Szybko i tanio

Od końca listopada 1999 roku Telekomunikacja Polska SA udostępnia mieszkańcom niektórych miast usługę, która nosi nazwę „Szybki Dostęp do Internetu" (SDI). SDI umożliwia wykorzystanie już istniejących linii telefonicznych do utworzenia stałego połączenia z Internetem, nie blokując jednocześnie normalnych połączeń telefonicznych na tych liniach.

TP SA wykorzystała rozwiązanie firmy Ericsson o nazwie „Home Internet Solution". System HiS składa się z terminala – niewielkiego urządzenia przypominającego modem zewnętrzny, instalowanego po stronie klienta, oraz serwera dostępowego po stronie centrali telefonicznej, podłączonego na stałe do Internetu.

Koszt instalacji łącza to w tej chwili wydatek rzędu 1000 PLN, miesięczny abonament wynosi mniej niż 200 PLN, niezależnie od stopnia wykorzystania łącza. W zamian za to HiS gwarantuje nam transfer danych w wysokości do 115 200 bps (maks. 57 600 bps przy jednocześnie prowadzonej rozmowie telefonicznej na tej samej linii).

Osoba, która wykupi usługę SDI od Telekomunikacji, otrzymuje nazwę użytkownika i hasło, niezbędne do połączenia, oraz stały numer IP. Daje to możliwość zainstalowania w domu niewielkiego serwera internetowego, a także podłączenia sieci lokalnej (np. w bloku) do Internetu.

Instalacja terminala

Samo podłączenie terminala SDI jest bardzo proste. Na tylnej ściance można znaleźć gniazda: zasilające, szeregowe (do podłączenia do komputera) oraz dwa telefoniczne – jedno do podłączenia linii, a drugie – aparatu telefonicznego (urządzenie musi być włączone między kablem przychodzącym z centrali a siecią telefoniczną wewnątrz mieszkania).

Kabel szeregowy z jednej strony podłączamy do terminala, z drugiej do jednego z gniazd szeregowych w komputerze. Często zdarza się, że „wąskie" gniazdo (typu DB 9) jest już zajęte, na przykład przez myszkę szeregową – wtedy należy dokupić przejściówkę, umożliwiającą podłączenie wtyczki 9-pinowej do 25-pinowego gniazda.

Oprócz komputera do terminala podłączamy jeszcze kable telefoniczne i zasilanie. Na przednim panelu urządzenia dostępowego powinny zapalić się kontrolki sygnalizujące gotowość urządzenia do pracy.

Kontrola terminala

Następną czynnością jest sprawdzenie działania połączenia na odcinku komputer-terminal. W zależności od tego, czy podłączyliśmy go do portu określanego jako COM1 czy też COM2, trzeba sprawdzić, czy możemy porozumieć się z HiS-em za pośrednictwem pliku urządzenia **/dev/ttyS0** lub **/dev/ttyS1**. Dla wygody można utworzyć dowiązanie symboliczne (link), na przykład **/dev/sdi**, wskazujące na odpowiednie urządzenie **tty**. Robimy to, używając polecenia **ln –s /dev/ttyS<X> /dev/sdi**.

Po utworzeniu dowiązania uruchamiamy dowolny emulator terminala, na przykład minicom. Ustawiamy urządzenie **/dev/sdi** i prędkość transmisji *115200 kbps*. Gdy program jest już skonfigurowany, wyłączenie i ponowne włączenie terminala powinno spowodować pojawienie się na ekranie paru przypadkowych znaków, a na przednim panelu urządzenia powinna zapalić się kontrolka DTR (*Data Transmission Ready*). Od tej pory terminal reaguje na polecenia. Wpisanie komendy **ATDT 1** przy prawidłowym połączeniu dają wtedy widoczny efekt w postaci statusu wykonania polecenia **OK** i znaku zachęty **Command>**. Terminal HiS ma wbudowaną obsługę niektórych kodów Hayesa typowych dla modemów, takich jak ATDT <numer>, ATH lub ATS0=<cyfra>.

Konfiguracja PPP

Jeśli wszystko działa, to znaczy, że mamy sprawny i podłączony terminal i można zająć się konfiguracją połączenia PPP. Do tego potrzebne nam będzie jądro z obsługą protokołu PPP – większość dystrybucyjnych kerneli otrzymujemy razem z modułem **ppp.o**, choć można też wkompilować do jądra obsługę PPP na stałe (w konfiguracji jądra jest to opcja **Network device support -> PPP**).

Oprócz kernela potrzebny jest program pppd – w dystrybucjach wyposażonych w systemy zarządzania pakietami programów znajduje się on zazwyczaj w pakiecie **ppp** lub **pppd**.

Po zainstalowaniu odpowiednich aplikacji możemy skonfigurować samo połączenie. Czynność ta przypomina konfigurację połączenia modemowego PPP – z tą różnicą że w przypadku HIS-a nie trzeba wybierać numeru telefonu. Ponadto nie odpowiadamy na pytanie o nazwę użytkownika i hasło, tak jak się to nieraz zdarza podczas konfiguracji połączeń telefonicznych – cała autoryzacja użytkownika odbywa się za pomocą protokołu PAP już po zestawieniu połączenia PPP.

Program pppd wymaga pliku z opcjami PPP dla połączeń SDI oraz zbioru **pap–secrets** z danymi potrzebnymi do autoryzacji użytkownika po stronie centrali. Pierwszy z wymienionych plików można utworzyć jako **/etc/ppp/peers/sdi** – jest to standardowe położenie dla plików z opcjami dla poszczególnych usługodawców, z którymi nasz komputer się łączy. W tym zbiorze powinny się znaleźć wpisy:

```
lock
noauth
modem
crtscts
user <nazwa_uzytkownika>
remotename tpsa-sdi
defaultroute
```

Oznaczają one odpowiednio:

- **lock** – urządzenie ma być blokowane dla innych programów (przez utworzenie pliku **/var/lock/LCK..<nazwa_urzadzenia>**);
- **noauth** – ponieważ połączenia PPP są symetryczne (tzn. nie ma w nich formalnego rozróżnienia na serwer oraz klienta), nasz komputer może wymagać przedstawienia nazwy użytkownika i hasła od centrali. Omawiany wpis wyłącza konieczność autoryzacji centrali w lokalnej maszynie;
- **modem** – pppd ma wykorzystywać linie kontrolne modemu;
- **crtscts** – pppd ma używać sprzętowej kontroli transmisji (za pomocą sygnałów RTS/CTS);
- **user <nazwwa>** – nasze pppd powinno się przedstawić zdalnej maszynie jako *<nazwa>* (domyślnie jest używana nazwa komputera, a to na ogół nie odpowiada nazwie użytkownika nadanej nam przez TP SA);
- **remotename tpsa-sdi** – pppd podczas autoryzacji powinno wybrać wpis z pliku *pap-secrets*, gdzie w polu serwer znajduje się właśnie wpis **tpsa-sdi**;
- **defaultroute** – po ustanowieniu połączenia PPP pppd powinno ustawić z jego pomocą ścieżkę domyślną. Wszystkie pakiety, dla których komputer nie zna drogi docelowej, mają być wysyłane właśnie przez to połączenie (na ogół dotyczy to wszystkich adresów poza siecią lokalną).

Dodatkowo w pliku **sdi** można dopisać jeszcze takie opcje, jak **mtu** i **mru**, czyli maksymalną ilość danych, która powinna być przesyłana w jednym pakiecie – im większa jest ta wartość, tym szybciej będą przesyłane pliki, przy założeniu, że połączenie jest wystarczająco szybkie. W przypadku kiepskich połączeń lepsze są mniejsze wartości – w razie zgubienia pakietu trzeba będzie przesłać ponownie mniejszą ilość danych. Możemy ustawić wartości 576 dla obu wymienionych opcji.

Czasem zachodzi też potrzeba przesyłania niektórych znaków w postaci zakodowanej (dlatego że są to na przykład znaki kontrolne terminala i ich przesyłanie w czystej formie zakłócałoby pracę urządzeń dostępowych). Możemy wtedy użyć opcji **escape** i **asyncmap**. Do celów testowych warto dodać opcję **debug** – dzięki niej pppd będzie udostępniało dużo więcej informacji na temat tego, co właśnie robi. Więcej informacji znajdziemy w podręczniku systemowego pppd w sekcji 8.

Drugi z wymaganych przez pppd plików to **/etc/ppp/pap–secrets**. Wpisujemy w nim nazwę użytkownika oraz hasło dostarczone przez providera, czyli w tym wypadku TP SA Wpis powinien mieć postać:

```
<użytkownik> tpsa-sdi <hasło>
  [numer_IP]
```

Numer IP na końcu wiersza jest opcjonalny. Podanie go spowoduje, że nasz program pppd odmówi sfinalizowania połączenia, jeśli otrzyma inny niż podany adres dla lokalnej maszyny.

Warto zwrócić uwagę na to, aby plik **pap–secrets** miał prawa dostępu ustawione tylko dla właściciela roota) – zwykły użytkownik nie będzie wtedy mógł odczytać wpisanego tam jawnym tekstem hasła.

Rozpoczynamy pracę

Jeśli całość jest już ustawiona, można wypróbować połączenie. Z konta roota należy wydać polecenie:

```
/usr/sbin/pppd /dev/sdi 115200
  call sdi
```

co oznacza: połącz się przez urządzenie */dev/sdi* z prędkością *115 200 kbps* i wykorzystaj opcje z pliku */etc/ppp/peers/sdi*. W tym momencie demon pppd powinien nawiązać połączenie i uruchomić interfejs PPP, na ogół o nazwie ppp0. Korzystając z innej konsoli, możemy uruchomić polecenie **watch /sbin/ifconfig**, jeśli w naszym systemie używamy poleceń **ifconfig/route**, lub **watch /sbin/ip** – gdy korzystamy z **iproute2**. Po wywołaniu pppd powinno w krótkim czasie pojawić się właśnie wspomniane *ppp0*. Jeśli tak się nie stanie, warto zajrzeć do logów – prawdopodobnie będzie tam podany powód problemów z utworzeniem połączenia.

Plik /etc/ppp/peers/sdi

```
lock
noauth
modem
crtscts
user <nazwa_uzytkownika>
remotename tpsa-sdi
defaultroute
```

Plik /etc/ppp/pap-secrets

```
<uzytkownik> tpsa-sdi <haslo>
```

Plik /etc/resolv.conf

```
# nazwa lokalnej domeny
domain moja-domena.com.pl

# adresy serwerow DNS
nameserver 194.204.159.1
nameserver 194.204.152.34
```

W momencie pojawienia się nowego interfejsu PPP można już korzystać z Sieci. Wystarczy tylko jeszcze za pomocą dyrektywy **nameserver** ustawić w pliku **/etc/resolv.conf** adresy serwerów DNS podane przez TP SA i sprawdzić działanie Sieci na przykład poleceniem **ping www.chip.pl**. Jeśli dostaniemy odpowiedź od zdalnego komputera, oznacza to, że jesteśmy podłączeni do Internetu. Komputer można już skonfigurować jako niewielki (ze względu na stosunkowo niską przepustowość łącza) serwer z danym adresem IP oraz nazwą przydzieloną przez TP SA, na ogół w postaci: *p<litera><numer>.<miasto>.sdi.tpnet.pl*.

Za niewielkie pieniądze lub czasem wręcz bezpłatnie zdobędziemy własną domenę. Dzięki temu będzie można się dostać do naszego komputera również przez podanie adresu URL.

Sieć lokalna

Tak skonfigurowany serwer może udostępniać wyjście na świat komputerom znajdującym się w sieci lokalnej. Jest kilka możliwych rozwiązań dostępu z sieci wewnętrznej do hostów w Internecie od najprostszych serwerów proxy, przez „przezroczyste" serwery proxy, aż po maskaradę (w tym wypadku są to jedyne możliwe rozwiązania – TP SA przydziela dla jednego połączenia tylko jeden numer IP i nie da się nim obdzielić kilku komputerów inaczej jak tylko przez wyróżnienie jednego z nich jako tzw. bramki dostępowej).

Najprostszym sposobem na udostępnienie sieci lokalnym komputerom jest zainstalowanie serwera proxy. Będzie się on zajmował zbieraniem i wykonywaniem pochodzących od maszyn w sieci lokalnej zleceń pobierania plików z Internetu, a następnie przekazywaniem ich dalej wewnątrz sieci lokalnej. Większość przeglądarek WWW pozwala na podanie adresu serwera proxy – jeśli wpiszemy adres naszego komputera z łączem SDI, a na nim uruchomimy program (np. *squid* albo *delegate*), który będzie zajmował się zbieraniem poleceń pobierania stron internetowych, komputer w sieci lokalnej będzie oferował przeglądanie stron, mimo że nie jest bezpośrednio podłączony do Sieci. Można w ten sposób pośredniczyć w przekazywaniu wielu różnych typów połączeń: HTTP, FTP, NNTP, Telnet itd.

Nieco innym sposobem jest postawienie tzw. przezroczystego serwera proxy (ang. *transparent proxy*). Działa on na podobnej zasadzie jak zwykłe proxy, ale nie wymaga od użytkownika konfigurowania w swoim oprogramowaniu niczego poza domyślną bramką dla pakietów. Bramka SDI, otrzymując pakiety, które powinny dotrzeć do zdalnego serwera, na przykład do portu usługi WWW, samodzielnie zajmuje się wysłaniem zlecenia do swojego proxy, a pobrane strony przekazuje z powrotem do komputera w sieci lokalnej. W ten sposób nasza maszyna lokalna „nie widzi" komputera z SDI, tylko bezpośrednio docelowy serwer.

Jeszcze inna metoda uzyskania dostępu do Internetu to użycie tzw. maskarady na serwerze wyposażonym w SDI. Maskarada polega na przyjmowaniu pakietów z sieci lokalnej, zamianie w nich adresów na własny i przesyłaniu ich w świat. W przypadku pakietów wracających z Internetu sytuacja wygląda dokładnie odwrotnie – adresy serwera SDI w pakiecie są zamieniane na odpowiednie adresy lokalne. Dla komputera w sieci lokalnej pracującego przez bramkę z maskaradą – podobnie jak przy transparentnym proxy serwerze – widoczne są jedynie hosty internetowe. Różnica polega na tym, że tu nie trzeba uruchamiać dodatkowych programów, gdyż wszystko dzieje się na poziomie obsługi pakietów w jądrze systemowym. Ta metoda jest zazwyczaj najlepsza, jeśli chcemy udostępniać coś więcej niż tylko kilka wybranych typów usług.

Ustawienia jądra

Włączenie jednej z usług: transparent proxy lub maskarady, wymaga odpowiedniego skonfigurowania i skompilowania jądra systemowego. Zwykłe serwery proxy nie potrzebują żadnych dodatkowych funkcji – są to zwyczajne programy i kernel nie musi w żaden specjalny sposób obsługiwać używanych przez nie pakietów. Zarówno jednak trans-proxy, jak i maskarada potrzebują w kernelu opcji, które w konfiguracji są oznaczone jako **Network firewalls** oraz **IP: firewalling**. Dodatkowo do ustawienia przezroczystych proxy potrzebna jest opcja IP: transparent proxy support, a dla maskarady – **IP: masquerading**. Dodatkowo czasem przydaje się ustawienie **IP: ICMP masquerading**, która dodaje możliwość przekazywania pakietów ICMP (generowanych na przykład przez pinga albo traceroute). Przy maskaradzie zostaną automatycznie skompilowane dodatkowe moduły, umożliwiające przekazywanie połączeń FTP, IRC, Real Audio itp.

Omawiane opcje można dodać do jądra przez wykonanie polecenia **make menuconfig** w katalogu z rozpakowanymi źródłami kernela i zaznaczenie potrzebnych pól w sekcji **Networking options**. Po skonfigurowaniu jądra kompilujemy je poleceniami **make clean**, **make dep**, **make zImage** i **make modules**. Osoby, które jeszcze nie wiedzą, jak kompiluje się jądro, odsyłamy do opisów zawartych w **Kernel–HOWTO** (patrz: odnośnik w ramce).

Maskarada na poważnie

Ponieważ maskarada jest spotykana znacznie częściej niż przezroczyste serwery proxy, opiszemy przede wszystkim konfigurację serwera maskaradującego. Gdy kernel ma już włączoną obsługę odpowiedniej usługi, można zacząć przygotowywać maszynę do przekazywania połączeń. Pierwszą rzeczą jest ustawienie forwardingu, czyli prostego przekazywania pakietów z jednego interfejsu na inny. Włącza się to przez wpisanie jedynki do pliku **/proc/sys/net/ipv4/ip_forward**:

```
echo "1" > /proc/sys/net/ipv4/
  ip_forward
```

Niektóre dystrybucje Linuksa (na przykład Red Hat) pozwalają na ustawianie tej właściwości w momencie uruchamiania systemu (za pomocą skryptów startowych i pliku **/etc/sysconfig/network**). Jeśli nasza dystrybucja nie oferuje takiej opcji, można wstawić powyższe polecenie na koniec dowolnego skryptu wykonywanego przy starcie (na przykład **/etc/rc.d/rc.local**).

Druga konieczna czynność to ustawienie reguł filtrowania pakietów: które pakiety, skąd, dokąd i jak mogą być przekazywane. W kernelach z serii 2.2.x służy do tego program **ipchains** (w 2.0.x był używany **ipfwadm**, nieco różniący się składnią). Najprostsze możliwe ustawienie maskarady przy jego użyciu to:

```
ipchains -A forward -j MASQ
  -s <adres_sieci_lokalnej>
  /<maska> -i ppp0
```

Po wywołaniu tego polecenia jądro rozpocznie maskaradę wszystkich pakietów pochodzących z adresów sieci lokalnej (oczywiście w wywołaniu trzeba podać własny adres sieci i netmaskę), wychodzących przez interfejs ppp0. Maskarada jest niepotrzebna w sieci wewnętrznej, a jedynie przy wysyłaniu pakietów poza jej obręb.

Dodatkowo, aby uniknąć przekazywania pakietów z sieci zewnętrznej do wewnętrznej (poza maskaradowanymi), można wywołać polecenie **ipchains -A forward –j DENY –s !<adres_sieci>/<maska>**, dzięki czemu wszystkie pakiety pochodzące spoza sieci lokalnej i nie przeznaczone dla serwera maskaradującego zostaną odrzucone.

Do wykorzystania specjalnych modułów maskarady, takich jak na przykład maskaradowanie połączeń FTP w trybie

Opis klas nieroutowalnych (prywatnych)

Aby ułatwić przydzielanie adresów w lokalnych sieciach komputerowych posługujących się protokołem IP, wśród wszystkich adresów sieciowych wyróżniono ich grupy, które są przeznaczone jedynie do użytku prywatnego – żadne urządzenie sieciowe nie powinno przekazywać pakietów z adresami jednej z tych klas lokalnych do sieci zewnętrznych. Te grupy adresów to:

10.0.0.0/8 (adresy od 10.0.0.0 do 10.255.255.255)
172.16.0.0/12 (adresy od 172.16.0.0 do 172.31.255.255)
192.168.0.0/16 (adresy od 192.168.0.0 do 192.168.255.255)

Typowym przypadkiem wykorzystania tych adresów jest sieć komputerowa podłączona do Internetu przez jeden komputer (tzn. sieć, dla której jest przeznaczony pojedynczy adres IP – tak jak w wypadku SDI) lub sieć odcięta od świata zewnętrznego. W takim wypadku wszystkie komputery w sieci wewnętrznej mogą nosić adresy lokalne.
Istnieje jeszcze jedna klasa adresów prywatnych: 127.0.0.0/8 (adresy od 127.0.0.0 do 127.255. 255.255), ale są one na ogół przypisywane wirtualnemu urządzeniu – na przykład pętli zwrotnej (loopback), która przekazuje otrzymane pakiety do samej siebie (ten sam pakiet jest wysyłany i odbierany – dzieje się to w obrębie jednego komputera). Dlatego wykorzystywanie tych numerów do adresowania końcówek w sieci jest zdecydowanie niewskazane.

aktywnym albo maskaradowanie IRC, potrzebne jest ich załadowanie do pamięci poleceniem **insmod**, na przykład **insmod ip_masq_ftp**, załaduje obsługę maskarady dla FTP. Te polecenia, tak samo jak ustawianie forwardingu, warto dopisać do jednego ze skryptów startowych. W jednej z ramek zamieszczony jest fragment programu dla systemów stosujących skrypty typu rc w stylu System V (Red Hat, SuSE) – można go skopiować na przykład do pliku **/etc/rc.d/init.d/sdi**, a potem porobić dowiązania symboliczne w katalogach **/etc/rc.d/rcX.d** z odpowiednimi nazwami.

Konfiguracja komputerów w sieci lokalnej jest bardzo prosta: jedyne konieczne ustawienia to bramka domyślna (default gateway) – tu powinien znaleźć się adres serwera z SDI, oraz serwery DNS – w przypadku maskarady można podać te same adresy, z których korzysta bramka, w przeciwnym zaś razie adres lokalnego serwera lub proxy DNS. Pod Linuksem domyślną bramkę dodaje się poleceniem:

```
route add default gw <adres_
  serwera_maskaradujacego>
  dev <interfejs>
```

Interfejs to np. **eth0**, jeśli do naszego serwera dostajemy się przez sieć podłączoną do pierwszej karty sieciowej. W systemach, w których używa się *iproute2*, podaje się polecenie:

```
ip route add default via <adres_
  bramki> dev <interfejs>
```

Wspomniane parametry da się zazwyczaj ustawić za pomocą skryptów startowych używanych standardowo w dowolnej dystrybucji, a w ostateczności można ręcznie dopisać odpowiednie regułki.

Przykładowy plik /etc/rc.d/init.d/sdi
(plik wymaga dostosowania do konkretnego systemu).

```
#!/bin/sh

NETWORK=10.0.0.0
NETMASK=255.0.0.0
SDI_IF=ppp0

case $1 in
start)
  echo -en "Uruchamianie SDI..."
  if
    /usr/sbin/pppd /dev/sdi 115200 call sdi
  then
    echo "zrobione"
  else
    echo "problemy!"
    exit 1
  fi

  echo -en "Ustawianie forwardingu..."
  echo "1" > /proc/sys/net/ipv4/ip_forward
  echo "zrobione"

  echo -en "Ustawianie maskarady..."
  if
    ipchains -A forward -j MASQ -s $NETWORK/$NETMASK
       -i $SDI_IF &&\
    ipchains -A forward -j DENY -s ! $NETWORK/$NETMASK
  then
    echo "zrobione"
  else
    echo "problemy!"
    exit 1
  fi

  echo -en "Ładowanie modułów do maskarady"
  if
    insmod ip_masq_ftp >/dev/null 2>&1
  then
    echo "zrobione"
  else
    echo "problemy!"
    exit 1
  fi

  touch /var/lock/subsys/sdi
  ;;
stop)
  echo -en "Usuwanie modułów maskarady"
  rmmod -f ip_masq_ftp >/dev/null 2>&1
  echo "zrobione"
  echo -en "Wyłączanie maskarady"
  ipchains -F forward
  echo "zrobione"
  echo -en "Wyłączanie forwardingu"
  echo "0" > /proc/sys/net/ipv4/ip_forward
  echo "zrobione"
  rm -f /var/lock/subsys/sdi
  ;;
status)
  if [ -f /var/lock/subsys/sdi ]; then
    echo "SDI jest włączone"
  else
    echo "SDI jest wyłączone"
  fi
  ;;
*)
  echo "Uzycie: $0 {start|stop|status}"
  ;;
esac
exit 0
```

Dodatkowe możliwości

Konfiguracja SDI oraz maskarady jest już w zasadzie zakończona. Wszystko, co zostaje do zrobienia, to uruchomienie na naszym lokalnym serwerze dostępowym dodatkowych usług, takich jak na przykład lokalny serwer DNS, serwer cache'ujący dla WWW (dzięki niemu można uniknąć zbytniego obciążenia łącza, jeśli oglądane strony często się powtarzają), serwer NNTP (z tych samych powodów co cache), serwer poczty itd. Można użyć programu in.timed do ustawienia serwera czasu – dzięki temu wszystkie komputery w sieci lokalnej będą w stanie automatycznie pobierać aktualną datę i godzinę. Dostępna jest również opcja uruchomienia lokalnego serwera IRCw celu umożliwienia łatwej komunikacji między poszczególnymi użytkownikami sieci lokalnej. Możliwości są właściwie nieograniczone i zależą jedynie od inwencji administratora sieci.

Sebastian Zagrodzki

INFO

http://www.jtz.org.pl/tlumaczenia.html

IP-Masquerade-mini-HOWTO, Kernel-HOWTO, Net-3-HOWTO, PPP-HOWTO

http://www.his-klub.prv.pl/
klub użytkowników HiS

http://www.tpsa.pl/sdi/
informacje o SDI z Telekomunikacji Polskiej