

OpenFlow & SDN for Network Providers

Smart Devices Need Smart Networks

Christos Kolias, Strategy & Development Group
Orange Silicon Valley

7th AdvancedTCA Summit, November 1-2, 2011



Agenda

- State of the Internet
- Why OpenFlow and Software-Defined Networking
- How can Network Carriers & Providers benefit
- Use Cases & Applications
- “Remote Control” app @ Orange SV

Disclaimer: the following constitute my own opinions, thoughts and views and not necessarily those of the company

Orange, a worldwide presence

- **Orange Silicon Valley (OSV)**, a wholly owned subsidiary of the Orange-France Telecom Group, is the Silicon Valley presence of **Orange**
- **Internet, Fixed, Mobile, TV** provider
- **Orange** is one of the major telcos, in **5** continents, **32** countries, **210 million** customers, **6 million** business customers
- **180,000** employees and **45.5 bn €** turnover in 2010



State of the Internet

State of the Internet

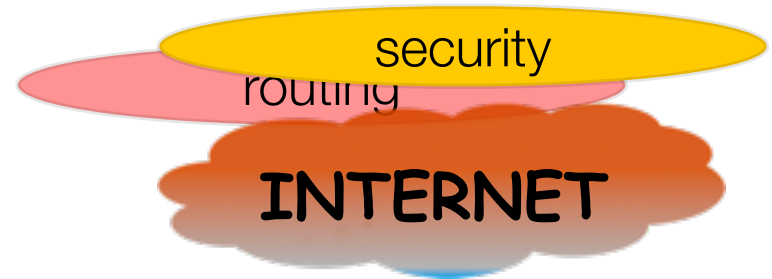
- Networks have become ossified, rigid, static, inefficient



INTERNET

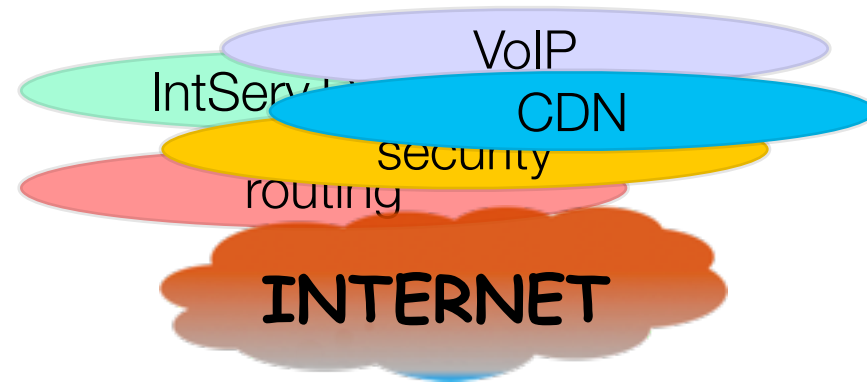
State of the Internet

- Networks have become ossified, rigid, static, inefficient
 - Built overlays over the years (VPN, MPLS, P2P, VoIP, CDN, etc) -> *too complex*



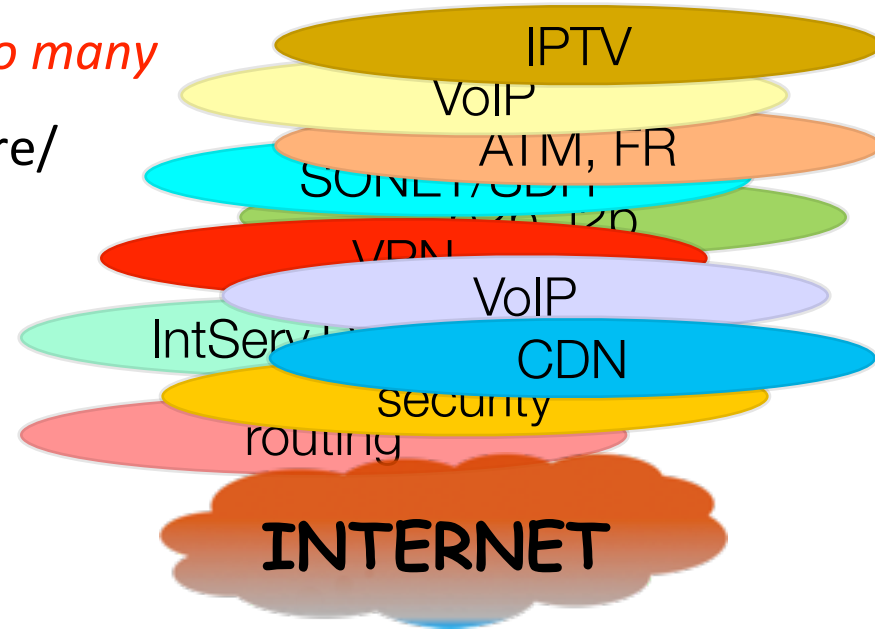
State of the Internet

- Networks have become ossified, rigid, static, inefficient
 - Built overlays over the years (VPN, MPLS, P2P, VoIP, CDN, etc) -> *too complex*
 - Load-balancer, Firewall, DNS server, WAN optimization, monitoring devices -> *too many*



State of the Internet

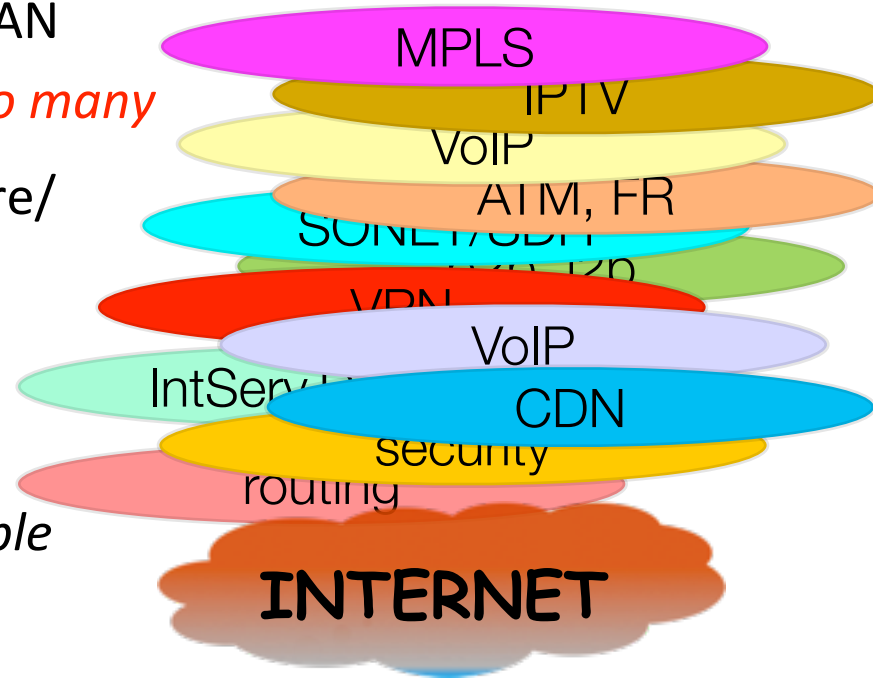
- Networks have become ossified, rigid, static, inefficient
 - Built overlays over the years (VPN, MPLS, P2P, VoIP, CDN, etc) -> *too complex*
 - Load-balancer, Firewall, DNS server, WAN optimization, monitoring devices -> *too many*
 - Routers come overloaded with software/features not used
 - -> *too much*



State of the Internet

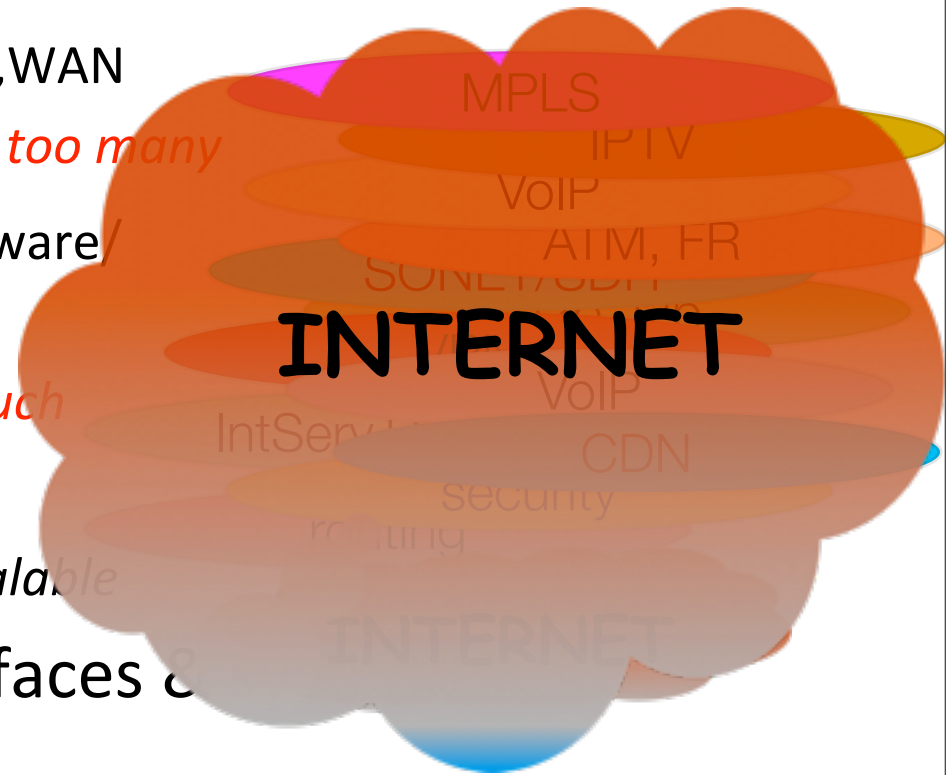
■ Networks have become ossified, rigid, static, inefficient

- Built overlays over the years (VPN, MPLS, P2P, VoIP, CDN, etc) -> *too complex*
- Load-balancer, Firewall, DNS server, WAN optimization, monitoring devices -> *too many*
- Routers come overloaded with software/features not used
 - -> *too much*
- Routing protocols, PIM, VLAN, STP
 - > *too old, not scalable*



State of the Internet

- Networks have become ossified, rigid, static, inefficient
 - Built overlays over the years (VPN, MPLS, P2P, VoIP, CDN, etc) -> *too complex*
 - Load-balancer, Firewall, DNS server, WAN optimization, monitoring devices -> *too many*
 - Routers come overloaded with software/features not used
 - -> *too much*
 - Routing protocols, PIM, VLAN, STP
 - > *too old, not scalable*
- Closed and proprietary interfaces &



- New requirements and challenges:
 - video
 - mobility/wireless
 - security
 - energy
 - virtualization
 - multi-tenancy
 - data centers
 - cloud computing
- Hard to innovate, determine how you want to handle traffic, e.g., define your own routing policy or security policy
- Result: networks have become difficult and expensive to build, operate and manage
- Few trends: openness, hardware commoditization & software differentiation, unified access, big data/big

What is Software-Defined Networking

What is Software-Defined Networking

“Users can define traffic flows and decide how these are treated in the network”

What is Software-Defined Networking

“Users can define traffic flows and decide how these are treated in the network”

“Installing small piece of firmware, users get direct access to the hardware”

- Remotely control network hardware with software in a dynamic, programmatic fashion -> *programmable networks*
- Supports separation of control (intelligence/brain) and data (speed/muscle): creates an abstraction and logical map of topology
- Decoupling of physical infrastructure and traffic mgmt
- Uses standard, open interfaces (to the hardware)
- Hardware agnostic
- It can change the way we build systems today

Why SDN?

“It is the economics, stupid”

Why SDN?

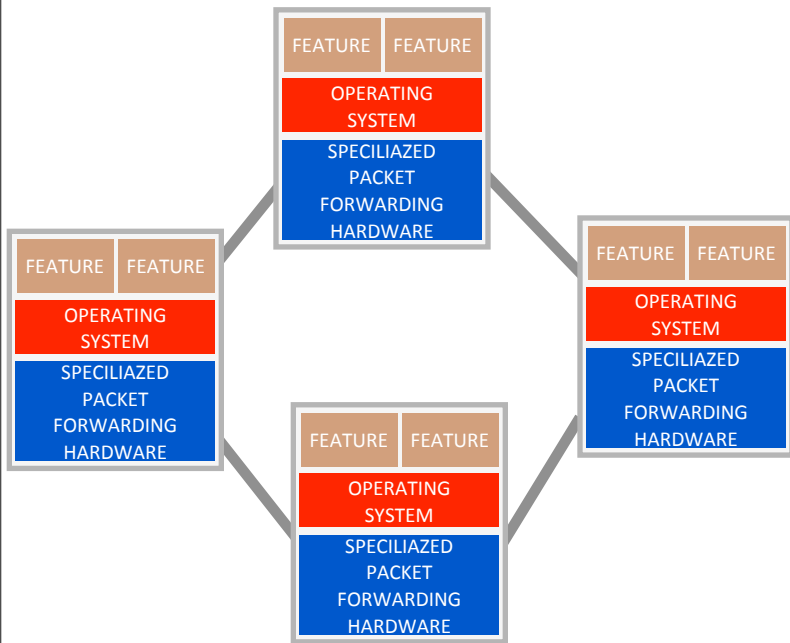
“It is the economics, stupid”

“People want to control their own network”

- Can run new services & apps on top of hardware
- Added flexibility & control
- Advanced capabilities & network functions
- Allows for optimization & customization: BYOR
- Empower the administrator/operator
- Network Expenses (NetEx) savings
- What about a network-wide OS?

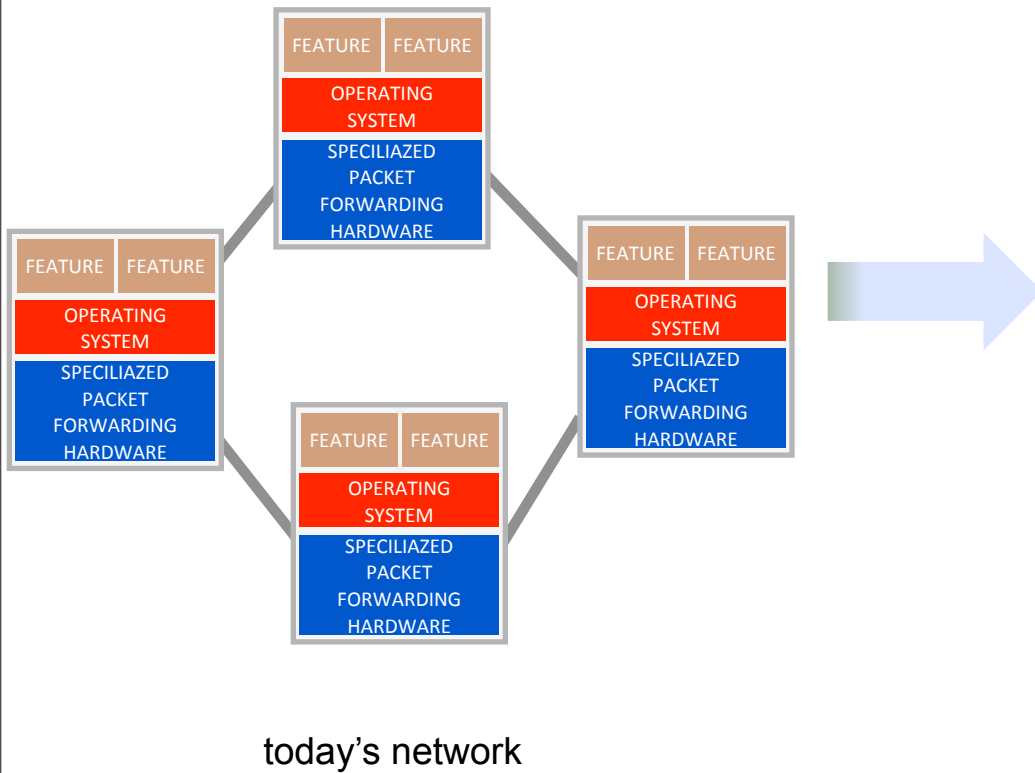
Evolution of Networking

Evolution of Networking

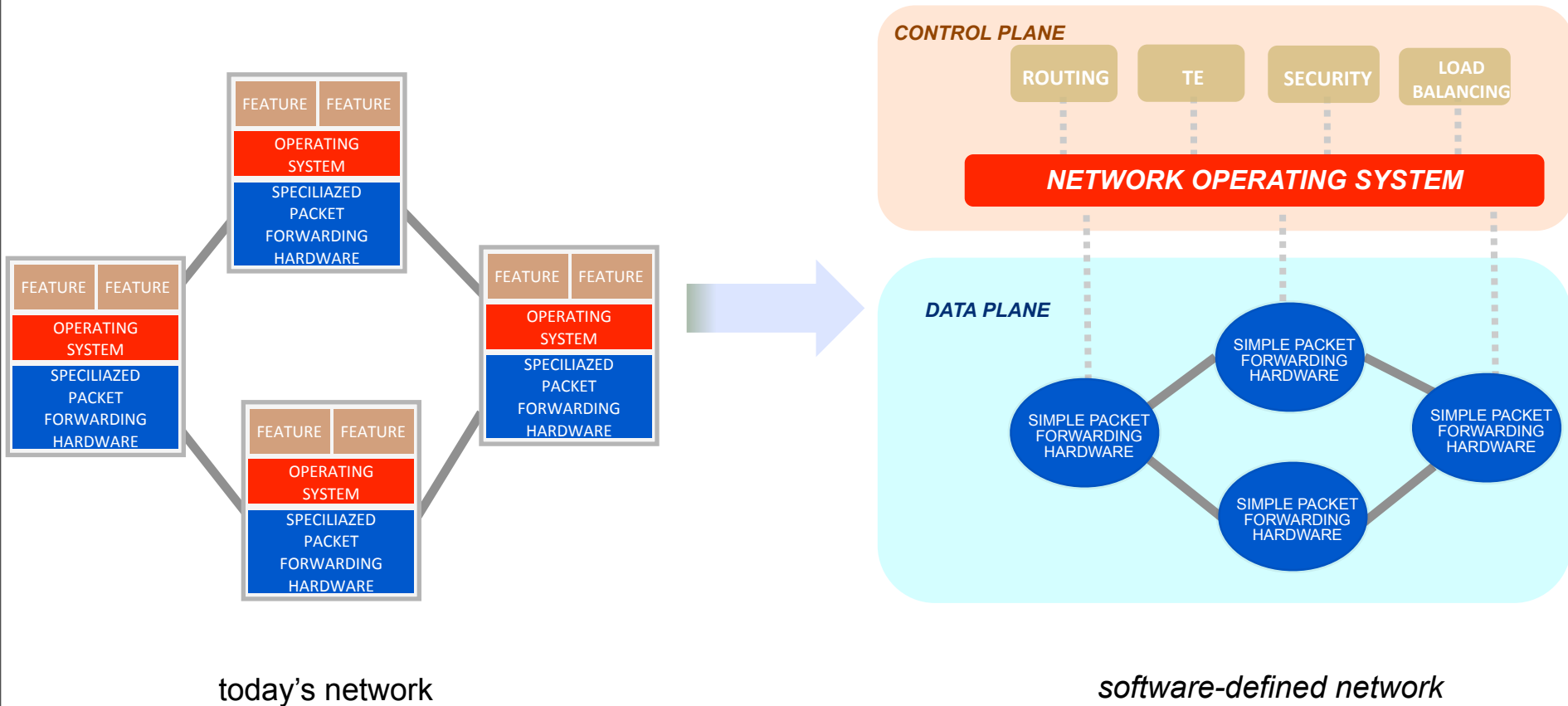


today's network

Evolution of Networking



Evolution of Networking



today's network

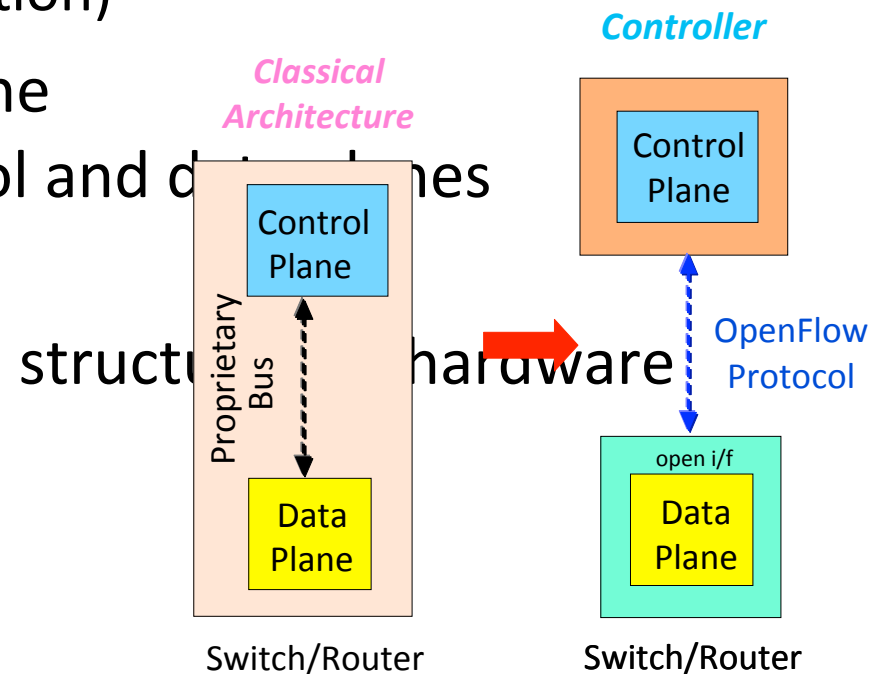
software-defined network

[source: ONF]

- the abstraction of the control plane creates a logical map of the network

OpenFlow

- Built around the concept of traffic flows
 - identify flows (matching, statistics)
 - dynamically manage flows (action)
- Supports clean separation of the control and data planes
- Takes advantage of existing hardware structures
- Firmware
- Vendor neutral
- Open source. Current spec 1.1



- Main components:
 - *openflow* table: inside each switch (e.g., TCAM)

in_port	MAC_src	MAC_dst	IP_src	IP_dst	Action	Stats
4	*	00:1f..	1.2.3.4	5.6.7.8	port 22	1000
20	*	*	*	2.3.4.5	drop	4
12	*	*	*	*	controller	235
1	7e:4b..	5a:2c..	6.7.8.1	*	port 1	37
8	6b:11..	01:ab..	8.7.6.5	4.3.2.1	all ports	89

- Main components:
 - *openflow* table: inside each switch (e.g., TCAM)

- Main components:
 - *openflow* table: inside each switch (e.g., TCAM)



The screenshot shows a PuTTY terminal window titled "192.168.2.50 - PuTTY" with a date of "2010/11/30 19:08:39 UTC". The terminal output displays the following information:

```
Number of Active FLOW entries per switch

  < in port >   <entry count>
  0/ 5[0x0005]   1
  0/ 6[0x0006]   1
  any            1
  total         3

FLOW entries information

[OpenFlow 1]

<pri> < in port > <entry no> < matched octets > < matched packets > <table>
32768 0/ 5[0x0005] 2515 1350056 990 nml
32768 0/ 6[0x0006] 2516 1280780 934 nml
100 any 2955 1085331089 862532 nml
```

- *controller*: outside switches, software
 - e.g., NOX – you can build your own controller
- *secure channel*: between controller and switches
- I can program network without modifying every switch
- Beyond MPLS, VLANs, ACL, QoS, RSVP

Benefits

- Programmable platform
- Virtualization
 - network virtualization
 - service virtualization
- Control & management of traffic
- Creates a flat L2 domain
- Evolution (or extinction) of VPNs, VLANs, MPLS ?



Net A



Net B

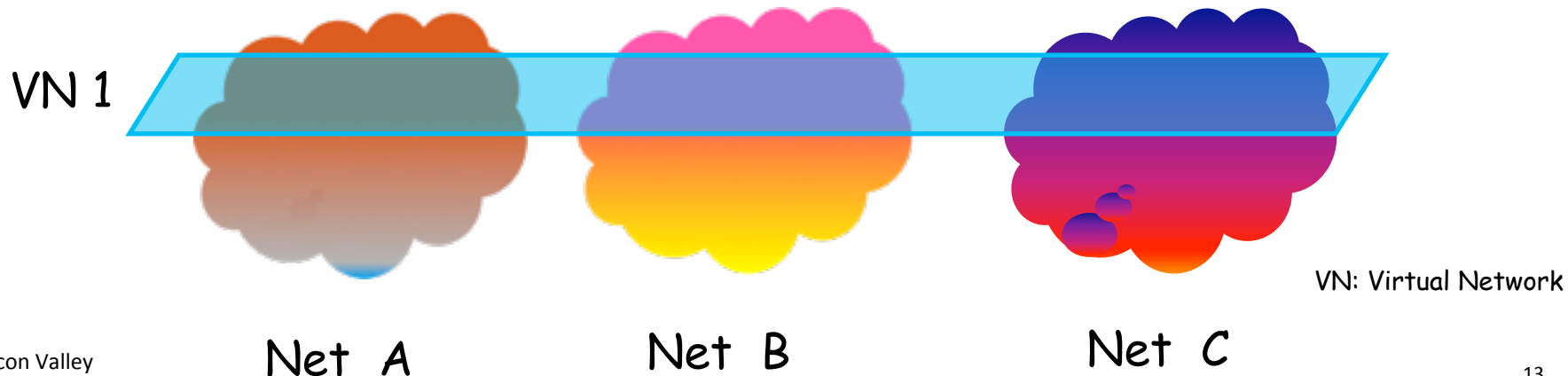


Net C

VN: Virtual Network

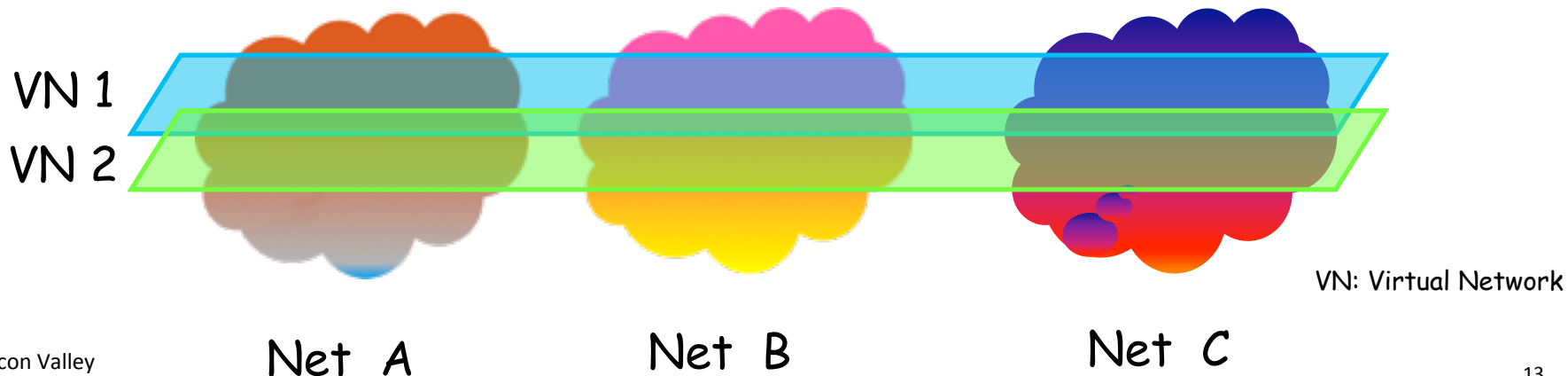
Benefits

- Programmable platform
- Virtualization
 - network virtualization
 - service virtualization
- Control & management of traffic
- Creates a flat L2 domain
- Evolution (or extinction) of VPNs, VLANs, MPLS ?



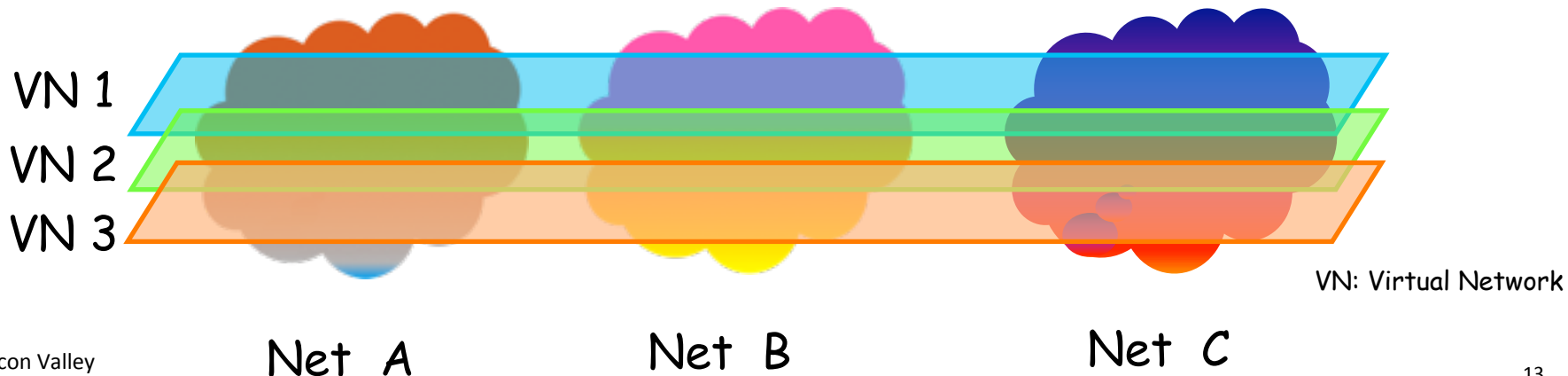
Benefits

- Programmable platform
- Virtualization
 - network virtualization
 - service virtualization
- Control & management of traffic
- Creates a flat L2 domain
- Evolution (or extinction) of VPNs, VLANs, MPLS ?



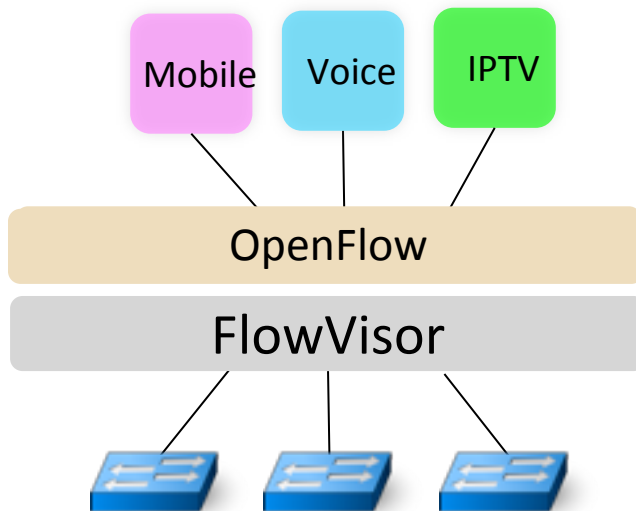
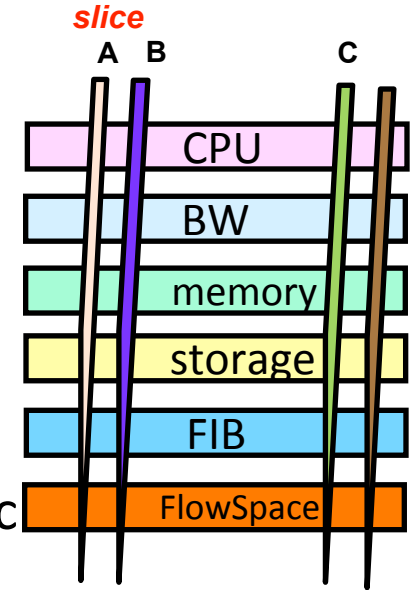
Benefits

- Programmable platform
- Virtualization
 - network virtualization
 - service virtualization
- Control & management of traffic
- Creates a flat L2 domain
- Evolution (or extinction) of VPNs, VLANs, MPLS ?



Network Virtualization with OpenFlow

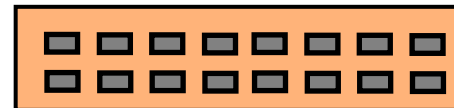
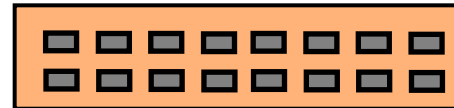
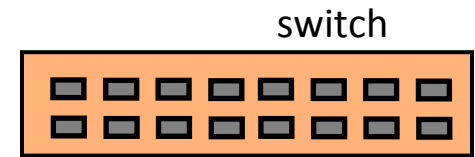
- Network Virtualization:
 - Optimize efficient use of resources
 - Multiple architectures, protocols & systems
- Slicing of network substrate
- Dynamic allocation of shared resources:
 - capacity, spectrum, buffers, storage, FIB, CPU, etc



- FlowVisor: a network virtualization layer
- Acts as proxy between switches and controllers
- Ensures slice and traffic isolation

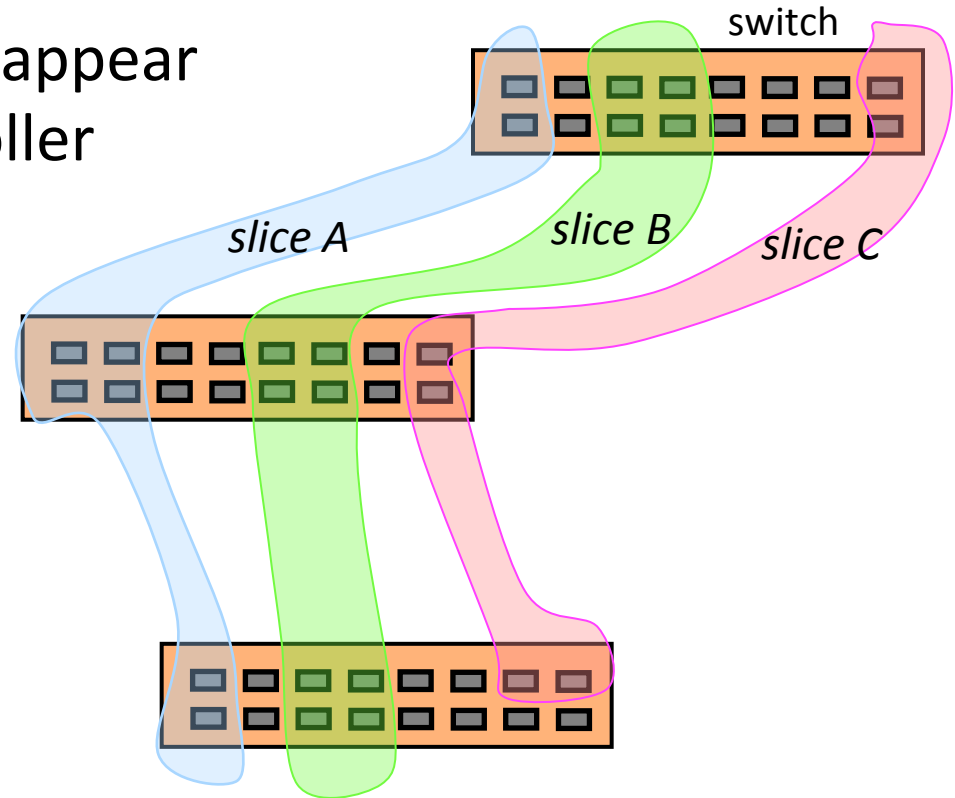
Network Slicing

- A slice can span across several switches (ports)
- Each slice is controlled by a single controller
- Switch's ports on that slice appear as a whole switch to controller
- Creates a virtual topology
- Nested implementation:
 - *multiple flowvisors*



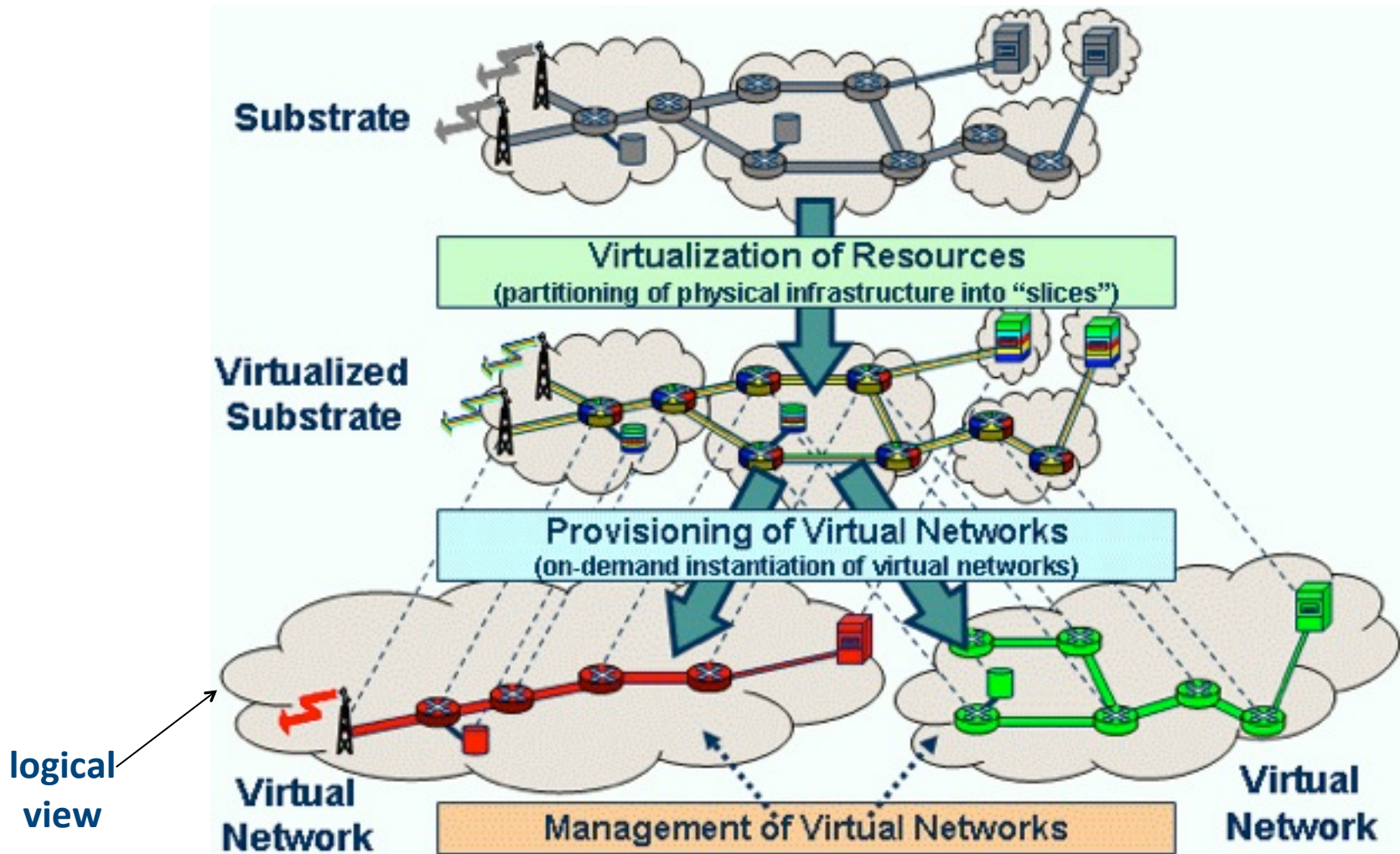
Network Slicing

- A slice can span across several switches (ports)
- Each slice is controlled by a single controller
- Switch's ports on that slice appear as a whole switch to controller
- Creates a virtual topology
- Nested implementation:
 - *multiple flowvisors*



Network Virtualization could be a “killer app” for OpenFlow

Network Virtualization



[source: The 4WARD Project]

Use Cases

- Routing (your own), no STP
- Traffic Off-Loading
- Mobility Management, wireless roaming
- Load Balancing
- Video distribution
- Traffic/Flow Management, Traffic Engineering
- Policy/Security Management
- IPv6: rewrite IPv4 headers into/from IPv6
- Power Management (ElasticTree)

Is the God Box real and finally here?

What providers care about

- Customer service and service quality (UEx)
- Minimize NetEx (CapEx+OpEx)
- Increase ARPU (more traffic does not mean more \$)
- Minimize downtime, outages
- Performance/delivery. Able to meet SLAs
- Build (enough) redundancy, resiliency
- Security, compliance, regulation
- Improve market penetration, offer new (customer-oriented) services, enter new markets
- Stay ahead of the competition, innovate

Bandwidth Control

- Allocate capacity (bps) per:
 - user
 - application
 - device
- Intelligent traffic off-loading (wireless, femtocells)
- Bandwidth scheduling
 - who gets what, when & for how long
- Physical Layer scheduling, e.g., who is using a radio channel and when



Home Networks

- Bandwidth control
- Security control and policy enforcement
- Roaming between devices and networks (e.g., WiFi, 4G, WiMax)
- Smart home/residential gateway
 - managed by provider
 - managed by user
- All-connected home
 - multiple devices
 - multiple user profiles
 - different needs



Home Networks

- Bandwidth control
- Security control and policy enforcement
- Roaming between devices and networks (e.g., WiFi, 4G, WiMax)
- Smart home/residential gateway
 - managed by provider
 - managed by user
- All-connected home
 - multiple devices
 - multiple user profiles
 - different needs



Network Operations

- Network optimization
 - Management, monitoring
 - Security
 - Discovery of network functions & capabilities
- Data Center and Virtual Data Centers
 - An OS for the data center?
 - Computing, storage, networking, apps
 - Flattened data center (L2)
- IT/Production Network
- Could SDN alter the relationship between plane and network management



Cloud Networking

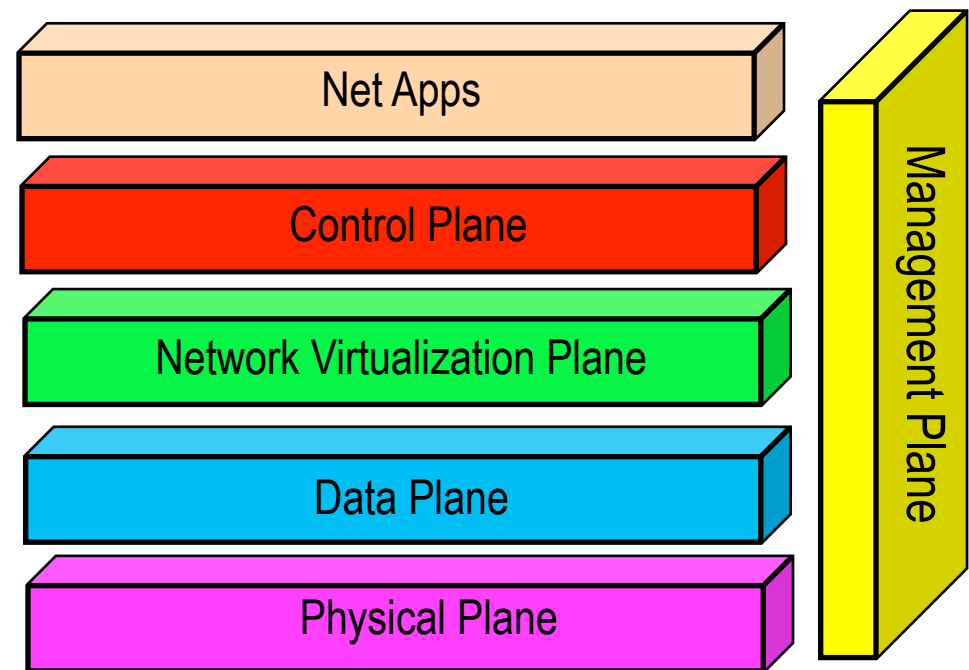
- Allocate on-demand networking & computing resources
- individualized, customizable network/service slices
- Flexible, dynamic deployment of services
- Ease of management
- HyperCloud: cloud services based on network virtualization
- Pay as you go and as you grow
- OpEx vs CapEx (owning equipment)

What would be the role of Network Management

- Further integration/orchestration with control plane?
 - administration
 - security
 - provisioning
 - monitoring
- Out-of-band or in-band?
- Run across full stack?
- Support for network virtualization

What would be the role of Network Management

- Further integration/orchestration with control plane?
 - administration
 - security
 - provisioning
 - monitoring
- Out-of-band or in-band?
- Run across full stack?
- Support for network virtualization



Net Apps Store

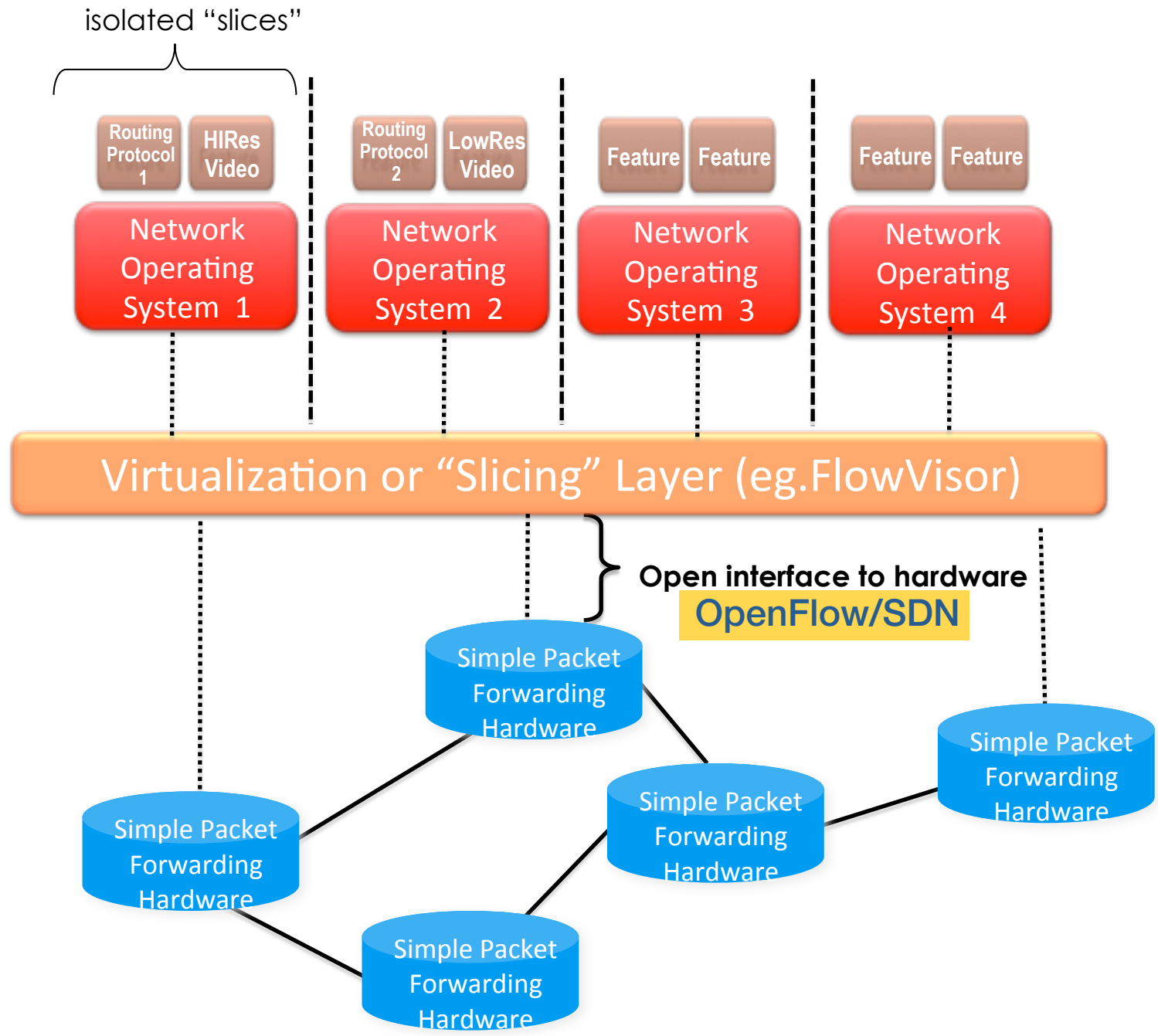
- Create a market for network applications
 - an app for WiFi offload
 - an app for bandwidth control
 - an app for policy control
 - an app for video/content distribution
- Revenue sharing model (telco-3rd parties)
 - Open APIs – network plug-ins
 - Location-based services
 - NaaS; what can operators do with all the intelligence & (user) data ? Deliver network as a service (not a dumb bit pipe).

What OpenFlow/SDN can do for operators

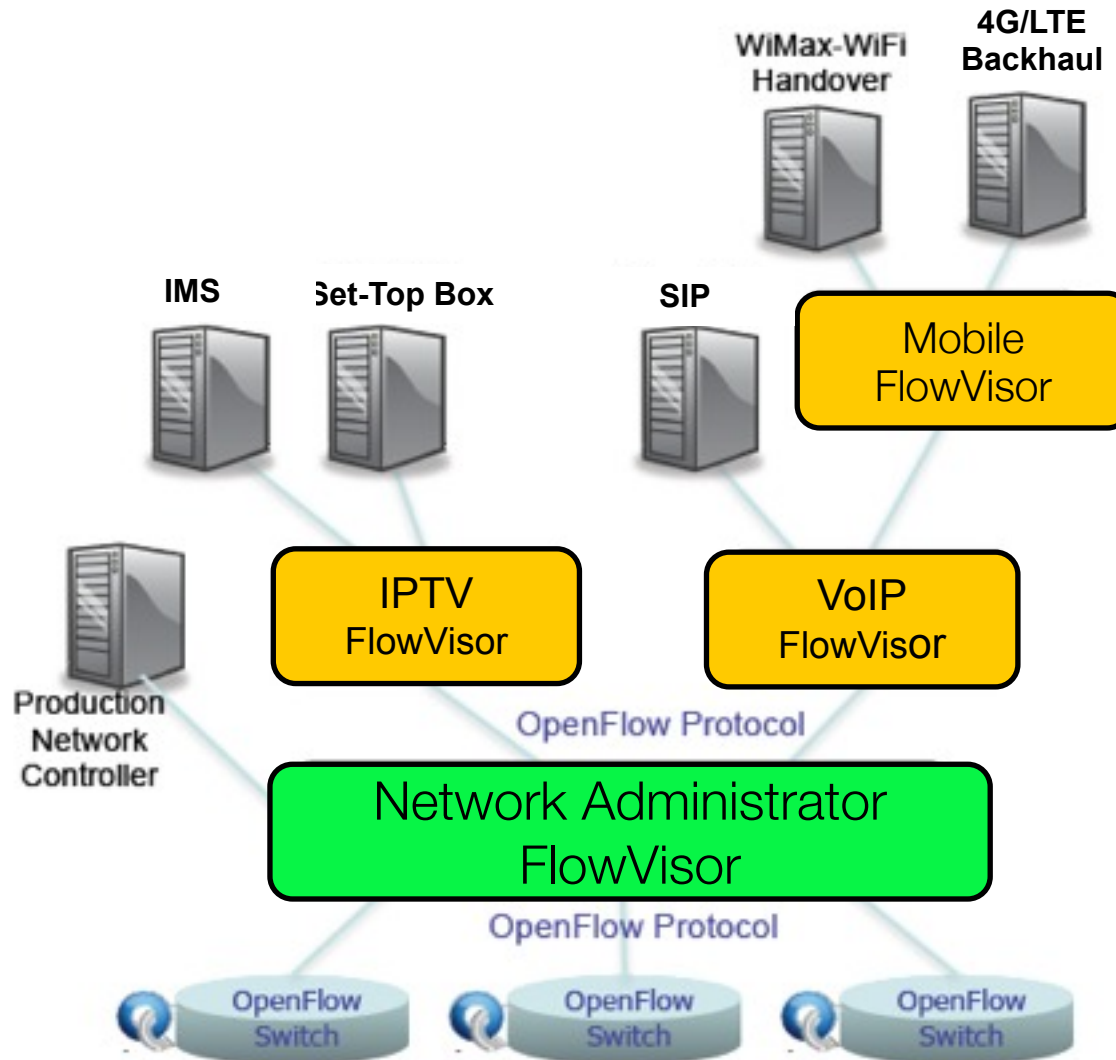
- Make network smarter
- Telco virtualization & telco clouds
- Deploy new (e.g., customizable, dedicated) services and build new business models, e.g., SLAs per slice
- Propel further innovation inside the operator's network
- Can it transform a “white” (commodity) box into a “god” box: a question (and answer) for the vendors
 - iOS vs Android paradigm: closed model (hw+sw) vs open model (3rd party sw)
- In the meantime, no plan to rip and replace existing boxes any time soon. How can OpenFlow complement existing infrastructure?

OpenFlow/SDN

network virtualization



telco network as a sliceable/virtualizable substrate

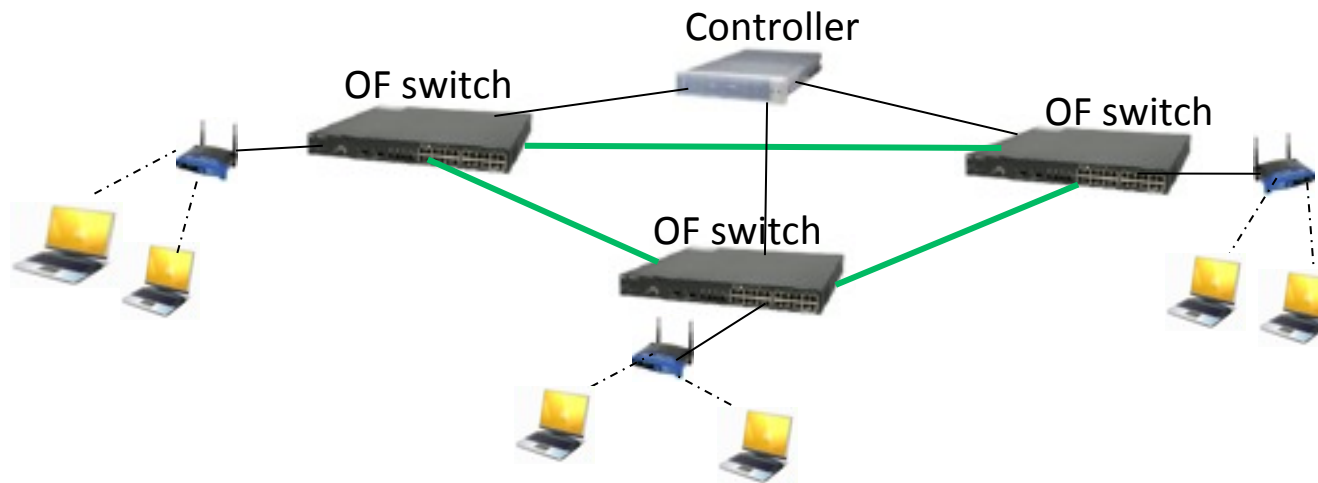


What OpenFlow/SDN can do for users

- Will OpenFlow lower my phone bill?
 - Not necessarily (or at least imminently), but will allow you to make smart choices and make your smartphone smarter and perhaps your home/car too!
- Empower user, decides which application uses which network
 - Provider managed vs. self-managed
- Better (and faster) UEx
(watching movies, shopping online)
- Consumerization of networking



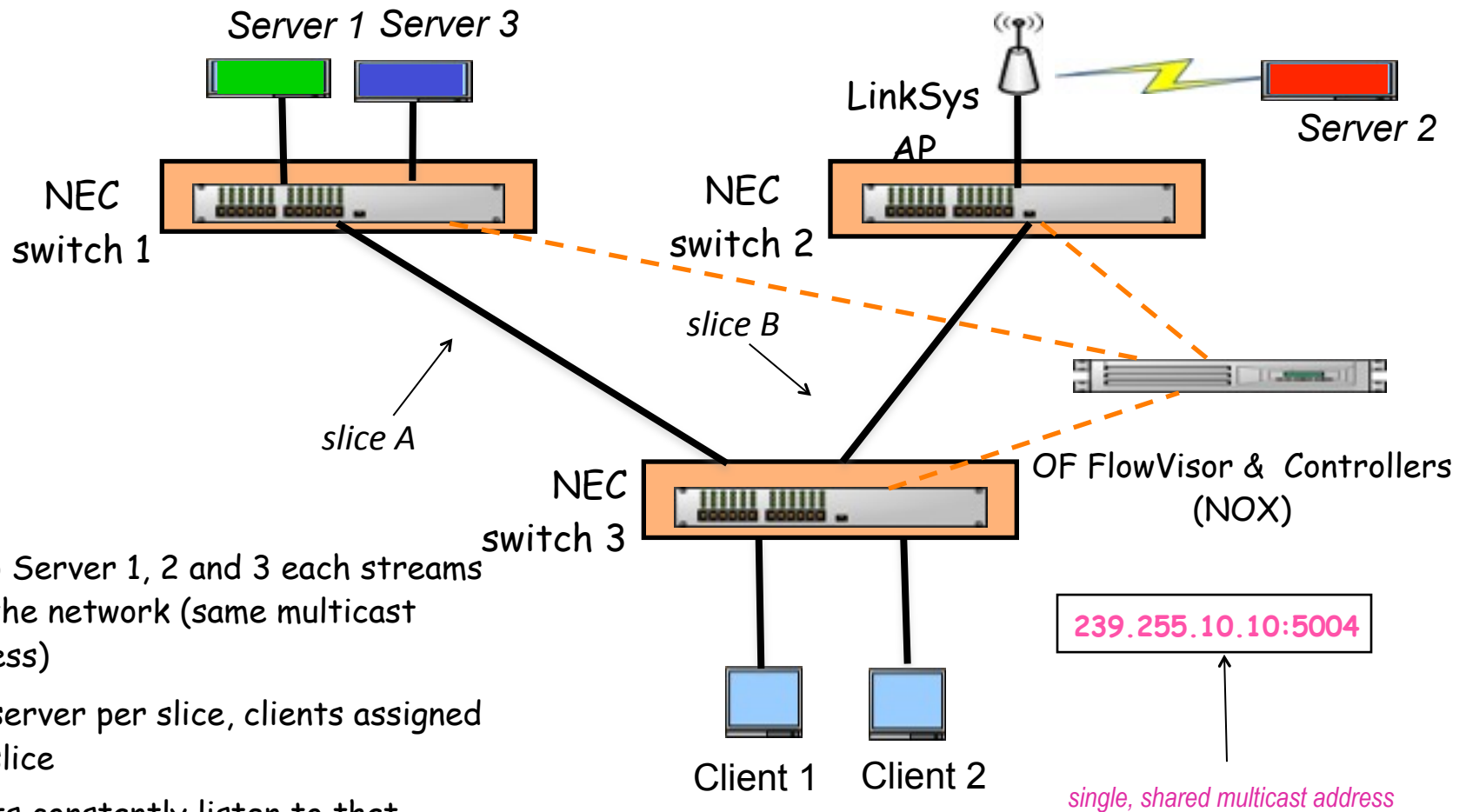
OpenFlow @ Orange Silicon Valley



- OSV leader in OpenFlow project within Orange
- NEC IP8800/S3640-24T2XW switches, LinkSys WRT54GL
- Built own controller (NOX-based)
- PoC (1): "Remote Control" App: taking control of video traffic
- PoC (2): Network Live Migration

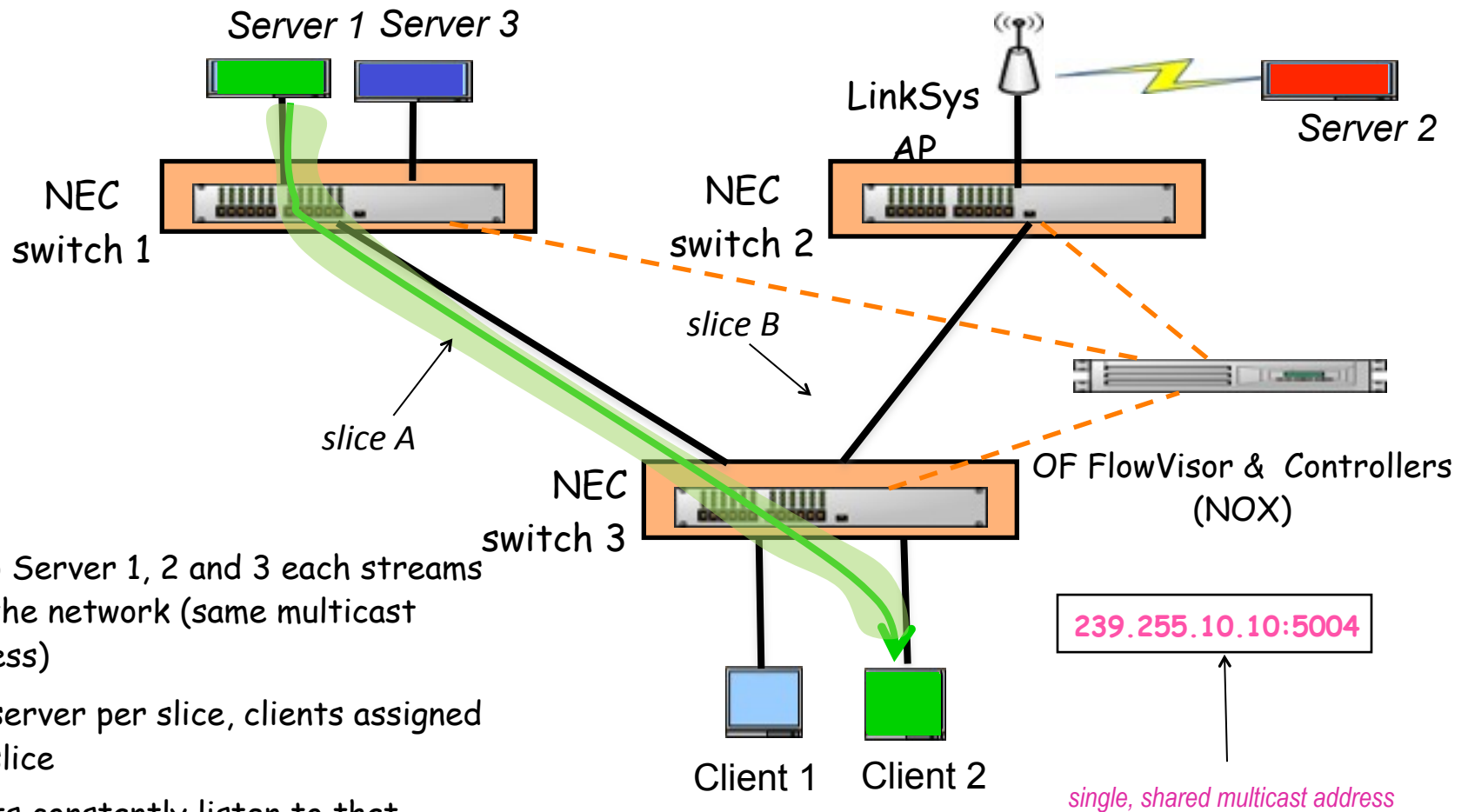


"Remote Control" App



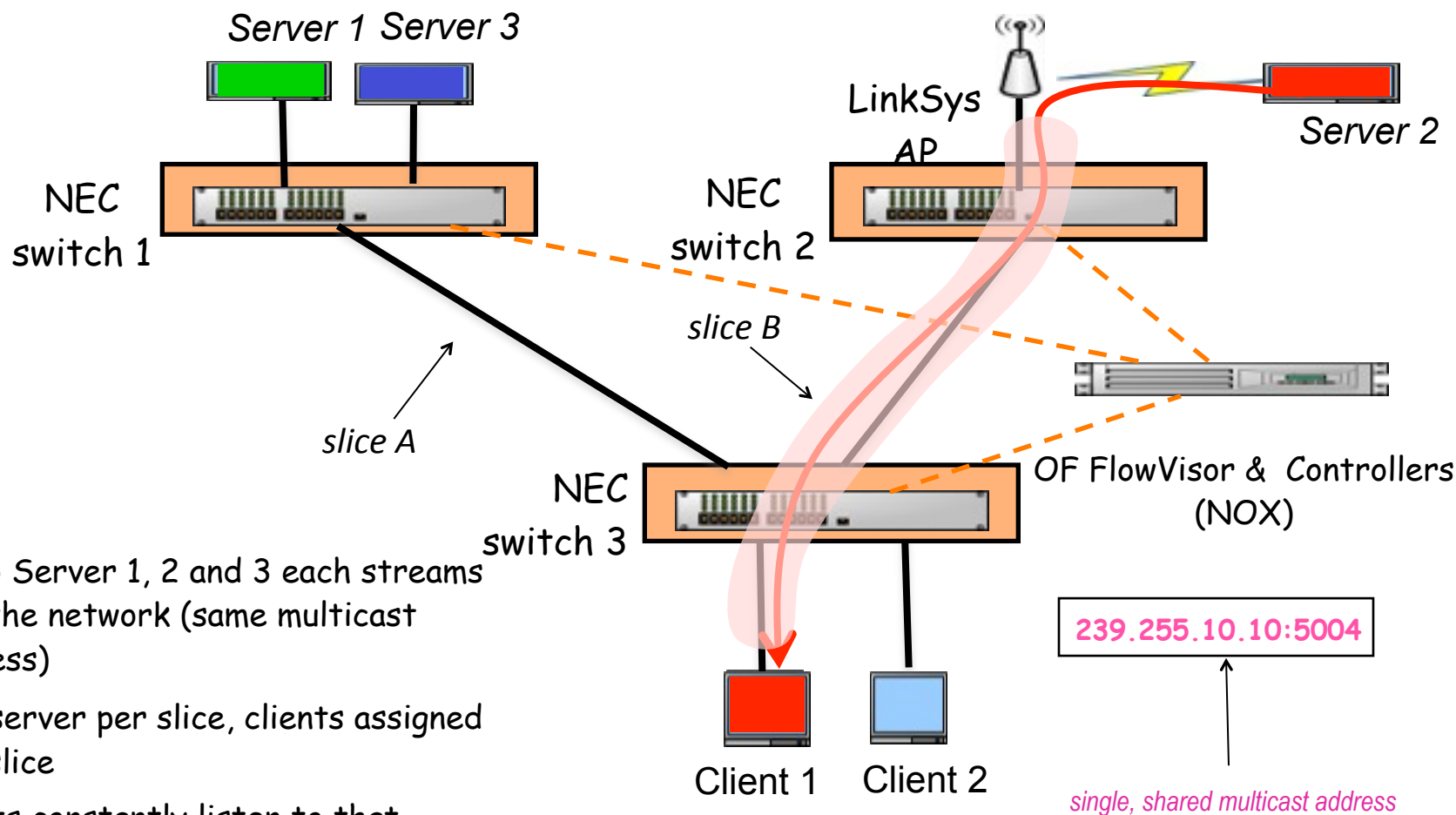
- Video Server 1, 2 and 3 each streams into the network (same multicast address)
- One server per slice, clients assigned to a slice
- Clients constantly listen to that multicast address. Client A receives stream 2 and Client B receives stream 1 (as controlled by switch)
- Define channels (slices) on every

"Remote Control" App



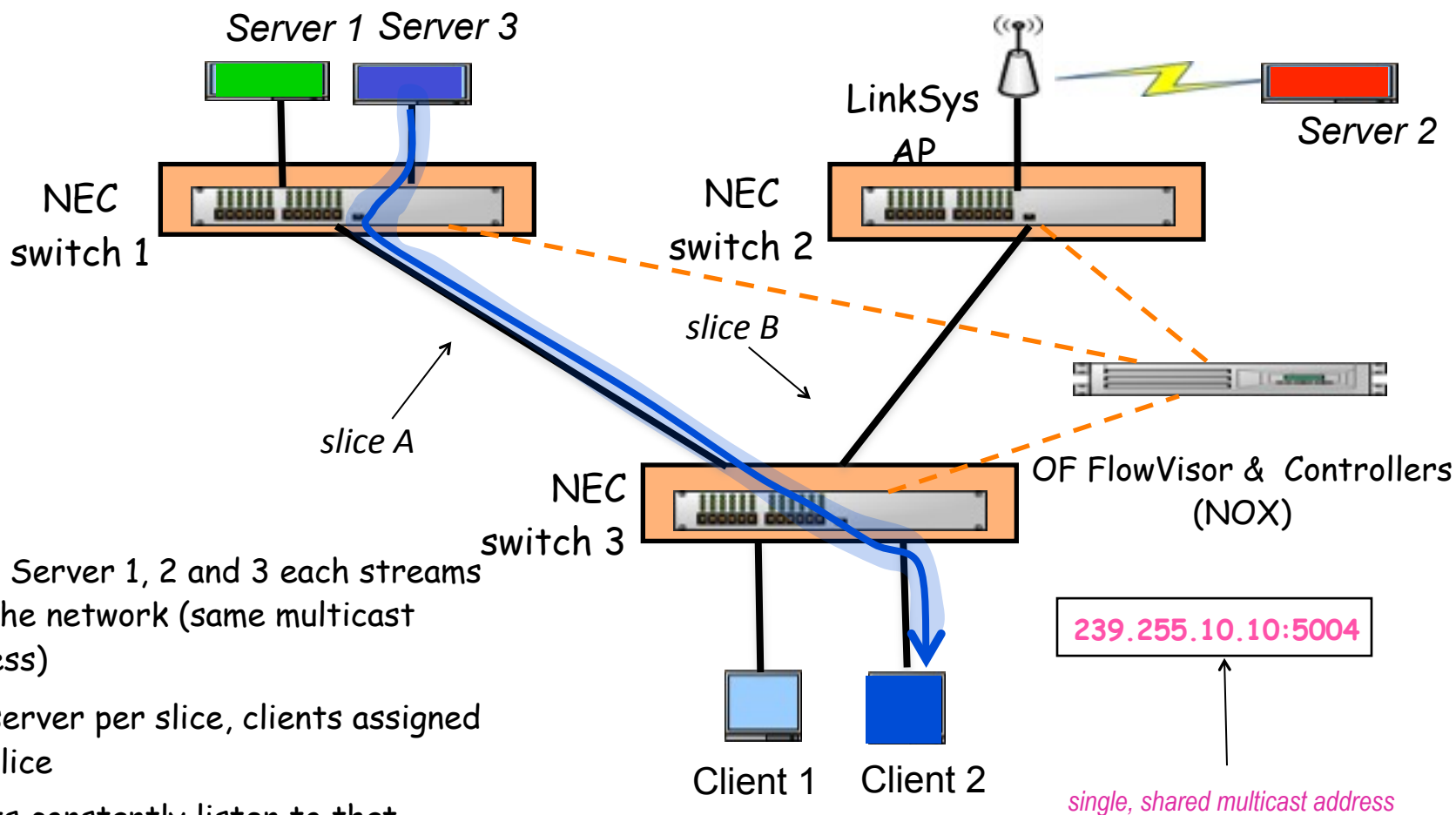
- Video Server 1, 2 and 3 each streams into the network (same multicast address)
- One server per slice, clients assigned to a slice
- Clients constantly listen to that multicast address. Client A receives stream 2 and Client B receives stream 1 (as controlled by switch)
- Define channels (slices) on every

"Remote Control" App



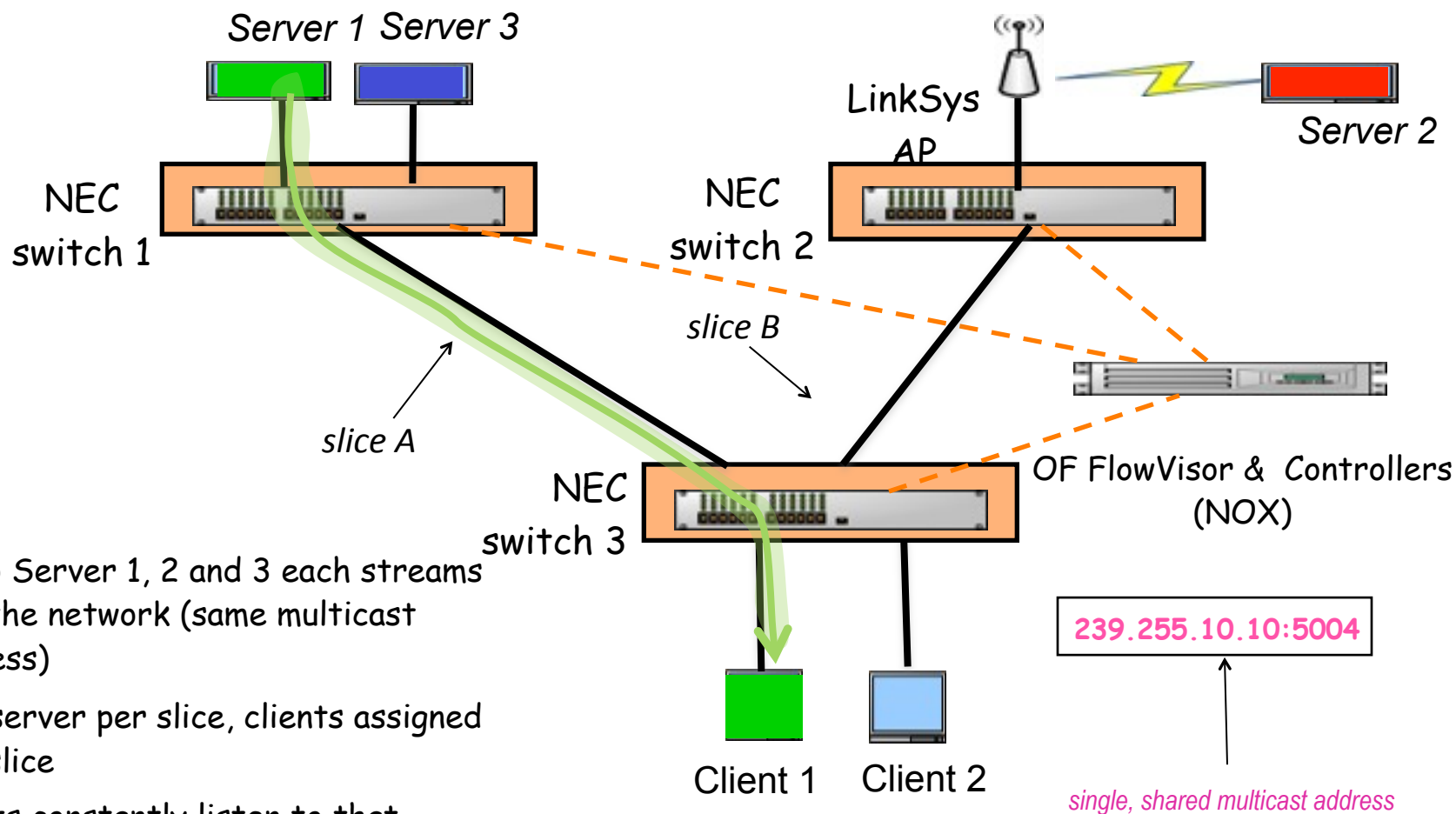
- Video Server 1, 2 and 3 each streams into the network (same multicast address)
- One server per slice, clients assigned to a slice
- Clients constantly listen to that multicast address. Client A receives stream 2 and Client B receives stream 1 (as controlled by switch)
- Define channels (slices) on every

"Remote Control" App



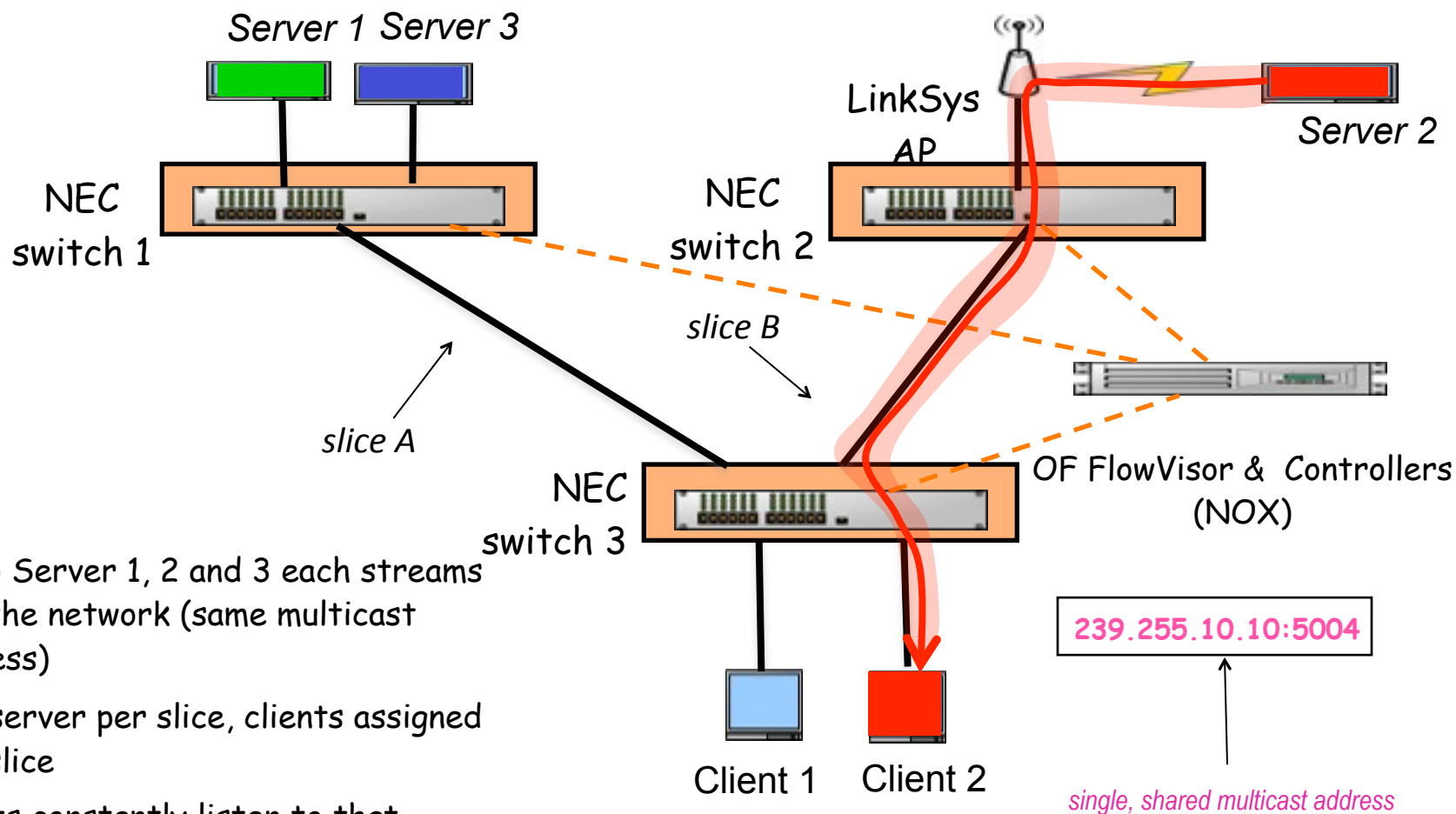
- Video Server 1, 2 and 3 each streams into the network (same multicast address)
- One server per slice, clients assigned to a slice
- Clients constantly listen to that multicast address. Client A receives stream 2 and Client B receives stream 1 (as controlled by switch)
- Define channels (slices) on every

"Remote Control" App



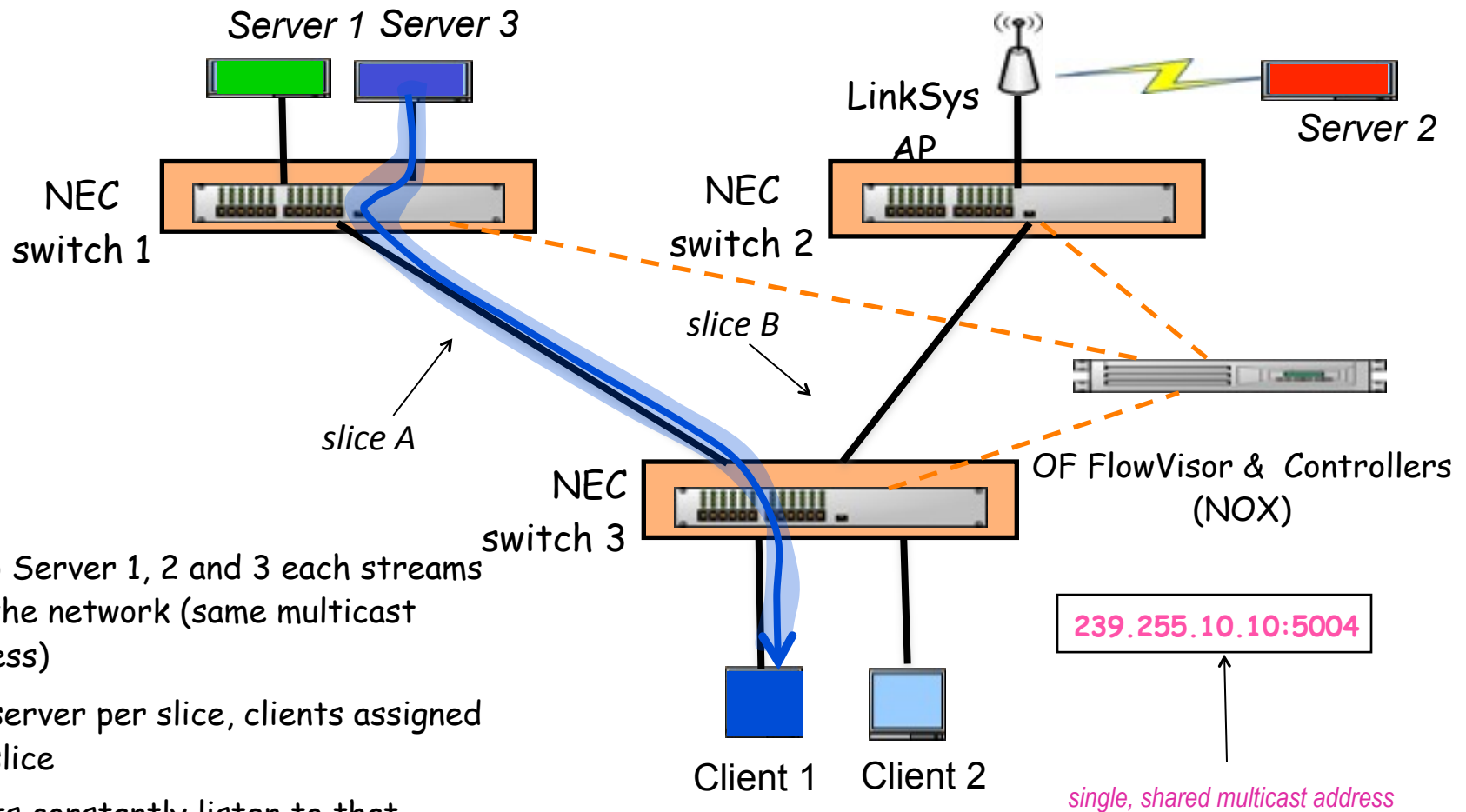
- Video Server 1, 2 and 3 each streams into the network (same multicast address)
- One server per slice, clients assigned to a slice
- Clients constantly listen to that multicast address. Client A receives stream 2 and Client B receives stream 1 (as controlled by switch)
- Define channels (slices) on every

"Remote Control" App



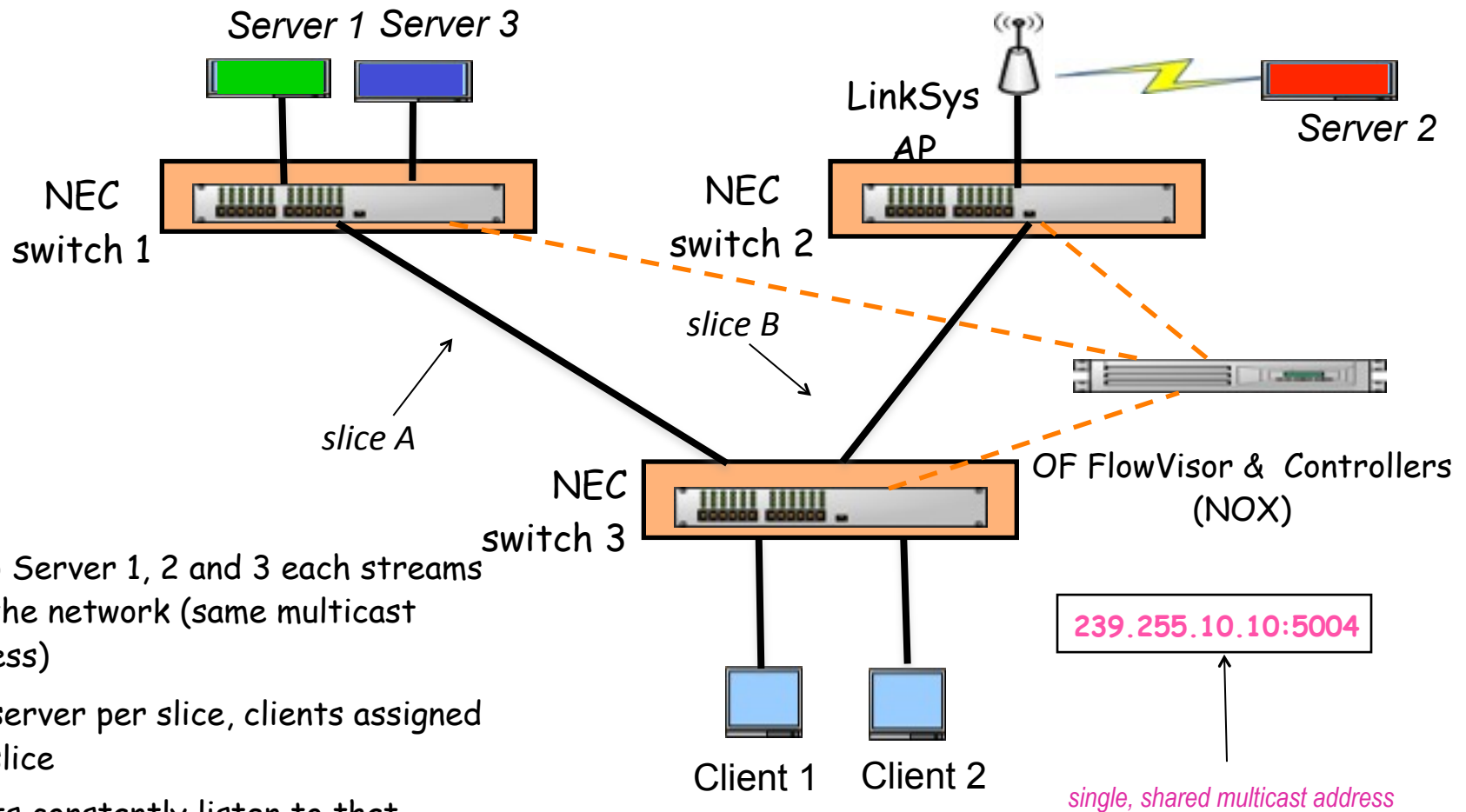
- Video Server 1, 2 and 3 each streams into the network (same multicast address)
- One server per slice, clients assigned to a slice
- Clients constantly listen to that multicast address. Client A receives stream 2 and Client B receives stream 1 (as controlled by switch)
- Define channels (slices) on every

"Remote Control" App



- Video Server 1, 2 and 3 each streams into the network (same multicast address)
- One server per slice, clients assigned to a slice
- Clients constantly listen to that multicast address. Client A receives stream 2 and Client B receives stream 1 (as controlled by switch)
- Define channels (slices) on every

"Remote Control" App



- Video Server 1, 2 and 3 each streams into the network (same multicast address)
- One server per slice, clients assigned to a slice
- Clients constantly listen to that multicast address. Client A receives stream 2 and Client B receives stream 1 (as controlled by switch)

• Detach channels (slices) on every

Conclusion

- OpenFlow/SDN is about:
 - Network Programmability
 - Network Virtualization
 - Intelligent Flow Management
- Ecosystem: controller providers, chip vendors, system vendors, virtualization/cloud vendors
- OpenFlow/SDN opens a door to a new world – a revolution or simply an evolution?
- Can redefine network architecture
 - BYOR: build your own router

thank you!

q & a

christos.kolias@orange.com

Orange, the Orange mark and any other Orange product or service names referred to in this material are trade marks of Orange.
All other trademarks and copyrights are the property of the respective owners.
© Orange – France Telecom Group.