# Virtualizing The Network For Fun and Profit

# Building a Next-Generation Network Infrastructure using EVPN/VXLAN

**By Richard A Steenbergen <richard@steenbergen.us>**

# A BRIEF HISTORY OF LAYER 2 NETWORKING

## Historically, we had a very clearly defined “LAN” and “WAN”.

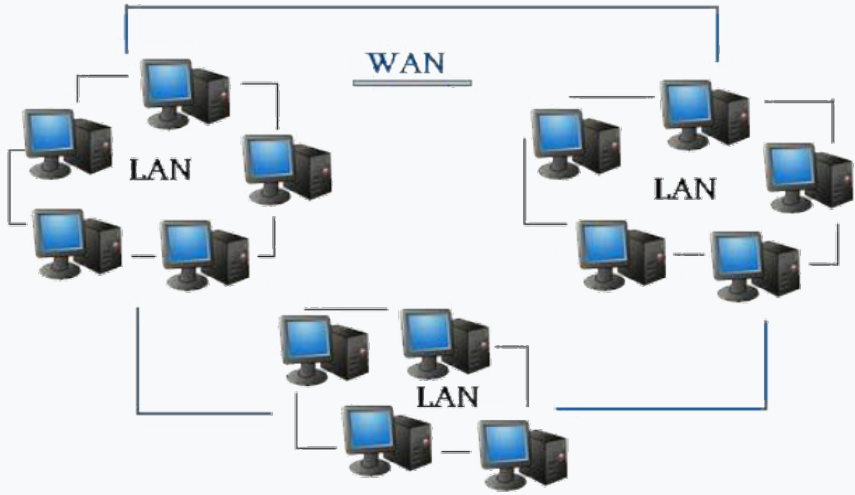
- LANs were cheap, small networks, that connected a handful of local devices.
  - But with major known scalability issues that limited their size and scope.

- WANs were built on Big Expensive Routers, that connected a lot of LANs together.
  - The individual circuit technology is less important.
  - But the overall networks, built on large numbers of these individual circuits, needed far more scalable technologies (such as IP, and MPLS).

# A BRIEF HISTORY OF LAYER 2 NETWORKING

## Today, nearly everything is built using Ethernet, a "LAN" protocol.

- Simplicity, low-cost, and massive market penetration, managed to beat out every other "technically superior" competing protocol in the end.
- But we still struggle with some of it's inherent scaling properties, creating protocol extension after protocol extension to try and make it more usable in a wider variety of roles.



# SCALABILITY PROBLEMS OF NATIVE ETHERNET

## Compared to Layer 3 technologies like IP, pure Ethernet just sucks:

- Loop prevention is critically important just to keep things operating.
  - Forcing us into ever more complex variants of protocols like Spanning-Tree.
- Distinct segments in a single domain are severely limited by 10-bit VLAN scale.
  - 4095 VLANs doesn't go far when you're trying to build large scale networks.
- Each device along the transit path needs to share the same VLAN configuration.
  - Provisioning a new service means touching each transit device along the way.
- Delivering redundancy is "incredibly complicated" to say the least, etc, etc.

## So we restrain native Ethernet to only the simplest environments.

- And reserve a special place in hell for those who break this cardinal rule.

# BUILDING SCALABILITY INTO ETHERNET

## So how DO we deliver Ethernet services on a large network at scale?

- We can cheat, and just deliver a dedicated layer 1 service instead.
  - For example: Dedicated optical waves, or OTN digital transport wrappers.
- Or, we emulate Ethernet services on top of a more complex/scalable network.

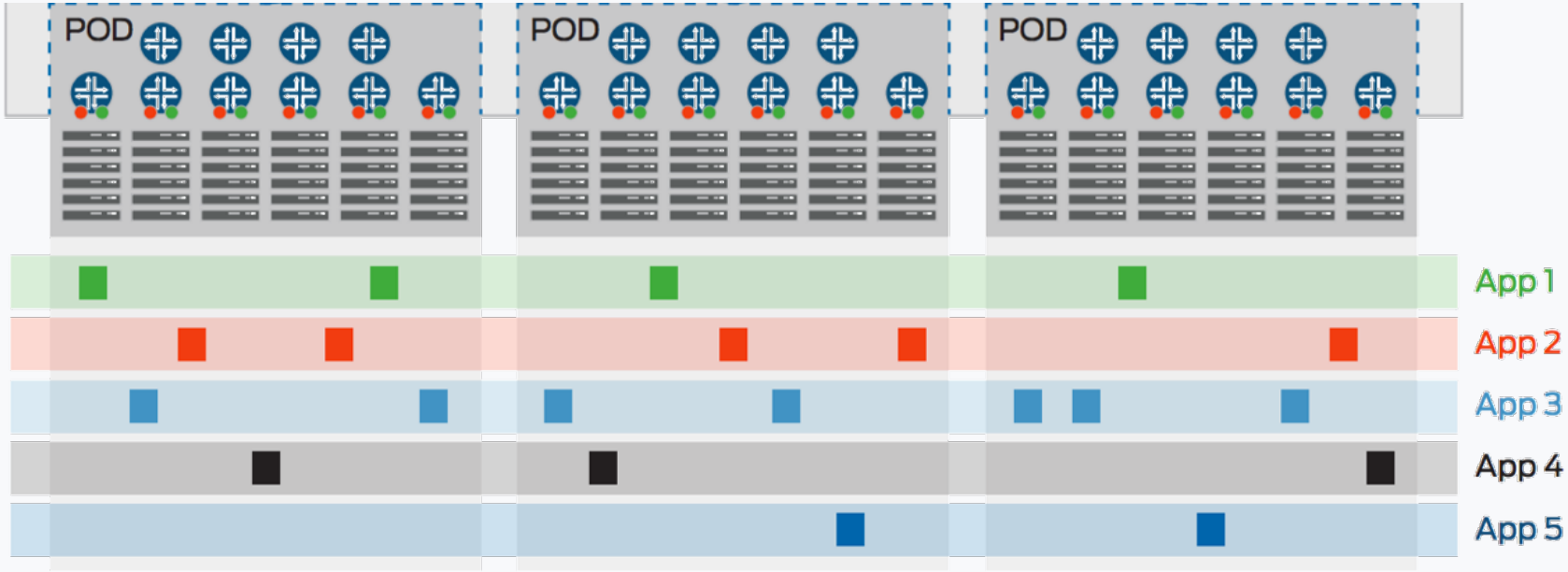
  Examples include:
  - Layer 2 Tunneling Protocol (L2TP) over IP
  - MPLS pseudo-wire services (draft-martini, VPWS, VPLS, etc)
  - MAC-in-MAC transport (Provider Backbone Bridging)

## These have worked well for simple "circuit emulation" tasks.

- For example, a carrier selling a virtual Ethernet circuit across an IP/MPLS network.
- But in the world of "cloud" services, even these have serious scaling problems.

# A MODERN VIRTUALIZED "CLOUD" ENVIRONMENT

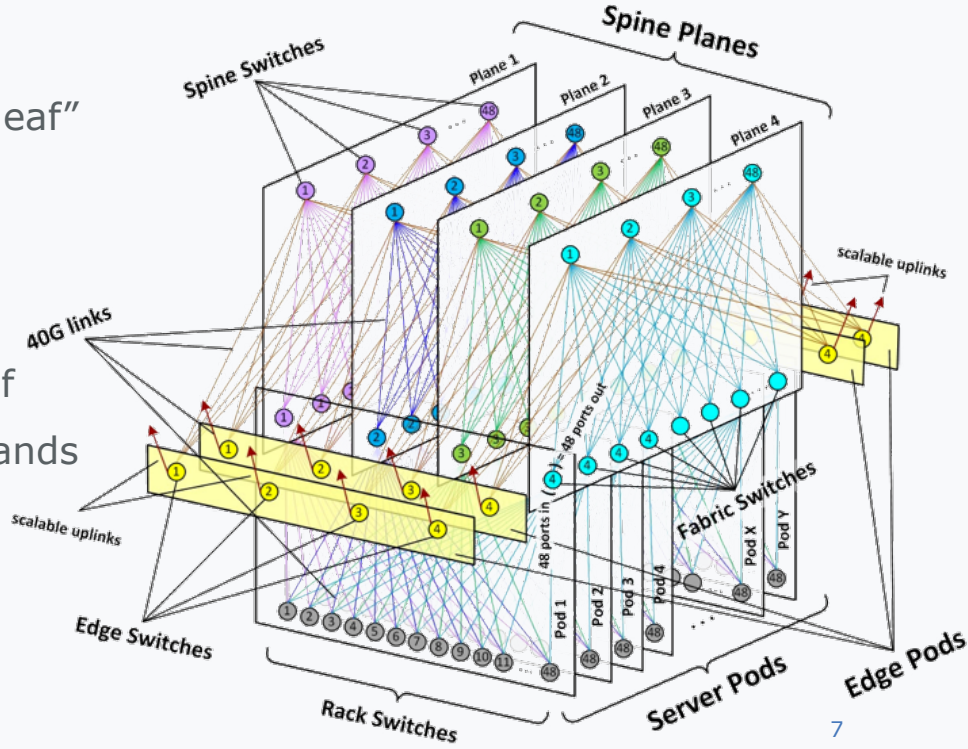
Workloads spread over a **great** many distributed servers, expecting LAN connectivity on the backend.



# DATACENTER NETWORKING FABRICS

## To support hyperscale cloud, new network architectures were needed

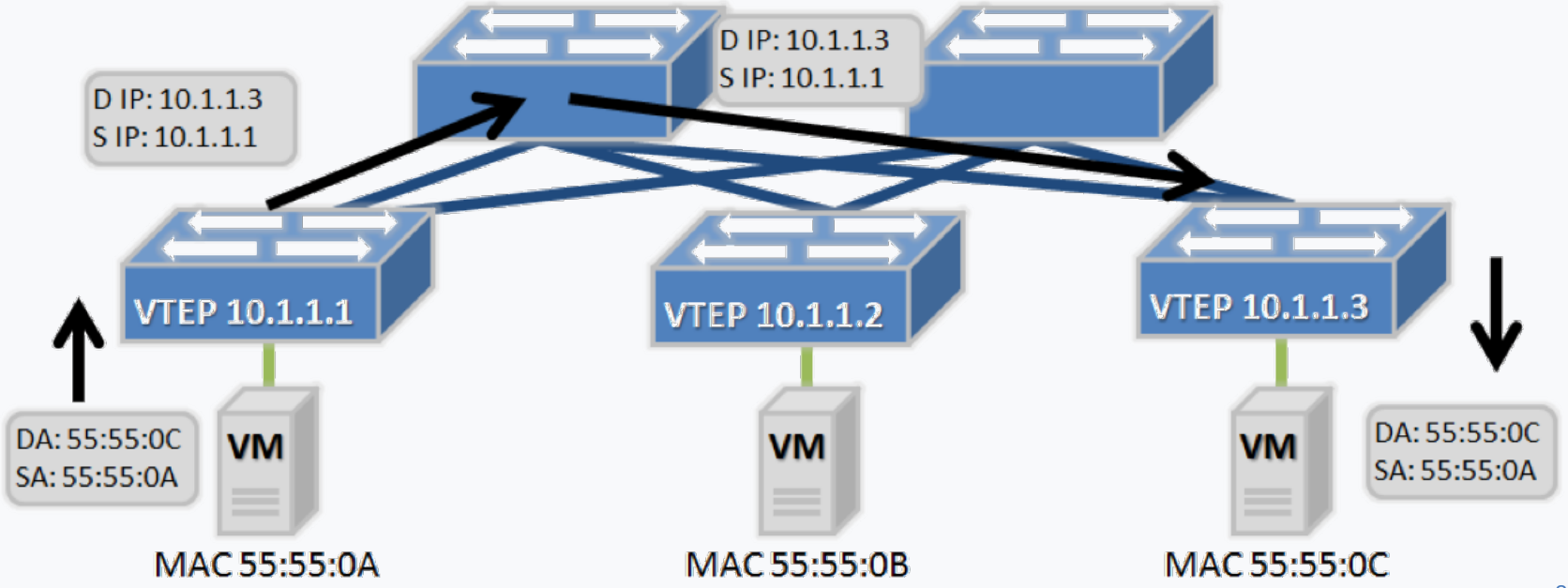
- Large scale cloud implementations needed to use more complex “spine/leaf” architectures, to economically connect huge numbers of servers, with large amounts of bandwidth.
- Multiply the hundreds of thousands of “services” they run, by tens of thousands of servers that must be connected, and you start to see the scale of the networking challenge.



# VIRTUAL EXTENSIBLE LAN (VXLAN)

## One notable attempt to improve on this was the VXLAN protocol.

- At it's simplest, can be described as tunneling Layer 2 frames over IP/UDP packets.



# VIRTUAL EXTENSIBLE LAN (VXLAN)

## Why? Remember our earlier discussion about how to scale Layer 2.

- The most scalable and successful techniques for providing Layer 2 services haven't involved running complex Layer 2 services natively, but rather, running far more scalable protocols and then emulating L2 Ethernet services across them.

- VXLAN takes this philosophy a step further, with a distinct "Underlay" and "Overlay".
  - Underlay – Use the simplest, cheapest, most scalable, most pervasive network technology we can find. Does that sound like "IP" to anyone else?
  - Overlay – Use the underlay to provide scalable and reliable transport, then overlay the services you want (e.g. L2 Ethernet) in a simple, light-weight, low-state way.

# THE ADVANTAGES OF VXLAN

## The original motivations were all about improving "cloud" deployments

- It increases the total number of possible distinct network segments significantly.
  - VLANs have a 10-bit ID field (4095 possible values)

    VXLANs have a 24-bit VNI field (16.7 million possible values)
- Allow VM mobility, independent of the physical network configuration.
- Provides multi-path forwarding using the ECMP properties of the underlay network.
- Reduces the amount of network state necessary to deliver service significantly.
  - Removes ALL service-specific state from the core network devices.
  - But it also eliminates the need for edge devices to negotiate each specific service instance with a protocol (such as LDP signaling, in the case of MPLS).

# THE DISADVANTAGES OF VXLAN

## But alas, VXLAN is not without it's own disadvantages.

- The protocol was designed for "known friendly" traffic, VM computing workloads.
- VXLAN lacks real "signaling protocols", and is expected to be configured via SDN.
  - Requires a centralized controller, which can be extremely difficult to scale.
- It uses simple "Flood and Learn" data-plane driven forwarding mechanisms.
  - Just like Ethernet, you get "Unknown Unicast" frames which you must flood to every member of the VXLAN, hoping to hear a reply and learn the real dest.
  - This is one of the inherent problems with large scale Ethernet networks already.
  - It can also have significant overhead and provide inefficient bandwidth use.
- It requires a multicast-enabled Underlay network for BUM traffic.
  - A serious problem for brownfield networks, or those who don't trust multicast.

# ENTER ANOTHER PROTOCOL, EVPN

## Oh great, another protocol. What does this one do?

- EVPN (Ethernet-VPN) is an evolution on earlier carrier L2VPN technologies like VPLS.
- Essentially, it's the advanced control-plane that VXLAN is missing.

## So what does EVPN provide that VXLAN doesn't?

- EVPN turns Ethernet service emulation from an unpredictable "flood and learn" model, to a predictable/scalable protocol-based routing of MAC addresses via BGP.
  - A known-scalable way to distribute millions of routes across millions of devices.
- EVPN adds support for all-active redundancy, load balancing, and loop prevention.
- It allows rapid MAC moves, which enhances VM mobility without hair-pinning.
- It eliminates the need for a multicast-enabled underlay network.
- It also natively supports Layer 3 Gateway functionality (more about this later).

# THE EVPN+VXLAN COMBINATION

## The chocolate and peanut butter of Ethernet emulation protocols

- EVPN provides the reliable, scalable control-plane you want for delivering Ethernet services at scale (large numbers of POPs, devices, and service instances).

- VXLAN provides the simple, light-weight, low-state forwarding plane, that has already seen widespread adoption from most major ASIC manufacturers.

- The end result is a highly scalable and high-performance Ethernet-based services framework, with all of the scaling properties of the underlying IP network, and without the requirement of running complex (and very vendor-limiting) protocols like MPLS.

# OK, BUT WHAT IF I'M NOT BUILDING A "CLOUD"?

## How does this apply to a pure Network Service Provider?

- Just because a concept originated to support "datacenter network" infrastructures, doesn't mean it isn't applicable in other areas, or across wider area networks.

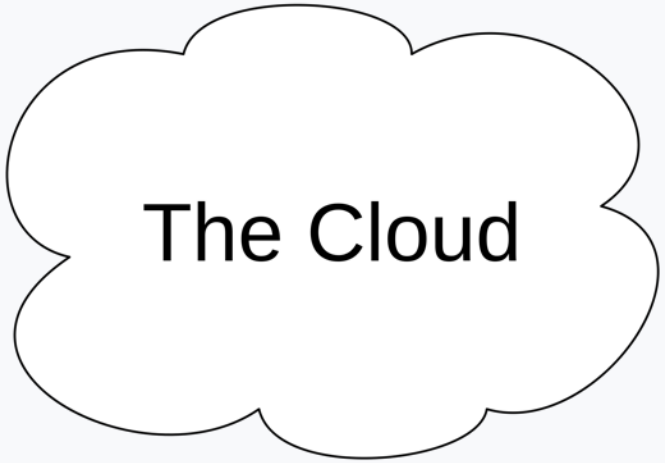
## Historically, we've used a "bare metal" approach to building NSP networks.

- Aiming to deploy the "correct" capacity/functionality hardware, in all the right places.
  - Which entails constant physical work, upgrading and grooming infrastructure.
- "Running an NSP" usually meant having full-table Big Expensive Routers everywhere.
  - Deployed in every location where you wanted to offer services, even if you weren't using all of the features and functionality of the box.
  - One first generation attempt to optimize this was the "label switching only core", but this meant $$$ "carrier-grade" MPLS routers, and restricted vendor choice.

# VIRTUALIZING THE NETWORK

## What if we could do for networking what "cloud" has done for compute?

- Start with a core "fabric" underlay network.
  - Optimized purely for providing massively scalable, cheap, dumb "connectivity", between any two points on the network.
  - Using optimized hardware, without the need for large FIBs, or MPLS features.
- Now, virtually connect your high touch network devices and customers using this fabric, to provide more advanced services (e.g. routing, firewalls, etc), but only when you need them.
- This is similar in concept to how a "switch fabric" works inside a router.



# VIRTUALIZING THE NETWORK

## EVPN/VXLAN finally provides the necessary technology to do this "well"

- This applies to traditional carrier networks, as well as datacenters and clouds.
- This model scales to support millions of virtual-circuits and MAC addresses, hundreds of thousands of devices, and hundreds of terabits/sec of bandwidth, cost effectively.
- EVPN L3 Gateway functionality allows your BERs to "join" any EVPN instance.
  - You don't need your Big Expensive Routers in the forwarding path of every packet in the network, you only need them to provide the necessary services, localized to provide acceptable latency and redundancy.
  - This means you can consolidate and aggregate services onto a smaller number of more purpose-built routers, getting more efficient utilization of your hardware.
  - You can dynamically select the "services" you need to apply, picking and choosing the correct/optimal devices on the fly, in a scalable and programmatic way.

# IMPACT TO A NETWORK COST MODEL

## What does all of this do to a carrier's economic model?

- Big Expensive Routers used in carrier-grade networks will often cost 25-50x more than the simpler "datacenter optimized" devices used to build "cloud" fabrics.
  - Commodity merchant silicon, used support large scale cloud applications, are a small fraction of the cost of traditional BER "god boxes", and falling.
- Historically this made sense, these devices were doing very specialized things.
  - Handling large-scale Internet FIBs, doing complex MPLS-TE functions, etc.
  - But a modern virtualized network can be far more cost effective.
- Even if you never eliminate the need for Big Expensive Routers in some locations, virtualizing their application allows you to consolidate workloads more efficiently.
  - Similar to keeping CPU cycles from going unused in a "cloud" environment.
- Pure transport becomes a far cheaper and more scalable service as well.

# EXTERNAL RESOURCES

## More information about EVPN/VXLAN

- https://ripe68.ripe.net/presentations/170-ripe-68-evpn.pdf
- https://www.juniper.net/assets/us/en/local/pdf/whitepapers/2000606-en.pdf
- http://www.cisco.com/c/dam/en/us/td/docs/switches/datacenter/nexus9000/sw/vxlan_evpn/VXLAN_EVPN.pdf

# THANKS FOR YOUR TIME!

Any Questions?

Richard A Steenbergen <richard@steenbergen.us>