Tworzenie sieci

- zaplanowanie struktury fizycznej i logicznej sieci
- przydzielenie adresów IP
- uruchomienie fizyczne sieci:
  - interface-y sieciowe komputerów
  - urządzenia sieciowe: transceivery, repeatery, huby, switche, routery
  - połączenia
- skonfigurowanie interface-ów sieciowych komputerów
- skonfigurowanie odwzorowania adresów fizycznych do adresów IP (arp)
- skonfigurowanie ścieżek statycznych i oprogramowania routera/ów
- skonfigurowanie rezolwera adresów symbolicznych i DNS-ów
- skonfigurowanie usług specjalnych: NFS, BOOTP, itp.
- skonfigurowanie serwerów ogólnych usług sieciowych: inetd, ftpd, sendmail, httpd, ...
- skonfigurowanie indywidualnych aplikacji sieciowych (np. SQL)

Konfiguracja interface'u sieciowego

- nazwa interface'u
- adres IP
- maska sieciowa
- adres broadcastu

```
sequoia> ifconfig hme0 156.17.9.3 up netmask 255.255.255.128 \
                                     broadcast 156.17.9.127
```

```
sequoia> ifconfig -a
lo0: flags=849<UP,LOOPBACK,RUNNING,MULTICAST> mtu 8232
        inet 127.0.0.1 netmask ff000000
hme0: flags=863<UP,BROADCAST,NOTRAILERS,RUNNING,MULTICAST> mtu 1500
        inet 156.17.9.3 netmask ffffff80 broadcast 156.17.9.127
```

Konfiguracja interface'u (Sun)

```
diablo> ifconfig -a
lo0: flags=1000849<UP,LOOPBACK,RUNNING,MULTICAST,IPv4> mtu 8232 index 1
        inet 127.0.0.1 netmask ff000000
hme0: flags=1000843<UP,BROADCAST,RUNNING,MULTICAST,IPv4> mtu 1500 index 2
        inet 156.17.9.14 netmask ffffff80 broadcast 156.17.9.127
hme0:1: flags=1000843<UP,BROADCAST,RUNNING,MULTICAST,IPv4> mtu 1500 index 2
        inet 156.17.9.33 netmask ffffff80 broadcast 156.17.9.127
hme0:2: flags=1000843<UP,BROADCAST,RUNNING,MULTICAST,IPv4> mtu 1500 index 2
        inet 156.17.9.49 netmask ffffff80 broadcast 156.17.9.127
```


Konfiguracja interface'u (Linux)

```
amargosa> ifconfig -a
lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Bcast:127.255.255.255  Mask:255.0.0.0
          UP BROADCAST LOOPBACK RUNNING  MTU:3584  Metric:1
          RX packets:811140 errors:0 dropped:0 overruns:0 frame:0
          TX packets:811140 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0

eth0      Link encap:Ethernet  HWaddr 02:60:8C:7A:99:1C
          inet addr:156.17.30.22  Bcast:156.17.30.31  Mask:255.255.255.224
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:10309963 errors:0 dropped:0 overruns:0 frame:5736
          TX packets:7436831 errors:0 dropped:0 overruns:0 carrier:0
          collisions:13844
          Interrupt:5 Base address:0x300 Memory:c8000-ca000

eth1      Link encap:Ethernet  HWaddr 02:60:8C:7A:96:D5
          inet addr:156.17.9.6  Bcast:156.17.9.127  Mask:255.255.255.128
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:14298997 errors:39 dropped:0 overruns:0 frame:199369
          TX packets:8890668 errors:70 dropped:0 overruns:0 carrier:151
          collisions:444116
          Interrupt:9 Base address:0x310 Memory:cc000-ce000
```

Protokół arp

```
shasta-749> arp -a
Net to Media Table: IPv4
Device   IP Address               Mask      Flags   Phys Addr
------ -------------------- --------------- ----- ---------------
eri0   printer.palnet       255.255.255.255       00:01:e6:23:36:a1
eri0   router.palnet        255.255.255.255       00:03:0a:00:90:ee
eri0   wifi.palnet          255.255.255.255       00:12:a9:55:5d:b7
eri0   sierra.palnet        255.255.255.255       08:00:20:7d:f1:3a
eri0   shasta.palnet        255.255.255.255 SP    00:03:ba:08:47:7b
eri0   BASE-ADDRESS.MCAST.NET 240.0.0.0       SM    01:00:5e:00:00:00
```

W bezpiecznych sieciach można wykorzystywać protokół ARP do samodzielnego odkrywania powiązań adresów fizycznych MAC urządzeń sieciowych z ich (deklarowanymi) adresami IP. W pozostałych przypadkach można skonstruować sztywną tablicę translacji tych adresów, i stosować się do niej sztywno.

Routing (trasowanie?)

Czynność wyboru ścieżki sieciowej, do której należy wysłać dany pakiet sieciowy. Decyzja jest podejmowana na podstawie docelowego adresu IP pakietu, i jej wynikiem jest wybór komputera w (jednej z) sieci lokalnej(ych), do której(ych) dany komputer jest podłączony. Routing jest czynnością wykonywaną w ramach protokołu IP (warstwy sieciowej, w nomenklaturze ISO).

Routing realizowany jest w sposób niezwykle prosty: jądro Unixa posiada tablicę ścieżek sieciowych, określająca powiązania docelowych adresów IP komputerów i całych sieci, z bramami, czyli adresami IP komputerów w sieci lokalnej, czyli takich, do których przesłanie jest bezpośrednie.

Może istnieć wiele ścieżek w tej tablicy, i wybierana jest zawsze najlepiej dopasowana, to znaczy najbardziej szczegółowa ścieżka zgodna z danym adresem docelowym. W braku takiej ścieżki używana jest specjalna ścieżka domyślna, a gdy jej nie ma, pakietu nie da się wysłać do miejsca przeznaczenia, i routing kończy się niepowodzeniem.

Routing — tablica ścieżek sieciowych

zwykły komputer

```
sequoia> netstat -rn

Routing Table: IPv4
  Destination           Gateway           Flags  Ref   Use   Interface
-------------------- -------------------- ----- ----- ------ ---------
156.17.9.0           156.17.9.3            U        1  27617  le0
224.0.0.0            156.17.9.3            U        1      0  le0
default              156.17.9.16           UG       1  87628
127.0.0.1            127.0.0.1             UH       1     58  lo0
```

komputer służący jako router kilku sieci lokalnych

```
amargosa-201> netstat -rn
Kernel IP routing table
Destination     Gateway         Genmask         Flags   MSS Window  irtt Iface
156.17.9.160    156.17.9.13     255.255.255.224 UG        0 0          0 eth1
156.17.9.128    156.17.9.22     255.255.255.224 UG        0 0          0 eth1
156.17.30.0     0.0.0.0         255.255.255.224 U         0 0          0 eth0
156.17.9.0      0.0.0.0         255.255.255.128 U         0 0          0 eth1
127.0.0.0       0.0.0.0         255.0.0.0       U         0 0          0 lo
0.0.0.0         156.17.9.16     0.0.0.0         UG        0 0          0 eth1
```



```
tahoe-201> netstat -rn

Routing Table: IPv4
  Destination           Gateway           Flags  Ref   Use   Interface
-------------------- -------------------- ----- ----- ------ ---------
156.17.19.158        156.17.30.126         UGH      1      0
156.17.19.190        156.17.30.126         UGH      1      0
156.17.19.191        156.17.30.126         UGH      1      0
156.17.19.188        156.17.30.126         UGH      1      0
156.17.19.156        156.17.30.126         UGH      1      0
156.17.19.189        156.17.30.126         UGH      1      0
156.17.236.2         156.17.30.126         UGH      1      0
156.17.19.154        156.17.30.126         UGH      1      0
156.17.19.186        156.17.30.126         UGH      1      0
156.17.19.155        156.17.30.126         UGH      1      0
156.17.19.187        156.17.30.126         UGH      1      0
156.17.19.184        156.17.30.126         UGH      1      0
156.17.19.153        156.17.30.126         UGH      1      0
156.17.19.182        156.17.30.126         UGH      1      0
156.17.63.250        156.17.30.126         UGH      1      0
156.17.19.180        156.17.30.126         UGH      1      0
156.17.249.158       156.17.30.126         UGH      1      0
156.17.19.149        156.17.30.126         UGH      1      0
156.17.19.146        156.17.30.126         UGH      1      0
156.17.19.144        156.17.30.126         UGH      1      0
156.17.19.176        156.17.30.126         UGH      1      0
156.17.19.145        156.17.30.126         UGH      1      0
156.17.19.177        156.17.30.126         UGH      1      0
156.17.19.174        156.17.30.126         UGH      1      0
156.17.19.143        156.17.30.126         UGH      1      0
156.17.19.175        156.17.30.126         UGH      1      0
156.17.229.217       156.17.30.126         UGH      1      0
156.17.19.140        156.17.30.126         UGH      1      0
... dalszych 617 sciezek
```

Aktualizacja tablicy ścieżek

Utrzymywanie tablicy ścieżek z stanie aktualnym jest niełatwym zadaniem. Ścieżki sieciowe można tworzyć ręcznie, są to tzw. ścieżki statyczne, jak również automatycznie, w wyniku ciągłej wymiany informacji między programami odpowiedzialnymi za aktualizację informacji o połączeniach między komputerami.

Routing statyczny oznacza całkowicie ręczną aktualizację tablicy ścieżek. Ścieżki są zwykle wpisywane w czasie startu systemu z plików startowych, i ewentualnie zmieniane ręcznie przez administratora, gdy pojawia się taka potrzeba, na przykład przy przeadresowaniu komputerów i sieci lokalnych (wtedy zwykle robi się to przez restart systemu).

```
route add default 156.17.9.16 1

sequoia-295> cat /etc/defaultrouter
156.17.9.16

panamint-204> cat /etc/sysconfig/network
NETWORKING=yes
HOSTNAME=panamint
GATEWAY=156.17.9.16
NISDOMAIN=stud.ict.pwr.
```


Aktualizacja ścieżek na podstawie ICMP

Protokół komunikatów kontrolnych ICMP warstwy sieciowej (IP) posiada opcje odpowiadania na pakiety, co do których router wie, że albo nie mogą być poprawnie doręczone (ICMP Host Unreachable), albo że istnieje dla tych pakietów prostsza ścieżka do przeznaczenia (ICMP Host Redirect).

```
sequoia-379> ping 156.17.9.131
PING 156.17.9.131: 56 data bytes
ICMP Host redirect from gateway tahoe (156.17.9.16)
 to ithaca (156.17.9.22) for 156.17.9.131
ICMP Host Unreachable from gateway ithaca (156.17.9.155)
 for icmp from sequoia (156.17.9.3) to 156.17.9.131
ICMP Host Unreachable from gateway ithaca (156.17.9.155)
 for icmp from sequoia (156.17.9.3) to 156.17.9.131
ICMP Host Unreachable from gateway ithaca (156.17.9.155)
 for icmp from sequoia (156.17.9.3) to 156.17.9.131
^C
----156.17.9.131 PING Statistics----
3 packets transmitted, 0 packets received, 100% packet loss
```

Na podstawie takich informacji można aktualizować tablice ścieżek.

Routing dynamiczny

Ścieżki statyczne wpisane „ręcznie” do tablicy ścieżek pozostają tam na stałe. Odróżniają się one od ścieżek dynamicznych, wpisywanych przez tzw. demony routingu, stale aktualizujące tablicę ścieżek przez wpisywanie i usuwanie ścieżek. Te programy posiadają wiedzę aprioryczną o połączeniach w sieci lokalnej, i wymieniają się tą wiedzą z innymi takimi programami, stosując w tym celu jeden z kilku istniejących protokołów komunikacyjnych, tzw. protokołów routingu.

Najstarszym takim protokołem jest RIP (*Routing Information Protokol*), pozwalający na wyrażanie informacji o połączeniach i ich kosztach w formie liczby przeskoków (*hop-count*). Standardowy demon routingu realizujący RIP w systemie Unix to `routed`. Przykładowy plik konfiguracyjny:

```
sequoia-295> cat /etc/gateways
net 0.0.0.0 gateway 156.17.9.16 metric 1 passive
net 156.17.30.0 gateway 156.17.9.6 metric 1 active
```


Protokół RIP, podobnie jak demon `routed`, ma wiele ograniczeń, i niezwykle prostą konstrukcję opierającą się na rozgłaszaniu posiadanej informacji o ścieżkach. Ta prostota jest wielką zaletą w konstruowaniu małych sieci o niewielkich wymaganiach. Jednak w poważniejszych sytuacjach konieczne jest stosowanie innych programów i protokołów.

W systemie Linux dostępny jest bardzo dobry demon routingu `gated`. Obsługuje on wiele protokołów routingu, np. OSPF. Przykład konfiguracji:

```
amargosa> cat /etc/gated.conf
rip yes {
  preference 200 ;
  interface "eth1" noripin ripout ;
  interface "eth0" noripin noripout ;
};
ospf yes {
  area 156.17.40.0 {
    networks {
      156.17.9.0 masklen 25;
      156.17.30.0 masklen 27;
    };
    interface "eth0"  cost 20 {enable ;};
    interface "eth1"  cost 2 {enable ;};
  };
};
```

Badanie ścieżek sieciowych

```
tahoe-229> traceroute diablo.ict.pwr.wroc.pl
traceroute: Warning: Multiple interfaces found; using 156.17.9.16 @ iprb0
traceroute to diablo.ict.pwr.wroc.pl (156.17.9.14), 30 hops max, 40 byte packets
 1  diablo.ict.pwr.wroc.pl (156.17.9.14)  0.765 ms  0.485 ms  0.454 ms
tahoe-230> traceroute cs.berkeley.edu
traceroute: Warning: cs.berkeley.edu has multiple addresses; using 169.229.60.28
traceroute: Warning: Multiple interfaces found; using 156.17.30.103 @ dnet0
traceroute to cs.berkeley.edu (169.229.60.28), 30 hops max, 40 byte packets
 1  156.17.30.126 (156.17.30.126)  31.216 ms  12.900 ms  5.045 ms
 2  wask-elek.wask.wroc.pl (156.17.252.169)  32.249 ms  2.301 ms  5.074 ms
 3  g48centrum-do-fa-rolnrsm.wask.wroc.pl (156.17.252.166)  9.034 ms  15.443 ms  8.326 ms
 4  g12wcss-g48centrum.wask.wroc.pl (156.17.255.128)  3.122 ms  2.545 ms  3.012 ms
 5  g12centrum-g12wcss.wask.wroc.pl (156.17.255.116)  5.515 ms  2.565 ms  2.854 ms
 6  janus-do-g12centrum.wask.wroc.pl (156.17.255.154)  2.854 ms  3.967 ms  11.354 ms
 7  z-wroclawia.poznan-gw.10Gb.pol34.pl (212.191.224.105)  19.754 ms  14.388 ms  9.459 ms
 8  pol-34.pl1.pl.geant.net (62.40.103.109)  9.385 ms  8.441 ms  8.962 ms
 9  pl.se1.se.geant.net (62.40.96.113)  31.034 ms  37.975 ms  32.183 ms
10  se.uk1.uk.geant.net (62.40.96.126)  64.393 ms  72.984 ms  70.521 ms
11  uk.ny1.ny.geant.net (62.40.96.169)  136.616 ms  138.640 ms  134.374 ms
12  198.32.11.61 (198.32.11.61)  140.980 ms  135.228 ms  139.610 ms
13  chinng-nycmng.abilene.ucaid.edu (198.32.8.82)  174.790 ms  172.945 ms  180.618 ms
14  * * *
15  kscyng-iplsng.abilene.ucaid.edu (198.32.8.81)  177.268 ms  186.013 ms  184.691 ms
16  snvang-kscyng.abilene.ucaid.edu (198.32.8.102)  213.356 ms  216.277 ms  212.206 ms
17  losang-snvang.abilene.ucaid.edu (198.32.8.94)  230.907 ms  231.256 ms  233.520 ms
18  hpr-lax-gsr1--abilene-LA-10ge.cenic.net (137.164.25.2)  226.998 ms  228.701 ms  220.408 ms
19  dc-lax-dc1--lax-hpr1-ge.cenic.net (137.164.22.12)  225.256 ms  344.021 ms  231.914 ms
20  dc-sac-dc1--lax-dc1-pos.cenic.net (137.164.22.127)  230.827 ms  232.051 ms  231.200 ms
21  dc-oak-dc2--csac-dc1-ge.cenic.net (137.164.22.110)  232.613 ms  234.784 ms  231.993 ms
22  ucb--oak-dc2-ge.cenic.net (137.164.23.30)  234.853 ms  233.284 ms  233.320 ms
23  doecev-soda-br-6-4.EECS.Berkeley.EDU (128.32.255.170)  232.299 ms  234.415 ms  234.769 ms
24  sbd2a.EECS.Berkeley.EDU (169.229.59.226)  235.138 ms  237.731 ms  241.399 ms
25  relay2.EECS.Berkeley.EDU (169.229.60.28)  239.828 ms  310.403 ms  233.796 ms
```



```
panamint-214> /usr/sbin/traceroute lab103
traceroute to lab103.ict.pwr.wroc.pl (156.17.41.130), 30 hops max, 38 byte packets
 1  tahoe (156.17.9.16)  0.281 ms  0.197 ms  0.188 ms
 2  hip.ict.pwr.wroc.pl (156.17.30.101)  0.557 ms  0.539 ms  0.753 ms
 3  lab103.ict.pwr.wroc.pl (156.17.41.130)  1.012 ms  0.699 ms  0.636 ms

panamint-212> /usr/sbin/traceroute cs.berkeley.edu
traceroute: Warning: cs.berkeley.edu has multiple addresses; using 169.229.60.163
traceroute to cs.berkeley.edu (169.229.60.163), 30 hops max, 38 byte packets
 1  tahoe (156.17.9.16)  0.434 ms  0.213 ms  0.189 ms
 2  elek.wask.wroc.pl (156.17.30.126)  3.366 ms  11.135 ms  4.716 ms
 3  wask-elek.wask.wroc.pl (156.17.252.169)  33.966 ms  1.768 ms  2.371 ms
 4  g48centrum-do-fa-rolnrsm.wask.wroc.pl (156.17.252.166)  6.845 ms  2.301 ms  2.638 ms
 5  g12wcss-g48centrum.wask.wroc.pl (156.17.255.128)  2.122 ms  1.908 ms  2.890 ms
 6  g12centrum-g12wcss.wask.wroc.pl (156.17.255.116)  3.200 ms  2.405 ms  2.998 ms
 7  janus-do-g12centrum.wask.wroc.pl (156.17.255.154)  1.642 ms  2.274 ms  1.737 ms
 8  z-wroclawia.poznan-gw.10Gb.pol34.pl (212.191.224.105)  6.170 ms  7.317 ms  8.560 ms
 9  pol-34.pl1.pl.geant.net (62.40.103.109)  8.037 ms  7.690 ms  8.391 ms
10  pl.se1.se.geant.net (62.40.96.113)  30.319 ms  30.195 ms  32.243 ms
11  se.uk1.uk.geant.net (62.40.96.126)  66.266 ms  66.338 ms  64.862 ms
12  uk.ny1.ny.geant.net (62.40.96.169)  135.491 ms  146.833 ms  143.482 ms
13  198.32.11.61 (198.32.11.61)  144.286 ms  145.076 ms  139.881 ms
14  chinng-nycmng.abilene.ucaid.edu (198.32.8.82)  186.096 ms  199.670 ms  185.151 ms
15  * iplsng-chinng.abilene.ucaid.edu (198.32.8.77)  294.894 ms  307.807 ms
16  kscyng-iplsng.abilene.ucaid.edu (198.32.8.81)  191.682 ms  180.568 ms  213.788 ms
17  snvang-kscyng.abilene.ucaid.edu (198.32.8.102)  233.531 ms  218.766 ms  222.053 ms
18  losang-snvang.abilene.ucaid.edu (198.32.8.94)  220.544 ms  223.522 ms  220.022 ms
19  hpr-lax-gsr1--abilene-LA-10ge.cenic.net (137.164.25.2)  222.118 ms  219.888 ms  219.943 ms
20  dc-lax-dc1--lax-hpr1-ge.cenic.net (137.164.22.12)  223.883 ms  219.691 ms  220.395 ms
21  dc-sac-dc1--lax-dc1-pos.cenic.net (137.164.22.127)  228.963 ms  235.668 ms  247.016 ms
22  dc-oak-dc2--csac-dc1-ge.cenic.net (137.164.22.110)  238.809 ms  237.555 ms  237.553 ms
23  ucb--oak-dc2-ge.cenic.net (137.164.23.30)  233.315 ms  237.720 ms  344.929 ms
24  doecev-soda-br-6-4.EECS.Berkeley.EDU (128.32.255.170)  236.840 ms  236.123 ms  236.656 ms
25  sbd2a.EECS.Berkeley.EDU (169.229.59.226)  232.506 ms  232.507 ms  256.932 ms
26  soda-cr-1-2-cory-cr-1-1.EECS.Berkeley.EDU (169.229.59.234)  233.133 ms  232.438 ms  232.280 ms
27  relay1.EECS.Berkeley.EDU (169.229.60.163)  233.622 ms  238.531 ms  234.688 ms
```

Translacja nazw symbolicznych — rezolwer

- rezolwer: algorytm translacji adresów symbolicznych na adresy IP stosowany w danym komputerze

- rezolwer ma zwykle postać biblioteki dynamicznej kompilatora C (`libresolv`) implementującej funkcje: `gethostbyname` i `gethostbyaddr`

- z biblioteką tą linkują się zarówno programy systemowe jak i programy użytkowników, co pozwala na jednolitą interpretację danej nazwy

- W starszych systemach rezolwer korzysta z tablicy translacji adresów `/etc/hosts`, i opcjonalnie z serwera lub serwerów DNS zgodnie ze specyfikacją w pliku `/etc/resolv.conf`

```
sierra> cat /etc/hosts

127.0.0.1       localhost
172.16.0.1      sierra.palnet   sierra loghost
172.16.0.3      shasta.palnet   shasta

156.17.9.3      sequoia.ict.pwr.wroc.pl sequoia

156.17.4.4      tryglaw.ii.uni.wroc.pl  tryglaw

156.17.181.99   yoyo.ar.wroc.pl         yoyo
156.17.181.101  geo1.ar.wroc.pl         geo1


sierra> cat /etc/resolv.conf
domain palnet
search palnet ict.pwr.wroc.pl ii.uni.wroc.pl ar.wroc.pl
nameserver 172.16.0.1
```

- W nowszych systemach rezolwer może korzystać z wielu różnych źródeł danych, takich jak: lokalne pliki konfiguracyjne, system DNS, bazy danych NIS lub LDAP. Odwołuje się do tych źródeł zgodnie z precyzyjnym algorytmem określającym w jakiej kolejności z nich korzystać i w jakich sytuacjach odwoływać się do innych źródeł. Algorytm zadany jest plikiem konfiguracyjnym `/etc/nsswitch.conf` natomiast plik `/etc/resolv.conf` określa tylko kolejność zapytań do serwerów DNS.

  Przykłady specyfikacji rezolwera z pliku `/etc/nsswitch.conf`:

  ```
  hosts:      files dns
  ```

  Sprawdzamy najpierw w pliku `/etc/hosts`, a tylko dla nazw, których tam nie ma, odwołujemy się do systemu DNS.

  ```
  hosts:      ldap dns [NOTFOUND=return] files
  ```

  Sprawdzamy kolejno: w bazie danych LDAP i systemie DNS. Gdy system DNS odpowie, że nazwy nie ma, to traktujemy tę odpowiedź jako ostateczną i kończymy zapytanie. Tylko gdyby system DNS nie odpowiadał, sprawdzamy w lokalnym pliku `/etc/hosts`

Translacja nazw symbolicznych — system DNS

- DNS (*Domain Name System*) — hierarchiczny, rozproszony system nazw symbolicznych w Internecie
- oparty na oddelegowaniu administracji domenami różnym instytucjom, korzystającym z własnych serwerów DNS, automatycznie wymieniającym między sobą informacje o administrowanych przez siebie domenach

  domena — poddrzewo hierarchicznego drzewa nazw
- własności: nadmiarowość, replikacja, buforowanie, duża niezawodność i tolerancja błędów, optymalizacja procesu uzyskiwania odpowiedzi w warunkach rzadkich zmian
- serwer DNS — program, którego zadaniem jest podawanie translacji adresu określonego w zapytaniu klienta, i komunikujący się z innymi serwerami DNS, w celu jej znalezienia
- serwery DNS mogą posiadać redundancję — dla danej domeny można wprowadzić oprócz serwera głównego (**primary**), równoważne serwery dodatkowe (**secondary**)

Serwery systemu DNS

- serwer DNS domyślnie jest **rekurencyjny**; w sytuacji gdy nie zna odpowiedzi na otrzymane zapytanie, sam kontaktuje się z innymi serwerami aby ją uzyskać, i udzielić pytającemu klientowi

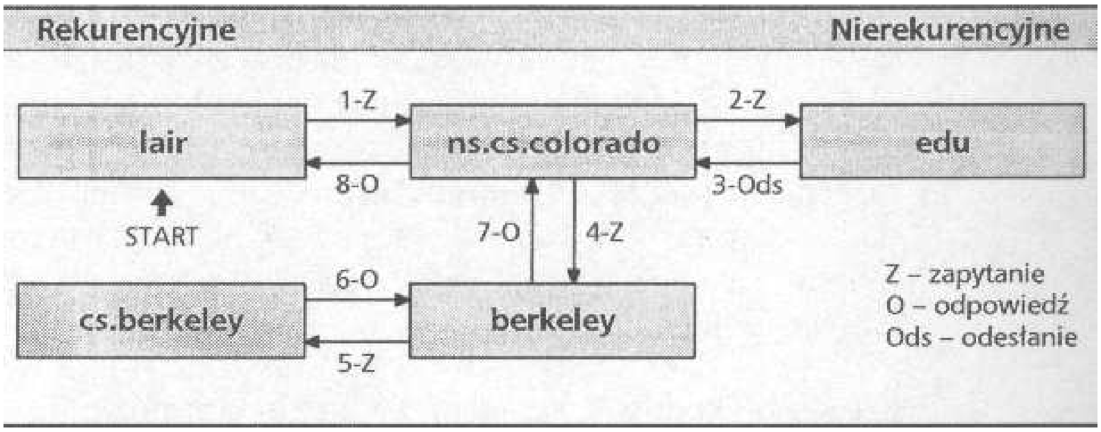
  rekurencyjny serwer jest właściwym rozwiązaniem dla sieci lokalnej, ponieważ pozwala klientom zawsze uzyskiwać odpowiedzi na swoje pytania, a ponadto może przechowywać uzyskane odpowiedzi, i udzielać ich potem kolejnym klientom bez ponownego odpytywania rekurencyjnego

- serwer DNS może być również **nierekurencyjny**; w przypadku nieznajomości odpowiedzi serwer taki nie pyta się innych serwerów, tylko odpowiada tzw. odsyłaczem (ang. *referral*), podającym adres innego, bardziej właściwego dla danej domeny serwera DNS

  serwery DNS wyższego poziomu w hierarchii Internetu (np. serwery główne takich domen jak .com albo .pl) są z zasady nierekurencyjne, więc tym bardziej nie przechowują informacji, które ich nie dotyczą

Dla domen pośrednich pomiędzy siecią lokalną a domeną główną Internetu musimy wybrać pomiędzy pracą rekurencyjną a nierekurencyjną serwera DNS. Jednak nierekurencyjny serwer nie może obsługiwać normalnych klientów, nieprzygotowanych na otrzymanie na swoje zapytanie odpowiedzi w postaci odsyłacza.

Przykład sekwencji odwołań do serwerów DNS dla zapytania o nazwę `mammoth.cs.berkeley.edu` wykonanego na komputerze `lair.cs.colorado.edu`:



Serwery DNS

primary — jest tylko jeden taki serwer dla strefy (ang. *zone*); strefa jest częścią domeny administrowaną przez serwer

secondary — takich może być dla danej strefy wiele, automatycznie aktualizują one swoje dane i ich odpowiedź jest równoważna odpowiedzi serwera *primary*

caching-only — nie jest właściwym źródłem informacji o żadnej strefie, nie posiada własnych informacji tylko realizuje funkcję rekurencyjnego odpytywania innych serwerów i przechowuje informacje przez dozwolony okres; można go uważać za rodzaj aktywnego klienta; jeśli nie chcemy zakładać w danym systemie serwera DNS, ale chcemy zaoszczędzić na ruchu sieciowym do zewnętrznych serwerów DNS, to możemy założyć właśnie serwer *caching-only*

Jeden serwer (uruchomiona instancja programu) może być serwerem primary dla jednej strefy (lub kilku), i serwerem secondary dla grupy innych stref, albo może być czystym serwerem caching-only.

Konfiguracja serwera DNS

Pliki konfiguracyjne serwera DNS:

- /etc/named.conf — określa zestaw stref administrowanych przez serwer i położenie plików definiujących strefy
- pliki definiujące strefy — zbiory rekordów zasobów (*resource records*):
  - rekord SOA — definiuje strefę i jej podstawowe parametry
  - rekord NS — identyfikuje serwery DNS autorytatywne (właściwe) dla danej strefy
  - rekord A — powiązanie pojedynczej nazwy z adresem IP
  - rekord TXT — dodatkowe informacje dla rekordu A
  - rekord HINFO — dodatkowe informacje o typie komputera rekordu A
  - rekord MX — określa obsługę usług pocztowych dla adresu
  - rekord CNAME — określa dodatkową nazwę (alias)
  - rekord PTR — powiązanie odwrotne: adresu IP z nazwą symboliczną

Konfiguracja usług sieciowych — superserwer `inetd`

```
# Syntax for socket-based Internet services:
#  <service_name> <socket_type> <proto> <flags> <user> <server_pathname> <args>
#
ftp     stream  tcp     nowait  root    /usr/sbin/in.ftpd       in.ftpd
telnet  stream  tcp     nowait  root    /usr/sbin/in.telnetd    in.telnetd
name    dgram   udp     wait    root    /usr/sbin/in.tnamed     in.tnamed
shell   stream  tcp     nowait  root    /usr/sbin/in.rshd       in.rshd
login   stream  tcp     nowait  root    /usr/sbin/in.rlogind    in.rlogind
exec    stream  tcp     nowait  root    /usr/sbin/in.rexecd     in.rexecd
comsat  dgram   udp     wait    root    /usr/sbin/in.comsat     in.comsat
talk    dgram   udp     wait    root    /usr/sbin/in.talkd      in.talkd
uucp    stream  tcp     nowait  root    /usr/sbin/in.uucpd      in.uucpd
#
# Tftp service is provided primarily for booting.  Most sites run this
# only on machines acting as "boot servers."
#
tftp    dgram   udp     wait    root    /usr/sbin/in.tftpd in.tftpd -s /tftpboot
#
# Finger, systat and netstat give out user information which may be
# valuable to potential "system crackers."  Many sites choose to disable
# some or all of these services to improve security.
#
finger  stream  tcp     nowait  nobody  /usr/sbin/in.fingerd    in.fingerd
systat  stream  tcp     nowait  root    /usr/bin/ps             ps -ef
netstat stream  tcp     nowait  root    /usr/bin/netstat        netstat -f inet
```



```
#
# These services implemented internally by inetd
#
time    stream  tcp     nowait  root    internal
time    dgram   udp     wait    root    internal
echo    stream  tcp     nowait  root    internal
echo    dgram   udp     wait    root    internal
discard stream  tcp     nowait  root    internal
discard dgram   udp     wait    root    internal
daytime stream  tcp     nowait  root    internal
daytime dgram   udp     wait    root    internal
chargen stream  tcp     nowait  root    internal
chargen dgram   udp     wait    root    internal
#
# RPC services syntax:
# <rpc_prog>/<vers> <endpoint-type> rpc/<proto> <flags> <user> <pathname> <args>
rquotad/1  tli rpc/datagram_v wait root /usr/lib/nfs/rquotad  rquotad
walld/1    tli rpc/datagram_v wait root /usr/lib/netsvc/rwall/rpc.rwalld rpc.rwalld
rstatd/2-4 tli rpc/datagram_v wait root /usr/lib/netsvc/rstat/rpc.rstatd rpc.rstatd
```

Kwestie bezpieczeństwa związane z `inetd`

Zagadnienia bezpieczeństwa pojawiają się w wielu elementach konfiguracji sieciowej, jednak kilka kwestii jest bardzo silnie związanych z serwerem usług sieciowych `inetd`:

- kontrola dostępu sieciowego za pomocą systemu zwanego *TCP wrapper*
- programy r∗

TCP wrapper

Konfiguracja serwera `inetd` zabezpieczonego przez program `tcpd` (*TCP wrapper*):

```
diablo-240> grep tcpd /etc/inetd.conf
#ftp    stream  tcp6    nowait  root    /usr/local/sbin/tcpd    in.ftpd
#telnet stream  tcp6    nowait  root    /usr/local/sbin/tcpd    telnetd
#shell  stream  tcp6    nowait  root    /usr/local/sbin/tcpd    in.rshd
tftp    dgram   udp6    wait    root    /usr/local/sbin/tcpd    in.tftpd -s /tftpboot
finger  stream  tcp6    nowait  nobody  /usr/local/sbin/tcpd    in.fingerd
```



```
diablo-237> cat /etc/hosts.allow
in.tftpd : mono :allow
ALL : katmai.magma-net.pl : twist=(/usr/bin/echo 'Indagacja udana')
ALL : meta.members.com.pl : twist=(/usr/bin/echo 'Poskanujcie sie sami')
telnetd : a06.ie.pwr.wroc.pl : allow
proftpd : 156.17.9.130 :allow
proftpd : 156.17.208.138: allow
#ALL : smietanka.t16.ds.pwr.wroc.pl : deny
# Dla dyplomanta
sshd : 195.94.196.142 : allow
sshd : .astercity.net : allow
#a tu koniec
ALL : pr168.wroclaw.sdi.tpnet.pl : twist=(/usr/bin/echo 'Zapraszam do 07 na rozmowe')
ALL : 156.17.9.0/255.255.255.128 rcf931 : allow
ALL :  UNKNOWN : twist=(/usr/bin/echo 'You must have proper DNS entry')
rpcbind : 156.17.9.0/255.255.255.128 : rfc931 : allow
#NA kompilator sun1000
rpcbind : 156.17.1.47 : rfc931: allow
rpcbind : 156.17.5.93 : rfc931: allow
rpcbind : ALL : rfc931 : deny
proftpd : 213.25.229.96 :allow
proftpd : 213.25.228.64 :allow
proftpd : 217.96.155.150 :allow
sshd : ALL : rfc931 :allow
in.rshd : a06.ie.pwr.wroc.pl :allow
ALL : ALL : rfc931 :deny
```

Usługi r∗

Następujące polecenia pozwalają użytkownikowi posiadającemu konto na innej maszynie unixowej (lub Unixo-podobnej) na wykonywanie operacji bez konieczności logowania się, o ile zdalna maszyna realizuje odpowiednie usługi, i uważa maszynę lokalną za **równoważną** sobie:

rcp — kopiowanie plików

rlogin — włączanie się na zdalną maszynę

rsh — wykonywanie poleceń na zdalnej maszynie przez interpreter poleceń

Usługi te wykorzystują koncepcję równoważności maszyn (ang. *host equivalence*) zakładającą, że użytkownik prawidłowo wlogowany na jednej z równoważnych maszyn, może być wpuszczony na drugą bez autoryzacji.

Równoważność maszyn może zdefiniować administrator w pliku `/etc/hosts.equiv`. Może ją również zdefiniować użytkownik dla swojego konta za pomocą pliku `~/.rhosts`.

```
$HOME/.rhosts                     /etc/hosts.equiv
hostname    username              hostname
                                  +           username
                                  -hostname
```


Usługi r∗ — zagrożenia

Równoważność maszyn jest uznawana za niebezpieczną i niepożądaną własność, a ze względu na brak kontroli nad jej stosowaniem przez użytkowników, normalną praktyką jest wyłączanie usług r∗, zwykle obsługiwanych przez `inetd`.

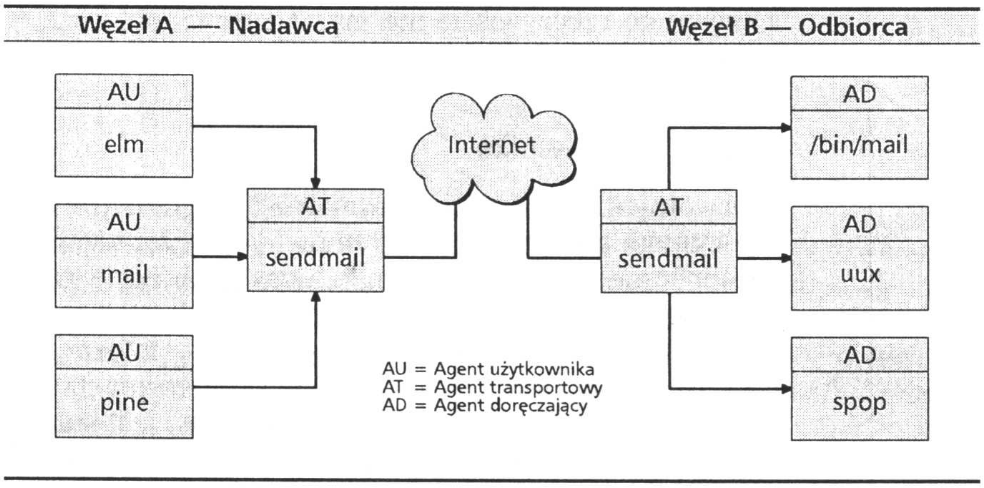
Jedną z przyczyn, dla których równoważność maszyn jest zagrożeniem, z którego administrator może nie zdawać sobie sprawy, jest niezrozumienie zasad interpretowania plików `.rhosts` i `/etc/hosts.equiv`:

- jeśli administrator wyłączy jakiś zdalny komputer lub użytkownika w pliku `/etc/hosts.equiv` to indywidualni użytkownicy i tak mogą udzielić im zezwoleń w swoich plikach `.rhosts`
- pozycja `hostname username` wpisana w pliku `/etc/hosts.equiv` jest interpretowana w ten sposób, że pozwala zdalnemu użytkownikowi `username` włączać się na konto **dowolnego** użytkownika lokalnego

Dobrym rozwiązaniem jest również stosowanie programów `scp` i `ssh` zamiast `rcp` i `rsh`, i można wręcz podłożyć programy s∗ w miejsce programów r∗ w systemie. W ten sposób efektywnie eliminujemy programy r∗ i ich odpowiadające usługi sieciowe, ale jednocześnie jakby „zachęcamy" użytkowników do stosowania bardziej bezpiecznych programów.

Przepływ poczty elektronicznej przez sieć

- UA - agent użytkownika, np. mailx, pine, mutt, outlook
- MTA - agent transferowy, np. sendmail, qmail, postfix
- DA - agent doręczający, np. lmail, procmail



System poczty elektronicznej — zagadnienia

- Protokół wymiany poczty SMTP: w założeniu dowolny komputer może przesłać pocztę do odbiorcy o dowolnym adresie, korzystając z dowolnego innego komputera jako przekaźnika.

  W ten sposób pełnosprawny serwer pocztowy jest tzw. otwartym przekaźnikiem (ang. *open relay*) poczty. Ta cecha powoduje, że, utrudnione, lub wręcz niemożliwe jest zwalczanie *spam*-u, czyli masowo rozsyłanych przesyłek reklamowych.

  Rozwiązania: możliwe, ale niezgodne z dotychczasowymi standardami.
- Inną cechą protokołu SMTP jest autentykacja klienta, a raczej jej brak. Dowolny komputer może wysłać pocztę w czyimkolwiek imieniu, przedstawiając ją jakby pochodziła od kogoś innego.

  Ponownie, rozwiązanie tego problemu w ramach istniejącego systemu wymiany poczty elektronicznej nie jest możliwe.
- Brak szyfrowania jest innym mankamentem poczty elektronicznej — listy są wymieniane w sieci otwartym tekstem. Użytkownicy mogą szyfrować treści swoich przesyłek, ale jest to uciążliwe, i przerzuca na nich problem, który jest niedostatkiem systemu poczty elektronicznej.

  Szyfrowanie pozwoliłoby rozwiązać wiele problemów poczty e-mail, przez wprowadzenie odpowiednich nowych standardów i wymagań.

Sendmail — zasada działania

Historycznie sendmail słynął z dziurawych zabezpieczeń, a raczej ich braku. Jednak jest najczęściej stosowanym w systemach Unix programem MTA. Jest skomplikowanym systemem, o dużych możliwościach konfiguracji.

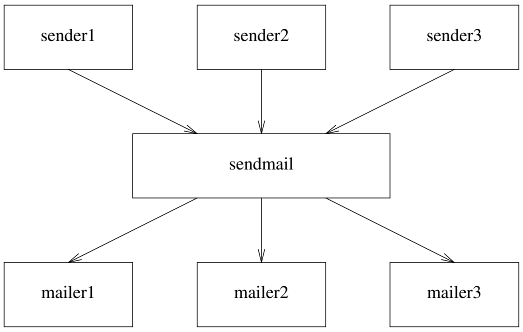


Figure 1 — Sendmail System Structure.

Dla sendmaila obojętne jest skąd pochodzi przesyłka, którą dostał. Jego zadaniem jest rozpoznać jej przeznaczenie, i przekazać ją jednemu z **mailerów**. Mailerami są, na przykład, lokalny program doręczania przesyłek do skrzynek pocztowych użytkowników (mailer local), oraz mechanizm wysyłania przesyłek do innego komputera przez sieć (mailer smtp).

Sendmail — konfiguracja (1)

Sendmail jest systemem regułowym, którego działanie sterowane jest zestawem reguł przepisujących nagłówki i/lub treść listu, jak również rozpoznających adres docelowy i wybierających odpowiedni mailer.

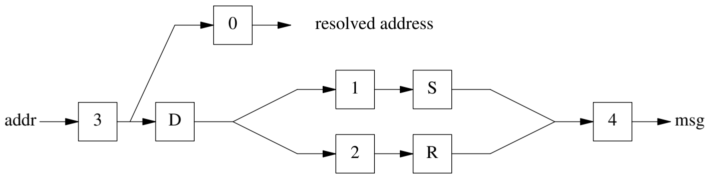


Figure 1 — Rewriting set semantics
D — sender domain addition
S — mailer-specific sender rewriting
R — mailer-specific recipient rewriting

Sendmail posiada pewne wydzielone zbiory reguł, a lokalna konfiguracja może zdefiniować dodatkowe, do zadań specjalnych. Napisanie pełnego zestawu reguł sendmaila od podstaw jest trudne, i rzadko dziś stosowane. Zamiast tego stosuje się pomocnicze systemy automatycznej generacji zestawu reguł sendmaila.

Sendmail — konfiguracja (2)

- „normalne” drogi przepływu poczty
- specjalne drogi przepływu poczty
- specjalne zakazy i zezwolenia
- dodatkowe konwersje adresu, np. ukrywanie części adresu
- aliasy

```
OSTYPE(hpux10)dnl
FEATURE(masquerade_envelope)dnl
MASQUERADE_AS(stud.ict.pwr.wroc.pl)dnl
MASQUERADE_DOMAIN(inyo.ict.pwr.wroc.pl)dnl
MASQUERADE_DOMAIN(diablo.ict.pwr.wroc.pl)dnl
MASQUERADE_DOMAIN(panamint.ict.pwr.wroc.pl)dnl
define(‘SMART_HOST’, smtp:diablo.ict.pwr.wroc.pl)dnl
MAILER(local)dnl
MAILER(smtp)dnl
LOCAL_NET_CONFIG
Cw stud.ict.pwr.wroc.pl
Cw inyo.ict.pwr.wroc.pl
Cw diablo.ict.pwr.wroc.pl
Cw panamint.ict.pwr.wroc.pl
```

System NFS — koncepcje

- sieciowy system plików — umożliwia współdzielenie systemów plików (albo kartotek) między komputerami
- prawa dostępu oparte na identyfikatorach użytkownika — wymaga jednolitego stosowania identyfikatorów w całej jednostce
- serwer bezstanowy (połączenie może „przeżyć” restart serwera), wersja 3 protokołu NFS jest stanowa
- liczne parametry serwera i klienta związane z niuansami transmisji sieciowej, praw dostępu, itp.

Konfiguracja systemu NFS — serwer

```
amargosa% cat /etc/exports
/var/spool/mail         sequoia(rw) tahoe(rw) mojave(rw)
/tmp                    sequoia(rw)


whitney% cat /etc/exports
/export/home       reksio(rw,sync) sequoia(rw,sync) tahoe(rw,sync) \
  inyo(rw,sync,no_root_squash,insecure_locks,insecure) diablo(rw,sync) \
  panamint(rw,sync) amargosa(rw,sync) becks(rw,sync) honolulu(rw,sync)
/export/home2       reksio(rw,sync) sequoia(rw,sync) tahoe(rw,sync) \
  inyo(rw,sync,no_root_squash,insecure_locks,insecure) diablo(rw,sync) \
  panamint(rw,sync) honolulu(rw,sync)
/export/mail    inyo(rw,sync,no_root_squash,insecure) diablo(rw,sync,no_root_squa
  panamint(rw,sync,no_root_squash) sequoia(rw,sync,no_root_squash)
/tftpboot/156.17.9.20          tioga.ict.pwr.wroc.pl(rw,sync,no_root_squash)
/usr                           tioga.ict.pwr.wroc.pl(ro)


diablo% cat /etc/dfs/dfstab
share -F nfs -o rw=panamint:inyo:carlsberg:miller:faxe:\
  tuborg:corona:grolsch:amstel:eb:lech:heineken:okocim:\
  zywiec,root=panamint:inyo   /var/mail
```

Konfiguracja systemu NFS — klient

- plik konfiguracyjny klienta NFS

```
panamint% cat /etc/fstab
...
whitney:/home      /home            nfs   defaults 0 0
diablo:/var/mail   /var/spool/mail  nfs   noexec 0 0
```

- klient może używać automountera

```
sequoia% cat /etc/auto_master
+auto_master
/net   -hosts   -nosuid,nobrowse,noexec,soft,intr,noquota
/home  auto_home  -nobrowse,nosuid
/mail  auto_mail  -nobrowse,soft,intr,noquota
```

- plik konfiguracyjny `/etc/auto_home`

```
sequoia% cat /etc/auto_home
witold sequoia:/export/home/witold
qc whitney:/home/qc
...
```

Protokół NTP synchronizacji czasu

Ważną kwestią we współczesnych systemach komputerowych jest kwestia utrzymywania poprawnego, i dość dokładnego czasu. Pierwszym problemem jest w ogóle konfiguracja zegara systemowego, strefy czasowej, obsługa czasu letniego, oraz synchronizacja zegara sprzętowego z zegarem systemowym. Te procedury różnią się w zależności od wersji Unixa i sprzętu, na którym pracuje system, i nie będziemy się w tym kursie w nie wgłębiać.

Drugim problemem, zwłaszcza dla systemów pracujących bardzo długo, np. wiele miesięcy, jest problem niedokładności zegara sprzętowego i wprowadzania korekt. Te niedokładności mogą wynosić wiele sekund i powodować takie problemy jak:

- trudności (lub niemożność) ustalenia dokładnej sekwencji zdarzeń analizowanych w logach systemowych (w ciągu kilku sekund mogą tam być zarejestrowane tysiące zdarzeń),

- niepoprawna praca sieciowych systemów plików (NFS), sieciowych macierzy dyskowych (SAN), itp.

Drugi problem można rozwiązać korzystając z serwerów czasu protokołu NTP.


```
# ntpdate -d sierra.palnet
15 May 09:11:37 ntpdate[5805]: ntpdate 3-5.93e+sun 03/06/05 23:16:45 (1.4)
transmit(172.16.0.1)
receive(172.16.0.1)
transmit(172.16.0.1)
receive(172.16.0.1)
transmit(172.16.0.1)
receive(172.16.0.1)
transmit(172.16.0.1)
receive(172.16.0.1)
transmit(172.16.0.1)
server 172.16.0.1, port 123
stratum 2, precision -15, leap 00, trust 000
refid [150.254.183.15], delay 0.02664, dispersion 0.00002
transmitted 4, in filter 4
reference time:    c812a805.12d5b000  Mon, May 15 2006  9:11:01.073
originate timestamp: c812a829.de0c0000  Mon, May 15 2006  9:11:37.867
transmit timestamp:  c812a829.c7f3b000  Mon, May 15 2006  9:11:37.781
filter delay:  0.02702  0.02679  0.02670  0.02664
                   0.00000  0.00000  0.00000  0.00000
filter offset: 0.084593 0.084692 0.084707 0.084664
                   0.000000 0.000000 0.000000 0.000000
delay 0.02664, dispersion 0.00002
offset 0.084664

15 May 09:11:37 ntpdate[5805]: adjust time server 172.16.0.1 offset 0.084664 sec

ntpdate -t 4 -v swisstime.ethz.ch sierra.palnet cyber.ict.pwr.wroc.pl
15 May 09:48:01 ntpdate[6125]: ntpdate 3-5.93e+sun 03/06/05 23:16:45 (1.4)
15 May 09:48:02 ntpdate[6125]: adjust time server 172.16.0.1 offset -0.000881 sec
```

Serwer xntpd

Program `ntpdate` pozwala korygować zegar komputera i może być wywoływany z crona. Jednak znacznie pewniejsze i dokładniejsze ustawienie czasu pozwala osiagnąć program `xntpd`, który wykorzystując szereg zaawansowanych algorytmów analizy danych może korygować czas na podstawie źródeł sprzętowych, serwerów internetowych, a także sam może być serwerem protokołu NTP.

Konfiguracja serwera `xntpd`:

```
server cyber.ict.pwr.wroc.pl
server wask.wask.wroc.pl
server vega.cbk.poznan.pl
server swisstime.ethz.ch
broadcast 224.0.1.1 ttl 4
enable auth monitor
driftfile /var/ntp/ntp.drift
statsdir /var/ntp/ntpstats/
filegen peerstats file peerstats type day enable
filegen loopstats file loopstats type day enable
filegen clockstats file clockstats type day enable
logconfig =syncstatus +sysevents
```


Obserwacja pracy serwera NTP — program `ntptrace` pokazuje ścieżkę serwerów NTP, według których bieżący serwer ustawiał aktualny czas:

```
shasta-694> ntptrace sierra
sierra: stratum 2, offset 0.003171, synch distance 0.06274
vega.cbk.poznan.pl: stratum 1, offset -0.042926, synch distance 0.00140, refid 'PPS'
```

Program `ntpq` służy do wydawania różnych zapytań i poleceń serwerowi NTP. Między innymi może wyświetlić konfigurację jego partnerów:

```
shasta-695> ntpq -p sierra
    remote           refid      st t when poll reach   delay   offset    disp
==============================================================================
 NTP.MCAST.NET   0.0.0.0        16 -    -   64    0     0.00    0.000 16000.0
 cyber.ict.pwr.w wask.wask.wroc. 3 u  921 1024  377    69.40  -10.061    4.85
+wask.wask.wroc. tik.cesnet.cz   2 u   40   64  377    71.04  -11.131   34.44
*vega.cbk.poznan .PPS.           1 u   58   64  377    62.26   -8.317   19.79
+anna.mat.uni.to ntps1-0.cs.tu-b 2 u   24   64  177    67.87   -9.495    6.97
-swisstime.ee.et swisstime2.ee.e 2 u   66  256  277    69.60    3.913   24.22
```

Znaczek w pierwszej kolumnie symbolizuje wykorzystanie informacji przez algorytm NTP: spacja i minus oznaczaja odrzucenie, plus oznacza zakwalifikowanie źródła do ostatecznego zbioru serwerów, a gwiazdka wybór serwera do synchronizacji czasu.

Zarządzanie sieciami

W większych instytucjach istnieją rozbudowane instalacje sieciowe i pojawia się problem usprawnienia zarządzania nimi:

- wykrywanie błędów w sieciach, bramach, serwerach
- mechanizmy zawiadamiania administratora o problemach
- monitorowanie w celu podejmowania decyzji o wyrównywaniu obciążenia, rozbudowy, inwestycji
- dokumentowanie
- ułatwienie czynności administracyjnych z centralnego miejsca

Wykrywanie uszkodzeń sieci

- wynikają z: nieprawidłowych połączeń, awarii mediów lub urządzeń, przeciążenia określonych elementów
- można je skutecznie prowadzić za pomocą prostych, ogólnie dostępnych narzędzi: ping, traceroute, netstat
- ping: stan poszczególnych serwerów, drożność połączeń, opóźnienia transmisji
- traceroute: dynamiczny obraz połączeń, miejsce niedrożności sieci
- netstat: stan interface'ów pojedynczej maszyny, liczba błędów, liczba kolizji w segmencie sieci
- wartości uzyskane tymi metodami mają znaczenie względne, i aby były znaczące, trzeba wykonać pomiarów wielokrotnie, w nieregularnych odstępach czasu, i wyeliminować te spowodowane mniej istotnymi stanami chwilowymi
- pomiary te mogą być wykonywane ręcznie lub automatycznie (skrypty); w tym drugim przypadku trzeba jeszcze zorganizować mechanizm rejestracji i powiadamiania administratora

Protokoły zarządzania sieciami — SNMP

Zastosowanie protokołu zarządzania siecią wprowadza porządek i systematykę. Wszystkie urządzenia sieciowe posługują się jednym językiem, i — jeśli protokół zostanie konsekwentnie zaimplementowany — można je monitorować, konfigurować, i resetować zdalnie, z jednego miejsca.

Bardzo prostym, wprowadzonym już w latach 80-tych, ale jedynym ogólnie przyjętym protokołem zarządzania siecią jest *Simple Network Management Protocol*. Definiuje on hierarchiczną przestrzeń nazw zarządzanych danych, oraz sposób czytania i zapisu tych danych w każdym węźle.

SNMP wprowadza szereg terminów, którymi określa zarówno swoje własne elementy, jak również obiekty w zarządanej sieci, i posiadane przez nie dane.

SNMP: podstawowe pojęcia

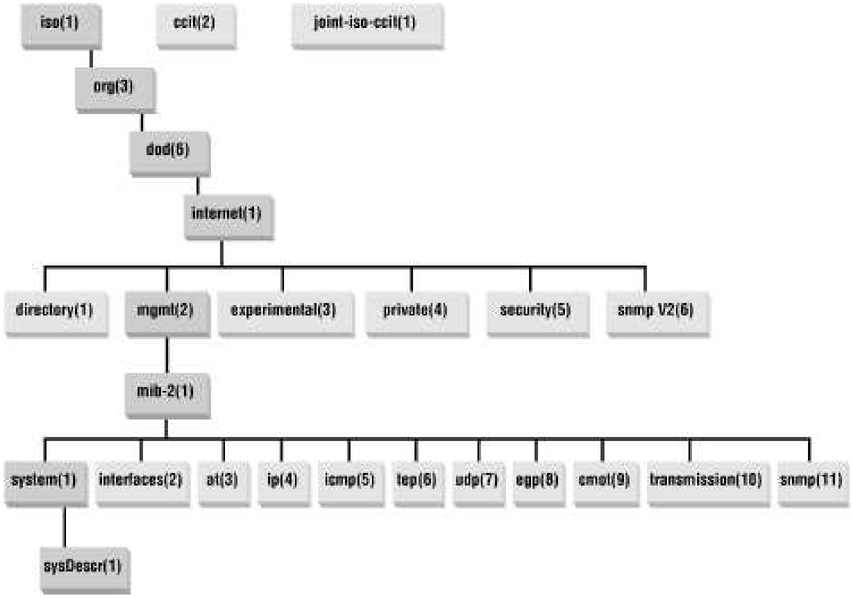
Urządzenia istniejące w zarządzanej sieci określane są jako jednostki (*network entities*), natomiast poszczególne dane w nich istniejące nazywane są obiektami. Natura poszczególnych obiektów może być różna: mogą być sprzętowymi przełącznikami na urządzeniu, programowymi rejestrami konfiguracyjnymi, licznikami, itp. SNMP jednolicie traktuje je jako obiekty i definiuje operacje, które mogą być na nich wykonywane. Każdy obiekt posiada nazwę i wartość, a więc może być uważany za zmienną (lub stałą).

Obiekty grupowane są w hierarchiczne, drzewiaste struktury, nazywane MIB (*Management Information Base*). SNMP wprowadził zestaw rozpoznawanych obiektów, na przykład, w MIB I: opis systemu, liczba interface'ów sieciowych, adresy IP poszczególnych interface'ów, liczniki pakietów przychodzących i wychodzących przez poszczególne. interface'y, oraz tablicę aktywnych połączeń TCP.

Wartością obiektu MIB może być: liczba całkowita, string, identyfikator obiektu, lub wartość pusta. Wartości mogą być grupowane w sekwencje wartości różnych typów, a sekwencje w tabele.

SNMP: drzewo MIB

Obiekty na każdym poziomie w hierarchicznym drzewie posiadają numery, jak również nazwy symboliczne. Konkretny obiekt na drzewie określony jest zatem ścieżką, którą zapisuje się z kropkami, podobnie jak domenowe adresy IP. Na przykład, jak widać poniżej, obiekt `sysDescr` zawierający tekstowy opis systemu danego urządzenia położony jest na ósmym poziomie w drzewie MIB i ma ścieżkę 1.3.6.1.2.1.1.1



Ścieżki obiektów mogą być względne lub bezwzględne. Bezwzględne ścieżki zaczynają się kropką, natomiast względne ścieżki interpretowane są tak, jakby zaczynały się od obiektu `mib-2`. Zatem poprawnymi i równoważnymi ścieżkami dla obiektu `sysDescr` są zarówno .1.3.6.1.2.1.1.1 jak i 1.1 a ponieważ możliwe jest również stosowanie nazw symbolicznych, zatem poprawną ścieżką tego samego obiektu jest też
`.iso.org.dod.internet.mgmt.mib-2.system.sysDescr.0`.

SNMP: operacje

Operacje definiowane przez SNMP nazywają się w jego terminologii PDU (*Protocol Data Units*):

- `get-request` — zapytanie obiektu o wartość zmiennej
- `get-next-request` — zapytanie o wartość następnej zmiennej, np. przy sekwencyjnym listowaniu tablicy danych
- `set-request` — żądanie ustawienia wartości zmiennej
- `trap/snmpV2-trap` — konfiguruje urządzenie tak, aby zawiadamiało swoją jednostkę zarządzającą o jakimś zdarzeniu, na przykład takim jak restart urządzenia, lub przekroczenie przez jakiś licznik pewnej wartości progowej
- `response` — odpowiedź urządzenia, potwierdzenie, itp.

SNMP: przykłady

```
sequoia-475> snmpget -c public -v1 hp5 SNMPv2-SMI::mib-2.43.10.2.1.4.1.1
SNMPv2-SMI::mib-2.43.10.2.1.4.1.1 = Counter32: 4528

sequoia-476> snmpget -v 1 -c public 156.17.9.39 SNMPv2-MIB::sysContact.0
SNMPv2-MIB::sysContact.0 = STRING: Super Administrator

shasta-730> snmpget -c public -v1 printer SNMPv2-SMI::mib-2.43.11.1.1.6.1.1
SNMPv2-SMI::mib-2.43.11.1.1.6.1.1 = STRING: "Toner Cartridge HP C8061X"
shasta-731> snmpget -c public -v1 printer SNMPv2-SMI::mib-2.43.11.1.1.8.1.1
SNMPv2-SMI::mib-2.43.11.1.1.8.1.1 = INTEGER: 4500
shasta-732> snmpget -c public -v1 printer SNMPv2-SMI::mib-2.43.11.1.1.9.1.1
SNMPv2-SMI::mib-2.43.11.1.1.9.1.1 = INTEGER: 2340

sequoia-478> snmpwalk -v 1 -c public hp5
SNMPv2-MIB::sysDescr.0 = STRING: HP ETHERNET MULTI-ENVIRONMENT,ROM A.05.03,JETDIRECT,JD24,EEPROM A
SNMPv2-MIB::sysObjectID.0 = OID: SNMPv2-SMI::enterprises.11.2.3.9.1
...

sequoia-481> snmpset -v 1 -c student_rw 156.17.9.39 SNMPv2-MIB::sysLocation.0 s "test"
SNMPv2-MIB::sysLocation.0 = STRING: test

sequoia-482> snmpwalk -v 2c -c public  156.17.9.56
SNMPv2-MIB::sysDescr.0 = STRING: Linux reksio 2.6.14.5 #8 PREEMPT Fri Jan 13 11:11:04 CET 2006 i68
...

sequoia-483> snmpwalk -v 2c -c grupa_ro  156.17.9.56
...
```

Przykład użycia operacji `trap` (wersja 1):

```
> snmptrap -v 1 -c public  diablo:6666 ''  '' 1 2 ''

Komunikat daemona:
2006-05-24 19:56:33 10.0.0.12(via 84.40.238.85) TRAP, SNMP v1, community public
        SNMPv2-SMI::enterprises.3.1.1 Warm Start Trap (2) Uptime: 3:26:25.24
```

Operacje `trap` w wersji 2 mają inną postać w protokole i inna jest składnia wywołania polecenia `snmptrap`:

```
snmptrap -v 2c -c private diablo:6666 '' SNMPv2-MIB::warmStart

Komunikat daemona:
2006-05-24 20:02:27 xdsl-11093.wroclaw.dialog.net.pl [84.40.238.85]:
        DISMAN-EVENT-MIB::sysUpTimeInstance = Timeticks: (1273873) 3:32:18.73
SNMPv2-MIB::snmpTrapOID.0 = OID: SNMPv2-MIB::warmStart
```


SNMP: uprawnienia

Pierwsze wersje SNMP (1 i 2) oparte były na prostym modelu przyznawania uprawnień, polegającym na wprowadzeniu grup zwanych *communities*.

Przynależność do grupy wymaga znajomości jej nazwy. Znajomość nazwy grupy jest również wystarczająca do udowodnienia przynależności do grupy, zatem nazwa grupy pełni rolę hasła.

Każdy obiekt w MIB posiada jeden z trzech trybów dostępu: `read-only`, `read-write`, i `none`.

W SNMP wersji 3 wprowadzono mechanizmy użytkowników, haseł i szyfrowania.

Materiały sieciowe na temat SNMP:

Strona CISCO poswiecona SNMP:
`http://www.cisco.com/en/US/tech/tk648/tk362/tk605/tsd_technology_support_sub-protocol_home.html`

Baza obiektow SNMP:
`http://tools.cisco.com/Support/SNMP/do/BrowseOID.do?local=en`