Wireless Howto

Roberto Arcomano

bertolinux@galactica.it

도운이

고지훈

gobears@kldp.org

리눅스 시스템을 위한 Wireless HOWTO는 Arcomano, bertolinux@galactica.it가 2001년 1월 12일에 발표한 버전 1.2 문서로 빠른 속도(약 11Mbps 이상)를 지원하는 새로운 네트워크 카드의 신기술에 관한 문서입니다. 이 문서는 리눅스에서 무선 네트워크를 설치하고, 문제를 해결하며, 위치적인 문제와 그 외에 것에 대해서 설명하고 있습니다. 가장 최근의 문서는 다음의 url에서 찾을 수 있습니다. http://web.tiscalinet.it/bertolinux

고친 과정

고침 번역 v1.12001년 4월 14일고친이 gobears

한국어 번역판

고침 v1.2 12 January 2001

1. 소개

1.1. 소개

이 문서는 무선 네트워크를 어떻게 설치하고 문제점은 무엇이 있는지 설명합니다. 유선 네트워크와는 달리 무선 네트워크는 잘 작동하기 위해서는 약간의 묘수가 필요합니다. 당신은 안테나와 안테나의 방향성과 로밍(roaming)에 대해서도 알아야 합니다. 의견은 언제나 환영하며, Jean Tourrilhes Wireless Howto (http://www.hpl.hp.com/personal/Jean_Tourrilhes/Linux)에서 좀더 재미있는 점을 발견할 수 있을 겁니다.

모든 의견과 질문은 이메일 (mailto:bertolinux@galactica.it)로 해 주시기를 부탁드립니다. 번역에 관한 내용과 한국어 질문은 gobears@kldp.org (mailto:gobears@kldp.org)로 해 주시기 바

랍니다.

1.2. 저작권

이문서는GNU GPL (http://www.gnu.org/copyleft/gpl.html)를 따릅니다.

1.3. 번역자의 생각

외국-특히 미국-에서는 무선 네트워크 구축이 상당한 힘을 가지고 추진되고 있습니다. 내용중 자주 보시게 될 Access Point 라는 장치-일종의 중계기 입니다.-도 미국에서는 저렴한 가격으로 공급 되고 있습니다. 리눅스에서도 무선 네트워크의 구성이 가능하다는 생각으로 읽어주셨으면 합니다.

-가자곰-

2. 배경지식

2.1. 무선이란 무엇인가?

무선이란 떨어져 있는 컴퓨터들을 연결해 주는 새로운 기술을 말합니다. 이 기술은 2.4 GHz에서 작동하는 TX/RX를 지원하는 무선 네트워크 카드에 의해서 구현됩니다. 사용되는 소프트웨어 인터페이스는 Ethernet-like이며 전세계에서 각각 다른 주소로 사용되는 무선 네트워크 카드에서 작동합니다. 사용되는 전력은 10-20 mW에서 100mW까지 입니다. (표준 IEEE 802.11 and FCC/CEPT licenses를 살펴보시기 바랍니다.)

2.2. 두개의 무선 네트워크 카드의 최대 전송거리는 얼마인가?

가장 중요한 점은 두개의 무선 네트워크 카드사이에 장애물이 없어야 한다는 것입니다. 당신은 서로 다른 무선 네트워크 카드의 안테나를 육안이나 혹은 쌍안경을 써서라도 볼 수 있어야 합니다.

사용 가능한 거리는 안테나와 증폭기에 따라서 달라집니다. : 무지향성 안테나를 사용하면 2-300m, 지향성 안테나를 사용하면 1km, 무지향성 증폭 안테나 (200mW)를 사용하면 2-3km, 접시형 안테나를 사용하면 수 km, 접시형 안테나와 지향성 증폭 안테나(수 와트)를 사용하면 50-60km 가능합니다.

당신이 사용하고 있는 무선 네트워크 카드가 FCC/CEPT 규정이나 당신이 있는 곳의 해당 법령을 위반하지 않는지 확인해 보아야 합니다.

2.3. 무선 네트워크와 유선 네트워크와의 차이점은 무엇인가?

유선 네트워크는 적어도 낮은 레벨에서는 설치하기가 매우 간단합니다. 하지만, 무선 네트워크는 설치, 유지, 문제해결 등 이 모든 문제가 매우 어렵습니다. 하드웨어와 소프트웨어 설치, 그리고 버그 잡기 등은 무선 네트워크의 일반적인 문제이며, 매우 치명적이 될 수 있습니다.

1. 당신은 알맞은 무선 네트워크 카드를 선정해야 합니다. 다양한 요구사항과 특징을 가지는 많은 종류의 무선 네트워크 카드가 있습니다. 당신이 작은 규모의 LAN/WAN을 구성하려고 한다면 IEEE 802.11 규정을 준수하는 무선 네트워크 카드를 구입해야 합니다.
2. 많은 카드들이 PCMCIA 방식입니다. 당신은 먼저 PCMCIA 리눅스 소스를 설치해야 합니다.
3. 당신은 2개의 시스템으로 테스트를 해 보아야 합니다. 한번은 짧은 거리에서 한번은 좀 더 먼거리에서 하시기를 추천합니다.
4. 당신은 다양한 날씨 상태(눈, 비 올 때 등)에서 테스트해 보아야 합니다.
5. 설치를 끝내십시오.:-)

중계기(repeater)를 설치했다면- 리눅스 박스(Linux box)는 많은 수의 유무선 네트워크 카드를 지원합니다- 당신은 원거리 네트워크 구성에서 문제를 발견할 수도 있습니다.

2.4. 무선 네트워크를 설치하는데 무엇을 알아야 하나?

무선 네트워크를 설치하는 것은 다음을 요구합니다.

소프트웨어 요구사항 :

1. generic Linux NET3-4-HOWTO에 다루는 IP address, netmask, routing 과 같은 일반적인 네트워크 지식
2. Bridge Mini-Howto and in Linux Kernel Source (2.2.x or 2.4.x) 에 있는 proxy ARP나 바인딩(binding), proc fs 같은 Proxy-ARP-Subnet 지식.
3. ADHOC, INFRASTRUCTURE, ACCESS POINT channel concept, outdoor 와 indoor defines 와 같은 무선 네트워크 액세스 모드 정보. 이 정보는 IEEE 표준 802.11, CEPT등에서 관련 정보를 얻을 수 있습니다.

소프트웨어 이외의 요구사항

1. physical mounting, pointing 그리고 안테나에 관한 기초적인 경험
2. 두개의 다른 무선 네트워크 사이에서 간섭없이 진행할 수 있게 하는 주의 깊은 PC 하드웨어 설치.

그리고 커다란 행운 :-)

2.5. 왜 무선 네트워크 카드를 설치해야 하나? 그리고 무엇을 기대할 수 있나?

왜냐구요? 당신이 유선 네트워크에 만족하지 못하고 있기 때문입니다.

무선 네트워크 카드는 정원이나 공원 혹은 집 사이에서 네트워크를 구성할 수 있게 해 줍니다.(단지 당신이 다른 카드를 볼 수 있는 장소에 있어야 하지만요.)

무선 네트워크 카드 에서도 Ethernet 카드에서 사용되는 것과 같은 하이 레벨 프로토콜(TCP/IP)이 사용됩니다. 하지만, 윈도우 파일 공유 프로그램을 사용한다면 좀더 많은 주의가 요구됩니다. 왜냐하면 당신이 리눅스를 포워드(forward)로 사용한다면 당신은 IP 포워더(forwarder)가 브로드케스트 메시지를 통과 하지 못하게 한다는 경고를 발견하게 될 것입니다.(자세한 것은 NetBIOS protocol를 참조하세요) 이러한 경우 당신은 Network Browsing을 지원하는 WINS server를 이용해야 합니다.

무선 네트워크는 중앙 접점을 통해서 작은 규모의 LAN/WAN을 만들 수 있게 해 줍니다.

한 나라의 모든 케이블이 무선 네트워크로 꾸며지는 것을 상상해 보십시오.

나라의 모든 사람들과 광 대역폭으로 파일을 공유하고 음성을 전달하고, 영상을 공유한다는 것을 상상해 보십시오.

이 모든 것은 실현 가능합니다. 이미 몇몇 나라에서는 무선 네트워크 카드와 Wireless Access Point와 중계기로 작동될 수 있는 Wireless Linux Boxes를 이용하여 실현되고 있습니다. (IP 레벨의 라우터나 혹은 data-link 레벨의 프리지 드라이버에 관해서 더 알고 싶으시다면 Bridge Http Link (http://openrock.net/bridge)나 Bridge Ftp Link (ftp://openrock.net/bridge)사이트를 방문해 보시기 바랍니다.)

2.6. 이 Howto 문서는 어떤 무선 네트워크 카드를 다루고 있나?

이 문서에서 일반적인 무선 네트워킹을 소개하는 일반적인 설치 과정으로 시작하여 각 카드들의 예를 설명하고 있습니다. 그리고 몇몇 트릭은 좀 더 나은 성능을 가져올 수 있습니다.

무선 네트워크 카드 목록:

1. Proxim Symphony -http://www.proxim.com
2. Webgear AviatorPRO 2.4 (pcmcia support needed) - http://www.webgear.com
3. Lucent Wavelan I, II, Orinoco - http://www.lucent.com and http://www.orinoco.net
4. Cabletron - http://www.cabletron.com
5. YDI am930_isa - http://www.ydi.com
6. Siemens Radio Modem (Dect) - http://www.siemens.com
7. RadioLan (5 GHZ) - http://www.radiolan.com

보다 완벽한 목록은 Jean Tourrilhes Wireless Howto (http://www.hpl.hp.com/personal/Jean_Tourrilhes/Linux)에서 보실 수 있습니다.

Siemens Radio Modem은 802.11 규정을 준수하는 무선 네트워크 카드는 아닙니다. 이 카드는 시리얼로 동작하고 1800MHz에서 DECT 테크놀러지를 이용하여 모뎀처럼 작동합니다. 자세한 사항은 부록 B에 기술되어 있습니다.

RadioLan 카드는 윈도우 9X 환경에서 5.4GHz대에서 작동합니다. 그리고, RadioLan Access Point는 유무선 사이의 브리지 역할을 합니다. (그러나, 아직 알려진 리눅스 드라이버가 없습니다.)

2.7. 비용은 얼마나 드나?

위에 열거되어 있는 무선 네트워크는 매우 비싸지 안습니다.(비싼 것 같은데:역자 주) bridge로 작동하는 2개의 무선 네트워크 카드(Lucent 제품)는 수백에서 수천 달러 정도 합니다.

3. 무선 네트워크에 관한 기술적인 정보

기초적인 무선 네트워크 환경에 대해서 이해할 수 있는 기술적인 정보를 이야기 합니다.

3.1. 물리층(layer)

우선, ISO/OSI에서 이야기하는 3종류의 레이어가 있습니다.

1. FHSS, Frequency Hopping Spread Spectrum
2. DSSS, Direct Sequence Spread Spectrum
3. Infrared connections : 이 문서에서는 다루지 않습니다.

3.2. 구성

두 가지의 구성 방법이 있습니다.

1. AdHoc mode(Independent mode라고도 불리는 것): BSS(Basic Service Set)를 가지고 있는 독립적인 네트워크를 구성할 때의 방식. 각각의 스테이션은 같은 BSS(Basic Service Set)를 가지고 있어야 합니다.
2. Infrastructure mode : 각각의 BSS(Basic Service Set)를 가지는 수개의 네트워크가 서로 통신할 때 사용하는 방식. 가까운 액세스 포인트로 중계할 수 있는 기능을 가지는 "attach" 스테이션이 있어야 합니다.

Adhoc 은 측량하기는 좀 힘들지만 좀더 간단한 방법입니다. 여러 개의 호스트들이 서로 직접 연결되게 해 줍니다. 제한적인 요구사항이 있는데 모든 네트워크 상에 있는 호스트들이 서로

볼 수 있는 위치에 있어야 한다는 것입니다. 중간에 장애물이 있어서는 안 됩니다.(적어도 이론상으로 그렇습니다. 이러한 문제를 해결하는 방법은 5.4절을 참조해 주시기 바랍니다.)

```
                         Adhoc mode

                        A - - - - - C
                          \       /
                        |   \   /   |
                              /\
                        |   /    \  |
                          /       \
                        B - - - - - D
```

Infrastructure 환경에서는 공유된 네트워크의 다른 호스트들과 연결하기 위해서 Access Point를 이용합니다.

```
Infrastructure mode
                           ESS

          A - - - | - Access Point - -  Access Point - | - - - D

          B - - - |   BSS1                   BSS2      | - - - E

          C - - - |                                    | - - - F
```

B와 C는 D,E,F를 볼수 없습니다. 그러나 같은 ESS를 이용하는 것 때문에 서로 통신 할 수 있습니다. 중요 : A,B,C도 서로를 볼 수 없습니다.

참고적으로 장거리와 단거리를 구분하는 indoor와 outdoor라는 용어가 있습니다.

3.3. 호환성

세상에는 굉장히 많은 무선 네트워크 카드가 있다는 것을 잊지 마시기 바랍니다. 모든 카드가 항상 다른 카드들과 통신할 수 있는 것도 아닙니다. 카드들이 서로 통신하기 위해서는 다음을 따라야 합니다.

1. 같은 구성 모드 : 모두 Adhoc 모드이던가 아니면 모두 Infrastructure모드 일 것
2. 같은 물리층(layer) : 모두 DSSS 이던가 아니면 모두 FHSS 일 것
3. 같은 프로토콜 (예를 들면 Proxim은 특유의 특정 프로토콜을 사용하기 때문에 다른 FHSS 카드와는 통신할 수 없습니다.)

3.4. Adhoc 이나 Infrastructure 모드중 어느 것을 사용해야 하나?

Access Point는 굉장히 유용하고 문제점을 고치는 데도 또한 좋습니다. 그러나 매우 비쌉니다. 이상적으로는 좀더 밀집된 네트워크 상에서는 Infrastructure모드가 좋고 소수의 호스트를 이용할 때는 Adhoc 모드가 좋습니다. 왜냐하면 비용 때문이지요

어쨌든 잘 작동하는 무선 네트워크를 가지기 위해서는 아마도 많은 돈을 써야 할 것입니다. 반면에 돈을 조금만 쓴다면 몇몇 문제에 직면할 것입니다.

3.5. Linux Box(리눅스 컴퓨터)는 AccessPoint처럼 동작할 수 없나?

좋은 질문입니다. 불행하게도 이것을 가능 하게 할 소프트웨어(혹은 프리웨어)가 없습니다. 그래서 우선 Adhoc 모드를 이용하거나 혹은 더 많은 것을 사야 하는 것입니다.

주의 : 당신이 소프트웨어 코드 변환에 전문가라면, Access Point 펌웨어를 다운 받아서 리눅스에 맞게 변환 시킬 수 있을 겁니다.

4. 필요한 도구들

4.1. 하드웨어 요구사항

리눅스 컴퓨터가 필요합니다.(최소 486 이상 펜티엄 100이상 추천 16MB 램 이상) 그리고, 무선 네트워크 카드와 원한다면 안테나도 필요합니다. 그리고 반대 쪽도 같은 것이 필요합니다.(Win9x 또는 WinNT 등 선호하는 것) 그래야만 시험을 할 수 있으니까요.

4.2. 소프트웨어 요구사항

필요한 것

1. 최근의 안정된 커널 소스(2.2.x 이상)
2. pcmcia카드를 샀다면 최근의 안정된 pcmcia 소스 (pcmcia-cs)
3. 무선 네트워크 카드 드라이버 : 드리이버 파일을 가지고 계시지 않다면 판매자의 사이트나 만든 회사의 사이트에서 해당하는 파일을 다운 받을 수 있습니다. 만약 찾을 수 없다면 Jean Tourrilhes Wireless Howto (http://www.hpl.hp.com/personal/Jean_Tourrilhes/Linux)에서 찾을 수 있을 것입니다.

위에서도 조차 리눅스용 드라이버를 발견할 수 없다면 당신은 윈도우용 드라이버가 리눅스로 변환되기를 기댜려야 할 것입니다. (행운을 빕니다:-))

이 모든 것을 구한 후에 새로운 커널을 다시 컴파일 해야 하고, 무선 네트워크 카드가 요구한다면 pcmcia 소스도 다시 컴파일 해야 합니다. 무선 네트워크 카드 드라이버도 물론 다시 컴파일 해야 합니다. 이것은 일반적인 경우이고, 어떤 카드들은 세 번째 만 해도 되는 경우가 있고,

어떤 카드들은 첫 번째와 세 번째만 해야 되는 경우도 있습니다. 이것은 카드에 따라 달라 집니다.

5. 무선 네트워크 설치의 개요

5.1. 기본적인 과정

이전에 필요한 모든 문서를 얻고, 필요한 모든 파일을 컴파일 했다면 무선 네트워크의 구성에 필요한 가장 기본적인 과정을 이행해야 합니다.

1. 로우 레벨 커널 구성(Low level kernel config)에서 리눅스 커널이 무선 네트워크 카드를 인식하도록 해주어야 합니다. (Ioport와 interrupts, dma등과 같은 로우 레벨) : 무선 네트워크 카드가 제대로 설치 구성 되었다는 커널 메시지를 볼 수 있어야 합니다.
2. Data-link level setting에서 각각의 무선 네트워크 카드에 맞는 특정의 값을 설정해 주어야 합니다. 예를 들면 Proxim Symphony 제품에는 PCMCIA카드 설정을 해주는 "r12cfg"라는 유틸리티가 있습니다. 모든 무선 네트워크 카드를 서로 통신할 수 있게 설정해 주어야 합니다.
3. IP setting : 바로 ifconfig 와 route capabilities를 사용하여 IP setting을 변경할 수 있습니다.
4. 좀더 나은 성능을 위한 방법과 충돌을 피하는 법 : 무선 네트워크는 기본적으로 작동하고 있습니다. 덧붙여서 proxy-arp, icmp, echo redirect, bridging, channel change와 같은 작은 설정사항을 조정해야 될 수도 있습니다.

N.B. : 단계1, 2, 3은 표준 ISO/OSI규정을 준수합니다. 단계4에서는 netmask 255.255.255.255에 의해서 발생될 수 있는 문제를 해결할 수 있는 방법이 더해 져야 합니다. 사실상 32bit netmask는 표준 ISO/OSI를 위반하는 것입니다. 이것은 Broadcast를 위해 같은 주소를 사용하는 것이나 존재하지 않는 네트워크 어드레스나 IP machine을 사용하는 것을 네트워크가 강요할 때 발생합니다.

두 번째 단계는 유선 네트워크에서는 필요가 없습니다. 왜냐하면 특별한 설정과정이 필요 없기 때문입니다.

5.2. Low Level Kernel 설정

항상 관리자(administrator)에게는 이것이 문제입니다. 커널이 하드웨어를 볼 수 있게 해 주어야 합니다.

무선 네트워크 카드는 대부분이 PCMCIA 방식이라서 더 복잡합니다. 우선 커널이 PCMCIA 카드를 볼 수 있도록 해 주어야 합니다. 그리고, 알맞은 하드웨어 드라이브를 설치해야 합니다.

그래서, 다음과 같은 방법으로 PCMCIA 카드를 구성할 수 있습니다.

1. http://www.kernel.org에서 커널 소스를 /usr/src/linux로 인스톨 합니다.
2. ftp://projects.sourceforge.net/pub/pcmcia-cs에서 /usr/src/pcmcia로 PCMCIA 소스를 인스톨 합니다.
3. linux 디렉터리(/usr/src/linux)에 있는 README파일을 잘 읽고 커널을 다시 설정하고 다시 컴파일합니다.
4. PCMCIA 소스를 설정하고 다시 컴파일 합니다. : /usr/src/pcmcia 에 있는 configure와 make 이용하면 됩니다. 드라이버를 확인하고 인스톨 해야 합니다. 다음의 과정이 따를 수 있습니다. "make all" 입력 한 후 "make install" 을 입력해야 됩니다.
5. 인스톨 후에 유용한 설정파일을 /etc/pcmcia에서 발견할 수 있습니다.

PCMCIA카드가 아닌 경우 :

1. 리눅스에서 작동하는 드라이버 파일(대부분 아닐 겁니다.)이 있다면, 해당 디렉터리에 인스톨 하고 컴파일합니다.

한번 로드해야 할 모듈이름을 당신은 알고 있습니다. : PCMCIA 설정에서 오직 한번 PCMCIA 데몬을 실행하는 것이 필요합니다. (레드헷 에서는 /etc/rc.d/init.d/pcmcia 입니다.) 옵션을 사용하면 ioport, irq, data-link settings(5.3절을 보시기 바랍니다.)을 할 수 있습니다. 어쨌든 제대로 설치된 드라이버와 하드웨어가 있다면 당신이 유용하게 사용할 수 있는 tool은 다음에 있습니다.

1. `tail /var/log/messages`은 syslog에 대해서 설명합니다.
2. `dmesg` 에 좀더 많은 정보가 있습니다.
3. `/proc` dir: ioports, devices, irq 파일, 드라이버의 특정한 서브 디렉터리

5.3. Data-link level 설정

Data-link level설정이 무엇인가?

유선 네트워크는 서로 연결되는 것만 필요합니다. 그리고 TCP/IP 설정을 해줄 수 있습니다.

반면에 무선 네트워크는 다음과 같은 data-link settings이 필요합니다.

1. 가지고 있는 무선 네트워크가 어떤 종류인가? (Adhoc 혹은 Infrastructure)
2. 어떤 채널을 사용해야 하나?
3. 내가 속해있는 subnet(BSSID)은 무엇인가? 나의 ESS ID는 무엇인가?
4. 암호화 알고리즘에 의해서 나의 통신은 보호되고 있나? 있다면 암호화 길이는 어떻게 되나?

위에서 본 것과 같이 조정해야 할 많은 셋팅이 있습니다. 이것은 무선 네트워크의 구조에 의해서 생기는 문제입니다. 가까운 거리에 있는 어떤 사람이 당신의 팻킷을 이용할 수 있고, 또

한 안테나를 지향 시키는 것과 TCP/IP 셋팅을 하는 것으로 당신의 서비스를 이용할 수 있습니다.

추가로 많은 무선 네트워크 subnet은 서로에게 간섭을 일으킬 수 있습니다.

그래서 다음과 같이 하는 것을 권장합니다.

1. load-time module에 다음과 같은 옵션을 줍니다. : `modprobe ray_cs essid='LINUX'` 혹은
2. run-time driver의 유틸리티를 사용합니다. : `rl2cfg eth1 master`

5.4. IP 설정

이것은 당신이 직면할 세번째 문제입니다. 이 상황은 당신의 네트워크가 더 큰 네트워크에 포한 될 때 발생합니다.

무선 IP 네트워킹은 당신이 무선 IP 네트워킹에 스트레스를 주지 않는다면 또한 그 문제로 당신이 스트레스를 받지 않는 다는 것을 명심하시기 바랍니다.(필자 주 : 어려운 말이죠. 스트레스 받지 말라는 말입니다.)

5.4.1. 간단한 구성

```
모든 호스트들이 서로를 볼 수 있는 상태

A - - - - - C
  \       /
|   \   /   |
      /\
|   /     \ |
  /       \
B - - - - - D
```

위와 같은 간단한 구성에서는 특별한 것이 요구되지 않습니다.(IP 레벨에서) : 각각의 호스트에 IP 주소를 설정하고 global netmask로 서로를 묶습니다.

5.4.2. 좀 더 복잡한 구성

```
A는 직접 B를 볼 수 없는 상태
     A <- - - -
   NO\        |
 TALK\        C
     \        |
     B <- - - -
```

A와 B는 C를 통해서만 통신할 수 있습니다.

네트워크가 Infrastructure모드이고 C가 Access Point 라면 문제가 발생하지 않습니다. Adhoc모드라면 당신은 하나의 호스트를 "master" 능력을 가지는 것으로 셋팅할 수 있습니다. 호스트는 BSS를 생성할 수 있고, 다른 호스트들은 만들어진 BSS에 참가할 수 있습니다.

완전한 연결은 IP 레벨에서 가능합니다 : A 와 B는 같은 C의 인터페이스를 이용하여 C와 통신할 수 있습니다. 그래서 A에서 B로 ping을 시도한다면 C로부터 많은 ICMP REDIRECT 패킷을 받게 될 것입니다. 왜냐하면 C는 A가 이미 request가 오는 네트워크의 목적지라는 것을 알고 있기 때문입니다.

해결 방법 : `echo 0 > /proc/sys/net/ipv4/conf/ethx/send_redirects`라고 입력 하시기 바랍니다.

A와 C를 무엇으로 netmask를 설정해야 하나? 만약 A를 A와 C를 포함하는 netmask로 설정한다면 작동하지 않을 것입니다. 왜냐하면 A는 gateway C를 사용할 수 없기 때문입니다. 그러나 알 수 없는 목적지 MAC address로의 ARP요청은 만들 수 있습니다.

Proxy ARP를 사용해야 한다고 생각할 수 있습니다. 신호를 요구하는 곳과 신호를 받는 곳이 다른 인터페이스를 사용할 때만 proxy ARP가 작동하는 것 때문에 이러한 효과 없이 proxy arp를 사용할 수 있어야 하는데, 이것은 우리의 경우와는 관계가 없는 것입니다.

그래서 아주 작은 netmask를 해야 합니다.(Win9x의 경우 255.255.255.254, Win NT의 경우 255.255.255.248) 그리고, A와 C가 같은 net address를 사용하지 않는지 확인해야 합니다.

예

1. IP(A) = x.y.z.2/31, IP(B) = x.y.z.3/31. 이 경우는 작동하지 않습니다. 왜냐하면 A가 B에게 ARP를 같은 네트워크에서 요구 하기 때문입니다. A와 B가 같은 인터페이스에 속해 있기 때문에 C는 동작하지 않습니다.
2. IP(A) = x.y.z.1/31, IP(B)= x.y.z.2/31. 이 경우에는 작동합니다. A가 B에 관한 것을 C에게 물어보기 때문입니다.

일반적인 경우 netmask 255.255.255.254 시스템은 오직 마지막 비트를 변경하는 것으로 동작합니다.

높은 레벨의 호환성은 TCP/IP forcing로 가능합니다.

노트 : Access Point를 사용한다면(Infrastructure mode로), redirect 문제는 발생하지 않을 것입니다. 왜냐하면 Access Point가 모든 data-link level의 문제를 해결하기 때문입니다. 그러나 Access Point는 매우 비쌉니다. P133 32MB Ram와 두개의 카드를 사용하는 것이 더 경제적입니다.

5.4.3. 인터넷 연결

```
A - - - - - C - - Internet
  \       /
|   \   /   |
      /\
|   /    \  |
  /       \
```

```
       B - - - - - - D
```

여러 가지 상황이 있습니다.

1. C는 오직 하나의 공용 IP 주소 입니다. 다른 네트워크 카드들은 인터넷 연결을 가능하게 하기 위해서 개인 IP(192.168.x.y와 같은)로 설정할 수 있습니다. A와 B와 D는 C라는 기본 게이트웨이를 가지는 것입니다.
2. 인터넷에서 사용할 수 있는 공용 netmask를 가질 수 있습니다. 그리고 C는 네트워크에서 기본 게이트웨이가 됩니다. 당신은 오직 A,B,D가 기본게이트웨이로 C를 지정하도록 하면 됩니다.
3. 인터넷에서 사용할 수 있는 공용 netmask를 가질 수 있습니다. 그리고 C는 기본 게이트웨이가 아닐 수 있습니다. 두 가지 해결 방법이 있습니다. 하나는 기본 게이트웨이가 C로 모든 정보를 보내는 방법이고, 다른 하나는 C에 관한 Proxy ARP를 사용 가능하게 하는 방법입니다. `echo 1 > /proc/sys/net/ipv4/conf/ethx/proxy_arp` 를 입력하여 기본 게이트웨이를 C로 설정하시면 됩니다. Proxy ARP는 TCP/IP forcing이지만 잘 동작 합니다.

5.4.4. 유선과 무선이 혼합된 네트워크

```
        Internet
                \
                 \        E
                  \     /
                   \  /        Wireless
       A - - - - - - C - - F
          \        /   \
       |    \   /    |    \
Wired         /\          G
       |    /    \   |
          /        \
       B - - - - - - D
```

C에는 두개의 네트워크가 접속되어 있습니다. 오른쪽에는 무선 네트워크 왼쪽에는 유선 네트워크

좀더 많은 인터넷 접속을 가질 수 있습니다. 그러기 위해서는 C에는 세 개의 네트워크 카드가 있어야 합니다.

각 호스트에는 어떻게 IP 주소를 설정해야 할까요? 두 가지 해결 방법이 있습니다.

1. 네트워크를 두 개의 서브네트워크로 나눕니다. : 예를 들면 192.168.1.0/24와 192.168.2.0/24처럼 말입니다. 이러한 방법은 굉장히 빠른 해결책입니다. 그러나 이러한 방법은 사용하지 않는 IP 주소를 만들게 되므로 모든 IP주소를 사용할 수 없게 됩니다.

2. 두개의 모든 인터페이스를 위해 C의 porxy APR를 가능하도록 설정합니다. Net address와 netmask와 같은 네트워크 parameters들은 유선이나 무선 네트워크가 동일하게 됩니다. 그러나 Proxy ARP는 유선과 무선 네트워크의 IP 주소를 선택할 수 있게 해 줍니다.

두 번째 방법을 실험해 보겠습니다.

가정 : 공용 subnet이 x.y.z.0/24이라고 가정합니다.

인터페이스는 다음과 같습니다.

1. ifconfig eth0 x.y.z.C netmask 255.255.255.255 (Wired)
2. ifconfig eth1 x.y.z.C netmask 255.255.255.255 (Wireless)
3. ifconfig eth2 x.y.z.C netmask 255.255.255.255 (to Internet)

eth2의 Static routes입니다.

1. route add IPGW dev eth2
2. route add default gw IPGW

이 route는 모든 인터넷 주소를 기본 게이트웨이에 요구한다는 것을 의미합니다. 이미 말한 바와 같이 리눅스에게 라우터가 어디에 있고, 기본 요구를 라우터에 해야 한다고 설정해야 합니다.

Eth0의 Static routes입니다.

1. route add x.y.z.A dev eth0
2. route add x.y.z.B dev eth0
3. route add x.y.z.D dev eth0

호스트 A,B,D는 유선 네트워크 입니다.

Eth1의 Static routes입니다.

1. route add x.y.z.E dev eth1
2. route add x.y.z.F dev eth1
3. route add x.y.z.G dev eth1

호스트 E,F,G는 무선 네트워크 입니다.

호환성은 매우 높습니다. 각각의 호스트에 수작업으로 설정을 다시 하실 수 있습니다.

6. 설치

여기에 비싸지 않는 무선 네트워크 카드를 설치하는 방법이 있습니다.(유용하기를 빌겠습니다.)

6.1. 일반적인 설치 정보

무선 네트워크 카드의 인터페이스는 일반적인 Ethernet 카드의 것과 유사합니다. 그러므로 `/etc/conf.modules`에 다음과 같은 사항을 추가할 수 있습니다.

1. `alias ethx module`, ethx에는 원하는 무선 네트워크의 인터페이스가 들어가고 module에는 커널 모듈의 이름이 들어갑니다.
2. `options module io=0xAAA irq=I ...`, 0xAAA에는 설치하는 카드의 IO base 주소가 들어갑니다. I는 IRQ번호 입니다. 그리고 다른 parameter들이 들어갈 수 있습니다.

`ifconfig`를 이용하여 위의 모든 설정을 할 수도 있습니다.

6.2. Proxim Symphony

네트워크 종류 : OpenAir 프로토콜을 사용하는 FHSS, Adhoc만 가능합니다.

http://www.proxim.com에서 원하는 리눅스 드라이버를 구할 수 있습니다.

드라이버는 소스로 제공되므로 컴파일 해야 합니다.

1. `untar`로 빈 디렉터리에 소스를 풉니다.
2. 도움말을 위해 `make`를 입력합니다.
3. 모듈을 만듭니다. : `make modules_install`를 입력합니다.
4. `/etc/conf.modules`을 수정한 후에 드라이버를 실행합니다. `ifconfig` 명령어로 실행할 수 있습니다.

r12cfg 유틸리티로 data-link level셋팅을 변경합니다.

1. `rl2cfg dev ethx sta` to set it to station (Slave)
2. `rl2cfg dev ethx msta` to set it to master station (Master)
3. `rl2cfg dev ethx alt` to set it to automatically mode

위의 과정이 전부입니다.

6.3. Webgear Aviator 2.4 and AviatorPro

네트워크 종류 : Aviator 2.4 는 FHSS, Adhoc만 Aviator 2.4는 Infrastructure로 가능합니다.

이 카드들을 사용하기 위해서는 좀더 많은 과정을 거쳐야 합니다. 왜냐하면 PCMCIA소스를 컴파일 해야 하기 때문입니다.

웹사이트는 다음과 같습니다. http://www.webgear.com

설정방법

1. PCMCIA 소스를 받아서 `/usr/src/pcmcia`에 풉니다.
2. 웹 사이트에서 드라이버 파일을 받습니다. 그리고, `tar zxvf driver.tgz`로 `/usr/src/pcmcia`에 풉니다.
3. PCMCIA를 다시 구성합니다.
4. `/etc/pcmcia/config.opts`에서 source `./ray_cs.opts`를 볼수 있습니다.
5. `/etc/pcmcia/ray_cs.opts`에 `module ray_cs opts ...`와 같은 줄을 발견할 수 있습니다. data-link level의 셋팅을 변경하기 위해서는 "opts"를 수정합니다.

논의

- pc_debug=x , where x is the log level
- net_type=x, x=0 for AdHoc, x=1 for Infrastructure
- essid=x, x is the ESSID

마지막으로 `pc_debug > 0`로 설정된 상태를 정검할 수 있습니다. 앞의 명령어를 실행하면 다음과 같은 data-link 메시지를 볼 수 있습니다. ; 새로운 카드가 생겼을 때, "network started" , 다른 새로운 무선 네트워크 카드가 연결 되었을 때, "network joined" 등

또한 `/proc/ray_cs` 파일로 도움을 얻을 수 있습니다. : 속해 있는 Subnet에서 잘못 된 BSSID의 정보를 제공해 줍니다.

6.4. Lucent Wavelan I, II, Orinoco products and Cabletron

네트워크 종류 : DSSS, Adhoc, Infrastructure

Lucent의 매우 전문적인 제품을 만듭니다.

웹사이트는 다음과 같습니다. http://www.lucent.com 또는 http://www.orinoco.net

설치 방법은 WebGear의 것과 유사 합니다. : WebGear의 단계 1,2,3을 따라 하시기 바랍니다.

그리고 `/etc/pcmcia/config.opts` 파일에 다음을 추가하시기 바랍니다. : Lucent Wavelan I module을 위해서 wavelan_cs opts ..., 그리고 Lucent Wavelan II or Orinoco을 위해서 wavelan2_cs opts ...

옵션(opt)은 다음과 같이 지정할 수 있습니다.

1. port_type=x : x가 Adhocd이면 3, Infrastructure이면 1
2. channel=x, x=channel, option은 AdHoc모드에서만 사용합니다.
3. transmit_rate=x : x는 사용하는 카드의 속도에 주의해서 정합니다.

이론상으로는 하나의 리눅스 컴퓨터에 두개의 Lucent 네트워크 카드를 설치하는 것이 가능합니다. 하나는 Adhoc모드로 또 하나는 Infrastructure모드로 할 수 있습니다. 하지만 오직 하나의 카드만이 동작 할 것입니다. 왜냐하면 pcmcia서비스가 시작될 때, 두개의 카드가 같은 옵션을 사용하기 때문입니다. 그래서, 새로운 리눅스 모듈을 만들어야 합니다.(또는 유저 모드의 프로그램일 수도 있습니다.) 이 모듈은 Adhoc모드의 access mode와 channel과 같은 data_link의 parameters들을 변경할 수 있어야 합니다.

channel parameter는 일반적으로 가까운 곳의 다른 무선 네트워크의 TX/RX를 간섭을 피하는 목적으로 사용됩니다.

Lucent드라이버는 Cabletron카드의 드라이버로도 사용할 수 있습니다. 해당 사이트는 다음과 같습니다. http://www.cabletron.com

6.5. YDI

네트워크 종류 : DSSS, Adhoc, Infrastructure

YDI는 전문적인 안테나를 장착한 카드 혹은 증폭기를 장착한 모델을 판매하는 회사입니다.

웹 사이트 : http://www.ydi.com

설치 방법

1. 빈 디렉터리에 untar 명령어로 드라이버 파일을 풉니다.
2. make 명령어로 컴파일 합니다.
3. make install 명령어로 am930_isa드라이버와 wlanctl유틸리티를 설치합니다.

수동으로 "wlanctl" 유틸리티를 써서 설정할 수 있고, `scripts/wlan`파일이나 `scripts/rc.wlan`을 써서 자동으로 네트워크를 구성할 수 있습니다.

수동 설정일 경우 세팅 방법은 다음과 같습니다.

1. `wlanctl scan ...` : 이미 있는 BSSs 들을 찾습니다.
2. `wlanctl netlist` : 1의 결과를 보여 줍니다.
3. `wlanctl bsscreate ... ssid` : ssid parameter로 새로운 네트워크를 만듭니다.
4. `wlanctl bssjoin bssid` bssid에 의해서 검증된 네트워크에 참가 합니다.
5. `wlanctl authen` 와 `wlanctl assoc` 서비스 승인을 위해서 사용합니다.

7. 무선에 관한 추가 정보

7.1. 무선 리눅스 배포본

무선 배포본인 FlyingLinux는 MobileIPv4와 DHCP-based wireless access의 이용 가능성을 연구하는 산학 협동 연구소인 Teleinformatics KTH 의 Telecommunication Systems Lab에서 1999년 10월 시작되었습니다.

이 연구의 결과로 백명의 학생들과 선생님들이 2000년 3월부터 5월까지 있었던 2G1303 프로젝트 코스동안 FlyingLinux 환경을 사용할 수 있었습니다.

FlyingLinux는 최소의 이동서비스에 기초한 리눅스입니다. 우리는 Kerberos support 와 OpenSSH를 포함하는 보안 문제에 관해서 주의를 기울이고 있습니다.

FlyingLinux는 오픈 소스 운동의 한 부분입니다. 우리의 소프트웨어는 GPL 라이센스를 기반으로 KTH에서 개발되고 있습니다.

무선 리눅스 배보본에 관한 사항은 다음의 사이트에서 찾을 수 있습니다.
http://www.flyinglinux.net Mantainer: Alberto Escudero Email (mailto:aep@it.kth.se), Home Page (http://www.it.kth.se/~aep)

8. FAQ

질문 : BBSID와 ESSID의 차이점은 무엇인가? 언제 ESSID가 필요한가?

답변 : BBSID는 BSS 좁은 공간을 구별하는 48비트의 숫자입니다. 이 안에서 모든 호스트들은 ESSID를 가지고 통신할 수 있게 된다. ESSID는 가변길이로 서로 다른 BSS사이의 통신을 가능하게 하는 것으로 BSS는 확장되어 ESS가 된다. 각각의 BSS를 위한 Access Point가 있고, 모든 BSS들은 같은 ESS를 통하여 통신하게 된다. 굉장히 큰 네트워크에 2개 이상의 Access Point를 가지고 있다면 ESSID가 필요할 것입니다.

질문 : Access Point를 사야만 하나?

답변 : 당신이 발견한 제품이 다소 비싸다면, Access Point를 사용하는 가장 중요한 이유는 카드와 Access Point가 같은 물리 layer를 사용하는 것입니다. 또한 모든 FHSS나 DSSS와 호환됩니다. Proxim RangeLan2카드는 표준 FHSS와 호환이 안됩니다. 이유는 OpenAir 프로토콜을 사용하기 때문입니다.

질문 : 설정된 채널을 무엇을 위해 사용되는가?

답변 : 다른 BSS를 사용하는 많은 네트워크가 있고, 충돌 문제를 가지고 있다면, 액세스 포인트의 채널을 바꾸거나 Adhoc 모드의 호스트들이 이러한 문제를 해결하는데 도움을 줄 것입니다.

질문 : 왜 채널이나 Infrastructure host들을 설정할 수 없나?

답변 : 왜냐하면 채널은 Access Point에 의해서 결정되기 때문입니다.

9. Appendix A - Netmask 255.255.255.255, proxy arp and bridging

Here we view some Linux advantages in Wireless Internetworking.

Linux let you specify a netmask like 255.255.255.255 for an interface which can help you assign IP addresses in to any interface you want, for example one in eth0, another in eth1 and so on...

This has not particularly side-effects.

In addition you have proxy arp setting under /proc/sys/net/ipv4/conf/ethx/proxy_arp where ethx if your interface.

If you "echo 1 > proxy_arp" you enable proxy_arp for that interface while with "echo 0 > proxy_arp" you disable it.

What's proxy_arp? Quickly proxy arp help you when you want a router answer to an ARP request if the destination address is in another interface of the linux router.

Example:

192.168.1.1 ---- 192.168.1.2 Linux router 192.168.2.2 ----192.168.2.1

To get this example working you should:

Without proxy-arp

1. In 192.168.1.1 host to set 192.168.1.2 as gateway
2. In 192.168.2.1 host to set 192.168.2.2 as gateway
3. pinging with success from any edge.

With proxy-arp

1. In 192.168.1.1 host to set 192.168.1.2 as gateway
2. do not set gateway for 192.168.2.1 host but enable proxy_arp for right interface of the router.
3. pinging with success from any edge

Proxy-arp in 2. case let the linux router answer when you ping from 192.168.2.1 host, saying that it has the 192.168.1.1 host so it can answer for it. After, when the source start sending ICMP packet, Linux router knows that it have to redirect it to the real host 192.168.1.1

In Wireless network proxy arp can help you if you have many Linux boxes that acts as IP Forwarders and you wouldn't set to all hosts a number of static route.

You also can experiment Linux bridging in Wireless network:

1. install a recent stable kernel

2. download good bridge utils at http link (http://openrock.net/bridge) or ftp link (ftp://openrock.net/bridge)

Bridging should be more simple to manage.

10. Appendix B - Siemens DECT Radio Modem

Web Site: http://www.siemens.com

What's that? These 2 components are not real PC cards but are more like modems that you use as external device.

Host1-serial - RadioModem1 - - - - - RadioModem2 - serial-Host2

How can I connect it?

If you see them in an abstract vision you can model them like this:

Host1-serial - - NULL MODEM CABLE - - serial-Host2

So you'd have a connection between 2 far serials with 2 possible configurations:

1. Linux with Windows, Linux has a daemon that answer to a ppp call while Windows has a Dial up connection under Remote Access.
2. Linux with Linux, where you can run (on all 2 hosts) a ppp connection, with IP addresses inverted.

For 1 you can use this simple script in Linux:

"/usr/sbin/pppd -detach lock idle 300 crtscts connect "/usr/sbin/chat -v TIMEOUT 5 AT OK AT OK AT OK AT OK" IPLINUX:IPWINDOWS /dev/ttySx 115200 disconnect "/usr/sbin/chat -v AT OK" ms-dns IPDNS"

where:

- /dev/ttySx is your serial port,
- IPDNS is the IP address of your dns server,
- IPLINUX is the Linux IP address as IPWINDOWS is the Windows IP address.

The above script is need for letting Windows believe there is a modem on the serial end!

In Windows you need to create a Dial Up connection with crtscts enabled, speed at 115200 and with a stupid number to call (need by Remote Access but absolutely not used).

With 2 Linux boxes you only have to run a very simple script like this at each Linux side:

"/usr/sbin/pppd passive local crtscts IPLINUX1:IPLINUX2 /dev/ttySx 115200 noauth persist" where you'll invert IPLINUX1 with IPLINUX2 at the other edge.

Note that you can do authentication even Linux-Windows than in Linux-Linux if you want.