Wojciech Myszka

TCP/IP: DNS, protokołąy, gniazda, IP v.6

wer. 31 z drobnymi modyfikacjami!

2021-04-13 12:13:57 +0200

Spis treści

Część I

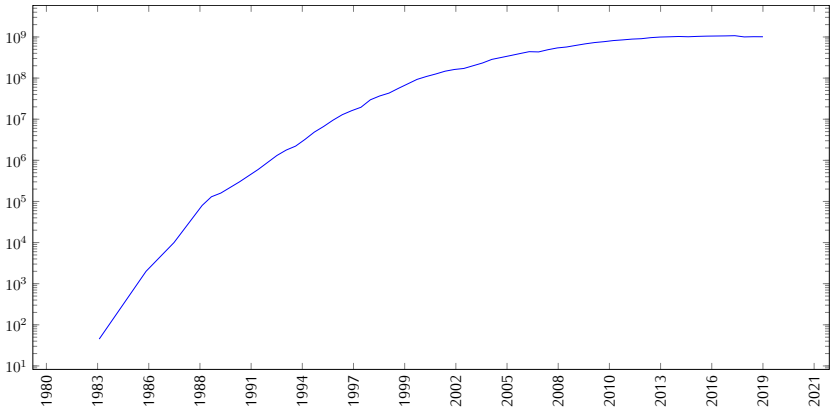
DNS

1. System Nazw Domenowych

Skoro wiemy już jak odbywa się komunikacja między komputerami, jak adresy IP używane są do wyboru sposobu komunikacji: bezpośrednio, w ramach tej samej sieci czy za pośrednictwem bramy (gdy węzeł docelowy jest w innej sieci) pozostaje kwestia „tłumaczenia” nazw symbolicznych używanych powszechnie (`www.gogle.com`, `onet.pl`,...) na adresy IP.

Jeżeli odwołać się do historii to pierwotnie w tym celu używano pliku `/etc/hosts`. Ale w tamtych czasach liczba węzłów sieci była znacznie, znacznie mniejsza.

Rysunek 1 przedstawia wzrost liczby węzłów sieci Internet w czasie. Dane z Internet System Consortium są dosyć wiarygodne, ale obejmują



Rysunek 1. Internet Domain Survey Host Count

lata 1993–2017. Dodałem do tego mniej już wiarygodne dane na temat czasów wcześniejszych (na osi Y jest skala logarytmiczna).

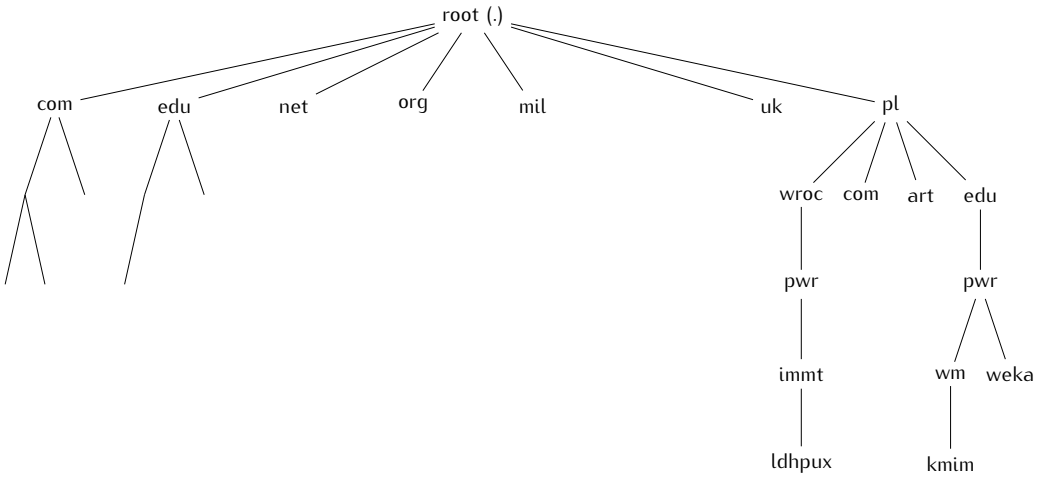
Podstawowy program do obsługi DNS powstał na początku lat 80. Nazywa się (do dziś) BIND — *Berkeley Internet Name Domain*.

Domain Name System

1. *System Nazw Domenowych* (częściej Domain Name System, albo DNS) to rozproszona baza danych służąca do zarządzania konwersją adresów IP na (bardziej) czytelne dla ludzi nazwy symboliczne.
2. Jest to całkiem złożony system informatyczny (i bardzo złożony system prawny). Złożoność systemu prawnego związana jest z prawami własności do nazw domenowych, które często powiązane są markami czy nazwami własnymi firm. Dochodziło do tego, że firma Apple zgłaszałą pretensje do nazwy domenowej **ap.pl** — fakt, że można ją wymawiać jako „jabłko" (po angielsku).
   — z jednej strony wykorzystywany jest do nadawania „osobowości internetowej" nowym podmiotom pojawiającym się w sieci,
   — z drugiej — nieodzowny fragment infrastruktury internetu zapewniający realizację usług różnych protokołów najwyższej warstwy.
3. Nazwy domenowe tworzą strukturę drzewiastą:
   — korzeniem drzewa jest domena główna *root* oznaczana jako . (kropka)
   — kolejne człony (o długości do 63 znaków każdy) nazw oddzielane są kropkami,
   — domena to poddrzewo hierarchii obejmujące szereg pod-domen o wspólnym przyrostku: **com.pl**, **edu.pl**, **wroc.pl**, **net.pl**, **art.pl**,...
   — nazwy domen mogą zawierć znaki, cyfry i znak minus (kiedyś sugerowano, że nazwa domeny powinna zaczynać się od litery, ale okazało się, że sytuacja, gdy zaczyna się od cyfry nie stanowi żadnego problemu, tym bardziej, że takiej nazwy domagała się firma 3com założona 1979 przez Roberta Metcalfe'a, jednego z „ojców" Internetu),
   — od pewnego czasu można używać nie tylko znaków ASCII, ale również znaków Unicode, ale z różnych, praktycznych względów adresy

takie są rzadko stosowane. Choć korzysta z tego bardzo chętnie Wikipedia i adres https://pl.wikipedia.org/wiki/Przełącznik_sieciowy tłumaczony jest na https://pl.wikipedia.org/wiki/Prze%C5%82%C4%85cznik_sieciowy. Jeszcze śmieszniej to wygląda w przypadku języka chińskiego czy japońskiego...: https://zh.wikipedia.org/wiki/%E9%80%9A%E8%A8%8A%E5%9F%A0.
Tekst zapisywany jest w kodowaniu UTF-8 gdzie każdy znak spoza zakresu ASCII kodowany jest jako sekwencja od dwu do sześciu bajtów, a każdy bajt przedstawiany jest jako dwie cyfry szesnastkowe poprzedzone znakiem procent.

DNS



Rysunek 2. Uproszczony schemat fragmentu drzewa DNS

Serwery DNS

— Z każdą nazwą domenową związany jest serwer przechowujący wszystkie informacje o domenie:
  — adres e-mail osoby odpowiedzialnej,
  — adresy serwerów zapasowych, (serwery zapasowe dostarczają informacji w przypadku awarii — lub braku dostępu do — serwera głównego),
  — okres ważności informacji oraz okres odświeżania informacji przez serwery zapasowe,
  — numer seryjny domeny.

' Program dig jest jednym z podstawowych narzędzi technicznych pozwalających badać stukturę nazw domenowych. Służy on do zadawania elementarnych pytań serwerom DNS. Najporstsze z nich to pytanie o rekord SOA (Start of Authority). Tylko serwer, który jest autorytatywny (czyli posiada informacje o odpytywanej nazwie domenowej) dla domeny potrafi na te pytanie odpowiedzieć.

Informacje tam zawarte zostały opisane na przykładzie domeny głównej.

Program dig to narzędzie służące do komunikowania się z serwerami DNS. Obsługa protokołu DNS odbywa się zazwyczaj z użyciem UDP, chociaż, może też być prowadzona z użyciem TCP.

dig soa .
Odpowiedź serwera DNS:

. 6900 IN SOA a.root-servers.net. nstld.verisign-grs.com. 2021041300 1800 900 604800 86400

. — nazwa domeny (kropka: domena główna)

6900 — czas ważności **tego** rekordu

IN SOA — oznaczenie rekordu Start of Authority

a.root-servers.net. — adres serwera głównego (kropka na końcu oznacza pełny adres, bez kropki może być on jeszcze rozszerzany)

nstld.verisign-grs.com. — adres odpowiedzialnego (pierwszą kropkę zamieniamy na znak @, ostatnią usuwamy)

2021041300 — numer seryjny rekordu

1800 — okres odświeżania

900 — okres powtarzania (gdy nie uda się odświeżyć)

604800 — okres ważności (w sekundach, tydzień)

86400 — minimalny okres ważności rekordu (gdy nie podano inaczej, w tym wypadku doba)

Zapytanie o serwery odpowiedzialne dla domeny.

dig ns .

```
. 83939 IN NS a.root-servers.net.
. 83939 IN NS b.root-servers.net.
. 83939 IN NS c.root-servers.net.
. 83939 IN NS d.root-servers.net.
. 83939 IN NS e.root-servers.net.
. 83939 IN NS f.root-servers.net.
. 83939 IN NS g.root-servers.net.
. 83939 IN NS h.root-servers.net.
. 83939 IN NS i.root-servers.net.
. 83939 IN NS j.root-servers.net.
. 83939 IN NS k.root-servers.net.
. 83939 IN NS l.root-servers.net.
. 83939 IN NS m.root-servers.net.
```

Liczba występująca po nazwie domeny oznacza pozostały okres ważności tej informacji, po przekroczeniu jego — informacja jest kasowana z pamięci podręcznej.

dig ns immt.pwr.wroc.pl

```
immt.pwr.wroc.pl. 86400 IN NS sun2.pwr.wroc.pl.
immt.pwr.wroc.pl. 86400 IN NS ldhpux.immt.pwr.wroc.pl.
immt.pwr.wroc.pl. 86400 IN NS dns1.pwr.wroc.pl.
immt.pwr.wroc.pl. 86400 IN NS dns2.pwr.wroc.pl.
immt.pwr.wroc.pl. 86400 IN NS temisto.immt.pwr.wroc.pl.
```

Zazwyczaj stacje końcowe nie zadają zapytań DNS serwerom autorytatywnym (posiadającym informacje z „pierwszej ręki") tylko serwerom dostawców usług. Te zaś przechowują raz zdobyte informacje w pamięci podręcznej, żeby nie generować zbyt wielkiego ruchu. Informacje w tej pamięci ulegają przeterminowaniu.

Rozwiązywanie adresów

1. Baza danych ma strukturę hierarchiczną
2. Wystarczy znać adres któregokolwiek serwera root, żeby zacząć rozwiązywać adresy.
   Załóżmy, że interesuje mnie numeryczna wartość adresu `ldhpux.immt.pwr.wroc.pl`. Pytam o to serwera głównego.

   dig @l.root-servers.net. ldhpux.immt.pwr.wroc.pl

   ```
   ;; AUTHORITY SECTION:
   pl. 172800 IN NS a-dns.pl.
   pl. 172800 IN NS b-dns.pl.
   pl. 172800 IN NS c-dns.pl.
   ...
   ```

   Serwer główny nigdy nie udziela tak szczegółowych informacji. Podaje natomiast informacje o adresach serwerów „pod-domeny"(w tym wypadku **.pl**).

   dig @a-dns.pl. ldhpux.immt.pwr.wroc.pl

   ```
   ;; AUTHORITY SECTION:
   wroc.pl. 86400 IN NS bilbo.nask.org.pl.
   wroc.pl. 86400 IN NS wask.wask.wroc.pl.
   wroc.pl. 86400 IN NS ldhpux.immt.pwr.wroc.pl.
   wroc.pl. 86400 IN NS ns1.net.icm.edu.pl.
   wroc.pl. 86400 IN NS sun2.pwr.wroc.pl.
   wroc.pl. 86400 IN NS kirdan.nask.net.pl.
   ```

   dig @bilbo.nask.org.pl. ldhpux.immt.pwr.wroc.pl

   ```
   ;; AUTHORITY SECTION:
   pwr.wroc.pl. 10800 IN NS sun2.pwr.wroc.pl.
   pwr.wroc.pl. 10800 IN NS dns2.pwr.wroc.pl.
   pwr.wroc.pl. 10800 IN NS dns.pwr.wroc.pl.
   pwr.wroc.pl. 10800 IN NS wask.wask.wroc.pl.
   pwr.wroc.pl. 10800 IN NS ns2.net.icm.edu.pl.
   pwr.wroc.pl. 10800 IN NS ns1.net.icm.edu.pl.
   ```

   dig @ns2.net.icm.edu.pl. ldhpux.immt.pwr.wroc.pl

```
;; AUTHORITY SECTION:
immt.pwr.wroc.pl. 86400 IN NS sun2.pwr.wroc.pl.
immt.pwr.wroc.pl. 86400 IN NS kufel.immt.pwr.wroc.pl.
immt.pwr.wroc.pl. 86400 IN NS ldhpux.immt.pwr.wroc.pl.
immt.pwr.wroc.pl. 86400 IN NS dns2.pwr.wroc.pl.
```

dig @sun2.pwr.wroc.pl. ldhpux.immt.pwr.wroc.pl

```
;; ANSWER SECTION:
ldhpux.immt.pwr.wroc.pl. 259200 IN A 156.17.8.1
```

W ten sposób, zapytanie przekazywane jest coraz niżej, aż do serwera, który potrafi udzielić autorytatywnej odpowiedzi.

3. Proces przeprowadziłem ręcznie, ale normalnie odbywa się on automatycznie.
4. Każdy komputer (na ogół) ma pamięć podręczna, w której przechowuje „zdobyte" dotychczas informacje. Przyśpiesza to proces rozwiązywania adresów.
   Z drugiej strony jeżeli jakaś informacja jest w pamięci podręcznej (i jest błędna)...
5. Przed opracowaniem systemu rozproszonej bazy danych używano pliku. Nazywa się on `hosts` i znajduje się w kartotece `/etc/`.
6. Jest on obecny również w systemie Windows: `C:\WINDOWS\system32\drivers\etc\hosts`.

2. Adresy „odwrotne"

— Wcześniej opisałem sposób translacji adresu symbolicznego na adres numeryczny.
— Istnieje równiez procedura odwrotna — translacji adresu numerycznego na symboliczny,
— W tym celu utworzno „sztuczną" domenę `in-addr.arpa`.
— Jej poddomeny to kolejne części numeryczne adresu IP w kolejnosci **odwrotnej**.
— Chcąc uzyskać nazwę internetową węzłą o adresie numerycznym 156.17.8.1 odpytujemy o `1.8.17.156.in-addr.arpa`.

```
nslookup 156.17.8.1
1.8.17.156.in-addr.arpa name = ldhpux.immt.pwr.wroc.pl.
```

Albo inaczej

```
dig -x 156.17.8.1
;; ANSWER SECTION:
1.8.17.156.in-addr.arpa. 86400 IN PTR ldhpux.immt.pwr.wroc.pl.
```

Adres 127.0.0.0/8 to adres „localhost" zapewniający funkcjonowanie oprogramowania sieciowego bez fizycznej karty sieciowej, w ramach jednego komputera. W wielu przypadkach ułatwia to pisanie aplikacji „uni-

wersalnych", mogących działać również w środowisku „bez dostępu do (rzeczywistej) sieci".

3. Protokoły, porty i gniazdka

Port w języku polskim ma bardzo wiele znaczeń, ale większość z nich kojarzy się z takim miejscem, które służy do komunikacji (niezależnie od środka komunikacji). Ja będę używał tego określenia w kontekście sieciowym.

Porty

— Jednym z bardzo ważnych zadań stosu sieciowego jest zapewnienie dostarczania przekazywanych przez sieć informacji do właściwych aplikacji.
— Jest tylko jedno[1] gniazdko sieciowe oraz wiele aplikacji i wielu użytkowników z niego korzystających.
— Zapewniają to dodatkowe informacje przesyłane z każdym pakietem. Są to:
  — deklaracja protokołu (TCP, UDP,...),
  — gniazdo (*socket*),
  — port,
  — adresy IP (źródłowy i docelowy).

Kilka definicji (1)

Definicja 1 (Port protokołu)**.** Port (protokołu) to szesnastobitowa liczba całkowita bez znaku używana w komunikacji sieciowej do definicji procesu z niego korzystającego. Porty dzielą się na:
  — ogólnie znane (*well known*) (liczby z zakresu od 0 do 1023), oraz
  — dynamiczne (przydzielane w miarę potrzeby).
Port jest jednym z atrybutów gniazda.

Porty ogólnie znane...

...to porty, pod którymi nasłuchują najważniejsze usługi sieci Internet:
80 serwery WWW (protokół http),
443 serwery WWW (protokół https),
25 serwery SMTP (poczty elektronicznej)
20, 21 ftp
53 Domain Name System
631 drukarki (IPP)
143, 993 IMAP (poczta elektroniczna — klient)
Plik `/etc/services` zawiera większość definicji portów i związanych z nimi usług.

Wiele z portów ma identyczne znaczenie dla protokołu TCP i UDP (usługi mogą być dostępne w obu protokołach).

[1] Mnóstwo zastrzeżeń tu!

Przydziałem numerów portów zajmuje się IANA (http://www.iana.org/assignments/port-numbers)

Kilka definicji (2)

Definicja 2 (Gniazdo). **Gniazdo** to abstrakcyjny dwukierunkowy punkt końcowy połączenia. Dwukierunkowość oznacza, że można dane odbierać i wysyłać. Podstawowe atrybuty gniazda:

- typ gniazda (protokół przesyłu informacji),
- lokalny adres (na przykład IP, Ethernet,...),
- opcjonalnie lokalny numer portu definiujący proces wymieniający dane przez gniazdo.

Dodatkowo może to być:

- zdalny adres,
- opcjonalnie zdalny numer portu definiujący zdalny proces z niego korzystający.

Plik `/etc/protocols` zawiera zakodowane numerycznie nazwy protokołów:

```
# Internet (IP) protocols
#
# Updated from http://www.iana.org/assignments/protocol-numbers
# and other sources.

ip      0       IP              # internet protocol, pseudo protocol number
hopopt  0       HOPOPT          # IPv6 Hop-by-Hop Option [RFC1883]
icmp    1       ICMP            # internet control message protocol
igmp    2       IGMP            # Internet Group Management
ggp     3       GGP             # gateway-gateway protocol
ipencap 4       IP-ENCAP        # IP encapsulated in IP (officially ``IP'')
st      5       ST              # ST datagram mode
tcp     6       TCP             # transmission control protocol
egp     8       EGP             # exterior gateway protocol
igp     9       IGP             # any private interior gateway (Cisco)
pup     12      PUP             # PARC universal packet protocol
udp     17      UDP             # user datagram protocol
hmp     20      HMP             # host monitoring protocol
```

Przykład

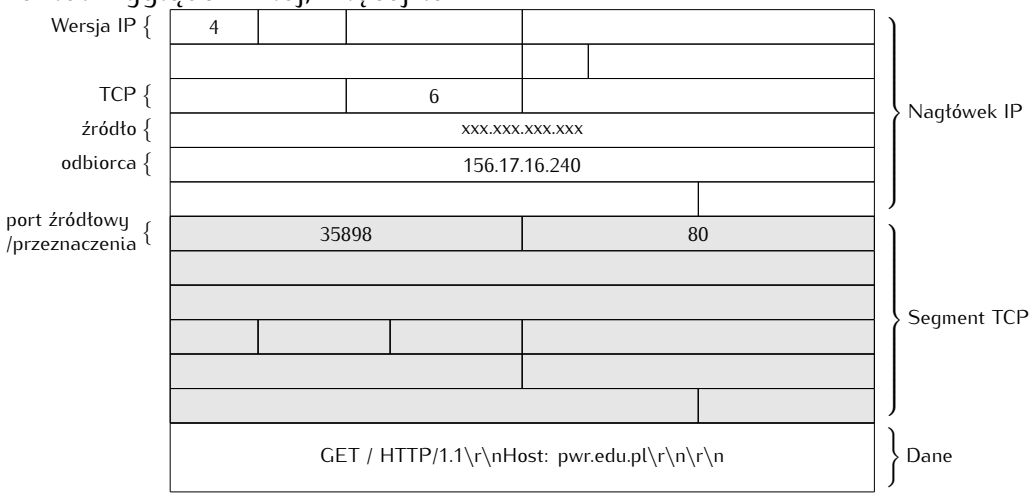
Poniżej dosyć długi przykład opisujący (w przybliżeniu) co się dzieje podczas komunikacji.

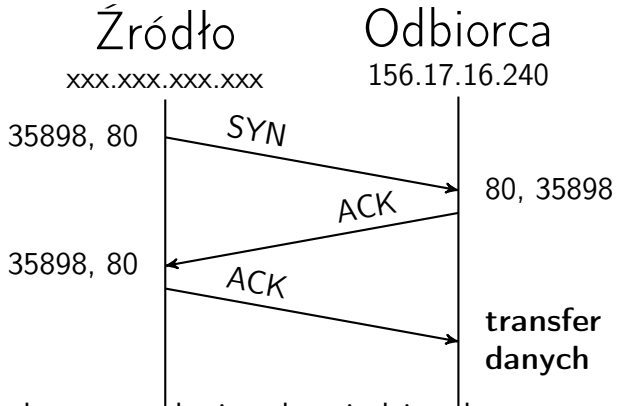
1. Uruchamiamy przeglądarkę i wpisujemy adres strony: http://pwr.edu.pl/.
2. Otwierany jest port do komunikacji (system operacyjny tworzy go i nadaje mu numer), na przykład 35898. Przeglądarka będzie z niego korzystała w sposób dosyć podobny jak z pliku na dysku.
3. Przeglądarka z całego adresu wydobywa adres serwera (wszystko to co jest za dwiema ukośnymi kreskami, a przed jedną ukośną) i wysyła do serwera (czyli pod adres pwr.edu.pl) zapytanie, które wygląda jakoś tak:

```
GET / HTTP/1.1
Host: pwr.edu.pl
```

(powyższe, to są dane informacje wysyłane do serwera). *Wszystko odbywa się w warstwie aplikacji.*

4. Najpierw dokonywana jest translacja adresu symbolicznego (pwr.edu.pl) na adres numeryczny (156.17.16.240)
5. Sprawdzane jest, czy adres znajduje się w „naszej" sieci (*NIE*).
6. Ustalany jest adres IP najlepszej bramy.
7. Ustalany jest adres fizyczny bramy (ARP).
8. Wysyłany jest pakiet otwierający połączenie TCP (*SYN*) ze zdalnym serwerem WWW, ale wysyłany jest on do lokalnego węzła pełniącego rolę bramy. Kolejne bramy będą w sposób przezroczysty przekazywały pakiet przepakowując go odpowiednio do użytego medium komunikacyjnego).
   Używany jest adres IP serwera WWW i numer portu docelowego 80 (WWW).
   Pakiet wygląda mniej, więcej tak:



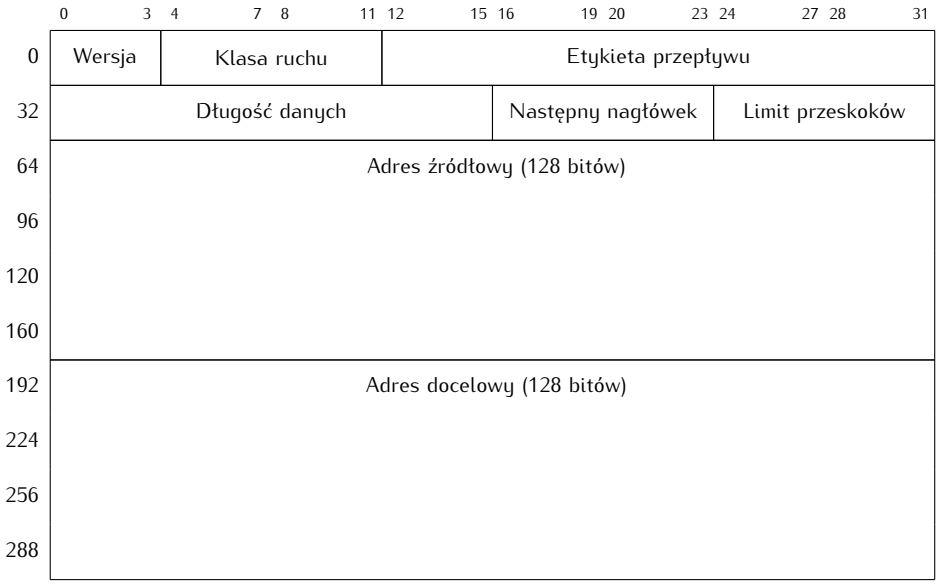


9. Podczas wysyłania odpowiedzi wykonywane są czynności podobne do tych opisanych w punktach 5, 6, 7.
   (Serwer zazwyczaj korzysta z informacji zawartych w pamięci podręcznej.)

Część II

Internet Protocol v.6

4. Datagram IPv6



Datagram IP ver. 6

— **Wersja** (4 bity) — definiująca wersję protokołu, w przypadku IPv6 pole to zawiera wartość 6 (bitowo 0110)
— **Klasa ruchu** (8 bitów) — określa sposób w jaki ma zostać potraktowany pakiet danych. W poprzedniej wersji protokołu pole to nazywało się **Type of Service**, jednak ze względu na to, że w IPv6 stosowane są inne mechanizmy priorytetowania danych, nazwę tego pola zmieniono
— **Etykieta przepływu** (20 bitów) — pomagające odróżnić pakiety, które wymagają takiego samego traktowania (ich pole klasy ruchu ma tę samą wartość)
— **Długość danych** (16 bitów) — wielkość pakietu, nie wliczając długości podstawowego nagłówka (wliczając jednak nagłówki rozszerzające)
— **Następny nagłówek** (8 bitów) — identyfikuje typ następnego nagłówka, pozwalając określić czy jest to nagłówek rozszerzający czy nagłówek warstwy wyższej. W przypadku tego drugiego, wartość pola jest identyczna z wartością pola w protokole IPv4
— **Limit przeskoków** (8 bitów) — określa ilość węzłów, po odwiedzeniu których pakiet zostaje porzucony. W poprzedniej wersji protokołu pole to nosiło nazwę time to live i zawierało liczbę skoków, która była zmniejszana przez każdy odwiedzony węzeł
— **Adres źródłowy** (128 bitów) — adres węzła, który wysłał pakiet
— **Adres docelowy** (128 bitów) — adres węzła do którego adresowany jest pakiet

5. Adresy IPv6

Adresy IPv6

— 128 bitów (340 282 366 920 938 463 463 374 607 431 768 211 456) $3{,}4028210^{38}$
— Adresy zapisywane są w postaci: `xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx` gdzie każdy znak `x` to cyfra szesnastkowa reprezentująca 4 bity adresu.
— W wersji tekstowej adresu może pojawić się (jeden raz) podwójny dwukropek `::`. Oznacza on ciągłe pole złożone z samych zer.
— Adres IPv6 mojego laptopa (w czasie pisania tych słów) to: `fe80::e565:6231:d639:2fba` czyli `fe80:0000:0000:0000:e565:6231:d639:2fba`

Najważniejsze różnice

— Liczba dostępnych adresów i różne tego konsekwencje:
  — własny adres IP dla (praktycznie) każdego sensownego gadżetu elektronicznego.
  — (teoretyczna) możliwość połączenia każdego węzła z każdym bez uciążliwych pośredników (typu NAT),
— Szyfrowanie w standardzie.
— Prostszy nagłówek IP.
— W IPv4 konfiguracja adrsu ręczna lub automatyczna (specjalne oprogramowanie zainstalowane w sieci) w przypadku IPv6 — ma się to odbywać „samoistnie" (ale prace są w początkowej fazie)

Część III

Przydział adresów IP

6. Sposoby przydziału adresów

IP v4

1. Ręczna
2. Automatyczna — serwer DHCP.
   Na podstawie adresu MAC karty przydzielany jest adres IP o ograniczonym czasie ważności.
   Administrator sieci konfiguruje serwer DHCP.
3. Serwer DHCP może współpracować z serwerami DNS.

IP v6

1. Automatyczna.
   Podstawowym identyfikatorem komputera jest EUI64 tworzony automatycznie na podstawie adresu MAC karty sieciowej przez wstawienie

w środek adresu MAC ciągu 0xFFFE i zanegowanie siódmego najstarszego bitu adresu. Ale generuje to przewidywalne adresy.
Stąd zazwyczaj generowane są one inaczej.

W obu przypadkach mogą być również przydzielane adresu *link-local* z zakresu 169.254.0.0/16 (IPv4) i FE80::/10 (IPv6).

W każdym przypadku, również dla adresów IPv6 można stosować przydział ręczy, czy korzystać z serwerów DHCP(v6).

Podstawowy problem z adresacja (IPv6) polega na tym, że dostępnych adresów jest tak dużo, iż (teoretycznie) można się nimi nie przejmować. Natomiast cały problem sprowadza się do tego, że:
— nie można dopuścić aby adresy się powtarzały,
— aby zapewnić, żeby komputery z sieci publicznej miały zapewniony routing.

Adresy IP v6

1. „Górne" 64 bity ustalane są w sposób ułatwiający routing w sieci globalnej.
2. Dla WASK jest to 2001:a48::/32
3. Kolejne bity mogą być „standaryzowane" na potrzeby routingu wewnątrz organizacji.
4. Specjalna kategoria to adresy postaci: FE80::/10 (1111 1110 10). Są to, tak zwane *link local addresses* nieroutowalne adresy węzłów podłączonych do medium lokalnego. Przydzielane są one automatycznie. Pozwalają one na komunikację między komputerami podłączonymi do tego medium.