HSR Cloud Infrastructure Lab 6

Analyze VXLAN and TRILL

Emanuel Duss, Roland Bischofberger

2014-11-13

Inhaltsverzeichnis

1 Einführung

Wir beschäftigen uns mit VXLAN und TRILL. Abgabe bis 13.11.2014 23:59h an Beat Stettler beat.stettler@ins.hsr.ch.

2 Aufgabe 1: Anforderungen an ein modernes Rechenzentrum

2.1 Herausforderungen

Die Herausforderungen in einem modernen Rechenzentrum sind die immer steigenden Datenmengen, welche generiert werden und verarbeitet werden müssen. Einerseits müssen die Server eine genügend grosse Leistung erbringen. Aber ohne ein Netzwerk, welches mit einer immer wachsenden Datenmenge umgehen kann, bringen auch die stärksten Server nichts.

In grossen Datencentern fliesst viel Traffic unter den Servern hin- und her. Beispielsweise besucht ein Kunde die Webseite von einem Anbieter. Ein simpler HTTP GET Request kann eine ganze Kette von Aktionen ins rollen bringen, welche viel Traffic zwischen den Servern verursacht.

Beispielsweise kommuniziert ein Kunde mit dem Webserver um etwas zu bestellen. Der Webserver muss dann in der Businesslogik die Produkte und deren Verfügbarkeit abfragen. Dazu muss der Server, welcher für die Businesslogik zuständig ist auf einen Datenbankserver zugreifen. Nicht zu vergessen sind auch die Backups, welche gemacht werden müssen, oder das verschieben von virtuellen Maschinen innerhalb eines Datacenters (oder sogar in ein remote Datacenter).

Im nächsten Abschnitt werden die Anforderungen an ein modernes Datacenter aufgelistet.

2.2 Anforderungen

Bei der Konstruktion von einem neuen Datacenter muss man einges beachten:

- Rechtliche Compliance
- Abhängigkeit
- Datenschutz / Ort der Speicherung
- Schnelles Netz
- Sicherheit
- Skalierbarkeit
- Verfügbarkeit

2.3 Netzwerkarchitektur

- Skalierbar
- Schneller Datendurchsatz zwischen den Servern
- Multinenancy (Abtrennung von mehreren Kunden auf dem selben Netz)

3 Aufgabe 2: VXLAN

3.1 Terminologie:

VLAN Virtual Local Area Network

VM Virtual Machine

VNI VXLAN Network Identifier (or VXLAN Segment ID)

VTEP VXLAN Tunnel End Point. An entity that originates and/or terminates VXLAN tunnels

VXLAN Virtual eXtensible Local Area Network

VXLAN Gateway an entity that forwards traffic between VXLANs or non-VXLAN-cappable devices

3.2 Funktion und Header

3.2.1 Funktion

VXLAN steht für "Virtual eXtensible Local Area Network" und ist im RFC 7348 spezifiziert [1].

Es erlaubt die Erweiterung von Layer 2 Netzwerken über Layer 3 Netzwerke. Dies kommt vor allem der Anforderung entgegen, dass die Server von einem Tenant innerhalb des Datacenters oder sogar über mehrere Datacenter so arrangiert werden können, dass CPU, Netzwerk und Speicherressourcen optimal verteilt und genutzt werden können.

Ein Problem ohne VXLAN kann sein, dass die einzelnen Tenants intern eigene MAC Adressen und VLAN IDs benutzen mit denen von anderen Tenants kollidieren würden. Dies kann mit einer VXLAN-Enkapsulierung verhindert werden. Jedes VXLAN wird mit einem VNI idenzifiziert, welches wie bei VLAN als ID für das jeweilige Layer 2 Netzwerk fungiert.

VXLAN Pakete dürfen nicht fragmentiert werden. Deshalb muss darauf geachtet werden, dass im Layer 3 Netzwerk zwischen den VTEPs die MTU genügend gross ist, damit die Pakete nicht fragmentiert und somit von dem Empfänger VTEP verworfen werden. Als VTEP kann ein Switch oder auch ein Server fungieren, welche VXLAN fähig sind. Dabei können die Geräte physisch wie auch virtuell sein.

Welche VM sich hinter welchem VTEP befindet, wird dynamisch und automatisch gelernt. Der folgende Ablauf beschreibt kurz, wie die Verbindung zwischen zwei VMs, welche dem gleichen VXLAN zugewiesen sind, hergestellt wird.

0. VM1 möchte mit VM2 per IP kommunizieren und befindet sich im gleichen Subnetz
1. VM1 sendet einen ARP Multicast Request für VM2
2. VTEP1 nimmt den Request entgegen und verpackt diesen in ein VXLAN Paket und sendet einen IP Multicast raus
3. VTEP2 erhält Multicast und lernt, dass VM1 unter der IP von VTEP1 erreichbar ist
4. VM2 erhält ARP Layer2 Paket und bekommt nichts von dem VXLAN zwischendurch mit.

Zum obigen Ablauf ist noch zu ergänzen, dass jedes VXLAN basierend auf einer Multicastgruppe realisiert ist, somit die VXLAN Pakete jeweils innerhalb Layer 3 Netzes zwischen den VTEPs auf eine Multicastgruppe "gemappt" werden.

Wenn nun eine VM von einem POD in einen anderen gezügelt wird, wird diese einen ARP Request aussenden und somit wissen die VTEPs wieder wo diese VM zu finden ist. Die VTEPs speichern die Zuweisung, welche VM MAC, in welchem VXLAN unter welchem VTEP erreichbar ist in einer Tabelle ab. Diese Einträge haben eine bestimmte TTL. Nach dieser TTL wird der Eintrag aus der Tabelle entfernt.

Um Pakete aus einem VXLAN in ein anderes VXLAN oder auch ein VLAN weiterzuleiten, wird ein VXLAN Gateway benötigt, welcher den VXLAN Header ausgehend entfernt und eingehend unter Umständen hinzufügt. Somit können auch Pakete aus einem VXLAN in ein Netz gesendet werden, welches nicht VXLAN fähig ist.

[1] https://tools.ietf.org/html/rfc7348

3.2.2 Header

Das gesamte L2 Paket wird in UDP verpackt. Dies führt zu 50 Bytes Overhead.

In folgender Figur ist ein gesamtes Paket aufgezeigt. Dabei ist schön aufgezeigt, dass das originale Ethernet Paket in ein UDP Paket verpackt wird und dazwischen noch der VXLAN Header zu finden ist.

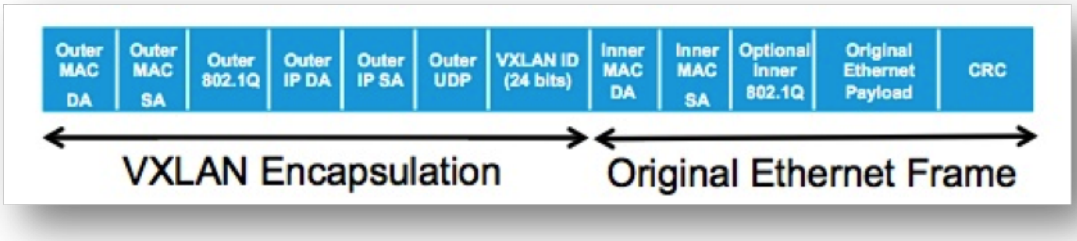


Abbildung 1: VXLAN Paket

Der 8 Byte Header des VXLAN ist wie folgt aufgebaut.

- Flags: 8 bit
- Reserve Felder: 24bit
- VXLAN Network Identifier (VNI): 24 bit
- Reservefelder: 8 bit

VXLAN Header

```
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|R|R|R|R|I|R|R|R|            Reserved                           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                VXLAN Network Identifier (VNI) |   Reserved    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

3.3 Ziel von VXLAN und warum wurde VXLAN entwickelt?

VXLAN wurde explizit für den Betrieb von VMs in einer Cloudumgebung entwickelt. Mit VXLAN kann eine multitenancyfähigkeit mit einer möglichst grossen Mobilität auch über mehrere Datenzentren gewährleistet werden. Es kann somit gewährleistet werden, dass mehrere über verschiedene Datenzentren verteilte VMs sich im gleichen Layer 2 Netzwerk befinden.

3.4 Wo kann VXLAN genutzt werden? (Use Cases)

Folgt einige spezifische Anwendungsbeispiele von VXLAN:

- In einem Datenzentrum werden Virtualisierungsserver eingesetzt. Nun kommt eine neue VM dazu wobei der physische Virtualisierungsserver keine weiteren Kapazitäten hat. Jetzt können einige VMs von diesem Virtualisierungsserver auf einen anderen gezügelt werden und dank VXLAN bleiben diese im gleichen Layer 2 Netzwerk.
- Die Ressourcen eines Rechnerverbundes können optimal auf die Bedürfnisse an CPU, Storage oder Bandbreite angepasst werden, weil die VMs beliebig platziert werden können.

3.5 Ist VXLAN mit traditionellen Protokollen vergleichbar?

3.5.1 VLAN

Mit VLANs kann auch Multitenancy sichergestellt werden. Jedoch können VLANs nicht über Layer3-Grenzen verteilt werden. Es können auch weniger Tenants gebildet werden. 802.1q kann rund 4k Tenants wobei VXLAN bis zu 16M Tenants unterstützt.

Ein Nachteil von VXLAN gegenüber 802.1q ist, dass es einen relativ grossen Overhead produziert, indem das Ethernet Paket noch in ein UDP/IP Paket verpackt wird.

3.5.2 L2TP

VXLAN ist funktional ähnlich wie ein Layer 2 Tunnel über ein IP Netzwerk. Jedoch agiert VXLAN eher wie ein Overlaynetzwerk und nicht wie ein Punkt zu Punkt Tunnel wie z.B. L2TP. Der Vorteil von VXLAN ist jedoch, dass VXLAN viel dynamischer eingesetzt werden kann, da nicht jede Punkt zu Punkt Verbindung von einem Endpunkt zum anderen einzeln definiert werden muss. Die Enpunkte werden in einem Learning Verfahren, wie im Abschnitt Funktion beschrieben, gelernt. Somit muss nicht jedesmal wenn eine VM gezügelt wird, ein Tunnel angepasst werden.

3.5.3 MPLS

MPLS bietet auch die Möglichkeit Layer 2 Pakete zu tunneln. Jedoch sind die MPLS Komponenten teurer und wird auch MPLS nicht von den virtuellen Switches bzw. virtualisierungs Servern unterstützt, was somit eine Terminierung des Tunnels auf diesen Geräten verunmöglicht. VXLAN ist auch hier durch das automatische Lernen der Endpunkte klar im Vorteil gegenüber MPLS.

Falls ja, was sind die Limitationen und Unterschiede? Falls ja, welche Limitationen und Unterschiede?

3.6 Vorteile / Nachteile von VXLAN

Vorteile:

- Layer 2 kann über ein Layer 3 Netzwerk gespannt werden
- Dynamisch erweiterbar durch die Definition von neuen VXLANs
- Tunnel müssen nicht alle einzeln definiert werden

Nachteile:

- relativ grosser Overhead
- Noch nicht von allen Herstellern unterstützt

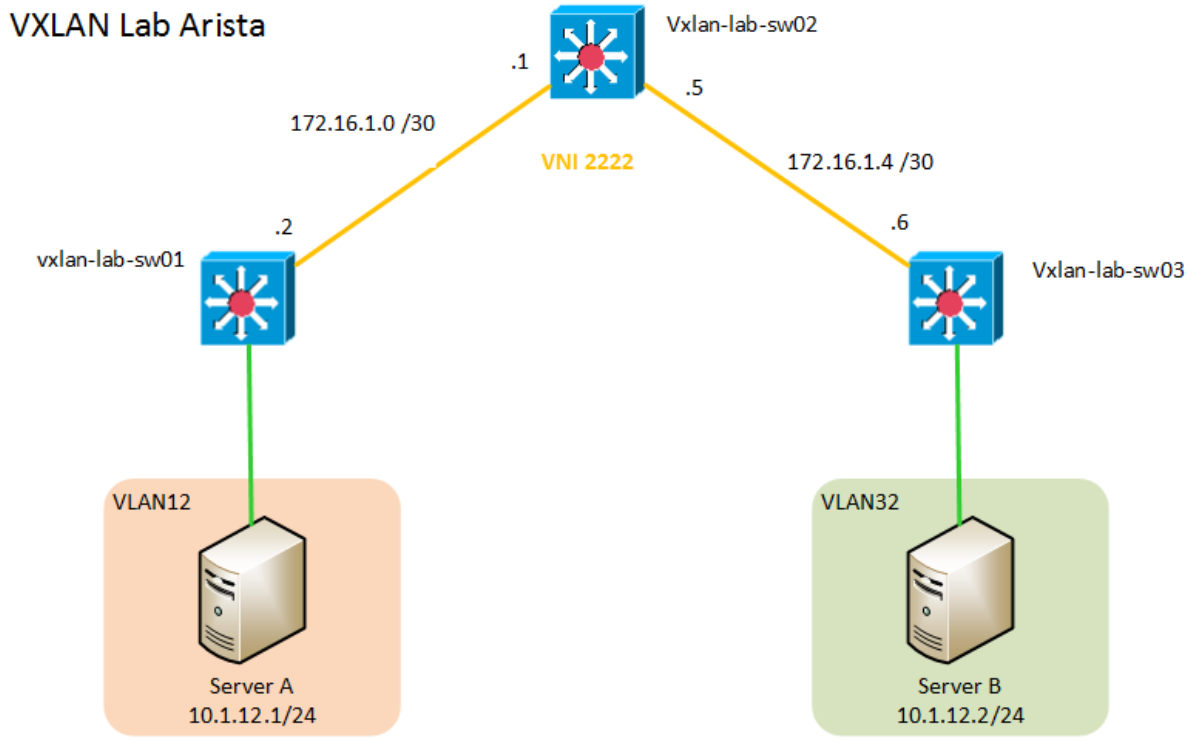
3.7 Andere moderne Technologien wie VXLAN?

3.7.1 NVGRE

NVGRE ist funktional zu VXLAN identisch. Es übernimmt die gleichen Funktionen und hat auch 24bit um die NetworkID festzulegen. Jedoch nutzt NVGRE Generic Routing Encapsulation(GRE) um andere Protokolle einzukapseln und diese in Form eines Tunnels über ein IP Netzwerk zu transportieren. Der Overhead bewegt sich im gleichen Rahmen wie VXLAN. Der grösste Unteschied der Protokolle ist, dass Sie von anderen Firmen unterstützt und gepusht werden.

3.8 VXLAN im Lab

Wir haben im Lab zum eine VXLAN mit Arista Switches aufgebaut. Das Lab ist wie folgt aufgebaut:



Wir haben im Lab eine TCP Verbindung vom Server B zum Server A aufgebaut. Dabei haben wir dann über einen Monitoring Port auf dem vxlan-lab-sw02 die Pakete mitgeschnitten. Dies ist ein Protokollstack eines solches Pakets:

```
Frame 47: 110 bytes on wire (880 bits), 110 bytes captured (880 bits) on interface 0
Ethernet II, Src: AristaNe_69:17:71 (00:1c:73:69:17:71), Dst: AristaNe_69:18:dd (00:1c:73:69:18:dd)
Internet Protocol Version 4, Src: 172.16.2.6 (172.16.2.6), Dst: 172.16.2.2 (172.16.2.2)
User Datagram Protocol, Src Port: 50765 (50765), Dst Port: vxlan (4789)
Virtual eXtensible Local Area Network
Ethernet II, Src: FujitsuT_af:97:92 (00:19:99:af:97:92), Dst: FujitsuT_af:97:ab (00:19:99:af:97:ab)
Internet Protocol Version 4, Src: 10.1.12.2 (10.1.12.2), Dst: 10.1.12.1 (10.1.12.1)
Transmission Control Protocol, Src Port: miva-mqs (1277), Dst Port: commplex-link (5001), Seq: 1, Ack: 1, Len: 0
```

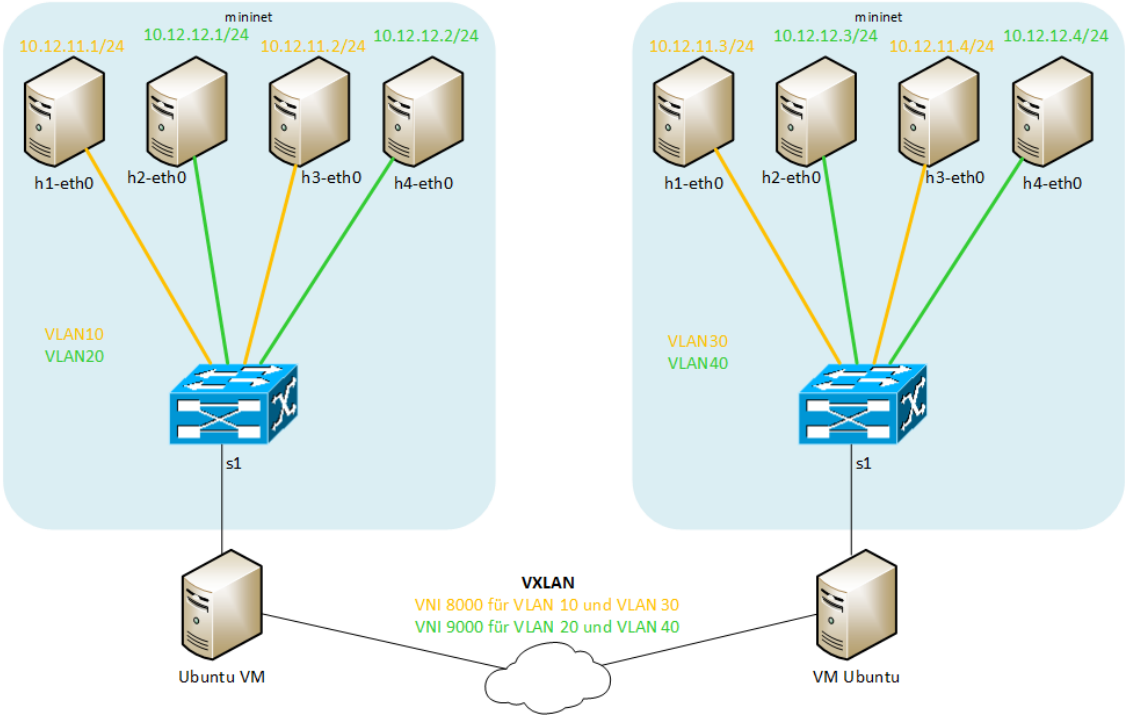
Dabei sieht man schön dass die TCP Pakete in ein VXLAN UDP Paket eingepackt wird. Wir haben hier farbig markiert, welcher Teil des Paketes in welchem Netzwerkabschnitt ausschlaggebend ist. Die Farben der Markierungen entsprechen der Farbe der Verbindungen auf der logical Map.

3.8.1 VXLAN mit Mininet

Ausserdem haben wir mit Mininet zwei kleine Netze virtuell simuliert. Dabei wurden die Hosts wie auch der Switch virtualisiert. Die Switches fungierten dabei als VTEP. Die Hosts sind über ein VLAN verbunden, wobei diese VLANs jeweils auf einen bestimmten VNI gemappt werden. Die Hosts können sich nun innerhalb des einen mininet erreichen, wenn Sie im gleichen VLAN sind. Auch können sich die Hosts nun auch "mininet-übergreifend" erreichen. Jedoch geht dies nur bei den Hosts, deren VLAN auf den gleichen VNI gemappt ist.

Dies ist die Übersicht über das Lab:

VXLAN Lab mininet



4 Aufgabe 3: TRILL

4.1 Funktion und Header

4.1.1 Funktion

Trill steht für "Transparent Interconnection of Lots of Links" und wurde 2011 im RFC 6325[2] spezifiziert. Updates zu diesem Standard gab es im Mai 2014 im RFC 7172[3].

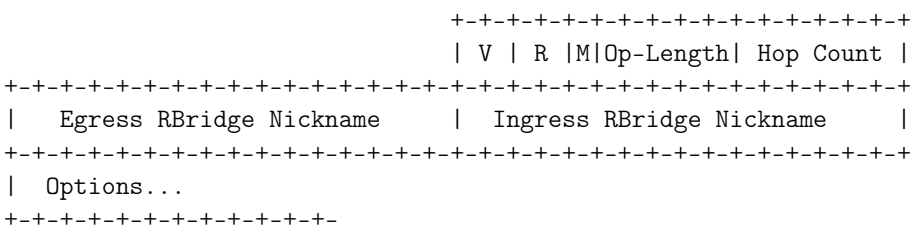
Zur Terminologie: Die Geräte, welche TRILL implementieren, nennt man RBridges.

Die Funktion von TRILL ist es auf Layer 2 ein Routing zu betreiben. Betreibt man auf dem Layer 2 ein Routing, ist kein Spanning Tree Protocol mehr nötig. Der grösste Vorteil darin ist, dass es keine blockierten Ports mehr gibt und das Netz schneller konvergiert.

4.1.1.1 RBridge IDs Die RBridges haben eine eindeutige ID, welche man Bridge Identifier nennt und im TRILL Header 2 Bytes beansprucht. Somit sind in einem solchen Layer 2 Netz maximal $2^{64} = 64k$ RBridges möglich. Die RBridges wählen ihre Bridge Identifier entweder dynamisch aus oder sie werden durch den Administrator statisch gesetzt.

4.1.2 Header

4.1.2.1 Header Felder Der Header von TRILL sieht so aus (aus dem RFC 6325):



Die wichtigsten Felder aus dem Header:

- V: Versionsnummer. 2 Bit. Aktuell: Version 0
- R: Reserved: Reservierte 2 Bit für die Zukunft
- M: Multi-Destination. 1 Bit. 0 = Unicast; 1 = Distribution Tree
- Hop Count: 6 Bit. Hop Count wird bei jeder RBridge dekrementiert. Bei 0 wird das Paket verworfen.
- Egress/Ingress RBridge Nicknames: 2 Byte. Eindeutige ID für eine Bridge.
  - Egress: Source RBridge
  - Ingress: Destination RBridge

4.1.2.2 Loop Prevention Durch den Hop Count im TRILL Header werden Loops verhindert. Die RBridges sollen das Hop Count Feld auf die erwartete Anzahl Hops stellen.

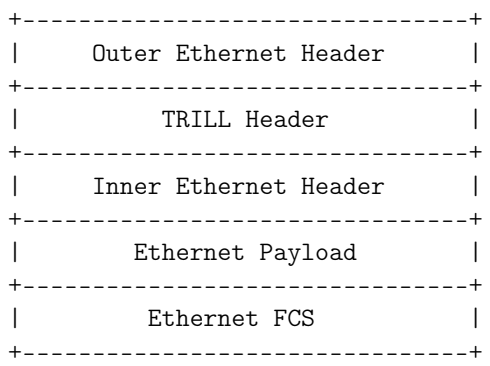
4.1.2.3 Nicknames Die Nicknamen der RBridges wählen diese selber aus. Sie müssen darauf achten, dass es keine Konflikte gibt. Daher wird oft die Systemzeit, das Datum und weitere Entropie in die 2 Byte ID miteinbezogen, um Kollisionen vorzubeugen. Nach einem Reboot sollten die RBridges den Nicknamen behalten. - 0x00 bedeutet dass der Nickname unbekannt ist - Es dürfen Nicknames zwischen 0xFFC0 und 0xFFFF ausgewählt werden.

[2] http://tools.ietf.org/html/rfc6325
[3] https://tools.ietf.org/html/rfc7172

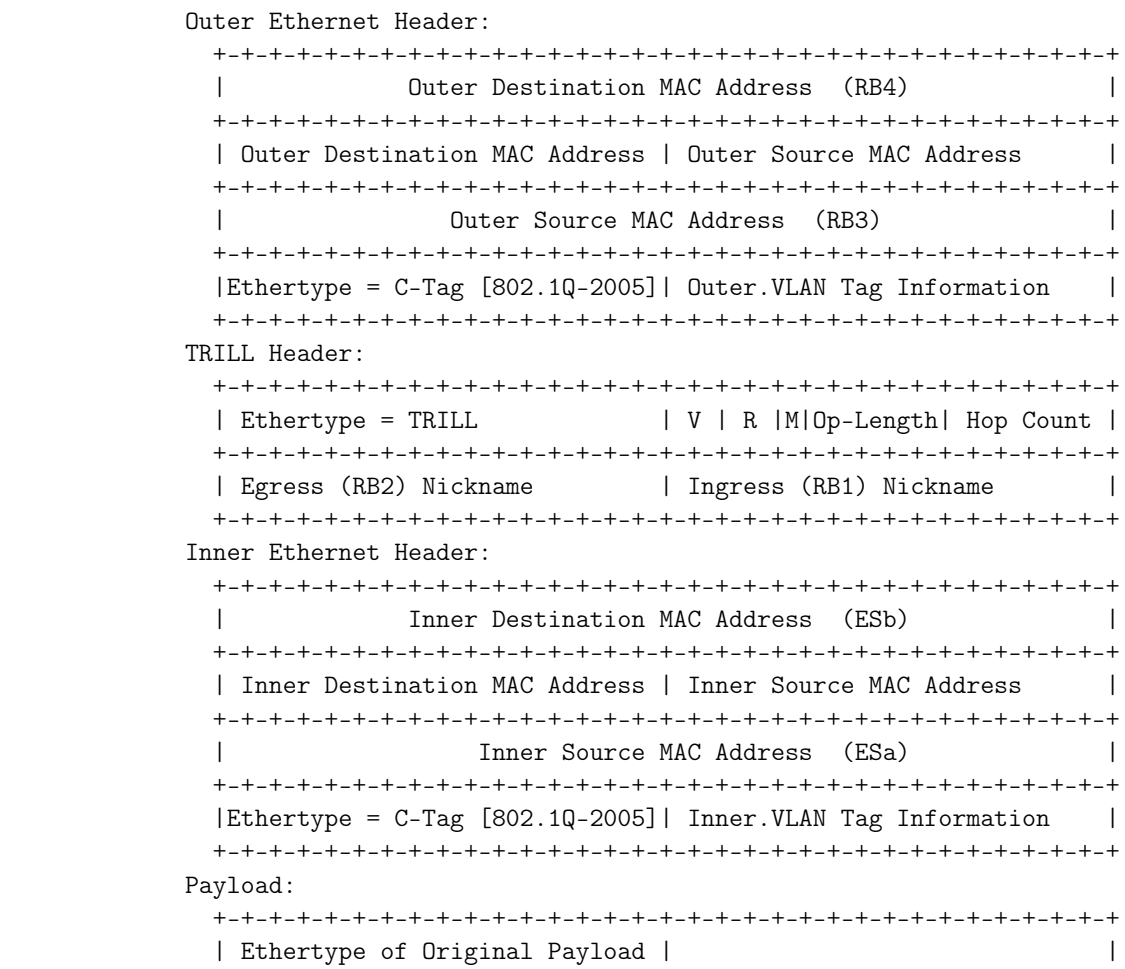
4.1.3 Kaplusierung von Ethernet Frames

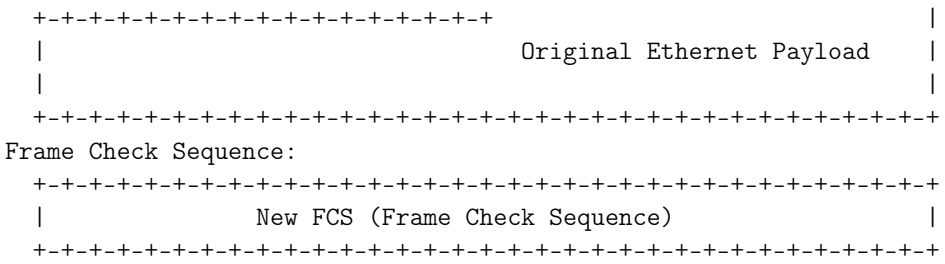
Die erste RBridge welche ein Ethernet Frame von einem Client erhält nennt man ingress Bridge. Die ingress Bridge packt das Ethernet Frame in ein TRILL Paket ein. Dabei wird der TRILL Header (siehe unten) eingefügt. In diesem TRILL Header wird die last RBridge spezifiziert. Die last RBridge muss das Ethernet Frame wieder auspacken. Diese last Bridge nennt man auch egress Bridge und ist mit der Destination IP Adresse eines Routers (Layer 3) vergleichbar.

Dabei wird das Ethernet Paket folgendermassen eingekapselt (Bild aus RFC):



Hier eine etwas detailliertere Aufstellung (ebenfalls auf dem RFC). Folgendes Paket wurde zwischen zwei transit RBridges aufgezeichnet, welche die TRILL Frames ohne Veränderung weiterleiten.





Es gibt neu zwei Ethernet Header: Der outer Ethernet Header wird für die Übertragung des eigentlichen Ethernet Pakets verwendet. Dieses beinhaltet den TRILL Header, welcher über den Ethertype 0x8100 (C-Tag) spezifiziert ist. Im TRILL Header stehen die ingress und engress (Absender und Empfänger) RBridges, welche das zu übertragende Ethernet Frame vom Client ein- bzw. dekapsulieren. Der Inner Ethernet Header kommt aus dem original Ethernet Frame vom Client. Dieses Ethernet Frame wird mittels TRILL auf Layer 2 geroutet.

Bei einer transit RBridge, welche die Pakete nur weiterleiten, wird nur der äussere Ethernet Heaer verändert. Der TRILL Header bleibt gleich. Auch die VLAN Tags bleiben beibehalten.

4.1.4 VLANs

Es können weiterhin VLANs verwendet werden. Es können beide Ethernet Frames (Outer und Inner Ethernet Frame) mit einem VLANTag ausgerüstet werden. Hierbei spricht man vom Inner VLAN Tag und vom Outer VLAN Tag. Das Inner VLAN Tag ist das VLAN vom eigentlichen VLAN, über welches die ingress RBridge das Ethernet Frame engegengenommen hat. Dieses VLAN Tag wird von den RBridges nie geändert. Das Outer VLAN Tag wird von der Designated RBridge bestimmt und bestimmt somit das VLAN auf dem die TRILL Frames übertragen werden.

4.1.5 Frames

In TRILL gibt es fünf verschiedene Frames, welche auf Layer 2 operieren:

- Control Frames (Brudge BPDU)
- TRILL Data Frames
- TRILL Control Frames
- weitere TRILL Frames

4.1.6 Link State Protocol (IS-IS)

TRILL verwendet IS-IS als Routing Protokoll. Dabei erstellt TRILL eine separate IS-IS Instanz und ist von einer eventuell bestehenden Layer 3 IS-IS Instanz komplett losgelöst. Schön an TRILL ist, dass dies für den Administrator unsichtbar ist, das heisst, er muss quasi keine Ahnung über IS-IS haben und kann trotzdem TRILL konfigurieren. Es wird IS-IS verwendet, da dies direkt auf Layer 2 läuft und somit keine Layer 3 Informationen wie IP Adressen benötigt. IS-IS ist durch das gute Design einfach erweiterbar.

Jede RBridge hat eine eindeutige 48 Bit IS-IS ID, welche praktischerweise von der MAC Adresse abgeleitet wird.

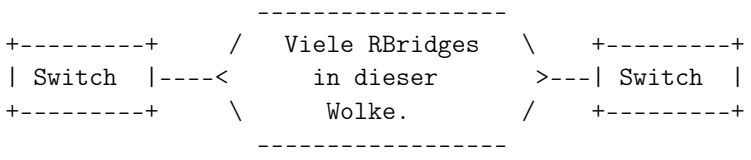
Die Designated Bridge wird über das TRILL-IS-IS Protokoll ausgehandelt. Dieses Protokoll versendet Ethernet Frames an Multicast Adressen (All-IS-IS-RBridges), ebenfalls mit dem C-Tag Ethertype. Damit sich die RBridges gegenseitig finden, senden diese periodisch TRILL-Hello Frames raus.

Die RBridges bilden über das IS-IS Protokoll Bäume, so dass alle egress Bridges erreichbar sind. Sind zu einer Bridge mehrere Pfade möglich, werden diese (im Gegensatz zu STP) auch genutzt. Somit hat man mehr Bandbreite zur Verfügung, die ausgeschöpft werden kann.

4.2 Ziel von TRILL

Das Ziel von TRILL ist es auf Layer 2 ein Routing zu betreiben. Dabei spricht man häufig auch von einem MAC Address Routing. Betreibt man auf dem Layer 2 ein Routing, ist kein Spanning Tree Protocol mehr nötig. Der grösste Vorteil darin ist, dass es keine blockierten Ports mehr gibt und das Netz schneller konvergiert.

Ein weiteres Ziel von TRILL war es, TRILL einfach in bestehende Netze zu integrieren. Layer 2 Geräte, welche TRILL nicht verstehen, sehen (möglicherweise vielen) TRILL RBridges als eine "Wolke":



4.3 Warum wurde TRILL entwickelt?

Das Spanning Tree Protocol (STP) wurde von Radia Perlmann designed. STP ist für Datacenter jedoch nicht sehr gut geeignet. STP bildet einen Tree und blockiert die Pfade, welche im Tree nicht genutzt werden. Durch das blockieren dieser Ports wird redundanz gewährleistet. Fällt ein Link zwischen zwei Switches aus, wird dies bemerkt und ein bis anhin blockierter Pfadi wird aktiviert. STP konvergiert nicht sehr schnell, und auch die verbesserten Varianten wie Rapid STP (RSTP) haben ihre Mängel, denn auch RSTP blockiert Links im Netz. Die Bandbreite der blockierten Links kann ebenfalls nicht ausgenutzt werden.

Die selbe Frau, welche STP designed hat, erfand auch TRILL. TRILL steht für Transparent Interconnection of Lots of Links. TRILL ist ein kompletter Ersatz für STP. Wird in einem Netz TRILL eingesetzt, kann komplett auf STP verzichtet werden. Die Geräte, welche TRILL implementieren werden RBridges genannt. Bei TRILL gibt es keine blockierten Ports mehr. Zu einem Ziel kann jede RBridge 16 Pfade haben. Jede RBridge kann alle Pfade nutzen. Somit kann mehr Bandbreite gegenüber von STP erreicht werden. Aus diesem Grund eignet sich TRILL auch vorallem in Datacenter, da oft viel Traffic innerhalb des Datacenters fliesst. Durch das so genannte Equal Cost Multipath Routing (ECMP) werden alle Pfade mit den gleichen tiefsten Kosten zu einem Ziel in die Routingtabelle eingetragen und können deshalb parallel ausgenutzt werden.

4.4 Wo kann TRILL genutzt werden? (Use Cases)

4.4.1 TRILL in Datacentern mit hoher Bandbreite

Da jede RBridge zu einem Ziel bis zu 16 Pfade nutzen kann, ist eine viel höhere Bandbreite als bei STP nutzbar. Kombiniet man TRILL beispielsweise mit Port Channels können noch schnellere Bandbreiten zwischen den RBridges erreicht werden. In Datacentern fliesst viel Traffic innerhalb des Datacenters hin- und her. Würde man STP einsetzen, könnte man die Links der blockierten Ports nicht nutzen. Bei TRILL hingegen sind alle Pfade zwischen den RBridges nutzbar.

4.4.2 TRILL in Netzen mit nicht TRILL fähigen Geräten

Befinden sich im selben Layer 2 Netz weitere Geräte, welche TRILL nicht unterstützen, können diese trotzdem eingesetzt werden. Die TRIL Geräte sprechen untereinander weiterhin TRILL. Die Geräte, welche kein TRILL sprechen, werden an die TRILL "Wolke" angeschlossen. Die nicht TRILL Fähigen Geräte erkennen die TRILL Wolke als ein einziger Switch und können mit diesem STP sprechen. (Vgl. Figur in "Ziele von TRILL".)

4.5 Ist TRILL mit traditionellen Protokollen vergleichbar?

4.5.1 Spanning Tree Protocol

TRILL ist mit dem Spanning Tree Protocol (STP) vergleichbar.

Limitationen:

- TRILL hat gegenüber STP keine Limitationen, da TRILL auch dafür designed wurde STP zu ersetzen.

Unterschiede:

- TRILL hat im gegensatz zu STP ein Hop Count feld, welches Loops verhindert.
- Gehen Spanning Tree Messages verloren können bei STP trotzdem temporäre Loops entstehen.

4.5.2 IS-IS

TRILL verwendet IS-IS als Routingprotokoll.

Limitationen:

- TRILL wird nur auf Layer 2 eingesetzt. IS-IS wird hingegen als Routing Protokoll auf Layer 3 eingesetzt.

Unterschiede:

- IS-IS ist sehr alt (wurde 1990 im RFC 1142 spezifiziert)
- Es wurden neue IS-IS Message Typen eingeführt (wie z.B. TRILL Link Hello, TRILL Neighbor, TRILL MTU Probe und noch weitere. . .)

4.6 Vorteile / Nachteile von TRILL

4.6.1 Vorteile von TRILL

- Spanning Tree ist nicht mehr notwendig
- Schnellere Konvergenzzeit (da kein STP mehr)
- Alle Links sind up und aktiv (da kein STP mehr)
- Kein Wissen über IS-IS notwendig
- Skaliert sehr gut (viele neue RBridges sind in bestehende Infrastruktur gut integrierbar)

4.6.2 Nachteile von TRILL

- Nicht alle Geräte unterstützen TRILL
- Hersteller implementierten oft eine eigene leicht angepasste Implementierung (Bei Cisco heisst es Fabric Path)

4.7 Andere moderne Technologien wie TRILL?

Falls ja, was sind die Limitationen und Unterschiede?

Als Alternative zu TRILL bietet sich Shortest Path Bridging (SPB) an. SPB wurde von IEEE als Erweiterung von VLAN entwickelt. Das Ziel von SPB ist, dem Spanning Tree Protocol eine schnellere Konvergenz zu ermöglichen. Der optimale Weg wird ebenfalls wie bei TRILL über IS-IS berechnet. Auch bei SPB können mehrere Pfade zum selben Ziel gleichzeitig genutzt werden. Bei SPB wird im Gegensatz zu TRILL weiterhin geswitcht und kein Routing durchgeführt. SPB setzt auf bereits bestehende Header Formate. Deshalb könnten moderne Switches über ein OS Update auf SPB aufgerüstet werden. Bei TRILL ist eine komplett neue Implementation des TRILL Headers nötig.

4.8 Wie wurde TRILL im LAB konfiguriert?

4.8.1 Konfiguration der Server

Hostname	IP Adresse
Server1	10.1.12.1
Server2	10.1.12.2

- Es ist kein Default Gateway nötig, da nicht geroutet werden muss.
- Es ist kein Nameserver nötig, da keine Namensauflösung gemacht werden muss.

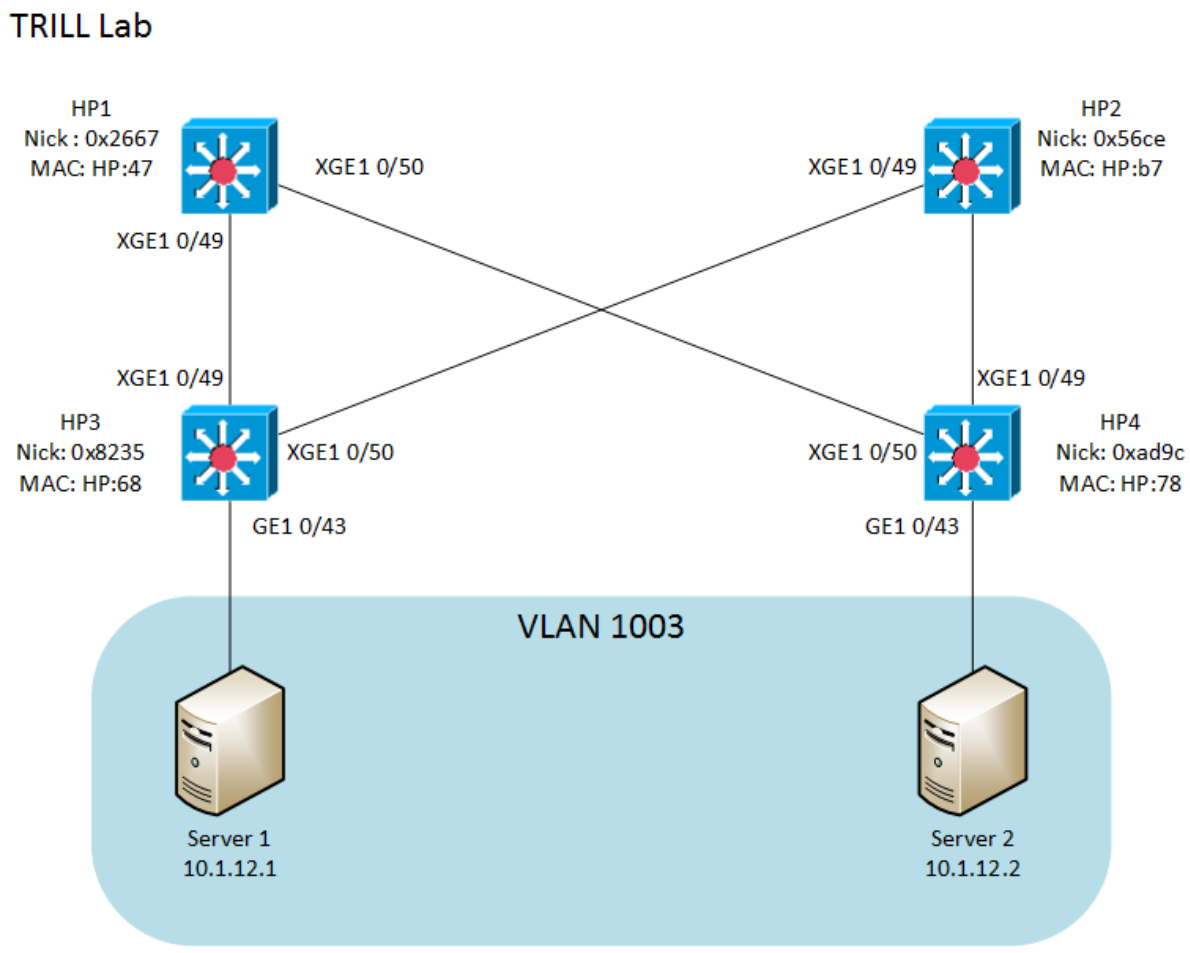
4.8.2 Konfiguration der Switches

Hostname	TRILL Nickname	IP Adresse (nur für SSH Zugriff verwendet)
HP1	0x2b67	10.5.0.11
HP2	0x56ce	10.5.0.12
HP3	0x8235	10.5.0.13
HP4	0xad9c	10.5.0.14

4.8.3 VLAN

Unsere Gruppe verwendet für den Laborversuch das VLAN 1003.

4.8.4 Laboraufbau



4.8.5 Trees

Die Switches haben folgende Neighbors:

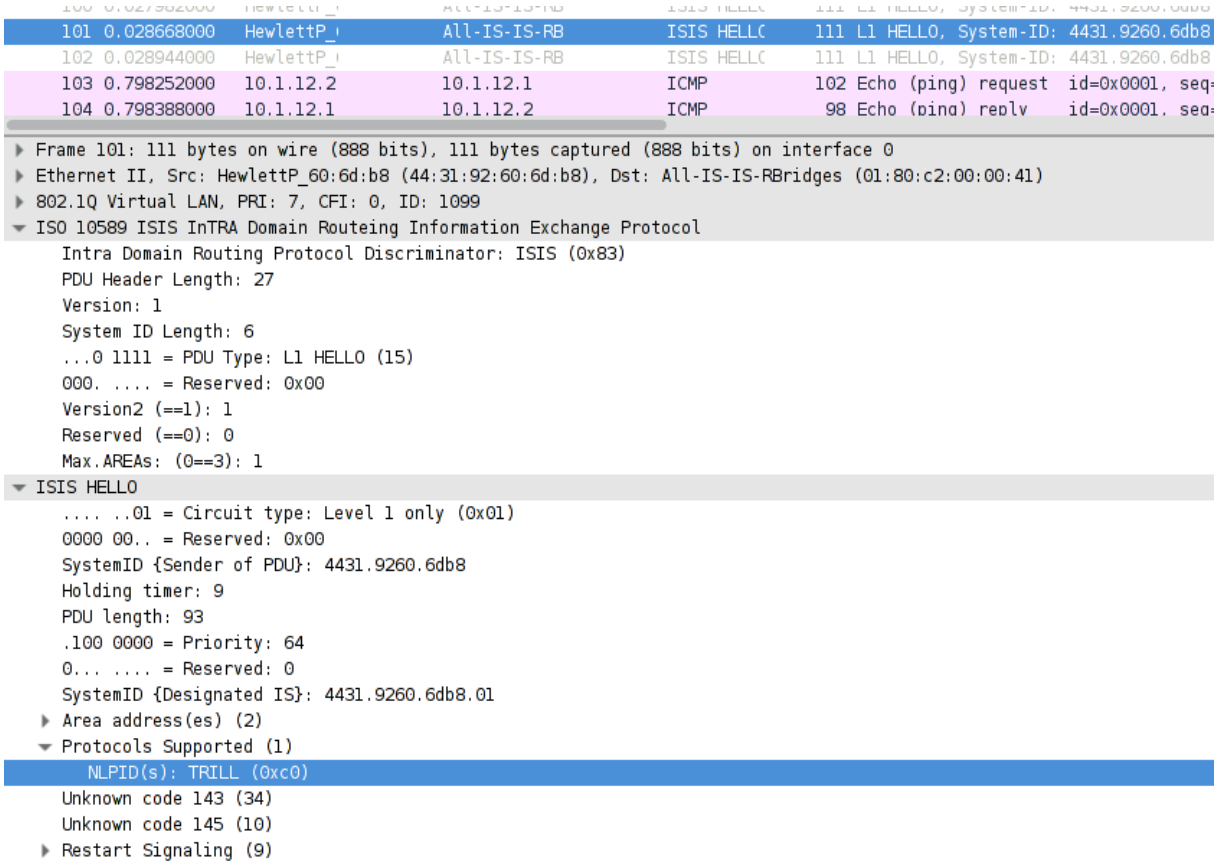
Switch	Neighbors
HP1	HP3, HP4
HP2	HP3, HP4
HP3	HP1, HP2
HP4	HP1, HP2

Diese Informationen wurden mit `display trill adjacent-table` und `display lldp neighbor-information list` ermittelt.

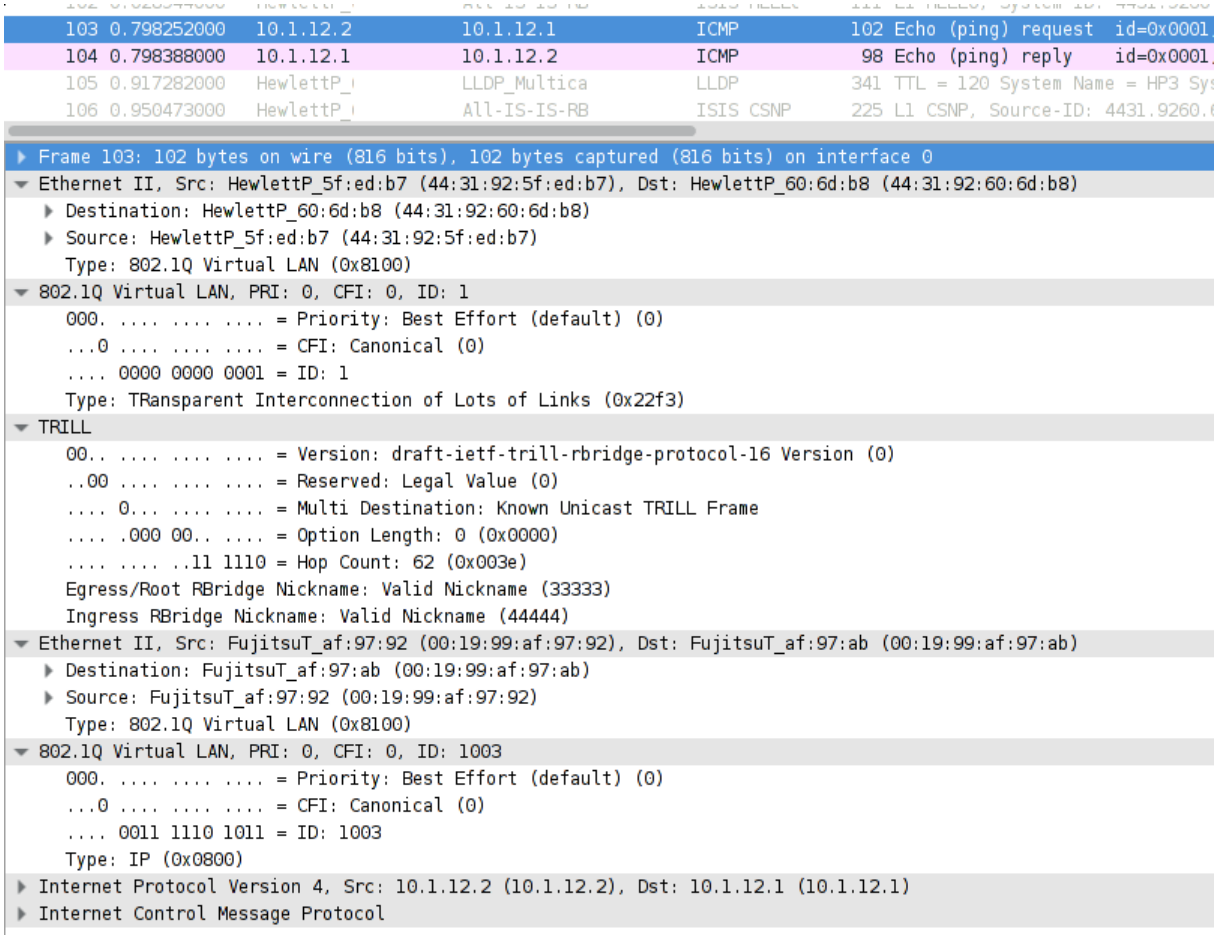
Dies entspricht auch der Zeichnung vom Laboraufbau.

4.8.6 Packet Flow

Wir sehen die regelmässigen TRILL IS-IS Hello Pakete in Wireshark. Als unterstütztes Protokoll wird TRILL angegeben:

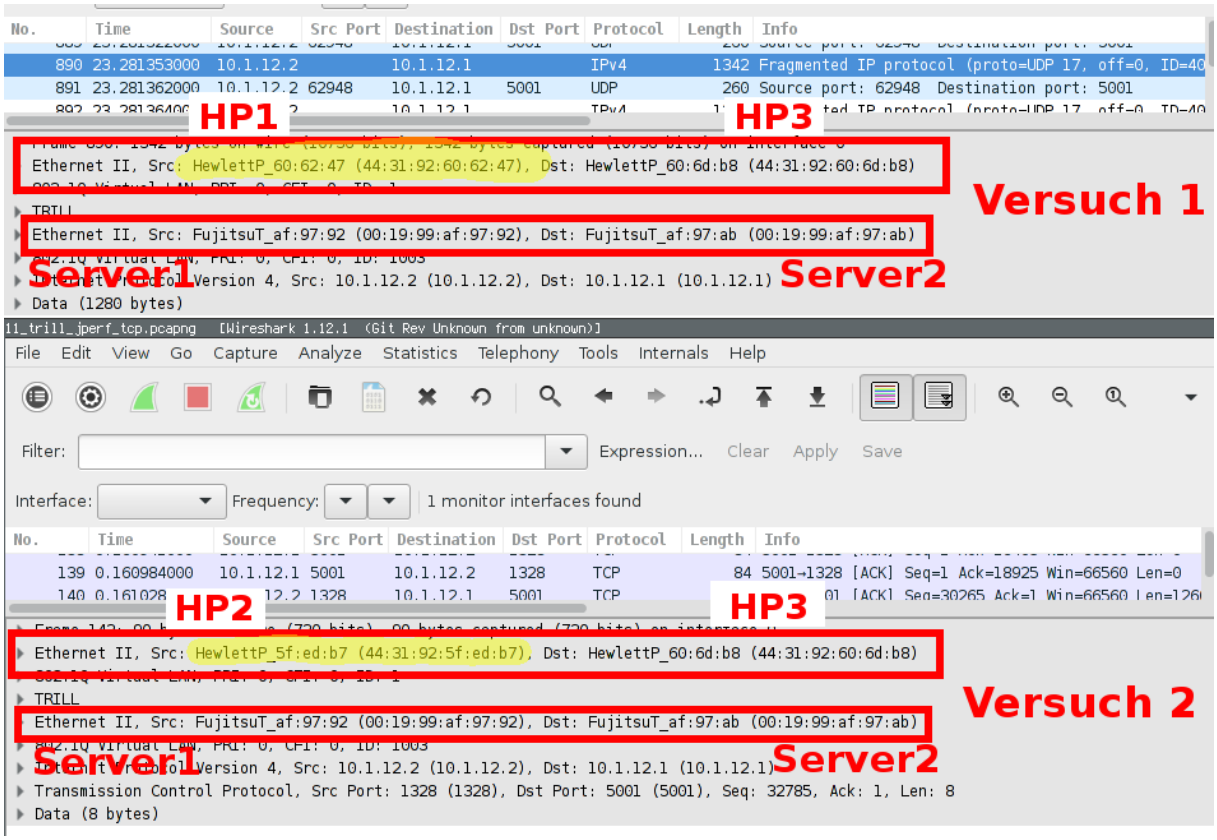


Schneidet man mit Wireshark ein ICMP Request mit, welcher von Server 2 zu Server 1 gesendet wird, sieht man, wie die Kapsulierung von TRILL Funktioniert:



Im Outer Ethernet Frame steht als Absender die MAC Adresse vom Switch HP2 und als Empfänger MAC Adresse die vom Switch HP3. Der Switch HP2 leitet das Paket an den Switch HP3 weiter. Dieser entfernt den TRILL Heaer und leitet das Paket an die MAC Adresse vom Inner Ethernet Header weiter. Dort kommt auch wieder das eingeschobene VLAN Tag zum Zug. Danach kommt der normale Payload, welche vom Server 2 erzeugt wurde.

Wir testeten auch, ob sich der Packetflow tatsächlich ändert, oder ob weiterhin Ports wie bei STP blockiert werden. Es funktionierte wunderbar: Einmal ging der Traffic über den Switch HP2 (wie im oberen Frame gezeigt) und einmal über den Swich HP1. Dies ist in folgendem Screenshot von Wireshark nochmals verdeutlicht:



Der Inner Ethernet Header bleibt exakt gleich. Der Outer Ethernet Header hat sich verändert: Da es wie bisher an den Switch HP3 geschickt wird, bleibt die Destination Adresse die selbe, aber die Source MAC Adresse vom Outer Ethernet Header ist unterschiedlich. Einmal geht es über HP2 und einmal über HP1. Der Unterschied ist im Screenshot gelb markiert.

Es werden somit wie erwartet beide Pfade ausgenutzt. Somit ist mehr Bandbreite verfügbar.

4.8.7 MAC Learning

Empfängt eine ingress Bridge zum ersten Mal ein Ethernet Frame von einem Client trägt es dessen MAC Adresse in eine Art ARP Tabelle ein. Die ingress Bridge weiss jedoch noch nicht wohin sie das Paket schicken soll, da die Ziel MAC Adresse (welche danach im Inner TRILL Frame steckt) noch keiner Egress RBridge (für den TRILL Header) zugeordnet werden kann. Deshalb wird das erste TRILL Paket multicast an alle RBridges gesendet. Jene egress RBridge, welche das Paket an die Ziel MAC Adresse ausliefern kann wird gegebenenfalls auch antworten (sofern der Client eine Antwort sendet). Diese egress Bridge hat als sie das erste Paket erhalten hat, die MAC Adresse vom sendenden Client in die ARP Tabelle aufgenommen. Daher weiss die egress Bridge, welche Bridge die ingress Bridge ist und über welchen Nickname diese erreichbar ist. Durch das IS-IS Protokoll kann es jetzt verschiedene Wege zum Ziel geben. Ein Weg wird gewählt.

Die transit RBridges lernen keine MAC Adressen vom inner Ethernet Header.

4.9 References

Die meisten detaillierten Infos stammen aus dem offiziellen RFC:

- RFC 6325[4]

Einen Grobüberblick bekommt man mit diesen zwei Videos:

[4] http://tools.ietf.org/html/rfc6325

- Youtube: MicroNugget: What is TRILL? https://www.youtube.com/watch?v=I7mvUrc90jQ
- Youtube: Extension and Virtualization using Layer 3 protocols: Part 2: TRILL: https://www.youtube.com/watch?v=YwYURoelGuI