



TURNING
INTO REALITY



Explorando Soluções de Roteamento IP em Redes OpenFlow

Christian Esteve Rothenberg, Ph.D.
Converged Networks Division (DRC)



Agenda



Brief Introduction



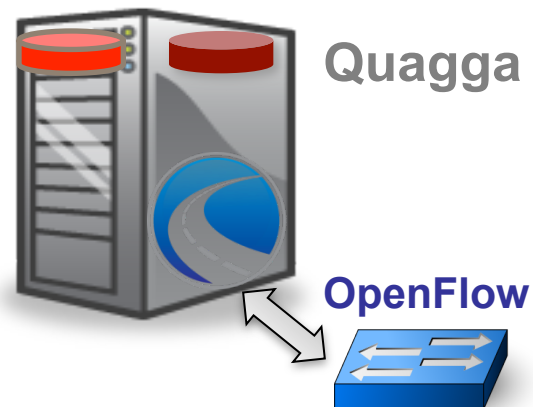
Basics of OpenFlow/SDN



RouteFlow

**Project
Architecture
Prototype
Demos
Use Cases**

Open discussion!





Brief Introduction

CPqD

- Major telecom R&D center in LATAM with expertise in various areas:
 - Optical (WDM, PON), Wireless (WiMax, LTE), IP (IMS/NGN, SDN/OpenFlow), OSS/BSS, Digital TV, SmartGrid/SmartCities
 - Today with ~1200 highly-skilled employees
 - Created in 1976 as R&D branch of Telebras - Brazilian telecom monopoly
 - Not-for-profit private foundation since 1998 after Telebras was privatized
- Purpose is to foster innovation to help (mainly) Brazilian companies and society
 - Focus on technology R&D
 - Bridge the gap between universities and the industry
- Near highly-ranked universities in Brazil
 - History of collaborations



CPqD – Network division

- ~200 researchers and engineers doing R&D
- Various product technologies transferred to the industry (mostly Brazilian companies) since privatization
- 37 patents filled in 2011
- 52 papers published in 2011
- Most successful spin-offs (and consumers of CPqD technology)
 - Tropic: created in 1999 with focus on NGN/IMS; US\$ 120M revenue in 2011, with growing presence in South America
 - Padtec: created in 2001 with focus on WDM; US\$ 150M revenue in 2011; WDM market leader in Brazil, with growing presence in South America and Europe



OpenFlow data plane developments at CPqD



OpenFlow 1.0 Switch (done 2010!)

- 24 x 10/100/1000
- 2 x 10Gb
- L2/L3
- ~2000 flow entries
- No protocol stack

OpenFlow 1.3 ROADM (ongoing)

- 5-degree WSS for mesh networks
- Contentionless-Directionless-Colorless
- Flexgrid
- Virtualization-capable
- Multicast-capable



OpenFlow 1.3 – 1a. no mundo

Search Mailing List Archives

Limit search to: Subject & Body Subject Author

Sort by: Reverse Sort

Limit to: All This Week Last Week This Month Last Month

Select Date Range through

[openflow-discuss] Announcing OpenFlow version 1.3 software switch

Christian Esteve Rothenberg [estevr at cpqd.com.br](mailto:estevr@cpqd.com.br)

Fri Nov 9 11:41:49 PST 2012

- Previous message: [\[openflow-discuss\] POX and cbench](#)
- Next message: [\[openflow-discuss\] Announcing OpenFlow version 1.3 software switch and controller prototypes](#)
- Messages sorted by: [\[date \]](#) [\[thread \]](#) [\[subject \]](#) [\[author \]](#)

Dear OpenFlow fellows,

giving continuity to the OpenFlow 1.X developments by Ericsson and CPqD (cf. v1.2 Tool-Kit [1]) we are glad to announce the release of a version 1.3 software switch [2] (based on Stanford reference design extended by Ericsson for v1.1) and a companion NOX OpenFlow 1.3 controller [3] (based on NOX Zaku and using oflib 1.3 of the software switch).



Forte candidata a se tornar a implementação de referência da ONF

Propósito da Apresentação

- Oferecer uma solução de software gratuita que pode
 - resolver seus problemas prontamente
 - ser usada como base para resolver seus problemas
- Ouvir de vocês, que projetam e operam redes diariamente
 - que problemas precisam resolver
 - se RouteFlow tem potencial para resolver esses problemas
 - se gostariam de usar RouteFlow
- Trabalhar com vocês para resolver esses problemas usando RouteFlow

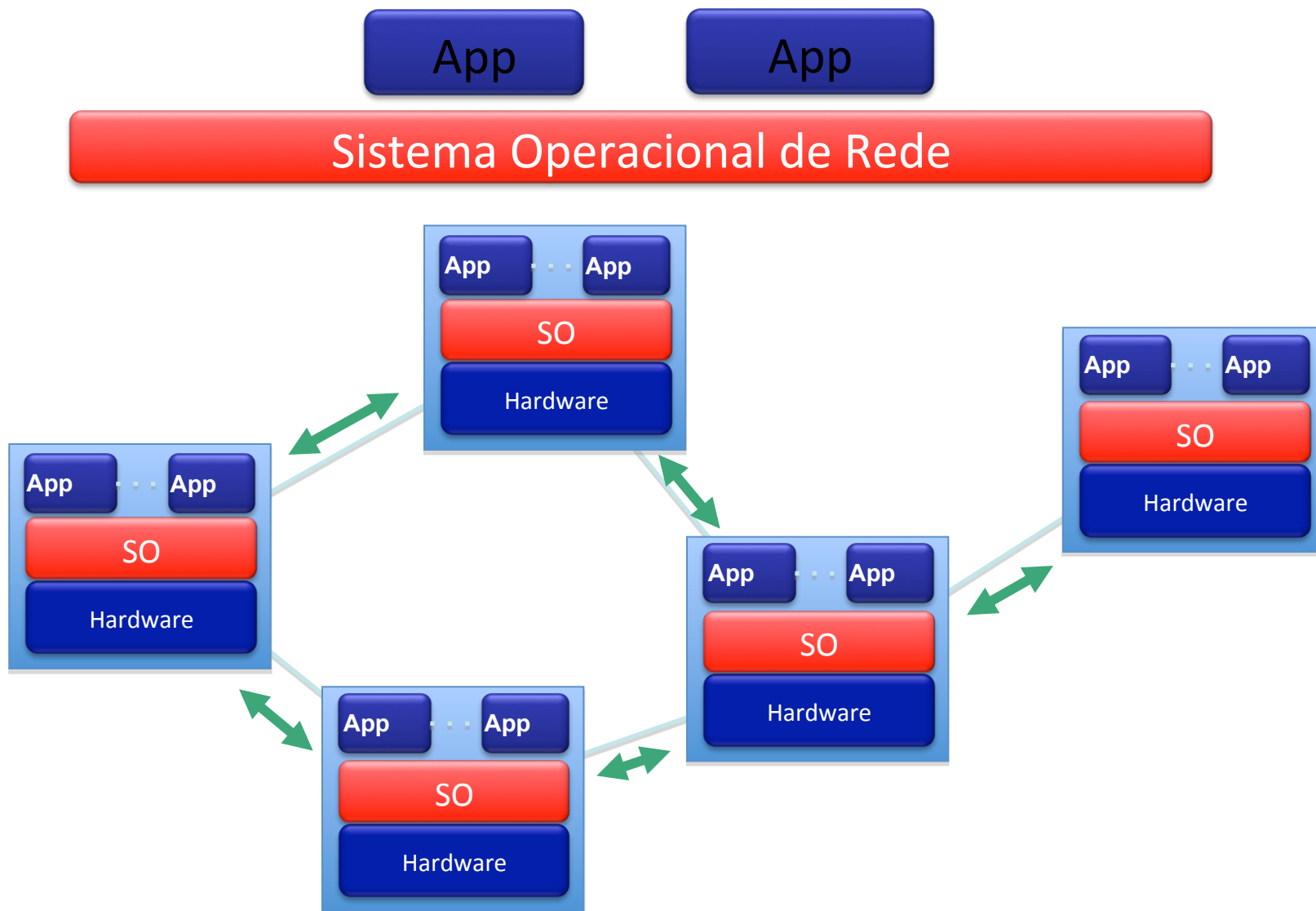


SDN/OpenFlow

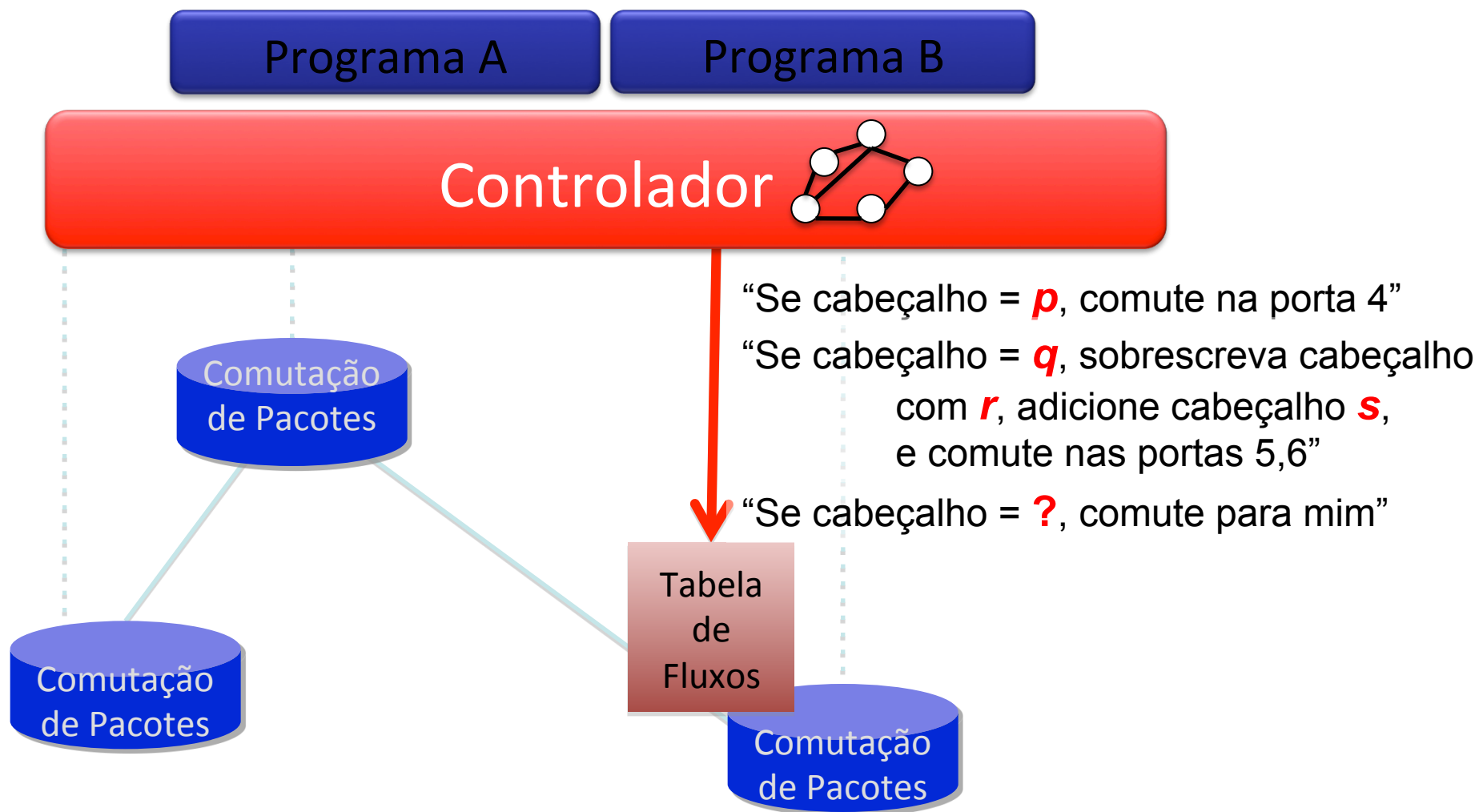
Brief Introduction

(slides from Nick Mckeown, U. Stanford, USA)

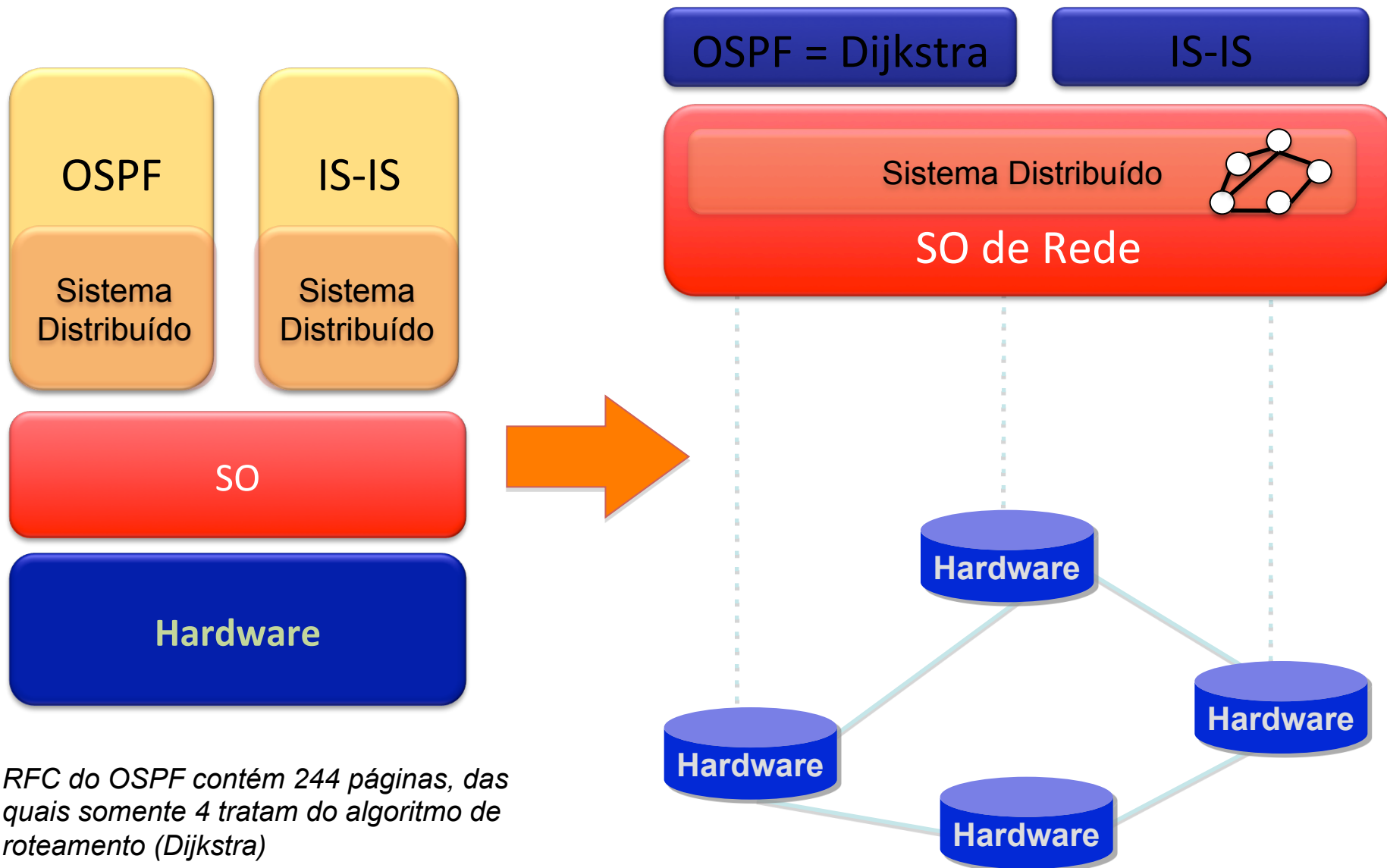
Rede Definida por Software (SDN)



OpenFlow: Exemplo básico

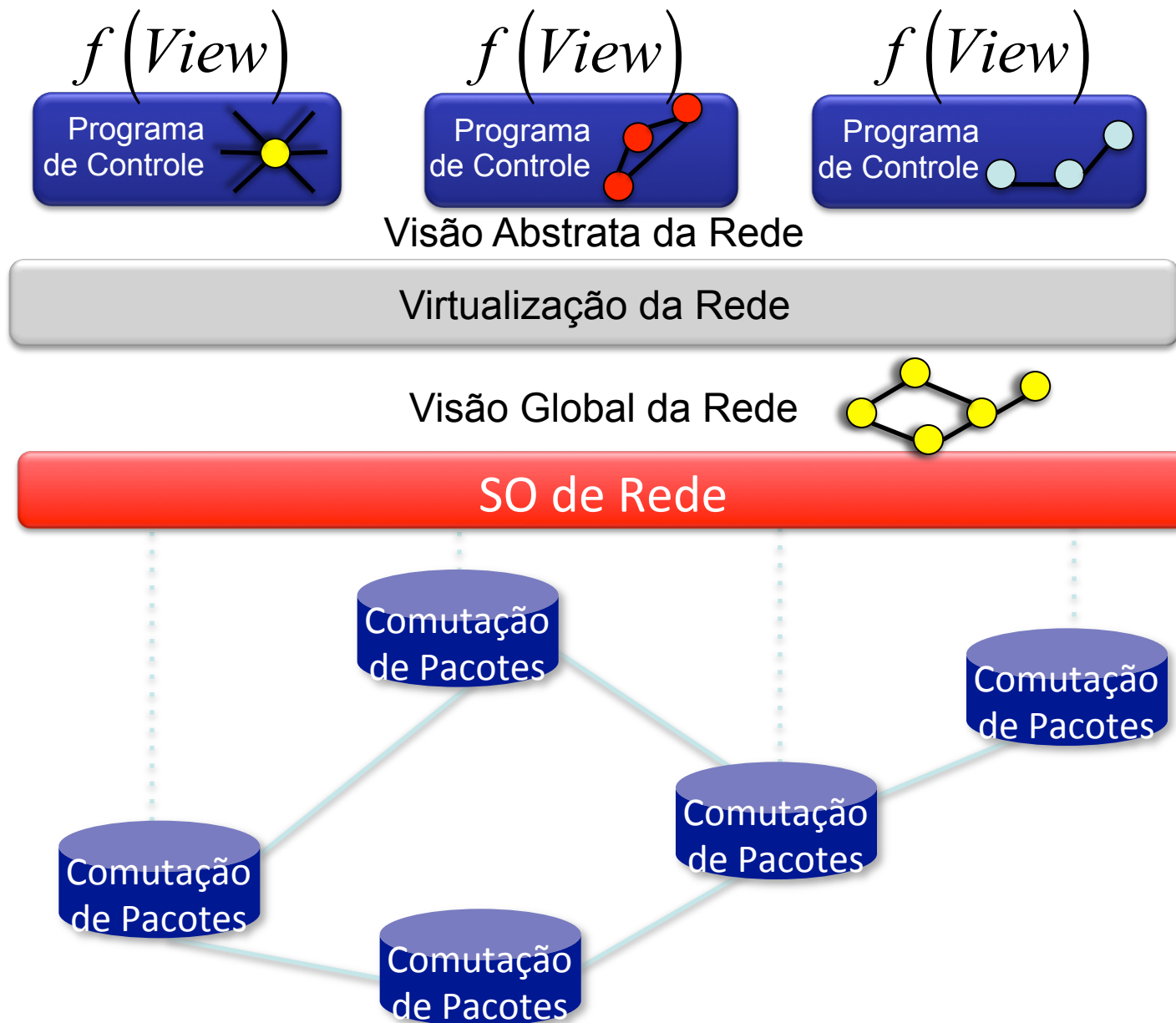


Exemplo: Roteamento Traditional x SDN



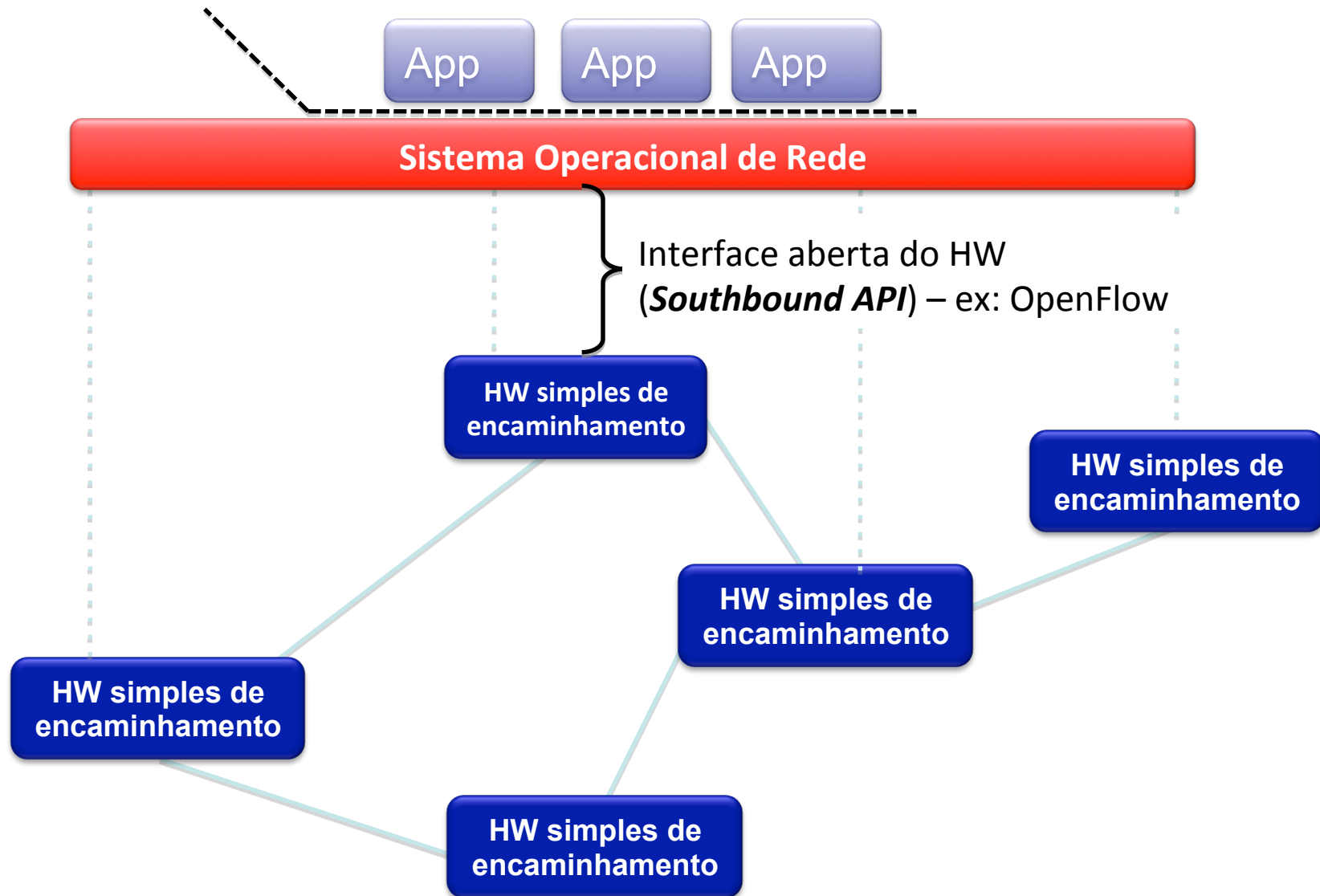
RFC do OSPF contém 244 páginas, das quais somente 4 tratam do algoritmo de roteamento (Dijkstra)

Visão atual de SDN: Programabilidade e Virtualização



Arquitetura SDN

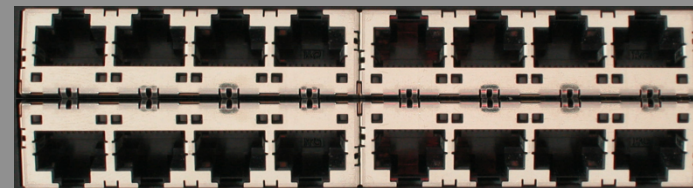
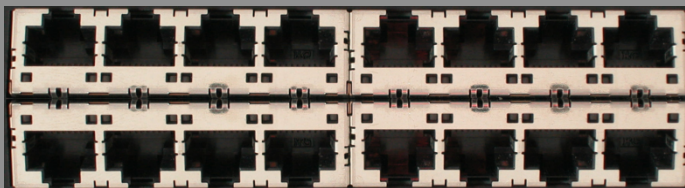
Interface aberta do SO de Rede (*Northbound API*)





How does OpenFlow work?

Ethernet Switch



OpenFlow Controller

OpenFlow Protocol (SSL/TCP)

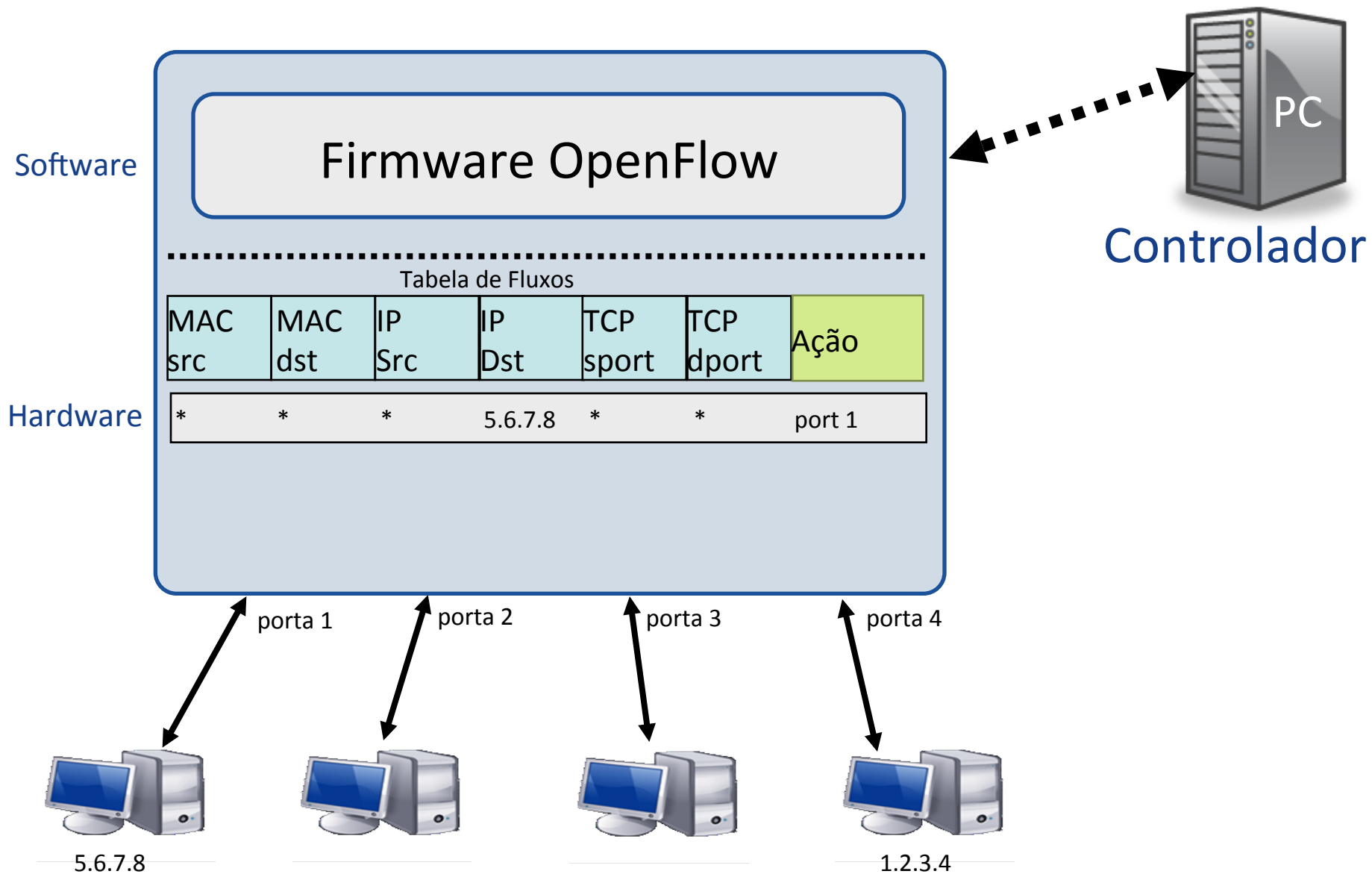


Control Path

OpenFlow

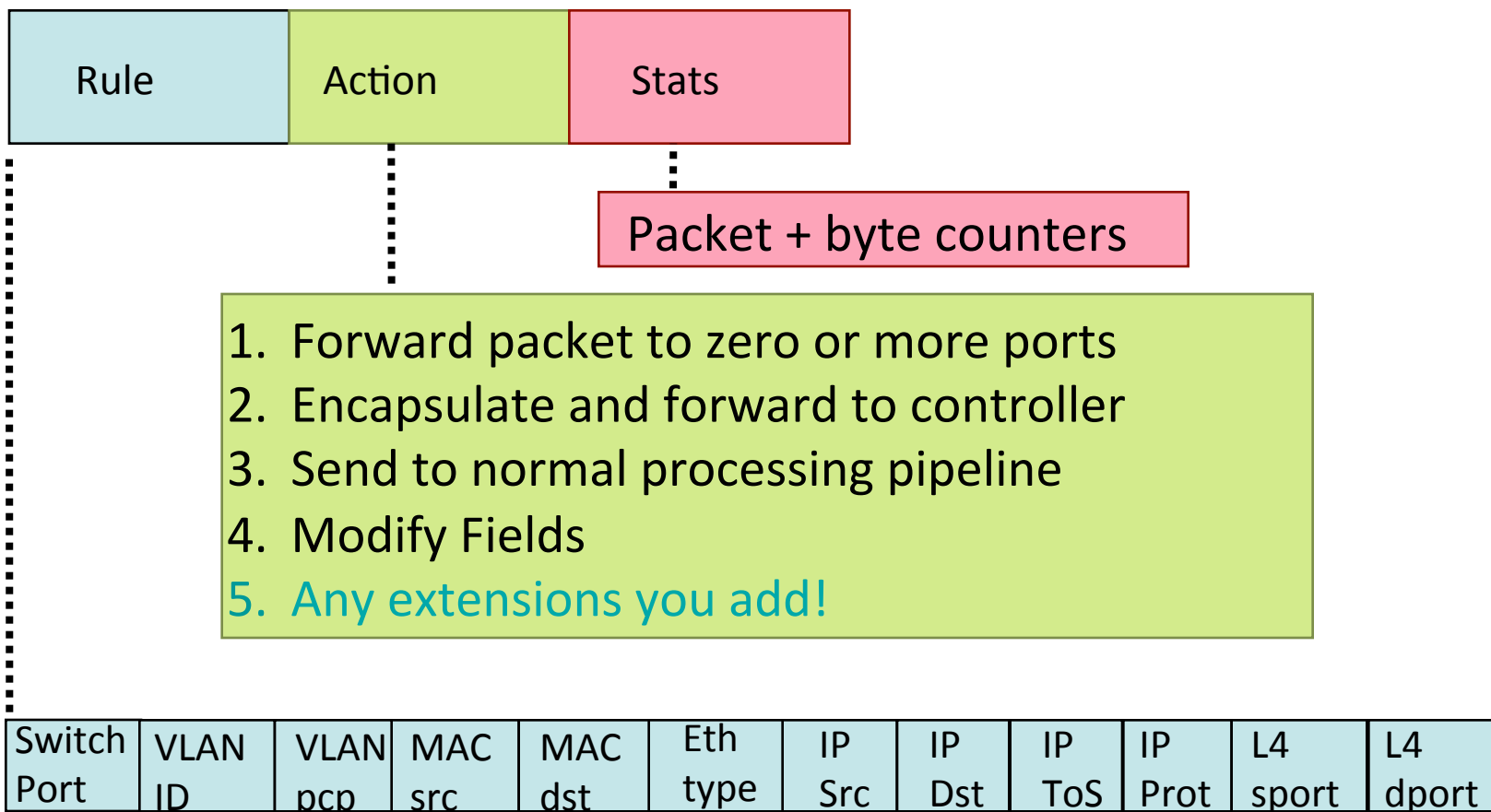
Data Path (Hardware)

OpenFlow: Interface de Programação do Hardware



OpenFlow Basics

Flow Table Entries



+ mask what fields to match

OpenFlow: Flexibilidade da abstração de fluxo

Comutação de quadros

Switch Port	MAC src	MAC dst	Eth type	VLAN ID	IP Src	IP Dst	IP Prot	TCP sport	TCP dport	Ação
*	*	00:1f:..	*	*	*	*	*	*	*	Porta 6

Comutação de fluxos

Switch Port	MAC src	MAC dst	Eth type	VLAN ID	IP Src	IP Dst	IP Prot	TCP sport	TCP dport	Ação
port3	00:20..	00:1f..	0800	vlan1	1.2.3.4	5.6.7.8	4	17264	80	Porta 6

Firewall

Switch Port	MAC src	MAC dst	Eth type	VLAN ID	IP Src	IP Dst	IP Prot	TCP sport	TCP dport	Ação
*	*	*	*	*	*	*	*	*	22	descartar

OpenFlow: Flexibilidade da abstração de fluxo

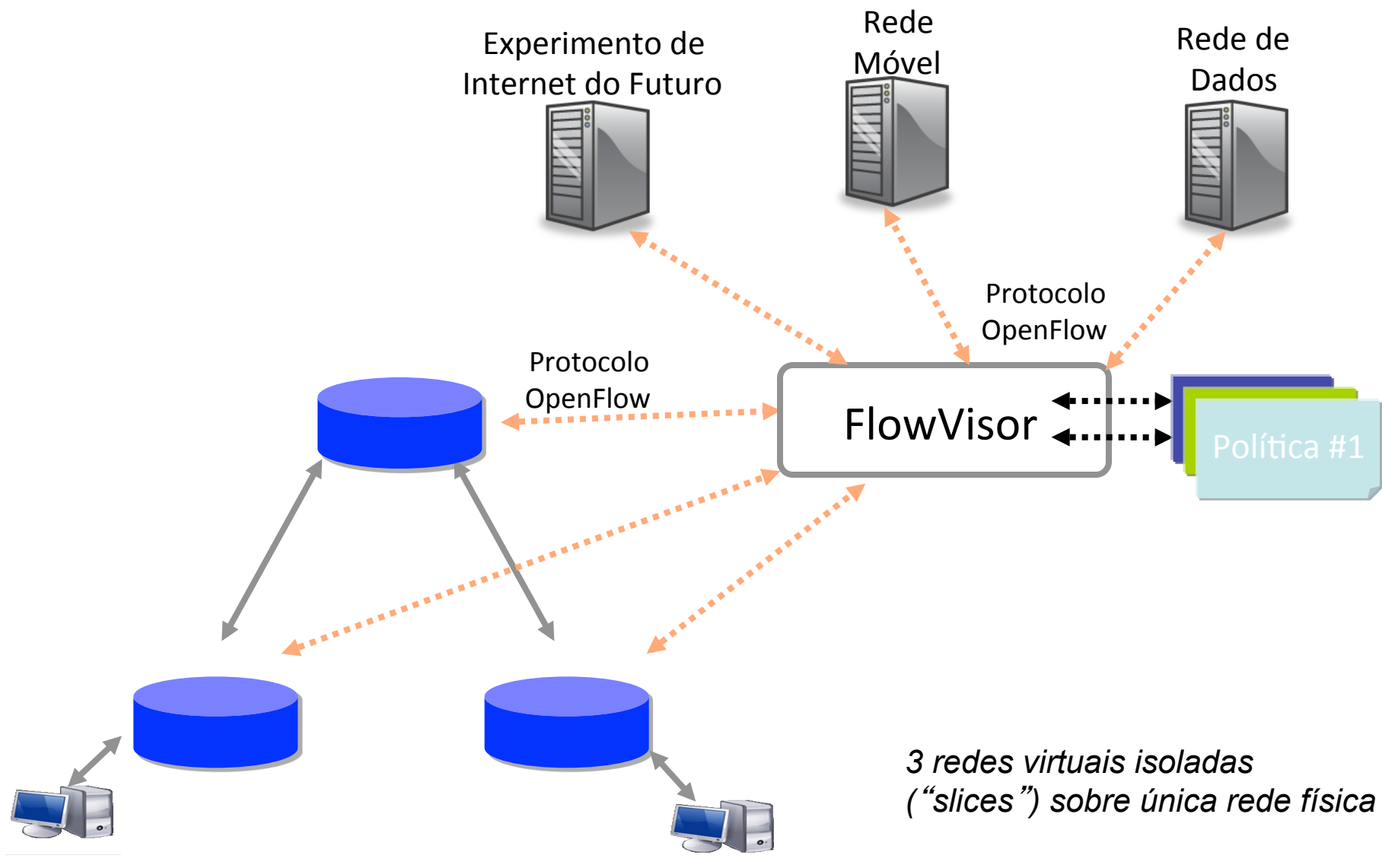
Roteamento

Switch Port	MAC src	MAC dst	Eth type	VLAN ID	IP Src	IP Dst	IP Prot	TCP sport	TCP dport	Ação
*	*	*	*	*	*	5.6.7.8	*	*	*	Porta 6

Comutação de VLAN

Switch Port	MAC src	MAC dst	Eth type	VLAN ID	IP Src	IP Dst	IP Prot	TCP sport	TCP dport	Ação
*	*	00:1f..	*	vlan1	*	*	*	*	*	Porta 6, Porta 7, Porta 9

FlowVisor: Virtualização OpenFlow



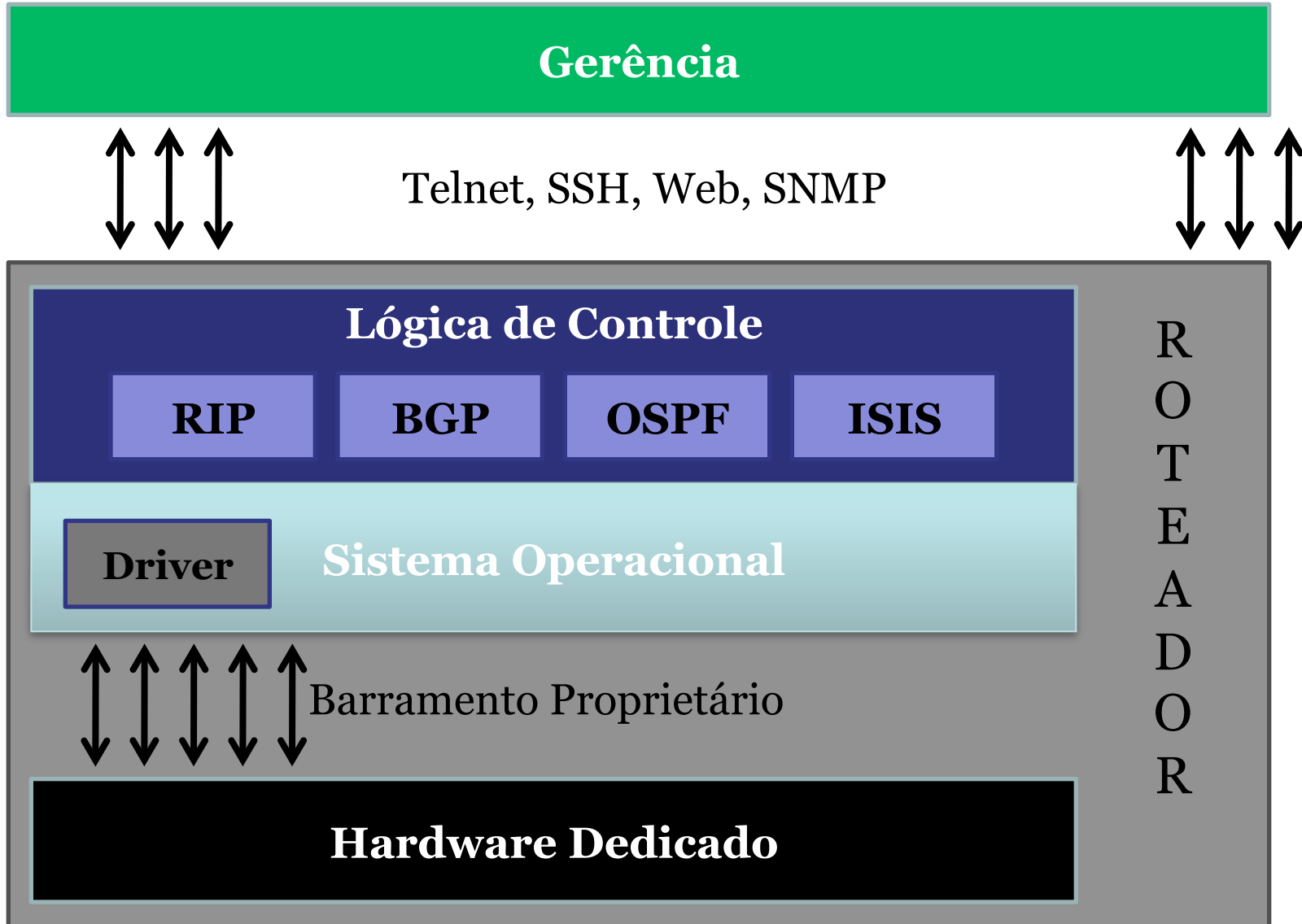
3 redes virtuais isoladas ("slices") sobre única rede física



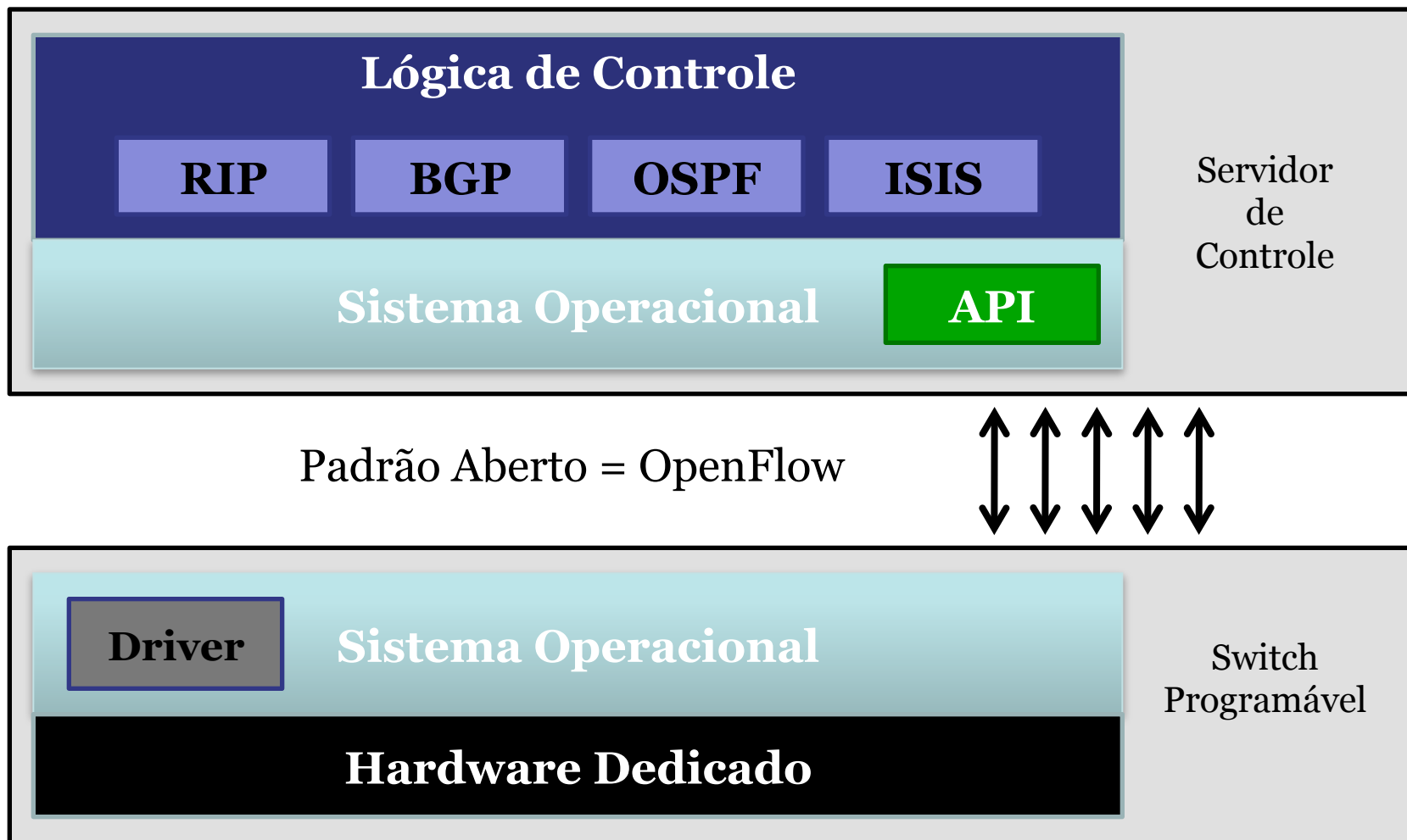
RouteFlow

(Transição de roteamento tradicional para roteamento ideal em SDN)

Modelo Atual: planos de controle e encaminhamento fortemente acoplados



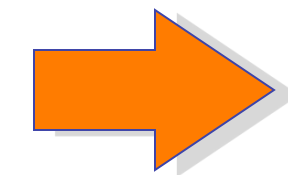
Modelo novo: planos de controle e encaminhamento separados por meio do protocolo OpenFlow



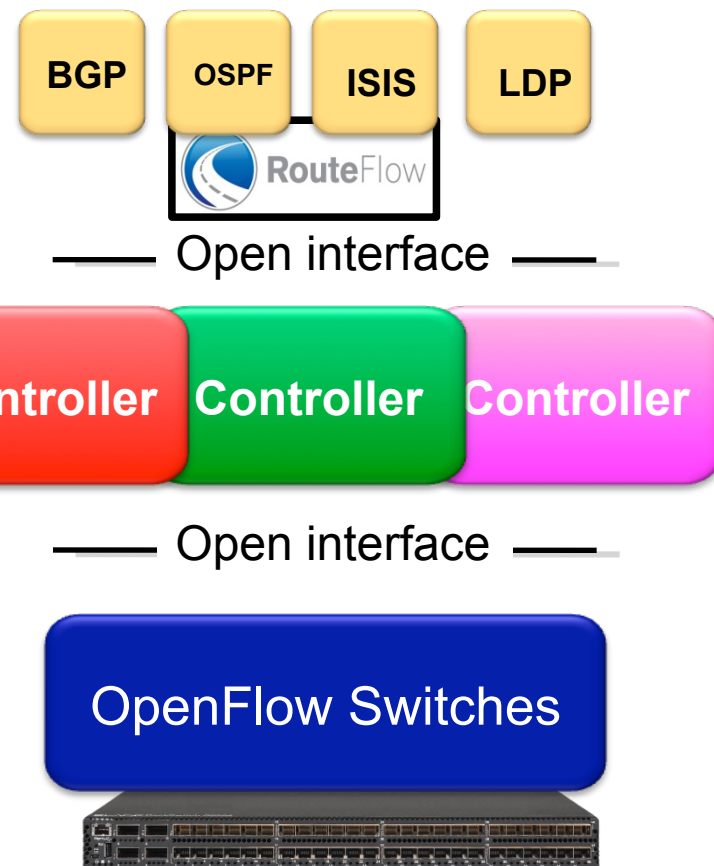
Software Defined IP Routing



High cost
Specialized config.
Closed source
Slow innovation

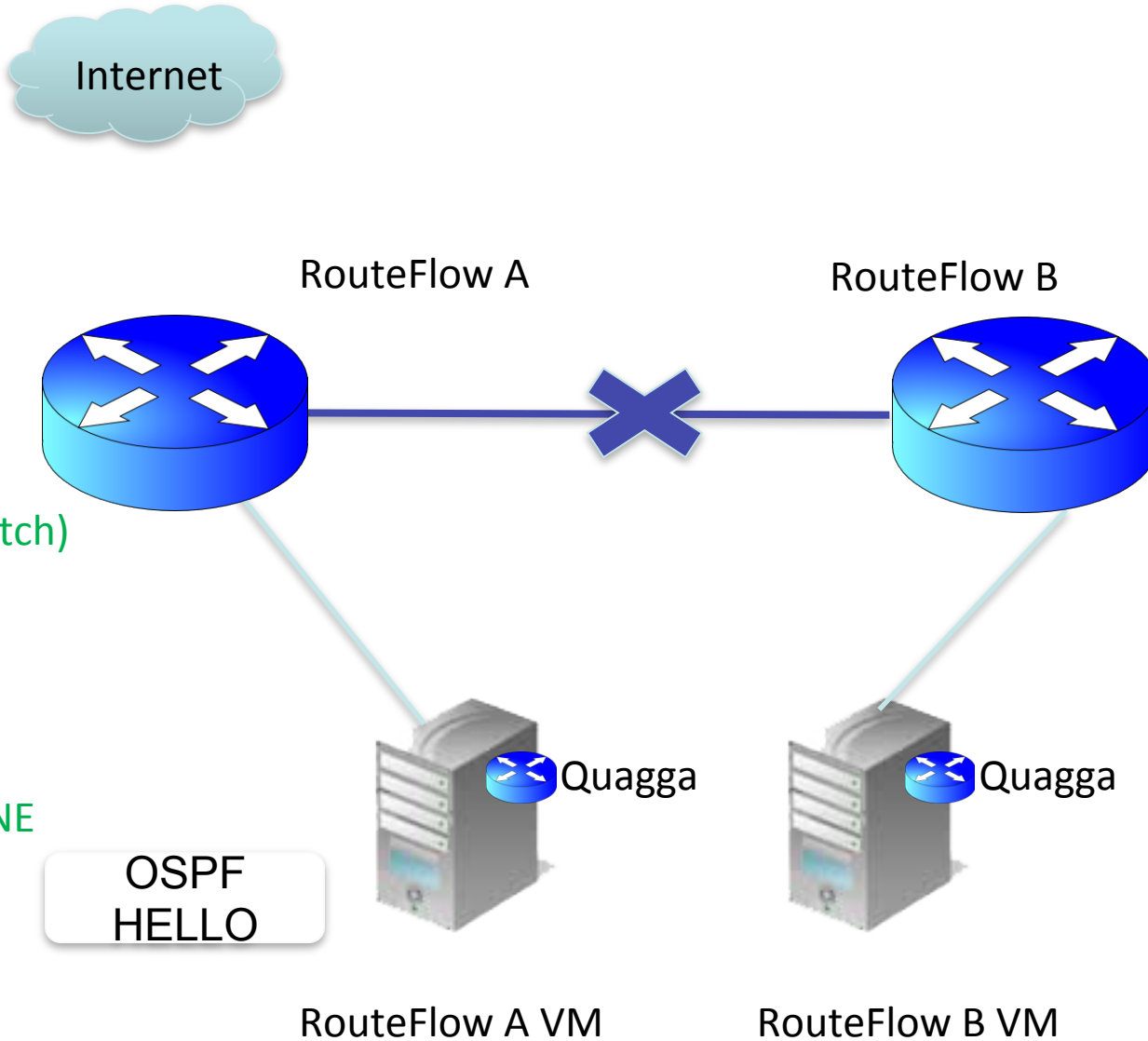


Source: McKeown

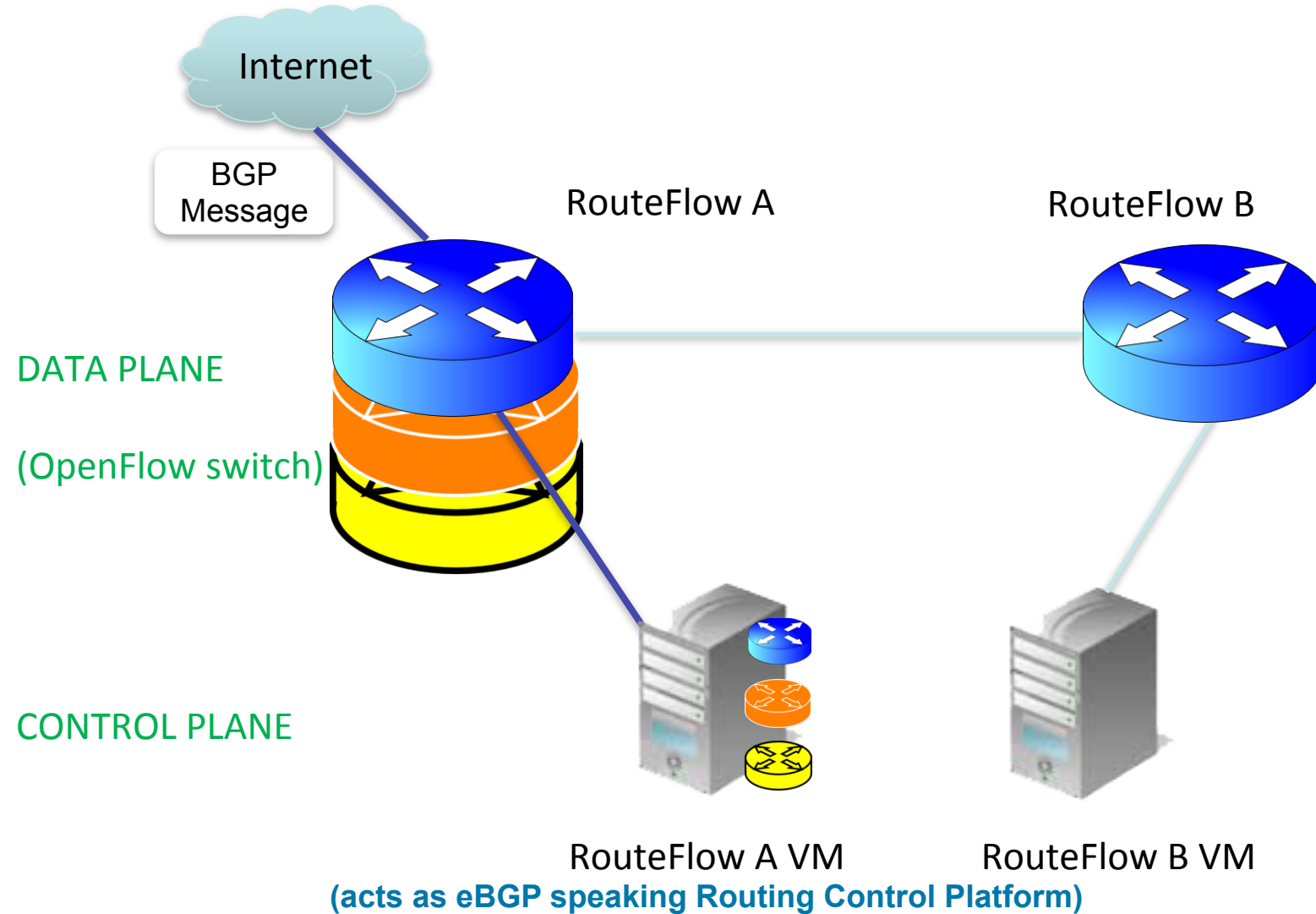


Low cost (commodity)
Multi-vendor modularity
Open source
Rapid innovation

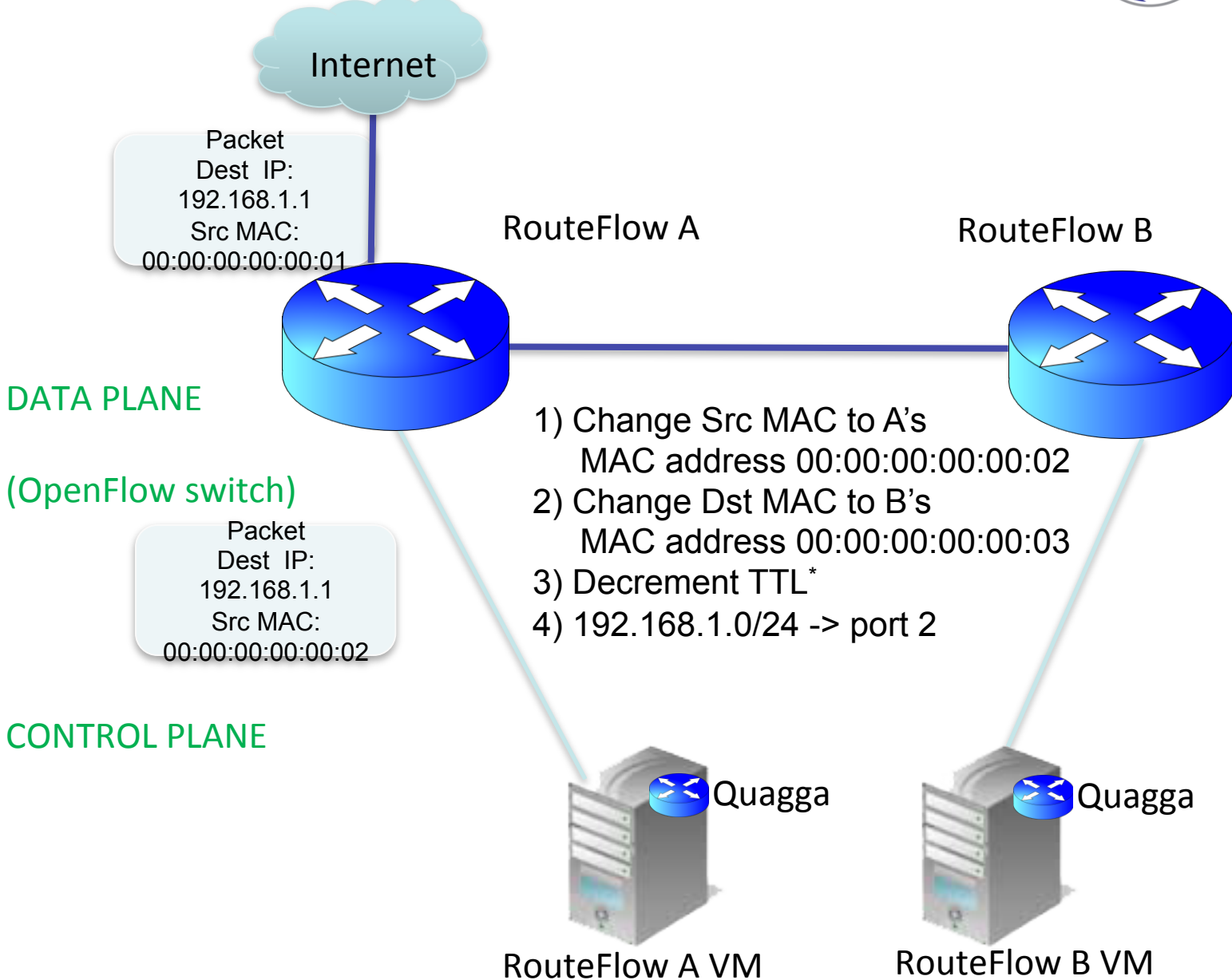
RouteFlow: Control Traffic (IGP)



RouteFlow: Control Traffic (BGP)



RouteFlow: Data Traffic



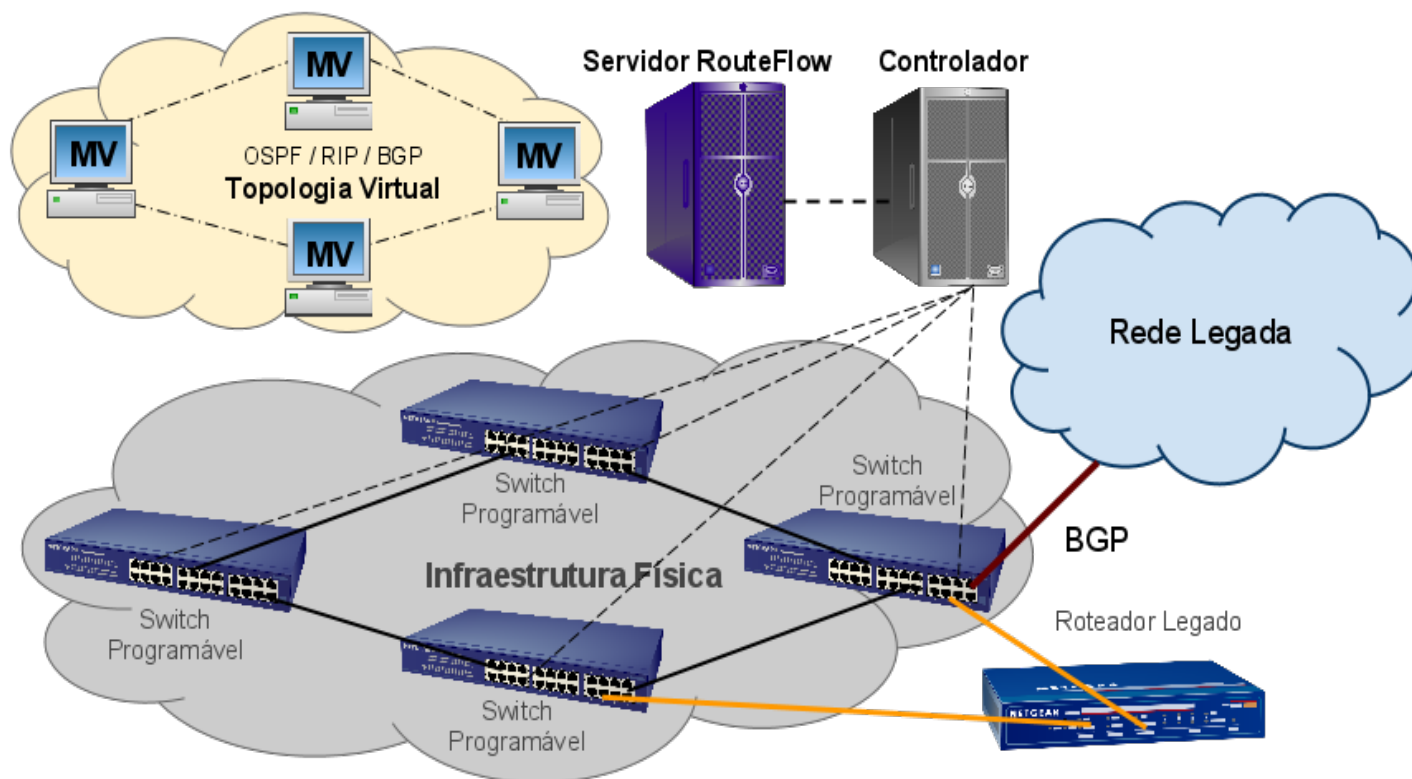
DATA PLANE

(OpenFlow switch)

CONTROL PLANE

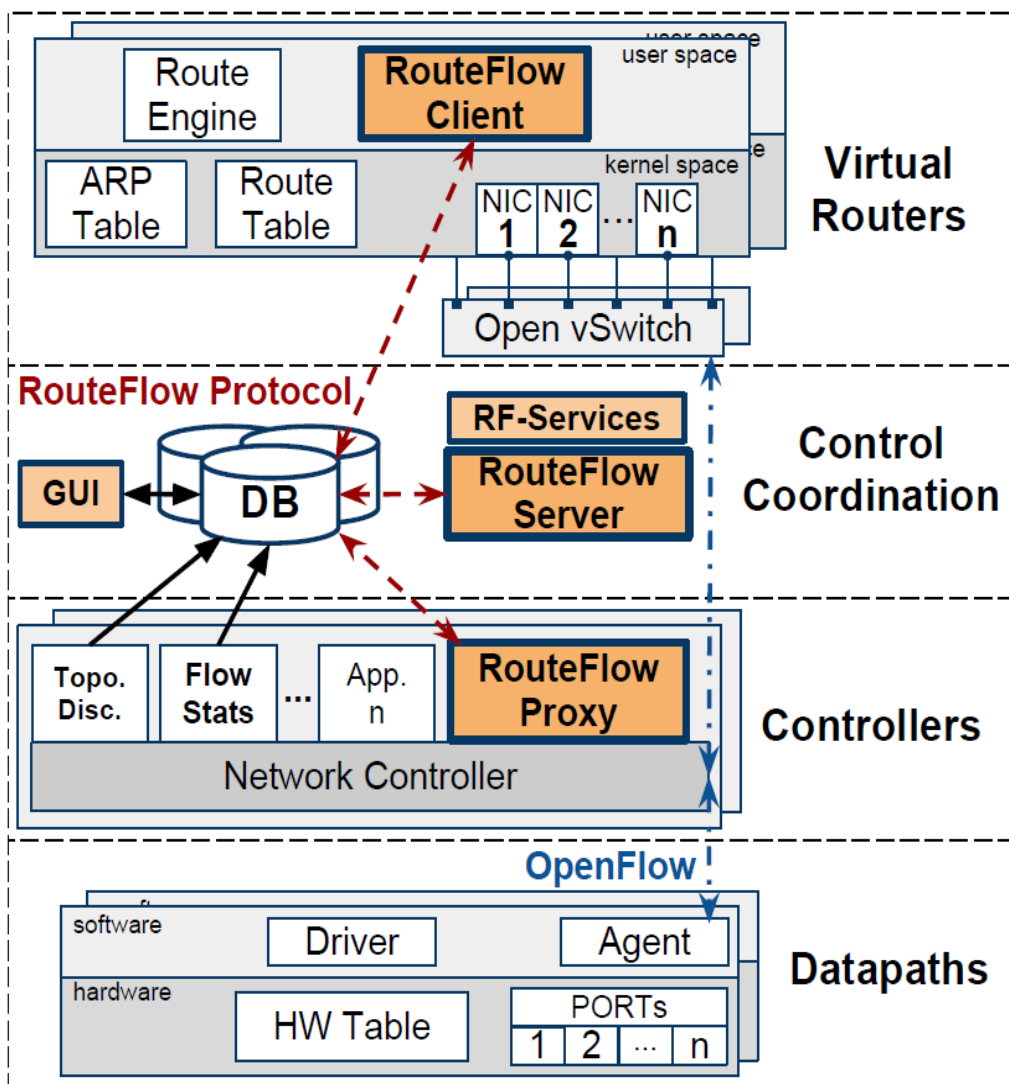


RouteFlow



1. Quagga on steroids
2. Low cost routing solutions
3. Innovative routing solutions
4. Transition path to OpenFlow/SDN

Arquitetura

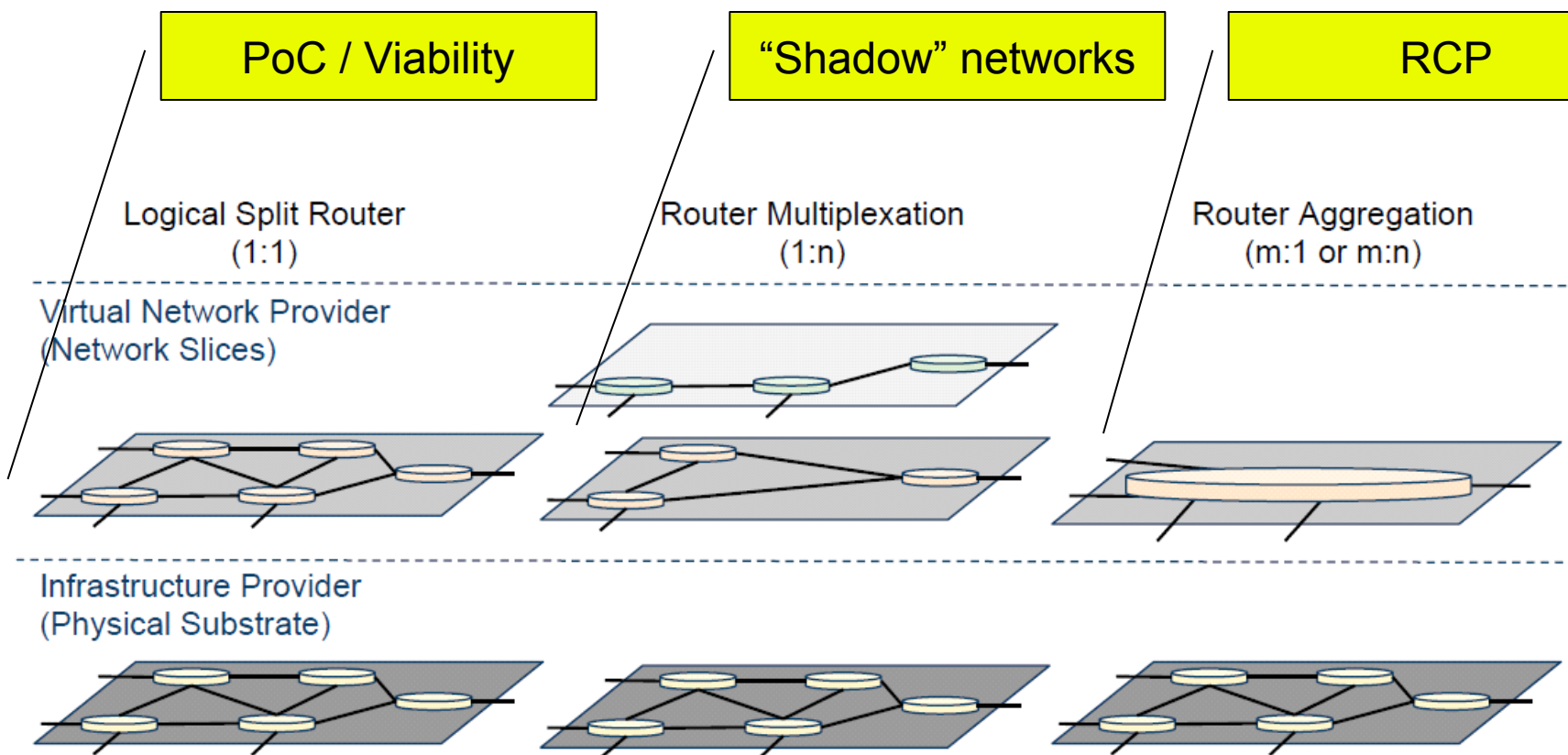


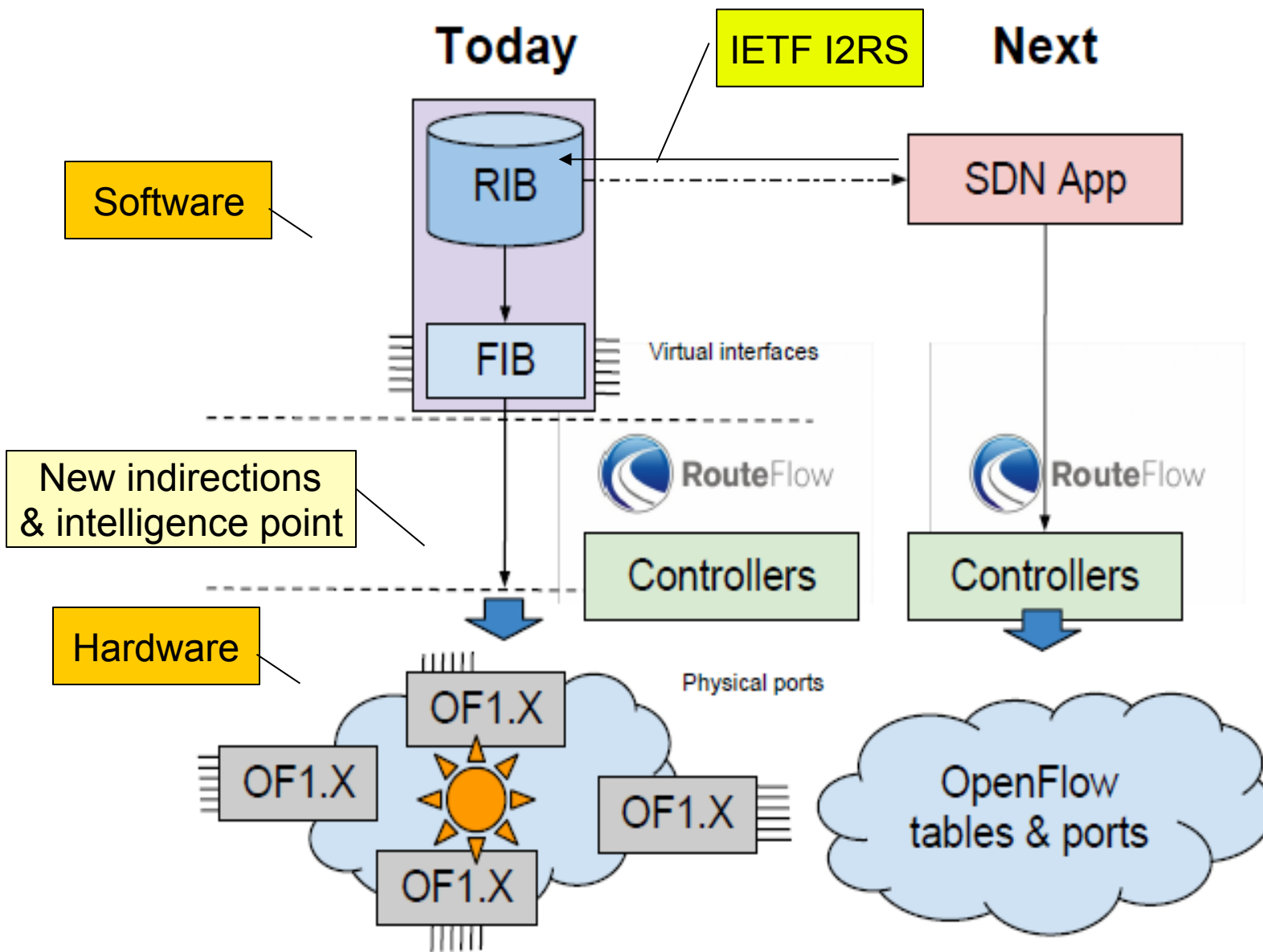
Key Features

- Modular architecture
 - RF-Proxy
 - RF-Server
 - RF-Client
- Database layer
 - JSON-based IPC
 - Resilient core state
 - Programmer-friendly
- Multi-Controller support
 - NOX, POX, Ryu, Floodlight,
 - Trema (ongoing)

Modes of operation

- From logical routers (akin VRFs) to single node abstractions over flexible virtual networks.
- New design choices on the distribution of the control nodes.



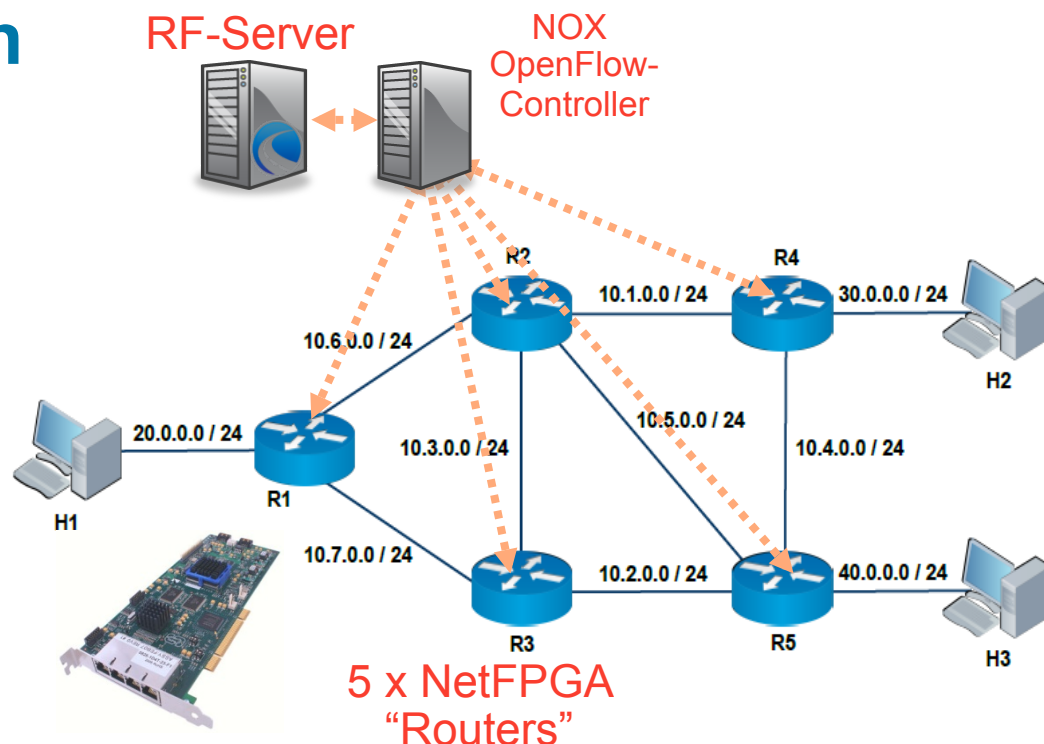


Prototype evaluation

- Back in 2010...

- Lab Setup

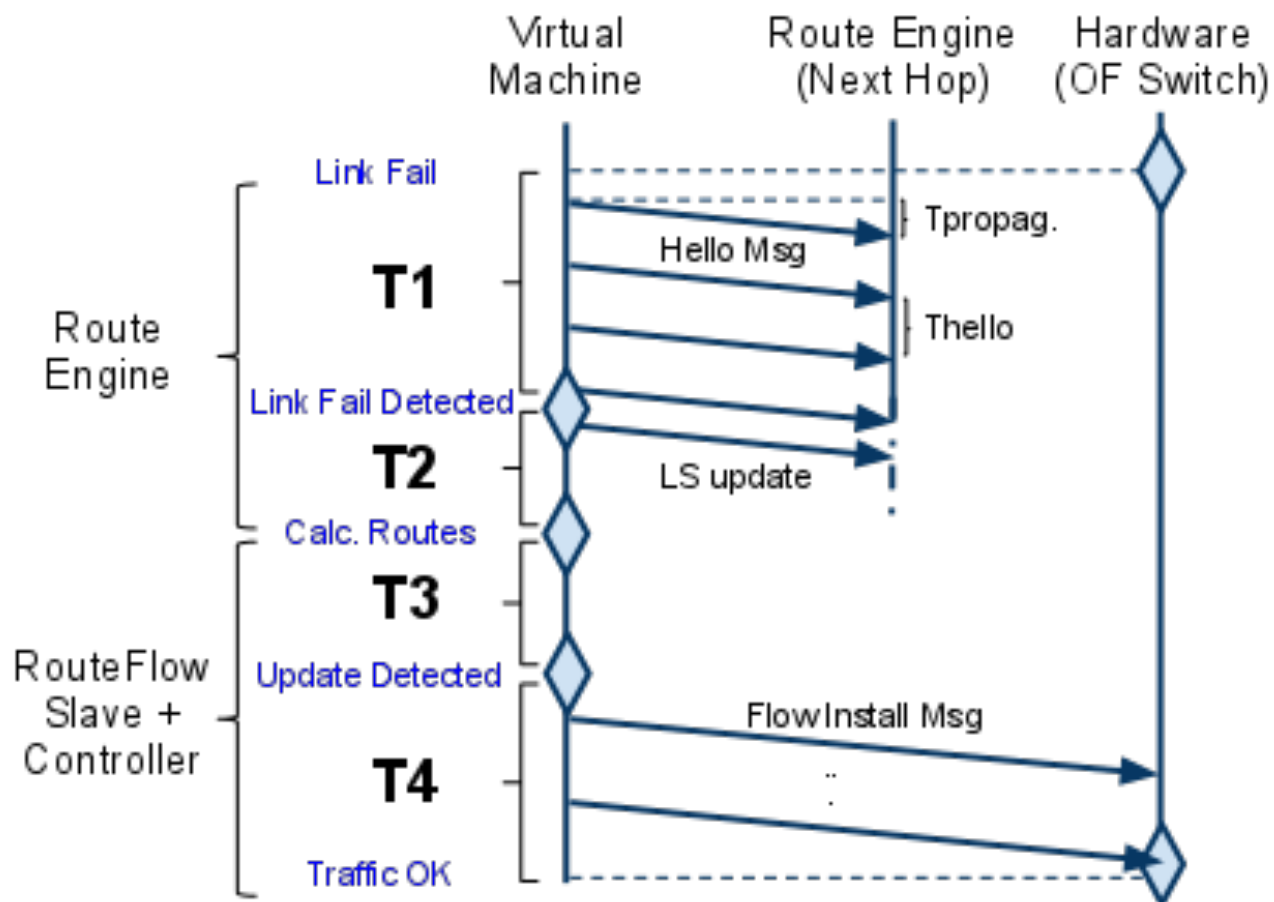
- NOX controller
- Quagga routing engine
- 5 x NetFPGAs switches



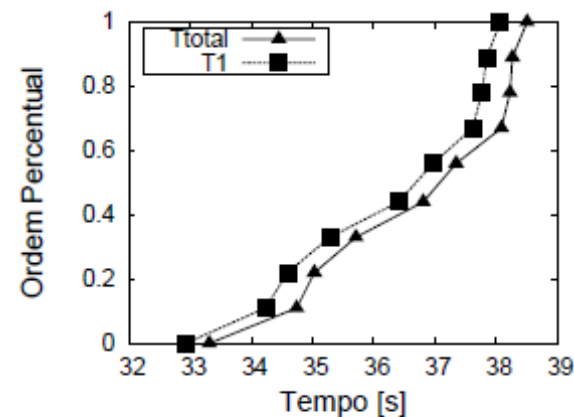
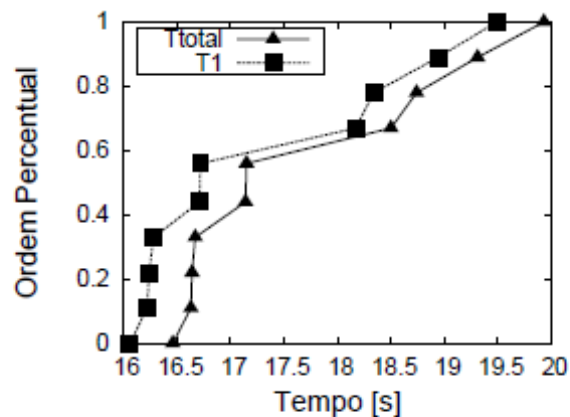
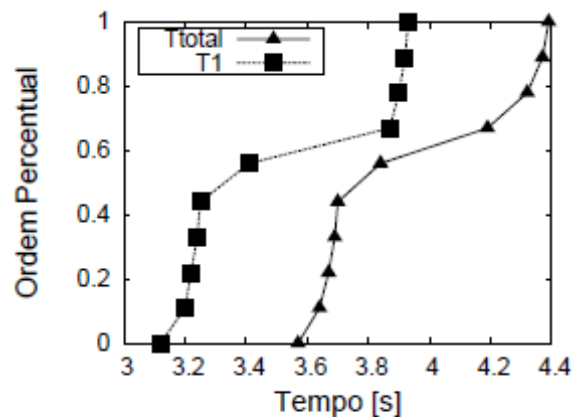
- Results

- Interoperability with traditional networking gear
- Route convergence time is dominated by the protocol time-out configuration (e.g., 4 x HELLO in OSPF) not by slow-path operations
- Compared to commercial routers (Cisco, Extreme, Broadcom-based), larger latency only for those packets that need to go to the slow-path:
 - Miss FIB entry, need processing by the OS networking / routing stack e.g., ARP, PING, routing protocol messages

Teste de Convergência com OSPF



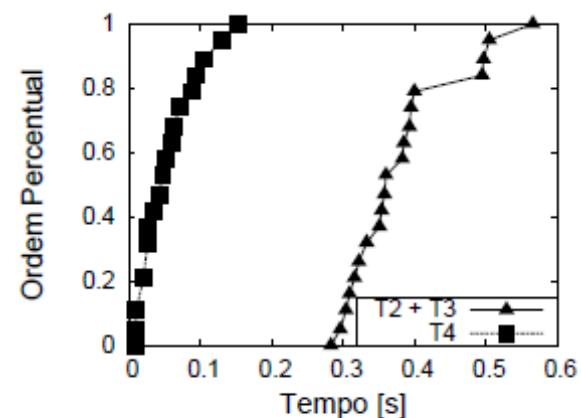
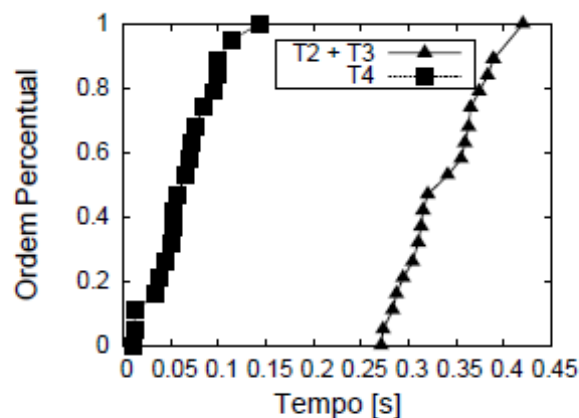
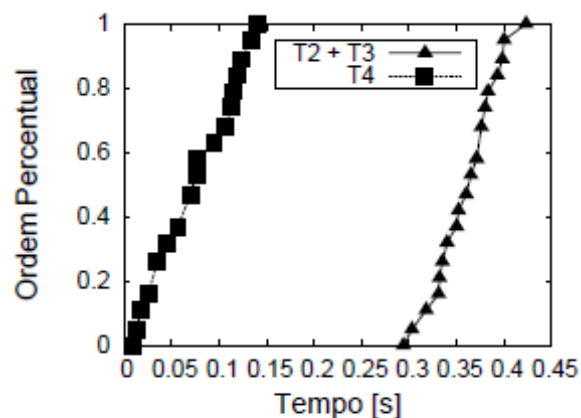
Teste de Convergência com OSPF



(a) T_{total} e T_1 para T_{Hello} de 1s.

(b) T_{total} e T_1 para T_{Hello} de 5s.

(c) T_{total} e T_1 para T_{Hello} de 10s.



(d) T_{2+3} e T_4 para T_{Hello} de 1s.

(e) T_{2+3} e T_4 para T_{Hello} de 5s.

(f) T_{2+3} e T_4 para T_{Hello} de 10s.

Teste de SlowPath e FastPath

Equipamento	Slow Path [ms]		Fast Path [ms]	
	Tmed.	T90%	Tmed.	T90%
CISCO 3560-e Catalyst	5,46	7,75	0,100	0,130
Extreme x450-e	11,30	14,00	0,106	0,141
CPqD Enterprise	14,20	17,30	0,101	0,147
RouteFlow*	22,00	28,00	0,082	0,119

- Cálculo do RTT com pacotes ICMP;
- Avaliação do overhead das mensagens direcionadas a uma pilha de controle remoto;
- Fast Path: encaminhamento em line rate;

*Newest RouteFlow design and NetFPGA

Demonstration at I Open Networking Summit (Stanford)

OPEN NETWORKING SUMMIT
 NOW PRE-REGISTERING FOR APRIL 2012 | STANFORD UNIVERSITY

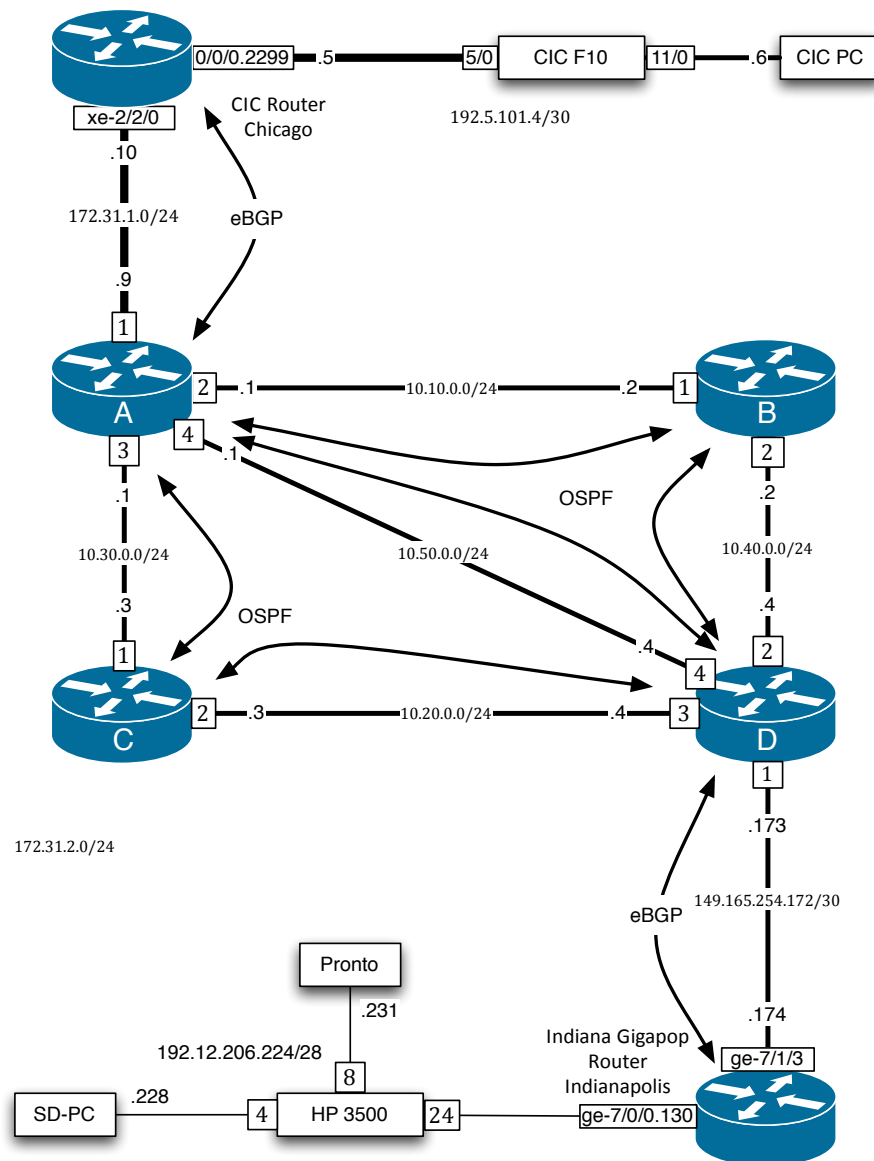
HOME | WHY SDN? | CONFERENCE | SPEAKERS | REGISTRATION | PAST CONFERENCES | HOTEL & VISITOR INFO | MEDIA | CONTACT



Field Trial at the University of Indiana

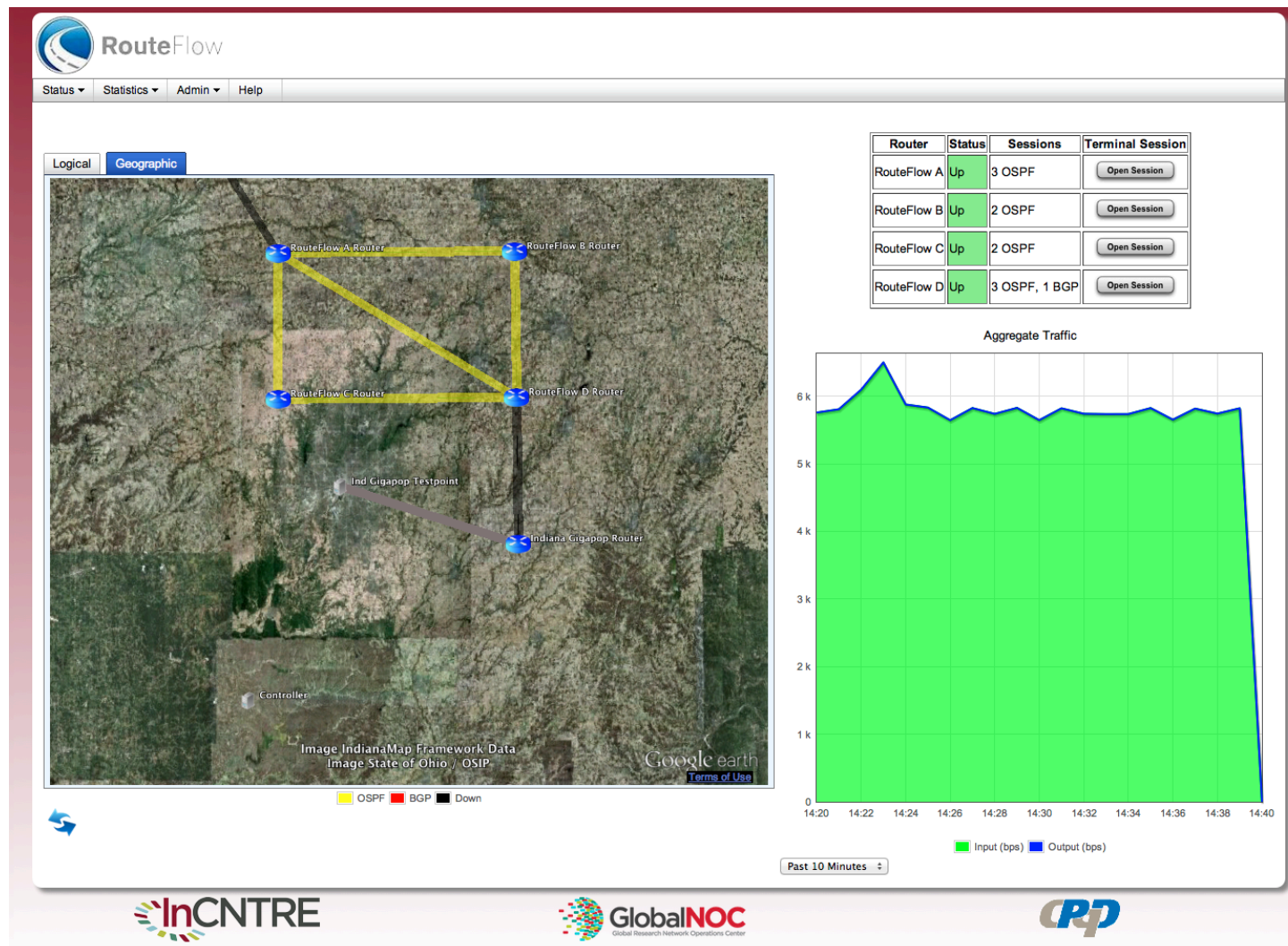
Network setup

- 1 physical OpenFlow switch
 - Pronto 3290
- 4 Virtual routers out of the physical OpenFlow switch
- 10 Gig and 1 Gig connections
- 2 BGP connections to external networks
 - Juniper routers in Chicago and Indianapolis
- Remote Controller
- New User Interface



Field Trial at the University of Indiana

User Interface



Compare interfaces over the last 30 years

“PC” user interfaces

```

Current date is Tue 1-01-1980
Enter new date:
Current time is 7:40:27.13
Enter new time:

The IBM Personal Computer BIOS
Version 1.10 (C)Copyright IBM Corp 1981, 1982

ROM BIOS
DISKBOOT  CDM  FDISKBT  CDM  CHECKSD  CDM  SYS  CDM  DISKCOPY  CDM
DISKCOPY  CDM  CDM  CDM  EXEC2BIN  EXE  PASTE  CDM  EXLIM  CDM
DEBUG  CDM  LINK  EXE  INVIC  CDM  INVICR  CDM  ANT  BAS
SAMPLES  BAS  PARTAGE  BAS  CALDRMR  BAS  CALDRMR  BAS  MUSIC  BAS
FORMATY  BAS  CIRCLE  BAS  PITCHWT  BAS  SPACE  BAS  DWL  BAS
CDMS  BAS
      26 File(s)
AMIR command.com
COMMAND  CDM  4950  5-07-82  12:00
      1 File(s)
AP:
  
```



Network user interfaces

```

Router>en
Router#service-module gi/0 session
Trying 172.25.25.25, 2066 ... Open

Switch>en
Switch#conf t
Enter configuration commands, one per line. End with CNTL/Z.
Switch(config)#int fas
Switch(config)#int fastethernet 1/0/1
Switch(config-1f)#ip address 10.10.10.2 255.255.255.0
Switch(config-1f)#no shut
Switch(config-1f)#end
Switch#
Building configuration...
[OK]
01:09:35: %SYS-5-CONFIG_I: Configured from console by console
Switch#
  
```

```

Router>en
Router#service-module gi/0 session
Trying 172.25.25.25, 2066 ... Open

Switch>en
Switch#conf t
Enter configuration commands, one per line. End with CNTL/Z.
Switch(config)#int fas
Switch(config)#int fastethernet 1/0/1
Switch(config-1f)#ip address 10.10.10.2 255.255.255.0
Switch(config-1f)#no shut
Switch(config-1f)#end
Switch#
Building configuration...
[OK]
01:09:35: %SYS-5-CONFIG_I: Configured from console by console
Switch#
  
```

Demystifying Configuration Challenges and Tradoffs in Network Based ISP Services (Benson, Akella, Shaikh SIGCOMM 2011)

Advancing the RouteFlow GUI...

The screenshot shows the RouteFlow GUI interface. On the left, there is a network topology diagram with nodes labeled 'switch8', 'switch9', and 'RouteFlow Server'. On the right, a detailed view for 'switch8' is shown, including its description (Nokia Networks, Inc. Open vSwitch), aggregated statistics (438 packets, 36704 bytes), and a table of matches and actions.

#	Matches	Actions	Packets	Bytes
0	ip_dst: 96.10.23.0e16:0d; net_dst: 40.0.0.2	SET_DL_DST: 02 ba ba ba ba; SET_DL_SRC: 96.10.23.0e16:0d; OUTPUT: port 2;	0	0
1	ip_dst: 96.10.23.0e16:0d; net_dst: 172.31.4.100	SET_DL_DST: 8a 7a 02 ea 08 7a; SET_DL_SRC: 96.10.23.0e16:0d; OUTPUT: port 1;	0	0
2	ip_dst: 96.10.23.0e16:0d; net_dst: 20.0.0.3	SET_DL_DST: 7a 0c 8f 6a 5a 70; SET_DL_SRC: 96.10.23.0e16:0d; OUTPUT: port 3;	0	0
3	ip_dst: 96.10.23.0e16:0d; net_dst: 90.0.0.1	SET_DL_DST: f2 34 79 ab 7d 09; SET_DL_SRC: 96.10.23.0e16:0d; OUTPUT: port 4;	0	0
4	read: ip_src: 0; ip_dst: 0	OUTPUT: port 65535;	403	33722
5	ip_dst: 96.10.23.0e16:0d; net_dst: 172.31.1.0/24	SET_DL_DST: 02 34 79 ab 7d 09; SET_DL_SRC: 96.10.23.0e16:0d; OUTPUT: port 4;	2	196
6	ip_dst: 96.10.23.0e16:0d; net_dst: 10.0.0.0/24	SET_DL_DST: 02 ba ba ba ba; SET_DL_SRC: 96.10.23.0e16:0d; OUTPUT: port 2;	0	0
7	ip_dst: 96.10.23.0e16:0d; net_dst: 30.0.0.0/24	SET_DL_DST: 7a 0c 8f 6a 5a 70; SET_DL_SRC: 96.10.23.0e16:0d; OUTPUT: port 3;	0	0
8	ip_dst: 96.10.23.0e16:0d; net_dst: 172.31.3.0/24	SET_DL_DST: 7a 0c 8f 6a 5a 70; SET_DL_SRC: 96.10.23.0e16:0d; OUTPUT: port 3;	2	196
9	ip_dst: 96.10.23.0e16:0d; net_dst: 172.31.2.0/24	SET_DL_DST: 02 ba ba ba ba; SET_DL_SRC: 96.10.23.0e16:0d; OUTPUT: port 2;	19	1862

Topology and Statistics

The screenshot shows the RouteFlow GUI interface displaying a table of VM associations. The table has columns for VM ID, VM port, VS ID, VS port, DP ID, and DP port. A yellow callout box provides a legend for the table's content.

VM ID	VM port	VS ID	VS port	DP ID	DP port
0x00E1E7B8789660	1	0x000002320844ACE	3	0x0000000000000007	1
0x00E1E7B8789660	2	0x000002320844ACE	13	0x0000000000000007	2
0x00E1E7B8789660	3	0x000002320844ACE	12	0x0000000000000007	2
0x0003F707E429D	1	0x000002320844ACE	9	0x0000000000000008	1
0x0003F707E429D	2	0x000002320844ACE	1	0x0000000000000008	2
0x0003F707E429D	3	0x000002320844ACE	8	0x0000000000000008	3
0x0003F707E429D	4	0x000002320844ACE	7	0x0000000000000008	4
0x000136E72E1B334	1	0x000002320844ACE	10	0x0000000000000005	1
0x000136E72E1B334	2	0x000002320844ACE	2	0x0000000000000005	2
0x000136E72E1B334	3	0x000002320844ACE	4	0x0000000000000005	3
0x000136E72E1B334	4	0x000002320844ACE	5	0x0000000000000005	4
0x0007C3C33597DF3	1	0x000002320844ACE	14	0x0000000000000006	1
0x0007C3C33597DF3	2	0x000002320844ACE	11	0x0000000000000006	2
0x0007C3C33597DF3	3	0x000002320844ACE	6	0x0000000000000006	3

The table shows the current associations of VMs and datapaths in the RouteFlow network.

- A row containing with all columns filled means an active entry.
- A row with only a VM ID represents a registered, idle and never used VM.
- A row with only a DP ID represents a registered, idle and never associated datapath.
- A row with only a VM ID and DP ID represents an association that is inactive. The reason could be an offline datapath.

Resource Status and Mapping

The screenshot shows the RouteFlow GUI interface with a message filter applied. The filter is set to 'Slave -> Server' and 'VMRegisterRequest'. The message list shows various messages between a slave and a server.

Filters	Slave -> Server	Server -> Controller
By message type:		
Slave -> Server		
<input type="checkbox"/> VMRegisterRequest		
<input checked="" type="checkbox"/> VMRegisterResponse		
<input type="checkbox"/> VMConfig		
RouterInfo		
<input type="checkbox"/> VMConfig		
Server -> Controller		
<input type="checkbox"/> FlowAdd		
<input type="checkbox"/> DatapathConfig		
<input type="checkbox"/> DatapathJoin		
<input type="checkbox"/> DatapathLeave		
<input type="checkbox"/> VMMap		

To	Status	Type
0x0003F707E429D read VMConfig	if-server read	DatapathJoin
0x00E1E7B8789660 read VMConfig	if-server read	DatapathJoin
0x0007C3C33597DF3 read VMConfig	if-server read	DatapathJoin
0x000136E72E1B334 read VMConfig	ip_id: 0x0000000000000000	
0x0003F707E429D read VMRegisterResponse	is_flow: false	
0x0007C3C33597DF3 read VMRegisterRequest	if-server read	DatapathJoin
0x00E1E7B8789660 read VMRegisterResponse	if-server read	DatapathJoin
if-server read VMRegisterRequest		
if-server read VMRegisterRequest		
if-server read VMRegisterRequest		
if-server read VMRegisterRequest		

RouteFlow Protocol

The screenshot shows the OpenNMS GUI interface displaying resource graphs and results. The graphs show traffic volume over time for RPKETS, TPCKETS, and RBYTES. The results section shows a list of messages.

Resource Graph Results
User: amir@opennms.org | Log out
Mar 27, 2012 16:20 EDT

Home | Reports | Resource Graphs | Results

Time period: [Last Day] | From: Mon Mar 26 16:20:38 EDT 2012 | To: Tue Mar 27 16:20:38 EDT 2012

Node: localhost
OpenFlow Stats: s5-eth-6 (Index 1.4)

RPKETS
Attribute [packets/s] | Avg: 1.30 | Min: 1.00 | Max: 3.95

TPCKETS
Attribute [packets/s] | Avg: 1.34 | Min: 1.33 | Max: 1.95

RBYTES
Attribute [packets/s] | Avg: 1.34 | Min: 1.33 | Max: 1.95

OpenNMS SNMP

Demonstration at Supercomputing 11

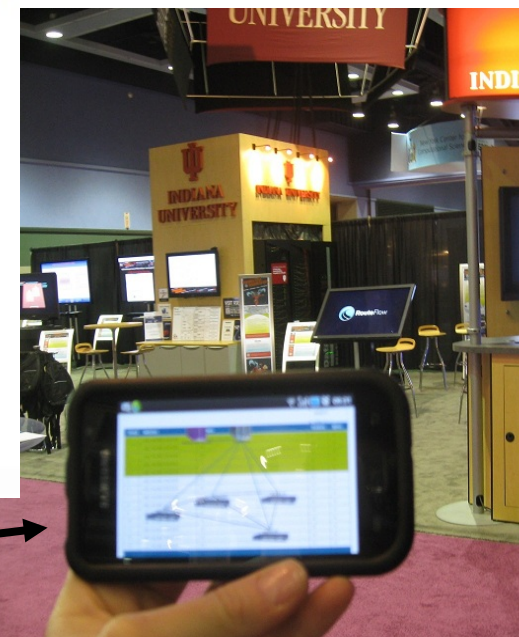
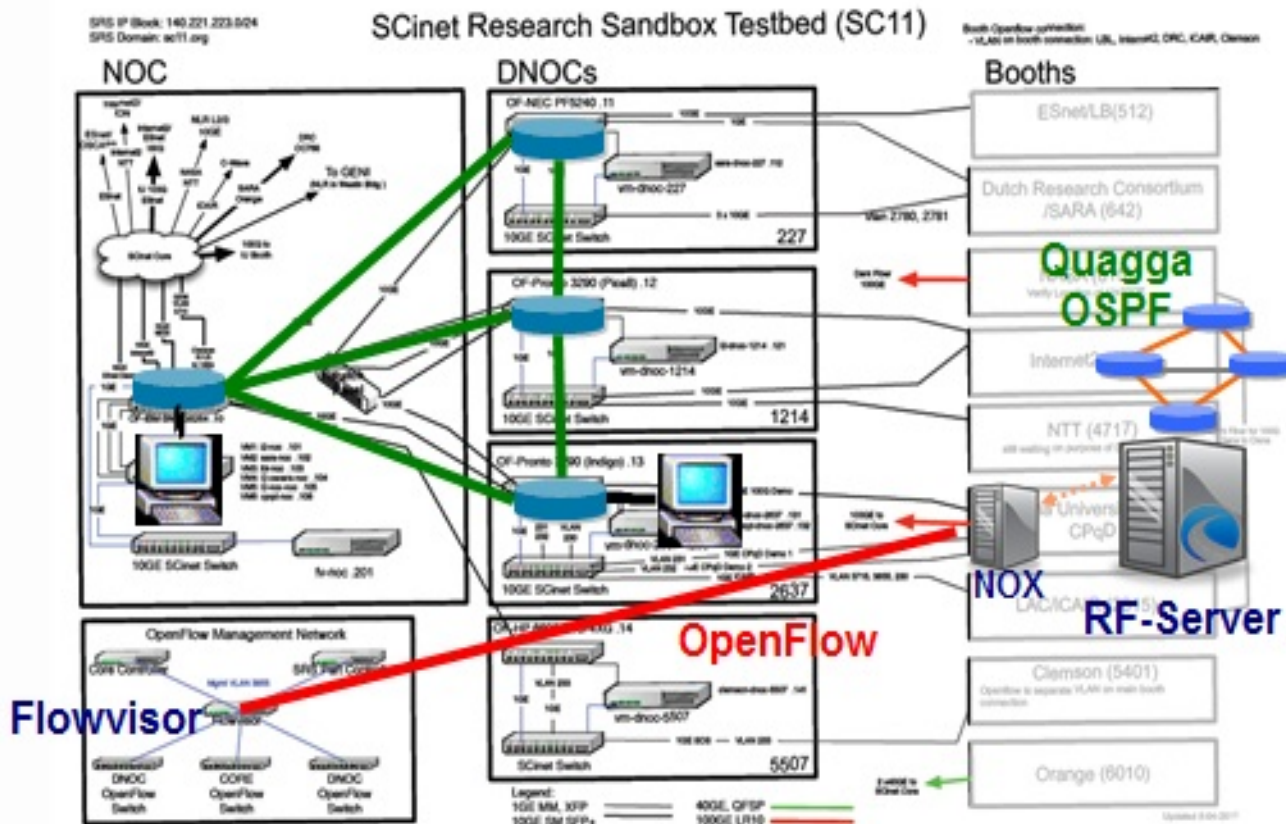


RouteFlow

SRS Demo @ SC11



Seattle, WA
Connecting communities
through HPC



Routing configuration at
your fingertips

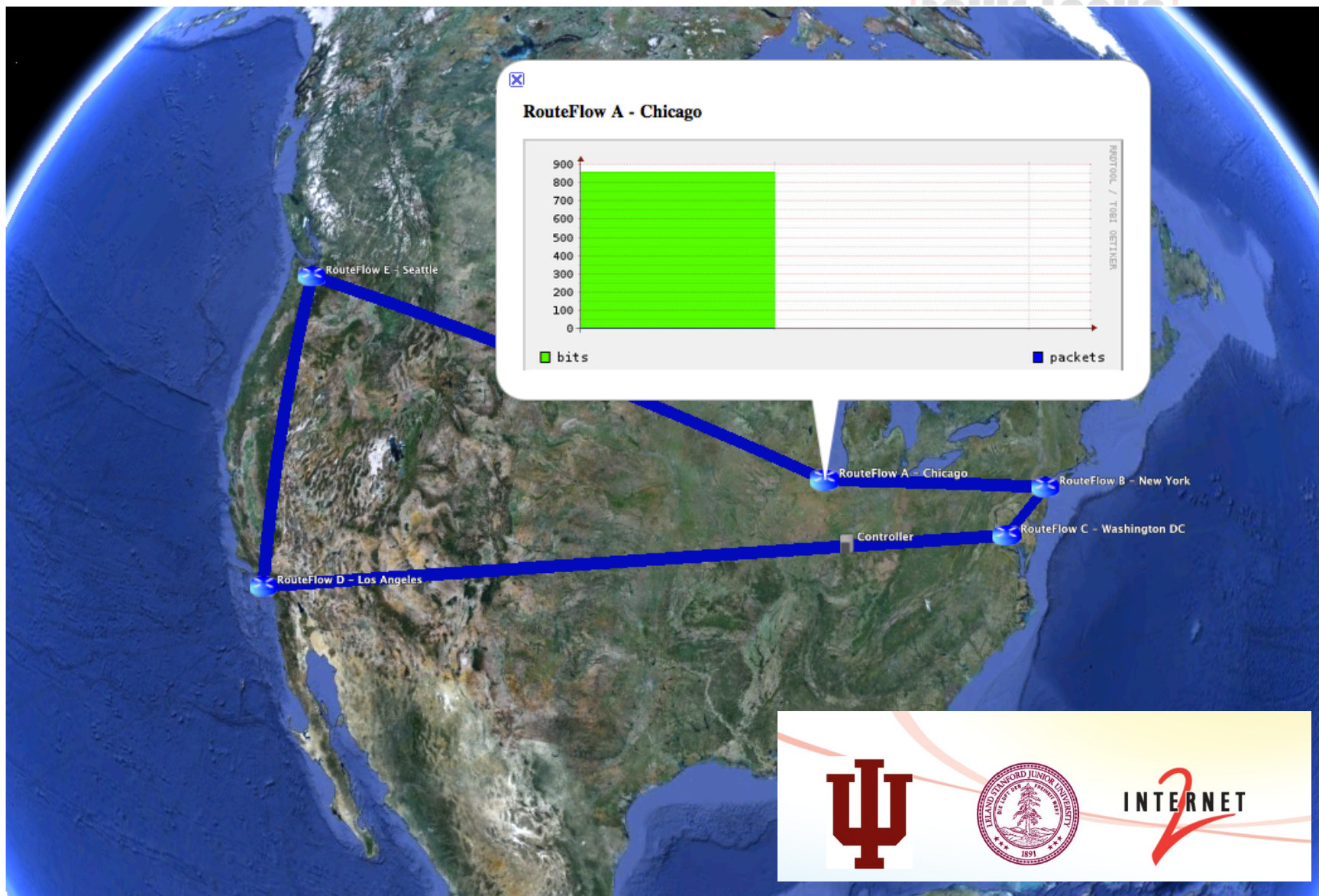


II Open Networking Summit (Santa Clara)



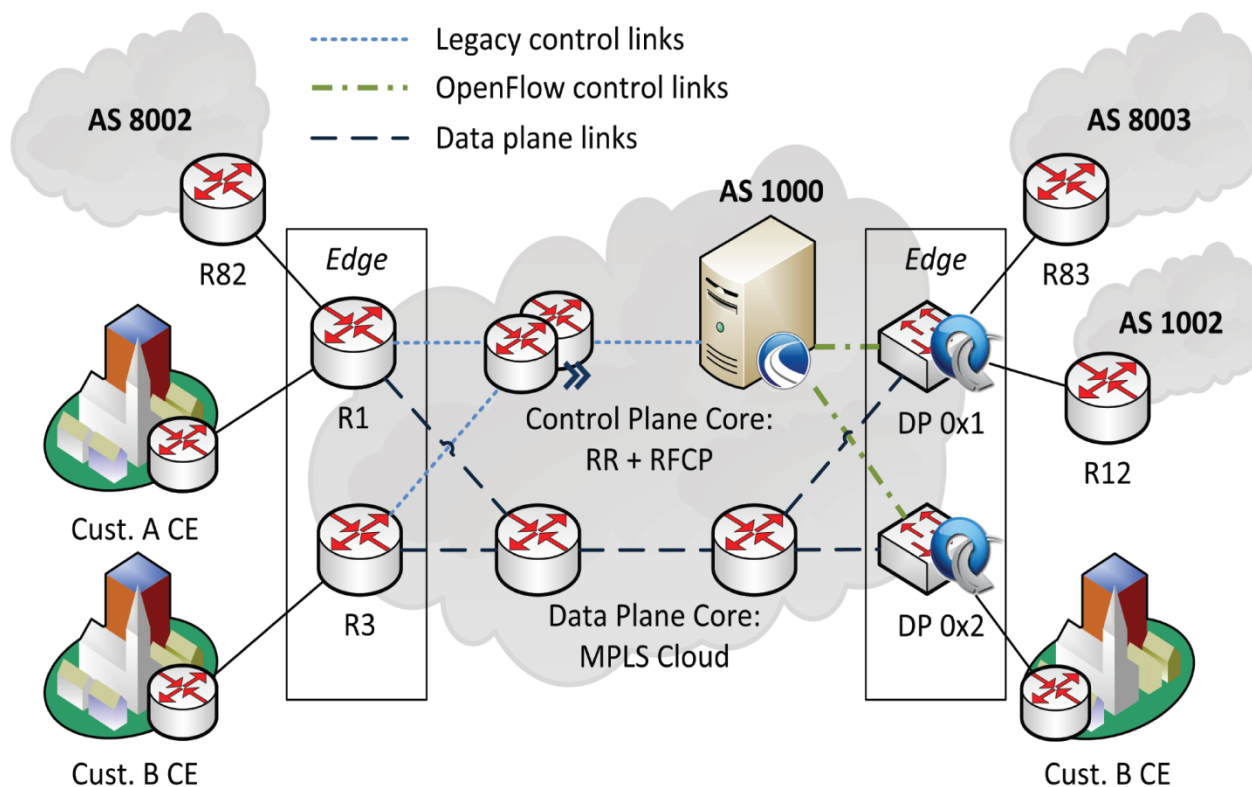
RouteFlow NDDI Deployment

[Joint Techs]



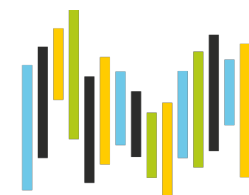
Controller-Centric Hybrid Networking

- A migration path to roll out OpenFlow technology
- Not a revolution, but an evolution of current iBGP RRs to essentially eBGP Route Controllers
 - “BGP-free edge”: A cost-effective simplified edge for SW-driven innovations

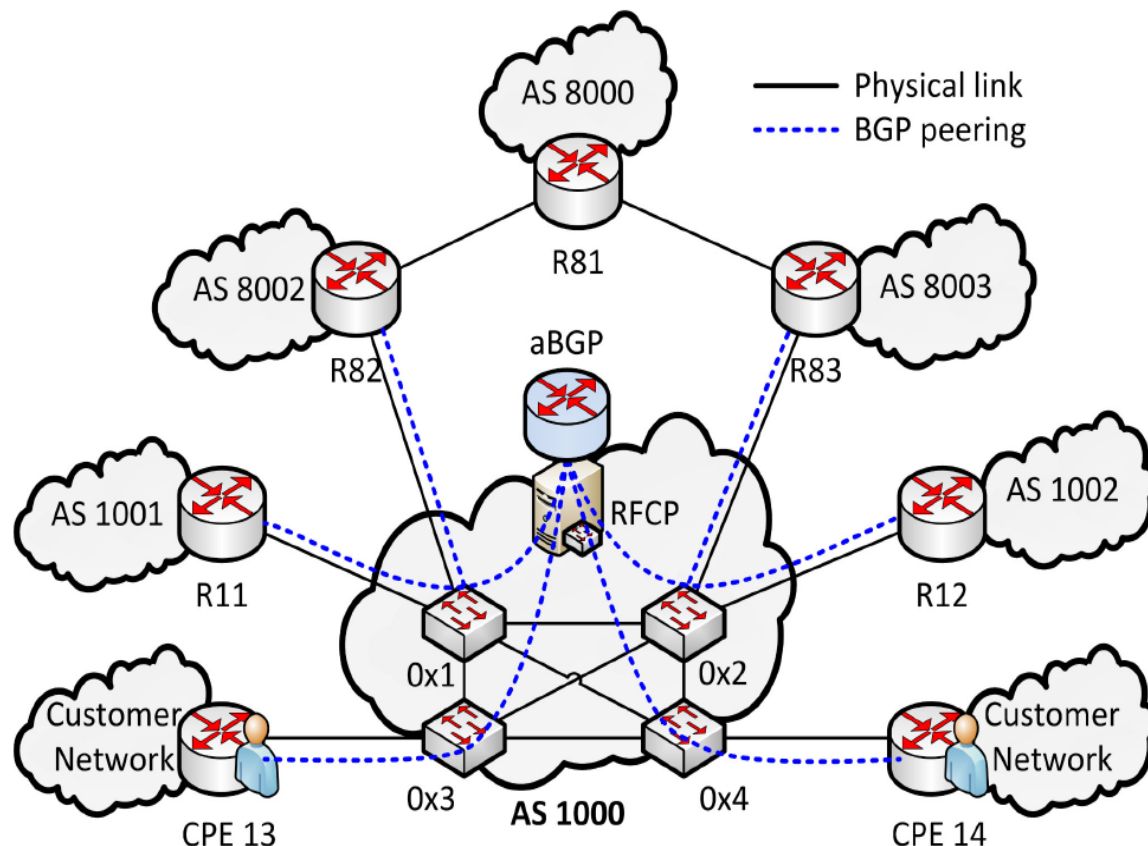


Aggregated BGP routing service → RCP

- Single node abstraction of a domain-wide eBGP router
 - Think modern multi-chassis routing architectures with external route processors and OpenFlow switches acting as line cards
- Aggregation logic defined in the RF-Server



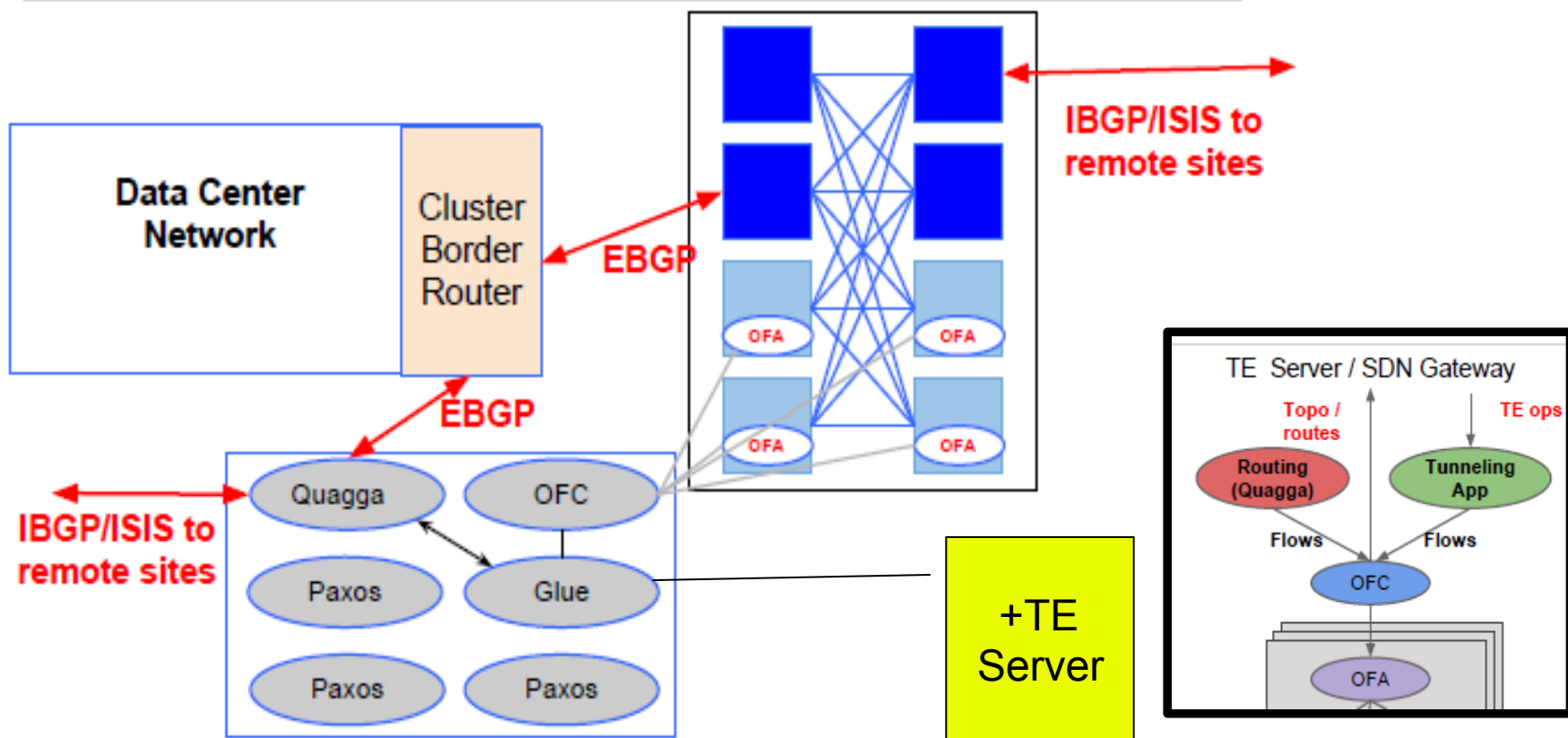
SIGCOMM
2012 HELSINKI



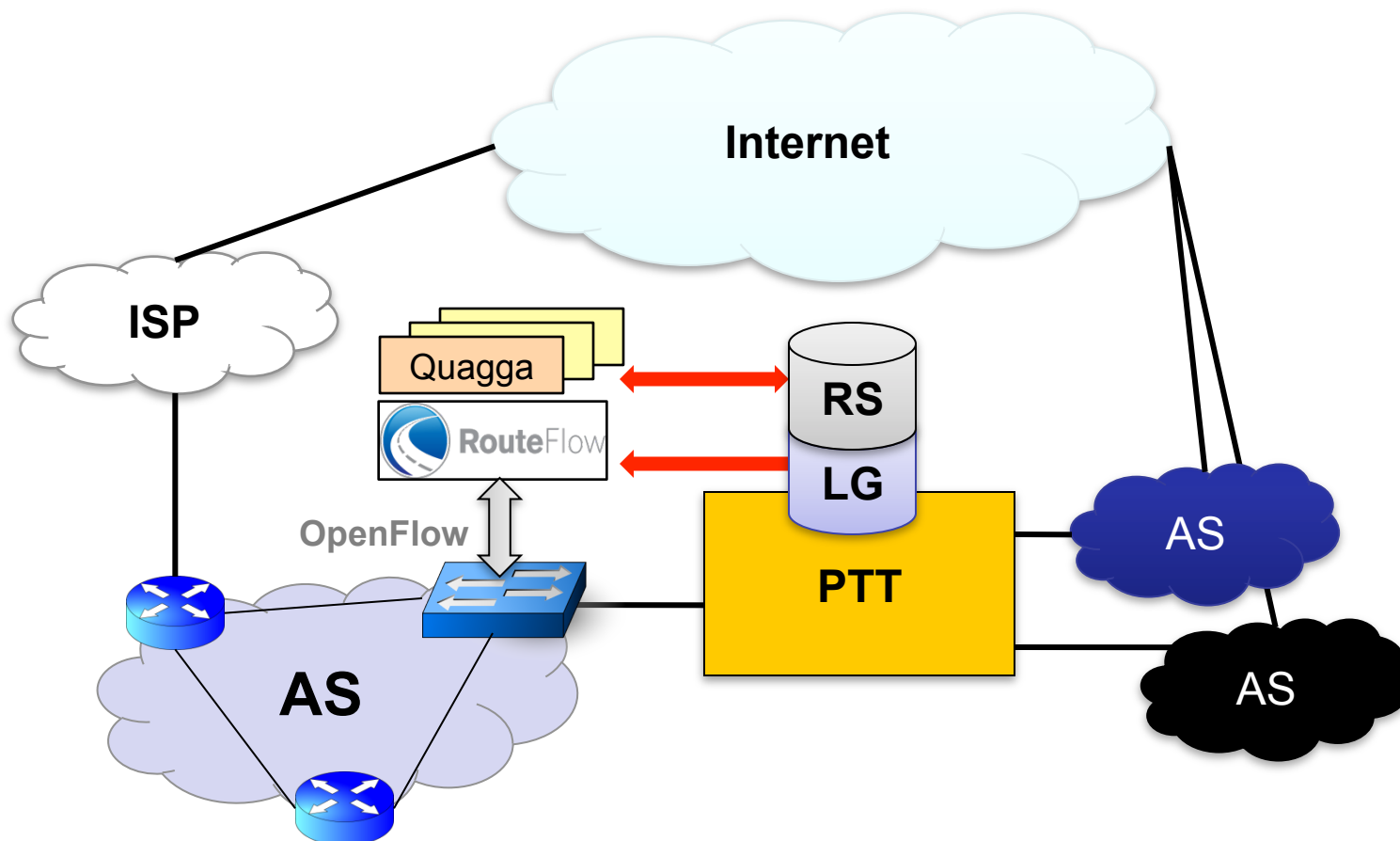
Google Software Defined WAN Architecture

Mixed SDN Deployment

Google



RouteFlow em rede cliente do PTT



Quagga on Steroids

SW Forwarding

- Only for OpenFlow cache misses
- Full FIB (virtually unlimited)

HW Forwarding

- FIB cache*
(2 – 4 K entries)

* use Simple VA + smart caching

Quagga on Linux PC



Server PC
~ 1K US\$

Data channel
(on flow table miss)
1:1GE or N:10GE ports

Control channel
(OpenFlow 1.0)

OpenFlow
Switch



Pronto 3290
(48x 1G + 4x 10G)
~ 2K US\$ (FOB)

Next Hop Interfaces

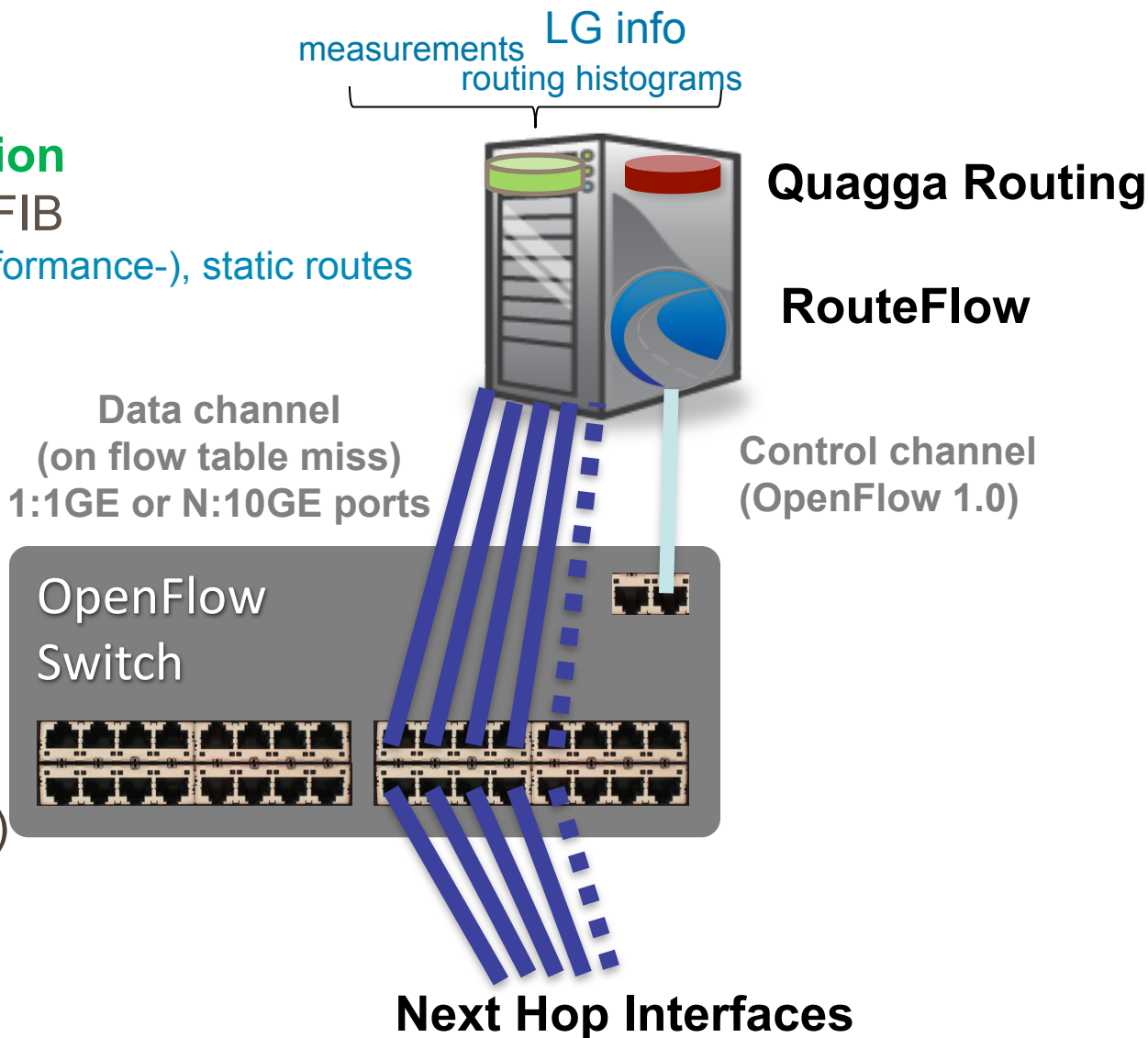
Quagga on Steroids + Customized Control

New control application

- Overwrites dynamic FIB
 - e.g., PBR (cost- / performance-), static routes
- Multi-homing/pathing
- Back-up routes
- Security rules

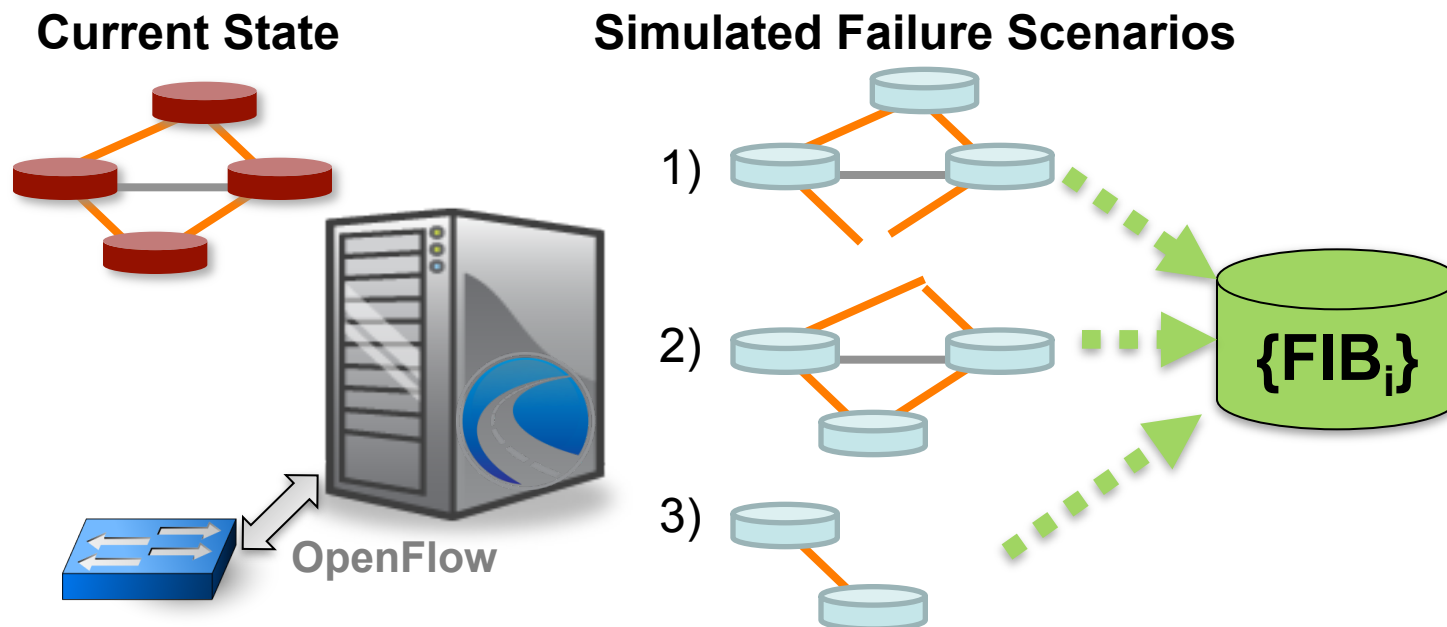
HW Forwarding

- FIB cache
- + new prioritized entries (e.g. ACL, PBR)



Fast convergence and availability

- Non-stop forwarding ready!
 - OpenFlow controller and IP control plane can be unavailable but FIB state is kept stale in the switches to forward packets
- Shadow control plane to test failure events and store next valid consistent states.
 - Upon failure, delta flow entries can be pushed down



Security

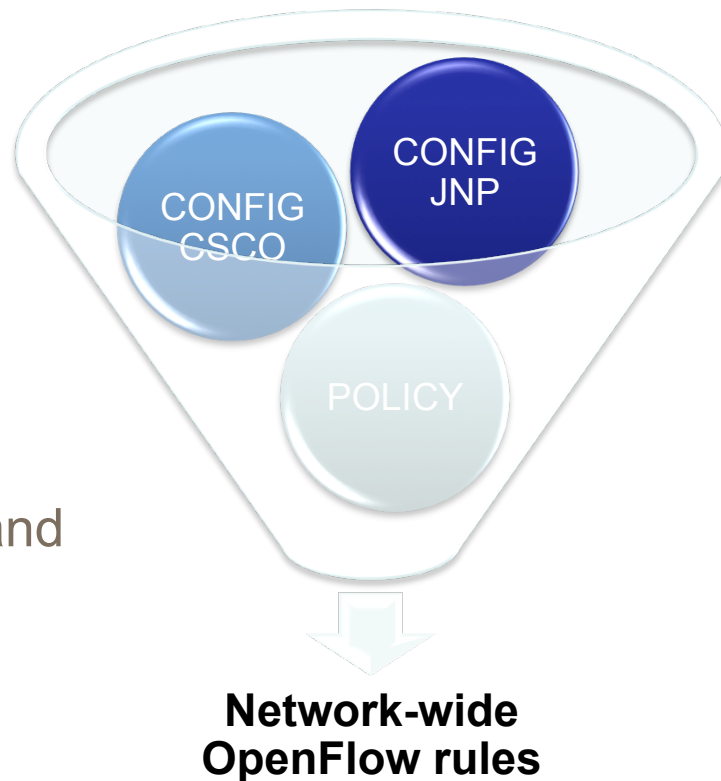
Towards a network-wide firewall

- Import existing device-level configurations
- Consolidate and review
- Import OpenFlow topology
- Transform into OpenFlow-rules
- Provide single point of configuration, roll-back, annotated histogram (who and why and for how long inserted a rule?), etc.
- Develop APIs for other systems, e.g., IDS

Secure Inter-domain routing (S*-BGP, etc..)

Data plane blackholes

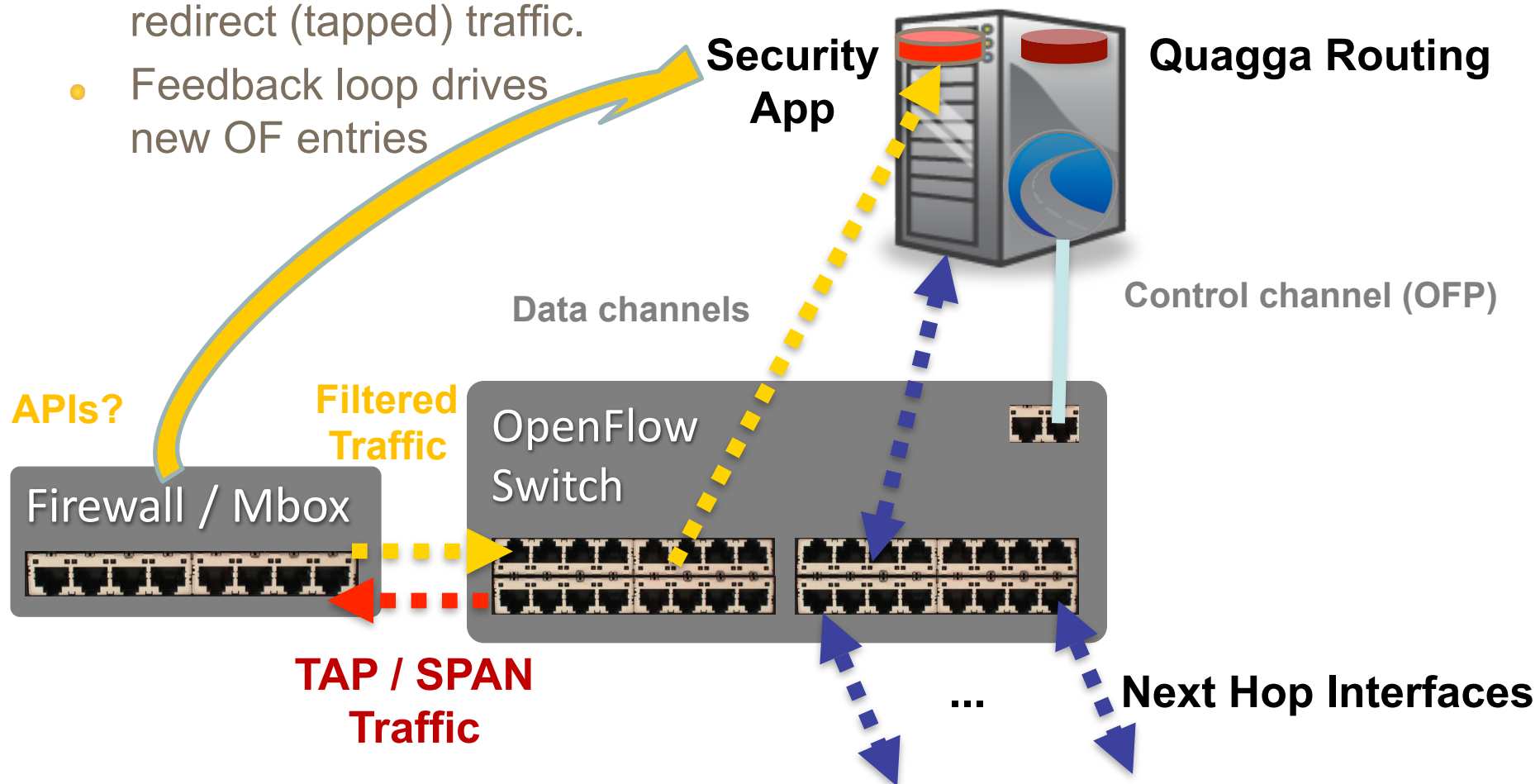
Middlebox injections



Security

Hybrid firewalling

- Re-use existing firewalls / middleboxes (IDS, Net/SDlow) and redirect (tapped) traffic.
- Feedback loop drives new OF entries



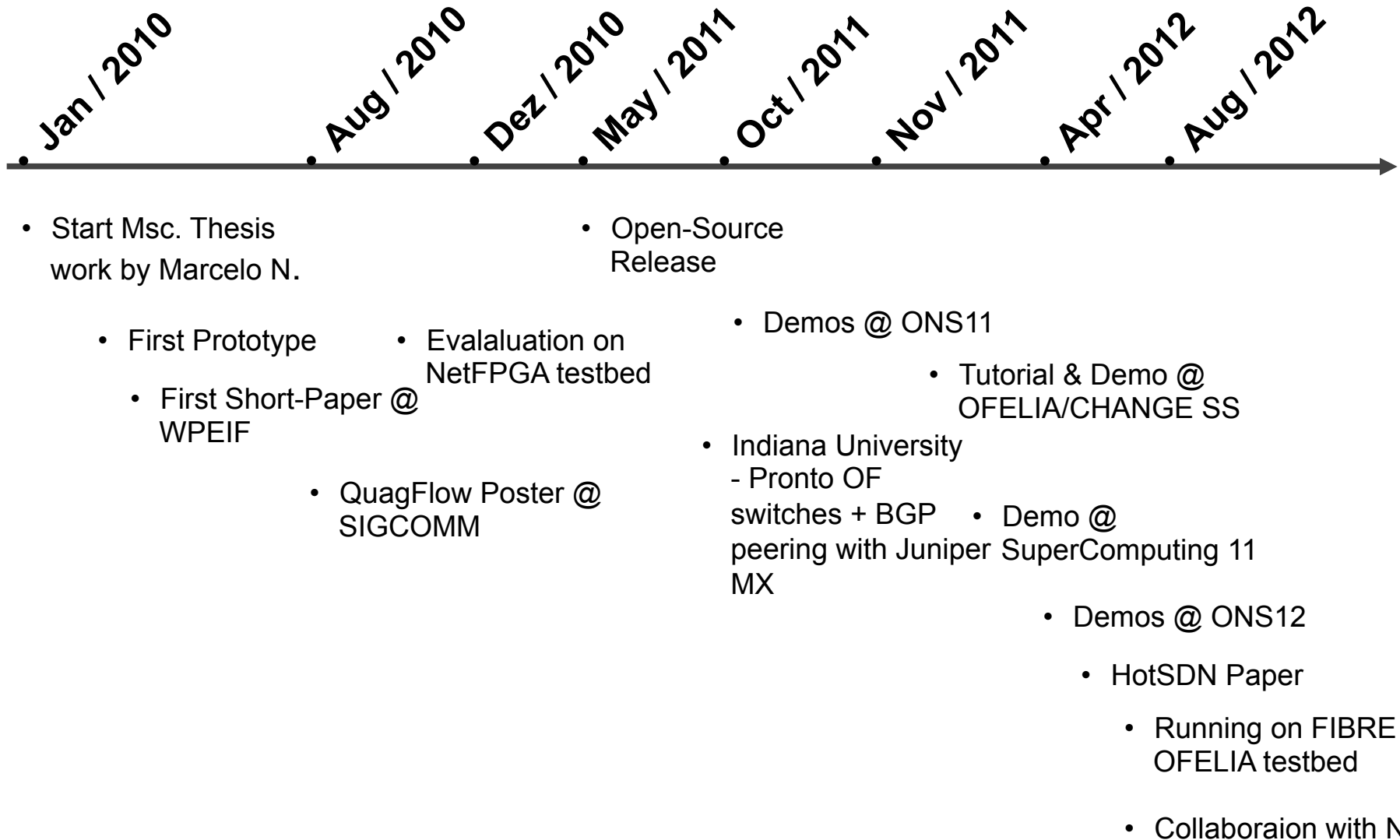
Routing-centric use cases under research

- Engineered path selection
 - Think Google WAN, performance-based routing, etc.
- Optimal best path reflection
 - Per ingress/customer [draft-ietf-idr-bgp-optimal-route-reflection-01]
- Path protection with prefix independent convergence
 - Hierarchical FIBs w/ OF 1.X Tables + LFA route-precomputation
- Security
 - Data plane blackholes and middlebox injections,
 - Secure Inter-domain routing ideas (crypto intense S*-BGP, etc..)
- Simplifying customer multi-homing
 - Easy to set and control cost/performance/policy-based routing
- IPv6 migration
 - Flow matching for service termination in v4-v6 migration solutions

Challenges

- Centralized BGP
 - Shown to scale well in modern CPU architectures
 - Centralized does not mean not distributed (but removal from edge)
- Small OpenFlow table sizes
 - Transient limitation?
 - Expose existing FIB data structures as an IP lookup OF table?
 - Smart RIB&FIB reduction (e.g., simple [draft-ietf-grow-simple-va-04])
 - HW/SW flow offloading
- Limited OpenFlow processing in datapath
 - Transient / Un-optimized implementations
- High availability
 - Previous ideas from distributed RCPs
 - Database-centric designs
 - Development in-progress of “BGP SHIM” for transparent eBGP redundancy

RouteFlow Project History

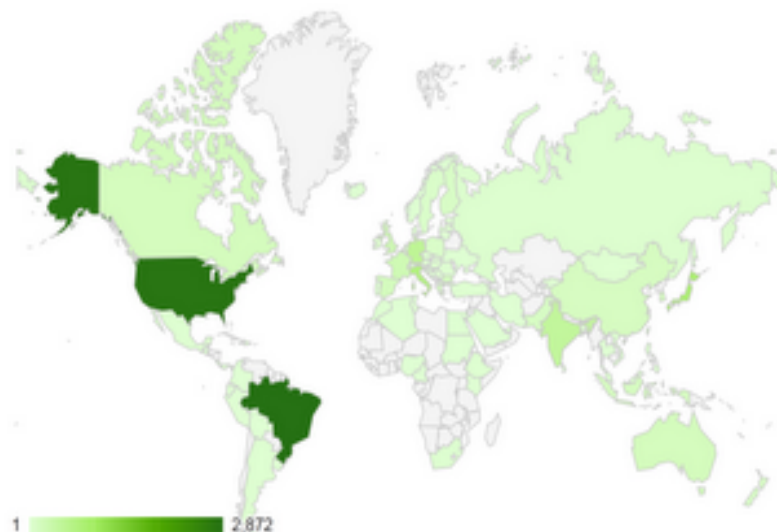
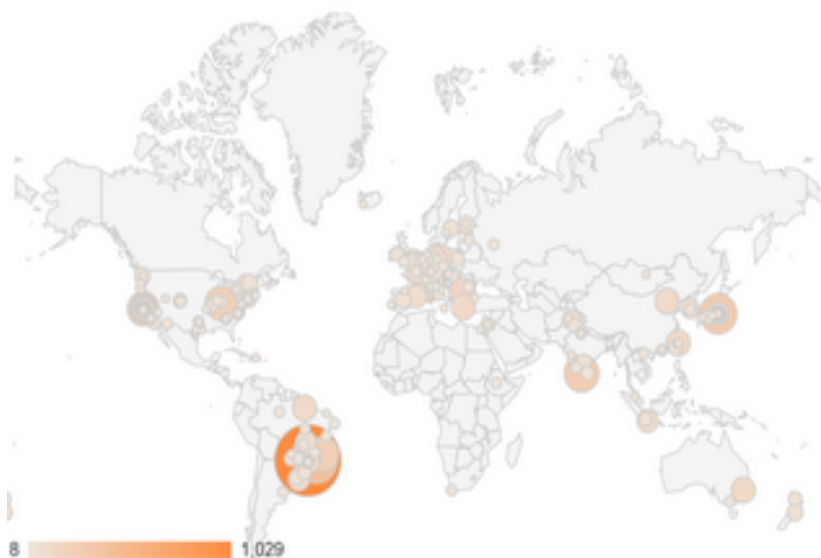


... building a community



Visits: 24,000+ (11,000+ Unique)

From over 2000 cities of 110+ countries all over the globe!



<http://go.cpqd.com.br/routeflow/>



582
days since
Project Launch



1000s
downloads!



Collaborations and community developments

- Web-based UI & Internet 2 HW pilot [C. Small, Indiana] ✓
- Aggregated BGP Routing Service [C. Corrêa, Unirio] ✓
- SNMP plugin [J. Stringer, Google] ✓
- Optimal BGP best path reflection [R. Raszuk, NTT-MCL] ☹
- Open Label Switched Router [OSRF; Google] ☹
- OpenFlow v1.2 and v1.3 [w/ Ericsson] ☹



RouteFlow



Open Source Routing



Evolving the IP routing landscape with OpenFlow/SDN

Comentário finais

RouteFlow como “*Quagga on steroids*”

- Open-source routing SW + line-rate forwarding on HW

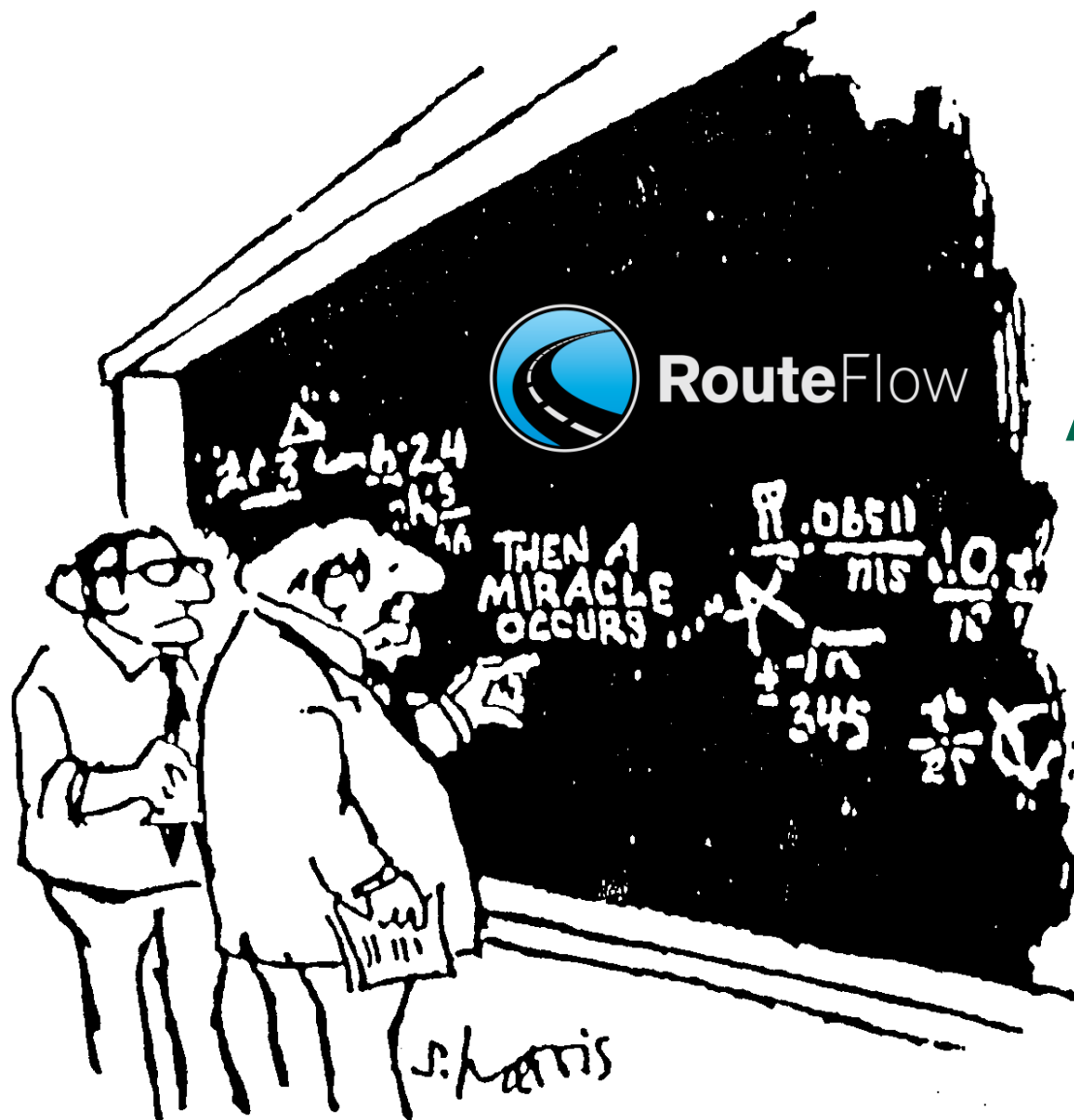
RouteFlow como exemplo e oportunidade para inovar no roteamento IP

- **Parceria com usuários “beta” nacionais**
 - com requisitos e dores de cabeça reais,
 - com orçamentos restritos,
 - sem limites na vontade de fazer acontecer
- **Ex: Policy-based routing (cost-, performance), convergência rápida, segurança, migração para IPv6, etc.**

OpenFlow/SDN como tecnologia nos PTT

- Automation, self-service provisioning, validation, security, etc.

Questions?



Thank you!

Ask and contribute!

routeflow-discuss@googlegroups.com

Learn more!

<https://go.cpqd.com.br/routeflow>

Get the Code!

<https://github.com/CPqD/RouteFlow>

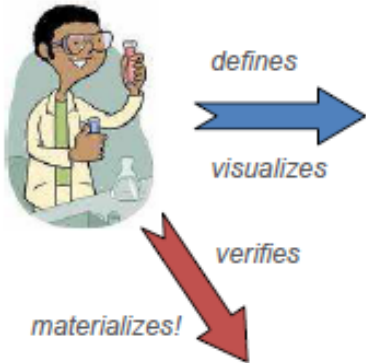
BACKUP

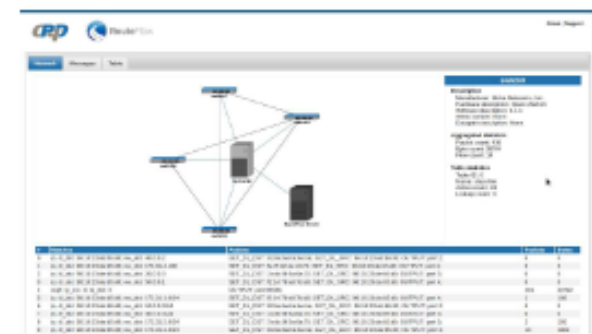
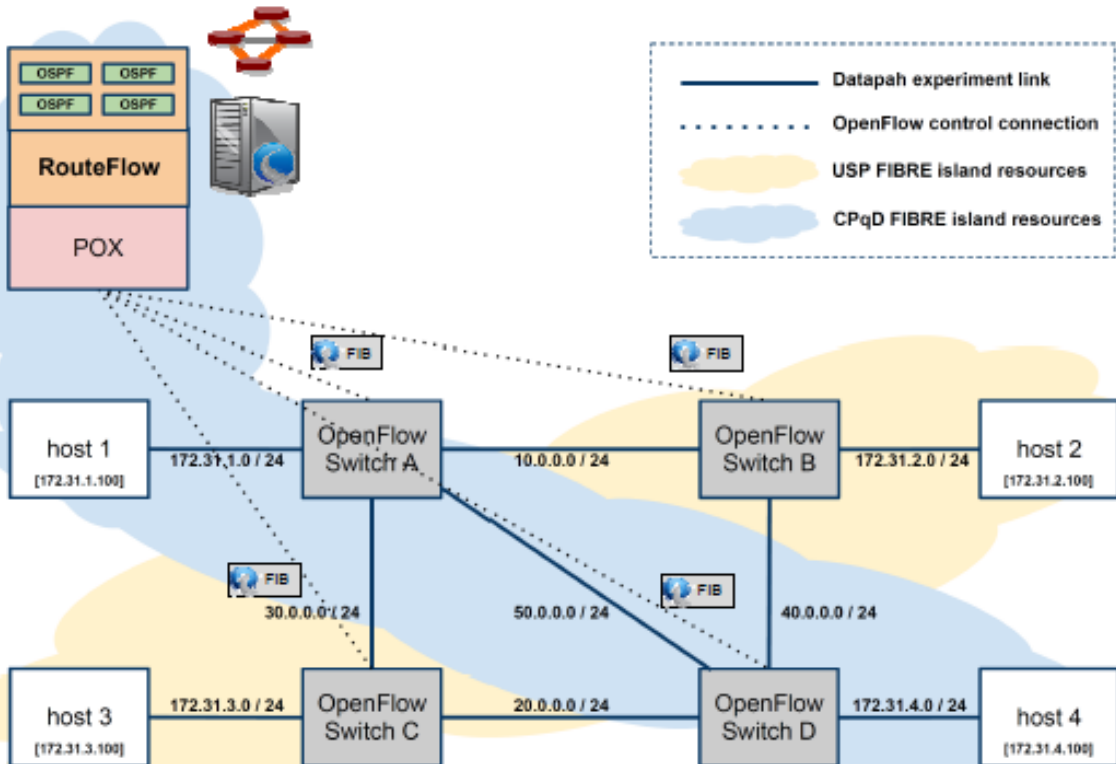
IP Network as a Service (NaaS)

FIBRE Aggregate

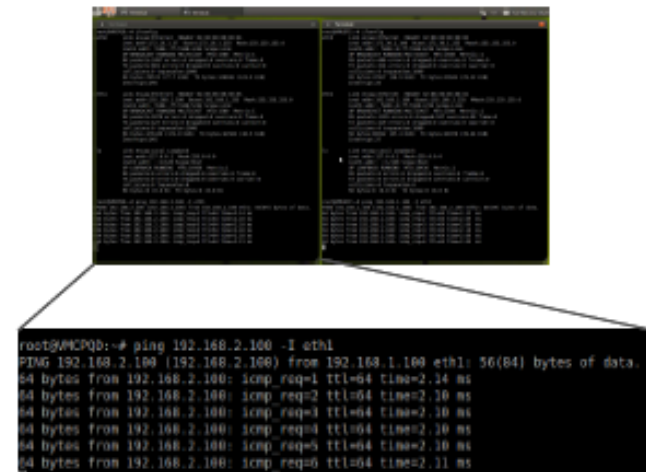
FIBRE Slice

RouteFlow GUI



VM end-to-end PING



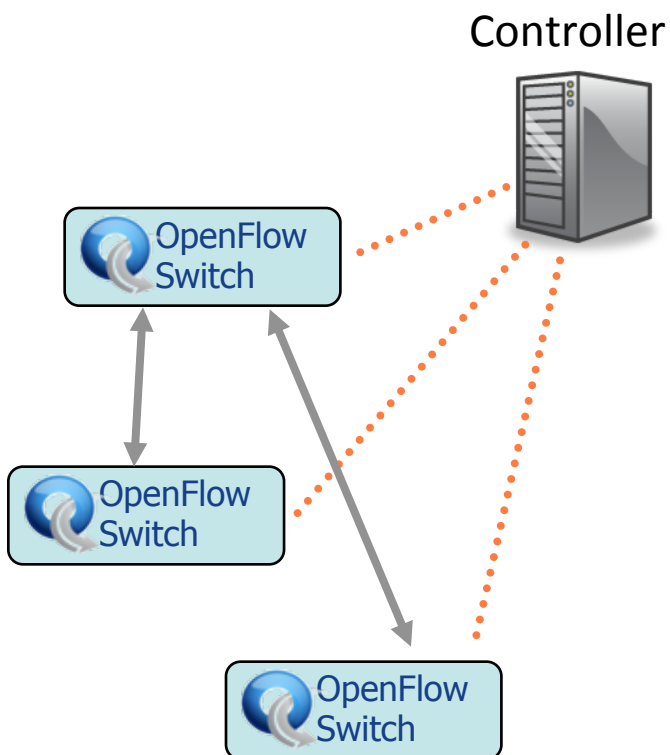
```

root@MCPQB:~# ping 192.168.2.100 -I eth1
PING 192.168.2.100 (192.168.2.100) from 192.168.1.100 eth1: 56(84) bytes of data:
64 bytes from 192.168.2.100: icmp_req=1 ttl=64 time=2.34 ms
64 bytes from 192.168.2.100: icmp_req=2 ttl=64 time=2.39 ms
64 bytes from 192.168.2.100: icmp_req=3 ttl=64 time=2.39 ms
64 bytes from 192.168.2.100: icmp_req=4 ttl=64 time=2.39 ms
64 bytes from 192.168.2.100: icmp_req=5 ttl=64 time=2.39 ms
64 bytes from 192.168.2.100: icmp_req=6 ttl=64 time=2.31 ms
  
```

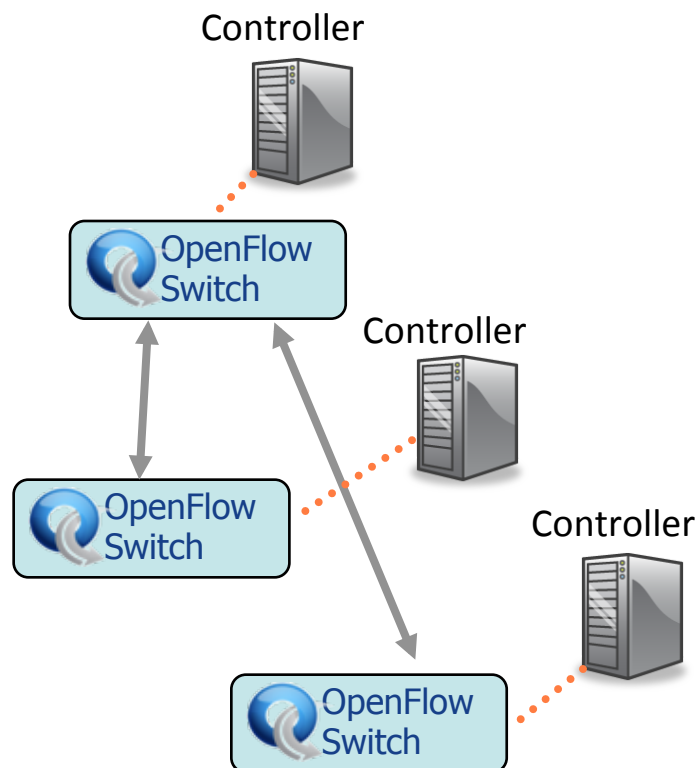
Centralized vs Distributed Control

Both models are possible with OpenFlow

Centralized Control



Distributed Control



Flow Routing vs. Aggregation

Both models are possible with OpenFlow

Flow-Based

- Every flow is individually set up by controller
- Exact-match flow entries
- Flow table contains one entry per flow
- Good for fine grain control, e.g. campus networks

Aggregated

- One flow entry covers large groups of flows
- Wildcard flow entries
- Flow table contains one entry per category of flows
- Good for large number of flows, e.g. backbone

Reactive vs. Proactive (pre-populated)

Both models are possible with OpenFlow

Reactive

- First packet of flow triggers controller to insert flow entries
- Efficient use of flow table
- Every flow incurs small additional flow setup time
- If control connection lost, switch has limited utility

Proactive

- Controller pre-populates flow table in switch
- Zero additional flow setup time
- Loss of control connection does not disrupt traffic
- Essentially requires aggregated (wildcard) rules