

DESCRIPTION IN THE PRECEDING FIGURE: THE INTERFACE IP 2.10 (MARKED IN RED) IN NODE2 IS THE INTERFACE CONNECTED TO THE EXTERNAL DEVICE, AND THERE IS NO STANDBY AODV TO TAKE OVER. The rest of the interfaces in the diagram (IP marked in black) are taken over by AODV.

Problem: SINCE 2.10 RECEIVES A PACKET WITH DESTINATION 1.3, NODE2 HAS A 1.0 \24 DIRECT ROUTE, TRIGGERING THE KERNEL TO SEND AN ARP REQUEST FOR THE MAC ADDRESS 1.3. The ARP request cannot trigger AODV to send a routing request, and the packet cannot reach NODE5.

Solution: 1. Develop a program responsible for receiving packets from 2.10, and send the data to the kernel after receiving the packets, so that AODV can use HOOK to capture packets and trigger routing requests.

2. If the AODV interface sends back packets, you need to determine the port number and protocol type to decide whether to forward the packets to external devices.

Implementation: 1. Receive the interface data through the LIBCAP library (see the router program), and then use the sendto function to send the data to the kernel. The Sendto sample code is as follows.

#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() {

Create a UDP socket

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

if (sockfd < 0) {

perror("socket creation failed");

exit(EXIT\_FAILURE);

}

Populate the destination information

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!";

Send data

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");

Close the socket

close(sockfd);

return 0;

}