



POLITECHNIKA POZNAŃSKA

Poznań University of Technology



Distributed Systems Group

Sieci komputerowe - administracja

warstwa sieciowa

Andrzej Stroiński

andrzej.stroinski@cs.put.edu.pl

<http://www.cs.put.poznan.pl/astroinski/>

29.03.2017

- zapewnia **adresowanie w sieci**
- ustala **trasę pomiędzy węzłami w sieci** (*ang. **routing***)
- definiuje **pakiety**
- definiuje **rozmiar pakietów**
- zapewnia **fragmentację pakietów** (*ang. **fragmentation***)
- **standardy:** IP, IPX, ICMP, ARP, DDP

- przypisywanie **adresów sieciowych urządzeniom**
- **przekazywanie pakietów** pomiędzy sieciami
- adres IP identyfikuje urządzenia sieciowe i określa ich lokalizację
- **nagłówek IP:**
 - **identyficação**
numer pakietu złożonego
 - **offset**
przesunięcie pakietu w złożonym pakiecie

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
Version				IHL				Type of service								Total length															
Identification																Flags				Fragment offset											
Time to live								Protocol								Checksum															
Source IP address																															
Destination IP address																															
Options (if data offset > 5; padded at the end with "0" bytes if necessary)																															
...																															

Time to live (TTL) – maksymalna liczba przeskoków (hops), którą może pokonać pakiet
 Flags (3b) + Fragment offset (13b) – jeśli datagram to fragment większego komunikatu

internet control message protocol (ICMP)



- przesyła informacje o **stanie** i **błędach w sieci**
- wspiera bezpołączeniowy IP
 - **brak hosta** docelowego
 - **błędy w trasowaniu**
- pakiety ICMP może być generowane przez przez dowolne urządzenie sieciowe

0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31

Type	Code	Checksum
Data (depends on the type)		
...		

internet control message protocol (ICMP)



- typy możliwych komunikatów

http://pl.wikipedia.org/wiki/IPv4#Nag.C5.82.C3.B3wek_IP

- **ping**

- echo replay (0)
- destination unreachable (3)
- echo request (8)

- **traceroute**

- echo replay (0)
- time exceeded (11)

traceroute google.com

```
traceroute to google.com (173.194.38.137), 30 hops max, 60 byte packets
 1 192.241.160.253 (192.241.160.253) 0.564 ms 0.539 ms 0.525 ms
 2 192.241.164.241 (192.241.164.241) 0.487 ms 0.435 ms 0.461 ms
 3 xe-3-0-6.ar2.nyc3.us.nlayer.net (69.31.95.133) 1.801 ms 1.802 ms 1.762 ms
 4 144.223.28.73 (144.223.28.73) 0.583 ms 0.562 ms 0.550 ms
 5 144.232.1.21 (144.232.1.21) 1.044 ms 1.048 ms 1.036 ms
 6 74.125.49.212 (74.125.49.212) 0.494 ms 0.688 ms 0.643 ms
 7 209.85.248.180 (209.85.248.180) 0.650 ms 209.85.248.178 (209.85.248.178) 0.621 ms 0.625 ms
 8 72.14.236.208 (72.14.236.208) 0.618 ms 72.14.236.206 (72.14.236.206) 0.898 ms 72.14.236.208
   (72.14.236.208) 0.872 ms
 9 72.14.239.93 (72.14.239.93) 7.478 ms 7.989 ms 7.466 ms
10 72.14.232.73 (72.14.232.73) 20.002 ms 19.969 ms 19.975 ms
11 209.85.248.228 (209.85.248.228) 30.490 ms 72.14.238.106 (72.14.238.106) 34.463 ms 209.85.248.228
   (209.85.248.228) 30.707 ms
12 216.239.46.54 (216.239.46.54) 42.502 ms 42.507 ms 42.487 ms
13 216.239.46.159 (216.239.46.159) 76.578 ms 74.585 ms 74.617 ms
14 209.85.250.126 (209.85.250.126) 80.625 ms 80.584 ms 78.514 ms
15 72.14.238.131 (72.14.238.131) 80.287 ms 80.560 ms 78.842 ms
16 209.85.250.228 (209.85.250.228) 171.997 ms 173.668 ms 170.068 ms
17 66.249.94.93 (66.249.94.93) 238.133 ms 235.851 ms 235.479 ms
18 72.14.233.79 (72.14.233.79) 233.639 ms 239.147 ms 233.707 ms
19 sin04s01-in-f9.1e100.net (173.194.38.137) 236.241 ms 235.608 ms 236.843 ms
```

traceroute



```
# traceroute -m 255 google.com
```

zmiana liczby hopów

```
# traceroute google.com 70
```

zmiana rozmiaru pakietu kontrolnego

internet group management protocol (IGMP v1, v2, v3?)



- protokół zarządzający komunikacją **multicast**
 - **dodawanie / usuwanie hostów**
 - **sprawdzanie uczestnictwa** w grupie multicast
 - wysłanie na IP grupy zapytania z TTL=1 (przez router)
 - odbiorcy będący w grupie po losowym czasie odpowiadają na adres grupy
 - jeśli chociaż jedna odpowiedź trafi do routera to zapisuje on sobie informacje, że istnieją komputery w tej sieci należące do poszukiwanej grupy

address resolution protocol (ARP)



- **problem**

- komputery w sieci identyfikuje się za pomocą adresu IP (warstwa 3)
- na poziomie warstwy drugiej (łącza danych) węzły w sieci są identyfikowane za pomocą adresu MAC
- **jak skojarzyć adres MAC z adresem IP?**
- statyczna konfiguracja odwzorowania
 - nieskalowana i problematyczna
- dynamiczna
 - ARP

adres resolution protocol (ARP)



- Działanie

- zapytanie: nadawca rozgłasza w sieci fizycznej pytanie o adres MAC urządzenia o adresie IP **X.X.X.X**; pytanie zawiera adresie MAC **aa:aa:aa:aa:aa:aa** nadawcy,
- odpowiedź: urządzenie, które dostało zapytanie i którego adres IP **to X.X.X.X** wysyła na **aa:aa:aa:aa:aa:aa** swój adres MAC **bb:bb:bb:bb:bb:bb**,
- (optymalizacja) nadawca zapisuje w prywatnej tablicy ARP-cache odwzorowanie **X.X.X.X** do **bb:bb:bb:bb:bb:bb**. Wpisy w ARP-cache są tymczasowe.

adres resolution protocol (ARP)



- reverse ARP
 - tłumaczenie MAC na IP,
 - obecnie nieużywane.
- Proxy ARP
 - używany, gdy niektóre komputery w sieci nie mają zdefiniowanej bramy,
 - gdy komputer próbuje wysłać komunikat do komputera spoza sieci, router może odpowiedzieć na zapytanie ARP swoim adresem MAC (adresem MAC interfejsu znajdującego się w tej sieci, z której pochodzi zapytanie),
 - router działa jako brama, choć komputer o tym nie wie

address resolution protocol (ARP)



```
# ip neigh show
192.168.1.1 dev eth0 lladdr 98:fc:11:bc:f1:a0 REACHABLE
# ping 192.168.1.100
PING 192.168.1.100 (192.168.1.100) 56(84) bytes of data.
64 bytes from 192.168.1.100: icmp_seq=1 ttl=64 time=83.1 ms
64 bytes from 192.168.1.100: icmp_seq=2 ttl=64 time=128 ms
^C
```

```
--- 192.168.1.100 ping statistics ---
```

```
# ip neigh show
192.168.1.100 dev eth0 lladdr 90:72:40:67:24:c7 REACHABLE
192.168.1.1 dev eth0 lladdr 98:fc:11:bc:f1:a0 REACHABLE
```

(minute później)

```
# ip neigh show
192.168.1.100 dev eth0 lladdr 90:72:40:67:24:c7 STALE
192.168.1.1 dev eth0 lladdr 98:fc:11:bc:f1:a0 REACHABLE
```

address resolution protocol (ARP)



```
# ip neigh add 192.168.1.101 lladdr aa:bb:cc:dd:ee:ff dev eth0
# ip neigh show
192.168.1.100 dev eth0 lladdr 90:72:40:67:24:c7 REACHABLE
192.168.1.101 dev eth0 lladdr aa:bb:cc:dd:ee:ff PERMANENT
192.168.1.1 dev eth0 lladdr 98:fc:11:bc:f1:a0 DELAY
# ip neigh del 192.168.1.101 dev eth0
# ip neigh show
192.168.1.100 dev eth0 lladdr 90:72:40:67:24:c7 REACHABLE
192.168.1.101 dev eth0 FAILED
192.168.1.1 dev eth0 lladdr 98:fc:11:bc:f1:a0 REACHABLE

# ip neigh flush dev eth0
# ip link set arp off dev eth0
# ip link set arp on dev eth0
```

address resolution protocol (ARP)



```
# arp
```

Address	HWtype	HWaddress	Flags	Mask	Iface
ap11	ether	98:fc:11:bc:f1:a0	C		eth0

```
# arp -i eth0 -s 192.168.1.101 aa:bb:cc:dd:ee:ff
```

```
# arp -i eth0 -d 192.168.1.101
```

```
# cat /proc/sys/net/ipv4/neigh/eth0/gc_stale_time  
60
```

```
# arping -I eth0 192.168.1.100
```

```
ARPING 192.168.1.100 from 192.168.1.103 eth0
```

```
Unicast reply from 192.168.1.100 [90:72:40:67:24:C7] 106.945ms
```

```
Unicast reply from 192.168.1.100 [90:72:40:67:24:C7] 51.238ms
```

```
Unicast reply from 192.168.1.100 [90:72:40:67:24:C7] 74.105ms
```

```
^CSent 4 probes (1 broadcast(s))
```

```
Received 3 response(s)
```

1. Podłącz swój komputer (poprzez port p4p1) do koncentratora (na zapleczu).
2. Skonfiguruj interfejs p4p1, tak by wszystkie komputery w rzędzie działały w jednej sieci (unikalne sieci między rzędami).
3. Zbadaj jak zmienia się tablica ARP, gdy uruchamiasz program `ping` z argumentami będącymi adresami IP komputerów z Twojej sieci i adresami komputerów w innych rzędach (należących do innych sieci).

dynamic host conf. protocol (DHCP)



- protokół służący do uzyskania przez węzeł sieci od serwera w sieci (lokalnej) danych do konfiguracji
 - adres IP
 - adres IP bramy
 - wartość MTU
 - nazwa DNS
 - adresy serwerów DNS
 - ...

- sposoby przydzielania adresu IP
 - **przypisanie na stałe adresu IP** do konkretnego urządzenia (po adresie MAC),
 - **przydział automatyczny** z wcześniej zdefiniowanej puli adresów,
 - **przydział dynamiczny** pozwalający urządzeniom na ponowne wykorzystanie poprzednio przydzielonych adresów
 - pula adresów zdefiniowana jak w przydziale automatycznym
 - każdy adres jest przydzielany (dzierżawiony) na określony czas,
 - urządzenia starają się odświeżać dzierżawę.

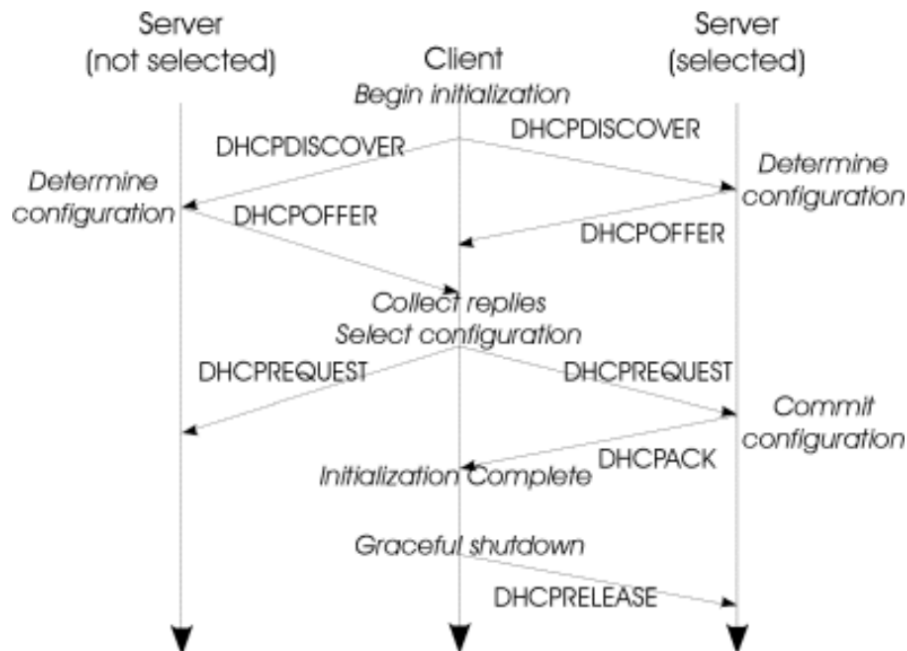
dynamic host conf. protocol (DHCP)



18

Inne komunikaty:

- DHCPNAK – odmowa przydziału parametrów konfiguracji,
- DHCPDECLINE – informacja, że adres sieciowy jest już używany,
- DHCPINFORM – żądanie przydziału parametrów konfiguracji bez przydziału adresu IP.



dynamic host conf. protocol (DHCP)



```
# dhclient -r -v eth0
```

```
Killed old client process
```

```
Internet Systems Consortium DHCP Client 4.3.4
```

```
...
```

```
Listening on LPF/eth0/1c:6f:65:82:11:8e
```

```
Sending on LPF/eth0/1c:6f:65:82:11:8e
```

```
Sending on Socket/fallback
```

```
DHCPRELEASE on eth0 to 150.254.31.62 port 67 (xid=0xb4d0356)
```

```
# dhclient -v eth0
```

```
Internet Systems Consortium DHCP Client 4.3.4
```

```
...
```

```
Listening on LPF/eth0/1c:6f:65:82:11:8e
```

```
Sending on LPF/eth0/1c:6f:65:82:11:8e
```

```
Sending on Socket/fallback
```

```
DHCPDISCOVER on eth0 to 255.255.255.255 port 67 interval 7 (xid=0x1b07331)
```

```
DHCPOFFER from 150.254.31.62
```

```
DHCPREQUEST on eth0 to 255.255.255.255 port 67 (xid=0x1b07331)
```

```
DHCPACK from 150.254.31.62 (xid=0x1b07331)
```

```
bound to 150.254.31.4 -- renewal in 1781 seconds.
```

dynamic host conf. protocol (DHCP)



```
# dhclient eth0
(rownolegle: journalctl [-e][-f] lub tail [-f] /var/log/messages)
# journalctl -f
...
Mar 24 14:44:45 tux dhclient[6363]: Created duid
                        "\000\004q\212W6$\374A\275\237W)\265\375\0138U".
Mar 24 14:44:45 tux dhclient[6363]: DHCPDISCOVER on eth0 to \
                        255.255.255.255 port 67 interval 8 (xid=0x8519f52c)
Mar 24 14:44:45 tux dhclient[6363]: DHCPREQUEST on eth0 to \
                        255.255.255.255 port 67 (xid=0x8519f52c)
Mar 24 14:44:45 tux dhclient[6363]: DHCPOFFER from 150.254.31.62
Mar 24 14:44:45 tux dhclient[6363]: DHCPACK from 150.254.31.62 (xid=0x8519f52c)
Mar 24 14:44:45 tux NET[6391]: /usr/sbin/dhclient-script : \
                        updated /etc/resolv.conf
Mar 24 14:44:47 tux dnsmasq[1272]: reading /etc/resolv.conf
Mar 24 14:44:47 tux dnsmasq[1272]: using nameserver 150.254.30.30#53
Mar 24 14:47:47 tux dnsmasq[1272]: using nameserver 150.254.5.4#53
Mar 24 14:44:47 tux dhclient[6363]: bound to 150.254.31.4 -- \
                        renewal in 1577 seconds.

# ip addr show eth0
...
    inet 150.254.31.4/25 brd 150.254.31.127 scope global dynamic eth0
        valid_lft 2898sec preferred_lft 2898sec
```

dynamic host conf. protocol (DHCP)



```
$ more /var/lib/dhclient/dhclient.leases
default-duid "\000\004q\212W6$\374A\275\237W)\265\375\0138U";
lease {
    interface "eth0";
    fixed-address 150.254.31.4;
    option subnet-mask 255.255.255.128;
    option routers 150.254.31.1;
    option dhcp-lease-time 3600;
    option dhcp-message-type 5;
    option domain-name-servers 150.254.30.30,150.254.5.4;
    option dhcp-server-identifier 150.254.31.62;
    option broadcast-address 150.254.31.127;
    option domain-name "cs.put.poznan.pl";
    renew 5 2017/03/24 14:40:12;
    rebind 5 2017/03/24 15:02:59;
    expire 5 2017/03/24 15:10:29;
}
```