

上图描述：图中NODE2中的接口IP 2.10(标红色)是与外部设备连接的接口，没有备AODV接管。图中的其余接口（IP标黑）的都被AODV接管。

问题： 由于2.10接收到一个目的地是1.3的报文后，由于NODE2有一个1.0 \24的直连路由，所以就会触发内核发送一个请求1.3的MAC的ARP请求。ARP请求并不能触发AODV去发送路由请求，导致报文无法顺利到达NODE5。

解决： 1、开发一个程序负责从2.10接收报文，收到报文后将数据发送给内核，这样AODV就可以利用HOOK将报文捕获触发路由请求。

2、如果AODV接口回传报文则需要判断端口号和协议类型决定是否将报文转发到外部设备。

实现： 1、通过LIBCAP库接收接口数据（可参见router程序），然后使用sendto函数将数据发送给内核。Sendto示例代码如下。

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#include <unistd.h>

#include <sys/types.h>

#include <sys/socket.h>

#include <netinet/in.h>

int main() {

// 创建一个UDP套接字

int sockfd = socket(AF\_INET, SOCK\_DGRAM, 0);

if (sockfd < 0) {

perror("socket creation failed");

exit(EXIT\_FAILURE);

}

// 填充目的地址信息

struct sockaddr\_in dest\_addr;

memset(&dest\_addr, 0, sizeof(dest\_addr));

dest\_addr.sin\_family = AF\_INET;

dest\_addr.sin\_port = htons(12345); // 目的端口号

dest\_addr.sin\_addr.s\_addr = inet\_addr("127.0.0.1"); // 目的IP地址

char\* message = "Hello, kernel!";

// 发送数据

int send\_len = sendto(sockfd, message, strlen(message), 0, (const struct sockaddr\*)&dest\_addr, sizeof(dest\_addr));

if (send\_len < 0) {

perror("sendto failed");

close(sockfd);

exit(EXIT\_FAILURE);

}

printf("数据包已发送到内核。\n");

// 关闭套接字

close(sockfd);

return 0;

}