Test

目前是上行：

mn端：

初始化（2个端口）:

1. 配置coa->发bu->接ba（针对每个端口）

2. 隧道（xfrm）两条：

tun1 (XinYinIP)(coa1, ha)；设备地址：索引地址（比如：1）

tun2 (XinYinIP) (coa2, ha)； 设备地址：索引地址（比如：2）

设计有无问题？

3. rule

886: from 202.187.11.38 lookup 248

887: from 202.157.11.38 lookup 249

888: bid == 100 lookup 250

889: bid == 200 lookup 251

890: from all lookup 250

4. route table

248路由表

default via 202.187.0.254 dev eth1

249路由表

default via 202.157.0.254 dev eth2

250路由表

default via tun1

251路由表

default via tun2

流程：

1. send

2. 系统默认调用 路由查询（其实没用）, 得到路由表250（tun1），

3. 调用xfrm lookup函数（是否配置启动xfrm后就能调用该函数），对数据包查询xfrm policy,因为一开始数据包没有源地址，数据包会获取tun1的设备的索引地址，查询成功后 template和state的比对等操作。生成一个dst链表。

4.对数据包添加包头。包头里放的是索引地址（源地址），目的地址

5. netfilter的output点

Iptables -A OUTPUT -t mangle –p udp –dport 1234 –j MARK –set-mark 200这样每个目的端口是1234的UDP数据包在经过Netfilter的OUTPUT点时，都会被打上200这个标记

（每个端口号对应一个业务，端口号是否在sk\_buff中可以获取到）

6. 加钩子函数，重新调用xfrm lookup函数，（选择子里多了bid,是否可扩展选择子）根据template和state的比对等操作，生成一个dst链表（选择相应的xfrm操作， xfrm这块流程不清晰）：

额外工作：修改数据包的包头索引（前面系统加进去的包头），该修改是否可行

7. 发包。

HA端：

和mn类似