The Linux Networking Overview HOWTO

Daniel Lopez Ridruejo, *ridruejo@rawbyte.com* <mailto:ridruejo@rawbyte.com> v0.31, 17 September 1999

남상현 (*nsh@asp-linux.co.kr* <mailto:nsh@asp-linux.co.kr>) v0.31 2000년 7월 29일

이 문서의 목적은, Linux operating system의 네트워크 능력의 개요를 전하고, 좀 더 나아가 정보와 구체적인 방식을 조사하는 장소를 소개하는 것이다.

목 차

제 1 절 도입

이 문서의 목적은, Linux operating system의 네트워크 능력의 개요를 전하는 것이다. Linux의 힘은 정보를 풍부하게 입수할 수 있는 것이다. 대부분의 관련 정보는 다수 입수할 수 있지만, 정보의 대부분은 구현에 초점이 맞혀 있다. Linux 초보자에 있어서는, 특히 Windows 환경으로부터 이동했던 방법은, Linux가 네트워크에서 할 수 있는 것을 모르게 한다. 그래서, 본 문서에서는, Linux가 네트워크에서 할 수 있는 것의 전체를 대강 긍칸터보는 것에있다. 각각의 항목 설명은 간결하지만, 좀 더 나아가 정보를 찾을수 있는 장소를 소개 한다. 이 정보는, HOWTO , faq , 각 프로젝트 의 Web 페이지, 저의 경험등, 많은 정보원으로부터 모은 것이다. 이것 들의 정보는 모두가 그 저자들덕분이다. 이 저자들과 활동이 없었으면 본 문서는 만들 수 없었을 것이고, 처음부터 필요도 없었을 것이다.

제 2 절 Linux

2.1 Linux 란 무엇인가?

Linux의 저자의 처음는 Linus Torvalds 씨이다. Linus 씨가 최초의 버젼을 만든 이래, 무수한 사람들의 힘으로 Linux는 진보해 왔다. Linux 는 전적으로 scratch부터 만들어진 Unix operating system의 clone 이다. Linux 에서 가장 흥미있는 것의 하나는 온 세계적으로 동시에 개발이 행해 진다는 것이다.

Linux는 **GNU General Public License (GPL)**의 저작권을 따른다. GPL은 Free Software Foundation (FSF)에의해서 기록된 라이센스로, 소프트웨어의 자유로운 배포를 제한하지 않도록 되어 있다. 간단하게 말하면 이 라이센 스에서는, 소프트웨어의 사본에 요금을 청구할 수 있지만, 그 사본을 취한 사람이 그 사본을 무료로 배포하는것을 금지하지 않는다. 또, 이 라이센스에서는 소스 코드가 반드시 공개되어야한다. 이것은 프로그램 에 도움이 될 것이다. 같은 저작권을 따르고 있는 한, 누구라도 Linux에 수정 을 추가 할 수 있고, 수정 프로그램의 배포도 할 수 있다.

2.2 무엇이 Linux를 다르게 만드나?

왜 Linux를 사용하가? Linux는 다른 operating system보다 일반적으로 염가(적어도 고가는 아닌)이고, 많은 상용 시스템보다 문제가 적은 것으로 되어있다. 하지만 Linux를 구별할수 있는 것은 비용 보다는(결국은 OS가 무료라도, 만족할 수 없다면 누가 원하겠는가?) Linux 의 뛰어난 가능성이다.

- Linux는 진정한 32비트 멀티테스크 operating system입니다. 대 학에서부터 대기업까지, 넓은 범위로 충분히 사용할 수 있는 힘과 능력이 있다. (역주 : 64bit 판도 있다.)
- Linux는 low-end 386박스 연구에서 부터 대규모의 ultra-parallel 머신의 하드웨어에서 실행된다.

- Out-of-the-box 버전은 Intel, Sparc, Alpha 구조에서 가능하다. 그리고, Power PC와, SGI, Ultra Sparc, AP1000+, Strong ARM, MIPS R3000/R4000와 기타 내장 시스템에서 실험적으로 서포트 되고 있다.
- 마지막으로, 네트워크를 구성 할 때에는 Linux가 적절하다. 그 이유는, 네트워크 기능이 OS 자체에 내장되어 있는 것과, 많은 어플리케이션을 free로 입수 할 수 있는 것도 있지만, 그것이 다는 아니다. 과중한 부하에도 견디는 힘이 이유이다. 이것 은 오픈 소스의 프로젝트 내에서 몇 년의 디버깅과 테스트를 경과해 처음으로 얻어진 것이다.

제 3 절 네트워크 프로토콜

Linux는 다양한 네트워크 프로토콜을 많이 서포트 한다.

3.1 TCP/IP

인터넷 프로토콜(Internet Protocol)은, 최초에는 미국 국방성(DoD)용으로 20년전에 개발된 것이다. 이것은 이기종 컴퓨터간의 상호 접속이 주요한 목적이었다. TCP/IP suite 프로토콜은 계층 구조이므로, 어플리케이션 을 네트워크용 하드웨어로부터 분리할 수 있다.

TCP/IP는 계층적모델을 기본 으로하지만, 기능적계층을 엄격하게 따르는 것보다는 실제로 정확하게 연결되는 것을 중요시하고 있다. 이러한 이유에 의해, 네트워크의 상호 접속 프로토콜은 OSI가 아닌 TCP/IP가 사실상의 표준으로 되어졌다.

Linux는 등장했을 때부터 TCP/IP 네트워크 기능을 가지고 있었다. 이것 은 scratch로부터 실행되었다. 이것은 가장 강하고, 빠르고, 신뢰성이 높은 구현으로, Linux 성공 열쇠의 하나이다.

관련있는 HOWTO :*metalab.unc.edu/mdw/HOWTO/NET-3-HOWTO.html* `<http://metalab.unc.edu/mdw/HOWTO/NET-3-HOWTO.html>`

3.2 TCP/IP 제 6판

IPv6은 IPng (IP Next Generation)라는, IPv4 프로토콜에서 문제 가 되고 있던 많은 점을 개선한 upgraded판이다. IPv4에서의 문제 란, 이용가능한 IP 어드레스의 부족, 시간에 민감한 트래픽을 취급하는 메카니즘 부재, 네트워크 layer에서의 보안이 없는 등이다.

보다 큰 어드레스 공간을 사용하려면, 어드레스 지정 방법의 개선이 동반하지만, 이것은 루틴의 성능에 크게 영향을 주게 된다. Linux용 베타 판에서는 구현 되어 있고, 실용판은 2.2.0의 Linux 커넬 배포에 내장 되리라 예 상된다.

- Linux IPv6 의 HOWTO : linux-ipv6.faq.htm
- IPv6의 시작 페이지 : *playground.sun.com/pub/ipng/html/ipng-main.html* `<http://playground.sun.com/pub/ipng/html/ipng-main.html>`

3.3 IPX/SPX

IPX/SPX (Internet Packet Exchange/Sequenced Packet Exchange)는, Xerox Network Systems (XNS) 프로토콜을 기본 으로하여 Novell이 개발한 프로트콜군으로서, 소유권이 첨부된 것이다. IPX/SPX는 1980 년대 처음에 Novell, Inc.의 NetWare에서 사용된 프로토콜로서 유명하게 되었다. NetWare는 제 일세대로 LAN의 network operating system (NOS)인 de facto 표준이 되었다. Novell은 자신의 NOS에, 비지니스형 어플리케이션군과 클라이언트측에서의 접속 유틸리티를 추가하였다.

Linux는 매우 안전한 IPX/SPX가 구현되어 있고, 설정에 의해 다음의 것 에 사용할 수 있다.

- IPX router
- IPX bridge
- NCP client and/or NCP Server (for sharing files)
- Novell Print Client, Novell Print Server

  또,
- PPP/IPX를 유효하게 하고, Linux 박스를 PPP의 서버 / 크라이언 트로서 사용할 수 있다.
- IP 내에서 IPX를 tunnelling시키는 것으로서, IP를 통한 유일한 링크를 허가해 두개의 IPX의 네트워크를 접속할 수 있다.

  그것에 추가해서, Caldera *www.caldera.com* `<http://www.caldera.com>`는 Linux에서의 Novell NetWare를 상용 지원하고 있다. Caldera는 Novell 사로부터 허가를 받은 기술을 내장해 완전한 Novell NetWare 클라이언트를 공급하고 있다. 이 클라이언트는 Novell 3.x 와 4.x 파일 서버로 완전한 접근을 제공하고, NetWare 디렉토리 서비스(NDS) 와 RSA 암호화기능 등도 있다.
- IPX 의 HOWTO: *metalab.unc.edu/mdw/HOWTO/IPX-HOWTO.html* `<http://metalab.unc.edu/mdw/HOWTO/IPX-HOWTO.html>`

3.4 AppleTalk Protocol Suite

AppleTalk는 Apple의 internetworking 기술의 이름이다. 이것은 peer-to-peer 네트워크 모델로, 파일과 프린터의 공유등의 기본적인 것을 할 수 있다. 각 머신은 동시에 클라이언트와 서버에서 동시에 활동할 수 있고, 이것 에 필요한 소프트웨어와 하드웨어는 어느 Apple 컴퓨터에도 들어있다.

Linux는, AppleTalk 네트워크를 완벽하게 제공할 수 있다. Netatalk는 최초에 BSD 계통의 시스템용이었던 것에서, AppleTalk Protocol Suite를 커넬레벨에서 실현한 것이다. Netatalk를 사용해, AppleTalk의 routing, Unix 와 AFS의 파일 시스템인 AFP (AppleShare)상에서 제공, Unix 프린터와 AppleTalk 프린터에로의 억세스를 PAP 상에서 제공 할 수 있다.

5.1 절에 좀 더 많은 정보가 있으므로 보시오.

3.5 WAN Networking: X.25 , Frame-relay, 기타

Linux 용 T-1 와, T-3 , X.25 , Frame Relay의 제품을 몇몇 서드파티가 공급하고 있다. 이것들의 타입으로 접속하려면 특별한 하드 웨어가 일반적으로 필하다. 그 하드웨어의 Vendor는 프로토콜 지원 드라이버도 공급 한다

o Linux 용의 WAN 관계 정보원 : *www.secretagent.com/networking/wan.html* `<http://www.secretagent.com/networking/wan.html>`

3.6 ISDN

Linux 의 커널에는 내장 ISDN 기능이 있다. isdn4linux에서 ISDN PC 카드를 콘트롤해, Hayes 커멘드 세트("AT" 명령어) 로 모뎀 에뮬레이터를 실행 할 수 있다. (내장 디바이스를 사용해) HDLC 를 경유하여 접속된 단말 프로그램을 이용하여, PPP에 의한 인터넷의 완전한 접속, 오디오 어플리케이션까지, 넓은 범위의 일을 할 수 있다.

o isdn4linux 의 FAQ : *www.lrz-muenchen.de/~ui161ab/www/isdn/faq_e.html* `<http://www.lrz-muenchen.de/~ui161ab/www/isdn/faq_e.html>`

3.7 PPP , SLIP , PLIP

Linux 커넬에는 내장 PPP (Point-to-Point-Protocol)와, SLIP (Serial Line IP) , PLIP (Parallel Line IP)의 지원을 한다. PPP는 개인이 자신의 ISP (Internet Service Provider)로서 억세스 에 제일 자주 사용되는 방법이다. PLIP는 2개의 머신을 돈을 들이지 않고 접 속할 수 있는 방법이다. PLIP는 패러렐 포트와 특수 cable를 사용해 10kBps 로부터 20kBps 의 속도가 나온다.

- Linux PPP HOWTO *metalab.unc.edu/mdw/HOWTO/PPP-HOWTO.html* `<http://metalab.unc.edu/mdw/HOWTO/PPP-HOWTO.html>`
- PPP/SLIP emulator *metalab.unc.edu/mdw/HOWTO/mini/SLIP-PPP-Emulator.html* `<http://metalab.unc.edu/mdw/HOWTO/mini/SLIP-PPP-Emulator.html>`
- PLIP의 정보는 다음 사이트에서 찾을 수 있다. The Network Administrator Guide *metalab.unc.edu/mdw/LDP/nag/nag.html* `<http://metalab.unc.edu/mdw/LDP/nag/nag.html>`

3.8 아마츄어 무선

Linux 커넬에는 내장된 아마츄어 무선 프로토콜이 지원되고 있다.

특히 흥미있는 것은 AX.25의 지원이다. AX.25 프로토콜은 connection 형과 connectionless형의 모두를 제공한다. 이것 만으로 point-point 링크를 하거나, TCP/IP 와 NetRom 등의 다른프로트콜을 옮기는데 사용한다.

AX.25는 구조적으로는 X.25 level 2와 같지만, 아마츄어 무선 환경에 적합한 몇개의 확장을 가지고있다.

o Amateur radio HOWTO *metalab.unc.edu/mdw/HOWTO/HAM-HOWTO.html* `<http://metalab.unc.edu/mdw/HOWTO/HAM-HOWTO.html>`

3.9 ATM

Linux에서 ATM 지원은 현재는 pre-alpha 단계이다. 실험적인 배포가 있어, 가공하지 않은 ATM 접속(PVC 와 SVC)이나, ATM 상에서의 IP , LAN 에뮬레이션등을 지원하고 있다.

o Linux ATM-Linux home page *lrcwww.epfl.ch/linux-atm/* `<http://lrcwww.epfl.ch/linux-atm/>`

제 4 절 지원되는 네트워크용 하드웨어

Linux는, 진부한 것을 포함해 매우 다양한 네트워크용 하드웨어를 지원한다.

약간의 흥미로운 문서

- Hardware HOWTO *metalab.unc.edu/mdw/HOWTO/Hardware-HOWTO.html* `<http://metalab.unc.edu/mdw/HOWTO/Hardware-HOWTO.html>`
- Ethernet HOWTO *metalab.unc.edu/mdw/HOWTO/Ethernet-HOWTO.html* `<http://metalab.unc.edu/mdw/HOWTO/Ethernet-HOWTO.html>`

제 5 절 파일과 프린터의 공유

PC를 기본로한 Local Area Networks가 많은것은, 파일 공유와 프린터 공유 서비스를 사용자에게 제공하는 것을 제일의 목적으로 하기 때문이다. 기업에서 파일과 프린터 공유의 서버로서 Linux는 훌륭한 해결책이 된다.

5.1 Apple 환경

이전 섹션에서 설명한 것처럼, Linux는 AppleTalk의 프로토콜 패밀리를 지원 한다. Linux의 netatalk를 사용하면, Macintosh의 클라이언트로부터 본 Linux 시스템을 네트워크상의 또 하나의 Macintosh로서 인 식할 수 있고, Linux 서버에 접속된 파일과 프린터를 공유할 수 있다.

Netatalk faq 와 HOWTO

- *thehamptons.com/anders/netatalk/* `<http://thehamptons.com/anders/netatalk/>`
- *www.umich.edu/˜rsug/netatalk/* `<http://www.umich.edu/˜rsug/netatalk/>`
- *www.umich.edu/˜rsug/netatalk/faq.html* `<http://www.umich.edu/˜rsug/netatalk/faq.html>`

5.2 Windows 환경

samba는, 대부분 Unix 들(특히 Linux)를 Microsoft 네트워크의 클라이언트로도 서버로도 통합화할 수 있는 어플리케이션 suite이다. 서버에 사용하면, Windows 95 와, Windows for Workgroup , DOS , Windows NT의 클라이언트가 Linux의 파일과 프린터 서비스에 억세스할 수 있다. samba는, 파일과 프린터 서비스에 대해서 Windows NT 를 완전히 바꿀수 있고, 클라이언트로 프린터 드라이버를 자동적으로 다운로드 할 수 있다. 클라이언트에 사용하면, 로컬에 export한 Windows 공유 파일을 Linux 워크스테이션이 마운트할 수 있다.

```
SAMBA Meta-FAQ에 의하면

     「 많은 유저의 보고에 의하면, 다른  SMB의 실행과 비교해 Samba
     는보다 안정되고, 보다 빠르고, 보다 많은 클라이언트와 호환성이 있다.
     crashing 없이 수만의 유저에게까지 규모 확대가 가능한
     유일의 SMB 서버가 Samba다, 라고 대규모 설치 관리자들은 말한다. 」
```

- Samba project home page *samba.anu.edu.au/samba/* `<http://samba.anu.edu.au/samba/>`
- SMB HOWTO *metalab.unc.edu/mdw/HOWTO/SMB-HOWTO.html* `<http://metalab.unc.edu/mdw/HOWTO/SMB-HOWTO.html>`
- Printing HOWTO *metalab.unc.edu/mdw/HOWTO/Printing-HOWTO.html* `<http://metalab.unc.edu/mdw/HOWTO/Printing-HOWTO.html>`

5.3 Novell 환경

이전의 섹션에서 말한 것처럼, Linux는 NCP의 클라이언트 또는 서버로서 적용할 수 있기 때문에, Novell의 네트워크상에서 Novell 와 Unix의 양쪽 클라이언트에 파일과 프린터를 서비스할 수 있다.

o IPX HOWTO *metalab.unc.edu/mdw/HOWTO/IPX-HOWTO.html* `<http://metalab.unc.edu/mdw/HOWTO/IPX-HOWTO.html>`

5.4 Unix 환경

Unix의 네트워킹 환경에서는 NFS를 사용한 파일 공유가 자주 행해진다. NFS는 네트워크에서 파일을 공유하는 것으로, 근원은 Sun Microsystems가 개발한 프로토콜이다. NFS 에서는 서로의 파일을 로컬 파일에 있는것과 같이 공유 한다 NFS 서버가 "export" 하는 파일 시스템을, 클라이언트가"mount" 한다. 마운 트된 파일 시스템은, 클라이언트 기계로부터는 로컬 파일시스템의 일부분과 같게 보인다.

기계를 시작할 때에 루트의 파일 시스템을 마운트할 수 있기 때문에, diskless의 클라이언트를 시작하고, 서버의 모든 파일로 억세스 할 수 있다. 즉, 하드 디스크 없이 충분히 기능적인 컴퓨터로 될 수 있다.

Coda는(NFS 같은)네트워크 파일 시스템에서, 비접속 상태로 조작, persistant caching, 그 외의 편리한 것을 서포트 한다. 이것은 2.2.x의 커넬내에 포함되어있다. 저속이나 신뢰할 수 없는 네트워크, 또, 랩탑용으로 매우 적당하다.

NFS 관련 문서

- *metalab.unc.edu/mdw/HOWTO/mini/NFS-Root.html* <http://metalab.unc.edu/mdw/HOWTO/mini/NFS-Root.html>
- *metalab.unc.edu/mdw/HOWTO/mini/Diskless.html* <http://metalab.unc.edu/mdw/HOWTO/mini/Diskless.html>
- *metalab.unc.edu/mdw/HOWTO/mini/NFS-Root-Client.html* <http://metalab.unc.edu/mdw/HOWTO/mini/NFS-Root-Client.html>
- *www.redhat.com/support/docs/rhl/NFS-Tips/NFS-Tips.html* <http://www.redhat.com/support/docs/rhl/NFS-Tips/NFS-Tips.html>
- *metalab.unc.edu/mdw/HOWTO/NFS-HOWTO.html* <http://metalab.unc.edu/mdw/HOWTO/NFS-HOWTO.html>

CODA는 여기서 찾을수 있다 : *www.coda.cs.cmu.edu/* <http://www.coda.cs.cmu.edu/>

제 6 절 인터넷 / 인트라네트

Linux는 인트라네트 / 인터넷의 서버용으로 사용할 수 있는 뛰어난 platform이다. 인트라네트라는 용어는, 조직 내부에서의 정보 유통과 공유 를 주목적으로 인터넷 기술을 사용한다는 의미이다. Linux가 할 수 있는 인터넷과 인트라네트 서비스는, 아래의 섹션에서 설명하는, Mail , News , WWW 서버 그외 상당수가 있다.

6.1 Mail

6.1.1 Mail 서버

Sendmail는 Unix platform을 위한 de facto 표준 메일 서버 (Mail Transport Agent , MTA 라고 한다)이다. Sendmail는 robust, scalable, 필요한 하드웨어를 적절히 설정하면 수천 유저의 메일을 간단하게 취급할 수 있다. sendmail의 대안으로 설계되어 있는 smail 과 qmail 등의 메일서버도 있다.

- Sendmail web site *www.sendmail.org/* <http://www.sendmail.org/>
- Smail faq *www.sbay.org/smail-faq.html* <http://www.sbay.org/smail-faq.html>
- Qmail web site *www.qmail.org* <http://www.qmail.org>

Mail HOWTOs:

- *metalab.unc.edu/mdw/HOWTO/Mail-HOWTO.html* <http://metalab.unc.edu/mdw/HOWTO/Mail-HOWTO.html>
- *metalab.unc.edu/mdw/HOWTO/mini/Qmail+MH.html* <http://metalab.unc.edu/mdw/HOWTO/mini/Qmail+MH.html>

- *metalab.unc.edu/mdw/HOWTO/mini/Sendmail+UUCP.html* <http://metalab.unc.edu/mdw/HOWTO/mini/Sendmail+UUCP.html>
- *metalab.unc.edu/mdw/HOWTO/mini/Mail-Queue.html* <http://metalab.unc.edu/mdw/HOWTO/mini/Mail-Queue.html>

6.1.2 Remote access to mail

기업 과 ISP에서는 대부분, 유저는 자신의 메일을 자신의 디스크로부터 멀리 접근 할 수 있다. 이것을 하는데 POP (Post Office Protocol)와 IMAP (Internet Message Access Protocol) 서버를 포함한, 몇개의 대안이 Linux에 있다. POP 프로토콜은 서버로부터 클라이언트로 메세지를 전송하는데 자주 사용한다. IMAP는, 서버내의 메세지를 조작하거나, 서버내의 폴더를 리모트로 작성, 삭제하거나, 공유메일 폴더로 동시 접근 등을 할 수 있다.

- IMAP 와 POP 의 간단한 비교 *www.imap.org/imap.vs.pop.brief.html* <http://www.imap.org/imap.vs.pop.brief.html>

메일관련 HOWTO

- *metalab.unc.edu/mdw/HOWTO/Mail-HOWTO.html* <http://metalab.unc.edu/mdw/HOWTO/Mail-HOWTO.html>
- *metalab.unc.edu/mdw/HOWTO/mini/Cyrus-IMAP.html* <http://metalab.unc.edu/mdw/HOWTO/mini/Cyrus-IMAP.html>

6.1.3 Mail User Agent

Linux에는 다수의 MUA (Mail User Agent)가 있다. graphical와 text mode 의 양쪽이다. 가장 넓리 사용되는, pine 와, elm , mutt , Netscape가 있다.

- 메일관련 소프트웨어의 일람 *www.linuxlinks.com/Software/Internet/Mail/* <http://www.linuxlinks.com/Software/Internet/Mail/>
- *metalab.unc.edu/mdw/HOWTO/mini/TkRat.html* <http://metalab.unc.edu/mdw/HOWTO/mini/TkRat.html>

6.1.4 메일링 리스트 소프트웨어

메일링 리스트를 관리하는 MLM (Mail List Management) 프로그램 은, Unix에서 일반적으로 이용되는 것과 Linux 특유의 것이 다수 있다.

- 현존하는 MLM를 효과적으로 비교 대조한 것을 여기서 찾을 수 있다 : *ftp.uu.net/usenet/news.answers/mail/list-admin/* <ftp://ftp.uu.net/usenet/news.answers/mail/list-admin/>
- Listserv *www.lsoft.com/* <http://www.lsoft.com/>
- Majordomo home page *www.greatcircle.com/majordomo/* <http://www.greatcircle.com/majordomo/>

6.1.5 Fetchmail

fetchmail은 메일관련의 편리한 유틸리티이다. 이것은(SLIP 와 PPP 접속등의)on-demand TCP/IP 링크로 사용하는 것을 목적으로, 리모트 메일을 수집하거나 포워드하는 것이다. 이것은 free, 복합적인, robust 인 문서가 다수 있다. fetchmail은 인터넷에서 현재 이용되는 리모트 메일의 프로토콜 모두를 서포트 한다. 심지어 IPv6 와 IPSEC도 서포트 한다

fetchmail는 리모트 메일 서버로부터 메일을 모으고, SMTP에 의해 포워드 한다. 그렇기 때문에 포워드된 뒤로는 mutt 와, elm , BSD Mail 등의 MUA를 사용해 읽을 수 있고, 시스템 MTA는 보통 MTA's filtering, forwarding, aliasing 기능 모두를 같은 것처럼 사용할 수 있다.

fetchmail를 사용해, ISP의 하나의 메일박스로부터 메일을 모으고, 헤더의 어드레스에 기초를 두고 SMTP로 포워드시키도록 하면, 하나 의 DNS 도메인 전체에서 POP/IMAP로부터 SMTP에로의 게이트웨이가 된다.

보내는 메일을 전부 ISP의 하나의 메일박스에 모아 두고, 그곳 으로부터 인터넷으로 보내고, 들어 오는 메일을 같은 메일박스로부터 전부 취득하도록 fetchmail를 설정할 수 있기 때문에, 소기업의 메일을 하나의 메일박스에 집중시킨다.

o Fetchmail home page *sagan.earthspace.net/~esr/fetchmail/* `<http://sagan.earthspace.net/~esr/fetchmail/>`

6.2 Web 서버

대부분의 Linux 배포본에는 Apache *www.apache.org* `<http://www.apache.org>`가 들어 있다. Apache는 http://www.netcraft.co.uk/survey/에 의하면 인터넷에서 최고의 서버로, 인터넷 사이트의 반 이상이 Apache나 파생된것으로 운영된다. Apache는 모듈러화된 설계, 안정성과 속도가 뛰어나다. 적합한 하드웨어를 적절히 설정하면 Apache는 최대한의 부하를 처리한다. Yahoo , AltaVista , GeoCities , Hotmail는 이 서버 를 주문 제작한 버전을 사용하고 있다.

Apache는 옵션으로 SSL (안전한 transactions을 가능하게 한다)를 사용할 수 있다. 이 지원은 다음 사이트에서 입수할 수 있다.

- *www.apache-ssl.org/* `<http://www.apache-ssl.org/>`
- *raven.covalent.net/* `<http://raven.covalent.net/>`
- *www.c2.net/* `<http://www.c2.net/>`

관련된 HOWTO

- *metalab.unc.edu/mdw/HOWTO/WWW-HOWTO.html* `<http://metalab.unc.edu/mdw/HOWTO/WWW-HOWTO.html>`
- *metalab.unc.edu/mdw/HOWTO/Virtual-Services-HOWTO.html* `<http://metalab.unc.edu/mdw/HOWTO/Virtual-Services-HOWTO.html>`
- *metalab.unc.edu/mdw/HOWTO/Intranet-Server-HOWTO.html* `<http://metalab.unc.edu/mdw/HOWTO/Intranet-Server-HOWTO.html>`
- Linux를 위한 Web server *www.linuxlinks.com/Software/Internet/WebServers/* `<http://www.linuxlinks.com/Software/Internet/WebServers/>`

6.3 Web 브라우저

Linux platform 용의 Web 브라우저는 많이 있다. Netscape Navigator는 처음부터 선택되어 온 것이고, 이번에 공개된 Mozilla (http://www.mozilla.org) 에는 Linux 판이 있을 것이다. 또 하나의 자주 사용되고 있는 텍스트 베이스의 Web 브라우저는 lynx 이다. 이곳은 그래픽 환경이 사용할 수 없는 경우에 고속으로 사용할 수 있다.

- Linux 용 브라우저 소프트 웨어 *www.linuxlinks.com/Software/Internet/WebBrowsers/* `<http://www.linuxlinks.com/Software/Internet/WebBrowsers/>`
- *metalab.unc.edu/mdw/HOWTO/mini/Public-Web-Browser.html* `<http://metalab.unc.edu/mdw/HOWTO/mini/Public-Web-Browser.html>`

6.4 FTP 서버와 클라이언트

FTP란 파일 전송 프로토콜(File Transfer Protocol)이다. 클라이언트가 FTP 서버에 접속해서 파일을 꺼낼 수 있다.(다운 로드 할 수있다.) Linux 용 FTP 서버와 클라이언트는 다수 있고, 대부분의 배포판에 보통은 들어 있다 . 텍스트 베이스의 클라이언트도 GUI 베이스의 것도 있다. Linux 용 FTP 관련 소프트웨어는(서버 와 클라이언트) *metalab.unc.edu/pub/Linux/system/network/file-transfer/* `<http://metalab.unc.edu/pub/Linux/system/network/file-transfer/>`에서 찾을 수 있다.

6.5 News 서비스

Usenet (news 로서도 알려진)는 큰 게시판 시스템 (bulletin board system)이다. 모든 종류의 화제를 망라하고, 계층적으로 조직화 되어 있다. 인터넷상에서 컴퓨터의 네트워크(Usenet) 가 기사를 교환할 때에는 NNTP 프로토콜을 사용한다. Linux 용 NNTP를 구현한 것은 몇개가 있다. 무거운 부하의 사이트용도, 소수의 뉴스 그룹 만을 수신하는 작은 사이트용도 있다.

- INN home page *www.isc.org/* `<http://www.isc.org/>`
- Linux의 news 관련 소프트웨어 *www.linuxlinks.com/Software/Internet/News/* `<http://www.linuxlinks.com/Software/Internet/News/>`

6.6 Domain Name System

DNS 서버는 IP 어드레스를 이름(사람이 읽을 수 있다)으로 번역하는 것이다. DNS 서버는 온 세상의 IP 어드레스 전부 알지 못하지만, 모르는 어드레스를 다른 서버에게 물을 수 있다. DNS 서버는, 요구된 IP 어드레스나, 테이블 중 에 그 이름이 눈에 띄지 않다고 말하는 레포트중, 어느 것을 유저에 돌려준다.

Unix(와 인터넷의 대다수)의 네임 서비스는 named라는 프로그램으로 실행되어 진다. 이것은 The Internet Software Consortium 의 bind 패키지의 일부이다.

- BIND *www.isc.org/* `<http://www.isc.org/>`
- DNS HOWTO *metalab.unc.edu/mdw/HOWTO/DNS-HOWTO.html* `<http://metalab.unc.edu/mdw/HOWTO/DNS-HOWTO.html>`

6.7 DHCP , bootp

DHCP 와 bootp 는, 클라이언트가(자신의 IP 번호등)네트워크 정보 를 서버로부터 받을 수 있는 프로토콜이다. 특히 대규모 네트워크와 유동적인 유저를 가지고 있는 네트워크에서는, 네트워크 관리가 간단하므로 , 많은 조직에서 이것을 사용하기 시작 했다.

관련 문서

o DHCP HOWTO *metalab.unc.edu/mdw/HOWTO/mini/DHCP.html* `<http://metalab.unc.edu/mdw/HOWTO/mini/DHCP.html>`

6.8 NIS

Network Information Service (NIS)는 네트워크상의 정보를 찾는 단순한 서비스로서, 데이타베이스와 processes로 구성된다. 네트워크 전체에서 알아 두어야 할 정보를 네트워크상의 모든 기계에 제공하는 것이 NIS의 목적 이다. 예를 들면, NIS를 실행하고 있는 네트워크에서는 관리자가 네트워크 내의 기계 전부에 패스워드 엔트리를 추가해 둘 필요가 없어진다. 메 인 데이타베이스에 패스워드 엔트리를 두는 것만으로 임의의 기계로 로그 인 할 수 있다.

관련 HOWTO

o NIS HOWTO *metalab.unc.edu/mdw/HOWTO/NIS-HOWTO.html* `<http://metalab.unc.edu/mdw/HOWTO/NIS-HOWTO.html>`

6.9 인증 (Authentication)

기종이 혼재된 네트워크에서 유저의 인증을 하는 방법도 다양하다.

- Linux/Windows NT 용으로는 :*www.mindware.com.au/ftp/smb-NT-verify.1.1.tar.gz* `<http://www.mindware.com.au/ftp/smb-NT-verify.1.1.tar.gz>`
- Unix 상에서 유연한 인증을 할 수 있는 PAM (pluggable authentication module) : PAM library *www.kernel.org/pub/linux/libs/pam/index.html* `<http://www.kernel.org/pub/linux/libs/pam/index.html>`
- 마지막으로 Linux에서 LDAP *www.umich.edu/~dirsvcs/ldap/index.html* `<http://www.umich.edu/~dirsvcs/ldap/index.html>`

제 7 절 어플리케이션의 Remote 실행

Unix에서 제일 놀랄만한 기능(신규 유저는 거의 모르는 것)은, 어플리케이션의 리모트실행과 분산 실행의 지원이 잘 지원되는 것이다.

7.1 Telnet

Telnet는, 리모트 컴퓨터를 유저가 실제로 그 사이트에 있는것과 같이 사용할 수 있는 프로그램이다. Telnet은 정확한 리모트관리를 할 수 있으므로, 가장 강력한 Unix 용 툴중 하나이다. Telnet은 사용자 측면에서도 편리한 프로그램이다. 인터넷상의 어디에서도 자신의 모든 파일과 로그램에 억세스할 수 있다. X 서버와 결합하면(지연 시간은 별도로) 콘솔 앞에 있는 것과 지구의 반대측으로부터 사용하는 것은 차이가 없다. Telnet의 demon과 클라이언트는 대부분의 배포본에 들어 있다.

암호화된 리모트 쉘 세션은 SSH (*www.ssh.fi/sshprotocols2/index.html* `<http://www.ssh.fi/sshprotocols2/index.html>`)를 사용 할 수 있기 때문에, 안전하게 리모트관리를 효율적으로 할 수 있다.

o Telnet 관련 소프트웨어 *metalab.unc.edu/pub/Linux/system/network/telnet/* `<http://metalab.unc.edu/pub/Linux/system/network/telnet/>`

7.2 Remote commands

Unix, 특히 Linux에서는, 쉘 prompt에서 다른 컴퓨터와 대화 할 수 있는 리모트 커멘드가 있다. 예를 들어, rlogin는 Telnet과 같이 리모 트 머신에 로그인할 수 있고, rcp는 머신간으로 리모트로 파일 전 송할 수 있는 등이다. 마지막에, 리모트쉘 커멘드의 rsh는, 리모트 머신에 실제로 로그인하지 않고 그 머신상에서 커멘드를 실행할 수 있다.

7.3 X 윈도우 시스템

X Window System은 1980 년대 후반에 MIT에서 개발되어, 곧바로 Unix 그래 픽 워크스테이션용의 업계 표준 윈도우 시스템이 되었다. 이 소프트웨어는 free로 입수가능하고, 매우 융통성 있고, 광범위한 하드웨어 플랫폼으로 사용할 수 있다. X Window System은 모든 X 서버 와(하나 이상의) X 클라이언트로 구성되어 있고, 이 두개는 다른 것이다. 서버와 클라이언트의 차이를 인식하는 것은 중요하다. 서버는 디스프 레이를 직접 콘트롤하는 것으로, 키보드, 마우스, 디스플레이등 입출력의 모든 책임을 진다. 그 한편 클라이언트는 직접 화면에는 억세스 하지 않는다. 클라이언트는 서버와 통신하고, 서버가 모든 입출력을 취급한다. 어플리케이션을 실행하는 등 무엇인가"실제의" 컴퓨터로서의 일은 클라이언트가 한다. 클라이언트는, 서버에 윈도우를 하나 이상 오픈시키고, 그 클라이언트를 위한 입출력을 취급하는 일을 서버에게 할당한다.

간단하게 말하면, 유저가 리모트 머신으에 로그인하고, 예를 들면 Web 브라우 저 오픈등의 과정을 실행하고, 그 결과를 유저 자신의 머신상에 나타내게 한다, 이것은 X Window System 에서 가능하다. 이 과정은 실제로는 서버에서 실행되고 있으므로, 클라이언트에서는 매우 작은 CPU 파워만을 필요로 한다. 실제로 순수하게 X 서버가 되는 것을 기본 목적으로 한 컴퓨터가 있는데, 이것을 X 단말이라고 부른다.

Linux 용으로 X Window System 를 이식한 free본은 여기서 발견된다 : Xfree *www.xfree86.org/* `<http://www.xfree86.org/>`. 이것은 대부분의 Linux 배포본에 들어있다.

관련 HOWTO

o Remote X Apps HOWTO *metalab.unc.edu/mdw/HOWTO/mini/Remote-X-Apps.html* `<http://metalab.unc.edu/mdw/HOWTO/mini/Remote-X-Apps.html>`

7.4 VNC

VNC는 가상 네트워크 컴퓨팅(Virtual Network Computing)이다. 간단하게 말하면, 실행되고 있는 머신만이 아닌 인터넷상의 어디 에서든, 광범위한 머신 아키텍쳐로부터도, 컴퓨팅 "데스크탑"환경을 보이도록 하는 것이 VNC이다. VNC 의 클라이언트와 서버는 Linux 용도 다른 플랫폼용도 다수 있다. Windows NT 나 95 상에서 MS-Word를 실행해, 그 출력을 Linux 머신 으로 표시할 수 있다. 그 반대로도 할 수 있다. Linux 머신으로 어플리케이션을 실행하고, 그 출력을 다른 Linux 와 Windows 머신으로 표시할 수 있다. Java의 클라이언트도 있으므로, Web 브라우저 내부에서 리모트 디스플레를 실행할 수 있다. 마지막으로, SVGAlib의 그래픽스 라이브러리를 사용해 Linux로 이식한 것을 사용하면, 386s에서 불과 4Mb의 RAM 머신을 완전하게 동작하는 X 단말가 가능하다.

o VNC 의 Web 사이트 *www.orl.co.uk/vnc/* `<http://www.orl.co.uk/vnc/>`

제 8 절 네트워크 상호 접속

Linux에는 풍부한 네트워킹 기능이 있다. Linux 박스를 router, bridge, 등으로 활동 할 수 있다. 그 몇개를 아래에 기술한다.

8.1 루터 (Router)

Linux는 커넬에 routing 기능 지원을 수 있다. 상용 router 의 몇분의 일의 비용으로 Linux 박스를 IP 나 IPX router로 충분히 활동 할 수 있다. 최근의 커넬에서서는 머신을 주요 router로서 사용하기 위한 특별한 옵션이 있다.

- Multicasting: Linux 머신을, 송신지 어드레스를 몇개 가지고있는 IP 패킷에 대하여 router로 활동하도록 허락한다. 이것은 MBONE로서 필요하게 되는 것이다. MBONE는 인터넷의 정점에 있는 넓은 대역폭의 통신망 에서, 오디오와 비디오의 방송을 전하는 것이다.
- IP policy routing: router는 보통, 수신한 패킷의 최종적인 송신지 어드레스만을 보고 무엇을 할지를 정한다. 그러나, 송신된 어드레스와 패킷이 들어 오는 네트워크 디바이스 인지를 고려한 routing도 할 수 있다.

  floppy 1매로 지장없이 가동하는 Linux 루터를 만드는 것을 목표의 하나 로한 관련 프로젝트가 몇개 있다.

- Linux router project *www.linuxrouter.org* `<http://www.linuxrouter.org>`

8.2 Bridge

Linux의 커넬에 Ethernet Bridge 기능을 내장 할 수 있다. Ethernet Bridge란, Ethernet 각각의 세그먼트를 모아, 하나의 Ethernet 처럼 보이게 하는 것이다. 몇개의 Bridge를 함께 사용해, IEEE802.1의 spanning tree 알고리즘을 이용한 Ethernet으로 결구된 큰 네트워크를 만들 수 있다. 이것은 표준이기 때문에, Linux의 Bridge는 다른 third party Bridge 제품과 상호 접속할 수 있다. 패키지를 추가하면 , IP 나 IPX 와 MAC 어드레스에 의한 필터링도 할 수 있다.

관련 HOWTO

- Bridge+Firewall *metalab.unc.edu/mdw/HOWTO/mini/Bridge+Firewall.html* `<http://metalab.unc.edu/mdw/HOWTO/mini/Bridge+Firewall.html>`
- Bridge *metalab.unc.edu/mdw/HOWTO/mini/Bridge.html* `<http://metalab.unc.edu/mdw/HOWTO/mini/Bridge.html>`

8.3 IP Masquerade

IP Masquerade은 Linux에서 개발중인 네트워크 기능이다. IP Masquerade를 유효하게 한 Linux의 호스트를 인터넷에 접속하고 있는다. 그 때, 그 호스트에게 접속하고 있는(같은 LAN에서도, 모뎀 경유의 접속에서 가능하다.) 컴퓨터도, 공식적으로 할당되어진 IP 어드레스를 가지고 있지 않아도, 인터넷에 연결되어진다. 모뎀 접속 하나만을 사용해 많은 사람이 인터넷에 억세스할 수 있기 때문에, 비용 절감이 되기도 하고, 보안 증대에도 도움이 된다.(공식적으로 할당되어지지 않은 어드레스는 네트워크의 밖으로부터 억세스 할 수 없으므로, 이 호스트는 firewall의 역할을 할 수 있다).

IP Masquerade 관련 문서

- *ipmasq.home.ml.org/* `<http://ipmasq.home.ml.org/>`

- *www.indyramp.com/masq/links.pfhtml* `<http://www.indyramp.com/masq/links.pfhtml>`
- *metalab.unc.edu/mdw/HOWTO/mini/IP-Masquerade.html* `<http://metalab.unc.edu/mdw/HOWTO/mini/IP-Masquerade.html>`

8.4 IP Accounting

Linux 커넬의 이 옵션은, IP 네트워크 트래픽의 기록을 취해, 패킷의 로깅과 통계를 작성하는 것이다. 일련의 규칙을 정 의할 수 있어, 패턴에 일치하는 특정의 패킷 시에 카운터를 증가시키거나, 승인 / 거부등을 할 수 있다.

8.5 IP aliasing

Linux 커넬의 IP alias기능은, 하나의 저레벨 네트워크 디바이스 드라이버상에 복수의 네트워크 어드레스를 설정할 수 있도록 한다. (예를 들면, 하나의 Ethernet 카드에 두개의 IP 어드레스등). 일반적으로 사용하는것은, 청취하고 있는 어드레스에 의해 역할을 다르게 하는 것 같은 서비스(예로는 "멀티 호스트" , "가상 도메인", "가상 호스트 서비스")이다.

관련 HOWTO

o IP Aliasing HOWTO *metalab.unc.edu/mdw/HOWTO/mini/IP-Alias.html* `<http://metalab.unc.edu/mdw/HOWTO/mini/IP-Alias.html>`

8.6 Traffic Shaping

Traffic Shaper는 가상 네트워크 디바이스로, 또 하나의 네트워크 디바이스로부터 자료 흐름의 속도를 제한할 수 있는 것이다. ISP 등으로 각 클라이언트가 사용하는 대역폭의 량을 통제하는 policy를 실현하는데 특히 도움이 된다. (Web 서비스 전용이지만)클라이언트로부터의 IP 접속수나 사용 대역폭을 제한하는 Apache 모듈도 사용할 수 있을 것이다.

o *metalab.unc.edu/mdw/HOWTO/NET-3-HOWTO.html#ss6.15* `<http://metalab.unc.edu/mdw/HOWTO/NET-3-HOWTO.html##ss6.15>`

8.7 FireWall

FireWall이란, 공공 부분(인터넷 전체)로부터 사적인 네트워크를 보호하는 디바이스이다. FireWall은 각 패킷 내에 들어있는 송신원, 송신지, 포트, 패킷 타입의 정보를 사용해, 패킷의 흐름을 콘트롤하는 것이다.

Linux에는 다양한 FireWall 툴킷이 있고, 커넬에 내장 지원도 있다. 또 다른 FireWall은 TIS 와 SOCKS 이다. 이것들의 툴킷은 매우 정확도가 높고, 다른 툴과 조합해서 사용하면, 모든 종류의 트래픽과 프로토콜의 blocking / redirection 을 할 수 있다. configuration file이나 GUI 프로그램을 사용해 설 정하고, 복수의 다른 policy를 실현할 수 있다.

- TIS home page *www.tis.com* `<http://www.tis.com>`
- SOCKS *www.socks.nec.com/socksfaq.html* `<http://www.socks.nec.com/socksfaq.html>`
- Firewall HOWTO *metalab.unc.edu/mdw/HOWTO/Firewall-HOWTO.html* `<http://metalab.unc.edu/mdw/HOWTO/Firewall-HOWTO.html>`

8.8 Port forwarding

CGI-bin이나 Java 애플릿을 사용해 데이타베이스의 다른 서비스로 접근시켜 대화식으로 하는 Web 사이트가 증가하고 있다 . 이 접근은 보안 문제를 일으키기 쉽기 때문에, 데이타베이스를 가지고 있는 머신은 인터넷 에 직접 접속시키지 말아야한다.

port forwarding은 이 접근 문제에 대하여 거의 이상적인 해결책을 제 제공할 수 있다. firewall상에서 특정의 포트 번호로 들어 오는 패킷이 다시 기록될수있고, 실제의 서비스를 제공하는 내부 서버에 포워드 되어진다. 내부 서버로부터의 응답 패킷은 그 firewall로부터 나왔던 것처럼 보이도록 하기 위하여 다시 기록된다.

port forwarding의 정보는 여기 *www.ox.compsoc.net/˜steve/portforwarding.html* `<http://www.ox.compsoc.net/˜steve/portforwarding.html>`에서 발견 할 수 있다.

8.9 Load Balancing

데이타베이스 / Web 서버에 다수의 클라이언트가 동시에 억세스하는 경우 에는, 보통은 부하 밸런스를 취할 필요가 있다. 같은 서버를 복수로 두고 부하가 적은 서버에 요구를 redirect하는 것이 바람직 할 것이다. 이것은 Network Address Translation techniques (NAT)로 실현할 수 있다. 이것의 서브 세트가 IP masquerading 이다. 네트워크 관리자는, Web 서비스등의 어플리케이션을 하나의 서버에서 공급하는 대신에, 복수의 서버에 하나의 IP 어드레스를 공유시킨 복수 서버의 논리적인 집합으로 대체할 수 있다. 들어오는 접속은 특정한 하나의 서버를 향해 들어온다. 그 서버는 load balancing 알고리즘을 사용한다. 그 가상 서버는 입력과 출력의 패킷을 다시 기록하고, 서버는 하나인것같은 클라이언트 로 가장해, 투과적으로 억세스시킨다.

Linux IP-NAT의 정보는 여기 *www.csn.tu-chemnitz.de/HyperNews/get/linux-ip-nat.html* `<http://www.csn.tu-chemnitz.de/HyperNews/get/linux-ip-nat.html>`에서 볼수있다.

8.10 EQL

EQL는 Linux의 커넬에 통합 되어 있다. 어딘가의 컴퓨터에게 시리얼 접속이 2개있어(보통 이것에는 2개의 전화선과 모뎀 2개가 필요하다), 거기에 SLIP이나 PPP (전화선으로 인터넷 트래픽을 송신하는 프로토콜)을 사용하고 있을 것이다. 이 때 EQL 드라이버를 사용하면 2배의 속도인 하나의 접속 처럼 사용할 수 있다. 당연하지만, 이것은 상대측에서도 서포트 되어야만한다.

o *metalab.unc.edu/mdw/HOWTO/NET-3-HOWTO-6.html#ss6.2* `<http://metalab.unc.edu/mdw/HOWTO/NET-3-HOWTO-6.html##ss6.2>`

8.11 Proxy Server

Proxy라는 용어는「다른 사람의 대리로서 무엇인가를 하는 것」이다. 네트워크 용어로 말하면, Proxy Server 컴퓨터는 복수의 클라이언트를 대리로 할 수 있다. HTTP Proxy에서는 다른 머신(머신 A)로부터 Web 페이지 요구를 받는다. 이 Proxy는 요구된 페이지를 얻은 결과를 머신 A 에 돌려준다. 이 Proxy는 요구된 페이지의 cache를 가질 수 있기 때문에, 다른 머신이 같은 페이지를 부탁해 왔다면 cache 내의 복사본를 대신 돌려준다. 이것으로 대역폭 자원을 유효하게 이용할 수 있고, 빠르게 응답할 수 있다. 부가적으로, 클라이언트 머신이 외부 세계에 직접 접속하지 못하기 때문에, 내부 네트워크를 안전하게 하는 방법으로 사용할 수 있다. 잘 설정된 Proxy는 효율을 향상시키는 것과 동시에, 좋은 firewall이 될 수 있다.

Linux용의 Proxy Server가 몇개 있다. 인기있는 솔루션은 Apache 프락시 모듈이다. 좀 더 완전하고 강력한 HTTP 프락시의 실현은 SQUID이다.

- Apache *www.apache.org* `<http://www.apache.org>`

- Squid *squid.nlanr.net/* <http://squid.nlanr.net/>

8.12 Diald on demand

on demand의 dial 목적은, 사용자가 리모트 사이트에 상시 접속하고 있는 것처럼 보이게 하는 것이다. 보통은, 흥미있는(흥미를 일으키는, 규칙/우선 순위/ 사용 권한의 세트로 보통 정의된다.)패킷이 도달 하는 장소에 패킷 트래픽을 감시하는 demon이 있어, 그 demon이 리모트 끝에 접속 한다. 일정시간 그 채널이 비어 있으면 demon이 접속을 끊는다.

o Diald HOWTO *metalab.unc.edu/mdw/HOWTO/mini/Diald.html* <http://metalab.unc.edu/mdw/HOWTO/mini/Diald.html>

8.13 tunnelling, mobile IP, virtual private network

Linux의 커넬은 프로토콜 tunnelling(캡슐화)를 할 수 있다. IP밖에 통과시킬 수 없는 링크에서도 IPX 네트워크 같은 종류를 접속할 수 있는 것처럼, IP를 개입시킨 IPX의 tunnelling을 할 수 있다. 또 IP-IP tunnelling도 할 수 있고, 이것은 모빌 IP , 멀티 캐스트, 아마츄어 무선을 서포트하는데 중요하다. http://metalab.unc.edu/mdw/HOWTO/NET-3-HOWTO-6.html#ss6.8를 보시오.

mobile IP는, 인터넷에서 mobile node로 IP datagrams을 투 과적으로 루팅할 수 있는 확장 사양을 정의하고 있다. 각 mobile node는, 인터넷의 현재 접속점에는 무관계하게 언제나 자신의 홈 어드레스에의해 식별된다. mobile node는 자신의 홈에서 떨어져 있는 동안에 지금 어디서 인터넷으로 접속되고 있는지의 정보(care-of address)도 가져고 있다. 이 프로토콜은 그 정보를 홈 agent에게 등록 하고, 홈 agent는 datagram을 care-of address의 터널을 통해 mobile node를 향해 송신 한다. 그 터널의 끝에 도달하면 각 datagram는 mobile node로 배달된다.

Point-to-Point Tunneling Protocol (PPTP)는, 안전한 virtual private network(VPN)으로서 인터넷을 이용 가능하게 하는 네트워크 기술이다. PPTP는 Windows NT Server에 내장된 Remote Access Services(RAS) 서버에 통합 되어 있다. PPTP에 의해 로컬 ISP로 dialing도, 인터넷에로 직접 접속도, 유저는 자신의 디 스크에 있는 것과 같이 자신의 네트워크로 억세스할 수 있다. PPTP는 close 프로토콜로 최근 보안이 문제시 되고있다 . 이것 대신에 Linux를 베이스로한 대체품을 사용하도록 강하게 권유한다, 이곳은 주의깊게 검토 테스트된 open 표준이기 때문이다.

- Linux용 PPTP 클라이언트의 구현은 여기 *www.pdos.lcs.mit.edu/˜cananian/Projects/PPTP/* <http://www.pdos.lcs.mit.edu/˜cananian/Projects/PPTP/>에서 입수할 수 있다.
- Linux에서의 PPTP에 대한 정보는 여기 *bmrc.berkeley.edu/people/chaffee/linux_pptp.html* <http://bmrc.berkeley.edu/people/chaffee/linux_pptp.html>에서 입수 할 수 있다.

모빌 IP

- *www-uk.hpl.hp.com/people/jt/mip.html* <http://www-uk.hpl.hp.com/people/jt/mip.html>
- *metalab.unc.edu/mdw/HOWTO/NET-3-HOWTO-6.html#ss6.12* <http://metalab.unc.edu/mdw/HOWTO/NET-3-HOWTO-6.html##ss6.12>

Virtual Private Networks 관련 문서

- *metalab.unc.edu/mdw/HOWTO/mini/VPN.html* <http://metalab.unc.edu/mdw/HOWTO/mini/VPN.html>
- *sites.inka.de/sites/bigred/devel/cipe.html* <http://sites.inka.de/sites/bigred/devel/cipe.html>

제 9 절 네트워크 관리

9.1 네트워크 관리 어플리케이션

네트워크 관리와 리모트관리에 초점을 맞춘 툴들이 놀랄만큼 많이 있다. 흥미있는 리모트관리 프로젝트는 linuxconf 와 webmin 이다.

- Webmin *www.webmin.com/webmin/* `<http://www.webmin.com/webmin/>`
- Linuxconf *www.solucorp.qc.ca/linuxconf/* `<http://www.solucorp.qc.ca/linuxconf/>`

기타의 툴에는 네트워크 트래픽 분석툴, 네트워크 보안 툴, 감시툴, 설정툴등이 있다. 이것들의 다수는 Metalab *www.metalab.unc.edu/pub/Linux/system/network/* `<http://www.metalab.unc.edu/pub/Linux/system/network/>` 에서 찾을 수 있다.

9.2 SNMP

SNMP (Simple Network Management Protocol)는 인터넷의 네트워크 관리 서비스용 프로토콜이다. routers, bridges, network cards, switches, 등의 원격 감시와 설정을 할 수 있다. Linux 용으로 사용할 수 있는 librarie, 클라이언트, demon, SNMP에 의한 감시용 프로그램이 대 량 있다. SNMP 와 Linux 소프트웨어를 가지고 있는 사이트가 *linas.org/linux/NMS.html* `<http://linas.org/linux/NMS.html>`이다.

제 10 절 대기업의 Linux 네트워킹

네트워크를 거의 100% 시간동안 이용가능하다고 보증하는 메카니즘이 필요하게 되는 상황이 있다. 이것에 관련된 기술을 아래의 섹션에서 몇개 말하겠다. 이하의 많은 것은 *linas.org/linux/index.html* `<http://linas.org/linux/index.html>` 와 Linux High-Availability HOWTO *sunsite.unc.edu/pub/Linux/ALPHA/linux-ha/High-Availability-HOWTO.html* `<http://sunsite.unc.edu/pub/Linux/ALPHA/linux-ha/High-Availability-HOWTO.html>`의, Linas Web 사이트에서 찾았다.

10.1 고가용성(High Availability)

IT 시스템 전체가 하나의 장해로 움직이지 않게 되는 것(단일 장해점)을 피하는 것에는 중복이 이용된다. 네트워크 카드가 한 장, SCSI 디스크 가 하나밖에 없는 서버라면, 단일의 장해점이 두군데있는 것이 된다. 생각지 않은 고장이 일어나도 유저의 일을 재빠르게 계속시켜, 유저 가 깨닫게 하지 않도록 하는 것이 목적이다. 고가용성 소프트웨어는 스크립트 와 툴이 세트가 되어, 장해를 자동 감시 검출하고, 적절한 순서로 통상 운용으로 복귀시키고, 시스템 관리자에게 통지하는 것이다.

10.2 RAID

RAID ,(Redundant Array of Inexpensive Disks)는, 몇개의 디스크에 걸쳐 정보를 분산시키는 방법이다. 사용하는 기술은 disk striping(RAID 레벨 0)이나, disk mirroring (RAID 레벨 1)등이다. 이것에 의해 읽기 쓰기 양방의 대기 시간이 적게 되고, 또한 /또는 대역폭이 넓게 되고, 하드 디스크의 크래쉬로부터 복원할 수 있을 가능성을 높일 수 있다. RAID 설정의 종류는 6종류 이상이 정의 되어 있다. Linux 사용자가 사용할 수 있는 RAID는 소프트웨어 RAID , 외부부착 DASD 박스, RAID 디스크 콘트롤러의 3 개중에서 선택할 수 있다.

- 소프트웨어 RAID: 커넬의 내에 있는 디스크 제어(block device) 의 프로그램에 의해서, 순수하게 소프트웨어 만으로 각종 RAID 레벨 을 실현하는 것다.
- 외부부착 DASD: DASD(Direct Access Storage Device)는 스스로 전원을 가지는 독립된 상자로서, 하드 드라이브를 내장하는 캐비넷과 샤시를 가지고, Linux로부터는 또 하나의 SCSI 디바이스 처럼 보이게 하는것이다. 많은 점에서 DASD는 RAID 시스템의 가장 강력한 방볍을 제공 한다
- RAID 디스크 콘트롤러: 디스크 콘트롤러는 ISA/EISA/PCI 버스 에 삽입되는 아답터 카드이다. 보통 디스크 콘트롤러 카드와 같이 카드로부터 디스크 드라이버에 cable를 붙이지만, 보통 디스크 콘트롤러와는 달른 RAID 콘트롤러는 그 카드 자신이 RAID를 내장하고 있어, 각종 RAID 레벨 구현에 필요한 동작의 모든 것을 카드가 한다.

관련 HOWTO

- *metalab.unc.edu/mdw/HOWTO/mini/DPT-Hardware-RAID.html* `<http://metalab.unc.edu/mdw/HOWTO/mini/DPT-Hardware-RAID.html>`
- *metalab.unc.edu/mdw/HOWTO/Root-RAID-HOWTO.html* `<http://metalab.unc.edu/mdw/HOWTO/Root-RAID-HOWTO.html>`
- *metalab.unc.edu/mdw/HOWTO/mini/Software-RAID.html* `<http://metalab.unc.edu/mdw/HOWTO/mini/Software-RAID.html>`

linas.org 에서의 RAID

o *linas.org/linux/raid.html* `<http://linas.org/linux/raid.html>`

10.3 Redundant networking

IP Address Takeover (IPAT)). 이것은, 네트워크 어댑터 카드가 만가졌을 때에, 동일 node나 다른 node에서 움직이고 있는 네트워크 카드에, 그 IP 어드레스를 계승하게 하는 것이다.

MAC Address Takeover. IP 어드레스 계승이 발생된 때는, 네트워크내의 모든 node에서 자신의 ARP 캣쉬(IP 어드레스와 MAC 어드레스와의 대응)을 업데이트하는 것은 확인 되어야 한다.

자세한 것은 High-Availability HOWTO 를 보시오 : *metalab.unc.edu/pub/Linux/ALPHA/linux-ha/High-Availability-HOWTO.html* `<http://metalab.unc.edu/pub/Linux/ALPHA/linux-ha/High-Availability-HOWTO.html>`

제 11 절 정보원

Linux에서 네트워크상의 문제가 있어도 그 질문을 저의 전자 메일로 하지 말아 주시오. 저에게는 그것에 답할 시간이 없기 때문이다. 그것보다도 comp.os.linux.networking 뉴스 그룹(여기는, *www.dejanews.com* `<http://www.dejanews.com>`를 통해 억세스 가능하다.)으로 질문을 post하는 편이, 동움을 받을 수 있을 것이다. 여기로 post하기 전에 관련 문서를 읽어 주시오. 그 다음에, 이 뉴 스 archive를 검색 한다. 누군가가 같은 질문을 하고 (그리고 누군가가 회답하고)있는 것도 있기 때문이다. post할 때에는, 당신이 행한 일들과 에러메세지를 단계별로 설명하기 바란다.

정보를 얻을수 있는 사이트

- Linux: *www.linux.org* `<http://www.linux.org>`
- Linux Documentation Project: *metalab.unc.edu/mdw/linux.html* `<http://metalab.unc.edu/mdw/linux.html>` (Linux Network Administrator Guide도 조사한다)

- Freshmeat: 최신 Linux 소프트웨어 배포본 *www.freshmeat.net* <http://www.freshmeat.net>
- Linux 링크: *www.linuxlinks.com/Networking/* <http://www.linuxlinks.com/Networking/>

제 12 절 문서의 개정 이력

- 0.31 (1999년 9월 17일) Linux router project의 어드레스 변경 (John Ellis에게 감사)과 PPTP 링크를 또 하나 추가 (Benjamin Smith에게 감사)
- 0.30 (1999년 4월 6일) CODA의 섹션추가 (Brian Ristuccia *brianr@osiris.978.org* <mailto:brianr@osiris.978.org>에게 감사)
- 0.2-0.29 Bugfixes:-) (본문서말미 사례 참조)
- 0.1 (1998년 6월 5일)

제 13 절 사례와 면책

본문서는 다수의 사람들의 업적에 기초를 두고 있다. Linux가 현재 최선의 네트워크 operating system의 하나로 될 때까지 이 사람들이 성장시킨 것이다. 모든 것은 이 사람들의 덕분이다. 저자는 많이 노력하여, 본문서를 간단하고 정확하게, 완전하지만 너무 길지 않도록 했다. 그럼에도 불구하고, 이 같은 상황에서도 저자는 책임은 지지 않는다. 여기에 쓰고 있는 정보는 여러분 자신의 리스크로 이용해 주세요. 본문서의 시사, 정정, 전반적인 코멘트는, 저 자에게 자유롭게 전자메일을 보내 주세요. 그것으로 본 문서를 개선하고 싶습니다. 본 문서 차후 개정판에서는 혹시, radius , wget 등의 Web/ftp를 미러링하는 툴, 트래픽 analyzer, CORBA 등 을 실을지도 모릅니다. 기타의 화제도 저자에게 가르쳐 주면, 그것이 적당 하다고 생각되면 실을지도 모릅니다. 저자의 연락처는 daniel@rawbyte.com 이다.

마지막으로, Finnbjorn av Teilgum *efat@kampsax.dtu.dk* <maitlto:efat@kampsax.dtu.dk>, Cesar Kant *ckant@fazenda.gov.br* <mailto:ckant@fazenda.gov.br>, Mathieu Arnold *arn@mygale.org* <mailto:arn@mygale.org> 와 Hisakuni Nogami *hisakuni@interlink.or.jp* <mailto:hisakuni@interlink.or.jp>, Phil Garcia *pgarcia@mail.execpc.com* <mailto:pgarcia@mail.execpc.com>가 이 HOWTO를 잘 조사해 코멘트 해준 것에 감사하고 싶다. 그들의 도움은 매우 고마운 것이다.

본 문서는 여기 *www.rawbyte.com/lno* <http://www.rawbyte.com/lno> 에서 찾을 수 있다.

Daniel Lopez Ridruejo 1999년 9월 17일