A Network Virtualization Overlay Solution using EVPN

Michele Finelli
m@biodec.com
BioDec

Index

- The problem
- Long time ago . . .
- Current solutions
- EVPN
- Conclusions

Index

- The problem
- Long time ago . . .
- Current solutions
- EVPN
- Conclusions

Many, many containers

- A microservice architecture requires order of magnitude more components than a traditional (*virtual machine*) based solution.
- The natural mapping of microservices is over containers, so we have to deal with hundreds (thousands) of containers.

Many, many containers

- A microservice architecture requires order of magnitude more components than a traditional (*virtual machine*) based solution.
- The natural mapping of microservices is over containers, so we have to deal with hundreds (thousands) of containers.

Many, many containers

- A microservice architecture requires order of magnitude more components than a traditional (*virtual machine*) based solution.
- The natural mapping of microservices is over containers, so we have to deal with hundreds (thousands) of containers.

Example

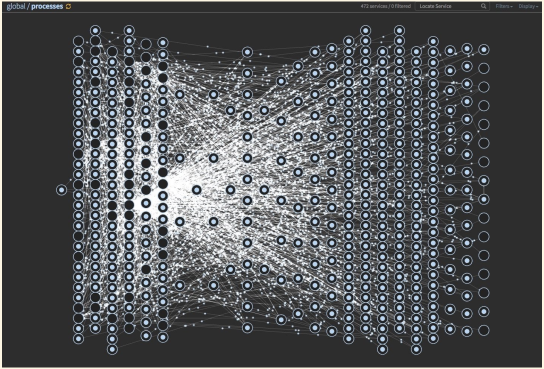


Figure: Allegrotech architecture from “Hitting the wall” - http://allegro.tech/2017/03/hitting-the-wall.html

1 container = 1 IP address

- Each container has an IP address . . .
- . . . and a *virtual Ethernet* interface . . .
- . . . with its own *MAC address*.

A large deployment usually ends up with a large *layer 2* network setup.

1 container = 1 IP address

- Each container has an IP address . . .
- . . . and a *virtual Ethernet* interface . . .
- . . . with its own *MAC address*.

A large deployment usually ends up with a large *layer 2* network setup.

1 container = 1 IP address

- Each container has an IP address . . .
- . . . and a *virtual Ethernet* interface . . .
- . . . with its own *MAC address*.

A large deployment usually ends up with a large *layer 2* network setup.

1 container = 1 IP address

- Each container has an IP address . . .
- . . . and a *virtual Ethernet* interface . . .
- . . . with its own *MAC address*.

A large deployment usually ends up with a large *layer 2* network setup.

1 container = 1 IP address

- Each container has an IP address …
- … and a *virtual Ethernet* interface …
- … with its own *MAC address*.

A large deployment usually ends up with a large *layer 2* network setup.

Layer 2 is flat

- Layer 2 has no **segmentation** of network traffic.
- Except for *VLAN* (Virtual Local Area Network).

> *In the case when the VMs in a data center are grouped according to their Virtual LAN (VLAN), one might need thousands of VLANs to partition the traffic according to the specific group to which the VM may belong. The current VLAN limit of 4094 is inadequate in such situations.*
>
> *RFC 7348*

Layer 2 is flat

- Layer 2 has no **segmentation** of network traffic.
- Except for *VLAN* (Virtual Local Area Network).

> *In the case when the VMs in a data center are grouped according to their Virtual LAN (VLAN), one might need thousands of VLANs to partition the traffic according to the specific group to which the VM may belong. The current VLAN limit of 4094 is inadequate in such situations.*
>
> *RFC 7348*

Layer 2 is flat

- Layer 2 has no **segmentation** of network traffic.
- Except for *VLAN* (Virtual Local Area Network).

> *In the case when the VMs in a data center are grouped according to their Virtual LAN (VLAN), one might need thousands of VLANs to partition the traffic according to the specific group to which the VM may belong. The current VLAN limit of 4094 is inadequate in such situations.*
>
> *RFC 7348*

Layer 2 is flat

- Layer 2 has no **segmentation** of network traffic.
- Except for *VLAN* (Virtual Local Area Network).

> *In the case when the VMs in a data center are grouped according to their Virtual LAN (VLAN), one might need thousands of VLANs to partition the traffic according to the specific group to which the VM may belong. The current VLAN limit of 4094 is inadequate in such situations.*
>
> *RFC 7348*

ff:ff:ff:ff:ff:ff ucking broadcast

- ARP resolution, in a nutshell, is replying to a *broadcast* request.
- This means *a lot of traffic*.
- By design.

ff:ff:ff:ff:ff:ff ucking broadcast

- ARP resolution, in a nutshell, is replying to a *broadcast* request.
- This means *a lot of traffic*.
- By design.

ff:ff:ff:ff:ff:ff ucking broadcast

- ARP resolution, in a nutshell, is replying to a *broadcast* request.
- This means *a lot of traffic*.
- By design.

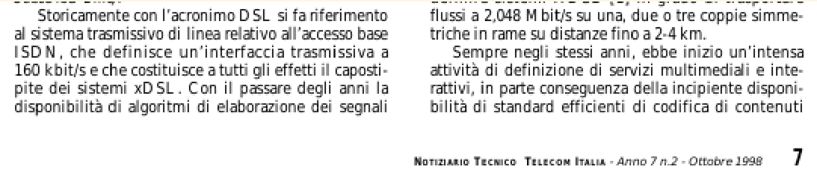
ff:ff:ff:ff:ff:ff ucking broadcast

- ARP resolution, in a nutshell, is replying to a *broadcast* request.
- This means *a lot of traffic*.
- By design.

Index

- The problem
- Long time ago . . .
- Current solutions
- EVPN
- Conclusions

ADSL



Storicamente con l'acronimo DSL si fa riferimento al sistema trasmissivo di linea relativo all'accesso base ISDN, che definisce un'interfaccia trasmissiva a 160 kbit/s e che costituisce a tutti gli effetti il capostipite dei sistemi xDSL. Con il passare degli anni la disponibilità di algoritmi di elaborazione dei segnali

flussi a 2,048 Mbit/s su una, due o tre coppie simmetriche in rame su distanze fino a 2-4 km.

Sempre negli stessi anni, ebbe inizio un'intensa attività di definizione di servizi multimediali e interattivi, in parte conseguenza della incipiente disponibilità di standard efficienti di codifica di contenuti

NOTIZIARIO TECNICO TELECOM ITALIA - *Anno 7 n.2 - Ottobre 1998* 7

Before ADSL !

- There was the *Plain Old Telephone Service*,
- (it had 56Kb)
- and was billed *by the time the line was up*.

Before ADSL !

- There was the *Plain Old Telephone Service*,
- (it had 56Kb)
- and was billed *by the time the line was up*.

Before ADSL !

- There was the *Plain Old Telephone Service*,
- (it had 56Kb)
- and was billed *by the time the line was up*.

Before ADSL !

- There was the *Plain Old Telephone Service*,
- (it had 56Kb)
- and was billed *by the time the line was up*.

An interesting problem

- People wanted to call their provider when they wanted to access the Internet, and hang up after some inactivity timeout.
- But they did not want to manually call, or the office modem was attached to a server, so the phone call should happen automatically.

An interesting problem

- People wanted to call their provider when they wanted to access the Internet, and hang up after some inactivity timeout.
- But they did not want to manually call, or the office modem was attached to a server, so the phone call should happen automatically.

An interesting problem

- People wanted to call their provider when they wanted to access the Internet, and hang up after some inactivity timeout.
- But they did not want to manually call, or the office modem was attached to a server, so the phone call should happen automatically.

Linux setup

- When the modem was connected, the `ppp` daemon would number a `ppp0` interface, and set it as the default gateway.
- Otherwise there was just a single `eth0` linked to the LAN and *no default gateway*.
- How would you know **automatically** that somebody wanted to access the Internet ?
- You should see traffic directed to the Internet gateway.

Linux setup

- When the modem was connected, the `ppp` daemon would number a `ppp0` interface, and set it as the default gateway.
- Otherwise there was just a single `eth0` linked to the LAN and *no default gateway*.
- How would you know **automatically** that somebody wanted to access the Internet ?
- You should see traffic directed to the Internet gateway.

Linux setup

- When the modem was connected, the `ppp` daemon would number a `ppp0` interface, and set it as the default gateway.
- Otherwise there was just a single `eth0` linked to the LAN and *no default gateway*.
- How would you know **automatically** that somebody wanted to access the Internet ?
- You should see traffic directed to the Internet gateway.

Linux setup

- When the modem was connected, the `ppp` daemon would number a `ppp0` interface, and set it as the default gateway.
- Otherwise there was just a single `eth0` linked to the LAN and *no default gateway*.
- How would you know **automatically** that somebody wanted to access the Internet ?
- You should see traffic directed to the Internet gateway.

Linux setup

- When the modem was connected, the `ppp` daemon would number a `ppp0` interface, and set it as the default gateway.
- Otherwise there was just a single `eth0` linked to the LAN and *no default gateway*.
- How would you know **automatically** that somebody wanted to access the Internet ?
- You should see traffic directed to the Internet gateway.

Paradox

- To save money, the modem should have been switched off when unused, so by default there was no ppp device, *i.e.* **no default gateway**.
- To access the Internet, traffic must have been routed to the default gateway.
- Since there was no gateway, nothing would ever run the ppp daemon and the Internet would never work.

Paradox

- To save money, the modem should have been switched off when unused, so by default there was no ppp device, *i.e.* **no default gateway**.
- To access the Internet, traffic must have been routed to the default gateway.
- Since there was no gateway, nothing would ever run the ppp daemon and the Internet would never work.

Paradox

- To save money, the modem should have been switched off when unused, so by default there was no ppp device, *i.e.* **no default gateway**.
- To access the Internet, traffic must have been routed to the default gateway.
- Since there was no gateway, nothing would ever run the ppp daemon and the Internet would never work.

Paradox

- To save money, the modem should have been switched off when unused, so by default there was no ppp device, *i.e.* **no default gateway**.
- To access the Internet, traffic must have been routed to the default gateway.
- Since there was no gateway, nothing would ever run the ppp daemon and the Internet would never work.

Sad picture

Insert your favorite picture of a sad kitten here.

Fake it!

- The solution was to always have a **fake** Internet (default) gateway.
- Since it could not have been a ppp device, it was a `slip` device — *Serial line Internet Protocol*.
- Which was basically a **named pipe**.

Fake it!

- The solution was to always have a **fake** Internet (default) gateway.
- Since it could not have been a ppp device, it was a `slip` device — *Serial line Internet Protocol*.
- Which was basically a **named pipe**.

Fake it!

- The solution was to always have a **fake** Internet (default) gateway.
- Since it could not have been a ppp device, it was a `slip` device — *Serial line Internet Protocol*.
- Which was basically a **named pipe**.

Fake it!

- The solution was to always have a **fake** Internet (default) gateway.
- Since it could not have been a ppp device, it was a `slip` device — *Serial line Internet Protocol*.
- Which was basically a **named pipe**.

Learning by seeing the traffic

- The first packet travelling through the slip device was given as the *standard input* to a command that discarded it and launched the ppp daemon.
- Upon completion of the ppp setup, the default gateway would be switched to the new (working) interface.
- Putting down the ppp interface would put again the slip device as the default gateway.

Learning by seeing the traffic

- The first packet travelling through the slip device was given as the *standard input* to a command that discarded it and launched the ppp daemon.
- Upon completion of the ppp setup, the default gateway would be switched to the new (working) interface.
- Putting down the ppp interface would put again the slip device as the default gateway.

Learning by seeing the traffic

- The first packet travelling through the slip device was given as the *standard input* to a command that discarded it and launched the ppp daemon.
- Upon completion of the ppp setup, the default gateway would be switched to the new (working) interface.
- Putting down the ppp interface would put again the slip device as the default gateway.

Learning by seeing the traffic

- The first packet travelling through the slip device was given as the *standard input* to a command that discarded it and launched the ppp daemon.
- Upon completion of the ppp setup, the default gateway would be switched to the new (working) interface.
- Putting down the ppp interface would put again the slip device as the default gateway.

Index

The problem

Long time ago . . .

Current solutions

EVPN

Conclusions

BIODEC
Evolving ICT Infrastructures

VXLAN

- The current solution to the VLAN problems is called *VXLAN* — *Virtual eXtensible LAN*.
- It is officially documented by the IETF in RFC 7348.
- A common implementation of VXLAN is the *Open vSwitch* software (but **no multicast** as of 2.7.90).

VXLAN

- The current solution to the VLAN problems is called *VXLAN* — *Virtual eXtensible LAN*.
- It is officially documented by the IETF in RFC 7348.
- A common implementation of VXLAN is the *Open vSwitch* software (but **no multicast** as of 2.7.90).

VXLAN

- The current solution to the VLAN problems is called *VXLAN* — *Virtual eXtensible LAN*.
- It is officially documented by the IETF in RFC 7348.
- A common implementation of VXLAN is the *Open vSwitch* software (but **no multicast** as of 2.7.90).

VXLAN

- The current solution to the VLAN problems is called *VXLAN* — *Virtual eXtensible LAN*.
- It is officially documented by the IETF in RFC 7348.
- A common implementation of VXLAN is the *Open vSwitch* software (but **no multicast** as of 2.7.90).

The main idea

- VXLAN is a Layer 2 overlay scheme on a Layer 3 network.
- Each overlay is termed a **VXLAN segment**. Only VMs within the same VXLAN segment can communicate with each other.
- Each VXLAN segment is identified through a 24-bit segment ID, termed the **VNI** – VXLAN Network Identifier.
- This allows up to 16 millions of VXLAN segments to coexist within the same administrative domain.

The main idea

- VXLAN is a Layer 2 overlay scheme on a Layer 3 network.
- Each overlay is termed a **VXLAN segment**. Only VMs within the same VXLAN segment can communicate with each other.
- Each VXLAN segment is identified through a 24-bit segment ID, termed the **VNI** – VXLAN Network Identifier.
- This allows up to 16 millions of VXLAN segments to coexist within the same administrative domain.

The main idea

- VXLAN is a Layer 2 overlay scheme on a Layer 3 network.
- Each overlay is termed a **VXLAN segment**. Only VMs within the same VXLAN segment can communicate with each other.
- Each VXLAN segment is identified through a 24-bit segment ID, termed the **VNI** – VXLAN Network Identifier.
- This allows up to 16 millions of VXLAN segments to coexist within the same administrative domain.

The main idea

- VXLAN is a Layer 2 overlay scheme on a Layer 3 network.
- Each overlay is termed a **VXLAN segment**. Only VMs within the same VXLAN segment can communicate with each other.
- Each VXLAN segment is identified through a 24-bit segment ID, termed the **VNI** – VXLAN Network Identifier.
- This allows up to 16 millions of VXLAN segments to coexist within the same administrative domain.

The main idea

- VXLAN is a Layer 2 overlay scheme on a Layer 3 network.
- Each overlay is termed a **VXLAN segment**. Only VMs within the same VXLAN segment can communicate with each other.
- Each VXLAN segment is identified through a 24-bit segment ID, termed the **VNI** – VXLAN Network Identifier.
- This allows up to 16 millions of VXLAN segments to coexist within the same administrative domain.

A tunnelling solution

- In other words, a VXLAN segment may be seen as a tunnel between two **VTEP** – VXLAN Tunnel End Point
- The key issue is how the table that associates a MAC address to a given VTEP IP address is updated.
- Usually the association of the MAC address to the IP address of the VTEP is discovered via source-address learning.

A tunnelling solution

- In other words, a VXLAN segment may be seen as a tunnel between two **VTEP** – VXLAN Tunnel End Point
- The key issue is how the table that associates a MAC address to a given VTEP IP address is updated.
- Usually the association of the MAC address to the IP address of the VTEP is discovered via source-address learning.

A tunnelling solution

- In other words, a VXLAN segment may be seen as a tunnel between two **VTEP** – VXLAN Tunnel End Point
- The key issue is how the table that associates a MAC address to a given VTEP IP address is updated.
- Usually the association of the MAC address to the IP address of the VTEP is discovered via source-address learning.

A tunnelling solution

- In other words, a VXLAN segment may be seen as a tunnel between two **VTEP** – VXLAN Tunnel End Point
- The key issue is how the table that associates a MAC address to a given VTEP IP address is updated.
- Usually the association of the MAC address to the IP address of the VTEP is discovered via source-address learning.

Learning by seeing the traffic - again

- When a packet for a given MAC flows through the VTEP, it gets forwarded to the VTEP of the segment where the destination MAC belongs.
- The receiving VTEP then updates its own table with the information about the originating MAC address, recording the VTEP where it came from.
- (There are other issues of broadcasting, master controllers, etcetera, but let us skip that).

Learning by seeing the traffic - again

- When a packet for a given MAC flows through the VTEP, it gets forwarded to the VTEP of the segment where the destination MAC belongs.
- The receiving VTEP then updates its own table with the information about the originating MAC address, recording the VTEP where it came from.
- (There are other issues of broadcasting, master controllers, etcetera, but let us skip that).

Learning by seeing the traffic - again

- When a packet for a given MAC flows through the VTEP, it gets forwarded to the VTEP of the segment where the destination MAC belongs.
- The receiving VTEP then updates its own table with the information about the originating MAC address, recording the VTEP where it came from.
- (There are other issues of broadcasting, master controllers, etcetera, but let us skip that).

Learning by seeing the traffic - again

- When a packet for a given MAC flows through the VTEP, it gets forwarded to the VTEP of the segment where the destination MAC belongs.
- The receiving VTEP then updates its own table with the information about the originating MAC address, recording the VTEP where it came from.
- (There are other issues of broadcasting, master controllers, etcetera, but let us skip that).

Index

- The problem
- Long time ago . . .
- Current solutions
- EVPN
- Conclusions

The main idea

- What if there were things like the VTEPs, but with the feature of being able to exchange information about their ARP table, in real time ?
- Updating their peers without having to wait for traffic to come through to learn the existence of a new MAC ?
- Does there exists a technology to exchange a kind of *routing information* among peers ?

The main idea

- What if there were things like the VTEPs, but with the feature of being able to exchange information about their ARP table, in real time ?
- Updating their peers without having to wait for traffic to come through to learn the existence of a new MAC ?
- Does there exists a technology to exchange a kind of *routing information* among peers ?

The main idea

- What if there were things like the VTEPs, but with the feature of being able to exchange information about their ARP table, in real time ?
- Updating their peers without having to wait for traffic to come through to learn the existence of a new MAC ?
- Does there exists a technology to exchange a kind of *routing information* among peers ?

The main idea

- What if there were things like the VTEPs, but with the feature of being able to exchange information about their ARP table, in real time ?
- Updating their peers without having to wait for traffic to come through to learn the existence of a new MAC ?
- Does there exists a technology to exchange a kind of *routing information* among peers ?

BGP

- In fact there are many alternatives, because the same problem **for IP networks** has been addressed since the birth of the Internet.
- One of the most known and widely used protocol is the *Border Gateway Protocol* (BGP)

BGP

- In fact there are many alternatives, because the same problem **for IP networks** has been addressed since the birth of the Internet.
- One of the most known and widely used protocol is the *Border Gateway Protocol* (BGP)

BGP

- In fact there are many alternatives, because the same problem **for IP networks** has been addressed since the birth of the Internet.
- One of the most known and widely used protocol is the *Border Gateway Protocol* (BGP)

EVPN specifications

- “A Network Virtualization Overlay Solution using EVPN” – which is the Internet Draft `draft-ietf-bess-evpn-overlay`
- “Requirements for Ethernet VPN (EVPN)” – `rfc7209`
- “BGP MPLS-Based Ethernet VPN” – `rfc7432`

Using EVPN

- EVPN is currently supported on some vendor network switching equipment.
- There is an implementation called *bagpipe-bgp* now part of OpenStack.

Using EVPN

- EVPN is currently supported on some vendor network switching equipment.
- There is an implementation called *bagpipe-bgp* now part of OpenStack.

Using EVPN

- EVPN is currently supported on some vendor network switching equipment.
- There is an implementation called *bagpipe-bgp* now part of OpenStack.

Index

The problem

Long time ago . . .

Current solutions

EVPN

Conclusions

Keep an eye on this technology

- If your use cases include *live migrating a container across datacenters* . . .
- then EVPN is something to track.
- It could be that you will use EVPN nonetheless, if it gets included in other project (see bagpipe-bgp and OpenStack).

Keep an eye on this technology

- If your use cases include *live migrating a container across datacenters* . . .
- then EVPN is something to track.
- It could be that you will use EVPN nonetheless, if it gets included in other project (see bagpipe-bgp and OpenStack).

Keep an eye on this technology

- If your use cases include *live migrating a container across datacenters* . . .
- then EVPN is something to track.
- It could be that you will use EVPN nonetheless, if it gets included in other project (see bagpipe-bgp and OpenStack).

Keep an eye on this technology

- If your use cases include *live migrating a container across datacenters* . . .
- then EVPN is something to track.
- It could be that you will use EVPN nonetheless, if it gets included in other project (see bagpipe-bgp and OpenStack).

Questions

?

Thanks !

Email m@biodec.com

Useless social @gaunilone

4DevOps.ch conference Last-Minute HQ, Corso San Gottardo 30, 6830, Chiasso, Switzerland, 24 of May 2017
http://4devops.ch/

4DevOps.ch workshops Palazzo dei Congressi, Piazza Indipendenza 4, 6900 Lugano, Switzerland, 23 of May 2017 — the day before the conference.