

# Sieci komputerowe

Tadeusz Kobus, Maciej Kokociński  
Instytut Informatyki, Politechnika Poznańska

# Warstwa transportowa

# Warstwa transportowa

- Zapewnienia bezbłędną komunikację między komputerami (end-to-end).
- Odpowiada za dzielenie danych na fragmenty i kontrolę ich przesyłania.
- Ustanawia wirtualne połączenia, utrzymuje je i później likwiduje.
- Przykładowe standardy:
  - TCP
  - UDP
  - UDPlite
  - DCCP
  - SCTP

# Połączenia w warstwie transportowej

Połączenie jest identyfikowane przez:

- adres lokalny, port lokalny,
- protokół,
- adres zdalny, port zdalny.

Połączeniom w systemie op. odpowiadają **gniazda (sockets)**.

Port:

- 16-bitowy identyfikator usługi/aplikacji/procesu działającego na konkretnym urządzeniu (na konkretnym adresie IP):

1	–	1023	uprzywilejowane (system/well-known/privileged)
1024	–	49151	zarejestrowane (user/registered)
49152	–	65535	efemeryczne (dynamic/private/ephemeral)
- lista numerów portów przydzielona do usług przez IANA ([link](#)),
- lista podstawowych numerów portów – `/etc/services`.

# Aktywne połączenia (1)

```
# netstat -tunap
```

```
Active Internet connections (servers and established)
```

Prot	RQ	SQ	Local Address	Foreign Address	State	PID/Program name
tcp	0	0	0.0.0.0:17500	0.0.0.0:*	LISTEN	1618/dropbox
tcp	0	0	127.0.0.1:17600	0.0.0.0:*	LISTEN	1618/dropbox
tcp	0	0	0.0.0.0:22	0.0.0.0:*	LISTEN	972/sshd
tcp	0	0	127.0.0.1:631	0.0.0.0:*	LISTEN	971/cupsd
tcp	38	0	150.254.31.4:55148	108.160.172.204:443	CLOSE_WAIT	1618/dropbox
tcp	0	0	150.254.31.4:47070	108.160.163.108:443	ESTABLISHED	1618/dropbox
tcp	0	0	150.254.31.4:55904	64.233.162.189:443	ESTABLISHED	1855/opera
tcp	0	0	150.254.31.4:55802	54.192.228.32:443	ESTABLISHED	1618/dropbox
...						
tcp6	0	0	:::22	:::*	LISTEN	972/sshd
tcp6	0	0	:::1:631	:::*	LISTEN	971/cupsd
udp	0	0	0.0.0.0:17500	0.0.0.0:*		1618/dropbox
udp	0	0	0.0.0.0:50321	0.0.0.0:*		751/avahi-daemon
udp	0	0	0.0.0.0:5353	0.0.0.0:*		751/avahi-daemon
udp	0	0	0.0.0.0:2046	0.0.0.0:*		1277/dhclient
udp	0	0	0.0.0.0:68	0.0.0.0:*		1277/dhclient
udp6	0	0	:::39661	:::*		1277/dhclient

## Aktywne połączenia (2)

```
# netstat -tunap | grep ssh
```

```
tcp    0  0  0.0.0.0:22                0.0.0.0:*          LISTEN      972/sshd
tcp    0  0  150.254.31.4:60824       150.254.30.50:22   ESTABLISHED 3985/ssh
tcp6   0  0  :::22                    :::*                LISTEN      972/sshd
```

```
# netstat -tuap | grep ssh
```

```
tcp    0  0  0.0.0.0:ssh               0.0.0.0:*          LISTEN      972/sshd
tcp    0  0  dcs-rw-2.cs.put.p:60824  hpc.cs.put.poznan.p:ssh ESTABLISHED 3985/ssh
tcp6   0  0  [::]:ssh                 [::]:*              LISTEN      972/sshd
```

# Zadanie 1

1. Przy pomocy programu netstat zbadaj, na jakich portach uruchamiane są serwery:
  - HTTP,
  - HTTPS,
  - FTP,
  - telnet,
  - DNS.

# User Datagram Protocol (UDP)

- **Bezpołączeniowy** protokół transportowy.
- **Brak kontroli przepływu i gwarancji dostarczenia** – weryfikacja niezawodności w komunikacji musi być implementowana przez aplikacje.
- Weryfikacja integralności przez sumy kontrolne.
- Struktura nagłówka **datagramu**:

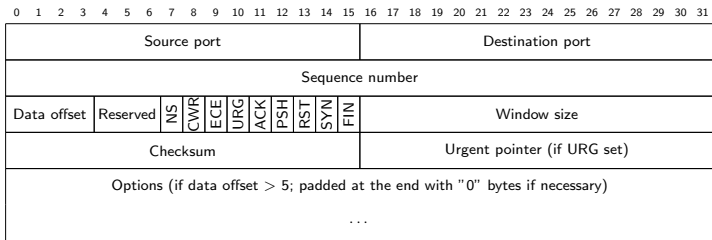
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
Source port																Destination port															
Length																Checksum															

- Protokoły/programy wykorzystujące UDP:
  - DNS
  - DHCP
  - RIP
  - strumieniowanie multimedialnych
  - gry sieciowe



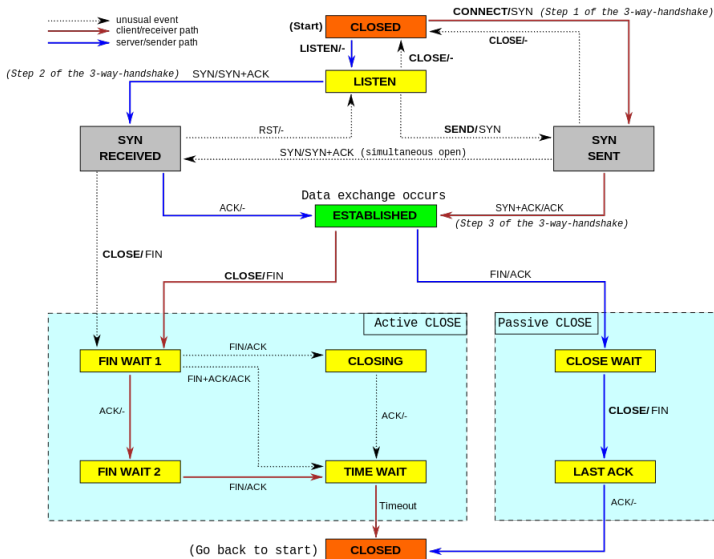
# Transmission Control Protocol (TCP) (1)

- Połączeniowy protokół sterowania transmisją.
- Kontrola przepływu (strumień), potwierdzenia, buforowanie.
- Struktura nagłówka segmentu:

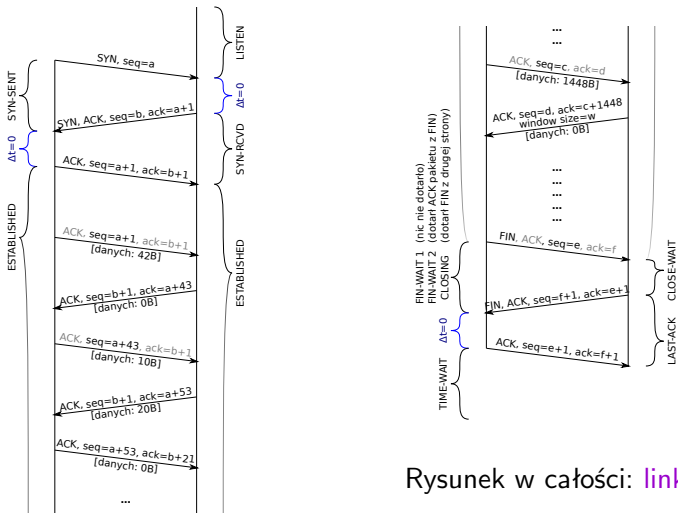


- Protokoły/programy wykorzystujące TCP:
  - HTTP
  - HTTPS
  - SMTP
  - POP3
  - IMAP
  - SSH
  - FTP

# Transmission Control Protocol (TCP) (2)



# Transmission Control Protocol (TCP) (3)



Rysunek w całości: [link](#)

# Operacje na gniazdach

`ncat` – jak `cat`, ale dla gniazd a nie dla plików

- kopiowanie danych przez sieć,
- przekierowywanie ruchu dla istniejących połączeń,
- tworzenie prostych usług i klientów usług,
- ...

(server)

```
# ncat -l 127.0.0.1 74 > out.txt
```

(client)

```
# ncat 127.0.0.1 74 < in.txt
```

## Zadanie 2

### Połączenie TCP

```
(server)
# ncat -l 127.0.0.1 74
... [ENTER]
```

```
(client)
# ncat 127.0.0.1 74
... [ENTER]
```

### Połączenie UDP

```
(server)
# ncat -ul 127.0.0.1 74
... [ENTER]
```

```
(client)
# ncat -u 127.0.0.1 74
... [ENTER]
```

1. Co widać w wiresharku?
2. Co pokazuje netstat (na różnych etapach uruchomienia programów)?
3. Kto pierwszy może pisać, klient czy serwer?
4. Jaka usługa jest przypisana do portu 74 (patrz /etc/services)?

# Serwer stron WWW (1)

```
http://www.cs.put.poznan.pl/tkobus
```

```
# ncat www.cs.put.poznan.pl 80
```

```
GET /tkobus/ HTTP/1.0 [ENTER]
```

```
[ENTER]
```

```
HTTP/1.1 200 OK
```

```
Date: Fri, 04 Dec 2015 13:15:54 GMT
```

```
Server: Apache
```

```
Last-Modified: Fri, 20 Nov 2015 22:02:53 GMT
```

```
ETag: "191a67-2887-52500097d7d40"
```

```
Accept-Ranges: bytes
```

```
Content-Length: 10375
```

```
Connection: close
```

```
Content-Type: text/html
```

```
<!DOCTYPE html>
```

```
<html xmlns="http://www.w3.org/1999/xhtml">
```

```
...
```

```
</html>
```

## Serwer stron WWW (2)

hello.html:

```
<html>
  <body>
    <font color="green">Hello world!</font>
  </body>
</html>
```

```
# ncat -l 127.0.0.1 80 < hello.html
```

## Serwer stron WWW (2)

hello.html:

```
<html>
  <body>
    <font color="green">Hello world!</font>
  </body>
</html>
```

```
# ncat -l 127.0.0.1 80 < hello.html
GET / HTTP/1.1
Host: 127.0.0.1
Connection: keep-alive
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,
                                             image/webp,*/*;q=0.8
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML,
           like Gecko) Chrome/46.0.2490.80 Safari/537.36 OPR/33.0.1990.58
Accept-Encoding: gzip, deflate, lzma, sdch
Accept-Language: en-US,en;q=0.8
```



## Serwer stron WWW (3)

Wiele połączeń: -k, --keep-open

```
# ncat -lk 127.0.0.1 80 -c 'echo -e "HTTP/1.1 200 OK\n"; cat hello.html'
```

```
# ncat -o out -lk 127.0.0.1 80 -c \  
    'echo -e "HTTP/1.1 200 OK\n"; \  
    echo -e "<html><body>It is $(date)</body></html>"'
```

# Inne fajne przykłady

```
# ncat -lk 127.0.0.1 8081 -c 'while true; do read i && echo [echo] $i; done'
```

```
# ncat -lk 192.168.1.101 8081 --max-conns 3 --allow 192.168.1.0/24  
--exec "/bin/bash"
```

(server)

```
# ncat -l 127.0.0.1 74 --chat
```

(client)

```
# ncat 127.0.0.1 74
```