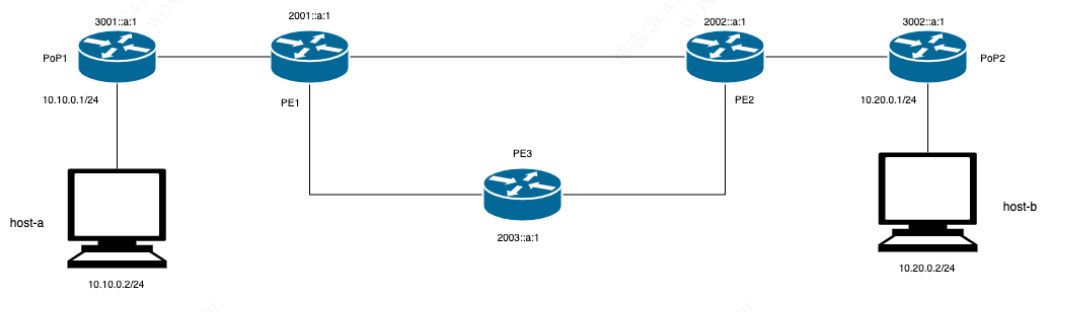
测试网络拓扑图



通过Linux命名空间模拟不同节点，步骤如下：

**网络拓扑初始化：**

创建命名空间模拟各个节点，并创建veth设备打通不同的命名空间

|  |
| --- |
| # 清除测试命名空间  ip netns |awk '{print $1}'|xargs -I {} ip netns delete {}    # 创建各个节点命名空间  ip netns add host-a  ip netns add host-b  ip netns add pop1  ip netns add pop2  ip netns add pe1  ip netns add pe2  ip netns add pe3    # 在命名空间内设置各个系统参数  ip netns list | awk '{print$1}' | xargs -I {} ip netns exec {} sysctl -w net.ipv4.ip\_forward=1  ip netns list | awk '{print$1}' | xargs -I {} ip netns exec {} sysctl -w net.ipv4.conf.all.rp\_filter=0  ip netns list | awk '{print$1}' | xargs -I {} ip netns exec {} sysctl -w net.ipv6.conf.all.forwarding=1  ip netns list | awk '{print$1}' | xargs -I {} ip netns exec {} sysctl -w net.ipv6.conf.all.seg6\_enabled=1  ip netns list | awk '{print$1}' | xargs -I {} ip netns exec {} sysctl -w net.ipv4.conf.default.rp\_filter=0  ip netns list | awk '{print$1}' | xargs -I {} ip netns exec {} sysctl -w net.ipv6.conf.default.forwarding=1  ip netns list | awk '{print$1}' | xargs -I {} ip netns exec {} sysctl -w net.ipv6.conf.default.seg6\_enabled=1  ip netns list | awk '{print$1}' | xargs -I {} ip netns exec {} sysctl -w net.ipv4.conf.lo.rp\_filter=0  ip netns list | awk '{print$1}' | xargs -I {} ip netns exec {} sysctl -w net.ipv6.conf.lo.forwarding=1  ip netns list | awk '{print$1}' | xargs -I {} ip netns exec {} sysctl -w net.ipv6.conf.lo.seg6\_enabled=1  ip netns list | awk '{print$1}' | xargs -I {} ip netns exec {} sysctl -w net.vrf.strict\_mode=1    # 创建veth设备，打通不同的节点命名空间  ip link add dev dev-a type veth peer name dev-pop1-a  ip link add dev dev-b type veth peer name dev-pop2-b  ip link add dev dev-pop1-pe1 type veth peer name dev-pe1-pop1  ip link add dev dev-pop2-pe2 type veth peer name dev-pe2-pop2  ip link add dev dev-pe1-pe2 type veth peer name dev-pe2-pe1  ip link add dev dev-pe2-pe3 type veth peer name dev-pe3-pe2  ip link add dev dev-pe3-pe1 type veth peer name dev-pe1-pe3  ip link set dev-a netns host-a  ip link set dev-b netns host-b  ip link set dev-pop1-pe1 netns pop1  ip link set dev-pop1-a netns pop1  ip link set dev-pop2-pe2 netns pop2  ip link set dev-pop2-b netns pop2  ip link set dev-pe1-pe2 netns pe1  ip link set dev-pe1-pe3 netns pe1  ip link set dev-pe1-pop1 netns pe1  ip link set dev-pe2-pe3 netns pe2  ip link set dev-pe2-pe1 netns pe2  ip link set dev-pe2-pop2 netns pe2  ip link set dev-pe3-pe1 netns pe3  ip link set dev-pe3-pe2 netns pe3    # 设置各个命名空间接口地址，路由表：  ip netns exec pop1 ip link set dev-pop1-a up  ip netns exec pop1 ip link set dev-pop1-pe1 up  ip netns exec pop1 ip link set lo up  ip netns exec pop1 ip addr add 3001::a:1/96 dev dev-pop1-pe1  ip netns exec pop1 ip addr add 10.10.0.1/24 dev dev-pop1-a  ip netns exec pop1 ip route add 2001::a:1/128 dev dev-pop1-pe1  ip netns exec pop1 ip route add 2001::/96 via 2001::a:1 dev dev-pop1-pe1  # pop1和pop2需要设置vrf表，用于END.DT4指令指定路由表  ip netns exec pop1 ip link add vrf-pop1-a type vrf table 101  ip netns exec pop1 ip link set dev dev-pop1-a master vrf-pop1-a  ip netns exec pop1 ip link set vrf-pop1-a up      ip netns exec pop2 ip link set dev-pop2-b up  ip netns exec pop2 ip link set dev-pop2-pe2 up  ip netns exec pop2 ip link set lo up  ip netns exec pop2 ip addr add 3002::a:1/96 dev dev-pop2-pe2  ip netns exec pop2 ip addr add 10.20.0.1/24 dev dev-pop2-b  ip netns exec pop2 ip route add 2002::a:1/128 dev dev-pop2-pe2  ip netns exec pop2 ip route add 2002::/96 via 2002::a:1  dev dev-pop2-pe2  ip netns exec pop2 ip link add vrf-pop2-b type vrf table 101  ip netns exec pop2 ip link set dev dev-pop2-b master vrf-pop2-b  ip netns exec pop2 ip link set vrf-pop2-b up      ip netns exec pe1 ip link set dev-pe1-pe2 up  ip netns exec pe1 ip link set dev-pe1-pe3 up  ip netns exec pe1 ip link set dev-pe1-pop1 up  ip netns exec pe1 ip link set lo up  ip netns exec pe1 ip addr add 2001::a:1/96 dev dev-pe1-pop1  ip netns exec pe1 ip addr add 2001::a:1/96 dev dev-pe1-pe2  ip netns exec pe1 ip addr add 2001::a:1/96 dev dev-pe1-pe3  ip netns exec pe1 ip route add 3001::a:1/128 dev dev-pe1-pop1  ip netns exec pe1 ip route add 2002::a:1/128 dev dev-pe1-pe2  ip netns exec pe1 ip route add 2003::a:1/128 dev dev-pe1-pe3  ip netns exec pe1 ip route add 3001::/96 via 3001::a:1 dev dev-pe1-pop1  ip netns exec pe1 ip route add 2002::/96 via 2002::a:1 dev dev-pe1-pe2  ip netns exec pe1 ip route add 2003::/96 via 2003::a:1  dev dev-pe1-pe3      ip netns exec pe2 ip link set dev-pe2-pe1 up  ip netns exec pe2 ip link set dev-pe2-pe3 up  ip netns exec pe2 ip link set dev-pe2-pop2 up  ip netns exec pe2 ip link set lo up  ip netns exec pe2 ip addr add 2002::a:1/96 dev dev-pe2-pop2  ip netns exec pe2 ip addr add 2002::a:1/96 dev dev-pe2-pe1  ip netns exec pe2 ip addr add 2002::a:1/96 dev dev-pe2-pe3  ip netns exec pe2 ip route add 3002::a:1/128 dev dev-pe2-pop2  ip netns exec pe2 ip route add 2001::a:1/128 dev dev-pe2-pe1  ip netns exec pe2 ip route add 2003::a:1/128 dev dev-pe2-pe3  ip netns exec pe2 ip route add 3002::/96 via 3002::a:1 dev dev-pe2-pop2  ip netns exec pe2 ip route add 2001::/96 via 2001::a:1 dev dev-pe2-pe1  ip netns exec pe2 ip route add 2003::/96 via 2003::a:1 dev dev-pe2-pe3      ip netns exec pe3 ip link set dev-pe3-pe1 up  ip netns exec pe3 ip link set dev-pe3-pe2 up  ip netns exec pe3 ip link set lo up  ip netns exec pe3 ip addr add 2003::a:1/96 dev dev-pe3-pe1  ip netns exec pe3 ip addr add 2003::a:1/96 dev dev-pe3-pe2  ip netns exec pe3 ip route add 2001::a:1/128 dev dev-pe3-pe1  ip netns exec pe3 ip route add 2002::a:1/128 dev dev-pe3-pe2  ip netns exec pe3 ip route add 2001::/96 via 2001::a:1 dev dev-pe3-pe1  ip netns exec pe3 ip route add 2002::/96 via 2002::a:1 dev dev-pe3-pe2 |

**编排SRv6指令：**

host-a设置pop1为网关，host-b设置pop2为网关。通过pop点封装SRv6报文，经过PE点打通。

POP通过封装BSID，引流到PE点。PE点负责将BSID映射为路径SID列表

指令详细说明如下：

|  |
| --- |
| # pop1设置往host-b网段10.20.0.0/24的报文，封装为SRv6报文。指令列表插入（PE1的BSID，PoP2的VPN SID）  # pop1和pop2的的VPN SID，实质上为END.DT4指令，关联租户的vrf表  ip netns exec pop1 ip route add 10.20.0.0/24 encap seg6 mode encap segs 2001::b1:1,3002::c1:1 dev dev-pop1-pe1    # pop1设置回程的VPN SID为END.DT4指令， 查找的vrf表为101  ip netns exec pop1 ip route add 3001::c1:1 encap seg6local action End.DT4 vrftable 101 dev vrf-pop1-a    # 设置PE1的BSID为End.B6.Encaps指令，封装PE路径为(PE3,PE2), 路径上PE2的指令设置为END.DX6指令，引导流量在PE2解开最外层的SRv6封装  ip netns exec pe1 ip -6 route add 2001::b1:1 encap seg6local action End.B6.Encaps srh segs 2003::a1:1,2002::a2:1 dev dev-pe1-pop1  # 设置PE1的END指令  ip netns exec pe1 ip -6 route add 2001::a1:1 encap seg6local action End dev dev-pe1-pop1  # 设置PE1的END.DX6指令  ip netns exec pe1 ip -6 route add 2001::a2:1 encap seg6local action End.DX6 nh6 :: dev dev-pe1-pop1    # 设置PE3的END指令  ip netns exec pe3 ip -6 route add 2003::a1:1 encap seg6local action End dev dev-pe3-pe1    # 设置PE2的END指令  ip netns exec pe2 ip -6 route add 2002::a1:1 encap seg6local action End dev dev-pe2-pe3  # 设置PE2的END.DX6指令  ip netns exec pe2 ip -6 route add 2002::a2:1 encap seg6local action End.DX6 nh6 :: dev dev-pe2-pop2  # 设置PE2的BSID为End.B6.Encaps指令，封装PE路径为(PE1)，路径上PE1的指令设置为END.DX6指令，引导回程流程在PE1上解开最外层SRv6封装  ip netns exec pe2 ip -6 route add 2002::b1:1 encap seg6local action End.B6.Encaps srh segs 2001::a2:1 dev dev-pe2-pe1    # pop2设置回程往host-a网段的报文，封装为SRv6报文。指令列表插入（PE2的BSID，PoP1的VPN SID）  ip netns exec pop2 ip route add 10.10.0.0/24 encap seg6 mode encap segs 2002::b1:1,3001::c1:1 dev dev-pop2-pe2  # pop2设置VPN SID为END.DT4指令， 查找的vrf表为101  ip netns exec pop2 ip route add 3002::c1:1 encap seg6local action End.DT4 vrftable 101 dev vrf-pop2-b |