

Krzysztof Bogusławski

Akademickie
Centrum
Informatyki PS



Wydział Informatyki PS



Wydział Informatyki

Sieci komputerowe i Telekomunikacyjne

RUTING

Krzysztof Bogusławski
tel. 4 333 950
kbogu@man.szczecin.pl

MENU

1. KONFIGURACJA INTERFEJSU

2. KONFIGURACJA RutowANIA

Rutowanie - przekazywanie datagramów na podstawie informacji zawartych w tablicy rutowania.

MENU

1. KONFIGURACJA INTERFEJSU

2. KONFIGURACJA RUTOWANIA

Konfiguracja interfejsu konieczna jest w sytuacji, gdy protokoły sieciowe współpracują z więcej niż jednym rodzajem fizycznej sieci. W przeciwnym razie, oprogramowanie korzystające z jedyne go dostępnego interfejsu dysponuje niezbędnymi informacjami na jego temat i konfiguracja ręczna przez administratora jest zbędna.

MENU

1. IFCONFIG - *ustawianie interfejsów*

2. IFCONFIG - *sprawdzanie ustawień*

3. NETSTAT

W konfiguracji interfejsu sieciowego najbardziej przydatnymi poleceniami są *ifconfig* i *netstat*. *Ifconfig* służy głównie do wykonywania zadań czysto konfiguracyjnych, może być jednak także użyty przy sprawdzaniu dokonanych ustawień. *Netstat* natomiast służy praktycznie wyłącznie do uzyskiwania danych na temat statusu interfejsów.

MENU

1. IFCONFIG - *ustawianie interfejsów*

2. IFCONFIG - sprawdzanie ustawień

3. NETSTAT

Polecenie *ifconfig* (interface configure) służy m.in. do określania interfejsu TCP/IP, przypisywania mu adresu IP, maski podsieci i adresu rozgłoszeniowego. Oprócz wymienionych wyżej funkcji, *ifconfig* stosowany jest także do udostępniania lub blokowania interfejsu i protokołu ARP, stosowania tzw. *trailers* i definiowania parametru „*routing metric*”, używanego przez RIP.

Podstawowa składnia polecenia **ifconfig** :

ifconfig *interfejs adres netmask maska broadcast adres*

interfejs

- nazwa konfigurowanego interfejsu

adres

- adres IP sieciowego interfejsu; może być przedstawiony w postaci literowej, lecz zaleca się stosowanie postaci liczbowej

netmask *maska* - maska podsieci dla danego interfejsu
(o ile sieć jest podzielona na podsieci)

broadcast *adres* - używany w sieci adres rozgłoszeniowy

Przykład:

```
#ifconfig le0 128.66.12.2 netmask 255.255.255.0 \  
broadcast 128.66.12.255
```

Udostępnianie i blokowanie interfejsów.

Do udostępniania interfejsów za pomocą polecenia `ifconfig` służy argument *up*, natomiast do blokowania, argument *down*.

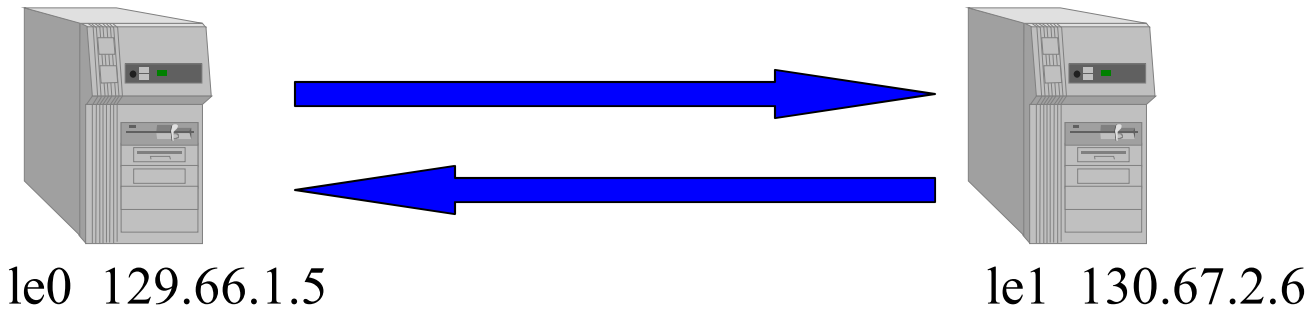
Argument *down* jest szczególnie przydatny przy rekonfiguracji interfejsu. Pewne parametry jak na przykład adres IP, nie mogą być zmienione, jeśli interfejs nie jest zablokowany. Najpierw więc blokujemy interfejs, rekonfigurujemy go, a następnie interfejs odblokowujemy.

Przykład:

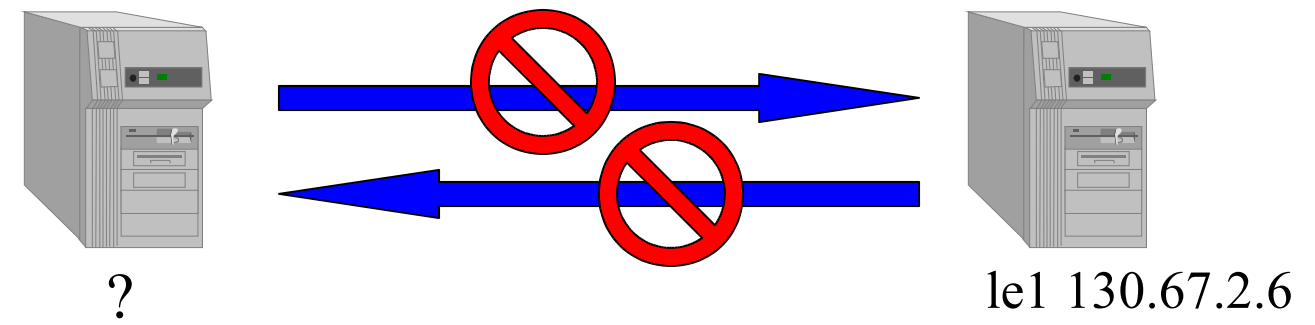
```
#ifconfig le0 down
```

```
#ifconfig le0 129.66.1.2 up
```

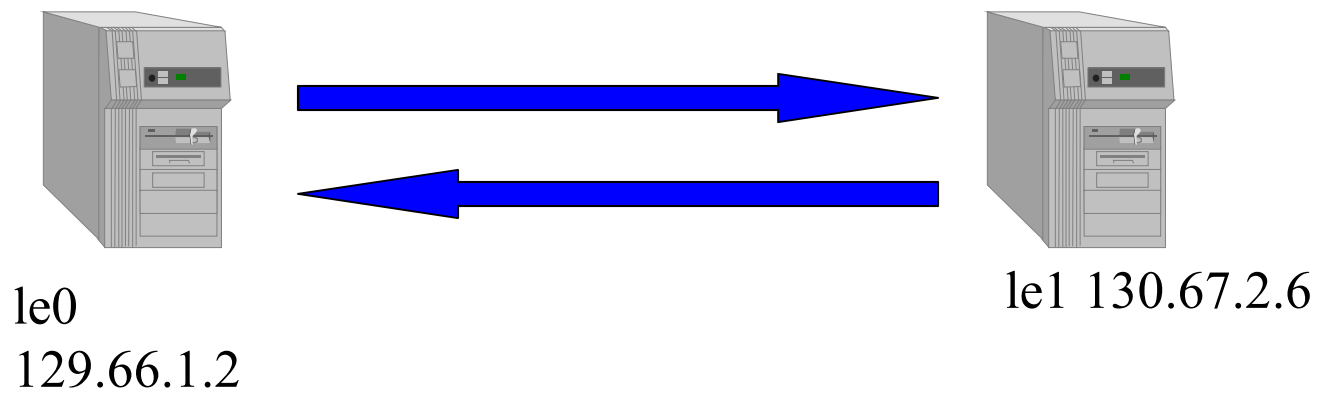

Symulacja zmiany adresu interfejsu



#ifconfig le0 down



#ifconfig le0 129.66.1.2 up



ARP i trailers

Opcje *arp* i *trailers* dotyczą tylko interfejsów Ethernet.

trailers - zezwala na negocjacje dotyczące uzupełniania pakietów IP lub blokuje je; umieszczenie tej opcji redukuje liczbę operacji kopiowania danych, jaką muszą dokonać systemy odbierające informacje

-trailers - blokuje *trailers*

arp - zezwala na stosowanie protokołu ARP (opcja domyślna)

-arp - blokuje *arp*

Metric

Parametr *metric* decyduje w jaki sposób RIP korzysta z informacji o wyborze dróg połączeń. Im wyższa wartość *metric*, tym określona trasa będzie miała mniejsze szanse na wykorzystanie przez RIP.

Przykładowo wydanie poniższego polecenia spowoduje, iż RIP skorzysta z trasy poprzez interfejs 26.104.0.19, dopiero gdy drogi o kosztach 0, 1, 2 okażą się w jakiś sposób niedostępne lub niewystarczające.

Przykład:

```
#ifconfig std0 26.104.0.19 metric 3
```

MENU

1. IFCONFIG - *ustawianie interfejsów*

2. IFCONFIG - ***sprawdzanie ustawień***

3. NETSTAT

Oprócz stosowania polecenia ifconfig podczas konfiguracji interfejsów, można także użyć go do uzyskania informacji o statusie już skonfigurowanych interfejsów.

Sprawdzanie interfejsu za pomocą polecenia *ifconfig*.

Wywołanie polecenia *ifconfig* tylko z nazwą interfejsu powoduje wyświetlenie informacji na temat przypisanych mu wartości.

Przykładowe dane uzyskane w ten sposób to:

```
le0 : flags=63<UP, BROADCAST, NOTRAILERS, RUNNING>  
      inet 128.66.12.2 netmask ffff0000 broadcast 128.66.0.0
```

Znaczniki (*flags*):

- **UP** - interfejs jest dostępny
- **BROADCAST** - stosowane są adresy rozgłoszeniowe
- **NOTRAILERS** - interfejs nie uzupełnia pakietów
- **RUNNING** - interfejs działa

inet - adres przypisany do interfejsu

netmask - stosowana maska podsieci (zapis heksadecymalny)

broadcast - adres rozgłoszeniowy

MENU

1. IFCONFIG - *ustawianie interfejsów*

2. IFCONFIG - *sprawdzanie ustawień*

3. NETSTAT

Program *netstat* przedstawia status wszystkich (zarówno fizycznych jak i programowych) interfejsów. Przydatny jest także do uzyskiwania statystyk sieciowych dla danej sieci.

Najczęściej stosowana składnia polecenia *netstat* ma postać:

netstat -ain

a - wszystkie interfejsy

i - status skonfigurowanych interfejsów

n - udostępniane informacje są w postaci liczbowej

Uzyskane informacje przedstawione zostaną wg. poniższych kategorii:

Name - nazwa przypisana do interfejsu (* - interfejs nie działa)

Mtu - Maximum Transmission Unit; określa rozmiar najdłuższej ramki przesłanej bez fragmentacji przez dany interfejs

Net/Dst - sieć lub komputer dostępny przez dany interfejs

Address - adres IP przypisany do interfejsu

Ipkts - liczba pakietów odebranych przez dany interfejs

Ierrs - liczba pakietów odebranych przez interfejs z błędami

- Opkts* - liczba pakietów wysłanych przez dany interfejs
- Oerrs* - liczba pakietów wysłanych przez dany interfejs z błędem
- Collis* - liczba wykrytych kolizji przy nadawaniu
- Queue* - liczba pakietów oczekujących w kolejce na transmisję przez dany interfejs

Przykład:

```
#netstat -ain
```

Name	Mtu	Net/Dest	Address	Ipkts	Ierrs	Opkts	Oerrs	Collis	Queue
lo0	1500	128.66.0.0	128.66.12.2	1547	1	1127	0	135	0
lo0	1536	127.0.0.0	127.0.0.1	133	0	133	0	0	0

Polecenie *netstat* służy także do przedstawiania zawartości tablicy routowania utworzonej przez polecenie *ifconfig* (rutowanie minimalne). Składnia przyjmie wtedy następującą postać:

```
netstat -rn
```

Przykładowe informacje uzyskane w ten sposób:

Routing tables

Destination	Gateway	Flags	Refcnt	Use	Interface
127.0.0.1	127.0.0.1	UH	1	132	lo0
128.66.12.0	128.66.12.2	U	26	49041	le0

Flags:

- *U* - (up) trasa gotowa do wykorzystania
- *G* - (gateway) trasa prowadzi przez zewnętrzne gateway'e
(w tym przypadku nie występuje, gdyż obie drogi są bezpośrednimi trasami przez lokalne interfejsy)
- *H* - (host) trasa prowadzi tylko do jednego komputera

Z wcześniejszego zapisu przedstawiającego efekt wydania polecenia

netstat -rn na przykładowej maszynie wynikają dwa istotne

wnioski:
• Adresem sieci dla zapętlenia jest 127.0.0.0, a docelowym komputerem jest *localhost*. Zapis ten jest efektem wydania polecenia *ifconfig* podczas konfiguracji interfejsu *lo0*.

W każdej tablicy rutowania musi znaleźć się taki zapis dotyczący *loopback*.

• Trasa do sieci 128.66.12.0 prowadzi poprzez interfejs *le0* o adresie 128.66.12.2

MENU

1. KONFIGURACJA INTERFEJSU

2. KONFIGURACJA Rutowania

Konfiguracja rutowania opiera się na w znacznej mierze na protokołach rutowania i tablicach rutowania. Błędna konfiguracja rutowania, niezależnie od stosowanych rozwiązań, protokołów, czy samego sprzętu, uniemożliwi poprawną komunikację między systemami w sieci.

MENU

1. RUTOWANIE

2.TABLICA RUTOWANIA

3.PROTOKOŁY RUTOWANIA

4.REALIZACJA PROTOKOŁÓW

1. RUTOWANIE

2. TABLICA RUTOWANIA

3. PROTOKOŁY RUTOWANIA

4. REALIZACJA PROTOKOŁÓW

MENU

Rutowanie jest spoiwem łączącym Internet w całość. Bez niego cały ruch TCP/IP byłby ograniczony do jednej fizycznej sieci. Rutowanie pozwala danym z określonej sieci lokalnej osiągnąć cel usytuowany w dowolnym miejscu na świecie. Ważne przy tym jest to, iż trasa prowadzić może przez wiele sieci pośrednich.

Spotykane są 3 rodzaje rutowania, jest to rutowanie:

- ***minimalne*** - stosowane, gdy nie ma połączenia z innymi sieciami TCP/IP, a także gdy nie są stosowane podsieci. Minimalna tablica rutowania konstruowana jest za pomocą polecenia *ifconfig*
- ***statyczne*** - stosowane, gdy trasy przekazywania danych są niezmiennie, oraz gdy sieć posiada ograniczoną liczbę gateway'ów do innych sieci. Statyczne tablice rutowania tworzone są za pomocą polecenia *route*
- ***dynamiczne*** - stosowane, gdy sieć posiada kilka tras do tego samego miejsca przeznaczenia. Dynamiczne tablice rutowania budowane są na podstawie informacji wymienianych przez protokoły rutowania, które dzięki swojej konstrukcji mogą szybciej niż administrator reagować na zmiany zachodzące w sieci (zmieniać tablice routingu).

1. RUTOWANIE

2. TABLICA RUTOWANIA

MENU

3. PROTOKOŁY RUTOWANIA

4. REALIZACJA PROTOKOŁÓW

Tablica rutowania (niezależnie od tego czy jest minimalna, statyczna, czy dynamiczna), jest podstawą trasowania, bez której spotykane dzisiaj rozwiązania okazałyby się w ogromnej większości bezużyteczne. Stanowi punkt odniesienia i źródło informacji niezbędnych podczas wyznaczania marszruty.

Statyczna tablica rutowania

Statyczne tablice rutowania zawierają informacje o trasach przez zewnętrzne gateway'e, dzięki czemu możliwe jest łączenie się z odległymi systemami.

Jednym ze sposobów tworzenia statycznych tablic rutowania jest polecenie *route*.

Składnia:

route add *adres_1 adres_2 miara rutowania*

add - dodaje trasę

delete - usuwa trasę (w takim przypadku występuje zamiast *add*)

adres_1 - adres osiągalny za pośrednictwem danej trasy (adres IP, nazwa sieci z */etc/networks*, nazwa komputera z */etc/hosts*, lub słowo kluczowe *default*)

adres_2 - adres gateway'a; musi to być adres systemu znajdującego się w sieci dołączonej bezpośrednio

miara rutowania - niezbędna przy dodawaniu tras; 0 - trasa prowadzi przez lokalny interfejs; 0< - trasa prowadzi przez zew. gateway

default - jeżeli wystąpi zamiast adresu docelowego, to utworzona zostanie trasa domyślna, używana gdy nie ma konkretnej marszruty do miejsca przeznaczenia (przydatne gdy sieć posiada tylko jeden gateway).

Po zainstalowaniu domyślnej trasy, można upewnić się, że została ona poprawnie dodana do tablicy routingu za pomocą, poznanego przy okazji omawiania konfiguracji interfejsu, polecenia *netstat -rn*.

Przykład:

```
#route -n add default 128.66.12.1 1  
  add net default: gateway 128.66.12.1 1
```

```
#netstat -rn
```

Routing tables

Destination	Gateway	Flags	Refcnt	Use	Interface
127.0.0.1	127.0.0.1	UH	1	132	lo0
default	128.66.12.1	UG	0	0	le0
128.66.12.0	128.66.12.2	U	26	49041	le0

W celu sprawdzenia, czy skonfigurowany system jest w stanie połączyć się z zewnętrznym systemem, stosowane jest polecenie ping.

Przykład:

```
#ping 26.40.0.17
```

```
  PING 26.40.0.17 : 56 data bytes
```

```
 64 bytes from foo.army.mil (26.40.0.17) : icmp_seq=0. time=110. ms
```

```
 64 bytes from foo.army.mil (26.40.0.17) : icmp_seq=1. time=100. ms
```

```
^C
```

```
--26.40.0.17 PING Statistics--
```

```
 2 packets transmitted, 2 packets received, 0% packet loss
```

```
 round trip (ms) min/avg/max = 100/105/110
```

Oznacza to, że trasa do komputerów w Internecie jest poprawna. W przypadku, gdy wywołamy program *ping* by skomunikować się z komputerem w innej sieci (np. podsieci sieci lokalnej), domyślny gateway wyśle przesyłkę przekierowującą typu *ICMP Redirect*, określającą korzystniejszą trasę do danej sieci (przez inny gateway).

Program *ping* wśród standardowych zapisów umieści także wpis informujący o obecności przesyłki tego typu, np.:

ICMP Host redirect from gateway almond.nuts.com (128.66.12.1) to pecan.nuts.com (128.66.12.3) for filbert.nuts.com (128.66.1.2)

Z tego zapisu wynika, iż gateway almond „radzi” by dla pakietów kierowanych pod adres 128.66.1.2, korzystać z gateway’ a pecan.

W takim wypadku w tablicy rutowania pojawi się zapis:

Destination	Gateway	Flags	Refcnt	Use	Interface
128.66.1.3	128.66.12.3	UGHD	0	514	le0

Znacznik D informuje, iż zapis dodany został przez *ICMP Redirect*. By uniknąć powtarzania przesyłek *ICMP Redirect*, należy ustawić marszrutę dla każdej sieci.

Po restarcie systemu, zawartość statycznej tablicy rutowania zostaje utracona.

1. RUTOWANIE

2. TABLICA RUTOWANIA

MENU

3. PROTOKOŁY RUTOWANIA

4. REALIZACJA PROTOKOŁÓW

Protokoły rutowania są to programy wymieniające informacje służące do budowy tablic rutowania. Mimo, iż każdy system dokonuje rutowania (wybierania tras przesyłanych danych), to nie w każdym systemach obecne są protokoły rutowania. Głównymi zadaniami protokołów rutowania jest określenie najlepszej trasy do każdego miejsca przeznaczenia i rozprowadzanie między systemami w sieci informacji o rutowaniu.

Wewnętrzne protokoły rutowania

- RIP
- Hello
- SPF
- OSPF

RIP (Routing Information Protocol)

Wybiera trasy charakteryzujące się najmniejszą „liczbą przeskoków” (hop count). Liczba przeskoków to liczba gateway’ów, przez które muszą przejść dane by trafić do miejsca przeznaczenia. Najlepsza trasa prowadzi przez jak najmniejszą liczbę gateway’ów (algorytm dystansowo - wektorowy). *RIP* nie uwzględnia obciążenia marszrut i ich opóźnienia przy przesyłaniu marszrut. *RIP* znajduje zastosowanie głównie w niedużych sieciach, ponieważ odrzuca on trasy prowadzące przez więcej niż 15 gateway’ów.

Hello

Hello wybiera trasy na podstawie opóźnienia, czyli czasu zużytego przez datagram na pokonanie drogi ze źródła do miejsca przeznaczenia. Pakiet protokołu *Hello* posiada znacznik określający czas jego wysłania, pozwalający określić czas jak długo pakiet docierał do celu.

SPF (Shortest Path First)

SPF, podobnie jak *RIP*, preferuje najkrótsze trasy. Jednak *SPF* może być wykorzystywany w sieciach z większą liczbą ruterów (ponad 15).

OSPF (Open Shortest Path First)

OSPF posiada zdolność wielościeżkowego rutowania o jednakowych kosztach (kilka tras do tego samego miejsca przeznaczenia).

Implementacje IP w UNIX'ie zakładają, że w tablicy routingu jest tylko jedna trasa do każdego miejsca przeznaczenia. IP korzysta z pierwszej napotkanej marszruty, ignorując pozostałe, dlatego *OSPF* nie jest tak szeroko stosowany.

Zewnętrzne protokoły rutowania

- EGP
- BGP

EGP (Exterior Gateway Protocol)

Gateway z wykorzystujący *EGP* rozgłasza, że za jego pośrednictwem dostępne są sieci będące częścią jego systemu autonomicznego (AS). *EGP* nie próbuje wybrać najlepszej drogi, a tylko uaktualnia informacje dystansowo - wektorowe nie przetwarzając ich.

BGP (Border Gateway Protocol)

- wymienia informacje o dostępności systemów autonomicznych
- na podstawie dostarczonych danych o trasach jest w stanie wybrać tę najlepszą. Informacje te noszą nazwę atrybutów ścieżki .
(path attributes). Zawierać mogą one dane służące do wyboru tras na podstawie preferencji administracyjnych (rutowanie o podstawie politycznej).
- rozszerza możliwości systemów w zakresie wyboru tras
- pozwala na zaimplementowanie wytycznych dla rutowania

1. RUTOWANIE

2. TABLICA RUTOWANIA

3. PROTOKOŁY RUTOWANIA

4. REALIZACJA PROTOKOŁÓW

MENU

By protokoły mogły spełniać swoje zadanie, czyli koordynować, ułatwiać i umożliwiać rutowanie, niezbędne są mechanizmy służące do realizacji tych protokołów i jednocześnie czuwające nad ich prawidłową pracą. Przykładem może być np. omówiony dalej *gated*.

Protokół **RIP** realizowany jest przez demona **routed**, który dynamicznie buduje tablicę rutowania korzystając z otrzymanych informacji.

Zadania *routed*:

Na podstawie pakietów informacyjnych **RIP** modyfikuje swoją tablicę rutowania:

- dodaje trasę, jeśli wcześniej nie miał jej w tablicy, lub gdy jej koszt jest niższy od kosztu trasy już występującej
- usuwa trasę jeśli jej koszt jest większy od 15, a także gdy gateway, przez który wiedzie trasa, nie przysyła pakietów informacyjnych (**RIP** zakłada, że taki gateway nie działa poprawnie i nie jest wiarygodny)

Routed może uzupełnić informacje otrzymane z innych systemów i dopiero wtedy tworzyć tabelę. Dodatkowe informacje czerpie wtedy z pliku */etc/gateways* w momencie startu systemu.

Składnia wpisów w pliku */etc/gateways*:

net *adres docelowy* **gateway** *adres gateway'a* **metric** *wartość* **active**

net - występujący w następnej kolejności adres, jest adresem sieci

host - jeśli występuje zamiast *net*, to adres wskazuje na komputer

metric - określa wartość miary służącej do podejmowania decyzji o marszrutowaniu; liczba ta określa liczbę gateway'ów, przez które muszą przejść dane by osiągnąć cel.

active - oznacza, iż trasy mogą być modyfikowane przez RIP

(trasa może zostać usunięta, gdy „jej” gateway nie odpowiada)

passive - oznacza, iż od danego gateway'a nie oczekuje się informacji o marszrutowaniu. Stosowane gdy chcemy uniknąć usunięcia z tablicy wpisu o trasach przez gateway (gdy nie potwierdza swoich tras); w ten sposób trasa staje się trasą statyczną.

Realizacja protokołu **EGP**

EGP aby nawiązać wymianę danych pomiędzy gateway'ami posługującymi się **EGP**, stosuje dialog inicjujący w postaci:

Hello

I-Heard-You (I-H-U)

Taka wymiana komunikatów nosi nazwę poznawania sąsiada, a komputery komunikujące się za pomocą **EGP** nazywane są wprost sąsiadami (*neighbours*).

Po zapoznaniu się z sąsiadem następuje wymiana pakietów:

poll - żądanie informacji o rutowaniu

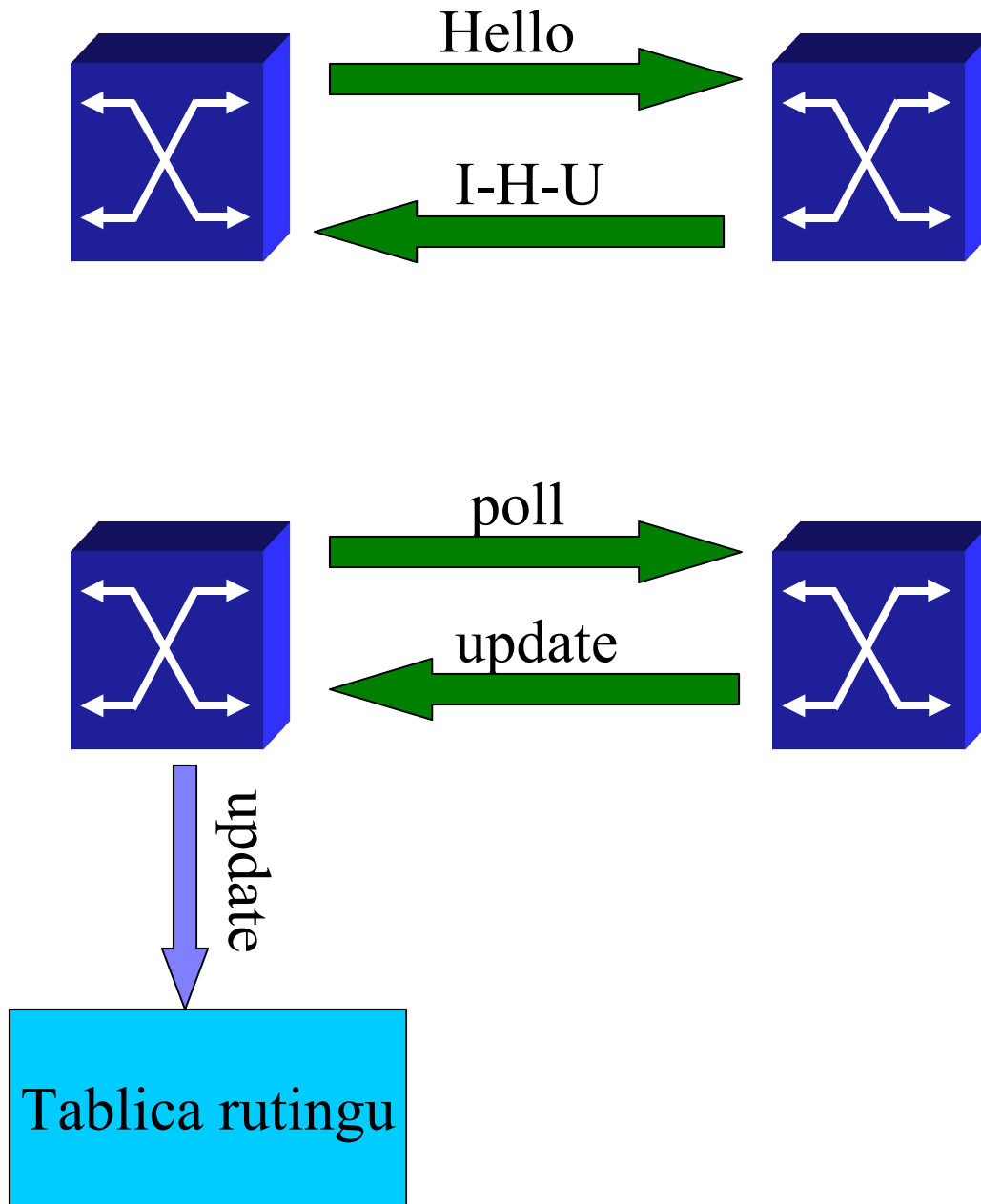
update - odpowiedź zawierająca właściwe informacje o rutowaniu

Jeżeli system otrzyma od swojego sąsiada **EGP** przesyłkę *poll*, natychmiast odpowiada swoim *update*.

cd poprzedniej strony

Po otrzymaniu przesyłki *update*, system modyfikuje swoją tablicę na podstawie zawartych w niej danych. Jeżeli po 3-krotnym wysłaniu żądania typu *poll* ciągle brak odpowiedzi, EGP zakłada, że sąsiad jest niedostępny i usuwa z tablicy wszystkie prowadzące przez niego trasy.

Realizacja protokołu EGP



EGP może działać jako oddzielny proces (*egpup*), lub jako część Gateway Routing Deamon (*gated*).

Jeżeli działa jako *egpup*, to przy starcie systemu, dane czytane są z pliku */etc/egp.init* (lub */etc/egp.conf* - zależnie od systemu), który zawiera następujące opcje:

- **autonomoussystem** *asn* - określa numer systemu autonomicznego; *asn* - oficjalny numer przydzielony do danego systemu
- **egpneighbour** *neighbour* - określa zdalny gateway jako naszego sąsiada EGP. Wartością jest nazwa lub adres IP zdalnego gateway'a
- **egpmaxacquire** *liczba* - maksymalna liczba sąsiadów, z którymi można nawiązać komunikację. EGP może „zapoznać się” z więcej niż jednym sąsiadem naraz (powoduje to przeciążenie protokołu, ale zwiększa jego pewność, gdy jeden z sąsiadów niepewnie odpowiada na wysłane zapytania *poll*).

cd poprzedniej strony

- **egpnetsreachable** *sieć1 sieć2 sieć3 ...* - wskazuje sąsiadom jakie sieci są dostępne przez nasz gateway; jeżeli to wyrażenie zostanie użyte, to udostępnione zostaną wskazane sieci. Jeśli natomiast ten wpis nie wystąpi, to udostępnione zostaną wszystkie sieci
- **net** *przeznaczenie gateway adres metric liczba* - instaluje statyczną trasę (marszruta pasywna)
- **defaultgateway** *adres* - instaluje trasę domyślną; jest ona instalowana przy starcie systemu i usuwana po uzyskaniu informacji od sąsiada EGP. Ponowne jej zainstalowanie jest możliwe przy każdym restarcie systemu

Przykład : /etc/egp.init

```
# configure EGP
autonomoussystem 249
egpmaxacquire 1
egpneighbor 26.1.0.49
egpneighbor 26.21.0.104
egpnetsreachable 128.66.0.0
```

```
#
```


Gateway Routing Deamon (gated)

Gated jest pakietem oprogramowania łączącym w sobie RIP, Hello, BGP i EGP

Wykorzystanie:

- w systemach, w których działa kilka protokołów rutowania, **gated** łączy informacje o marszrutowaniu, otrzymywane od tych protokołów i wybiera najlepsze trasy
- trasy uzyskane za pomocą wewnętrznych protokołów mogą być rozgłaszane poprzez protokoły zewnętrzne. Pozwala to na przekazywanie na zewnątrz informacji dynamicznie dopasowujących się do zmian zachodzących w trasach wewnętrznych
- **gated** upraszcza konfigurację rutowania. Wszystkie protokoły są konfigurowane poprzez jeden plik (*/etc/gated.conf*) za pomocą poleceń o jednorodnej składni

- **gated** jest na bieżąco modyfikowany

Cechy **gated**:

- zewnętrzne funkcje (wymiana informacji o rutowaniu z innymi systemami) nie różnią się od innych realizacji protokołów
- wewnętrzne funkcje (wykorzystywanie otrzymanych z zewnątrz danych do modyfikacji tablicy rutowania) różnią się od spotykanych dotąd rozwiązań. **Gated** przetwarza bowiem otrzymane poprzez różne protokoły informacje o marszrutowaniu. W związku z tym musi je łączyć w sposób umożliwiający modyfikację tablicy

Miary protokołów rutowania zaimplementowane w *gated*

Gated używa miar protokołów przy rozgłaszaniu tras przez te protokoły. Reprezentują one wartości pomocne przy wyborze trasy (im dana miara jest mniejsza, tym trasa przez nią określana jest korzystniejsza).

PROTOKÓŁ	REPREZENTACJA MIARY	ZAKRES	NIE - OSIĄGALNE
RIP	Dystans (hop counts)	0 - 15	16
Hello	Opóźnienie w milisekundach	0 - 29999	30000
BGP	Nieokreślona	0 - 65534	65535
EGP	Dystans (nieużywany)	0 - 254	255

Wartości preferencyjne

Wartości preferencyjne używane są przez **gated** do łączenia informacji o marszrutowaniu pochodzących z różnych źródeł w jedną tablicę rutowania. Wartości pochodzą z zakresu 0 - 255 i podobnie jak w przypadku miar rutowania, mniejsza wartość jest bardziej preferowana.

Typ trasy	Domyślna wartość preferencyjna
Trasa bezpośrednia	0
Przekierowanie ICMP	20
Trasa statyczna	50
Protokół Hello	90
RIP	100
BGP	150
EGP	200

Wartości te mogą być ustawione tak, by preferować trasy przez jakiś interfejs sieciowy, określony protokół lub konkretny gateway. Są one używane tylko w pliku konf. i nie są rozsyłane przez protokoły.

Koniec

Inne prezentacje znajdują się na stronie:

<http://kbogu.man.szczecin.pl>