How to implement Auto-Configuration for W6100

Version 1.0

Table of Contents

1 Auto-Configuration Introduction

The automatic generation of a host IP address can be classified into two methods: **Stateful auto-configuration**, which obtains an address through a server such as DHCP, and **Stateless auto-configuration**, in which the host generates the address on its own.

In the server-based method, the host requests an address from a DHCP server, and the server assigns one of the available addresses to the host. Consequently, the server must maintain a large database and be managed rigorously.

In contrast, the stateless auto-configuration method allows the host to generate its own address using its interface ID along with prefix information obtained from a router or well-known prefixes. In this approach, the host is responsible for both the generation and assignment of its own IP address.

2 Stateless Auto-Configuration

Stateless Auto-Configuration includes methods for setting up a Link-Local Address and a Global Address.

2.1 Stateless Link-Local Address Auto-Configuration

A **link-local address** is valid only within a single link, and its propagation to external networks is blocked by the boundary router. However, such an address can still be utilized even in the absence of a router or DHCP server. A link-local address generated in this environment is not subject to time-based usage limitations and can be used at any time within the local link. These addresses can be automatically configured without the need for a router and are employed as source and destination node addresses in protocols such as the Neighbor Discovery Protocol (NDP). The format of a link-local address is illustrated in the figure below.

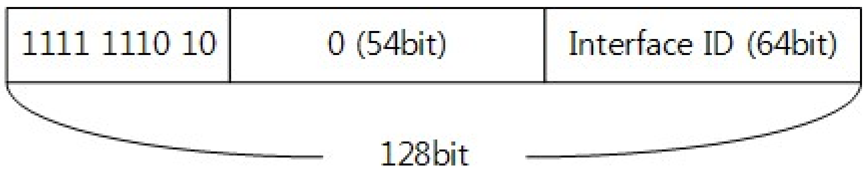

Figure 1 Structure of a Link-Local Address

The set of bits at the beginning of an IPv6 address is referred to as the **Prefix**, and the FE80::/10 prefix shown in the figure above is a predefined Prefix for link-local addresses. This Prefix is utilized to restrict the scope of packet delivery to a single link and serves to identify the network to which the packet belongs. Consequently, by using this Prefix, the host can autonomously generate its own link-local address without the need to obtain prefix information from a router. Such a link-local address is configured in the following scenarios, including during system boot:

- When initializing a network interface at system startup.
- When the interface is initialized after a temporary interface error or after being temporarily disabled for system management.
- When the interface is first attached to a link.
- When there is a system operation that enables the interface after it has been disabled for system management.

The general method for generating a link-local address involves appending the interface ID to the well-known Prefix. The space between the Prefix and the interface ID is filled with zeros. If the desired number of ID bits exceeds 118 bits (excluding the 10 bits of the Prefix), address generation becomes impossible. However, it is generally recommended to use 64 bits for the interface ID.

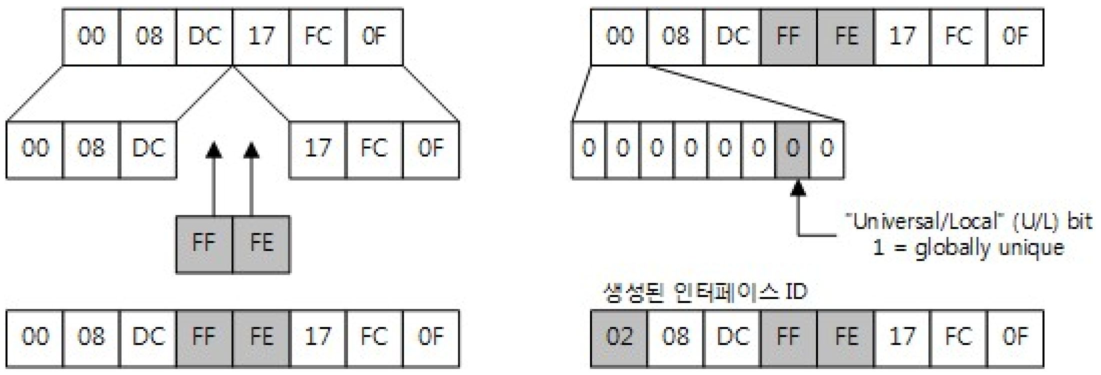

Figure 2 How to construct an Interface ID in an Ethernet environment

We will now examine the method of setting a link-local address in the widely used Ethernet environment. As previously explained, the Prefix is predefined, so to generate a link-local address in an Ethernet environment, it is necessary to construct the interface ID. The process for constructing the interface ID follows the method illustrated in Figure 2. For instance, if the Ethernet link-layer address is the 48-bit "00:08:DC:17:FC:0F", the corresponding 64-bit interface ID "02:08:DC:FF:FE:17:FC:0F" is generated. The "FF:FE" is inserted to conform to the IEEE EUI-64 (Extended Unique Identifier) format for Ethernet, and setting the 7th bit to "1" indicates that the interface ID has a global scope.

Once this interface ID is created, it is inserted into the interface ID field as shown in Figure 1, completing the link-local address. Since the uniqueness of this link-local address has not yet been verified within the link, it is considered a tentative address. The address must undergo the Duplicate Address Detection (DAD) process, as shown in Figure 3, to ensure its uniqueness.

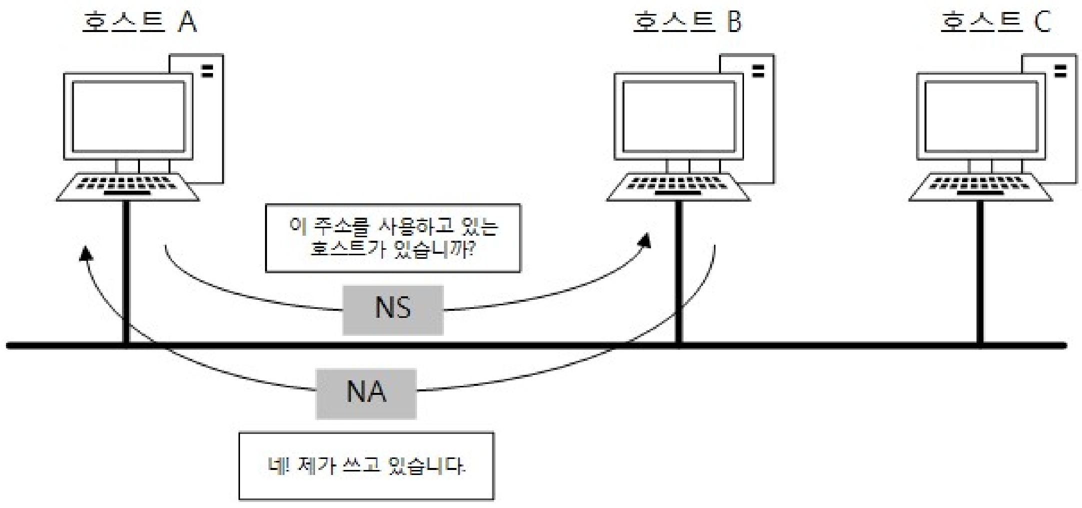

Figure 3 Duplicate Address Detection process

DAD (Duplicate Address Detection) is the process of checking for address conflicts before assigning an address to an interface. It is a mandatory step in the stateless address configuration method. Figure 3 illustrates the DAD process using the Neighbor Solicitation (NS) and Neighbor Advertisement (NA) message formats of the Neighbor Discovery Protocol.

After Host A generates a tentative address, it sends an NS packet to the network. If the host waits for a certain period without receiving an NA packet, it concludes that the uniqueness of the address has been verified and proceeds to assign the address to the interface. However, if an NA packet is received from another host, it indicates that the address is already in use, and the address cannot be assigned to the interface. In the event of a failed address assignment, the address must be obtained using the stateful address configuration method.

2.1.1 Sudo Code

TargetAddr is created and entered as shown in Figure 2

```
uint8_t flags;
setSLPIP6R(TargetAddr); // Set IPv6 Link Local Address
setSLRTR(0x2000); // Set Timeout value
setSLIMR(TIMEOUT|DAD_NS); // Timeout & DAD_NS Interrupt Masking
setSLCR(DAD_NS); // Execute DAD_NS command

do
{
    flags = getSLIR();
} while(flags == 0);

setSLIRCLR(0xFF); // Clear all SLIR flags
if((flags & TIMEOUT) == TIMEOUT) // DAD Success!
{
    printf("DAD Success!!\r\n");
    setLLAR(TargetAddr);
}
else if((flags & DAD_RECV) == DAD_RECV) // DAD Fail!
{
    printf("DAD Failed!!\r\n");
}
```

2.2 Stateless Global Address Auto-Configuration

To obtain a global address, unlike a link-local address where the prefix is predefined, it is necessary to obtain information required for address configuration, such as the prefix, from a router.

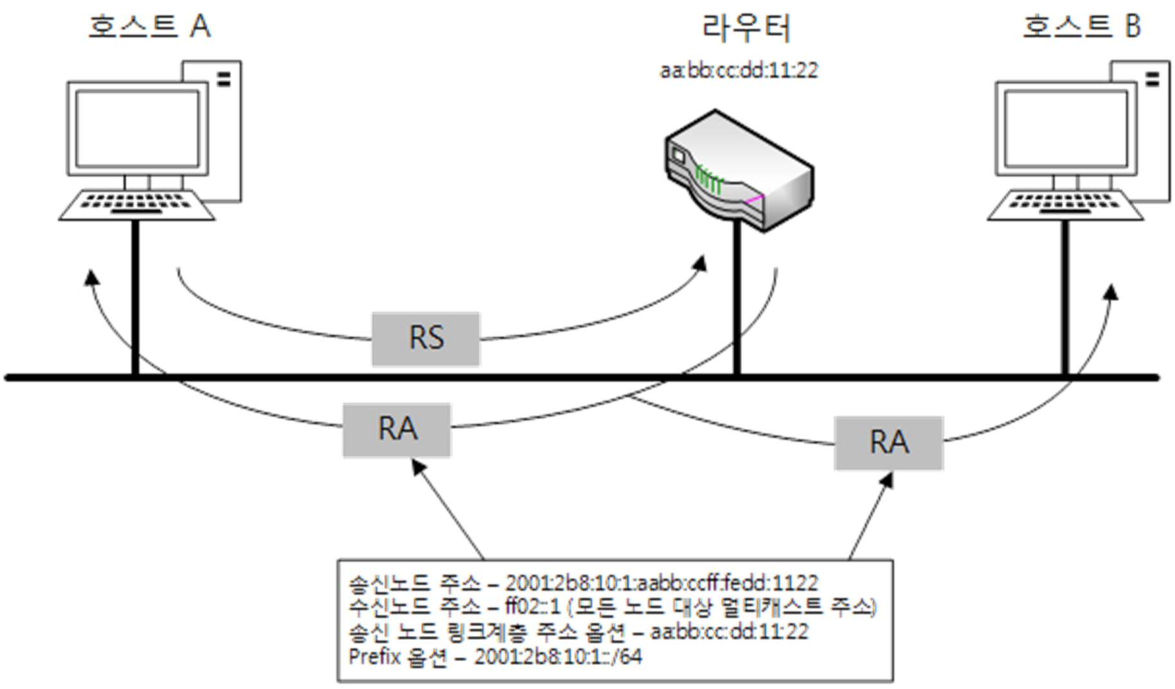
호스트 A
라우터
aa:bb:cc:dd:11:22
호스트 B
RS
RA
RA
송신노드 주소 – 2001:2b8:10:1:aabb:ccff:fedd:1122
수신노드 주소 – ff02::1 (모든 노드 대상 멀티캐스트 주소)
송신 노드 링크계층 주소 옵션 – aa:bb:cc:dd:11:22
Prefix 옵션 – 2001:2b8:10:1::/64

Figure 4 Process of obtaining address configuration information from a router

As shown in Figure 4, the host can either send a Router Solicitation (RS) message to the router to request an RA (Router Advertisement) packet containing this information or obtain it from the RA packets that the router periodically broadcasts. The information included in the RA packet consists of the prefix, valid lifetime, preferred lifetime, and the retransmission interval for the Neighbor Advertisement (NA) messages of each host. The host then uses the obtained prefix information along with its own interface ID to configure the global address.

The key difference between global address generation and link-local address generation is that the Duplicate Address Detection (DAD) process is not performed for global addresses. This exception applies only when the same 64-bit interface ID used for the link-local address generation is reused.

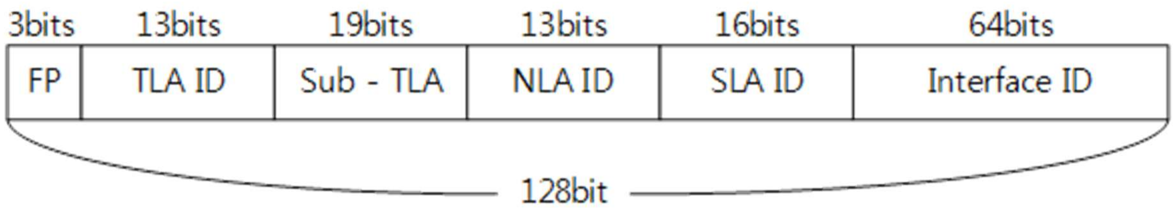

Figure 5 Global Address Format

To examine the address generation process in greater detail, as shown in Figure 4, Host A first sends an RS (Router Solicitation) packet to the all-routers multicast address. The router then responds by transmitting an RA (Router Advertisement) packet containing the address configuration information. The destination address of the RA packet is FF02::1, the all-nodes multicast address. This is because Host A has not yet been assigned a formal address, so the RA packet must be broadcast to all nodes to ensure that Host A can receive it.
After verifying the validity of the RA message, Host A uses its interface ID and the received prefix information to generate a global address, following the structure shown in Figure 5. Even after generating the address, Host A must continue receiving RA messages to update its prefix information and monitor the validity times. Additionally, to ensure smooth packet transmission, Host A must store and systematically manage information about neighboring nodes.

2.2.1 Sudo Code

This must be performed after the DAD for Stateless Link-Local Address Auto-Configuration.

```
DAD_run(LinklocalAddress);

setSLIMR(TIMEOUT&RA_RECV); // interrupt mask
setSLCR(AUTO_RS); //autoconfiguration RS command

do
{
    flags = getSLIR();
    printf(".");
}while(flags == 0);

setSLIRCLR(0xFF); // SLIR Clear

if((flags&Timeout)==Timeout)
{
    printf("Timeout!!\r\n");
}
else if((flags&RA_RECV)==RA_RECV)
{
    printf("Received RA!!\r\n");

    getRSPRXLEN(); // Prefix length
    getRSFLAGS();  // ICMPv6 Option Flag
    getRSVLIFETIME(VLIFE_array); //Valid Lifetime
    getRSPLIFETIME(PLIFE_array); // Preferred Lifetime
    getRSPRFIX(PRFIX_array); //Prefix Address
    getLLAR(LinklocalAddress);

    /* global front 7bit = prfix address */
    for(i=0 ; i <8 ; i++) global_addr [i] = PRFIX_array[i];
    /* global behind 7bit = link local address */
    for(j=8 ; j<16; j++) global_addr[j] = LinklocalAddress [j];

    setGUAR(global_addr);
}
```

3 Stateful Auto-Configuration

Stateful Auto-Configuration involves setting a Global Address using a DHCPv6 Server after performing Stateless Link-Local Address Auto-Configuration.

3.1 Stateful Global Address Auto-Configuration (DHCPv6)

DHCPv6 is the DHCP protocol designed for IPv6, supporting **Stateful Auto-Configuration** as a counterpart to Stateless Auto-Configuration. It provides a mechanism for reducing maintenance costs by centralizing the management of IP addresses, routing information, operating system

installation data, and directory service information on a limited number of DHCP servers. Furthermore, DHCPv6 is designed for easy extensibility through well-defined DHCP options, which allow the delivery of additional configuration parameters as needed.
In the following section, only the key features of DHCPv6 and the address acquisition process will be briefly described.

- DHCP can be used in conjunction with IPv6 Stateless Auto-Configuration.
- Except for security-related reasons, DHCP does not require manual configuration of network parameters on the client side.
- DHCP can coexist with static configurations, nodes that do not use DHCP, and existing network protocols.

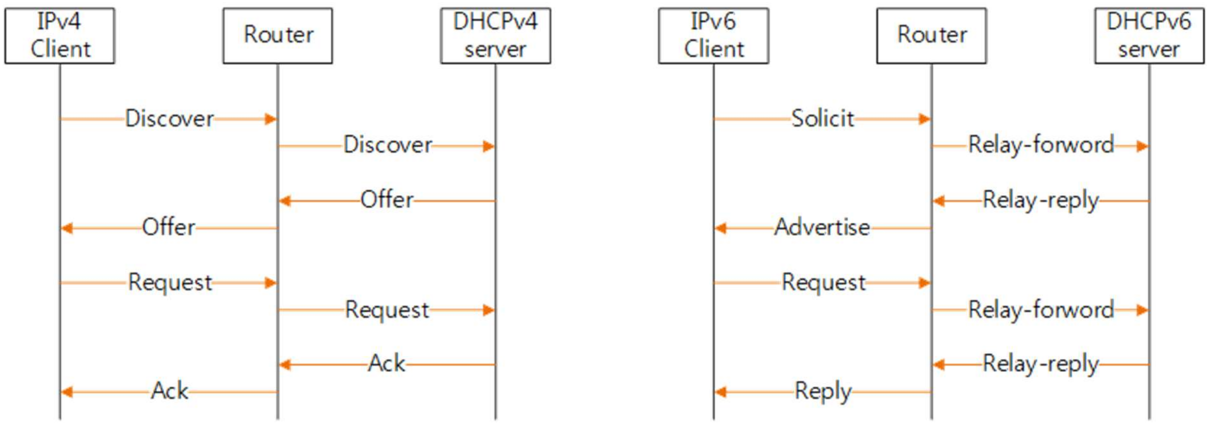

Figure 6 DHCP and DHCPv6 Process

DHCPv6 sends a **Solicit** Message instead of a Discover Message, but the content is similar: it is searching for a DHCPv6 server. Therefore, if you understand DHCP, you can also understand DHCPv6 without much difficulty.

3.1.1 Sudo Code

This must be done after Stateless Link-Local Address Auto-Configuration.

```
DAD_run(LinkLocalAddress);

DHCP_init(DHCP_SOCKET, test_buf);

while(1)
{
    switch(DHCP_run())
    {
        case DHCP_IP_ASSIGN:

        case DHCP_IP_CHANGED:
            check_flag = 1;
            if(check_flag)
            {
                check_flag = 0;
                close(TCP_SOCKET);
            }
            break;

        case DHCP_IP_LEASED:
            if(check_flag)
            {
                check_flag = 0;
            }

            /*
                TO DO YOUR NETWORK APPs
            */
            loopback_tcps(TCP_SOCKET, TCP_PORT, test_buf, AF_INET);
            break;

        case DHCP_FAILED:
            my_dhcp_retry++;
            if(my_dhcp_retry > MY_MAX_DHCP_RETRY)
            {
                my_dhcp_retry = 0;
                DHCP_stop();      // if restart, recall DHCP_init()
            }
            break;

        default:
            break;
    }
}
```

Document History Information

Version	Date	Descriptions
Ver. 1.0	01Oct2018	Release