

---

# FreeBSD

---

Korea FreeBSD Users Group – <http://www.kr.FreeBSD.org/>

## Chapter 21 : PPP SLIP

<rick@rickinc.com> .

<nobreak@kr.FreeBSD.org> .

: 2004 8 7

: 2005 3 8

: 1.0

FreeBSD Documentation Project FreeBSD Handbook 2004 8

7 . FreeBSD Handbook ,  
, CVS Repository  
,  
,  
가  
가 . KFUG(Korea FreeBSD Users Group)  
KFUG

“FreeBSD Handbook” “FreeBSD Documentation Project” , “ FreeBSD  
” “ (Young-oak Lee)” .

Copyright © 1995, 1996, 1997, 1998, 1999, 2000, 2001, 2002, 2003, 2004 The FreeBSD Documentation Project.

## 21 장 PPP 와 SLIP

### 21.1 개요

FreeBSD 는 컴퓨터를 다른 컴퓨터와 연결하는 다양한 방법을 가지고 있다. 다이얼-업(dial-up) 모뎀으로 네트워크나 인터넷에 연결하려면 PPP 나 SLIP 를 사용해야 된다. 이번 장에서는 이런 모뎀 기반 통신 서비스 설정에 대해 자세히 다룬다.

이번 장을 읽고 다음과 같은 사항을 알 수 있다:

- 유저 PPP는 어떻게 설정하는가
- 커널 PPP는 어떻게 설정하는가
- PPPoE(PPP over Ethernet)는 어떻게 설정하는가
- PPPoA (PPP over ATM)는 어떻게 설정하는가
- SLIP 클라이언트와 서버는 어떻게 설정하는가

이번 장을 읽기 전에 아래 사항을 알고 있어야 한다:

- 기본적인 네트워크 용어를 이해하고 있어야 된다.
- 기본적인 다이얼-업 연결과 PPP나 SLIP에 대해 이해하고 있어야 된다.

유저 PPP 와 커널 PPP 의 가장 큰 차이점이 무엇인지 궁금할 것이다. 유저 PPP 프로세스는 커널 보다 userland 에서 데이터를 인바운드(inbound)하고 아웃바운드(outbound) 한다. 커널과 userland 사이의 데이터를 복사하는 것은 노력이 따르지만 더 많은 기능을 수행할 수 있다 - 다양한 PPP 실행. 유저 PPP 는 외부와 통신할 때 tun 장치를 사용하지만 커널 PPP 는 ppp 장치를 사용한다.

**Note:** 이번 장 전반에 걸쳐 `pppd` 처럼 다른 PPP 소프트웨어와 구분할 필요가 없다면 유저 PPP 는 단순히 `ppp` 로 표시한다. 그리고 따로 표시하지 않는다면 이 섹션에서 설명하는 모든 명령은 root 로 실행한다.

### 21.2 유저 PPP 사용

## 21.2.1 유저 PPP

### 21.2.1.1 전제 조건

이문서는 다음과 같은 사항으로 가정한다:

- PPP를 사용하여 연결하는 ISP 계정을 가지고 있다.
- 모뎀이나 다른 장치가 시스템에 설치되어 있고 ISP와 연결할 수 있도록 정확하게 설정되어 있다.
- ISP의 전화번호를 알고 있다.
- 로그인 이름과 패스워드(보통 유닉스 스타일 로그인과 패스워드 또는 PAP나 CHAP 로그인과 패스워드).
- 한 개 또는 한 개 이상의 네임 서버 IP주소. 보통 네임 서버로 사용하도록 ISP에서 두 개의 IP를 받았을 것이다. ISP에서 IP를 주지 않았다면 `ppp.conf`에서 `enable dns` 명령을 사용하여 PPP가 네임 서버를 설정하도록 한다. 이 기능은 ISP의 PPP가 DNS 감지(negotiation)를 지원하는가에 따른다.

다음 정보는 ISP에서 제공하겠지만 모두 필요한 것은 아니다:

- ISP의 게이트웨이 IP주소. 게이트웨이는 여러분이 연결하는 *default route*로 설정되는 머신이다. 이 정보가 없다면 연결됐을 때 ISP PPP 서버가 정확한 값을 제공하도록 만들 수 있다.

이 IP 번호는 PPP에 의해 *HISADDR*로 참조된다.

- 사용해야 될 넷 마스크. ISP에서 제공하지 않는다면 무난하게 255.255.255.255를 사용할 수 있다.
- ISP에서 고정 IP와 호스트 이름을 제공했다면 입력한다. 그렇지 않으면 적당해 보이는 값을 설정한다.

필요한 정보를 모른다면 없다면 ISP에 문의한다.

**Note:** 이 섹션 전반에 걸쳐 보여 주는 많은 예제의 설정파일 내용은 라인에 번호를 부여했다. 이들 번호는 설명을 돕기 위해 부여한 것이므로 실제 파일을 의미하지 않는다. 또한 탭과 공백 문자로 들여 쓰는 것이 중요하다.

### 21.2.1.2 PPP 장치 노드 생성

일반적인 상황에서 대부분의 유저는 오직 하나의 tun 장치(/dev/tun0)만 필요하다. tun0 는 아래에서 tun*N*으로 바뀔 것이다. *N*은 여러분의 시스템에 맞는 유닛 번호다.

devfs(5)가 활성화되지 않은 FreeBSD(FreeBSD 4.X 와 이전 버전)는 tun0 장치가 있는지 확인해야 된다(devfs(5)가 활성화되었다면 요청에 의해 장치 노드가 생성되기 때문에 확인할 필요는 없다).

tun0 장치가 정확하게 설정되었는지 확인하는 가장 쉬운 방법은 장치를 다시 만드는 것이다. 장치를 다시 만들기 위해 다음 명령을 따른다:

```
# cd /dev
# sh MAKEDEV tun0
```

커널에 16 개의 터널 장치가 필요하면 다음 명령을 사용하여 생성한다:

```
# cd /dev
# sh MAKEDEV tun15
```

### 21.2.1.3 자동 PPP 설정

ppp 와 pppd(커널 레벨의 PPP 기능)는 /etc/ppp 디렉터리에 있는 설정파일을 사용한다. 유저 ppp 예제는 /usr/share/examples/ppp/에서 찾을 수 있다.

ppp 를 설정할 때 수정해야 될 파일들은 요구 사항에 따라 다르다. 설정할 것은 ISP 가 고정 IP 를(다시 말해 IP 주소 하나를 주어서 항상 이 IP 만 사용한다) 할당하는지 유동 IP 를(다시 말해 ISP 에 연결할 때마다 IP 주소가 바뀐다) 할당하는지에 따라 다르다.

## [PPP 설정]

## 1. PPP 와 고정 IP 주소

/etc/ppp/ppp.conf 파일을 수정해야 된다. 이것은 아래 예제와 비슷하다.

**Note:** :로 끝나는 시작 라인을 제외하고 다른 라인들은 아래에서 보여 준 대로 스페이스 또는 탭 키를 사용하여 들여쓰기 해야 된다.

```
1  default:
2      set log Phase Chat LCP IPCP CCP tun command
3      ident user-ppp VERSION (built COMPILATIONDATE)
4      set device /dev/cuaa0
5      set speed 115200
6      set dial "ABORT BUSY ABORT NOWWsCARRIER TIMEOUT 5 W
7              W"W" AT OK-AT-OK ATE1Q0 OK WWdATDTWWT TIMEOUT 40
CONNECT"
8      set timeout 180
9      enable dns
10
11  provider:
12      set phone "(123) 456 7890"
13      set authname foo
14      set authkey bar
15      set login "TIMEOUT 10 W"W" W"W" gin:--gin: WWU word: WWP col: ppp"
16      set timeout 300
17      set ifaddr x.x.x.x y.y.y.y 255.255.255.255 0.0.0.0
18      add default HISADDR
```

라인 1:

default 엔트리 확인. 이 엔트리의 명령은 ppp 가 동작할 때 자동으로 실행된다.

라인 2:

로그인 매개변수 활성화. 설정이 정확하게 동작하면 너무 큰 로그 파일을 만들지 않도록 이 라인이 메시지를 요약한다.

```
set log phase tun
```

**라인 3:**

PPP 가 peer 에게 어떻게 인증을 받을 수 있는지 지시한다. 교신과 링크 설정에 문제가 생기면 PPP 는 직접 peer 에게 인증을 받고 peer 관리자가 이런 문제를 조사할 때 유용한 정보를 제공한다.

**라인 4:**

모뎀이 접속하고 있는 장치를 확인한다. COM1 은 /dev/cuaa0 이고 COM2 는 /dev/cuaa1 이다.

**라인 5:**

연결하려는 속도를 설정한다. 115200 이 동작하지 않으면(새로운 모뎀에서 논리적인 이유로) 38400 을 사용해 본다.

**라인 6 & 7:**

다이얼 문자열. 유저 PPP 는 chat(8) 프로그램과 비슷한 expect/send 구문을 사용한다. 이 언어의 특성에 대한 더 많은 정보는 매뉴얼 페이지를 참고한다.

이 명령은 읽기 쉽게 다음 라인에서 계속된다. ppp.conf 의 명령은 라인의 마지막 문자가 "W" 문자이면 다음 라인으로 계속된다.

**라인 8:**

링크의 idle 타임아웃을 설정한다. 180 초가 기본값이기 때문에 이 라인은 상당히 직선적이다.

**라인 9:**

로컬 리졸버 설정을 peer 가 확인하도록 PPP 가 요청하도록 지시한다. 로컬 네임서버를 운용 중이라면 이 라인을 주석처리 하거나 삭제한다.

**라인 10:**

읽기 쉽도록 빈 라인이다. 빈 라인은 PPP 가 무시한다.

**라인 11:**

"provider"라고 부르는 라인은 provider 엔트리를 확인한다. 이곳을 ISP 이름으로 변경할 수 있기 때문에 연결을 시작하기 위해 나중에 load ISP를 사용할 수 있다.

**라인 12:**

이 공급자의 전화번호를 설정한다. 여러 개의 전화번호는 콜론(:)이나 파이프 문자(|)로 나눠서 지정한다. 이 분리 방법의 차이는 ppp(8)에 나와있다. 요약해보면 번호를 교대로 사용하려면 콜론(:)을 사용한다. 첫 번째 번호를 처음으로 사용하고 첫 번째 번호가 실패해서 다른 번호를 사용하려면 파이프 문자를 사용한다. 위에서 본 것처럼 전체 전화번호를 설정하려면 인용부호를 사용한다. 전화번호에서 공백으로 들여쓰기를 했다면 인용부호(")로 둘러싸야 한다.

**라인 13 & 14:**

유저 이름과 패스워드 확인. 유닉스 스타일 로그인 프롬프트를 사용하여 연결할 때 이런 값은 set login 명령에 WU와 WP 변수로 참조된다. PAP 나 CHAP 로 연결하여 인증할 때 이런 값이 사용된다.

**라인 15:**

PAP 나 CHAP 를 사용한다면 여기서 로그인할 수 없기 때문에 이 라인을 주석처리 하거나 삭제한다. 더 자세한 사항은 PAP 와 CHAP 인증을 참고한다.

로그인 문자열은 다이얼 문자열과 같은 chat 와 비슷한 구문이다. 이 예제에서 이 문자열은 다음과 같은 로그인 세션을 가진 유저를 위해 동작한다:

```
J. Random Provider
login: foo
password: bar
protocol: ppp
```

필요에 따라 이 스크립트를 수정한다. 처음으로 이 스크립트를 작성할 때 "chat" 로그를 활성화해서 대화가 제대로 되는지 확인할 수 있다.

**라인 16:**

연결의 기본 idle 타임아웃(초단위로)을 설정한다. 여기서 연결은 동작이 없이 300 초가 지나면 자동으로 끊어진다. 타임아웃을 원치 않으면 이 값을 0 으로 설정하거나 `-ddial` 명령 라인으로 변경한다.

**라인 17:**

인터페이스 주소 설정. 문자열 `x.x.x.x`는 ISP 에서 공급하는 IP 주소로 변경한다. 문자열 `y.y.y.y`은 ISP 의 게이트웨이(여러분이 연결하는 머신) IP 주소로 변경한다. ISP 에서 게이트웨이 주소를 제공하지 않았다면 `10.0.0.2/0` 을 사용한다. "추측해서"

주소를 사용해야 된다면 유동 IP 주소를 사용한 PPP 에서 지시한대로 /etc/ppp/ppp.linkup 에 엔트리를 생성한다. 이 라인이 생략됐다면 PPP 는 *-auto* 모드로 동작하지 않는다.

**라인 18:**

기본 경로로 ISP 의 게이트웨이를 추가한다. 특수한 단어 *HISADDR*은 라인 9 에서 지정한 게이트웨이 주소로 대체된다. 9 라인 이후에 이 라인이 없으면 *HISADDR*은 초기화되지 않기 때문에 중요하다.

ppp 를 *-auto* 로 사용하지 않으려면 이 라인을 ppp.linkup 파일로 이동시킨다.

고정 IP 로 ppp 가 *-auto* 모드에서 동작하면 연결하기 전에 라우팅 테이블 엔트리가 정확하게 설정되기 때문에 ppp.linkup 에 엔트리를 추가할 필요가 없다. 그러나 연결 후에 프로그램을 실행하기 위해 엔트리를 생성할 수 있다. 이것은 나중에 sendmail 예제에서 설명한다.

예제 설정파일은 /usr/share/examples/ppp/ 디렉터리에서 찾을 수 있다.

## 2. PPP 와 유동 IP 주소

인터넷 서비스 공급자가 고정 IP 주소를 할당하지 않는다면 로컬과 원격 주소 감지(negotiate)에 ppp 를 설정할 수 있다. "추측한" IP 주소를 설정하여 연결 후에 ppp 가 IP 설정 프로토콜(IPCP)을 사용하여 정확하게 설정하도록 한다. ppp.conf 설정은 고정 IP 를 사용한 PPP 와 같으므로 다음과 같이 변경한다:

```
17      set ifaddr 10.0.0.1/0 10.0.0.2/0 255.255.255.255
```

레퍼런스이기 때문에 역시 라인 번호를 포함하지 않는다. 최소한 한 칸 이상의 공백으로 들여 쓰기 해야 된다.

**라인 17:**

/ 문자 이후의 숫자는 ppp 가 사용하려는 주소비트 번호다. 상황에 맞게 정확한 IP 주소를 사용하려고 하겠지만 위의 예제는 항상 동작한다.

마지막 인자(0.0.0.0)는 PPP 에게 10.0.0.1 이 아닌 0.0.0.0 을 사용하여 교신을 시작하도록 하고 어떤 ISP 들에게 필요하다. PPP 가 *-auto* 모드에서 초기 경로를

설정하지 못하므로 `set ifaddr` 명령의 첫 번째 인자로 `0.0.0.0`을 지정하지 않는다.

`-auto` 모드에서 사용하지 않는다면 연결이 된 후에 사용되는 `/etc/ppp/ppp.linkup` 에 엔트리를 생성해야 된다. 여기서 `ppp` 가 인터페이스 주소에 할당되면 이제 라우팅 테이블 엔트리를 추가하는 것도 가능하다:

```
1 provider:
2 add default HISADDR
```

**라인 1:**

연결을 설정할 때 `ppp` 는 다음과 같은 규정에 따라 `ppp.linkup` 에서 엔트리를 찾는다: 첫째 `ppp.conf` 에 사용한 라벨과 같은 대응되는 라벨이 있는지 찾는다. 검색에 실패하면 게이트웨이 IP 주소 엔트리를 찾는다. 이 엔트리는 4 개의 8 진수 IP 스타일 라벨이다. 그래도 엔트리를 찾지 못하면 `MYADDR` 엔트리를 찾는다.

**라인 2:**

이 라인은 `HISADDR`가 지정하는 경로를 기본 경로로 추가하도록 `ppp` 에게 지시한다. `HISADDR`은 IPCP 에서 결정된 게이트웨이 IP 주소로 대체된다.

더 자세한 예제는 파일 `/usr/share/examples/ppp/ppp.conf.sample` 과 `/usr/share/examples/ppp/ppp.linkup.sample` 에서 `pmdemand` 엔트리를 본다.

**3. 입력 콜 받기**

LAN 에 연결된 머신에서 입력 콜을 받도록 `ppp` 를 설정할 때 패킷을 LAN 으로 포워드 할 것인지 결정해야 된다. 포워드 하겠다면 LAN 서브넷의 IP 주소를 `peer` 에게 할당하고 `/etc/ppp/ppp.conf` 파일에서 `enable proxy` 명령을 사용한다. 또한 `/etc/rc.conf` 파일에 다음 내용이 있는지 확인한다:

```
gateway_enable="YES"
```

**4. 어떤 getty 인가?**

FreeBSD 를 다이얼-업 서비스로 설정하기 섹션에서 `getty(8)`를 사용한 다이얼-업 서비스 활성화를 설명한다.

또 다른 `getty`는 다이얼-업을 위해 개선된 버전인 `mgetty`(<http://www.leo.org/~doering/mgetty/index.html>)다.

`mgetty`를 사용하는 장점은 `mgetty`가 모뎀과 활발히 통신한다. 이 의미는 `/etc/ttys`에서 포트가 꺼져있다면 모뎀은 응답하지 않는다.

최근 `mgetty` 버전(0.99 베타 버전부터)은 클라이언트 스크립트가 서버에 접근하는 횟수를 줄일 수 있도록 `ppp` 흐름을 자동으로 감지한다.

`mgetty`에 대한 더 많은 정보는 `Mgetty`와 `Autoppp`를 참고한다.

## 5. ppp 퍼미션

`ppp` 명령은 보통 `root` 유저에서 실행한다. 그러나 아래에서 설명하듯이 `ppp`를 서버 모드에서 일반 유저로 실행하려면 `/etc/group`의 `network` 그룹에 유저를 추가하여 `ppp`를 실행할 수 있는 권한을 할당한다.

그리고 `allow` 명령으로 유저들이 설정파일에서 하나 이상의 섹션에 접근할 수 있도록 해야 된다:

```
allow users fred mary
```

이 명령이 `default` 섹션에서 사용된다면 지정된 유저가 모든 것에 접근할 수 있다.

## 6. 유동 IP 유저를 위한 PPP 셸

다음 내용으로 `/etc/ppp/ppp-shell` 파일을 만든다:

```
#!/bin/sh
IDENT=`echo $0 | sed -e 's/^.*-W(.*)$/W1/'`
CALLEDAS="$IDENT"
TTY=`tty`

if [ x$IDENT = xdialup ]; then
    IDENT=`basename $TTY`
fi

echo "PPP for $CALLEDAS on $TTY"
echo "Starting PPP for $IDENT"

exec /usr/sbin/ppp -direct $IDENT
```

이 스크립트는 실행할 수 있어야 되고 다음 명령으로 ppp-dialup 이라는 심볼릭 링크를 만든다:

```
# ln -s ppp-shell /etc/ppp/ppp-dialup
```

그리고 모든 다이얼-업 유저 *살로* 이 스크립트를 이용한다. 다음은 /etc/passwd 에서 pchilds 라는(직접 패스워드 파일을 수정하지 않고 vipw 를 사용한다) 유저의 다이얼-업 PPP 예제다.

```
pchilds:*:1011:300:Peter Childs PPP:/home/ppp:/etc/ppp/ppp-dialup
```

다음과 같이 누구나 읽을 수 있는 0 바이트 파일을 가진 /home/ppp 디렉터리를 생성한다:

```
-r--r--r--  1 root   wheel      0 May 27 02:23 .hushlogin
-r--r--r--  1 root   wheel      0 May 27 02:22 .rhosts
```

## 7. 고정 IP 유저를 위한 PPP 셸

위에서처럼 ppp-shell 파일을 생성하고 고정 IP 를 할당 받은 각 계정의 ppp-shell 로 심볼릭 링크를 생성한다.

예를 들어 fred, sam 과 mary 3 명의 다이얼-업 고객이 있고 C 클래스 네트워크를 할당하려면 다음과 같이 입력한다:

```
# ln -s /etc/ppp/ppp-shell /etc/ppp/ppp-fred
# ln -s /etc/ppp/ppp-shell /etc/ppp/ppp-sam
# ln -s /etc/ppp/ppp-shell /etc/ppp/ppp-mary
```

이러한 유저들의 다이얼-업 계정 쉘을 위에서 생성한 심볼릭 링크로 설정한다(예를 들면 mary 의 쉘은 /etc/ppp/ppp-mary 다).

## 8. 유동 IP 유저를 위한 ppp.conf 설정

/etc/ppp/ppp.conf 파일에 다음과 같은 라인이 필요하다:

```
default:
    set debug phase lcp chat
    set timeout 0

ttyd0:
    set ifaddr 203.14.100.1 203.14.100.20 255.255.255.255
    enable proxy

ttyd1:
    set ifaddr 203.14.100.1 203.14.100.21 255.255.255.255
    enable proxy
```

**Note:** 들여쓰기가 중요하다.

*default:* 섹션이 각 섹션에 로드 된다. /etc/ttys 에서 활성화시킨 각 다이얼-업 연결에 위의 *ttyd0:*와 비슷한 엔트리를 생성해야 된다. 각 라인은 유동 IP 유저를 위한 IP 주소 풀(pool)에서 유일한 IP 주소를 받아야 된다.

## 9. 고정 IP 유저를 위한 ppp.conf 설정

위의 샘플 /usr/share/examples/ppp/ppp.conf 의 내용에 따라 각각의 고정 IP 가 할당된 다이얼-업 유저에게 섹션을 추가해야 된다. fred, sam 과 mary 예제로 계속 설명한다.

```
fred:
    set ifaddr 203.14.100.1 203.14.101.1 255.255.255.255

sam:
    set ifaddr 203.14.100.1 203.14.102.1 255.255.255.255

mary:
    set ifaddr 203.14.100.1 203.14.103.1 255.255.255.255
```

필요하다면 /etc/ppp/ppp.linkup 파일에 각 고정 IP 유저에 대한 라우팅 정보도 있어야 된다. 아래 라인은 클라이언트의 ppp 링크를 경유하는 C 클래스 203.14.101.0 네트워크의 경로를 추가한다.

```
fred:
    add 203.14.101.0 netmask 255.255.255.0 HISADDR

sam:
    add 203.14.102.0 netmask 255.255.255.0 HISADDR

mary:
    add 203.14.103.0 netmask 255.255.255.0 HISADDR
```

## 10. mgetty 와 AutoPPP

*AUTO\_PPP* 옵션으로 mgetty 를 설정하고 컴파일하면 PPP 연결의 LCP(종료 패킷을 교환하여 링크를 끊는데 사용된다) 상태를 mgetty 가 감지하고 자동으로 ppp 쉘을 생성한다. 그러나 기본 로그인/패스워드 절차가 없다면 유저는 PAP 나 CHAP 인증을 사용해야 된다.

이 섹션은 유저가 *AUTO\_PPP* 옵션(v0.99 배타나 이후 버전)으로 mgetty 를 성공적으로 설정, 컴파일하고 설치 했다고 가정한다.

다음 내용이 /usr/local/etc/mgetty+sendfax/login.conf 파일에 필요하다:

```
/AutoPPP/ - - /etc/ppp/ppp-pap-dialup
```

이 설정은 감지된 PPP 연결에 mgetty 가 ppp-pap-dialup 스크립트를 실행하도록 한다.

다음 내용이 있는 /etc/ppp/ppp-pap-dialup 파일(이 파일은 실행할 수 있어야 된다)을 생성한다:

```
#!/bin/sh  
exec /usr/sbin/ppp -direct pap$IDENT
```

/etc/ttys 에서 활성화된 각 다이얼-업을 위해 /etc/ppp/ppp.conf 에 일치하는 엔트리를 생성한다. 이것은 다행히 위에서 정의한 것과 상호 존재한다.

```
pap:  
  enable pap  
  set ifaddr 203.14.100.1 203.14.100.20-203.14.100.40  
  enable proxy
```

이 방법으로 로그인한 각 유저는 /etc/ppp/ppp.secret 파일에 유저이름/패스워드가 필요하거나 /etc/password 파일에서 PAP 를 통해 유저가 인증을 받도록 다음 옵션을 추가해야 된다.

```
enable passwdauth
```

특정 유저에게 고정 IP 를 할당하려면 /etc/ppp/ppp.secret 의 세 번째 인자에 번호를 지정할 수 있다. 예제는 /usr/share/examples/ppp/ppp.secret.sample 를 본다.

## 11. MS 확장

요청에 따라 DNS 와 NetBIOS 네임 서버 주소를 할당하도록 PPP 를 설정할 수 있다.

PPP 버전 1.x 에서 이렇게 확장하려면 다음 라인을 /etc/ppp/ppp.conf 의 관련 섹션에 추가해야 된다.

```
enable msextd
set ns 203.14.100.1 203.14.100.2
set nbns 203.14.100.5
```

그리고 PPP 버전 2 이상에서는 다음 내용을 추가한다:

```
accept dnst
set dns 203.14.100.1 203.14.100.2
set nbns 203.14.100.5
```

이 설정은 클라이언트에게 첫 번째와 두 번째 네임서버 주소 그리고 NetBIOS 네임서버 호스트를 알려준다.

버전 2 와 그 이상에서 *set dns* 라인이 생략됐다면 PPP 는 /etc/resolv.conf 에 있는 네임서버 주소를 사용한다.

## 12. PAP 와 CHAP 인증

어떤 ISP 는 인증 시스템을 설치해서 여러분의 연결 인증에 PAP 나 CHAP 인증 메커니즘을 사용한다. 이 경우 연결이 되면 ISP 는 login: 프롬프트를 보여주지 않지만 즉시 PPP 통신이 시작된다.

PAP 는 패스워드를 평범한 텍스트로 보내서 CHAP 보다 안전성이 떨어지지만 시리얼 라인을 통해서만 전송하기 때문에 패스워드는 문제가 되지 않는다.

다시 PPP 와 고정 IP 주소 또는 PPP 와 유동 IP 주소 섹션을 참고해서 다음 내용을 수정한다:

```
13      set authname MyUserName
14      set authkey MyPassword
15      set login
```

라인 13:

이 라인에 여러분의 PAP/CHAP 유저 이름을 지정한다. *MyUserName* 에 정확한 값을 입력해야 된다.



/etc/ppp/ppp.conf 에 다음 라인을 추가하면 활성화될 것이다:

```
nat enable yes
```

다른 방법으로 PPP NAT 는 명령어 라인 옵션 `-nat`으로 활성화할 수 있다. 또한 기본적으로 활성화되도록 /etc/rc.conf 에 `ppp_nat`를 추가할 수 있다.

이 기능을 사용한다면 다음과 같이 입력되는 연결을 포워딩 하거나:

```
nat port tcp 10.0.0.2:ftp ftp
nat port tcp 10.0.0.2:http http
```

외부의 모든 입력을 막는 /etc/ppp/ppp.conf 에 유용한 옵션을 찾을 수 있을 것이다.

```
nat deny_incoming yes
```

### 21.2.1.5 마지막 시스템 설정

ppp 를 설정하였지만 실행하기 전에 해야 되는 몇 가지가 있다. /etc/rc.conf 파일 수정과 관련된 일이다.

이 파일의 상단에 `hostname=` 라인을 설정해야 된다 예:

```
hostname="foo.example.com"
```

여러분의 ISP 가 고정 IP 주소와 이름을 부여했다면 호스트 이름으로 이 이름이 적절하다.

`network_interfaces` 변수를 본다. 요청에 따라 시스템이 ISP 에 연결하도록 설정하려면 `tun0` 장치가 리스트에 추가되어 있는지 확인하고 그렇지 않으면 삭제한다.

```
network_interfaces="lo0 tun0"
ifconfig_tun0=
```

**Note:** *ifconfig\_tun0* 변수는 비어 있어야 하고 */etc/start\_if.tun0* 라는 파일을 생성해야 된다. 이 파일에는 다음 라인이 필요하다:

```
ppp -auto mysystem
```

이 스크립트는 네트워크를 설정할 때 실행되고 자동 모드에서 ppp 데몬이 시작한다. 이 머신이 LAN의 게이트웨이라면 *-alias* 스위치를 사용할 수 있다. 더 자세한 사항은 매뉴얼 페이지를 참고한다.

*/etc/rc.conf*의 다음 라인에서 라우터 프로그램을 *NO*로 설정한다:

```
router_enable="NO"
```

routed가 ppp로부터 생성된 기본 라우팅 테이블 엔트리를 삭제하기 때문에 routed 데몬을 시작하면 안 된다.

*sendmail\_flags* 라인에 *-q* 옵션을 포함하지 않는 것도 중요할 수 있다. 그렇지 않으면 sendmail이 주소를 확인하기 위해 머신이 아이얼-아웃하는 원인이 될 것이다. 다음과 같이 설정할 수 있다:

```
sendmail_flags="-bd"
```

이 설정의 단점은 ppp가 연결될 때마다 다음 명령으로 sendmail이 메일 큐를 다시 확인하도록 해야 된다:

```
# /usr/sbin/sendmail -q
```

자동으로 메일 큐를 확인하도록 ppp.linkup에 !bg 명령을 사용할 수 있다:

```
1 provider:
2 delete ALL
3 add 0 0 HISADDR
4 !bg sendmail -bd -q30m
```

이렇게 하지 않고 SMTP 트래픽을 막도록 "dfilter"를 설정하는 것도 가능하다. 더 자세한 사항은 샘플 파일을 참고한다.

이제 유일하게 남은 것은 머신을 재 부팅하는 일이다.

재 부팅 후 다음 명령을 입력하여 dial provider 로 PPP 세션을 시작하거나:

```
# ppp
```

외부와의 통신이 발생할 때(그리고 start\_if.tun0 스크립트를 생성하지 않았다면) ppp 세션이 자동으로 연결되기를 원한다면 다음 명령을 입력한다:

```
# ppp -auto provider
```

### 21.2.1.6 요약

ppp 를 처음 설정할 때 다음 단계를 따라야 한다:

클라이언트 쪽:

- ① tun 장치를 커널에 빌드한다.
- ② /dev 디렉터리에 tunN 장치 파일이 있어야 된다.
- ③ /etc/ppp/ppp.conf에 엔트리를 생성한다. pmdemand 예제는 대부분의 ISP 설정에 적당하다.
- ④ 유동 IP를 가지고 있다면 /etc/ppp/ppp.linkup에 엔트리를 생성한다.
- ⑤ /etc/rc.conf 파일을 업데이트한다.
- ⑥ 전화접속 요청이 필요하면 start\_if.tun0 스크립트를 생성한다.

서버 쪽:

- ① tun 장치를 커널에 빌드 한다.
- ② tunN 장치 파일을 /dev 디렉터리에서 사용할 수 있어야 한다.
- ③ /etc/passwd(vipw(8) 프로그램을 이용하여)에 엔트리를 생성한다.
- ④ ppp -direct direct-server나 비슷한 명령을 실행하는 프로필을 이 유저의 홈 디렉터리에 생성한다.
- ⑤ /etc/ppp/ppp.conf에 엔트리를 생성한다. direct-server 예제가 적당할 것이다.
- ⑥ /etc/ppp/ppp.linkup에 엔트리를 생성한다.
- ⑦ /etc/rc.conf 파일을 업데이트 한다.

## 21.3 커널 PPP 사용

### 21.3.1 커널 PPP 설정

머신에 PPP 를 설정하기 전에 pppd 가 /usr/sbin 에 있고 /etc/ppp 디렉터리가 있어야 된다.

pppd 는 두 가지 모드로 동작할 수 있다:

- ① "클라이언트" - PPP 시리얼 회선이나 모뎀 라인으로 머신을 외부와 연결하기를 원할 것이다.
- ② "서버" - 머신이 네트워크에 연결되어 있고 ppp를 사용하여 다른 컴퓨터와의 연결에 사용된다.

두 가지 경우에 옵션 파일을 설정해야 된다(etc/ppp/options 이나 머신에서 한 명 이상의 유저가 PPP 를 사용한다면 ~/.ppprc 를 설정한다).

그리고 원격 호스트에 전화 연결하여 통신할 수 있는 모뎀/시리얼

소프트웨어(comms/kermit이 좀더 적합하다)가 필요하다.

## 21.3.2 pppd 클라이언트로 사용

다음 /etc/ppp/options 은 시스코 PPP 라인 터미널 서버에 연결할 때 사용할 수 있을 것이다.

```
crtscts      # enable hardware flow control
modem        # modem control line
noipdefault  # remote PPP server must supply your IP address.
              # if the remote host does not send your IP during IPCP
              # negotiation, remove this option
passive      # wait for LCP packets
domain ppp.foo.com  # put your domain name here

:<remote_ip> # put the IP of remote PPP host here
              # it will be used to route packets via PPP link
              # if you didn't specified the noipdefault option
              # change this line to <local_ip>:<remote_ip>

defaultroute # put this if you want that PPP server will be your
              # default router
```

접속은 다음 단계를 따른다:

- ① **kermit**으로(또는 다른 모뎀 프로그램) 원격 호스트에 전화 연결해서 유저 이름과 패스워드를(또는 원격호스트에 PPP 연결에 필요한) 입력한다.
- ② **kermit**을 종료한다(접속 상태를 유지하고).
- ③ 다음 명령을 실행한다:

```
# /usr/src/usr.sbin/pppd.new/pppd /dev/tty01 19200
```

적당한 속도와 장치 이름을 사용한다.

이제 컴퓨터가 PPP 로 접속되었다. 접속이 실패했다면 /etc/ppp/options 파일에 *debug* 옵션을 추가할 수 있고 문제를 추적하는 메시지를 콘솔에서 체크할 수 있다.

다음의 /etc/ppp/pppup 스크립트는 3 단계 모두 자동으로 수행한다:

```
#!/bin/sh
ps ax |grep pppd |grep -v grep
pid=`ps ax |grep pppd |grep -v grep|awk '{print $1;}'`
if [ "X${pid}" != "X" ] ; then
    echo 'killing pppd, PID=' ${pid}
    kill ${pid}
fi
ps ax |grep kermit |grep -v grep
pid=`ps ax |grep kermit |grep -v grep|awk '{print $1;}'`
if [ "X${pid}" != "X" ] ; then
    echo 'killing kermit, PID=' ${pid}
    kill -9 ${pid}
fi

ifconfig ppp0 down
ifconfig ppp0 delete

kermit -y /etc/ppp/kermit.dial
pppd /dev/tty01 19200
```

/etc/ppp/kermit.dial 은 전화 연결해서 원격 호스트에 필요한 모든 인증을 받는 **kermit** 스크립트다(이와 같은 예제 스크립트는 이 문서의 마지막에 있다)

PPP 라인을 끊기 위해 다음 /etc/ppp/pppdown 스크립트를 사용한다:

```
#!/bin/sh
pid=`ps ax |grep pppd |grep -v grep|awk '{print $1;}'`
if [ X${pid} != "X" ] ; then
    echo 'killing pppd, PID=' ${pid}
    kill -TERM ${pid}
fi

ps ax |grep kermit |grep -v grep
pid=`ps ax |grep kermit |grep -v grep|awk '{print $1;}'`
if [ "X${pid}" != "X" ] ; then
    echo 'killing kermit, PID=' ${pid}
    kill -9 ${pid}
fi

/sbin/ifconfig ppp0 down
/sbin/ifconfig ppp0 delete
kermit -y /etc/ppp/kermit.hup
/etc/ppp/ppptest
```

다음과 비슷한 /usr/etc/ppp/ppptest 를 실행하여 PPP 가 아직 동작 중인지 체크한다:

```
#!/bin/sh
pid=`ps ax | grep pppd |grep -v grep|awk '{print $1;}'`
if [ X${pid} != "X" ] ; then
    echo 'pppd running: PID=' ${pid-NONE}
else
    echo 'No pppd running.'
fi
set -x
netstat -n -l ppp0
ifconfig ppp0
```

모뎀을 정지시키려면 다음 내용이 있는 /etc/ppp/kermit.hup 을 실행한다:

```
set line /dev/tty01 ; put your modem device here
set speed 19200
set file type binary
set file names literal
set win 8
set rec pack 1024
set send pack 1024
set block 3
set term bytesize 8
set command bytesize 8
set flow none

pau 1
out +++
inp 5 OK
out ATH0W13
echo W13
exit
```

여기 kermit 대신 chat 을 사용한 다른 방법이 있다.

다음 두 개의 파일이 pppd 에 접속하는데 충분하다.

/etc/ppp/options:

```
/dev/cuaa1 115200

rtscts      # enable hardware flow control
modem       # modem control line
connect "/usr/bin/chat -f /etc/ppp/login.chat.script"
noipdefault # remote PPP serve must supply your IP address.
            # if the remote host doesn't send your IP during
            # IPCP negotiation, remove this option
passive     # wait for LCP packets
domain <your.domain> # put your domain name here
```

```

:      # put the IP of remote PPP host here
      # it will be used to route packets via PPP link
      # if you didn't specified the noipdefault option
      # change this line to <local_ip>:<remote_ip>

defaultroute  # put this if you want that PPP server will be
              # your default router
    
```

/etc/ppp/login.chat.script:

**Note:** 다음은 한 라인이다.

```

ABORT BUSY ABORT 'NO CARRIER' "" AT OK ATDT<phone.number>
CONNECT "" TIMEOUT 10 ogin:wwwogin: <login-id>
TIMEOUT 5 sword: <password>
    
```

이 스크립트들을 정확히 수정해서 설치하고 이제 필요한 것은 다음과 같이 pppd 를 실행한다:

```
# pppd
```

### 21.3.3 pppd 서버로 사용

/etc/ppp/options 은 다음 내용과 비슷한 내용을 가지고 있다:

```

crtscts      # Hardware flow control
netmask 255.255.255.0  # netmask ( not required )
192.114.208.20:192.114.208.165  # ip's of local and remote hosts
                                # local ip must be different from one
                                # you assigned to the ethernet ( or other )
                                # interface on your machine.
                                # remote IP is ip address that will be
                                # assigned to the remote machine

domain ppp.foo.com      # your domain
    
```

```
passive          # wait for LCP
modem            # modem line
```

다음 /etc/ppp/pppserv 스크립트는 pppd 를 서버로 동작하게 한다:

```
#!/bin/sh
ps ax |grep pppd |grep -v grep
pid=`ps ax |grep pppd |grep -v grep|awk '{print $1;}'`
if [ "X${pid}" != "X" ] ; then
    echo 'killing pppd, PID=' ${pid}
    kill ${pid}
fi
ps ax |grep kermit |grep -v grep
pid=`ps ax |grep kermit |grep -v grep|awk '{print $1;}'`
if [ "X${pid}" != "X" ] ; then
    echo 'killing kermit, PID=' ${pid}
    kill -9 ${pid}
fi

# reset ppp interface
ifconfig ppp0 down
ifconfig ppp0 delete

# enable autoanswer mode
kermit -y /etc/ppp/kermit.ans

# run ppp
pppd /dev/tty01 19200
```

서버를 정지 시킬 때 /etc/ppp/pppservdown 스크립트를 사용한다:

```
#!/bin/sh
ps ax |grep pppd |grep -v grep
pid=`ps ax |grep pppd |grep -v grep|awk '{print $1;}'`
if [ "X${pid}" != "X" ] ; then
    echo 'killing pppd, PID=' ${pid}
    kill ${pid}
fi
ps ax |grep kermit |grep -v grep
pid=`ps ax |grep kermit |grep -v grep|awk '{print $1;}'`
if [ "X${pid}" != "X" ] ; then
    echo 'killing kermit, PID=' ${pid}
    kill -9 ${pid}
fi
ifconfig ppp0 down
ifconfig ppp0 delete

kermit -y /etc/ppp/kermit.noans
```

다음 **kermit** 스크립트(/etc/ppp/kermit.ans)는 모뎀의 자동응답 모드를 활성화/비활성 한다:

```
set line /dev/tty01
set speed 19200
set file type binary
set file names literal
set win 8
set rec pack 1024
set send pack 1024
set block 3
set term bytesize 8
set command bytesize 8
set flow none

pau 1
out +++
inp 5 OK
```

```
out ATH0W13
inp 5 OK
echo W13
out ATS0=1W13 ; change this to out ATS0=0W13 if you want to disable
                ; autoanswer mode inp 5 OK
echo W13
exit
```

/etc/ppp/kermit.dial 스크립트는 원격호스트로 전화 연결해서 인증 받는데 사용된다. 이 파일을 필요에 따라 수정해야 된다. 이 스크립트에 로그인 이름과 패스워드를 넣고 모뎀과 원격호스트의 응답에 따라 입력문을 변경해야 된다.

```

;
; put the com line attached to the modem here:
;
set line /dev/tty01
;
; put the modem speed here:
;
set speed 19200
set file type binary          ; full 8 bit file xfer
set file names literal
set win 8
set rec pack 1024
set send pack 1024
set block 3
set term bytesize 8
set command bytesize 8
set flow none
set modem hayes
set dial hangup off
set carrier auto             ; Then SET CARRIER if necessary,
set dial display on         ; Then SET DIAL if necessary,
set input echo on
set input timeout proceed
```

```
set input case ignore
def W%x 0 ; login prompt counter
goto slhup

:slcmd ; put the modem in command mode
echo Put the modem in command mode.
clear ; Clear unread characters from input buffer
pause 1
output +++ ; hayes escape sequence
input 1 OKW13W10 ; wait for OK
if success goto slhup
output W13
pause 1
output atW13
input 1 OKW13W10
if fail goto slcmd ; if modem doesn't answer OK, try again

:slhup ; hang up the phone
clear ; Clear unread characters from input buffer
pause 1
echo Hanging up the phone.
output ath0W13 ; hayes command for on hook
input 2 OKW13W10
if fail goto slcmd ; if no OK answer, put modem in command mode

:sdial ; dial the number
pause 1
echo Dialing.
output atdt9,550311W13W10 ; put phone number here
assign W%x 0 ; zero the time counter

:look
clear ; Clear unread characters from input buffer
increment W%x ; Count the seconds
input 1 {CONNECT }
if success goto sllogin
```

```
reinput 1 {NO CARRIERW13W10}
if success goto sldial
reinput 1 {NO DIALTONEW13W10}
if success goto slnodial
reinput 1 {W255}
if success goto slhup
reinput 1 {W127}
if success goto slhup
if < W%x 60 goto look
else goto slhup

:slogin                                ; login
assign W%x 0                          ; zero the time counter
pause 1
echo Looking for login prompt.

:sloop
increment W%x                          ; Count the seconds
clear                                   ; Clear unread characters from input buffer
output W13
;
; put your expected login prompt here:
;
input 1 {Username: }
if success goto sluid
reinput 1 {W255}
if success goto slhup
reinput 1 {W127}
if success goto slhup
if < W%x 10 goto slloop                ; try 10 times to get a login prompt
else goto slhup                       ; hang up and start again if 10 failures

:sluid
;
; put your userid here:
;
```

```
output ppp-loginW13
input 1 {Password: }
;
; put your password here:
;
output ppp-passwordW13
input 1 {Entering SLIP mode.}
echo
quit

:slnodial
echo W7No dialtone. Check the telephone line!W7
exit 1

; local variables:
; mode: csh
; comment-start: ";"
; comment-start-skip: ";"
; end:
```

## 21.4 PPP 접속 문제 해결

모뎀을 사용하여 접속할 때 사용자가 해결해야 될 여러 가지 문제가 있다. 이 섹션은 발생할 수 있는 몇 가지 문제에 대해 다룬다. 예를 들어 전화접속으로 시스템에 로그인할 때 어떤 프롬프트가 나타나는지 정확히 알고 있어야 될 필요가 있을 것이다. 어떤 ISP는 *ssword* 프롬프트를 보여주고 다른 ISP는 *password*를 보여 준다; ppp 스크립트를 정확히 작성하지 않았다면 로그인은 실패한다. ppp 연결을 디버그하는 가장 일반적인 방법은 직접 연결하는 것이다. 다음 정보는 직접 단계적으로 연결을 수행한다.

### 121.4.1 장치 노드 체크

커널을 다시 설정했다면 sio 장치를 다시 확인한다. 커널을 설정하지 않았다면 걱정할 필요가 없다. 단지 다음 명령으로 dmesg 출력에서 모뎀 장치를 체크한다:

```
#dmesg | grep sio
```

sio 장치에 관련된 적절한 결과를 보게 되고 이 결과는 우리에게 필요한 COM 포트다. 모뎀이 표준 시리얼 포트처럼 동작한다면 sio1 이나 COM2 에 나열된 것을 보게 된다. 그렇다면 커널을 다시 빌드 할 필요 없이 시리얼 장치를 만들면 된다. /dev 디렉터리로 이동하여 위에서처럼 MAKEDEV 스크립트를 실행하여 시리얼 장치를 만들 수 있다. 이제 다음 명령으로 시스템에 시리얼 장치를 만든다:

```
# sh MAKEDEV cuaa0 cuaa1 cuaa2 cuaa3
```

DOS 에서 sio 모뎀이 sio1 나 COM2 와 매칭되면 모뎀 장치는 /dev/cuaa1 이 된다.

## 21.4.2 직접 연결

직접 ppp 를 제어하여 인터넷에 연결하는 것은 쉽고 빠르며, 연결을 디버그 하는 가장 좋은 방법으로 ISP 에서 ppp 클라이언트 연결을 어떻게 처리하는지 정보를 얻게 된다. 명령어 라인에서 PPP 를 시작한다. 이곳의 모든 예제에서 PPP 가 동작하는 머신의 호스트

```
example . ppp ppp :
```

```
# ppp
```

이제 ppp 가 시작되었다.

```
ppp ON example> set device /dev/cuaa1
```

모뎀 장치를 설정하였고 여기서 모뎀 장치는 cuaa1 이다.

```
ppp ON example> set speed 115200
```

연결 속도 설정에서 이 경우 115,200 kbps 를 사용한다.

```
ppp ON example> enable dns
```

ppp 가 리졸버를 설정할 수 있도록 /etc/resolv.conf 에 네임서버 라인을 추가하도록 한다.

ppp 가 호스트 이름을 결정하지 못했다면 나중에 직접 설정할 수 있다.

```
ppp ON example> term
```

"터미널" 모드로 변경하여 직접 모뎀을 제어할 수 있다.

```
deflink: Entering terminal mode on /dev/cuaa1  
type '~h' for help
```

```
at
```

```
OK
```

```
atdt 123456789
```

at 을 사용하여 모뎀을 초기화하고 atdt 와 ISP 번호를 사용하여 전화 걸기를 시작하게 한다.

```
CONNECT
```

연결을 확인하고 하드웨어와 관련 없는 문제가 발생한다면 여기서 문제를 해결할 것이다.

```
ISP Login:myusername
```

유저 이름을 입력하는 프롬프트에 ISP 가 제공한 유저 이름을 넣고 Enter 를 누른다.

```
ISP Pass:mypassword
```

패스워드를 입력하는 프롬프트에 ISP 가 제공한 패스워드를 입력한다. FreeBSD 에 로그인 하는 것처럼 패스워드는 나타나지 않는다.

```
Shell or PPP:ppp
```

ISP 에 따라서 이 프롬프트는 절대 나타나지 않을 수 있다. 여기서 ISP 에게 셸 사용을 요청하거나 ppp 를 시작할 수 있다. 이 예제에서는 인터넷에 연결하려고 ppp 사용을 선택했다.

```
Ppp ON example>
```

이 예제에서 첫 번째 P 는 대문자임을 주의한다. 성공적으로 ISP 에 연결됐음을 보여준다.

PPp ON example>

성공적으로 ISP 인증을 거쳤고 IP 주소 할당을 기다린다.

PPP ON example>

IP 주소를 받았고 연결은 성공적으로 완료됐다.

PPP ON example>**add default HISADDR**

현재 peer 만 유일하게 연결되었기 때문에 외부와 연결하기 전에 기본 경로(라우트)를 추가해야 된다. 예전 경로 때문에 기본 경로를 추가하는데 실패했다면 *add* 앞에 ! 문자를 입력할 수 있다. 아니면 실제로 연결하기 전에 이렇게 설정하여 새로운 경로를 적절히 설정할 수 있다.

모든 것이 정상이면 **CTRL + z** 를 사용하여 백그라운드로 바꿀 수 있는 인터넷 연결이 활성화되고 PPP 가 ppp 로 바뀌면 연결은 끊어진다. 연결 상태를 보여주기 때문에 확인하는데 유용하다. 대문자 P 는 ISP 와 연결되었음을 소문자 p 는 어떤 이유에서건 연결이 끊어진 것을 보여준다. ppp 는 2 가지 상태만 있다.

## 21.4.2.1 디버깅

직접 연결은 했지만 연결이 안된 것처럼 보이면 *set ctsrts off* 로 하드웨어 흐름제어에 사용되는 CTS/RTS 를 끈다. 주된 이유는 데이터를 통신 링크에 보내려고 하면 PPP 가 멈추는 어떤 PPP 터미널 서버에 연결되어 있어서 절대 오지 않는 CTS 나 Clear To Send 신호를 기다리게 된다. 그러나 이 옵션을 사용한다면 끝 단에서 끝 단까지 특정 문자(XON/XOFF 등)를 보낼 때 하드웨어 의존성을 없애기 위해 필요할 수 있는 *set accmap* 옵션도 사용해야 된다. 이 옵션에 대한 정보와 사용법은 ppp(8) 매뉴얼 페이지를 참고한다.

오래된 모뎀을 가지고 있다면 *set parity even* 을 사용해야 될 것이다. 패리티가 기본적으로 none 으로 설정되므로 오래된 모뎀과 몇몇 ISP 의 에러 체크에 사용된다.

ISP 가 교신을 시작하기 위해 여러분 측의 준비를 기다리는 동안 PPP 는 보통 교신 에러가 있는 명령어 모드를 되돌려 주지 않을 것이다. 여기서 ~p 명령을 사용하면 강제로 ppp 가 설정 정보를 보낸다.

로그인 프롬프트를 보지 못했다면 위의 예제에서 유닉스 스타일의 인증 대신 PAP 나 CHAP 인증을 사용해야 된다. PAP 나 CHAP 를 사용하려면 터미널 모드로 들어가기 전에 PPP 에게 다음 옵션을 추가한다:

```
ppp ON example> set authname myusername
```

*myusername* 은 ISP 에서 할당해준 유저 이름으로 대체한다.

```
ppp ON example> set authkey mypassword
```

*mypassword* 는 ISP 에게 할당 받은 패스워드로 대체한다.

연결은 잘 되었지만 도메인 이름을 찾지 못하는 것처럼 보인다면 IP 주소로 ping(8)을 사용하여 결과를 확인한다. 100% 패킷을 잃어버린다면 대부분 기본 경로를 할당하지 않았을 것이다. 연결되어 있는 동안 옵션 *add default HISADDR* 가 설정되어 있는지 다시 체크한다. IP 주소로 연결할 수 있다면 리졸버 주소가 */etc/resolve.conf* 에 추가되지 않았을 것이다. 이 파일은 다음과 비슷하다:

```
domain example.com
nameserver x.x.x.x
nameserver y.y.y.y
```

*x.x.x.x* 와 *y.y.y.y* 는 ISP 의 DNS 서버 IP 주소로 대체한다. 이 정보는 ISP 와 계약할 때 제공받았거나 그렇지 않았을 수 있지만 ISP 와 통화하여 해결한다.

syslog(3)으로 PPP 연결에 로그 기능을 제공할 수 있다. 그냥 다음 라인을 */etc/syslog.conf* 에 추가한다:

```
!ppp
*.*/var/log/ppp.log
```

대부분 이 기능이 이미 설정되어 있다.

## 21.5 PPP over Ethernet(PPPoE) 사용

이번 섹션은 PPP over Ethernet(PPPoE)를 어떻게 설정하는지 설명한다.

### 21.5.1 커널 설정

PPPoE 를 위해 특별히 커널을 설정할 필요는 없다. **netgraph** 지원이 필요하다면 커널에 빌드하지 않고 **ppp** 를 통해 동적으로 로드 한다.

**Netgraph:** Netgraph 는 잘 정의된 각각의 태스크를 수행하도록 디자인된 툴들을 조합하여 강력하고 유연한 네트워킹 기능을 수행하는 in-kernel 네트워킹 서브시스템이다.

### 21.5.2 ppp.conf 설정

여기에 정확히 동작하는 ppp.conf 의 예제가 있다:

```
default:
  set log Phase tun command # you can add more detailed logging if you wish
  set ifaddr 10.0.0.1/0 10.0.0.2/0

name_of_service_provider:
  set device PPPoE:xl1 # replace xl1 with your ethernet device
  set authname YOURLOGINNAME
  set authkey YOURPASSWORD
  set dial
  set login
  add default HISADDR
```

### 21.5.3 PPP 실행

root 에서 다음 명령을 실행한다:

```
# ppp -ddial name_of_service_provider
```

## 21.5.4 부팅할 때 PPP 시작하기

다음 내용을 /etc/rc.conf 파일에 추가하다:

```
ppp_enable="YES"
ppp_mode="ddial"
ppp_nat="YES" # if you want to enable nat for your local network, otherwise NO
ppp_profile="name_of_service_provider"
```

## 21.5.5 PPPoE 서비스 태그 사용

연결하기 위해 서비스 태그(service tag)를 사용해야 될 경우가 종종 있다. 서비스 태그는 네트워크에 있는 다른 PPPoE 서버들을 구분하는데 사용된다.

ISP 가 제공한 문서에서 필요한 서비스 태그 정보를 받는다. 이 정보가 없다면 ISP 의 기술 지원을 받는다.

마지막으로 포트 컬렉션에서 찾을 수 있는 Roaring Penguin PPPoE

프로그램(<http://www.roaringpenguin.com/pppoe/>)이 제시하는 방법을 시도해 볼 수 있다.

그러나 이 방법은 모뎀 프로그램을 사용하지 못하게 될 수 있기 때문에 시도하기 전에 다시 생각해 본다. 그냥 여러분의 ISP가 제공한 모뎀에 맞는 프로그램을 설치한다. 그리고 프로그램에서 System 메뉴에 접근한다. 여러분의 프로필 이름에 보통 *ISP*라고 설정되어 있다.

프로필 이름(서비스 태그)은 set device 명령(더욱 자세한 내용은 ppp(8) 매뉴얼 페이지를 본다)의 공급자 부분으로 ppp.conf 의 PPPoE 설정 엔트리에 사용된다. 다음과 비슷하다:

```
set device PPPoE:x11:ISP
```

*x11* 을 여러분의 이더넷 카드에 맞게 변경하는 것을 잊지 않는다.

/ISP를 위에서 찾은 프로필로 변경하는 것을 잊지 않는다.

추가적인 정보는 아래 사이트를 참고한다:

- Renaud Waldura의 Cheaper Broadband with FreeBSD on DSL(<http://renaud.waldura.com/doc/freebsd/pppoe/>).
- Udo Erdelhoff (독일인) Nutzung von T-DSL und T-Online mit FreeBSD(<http://www.ruhr.de/home/nathan/FreeBSD/tdsl-freebsd.html>).

## 21.5.6 3Com® HomeConnect® ADSL Modem Dual

### Link 로 PPPoE 설정하기

이 모뎀은 RFC 2515(<http://www.faqs.org/rfcs/rfc2516.html>) (L.Mamakos, K.Lidl, J.Evarts, D.Carrel, D.Simone 그리고 R.Wheeler)이 작성한 PPP over Ethernet(PPPoE)을 따르지 않는다. 대신 이더넷 프레임에 다른 패킷 타입 코드를 사용한다. 이 모뎀이 PPPoE 규정을 따라야 한다고 생각하면 3Com에(<http://www.3com.com/>) 항의한다.

FreeBSD가 이 장치와 통신하도록 하려면 `sysctl` 을 설정해야 된다. `sysctl` 설정은 `/etc/sysctl.conf` 를 수정하여 부팅할 때 자동으로 수행한다:

```
net.graph.nonstandard_pppoe=1
```

아니면 `sysctl net.graph.nonstandard_pppoe=1` 명령을 사용하여 즉시 결과를 알 수 있다.

시스템 전반에 걸친 설정이기 때문에 일반적인 PPPoE 클라이언트나 서버를 3Com® HomeConnect® ADSL Modem 과 함께 여기서 설명하는 것은 불가능하다.

## 21.6 PPP 를 ATM 으로 사용 (PPPoA)

이 섹션은 PPP over ATM(PPPoA)을 어떻게 설정하는지 다룬다. PPPoA 는 유럽의 DSL

공급자들이 흔히 사용한다.

## 21.6.1 Alcatel SpeedTouch™ USB 로 PPPoA 사용하기

펌웨어가 Alcatel의 라이선스([http://www.alcatel.com/consumer/dsl/disclaimer\\_ix.htm](http://www.alcatel.com/consumer/dsl/disclaimer_ix.htm))로 배포되어서 FreeBSD의 기본 시스템에 무료로 재 배포할 수 없기 때문에 이 장치를 지원하는 PPPoA는 FreeBSD 포트에서 공급한다.

소프트웨어를 설치하려면 단순히 포트 컬렉션을 이용한다. [net/pppoa](#) 포트를 설치하고 제공된 지시를 따른다.

많은 USB 장치처럼 Alcatel SpeedTouch™ USB 도 호스트 컴퓨터에서 운영에 맞는 펌웨어를 다운로드 해야 된다. FreeBSD 에서 자동으로 다운로드 할 수 있기 때문에 장치가 USB 포트에 연결될 때마다 펌웨어는 자동으로 로드 된다. 자동으로 펌웨어를 전송하도록 다음 정보를 `/etc/usbd.conf` 파일에 추가한다. 이 파일은 root 유저에서 수정해야 된다.

```
device "Alcatel SpeedTouch USB"
  devname "ugen[0-9]+"
```

```
  vendor 0x06b9
```

```
  product 0x4061
```

```
  attach "/usr/local/sbin/modem_run -f /usr/local/libdata/mgmt.o"
```

USB 데몬 `usbd` 를 활성화하려면 다음 라인을 `/etc/rc.conf` 에 추가한다:

```
usbd_enable="YES"
```

시작할 때 PPP 가 다이얼-업 하도록 설정하는 것도 가능하다. 다이얼-업 하도록 설정하려면 다음 라인을 root 유저로 `/etc/rc.conf` 에 추가한다.

```
ppp_enable="YES"
```

```
ppp_mode="ddial"
```

```
ppp_profile="adsl"
```

정확하게 동작하도록 [net/pppoa](#) 포트에서 제공하는 샘플 `ppp.conf`를 사용한다.

## 21.6.2 mpd 사용

다양한 서비스(특히 PPTP)에 연결하기 위해 **mpd**를 사용할 수 있다. **mpd**는 포트 컬렉션의 [net/mpd](#)에서 찾을 수 있다. 다양한 ADSL 모뎀은 모뎀과 컴퓨터 사이에 생성된 PPTP 터널을 필요로 하고 이러한 모뎀 중 하나가 Alcatel SpeedTouch Home이다.

첫째로 포트를 설치하고 여러분의 필요성과 ISP의 설정에 따라 **mpd**를 설정한다. 포트의 PREFIX/etc/mpd/에 문서화가 잘 되어있는 샘플 설정파일 세트가 있다. PREFIX의 의미는 포트가 설치되어 있는 디렉터리를 의미하고 기본적으로 /usr/local/에 위치한다. 완벽한 **mpd** 설정 가이드는 포트가 설치된 후 PREFIX/share/mpd/에서 HTML 포맷으로 확인할 수 있다. 여기에 **mpd**로 ADSL 서비스에 연결하는 샘플 설정이 있다. 이 설정은 두 개의 파일로 나뉘어있고 첫 번째는 mpd.conf 이다:

```
default:
    load adsl

adsl:
    new -i ng0 adsl adsl
    set bundle authname username ❶
    set bundle password password ❷
    set bundle disable multilink

    set link no pap acfcomp protocomp
    set link disable chap
    set link accept chap
    set link keep-alive 30 10

    set ipcp no vjcomp
    set ipcp ranges 0.0.0.0/0 0.0.0.0/0

    set iface route default
    set iface disable on-demand
    set iface enable proxy-arp
    set iface idle 0
```

```
open
```

- ❶ ISP 인증에 사용되는 유저 이름.
- ❷ ISP 인증에 사용되는 패스워드.

mpd.links 파일은 연결을 원하는 링크에 대한 정보를 가지고 있다. 예제 mpd.links 는 위의 예제에 덧붙인 것이다.

```
adsl:  
  set link type pptp  
  set pptp mode active  
  set pptp enable originate outcall  
  set pptp self 10.0.0.1 ❶  
  set pptp peer 10.0.0.138 ❷
```

- ❶ mpd 를 사용할 FreeBSD 컴퓨터의 IP 주소
- ❷ ADSL 모뎀의 IP 주소. Alcatel SpeedTouch Home 의 주소는 기본적으로 10.0.0.138 이다.

root 에서 다음 명령으로 연결을 쉽게 초기화할 수 있다.

```
# mpd -b adsl
```

다음 명령으로 연결 상태를 볼 수 있다.

```
% ifconfig ng0  
ng0: flags=88d1<UP,POINTOPOINT,RUNNING,NOARP,SIMPLEX,MULTICAST> mtu 1500  
  inet 216.136.204.117 --> 204.152.186.171 netmask 0xffffffff
```

mpd 를 사용하여 FreeBSD 에서 ADSL 서비스에 연결하는 것을 권장한다.

## 21.6.3 pptpclient 사용

[net/pptpclient](#)를 사용하여 FreeBSD를 다른 PPPoA 서비스에 연결하는 것도 가능하다.

`net/pptpclient`를 사용하여 DSL 서비스에 연결하려면 포트나 패키지를 설치하고 `/etc/ppp/ppp.conf`를 수정한다. 이들 작업은 `root` 권한이 필요하다. `ppp.conf`의 예제 섹션은 아래에 있다. 더 많은 `ppp.conf` 옵션은 **ppp** 매뉴얼 페이지 `ppp(8)`를 참고한다.

```
adsl:
set log phase chat lcp ipcp ccp tun command
set timeout 0
enable dns
set authname username ❶
set authkey password ❷
set ifaddr 0 0
add default HISADDR
```

- ❶ DSL 공급자의 유저계정 이름
- ❷ 계정의 패스워드

**주의:** `ppp.conf` 파일에 평범한 텍스트 형식으로 계정의 패스워드를 입력해야 되므로 아무도 이 파일의 내용을 볼 수 없도록 해야 된다. 다음 명령들은 이 파일을 `root` 계정으로만 읽을 수 있도록 한다. `chmod(1)`과 `chown(8)`에 대한 더 많은 정보는 매뉴얼 페이지를 참고한다.

```
# chown root:wheel /etc/ppp/ppp.conf
# chmod 600 /etc/ppp/ppp.conf
```

이 설정은 DSL 라우터의 PPP 세션 터널을 연다. 이더넷 DSL 모뎀에는 여러분이 연결할 LAN IP가 미리 설정되어 있다. Alcatel SpeedTouch Home의 경우 주소는 10.0.0.138이다. 라우터 문서에서 장치가 어떤 IP를 사용하는지 확인한다. 터널을 열고 `ppp` 세션을 시작하려면 다음 명령을 실행한다.

```
# pptp address isp
```

**Tip:** `pptp`는 프롬프트를 돌려주지 않기 때문에 이전 명령의 마지막에 앰퍼샌드("&")를 추가할 수 있다.

`tun` 가상 터널 장치는 `pptp`와 `ppp` 프로세스 사이에서 생성된다. 프롬프트를 되돌려

받았거나 pptp 프로세스가 연결을 확인했다면 다음과 같이 터널을 확인할 수 있다.

```
% ifconfig tun0
tun0: flags=8051<UP,POINTOPOINT,RUNNING,MULTICAST> mtu 1500
      inet 216.136.204.21 --> 204.152.186.171 netmask 0xfffff00
      Opened by PID 918
```

연결할 수 없다면 보통 telnet 이나 웹 브라우저로 라우터의 설정을 체크한다. 그래도 연결할 수 없다면 pptp 명령의 출력과 ppp 로그 파일 /var/log/ppp.log 의 내용을 검토해본다.

## 21.7 SLIP 사용

### 21.7.1 SLIP 클라이언트 설정

다음은 FreeBSD 머신을 고정 호스트 네트워크에서 SLIP(serial Line Internet Protocol: TCP/IP 네트워크에 다이얼-업 접속을 할 수 있도록 해주는 데이터링크 프로토콜) 클라이언트로 설정하는 방법이다. 유동적인 호스트이름 할당에 대해서는(다이얼-업 할 때 마다 IP 주소가 변경되는) 더욱 복잡한 설정이 필요할 것이다.

우선 어떤 시리얼 포트에 모뎀을 연결할지 결정한다. 많은 사람들은 /dev/modem 과 같은 심볼릭 링크를 실제 이름 /dev/cuaaN 에 설정한다. 이 방법은 실제 장치 이름을 추상화하여 모뎀을 다른 포트로 이동시킬 필요가 없게 된다. 시스템의 /etc 파일들과 .kermrc 파일을 전체적으로 수정해야 될 때 상당히 귀찮은 일이 될 수 있다.

**Note:** /dev/cuaa0 는 COM1, cuaa1 은 COM2 등으로 대응된다.

다음 내용을 커널 설정파일에 추가한다:

```
pseudo-device  sl      1
```

FreeBSD 5.X 에서는 다음 라인을 대신 사용한다:

```
device sl
```

위 라인은 GENERIC 커널에 포함되어 있어서 이 내용을 지우지 않았다면 문제가 없다.

### 21.7.1.1 최초 설정

- ① 머신의 게이트웨이와 네임서버를 `/etc/hosts` 파일에 추가한다. 필자의 경우 `hosts` 파일은 다음과 같다:

```
127.0.0.1          localhost loghost
136.152.64.181    water.CS.Example.EDU water.CS water
136.152.64.1     inr-3.CS.Example.EDU inr-3 slip-gateway
128.32.136.9     ns1.Example.EDU ns1
128.32.136.12    ns2.Example.EDU ns2
```

- ② FreeBSD 5.0 전 버전에서는 `bind` 전에 `hosts`가 `/etc/host.conf` 파일에 나열되어 있어야 된다. FreeBSD 5.0부터 이 파일 대신 `/etc/nsswitch.conf` 파일을 대신 사용하므로 이 파일의 `dns` 전의 `hosts` 라인에 `files`가 있어야 된다. 이러한 매개변수가 없다면 재미있는 일이 발생할 것이다.

- ③ `/etc/rc.conf` 파일을 수정한다.

- A. 다음 라인을 수정하여 호스트 이름을 설정한다:

```
hostname="myname.my.domain"
```

머신의 완벽한 인터넷 호스트 이름이 이곳에 필요하다.

- B. 다음 라인을 변경하여 `sl0`를 네트워크 인터페이스 리스트에 추가한다:

```
network_interfaces="lo0" 라인을
network_interfaces="lo0 sl0"으로 변경한다.
```

- C. 다음 라인을 추가하여 sl0의 시작 플래그를 설정한다:

```
ifconfig_sl0="inet ${hostname} slip-gateway netmask 0xfffff00 up"
```

- D. 다음 라인을 수정하여 기본 라우터를 설정한다.

```
defaultrouter="NO" 라인을  
  
defaultrouter="slip-gateway"으로 수정한다.
```

- ④ 다음 내용을 포함하는 /etc/resolv.conf 파일을 생성한다.

```
domain CS.Example.EDU  
nameserver 128.32.136.9  
nameserver 128.32.136.12
```

이들 설정은 네임 서버 호스트 설정이다. 물론 실제 도메인 이름과 주소는 여러분의 환경을 따른다.

- ⑤ root와 toor 패스워드를 설정한다(그리고 패스워드가 없는 다른 계정들까지).  
⑥ 머신을 재 부팅하고 정확한 호스트 이름인지 확인한다.

### 21.7.1.2 SLIP 접속

- ① 프롬프트에 slip를 입력하여 다이얼-업 한 후 머신 이름과 패스워드를 입력한다. 필요한 사항은 여러분의 환경에 따라 다르다. kermitt를 사용한다면 다음과 같은 스크립트를 사용한다:

```
# kermitt setup  
set modem hayes  
set line /dev/modem  
set speed 115200
```

```
set parity none
set flow rts/cts
set terminal bytesize 8
set file type binary
# The next macro will dial up and login
define slip dial 643-9600, input 10 =>, if failure stop, -
output slip\x0d, input 10 Username:, if failure stop, -
output silvia\x0d, input 10 Password:, if failure stop, -
output ***\x0d, echo \x0aCONNECTED\x0a
```

물론 여러분의 환경에 맞게 호스트 이름과 패스워드를 변경한다. 이렇게 한 후 해 kermit 프롬프트에 slip 을 입력하여 연결할 수 있다.

**Note:** 파일시스템에 평범한 텍스트로 패스워드를 두는 것은 일반적으로 좋지 못한 생각이다. 위험 요소가 된다.

- ② kermit을 그 상태로 두고(**Ctrl-z**를 사용하여 중단할 수 있다) root에서 다음 명령을 입력한다:

```
# slattach -h -c -s 115200 /dev/modem
```

라우터 밖의 다른 호스트에 ping 이 나간다면 연결이 된 것이다. ping 이 나가지 않는다면 slattach 인자를 -c 대신 -a로 바꾸고 다시 시도한다.

### 21.7.1.3 연결을 어떻게 끊는가

slattach 를 중지하기 위해 다음 명령을 실행한다:

```
# kill -INT `cat /var/run/slattach.modem.pid`
```

위 명령은 root 에서 실행해야 된다. kermit 으로(중단 시켰다면 fg 를 실행하여) 되돌아가서 빠져나간다(q).

slattach 매뉴얼 페이지에서는 인터페이스를 다운시키는데 ifconfig sl0 down 을 사용하도록 하지만 이것은 여기서 알려 준 방법과 차이가 없어 보인다(ifconfig sl0 도 같은 내용을

출력한다).

종종 모뎀이 전송을 끊는 것을 거부한다(가끔). 이 경우 단순히 kermit 를 시작하고 다시 빠져나간다. 보통 두 번 정도하면 끊어진다.

#### 21.7.1.4 문제 해결

작동하지 않는다면 메일링 리스트에 문의한다. 이제까지 사람들이 보고한 내용:

- slattach에서 `-c`나 `-a`를 사용하지 않는다(이것은 치명적이지 않지만 어떤 유저는 자신들의 문제를 해결했다고 보고했다).
- `s10` 대신 `s100`을 사용했다(어떤 폰트에서는 차이점을 찾기가 어려울 것이다).
- 인터페이스 상태를 보기 위해 `ifconfig s10`를 사용한다. 예를 들어 다음과 같이 입력한다:

```
# ifconfig s10
s10: flags=10<POINTOPOINT>
    inet 136.152.64.181 --> 136.152.64.1 netmask ffffff00
```

- ping 테스트에서 "no route to host"라는 메시지를 받았다면 라우팅 테이블과 관련된 문제일 것이다. `netstat -r` 명령을 사용하여 현재 경로를 볼 수 있다:
- 

```
# netstat -r
Routing tables
Destination      Gateway          Flags    Refs    Use  IfaceMTU
Rtt    Netmasks:

(root node)
(root node)

Route Tree for Protocol Family inet:
(root node) =>
default          inr-3.Example.EDU  UG          8    224515  s10 -    -
```



## 21.7.2.2 간단한 개요

일반적인 설정에서 FreeBSD 를 사용한 SLIP 서버는 다음과 같이 동작한다: SLIP 유저는 FreeBSD SLIP 서버 시스템에 다이얼-업하고 특수 유저 쉘 `/usr/sbin/sliplogin` 을 사용하는 특수한 SLIP 로그인 ID 로 로그인 한다. `sliplogin` 프로그램은 특수한 유저와 매칭되는 라인을 찾기 위해 `/etc/sliphome/slip.hosts` 파일을 검색한다. 매칭되는 라인을 찾았다면 이용할 수 있는 SLIP 인터페이스에 시리얼 라인을 연결하고 SLIP 인터페이스를 설정하기 위해 `/etc/sliphome/slip.login` 쉘 스크립트를 실행한다.

### 21.7.2.2.1 SLIP 서버 로그인 예제

예를 들어 SLIP 유저 ID 가 `Shelmerg` 라면 `/etc/master.passwd` 에서 `Shelmerg` 의 엔트리는 다음과 비슷하다:

```
Shelmerg:password:1964:89::0:0:Guy Helmer - SLIP:/usr/users/Shelmerg:/usr/sbin/sliplogin
```

`Shelmerg` 가 로그인하면 `sliplogin` 은 유저 ID 와 매칭되는 라인을 `/etc/sliphome/slip.hosts` 에서 찾는다; 예를 들어 `/etc/sliphome/slip.hosts` 의 라인은 다음과 같을 것이다:

```
Shelmerg      dc-slip sl-helmer      0xfffffc00      autocomp
```

`sliplogin` 이 매칭되는 라인을 찾으면 시리얼 라인은 이용할 수 있는 다음 SLIP 인터페이스에 연결한다. 그리고 다음과 같이 `/etc/sliphome/slip.login` 을 실행한다.

```
/etc/sliphome/slip.login 0 19200 Shelmerg dc-slip sl-helmer 0xfffffc00 autocomp
```

정상적이면 `/etc/sliphome/slip.login` 은 연결되어 있는 SLIP 인터페이스(위의 예제에서 `slip.login` 에 나열되어 첫 번째 매개변수가 주어진 `slip` 인터페이스 0)에 로컬 IP 주소(`dc-slip`), 원격 IP 주소(`sl-helmer`) 그리고 SLIP 인터페이스에 네트워크 마스크(`0xfffffc00`)와 부가적인 플래그(`autocomp`)를 설정하기 위해 `ifconfig` 를 사용한다. 문제가 발생하면 `sliplogin` 은 `syslog` 데몬을 통해서 쓸 만한 정보 메시지를 보통 `/var/log/messages` 에 로그로 남긴다(어떤 `syslogd` 가 로그로 남고 어디에 남는지 확인하려면 `/etc/syslog.conf` 를 체크하고 `syslogd(8)`와 `syslog.conf(5)` 매뉴얼 페이지를 참고한다).

이 정도로 해두고 시스템 설정으로 들어가보자.

### 21.7.2.3 커널 설정

FreeBSD의 기본 커널에는 보통 두 개의 SLIP 인터페이스(sl0와 sl1)가 정의되어 있다. 이들 인터페이스가 정의되어 있는지 netstat -i를 통해볼 수 있다.

netstat -i의 샘플 출력:

Name	Mtu	Network	Address	lpkts	lerrs	Opkts	Oerrs	Coll
ed0	1500	<Link>	0.0.c0.2c.5f.4a	291311	0	174209	0	133
ed0	1500	138.247.224	ivory	291311	0	174209	0	133
lo0	65535	<Link>		79	0	79	0	0
lo0	65535	loop	localhost	79	0	79	0	0
sl0*	296	<Link>		0	0	0	0	0
sl1*	296	<Link>		0	0	0	0	0

netstat -i에서 보여 주는 sl0와 sl1 인터페이스는 커널에 두 개의 SLIP 인터페이스가 빌드되어있음을 보여준다(sl0와 sl1 뒤의 \*는 인터페이스가 "다운"되어 있음을 보여준다).

그러나 FreeBSD의 기본 커널은 패킷을 포워드(기본적으로 FreeBSD 머신은 라우터로 동작하지 않는다) 하도록 설정되지 않았기 때문에 인터넷 RFC가 인터넷 호스트에(RFC 1009 [인터넷 게이트웨이에 필요한], 1122 [인터넷 호스트에 필요한 통신 레이어] 그리고 1127[호스트에서 필요한 RFC를 볼 수 있는]를 본다) 필요하다. FreeBSD SLIP 서버가 라우터로 동작하기를 원한다면 /etc/rc.conf 파일을 수정하고 gateway\_enable 변수를 YES로 변경한다.

그리고 재 부팅하여 새로운 설정을 적용해야 된다.

기본 커널 설정 파일(/sys/i386/conf/GENERIC)의 거의 마지막 부분에 있는 다음과 같은 라인을 주의한다:

```
pseudo-device sl 2
```

이것은 커널에서 사용할 수 있는 SLIP 장치 번호를 정의한다; 마지막 라인의 숫자는 운영체제가 동시에 운용할 수 있는 최대 SLIP 연결 숫자이다.

커널을 재 설정하는 것은 8 장의 FreeBSD 커널 설정을 참고한다.

### 21.7.2.4 Sliplogin 설정

앞에서 언급했듯이 /usr/sbin/sliplogin(sliplogin의 실제 매뉴얼 페이지는 sliplogin(8)을 본다) 설정의 일부분인 3 개의 파일이 /etc/sliphome 디렉터리에 있다: slip.hosts 는 SLIP 유저와 관련된 IP 주소를 정의한다; slip.login 은 보통 SLIP 인터페이스 설정이다; 그리고(부가적으로) slip.logout 은 시리얼 접속이 종료되면 slip.login 의 설정을 되돌린다.

#### [3 개의 파일 설정]

##### 1. slip.hosts 설정

/etc/sliphome/slip.hosts 는 공백으로 나누어진 최소한 4 가지 내용을 포함하는 라인이다:

- SLIP 유저의 로그인 ID
- SLIP 링크의 로컬 주소
- SLIP 링크의 원격 주소
- 네트워크 마스크

로컬과 원격주소는 호스트 이름일 것이고(/etc/hosts 에 의해 IP 주소로 분해되거나, FreeBSD 5.X 에서는 /etc/nsswitch.conf 그리고 FreeBSD 4.X 에서는 /etc/host.conf 의 설정에 따른 도메인 네임 서비스다) 네트워크 마스크는 /etc/networks 검색에 의해 분석할 수 있는 이름일 것이다. 샘플 시스템에서 /etc/sliphome/slip.hosts 는 다음과 같다:

```
#
# login local-addr      remote-addr      mask            opt1    opt2
#                               (normal,compress,noicmp)
#
Shelmerg dc-slip        sl-helmerg      0xfffffc00      autocomp
```

각 라인의 마지막에는 다음의 하나 이상의 옵션을 지정할 수 있다.

- *normal* -- 헤더를 압축하지 않는다.
- *compress* - 헤더를 압축한다.
- *autocomp* -- 원격에서 허락한다면 헤더를 압축한다.
- *noicmp* -- ICMP 패킷을 비활성 한다(그래서 어떠한 "ping" 패킷도 여러분의 대역폭을 사용하지 못한다).

여러분의 SLIP 링크에서 로컬과 원격 주소를 선택하는 것은, TCP/IP 서브넷을 사용할 것인가 또는 SLIP 서버에 "proxy ARP"를 사용할 것인지에 따라 달라진다(이것은 "진짜" proxy ARP 가 아니지만 설명을 위해서 이 용어를 사용한다). 어떤 방법을 사용할지 또는 IP 주소를 어떻게 할당할지 정확하지 않다면 SLIP 서버를 설명하기 전에 언급한 내용들을 참고하고 여러분의 네트워크 IP 관리자에게 도움을 청한다.

SLIP 클라이언트를 서브넷으로 나눈다면 여러분에게 할당된 네트워크 IP 번호의 서브넷을 만들고 이 서브넷 내의 IP 주소를 각 SLIP 클라이언트에 할당한다. 그리고 이 SLIP 서브넷으로부터 SLIP 서버를 경유하여 가장 가까운 IP 라우터의 경로를 정적으로 설정해야 될 것이다.

그렇지 않고 "proxy ARP" 방법을 사용한다면 여러분의 SLIP 서버 서브넷 범위의 IP 를 SLIP 클라이언트 IP 주소로 할당한다. 그리고 arp(8)을 사용하여 SLIP 서버의 ARP 테이블에서 proxy-ARP 엔트리를 관리하도록 /etc/sliphome/slip.login 과 /etc/sliphome/slip.logout 스크립트를 수정해야 된다.

## 2. slip.login 설정

전형적인 /etc/sliphome/slip.login 파일은 다음과 같다:

```
#!/bin/sh -
#
#      @(#)slip.login  5.1 (Berkeley) 7/1/90
#
# generic login file for a slip line.  sliplogin invokes this with
# the parameters:
#      1      2      3      4      5      6      7-n
#  slipunit ttyspeed loginname local-addr remote-addr mask opt-args
#
/sbin/ifconfig sl$1 inet $4 $5 netmask $6
```

이 slip.login 파일은 SLIP 인터페이스의 로컬, 원격 주소 그리고 네트워크 마스크를 ifconfig 명령으로 설정한다.

"proxy ARP" 방법(SLIP 클라이언트를 서브넷으로 나누는 대신)을 사용하기로 결정했다면 필요한 /etc/sliphome/slip.login 파일은 다음과 비슷할 것이다:

```
#!/bin/sh -
#
#      @(#)slip.login  5.1 (Berkeley) 7/1/90
#
# generic login file for a slip line.  sliplogin invokes this with
# the parameters:
#      1      2      3      4      5      6      7-n
#  slipunit ttyspeed loginname local-addr remote-addr mask opt-args
#
/sbin/ifconfig sl$1 inet $4 $5 netmask $6
# Answer ARP requests for the SLIP client with our Ethernet addr
/usr/sbin/arp -s $5 00:11:22:33:44:55 pub
```

slip.login 의 추가적인 라인 arp -s \$5 00:11:22:33:44:55 pub 은 SLIP 서버의 ARP 테이블에 ARP 엔트리를 생성한다. 이 ARP 엔트리는 이더넷의 다른 IP 노드에서 SLIP 클라이언트의 IP 주소를 질의할 때 SLIP 서버의 이더넷 맥 주소로 응답하도록 한다.

위의 예제를 사용할 때 이더넷 맥 주소(00:11:22:33:44:55)를 여러분의 시스템의 이더넷 카드 맥 주소로 대체한다. 그렇지 않으면 여러분의 "proxy ARP"는 절대로 동작하지 않는다! netstat -i 를 실행한 결과에서 여러분의 SLIP 서버 이더넷 맥 주소를 확인할 수 있다; 출력의 두 번째 라인이 다음과 비슷할 것이다:

```
ed0 1500 <Link>0.2.c1.28.5f.4a 191923 0 129457 0 116
```

이것은 이 시스템의 이더넷 맥 주소가 00:02:c1:28:5f:4a 임을 보여준다. netstat -i 에서 보여 준 이더넷 맥 주소에서 마침표를 콜론(:)으로 바꾸고 arp(8)이 요구하는 형식으로 주소를 변환하기 위해 하나의 16 진수 숫자에는 0 을 추가한다; 사용법에 대한 완벽한 정보는 arp(8) 매뉴얼 페이지를 본다.

**Note:** /etc/sliphome/slip.login 과 /etc/sliphome/slip.logout 을 생성하고 실행 가능한 비트(chmod 755 /etc/sliphome/slip.login /etc/sliphome/slip.logout)를 설정해야 된다. 그렇지 않으면 sliplogin 은 실행할 수 없다.

### 3. slip.logout 설정

/etc/sliphome/slip.logout 이 꼭 필요하지 않지만("proxy ARP"를 사용하지 않는다면) 생성하기로 했다면 기본 slip.logout 스크립트 예제가 여기 있다:

```
#!/bin/sh -
#
#      slip.logout
#
# logout file for a slip line.  sliplogin invokes this with
# the parameters:
#      1      2      3      4      5      6      7-n
#      slipunit tty speed loginname local-addr remote-addr mask opt-args
#
/sbin/ifconfig sl$1 down
```

"proxy ARP"를 사용한다면 /etc/sliphome/slip.logout 가 SLIP 클라이언트의 ARP 엔트리를 삭제하기를 원할 것이다:

```
#!/bin/sh -
#
#      @(#)slip.logout

#
# logout file for a slip line.  slplogin invokes this with
# the parameters:
#      1      2      3      4      5      6      7-n
#  slpunit ttyspeed loginname local-addr remote-addr mask opt-args
#
/sbin/ifconfig sl$1 down
# Quit answering ARP requests for the SLIP client
/usr/sbin/arp -d $5
```

arp -d \$5 는 SLIP 클라이언트가 로그인 했을때 slip.login 에 추가된 "proxy ARP" 엔트리를 삭제한다.

/etc/sliphome/slip.logout 을 생성하고 실행할 수 있도록(다시말해서 chmod 755 /etc/sliphome/slip.logout) 다시 설정한다.

### 21.7.2.5 라우팅에서 고려할 사항

SLIP 클라이언트와 나머지 네트워크(아마도 인터넷) 사이의 패킷 라우팅에 "proxy ARP" 방법을 사용하지 않는다면, SLIP 서버를 경유하여 SLIP 클라이언트가 속하는 서브넷까지의 경로를 가장 가까운 기본 라우터에 정적인 경로로 추가해야 될 것이다.

[정적인 경로에서 고려할 사항]

## 1. 정적인 경로

정적인 경로를 가장 가까운 기본 라우터에 추가하는 것이 문제가 될 수 있다(또는 권한이 없다면 불가능하다). Cisco 와 Proteon 에서 만든 몇몇 라우터처럼 여러분의 조직에 다중 라우터로 구성된 네트워크가 있다면 정적인 경로를 SLIP 서브넷에서 사용하도록 라우터를 설정하고 이 정적 경로를 다른 라우터에 알려야 되기 때문에 정적 경로에 기반하는 라우팅(static-route-based routing)이 동작하도록 특정 전문지식과 문제해결/시스템 변경 능력이 필요할 것이다.

## 2. GateD® 실행

정적 경로의 또 다른 문제는 FreeBSD SLIP 서버에 **GateD** 를 설치하고 적절한 라우팅 프로토콜(RIP/OSPF/BGP/EGP)을 사용하여 SLIP 서브넷에 대한 경로 정보를 다른 라우터에 알리도록 설정하는 것이다. gated 를 설정하기 위해 /etc/gated.conf 파일을 작성해야 된다; 여기 필자가 FreeBSD SLIP 서버에 사용하는 것과 비슷한 샘플이 있다:

**Note:** GateD®는 이제 소유권이 있는 소프트웨어이기 때문에 더 이상 공개적으로 소스코드를 이용할 수 없다(더 많은 정보는 GateD 웹 사이트에 있다). 이 섹션은 아직 사용되고 있는 예전 버전과 호환되도록 존재한다

```
#
# gated configuration file for dc.dsu.edu; for gated version 3.5alpha5
# Only broadcast RIP information for xxx.xxx.yy out the ed Ethernet interface
#
#
# tracing options
#
traceoptions "/var/tmp/gated.output" replace size 100k files 2 general ;

rip yes {
    interface sl noripout noripin ;
    interface ed ripin ripout version 1 ;
    traceoptions route ;
};

#
# Turn on a bunch of tracing info for the interface to the kernel:
```

```
kernel {
    traceoptions remnants request routes info interface ;
};

#
# Propagate the route to xxx.xxx.yy out the Ethernet interface via RIP
#

export proto rip interface ed {
    proto direct {
        xxx.xxx.yy mask 255.255.252.0 metric 1; # SLIP connections
    };
};

#
# Accept routes from RIP via ed Ethernet interfaces

import proto rip interface ed {
    all ;
};
```

위의 샘플 `gated.conf` 파일은 SLIP 서브넷 `xxx.xxx.yy` 대한 경로 정보를 RIP 를 사용하여 이더넷에 브로드캐스트 한다; `ed` 드라이버 대신 다른 이더넷 드라이버를 사용한다면 레퍼런스의 `ed` 인터페이스를 적절히 변경해야 된다. 이 샘플 파일은 **GateD** 의 활동을 디버깅하기 위해 `/var/tmp/gated.output` 에 로그로 남기도록 설정되어있다; **GateD** 가 정확히 동작한다면 로그 옵션을 삭제할 수 있다. `xxx.xxx.yy` 를 여러분의 SLIP 서브넷 네트워크 주소로 변경해야 된다(*proto direct* 문자열의 넷 마스크도 변경한다).

시스템에 **GateD** 를 설치하고 설정했다면 FreeBSD 시작 스크립트가 **routed** 대신 **GateD** 를 실행하도록 해야 된다. 가장 쉬운 방법은 `/etc/rc.conf` 에 `router` 와 `router_flags` 변수를 설정한다. **GateD** 의 명령어 매개변수에 대한 정보는 매뉴얼 페이지를 참고한다.