



Mobile 環境での **Network**

大浦 真@ LILO

ohura@netfort.gr.jp

<http://www.netfort.gr.jp/~ohura/>

目次

- ⑥ はじめに
- ⑥ 私の Network 環境
- ⑥ 基本的な考え
- ⑥ pdnsd
- ⑥ wwwoffle
- ⑥ whereami
- ⑥ その他
- ⑥ 終わりに

はじめに

- ⑥ 私の本職は大学院生。
- ⑥ 一台の note パソコンを自宅と大学の両方で使っている。
- ⑥ 自宅と大学で Proxy などの Network 環境が全く違うので切り替えが面倒。
- ⑥ 自動的に切り替えることはできないだろうか。
- ⑥ 今回は、私が使っている一例を紹介します。Debian に依存している部分が多いので、全ての人が同じように使えるかどうかはあやしいですが、かなり UNIX らしい使い方だと思います。

私の *Network* 環境 (1)

- ⑥ 一台の note パソコンを自宅と大学の両方で使っている。
- ⑥ note パソコンには、PC カードスロット一つと、内蔵 LAN ポートがある。
- ⑥ 自宅
 - △ 普通のプロバイダに加入。
 - △ PC カードのモデムでダイヤルアップ接続。
 - △ SMTP サーバ、POP サーバ、DNS サーバはプロバイダのものを使用。
 - △ Proxy はなし。
 - △ 内蔵 LAN ポートに HUB をつないで、他の Linux マシンもつないでいる。

私の *Network* 環境 (2)

⑥ 大学 (研究室)

- △ 無線 LAN で接続。
- △ ファイアウォールの中。
- △ DHCP でプライベート IP アドレスを取得。(学内の全てのクライアントマシンにクラス A の IP アドレスが割当てられていて、一つの大きな Network を構成している。)
- △ SMTP サーバは研究室のサーバを使用。
- △ DNS サーバは大学のものを使用。
- △ POP サーバはプロバイダのものを使用。大学の POP Proxy サーバを経由してメールを取得。
- △ HTTP(S) と FTP は Proxy 経由。

基本的な考え (1)

- ⑥ 通常は Network 毎に手動で切り替える。(Web browser での例。)

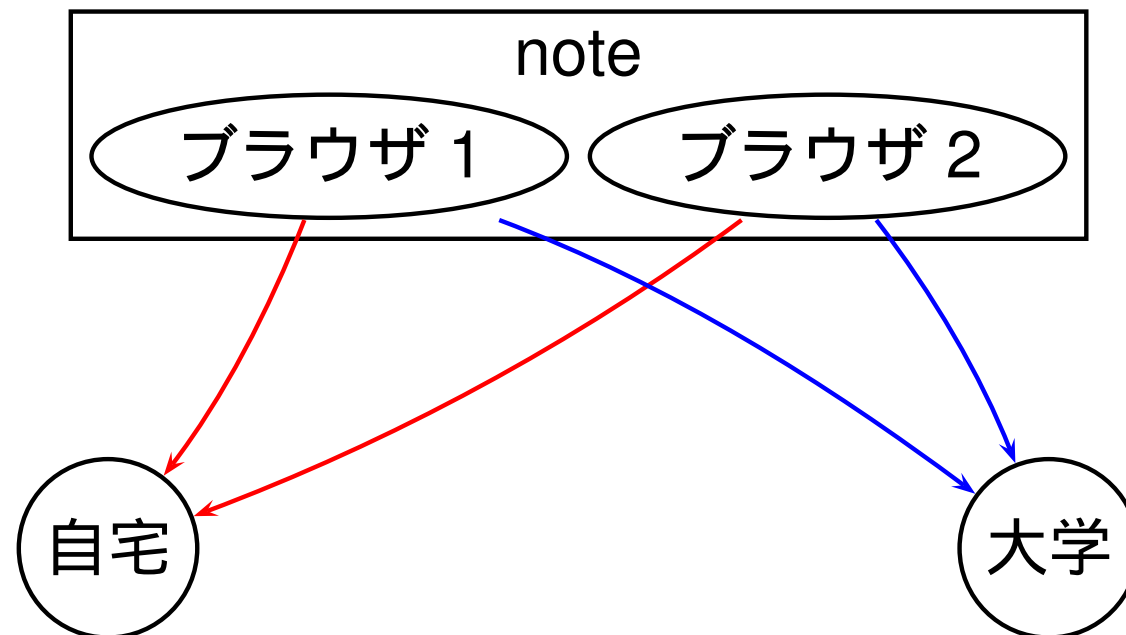


図 1: 手動で切り替え

基本的な考え (2)

- ⑥ 手元に必要なサーバを置いてしまおう。

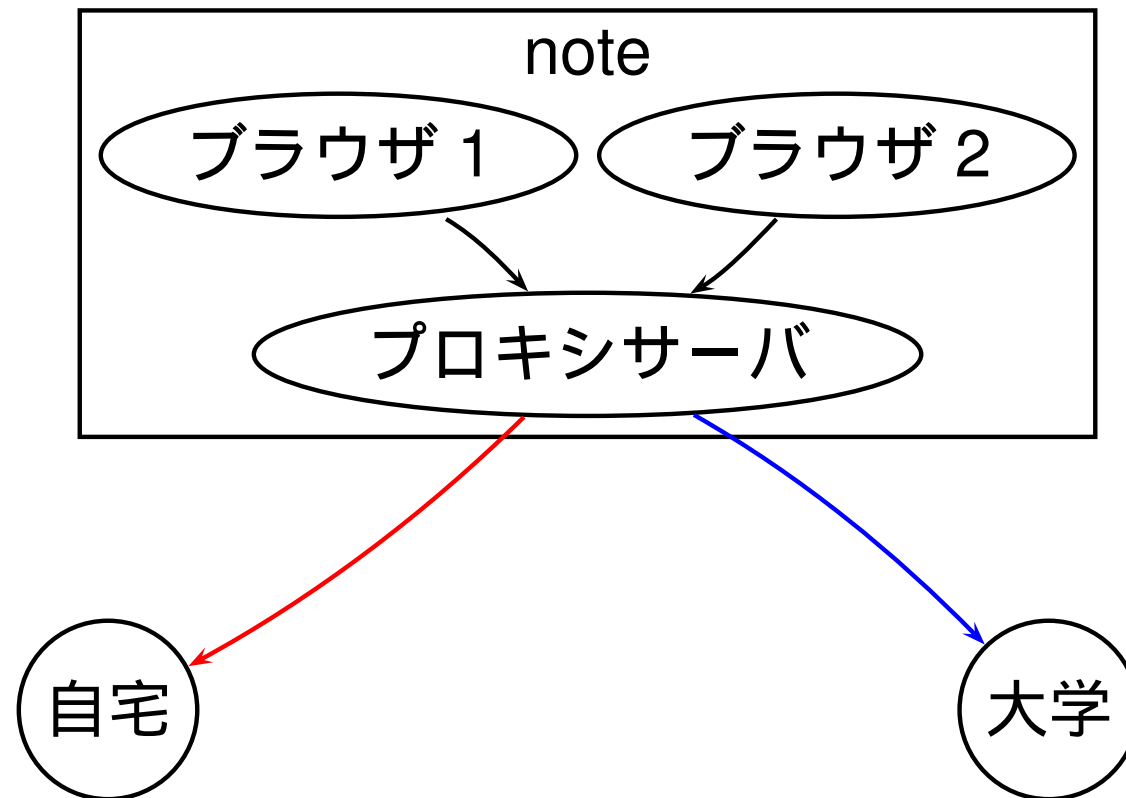


図 2: 自動切り替え

基本的な考え (3)

- ⑥ 図 1 の場合、Network が変わる度に、アプリケーション毎に Proxy の設定を変更しなければならない。
 - △ Mozilla(Netscape) の場合は、Preferences の画面で変更。
 - △ lynx や wget は環境変数を変更。
- ⑥ 手作業の変更は必ずと言っていいほど忘れる。
- ⑥ 図 2 の場合、各アプリケーションの Proxy の設定は、すべて localhost にしておけばいい。
- ⑥ 起動時 (PC カードを差した時) に Proxy サーバの設定を自動的に切り替えるようにする。

基本的な考え (4)

- ⑥ サーバとして何を使うか。
- ⑥ DNS は `pdnsd` を使う。Proxy は `wwwoffle`。SMTP は `postfix`。POP は `fetchmail`。
- ⑥ サーバの切り替えには、`whereami` を使う。

pdnsd(1)

- ⑥ UNIX では /etc/resolv.conf で DNS のアドレスを設定する。
- ⑥ XEmacs21 は起動時にしか /etc/resolv.conf を見ないようだ。
- ⑥ 起動後に resolv.conf が書き変わっても XEmacs21 はそれを知らないまま。
- ⑥ Mew でメールを送信する時に、XEmacs21 が固まってしまう。

pdnsd(2)

- ⑥ pdnsd: Proxy DNS サーバ
 - △ <http://home.t-online.de/home/Moestl/>
- ⑥ 手頃な DNS サーバ。
- ⑥ キャッシュ内容をハードディスク上に保存しておいてくれる。DNS へのアクセスが速くなる。(少しだけだけど。)
- ⑥ 複数の親 DNS を設定しておくと、ping が届く DNS だけを使ってくれるので、プロバイダの DNS と大学の DNS を両方設定しておけばいい。特に切替を意識する必要はない。

pdnsd(3)

- ⑥ resolv.conf には 127.0.0.1 を設定し、dhcp クライアントが resolv.conf を書き換えないように設定し、PPP の設定でも DNS を 127.0.0.1 にする。
- ⑥ dhcp クライアントが pump の場合、'-d' オプションをつける。
- ⑥ PPP では /etc/ppp/resolv/ 以下で設定。

wwwoffle(1)

- ⑥ wwwoffle: World Wide Web Offline Explorer
 - △ <http://www.gedanken.demon.co.uk/wwwoffle/>
- ⑥ 手頃な Web Proxy サーバ。(以前は squid を使っていたが、個人で使うにはちょっと大きさ。)
- ⑥ ダイヤルアップ接続の時に便利な機能がいくつか。
- ⑥ 接続中は後で閲覧できるようにページをキャッシュできる。
- ⑥ 接続していない時は、次回接続時に自動的にファイルを取ってくるように設定できる。
- ⑥ キャッシュを zlib で圧縮できる。
- ⑥ キャッシュは自動的に掃除される。

wwwoffle(2)

- ⑥ キャッシュ上のファイルを Namazu で検索することもできる。
mknmz-wwwoffle
<http://www.naney.org/comp/distrib/mknmz-wwwoffle/index.html>
- ⑥ もちろん、親 proxy を設定できる。

fetchmail

- ⑥ fetchmail: 有名な POP クライアント (作者も有名)
<http://www.catb.org/~esr/fetchmail/>
- ⑥ 時間間隔を指定して、デーモンモードで定期的に POP サーバからメールを取得できる。
- ⑥ 取得したメールは、‘/var/mail/ユーザ名’ のような local のメール配送システムに転送されるので、メールソフトで localhost からメールを取得するように設定する。
- ⑥ 自宅用の fetchmail のオプションを書いたスクリプトと大学用の fetchmail のオプションを書いたスクリプトを用意しておく。

whereami(1)

- ⑥ whereami: Network Autoconfiguration Utility
 - △ <http://debiana.net/whereami/>
- ⑥ ping が通るかどうかや、どの module が読み込まれているかなどを見て、今自分がいる場所 (location) を判断する。
- ⑥ その判断をもとに、サーバの設定を変更してサーバを再起動。
- ⑥ どうやら Debian にしか package がないようだ。
- ⑥ この際、Debian に乗り換えるか。

whereami(2)

⑥ インストール方法

- △ Debian だったら、`apt-get install whereami` で OK.
- △ Debian 以外の場合、まず、Debian のアーカイブから最新のパッケージを download。(作者のページには最新版がない。)
`http://ring.asahi-net.or.jp/pub/linux/debian/debian/pool/main/w/whereami/`
- △ Makefile には頼らず自分でファイルをコピー。

whereami(3)

⑥ アーカイブの中身

bin/whereami.pl whereami 本体の perl スクリプト。
/usr/sbin/ にでも入れておこう。whereami.sh は古い version のスクリプトなので使わない。

scripts/ 内部で使われるスクリプト類。
/usr/share/whereami/ に入れる。

whereami.conf, detect.conf 設定ファイル。
/etc/whereami/ に入れる。

html/, man/ マニュアル、ドキュメント類。

- ⑥ whereami.pl: whereami の本体。
何もオプションを付けずに実行すると設定されたされたテストが実行される。起動時や PC カードを差した時に実行されるスクリプトに入れておく。(Debian だったら、/etc/pcmcia/network など。)
- ⑥ コマンドラインオプションで location を上書きしたりもできる。

whereami の設定: *detect.conf*(1)

- ⑥ *detect.conf*: 設定ファイルその 1。自分がどこにいるか検出する。
- ⑥ 例

```
detect.conf
testmodule orinoco_cs wlan
testmodule serial_cs ppp

if wlan
    set INTERFACE eth1
    testdhcp '*.*.*.*' kuins
fi
```

whereami の設定: *detect.conf*(2)

- ⑥ 1行目、2行目。testmodule *module location* で *module* がロードされていたら、*location* にいると判断する。
 - △ *oricono_cs* は無線 LAN カードのモジュールなので、ロードされていたら、*wlan* にいると判断する。
 - △ *serial_cs* はモデムカードのモジュールなので、ロードされていたら、*ppp* にいると判断する。
- ⑥ 4行目から7行目。if *location...fi*
location にいる場合、対応する if と fi の間が実行される。

whereami の設定: *detect.conf*(3)

- ⑥ 5 行目。環境変数を設定。
- ⑥ 6 行目。testdhcp [*interface*,] *pattern location*
dhcp で *pattern* の IP アドレスを取得していたら、
location にいると判断。 *interface* は省略可。
- ⑥ 他にも testmii、testpppoe、testarp、testpci など
がある。
- ⑥ これらの test... の実体
は、 /usr/share/whereami/tests/ にあるスクリプト。成
功なら 0、失敗なら 1 を返す。

whereami の設定: *whereami.conf*(1)

- ⑥ *whereami.conf*: 設定ファイルその2。自分のいる場所にあわせて指定したコマンドを実行。
- ⑥ 例。

```
whereami.conf  
+kuins /usr/local/sbin/wwwoffle.kuins  
+kuins setmailrelay ling.bun.kyoto-u.ac.jp  
  
+ppp setmailrelay mb4.seikyou.ne.jp
```

whereami の設定: *whereami.conf*(2)

- ⑥ *+location action*
以前は別の場所において、新たに *location* に移動したら、*action* を実行。
- ⑥ *-location action*
location から、別の場所に移動したら、*action* を実行。
- ⑥ *=location action*
location にいたら、*action* を実行。

whereami の設定: *whereami.conf*(3)

- ⑥ *action* には、サーバを停止して、設定を変更して再起動するためのスクリプトでも書いておこう。
- ⑥ *whereami* に付属の *setmailrelay* というスクリプトは *sendmail* や *postfix*、*exim* などの MTA の設定を書き換え再起動してくれる。

その他

- ⑥ 他のネットワークサービス (ネットニュースや ntp など。) も同じように設定できる。
- ⑥ 私は2つの Network 環境を往復しているだけだが、もっと多くの環境の切り替えも簡単にできる。
- ⑥ whereami 付属のサンプル設定ファイルでは、全部で7つの環境を切り替えている。

終わりに

- ⑥ ifplugd という LAN ケーブルの抜き差しを監視できるプログラムもある。
<http://www.stud.uni-hamburg.de/users/lennart/projects/ifplugd/>
- ⑥ Debian だったら、インストールするだけで whereami と一緒に使える。
- ⑥ さすが Debian。
- ⑥ 今回紹介した方法は、localhost にサーバを立てなければならない方法だが、UNIX 系の OS でないとできない方法なのでぜひ試してみてください。