

SOCKET-less Command Application Note

Version 1.0.0

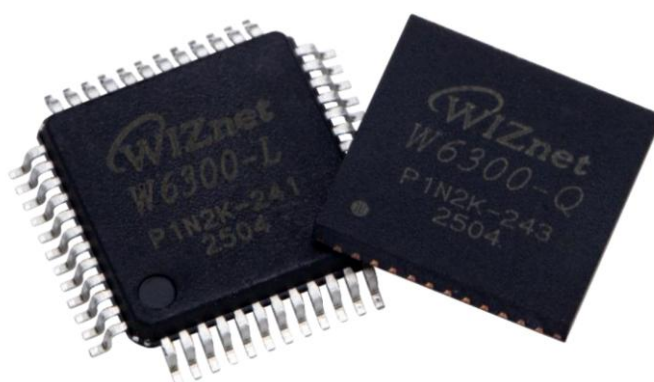


Table of Contents

1	Introduction	3
2	SOCKET-less Commands	3
2.1	ARP(Address Resolution Protocol).....	4
2.2	PING	7
2.3	NS(Neighborhood Solicitationn)	9
2.4	RS(Router Solicitation)	11
2.5	UNA (Unsolicited Neighbor Advertisement).....	14
3	Document History Information	16

List of Figures

Figure 1	ARP Command Operation Flow	4
Figure 2	PING Command Operation Flow	7
Figure 3	NS Command Operation Flow	10
Figure 4	RS Command Operation Flow	12
Figure 5	UNA Command Operation Flow	14

List of Tables

Table 1	Link-Local Address Format.....	10
Table 2	Global Unicast Address Format	12

1 Introduction

W6300은 별도의 SOCKET OPEN없이 SOCKET-less Commands를 통해 특정 Packet을 전송한다. Packet 전송에 대한 결과는 SLIR 레지스터를 통해 알 수 있으며, 특정 레지스터들을 통해 수신한 Reply Packet에 관한 정보들을 확인한다. 또한 SOCKET-less Commands는 이전 Command가 수행을 완료할 때까지 동시에 수행할 수 없다.

2 SOCKET-less Commands

SOCKET-less Commands는 특정 레지스터들을 통해 관련정보를 설정한 후에 Command를 수행시켜야 하며 Command의 종류는 아래와 같다.

- ARP4 Command
- PING4 Command
- ARP6 Command
- PING6 Command
- NS Command
- RS Command
- UNA Command

UNA Command를 제외한 Command들은 Request-Reply구조로 Request를 전송 한 후에 Reply 수신을 기다린다. Reply Packet을 SLRCR과 SLRTR을 통해 설정한 시간 내에 수신하지 못한 경우 TOUT Interrupt가 발생한다. Reply Packet을 수신한 경우에는 해당 Packet의 Interrupt가 발생한다.

Unsolicited NA Command의 경우 Reply가 없는 Packet으로 수신을 기다리지 않으며, 메시지 전송이 완료되었을 경우 TOUT Interrupt가 발생한다.

또한 SOCKET-less Commands는 동시에 수행할 수 없으므로 여러 개의 Command 수행을 원할 경우, 이전 Command에 대한 Interrupt를 확인 한 후 다음 Command를 수행 한다.

2.1 ARP(Address Resolution Protocol)

ARP란 Request-Reply 구조로 상대방의 MAC Address를 얻어오기 위한 message이다. ARP Request Message를 전송하여 해당 IP Address에 대한 MAC Address를 요청하며 ARP Response Message를 통해 MAC Address를 얻는다. IPv6에서는 ARP Request message 역할을 Neighbor Solicitation message가 하며, ARP Response message 역할을 Neighbor Advertisement message가 한다.

W6300은 기본 동작으로 TCP통신에서 Connect command나 IPRAW 또는 UDP 통신에서 Send Command를 설정할 경우, ARP를 수행한 후에 Mac address를 얻어 해당 주소로 Packet을 전송한다. Socket을 Destination Hardware address Mode로 설정 할 경우에는 ARP 없이 사용자가 Sn_DHAR에 설정한 MAC Address로 패킷을 전송한다. 이 외에 사용자가 별도로 ARP 전송을 원할 경우에는 SOCKET-less Command를 통해 전송이 가능하며 SLDHAR 레지스터를 통해 MAC Address 확인이 가능하다.

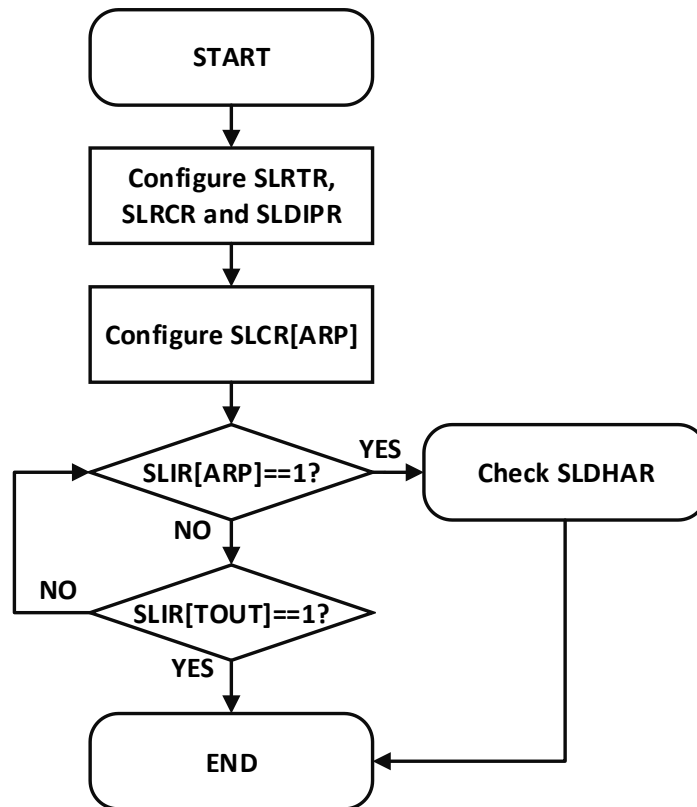


Figure 1 ARP Command Operation Flow

2.1.1 Example code

ioLibrary에 있는 ctlnetservice()를 이용하여 SOCKET-less Command를 통해 ARP Request를 전송한다. IPv4와 IPv6의 전송 방법은 동일하며 address의 length로 구분된다. ctlnetservice()의 return 값에 따라 ARP Response의 수신여부를 알 수 있다.

// IPv4 ARP Request 예제

```
wiz_ARP arp4_info = {
    .destinfo = {
        .ip = {192, 168, 0, 232},
        .len = 4
    }
};

if (ctlnetSERVICE(CNS_ARP, &arp4_info) == 0)
{
    printf("DHAR = %02X:%02X:%02X:%02X:%02X:%02X\r\n",
        arp4_info.dha[0], arp4_info.dha[1], arp4_info.dha[2],
        arp4_info.dha[3], arp4_info.dha[4], arp4_info.dha[5]);
}
else
{
    printf("No Target\r\n");
}
```

// IPv6 ARP Request 예제

```
wiz_ARP arp6_info = {
    .destinfo = {
        .ip = {
            0xFE, 0x80, 0x00, 0x00,
            0x00, 0x00, 0x00, 0x00,
            0x31, 0x71, 0x98, 0x05,
            0x70, 0x24, 0x4B, 0xB1
        },
        .len = 16
    }
};

if (ctlnetSERVICE(CNS_ARP, &arp6_info) == 0)
{
    printf("DHAR = %02X:%02X:%02X:%02X:%02X:%02X\r\n",
        arp6_info.dha[0], arp6_info.dha[1], arp6_info.dha[2],
        arp6_info.dha[3], arp6_info.dha[4], arp6_info.dha[5]);
}
```

```

}
else
{
    printf("No Target\r\n");
}
printf("No Target\r\n");
}

```

2.1.2 Success Case

ctlnetSERVICE()의 반환값이 0일 경우 ARP Response를 수신한 것으로 dha에 MAC Address가 저장 되어 있다.

No.	Time	Source	Destination	Protocol	Length	Info
659	43.521...	Wiznet_57:57:61	Broadcast	ARP	60	Who has 192.168.0.232? Tell 192.168.0.107
660	43.521...	AsustekC_2a:c2:e3	Wiznet_57:57:61	ARP	42	192.168.0.232 is at 08:62:66:2a:c2:e3

2.1.3 Fail Case

ctlnetSERVICE()의 반환값이 -1일 경우 Timeout처리되며 ARP Response를 수신 하지 못한 경우이다.

No.	Time	Source	Destination	Protocol	Length	Info
192	14.039...	Wiznet_57:57:61	Broadcast	ARP	60	Who has 192.168.0.9? Tell 192.168.0.107

Socket Less Command의 Retransmission Count의 초기값은 0으로 재전송이 발생 하지 않는다. 재전송을 발생 시키고 싶은 경우 Retransmission Count값을 설정하여야 하며 ctlnetwork()를 통해 설정한다. 예시로 sl_retry_count 를 3으로 설정하였으며 총 4개의 ARP Request가 전송되었다.

```

wiz_NetTimeout timeout = {
    .s_retry_cnt    = 8,
    .s_time_100us   = 2000,
    .sl_retry_cnt    = 3,
    .sl_time_100us   = 1000
};

ctlnetwork(CN_SET_TIMEOUT, &timeout);

```

No.	Time	Source	Destination	Protocol	Length	Info
815	20.479...	Wiznet_57:57:61	Broadcast	ARP	60	Who has 192.168.0.9? Tell 192.168.0.107
816	20.580...	Wiznet_57:57:61	Broadcast	ARP	60	Who has 192.168.0.9? Tell 192.168.0.107
818	20.680...	Wiznet_57:57:61	Broadcast	ARP	60	Who has 192.168.0.9? Tell 192.168.0.107
825	20.781...	Wiznet_57:57:61	Broadcast	ARP	60	Who has 192.168.0.9? Tell 192.168.0.107

2.2 PING

PING이란 Request-Reply 구조로 상대방의 네트워크 상태를 확인하기 위한 message이다. PING Request Message를 전송한 후 응답을 기다리며, Response Message를 통해 상대방의 네트워크 상태를 확인한다.

W6300으로 PING 전송을 원할 경우에는 SOCKET-less Command를 통해 전송이 가능하며, PING Request 전송 전에 자동으로 수행된 ARP과정을 통해 얻은 MAC Address를 SLDHAR 레지스터를 통해 확인 가능하다.

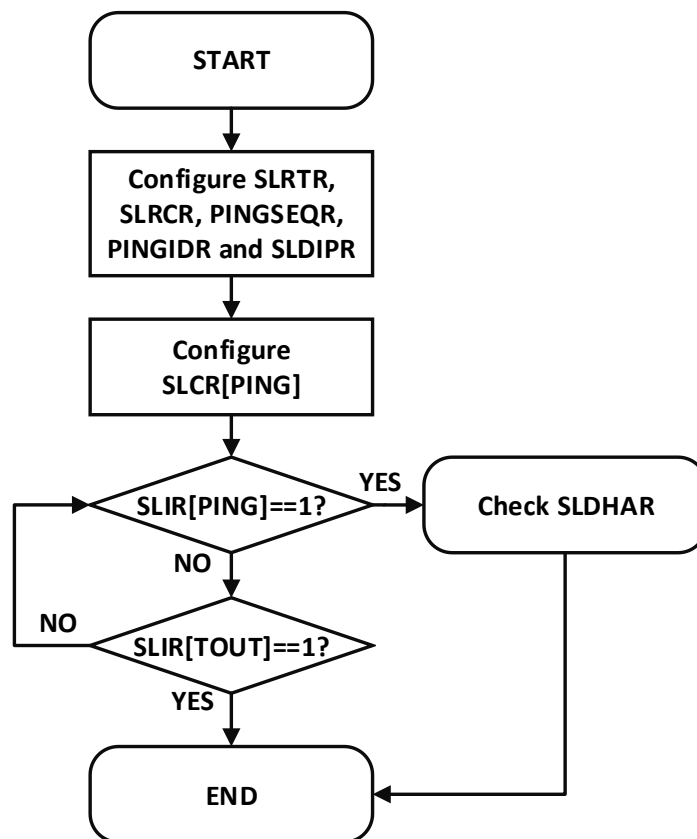


Figure 2 PING Command Operation Flow

2.2.1 Example code

Io6Library에 있는 ctlnetservice()를 이용하여 SOCKET-less Command를 통해 PING Request를 전송한다. IPv4와 IPv6의 전송 방법은 동일하며 address의 length로 구분된다. ctlnetservice()의 return 값에 따라 PING Response의 수신여부를 알 수 있다.

// IPv4 PING Request 예제

```
wiz_PING ping4_info = {
    .id = 0x1234,

    .seq = 0x4321,
    .destinfo = {
        .ip = {192, 168, 0, 232},
        .len = 4
    }
};

if (ctlnetSERVICE(CNS_PING, &ping4_info) == 0)
{
    printf("Receive a PING4 Response\r\n");
}
else
{
    printf("No Response\r\n");
}
```

// IPv6 PING Request 예제

```
wiz_PING ping6_info = {
    .id = 0x1234,
    .seq = 0x4321,
    .destinfo = {
        .ip = {
            0xFE, 0x80, 0x00, 0x00,
            0x00, 0x00, 0x00, 0x00,
            0x31, 0x71, 0x98, 0x05,
            0x70, 0x24, 0x4B, 0xB1
        },
        .len = 16
    }
};

if (ctlnetSERVICE(CNS_PING, &ping6_info) == 0)
{
    printf("Receive a PING6 Response\r\n");
}
```



```

}
else
{
    printf("No Response\r\n");
}

```

2.2.2 Success Case

ctltnetservice()의 반환값이 0일 경우 PING Response를 수신한 것으로 dha에 MAC Address가 저장된다.

Ping Request Packet의 ID와 Sequence값은 ping4_info에 설정한 id, seq로 적용된다.

No.	Time	Source	Destination	Protocol	Length	Info
55	3.9607...	Wiznet_57:57:61	Broadcast	ARP	60	Who has 192.168.0.232? Tell 192.168.0.107
56	3.9607...	AsustekC_2a:c2:e3	Wiznet_57:57:61	ARP	42	192.168.0.232 is at 08:62:66:2a:c2:e3
57	3.9608...	192.168.0.107	192.168.0.232	ICMP	74	Echo (ping) request id=0x1234, seq=17185/8515,
58	3.9609...	192.168.0.232	192.168.0.107	ICMP	74	Echo (ping) reply id=0x1234, seq=17185/8515,

2.2.3 Fail Case

ctltnetservice()의 반환값이 -1일 경우 Timeout 처리되며 크게 두 가지 경우가 있다.

첫 번째는 PING Request를 전송하기 전 ARP Response를 수신하지 못해 PING Request를 전송하지 못한 경우이다.

No.	Time	Source	Destination	Protocol	Length	Info
32	2.0309...	Wiznet_57:57:61	Broadcast	ARP	60	Who has 192.168.0.9? Tell 192.168.0.107

두 번째는 ARP Response를 수신하였으나, PING Request에 대한 응답을 얻지 못한 경우이다.

No.	Time	Source	Destination	Protocol	Length	Info
32	2.0309...	Wiznet_57:57:61	Broadcast	ARP	60	Who has 192.168.0.9? Tell 192.168.0.107
10...	41.108...	Wiznet_57:57:61	Broadcast	ARP	60	Who has 192.168.0.232? Tell 192.168.0.107
10...	41.108...	192.168.0.107	192.168.0.232	ICMP	74	Echo (ping) request id=0x1234, seq=17185/8515,

Retransmission Count를 증가시켜 재전송을 발생 시키고 싶은 경우 2.1.3을 참고하라.

2.3 NS(Neighborhood Solicitationn)

IPv6에서 Host IP Address는 stateful auto-configuration 방법과 stateless auto-configuration 방법으로 생성한다. Stateless auto-configuration은 Host가 스스로 주소를 생성하는 방식으로 LLA(Link Local Address)의 경우에는 NS Command를 통해 생성한다.

LLA는 정해진 Prefix 정보에 Interface ID 정보를 붙여서 생성하며, 생성한 LLA를 사용하기 위해서는 DAD(Duplicate Address Detection)과정을 수행해야 한다. DAD 과정이란 중복 주소 검출 과정으로 NS Packet을 전송한 후, NS Packet에 대한 NA Packet의 수신 여부로 검출이 가능 하다.

10 bits	54 bits	64 bits
1111 1110 10	0	Interface ID

Table 1 Link-Local Address Format

DAD 수행하기 위해 생성한 LLA를 SLDIPR에 설정한 후 NS Command를 통해 NS Packet를 전송한다. 일정시간 동안 NA Packet을 수신하지 못할 경우 유일성이 검증된 것으로 해당 LLA는 사용이 가능하다. 그러나 NS Packet에대한 응답으로 NA Packet을 수신할 경우, 해당 LLA는 이미 사용 중 이므로 사용 할 수 없다.

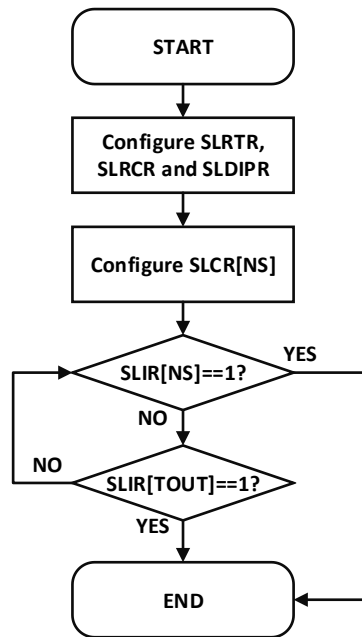


Figure 3 NS Command Operation Flow

2.3.1 Example Code

ioLibrary에 있는 ctlnetservice()를 이용하여 SOCKET-less Command를 통해 NS Packet을 전송하여 DAD를 수행한다. 매개변수로 사용하고자 하는 주소를 설정한다. ctlnetservice()의 return 값에 DAD 결과를 알 수 있다.

```

// IPv6 DAD 예제
uint8_t dad_info[16] = {
    0xFE, 0x80, 0x00, 0x00,
    0x00, 0x00, 0x00, 0x00,
    0x02, 0x08, 0xDC, 0xFF,
    0xFE, 0x57, 0x57, 0x61
};
  
```

```
if (ctlnetSERVICE(CNS_DAD, dad_info) == 0)
{
    printf("No Response (= Valid)\r\n");
}
else
{
    printf("Not Valid\r\n");
}
```

2.3.2 Success Case

ctlnetSERVICE()의 반환값이 0일 경우 NS에 대한 NA를 수신하지 못한 것을 뜻한다. 해당 IPv6 Address를 사용중인 Device가 없으므로 사용이 가능하다. Retransmission Count를 증가시켜 재전송을 발생 시키고 싶은 경우 2.1.3을 참고하라.

No.	Time	Source	Destination	Protocol	Length	Info
473	5.2732...	::	ff02::1:ff57:5761	ICMPv6	78	Neighbor Solicitation for fe80::208:dcff:fe57:5761

2.3.3 Fail Case

ctlnetSERVICE()의 반환값이 -1일 경우 NS에 대한 NA를 수신하였음을 뜻한다. 해당 IPv6 Address를 사용중인 Device가 존재하므로 해당 Address는 사용이 불가하다.

No.	Time	Source	Destination	Protocol	Length	Info
12...	15.080...	::	ff02::1:ff24:4bb1	ICMPv6	78	Neighbor Solicitation for fe80::3171:9805:7024:4bb1
12...	15.080...	fe80::3171:9805:7...	ff02::1	ICMPv6	86	Neighbor Advertisement fe80::3171:9805:7024:4bb1 (ov

2.4 RS(Router Solicitation)

NS에서 언급했듯이 IPv6에서 Host IP Address는 Stateless auto-configuration 방식으로 Host가 스스로 주소를 생성하며, GUA(Global Unicast Address)의 경우에는 RS Command를 통해 생성한다.

GUA의 경우에는 LLA와 달리 Prefix 정보가 정해져 있지 않다. 따라서 LLA의 경우에는 DAD를 수행하기 위해 NS Packet을 전송하였지만, GUA의 경우에는 Prefix 정보를 획득하기 위해 Router로 RS Packet을 전송한다.

Prefix 정보를 획득하기 위해 SLDIPR 레지스터에 Router의 IP Address를 설정한 후, RS Command를 통해 RS Packet을 전송한다. RS Packet에 대한 응답으로 RA Packet을 수신할 경

우, RA Packet에서 얻은 Prefix 정보와 Interface ID를 이용하여 GUA를 생성한다. Prefix Length, Valid Life Time, Preferred Life Time, Prefix Address를 각각 PLR, PFR, VLTR, PLTR, PAR 을 통해 확인한다.

48bits	16bits	64 bits
Prefix	Subnet ID	Interface ID

Table 2 Global Unicast Address Format

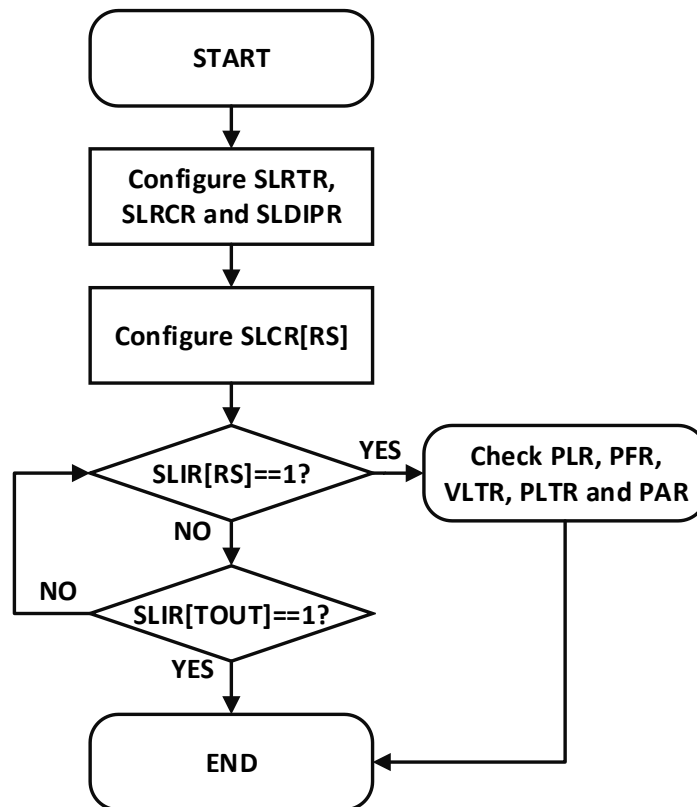


Figure 4 RS Command Operation Flow

2.4.1 Example Code

```

// RS 예제
wiz_Prefix info;

if (ctlnetSERVICE(CNS_SLAAC, &info) == 0)
{
    printf("Prefix Length = %d\r\n", info.len);
    printf("Prefix Flag = %d\r\n", info.flag);
    printf("Prefix Valid Lifetime = %d\r\n", info.valid_lifetime);
    printf("Prefix Preferred Lifetime = %d\r\n", info.preferred_lifetime);
}
  
```

```

printf("Prefix Address = %02X%02X:%02X%02X:%02X%02X:%02X%02X:"
"%02X%02X:%02X%02X:%02X%02X:%02X%02X\r\n",
info.prefix[0], info.prefix[1],
info.prefix[2], info.prefix[3],
info.prefix[4], info.prefix[5],
info.prefix[6], info.prefix[7],
info.prefix[8], info.prefix[9],
info.prefix[10], info.prefix[11],
info.prefix[12], info.prefix[13],
info.prefix[14], info.prefix[15]);
}
else
{
printf("Not Valid\r\n");
}

```

2.4.2 Success Case

ctltnetservice()의 반환 값이 0일 경우 RS에 대한 RA를 수신한 것으로 prefix length, prefix flag, prefix valid life time, prefix preferred life time, prefix address 가 저장된다. 단, 이값은 RS의 Option Field가 아래와 같이 첫 번째가 Source link-layer address, 두 번째가 Prefix information일 경우에만 보장된다. 이와 같은 환경이 보장되지 않는 경우 IPRAW6 SOCKET을 이용하여 해당 값들을 수신한다.

No.	Time	Source	Destination	Protocol	Length	Info
194	7.7701...	fe80::208:dcff:fe57:5761	ff02::2	ICMPv6	70	Router Solicitation from 00:08:dc:57:57:61
195	7.8548...	fe80::200:87ff:fe08:4c81	ff02::1	ICMPv6	110	Router Advertisement from 00:00:87:08:4c:81
196	7.8704...	Hitachi_10:8a:00...	Broadcast	ARP	60	Who has 102.168.10.1? Tell 102.168.10.1??

▶ Frame 195: 110 bytes on wire (880 bits), 110 bytes captured (880 bits)
 ▶ Ethernet II, Src: Hitachi_08:4c:81 (00:00:87:08:4c:81), Dst: IPv6mcast_01 (33:33:00:00:00:01)
 ▶ Internet Protocol Version 6, Src: fe80::200:87ff:fe08:4c81, Dst: ff02::1
 ▲ Internet Control Message Protocol v6
 Type: Router Advertisement (134)
 Code: 0
 Checksum: 0x7b54 [correct]
 [Checksum Status: Good]
 Cur hop limit: 64
 ▶ Flags: 0x00, Prf (Default Router Preference): Medium
 Router lifetime (s): 1800
 Reachable time (ms): 1800
 Retrans timer (ms): 0
 ▲ ICMPv6 Option (Source link-layer address : 00:00:87:08:4c:81)
 Type: Source link-layer address (1)
 Length: 1 (8 bytes)
 Link-layer address: Hitachi_08:4c:81 (00:00:87:08:4c:81)
 ▲ ICMPv6 Option (Prefix information : 2001:2b8:10:fffe::/64)
 Type: Prefix information (3)
 Length: 4 (32 bytes)
 Prefix Length: 64
 ▶ Flag: 0xc0, On-link flag(L), Autonomous address-configuration flag(A)
 Valid Lifetime: 86400
 Preferred Lifetime: 86400
 Reserved
 Prefix: 2001:2b8:10:fffe::

2.4.3 Fail Case

ctlnetSERVICE()의 반환값이 -1일 경우 Timeout처리되며 RS에대한 RA를 수신 하지 못한 경우 이다. Retransmission Count를 증가시켜 재전송을 발생 시키고 싶은 경우 2.1.3을 참고하라.

No.	Time	Source	Destination	Protocol	Length	Info
286	3.8233...	fe80::208:dcff:fe57:5761	ff02::2	ICMPv6	70	Router Solicitation from 00:08:dc:57:57:61

2.5 UNA (Unsolicited Neighbor Advertisement)

NA Message는 일반적으로 NS Message에 대한 응답으로 사용되는 Message이지만, 받지않은 경우에도 변경사항을 알리기 위해서 사용된다. UNA Command시 이러한 경우의 Message를 전송하며 다른 Command와 달리 Message 전송 완료 시 TOUT Interrupt가 발생한다.

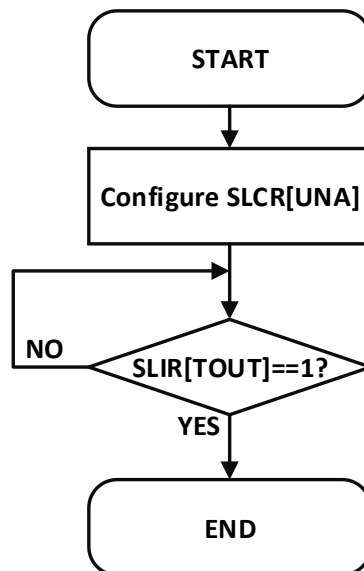


Figure 5 UNA Command Operation Flow

2.5.1 Example Code

```

//UNA 예제
if (ctlnetSERVICE(CNS_UN SOL_NA, 0) == 0)
{
    printf("Transmit UNA Packet\r\n");
}
else
{
    printf("Transmit Failed\r\n");
}
  
```

2.5.2 Success Case

ctlnetSERVICE()의 반환값이 0일 경우 NA를 전송한 것을 뜻한다.

No.	Time	Source	Destination	Protocol	Length	Info
245	3.4244...	fe80::208:dcff:fe57:5761	ff02::1	ICMPv6	86	Neighbor Advertisement fe80::208:dcff:fe57:5761

3 Document History Information

Version	Date	Descriptions
Ver. 1.0.0	Dec, 2025	Release

Copyright Notice

Copyright 2025 WIZnet Co., Ltd. All Rights Reserved.

Technical support : <https://maker.wiznet.io/forum>

Sales & Distribution: sales@wiznet.io

For more information, visit our website at <http://www.wiznet.io>