Minimizing ARP traffic in the AMS-IX switching platform using OpenFlow

Victor Boteanu Hanieh Bagheri

University of Amsterdam
System and Network Engineering

July 3, 2013

Research Questions

"Can OpenFlow be used to reduce broadcast ARP traffic in the AMS-IX ISP peering LAN?"

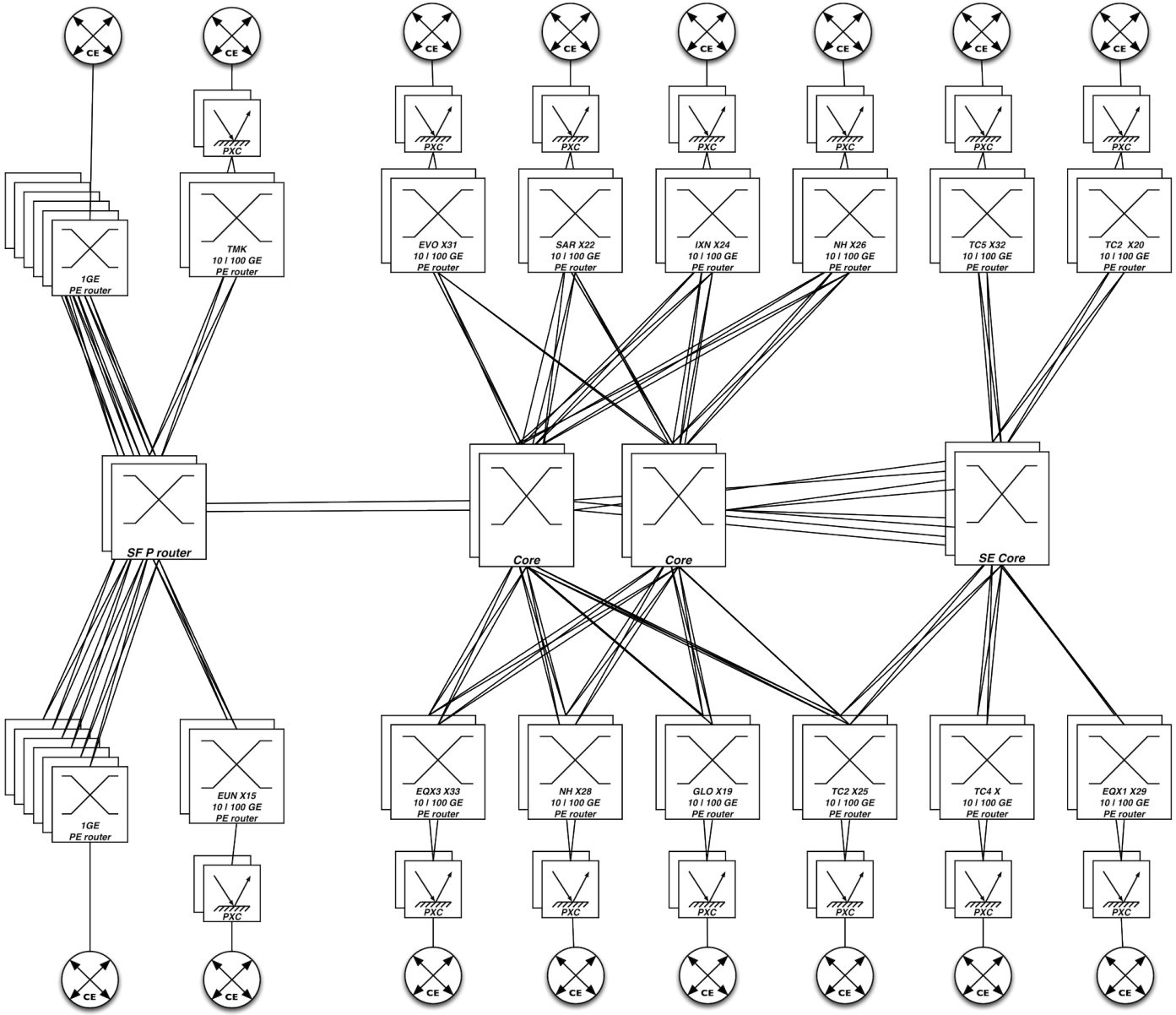
"Can ARP be replaced completely by OpenFlow in the core network?"

"Is OpenFlow a scalable solution for this scenario?"

Amsterdam Internet Exchange (AMS-IX)

- One of the largest IXPs in the world
- 600 ASNs, 1132 Ports
- Average traffic of 1.47 Tb/s, peak traffic of 2.25 Tb/s
- Capacity: 8.81 Tb/s
- MPLS/VPLS-based peering platform (Using Brocade hardware)
- Services: 1 GE, 10 GE, 100 GE, or multiples of these values

AMS-IX Peering Platform

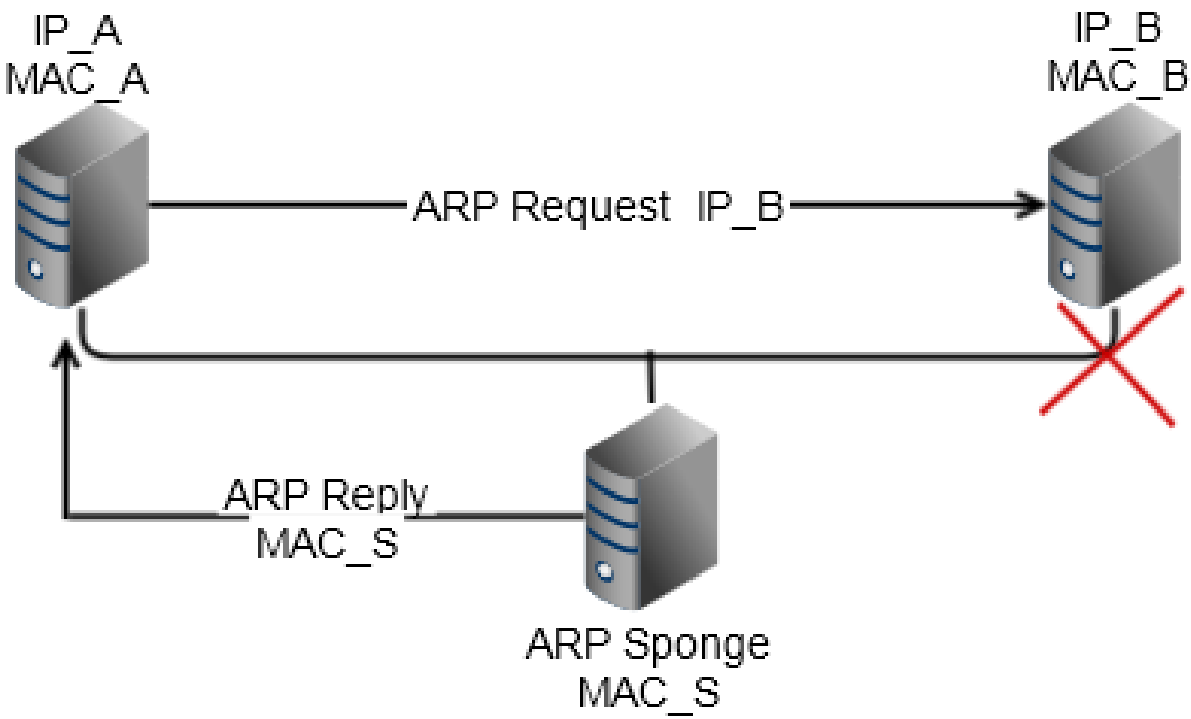


ARP in AMS-IX Peering Platform

- Customer routers use ARP to map IP to MAC addresses
- Broadcast nature of ARP:
  All nodes to get the request; Only one replies
- If no reply is received, keep trying.
  When a node is down $\rightarrow$ ARP storm
- Waste of CPU cycles in routers
- Current solution: ARP Sponge

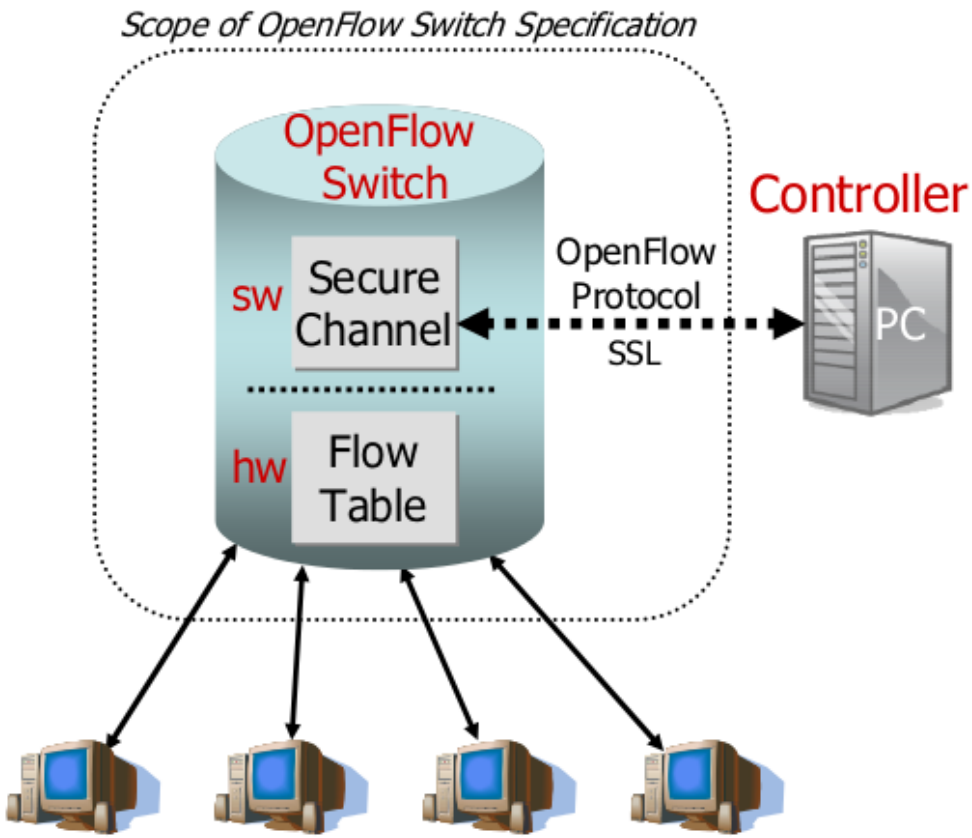
ARP Sponge

- Developed at AMS-IX to solve the ARP storm problem
- To "sponge" ARP requests for dead IP addresses
- It replies to ARPs on behalf of the (likely) dead nodes
- An effective solution, but does not solve the problem completely



OpenFlow

- An open standard for researchers to test new ideas and protocols
- Separation of control plane from data plane
- Architecture:



A Flow Table entry in OpenFlow

- Packet header

In Port	VLAN ID	Ethernet			IP			TCP	
		SA	DA	Type	SA	DA	Proto	Src	Dst

- Action:
  - forward to a specific port (or to all ports)
  - encapsulate and forwarding them to the controller
  - drop
- Statistics

Solutions

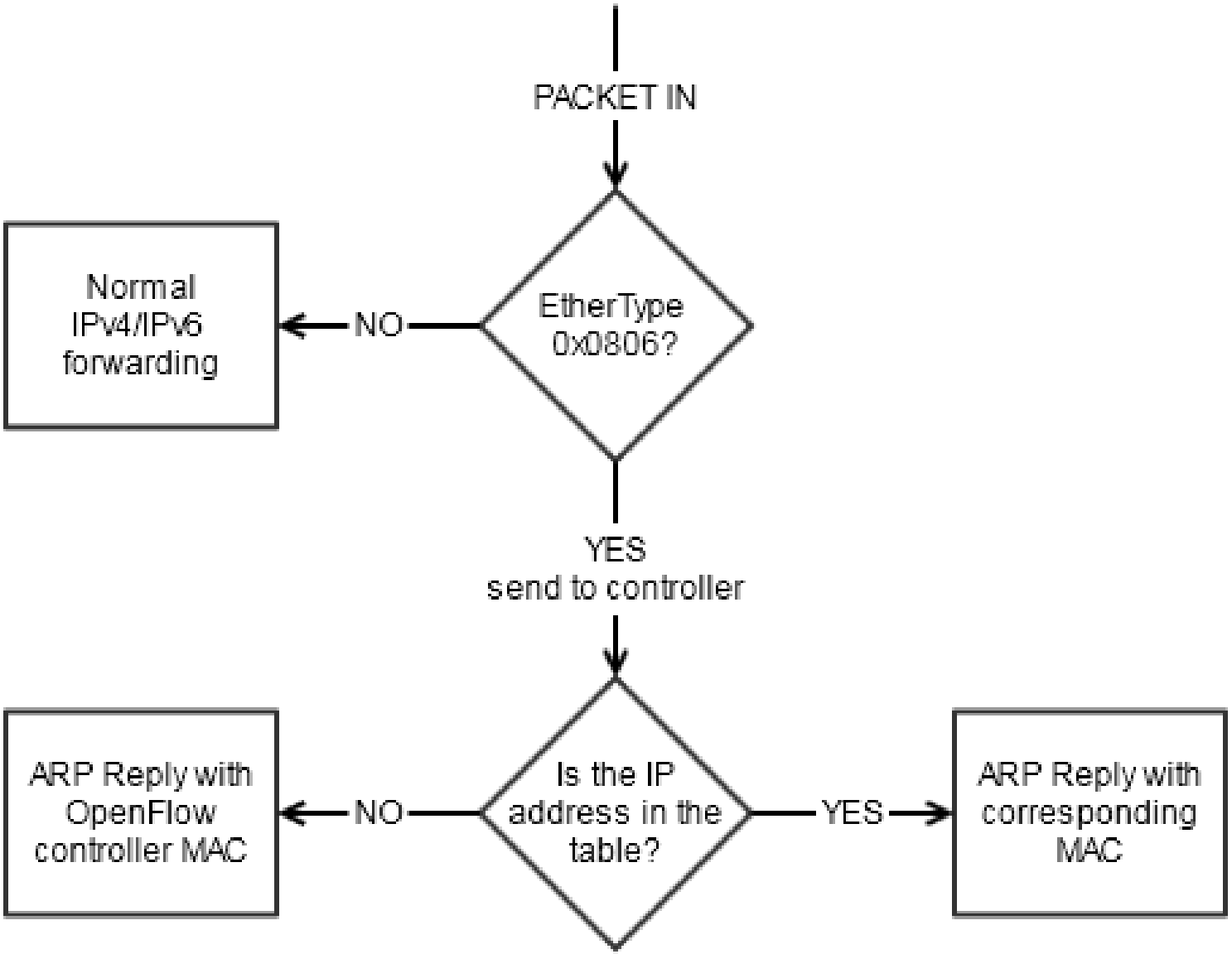
Solution 1

- The mapping between MAC and IP addresses is known beforehand
- Idea: import the mapping table to the OpenFlow controller
- Remove the ARP Sponge; Use the controller as an ARP proxy
- The mapping table is used for replying to the ARP requests
- The ARP requests are not broadcast anymore
- MAC spoofing prevention

Solution 1: steps

1. The mapping table is imported to the OpenFlow controller
2. The controller installs a flow in the switch:
   Flow: forward ARP messages to the controller.
3. The controller uses the table to make an ARP reply for the ARP request
4. The controller sends the ARP reply to the sender of ARP request

Solution 1


PACKET IN
EtherType 0x0806?
NO
Normal IPv4/IPv6 forwarding
YES
send to controller
Is the IP address in the table?
NO
ARP Reply with OpenFlow controller MAC
YES
ARP Reply with corresponding MAC

Solution 2

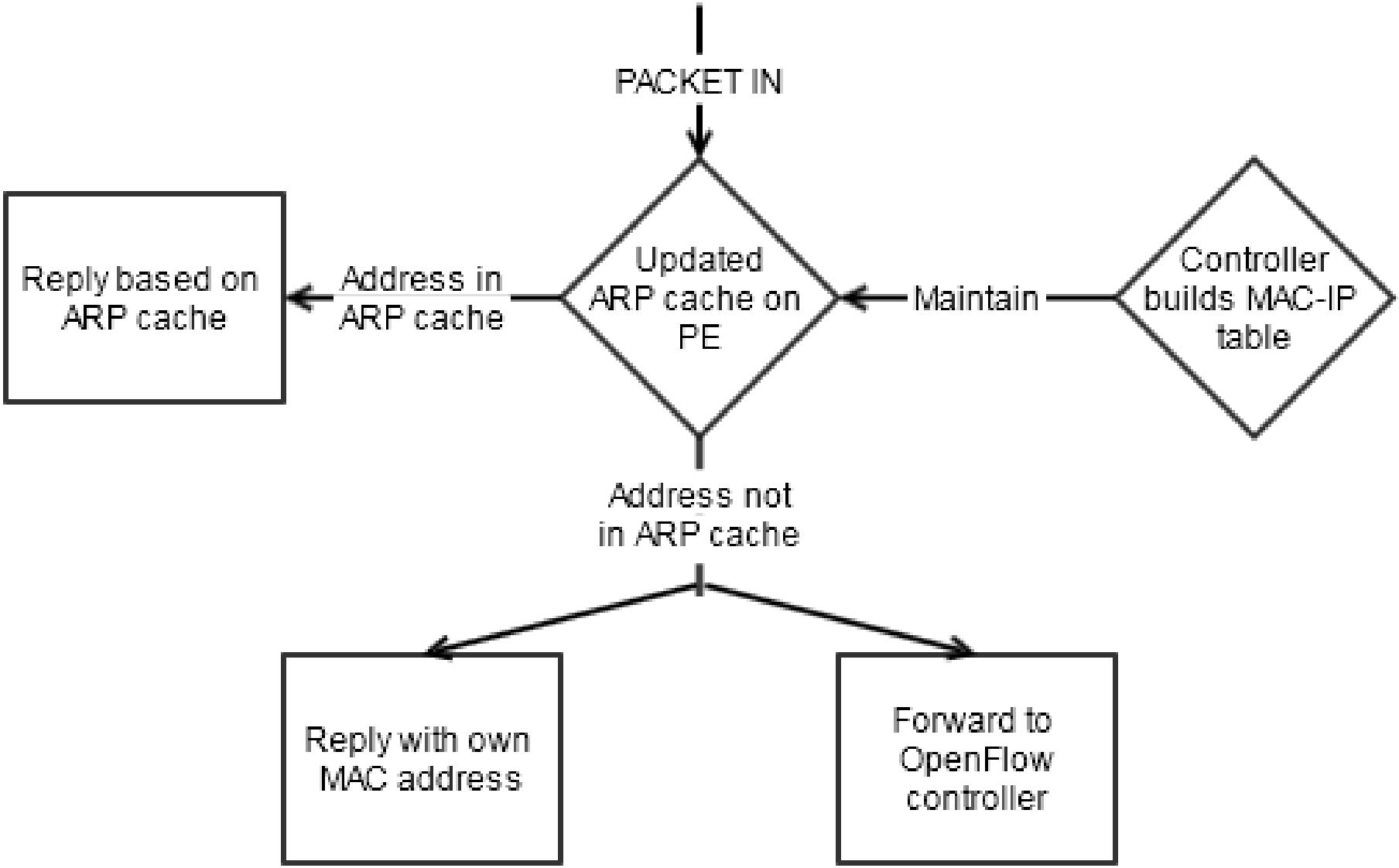
- Similar to solution 1, but more dynamic
- Learn the IP-to-MAC table from the exchanged ARP
- Use ARP messages to update the table:
  ARP Request+ ARP Reply + Gratuitous ARP
- Reply to ARP in case of a hit; otherwise flood the request once
- Latency of learning mechanism
  (for nodes that do not support gratuitous ARP)
- Still vulnerable to MAC spoofing

Solution 3

- Using the updated IP-to-MAC in the controller to update the ARP cache in each PE
- PE switch as ARP Proxy
- Need for special switch/controller support

1. The OpenFlow controller builds an IP-to-MAC table
2. The controller uses this table to update each PE's ARP cache
3. The switches can now respond to ARP requests based on their own ARP caches

Solution 3


PACKET IN
Updated ARP cache on PE
Address in ARP cache
Reply based on ARP cache
Maintain
Controller builds MAC-IP table
Address not in ARP cache
Reply with own MAC address
Forward to OpenFlow controller

Solution 4

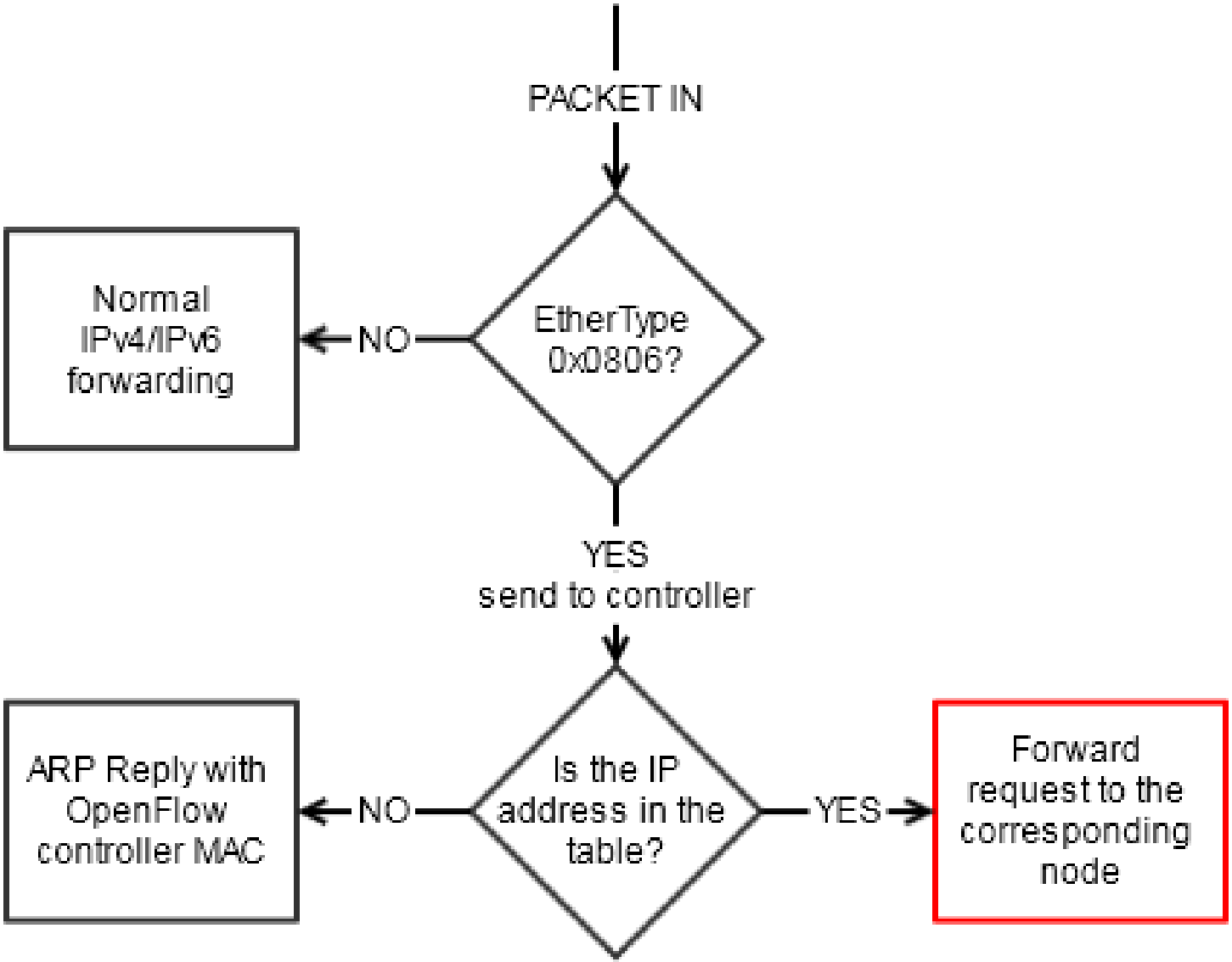
1. The controller installs a flow in the switch to forward every incoming ARP request to the ARP Sponge.
2. The ARP Sponge uses its internal ARP table to answer the ARP requests.
3. The ARP Sponge monitors the network and answers the ARP requests destined for the dead nodes using its own MAC address

The sponge's table needs to be kept updated.
Easier to implement; makes use of a known mechanism.
Learning period for the ARP sponge.

Solution 5

1. Import IP-MAC mapping to the controller
2. The controller installs a flow on the switch to forward all ARP to the controller
3. The controller looks up the destination IP addresses in the table to find the corresponding MAC address
4. If there is a match, it forwards the unicast ARP request to the corresponding MAC address; otherwise, it makes an ARP reply using the MAC address of the controller.
5. If the controller receives a gratuitous ARP from one of the clients, the IP-MAC table is updated

Solution 5

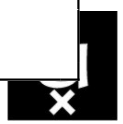

PACKET IN
EtherType 0x0806?
NO
Normal IPv4/IPv6 forwarding
YES
send to controller
Is the IP address in the table?
NO
ARP Reply with OpenFlow controller MAC
YES
Forward request to the corresponding node

Proof of Concept

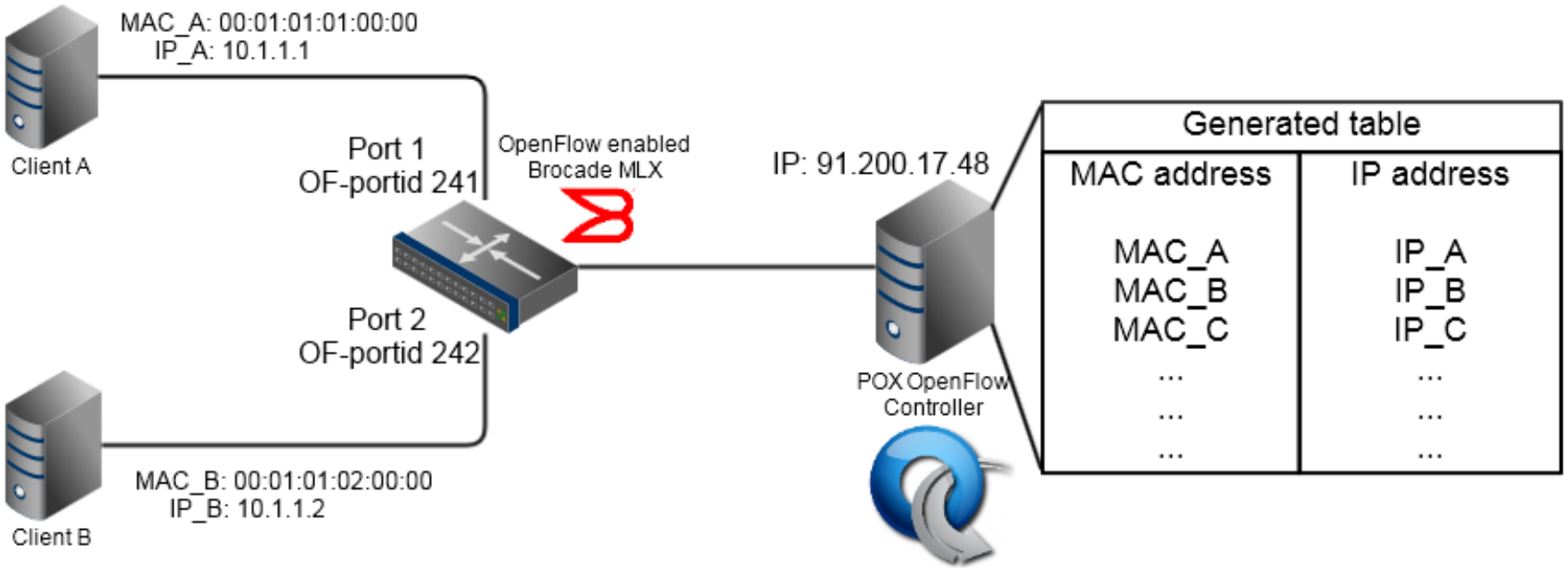
Proof of Concept

- Implementation of solution 1
- Switch = Brocade MLX; OpenFlow (v1) enabled
- Clients simulated using 2 ports from a traffic generator (hybrid-port mode)
- POX OpenFlow controller VM
- MAC-IP table built from an XML file that contains port information

```
<vlan id="501" mode="untagged">
    <mac-address>782b.cb5a.bb68</mac-address>
        <router ipaddr="195.69.145.0" fqdn="rs2.ams-ix.net" asnum
            ="6777">
        <attr id="route-server" value="1"/>
        <peering neighbor="3.14.159.2" />
        <peering neighbor="65.35.89.79"/>
        <peering neighbor="32.38.46.26"/>
        </router>
</vlan>
```



Proof of Concept




```
#sh openflow flows flowid 22586
Flow ID: 22586 Priority: 28672 Status: Active
        Rule:
            Destination Mac :       ffff.ffff.ffff
            Destination Mac Mask:           ffff.ffff.ffff
            Ether type:    0x00000806
        Action: FORWARD
            Out Port: send to controller
```

Implications

- CAM table programming issues caused by the lack of broadcast in the network
- Multiple controller consistency; active/passive controllers
- No official support Link Aggregation Groups (LAGs)
- Unknown unicast flooding if we reply with the MAC of a dead node

Research Questions

"Can OpenFlow be used to reduce broadcast ARP traffic in the AMS-IX ISP peering LAN?"

"Can ARP be replaced completely by OpenFlow in the core network?"

"Is OpenFlow a scalable solution for this scenario?"

Conclusions

- OpenFlow controller replies to all ARP requests
- All broadcasts are stopped at the PE
- All other traffic is unaffected (hybrid-port mode)
- Depends on the controller being there
- Still need (a bit of) ARP in some of the proposed solutions
- Need to account for CAM table programming

Future Work

- Implement and test other proposed solutions
- Test with multiple controllers
- Adapt the proposed solutions according to the latest Brocade implementations of OpenFlow
- Test solutions in other OpenFlow controllers (NOX, Floodlight)


-Question-