sx CN2

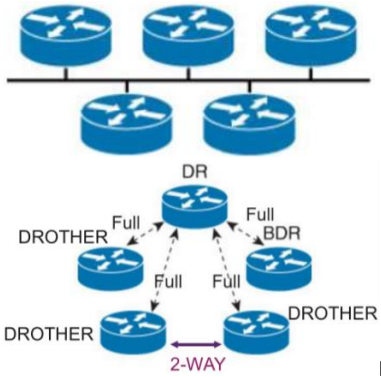
1 Open Shortest Path First

Link-State Routingprotokoll, Interior Gateway Protocol (IGP), Version 2 für IPv4 und Version 3 für IPv6, 4 Schritte:

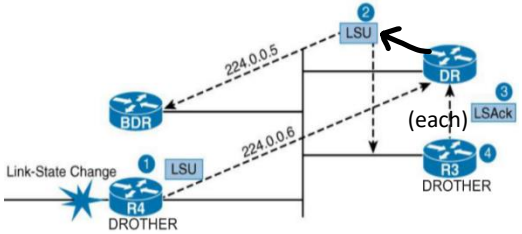
1. Neighbor adjacencies werden aufgebaut, Link-State Advertisements (LSA) ausgetauscht
2. Jeder Router baut seine Link-State Database (LSDB)
3. Jeder Router führt Dijkstra Algorithmus aus
4. Jeder Router erstellt seine Routing-Tabelle

1.1 Operation Modes

Point-to-Point



Multi-Access

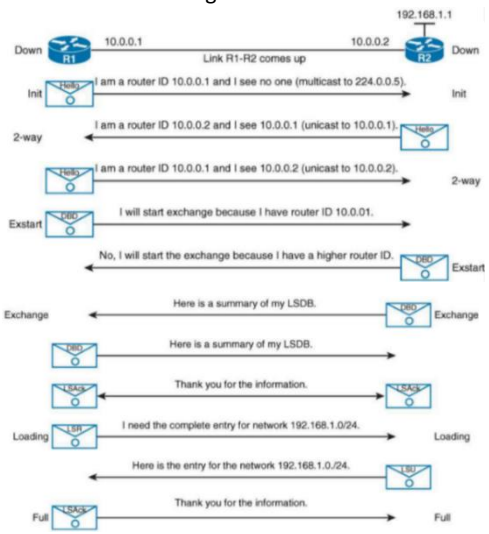


DR macht LSA forwarding und LSDB Sync für alle Router in Broadcast-Domain, DR/BDR haben Multicast 224.0.0.6, BDR ist Backup, jedes OSPF-enabled interface hat eine Prio von 0 – 255 (0 kann nicht DR werden), höchste Prio wird DR, zweithöchste BDR, wenn gleich höhere Router-ID, BDR übernimmt, wenn DR ausfällt, wenn DR dann wieder läuft, bleibt aber der BDR DR, auch bei neuem Router mit höherer Prio ändert sich der DR nicht

1.2 Neighbor Adjacency Forming

Pakete (OSPF L4 Protokoll, kein TCP/UDP):

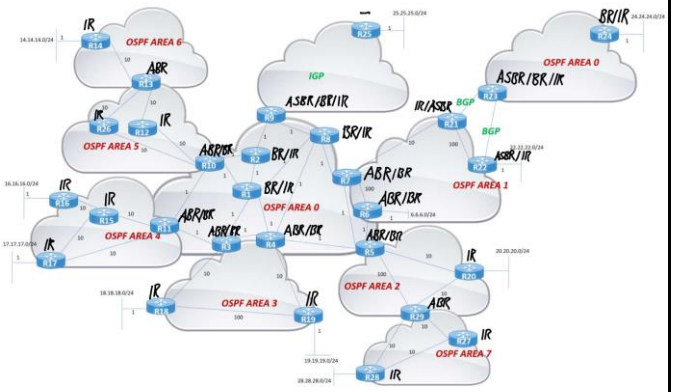
- **Hello**: Neighbor adjacencies finden, aufbauen, erhalten
- **Database Description (DBD)**: Zusammenfassung der gesamten LSDB eines Routers, aber nicht den ganzen Inhalt. Pro LSA-Eintrag ist Link-State type, advertising router address, link cost und seq no. angegeben.
- **Link-State Request (LSR)**: wird geschickt, um fehlenden LSA-Eintrag für LSDB zu bekommen
- **Link-State Update (LSU)**: Antwort auf ein LSR
- **Link-State Acknowledgement**: folgt auf LSU vom Router, welcher den LSR gesendet hat



- **DOWN**: Keine Hellos von beiden
- **INIT**: Hello auf Multicast erhalten ohne eigene Router ID enthalten, Router zeichnet Neighbor IDs auf und fügt sie in die Hellos ein
- **2-WAY**: Hello auf Unicast mit eigener ID erhalten, DR/BDR Election findet statt
- **EXSTART**: DBDs werden ausgetauscht, Master/Slave Election, Initiale Sequence Number wird definiert, stuck hier bei MTU-Mismatch
- **EXCHANGE**: DBDs werden ausgetauscht, fehlende LSA-Einträge werden der LSR-Liste hinzugefügt
- **LOADING**: LSR-Liste wird dem Neighbor geschickt, LSUs werden gesendet oder bis zum LSAck erneut gesendet
- **FULL**: Beide Neighbors haben eine identische LSDB

Um die Neighbor adjacency aufrecht zu erhalten, wird alle «Hello-Interval» (Cisco-Default 10s) ein Hello gesendet. Nach dem «Dead-timer» (4x Hello-Interval) wird der Neighbor als DOWN markiert. Die beiden Timer sind in den Hellos enthalten und müssen identisch sein auf allen Routern. Auch die MTU muss matchen, sie ist im ersten DBD enthalten. Die OSPF Router ID ist eine IPv4-Addresse nach absteigender Priorität: manuell konfiguriert, höchste Loopback-Interface IP, höchste Non-Loopback-Interface IP. Es können passive Interfaces konfiguriert werden, um Adjacencies zu deaktivieren. Ein originating Router sendet seinen LSA alle 30m mit höherer Seq No, bei Änderungen sofort. Jeder LSA hat eine Link-State Age Variable, welche nicht höher als 30m sein sollte, nach 60m wird der Eintrag aus der LSDB entfernt.

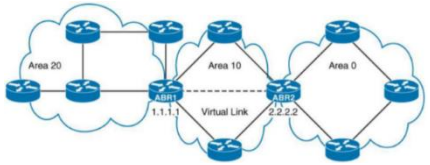
1.3 Areas



Bei jeder Netzwerkänderung muss der Dijkstra Algorithmus von den Routern ausgeführt werden. Daher sollten Router in Areas von höchstens 50 Routern unterteilt werden. Dijkstra wird dann nur bei Änderungen in der eigenen Area ausgeführt.

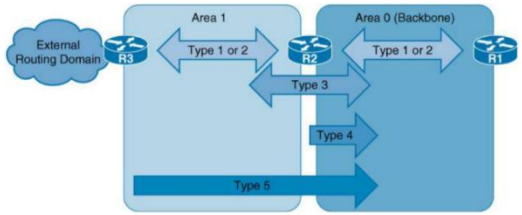
- Area 0: Backbone (keine Enduser)
- Area >0: Non-Backbone
- ABR: Area Border Router (Routing Tabelle pro verbu. Area)
- ASBR: Autonomous System Border Router
- IR: Internal Router (alle Interfaces in der gleichen Area)
- BR: Backbone Router (mind. 1 Interface in Area 0)
- Route Summarization nur bei ABR/ASBR

Regeln für Area 0: muss contiguous (zusammenhängend) sein, Non-Backbone Areas müssen mit Backbone Area verbunden sein, Traffic zwischen Non-Backbone Areas muss durch Backbone Area gehen, dafür braucht es möglicherw. Virtual Links:



LSA-Typen:

- **Type 1**: Router Link advertisement, Interfaces des Routers, typisch für Subnetze, innerhalb der Area (O)
- **Type 2**: Vom DR gesendet, enthält alle Router des Multi-Access Netzes, innerhalb der Area
- **Type 3**: Für alle Type 1&2 generiert der ABR einen Type 3, welcher dann von anderen ABRs in alle anderen Areas weitergeleitet wird, können auch summarized sein (O IA)
- **Type 4**: ASBR generieren einen Type 1 mit External Bit gesetzt, um die Route zu ihnen zu advertisen, ABRs wandeln den dann um zu einem Type 4, um den ASBR über alle Areas zu advertisen
- **Type 5**: ASBR advertised externe Netzwerk über alle Areas, können auch summarized sein (O E1/O E2 (Default=2))



Area Typen:

- **Standard**: alle Types erlaubt
- **Stub**: Type 5 blockiert, keine externen Routen und ASBRs, Gateway für externes ist nächster ABR
- **Totally Stubby**: Type 3, 4, 5 blockiert, keine Intra-Area/externe Routen und ASBRs, Gateway für alles ausserhalb der Area ist der nächste ABR
- **Not-so-Stubby (NSSA)**: Stub mit ASBR, nur externe Routen der eigenen Area, keine Default Route, ASBR generiert Type 7 innerhalb Area (O N1/O N2 (Default=2)), ABR leitet Type 7 als Type 5 weiter
- **Totally Not-so-Stubby (Totally NSSA)**: Totally Stubby mit ASBR, nur externe Routen der eigenen Area, Default Route über ABR, ASBR generiert Type 7 innerhalb Area (O N1/O N2), ABR leitet Type 7 als Type 5 weiter

1.4 Best Path Selection

Jedes Interface hat eine Cost von 1 – 65535
Cisco Default ist alles mit und über 100 Mbit/s hat Cost 1, die Reference-Bandwidth kann erhöht werden, um die Cost am Interface manuell anzupassen, kann die Interface Bandwidth oder die Cost direkt geändert werden

OSPF wählt immer den Weg mit niedrigster Cost

Externe Routen haben eine Metric (Default=20), welche der Cost addiert wird

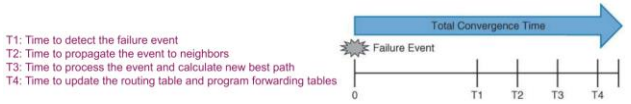
Mehrere Wege mit gleicher Cost: Load Balancing, genannt Equal-Cost MultiPath (ECMP)

Priorität absteigend: O, O IA, E1, N1, E2, N2

1.5 Schnellere Network Convergence

Um die Verfügbarkeit eines Netzwerks zu erhöhen, gibt es verschiedene Techniken, um Packet Loss zu minimieren und die Convergence zu verschnellern.

OSPF-Default ist >5s für Total Convergence:



Wenn der Link Failure nicht direkt verbunden ist, muss der Router sich auf OSPF verlassen. OSPF wartet 40s, bis der Neighbor als down markiert wird.

1.5.1 Fast Hellos

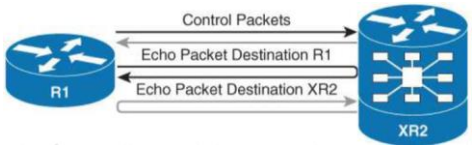
Der Hello und somit Dead Timer kann auf ein kürzeres Intervall gesetzt werden, um T1 zu verkürzen, jedoch reicht das möglicherweise nicht, braucht viel CPU und skaliert schlecht.

1.5.2 Bidirectional Forwarding Detection (BFD)

BFD behandelt OSPF als Client, der BFD-Sessions erstellt. Alle 50ms wird ein BFD-Paket an den Neighbor gesendet, nach 3 Fails (150ms) gibt es einen Session Failure, die Timer können auch angepasst werden. Es ist weniger CPU-intensiv als Fast Hellos. Zuerst werden mit Control Packets, welche von UDP-Port 49152 zu 3784 gehen, Session Parameter verhandelt. Danach gibt es zwei Modes:

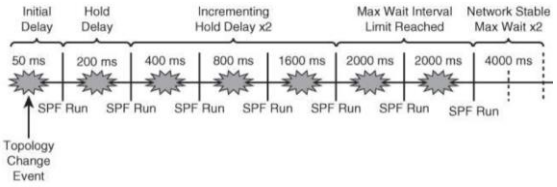
- **Asynchronous Mode ohne Echo**: Control Packets werden unidirectional ständig gesendet, keine Requests/Antworten. Session ist down, wenn eine gew. Anzahl nicht angekommen sind.

- **Asynchronous Mode mit Echo**: Nach den Control Packets werden Echo Pakete an Port 3785 gesendet, der Empfänger loopt das Echo zurück.



1.5.3 Fast Rerouting (FRR)

Link-State Routing Protokolle sind auf Stabilität anstatt Convergence Speed ausgelegt. Der Backoff Timer erlaubt schnelle Reaktion auf ein Event, aber verzögerte Reaktion auf Serien von Events. Throttling macht die Convergence langsamer, ist aber nötig, um unkontrollierte Fluktuationen zu verhindern.



Start timer: Initialer Delay nach Topology Change
Hold: Delay zwischen zwei aufeinanderfolgenden SPF-Kalkulationen, verdoppelt sich nach jeder SPF-Kalkulation oder wenn Route Flap erkannt wird
Max wait: Maximaler Delay zwischen aufeinanderfolgenden SPF-Kalkulationen

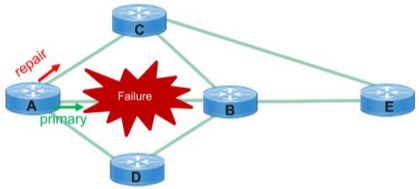
Konkrete Implementierung bei Cisco OSPF:

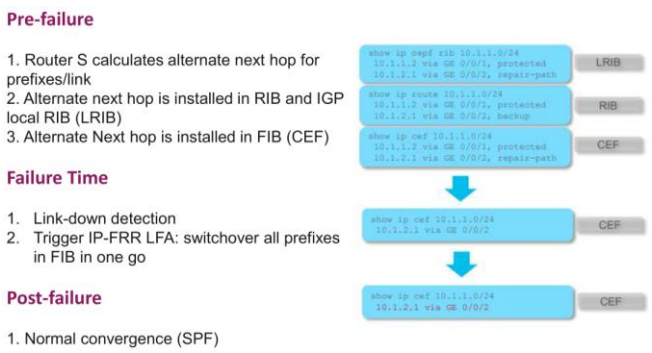
- **Max LSA Lifetime**: 3600s
- **LSA Refresh Interval**: 1800s
- **LSA Packet Pacing** (Delay zwischen LSUs): 33ms
- **LSA Generation Throttling** (Delay, bevor ein neuer LSA an Neighbors flooded wird): Start 0ms, Hold 5000ms, Max 5000ms
- **LSA-Arrival** (Delay zwischen Erhalt von neuem LSA und Verarbeitung): 1000ms

1.5.3.1 Loop Free Alternate (LFA)

Dabei wird nach dem Link Failure anstatt, dass das Control Plane die langen Convergence Calculations macht, direkt vom Data Plane auf einen Repair path gewechselt und danach die Convergence Calculations durchgeführt.

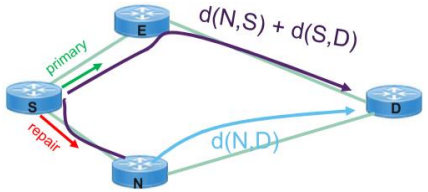
Der Router mit dem Failure macht die ganze Berechnung innerhalb von 50ms. Die Berechnung des Alternate Paths erfolgt im Hintergrund, das primary SPF hat Vorrang und es braucht mehr Memory.





Begriffe:

- **S**: Source Router (calculating Router):
- **E**: Neighbor Router (primary next Hop)
- **N**: Neighbor Router (alternativer next Hop)
- **D**: Destination Router (wo die Prefixes connected sind)
- **d(A, B)**: Tiefste Cost von A nach B



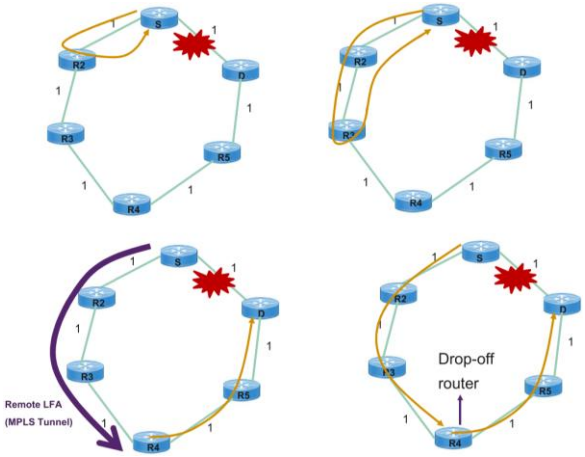
3 Bedingungen müssen erfüllt sein, damit ein LFA existiert:

1. LFA: $d(N,D) < d(N,S) + d(S,D)$

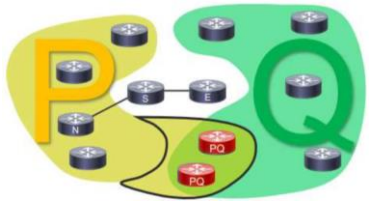
2. Downstream Path: $d(N,D) < d(S,D)$

3. Node Protection: $d(N,D) < d(N,E) + d(E,D)$

1.5.3.2 MPLS zur Erweiterung der LFA Coverage



- **P-Space**: Router, die von S erreicht werden können, ohne den SE-Link zu durchqueren
- **Q-Space**: Router, die von E erreicht werden können, ohne den SE-Link zu durchqueren
- **Extended P-Space (PQ)**: P-Space aller Neighbors von S



Als Drop-Off Router wird der nächste PQ-Router genommen. Der LFA-Tunnel muss nicht der shortest Path zum Drop-Off Router sein. Network Design: Dreiecke haben LFA, Vierecke/Fünfecke/Ringe unterstützen nur Remote LFA.

2 IS-IS

Link-State Routingprotokoll, Interior Gateway Protocol (IGP), Layer 3 Connectionless Network Protocol (CLNP) fürs ISO/OSI Modell

Unterschiede gegenüber OSPF:

- braucht kein IP
- älter als OSPF, macht gerade Comeback in SDN,
- einfacher & sicherer
- kann einfach erweitert werden mit type, length & value (TLV) Mechanismus
- IPv4/IPv6 Support
- 1000 Router pro Area gegenüber 100
- Updates sind ins LSPs zusammengefasst anstatt viele LSUs
- Weniger CPU intensiv
- Erkennt Failures schneller mit Default Timers

Host Geräte werden End Systems (ES) und Router Intermediate Systems (IS) genannt

Gleiche 4 Schritte wie OSPF

Jeder Router hat eine unique Network Service Access Point (NSAP) Adressse:

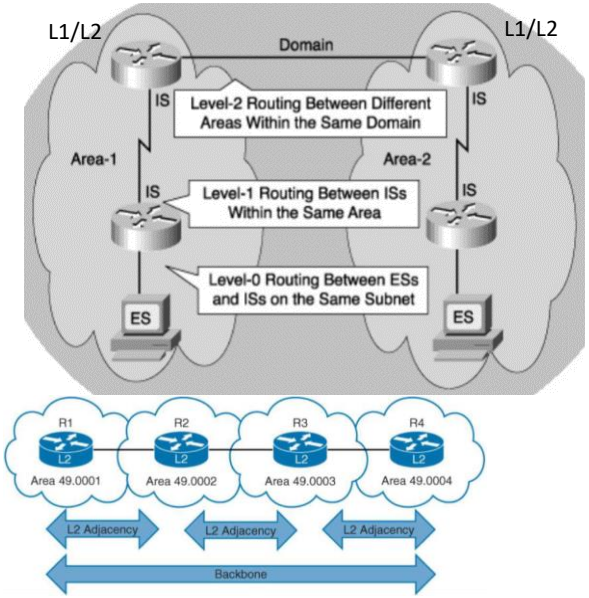
Fix	Fix	Variabel	Variabel	Fix
49	00	01	0000.0c12.3456	00
Area Address			System ID	NSEL
3 Bytes			6 Bytes	1 Byte

2.1 Areas/Levels

- L1: Kennt nur eigene Area
- L2: Kennt andere Areas (Backbone)
- L1/L2: Zwei Routingtables für die eigene und andere Areas
- Separate LSDB/LSP pro Level

L1 Router verwenden den nächsten L1/L2 Router für Destinations ausserhalb der Area. Da L1/L2 Router keine L2 Routen in die Areas advertisen, ist eine L1 Area wie eine Totally Stubby Area in OSPF. L1 Routen werden jedoch in L2 advertised.

Router und seine Interfaces können nur in einer Area sein die Area-Grenzen liegen auf den Links. Das Backbone kann mehrere Areas umfassen.



Area Design Rule: Backbone muss eine contiguous (zusammenhängende) Chain von L2 oder L1/L2 Routern sein.

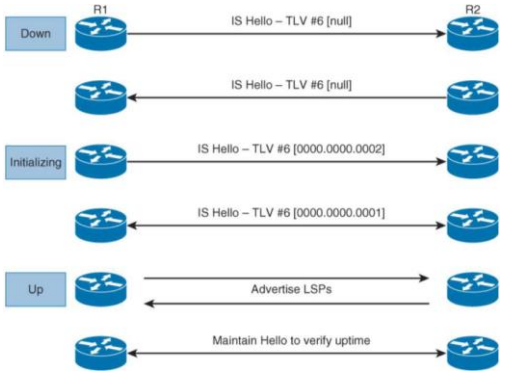
2.2 Neighbor Adjancency

Bei L2 wird immer eine Adjacency mit allen Routern im LAN/P2P hergestellt. Bei L1 wird eine Adjacency mit allen Routern im LAN/P2P hergestellt, sofern sie in der gleichen Area sind.

Pakete (L3):

- **IS-IS Hello (IIH)**: wird fürs Herstellen von neighbor adjancencies verwendet, bei Multi-Access: L1 an Multicast 01:80:C2:00:00:14, L2 an Multicast 01:80:C2:00:00:15, bei P2P: L1, L2, L1/2 Unicast an MAC, zum Prüfen ob die MTU matcht wird bis zum Maximum mit TLV 8 gepadded, dies kann optional nach den ersten 5 Hellos deaktiviert werden, um Bandbreite zu sparen
- **Link-State Packets (LSP)**: Informationen über den Router und angeschlossene Netzwerke mit Sequence No, Multicast bei LAN, Unicast bei P2P
- **Sequence Number Packets (SNPs)**: 2 Typen:
  Complete SNP: Zusammenfassung der LSPs in der LSDB
  Partial SNP: Request und Acknowledgement von fehlenden LSPs in der LSDB

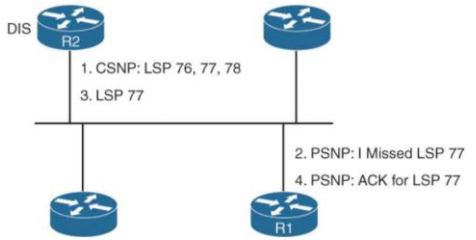
Fürs Herstellen einer Neighborship müssen die Levels übereinstimmen (L1/L2 geht mit beidem), ein gemeinsames Subnetz geteilt werden, die MTU übereinstimmen und der Authentication Type/Credentials übereinstimmen. Um die Neighborship zu erhalten, wird alle Hello-Timer (Default=10s bei DIS 3s) ein Hello gesendet, der Holding timer ist 3x der Hello-Timer. Wenn dann kein Hello empfangen wurde, wird die Adjancency aufgelöst. Bei passiven Interfaces hängen Netze ohne Router.



1. R1 sendet Hello ohne Neighbor TLV
2. R2 antwortet mit Hello ohne Neighbor TLV
3. R1 verifiziert die Parameter, wenn alles matcht wird ein neuer Eintrag in der Neighbor Table erstellt und der Status auf Initializing gesetzt
4. Es folgen 3 Hellos (R1-R2-R1) mit der MAC
5. Beide Router setzen Status auf Up
6. LSPs und evtl. SNPs werden ausgetauscht und wenn beide die gleiche Tabelle haben, wird der Dijkstra Algorithmus ausgeführt

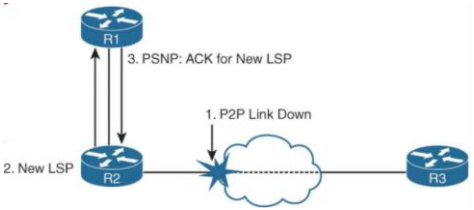
Bei Multi-access wird ein Distinguished Intermediate System (DIS) gewählt. Die höchste Priorität oder danach die höchste SNPA (MAC) gewinnt, dabei kann L1 und L2 einen unterschiedlichen DIS haben. Sobald ein neuer Router ins Netz kommt, kann er DIS werden, wenn er gewinnt. Kein Backup DIS wegen den kurzen Hello Timers.

Der DIS floodet LSPs ins LAN und erstellt/updated das Pseudonode, welches eine Verbindung zu allen Routern herstellt. Der DIS multicastet alle 10s CSNPs ins LAN:



P2P Adjacencies werden erkannt und es werden keine Ressource mit DIS Election und LSP Flooding verschwendet.

Ablauf Link Failure:



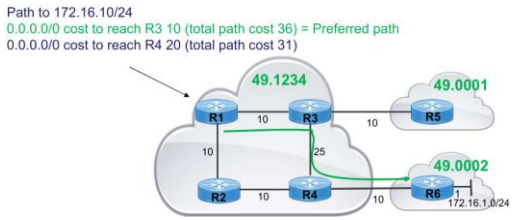
2.3 Best Path Selection

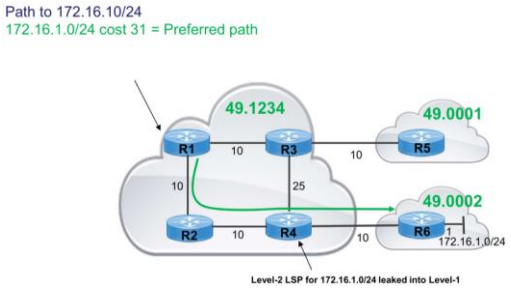
Den Interfaces können narrow Metrics zwischen 1 und 63 zugewiesen werden. Default ist 10 für alle Interfaces. Ausserdem gibt es die wide metric, welche in grossen Netzwerken verwendet werden sollte mit Werten von 1 - 16'777'215.

Der Best Path ist der mit der niedrigsten Metric mit absteigender Priorität:

1. L1 Intra-area (internal metric)
2. L2 Intra-area (internal metric)
3. Leaked Routes L2 → L1 (internal metric)
4. L1 external route (external metric)
5. L2 external route (external metric)
6. Leaked route L2 → L1 (external metric)

Route leaking (suboptimales Routing wegen Default Route nächster L1/L2 Router):



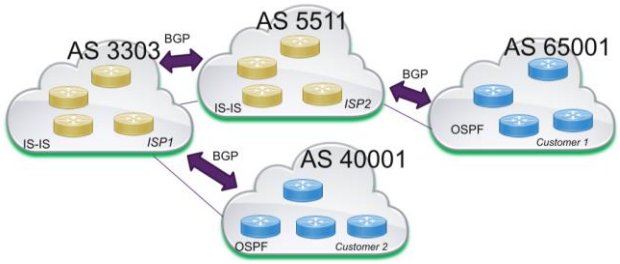


Route summarization passiert immer da, wo eine Route ein Level betritt.

3 Border Gateway Protocol

Exterior Gateway Protocol (EGP), wird benutzt, um Routing Infos zwischen ISPs und grossen Firmen auszutauschen

Autonomous System (AS): Netzwerk unter derselben administrativen Domain
Public AS: 1 – 64511 (werden von IANA verwaltet, welche RIRs wie RIPE in Europa assigned)
Private AS: 64512 – 65535



Zwei Router aus verschiedenen AS können ein Peering machen. Die beiden Router werden dann BGP-Speaker genannt. Die Kommunikation erfolgt über TCP-Port 179.

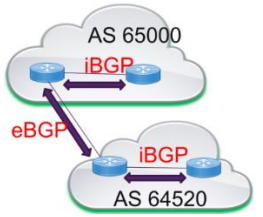
Es gibt 4 Message-Typen zwischen Peers:

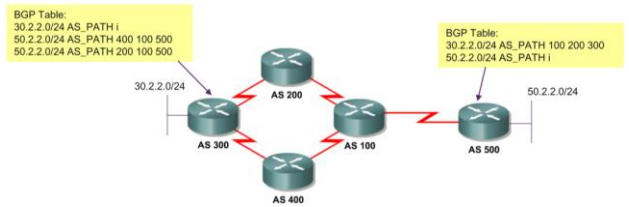
- **Open**: Peering erstellen, enthält AS, Router ID und Hold time
- **Update**: Routing Infos wie withdrawed routes, neue Routes mit Netzwerk/Maske oder path attributes übertragen
- **Keepalive**: alle 60s für Session-Keepalive
- **Notification**: Session beenden

BGP attributes:

- **Well-known mandatory** (must be supported and included in every update message): AS Path, Origin, Next Hop
- Well-known discretionary (must be supported but no included in every update): Atomic Aggregate, Local Preference
- Optional transitive (should be forwarded to other AS even if not understood): Community, Aggregator
- Optional non-transitive (should be removed if not understood and not forwarded): MED, Weight, Cluster ID, Originator ID, Cluster List

Beim Path Announcement hängt jeder Router seine AS vorne an beim AS_PATH, damit die Daten dann in die umgekehrte Richtung des Advertisements fliessen können. Wenn die eigene AS schon im AS_PATH vorhanden ist, wird das Paket verworfen.



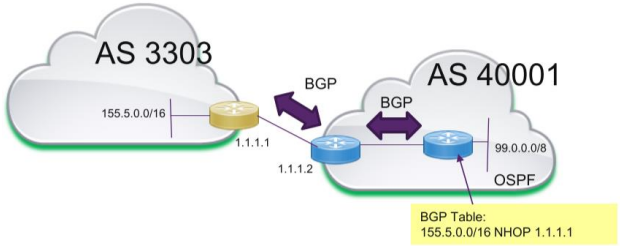


11 Schritte von BGP zum Finden des Best Path:

1. Höchste Weight (local zum Router)
2. Höchste Local Preference (global innerhalb AS)
3. Route ist locally originated
4. Kürzester AS Path
5. Tiefster Origin Code (IGP < EGP < Incomplete)
6. Tiefste MED (metric)
7. Externe (eBGP) paths vor internen (iBGP)
8. Für iBGP: path durch nächsten IGP neighbor
9. Für eBGP: ältester path
10. Path von Router mit tiefster BGP-Router-ID
11. Path von Router mit tiefster neighbor address

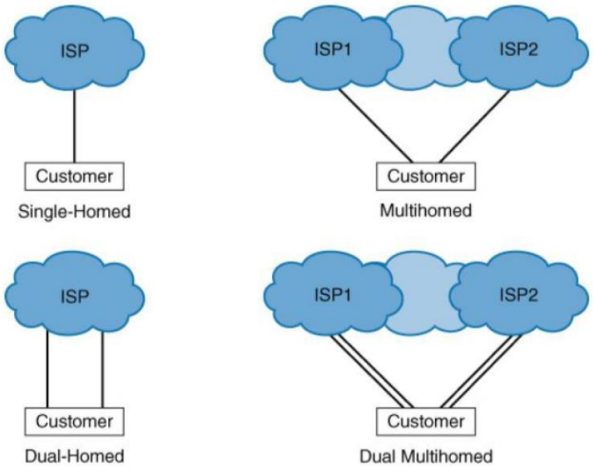
Jeder eBGP-Router ändert das AS_PATH- und NHOP-Attribut. Bei iBGP wird beides nicht modifiziert.

Lösung für diese Situation:

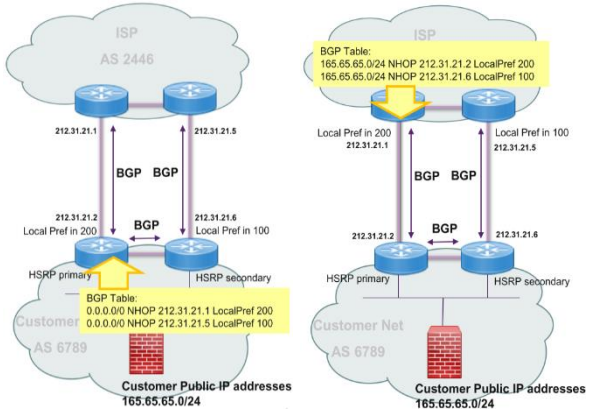


- Next-hop-self aktivieren, somit würde 1.1.1.2 sich selbst als Next-Hop eintragen, bevor die Route ins iBGP advertised wird
- Der Link zwischen AS 3303 und 40001 kann ins iBGP redistributed werden

3.1 Enterprise Internet Connectivity



- **Single-homed**: Customer-Router hat Default Route zum ISP, ISP hat statische Route zum Customer-Router mit IP-Prefix, geht mit oder ohne BGP
- **Dual-homed**: meist im Primary/Backup Design, ISP advertised Default Route, Customer advertised IP-Prefix, über die Local Preference kann bestimmt werden, welcher Router Primary ist für ausgehenden Traffic, über die MED kann bestimmt werden, welcher Router Primary ist für eingehenden Traffic, man kann auch bei einem Router AS_PATH prepending aktivieren, somit wird die eigene AS mehrmals angehängt und der Pfad ist länger, dann wird der andere Router genommen, eingehend hat man aber nicht wirklich die Kontrolle, wenn der ISP LOCAL_PREFS setzt, übersteuert das alle Massnahmen
- **Multi-homed**: active/active Design, beide Customer Router erhalten die ganze Internet Routing Tabelle, dadurch können die effizientesten Pfade genutzt werden, wichtig ist, dass die eigenen Router nur eigene Prefixes und die von



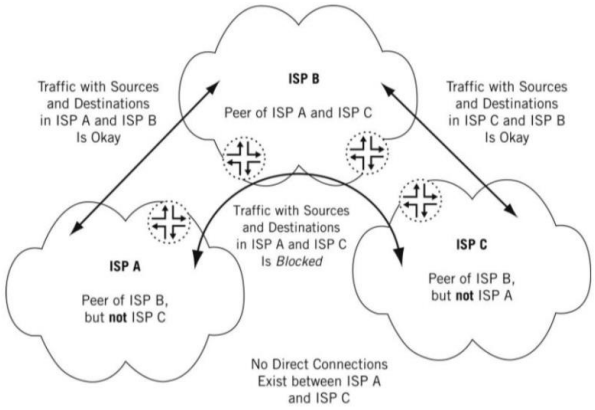
nur einem ISP advertisen, sonst werden die eigenen Router für den Internet-Transit verwendet

3.2 Internet

Das Internet ist eine Verbindung von 80'000 AS, es gibt keine zentrale Kontrolle, es gibt Verbindungen zwischen allen möglichen Firmen/ISPs. Da die AS-Nummern ausgegangen sind, wurde nun eine 4 Bytes ASN eingeführt

Transit: Business Beziehung von meist kleinerem und grösserem ISP mit Bezahlung, damit der kleinere über den grösseren das ganze Internet erreichen kann
Peering: Partnerschaft ohne Bezahlung für mehr Performance für beide, beide können dann bestimmte Prefixes des anderen erreichen

Teilweise wird auch Traffic blockiert, beispielsweise möchte ISP B hier nicht Transit machen:



Public Internet Exchange Point (IXP): Members peeren mit einem Route server (RS), alle member routes werden allen Peers advertised, der RS fügt seine AS dem AS_PATH hinzu, jedoch nicht dem NEXT_HOP, damit er aus dem Path bleibt, gleicher rechtlicher Vertrag für alle, z.B. SwissIX

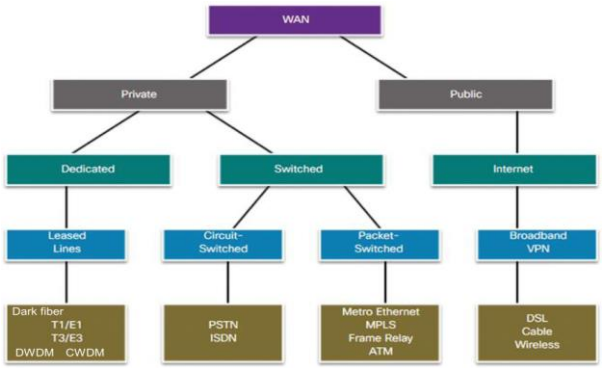
Private IXP: Individuelle Peerings zwischen AS

Mithilfe von Looking Glasses von ISPs können die Paths zu einer bestimmten IP von einem bestimmten Router des ISPs ausgelesen werden

4 Wide Area Network

WANs möchten Remote LANs verbinden

Beachtet werden muss: Bandbreite, Verfügbarkeit (SLA), Redundanz, QoS, Vertrauen in Provider, Design, Layer 2/3, Management



4.1 Private

4.1.1 Leased Line

Kunde bezahlt Service Provider für eine private eigene Layer 2 Leitung mit garantierter Bandbreite. Mit Glasfaser wird das Dark Fiber genannt, welches extrem teuer ist und schwierig zu bekommen ist.

Coarse wavelength division multiplexing (CWDM): 16 CWDM-Lambdas über eine physische Fiber, 1270nm – 1620nm mit 20nm Intervall, bis 120km, nur optisch ohne Elektronik, günstiger als DWDM

Dense wavelength division multiplexing (DWDM): 80x 10 Gbit/s Channels über eine physische Fiber, 1528nm – 1563nm, 0.8nm Intervall, bis 1000 km, teurer als CWDM, mit Elektronik, Unterseekabel

4.1.2 Circuit Switching

Bevor kommuniziert werden kann, muss mit einem Signaling Protocol ein dedizierter Circuit hergestellt werden, ineffizient wegen bursty traffic im Networking, z.B. bei Telefon eingesetzt

Integrated Services Digital Network (ISDN): legacy Internet Technologie, wure durch DSL ersetzt, analoge Signale durch public switched telephone network (PSTN) beim ISP, wird zu time-division multiplexed (TDM) digital signals umgewandelt

4.1.3 Packet Switching

Traffic wird in Packets gesplittet, welche über ein shared network geroutet werden, es wird kein Circuit benötigt, die Switching-Entscheidungen werden aufgrund der Informationen in den Packets gefällt

4.1.3.1 Connection-oriented (legacy)

- **Asynchronous Transfer Mode (ATM)**: Virtual Circuits unterstützen bis 622 Mbit/s
- **Frame-Relay**: Permanent Virtual Circuits (PVC) unterstützen bis 4 Mbit/s

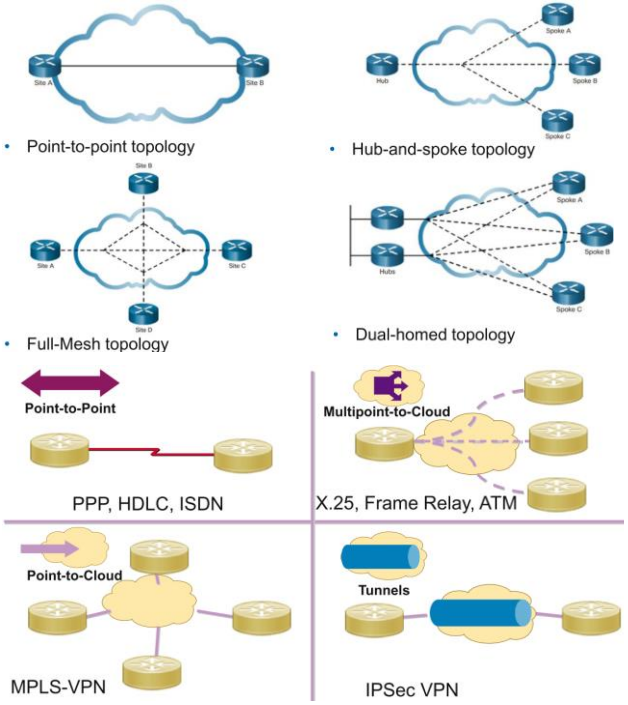
4.1.3.2 Connectionless

- **Metro Ethernet**: Ethernet war für LANs gedacht, kann aber auch über längere Distanzen benutzt werden. IEEE 1000BASE-SX (Fiber 550m), 1000BASE-LX (Fiber, 5km), 1000BASE-ZX (70km), grössere Distanzen mit Ethernet over MPLS (EoMPLS) & Virtual Private LAN-Service (VPLS)
- **MPLS**

4.2 Public

- **IPSec VPN**, 2 Arten: Site-to-Site (S2S, fixe -Verbindungen von Branch Offices), Remote-access (dynamische Verbindungen von einzelnen Remote-Clients)
- **Dynamic Multipoint VPN (DMVPN)**: basiert auf Generic Routing Encapsulation (GRE), jedes Uni- oder Multicast Packet kann encapsulated werden, NHRP (Next-hop resolution protocol), IPSec verschlüsselt, Cisco-proprietär, statische Hub-to-Spoke Tunnels, dynamische Spoke-to-Spoke Tunnels (Hub=Hauptsitz, Spoke=Branch Office)

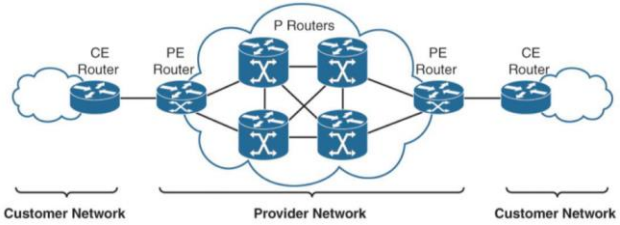
4.3 Topologien



5 Multi Protocol Label Switching

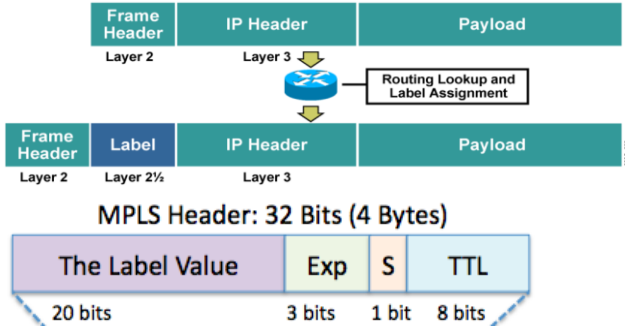
MPLS war eine Service Provider Technologie, wird jedoch mittlerweile in vielen Enterprise Netzwerken verwendet. Das Routing basiert auf Labels anstatt IP-Adressen und es können alle möglichen L2 oder L3 Protokolle transportiert werden.

Router Roles:



- **P**: Provider, Label switching router (LSR), Interior Gateway Protocol (IGP) und Label Distribution Protocol (LDP) läuft
- **PE**: Provider Edge, setzt und entfernt Labels (Edge LSR), IGP, LDP und Multiprotocol BGP (MP-BGP) läuft
- **CE**: Customer Edge, verbindet Kunde mit ISP

MPLS hat einen eigenen Ethertype (0x8847), das Paket wird vom PE-Router damit verpackt.

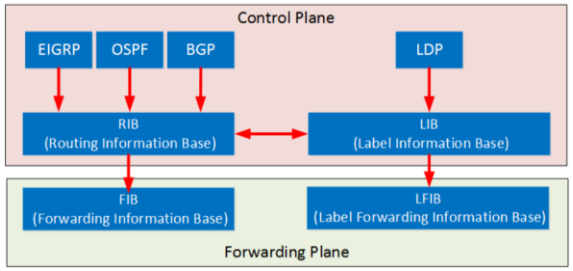


- MPLS-Label
- Experimental Field (Class of Service)
- Bottom-of-Stack indicator
- Time-to-Live (wir vom IP-Paket übernommen, dies kann jedoch deaktiviert werden, damit traceroute die ISP-Topologie nicht sieht)

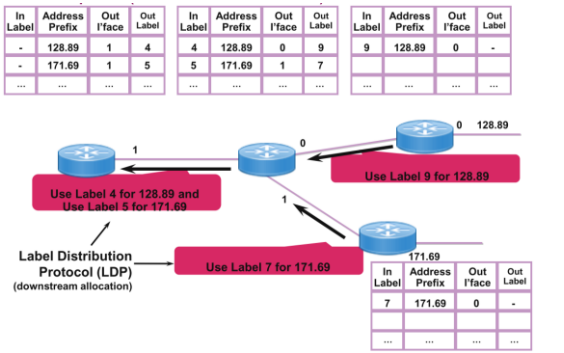
Es können mehrere MPLS-Header vorhanden sein, dann ist das S-Bit nur beim innersten gesetzt.

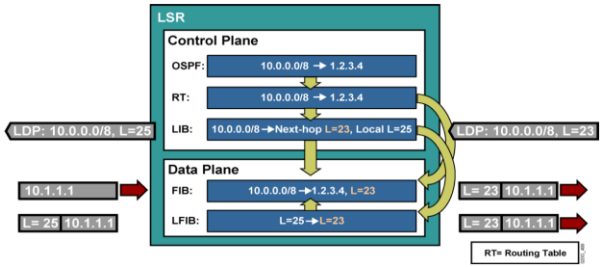
LDP sendet regelmässig Hello messages auf allen MPLS-enabled Interaces an 224.0.0.2 UDP/646. Andere Router antworten darauf mit dem Erstellen einer Session mit TCP/646.

Es gibt eine Control plane, welche den Destinations die Labels zuweist und die Labels verteilt mit LDP, und ein Data plane, welches das Switching der labeled Packets macht.



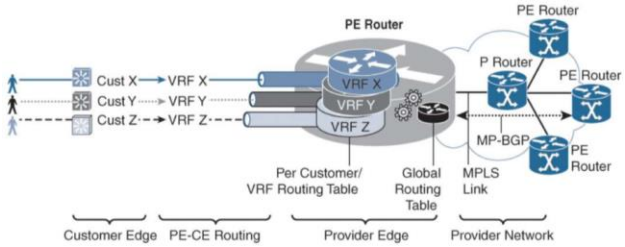
1. Routingtabellen werden mit einem IGP wie OSPF erstellt.
2. LDP assoziiert den einzelnen Prefixes individuelle In- und Out-Labels auf den Routern.





Mit Penultimate Hop Popping (PHP) wird das Label bereits vom P-Router vor dem PE-Router entfernt, somit muss der PE-Router das nicht mehr machen, das verbessert die Performance.

Weil mehrere Kunden die gleichen IP-Ranges haben können, werden Virtual Routing/Forwarding (VRFs) tables auf den PE-Routern verwendet.



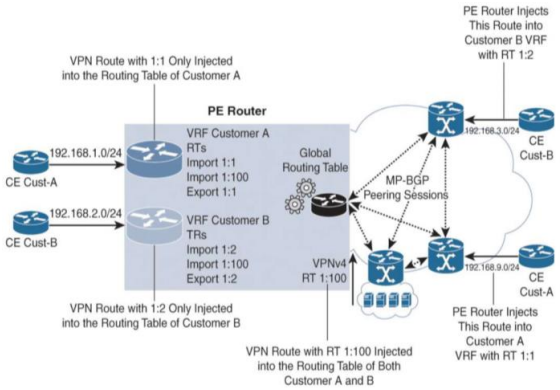
Der Route Distinguisher (RD) wird verwendet, um trotz gleichen IP-Ranges eine Route unique zu identifizieren. Zusammen mit der IPv4 Adresse ergibt das dann die VPNv4 Adresse, welche global im ganzen ISP unique ist.

RD: 1:1
IP address: 10.1.1.0
VPNv4 prefix: 1:1:10.1.1.0

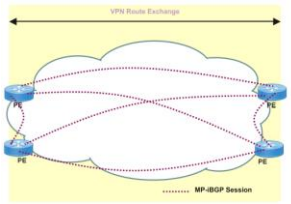
Route Distinguisher (8 Bytes)	IPv4 Address (4 Bytes)

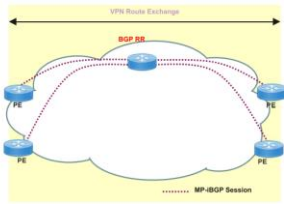
Die PE-Router müssen mit allen anderen PE-Routern mit MP-iBGP verbunden sein, um die VRFs zu synchronisieren. Ein MP-iBGP Paket hat neben der VPNv4 Adresse zwei weitere Felder:

- **Route Target (8 Bytes)**: Extended community welches identifiziert, welche Routen ins VRF importiert und exportiert werden dürfen
- **VPN-Label (3 Bytes)**: Label im inneren MPLS-Header, welches das VPN global identifiziert



Bei BGP müssen alle Router Full-Mesh verbunden sein, damit steigt die Anzahl Verbindungen exponentiell, daher ist ein Route Reflector (RR) gut.





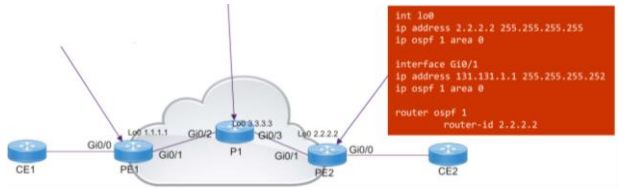
5.1 MPLS L3 VPN

Komponenten:

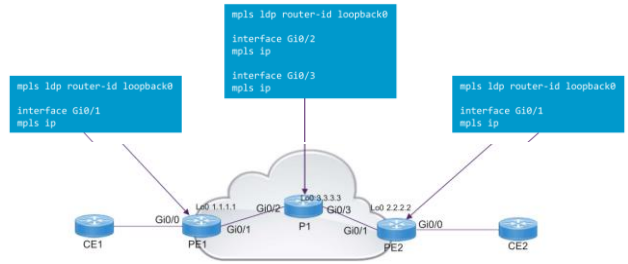
- **PE-CE Link**: CE routet Traffic zum PE, CE kann Netze mit static routes, eBGP, OSPF oder IS-IS advertisen
- **L3VPN Control plane**: Kunden-Routing wird mit VRFs separiert, Interfaces sind VRFs zugeordnet, VRFs werden mit MP-iBGP zu anderen PEs übertragen, meist mit RR
- **L3VPN Forwarding plane**: Kunden-VPN-Traffic wird mit VPN label separiert, Destination-PE erkennt VPN aufgrund des Labels

5.1.1 Konfiguration

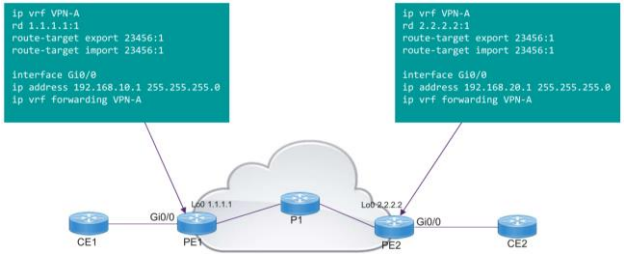
1. Provider OSPF



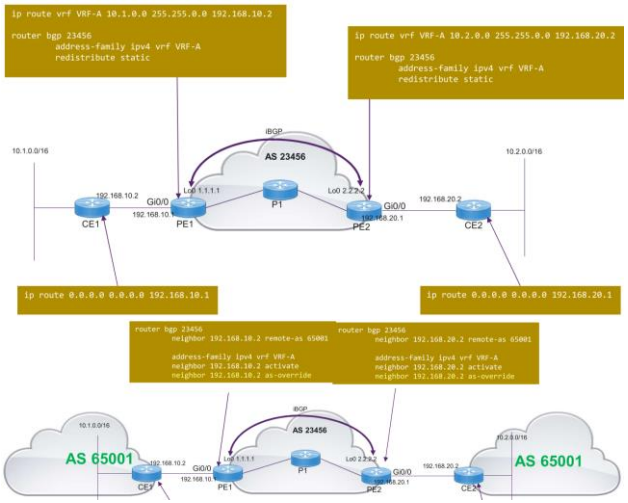
2. Provider LDP



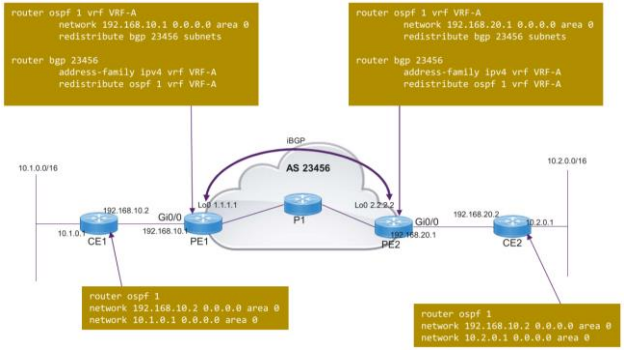
3. Provider VRF



4. Kunde Route Advertisement (Static/eBGP/OSPF)



(as-override bewirkt, dass nur AS_PATH=23456)

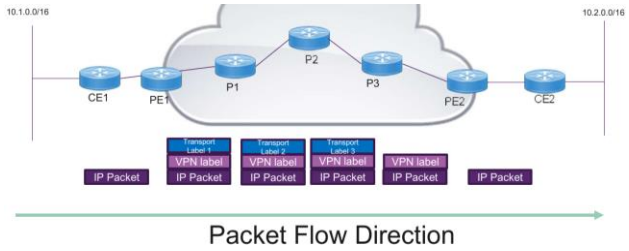


5.2 Route Propagation

Ablauf bezogen auf Topologie von Konfiguration:

1. CE1 advertised 10.1.0.0/16 an PE1
2. PE1 injected den Prefix in die VRF-Tabelle und prepended den RD
3. Die VPNv4 Route wird über MP-BGP an PE2 propagiert
4. PE2 weist die richtige VRF-Tabelle aufgrund des RT zu, der RD wird entfernt
5. PE2 sendet die Route an CE2

Label Handling:



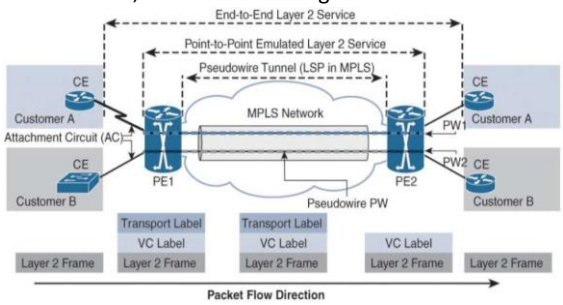
1. PE1 assigned VPN-Label, welches er von PE2 erhalten hat
2. PE1 assigned TL1 gelernt von P1
3. P1 assigned TL2 gelernt von P2
4. P2 assigned TL3 gelernt von P3
5. P3 weiss über LDP von PE2, dass er PHP machen muss

5.3 MPLS L2 VPN

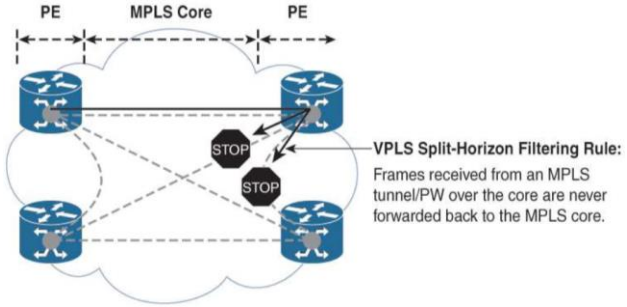
Anderer Name: Carrier Ethernet
Service Provider forwardet Traffic aufgrund MAC

Zwei Typen:

- **Virtual Private Wire Service (VPWS)**: E-Line, Pseudowire (PW), P2P Connectivity, Netzwerk ist komplett transparent für CE Router, kein MAC-Learning



- **Virtual Private LAN Service (VPLS)**: E-LAN, multipoint oder any-to-any connectivity, LAN-Segment wird emuliert, MAC-Learning/Forwarding, Flooding wenn Broad-/Multicast/Unknown Dest., Full-Mesh von Pes



6 Multicast

Wenn ein Server Daten an mehrere Clients senden möchte, müsste er für jeden Client ein einzelnes Unicast Paket erstellen, was eine Bandbreitenverschwendung ist. Daher gibt es Multicast, der Server muss nur ein Paket erstellen, die Router und Switches replizieren es, sodass es bei allen Clients der Multicast Gruppe ankommt. Vorteile: Effizienz, Performance, ermöglicht Distributed Applications

Multicast eignet sich für 1-to-n und n-to-n

Funktioniert nur mit UDP-Paketen, die Applikationen sollten also mit fehlenden/doppelten Paketen und Congestion umgehen können.

Reservierte Multicast-Adressen in 224.0.0.0/4:

- **Reserviert**: 224.0.0.0 – 224.0.0.255
  TTL=1, z.B. Adresse für alle OSPF-Router
- **IANA reserviert**: 224.0.1.0 – 224.0.1.255
  TTL > 1, z.B. für NTP
- **Administratively scoped (private)**: 239.0.0.0 – 239.255.255.255, ähnlich 192.168.0.0/16 bei IPs
- **Source Specific Multicast (SSM)**: 232.0.0.0 – 232.255.255.255

Reservierte MACs: 01:00:5E:00:00:00 – 01:00:5E:7F:FF:FF, für die Zuweisung IP-MAC sind jeweils die letzten 23 Bit identisch, somit haben jeweils 32 IPs die gleiche MAC, diese Überschneidung ist akzeptabel.

Wenn ein Host eine Multicast Gruppe joinen möchte, sendet er einen Internet Group Management Protocol (IGMP) **membership report** mit der zu joinenden Multicast-Adresse (Gruppe) **an 224.0.0.2** (alle Router). Der Router merkt sich dann das Interface und die Gruppe.

Es gibt keinen Mechanismus in **IGMPv1**, um eine Gruppe zu verlassen. Der Router schickt alle 60s eine general query message. Wenn 3 davon unbeantwortet bleiben, wird das Interface aus der Gruppe entfernt. Zum Verlassen einer Gruppe in **IGMPv2** sendet der Host eine **leave-group** message an 224.0.0.2. Der Router antwortet dann mit einem Gruppen-spezifischen query, bei keiner Antwort innerhalb von 3s wird das Interface aus der Gruppe entfernt. Bei **IGMPv3** gibt es zusätzlich die Möglichkeit, zusammen mit der Gruppe **Source-IPs zu includen und excluden**.

Multicast-Gruppen-Adressen sind nicht in der normalen Unicast Routing Tabelle gespeichert, sondern in einer speziellen Multicast-Tree Tabelle, welche von den Join/Leave messages befüllt wird.
Multicast-Routing kümmert sich darum, wo das Paket herkam, im Gegensatz zum normalen Routing, welches sich darum kümmert, wo das Paket hingeht.
Beim **Reverse Path Forwarding (RPF)** wird geprüft, ob das Paket mit einer Source Adresse auf dem Interface angekommen ist, wo es den Router verlassen würde, wenn die Source die Destination wäre. Wenn es zwei equal-cost Pfade gibt, wird dieser mit der höheren Next-Hop IP als RPF-Pfad genommen.

States:

- (*, G): Gruppe mit jeder Source
- (S, G): Gruppe mit bestimmter Source

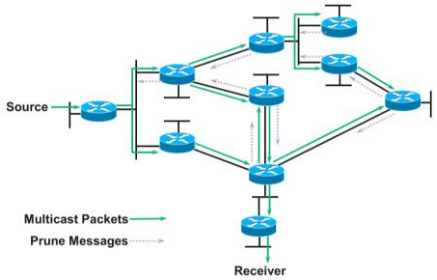
6.1 IGMP Snooping

Ohne dieses Switch-Feature werden Multicast Pakete einfach an alle Ports forwarded. Wenn es aktiviert ist, hört der Switch auf IGMP Join/Leave messages und forwarded Multicast Frames nur an die Ports, die der Group joined sind.

6.2 Protocol Independent Mcast (PIM)

6.2.1 Dense-mode

Push, Traffic wird ins ganze Netzwerk geflooded, bis er dort, wo er nicht gewünscht ist, pruned wird, alle Router erstellen (S, G) Eintrag



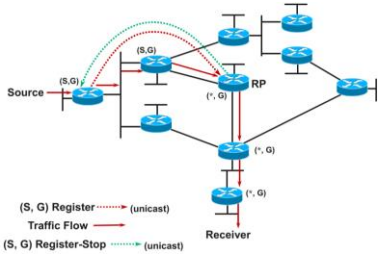
6.2.2 Sparse-mode

Pull, Traffic wird nur dorthin gesendet, wo er durch Joins gewünscht wurde

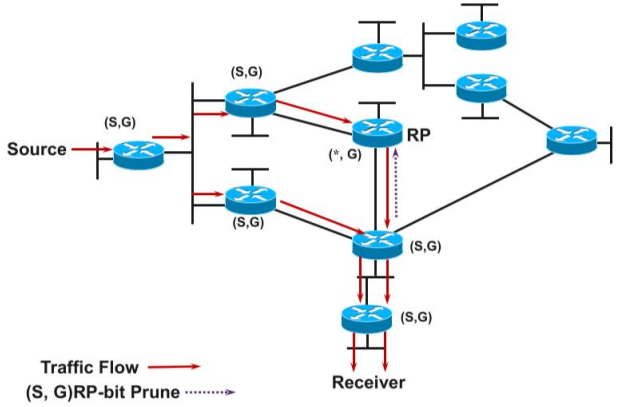
6.2.2.1 Any Source Multicast (ASM)

IGMPv1/v2 Joins, Rendezvous Point (RP) benötigt, IP davon muss jedem Router im Netz bekannt sein

1. Client sendet (*, G) Join zum Last-Hop Router (LHR), dieser leitet ihn dann weiter zum RP
2. Source sendet Mcast Traffic zum First-Hop Router, dieser sendet einen (S, G) Register zum RP
3. Da der RP bereits einen Client hat, sendet der RP einen (S, G) Join und einen (S, G) Register-Stop zum First-Hop Router (FHR)
4. Der FHR sendet nun den Mcast Stream zum Last-Hop Router als (S, G) Flow via RP

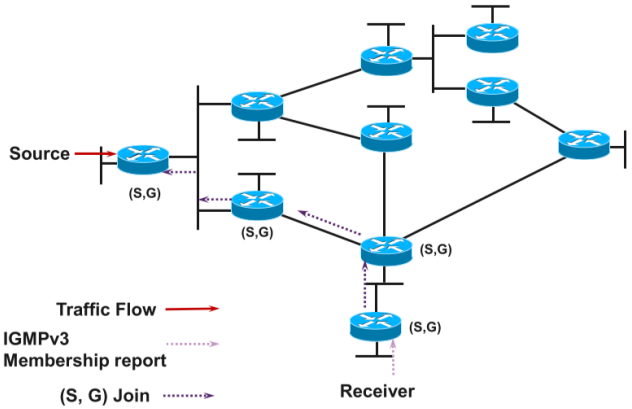


5. Der LHR überprüft seine Unicast Routing Tabelle, wie der beste Path zur Source ist, wenn ein besserer als über den RP gefunden wird, sendet der LHR einen (S, G) Join an die Source
6. Alle Router auf dem Weg zur Source fügen den (S, G) Eintrag hinzu
7. Router, welche den Traffic nun auf zwei Interfaces erhalten, senden einen (S, G) RP-bit prune Richtung RP, um den Pfad abzubauen, am Schluss sendet der RP den prune



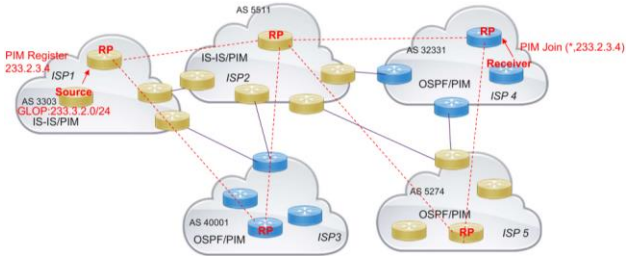
6.2.2.2 Specific Source Multicast (SSM)

IGMPv3 Joins mit Source, kein RP, Hosts sind für die source discovery verantwortlich, das ist viel einfacher als ASM, der LHR bei der Destination sendet einfach den (S, G) Join Richtung Source, somit wird von Anfang an der optimale Pfad verwendet.

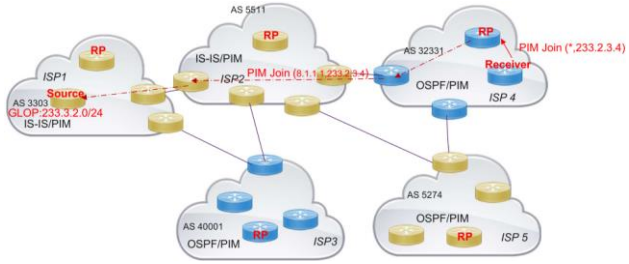


6.3 Mcast im Internet

Globally unique, static block: 233.0.0.0/8
16-bit BGP AS Nummer auf 233.X.Y.0/24 mapped



Die roten Linien sind MSDP-Peers, sie tauschen Infos über Mcast Sources und Groups aus. Der Receiver kennt somit die Source einer in der AS gejointen Gruppe und kann den optimalen Path bestimmen.



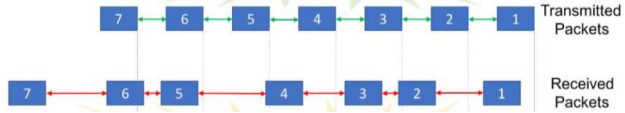
7 Quality of Service (QoS)

Netzwerke sind gebaut, um die User-Anforderungen zu erfüllen, QoS ist das Ziel. Alles, was die User-Wahrnehmung beeinflusst, fällt unter QoS. QoS sollte auch implementiert werden bei genug Bandbreite, weil Traffic bursty ist und Überlastungen in kurzen Zeitabschnitten nicht ganz ausgeschlossen werden können.

QoE = Quality of Experience

7.1 Performance-Eigenschaften

- **Delay**: Buffering verursacht Verzögerung, weil auf andere Pakete gewartet werden muss, nicht mehr als 150ms für VoIP, 400ms für Video, Data Traffic nicht sensitiv darauf wegen TCP Reorder
  - **E2E**: Benötigte Zeit für Destination App um Paket der Source App zu erhalten
  - **Network/One-way**: Vergangene Zeit zwischen erstem Bit auf Kabel und letztes Bit empfangen
  - **Transmission**: Zeit um das Paket aufs Kabel zu pushen, insignifikant bei High-Speed Links
  - **Packet processing**: Zeit, die das Netzwerkgerät benötigt, um das Paket zu verarbeiten
  - **Propagation**: Zeit, die das Signal braucht um über die Distanz zu reisen
- **Jitter**: Differenz zwischen dem höchsten und tiefsten Delay zwischen A und B, nicht mehr als 30ms für VoIP, 50ms für Video, kann mit einem De-Jitter Buffer beim Empfänger behoben werden, erst wenn der minimum playout Buffer voll ist, werden die Pakete in einer konstanten Rate zur Wiedergabe freigegeben



- **Packet loss in Prozent**: Prozent der verloreren Pakete auf dem Weg von Source zu Destination, nicht mehr als 1% für VoIP/Video, Data Traffic nicht sensitiv darauf wegen TCP Retransmit
- **Bandbreite**: mind. 30 Kbit/s für VoIP, 384 Kbit/s für Video

7.2 Buffering

Wenn verschiedene Link-Speeds im Netzwerk vorhanden sind, kann das Bottlenecks verursachen, die Router haben dann zwei Buffer:

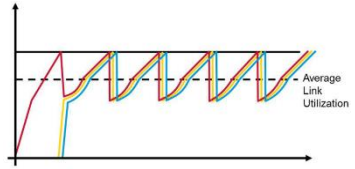
- **Incoming**: Transit, CPU-Overload füllt ihn, QoS Klassifizierung, Policing
- **Outgoing**: Enthält Pakete, die auf die Übertragung warten, QoS ist auf diesen Buffer fokussiert, CBWFQ, LLQ, Policing, Shaping

7.2.1 Queuing Algorithmen

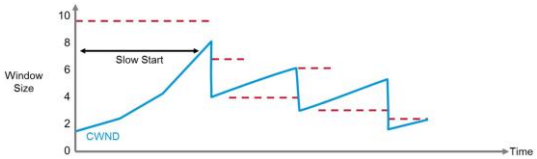
- **First In First Out (FIFO)**: keine Prio, schnell, geeig. für Links mit wenig Delay/Congestion
- **Priority Queuing**: mehrere Queues, Queue wird erst bearbeitet, wenn alle Queues mit höherer Prio leer sind, somit «verhungern» Queues mit niedriger Prio möglicherweise
- **Round-Robin**: mehrere Queues, es wird immer 1 Paket pro Queue der Reihe nach gesendet
- **Weighted Round-Robin/Weighted Fair Queuing (WFQ)**: mehrere Queues, es werden pro Runde so viele Pakete pro Queue wie in der Weight steht gesendet, Weight = IP Precedence
- **Class-Based Weighted Fair Queuing (CBWFQ)**: anstatt Weights Classes verwenden, bei denen die maximale Bandbreite, prozentuale Bandbreite und das Queue Limit gesetzt werden können, die Classes werden aufgrund von IP Precendence, DSCP, Source IPs, TCP-Ports, ... bestimmt
- **Low Latency Queuing (LLQ)**: fügt eine zusätzliche Class dem CBWFQ hinzu mit strikter Priorität für Low-Latency Apps wie VoIP

7.2.2 Queue Management

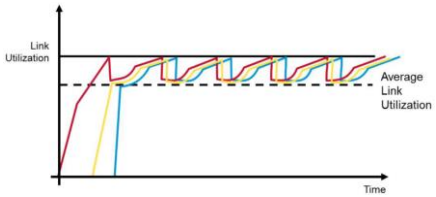
- **Tail Dropping**: neue Pakete werden gedroppt, wenn die Queue voll ist, keine differenzierten Drops mithilfe von Klassen, löst TCP Global Sync & Starvation aus
- **TCP Global Synchronization**: TCP-Sessions starten zu versch. Zeiten, Window Sizes werden grösser bis zum Tail Drop, viele Sessions werden gleichzeitig gedropped, alle starten wieder gleichzeitig, sehr ineffizient



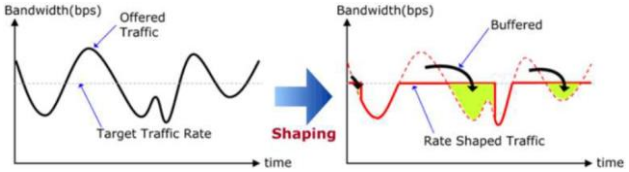
- **TCP Starvation**: passiert bei Congestion mit TCP & UDP Flows, nach dem Tail Drop verlangsamt sich TCP, UDP aber nicht, daher füllt UDP die Queues und lässt TCP «verhungern»



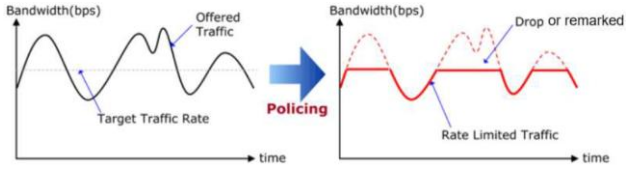
- **Random Early Detection (RED)**: ab dem Queue minimum Threshold wird eine gewisse Prozentzahl dropped, um global Sync zu verhindern, indem TCP seine Window Sizes reduziert, ab dem maximum Threshold werden alle Packets dropped, sollte für Voice Queue deaktiviert werden



- **Weighted RED**: Droppt selektiv Packets basierend auf IP precedence, DSCP or EXP
- **Shaping**: Packets, die das Bandbreiten-Limit überschreiten, werden buffered und zu einem Zeitpunkt mit weniger Last gesendet



- **Policing**: Packets, die das Bandbreiten-Limit überschreiten, werden dropped

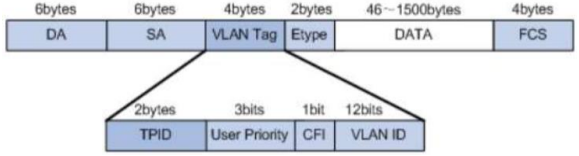


7.3 QoS Models

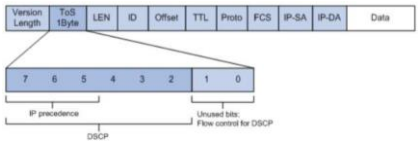
- **Best-Effort**: Default, alle Pakete gleich, kein QoS, skalierbar, einfach, schnell deployed, keine Garantien, kritische Daten behandelt wie E-Mails, wird im Internet verwendet (Netzneutralität)
- **Integrated Services (IntServ)**: E2E QoS für Echtzeit-Apps, QoS für User-spezifische Microflows
- **Differentiated Services (DiffServ)**: Soft QoS, Traffic wird in Klassen eingeteilt, skalierbar und kosteneffektiv, Policies auf Interfaces

7.4 QoS Klassifizierung

Layer 2:



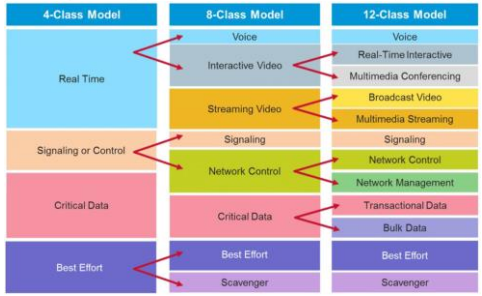
Layer 3:



Application	Layer 3 Classification		IETF
	PHB	DSCP	RFC
Network Control	CS6	48	RFC 2474
VoIP Telephony	EF	46	RFC 3246
Broadcast Video	CS5	40	RFC 2474
Multimedia Conferencing	AF41	34	RFC 2597
Real-Time Interactive or TelePresence	CS4	32	RFC 2474
Multimedia Streaming	AF31	26	RFC 2597
Call Signaling	CS3	24	RFC 2474
Low Latency or Transactional Data	AF21	18	RFC 2597
OAM	CS2	16	RFC 2474
High-Throughput or Bulk Data	AF11	10	RFC 2597
Best Effort	DF	0	RFC 2474
Low-Priority or Scavenger Data	CS1	8	RFC 3662

4-Class:

Signaling or Control
Real-Time
Critical
Critical
Real-Time
Critical
Signaling or Control
Signaling or Control
Critical
Best Effort
Best Effort
Best Effort



Network-Based Application Recognition (NBAR): Classification Engine die eine grosse Palette von Protokollen und Applikationen erkennt

7.5 Konfiguration Cisco

Define Classes of Traffic
"What traffic do we care about?"
Each class of traffic is defined using a class map.

Define QoS Policies for Classes
"What will be done to this traffic?"
Defines a policy map, which configures the QoS features associated with a traffic class previously identified using a class map.

Apply a Service Policy
"Where will this policy be implemented?"
Attaches a service policy configured with a policy map to an interface.

```
policy-map ingress
 class Customers
  police rate percent 50                    ← Parent level has only transmit and drop actions.
   conform-action transmit
   exceed-action drop
   service-policy CustomerA-policer         ← Parent level embeds child policy map.
!
policy-map CustomerA-policer                ← Child level uses percent-based policing.
 class CustomerA
  police rate percent 10                    ← Child level supports all action combinations.
   exceed-action set dscp cs1
   violate-action drop
!
interface GigabitEthernet0/0/0/0
 service-policy input ingress               ← Parent policy map is applied to the interface.
```

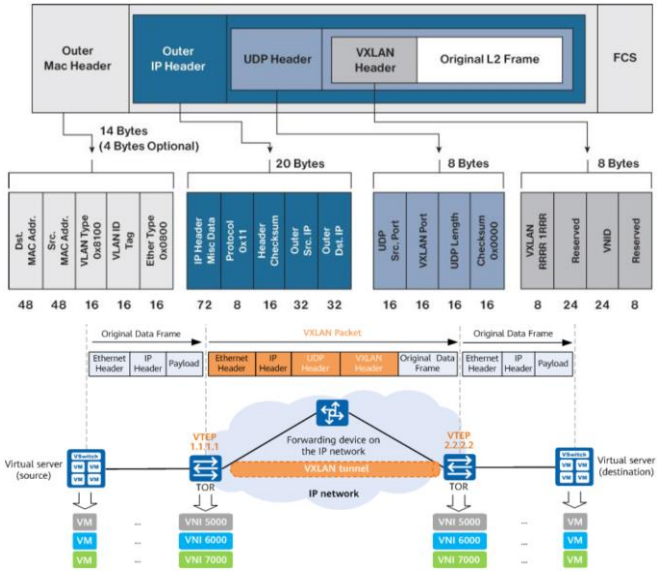
Network-Based Application Recognition (NBAR): Classification Engine die eine grosse Palette von Protokollen und Applikationen erkennt

8 EVPN-VxLAN

Das ist eine skalierbare und effiziente Art, um L2 Netzwerke über eine L3 Infrastruktur zu verbinden. Es hat somit den gleichen Zweck wie MPLS L2 VPN. Es ermöglicht Multi-Tenancy, Host Mobility und eine effiziente Nutzung der Netzwerk-Ressourcen.

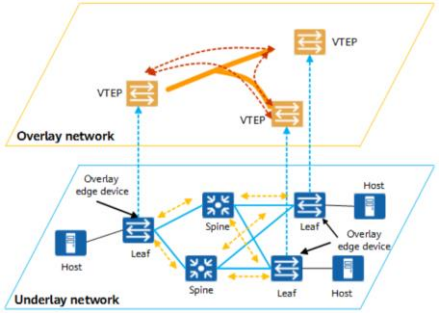
8.1 Begriffe

- **VxLAN**: Virtual eXtensible Local Area Network
- **VxLAN Network Identifier (VNI): 24Bit, 16M**
- **VxLAN Tunnel Endpoint (VTEP)**: erstellt und terminiert VxLAN Tunnels
- **Equal-Cost Multipath (ECMP)**
- **Protocol Independent Multicast (PIM)**
- **Top of Rack (ToR)**
- **EVPN Instance (EVI)**: EVPN-Instanz welche die PE-Router des EVPNs umspannt
- **Ethernet Segments (ES)**: Wenn eine Kunden-Site (Gerät oder Netzwerk) zu mind. einem PE mit einem Set von Ethernet Links verbunden ist, nennt man dieses Set ES
- **Ethernet Segment Identifier (ESI)**: Unique non-zero Identifier für ein ES



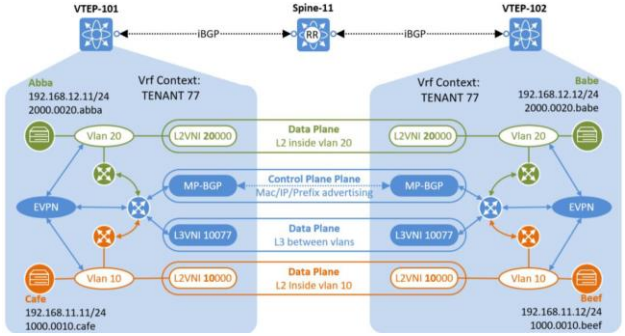
Fabric Designs:

- **Hierarchical**: Es gibt verschiedene Stufen und die Geräte sind jeweils mit zwei Geräten der oberen Stufe verbunden, zuunterst sind die Switches
- **Leaf-Spine**: Es gibt ein Core, darunter Spines (Router) und darunter Leafs (Switches), die mit allen Spines verbunden sind



Das Control plane wird entweder als Flood-and-learn oder MP-BGP EVPN betrieben. Beim EVPN werden die MACs/IPs von den VTEPs advertised, bei Flood-and-learn müssen die VTEPs einfach alles selbst lernen.

8.2 EVPN



Durch den Anycast Gateway kann jeder Leaf Switch als Default Gateway der angeschlossenen Geräte agieren, mehrere Leafs haben also die gleiche IP.

Integrated Routing and Bridging (IRB):

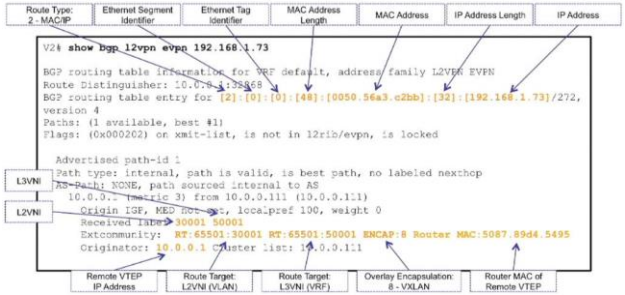
- **Asymmetrisch**: Ingress PE macht MAC Lookup, IP Lookup und nochmals MAC Lookup. Egress PE macht nur einen MAC Lookup.
- **Symmetrisch**: Ingress PE macht MAC Lookup, dann IP Lookup. Egress PE mach IP Lookup, dann MAC Lookup.

8.3 MP-BGP

Über BGP (meist mit RR) werden die Route Types an alle VTEPs gesendet, sodass alle den gleichen Stand haben. Die VTEPs speichern sich dann jeweils die Routes in die RIB/FIB inkl. Next Hop.

8.3.1 Route-type 2: Host Advertisement

VTEP sendet, wenn ein Host angeschlossen wird. Zwingend: MAC/L2VNI, optional: IP/L3VNI



8.3.2 Route-type 3: Inclusive Multicast Ethernet Tag Route

VTEP advertised seine Teilnahme an einer EVI.

- Prefix:

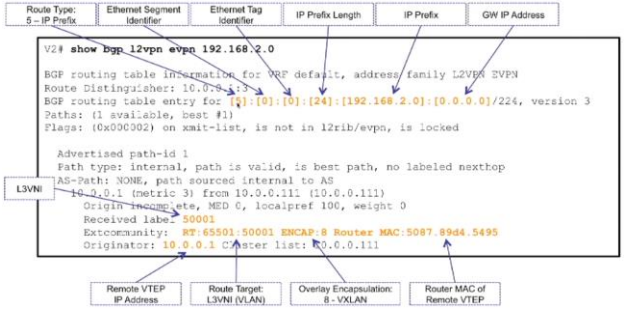
```
+---------------------------------+
| RD (8 octets)                   |
+---------------------------------+
| Ethernet Tag ID (4 octets)      |
+---------------------------------+
| IP Address Length (1 octet)     |
+---------------------------------+
| Originating Router's IP Address |
|        (4 or 16 octets)         |
+---------------------------------+
```

- PSMI (Provider Multicast Service Interface)

```
+---------------------------------+
| Flags (1 octet)                 |
+---------------------------------+
| Tunnel Type (1 octets)          |
+---------------------------------+
| MPLS Label (3 octets)           |
+---------------------------------+
| Tunnel Identifier (variable)    |
+---------------------------------+
```

8.3.3 Route-type 5: Subnet Route Advertisement

Es können directly connected, statische oder durch Routingprotokolle gelernte Routen von VTEPs advertised werden.



8.4 Host Detection/Deletion/Move

Detection: Host wird angeschlossen, MAC kommt in MAC Address-table, evtl. broadcastet der Host einen ARP Request, somit erhält der VTEP die IP (ARP/ND Snooping), VTEP sendet Type 2 Adv.
Deletion: Host wird disconnected, MAC Eintrag timed out nach 1800s, ARP Eintrag nach 1500s, VTEP withdrawed und löscht Host
Move: Neuer VTEP advertised Host mit neuem Next-Hop, höherer Seq Nummer und fügt die MAC Mobility Sequence dem Extcommunity hinzu, der alte VTEP erhält das neue Advertisement und aktualisiert die Einträge

8.5 Early ARP Termination

Wenn ein Host einen ARP Request L2-broadcastet nach der MAC einer IP, die dem VTEP bekannt ist, antwortet der VTEP direkt mit der MAC, anstatt den Request an den Host weiterzuleiten, der sich an einem anderen VTEP befindet.

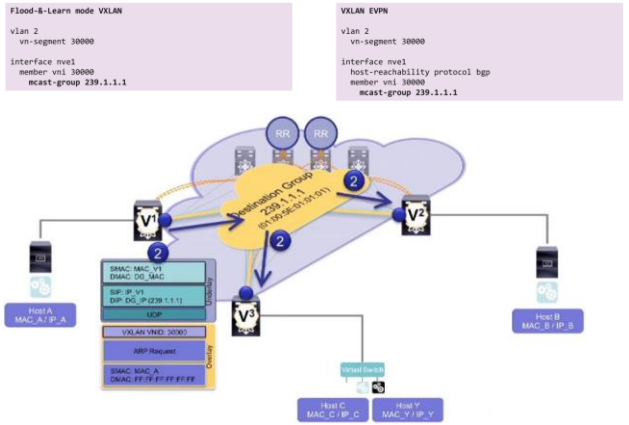
Fragt eine Host nach einer dem VTEP unbekannten IP, wird der Request an alle VTEPs der EVI broadcastet, welche ihn dann in ihre Netze broadcasten. Wenn eine Antwort kommt, advertised der VTEP mit dem gesuchten Host die MAC und evtl. IP.

8.6 Broadcast/Unknown Unicast/ Multicast (BUM) Traffic

2 Methoden, um diese Art von Traffic zu replizieren:

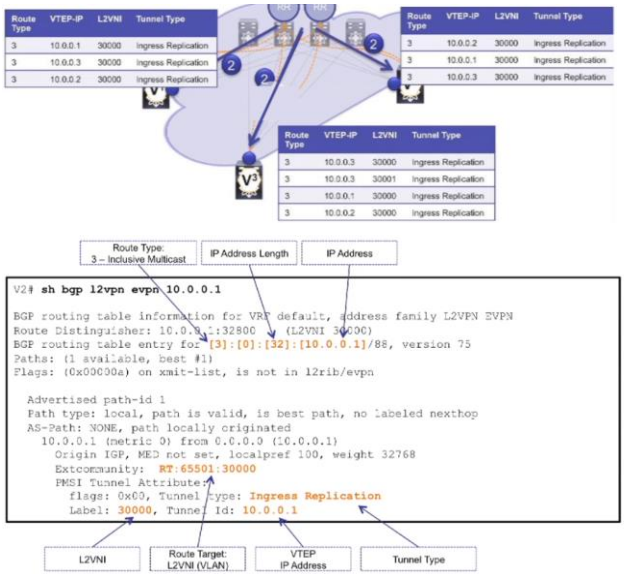
8.6.1 Multicast

Effizient/optimiert/empfohlen, Traffic wird als einzige Kopie an eine Multicast Gruppe mit allen benötigten VTEPs gesendet



8.6.2 Unicast (Ingress Replication)

Weniger effizient, sollte nur verwendet werden, wenn Multicast nicht vorhanden ist, Source VTEP muss eine Kopie für alle benötigten VTEPs erstellen und sie ihnen einzeln als Unicast senden



In Flood-&-Learn mode VXLAN, remote VTEPs for Ingress replication are statically configured

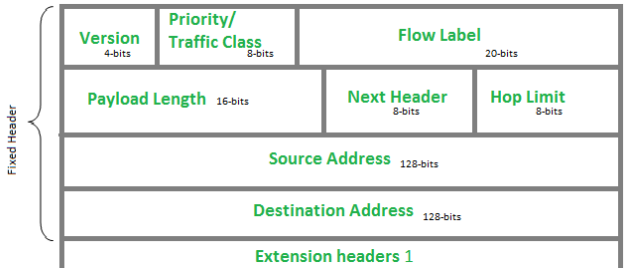
In VXLAN EVPN, remote VTEPs automatically learnt through EVPN inclusive Multicast Ethernet Tag (IMET) routes

9 IPv6 Security

9.1 IPv6

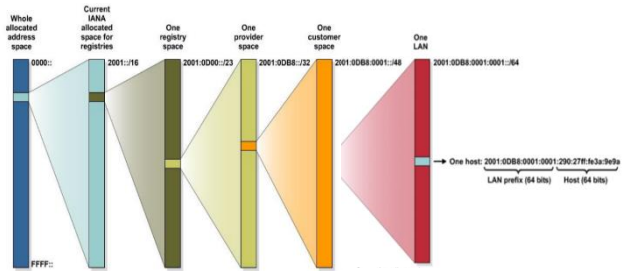
Vorteile gegenüber v4: genug Adressen, einfacheres Subnetting, kein NAT, End-to-End Transparenz, fördert Innovation

9.1.1 Header



- IPv6 Header Grösse: fix 40 Bytes
- Version: 6
- Traffic Class: DS Field von IPv4 (für QoS)
- Flow Label: für spezielle QoS Rules, normalerweise 0
- Payload Length: Grösse Paket ohne Header
- Next Header: Gibt an, ob TCP/UDP oder ein bestimmter Extension Header kommt
- Hop Limit: TTL von IPv4
- Fragmentation gibt es nicht, stattdessen führt jeder Host eine PATH MTU Discovery durch, so weiss er, was für eine MTU er verwenden muss für jede IPv6 Adresse. Ein IPv6 Router schickt dem Client eine ICMPv6 Packet Too Big message (type 2 code 0), wenn das Package zu gross ist.
- Checksumme gibt es nicht, weil das TCP/UDP schon haben

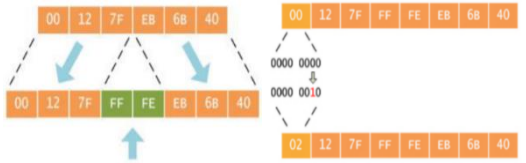
9.1.2 Adressen



Vereinfachung: Führende Nullen können weggelassen werden, Nullerblöcke nacheinander können einmal pro Adresse mit :: ersetzt werden

Beispiel: 2001:0DB8:0000:1133:0000:0000:0000:0200 → 2001:DB8:0:1133::200

Beim EUI-64 wird aus der MAC-Adresse die 64 Host Bits der IPv6 Adresse generiert (FFFE in Mitte einfügen, Bit 7 flippen)



Ein IPv6 Gerät hat meist mehrere Adressen:

- **Global Unicast**: weltweit einzigartig und routable, stammt aus dem Range 2000::/3, Generierungs-Optionen: Manuell, EUI-64, Stateless autoconfiguration (SLAAC, Prefix und Prefix Length kommt vom RA, der Rest von EUI-64), SLAAC+ statelless DHCP (gleich wie SLAAC, jedoch ist ein DHCP vorhanden für Optionen wie DNS), Stateful autoconfiguration (DHCPv6)
- **Link Local**: muss für jedes NIC vorhanden sein, kommt aus dem Range fe80::/10 (meist fe80::/64), Scope ist auf das lokale Netz beschränkt, Generierungs-Optionen: EUI-64 oder manuell, wird gebraucht für NDP
- **Unique Local Address**: wird verwendet für Kommunikation in mehreren Netzwerken, ist aber nicht im Internet routable, aus Range fc00::/7

Spezielle Unicast Adressen:

- Localhost: ::1
- Unspecified address: ::/128
- Dokumentations-Prefix: 2001:0db8::/32
- Discard-Prefix: 0100::/64
- Default Route/Unspecified: ::/0

Spezielle Multicast Adressen:

- FF02::1: alle Nodes (MAC dazu: 33:33:00:00:00:01)
- FF02::2: alle Router (MAC dazu: 33:33:00:00:00:02)
- FF05::1:3: alle DHCPv6 Server(MAC dazu: 33:33:00:01:00:03)
- Solicited Node: ff02::1:ff: + letzte 24 Bit der Unicast Adresse
  Beispiel: FE80::200:CFF:FE3A:8B18 → FF02::1:FF3A:8B18
  ist für **JEDE** IPv6 Ad. vorhanden, wird für ND/DAD verw.

Neighbor Discovery (ARP-Ersatz zur MAC-Adressenfindung):

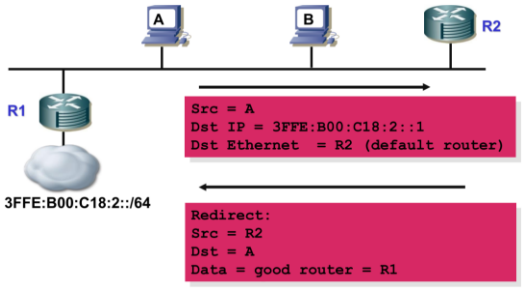
1. A möchte ein IPv6 Paket an die globale oder Link-local IPv6 Adresse von B schicken
2. A berechnet Solicited Node (SN) Multicast Adresse von B
3. Neighbor Solicitation senden (Src=IPv6 Adresse von A, Dst=SN von B, Data=MAC von A, Query=MAC von B?)
4. Neighbor Advertisement senden (Src=IPv6 Adresse von B,Dst=IPv6 Adresse von A, Data=MAC von B)

Duplicated Address Detection (DAD) funktioniert ähnlich wie Neighbor Discovery. Bei Step 3 ist die Destination jedoch die SN von sich selbst, kommt keine Antwort, ist die Adresse unique.

Die **Autokonfiguration** wird mit Router Solicitations (RS) und Router Advertisements (RA) gemacht:

- Hosts senden RS beim Booten, um RAs zu erhalten (Src=unspecified,Dst=alle Router)
- RA: Router Konfiguration, wird periodisch und nach RS gesendet (Src=Router Link local Adresse, Dst=alle Nodes, Data=options, prefix, lifetime, autoconfig flag)

Redirects:



9.2 ICMPv6

Types: 0: Reserved, 1: Destination unreachable, 2: Packet Too Big, 3: Time Exceeded, 4: Parameter Problem, 9: RA, 10: RS

Codes: 0: No route to destination, 1: Communication with the destination is administratively prohibited (e.g. firewall), 2: Beyond scope of the source address, 3: Address unreachable, 4: Port unreachable

Darf nicht blockiert sein, sonst funktioniert nichts mehr, weil ND, DAD und Autokonfiguration nicht mehr funktionieren.

Types, die **blockiert** werden sollten:

- Unallocated Errors: 5-99, 102-126
- Unallocated Informational: 155-199, 202-254
- Experimental: 100, 101, 200, 201
- Extensions: 127, 255

Types, die **vom und ins Internet erlaubt** werden müssen: 1 – 4, 128 (Echo Request) & 129 (Echo reply) empfohlen

9.2.1 Angriffe

- Router können readressiert werden mit Type 138 (Router Renumbering), daher blockieren
- Types 139 (Node Information Query) & 140 (Node Information Response) können zu viele Infos liefern, daher block
- ICMPv6 Errors können für versteckte Channels missbraucht werden. Daher enthalten die Errors Teile des originalen Pakets, wenn ein Error nicht eine Antwort auf einen stateful Flow in die andere Richtung ist, sollte er dropped werden.
- Es sollte eine Default Deny Policy für ICMPv6 Pakete geben und nur die explizit erlaubten Messages durchgehen.

9.3 Multicast Security

Es gibt keinen Broadcast in IPv6, stattdessen Multicast.

Um einen blinden Angriff zu starten, kann einfach eine Well-Known Multicast Adresse verwendet werden, es gibt keine Reconnaissance Phase. Am Netzwerk-Perimeter sollten daher global und Site-Local Scope Multicast Pakete blockiert werden.

Es können auch Multicast Adressen als spoofed Source Adressen verwendet werden, um eine DoS Attacke auszulösen. Pakete mit solchen Sources sollten immer dropped werden und nicht mit Errors beantwortet werden.

9.4 Extension Header Threats

Das Next Header Feld des IPv6 Headers identifiziert den folgenden Extension Header. Beim letzten ist das Feld leer. Folgende Extension Headers können missbraucht werden:

- **Hop-by-Hop Options**: Das Padding darin stellt sicher, dass das Paket auf einer Oktettgrenze endet, meist wird es nicht gebraucht, weil der Header und Options bereits aligned sind. Im Pad1 (1 Oktett Padding in den Options) oder PadN (variable Padding-Grösse in den Options) könnten sich Infos für einen verdeckten Channel befinden. Firewalls sollten Pakete mit mehreren Padding Options, Pakete mit mehr als 5 Bytes Padding und Padding, das nicht ausschliesslich aus Nullen besteht, blockieren.
  Dieser Extension Header muss von allen Hops auf dem Weg bearbeitet werden. Wenn viele Pakete mit Router Alert Optionen ankommen, kann das Performance-Probleme auf den Routern verursachen, daher sollten nur Router Alert 0 – 35 erlaubt werden und 36 – 65536 blockiert werden.
- **Routing Options**: Alle Router müssen diesen Header bearbeiten. Beim Type 0 bestimmt der Absender mit Source Routing den Weg des Pakets. Weil die Destination IPv6 Adresse bei jedem Hop ändert, ist es schwierig, solche Pakete rauszufiltern, die Firewall muss daher tief ins Paket schauen, alle Header und Optionen analysieren und entscheiden. RH0 Pakete, die die Interface IP des Routers im String of Nodes haben, sollten dropped werden. Cisco ACL Keyword: *no ipv6 source-route*
- **Fragmentation Options**: Fragmentation darf ausschliesslich vom Source Endgerät gemacht werden. Router senden einfach ein ICMPv6 Type 2 zurück, wenn es zu gross ist. Angreifer können probieren, im ersten Paket einen für die Firewall akzeptablen Header zu verstecken und dann in den restlichen Fragmenten den Angriff zu senden. Firewalls sollten Fragmente kleiner als 1280 Bytes droppen. Ausserdem können durch überlappende, aus der Reihe fallende, unvollständige oder nested Fragmente Hosts zum Absturz bringen, da das Zusammensetzen CPU/Memory braucht. Firewalls können nachfolgende Fragmente ohne L4 Header nicht einfach blockieren, da sonst auch legitime Fragmente blockiert werden. Die Lösung ist Virtual Fragment Reassembly (VFR), dabei setzt die Firewall Fragmente zusammen und kann sie so als Ganzes analysieren, es braucht aber entsprechend Hardware-Leistung. Keywd: *fragments*
- **Destination Options**: Dieser Extension Header muss vom Receiver bearbeitet werden. Router Alerts darin verursachen nur Performance-Probleme und können ganz blockiert werden.
- **Unknown Options Header**: L3 Geräte und Firewalls sollten alle Pakete mit unbekannten Extension Headers droppen und mit ICMPv6 Type 4 Code 1 antworten. Dafür kann bei Cisco ACLs das Keyword *undetermined-transport* verwendet werden, es werden auch Pakete ohne L4 Header dropped, OSPFv3 Hellos ohne L4 Header müssen explizit erlaubt werden.

9.5 Reconnaissance

Weil IPv6 Subnetze so riesig sind, ist es unmöglich, sie mit Ping Sweeps nach Hosts zu durchsuchen. Wenn ein Angreifer aber bereits im LAN ist, kann er einfach schauen, wer auf dem Multicast für alle Nodes (ff02::1) antwortet.
Hostlists sind sehr gesucht von Angreifern, sie können auch aus dem Neighbor Cache der Hosts (IPv6 Adresse / MAC) gezogen werden, anderen Logs oder DHCPv6 Records.

Um es dem Angreifer möglichst schwer zu machen, werden zufällige Adressen empfohlen und dass der erste Router nicht die erste oder letzte IP hat.

9.6 L3/L4 Spoofing

Mithilfe von Unicast Reverse Path Forwarding (RPF) kann überprüft werden, ob eine Source Adresse gespooft ist. Nämlich muss das Paket auf dem Interface ankommen, an welches es der Router schicken würde, wenn die Source die Destination wäre. **Strict Mode** dropped Pakete, die den RPF-Check failen**. Loose Mode** prüft, ob die Source irgendwo in der Routingtabelle vorhanden ist und prüft nicht das Interface.
Der allow-default Parameter wird verwendet für Lookups auf dem Interface mit der Default Route.

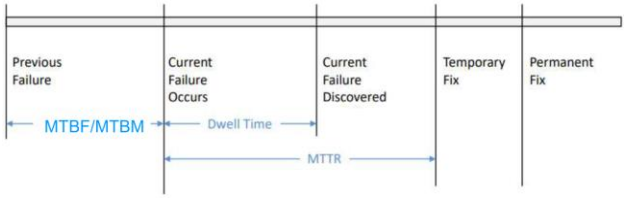
9.7 Local Network Attacks

Beim **SLAAC** kann ein Angreifer auf einen RS eines neuen Hosts mit einem **falschen RA antworten**, der einen falschen Präfix und Router enthält. Der Host generiert dann eine falsche Adresse und hat keine Internetverbindung (**Black-Holing**). RAs von Routern sollten daher mit höchster Prio gesendet werden. Ebenfalls können RAs von falschen MACs oder Ports blockiert werden. Mit der M-Flag in den RAs kann stateful DHCPv6 enforded werden, somit kann DHCPv6 Snooping mit MAC/IP betrieben werden.
Beim **Neighbor Discovery (ND)** möchte ein Host die MAC eines anderen Hosts herausfinden, er multicastet dazu den Type 135 Neighbor Solicitation Request an die Solicited Node Multicast Adresse. Der andere Host antwortet dann mit seiner MAC. Der Angreifer kann darauf **mit einer falschen MAC antworten**.
Duplicate Address Detection (DAD) ist ähnlich wie ND, der Angreifer kann einfach auf jedem Multicast antworten, sodass der Node **nie eine unique IP findet**. Es kann behoben werden, indem die Mappings von IPs/MACs den Hosts bekannt sind.

Mit dem Type 137 Redirect kann ein Router dem Host einen besseren Router vorschlagen. Der Redirect muss das originale Paket enthalten, der Angreifer kann also nicht einfach so auch einen Redirect senden. Das kann leicht umgangen werden, indem der Angreifer den Host pingt, somit den echo reply genau kennt und somit einen passenden Redirect senden kann. Redirects können deaktiviert werden auf Interfaces.

SLAAC, NDP, and DAD sind auf Link-Local und unspecified Adressen als Source beschränkt und haben ein Hop Limit von 255, somit sind alle diese Angriffe aufs lokale Netzwerk beschränkt.

10 Troubleshooting



MTBF/M: Mean time between Failure/Mistake
MTTR: Mean time to Repair
Dwell time (grey failure)
MTTI: Mean time to Innocence

Für Resilience muss die MTTR tief und die MTBF hoch sein.

10.1 MTBF

$$MTBF = \sum_{k=1}^{N} \frac{1}{(\frac{1}{MTBF_k})}$$

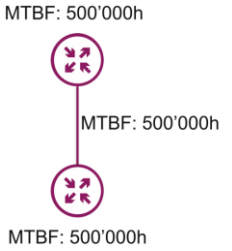
2500 Router je 64 Ports (219'000h), somit 160'000 optische Einheiten (879'000h) und 80'000 optische Links (879'000h)

$$MTBF\ Router = \frac{1}{2500 \cdot \frac{1}{219000}} = 87.6h$$

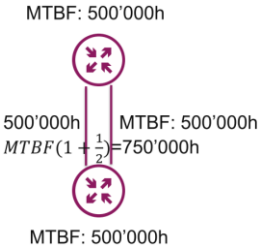
Bei der Optik grössere der Zahlen 160'000/80'000 nehmen:

$$MTBF\ Optik = \frac{1}{160000 \cdot \frac{1}{879000}} = 5.5h$$

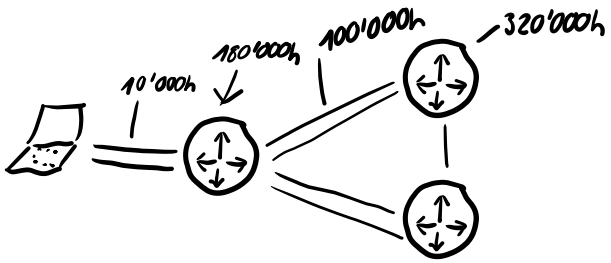
$$Availability = \frac{MTBF}{MTBF + MTTR}$$



$$\frac{1}{3 \cdot \frac{1}{500'000}} = 166'666h$$



$$\frac{1}{2 \cdot \frac{1}{500'000} + \frac{1}{750'000}} = 187'500h$$



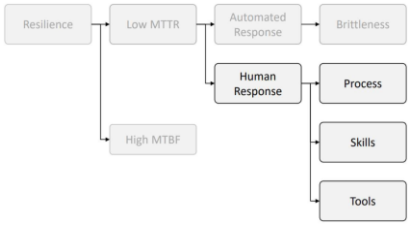
Access-Distribution Link: $1.5 * 100\ 000 = 150\ 000h$

Distribution Router: $1.5 * \frac{1}{\frac{1}{150\ 000} + \frac{1}{320\ 000}} = 153\ 191h$

Access Router: $1.5 * \frac{1}{\frac{1}{153\ 191} + \frac{1}{180\ 000}} = 82\ 758h$

Access-PC Link: $\frac{1}{\frac{1}{82\ 758} + \frac{1}{15\ 000}} = 12\ 698h$

10.2 MTTR



10.2.1 Automated Response

Eine Automated Response ist z.B. die Routing Protokoll Convergence, optional mit FRR. Automated Responses verkleinern die MTTR, wenn sie richtig designed sind und funktionieren, können sie aber verlängern bei schlechtem Design und unerwarteten Konsequenzen. Diese unerwarteten Konsequenzen nennt man Brittleness, beispielsweise wenn die automated Response einen Traffic Loop oder Crashing Loop generiert.

10.2.2 Human Response

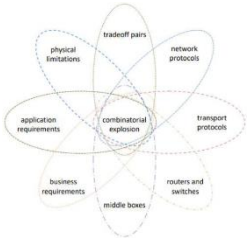
- **Technical Debt**: Differenz zwischen wie das System funktioniert und wie man denkt wie es funktioniert. Man wählt im Stress, wenn das Netzwerk down ist, oft eine einfache Lösung anstatt eine aufwändigere und bessere, welche länger standhält.
- **Temp Fix:** Man weiss, dass es so bald wieder kaputtgeht und die Technical Debt grösser wird, applied den Fix aber trotzdem, weil es schnell gehen muss.
- **Permanent Fix:** Man weiss, dass der Fix lang hält und die Technical Debt kleiner wird oder gleichbleibt.

10.2.2.1 Komponenten

Forwarding plane:

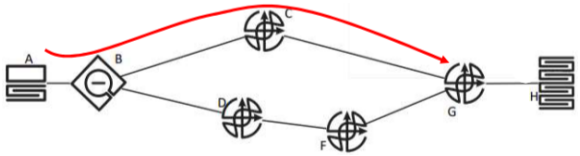
In einem grösseren Netzwerk weiss niemand im Team alles, stattdessen kennt jeder Engineer seinen Home Layer genau und seine Neighbor Layers ein bisschen.

Combinatorial Explosion:



ISO/OSI Layer: 1 Physical Layer, 2 Data Link Layer, 3 Network Layer, 4 Transport Layer, 5 Session Layer, 6 Presentation Layer, 7 Application Layer

Manipulability Theory: Beweisen, dass das Problem «hier» liegt, bevor man im «hier» tiefer geht.
Beispiel: Roter Path geht nicht, Annahme C macht Probleme



Man erstellt eine statische Route über D, wenn das Problem verschwindet, kann es am Outgoing Interface von B, dem ganzen Router C oder dem Incoming Interface von G liegen.

Beim Equal Cost Multipath (ECMP) gibt es exponentiell viele mögliche Paths, fürs Troubleshooting empfiehlt es sich, sowohl den Forward als auch Reverse Path durch das Ausschalten von Links durch einen bestimmten Path zu lenken.

10.2.2.2 Heuristics

Heuristics sind Eselsbrücken, um Troubleshooting zu vereinfachen. Folgendes Vorgehen ist praxiserprobt:

1. Zusammenhänge zwischen der Verhaltensweise und kürzlichen Software-Changes suchen.
2. Wenn keine Changes gefunden, Suche erweitern auf andere potenzielle Mitwirkende.
3. Die diagnostische Richtung in Richtung des ersten Gedankens lenken, zu dem die Symptome passen. Alternativ können auch Erfahrungen aus früheren Events genutzt werden, welche eine schwierige Diagnose hatten.
4. Bei Fixes sollten die Changes peer reviewed werden anstatt nur automatisch getestet werden, wenn überhaupt.

10.2.2.3 Random Walk

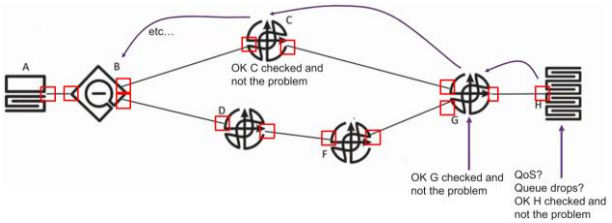
Beim Random Walk hört der Engineer auf seine Intention und sucht sich zufällig durch die möglichen Gründe des Ausfalls. Nicht die ganze Troubleshooting Session in einem Random Walk feststecken, die MTTR/MTTI verkürzt sich damit nicht, die Technik kann sehr ineffizient sein. Es könnte auch sein, dass man etwas sucht, das nicht existiert, trotzdem wird diese Technik meistens angewandt.

Es funktioniert jedoch sehr gut, wenn man etwas schon mal gesehen hat und die Lösung kennt, man nimmt so eine Abkürzung und findet die Ursache sehr schnell. Wenn man aber mehrmals denkt, man hätte es schon gesehen, verliert man schnell den Überblick über das ganze System.

In jedem Fall sollte man bei den Basics starten (kein it's always DNS), man fokussiert sich möglicherweise auf das Falsche. Den Fokus immer darauflegen, was das System macht und was es tun sollte.

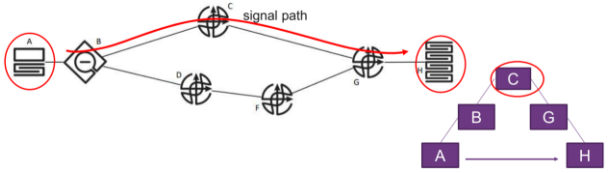
10.2.2.4 Linear Walk

Beim linear Walk sortiert man die Möglichkeiten und geht sie der Reihe nach durch. Es ist effizient, wenn man an der richtigen Stelle startet, oft ist er aber ineffizient.

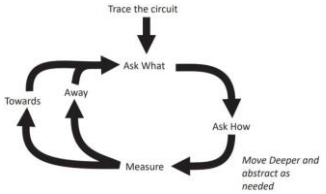


10.2.2.5 Sorted Walk

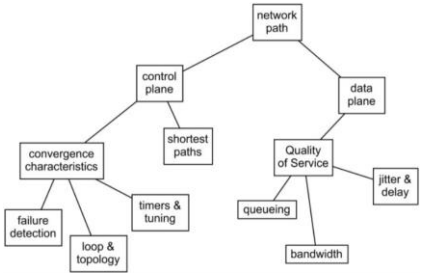
Beim Sorted Walk erstellt man einen Baum



Zunächst wählt man die Mitte, also C aus. Wenn das Signal bei C nicht korrekt ist, muss es bereits vorher, also in Richtung der Source kaputt gegangen sein, man macht bei B weiter. Wenn das Signal bei C korrekt ist, muss es nachher kaputt gegangen, sein, also in Richtung der Destination, man macht bei G weiter.



10.2.2.6 Troubleshooting for Network Paths



1. Man sollte das System bereits vor dem Failure genau kennen, dadurch kann man viel schneller arbeiten und verliert sich nicht.
2. Man sollte wissen, wie es normalerweise funktioniert, wie und warum es so funktioniert.
3. Man sollte sich die Half Split Punkte überlegen durch Aufzeichnen des Problems.
4. Massnahmen überlegen, was man ändern könnte, um das Problem zu verstehen ohne weiteren Schaden anzurichten.

11 Graph Theorie

11.1 Networks & Graphs

Ein Netzwerk ist ein System aus Nodes (Subcomponents), die durch Links (Connections, Relationships) verbunden sind. Der Begriff Netzwerk wird von Physikern und Engineers bevorzugt, der Begriff Graph von Mathematikern. Mathematisch ist ein Netzwerk ein Set von Vertices V (Nodes) und eine Liste von Edges E (Links).

Ein Link ist ein Paar von Nodes: $e = (v, v') \in E$

Bei **directed Netzwerken** spielt die Reihenfolge von v und v' eine Rolle, bei **undirected Netzwerken** nicht. In **weighted**

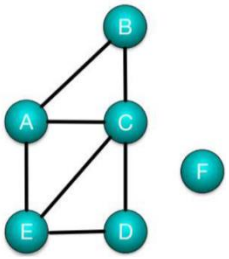
Netzwerken haben die Links zusätzlich noch ein Weight, welches die Wichtigkeit vom Link beschreibt.

Brücken von Königsberg: Ein Pfad, der alle Vertices einmal besucht und dann zum Start-Vertice Node zurückkehrt, existiert nur unter zwei Bedinungen:

- Die Anzahl Vertices muss gerade sein
- Kein Vertice oder zwei Vertices dürfen eine ungerade Anzahl Edges haben

Edge List:

$$\begin{bmatrix} (A,B) \\ (A,C) \\ (A,E) \\ (B,C) \\ (C,E) \\ (C,D) \\ (D,E) \end{bmatrix}$$



Adjacency Matrix:

	A	B	C	D	E	F
A	0	1	1	0	1	0
B	1	0	1	0	0	0
C	1	1	0	1	1	0
D	0	0	1	0	1	0
E	1	0	1	1	0	0
F	0	0	0	0	0	0

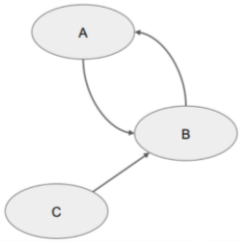
Adjacency List/Dict:

```
{'A': {'B', 'C'}, 'B': {'C'}, 'C': {'E', 'D'}, 'D': {'E'}}
```

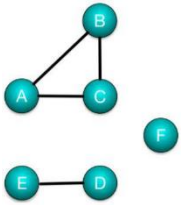
```
{'A': {'B': {}, 'C': {}, 'E': {}},
 'B': {'A': {}, 'C': {}},
 'C': {'A': {}, 'B': {}, 'D': {}, 'E': {}},
 'E': {'A': {}, 'C': {}, 'D': {}},
 'D': {'C': {}, 'E': {}}}
```

Mathematische Notation:

G = {V, E}
V = {A, B, C}
E = { {A,B}, {B, A}, {C, B} }



11.2 Graph Properties



Es ist möglich, dass Nodes disconnected sind (F) oder grössere Teile disconnected sind (E, D). Jeden Teil des Graphs nennt man **Component** oder **Subgraph**. Der Component mit der grössten Anzahl Nodes (A, B ,C) wird G**iant Connected Component** genannt.

Connectedness/Cycle-Arten:

- **Strongly connected**: Jedes Node ist erreichbar von jedem anderen Node mit Beachtung der Edge-Directions.
- **Weakly connected**: Jedes Node ist erreichbar von jedem anderen Node ohne Beachtung der Edge-Directions.

Ein **Cut Edge** disconnected das Graph, wenn er entfernt wird uns ist nicht in einem Cycle. Oder Degree=1

Ein Graph ist **complete**, wenn es einen Edge von jedem Vertex zu jedem Vertex gibt. $Degree\ von\ jedem\ Vertex\ = N-1$

Ein Graph ist **cyclic**, wenn er einen Cycle in sich hat.

Ein **connected** Graph kann **acyclic** sein. Ein **complete** graph kann **cyclic** sein. Ein **undirected** Graph, bei dem jedes Node **Degree=2** hat ist immer **cyclic**.

Ein **Path** hat keine Einschränkungen bei der Länge, ausser dass jeder Edge nur einmal besucht werden darf. Ein Path von u nach v ist ein Subgraph mit Edges $\{u, v0\}, \{v0, v1\}, \ldots, \{vn, v\}$

Ein **Walk** ist ähnlich wie Path, jedoch dürfen Edges und Vertices auch mehrmals besucht werden.

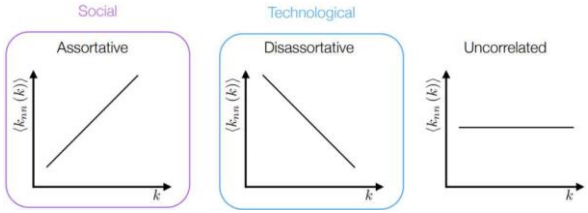
Der **Diameter** eines Graphs ist der durchschnittliche Shortest path zwischen allen Node-Paaren.

Der Degree k eines Nodes ist die Anzahl Connections des Nodes. Bei directed networks gibt es k_{in} und k_{out}.

k von A: 3 $\quad P(k) = {}^{1}/_{N} \sum_{k_i}^{N} 1$
k von B: 2 $\quad P(0) = {}^{1}/_{6}$
k von C: 4 $\quad P(1) = 0$
k von D: 2 $\quad P(2) = {}^{1}/_{3}$
k von E: 3 $\quad P(3) = {}^{1}/_{3}$
k von F: 0 $\quad P(4) = {}^{1}/_{6}$

Durchschnittlicher Degree der Nachbarn:

$$\langle k_{nn}(k) \rangle = \sum_{k'} k' P(k'|k)$$

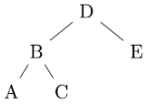


Assortativity coefficient r:

- Pos r (Rich-club): Nodes mit hohem (tiefem) Degree verbinden sich bevorzugt mit Nodes mit hohem (tiefem) Degree.
- Neg r (Hierarchisch, Baum): Nodes mit hohem (tiefem) Degree verbinden sich bev. mit Nodes mit tiefem (hohem) Degree.

11.3 Graph Traversal Algorithmus

11.3.1 Breadth First



Jede Hierarchiestufe wird nacheinander von links nach rechts traversiert

D-B-E-A-C

11.3.2 Depth First

Es wird ebenfalls jede Hierarchiestufe nacheinander von links nach rechts traversiert, jedoch erst wenn alle Kinder abgedeckt sind: D-B-A-C-E

11.3.3 Dijkstra

1. Beginn der Tabelle (aus Perspektive R9, R9 ist root)
2. Alle Neighbors von R9 in die Candidates schreiben (wenn der zweite Router bereits im Tree ist, nicht adden)
3. Die Cost to Root ablesen und in diese Spalte schreiben
4. Niedrigste Cost in den Tree übernehmen
5. Candidates werden aus 2 Gründen gestrichen:
   a. Der zweite Router (z.B. bei «R1, R2, 3» ist R2 der zweite) ist bereits bei einem Eintrag im Tree zweiter Router
   b. Es sind zwei Einträge in den Candidates mit dem gleichen zweiten Router, dann werden alle Einträge ausser dieser mit der niedrigesten Cost gestrichen

Das Ganze wird wiederholt, bis die Anzahl Einträge im Tree gleich der Gesamtanzahl Router ist

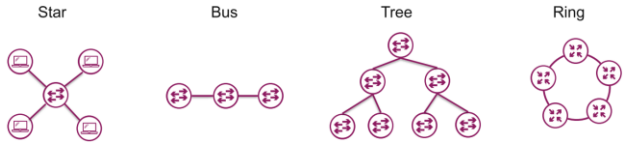
Candidate	Cost from the Root to the Candidate	Nodes within the Tree
R9,0	R9, 0 ✓	R9,0
R9, R4, 5	R9, R4, 5 ✓	R4,0
R9,R3, 7	R9,R3, 7	R9,R4,5

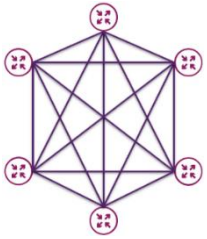
12 Network Design

Requirements an ein Netzwerk: Security, Availability, Scalability, Usability, Manageability, Performance, Adaptability, Price

Durch **Redundanz** können Verfügbarkeit und Kapazität erhöht werden, jedoch wird auch die Komplexität erhöht.

12.1 Topologien





← Full-Mesh

Building Blocks sind fertige Bauteile, die zu einem gut designten Netzwerk zusammengefügt werden können. **Vorteile**: Modularity, Grow, Failure Isolation, Standardization

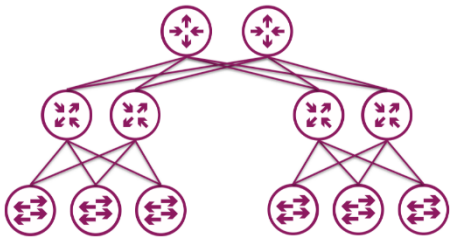
Beispiel: Cisco Validated Designs

- **Star**: Standard im Büro, viele Kabel, hohe Kosten, keine Redundanz, sehr flexibel, ungeeignet für Ortschaften (wegen Preis & Anzahl Kabel)
- **Ring**: gut geeignet für Ortschaften, wenig Kabel, günstig, unflexibel, weil die Geräte entlang des Rings angeordnet werden müssen und Geräte-Änderungen Unterbrüche verursachen

12.2 Enterprise Campus

Beim Design eines Enterprise Netzwerks sollten die Requirements im Kopf behalten werden, ein passender Vendor ausgewählt werden, die passenden Technologien ausgewählt werden und für die Zukunft geplant werden.

Hierarchie:



- **Flat**: In LANs/VLANs mit Clients wo jeder mit jedem kommunizieren kann
- **Hierarchisch**: Rest des Campus Designs mit Access/Distribution/Core und auch beim Addressing, Subnet-Summarization pro Gebäude, Areas pro Gebäude

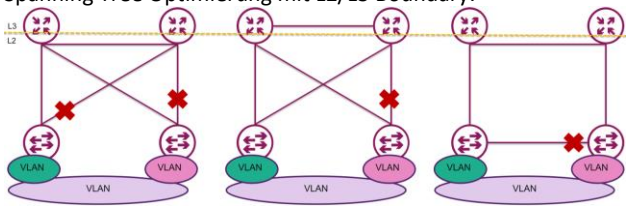
Prioritäten **Core/Backbone**: Scalability, Capacity, Simpel aber schnell, Redundancy, hohe Bandbreite, Convergence in ms, HA, keine Security/Intelligence

Prioritäten **Distribution**: First Hop Redundancy, Availability, Load Balancing, Routing Optimization, Security QoS Enforcement, evtl. L3 Boundary, Default Gateway (HSRP, VRRP, GLBP), OSPF Totally Stubby Area, L2 Termination, Access Lists, Fast Convergence für STP, Route Summarization

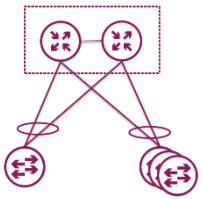
Prioritäten **Access**: High port density, PoE, Network Access control (z.B. 802.1X), QoS Klassifizierung, L2 Features: STP, IGMP Snooping, DHCP Snooping, ICMPv6 RA Guard, usw., evtl. L3 Boundary, STP Root Port, VLAN, 802.1Q Trunk

Beim **Collapsed Core Design** sind Core und Distribution zusammengelegt. Es ist dann eine Two-Tier Hierarchie.

Spanning Tree Optimierung mit L2/L3 Boundary:



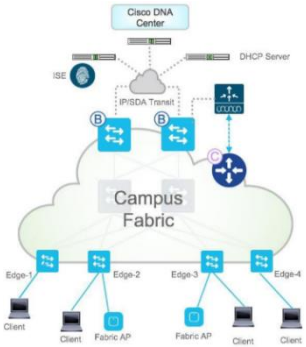
Access Switches können auch virtuell zu einem einzigen Switch zusammengefügt werden. Das simplified zwar die Topologie, ist aber technisch sehr komplex.



Layer 2 hat einige nützliche Features: STP, FHRP, Rapid STP, MST, LoopGuard, RootGuard, BPDU Guard, PortFast, ARP Inspection, DHCP Snooping, RA Guard

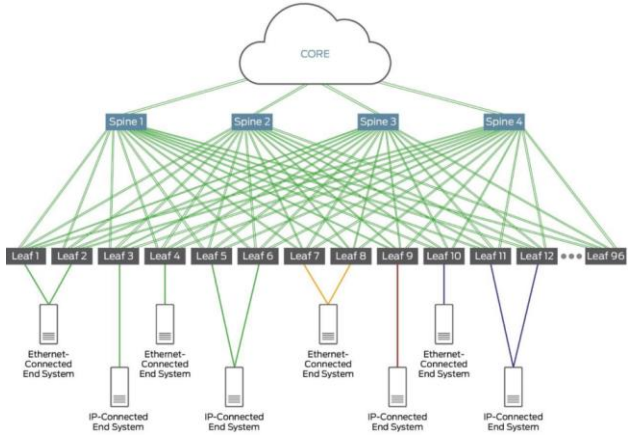
Jedoch skaliert L2 sehr schlecht und hat auch andere Probleme, welche nur teilweise behoben sind: Security, STP, Fault Isolation, Looping

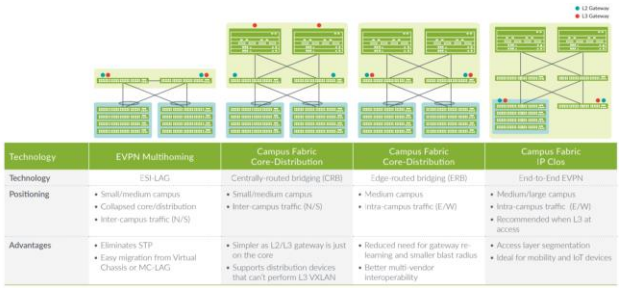
Daher sollte bereits auf dem Access Layer L3 eingesetzt werden, man hat kein STP und FHRP, Load Balancing, Convergence, simplere Konfiguration und mehr Möglichkeiten zur Segmentierung.



Bei einer Fabric oder Overlay mit VxLAN oder LISP kann man L2 und L3 trennen. Ebenfalls geht das mit einem EVPN in der Leaf-Spine Topologie. Auch mit MPLS können L2 und L3 unabhängig über ein Netzwerk geroutet werden.

Um die L2 Domain zu verkleinern und trotzdem über mehrere Switches zu verteilen können Overlays wie EVPN, LISP oder MPLS eingesetzt werden.





Technology	EVPN Multihoming	Campus Fabric Core-Distribution	Campus Fabric Core-Distribution	Campus Fabric IP Clos
Technology	ESI-LAG	Centrally-routed bridging (CRB)	Edge-routed bridging (ERB)	End-to-End EVPN
Positioning	• Small/medium campus • Collapsed core/distribution • Inter-campus traffic (N/S)	• Small/medium campus • Inter-campus traffic (N/S)	• Medium campus • Intra-campus traffic (E/W)	• Medium/large campus • Intra-campus traffic (E/W) • Recommended when L3 at access
Advantages	• Eliminates STP • Easy migration from Virtual Chassis or MC-LAG	• Simpler as L2/L3 gateway is just on the core • Supports distribution devices that can't perform L3 VXLAN	• Reduced need for gateway re-learning and smaller blast radius • Better multi-vendor interoperability	• Access layer segmentation • Ideal for mobility and IoT devices

12.3 Data Center

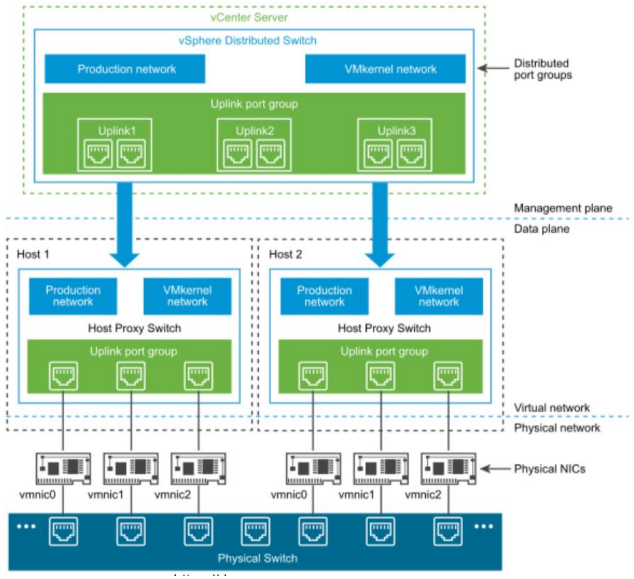
North-South Traffic: Ein- und Ausgehende Kommunikation vom Data Center
East-West Traffic: Kommunikation zwischen Servern im Data Center

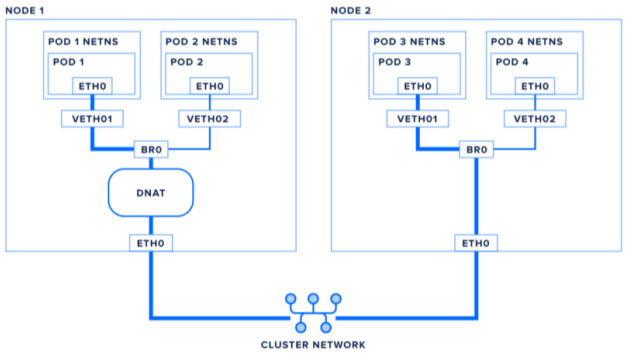
Prioritäten: Requirements im Kopf behalten, verschiedene Topologien, IP-Netzwerk, Storage Netzwerk, Building Blocks für Core, Aggregation und Access anwenden, Scalability, Resilience, Agility, Isolation, Multitenancy

Beliebt ist eine **Leaf-Spine Topologie** mit Top of Rack (**ToR**) Switches, die redundant an **Leafs** angeschlossen sind, welche wiederum redundant an den **Spine** angeschlossen sind.

Vorteile ToR gegenüber End of Rack (EoR): weniger Kabel, einfache Erweiterungen und Änderungen an den Racks, skalierbare Fiber Infrastruktur, geeignet für volle Racks

Vorteile EoR gegenüber ToR: weniger Switches benötigt, höhere Port-Auslastung der Switches, Switches sind alle an einem Ort, einfachere L2-Kommunikation zwischen Racks





VRRP: zwei Router haben gleiche IP aber andere MACs, ARP-Cache auf Hosts muss renewed werden, langsam

HSRP: zwei Router haben gleiche IP/MAC

GLBP: verschiedene MAC-Antworten auf ARP Requests, Load Balancing