

## LISP – Locator ID Separation Protocol

Technologie die **Routing** und **Overlay Tunneling** miteinander kombiniert.

### Tunneling

LISP Router sind über eine beliebige geroutete Infrastruktur miteinander verbunden, z.B.

- externe Verbindungsnetze
- entfernte WAN Strukturen
- Internet

Über diese Topologie baut LISP ein **fully-meshed overlay VN** auf, d.h. alle LISP Router sind über **getunnelte** Verbindungen direkt miteinander verbunden.

Datenverkehr, der zwischen den "LISP-sites" oder "zwischen "LISP-sites" und "non-LISP sites" übertragen werden soll,

wird beim Eingang in das LISP VN (overlay network) encapsuliert, d.h. dem eigentlichen Paket wird ein LISP Header und einem neuer IP Header vorangestellt.

Beim Ausgang aus dem Overlay VN wird das Paket dann wieder de-encapsuliert.

### Original Paket

zwei Geräte kommunizieren über das LISP overlay VN

IP Beispiel	ICMP
172.23.1.1 192.168.11.1	"echo-request"

### LISP Paket

LISP schreibt LISP, UDP und neuen IP Header vor das originale IP Paket

- **LISP** Header mit Flag Informationen
- **UDP** Header: Destination Port 4341
- **IP** Header: die IP/IPv6 Adressen der LISP Router (Konnektivität über das eigentliche "underlay" Netzwerk).

Die notwendigen DST IP/IPv6 Adressen für die neuen IP/IPv6 Header

(Tunnelendadressen der LISP Router) werden von LISP **Map Servern** bereitgestellt.

"new" IP Beispiel	UDP	LISP	"original" IP	ICMP
33.0.0.1 66.0.0.1	unpriv 4341	Flags	172.23.1.1 192.168.11.1	"echo-request"

### Wireshark – Mitschnitt

No.	Time	Source	Destination	Protocol	Length	Info
→ 11	25.464631	172.23.1.1	192.168.11.1	ICMP	150	Echo (ping) request

```

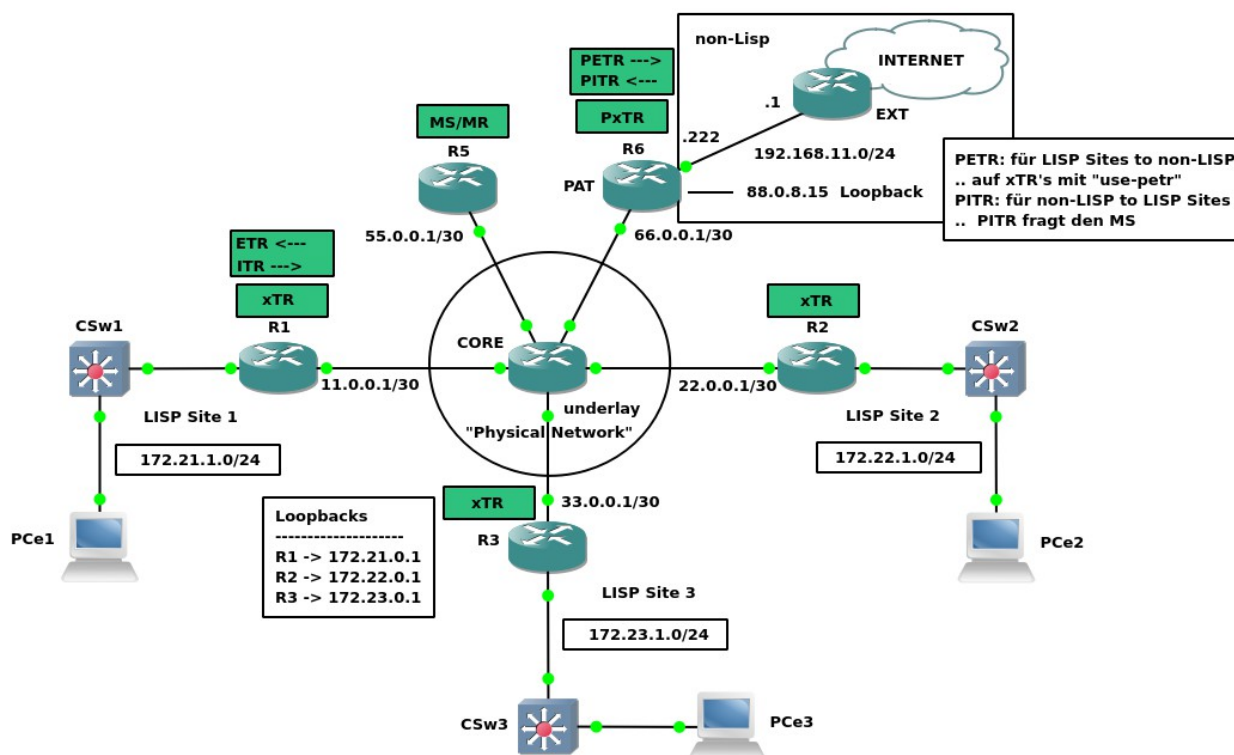
> Frame 11: 150 bytes on wire (1200 bits), 150 bytes captured (1200 bits) on interface -, id 0
> Ethernet II, Src: ca:03:27:cc:00:a8 (ca:03:27:cc:00:a8), Dst: ca:06:31:d8:00:00 (ca:06:31:d8:00:00)
> Internet Protocol Version 4, Src: 33.0.0.1, Dst: 66.0.0.1
> User Datagram Protocol, Src Port: 26559, Dst Port: 4341
> Locator/ID Separation Protocol (Data)
> Internet Protocol Version 4, Src: 172.23.1.1, Dst: 192.168.11.1
> Internet Control Message Protocol

```

## LISP Geräte und Bezeichnungen

Bezeichnung/Geräte	Beschreibung
<b>LISP site</b>	Netzwerk hinter einem LISP Router LISP Router verbinden LISP sites miteinander
<b>non-LISP site</b>	nicht an die LISP Topologie angeschlossene – externe Netze (z.B. Internet, WAN, ..) für die keine EIDs (aber Net-IDs bzw. Routing Targets) existieren
<b>EID</b> Endpoint identifier	IPv4/IPv6 Adressen von Geräten innerhalb einer LISP site
<b>RLOC</b> Routing locator	IPv4/IPv6 Adressen der LISP Router, über die das getunnelte fully-meshed LISP VN aufgebaut wird Tunnelendadressen der LISP Router
<b>ITR</b> Ingress tunnel router	<b>Encapsuliert</b> Datenverkehr <b>aus einer LISP site</b> , der an eine andere LISP Site oder an eine non-LISP Site adressiert ist → sendet traffic in das overlay network
<b>ETR</b> Egress tunnel router	<b>De-encapsuliert</b> Datenverkehr <b>in eine LISP site</b> , der aus einer andern LISP Site oder einer non-LISP site stammt → sendet traffic aus dem overlay network
<b>xTR</b> ITR/ETR tunnel router	Router der ingress und egress Funktion verbindet → meist verwendet
<b>PITR</b> Proxy ITR	<b>Encapsuliert</b> Datenverkehr <b>aus einer non-LISP site</b> , der an eine andere LISP Site oder an eine externe non-LISP Site adressiert ist → sendet traffic in das overlay network
<b>PETR</b> Proxy ETR	<b>De-encapsuliert</b> Datenverkehr <b>in eine non-LISP site</b> , der aus einer andern LISP Site oder einer non-LISP site stammt → sendet traffic aus dem overlay network
<b>PxTR</b> PITR/PETR tunnel router	Router der proxy ingress und egress Funktion verbindet → meist verwendet
<b>MS</b> Map Server	<b>Lernt EID-to-RLOC Mappings</b> (Zuordnungen) von ETR oder PETR Routern → Teilen dem MS mit, welche EIDs oder externen Ziele (Routing Ziele) über ihren RLOC (eigene Tunnelendpunktadresse) erreicht werden können
<b>MR</b> Map Resolver	<b>Empfängt Anfragen zu RLOCs von ETRs</b> von ITR oder PITR Routern .. → welcher RLOC (remote tunnelendpunkt next-hop IP) muss verwendet werden um eine bestimmte Ziel IP Adresse (EID oder externe IP) zu erreichen .. und <b>sendet diese Anfragen an den Map Server weiter</b> (nach de-encapsulation)
<b>MR/MS</b> Map Server/Resolver	Router der als Map Server und Resolver Funktion verbindet → meist verwendet

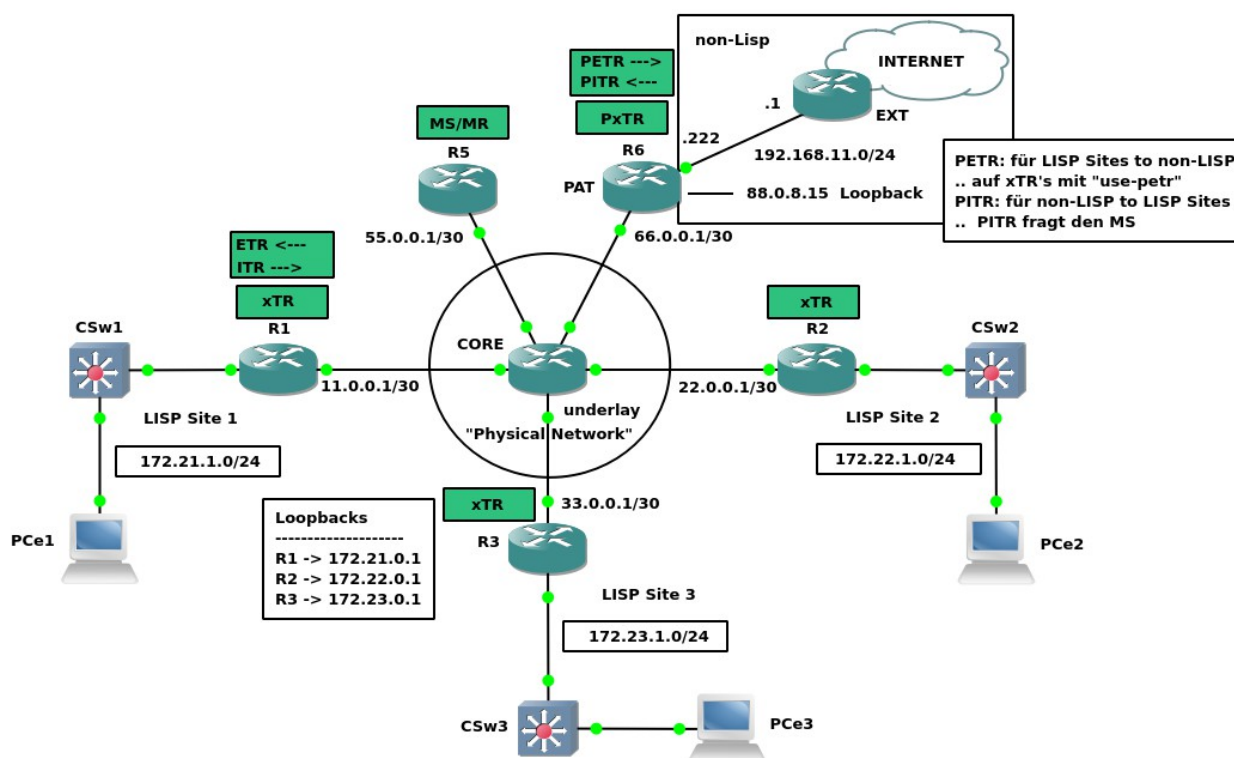
## LISP Generelle Arbeitsweise (lisp-site to lisp-site)



### Kommunikation PCe1 (lisp-site) zu PCe2 (lisp-site)

1. PCe1 erzeugt ein IP Paket um PCe2 zu erreichen  
SRC: 172.21.1.1  
DST: 172.22.1.1
2. R1 (**ITR** Funktion)
  - Empfängt das IP Paket von PCe1
  - Sendet **Anfrage an MR** (55.0.0.1)  
→ "ich benötige die RLOC Adresse für die DST EID 172.22.1.1"
  - Erhält **Anwort vom MS** (ebenfalls 55.0.0.1) .. der sie vom ETR (R2) gelernt hat.  
→ "RLOC für EID 172.22.1.1 ist 22.0.0.1 (Tunnelendadresse von R2)"
  - **Encapsuliert** den Datenverkehr mit LISP (Tunneling), trägt die ermittelt RLOC Adresse als DST IP in den neuen IP Header ein (Routing) und sendet das "getunnelte Paket" physikalisch über das underlay network Richtung Core Router  
SRC: 11.0.0.0  
DST: 22.0.0.1
3. R2 (**ETR** Funktion)
  - Empfängt LISP Paket von RLOC 11.0.0.1
  - **De-encapsuliert** das Paket ..
  - .. und sendet es gemäß DST IP (172.22.1.1) an PCe2 weiter.

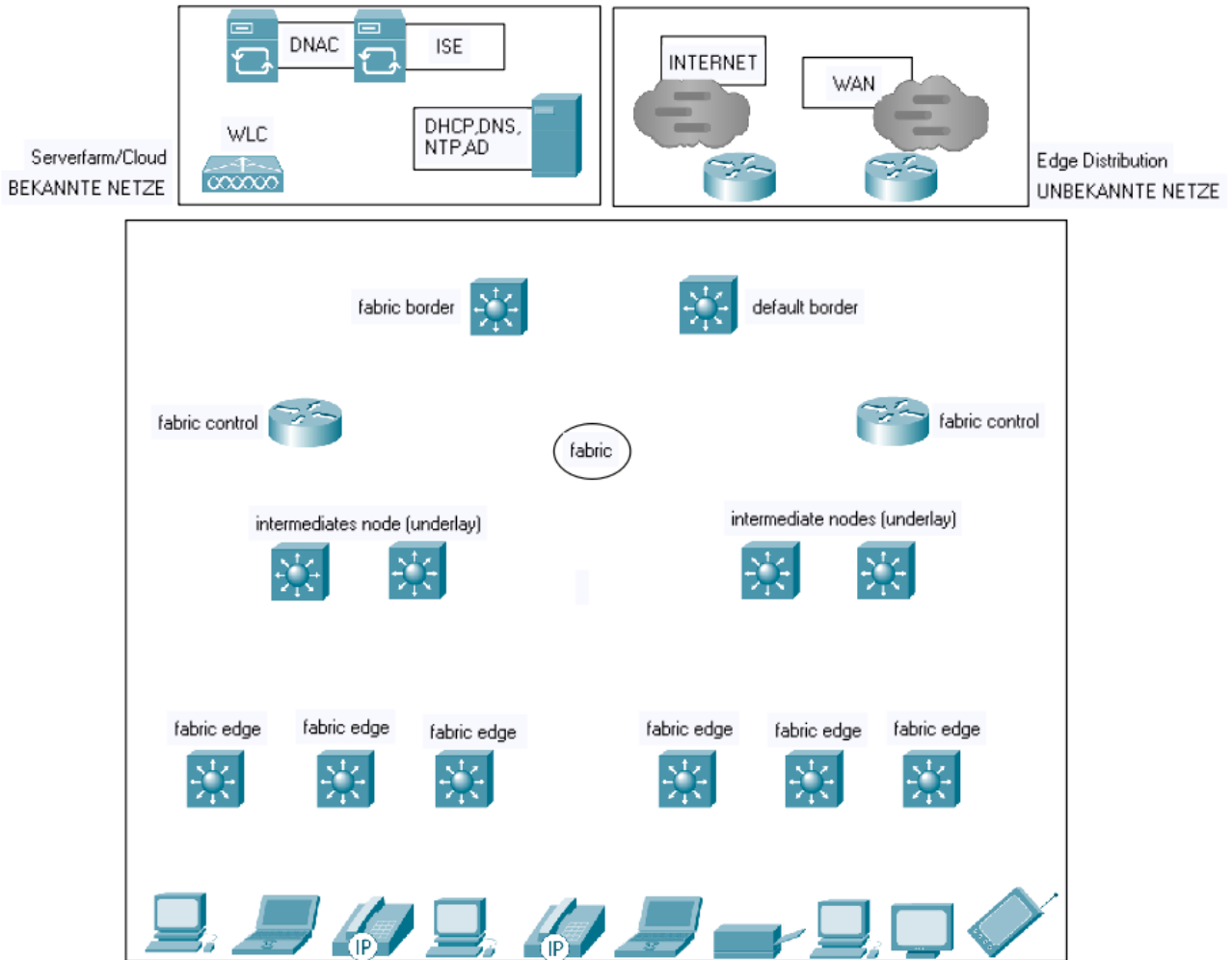
## LISP Generelle Arbeitsweise (lisp-site to non-lisp-site)



Kommunikation PCe1 (lisp-site) zu einer beliebigen IP ins Internet (non-lisp-site)

- PCe1 erzeugt ein IP Paket um z.B. die 9.9.9.9 im Internet zu erreichen  
SRC: 172.21.1.1  
DST: 9.9.9.9
- R1 (**ITR** Funktion)
  - Empfängt das IP Paket von PCe1
  - Sendet Anfrage an MR** (55.0.0.1)  
→ "ich benötige die RLOC Adresse für die DST EID 9.9.9.9"
  - Erhält Antwort vom MS** (ebenfalls 55.0.0.1) .. der sie vom Proxy ETR (R6) – in diesem Fall – als Routing Target 0.0.0.0/0 gelernt hat.  
→ "RLOC für EID 9.9.9.9 ist 66.0.0.1 (Tunnelendadresse von R6 – Proxy ETR)"
  - Encapsuliert** den Datenverkehr mit LISP (Tunneling), trägt die ermittelte RLOC Adresse als DST IP in den neuen IP Header ein (Routing) und sendet das "getunnelte Paket" physikalisch über das underlay network Richtung Core Router  
SRC: 11.0.0.1  
DST: 66.0.0.1
- R2 (**PETER** Funktion)
  - Empfängt LISP Paket von RLOC 11.0.0.1
  - De-encapsuliert** das Paket ..
  - .. und sendet es gemäß DST IP (9.9.9.9) via seiner Default Route (0.0.0.0/0) den Next-Hop Router (EXT) Richtung Internet weiter.

### LISP SDA Rollen



LISP Router	Rolle in Softward-defined Access Netzwerken
<b>LISP</b>	<b>fabric overlay control-plane</b>
LISP <b>MS/MR Router</b>	<b>control-plane node</b> (fabric control)
LISP <b>PxTR Router</b>	<b>border node</b> (fabric border)
LISP <b>xTR Router</b>	<b>edge node</b> (fabric edge)
<b>non-LISP Router</b>	<b>intermediate node</b> (underlay)

## LISP IPv4 Konfiguration und Troubleshooting

→ geschieht in einer SDA fabric vollautomatisch über den DNAC.

### MS/MR

LISP aktivieren – **router lisp**

- .. und Einstellung für die VRF Zuordnung der **locator-table** vornehmen ("default" verwenden, wenn keine VRFs existieren)
- Router als **Map Server/Map Resolver Router** aktivieren
- Definition von **LISP Sites** inklusive
  - Beschreibung,
  - Authentication mit Passwort (PSK) und
  - erlaubten prefixes

```
!
router lisp
locator-table { default | vrf vrf-name }
!
ipv4 map-server
ipv4 map-resolver
!
site lisp-site-name
description Beschreibung
authentication-key PSK
eid-prefix net-ID/prefix-length
exit
!
```

Troubleshooting

```
# show lisp site
# show ip lisp
# show ipv6 lisp
# debug lisp control-plane all
```

**xTR**LISP aktivieren – **router lisp**

- **locator-table** → RLOC Space  
über welche VRF erreichen sich die LISP Router  
→ "default" verwenden, wenn keine VRFs existieren
- Optional: **locator-set** → Lokale RLOC, Priorität und Gewichtung bei der Lastverteilung sinnvoll falls redundante ETR Router für die LISP site existieren – priority und weight  
→ das locator-set kann in der eid-table database-mapping Konfiguration als Referenz verwendet werden
- **eid-table** → EID Space  
→ für die "default" VRF muss als **Instance ID 0** verwendet werden  
→ pro VRF eine eigene **Instance ID** notwendig (LISP Virtualization)

..mit **database-mapping** werden EID-to-RLOC mapping Beziehungen festgelegt

- Optional: mit **loc-reach-algorithm rloc-probing** tauschen LISP Router Nachrichten untereinander aus, um die Erreichbarkeit der RLOC IP's zu verifizieren (sinnvoll für Troubleshooting)
- **ITR** Funktion aktivieren und Adresse des **MR** festlegen  
→ und Adresse des **Proxy ETR** festlegen
- **ETR** Funktion aktivieren und Adresse des **MS** festlegen

```
!
router lisp
  locator-table { default | vrf vrf-name }
!
  locator-set lisp-site-name
    RLOC-IP|IPv6 priority wert weight wert
  exit
!
  eid-table { default instance-id 0 | vrf vrf-name instance-id ID }
  database-mapping net-ID/prefix-length RLOC-IP|IPv6 priority wert weight wert
! .. oder Referenz auf das locator-set
  database-mapping net-ID/prefix-length locator-set lisp-site-name
  exit
!
  loc-reach-algorithm rloc-probing
!
  { ipv4 | ipv6 } itr
  { ipv4 | ipv6 } itr map-resolver IP|IPv6-map-resolver
  { ipv4 | ipv6 } use-petr IP|IPv6-proxy-etr
  { ipv4 | ipv6 } etr
  { ipv4 | ipv6 } etr map-server IP|IPv6-map-server key PSK
!
```

Virtuelles LISP Tunnel Interface anlegen (optional) – **interface lisp**

- wird für xTR und PxTR automatisch erzeugt .. (default LISP0)
- .. pro Instance ID (VRF) ein eigenes separates LISP SubIF (z.B. LISP0.42 für die Instance ID 42)
- optional die RLOC Adresse bestimmen (Tunnelendpunkt SRC IP Adresse), die der Router in RLOC-to-EID Mappings an den MS senden sollen. ACHTUNG: im äusseren LISP Header wird nichtdestotrotz die SRC IP der ausgehenden Schnittstelle verwendet.

```
!  
interface lisp IF-nummer  
  [ { ipv4 | ipv6 } lisp source-locator IF-ID ]  
!
```

Troubleshooting

```
# show { ipv4 | ipv6 } lisp  
# show { ipv4 | ipv6 } lisp map-cache [ instance-id ID ]  
# debug lisp control-plane all
```



## PxTR

LISP aktivieren – **router lisp**

- **locator-table** → RLOC Space  
über welche VRF erreichen sich die LISP Router  
→ "default" verwenden, wenn keine VRFs existieren
- **eid-table** → EID Space  
mit **map-cache** oder **ipv4 route import** wird angegeben, welcher Datenverkehr zu Zielen innerhalb von LISP führen ..  
.. und somit für welche DST IP Adressen er Anfragen an den Map Resolver senden und letztendliche über das LISP Overlay Netzwerk zustellen soll.
- Optional: mit **loc-reach-algorithm rloc-probing** tauschen LISP Router Nachrichten untereinander aus, um die Erreichbarkeit der RLOC IP's zu verifizieren (sinnvoll für Troubleshooting)
- Optional: Angabe der zu verwenden Source IP – **map-request-source** – zur Kommunikation mit dem Map Resolver - muss im Adressraum der eid-table sein.
- Aktivierung als **Proxy-ETR**
- Aktivierung als **Proxy-ITR** und Angabe der MS IP für die ITR Funktion

```
!
router lisp
  locator-table { default | vrf vrf-name }
!
  eid-table { default instance-id 0 | vrf vrf-name instance-id ID }
  map-cache 172.16.0.0/12 map-request
  .. oder
  ipv4 route-import map-cache static route-map route-map-name
  exit
!
  loc-reach-algorithm rloc-probing
!
  { ipv4 | ipv6 } map-request-source src-IP|IPv6
!
  { ipv4 | ipv6 } proxy-etr
  { ipv4 | ipv6 } proxy-itr RLOC-IP|IPv6
  { ipv4 | ipv6 } itr map-resolver IP|IPv6-map-resolver
  exit
!
```

Virtuelles LISP Tunnel Interface anlegen (optional) – **interface lisp**

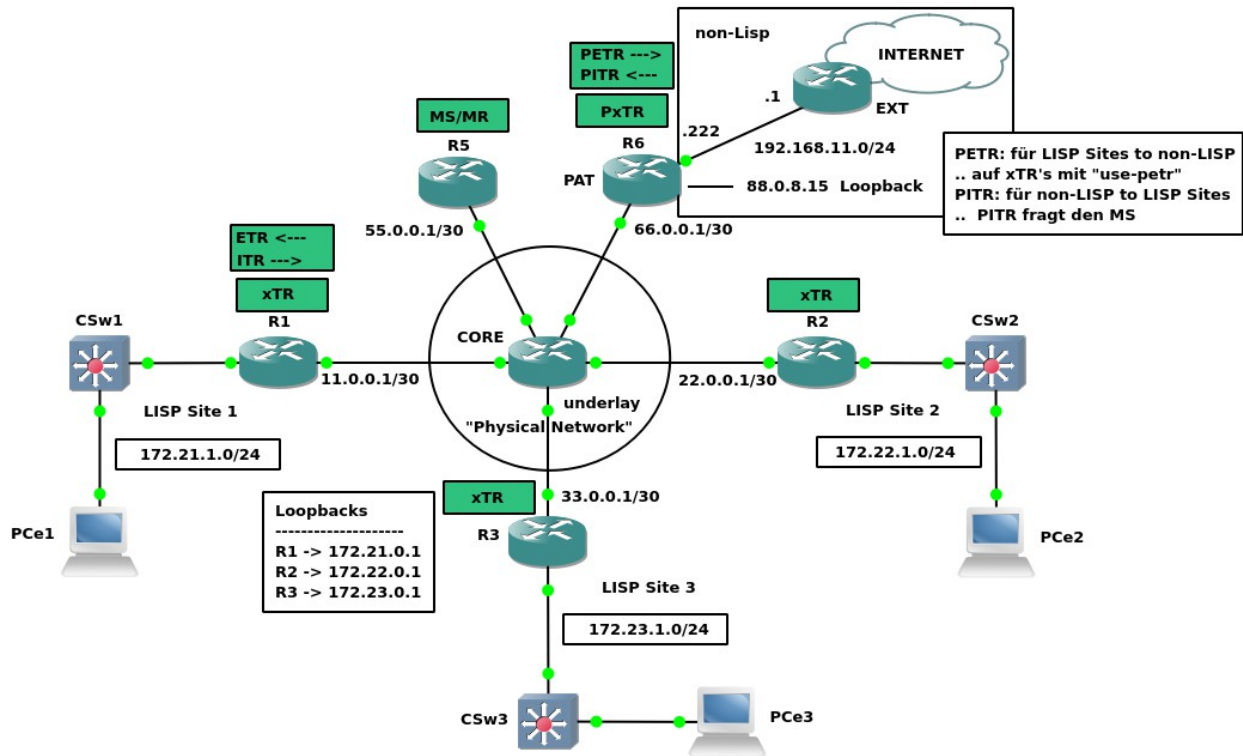
- wird für xTR und PxTR automatisch erzeugt .. (default LISP0)
- .. pro Instance ID (VRF) ein eigenes separates LISP SubIF (z.B. LISP0.42 für die Instance ID 42)
- optional die RLOC Adresse bestimmen (Tunnelendpunkt SRC IP Adresse), die der Router in RLOC-to-EID Mappings an den MS senden sollen. ACHTUNG: im äusseren LISP Header wird nichtdestotrotz die SRC IP der ausgehenden Schnittstelle verwendet.

```
!  
interface lisp IF-nummer  
  [ { ipv4 | ipv6 } lisp source-locator IF-ID ]  
!
```

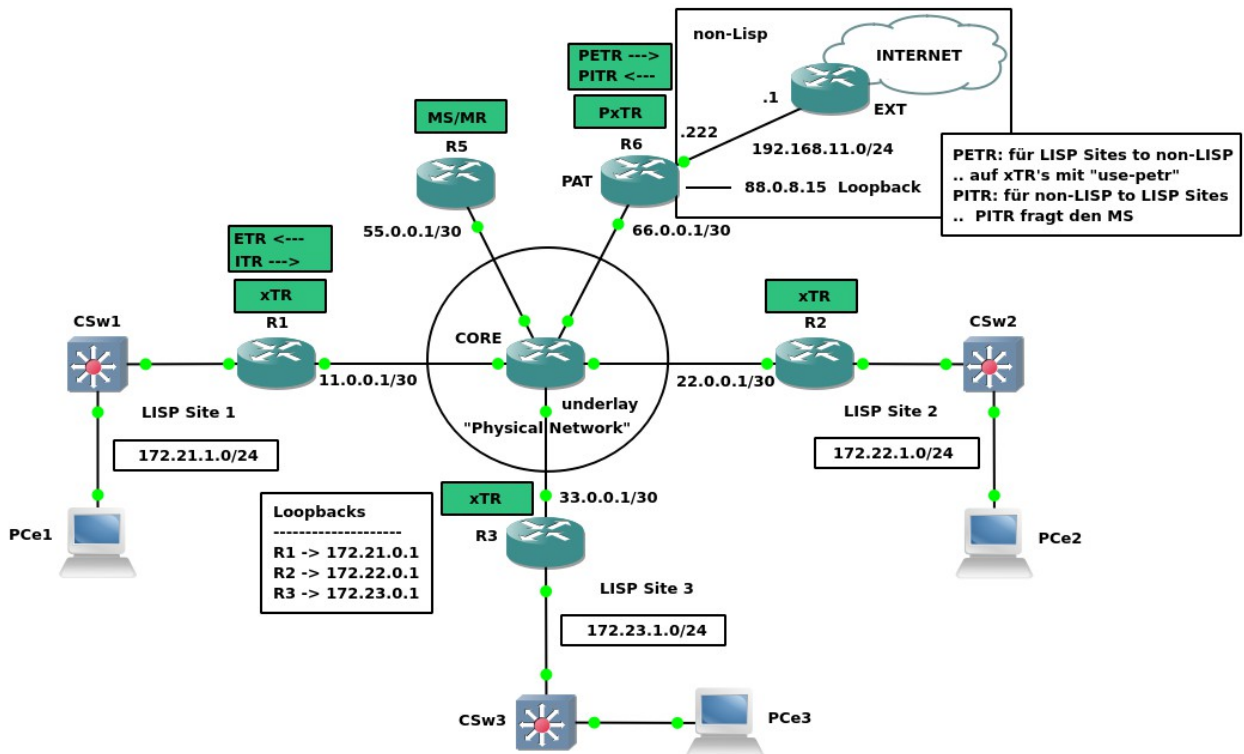
Troubleshooting

```
# show { ipv4 | ipv6 } lisp  
# show { ipv4 | ipv6 } lisp map-cache [ instance-id ID ]  
# debug lisp control-plane all
```

## Konfigurationsbeispiel

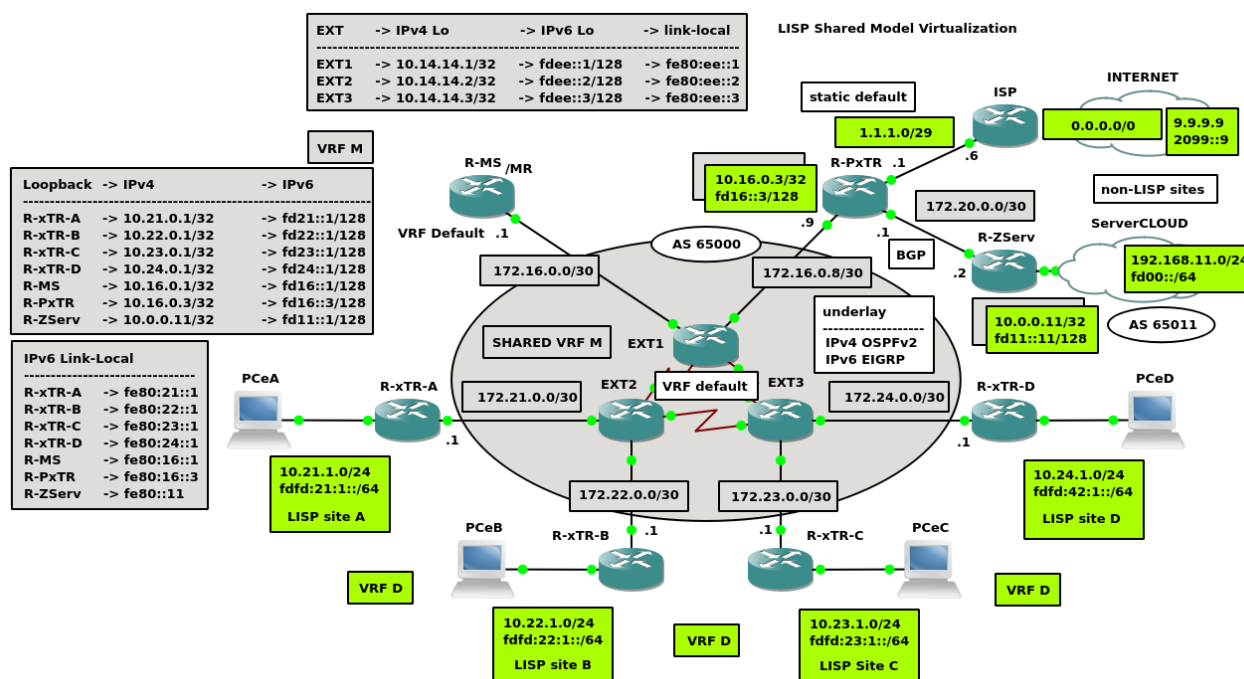


R5 → MS/MR	R6 → PxTR
<pre>! router lisp locator-table default ! site site1 description LISP Site 1 authentication-key SITE1KEY eid-prefix 172.21.0.0/24 eid-prefix 172.21.1.0/24 exit ! site site2 description LISP Site 2 authentication-key SITE2KEY eid-prefix 172.22.0.0/24 eid-prefix 172.22.1.0/24 exit ! site site3 description LISP Site 3 authentication-key SITE3KEY eid-prefix 172.23.0.0/24 eid-prefix 172.23.1.0/24 exit ! ipv4 map-server ipv4 map-resolver !</pre>	<pre>! router lisp eid-table default instance-id 0 map-cache 172.21.0.0/24 map-request map-cache 172.21.1.0/24 map-request map-cache 172.22.0.0/24 map-request map-cache 172.22.1.0/24 map-request map-cache 172.23.0.0/24 map-request map-cache 172.23.1.0/24 map-request exit ! loc-reach-algorithm rloc-probing ipv4 map-request-source 66.0.0.1 ipv4 map-cache-limit 100000 ipv4 proxy-etr ipv4 proxy-itr 66.0.0.1 ipv4 itr map-resolver 55.0.0.1 exit !</pre>



R1 → xTR	R2 → xTR
<pre> ! router lisp locator-set SITE1  11.0.0.1 priority 1 weight 1 exit ! eid-table default instance-id 0 database-mapping 172.21.0.0/24 locator-set SITE1 database-mapping 172.21.1.0/24 locator-set SITE1 exit ! loc-reach-algorithm rloc-probing ipv4 itr map-resolver 55.0.0.1 ipv4 itr ipv4 etr map-server 55.0.0.1 key SITE1KEY ipv4 etr ipv4 use-petr 66.0.0.1 exit !                     </pre>	<pre> ! router lisp ! eid-table default instance-id 0 database-mapping 172.22.0.0/24 22.0.0.1 priority 1 weight 1 database-mapping 172.22.1.0/24 22.0.0.1 priority 1 weight 1 exit ! loc-reach-algorithm rloc-probing ipv4 itr map-resolver 55.0.0.1 ipv4 itr ipv4 etr map-server 55.0.0.1 key SITE2KEY ipv4 etr ipv4 use-petr 66.0.0.1 exit !                     </pre>

## Konfigurationsbeispiel LISP Shared Model Virtualization



### EID/RLOC Space

- **EID Space**  
einem VRF wird mit einer Instance ID einem EID Space zugeordnet .. hier D
- **RLOC Space**  
der RLOC Space ist einem anderen VRF zugeordnet .. hier M  
Dabei müssen die physikalischen Adressen innerhalb dieser shared VRF erreicht werden können (hier die 172.er Adressen)

Die LISP Instance ID segmentiert LISP Control Plane von der Data Plane

### Virtualisierung im Beispiel

#### UNDERLAY

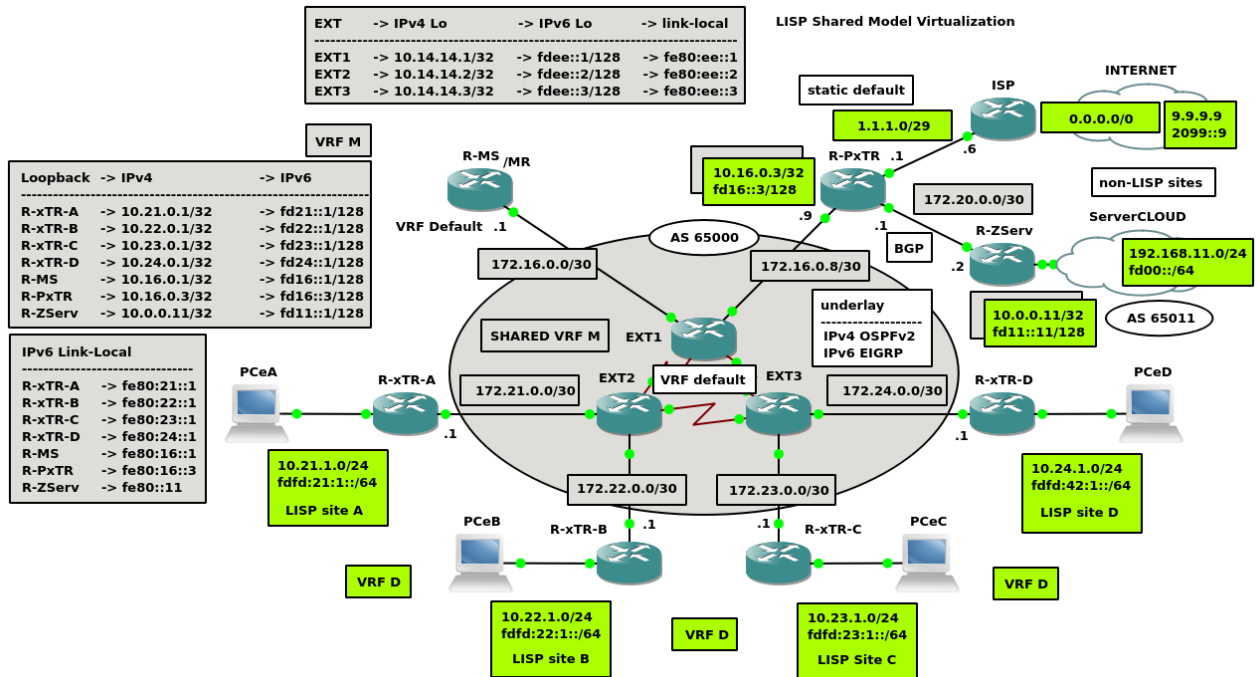
- VRF M only (kein VRF auf EXT Routern, MS und MR)
- dual-stack OSPFv3, IPv6 Named EIGRP Routing

#### LISP OVERLAY

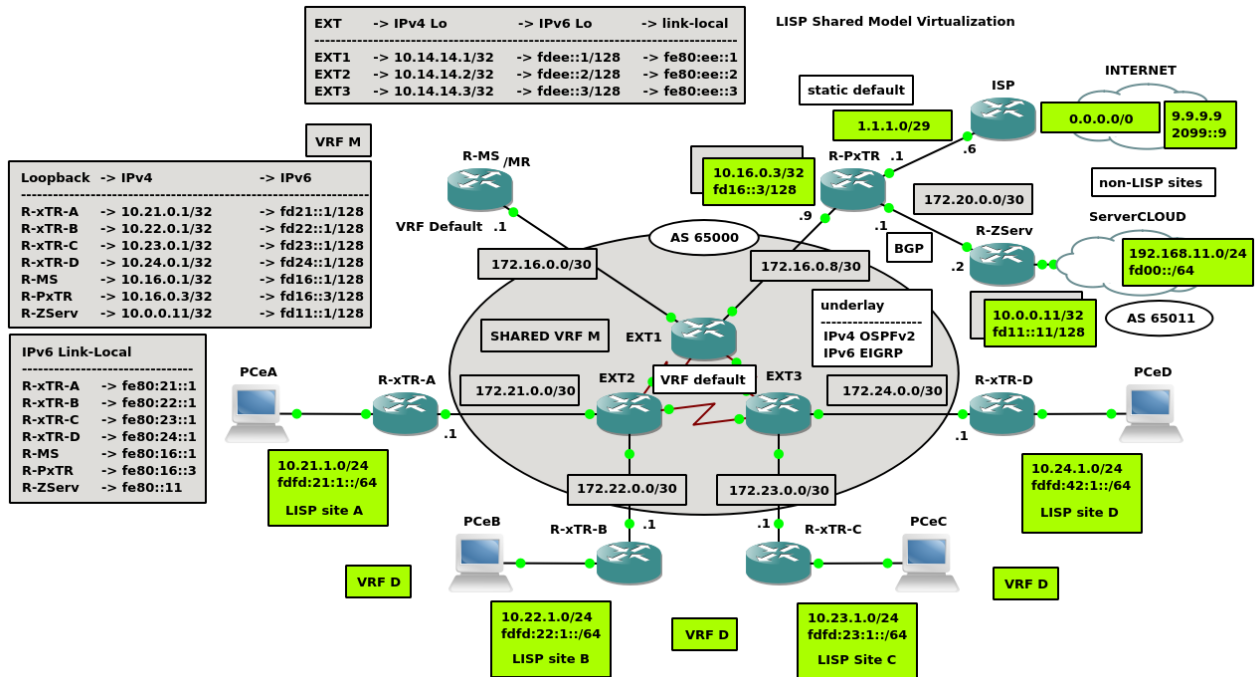
- LISP Sites → VRF D .. hinter xTR Routern
- Shared VRF → VRF M bzw. VRF default auf MS und MR (keine VRFs)
- non-LISP Sites → INTERNET (VRD D), CLOUD (VRF C) .. hinter PxTR

Traffic von/zu LISP Sites (VRF D) hinter den xTR Routern ..  
.. wird über getunneltes "shared" Network (VRF M) übertragen.

Generell sind natürlich multiple VRFs hinter jedem xTR Router, mit unterschiedlichsten Verbindungen möglich.



R-MS	R-PxTR
<pre> ! router lisp site A description -&gt; PCeA authentication-key cisco eid-prefix instance-id 42 10.21.0.0/16 accept-more-specifics eid-prefix instance-id 42 fd0e:21::/32 accept-more-specifics exit site B description -&gt; PCeB authentication-key cisco eid-prefix instance-id 42 10.22.0.0/16 accept-more-specifics eid-prefix instance-id 42 fd0e:22::/32 accept-more-specifics exit site C description -&gt; PCeC authentication-key cisco eid-prefix instance-id 42 10.23.0.0/16 accept-more-specifics eid-prefix instance-id 42 fd0e:23::/32 accept-more-specifics exit site D description -&gt; PCeD authentication-key cisco eid-prefix instance-id 42 10.24.0.0/16 accept-more-specifics eid-prefix instance-id 42 fd0e:24::/32 accept-more-specifics exit ! ipv4 map-server ipv6 map-server ipv4 map-resolver ipv6 map-resolver !                     </pre>	<pre> ! router lisp locator-table vrf M ! eid-table vrf D instance-id 42 map-cache 10.21.0.0/16 map-request map-cache 10.22.0.0/16 map-request map-cache 10.23.0.0/16 map-request map-cache 10.24.0.0/16 map-request map-cache fd0e:21::/32 map-request map-cache fd0e:22::/32 map-request map-cache fd0e:23::/32 map-request map-cache fd0e:24::/32 map-request exit ! loc-reach-algorithm rloc-probing ! ipv4 proxy-etr ipv4 proxy-itr 10.16.0.3 ipv4 itr map-resolver 10.16.0.1 ipv6 proxy-etr ipv6 proxy-itr fd16::3 ipv6 itr map-resolver fd16::1 !                     </pre>



R-xTR-A	R-xTR-D
<pre> ! router lisp  locator-table vrf M ! eid-table vrf D instance-id 42  database-mapping 10.21.1.0/24 10.21.0.1 priority 1  weight 1   database-mapping fdfd:21:1::/64 fd21::1 priority 1  weight 1  exit ! loc-reach-algorithm rloc-probing ! ipv4 etr ipv4 etr map-server 10.16.0.1 key cisco ipv4 itr ipv4 itr map-resolver 10.16.0.1 ipv4 use-petr 10.16.0.3 ipv6 etr ipv6 etr map-server fd16::1 key cisco ipv6 itr ipv6 itr map-resolver fd16::1 ipv6 use-petr fd16::3 ! </pre>	<pre> ! router lisp  locator-table vrf M ! eid-table vrf D instance-id 42  database-mapping 10.24.1.0/24 10.24.0.1 priority 1  weight 1   database-mapping fdfd:24:1::/64 fd24::1 priority 1  weight 1  exit ! loc-reach-algorithm rloc-probing ! ipv4 etr ipv4 etr map-server 10.16.0.1 key cisco ipv4 itr ipv4 itr map-resolver 10.16.0.1 ipv4 use-petr 10.16.0.3 ipv6 etr ipv6 etr map-server fd16::1 key cisco ipv6 itr ipv6 itr map-resolver fd16::1 ipv6 use-petr fd16::3 ! </pre>