GoPlane: Open Source BUM-less Networking for Large Scale Docker Deployment

August 19, 2015

Soramichi Akiyama
Software Innovation Center, NTT, Japan
akiyama.soramichi@lab.ntt.co.jp

A bit about Myself

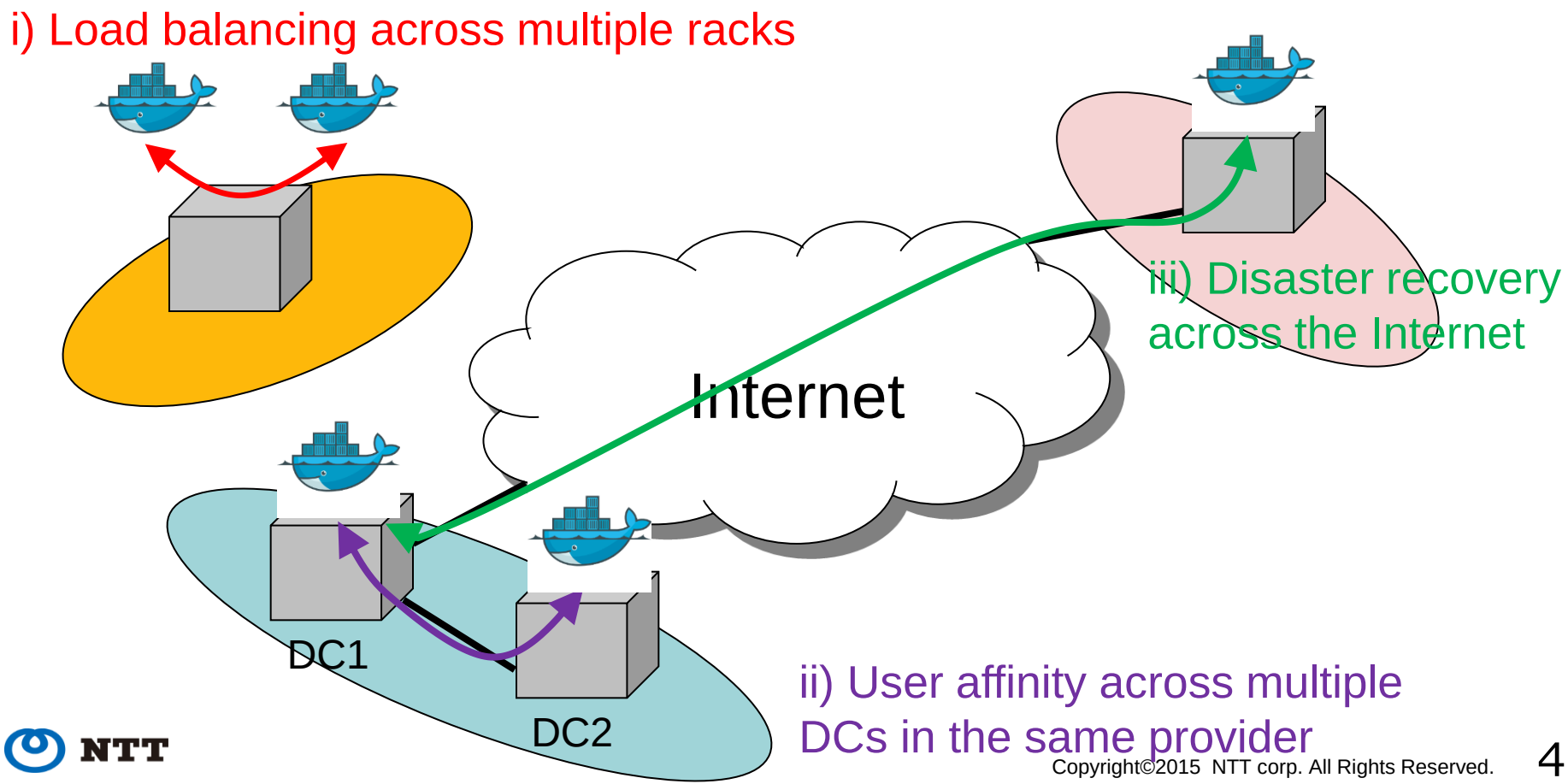
- **Researcher@Software Innovation Center, NTT**
- **Had been working on Virtual Machine Live Migration for 5 years**
  - Ask me if you want to know migration technologies other than “trace and replay”
- **Current work on software defined networking for DCs using baremetal switches**
- **More on me: http://www.soramichi.jp/**

A bit about NTT

- **Japanese counterpart of AT&T**
  - telephone, telegraph, ASDL, FTTH, ISP, DC, mobile
  - Changing toward timely and open

- **FLOSS we develop (avail@GitHub)**
  - Ryu Openflow controller
  - Sheepdog distributed storage (merged in QEMU upstrm)
  - Lagopus software switch

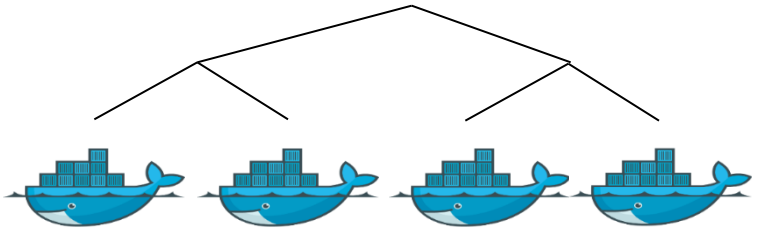
Large Scale Docker Deployment



How Docker Apps are Deployed

- **(Ideally) Deployable as developed**
- **This requires network tenant isolation**

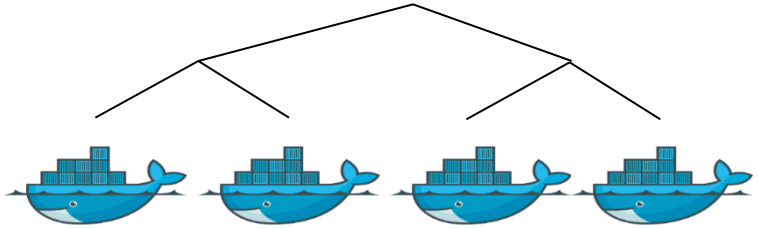
✓ No (visible) one else on the NW
✓ Use any IP you want



Your Macbook (or Linux machine)

Development

✓ No (visible) one else on the NW
✓ Use any IP you want



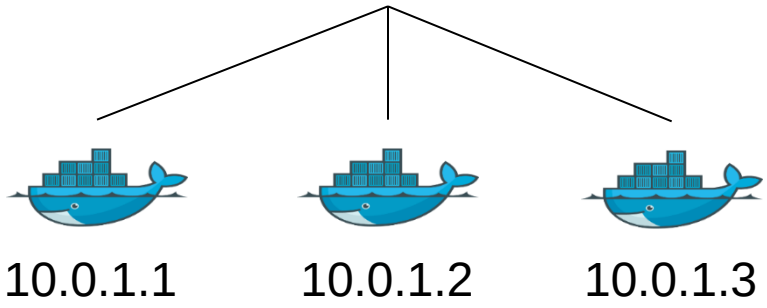
VM

Deployment

Overcoming L3 boundary

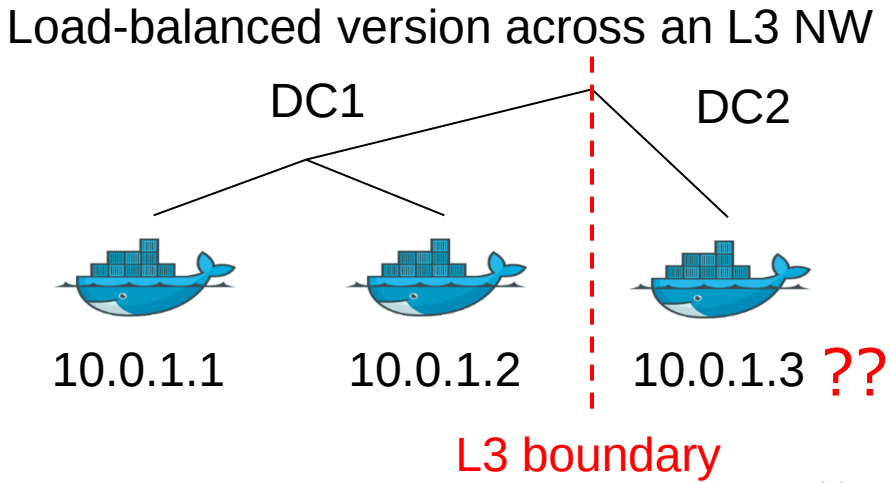
- **Large scale Docker deployment hits L3 boundaries**
  - Across racks, DCs, Internet
  - Large scale single L2 domain is in feasible

Original Docker app in a single L2 domain



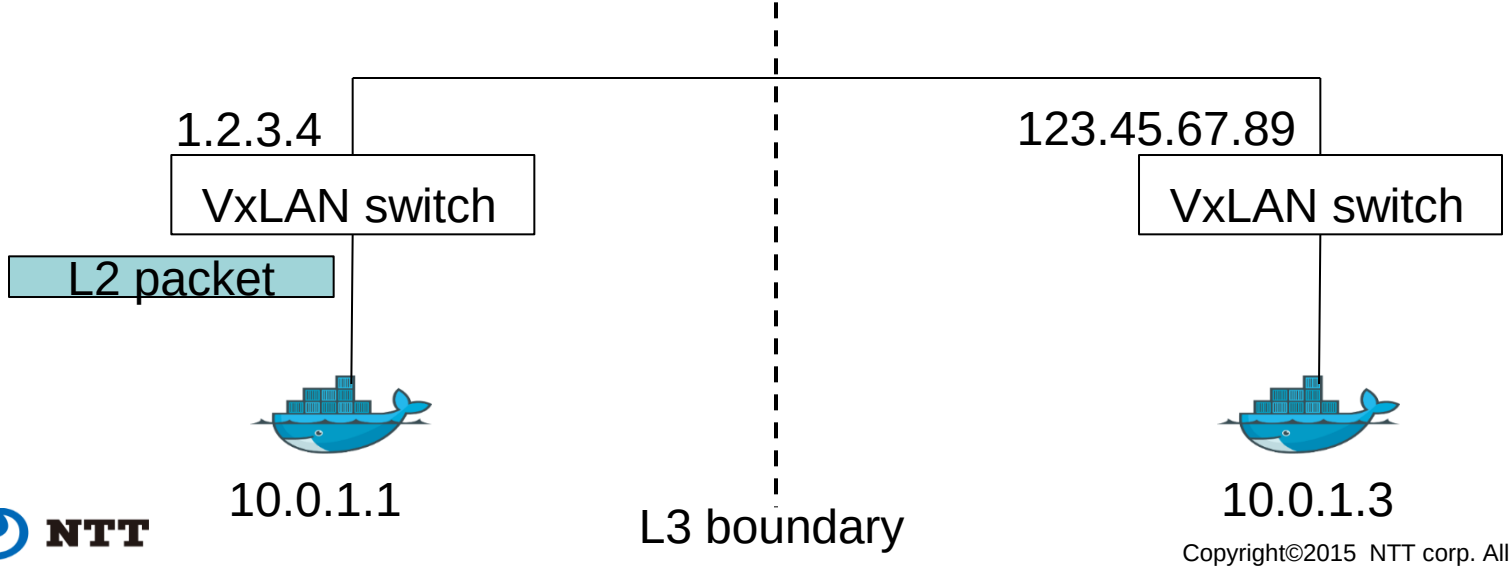
Overcoming L3 boundary

- **Large scale Docker deployment hits L3 boundaries**
  - Across racks, DCs, Internet
  - Large scale single L2 domain is in feasible



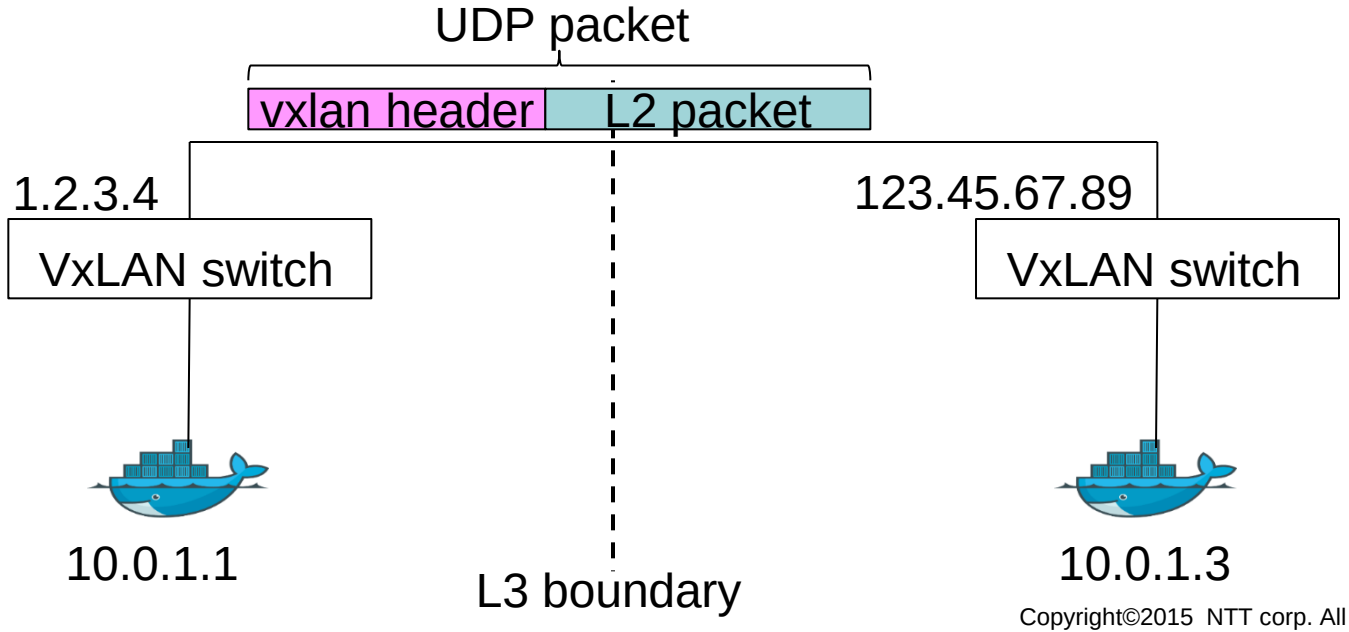
L2 over L3 using VxLAN

- **VxLAN: Virtual eXtensible LAN**
  - Encapsulate L2 packets with VxLAN headers
  - L2 networks can be extended over L3 networks



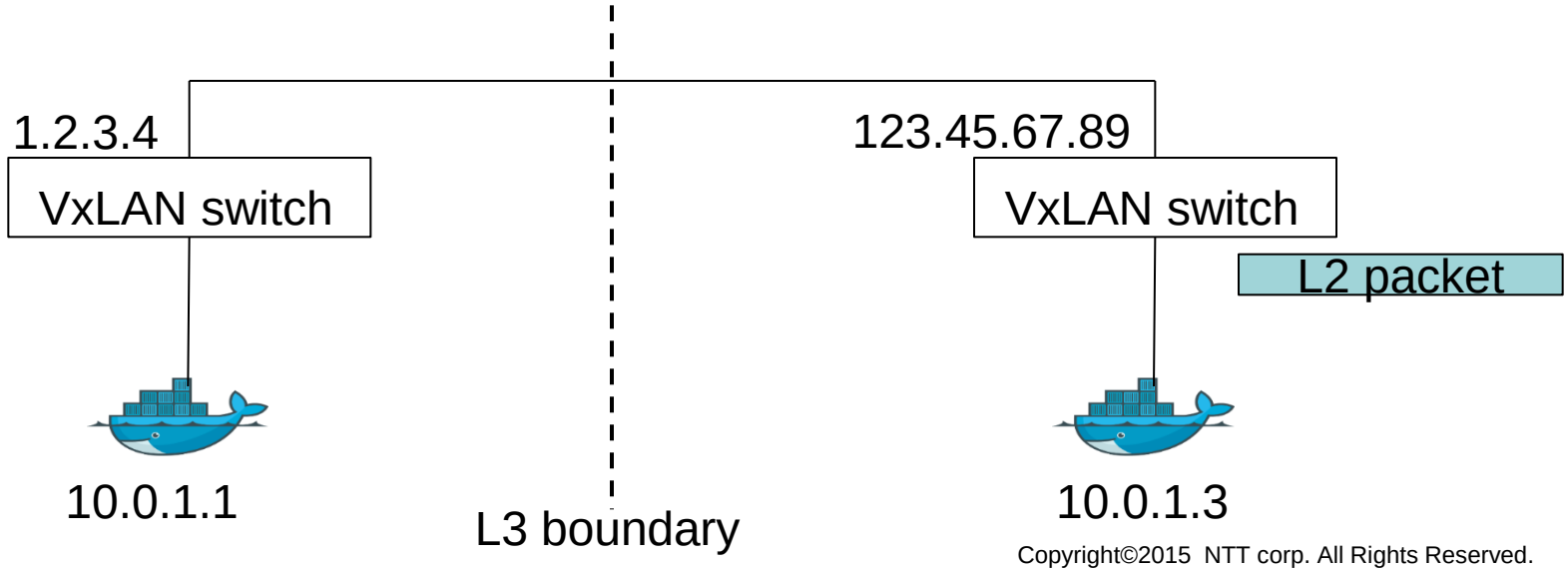
L2 over L3 using VxLAN

- **VxLAN: Virtual eXtensible LAN**
  - Encapsulate L2 packets with VxLAN headers
  - L2 networks can be extended over L3 networks

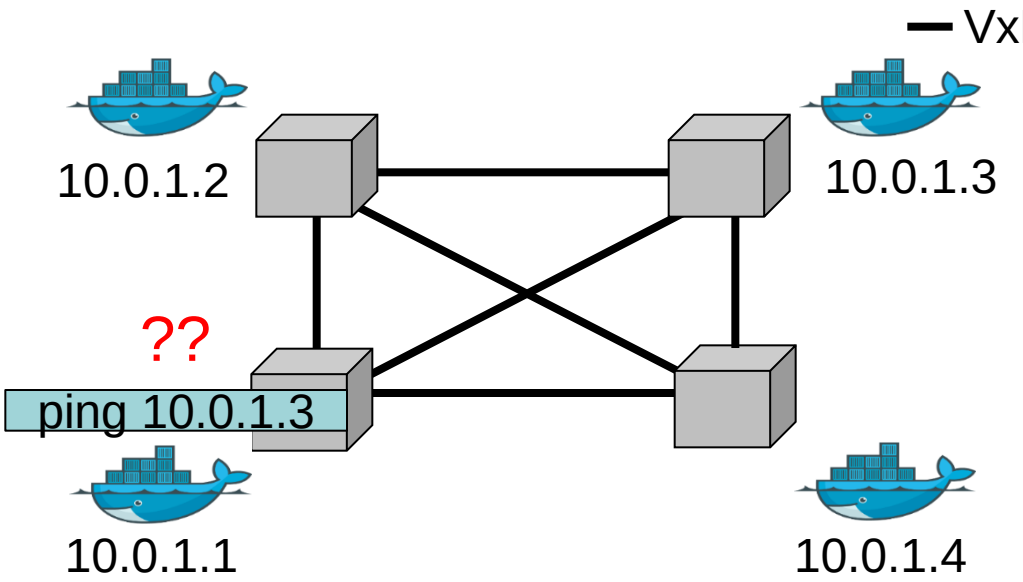


L2 over L3 using VxLAN

- **VxLAN: Virtual eXtensible LAN**
  - Encapsulate L2 packets with VxLAN headers
  - L2 networks can be extended over L3 networks



Problem: How to find the next hop?

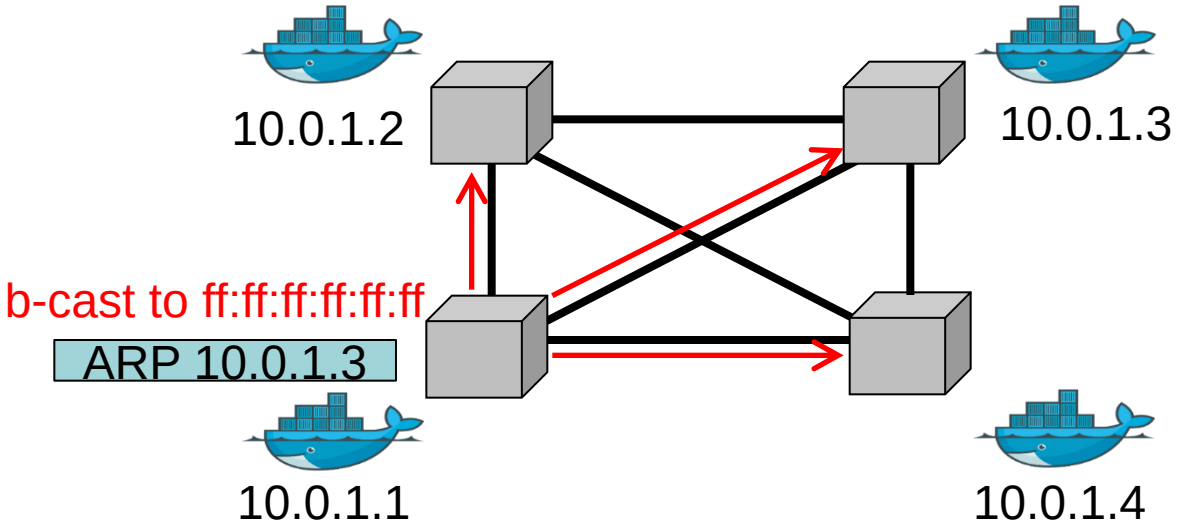


Where to send BUM packets?

- Broadcast
- Unknown Unicast
- Multicast

Naïve method: what Socketplane does

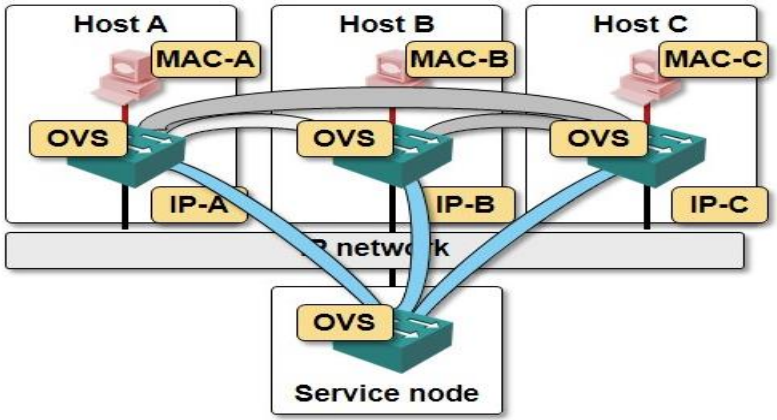
- **Dataplane flood and learn**
  - Just like normal L2 network
  - Broadcasting over L3 networks!



Existing approaches: Nicira service node

- **BUM packets are sent to *the* service node**
  - Reduce CPU load for packet replication

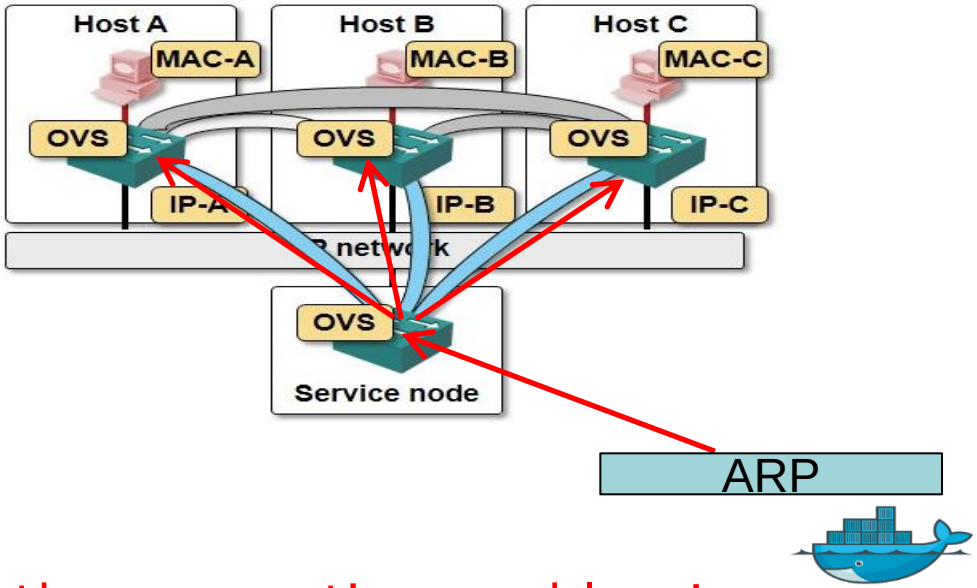
Figure cited from http://blog.ipspace.net/2013/08/nicira-nvp-control-plane.html



Existing approaches: Nicira service node

- **BUM packets are sent to *the* service node**
  - Reduce CPU load for packet replication

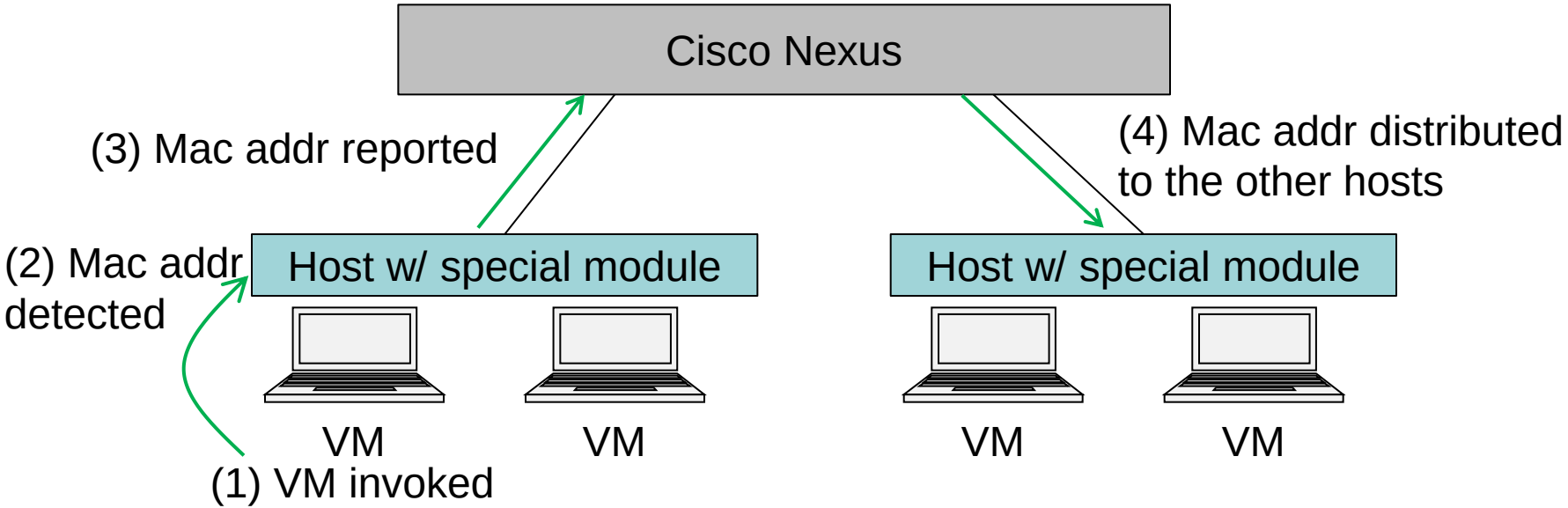
Figure cited from http://blog.ipspace.net/2013/08/nicira-nvp-control-plane.html



- No help to the congestion problem!

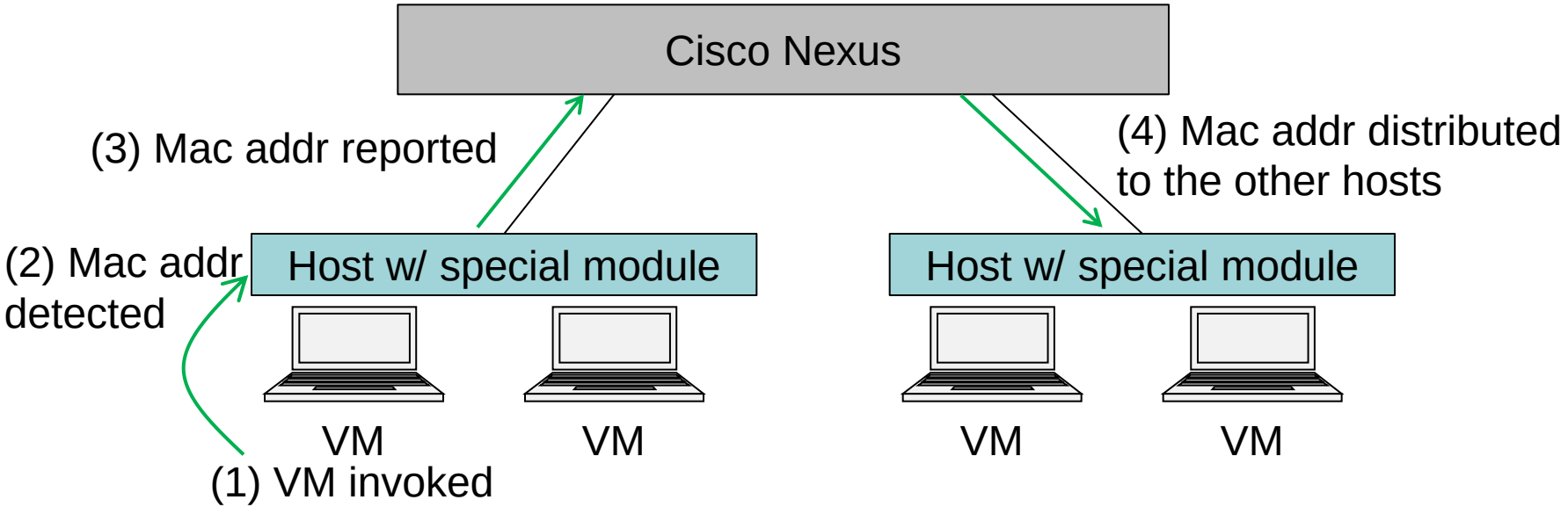
Existing approaches: Cisco floodless mode

- **Implemented in Cisco Nexus switches**
  - Mac addresses are reported to/distributed from ToR



Existing approaches: Cisco floodless mode

- **Implemented in Cisco Nexus switches**
  - Mac addresses are reported to/distributed from ToR

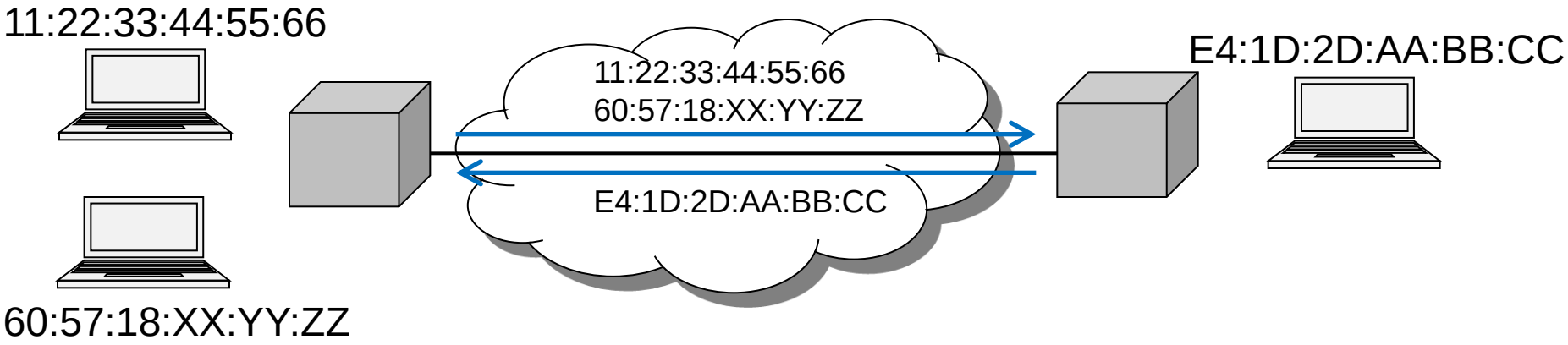


- Vendor lock-in, Cannot go across an L3 network

Our solution: OVS + BGP (EVPN)

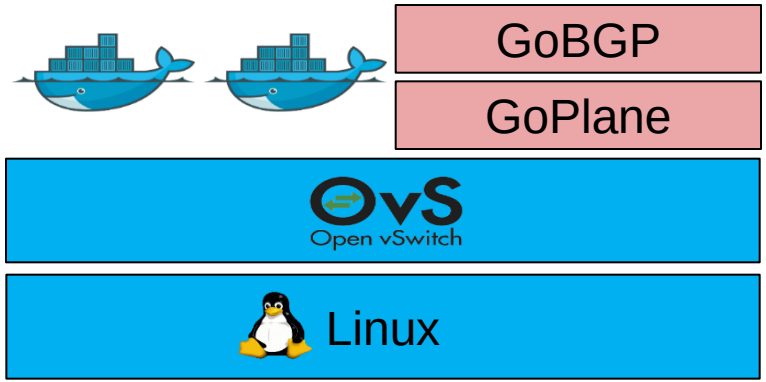
- **EVPN (Ethernet VPN): Simplified overview**
  - Extended ethernet across L3 networks
  - Standardized in RFC7432 on Feb 2015
  - Exchange MAC addresses thru BGP

Border Gateway Protocol



Our solution: OVS + BGP (EVPN)

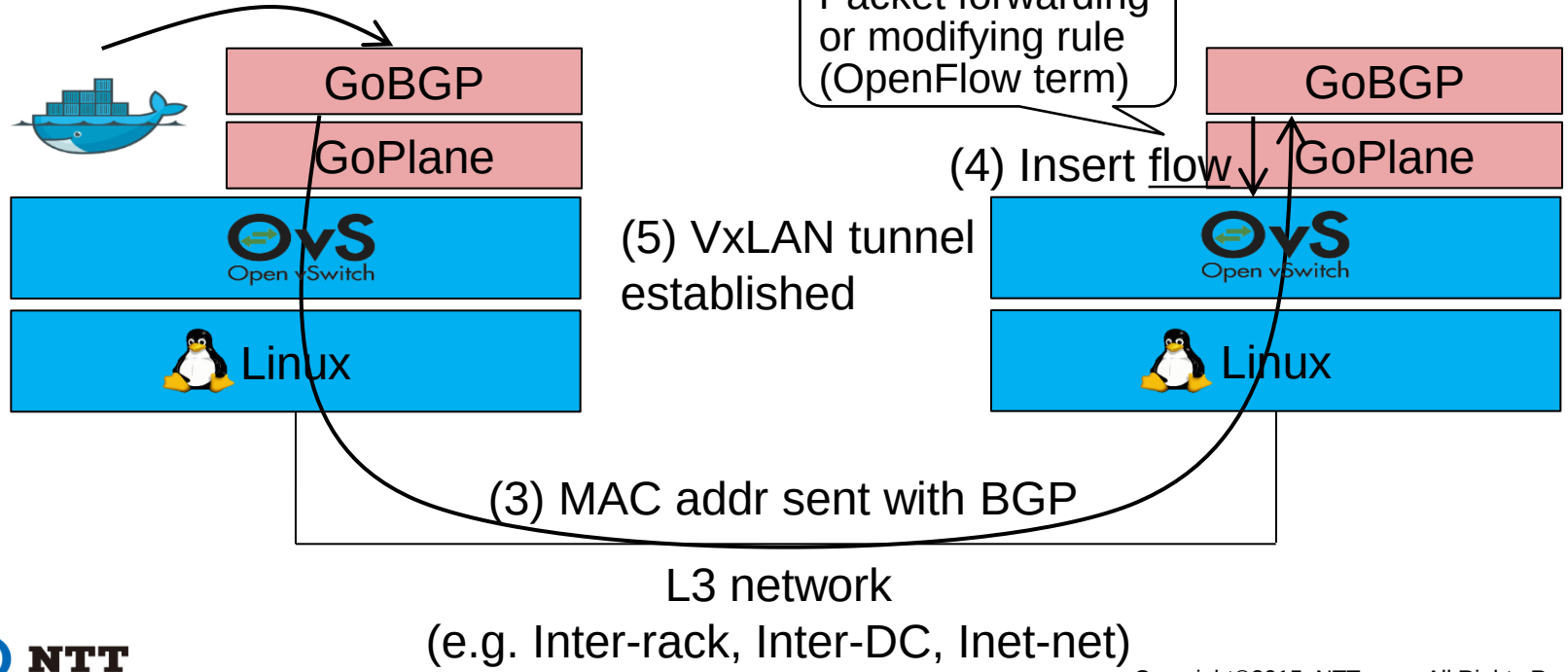
- **System Components**
  - **GoBGP**: Fully open and API-capable BGP daemon
  - **GoPlane**: Data plane management
  - **Open vSwitch**: VxLAN tunneling and flow control



What we built
Existent things

How it works: Mac address exchange

(1) Container invoked
(2) MAC addr notified to GoBGP



Flows Inserted

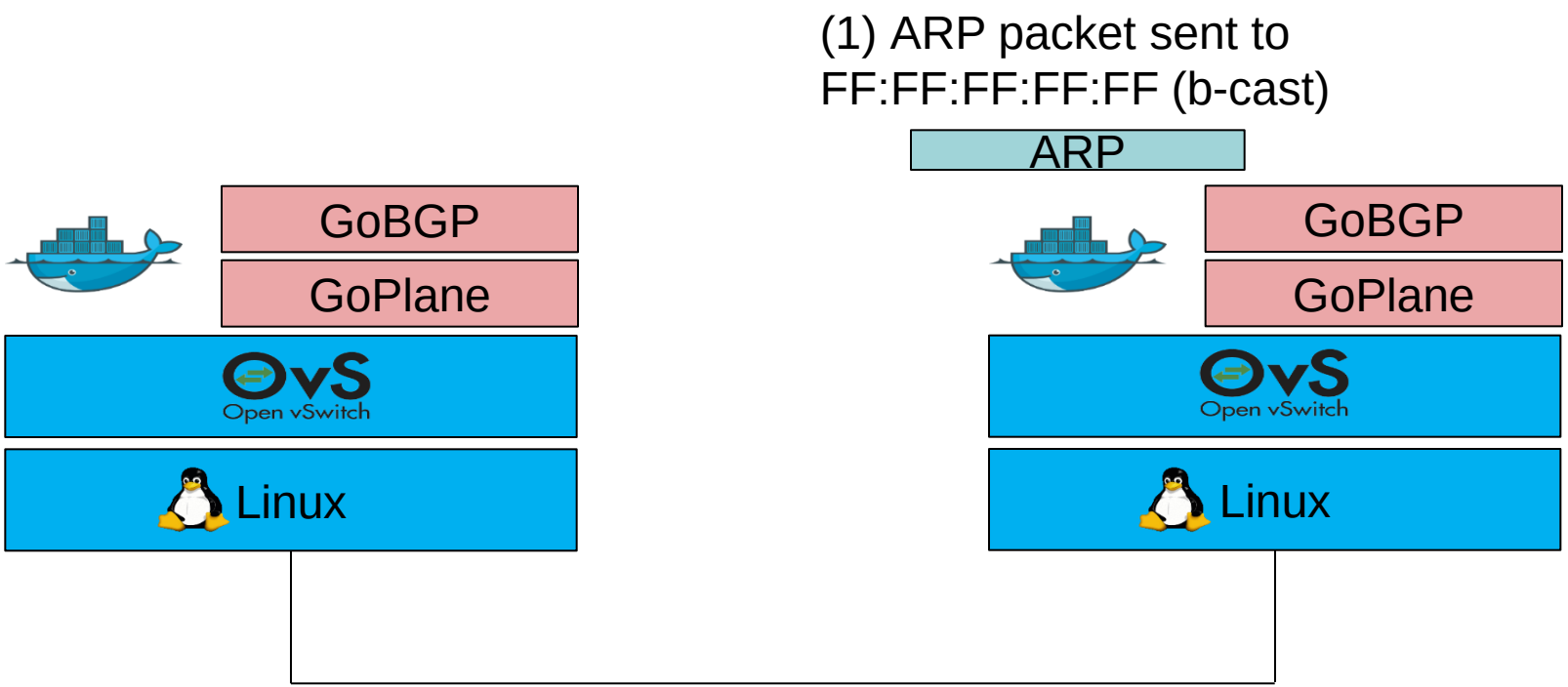
1. Remote tunnel selection flow

- Mac address → VxLAN tunnel beyond which the container with the mac exists

2. ARP responder flow

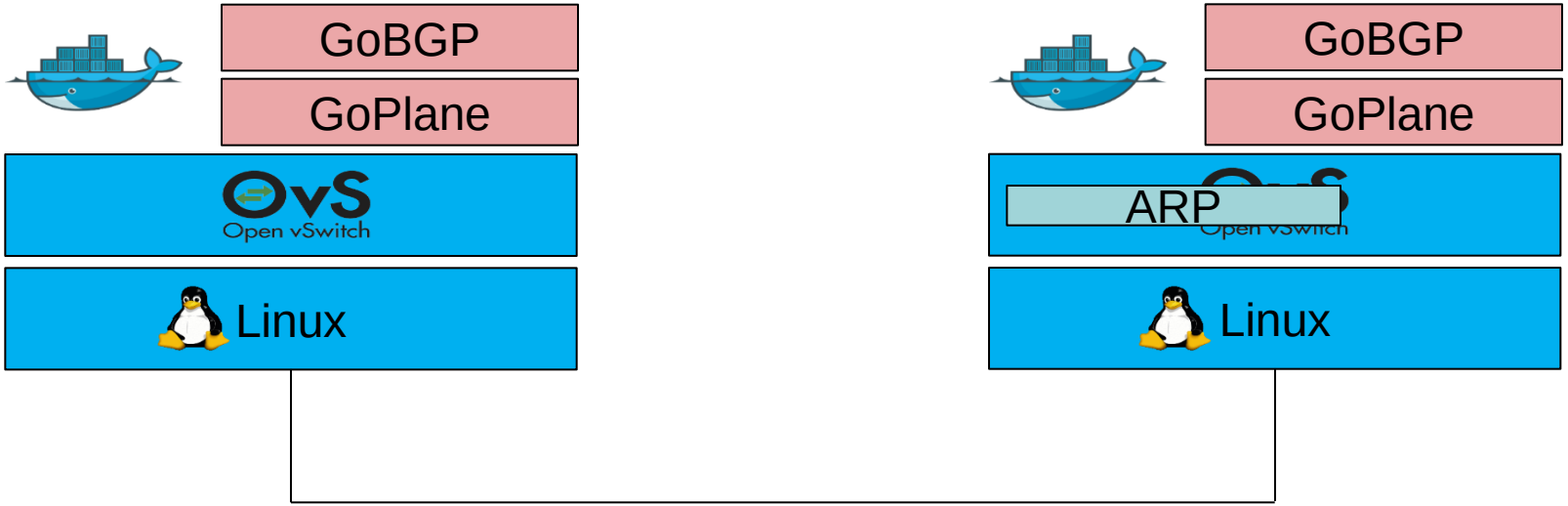
- Mutates an ARP request to a corresponding ARP response (described in the coming slides)

How it works: ARP & Response



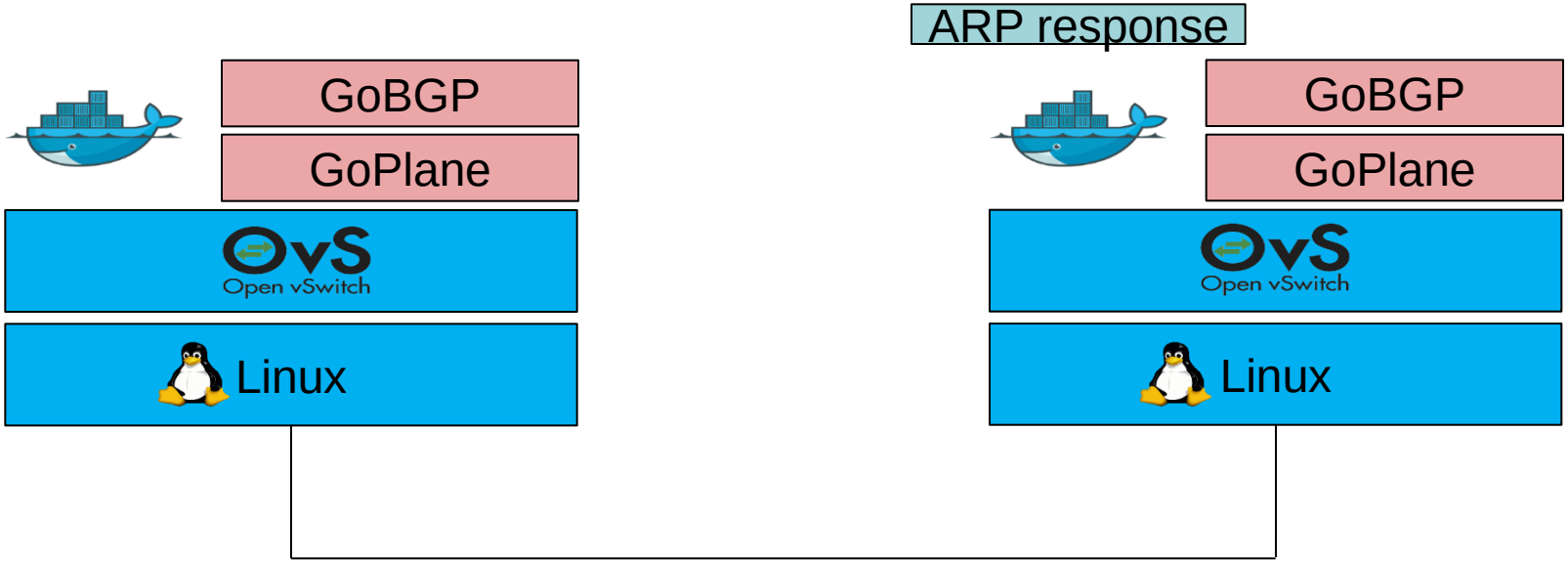
How it works: ARP & Response

(2) OVS receives the ARP,
mutates it to the response
(as OVS knows the MAC addresses!)



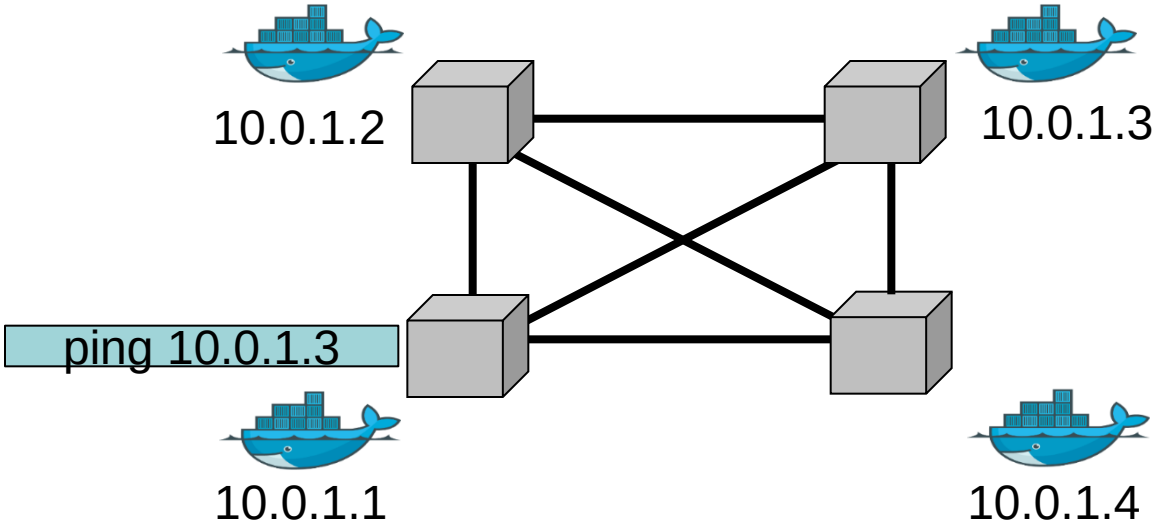
How it works: ARP & Response

(3) Container receives the response,
ARP packets never be shot to actual NW



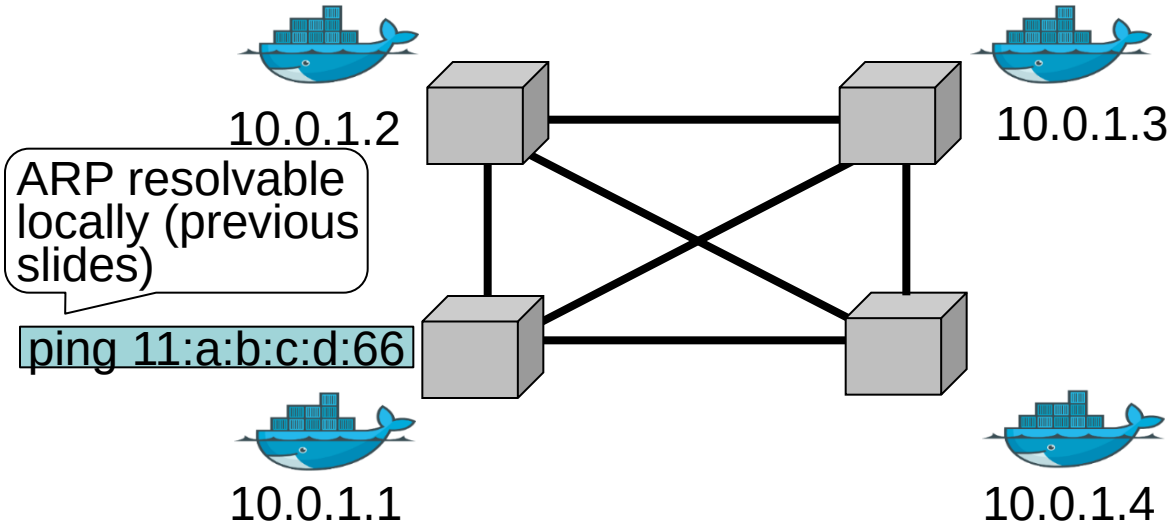
How it works: Unicast

- **Every unicast is “known”**
  - If the target mac is unknown by the OVS, that address does not exist



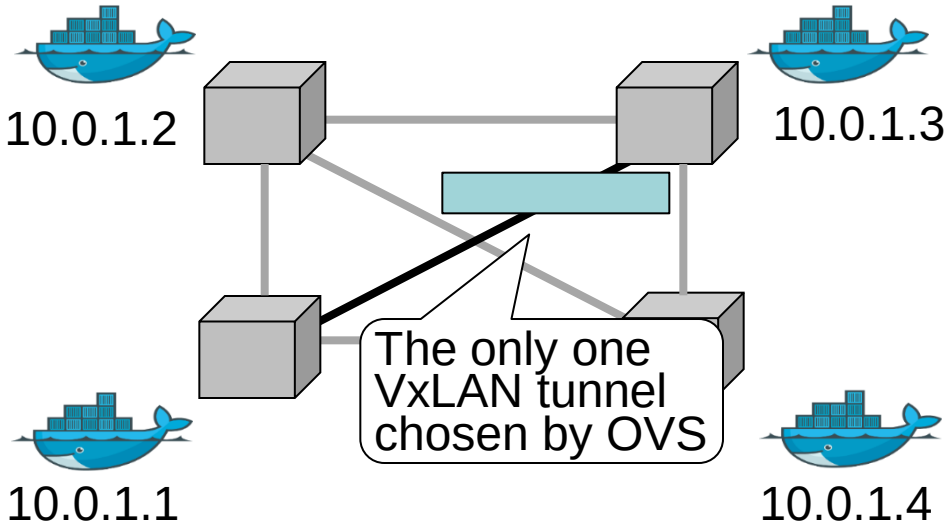
How it works: Unicast

- **Every unicast is “known”**
  - If the target mac is unknown by the OVS, that address does not exist



How it works: Unicast

- **Every unicast is “known”**
  - If the target mac is unknown by the OVS, that address does not exist



Discussion

- **Pros of our solution ☺**
  - Fully open, no proprietary technology
  - Flexibility by API-capable BGP daemon

  → Even the containers themselves can manage the network if you want (looking for a good use case)

- **Cons of our solution ☹**
  - BGP doesn't consider coherence

  → Special extension needed for IP management

GoBGP

- Open Source BGP daemon we develop
  - Available@Github https://github.com/osrg/gobgp
- Controlled with GRPC (via http)
  - Suitable for SDN
- Leverage multi-core CPUs
  - The most famous existing software BGP is single threaded

EVPN Interoperation@Interop Tokyo 2015

http://osrg.github.io/goplane/post/20150615/

Summary

- **L2 over L3 NW is required for large scale Docker deployment**
- **Existing solutions do not fulfill requirements**
  - Service node: no help for congestion
  - Cisco floodless mode: vendor lock-in, ToR only
- **We built a fully open solution, by a combination of BGP (EVPN) + OVS**
- **GoPlane and GoBGP available at GitHub!**

Q&A