Could IXPs Use OpenFlow To Scale?

Ivan Pepelnjak (ip@ipSpace.net)
Chief Technology Advisor

NIL Data Communications

Disclaimer

The presentation describes *potential future solution* that could be implemented on existing hardware, not features of an actual product.

Products or vendors mentioned in the presentation are not endorsed nor criticized. They just happen to have OpenFlow-enabled products.

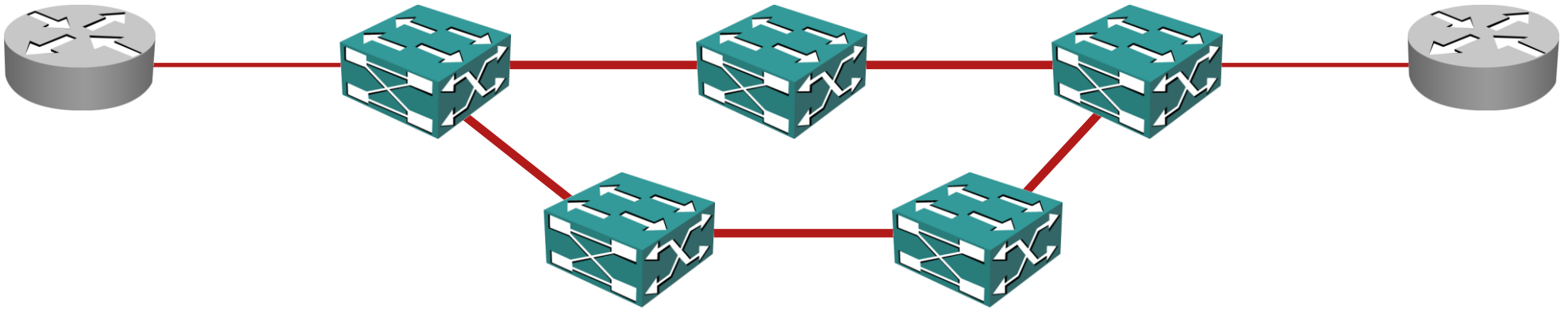
Who is Ivan Pepelnjak (@ioshints)

- Networking engineer since 1985
- Focus: real-life deployment of advanced technologies
- Chief Technology Advisor @ NIL Data Communications
- Consultant, blogger (blog.ioshints.info), book and webinar author
- Teaching “Scalable Web Application Design” at University of Ljubljana

Current interests:

- Large-scale data centers and network virtualization
- Networking solutions for cloud computing
- Scalable application design
- Core IP routing/MPLS, IPv6, VPN

Typical IXP Architecture

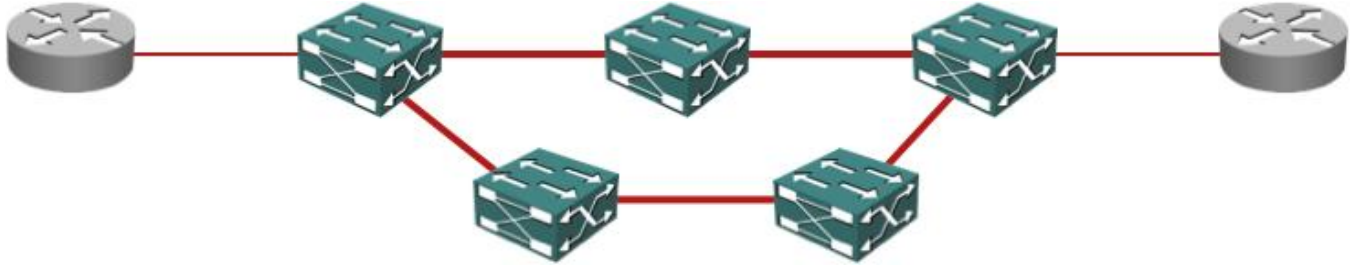


- Third-party routers connected to layer-2 IXP infrastructure
- Large, high-speed, geographically dispersed L2 infrastructure
- High-impact environment
- No influence on third-party device configuration or behavior
  ➔ tight control on network edge

IXP Menu of Pain

IXPs worldwide share the same challenges:

- Large L2 domains
- Need for equal-cost or unequal-cost multipath
- (optional) Traffic engineering
- Source MAC- and IP address control
- Tight protocol control (IP+ARP only)
- Broadcast storm control
- ARP storms



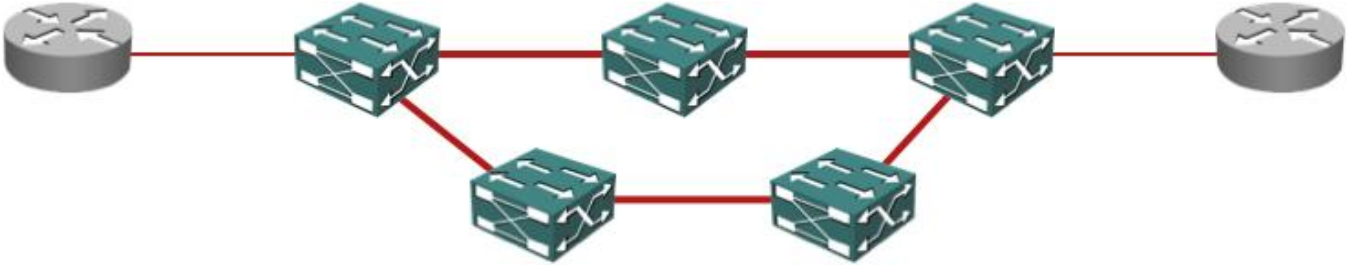
… And the Vendor Response Is …

Use our new shiny toys

- VPLS ➔ SPB/TRILL ➔ Ethernet VPN ➔ …

Unsolved problems remain:

- Little integration with provisioning tools
- Edge control is vendor/product specific
- No reduction in ARP traffic (apart from “ND will work better”)
- Plenty of opportunity for fat fingers



Could we solve the problem using OpenFlow?

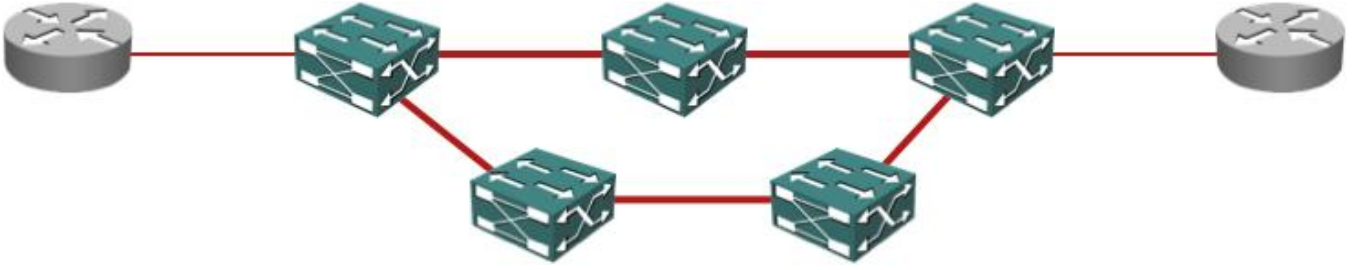
Requirements

Edge behavior:

- Single (or few) MAC addresses per port (dynamic)
- Single IP address per port (from provisioning system)
- L2 protocol filters (allow IPv4, IPv6, ARP)
- Port shutdown on BPDU receipt
- No broadcasting ➔ ARP proxy
- (Optional) ARP sponge

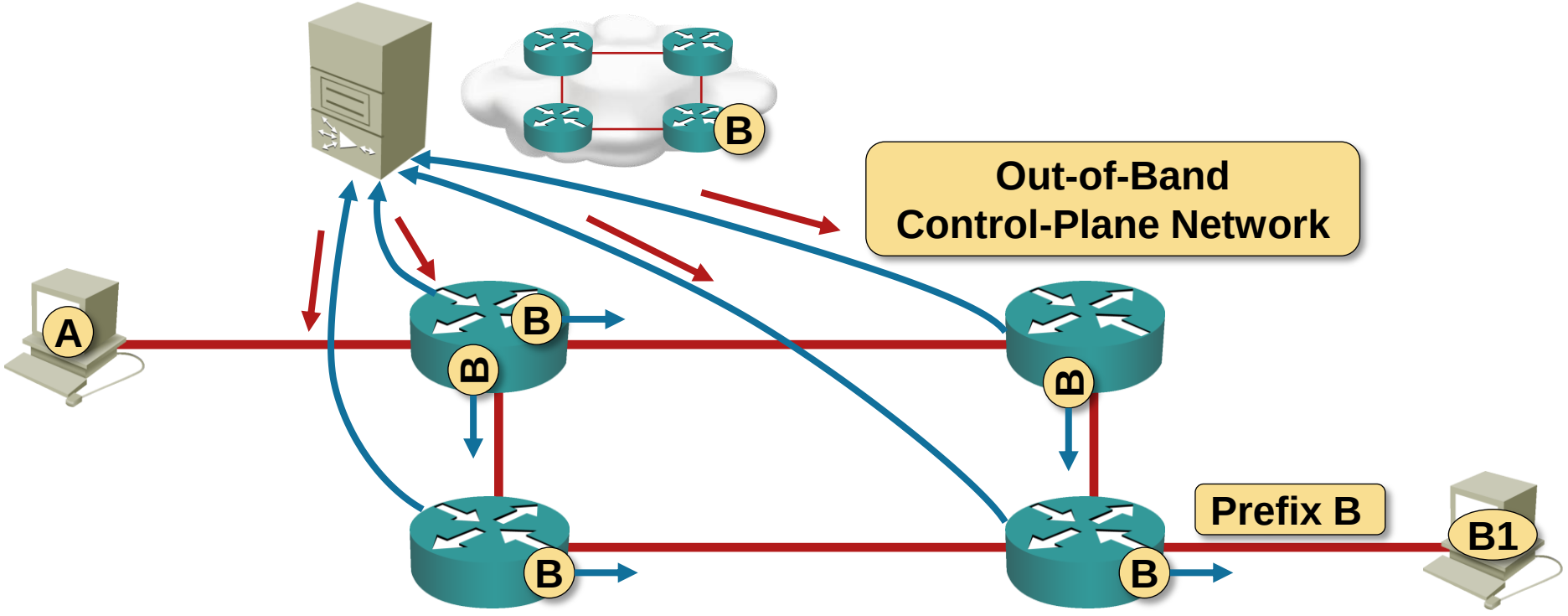
Core behavior

- L2 ECMP
- L2 TE



OpenFlow and Other SDN Tools – Review

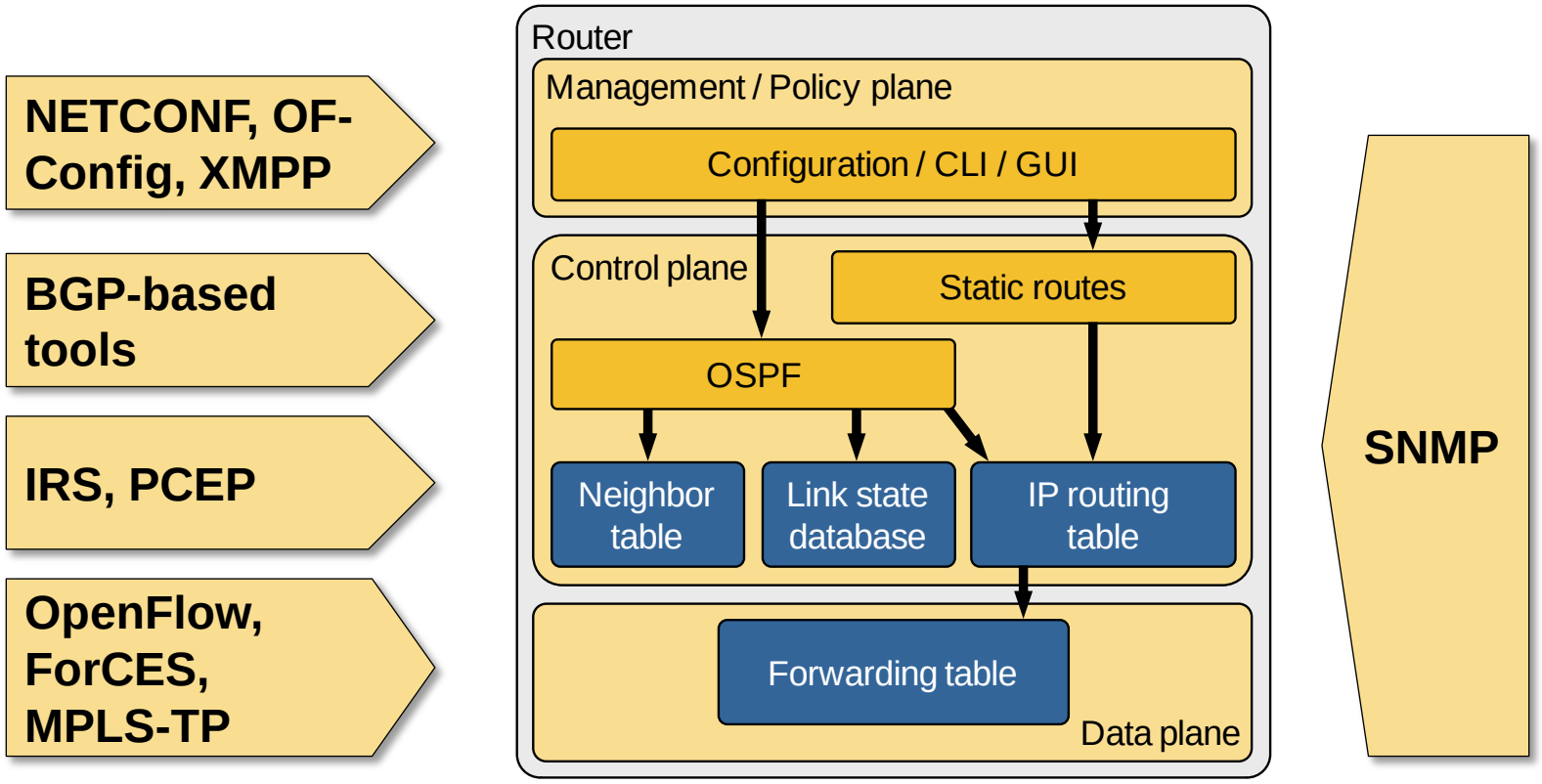
Brief Review of OpenFlow



Basic principles:

- Control / Management plane in a dedicated *controller*
- Networking devices perform forwarding and maintenance functions
- IP / SSL connectivity between controller and OpenFlow switch
- OpenFlow = Forwarding table (TCAM) download protocol

Brief Review of SDN Tools



- Vendor APIs: Cisco, Juniper
- Scripting: Cisco, Juniper, Arista, Dell, F5 ...

Selecting the Right Tool

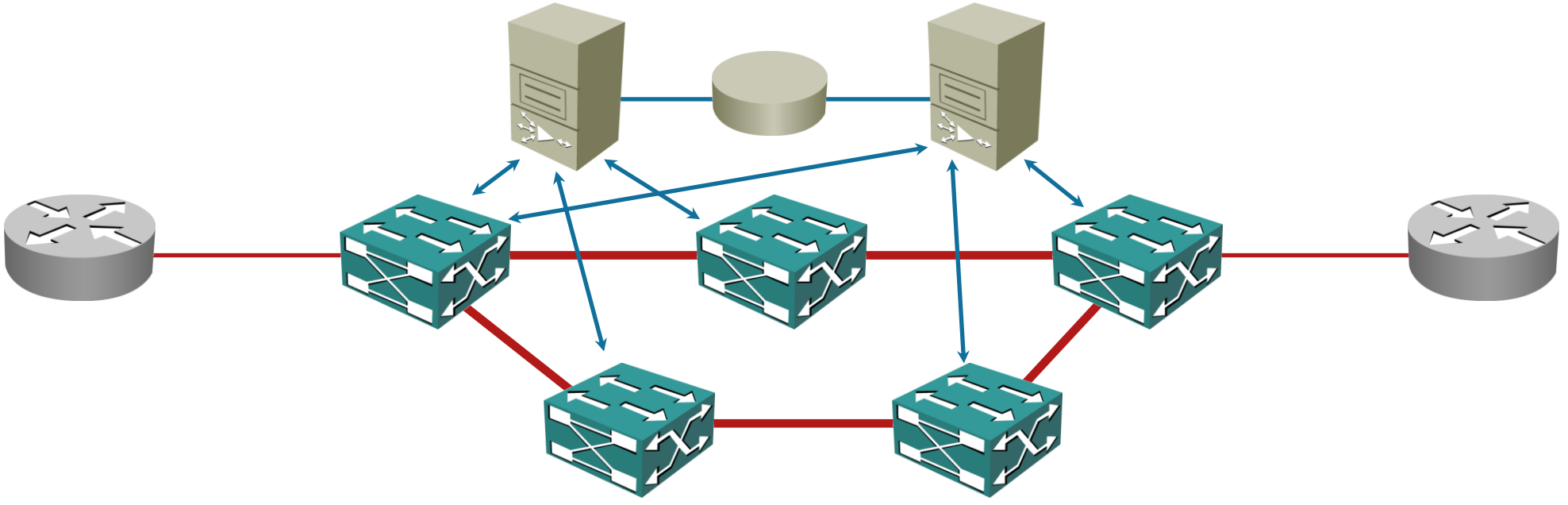
Overview of Existing Implementations

Requirement	Available in existing devices
Single (or few) MAC addresses per port	Yes
Single IP address per port	Yes (L3 port ACLs)
L2 protocol filters (allow IPv4, IPv6, ARP)	Yes
Port shutdown on BPDU receipt	Yes
No broadcasting ➔ ARP proxy	Hard
ARP sponge	Harder

- Most requirements could be implemented with existing devices + management-plane API (e.g. NETCONF)
- ARP proxy requires data plane intercept
- MPLS-TP lacks L3 functionality, ForCES is not widely implemented ➔ OpenFlow is the only viable tool

Proposed Solution

Proposed Architecture

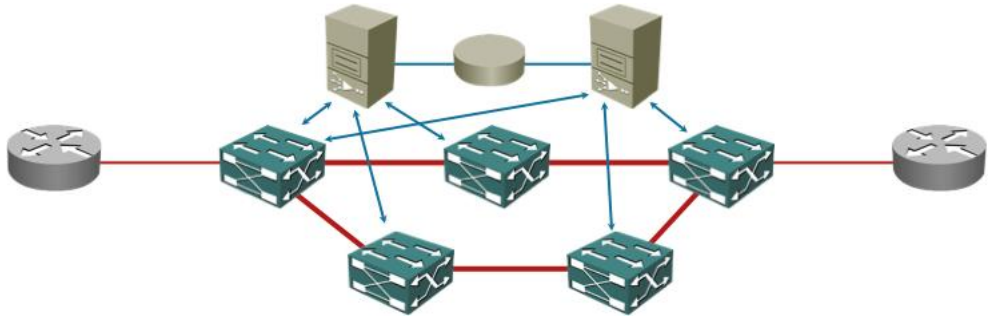


- Cluster of OpenFlow controllers
- Provisioning DB used for IP address information
- Phase 1: Edge functionality, *NORMAL* forwarding in core
- Phase 2: Core OpenFlow

Phase 1: OpenFlow on Edge

Principles

- OpenFlow used to control edge forwarding
- Core transport uses existing technologies

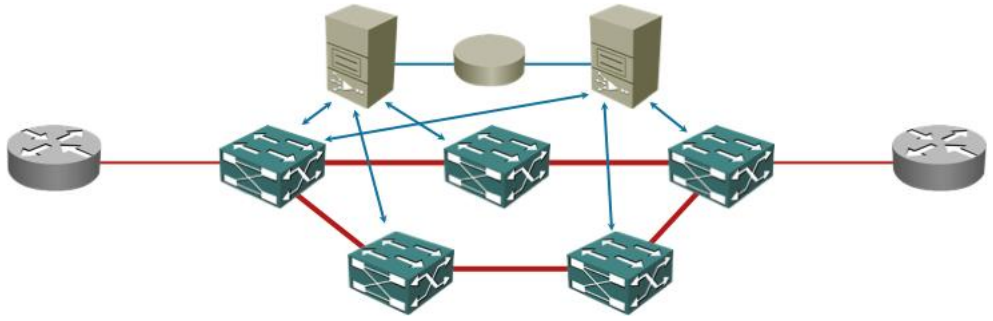


Edge behavior:

- Static (proactive) forwarding entries
- Allow off-link source IPv4+IPv6 addresses + known on-link IPv4+IPv6 address
- Drop traffic sent to *sponge* MAC address
- Forward IPv4+IPv6 packets from known source MAC address to *normal* processing
- Forward ARP traffic to controller (when ARP proxy is enabled)
- Forward STP BPDU to controller
- Drop all other traffic

Phase 1: Port Initialization

- Switch notifies controller on port status change
- Forwarding entries are automatically removed
- Switch forwards all inbound traffic to controller



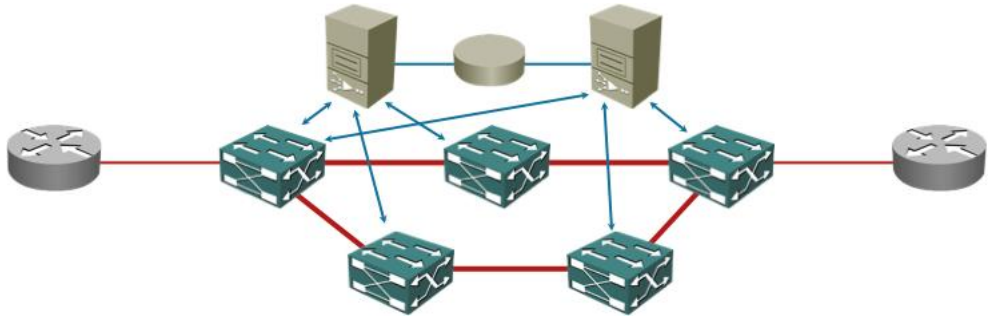
Controller port initialization behavior:

- Send ARP/ND request for expected IPv4+IPv6 address
- Use source MAC address in ARP/ND reply in L2 traffic filter
- Update ARP/ND cache
- Download forwarding entries to the switch

Phase 1: ARP Proxy

Principles:

- ARP processing done by OpenFlow controllers
- No end-to-end broadcasts



ARP proxy behavior

- IP-to-MAC bindings discovered during port initialization
- ARP information stored in scale-out database
  ➔ zero state in switches or OpenFlow controller
- OpenFlow controllers reply to ARP requests

ARP sponge (optional)

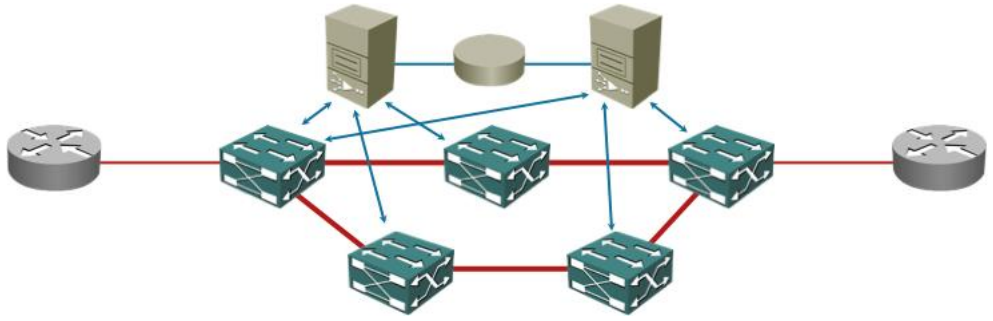
- Controller(s) perform IP address liveliness tests
- *Sponge* MAC address used in replies for unknown or dead IP addresses

ND behavior for further study

Phase 1: Gradual Deployment

Principles:

- Do not interfere with IXP operation
- Incremental OpenFlow deployment



Step 1 – deploy edge filters

- Implementable on individual switches
- Does not impact end-to-end forwarding
- Does not required OpenFlow on all switches
- Immediate results: stop misconfigurations and undesired protocols

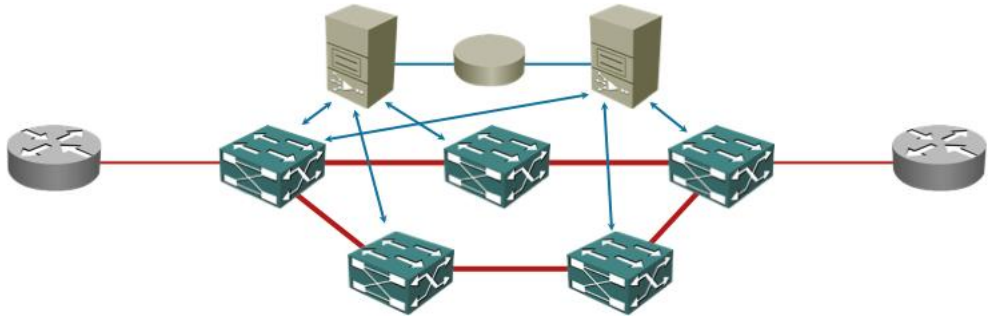
Step 2 – ARP proxy/sponge

- Deployed when all edge switches support OpenFlow
- Still no interference with core forwarding

Phase 1: Distributed System

Principles:

- No state in OpenFlow controllers
- All information in scale-out database
- Controller cluster per site



Benefits

- System survives partitioning

Drawbacks

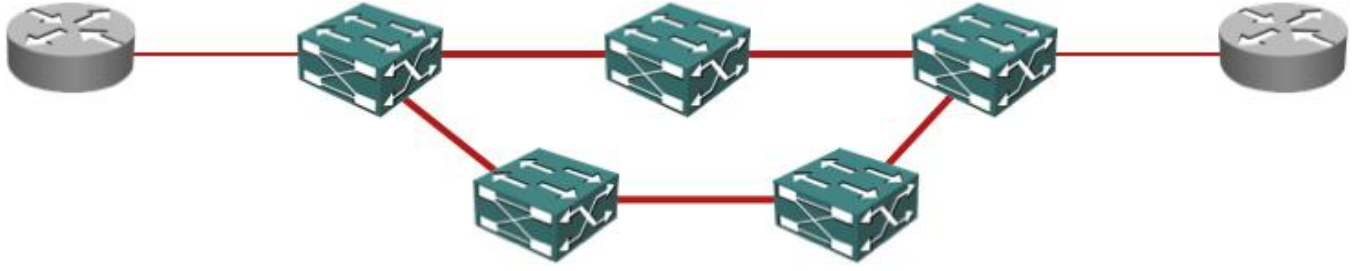
- Eventual consistency (MAC, ARP information only)

Phase 2 – End-to-End OpenFlow

- If needed, implement ECMP L2 core with OpenFlow
- Use existing switch hardware (if OpenFlow-enabled)
- Add L2 ECMP or TE functionality with OpenFlow controller

Potential existing products:

- Floodlight or Beacon open-source controllers
- ProgrammableFlow from NEC



Orders-of-magnitude harder than Phase 1

Conclusions

We could solve unique IXP problems with OpenFlow

Phased approach:

- OpenFlow at edges for IP+MAC validation
- ARP proxy and ARP sponge implemented with OpenFlow controller
- ECMP L2 core implemented with OpenFlow

Solution based on open source controllers (e.g. Floodlight, Daylight)

Many Thanks!

Brent Salisbury

Sander Steffann

Pierre Francois

Questions?

Send them to ip@ipSpace.net or @ioshints