

Wstęp do BGPv4 w oparciu o oprogramowanie Quagga.

Border Gateway Protocol (BGP)

Tekst ten ma na celu przedstawienie podstawowych założeń protokołu BGPv4, oraz przykładową konfigurację BGP w oparciu o oprogramowanie Quagga, w jednej z najczęściej spotykanych konfiguracji klienta z dwoma dostawcami internetu (multihoming).

Border Gateway Protocol (BGP) jest obecnie podstawowym protokołem, za pomocą którego Internet wymienia pomiędzy poszczególnymi sieciami, nazywanymi też systemami autonomicznymi (AS - ang. Autonomous System), informację o ich dostępności.

BGP w pierwszej wersji został opisany w RFC-1105 w 1989 roku. Wersja czwarta, która jest obecnym standardem używanym w Internecie, opisana jest w RFC-1771 z roku 1995. BGP od wersji pierwszej do czwartej znacznie zyskało na funkcjonalności. Zwiększył się rozmiar pojedynczej wiadomości, wprowadzone zostały atrybuty, za pomocą których można w BGP przenosić dodatkowe informacje. Atrybuty te były stopniowo rozbudowywane z wersji na wersję aby doprowadzić do obecnie wykorzystywanej wersji czwartej.

Podstawową funkcją BGP, tak jak każdego protokołu rutowania, jest wymiana informacji o zakresach adresów, które dostępne są poprzez określone rutery/systemy autonomiczne. Zakresy adresów są określone jako prefiksy (ang. prefix), a ścieżki opisujące kolejne systemy autonomiczne (AS) doprowadzające do wybranego prefiksu - jako trasy (ang. path). BGP nie przenosi informacji o konkretnych urządzeniach, zamiast tego posługuje się terminem systemów autonomicznych. Taki system najczęściej tworzą logicznie wydzielone organizacje, firmy czy dostawcy internetu. To jak w ramach AS przekazywana jest informacja o ścieżkach zależy już jedynie od administratorów danej sieci.

Aby brać udział w wymianie informacji przez BGP każdy system autonomiczny musi posiadać własny numer, który przydzielany jest przez IANA/RIPE. Jest to 32-bitowa liczba - Autonomous System Number (ASN), która jest następnie używana przez BGP przy wymianie informacji. Numery te są unikatowe, aby nie doszło do zapętlenia się routingu. Podobnie jak w przypadku numerów IP, także tutaj można wykorzystać prywatne numery AS. Numery te są często wykorzystywane w dużych sieciach opartych o BGP. Prywatne ASN są numerami z zakresu 64512-65535.

Przykładowe numery ASN:

GTS	-	8246
TPNET	-	5617

W bazach whois RIPE i pozostałych agend IANA przechowywane są informacje dotyczące numerów AS i jednostek, którym zostały przypisane. Aby dowiedzieć się więcej o którymś z numerów ASN można wykorzystać program whois:

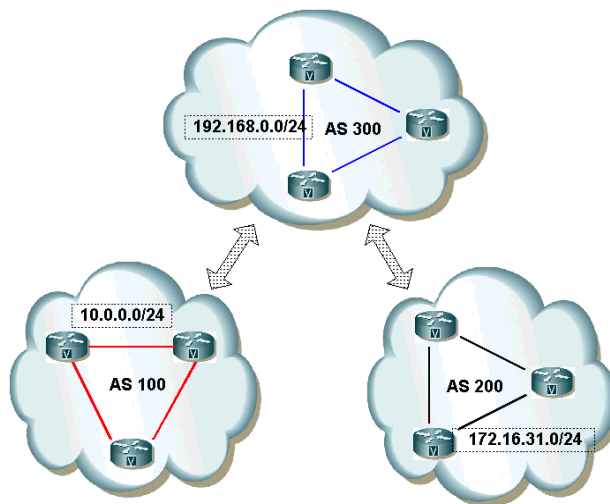
```
#whois as8246

aut-num:      AS8246
as-name:      INTERNET-TECHNOLOGIES-POLSKA-AS
descr:        Internet Technologies Polska Sp. z o.o.
descr:        GTS Internet Partners
...
```

Żeby wykorzystać BGP w sytuacji przez nas rozpatrywanej (klient – 2 różnych dostawców) poza ASN, potrzebne nam będą także adresy IP niezależne od dostawców. Procedura otrzymywania adresów PI (Provider Independent) jest bardzo podobna do procedury otrzymywania numeru AS. Cała procedura składa się z wymiany kilku lub kilkunastu maili z RIPE. Należy jednak pamiętać, że zanim wystąpimy o PI czy ASN musimy mieć wsparcie jednego z dostawców, który musi posiadać status LIR (Local Internet Registry).

Jak to działa – czyli wiedza tajemna.

Celem BGP jest przenoszenie informacji o dostępnych prefiksach poprzez poszczególne ASy. Jest to informacja nie tylko o trasach dostępnych, ale także o zmianach zachodzących w sieci BGP. Routery wykorzystujące BGP nazywane są neighborami. Zestawiają one połączenia (peeringi) pomiędzy sobą. W obrębie danego AS (iBGP) każdy router musi być połączony z każdym (można to obejść za pomocą mechanizmu route reflectorów). Routery nie muszą być ze sobą bezpośrednio połączone. W przypadku kiedy routery odległe są od siebie o kilka hopów, korzysta się z opcji ebgp-multi-hop.



Rysunek 1. BGP pomiędzy trzema odrębnymi sieciami.

Router należący do AS100 informuje pozostałe AS o dostępnej przez niego trasie, czyli 10.0.0.0/24. Podobnie pozostałe ASN informują sąsiadów jakie prefiksy są u nich i przez nich dostępne. AS300 poza tym, że rozgłasza informację o swoim własnym prefiksie 192.168.0.0/24, informuje także sąsiadów jakie inne prefiksy są przez niego dostępne. AS300 informuje AS200 o dostępnej przez niego ścieżce do AS100 i odwrotnie, AS100 jest informowany o AS200. W rzeczywistości wygląda to nieco bardziej skomplikowanie ze względu na ilość AS oraz ich strukturę i rozmieszczenie. Jeśli jest to router tranzytowy prowadzący do Internetu, tras, które otrzymuje może być bardzo dużo (obecnie około 130 000). Jeśli jest to router, który informuje tylko o własnych trasach/zakresach adresów ip, może ich być kilka, a nawet tylko jedna (oczywiście możemy otrzymać pełen zakres tras, ale nie zawsze ma to sens).

Poniższy przykład pokazuje ścieżki otrzymane od ASN 8246 - GTS/IPARTNERS. Jak widać poniżej, są to tylko trzy trasy:

```
router-bgpd# show ip bgp regexp ^8246$
BGP table version is 0, local router ID is 195.149.118.1
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal
```

Origin codes: i - IGP, e - EGP, ? - incomplete

Network	Next Hop	Metric	LocPrf	Weight	Path
*> 195.94.192.0/19	157.25.1.16			0	8246 i
*> 217.8.160.0/19	157.25.1.16			0	8246 i
*> 217.153.0.0/16	157.25.1.16			0	8246 i

Total number of prefixes 3

BGP w sieci TCP/IP

Warto wiedzieć jak BGP działa w warstwie transportowej modelu OSI/ISO. Do komunikacji używany jest protokół TCP oraz port 179. Dzięki TCP protokół rutowania nie musi się martwić o utrzymywanie połączenia i sprawdzanie poprawności danych. Niektóre z protokołów, takie jak EIGRP, wykorzystują własne protokoły stworzone specjalnie do komunikacji, w tym przypadku RTP, niektóre nie robią tego w ogóle, jak RIP czy IGRP, i korzystają z protokołów bezpołączeniowych jak UDP.

Powyżej warstwy transportowej BGP używa własnych mechanizmów do zestawiania sesji i wymiany danych. BGP tworzy trwałe połączenia pomiędzy ruterami komunikującymi się bezpośrednio. Używanych jest kilka typów komunikatów do komunikowania się ruterów: OPEN, UPDATE, KEEPALIVE, NOTIFICATION. Potrzebne są one do zestawienia sesji, informowania rutera sąsiada o zmianach i podtrzymywania oraz zamykania sesji.

Atrybuty BGP

Wraz z informacją o prefiksach i ich długości przesyłana jest dodatkowa informacja w pakietach UPDATE - atrybuty. Pozwalają one procesowi BGP dokonywać wyboru optymalnej ścieżki. Za ich pomocą administrator może wpływać na działanie procesu BGP i filtrować wybrane ścieżki.

Najczęściej używane atrybuty to `as_path`, `next_hop`, `origin`, `communities`, `local_preference`, `MED`. Na podstawie ich wartości lub zmieniając ich wartość przy przekazywaniu informacji do sąsiadów możemy wpływać na zachowanie się procesu BGP i wybór ścieżek.

Atrybut `as_path`

Informacja o prefiksach przekazywana jest między routerami danych AS. Przechodząc przez nie, dodatkowo jest on oznaczana numerami ASN sieci, przez które przechodzi. Atrybut ten nazywany jest `as_path`. Przekazywanie informacji o ścieżce `as_path`, oprócz informacji samej w sobie, zapobiega występowaniu pętli przez wykluczenie sytuacji, w której ASN mógłby się powtórzyć w `as_path`.

```
router-bgpd# show ip bgp 161.12.12.0
BGP routing table entry for 161.12.12.0/22
Paths: (2 available, best #1, table Default-IP-Routing-Table)
Not advertised to any peer
 12968 3549 3356          <- as_path
    62.111.160.101 from 62.111.160.101 (213.134.144.1)
      Origin IGP, localpref 100, valid, external, best
      Last update: Tue Dec 30 17:24:48 2003

 8246 5588 1239 3356     <- as_path
    157.25.1.16 from 157.25.1.16 (157.25.1.16)
      Origin IGP, localpref 100, valid, external
      Community: 5588:1001 5588:3001 8246:667 8246:1080
      Last update: Tue Dec 23 07:24:15 2003
```

Jak widać trasa do adresów 161.12.12.0 jest dostępna przez dwie ścieżki as_path. Wybrana została pierwsza jako krótsza (na wybór ścieżek można wpływać na bardzo wiele sposobów).

Quagga

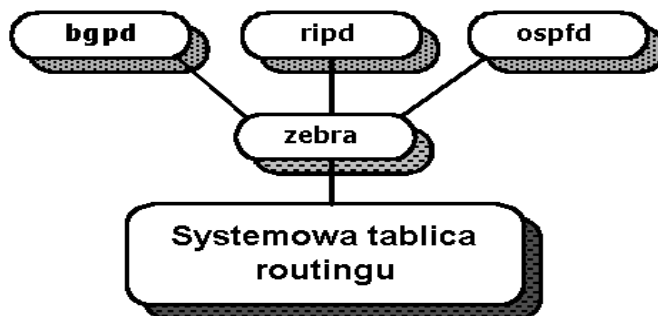
Quagga jest jednym z niewielu rozwiązań OpenSource pozwalających na stosowanie protokołów dynamicznego routingu w Linuxie. Jest to oprogramowanie bazujące na kodzie projektu Zebra (<http://www.zebra.org/>), której rozwój pozostawia ostatnio wiele do życzenia.

Poniższy opis, który pochodzi z dokumentacji, opisuje w krótki sposób czym jest Quagga:

"Quagga jest oprogramowaniem implementującym protokoły dynamicznego routingu dla protokołu TCP/IP takie jak RIPv1, RIPv2, RIPv6, OSPFv2, OSPFv3, BGP-4, i BGP-4+. Quagga posiada także zaimplementowany mechanizm Route Reflectorów i Route Serverów znany z BGP. Poza wsparciem dla protokołów routingu związanych z Ipv4, Quagga wspiera także protokoły związane z Ipv6."

Co jest ciekawe i wygodne dla ludzi znających interfejs Cisco to to, że autor Zebry, a obecnie także główny deweloper Quaggi, starali się go naśladować. Osoba, która używała interfejsu Cisco nie będzie miała kłopotu z konfigurowaniem Quaggi.

Quagga działa jako zespół demonów komunikujących się wzajemnie ze sobą. *zebra* odpowiada za komunikację z systemową tablicą routingu i komunikację z pozostałymi demonami *bgpd*, *ripd*, *ripngd*, *ospfd*, *ospf6d*. Te obsługują poszczególne protokoły i komunikują się jedynie z *zebrą* lub innymi routerami. Do poszczególnych demonów można się dostać poprzez telnet na port 2601 w przypadku *zebry* i 2605 w przypadku *bgpd*. Warto skorzystać z udogodnienia jakim jest vttysh, pozwalający na komunikację z wszystkimi działającymi demonami podczas jednego połączenia.



Rysunek 2. Schemat architektury Quaggi.

Oficjalnie wspierane platformy przez Quaggę to: GNU/Linux, FreeBSD, NetBSD, OpenBSD i Solaris, aczkolwiek Linux i FreeBSD wydają się być dominującymi platformami jeśli chodzi o wsparcie ze strony autorów.

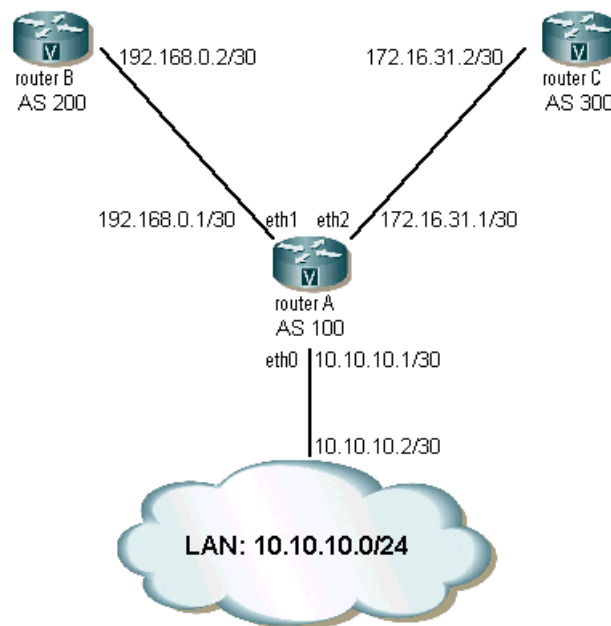
Przykładowa konfiguracja

Quagga stanowi w pełni funkcjonalny odpowiednik routera, który może być wykorzystywany na styku pomiędzy operatorami wykorzystującymi BGP. Poniżej przedstawiona zostanie konfiguracja Quaggi w sytuacji połączenia sieci z dwoma

dostawcami Internetu. Zakładamy, że połączenia są tej samej przepustowości i pozwalamy aby proces BGP sam decydował o wyborze najlepszej trasy.

Konfiguracja ta zapewnia nam zdublowanie połączeń przy pełnej niezależności adresowej (konieczne adresy PI). Z routera A wyprowadzane są dwa połączenia eBGP do AS200 i AS300. One zapewniają połączenie z resztą internetu i backup w przypadku awarii jednego z nich.

Jest to jedna z najczęstszych sytuacji, w których wykorzystanie BGP staje się konieczne.



Rysunek 3. Schemat sieci wykorzystanej w przykładowej konfiguracji.

Przy konfiguracji warto zwrócić uwagę na to aby nie rozgłaszać informacji otrzymywanych od AS200 do AS300 i odwrotnie, chyba że chcemy to robić celowo. Tak może być w przypadku gdy jesteśmy punktem tranzytowym. Route mapa o nazwie *localonly* nie pozwala na wysyłanie niczego co nie pochodzi z lokalnego AS. Wszystkie informacje wysyłane z AS100 posiadają pustą ścieżkę ASów, czyli atrybut *as_path* można dopasować za pomocą regexpa *^\$*.

Konfiguracja daemona *bgpd* odpowiadającego za komunikację z innymi routerami BGP:

```
1 hostname router-bgpd
2 password 8 xxxxxxxxxxxxxxxx
3 enable password 8 xxxxxxxxxxxxxxxx
4 log file /var/log/quagga.log
5 log trap informational
6 log record-priority
7 service password-encryption
8 !
9 router bgp 100
10 network 10.10.10.0/24
11 neighbor 192.168.0.1 remote-as 200
12 neighbor 192.168.0.1 description "Link do ISP 1"
13 neighbor 192.168.0.1 prefix-list 10 in
14 neighbor 192.168.0.1 route-map localonly out
```

```

15     neighbor 172.16.31.1 remote-as 300
16     neighbor 172.16.31.1 description "Link do ISP 2"
17     neighbor 172.16.31.1 prefix-list 10 in
18     neighbor 172.16.31.1 route-map localonly out
19     !
20     ip prefix-list 10 seq 10 deny 0.0.0.0/0
21     !
22     ip as-path access-list 50 permit ^$
23     !
24     route-map localonly permit 10
25         match as-path 50
26     !

```

Linie 1-7 są właściwie samoopisujące, definiujemy w nich: prompt wyświetlany po zalogowaniu się do terminala, hasła i opcje logowania. Polecenie w linii 7. powoduje, że hasła są przechowywane w postaci zaszyfrowanej.

Począwszy od linii 9. przechodzimy do konfiguracji procesu BGP. W linii 10. określamy informację jaką będziemy rozgłaszać do Internetu, czyli nasze adresy PI. Dzięki temu AS podłączone do nas będą otrzymywać informację o naszej sieci i propagować ją dalej do internetu.

Linie 11. i 15. opisują ASN naszego sąsiada.

W liniach 13. i 17. przy pomocy prefix listy 10. z linii 20. blokujemy przyjmowanie przez nasz proces BGP trasy domyślnej. Ponieważ otrzymujemy pełną informację o prefiksach dostępnych w internecie, nie potrzebujemy informacji o trasie domyślnej.

Linia 14. i 18. zapewnia, że my nie będziemy rozgłaszać do naszych sąsiadów informacji innej poza lokalną. Wykorzystywana jest w tym celu route mapa *localonly*, która dopasowuje atrybut *as_path* do listy 50. Zapobiega to tworzeniu sytuacji, w których moglibyśmy rozgłaszać z naszego AS informacje inne niż o naszej sieci.

Konfiguracja demona *zebra* odpowiadającego za komunikację z innymi demonami wchodzącymi w skład Quagga oraz kernelem:

```

1     hostname router-zebra
2     password 8 xxxxxxxxxxxxxxxx
3     enable password 8 xxxxxxxxxxxxxxxx
4     log file /var/log/quagga.log
5     log trap informational
6     log record-priority
7     service advanced-vty
8     service password-encryption
9     !
10    interface eth0
11        description "Siec wewnetrzna - AS100"
12        link-detect
13        ip address 10.0.0.1/24
14    !
15    interface eth1
16        description "ISP 1 - AS200"
17        link-detect
18        ip address 192.168.0.2/30
19    !
20    interface eth2
21        description "ISP 2 -AS300"
22        link-detect
23        ip address 172.16.31.2/30
24    !
25    interface lo
26        ip address 127.0.0.1/8
27    !

```

Jak widać powyżej, nie ma tu niczego specjalnego w konfiguracji. Jest to zwykła konfiguracja interfejsów. Warto zwrócić uwagę żeby nie dublować ich inicjowania. Jeśli

robimy to za pomocą zebry, pominiemy skrypty startowe inicjujące sieć.

Drugą ważną rzeczą jest to, aby proces zebry uruchamiany był przed pozostałymi daemonami wchodzącymi w skład Quaggi. W przeciwnym wypadku procesy takie jak bgpd nie będą przekazywać informacji do procesu zebry, a ten do kernela.

Jak widać, minimalna konfiguracja BGP za pomocą Quaggi w przypadku multihomingu jest wręcz banalna. Problemy zaczynają się kiedy chcemy mieć wpływ na poziom ruchu na poszczególnych interfejsach. Do tego typu kontroli ruchu przydają się właśnie wspomniane wcześniej atrybuty takie jak `as_path`, `local_preference` czy `communities`.

Kontrola działania

Zakładając, że mamy już skonfigurowane BGP pomiędzy routerami, warto znać kilka podstawowych poleceń, które będą niezbędne przy monitorowaniu jego działania.

Poniżej widać dwie sesje, z których krótsza działa od ponad czterech dni. Z obydwu sesji otrzymano ponad 128000 prefiksów, czyli tras do różnych sieci IP. Są to rzeczywiście działające sesje w konfiguracji podobnej do opisanej powyżej. Prompt "router-bgp#" jest promptem `bgpd`, natomiast "router-zebra#" jest promptem daemona `zebra`.

```
router-bgpd# show ip bgp summary
BGP router identifier 195.149.118.1, local AS number 29620
43688 BGP AS-PATH entries
322 BGP community entries

Neighbor      V    AS MsgRcvd MsgSent   TblVer   InQ  OutQ Up/Down   State/PfxRcd
62.111.160.101 4 12968 6575310 31966       0    0    0 4d17h27m 128107
157.25.1.16   4  8246 1226765 31877       0    0    0 01w0d15h 129167

Total number of neighbors 2
```

Na routerze, na którym działa Quagga zostały zestawione dwie sesje. Daemon Quaggi zajmujący się wprowadzaniem tras do tablicy routingu rozdzielił, dość nieproporcjonalnie jak widać, trasy pomiędzy dwóch dostawców.

```
gamma:~# ip route | grep 217.153.71.33 | wc -l
98072
gamma:~# ip route | grep 62.111.199.61 | wc -l
31419
gamma:~# ip route | wc -l
129492
```

Tu sprawdzamy jaka informacja jest przechowywana w procesie BGP o trasie 217.153.107.0. Jak widać poniżej, pierwsza została wybrana jako najlepsza (best #1), ponieważ ma krótszą ścieżkę AS – `as_path`.

```
router-bgpd# show ip bgp 217.153.107.0
BGP routing table entry for 217.153.0.0/16
Paths: (2 available, best #1, table Default-IP-Routing-Table)
Not advertised to any peer
8246
 157.25.1.16 from 157.25.1.16 (157.25.1.16)
   Origin IGP, localpref 100, valid, external, best
   Last update: Mon Mar 29 05:26:28 2004

12968 8246
```

```
62.111.160.101 from 62.111.160.101 (213.134.144.1)
Origin IGP, localpref 100, valid, external
Last update: Sun Mar 28 21:52:08 2004
```

```
router-zebra# show ip route 217.153.107.0
Routing entry for 217.153.0.0/16
Known via "bgp", distance 20, metric 0, best
Last update 2d16h25m ago
* 157.25.1.16 (recursive via 217.153.71.33)
```

W tablicy routingu została umieszczona jedynie trasa najlepsza.

Jeśli chcemy sprawdzić jakie trasy do sieci TPNET otrzymujemy od naszych peerów, wystarczy sprawdzić jakie informacje dostajemy o AS 5617.

```
router-bgpd# show ip bgp regexp 5617$
BGP table version is 0, local router ID is 195.149.118.1
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop          Metric LocPrf Weight Path
*> 63.167.185.0/24  62.111.160.101
*> 80.48.0.0/13     157.25.1.16      150      0 8246 5617 i
*                   62.111.160.101   0 12968 24748 5617 i
*> 83.0.0.0/11     157.25.1.16      150      0 8246 5617 i
*                   62.111.160.101   0 12968 24748 5617 i
```

I na koniec pokazana została pełna informacja o jednym z sąsiadów BGP naszego routera:

```
router-bgpd# show ip bgp neighbors 157.25.1.16
BGP neighbor is 157.25.1.16, remote AS 8246, local AS 29620, external link
Description: "Link do GTS"
BGP version 4, remote router ID 157.25.1.16
BGP state = Established, up for 01w0d15h
Last read 00:00:05, hold time is 180, keepalive interval is 60 seconds
Neighbor capabilities:
  Route refresh: advertised and received (old and new)
  Address family IPv4 Unicast: advertised and received
Received 1226920 messages, 2 notifications, 0 in queue
Sent 31874 messages, 10 notifications, 0 in queue
Route refresh request: received 0, sent 0
Minimum time between advertisement runs is 30 seconds
...
```

Quagga jak widać może być świetnym środkiem na rozpoczęcie korzystania z BGP bez wydawania przy okazji dużych sum pieniędzy. Z moich doświadczeń wynika, że jest to oprogramowanie stabilne i bez problemów współpracujące z urządzeniami firm komercyjnych, a przy tym znacznie tańsze.

To tylko krótki wstęp...

Jak wspomniałem na początku, jest to tylko krótki wstęp do bardzo ciekawego tematu jakim jest BGP. Brakuje w nim bardzo wielu rzeczy, po które odsyłam do RFC, a szczególnie polecam książkę Sama Halabiego, która stała się jednym z głównych podręczników administratorów używających BGP na codzień.

Jeśli nawet w danej chwili nie mamy okazji pracować z BGP, myślę, że warto poznać ten protokół. Każdy administrator sieci pracującej na styku z Internetem wcześniej czy później zetknie się z problemami związanymi pośrednio lub bezpośrednio z tym protokołem. Warto więc być przygotowanym;).

Materialy

"Internet Routing Architectures", Sam Halabi, Danny McPherson. ISBN: 157870233X.
RFC 1771 - A Border Gateway Protocol 4 (BGP-4): <http://www.ietf.org/rfc/rfc1771.txt>
Strona domowa projektu Quagga: <http://www.quagga.net/>