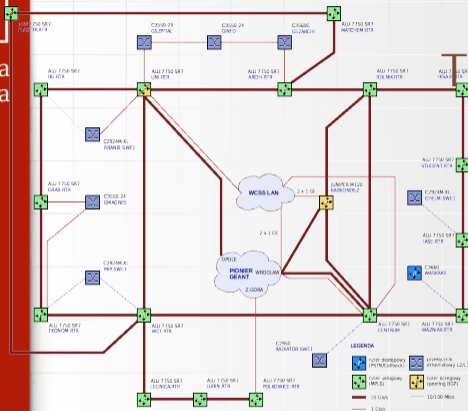




Politechnika
Wroclawska



TCP/IP: Adresy, trasowanie wer. 41 z drobnymi modyfikacjami!

Wojciech Myszka

2024-03-14 11:11:24 +0100



Część I

Internet Protocol v.4

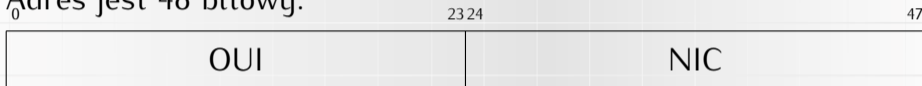


Sekcja 1

Adresy fizyczne

Adresy Ether

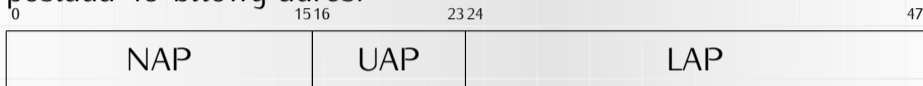
- ▶ Każda karta sieciowa w sieci Ethenrnet ma swój unikatowy (globalnie) adres MAC (*Medium Access Control address*)
- ▶ Adres jest 48 bitowy:



- ▶ OUI — Organizationally Unique Identifier (można go używać do **określenia produceta urządzenia**)
 - ▶ NIC — Network Interface Controller
- ▶ Adres zapisany jest w oprogramowaniu karty sieciowej.

Adresy Bluetooth

1. Podobnie w przypadku urządzeń Bluetooth — każde z nich posiada 48-bitowy adres:



2. OUI Ustalane przez producenta

- ▶ NAP — Not significant Address Part
- ▶ UAP — Upper Address Part
- ▶ LAP — Lower Address Part (nadawany przez producenta, jednoznacznie identyfikuje urządzenie)
- ▶ OUI — Organizationally Unique Identifier (można go używać do **określenia producenta urządzenia**)

Zapis adresów

Przyjęto się, że adresy zapisywane są szesnastkowo, poszczególne bajty (dwie cyfry) oddzielane są dwukropkiem:

ether 10:02:b5:a8:3c:d9 — karta bezprzewodowa

hci0 10:02:B5:A8:3C:DD — bluetooth

Wielkość liter nie ma znaczenia — tak jak w liczbach szesnastkowych!

Powyższe dla mojego laptopa. Jak widać producentem jest najprawdopodobniej jedna firma (Intel).



Sekcja 2

Adresy IPv4

Adresy IPv4 I

- ▶ 32 bity (4 miliardy węzłów): 4 294 967 296
- ▶ Adresy podzielone na kilka klas:
 - ▶ A: pierwszy bit adresu 0, 7 następnych identyfikuje sieć, ostatnie 24 — węzeł w sieci.
 - ▶ B: pierwsze dwa bity adresu 1 0, kolejnych 14 bitów identyfikuje sieć, a ostatnich 16 — węzeł w sieci.
 - ▶ C: pierwsze trzy bity adresu 1 1 0, 21 bitów to numer sieci, ostatnich osiem bitów identyfikuje węzeł w sieci.
 - ▶ D: pierwsze trzy bity adresu 1 1 1. Jest to specjalna, zarezerwowana klasa adresów.

Adresy IPv4 II

- ▶ Dla wygody adres dzielony jest na cztery bajty przedstawiane dziesiętnie.
Przykład: 156.17.8.1: 10011100 00010001 00001000 00000001
jest to adres z klasy B: sieć 156.17.
- ▶ Przyłączając się do Internetu wystąpiliśmy o pulę adresową i WCSS otrzymał do wyłącznej dyspozycji „grupę adresów klasy B” 156.17.0.0/16.
- ▶ Podumowując:
 - ▶ adresy klasy A pierwszy bajt mają mniejszy od 128,
 - ▶ klasa B to adresy z zakresu 128–191,
 - ▶ klasa C to adresy 192–223,
 - ▶ adresy większe od 223 to adresy zarezerwowane.

Adresy „prywatne”

- ▶ W każdej klasie adresowej zarezerwowaną grupę adresów do użytku „prywatnego”.
- ▶ Adresy takie nie są widoczne w światowym Internecie.
- ▶ Każdy może (dosyć dowolnie) z nich korzystać.
- ▶ Zarezerwowane adresy to:
 - ▶ W klasie A: sieć numer 10, czyli adresy z zakresu 10.0.0.1 – 10.255.255.254 (256^3 adresów),
 - ▶ W klasie B: sieć numer 172.16, czyli adresy z zakresu 172.16.0.1 – 172.31.255.254 (16×256^2 adresów),
 - ▶ W klasie C: sieć numer 192.168, czyli adresy z zakresu 192.168.0.0 – 192.168.255.254 (256^2 adresów).
- ▶ Dodatkowo adresy „link-local”: 169.254.0.0/16

Sieć i węzeł (host)

1. W adresach IPv4 pojawiło się pojęcie sieci i węzła.
2. Dostownie należy to tak rozumieć, że wszystkie węzły w sieci o tym samym numerze mają do siebie bezpośredni dostęp: **są w jednej fizycznej sieci.**
3. Weźmy (dla przykładu) adres 156.17.8.1.
 - ▶ pierwsze dwa bajty to numer sieci (156.17)
 - ▶ dwa następne bajty to numer węzła (8.1 czyli 00001000 00000001).
4. Zgodnie z modelem sieciowym do komunikacji w ramach tej samej sieci (tego samego medium) wystarczy warstwa fizyczna i transportowa.
5. Natomiast możliwości sprawnego zarządzania jedną siecią, która ma 2^{24} węzłów (klasa A) czy nawet tylko 2^{16} węzłów (klasa B) jest iluzoryczna.
6. Zachowano podział na „sieciową” i „węzłową” część adresu, ale sposób tego podziału pozostał w ostatecznej gestii użytkownika puli adresowej.

Maska

1. Z programowania (w języku C, ale nie tylko) powinna Państwu pozostać informacja, o bitowych operatorach logicznych $\&$ i $|$.
2. Operator AND może być używany do „wycinania” z wartości binarnych pól o zadanej długości i pozycji:

wartość	1	0	1	1	0	1	1	0
maska	0	0	0	1	1	1	0	0
<hr/>								
wynik	0	0	0	1	0	1	0	0
3. Maska złożona z jedynek „wycina” wartości, zera „przykrywają” je.



Maska

1. Z programowania (w języku C, ale nie tylko) powinna Państwu pozostać informacja, o bitowych operatorach logicznych $\&$ i $|$.
2. Operator AND może być używany do „wycinania” z wartości binarnych pól o zadanej długości i pozycji:

wartość	1	0	1	1	0	1	1	0
maska	0	0	0	1	1	1	0	0
<hr/>								
wynik	0	0	0	1	0	1	0	0

3. Maska złożona z jedynek „wycina” wartości, zera „przykrywają” je.

Maska sieciowa

Maska sieciowa

to informacja pozwalająca wyodrębnić z każdego adresu IP informację o numerze sieci (podsieci) i numerze węzła.

1. Podaje się ją jak „adres IP” jedyнкami wskazując ciągły obszar zarezerwowany na numer sieci (podsieci) albo
2. Jako liczbę bitów (licząc od najbardziej znaczącego czyli o lewej strony) przeznaczonych na numer sieci.

Przykład:

255.255.255.224

255.255.255.11100000

x.x.x.x/27

pierwszych 27 bitów to adres sieci



Adres sieci, adres emisji

1. Przyjęto uznawać adres ze wszystkimi zerami w polu hosta uważać za **numer podsieci** natomiast
2. Adres ze wszystkimi jedynekami w polu hosta nazywać **adresem rozgłoszeniowym** (*broadcast*) lub **adresem emisji**.



Maska sieci

Przykład

1. Załóżmy, że komputer o adresie 156.17.8.1 chce się skontaktować z komputerem o adresie 156.17.5.2

Maska sieci

Przykład

1. Załóżmy, że komputer o adresie 156.17.8.1 chce się skontaktować z komputerem o adresie 156.17.5.2
2. Skoro oba adresy należą do tej samej klasy B — nie powinno być właściwie żadnego problemu

Maska sieci

Przykład

1. Załóżmy, że komputer o adresie 156.17.8.1 chce się skontaktować z komputerem o adresie 156.17.5.2
2. Skoro oba adresy należą do tej samej klasy B — nie powinno być właściwie żadnego problemu
3. Administratorzy zdecydowali jednak o wewnętrznym podziale na podsieci.

Maska sieci

Przykład

1. Załóżmy, że komputer o adresie 156.17.8.1 chce się skontaktować z komputerem o adresie 156.17.5.2
2. Skoro oba adresy należą do tej samej klasy B — nie powinno być właściwie żadnego problemu
3. Administratorzy zdecydowali jednak o wewnętrznym podziale na podsieci.
4. W podsieci nadawcy zastosowana jest maska 255.255.255.224

Maska sieci

Przykład

1. Założmy, że komputer o adresie 156.17.8.1 chce się skontaktować z komputerem o adresie 156.17.5.2
2. Skoro oba adresy należą do tej samej klasy B — nie powinno być właściwie żadnego problemu
3. Administratorzy zdecydowali jednak o wewnętrznym podziale na podsieci.
4. W podsieci nadawcy zastosowana jest maska 255.255.255.224
5. Maska nakładana jest na adres nadawcy i na adres odbiorcy

156	17	8	00000001	156	17	5	00000010
255	255	255	11100000	255	255	255	11100000
<hr/>				<hr/>			
156	17	8	0	156	17	5	0

Maska sieci

Przykład

1. Założmy, że komputer o adresie 156.17.8.1 chce się skontaktować z komputerem o adresie 156.17.5.2
2. Skoro oba adresy należą do tej samej klasy B — nie powinno być właściwie żadnego problemu
3. Administratorzy zdecydowali jednak o wewnętrznym podziale na podsieci.

4. W podsieci nadawcy zastosowana jest maska 255.255.255.224

5. Maską nakładaną jest na adres nadawcy i na adres odbiorcy

156	17	8	00000001	156	17	5	00000010
255	255	255	11100000	255	255	255	11100000

156	17	8	0	156	17	5	0
-----	----	---	---	-----	----	---	---

6. Numery podsieci różnią się — nie można przeprowadzić komunikacji bezpośredniej; trzeba skorzystać z bramy.

Maska sieciowa

Więcej przykładów

1. W ramach sieci 156.17.8.0/27 mamy podsieci o następujących adresach:

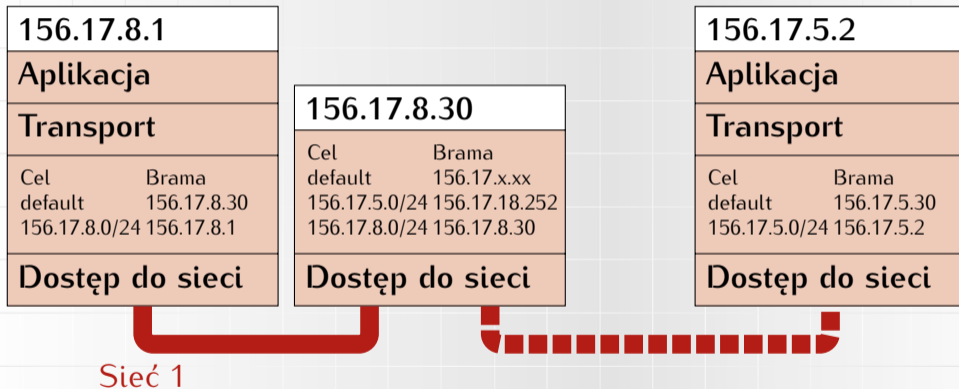
- | | |
|--------------------------|---------------------------|
| ▶ 156.17.8.00000000 (0) | ▶ 156.17.8.10000000 (128) |
| ▶ 156.17.8.00100000 (32) | ▶ 156.17.8.10100000 (160) |
| ▶ 156.17.8.01000000 (64) | ▶ 156.17.8.11000000 (192) |
| ▶ 156.17.8.01100000 (96) | ▶ 156.17.8.11100000 (224) |

2. Była to jedna z najpopularniejszych masek sieciowych (w czasach „cienkiego” ethernetu — miał on ograniczenie na liczbę węzłów w jednym segmencie równe 30)...

3. Węzeł o numerze zero zarezerwowany jest jako numer sieci, a węzeł o numerze IP złożonym z samych jedynek — to adres rozgłoszeniowy (czyli „do wszystkich”).

Trasowanie

Routing





Unix: ifconfig I

```
myszka@asusux:~$ ifconfig
```

```
enx9cebe8060394: flags=4099<UP,BROADCAST,MULTICAST> mtu 1500
  ether 9c:eb:e8:06:03:94 txqueuelen 1000 (Ethernet)
  RX packets 0 bytes 0 (0.0 B)
  RX errors 0 dropped 0 overruns 0 frame 0
  TX packets 0 bytes 0 (0.0 B)
  TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

```
lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
  inet 127.0.0.1 netmask 255.0.0.0
  inet6 ::1 prefixlen 128 scopeid 0x10<host>
  loop txqueuelen 1000 (Local Loopback)
  RX packets 24569 bytes 10372067 (10.3 MB)
  RX errors 0 dropped 0 overruns 0 frame 0
  TX packets 24569 bytes 10372067 (10.3 MB)
  TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```




Unix: ifconfig II

```
wlp1s0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
  inet 192.168.1.174 netmask 255.255.255.0 broadcast 192.168.1.255
  inet6 fe80::e565:6231:d639:2fba prefixlen 64 scopeid 0x20<link>
  ether 10:02:b5:a8:c3:d9 txqueuelen 1000 (Ethernet)
  RX packets 404793 bytes 581882348 (581.8 MB)
  RX errors 0 dropped 0 overruns 0 frame 0
  TX packets 126791 bytes 17418958 (17.4 MB)
  TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```



Unix: ip l

```
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN
    group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: wlp1s0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP
    group default qlen 1000
    link/ether 10:02:b5:a8:c3:d9 brd ff:ff:ff:ff:ff:ff
    inet 192.168.1.174/24 brd 192.168.1.255 scope global dynamic wlp1s0
        valid_lft 3401sec preferred_lft 3401sec
    inet6 fe80::e565:6231:d639:2fba/64 scope link
        valid_lft forever preferred_lft forever
3: enx9cebe8060394: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc
    pfifo_fast state DOWN group default qlen 1000
    link/ether 9c:eb:e8:06:03:94 brd ff:ff:ff:ff:ff:ff
```

Windows: ipconfig /all

```
ipconfig /all
```

Karta Ethernet Połączenie sieci bezprzewodowej:

```
Sufiks DNS konkretnego połączenia : chello.pl
Opis . . . . . : 802.11n Wireless LAN Card
Adres fizyczny. . . . . : 00-15-AF-DC-5F-5B
DHCP włączone . . . . . : Tak
Autokonfiguracja włączona . . . . : Tak
Adres IP. . . . . : 192.168.1.199
Maska podsieci. . . . . : 255.255.255.0
Brama domyślna. . . . . : 192.168.1.1
Serwer DHCP . . . . . : 192.168.1.1
Serwery DNS . . . . . : 62.179.1.61 62.179.1.63
Dzierżawa uzyskana. . . . . : 6 kwietnia 2018 09:17:26
Dzierżawa wygasa. . . . . : 6 kwietnia 2018 10:17:26
```



Address Resolution Protocol

1. Gdy już wiadomo czy adres docelowy znajduje się w sieci lokalnej czy zdalnej...
2. Zdecydować trzeba do kogo przestać pakiet:
 - 2.1 bezpośrednio do odbiorcy (adres w sieci lokalnej),
 - 2.2 do bramy sieciowej (adres w sieci zdalnej).
3. Warstwa transportowa korzysta z adresów fizycznych (o których była mowa wcześniej).
4. Potrzebny jest zatem mechanizm (działający wyłącznie wewnątrz sieci lokalnej) łączący adresy fizyczne z internetowymi.
5. Służy do tego protokół Address Resolution Protocol (ARP).
 - ▶ sprawdza sięczy w pamięci pomocniczej jest wpis wiążący adres IP z adresem fizycznym;
 - ▶ jeżeli nie ma — wysyłany jest specjalny pakiet **do wszystkich** (zawiera on adres rozgłoszeniowy MAC ff:ff:ff:ff:ff:ff) z adresem IP;
 - ▶ na wezwanie odpowiada wyłącznie węzeł o szukanym adresie IP;
 - ▶ po pewnym czasie informacja ARP się przeterminowuje.

ARP

Specjalny protokół służy do zdobywania informacji o adresach ARP powiązanych z adresami IP (w sieci lokalnej)

Aby prześledzić te pakiety użyłem polecenia

```
tcpdump -ennqti eno1 \( arp or icmp \)
```

Oto fragment z wymnianej pakietów:

```
00:1e:8f:38:41:f8 > ff:ff:ff:ff:ff:ff, ARP, length 60:  
Request who-has 156.17.6.200 tell 156.17.6.175, length 46  
a4:4c:c8:4a:6e:f6 > 00:1e:8f:38:41:f8, ARP, length 60:  
Reply 156.17.6.200 is-at a4:4c:c8:4a:6e:f6, length 46
```

Pierwsza linia to zapytanie wysłane z komputera o adresie MAC 00:1e:8f:38:41:f8 do wszystkich (adres MAC z samych jedynek) druga to odpowiedź komputera skierowana bezpośrednio do zainteresowanego.



ARP

```
arp -a
gateway (192.168.1.1) w 44:6a:b7:f7:31:38 [ether] na wlp1s0
? (192.168.1.176) w 3c:77:e6:88:6d:88 [ether] na wlp1s0
? (192.168.1.53) w c0:ee:fb:42:9e:2f [ether] na wlp1s0
? (192.168.1.2) w 38:d5:47:82:03:d4 [ether] na wlp1s0
```

```
ping 192.168.1.14
PING 192.168.1.14 (192.168.1.14) 56(84) bytes of data.
64 bytes from 192.168.1.14: icmp_seq=1 ttl=64 time=265 ms
```

```
arp -a
gateway (192.168.1.1) w 44:6a:b7:f7:31:38 [ether] na wlp1s0
? (192.168.1.14) w b8:27:eb:0f:99:e4 [ether] na wlp1s0
? (192.168.1.176) w 3c:77:e6:88:6d:88 [ether] na wlp1s0
? (192.168.1.53) w c0:ee:fb:42:9e:2f [ether] na wlp1s0
? (192.168.1.2) w 38:d5:47:82:03:d4 [ether] na wlp1s0
```

Trasowanie

- ▶ Osobną kwestią jest wybór bramy, którą należy wybrać jako pośrednika w ruchu do innych sieci.
- ▶ Bardzo często sytuacja jest bardzo prosta: z podsieci jest tylko jedno wyjście „w świat”.
- ▶ W takich przypadkach w konfiguracji sieciowej wystarczy zdefiniowanie **domyślnej bramy**.
- ▶ Gdy sytuacja jest bardziej skomplikowana — oprogramowanie musi zdobywać i przechowywać informacje o bramach/interfejsach używanych do kontaktów z innymi sieciami/węzłami.
- ▶ W przypadku gdy komputer ma kilka interfejsów sieciowych odpowiednie informacje zostaną wygenerowane automatycznie, ale informacje mogą być uaktualniane (protokół ICMP).



ip route

```
ip route
```

```
default via 192.168.1.1 dev wlp1s0 proto static metric 600
```

```
169.254.0.0/16 dev wlp1s0 scope link metric 1000
```

```
192.168.1.0/24 dev wlp1s0 proto kernel scope link src 192.168.1.174 metric 600
```

lub bardziej skomplikowane

```
default via 192.168.0.1 dev enx9cebe80444ef proto dhcp metric 100
```

```
default via 192.168.15.210 dev usb0 proto dhcp metric 101
```

```
default via 192.168.0.1 dev wlp1s0 proto dhcp metric 600
```

```
169.254.0.0/16 dev wlp1s0 scope link metric 1000
```

```
192.168.0.0/24 dev enx9cebe80444ef proto kernel scope link src 192.168.0.18 metric 600
```

```
192.168.0.0/24 dev wlp1s0 proto kernel scope link src 192.168.0.227 metric 600
```

```
192.168.15.0/24 dev usb0 proto kernel scope link src 192.168.15.214 metric 100
```




netstat -rn

```
netstat -rn
```

```
Kernel IP routing table
```

Destination	Gateway	Genmask	Flags	MSS	Window	irrtt	Iface
0.0.0.0	192.168.1.1	0.0.0.0	UG	0	0	0	wlp1s0
169.254.0.0	0.0.0.0	255.255.0.0	U~	0	0	0	wlp1s0
192.168.1.0	0.0.0.0	255.255.255.0	U~	0	0	0	wlp1s0



Tablica routingu + VPN

```
ip route
```

```
default via 10.8.0.5 dev tun0 proto static metric 50
```

```
default via 192.168.0.1 dev enx9cebe80444ef proto dhcp metric 100
```

```
default via 192.168.15.210 dev usb0 proto dhcp metric 101
```

```
default via 192.168.0.1 dev wlp1s0 proto dhcp metric 600
```

```
10.8.0.1 via 10.8.0.5 dev tun0 proto static metric 50
```

```
10.8.0.5 dev tun0 proto kernel scope link src 10.8.0.6 metric 50
```

```
156.17.8.5 via 192.168.0.1 dev enx9cebe80444ef proto static metric 100
```

```
169.254.0.0/16 dev wlp1s0 scope link metric 1000
```

```
192.168.0.0/24 dev enx9cebe80444ef proto kernel scope link src 192.168.0.18 metric 600
```

```
192.168.0.0/24 dev wlp1s0 proto kernel scope link src 192.168.0.227 metric 600
```

```
192.168.0.1 dev enx9cebe80444ef proto static scope link metric 100
```

```
192.168.15.0/24 dev usb0 proto kernel scope link src 192.168.15.214 metric 100
```



Tablica routingu w sieci VPN (do ćwiczeń)

ip address show dev tun0

```
6: tun0: <POINTOPOINT,MULTICAST,NOARP,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast
    link/none
    inet 10.8.0.10 peer 10.8.0.9/32 scope global tun0
        valid_lft forever preferred_lft forever
    inet6 fe80::5dd3:fbb0:6342:33b8/64 scope link flags 800
        valid_lft forever preferred_lft forever
```

ip route

```
0.0.0.0/1 via 10.8.0.9 dev tun0
default via 192.168.1.1 dev wlp1s0 proto static metric 600
10.8.0.1 via 10.8.0.9 dev tun0
10.8.0.9 dev tun0 proto kernel scope link src 10.8.0.10
128.0.0.0/1 via 10.8.0.9 dev tun0
156.17.8.21 via 192.168.1.1 dev wlp1s0
169.254.0.0/16 dev wlp1s0 scope link metric 1000
192.168.1.0/24 dev wlp1s0 proto kernel scope link src 192.168.1.174 metric 600
```

Protokoły trasowania

- ▶ Trasowanie w globalnym Internecie opiera się na koncepcji **Systemów Autonomicznych**.
- ▶ System Autonomiczny to fragment globalnej sieci Internet zarządzany przez jedną organizację.
- ▶ W ramach systemu autonomicznego używany jest jakiś protokół z grupy *Interior Gateway Protocols*.
- ▶ Na zewnątrz używane są protokoły z grupy *Exterior Gateway Protocols*.