

# 브로드캐스팅 트래픽을 최소화한 DHCP의 설계 및 구현

안 성 진<sup>†</sup> · 정 진 옥<sup>††</sup> · 함 영 환<sup>†††</sup> · 이 춘 희<sup>†††</sup> · 박 창 순<sup>†††</sup> · 김 해 진<sup>††††</sup>

## 요 약

DHCP는 호스트에게 IP(Internet Protocol) 주소나 구성 파라미터(Configuration Parameter)등을 동적으로 할당해주는 프로토콜이다. 그러나 이 프로토콜은 메시지를 교환하기 위해 브로드캐스팅을 이용하기 때문에 네트워크에 과중한 트래픽의 발생 요인이 된다. 이러한 오버헤드를 줄이기 위해 메시지 교환시에 가능하면 브로드캐스팅을 피하고, 간결하고 효율적인 클라이언트-서버의 동작 메커니즘을 설계하였다. 본 연구에서는 DHCP가 유발하는 브로드캐스팅 트래픽 특성을 고찰하여 메시지를 유니캐스트할 수 있는 방안과 트래픽을 줄일 수 있도록 알고리즘을 개선한 DHCP의 모델을 구현하였다.

## Design and Implementation of DHCP with Minimized Broadcasting Traffic

Seong Jin Ahn<sup>†</sup> · Jin Wook Chung<sup>††</sup> · Young Hwan Ham<sup>†††</sup>  
Choon Hi Lee<sup>†††</sup> · Chang Soon Park<sup>†††</sup> · Hae Jin Kim<sup>††††</sup>

## ABSTRACT

DHCP(Dynamic Host Configuration Protocol) is a protocol which dynamically allocates an IP address and/or host configuration parameters to a host. But because this protocol uses broadcasting for message exchange, it causes heavy traffic and overload on the network. By avoiding unnecessary broadcasting and using a simple and efficient implementation model, we can reduce such heavy traffic and overload. We have studied on characteristics of the broadcasting traffic caused by DHCP, and implement the model which can reduce traffic by improving algorithm and unicasting messages.

### 1. 서 론

TCP/IP는 전세계적으로 수백만대 이상의 호스트가 연결되어 있는 인터넷의 네트워크 표준으로 발전되어 왔다. TCP/IP네트워크는 중앙 집중형 네트워크

가 아니기 때문에 호스트들을 관리하기 위해서는 일일이 구성(configuration)정보를 관리자가 설정을 해주어야 하고 이의 변경 시에는 호스트마다 구성정보를 변경해 주어야 한다. 특히 IP 주소의 경우에는 서로간의 통신을 위해서 꼭 필요하고 그 수가 제한되어 있기 때문에 이의 효율적인 관리가 필요하다[1, 2].

이와 같은 호스트들의 중앙 집중적인 관리, IP 주소와 같은 한정된 자원의 효율적인 관리, 그리고 이동 호스트와 같이 주소를 일시적으로 사용하는 경우나 서브네팅(subnetting)이나 주소의 변경이 잦은 경우

† 정 회 원:성균관대학교  
†† 중신회원:성균관대학교  
††† 정 회 원:시스템공학연구소  
†††† 정 회 원:한국전자통신연구소  
논문접수:1995년 11월 13일, 심사완료:1996년 2월 23일

주소관리를 위한 프로토콜로서 DHCP(Dynamic Host Configuration Protocol)가 등장하게 되었다[3, 6].

DHCP을 사용하는 경우에는 호스트를 네트워크에 접속하자마자 서버에서 IP 주소를 할당해주므로 즉시 네트워크 단말로서 동작하여 이른바 “Plug & Play” 환경을 제공해 준다. 이러한 DHCP는 IP 주소를 할당하기 위한 프로토콜이기 때문에 메시지를 교환할 때에 IP 주소를 가지고 일대일 통신이 가능하다. 그러므로 이러한 메시지들은 브로드캐스팅되어야 하고 이 과정에서 다른 호스트들에게 불필요한 부하를 발생시키며 네트워크 트래픽을 증가시키는 요인이 된다[9]. 본 논문에서는 이러한 문제를 해결하기 위해서 메시지의 유니캐스트와 알고리즘의 개선을 통하여 브로드캐스팅 트래픽을 최소화하면서 효율적인 DHCP를 구현하기 위해 제안한 모델과 이의 구현에 대해서 설명하려 한다.

## 2. DHCP의 동작

### 2.1 주소 할당방식

DHCP를 이용하기 위해서는 DHCP 클라이언트와 서버가 필요하며 DHCP 클라이언트를 갖춘 호스트는 기동 시에 서버로부터 주소를 할당받아서 사용한다. 한편, DHCP서버는 네트워크 관리자가 필요로 하는 수의 적절한 IP 주소를 미리 등록해 둔다. DHCP는 관리자의 편의를 위해서 클라이언트에게 주소를 부여하는 세 가지 방식인 “동적 할당”, “자동 할당”, “직접 할당”을 제공한다.

동적 할당은 네트워크에 일시적으로 접속하는 단말에 효과적인 방법으로 클라이언트는 일정기간(lease)동안의 주소 사용을 요청하고 서버는 그 기간동안에는 할당된 주소를 다른 호스트에게 할당하지 않는다[6].

자동 할당은 DHCP서버는 클라이언트에게 영구적인 IP 주소를 할당하기 때문에 중간에 IP 주소의 변경이 없으므로 동적 할당과 달리 네임서버를 이용해 IP 주소를 알아내지 않고 상대방의 이름을 가지고 IP 주소를 특별히 정할수 있어 DHCP클라이언트간 또는 DHCP클라이언트와 기존 단말간의 통신이 쉬워진다.

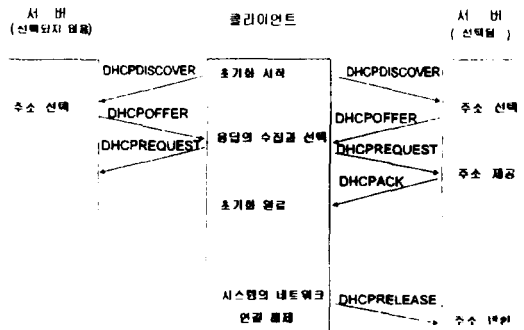
직접 할당은 네트워크 관리자가 호스트의 IP 주소를 할당하는 방식으로 DHCP는 할당된 주소를 단순

히 호스트에 옮기는 역할만을 한다. 서브네트워크에 호스트를 추가한다든지 삭제한다든지 할 때마다 DHCP서버의 설정변경이 필요하다.

### 2.2 클라이언트-서버의 상호동작

DHCP는 네트워크 주소의 할당을 위해서 클라이언트-서버 모델을 사용한다. 클라이언트가 서버로부터 주소를 할당받기 위한 상호동작 순서와 메시지는 아래와 같다.

- ① 클라이언트는 DHCPDISCOVER 메시지를 로컬 서브네트에 브로드캐스팅한다.
- ② 각 서버는 ‘yiaddr’ 필드에 이용가능한 주소가 들어있는 DHCPPOFFER 메시지로 응답한다.
- ③ 클라이언트는 하나이상의 DHCPPOFFER 메시지를 받는다. 그 중에서 서버를 하나 선택한다. 선택된 서버를 가리키는 ‘server identifier’ 옵션을 포함하는 DHCPREQUEST 메시지를 서버에게 보낸다.
- ④ 서버는 DHCPREQUEST 메시지를 받는다. 클라이언트를 위한 구성 파라미터를 포함하는 DHCPACK 메시지를 보낸다. DHCPREQUEST 메시지를 만족시킬 수 없는 경우 DHCPACK 메시지로 응답해야 한다.
- ⑤ 클라이언트는 DHCPACK 메시지를 받는다. 클라이언트가 더 이상 네트워크를 사용하지 않게 되었을 때는 DHCPRELEASE 메시지를 서버에게 보내 할당받은 주소를 반납한다.



(그림 1) 클라이언트와 서버의 상호동작  
(Fig. 1) Client-server interactions

위에서 설명한 클라이언트와 서버의 상호동작 순서와 메시지의 교환은 아래의 그림 1과 같고 여기에서 DHCPRELEASE 메시지를 제외한 메시지는 기본적으로 브로드캐스팅 된다.

### 3. 브로드캐스팅 트래픽의 최소화

#### 3.1 DHCP 브로드캐스팅 트래픽의 발생형태

DHCP 프로토콜은 호스트에게 IP 주소를 할당하기 위해 사용되는 프로토콜이기 때문에 IP 주소를 이용한 통신은 불가능하다. 그러므로 원칙적으로 클라이언트에게 IP 주소할당을 완료하기 전까지는 교환되는 메시지를 브로드캐스팅해야 한다. 이와 같은 브로드캐스팅 메시지는 서버의 메시지와 클라이언트의 메시지로 분류할 수 있다.

우선 클라이언트의 메시지를 보면 DHCPDISCOVER는 서버의 주소를 알지 못한 상태에서 서버를 알기 위해 보내는 메시지이기 때문에 브로드캐스팅해야 하고 DHCPREQUEST 메시지는 서버의 주소는 알고 있지만 여러 개의 서버로부터 응답이 온 경우 선택되지 않은 서버에게 그 사실을 알리기 위해 브로드캐스팅이 필요하다.

서버의 메시지인 DHCPPOFFER의 경우 아직 클라이언트에게 IP 주소가 할당되지 않았기 때문에 그리고 DHCPACK와 DHCPNAK의 경우 클라이언트측에서 자신의 IP 주소를 인식하지 못할 수도 있으므로 브로드캐스팅이 필요하다. IP 주소를 할당한 이후에 서버에게 보내는 메시지 즉 DHCPDECLINE과 DHCPRELEASE 메시지는 유니캐스트한다[6, 7]. 이상을 정리해 보면 다음과 같다.

〈표 1〉 메시지의 종류  
〈Table 1〉 DHCP message type

	서버 메시지	클라이언트 메시지
브로드캐스트	DHCPPOFFER DHCPACK, DHCPNAK	DHCPDISCOVER DHCPREQUEST
유니캐스트		DHCPDECLINE DHCPRELEASE

#### 3.2 DHCP브로드캐스팅 트래픽의 문제점

브로드캐스팅은 다수의 수신 측에게 단일 메시지

를 보내는 응용 서비스에서만 의미가 있는 것으로 전송계층으로 UDP를 사용해야만 가능하다. 이와 같은 브로드캐스팅을 사용하는 DHCP는 네트워크에 트래픽을 증가시키고 브로드캐스팅에 참여하지 않는 호스트들에게도 필터링을 위한 프로세싱 부하가 걸리게 하는 문제점이 있다[8, 9].

브로드캐스팅 트래픽은 메시지를 브로드캐스팅 해야하는 경우에 불가피하게 생길 수도 있지만 주소할당 알고리즘상의 문제로 인해 트래픽이 증가하는 경우도 생긴다. 이런 경우는 아래와 같다.

(1)서버가 DHCPPOFFER로 주소를 제공하고 나서 DHCPREQUEST로 응답을 받기 전에 다른 호스트의 요구에 의해 같은 주소를 할당하게 되는 경우 주소의 중복에 의해 두 호스트가 다시 재 기동하게 되고 루프가 발생될 수도 있다.

(2)서버와 클라이언트 양쪽에서 사용(lease)기간을 정할 수 있게 하는 경우 클라이언트에서 DHCPREQUEST의 사용기간을 서버가 들어줄 수 없는 경우 DHCPNAK로 응답하므로 클라이언트는 재 기동하게 되고 루프가 발생될 수 있다.

#### 3.3 트래픽 최소화 방안

브로드캐스팅 트래픽은 네트워크상의 여러 호스트에게 불필요한 부하를 준다. 그러므로 효율적인 DHCP를 설계하기 위해서는 이를 최소화할 수 있는 방안이 필요하다. 브로드캐스팅 트래픽을 최소화할 수 있는 방안은 크게 두 가지로 분류할 수가 있다.

##### (1) 메시지를 유니캐스트하는 방법

###### 1) DHCPREQUEST의 유니캐스트

DHCPREQUEST 메시지는 서버의 주소가 알려진 상태이기 때문에 유니캐스트가 가능하다. 이렇게 할 경우의 문제점은 서버가 2개 이상인 경우에 자신이 클라이언트에게 제공한 주소가 클라이언트에 의해 선택되지 않은 서버는 그 사실을 알 방법이 없게 된다(브로드캐스팅시에는 DHCPREQUEST메시지의 서버필드를 보고 알 수 있다). 이를 해결하기 위해서는 각 서버에게 DHCPREQUEST를 각각 유니캐스트하면 된다.

###### 2) DHCPPOFFER, DHCPACK, DHCPNAK의 유

**니캐스트**

DHCPOFFER, DHCPACK, DHCPNAK 메시지의 경우 서버측에서 클라이언트의 하드웨어 주소를 알 수 있으므로 이 하드웨어 주소와 새로 할당해 줄 IP 주소를 ARP캐쉬에 바인딩하여 엔트리를 만듦으로써 유니캐스트가 가능하다.

**3) REBINDING 상태에서 DHCPREQUEST의 유니캐스트**

클라이언트는 IP 주소를 할당받아서 사용하다가 사용기간이 종료되면 사용기간의 갱신을 위해서 지정된 서버에게 DHCPREQUEST 메시지를 보낸다. 이때 지정된 서버에서 응답이 없으면 DHCPREQUEST를 브로드캐스팅해야 한다. 이를 피하기 위해서는 미리 서버들의 주소를 기억하고 있다가 각각의 주소로 응답을 받을 때까지 유니캐스트하는 방법을 생각할 수 있다.

**(2) 알고리즘의 개선**

① 클라이언트가 여러개의 DHCPOFFER 메시지 중에서 하나를 선택할 때 처음으로 도착하는 메시지를 선택하는 방법.

② 주소중복으로 클라이언트가 처음부터 다시 시작하는 상태를 막기 위해서 서버측에서 DHCPOFFER 메시지 전송 후 그 주소를 할당된 것으로 표시하는 방법.

③ 클라이언트의 DHCPREQUEST에서 사용기간 요구시 서버에서 이를 들어줄 수 없는 경우 DHCPNAK로 응답하고 클라이언트는 재 기동하고 루프로 빠질 수 있는데 이것을 막기 위해 서버에서만 사용기간을 설정할 수 있도록 한다.

위에서 제안한 알고리즘의 개선방안과 기존의 RFC에서 나타나 있는 방식을 비교하면 표 2와 같다.

**<표 2> 제안방식과 RFC의 비교**

**<Table 2> The comparison between the proposed and the shown RFC**

	RFC	제안방식
DHCPOFFER의 선택	정의안됨	처음 도착 메시지
제공된 주소의 관리	정의안됨	할당된 것으로 표시
임대기간의 설정	클라이언트 또는 서버	서버

**4. 프로토콜 설계**

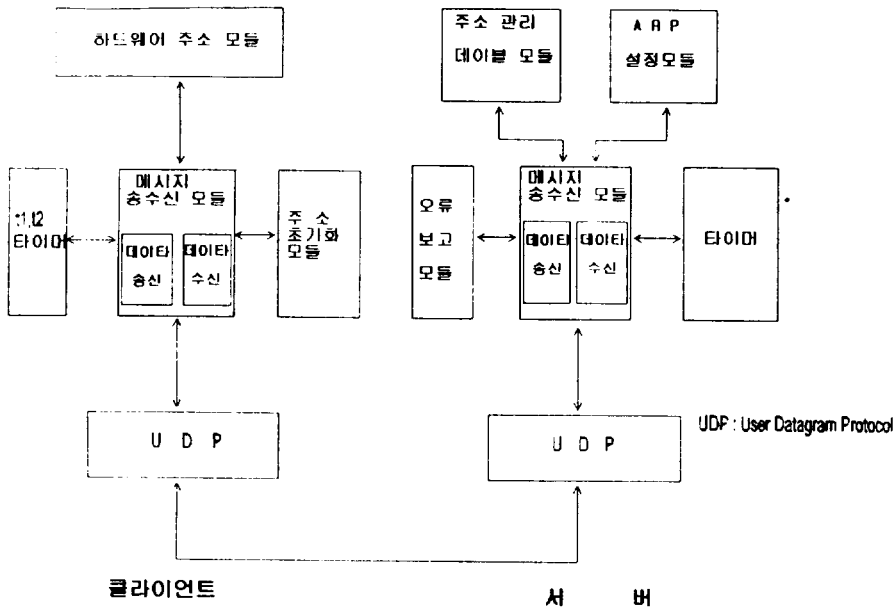
**4.1 구현 모델**

앞절에서 제시한 메시지의 유니캐스트와 개선된 알고리즘을 구현하기 위한 구현 모델은 아래의 그림 2와 같은 구성도를 가지며 기본적으로 링크계층은 IP, 네트워크 계층은 UDP를 사용한다[6, 7]. 클라이언트는 서버의 위치를 알아내고 IP 주소를 받기 위한 클라이언트의 메시지와 서버로부터의 메시지를 위한 메시지 처리 모듈이 있다. 그리고 IP 주소를 할당받은 후 시스템의 IP 주소를 초기화하기 위한 주소 초기화 모듈, 시스템의 하드웨어 주소를 얻어내기 위한 하드웨어 주소 모듈, 사용기간이 어느 정도 지난 후의 사용기간연장을 위한 타이머인 t1, t2타이머 모듈이 있다.

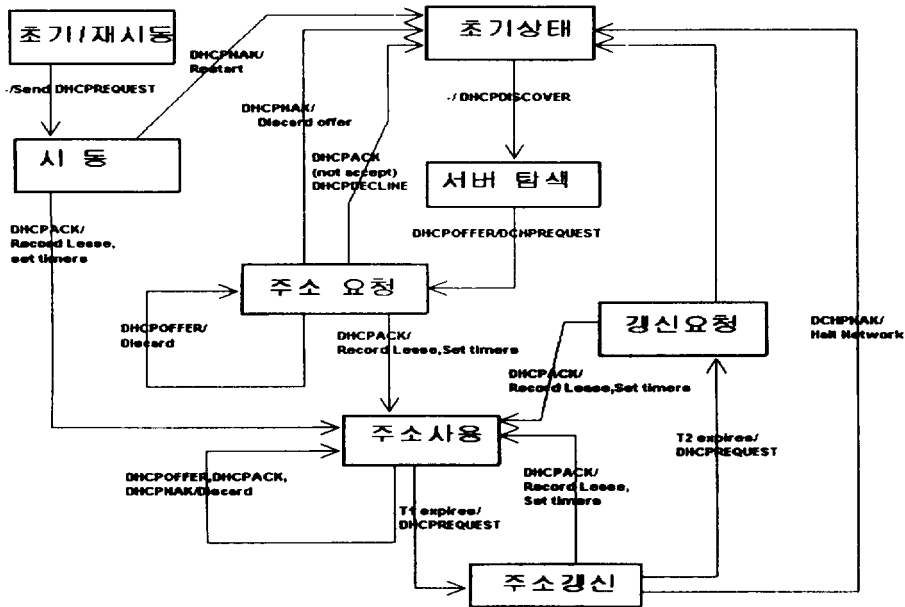
서버는 클라이언트의 메시지와 클라이언트에 주소를 할당하기 위한 메시지를 위한 메시지 처리 모듈이 있다. 그리고 에러를 처리하기 위한 에러처리 모듈, 클라이언트를 위한 IP 주소와 각종 파라미터를 저장하기 위한 호스트 구성정보 테이블 모듈, 클라이언트에게 메시지 유니캐스트하기 위한 서버의 ARP캐쉬 항목을 만들어 주는 ARP 초기화 모듈, 클라이언트에게 할당된 주소의 남은 사용기간을 주기적으로 검사하기 위한 타이머 모듈로 구성된다.

**4.2 클라이언트와 서버의 상태천이도**

초기상태는 클라이언트가 처음 네트워크에 접속되었을 때의 초기화 상태로 DHCPDISCOVER 메시지를 보낸 후 서버탐색 상태가 된다. 서버탐색 상태는 서버로부터의 주소할당 제의를 기다리는 상태에서 DHCPOFFER 메시지를 받으면 주소의 할당요구(DHCPREQUEST)를 한 후 DHCPACK를 기다리는 상태인 주소요청 상태가 된다. 주소요청 상태에서 DHCPACK를 받으면 주소를 사용할 수 있게되고 T1 시간이 지나면 사용기간을 연장하기 위해 주소갱신 상태가 되며 여기서 연장허가를 받으면 주소사용 상태로 돌아가나 그렇지 못하면 T2시간경과 후 갱신요청 상태로 간다. 갱신요청 상태에서 DHCPACK를 받으면 주소사용 상태로 가고 그렇지 못하면 초기상태로 간다. 클라이언트의 상태천이도는 아래의 그림 3과 같다.



(그림 2) DHCP 구현모델의 구성도  
(Fig. 2) DHCP implementation model diagram

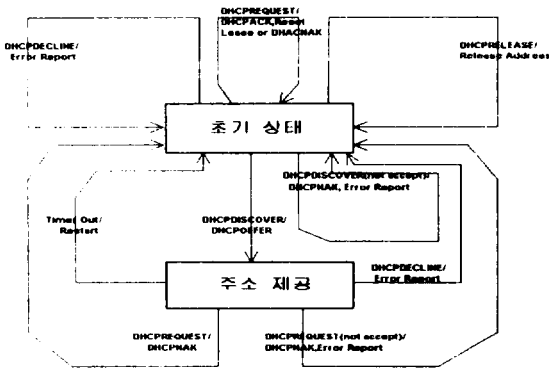


(그림 3) 클라이언트의 상태천이도  
(Fig. 3) The state transition diagram of the client

서버는 간단하게 두개의 상태만을 둔다. 초기 상태는 IP 주소의 제공, 연장, 거부, 해제를 처리하는 상태를 가리킨다.

이 상태에서 서버는 DHCPDISCOVER와 DHCPREQUEST 메시지에 대해서 DHCPACK나 DHCPNAK 메시지로 응답하고 DHCPDECLINE과 DHCPREL-

EASE메시지에 적합한 행동을 취한다. 초기 상태에서 DHCPDISCOVER메시지를 받았을 때 할당해 줄 주소가 있다면 DHCPPOFFER메시지로 응답하고, 주소제공 상태로 간다. 이 상태는 IP 주소를 제공하고 클라이언트에 의해 선택되기를 기다리는 상태로서 DHCPREQUEST와 DHCPDECLINE메시지를 처리한다. 서버의 상태천이도는 아래의 그림 4와 같다.



(그림 4) 서버의 상태천이도  
(Fig. 4) The state transition diagram of the client

4.3 서비스 프리미티브(메시지)

서비스 메시지는 크게 두 가지로 나눌 수 있다. 즉 클라이언트에서 서버로 보내는 메시지와 서버에서 클라이언트로 보내는 메시지가 있다. 클라이언트의 메시지는 DHCPDISCOVER, DHCPREQUEST, DHCPDECLINE, DHCPRELEASE가 있고 서버의 메시지는 DHCPPOFFER, DHCPACK, DHCPNAK 등이 있다. 위의 메시지들을 정리해보면 다음과 같다.

〈표 3〉 프리미티브의 파라미터  
(Table 3) The parameter of primitives

프리미티브	파라미터
DHCPDISCOVER	하드웨어 주소
DHCPREQUEST	하드웨어 주소, 서버 식별자
DHCPDECLINE	하드웨어 주소, 서버 식별자
DHCPRELEASE	하드웨어 주소, 서버 식별자
DHCPPOFFER	사용기간, 서버 식별자
DHCPACK	사용기간

5. 프로토콜 구현

5.1 구현환경

DHCP를 구현하기 위해 사용한 시스템은 Pentium PC(100Mhz)로서 시스템은 이더넷에 연결해서 LAN 환경에서 프로그램을 구현하고 테스트하였다. 운영체제로는 System V계열의 UNIX인 UNIXWARE 2.0을 사용했고 컴파일러는 UNIXWARE에서 제공하는 ANSI C 컴파일러를 사용하였다. UDP서비스를 이용하기 위한 전송계층 인터페이스로써 소켓(socket) 인터페이스를 이용하지 않고 System V계열의 표준인 TLI(Transport Layer Interface)를 사용하였다[8, 9, 10].

5.2 메시지의 유니캐스트 구현

5.2.1 메시지 구성

메시지 만들기 위한 자료구조는 구조체(struct)를 사용하여 메시지의 각 필드를 구조체의 필드로 선언하였다. 구조체의 필드는 아래의 그림 5와 같다.

```

struct dhcp {
    unsigned char dh_op;          /* packet opcode type */
    unsigned char dh_htype;      /* hardware addr type */
    unsigned char dh_hlen;       /* hardware addr length */
    unsigned char dh_hops;       /* gateway hops */
    unsigned long dh_xid;         /* transaction ID */
    unsigned short dh_secs;      /* second since boot began */
    unsigned short dh_flags;     /* RFC1532 broadcast, etc. */
    struct in_addr dh_ciaddr;     /* client IP address */
    struct in_addr dh_yiaddr;     /* 'your' IP address */
    struct in_addr dh_siaddr;     /* server IP address */
    struct in_addr dh_giaddr;     /* gateway IP address */
    unsigned char dh_chaddr[DH_CHADDR_LEN]; /* client hardware address */
    char dh_sname[DH_SNAME_LEN]; /* server host name */
    char dh_file[DH_FILE_LEN];   /* boot file name */
    unsigned char dh_vend[DH_VEND_LEN]; /* option area */
}
    
```

(그림 5) 메시지를 만들기 위한 구조체의 필드  
(Fig. 5) The message format of DHCP

IP 주소가 들어가는 필드는 IP 주소를 위해 TLI 라이브러리의 헤더파일에 미리 정의되어 있는 구조체인 "struct in\_addr"을 사용하였다.

"options" 필드를 제외한 필드는 구조체의 각 필드에 값을 넣어주면 된다. 하지만 "options" 필드는 어떤 위치에 어떤 필드의 값이 들어가도록 고정되어 있지 않다. "options" 필드의 시작부분은 4옥테트의 "magic cookie"(RFC 951에서 제안됨)라고 불리는 "99.130.83.99"로 채워진다.

"magic cookie" 다음에는 넣고자 하는 옵션의 코

드, 옵션 길이, 옵션의 실제 값 순으로 해서 "options" 필드에 network byte order로 그 값을 복사해서 넣어 준다. 더 이상 넣어줄 옵션이 없을 때는 옵션의 끝을 알리는 "End Option"인 "255"라는 옵션코드만을 넣어주고 "options" 필드의 나머지 부분은 "0"으로 채운다[12].

### 5.2.2 메시지의 브로드캐스팅

우선 시스템에 존재하는 전송계층 프로토콜을 알아내고 주어진 전송계층 프로토콜상에서 이용하고자 하는 서비스의 전송 계층 주소를 알아내기 위하여 TLI에서 제공하는 네트워크 선택 메카니즘과 Name-to-Address사상(寫像) 루틴을 사용하였다.

네트워크 선택 메카니즘은 어떤 전송계층 프로토콜이 존재하는지 알려주며 전송계층 프로토콜이 가상회선이나 데이터그램 서비스를 제공하는지 알려주고 전송계층 프로토콜을 접근하기 위해 사용되는 디바이스 파일을 보여준다. Name-to-Address사상 루틴은 전송계층에 독립적인 방식으로 네트워크 서비스의 전송계층 주소를 알 수 있게 한다. 메시지를 브로드캐스트하기 위해서는 특정한 호스트이름을 지정하는 것이 아니고 미리 정의된 특수한 값을 호스트 이름으로 해준다. 브로드캐스팅을 위한 특수한 값을 "HOST\_BROADCAST"이다. "HOST\_BROADCAST"라는 값은 UDP프로토콜이 사용하는 255.255.255.255라는 특수한 주소로서 자신의 시스템에 연결된 네트워크상의 모든 시스템들을 의미한다. 브로드캐스팅을 위해 필요한 것은 전송계층 프로토콜이 브로드캐스팅을 지원하도록 초기화시켜주는 것이다. 이런 일을 하는 함수는 netdir\_options()로서 인수로 ND\_SET\_BROADCAST라는 값을 주어야 한다.

### 5.2.3 메시지의 유니캐스트

#### (1)DHCPREQUEST의 유니캐스트

서버에서 보낸 DHCPPOFFER메시지의 서버 IP 주소 필드(siaddr)안의 값을 가지고 서버의 주소를 알 수 있기 때문에 이 주소를 서버의 주소로하여 메시지를 보내면 된다. 단 이 때 DHCPREQUEST의 경우에는 선택되지 않은 서버에 선택되지 않았다는 것을 알리기 위하여 메시지를 보내야 한다. 사용기간 연장을 위한 DHCPREQUEST에서도 지정된 서버에게서 응

답이 없으면 다른 서버에게도 메시지를 보낸다.

#### (2)DHCPPOFFER, DHCPACK, DHCPNAK의 유니캐스트

클라이언트에서 DHCPDISCOVER메시지에 있는 클라이언트의 하드웨어 주소 필드의 값을 읽고 그 클라이언트에게 할당해 줄 IP 주소를 하나의 쌍으로 하여 서버의 ARP캐쉬에 항목을 만든다. ARP캐쉬에 항목을 만들기 위해서는 /dev/arp라는 디바이스 파일을 열어서 ioctl시스템 호출을 써야 하는데 이 경우 시스템 관리자의 권한이 필요하다. 이와 같이 ARP캐쉬 항목을 만듦으로써 클라이언트의 새로운 IP 주소를 하드웨어 주소에 바인딩할 수 있고 이 주소로 DHCPPOFFER, DHCPACK, DHCPNAK 메시지를 보내면 클라이언트는 자신의 주소가 정의되지 않은 상태에서 하드웨어 주소가 일치하는 메시지를 수신할 수 있다.

#### (3)REBINDING상태에서 DHCPREQUEST의 유니캐스트

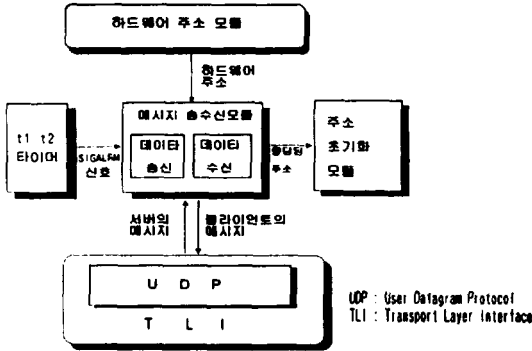
클라이언트는 IP 주소를 할당받아서 사용하다가 사용기간이 종료되면 사용기간의 갱신을 위해서 지정된 서버에게 DHCPREQUEST메시지를 보낸다. 이때 지정된 서버에서 응답이 없으면 DHCPREQUEST를 브로드캐스팅해야 한다. 이를 피하기 위해서는 초기화 과정에서 DHCPPOFFER메시지를 받을 때에 여러개의 서버로부터 메시지를 받았다면 그 서버들의 주소를 저장한다. 그리고 REBINDING상태에서 어느 한 서버로부터 응답이 있을 때까지 각각의 주소로 DHCPREQUEST메시지를 유니캐스트하면 브로드캐스팅을 할 때의 기능을 수행할 수 있다.

## 5.3 클라이언트의 구현

클라이언트를 각 모듈별로 설명하면 아래와 같다.

### 5.3.1 메시지 송수신 모듈(MTM)

서버의 위치를 알아내기 위하여 DHCPDISCOVER 메시지를 브로드캐스팅하고 서버로부터 DHCPPOFFER 메시지로 응답이 오면 이 메시지에서 제공한 주소를 DHCPREQUEST메시지로 요청한다. DHCPDISCOVER메시지의 브로드캐스팅 횟수를 줄이기 위해서 최초의 DHCPPOFFER메시지를 수신했을 때 부터는 송신을 중단하고 수신만 한다. 제공된 주소를 사용할



(그림 6) 클라이언트의 구성도  
(Fig. 6) The implementation diagram of the client

수 없는 경우에는 DHCPDECLINE 메시지를 보내고 더이상 주소를 사용할 필요가 없어진 경우에는 DHCPRELEASE 메시지로 반납한다.

알고리즘을 보면 아래의 그림 7과 같다.

```
Client_main()
{
    open the transport provier;
    while(1)
    {
        broadcast DHCPDISCOVER message
        until client receive the server's message:
        if(server's message is DHCPPOFFER)
            set up system with offered address:
        send DHCPREQUEST to the server
        until client receive the server's message:
        if(server's message is DHCPACK)
            call timer routine:
    }
}
```

(그림 7) 클라이언트의 알고리즘  
(Fig. 7) The algorithm of the client

메시지를 보내기 위해서 l\_sndudata 함수를 이용하고 수신을 위해서는 l\_rcvudata 함수를 사용한다. 함수의 인수로서 l\_unitdata 구조체의 addr 필드에는 송신자의 트랜스포트 주소가 들어가 있고 udata 필드에는 수신한 사용자 데이터가 들어가 있다.

5.3.2 타이머 모듈(Timer Module)

클라이언트는 T1, T2 두개의 시간을 유지한다. T1 과 T2는 클라이언트가 자신이 할당받은 주소의 사용 기간을 연장시킬 시간을 지정한다.

알고리즘은 아래의 그림 8과 같다.

타이머의 설정은 sigset() 함수와 alarm이라는 시스

```
timer()
{
    start:
    set t1 timer:
    while(1)
    {
        if(t1 timer expires)
        {
            set t2 timer:
            send DHCPREQUEST to the server
            until receive the server's message or timer expires:
            if(receive the server's DHCPACK message)
                goto start:
            else if(t2 timer expires)
            {
                send DHCPREQUEST to other server
                until receive the server's message or lease expires:
                if(receive the server's DHCPACK message)
                    goto start:
                else if(lease expires)
                    exit the while loop and return:
            }
        }
    } /* if */
} /* while */
```

(그림 8) 타이머의 알고리즘  
(Fig. 8) The algorithm of the timer

템 호출을 사용하였다[11].

sigset(SIGALRM, handler)  
alarm(T1)

BOUND 상태에서 DHCPREQUEST 메시지를 재전송할 때의 지연시간은 지수적 재설정 알고리즘(exponential backoff algorithm)을 쓴다. 지연시간은 처음에 4 + random(-1 to +1)로 시작해서 64초가 될 때까지 두배씩 증가하는데 이 지연시간의 합을 검사해서 T2 시간이 경과하면 REBINDING 상태가 된다. REBINDING 상태에서도 같은 방식을 지연시간의 합을 체크해서 사용기간이 경과하면 INIT 상태로 간다. 지연시간 역시 sigset() 함수와 alarm 시스템 호출을 이용해서 구현하였다.

5.3.3 주소 초기화 모듈(Address Configuration Module)

DHCPPOFFER에서 제공해 준 주소를 자신의 주소로 초기화시키는 일을 해주는 모듈이다. 시스템의 주소를 초기화 시켜주기 위해서는 우선 ifconfig 명령을 이용해서 주소를 설정해 주고 재부팅시에도 그 주소를 유지하려면 hosts 파일에 등록시켜 주면 된다.

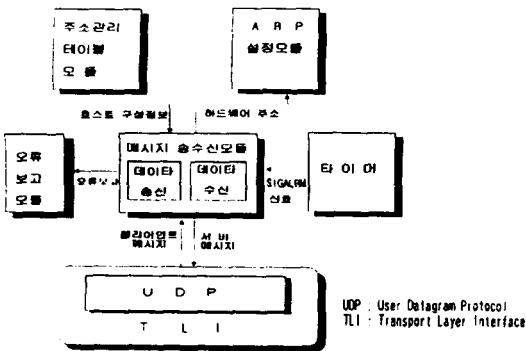
ifconfig 명령은 popen() 함수를 이용해서 프로그램 안에서 수행시킨다.

5.3.4 하드웨어 주소 모듈(Hardware Address Module)



클라이언트에서 자신의 하드웨어 주소를 얻어내기 위해서는 우선 이더넷을 위한 인터페이스를 알아내야 한다. Ioctl 시스템 호출을 이용하여 UNIXWARE 2.0상에서 이더넷 인터페이스를 알아보면 "TCM5X9 0"임을 알 수 있다. 그리고 이 인터페이스를 담당하는 디바이스 파일은 "/dev/TCM5X9\_0"이다. 이 파일을 열어서 시스템의 하드웨어 주소를 알아낼 수 있다.

5.4 서버의 구현



(그림 9) 서버의 구현

(Fig. 9) The implementation diagram of the server

5.4.1 메시지 송수신 모듈(MTM)

DHCPOFFER 메시지의 유니캐스트를 위해서 클라이언트의 IP 주소와 하드웨어 주소를 바인딩해서 서버의 ARP캐쉬에 항목을 만든다. 이를 위해서 "/dev/arp"라는 파일을 열어서 ioctl()시스템 호출을 사용해야 한다. 클라이언트에서 DHCPREQUEST 로 제공한 주소의 요청이 들어오면 DHCPACK 메시지를 보내어 할당과정을 종료한다. 클라이언트의 사용기간 요구에 대한 서버의 거절로 인한 무한루프의 가능성을 없애기 위해서 서버에서만 사용기간을 설정할 수 있도록 한다.

알고리즘을 보면 그림 10과 같다.

클라이언트와 마찬가지로 메시지를 송신하기 위해서는 TLI에서 제공하는 t\_sndudata()함수를 사용하고 수신을 위해서는 t\_rcvudata()함수를 사용한다.

5.4.2 오류보고 모듈(Error Report Module)

클라이언트에서 주소를 사용할 수 없는 경우에 보내는 DHCPDECLINE 메시지를 보내오면 서버에서

```

Server.main()
{
    open the transport provider.
    bind to the address the service will be offered over.
    while(1)
    {
        wait client's message until receive.
        if(message is DHCPDISCOVER)
        {
            make an entry in ARP cache.
            send DHCPOFFER with address to offer.
            wait client's message.
            if(message is DHCPREQUEST)
            send DHCPACK message with an address offered.
        }
        else if(message is DHCPREQUEST)
        {
            reset the lease for client.
            send DHCPACK message.
        }
        else if(message is DHCPRELEASE)
        {
            release the allocated address.
        }
    } /* end of while */
} /* end of main */
    
```

(그림 10) 서버의 알고리즘

(Fig. 10) The algorithm of the server

는 이를 해결하기 위해서 관리자에게 오류가 발생했음을 알리는 모듈이다. 관리자에게 적절한 오류 메시지를 보냄으로써 오류가 발생했음을 알린다.

5.4.3 타이머 모듈(Timer Module)

클라이언트는 자신이 더 이상 네트워크 단말로서 동작할 필요가 없는 경우 할당받은 주소를 반납하기 위해서 DHCPRELEASE 메시지를 보낸다. 하지만 DHCPRELEASE 메시지를 클라이언트가 반드시 보내야 하는 것은 아니므로 서버는 주소관리 테이블을 주기적으로 검사하여 사용기간이 어느 정도 남아있는지 알아야 하며 사용기간이 종료된 경우에는 클라이언트가 더 이상 그 주소를 사용하지 않는 것으로 알고 그 주소를 할당 안 된 것으로 해제할 수 있다. 주기적인 검사를 위해서 타이머 모듈에서는 sigset()함수와 alarm이라는 시스템 호출을 이용한다.

5.4.4 주소관리 테이블 모듈(Host Configuration Table Module)

클라이언트에게 할당해 줄 IP 주소를 관리하기 위하여 주소관리 테이블을 위한 모듈이 필요하다. 이를 위해서 그림 8에 나타난 주소관리 테이블의 레코드 구조를 이용하여 데이터를 저장했다.

데이터 파일에는 IP 주소와 이 주소를 할당받은 클라이언트의 여러 가지 정보가 들어간다. 여기서 플래

데이터 파일의 레코드 구조

IP주소 (16bytes)	플래그 (1byte)	호스트 이름 (30bytes)	하드웨어 주소 (16bytes)	서브네트 마스크 (16bytes)	사용 기간 (4bytes)
-------------------	----------------	---------------------	-------------------------	--------------------------	-------------------

인덱스 파일의 레코드 구조

IP주소 (16bytes)	데이터 레코드의 번호 (4 bytes)
-------------------	-----------------------------

(그림 11) 주소관리 테이블의 레코드 구조  
(Fig. 11) The record format of an IP address table

그 필드는 이 주소가 할당되었는지 안 되었는지를 표시하는 필드이다. 인덱스 파일에는 IP 주소와 이 주소가 포함된 데이터 파일의 레코드를 위치를 알기 위한 데이터 레코드의 순서 번호가 들어간다. IP 주소의 할당상태와 리스기간을 관리자가 보고자 할 때 IP 주소에 대해 순차적으로 볼 수 있도록 IP 주소를 인덱스로 하여 삭제와 삽입시에 인덱스 파일에 그 순서를 유지하도록 하는 인덱스 순차 파일로 구현하였다.

6. 결 론

IP 주소의 효율적인 관리와 네트워크 호스트들을 중앙에서 관리할 수 있도록 등장한 DHCP는 호스트에게 IP 주소나 구성파라미터 등을 할당하는 과정에서 일어나는 브로드캐스팅의 트래픽을 증가시킨다. 이러한 트래픽을 최소화하기 위해서 본 연구에서는 두가지 방안을 제시하고 이를 구현하였다. 첫번째 방안은 브로드캐스팅을 근본적으로 줄일 수 있도록 가능하면 메시지를 유니캐스트하자는 것으로 DHCPDISCOVER 메시지를 제외한 DHCPPOFFER, DHCPREQUEST, DHCPACK, DHCPNAK 메시지를 유니캐스트할 수 있도록 구현했다. 두번째 방안은 알고리즘을 개선하여 브로드캐스팅 메시지의 교환회수를 줄이고자 하는 것으로 DHCPPOFFER 메시지를 클라이언트측에서 선택할 때 가장 먼저 도착한 메시지를 선택하고 서버에서 DHCPPOFFER 메시지로 클라이언트에게 할당해준 주소는 표시를 함으로써 중복할당을 피할 수 있게 하였다. 또한 클라이언트가 사용기간을 요청할 경우, 이를 서버에서 거절할 때의 문제를 해결하기 위해서 서버에서만 사용기간을 설정할 수 있도록 하였다. 따라서 첫번째 방안대로 메시지를 유니캐스트했을 때에 클라이언트와 서버가 서로 메시지

를 교환할 수 있도록 구현하여 이를 메시지의 분석을 통해 확인하였고 두번째 방안에 의한 구현모델을 구현함으로써 제안된 모델이 제대로 동작할 수 있음을 입증하였다. 구현된 DHCP를 이용함으로써 브로드캐스팅으로 인한 네트워크의 부담을 최소화하면서 복잡해져가는 네트워크를 보다 효율적으로 관리할 수 있다. 본 연구에서는 트래픽을 최소화하기 위한 모델 설계와 구현을 하였고 앞으로는 효율적인 관리를 위하여 그래픽 사용자 인터페이스를 구현중에 있다.

참 고 문 헌

- [1] Douglas E. Comer, Internetworking with TCP/IP(Vol.1), Prentice Hall, pp.1-15, 1992.
- [2] Hans-Werner Braun, Peter S.Ford, Yakov Rekhter, "CIDR and the Evolution of the Internet Protocol", Proceedings of INET'93, pp.BBA-1-BBA-5, 1993.
- [3] Charles E. Perkins, Andrew Myles, David B. Johnson, "The Internet Mobile Host Protocol", Proceedings of INET'94/JENC5 pp.642-1-642-9, 1994.
- [4] W. Richard Stevens, TCP/IP Illustrated, Volume1, Addison Wesley, pp.169-178, 1994.
- [5] W.Wimer, "Clarifications and Extensions for the Bootstrap Protocol", RFC1532, 1993.
- [6] R.Droms, "Dynamic Host Configuration Protocol", RFC1541, 1993.
- [7] W. Richard Stevens, TCP/IP Illustrated, Volume2, Addison Wesley, pp.337-339, 1995.
- [8] Michael Padovano, "Networking Applications on UNIX system V Release 4", Prentice Hall, pp. 245-448, 1993.
- [9] Stephen A.Rago, UNIX SystemV Networking Programming, Addison-Wesley, pp.149-259, 1993.
- [10] 김종우, 강현중, 정진욱, "효율적인 네트워크 관리 시스템을 위한 프로토콜 의 설계 및 구현", '94 정보처리학회 추계 학술발표 논문집, pp.476-479, 1994.
- [11] 정진욱, 안성진, UNIX 프로그래밍 기술, (주)컴퓨터 출판, pp.345-396, 1995.

[12] S.Alexander, R.Droms, "DHCP Options and BOOTP Vendor Extensions", RFC1533, 1993.



**안 성 진**

1988년 성균관대학교 정보공학과 졸업(학사)  
1990년 성균관대학교 대학원 정보공학과 졸업(석사)  
1990년~1995년 시스템공학연구소 연구원  
1995년~현재 성균관대학교 정보공학과 박사과정

관심분야: 네트워크 관리, 트래픽분석, Unix 네트워킹

**정 진 욱**

1974년 성균관대학교 전자공학과 학사  
1979년 성균관대학교 전자공학과 석사  
1991년 서울대학교 계산통계학과 박사  
1982년~1985년 한국과학기술연구소 실장  
1981년~1982년 Racal Milgo Co. 객원연구원  
1985년~현재 성균관대학교 정보공학과 교수  
관심분야: 네트워크관리, 네트워크보안, 고속 및 무선 통신프로토콜



**함 영 환**

1994년 성균관대학교 정보공학과 졸업(학사)  
1996년 성균관대학교 대학원 정보공학과 졸업(석사)  
1996년~현재 시스템공학연구소 연구원

관심분야: 네트워크 관리, 네트워크 프로토콜



**이 춘 희**

1964년 성균관대학교 영어영문학과 졸업(학사)  
1985년 연세대학교 산업대학원 전산학과 졸업(석사)  
1993년 아주대학교 컴퓨터공학과 박사과정 수료  
1968년~현재 시스템공학연구소 책임연구원

관심분야: 운영체제, 분산처리, 시스템 및 네트워크 관리



**박 창 순**

1975년 서울대학교 응용수학과 졸업(학사)  
1992년 연세대학교 산업대학원 전산학과 졸업(학사)  
1992년~현재 충남대학교 전산학과 박사과정  
1979년~현재 시스템공학연구소 책임연구원

관심분야: 시스템 성능평가, 병렬처리

**김 해 진**

1983년 경북대학교 공과대학 컴퓨터공학과 졸업(학사)  
1995년 충남대학교 대학원 전산학과 졸업(석사)  
1983년~현재 한국전자통신연구소 시스템소프트웨어 연구실장

관심분야: 운영체제, 실시간처리, 고장감내