

컴퓨터 네트워크 assignment1

2018170754 유인혁

(1) Explain why this function is needed

```
struct my_ip_routing_entry * route_output(struct my_ip_header my_iph) {
    struct my_ip_routing_entry * ptr = routing_entry_list;

    while(ptr != NULL) {
        if(ptr->dest_len <= my_iph.dest_len) {
            printf("%s %s %d %d\n", ptr->dest, my_iph.dest, ptr->dest_len, my_iph.dest_len);
            if(strncmp(ptr->dest, my_iph.dest, ptr->dest_len) == 0) {
                printf("Routing entry is found (In this assignment, assume that the firstly found routing entry is longest matching)\n");
                return ptr;
            }
        }

        ptr = ptr->next;
    }

    return NULL;
}
```

목적 ip주소의 prefix와 내 ip주소 prefix가 같은지 비교합니다. dest_len의 길이만큼 ptr->dest_len과 my_iph.dest를 비교해서 같으면 0을 반환하고 라우팅 엔트리를 찾았다고 해줍니다. 내 네트워크 안에 있으면 ptr에 next hop의 라우팅 엔트리를 넣어줍니다.

(2) How code? (Explain your code execution)

처음 함수를 실행하면 main() 함수 내에서 interface를 setup하는 함수 setup_my_interface()를 사용합니다. raw socket을 만들고 interface_info 구조체 내에 해당 interface의 이름, ip주소 등을 list로 만들어 줍니다. environment_setup은 맥 주소와 ip주소를 ARP cache entry에 넣어주고 라우팅 엔트리에 destination ip, next-hop, output interface를 기록해줍니다.

vm1은 my_ip_send() 함수를 사용해 ip주소와 ttl을 입력합니다. 해당 라우팅 엔트리를 기록하고 my_ip_output() 함수를 호출합니다. output interface가 어디인지를 결정한 다음, 엔트리에서 next-hop을 찾아주며, 만일 next-hop이 비어 있다면 direct로 dest IP를 넣어줍니다. cache list에서는 output interface의 엔트리를 찾아오고 next-hop에 대한 mac 주소를 가져와 IP헤더와 Ethernet header를 씌워 줍니다.

```
next_hop = reentry->next_hop;
if(next_hop == NULL){
    next_hop = (unsigned char*)malloc(strlen(my_iph.dest) + 1);
    memcpy(next_hop, my_iph.dest, strlen(my_iph.dest)+1);
}

// from cache_list bring mac
cache_ptr = output_interface->cache_list;
while(cache_ptr != NULL) {
    // Retrieve ARP Cache to find the mac address for the next hop
    // next_hop에 대한 Mac 주소 가져오는 부분
    // !!! Fill the blank (Setup next_hop_mac variable)

    if(strncmp(cache_ptr->dest, next_hop, cache_ptr->dest_len) == 0){
        next_hop_mac = (unsigned char *)malloc(6);
        memcpy(next_hop_mac, cache_ptr->dest_mac, 6);
        break;
    }

    cache_ptr = cache_ptr->next;
}
```

```
// !!! Fill the blank (Set dest, source len, source, ttl)
memcpy(buffer_ptr, my_iph.dest, my_iph.dest_len + 1);
buffer_ptr += my_iph.dest_len;

//source IP length
*((unsigned short*)buffer_ptr) = htons(my_iph.source_len);
buffer_ptr += sizeof(unsigned short);

//source IP
memcpy(buffer_ptr, my_iph.source, my_iph.source_len + 1);
buffer_ptr += my_iph.source_len;

//ttl
*buffer_ptr = my_iph.ttl;
```

(IP헤더를 씌우는 과정)

이렇게 저장된 buffer는 vm2 main으로 받아와 processEthernet() 함수로 들어갑니다.

```
unsigned char * ptr = buffer;
ptr += sizeof(eh);

ethtype = htons(eh.eth_type);
if( ethtype == 0xfffe){
    my_ip_receive(ptr);
}
```

이 함수에서 Ethernet헤더가 벗겨지고 my_ip_receive() 함수로 이동해 ip 헤더를 기록하고 my_ip_forward()로 이동합니다.

```
if(my_iph.ttl > 0){
    my_iph.ttl -= 1;
    if(my_iph.ttl == 0) return -1;
}else{
    return -1;
}
```

여기서 ttl을 깎고 my_ip_output() 함수를 수행하며 vm3로 패킷을 전달하게 됩니다.

(3)Without default gateway entry, how can you reduce the number of VM1's routing table entry?

디폴트 게이트웨이 엔트리가 없이 vm1의 엔트리 수를 줄이려면 성북구 안암동 1,2로 나뉘어 있는 것을 각각 성북구 안암동/18, 성북구 보문동/19로 바꿔주어 줄이거나 성북구/9로 서브넷 마스크 범위를 바꾸는 방법을 사용할 수 있습니다.

(4)How can you configure the default routing entry of VM1?

디폴트 게이트웨이를 쓰면 destination은 default gateway이고 next-hop은 외부로 나가는 인터페이스인 동대문구가 된다.