# MAC转发表导致的网络丢包的问题

## 概述

防火墙在透明模式下，与交换机一样是根据MAC转发表来转发数据流的。查看转发表的命令如下：

比如我们要看透明域tp的MAC转发表，那么就查看tp.b的桥转发表。

dump netlink brctl name host tp.b

show bridge control interface tp.b host.

fdb: size=256, used=4, num=4, depth=1

Bridge tp.b host table

port no device devname mac addr vlan ttl attributes

 2 44 m1/4 88:b6:6b:09:00:03 1 0 Local Static

 1 43 m1/3 88:b6:6b:09:00:02 0 Local Static

 2 44 m1/4 88:b6:6b:09:00:03 0 Local Static

转发表是根据接口收到的数据包的源MAC地址来生成的。如果防火墙长时间没有收到某个MAC发出的流量，则会把它从转发表中删除。假如某个MAC从转发表中删除了，但是发往该MAC地址的流量怎么办？

通常采用flooding方式，往所有接口上转发该流量。

但是防火墙与交换机不同的地方在于，它flooding流量是有先决条件的，就是得有相应的策略允许flooding。举例如下：

1、如下图所示，192.168.1.1访问192.168.1.2，我们只需要建立策略，port1到port2 192.168.1.1到192.168.1.2的策略。

2、当192.168.1.1发起流量源MAC地址是11:11，目标MAC地址11:12，防火墙上就会生成MAC地址11:11 port1的转发表，

3、当192.168.1.2回应数据包发给192.168.1.1时，因为目标mac是11:12,就会根据转发表找到port1接口转发出去。

4、但是如果由于种种原因，MAC转发表中没有11:11的时候，192.168.1.2回应的数据包就会按照规则flood到所有接口上，但是这样操作的前提是得有从port2到port1的策略。实际上我们不可能为回应的数据包建立策略，防火墙就不会flooding该数据包，于是网络就不通了。



**什么时候会发生这种情况呢？通常有两种情况，一是双MAC地址的时候，比如192.168.1.1有两个MAC地址，它发送的时候用一个MAC地址，接收的时候用另外一个MAC地址。这样就会导致MAC转发表上没有它用于接收的MAC地址。另外一种情况就是异步的时候。发送和接收流量不在一条链路上。**

## 二、测试

**测试目的**：透明墙收到数据包不转发原因及解决方法

**防火墙配置**：

A防火墙：开启异步路由、4口→3口全通策略，3口到4口禁止。

B防火墙：开启异步路由、4口→3口全通策略，3口到4口禁止。

**路由路径**：PC1→FWB→PC2→FWA→PC1

**测试拓扑**：



**测试方法**：PC1 访问PC2，在防火墙A抓包分析。

**现象**：PC1 无法 ping 通PC2 , FW-A 抓包如下，数据进入3口不转发。

A (root) # dump sniffer packet any "host 192.168.1.2" 4

interfaces=[any]

filters=[host 192.168.1.2]

1.000000 port3 in 192.168.1.2 -> 172.16.1.3: icmp: echo reply

6.000000 port3 in 192.168.1.2 -> 172.16.1.3: icmp: echo reply

11.000000 port3 in 192.168.1.2 -> 172.16.1.3: icmp: echo reply

16.000000 port3 in 192.168.1.2 -> 172.16.1.3: icmp: echo reply

21.000000 port3 in 192.168.1.2 -> 172.16.1.3: icmp: echo reply

查看FW-A 转发数据包目的MAC，此MAC地址应该在port4口，如果有就可以访问了。



查看FW-A 转发表项，没有发现此MAC地址

A (global) # dump netlink brctl name host root.b

show bridge control interface root.b host.

fdb: hash(size=256, used=8, depth=1) num=8, max=4096

Bridge root.b host table

port no device devname fdom mac addr ttl attributes

 2 5 port4 0 a4:be:2b:aa:5b:b0 1

 2 5 port4 0 00:e0:fc:09:bc:f9 59

 1 4 port3 0 94:28:2e:42:c8:52 20

 2 5 port4 0 00:90:27:fe:cf:43 0 Local Static

 1 4 port3 0 00:90:27:fe:cf:42 0 Local Static

 4 7 port6 0 00:90:27:fe:cf:45 0 Local Static

 3 6 port5 0 00:90:27:fe:cf:44 0 Local Static

 1 4 port3 0 94:28:2e:42:c8:45 3

**解决方法**：在防火墙FW-A port3口执行: A (port3) # **set peer-interface port4 ,**强制将数据包转发到port4口，PC1就可以访问 PC2；该命令删除后，访问中断；另一种方法在端口下手工绑定MAC。

但为什么此mac没有在转发表呢？ 原因就是：此mac地址近几分钟没有发出过数据包，防火墙上就没有记录，arp广播一般几分钟发一个，防火墙转发表生存时间为300秒，到期后又会中断，这就是为什么一会通一会不通的原因。

FW-A截图：2.2.2.1 发出 ARP请求，2.2.2.2回复告知转发数据包目的MAC，立即生成转发表项目，P1C可以ping通PC2了，300秒结束后，就会中断。





**备注**：如果防火墙FWA 3→4口为全通策略时，不会出现无法访问现象，为什么呢？

正常收到数据包后，检查防火墙root.b转发表,若没有转发目的MAC地址，会向所有端口flooding，但由于防火墙处理机制原因，会检查下策略，如果没有策略匹配就不转发，如果加上相应策略，就转发该数据包了。

在防火墙FWA 3→4口添加（反向策略）192.168.1.0/24 → 172.16.1.0/24 策略后，即可通信一直无丢包。以下为debug flow抓包结果：

A (root) # id=36870 trace\_id=7 msg="vd-root received a packet(proto=1, 192.168.1.2:1->172.16.1.3:0) from port3."

id=36870 trace\_id=7 msg=**"send out via dev-port4,** dst-mac-a4:be:2b:aa:5b:b6"

id=36870 trace\_id=8 msg="vd-root received a packet(proto=1, 192.168.1.2:1->172.16.1.3:0) from port3."

id=36870 trace\_id=8 msg="send out via dev-port4, dst-mac-a4:be:2b:aa:5b:b6"

id=36870 trace\_id=9 msg="vd-root received a packet(proto=1, 192.168.1.2:1->172.16.1.3:0) from port3."

id=36870 trace\_id=9 msg="send out via dev-port4, dst-mac-a4:be:2b:aa:5b:b6"

id=36870 trace\_id=10 msg="vd-root received a packet(proto=1, 192.168.1.2:1->172.16.1.3:0) from port3."

id=36870 trace\_id=10 msg="send out via dev-port4, dst-mac-a4:be:2b:aa:5b:b6"

id=36870 trace\_id=11 msg="vd-root received a packet(proto=1, 192.168.1.2:1->172.16.1.3:0) from port3."

id=36870 trace\_id=11 msg="send out via dev-port4, dst-mac-a4:be:2b:aa:5b:b6"

id=36870 trace\_id=12 msg="vd-root received a packet(proto=1, 192.168.1.2:1->172.16.1.3:0) from port3."

id=36870 trace\_id=12 msg="send out via dev-port4, dst-mac-a4:be:2b:aa:5b:b6"