

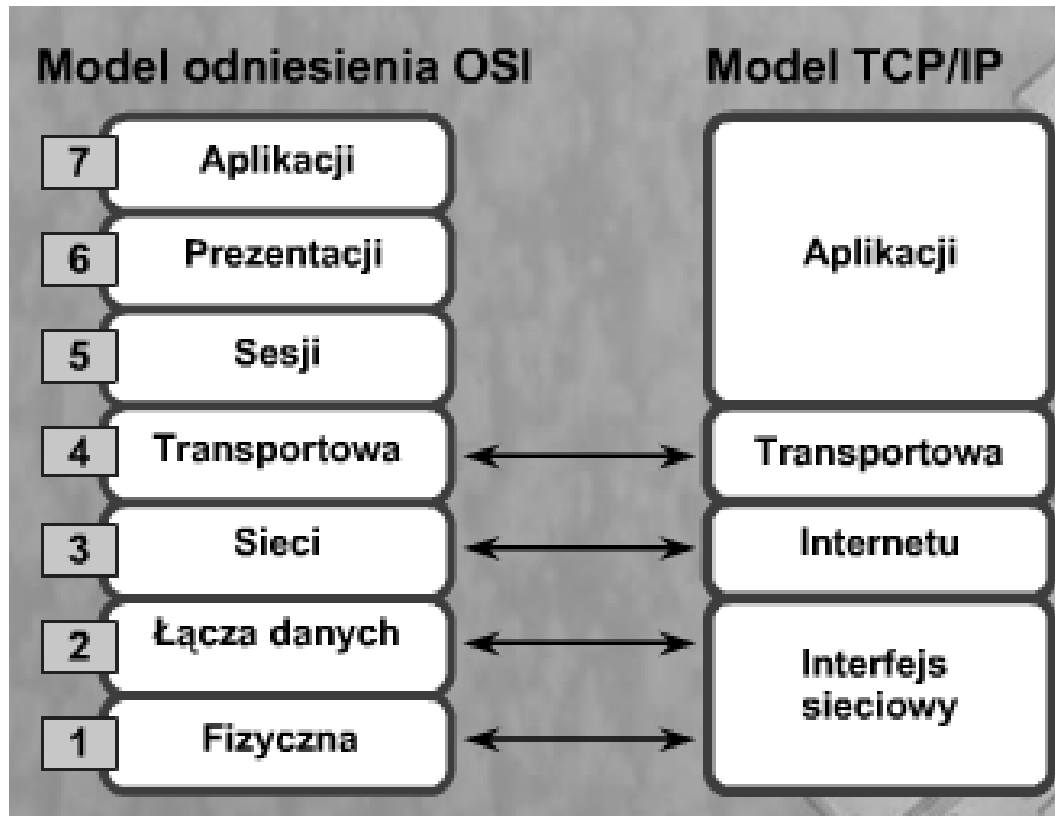


# **PODSTAWY SIECI KOMPUTEROWYCH**

**Modele warstwowe - Model OSI  
- warstwa sesji.**

# Funkcje warstwy

- ustawia, zarządza i kończy połączenia pomiędzy lokalną i zdalną aplikacją.



# Warstwa sesji

otrzymuje od różnych aplikacji dane, które muszą zostać odpowiednio zsynchronizowane.

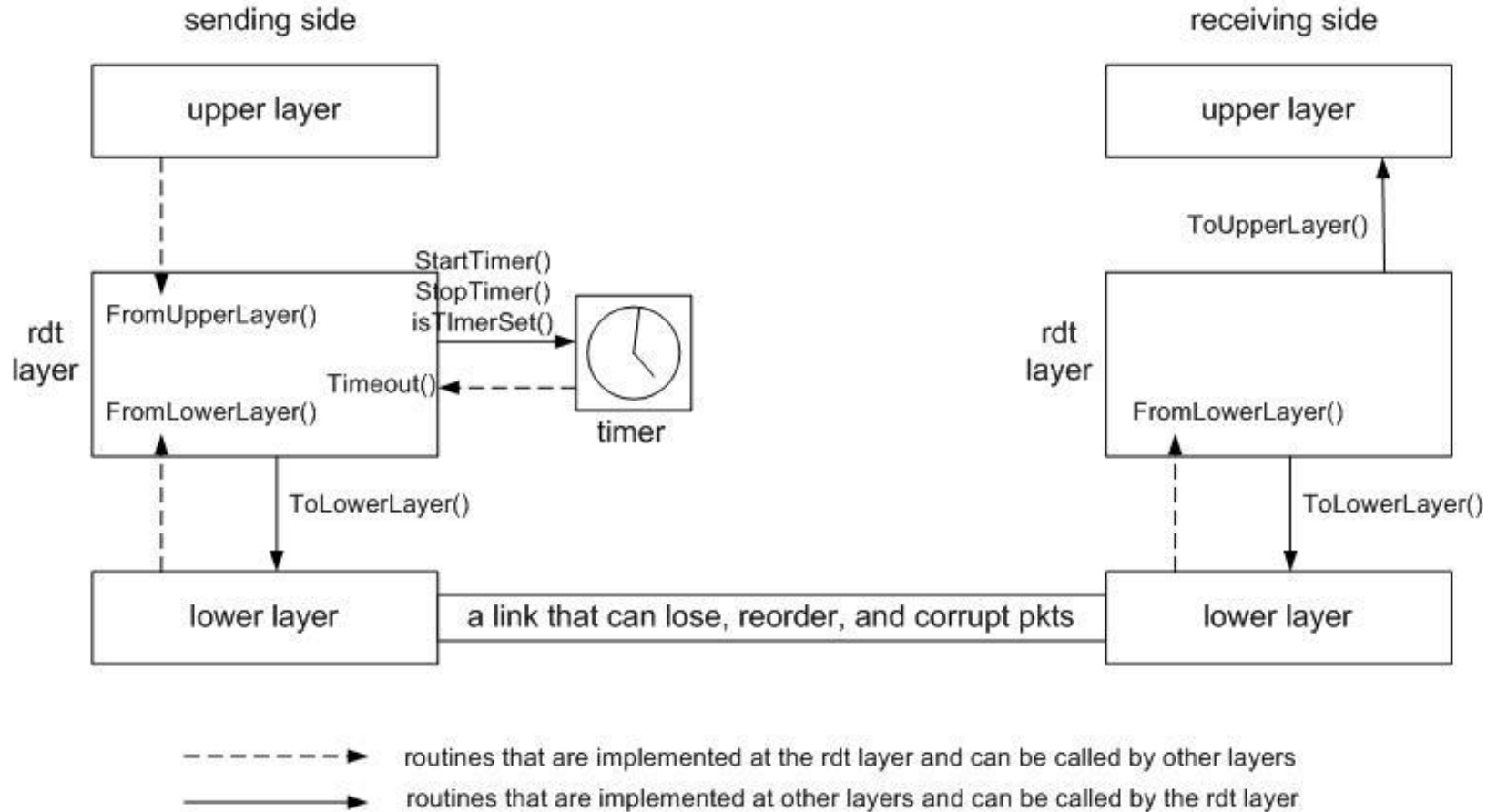
Synchronizacja występuje między warstwami sesji systemu nadawcy i odbiorcy.

Warstwa sesji wie, która aplikacja łączy się z którą, dzięki czemu może zapewnić właściwy kierunek przepływu danych – nadzoruje połączenie.

Wznawia połączenie po przerwaniu.

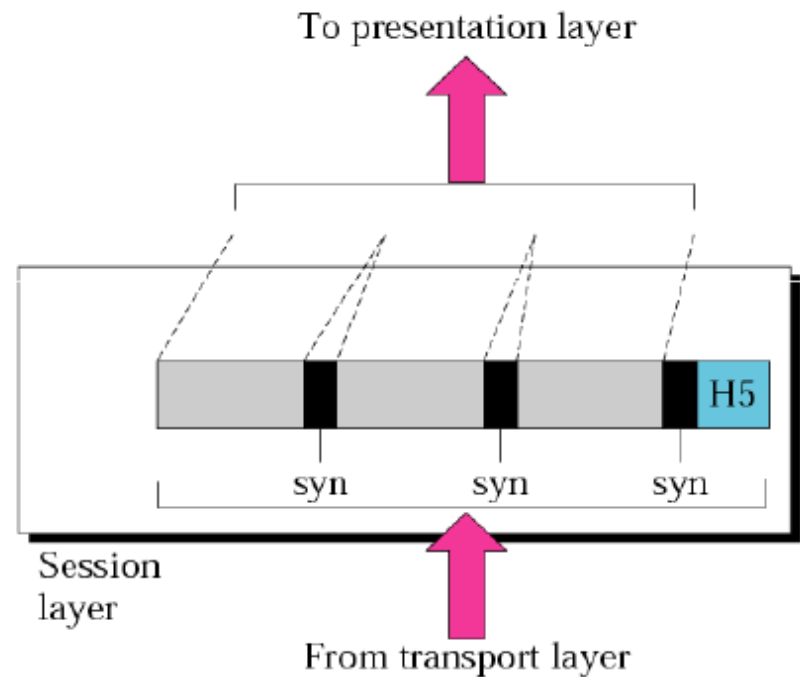
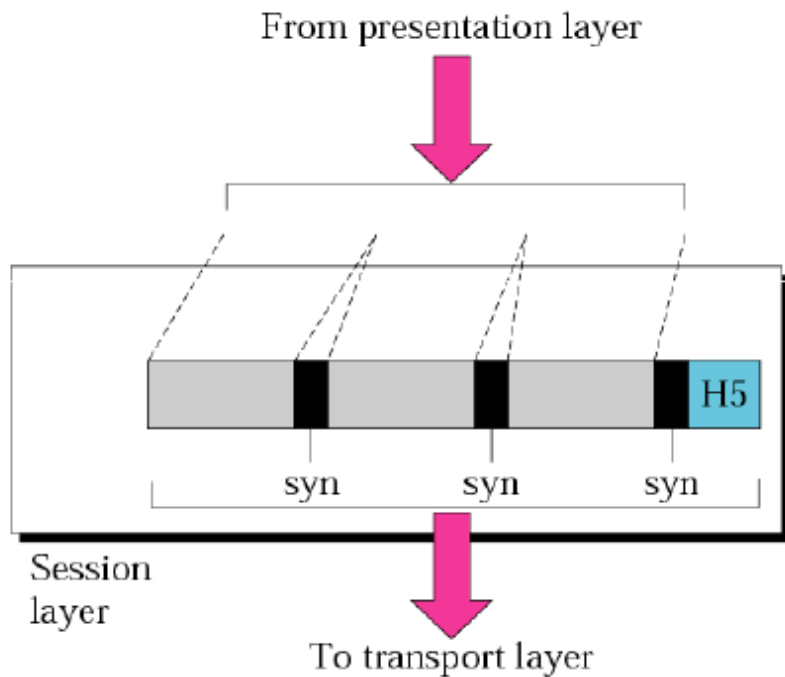
# Zadanie warstwy sesji

Synchronizacja między warstwami sesji systemu nadawcy i odbiorcy.



# Zadanie warstwy sesji

Synchronizacja między warstwami sesji systemu nadawcy i odbiorcy.



# Warstwa sesji

Warstwa sesji określa standardy i funkcje:

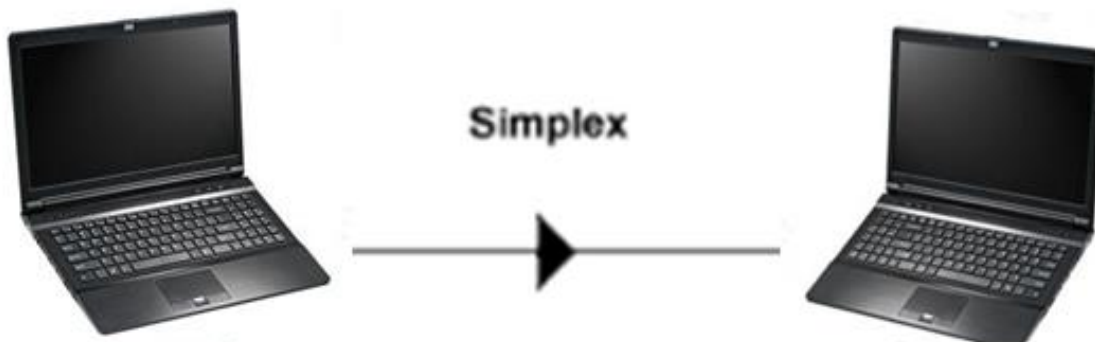
- zarządzania sesją, buduje i zamyka połączenie (w przypadku komunikacji połączeniowej)
- przesyłania danych,
- zarządzania wyjątkami (obsługa wyjątków),
- zarządzania prawami dostępu do usług warstwy Sesji (token management).

W warstwie sesji określane są parametry transmisji między komunikującymi się aplikacjami.

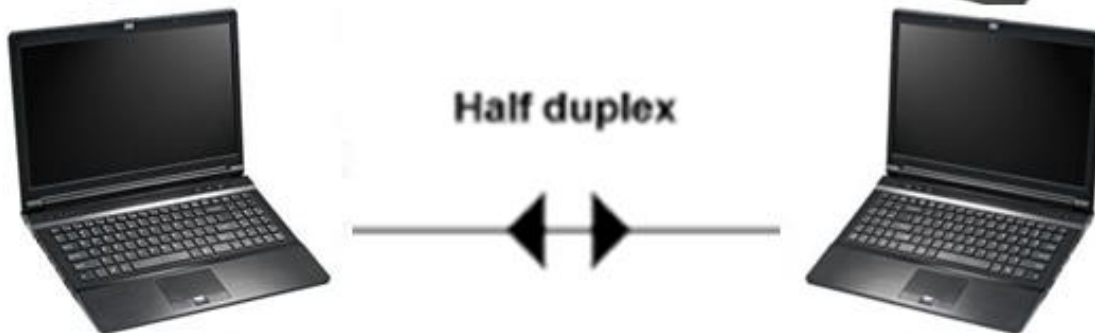
Określany jest np. tryb wymiany danych: simplex, half-duplex, full duplex.

# Tryb przesyłania danych

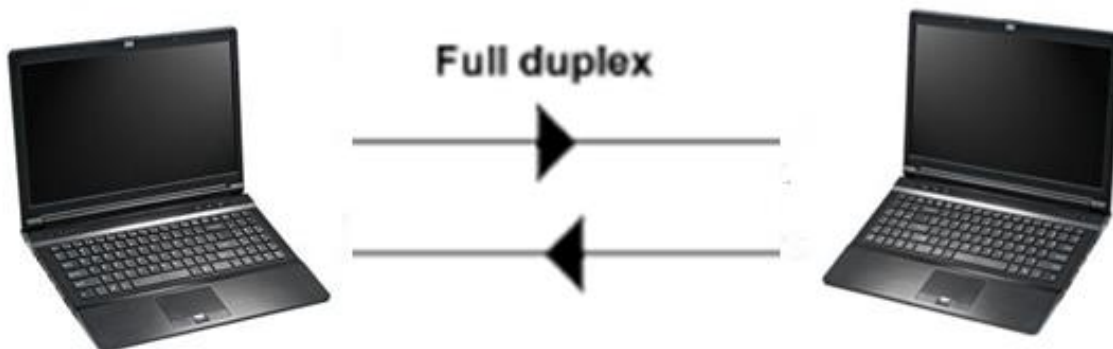
- simpleks



- pół-dupleks



- pełny dupleks





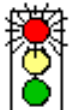




# Warstwa sesji

Przykłady protokołów i standardów warstwy sesji:

- X Window System
- Network File System (NFS)
- Remote Procedure Call (RPC)
- AppleTalk Session Protocol (ASP)
- Network Basic Input/Output System (NetBIOS)
- DNA Session Control Protocol, (DIGITAL Network Architecture - DNA).



# Przykładowy protokół: NetBIOS

OSI MODEL	
7	 <b>Application Layer</b> Type of communication: E-mail, file transfer, client/server.
6	 <b>Presentation Layer</b> Encryption, data conversion: ASCII to EBCDIC, BCD to binary, etc.
5	 <b>Session Layer</b> Starts, stops session. Maintains order.
4	 <b>Transport Layer</b> Ensures delivery of entire file or message.
3	 <b>Network Layer</b> Routes data to different LANs and WANs based on network address.
2	 <b>Data Link (MAC) Layer</b> Transmits packets from node to node based on station address.
1	 <b>Physical Layer</b> Electrical signals and cabling.

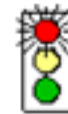
## NetBIOS

NetBIOS

NetBEUI,  
SPX,  
TCP

IPX,  
IP

5



## Session Layer

Starts, stops session.  
Maintains order.

NetBIOS

# Przykładowy protokół: NetBIOS

**NetBIOS** (ang. *Network Basic Input/Output System*) - protokół sieciowy oryginalnie zaprojektowany przez firmę IBM.

Zapewnia podstawowy interfejs łączenia aplikacji z innymi aplikacjami w innych komputerach znajdujących się w tej samej sieci lokalnej oraz umożliwia współdzielenie danych.

NetBIOS spełnia następujące założenia:

- aplikacje nie zajmują się szczegółami transportu
- wszystkie stacje w danej sieci traktowane są jednakowo
- usługi nie zależą od tego, jak sieć została zrealizowana sprzętowo
- obsługiwane są "przyjazne" nazwy - nie ma potrzeby posługiwania się adresami

# Przykładowy protokół: NetBIOS

NetBIOS dostarcza trzy odrębne usługi:

- usługa nazw służąca do rejestracji i przyznawania nazw (port 137/udp)
- usługa przekazywania datagramów dla komunikacji bezpołączeniowej (port 138/udp)
- usługa sesji dla komunikacji połączeniowej (port 139/tcp)

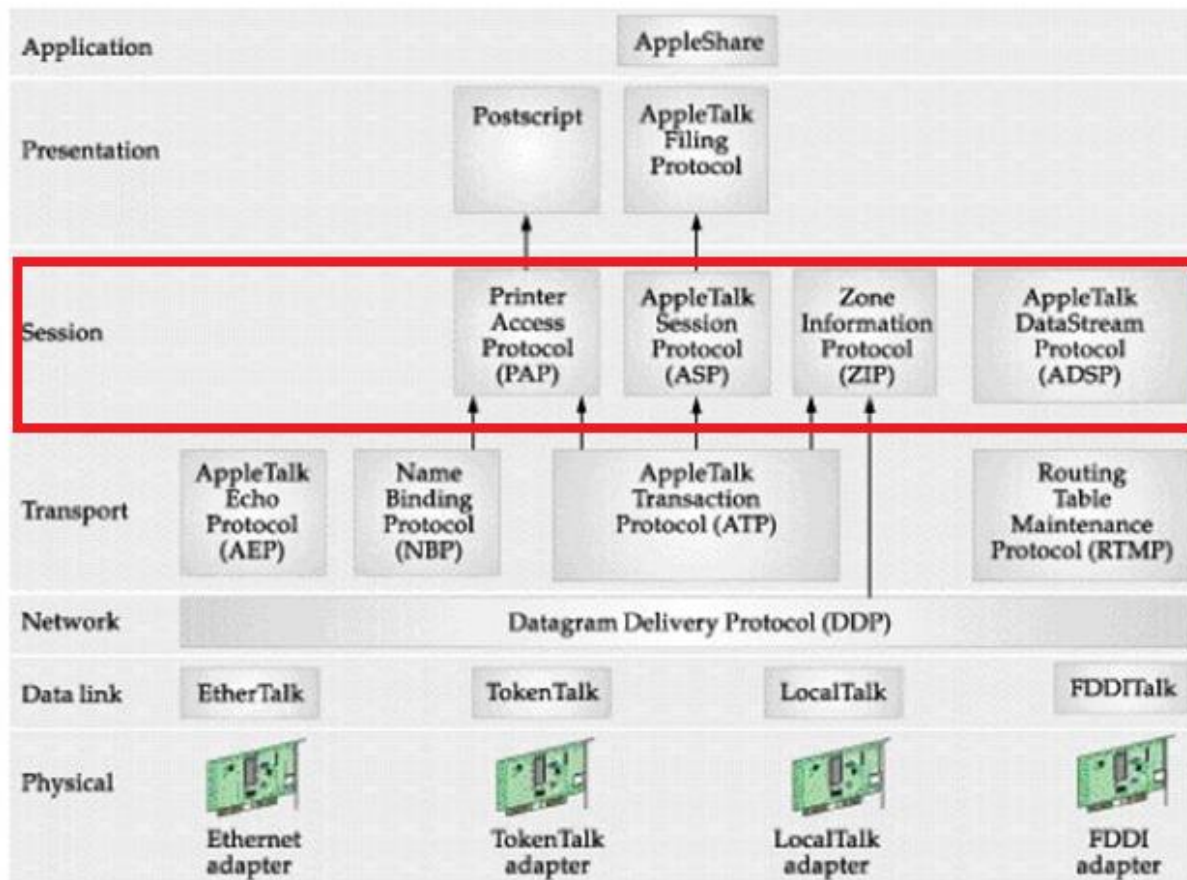
Wszystkie powyższe usługi są zaimplementowane w NBT.

NetBIOS wykorzystuje dwie metody komunikacji.

- **bezpoleczeniowa** bez potwierdzenia dotarcia datagramów (wykorzystywana przy broadcast-cie)
- **połączeniowa** z potwierdzeniami dotarcia wysyłanych datagramów

# Przykładowy protokół: ASP

Aplikacji		Sesji	Transportu	Datagramowa	Dostęp do sieci	
Aplikacji	Prezentacji	Sesji	Transportu	Sieci		Łącza danych Fizyczna



# Warstwa sesji / Gniazda

Gniazda są standardowym mechanizmem komunikowania się poprzez sieć TCP/IP.

Każde gniazdo posiada unikalny numer portu.

System rezerwuje pewne numery portów dla gniazd ogólnie znanych usług (np. 80 – HTTP, 21 – FTP).

Aby nie trafić na porty zajęte już przez inną aplikację lub usługę systemową, aplikacje użytkowników powinny wybierać porty z zakresu 1024 – 49151.

# Warstwa sesji / Gniazda

Połączenia z ogólnie znanym gniazdem na maszynie serwera są przechwytywane i przekierowane do nowego gniazda o losowym numerze portu.

Dzięki temu ogólnie znane gniazdo dostępne jest cały czas do obsługi nowych połączeń.

Istnieje możliwość tworzenia gniazd bezpołączeniowych. Ich użycie pozwala na uzyskanie większej efektywności, ale nie gwarantuje poprawnego dostarczenia danych.

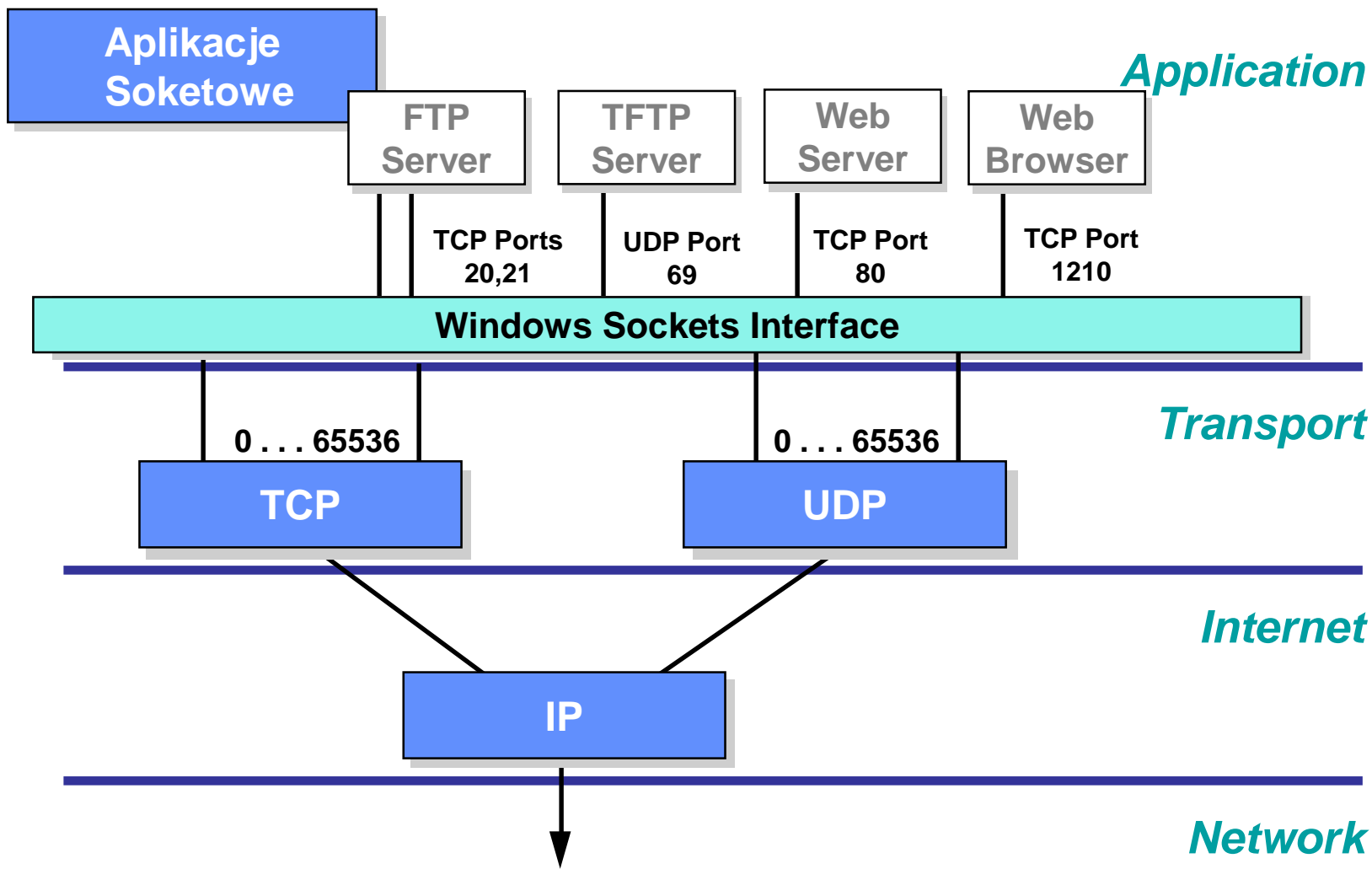
# Warstwa sesji / Gniazda

Serwery tworzą gniazda o ogólnie znanych numerach portów. Następnie serwer nasłuchuje połączeń przychodzących przez to gniazdo.

Klient po swojej stronie tworzy gniazdo o ogólnym numerze portu, przypisywanym losowo przez system.

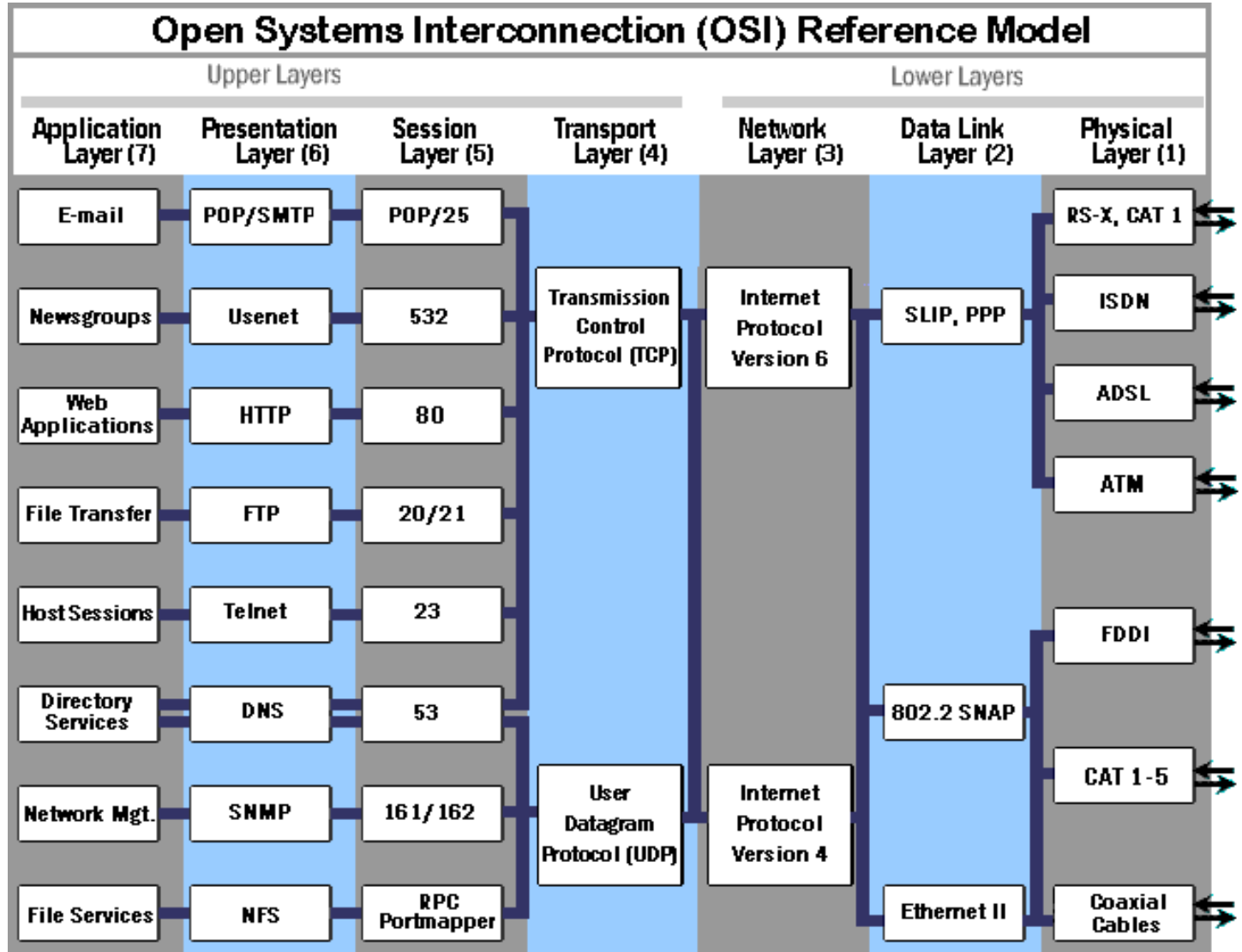
Gniazdo to następnie używane jest do nawiązania połączenia z gniazdem serwera.

# Warstwa sesji / Porty i Gniazda

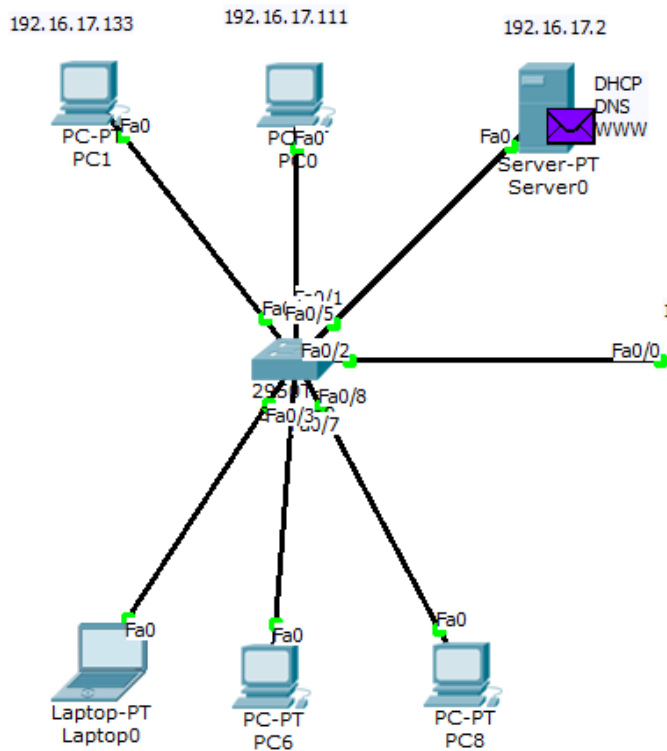




# Warstwa sesji / OSI



# CPT – warstwa sesji / OSI



komputery otrzymują adres ip z DHCP

PDU Information at Device: Server0

OSI Model    Inbound PDU Details    Outbound PDU Details

At Device: Server0  
Source: Laptop0  
Destination: HTTP CLIENT

Layer	Inbound PDU Details	Outbound PDU Details
Layer 7: HTTP	Layer 7: HTTP	
Layer 6	Layer 6	
Layer 5	Layer 5	
Layer 4: TCP Src Port: 1031, Dst Port: 80	Layer 4: TCP Src Port: 80, Dst Port: 1031	
Layer 3: IP Header Src. IP: 192.16.17.12, Dest. IP: 192.16.17.2	Layer 3: IP Header Src. IP: 192.16.17.2, Dest. IP: 192.16.17.12	
Layer 2: Ethernet II Header 0090.2184.170D >> 0003.E40C.4D30	Layer 2: Ethernet II Header 0003.E40C.4D30 >> 0090.2184.170D	
Layer 1: Port FastEthernet0	Layer 1: Port(s): FastEthernet0	

1. The device receives a TCP PUSH+ACK segment on the connection to 192.16.17.12 on port 1031.
2. Received segment information: the sequence number 1, the ACK number 1, and the data length 100.
3. The TCP segment has the expected peer sequence number.
4. TCP processes payload data.
5. TCP reassembles all data segments and passes to the upper layer.

Device	At Device	Type
Laptop0	Laptop0	HTTP
Laptop0	Laptop0	HTTP
Switch0	Switch0	HTTP
Server0	Server0	HTTP

Capture / Play    Capt

Show

# Ws – warstwa sesji / OSI

The image shows a Wireshark capture of network traffic on an Atheros NIC. The main pane displays a list of captured packets. Packet 7 is selected, showing an HTTP 200 OK response from 204.193.144.31 to 192.168.2.226. The packet details pane is expanded to show the Hypertext Transfer Protocol section, which includes the status line 'HTTP/1.0 200 OK' and various headers such as 'Cache-Control: private', 'Content-Length: 648', and 'Content-Type: text/html; charset=utf-8'. The packet bytes pane at the bottom shows the raw data in hexadecimal and ASCII, with the ASCII column displaying the text 'Content-Type: text/html; charset=utf-8'.

No.	Time	Source	Destination	Protocol	Info
1	0.000000	d0:67:e5:f3:3b:da	Broadcast	ARP	who has 192.168.2.224? Tell 192.168.1.254
2	0.203815	192.168.157.120	192.168.255.255	NBNS	Name query NB S257K4<20>
3	0.231166	fe80::e4fe:e43f:a12a::ff02::1:2		DHCPv6	Solicit XID: 0x56230a CID: 000100011b7e41ae002564a0c32a
4	0.312227	204.193.144.31	192.168.2.226	HTTP	HTTP/1.0 200 OK (text/html)
5	0.324075	192.168.2.226	204.193.144.31	TCP	49344 > http [FIN, ACK] Seq=1 Ack=992 Win=252 Len=0
6	0.324466	204.193.144.31	192.168.2.226	TCP	http > 49344 [FIN, ACK] Seq=992 Ack=2 Win=245 Len=0
7	0.324603	192.168.2.226	204.193.144.31	TCP	49344 > http [ACK] Seq=2 Ack=993 Win=252 Len=0
8	0.374642	Cisco_ba:49:2b	Spanning-tree-(for-br	STP	Conf. Root = 32768/0/00:24:f7:ba:49:0f Cost = 0 Port = 0x801c
9	0.579753	d0:67:e5:f3:3b:da	Broadcast	ARP	who has 192.168.222.234? Tell 192.168.1.254
10	0.964417	192.168.157.120	192.168.255.255	NBNS	Name query NB S257K4<20>
11	1.120402	fe80::e4fe:e43f:a12a::ff02::c		SSDP	M-SEARCH * HTTP/1.1
12	1.228015	d0:67:e5:f3:3b:da	Broadcast	ARP	who has 192.168.2.225? Tell 192.168.1.254
13	1.528250	Foxconn_a9:21:4d	Broadcast	ARP	who has 192.168.19.1? Tell 192.168.2.75
14	1.575807	d0:67:e5:f3:3b:da	Broadcast	ARP	who has 192.168.222.234? Tell 192.168.1.254

Internet Protocol, Src: 204.193.144.31 (204.193.144.31), Dst: 192.168.2.226 (192.168.2.226)  
Transmission Control Protocol, Src Port: http (80), Dst Port: 49344 (49344), Seq: 1, Ack: 1, Len: 991  
Hypertext Transfer Protocol  
HTTP/1.0 200 OK\r\n  
Cache-Control: private\r\n  
Content-Length: 648\r\n  
Content-Type: text/html; charset=utf-8\r\n  
Server: Microsoft-IIS/7.5\r\n  
X-Powered-By: ASP.NET\r\n  
Date: wed, 15 Oct 2014 05:57:29 GMT\r\n  
X-Cache: MISS from proxy.zsl.gda.pl\r\n  
X-Cache-Lookup: MISS from proxy.zsl.gda.pl:8080\r\n  
Via: 1.0 proxy.zsl.gda.pl (squid/3.1.19)\r\n  
Connection: keep-alive\r\n  
\r\n  
Line-based text data: text/html

0070 34 38 0d 0a 43 6f 6e 74 65 6e 74 2d 54 79 70 65 48..Cont ent-Type  
0080 3a 20 74 65 78 74 2f 68 74 6d 6c 3b 20 63 68 61 ; text/h tml; cha  
0090 72 73 65 74 3d 75 74 66 2d 38 0d 0a 53 65 72 76 rset=utf -8..Serv  
00a0 65 72 3a 20 4d 69 63 72 6f 73 6f 66 74 2d 49 49 er: Micr osoft-II  
00b0 53 2f 37 2e 35 0d 0a 58 2d 50 6f 77 65 72 65 64 S/7.5.X -Powered  
00c0 24 42 70 22 20 41 52 50 20 40 45 54 0d 03 44 61 by: ASP .NET 02

HTTP Content-Type header (http.content\_ty... Packets: 89 Displayed: 89 Marked: 0 Dropped: 0

# Ws – warstwa sesji / NetBIOS

```
921 19.155699 153.19.157.25 62.61.48.236 HTTP [TCP Retransmission] HTTP
922 19.155751 62.61.48.236 153.19.157.25 TCP [TCP Dup ACK 916#1] 50885
```

```
⊕ Frame 923: 92 bytes on wire (736 bits), 92 bytes captured (736 bits)
⊕ Ethernet II, Src: AsustekC_0c:2d:85 (00:18:f3:0c:2d:85), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
⊕ Internet Protocol, Src: 62.61.34.214 (62.61.34.214), Dst: 62.61.34.255 (62.61.34.255)
⊕ User Datagram Protocol, Src Port: netbios-ns (137), Dst Port: netbios-ns (137)
```

## NetBIOS Name Service

Transaction ID: 0x8148

### Flags: 0x0110 (Name query)

```
0... .. = Response: Message is a query
.000 0... .. = Opcode: Name query (0)
.... ..0. .... = Truncated: Message is not truncated
.... ...1 .... = Recursion desired: Do query recursively
.... .... ..1 .... = Broadcast: Broadcast packet
```

Questions: 1

Answer RRs: 0

Authority RRs: 0

Additional RRs: 0

### Queries

⊕ MSHOME<1b>: type NB, class IN

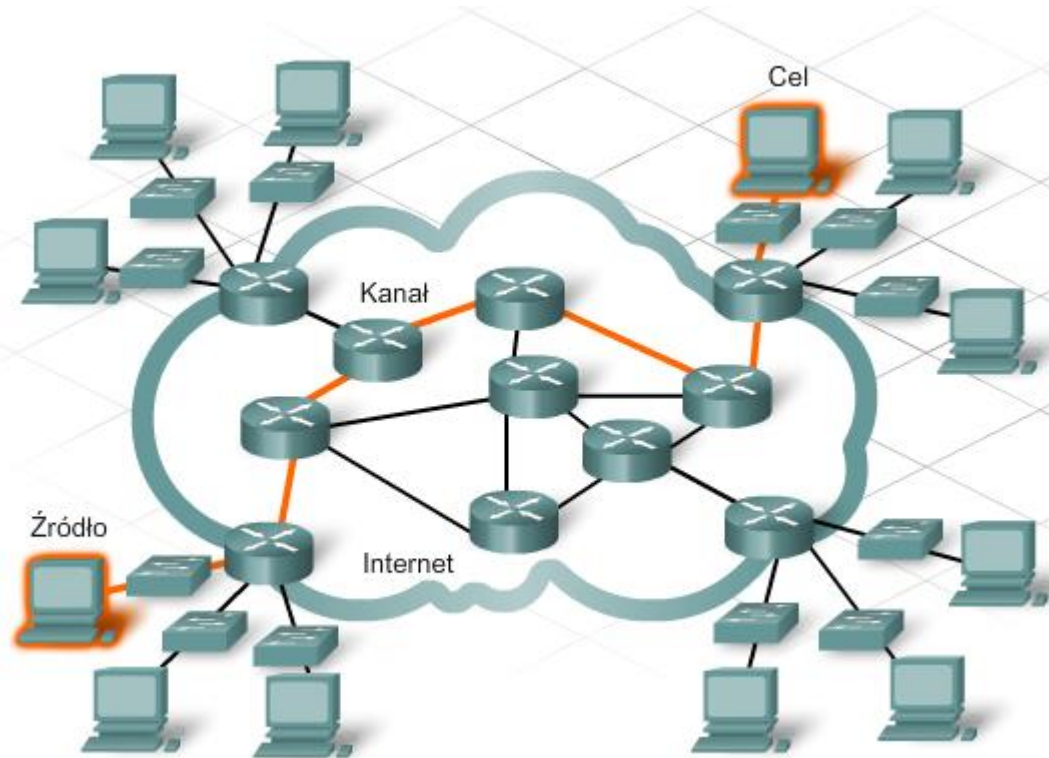
```
0020 22 ff 00 89 00 89 00 3a 41 5e 81 48 01 10 00 01 ".....: A.H....
0030 00 00 00 00 00 00 20 45 4e 46 44 45 49 45 50 45 ..... E NFDEIEPE
0040 4e 45 46 43 41 43 41 43 41 43 41 43 41 43 41 43 NEFCACAC ACACACAC
0050 41 43 41 43 41 42 4c 00 00 20 00 01 ACACABL. . .
```

NetBIOS Name Service (nbns), 50 bytes

Packets: 1058 Displayed: 1058 Marked: 0 Dropped: 0

# Przesyłanie danych w sieci

- Tryb połączeniowy - przed rozpoczęciem komunikacji następuje nawiązanie logicznego połączenia pomiędzy oboma urządzeniami.
- Tryb bezpołączeniowy - komunikaty wysyłane są niezależnie.





# Tryb połączeniowy

Po wybraniu najkrótszej trasy z węzła A do węzła Y, która przebiega przez węzły C i D wysyłane jest żądanie zestawienia połączenia od węzła początkowego A do węzła kolejnego C.

Po otrzymaniu potwierdzenia żądanie przekazywane jest dalej od C do D i następnie do Y.

Po zestawieniu całej trasy od węzła ostatniego wysyłane jest potwierdzenie do węzła początkowego.

Po zestawianiu trasa następuje przesyłanie danych.

Po zakończeniu przesyłania następuje rozłączanie trasy - od węzła początkowego do końcowego, z potwierdzeniem w kierunku odwrotnym.

# Tryb połączeniowy

- posiada mechanizmy kontroli błędów:
  - potwierdzenie zestawienia połączenia,
  - gdy zostanie przekroczony limit czasu (brak potwierdzenia odbioru ramki od stacji docelowej ) - retransmisja danych,
  - suma kontrolna sprawdzana w węźle docelowym.

# Tryb bezpołączeniowy

Nie ma tu potwierdzeń zestawienia połączenia.

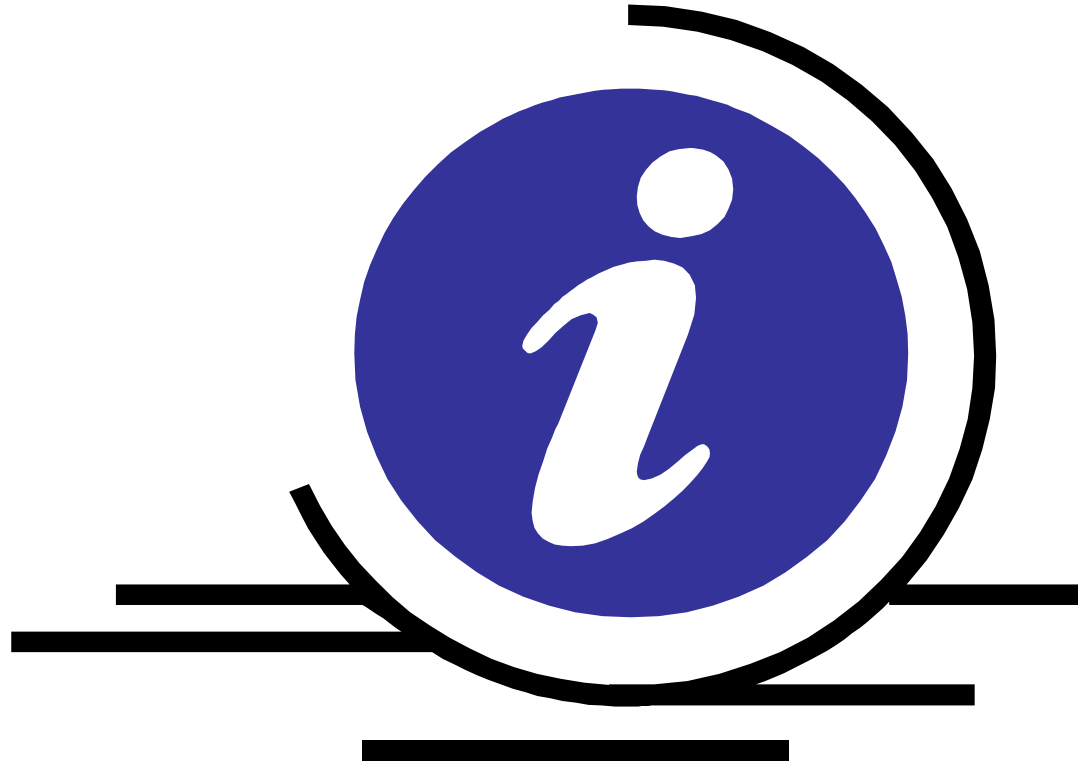
Zaraz po znalezieniu drogi rozpoczyna się transmisja.

Podobnie jak w trybie połączeniowym - gdy pakiet dotrze do węzła docelowego wysyłane jest potwierdzenie.

Pakiety wysyłane są niezależnie od siebie, może się zatem zdarzyć taka sytuacja, iż pójdą one różnymi trasami i dotrą do celu w innej kolejności. Stąd też potrzeba ich numerowania oraz układania w odpowiednim porządku w stacji docelowej.



# Pytania

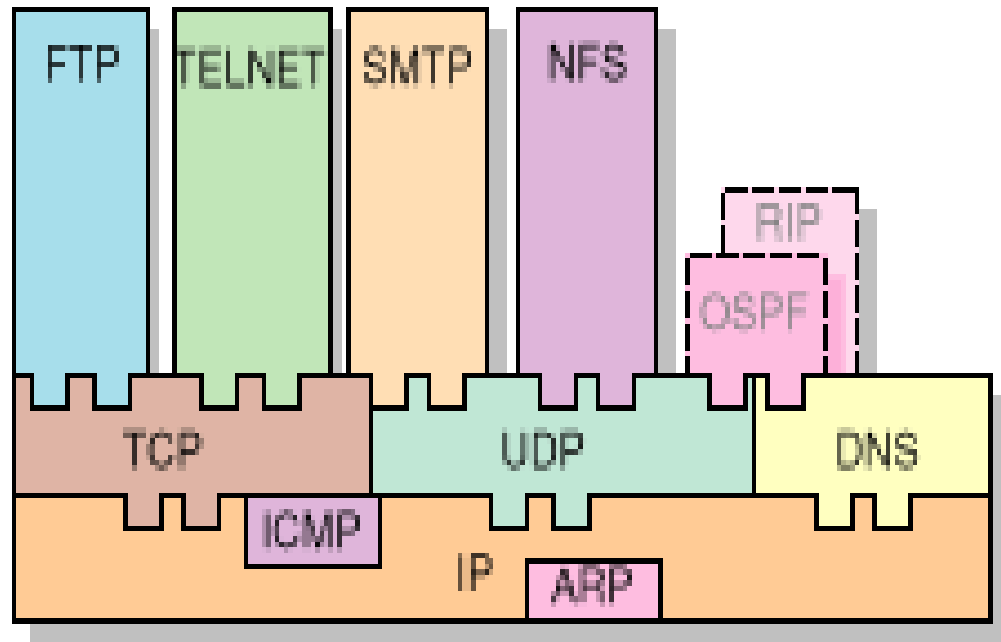


# ISO/OSI i stos protokołów TCP/IP

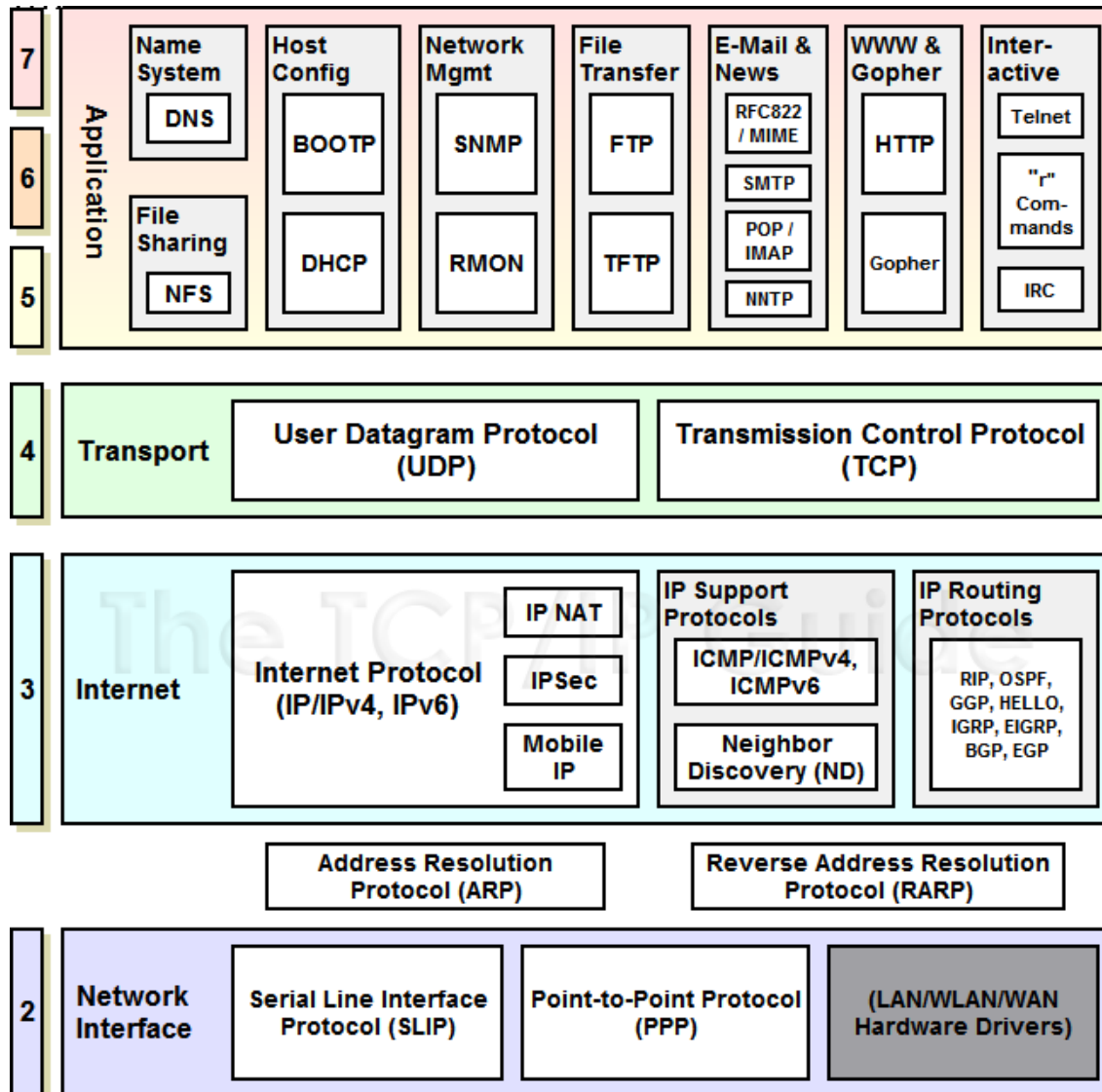
## ISO/OSI



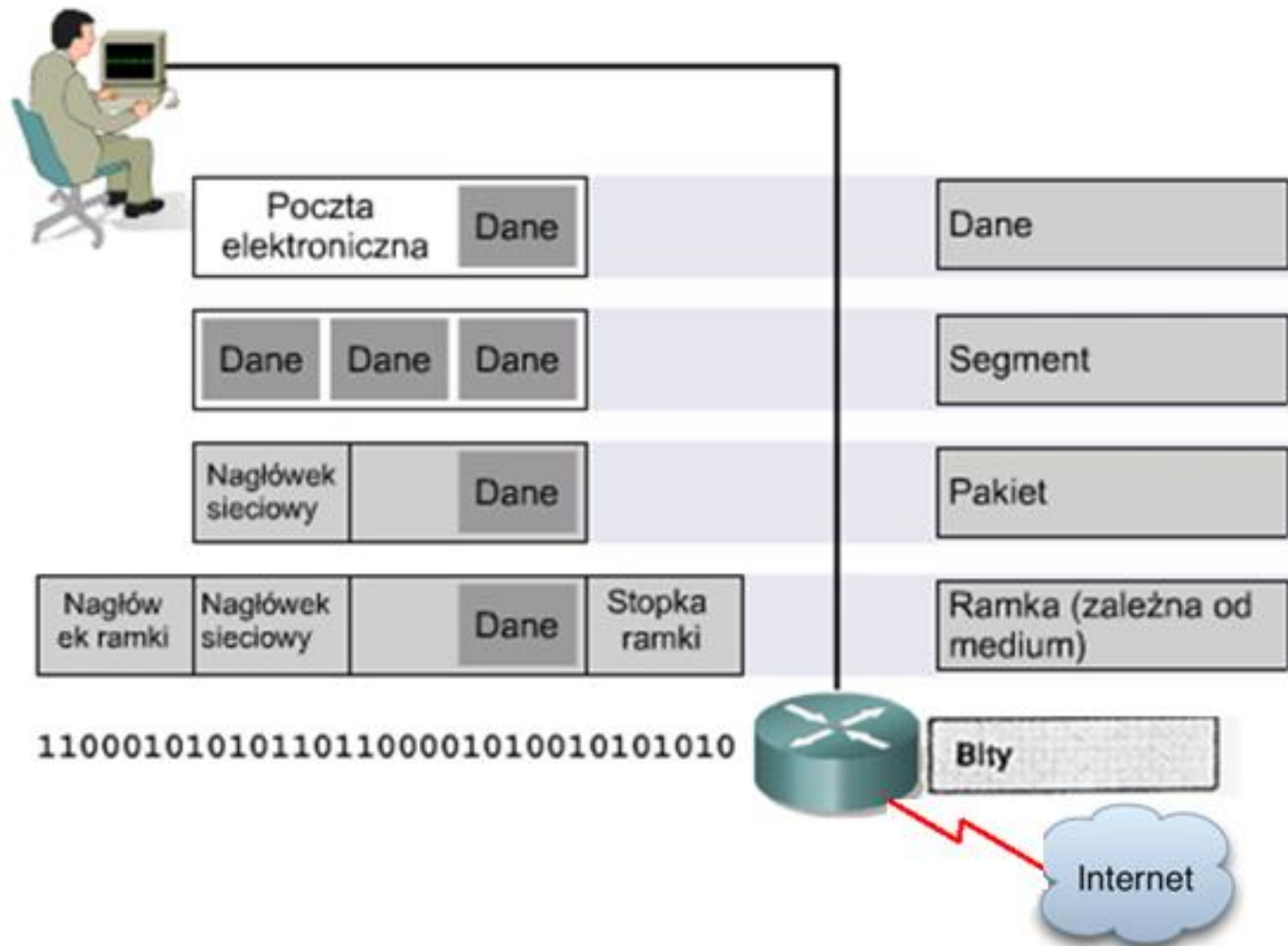
## TCP/IP



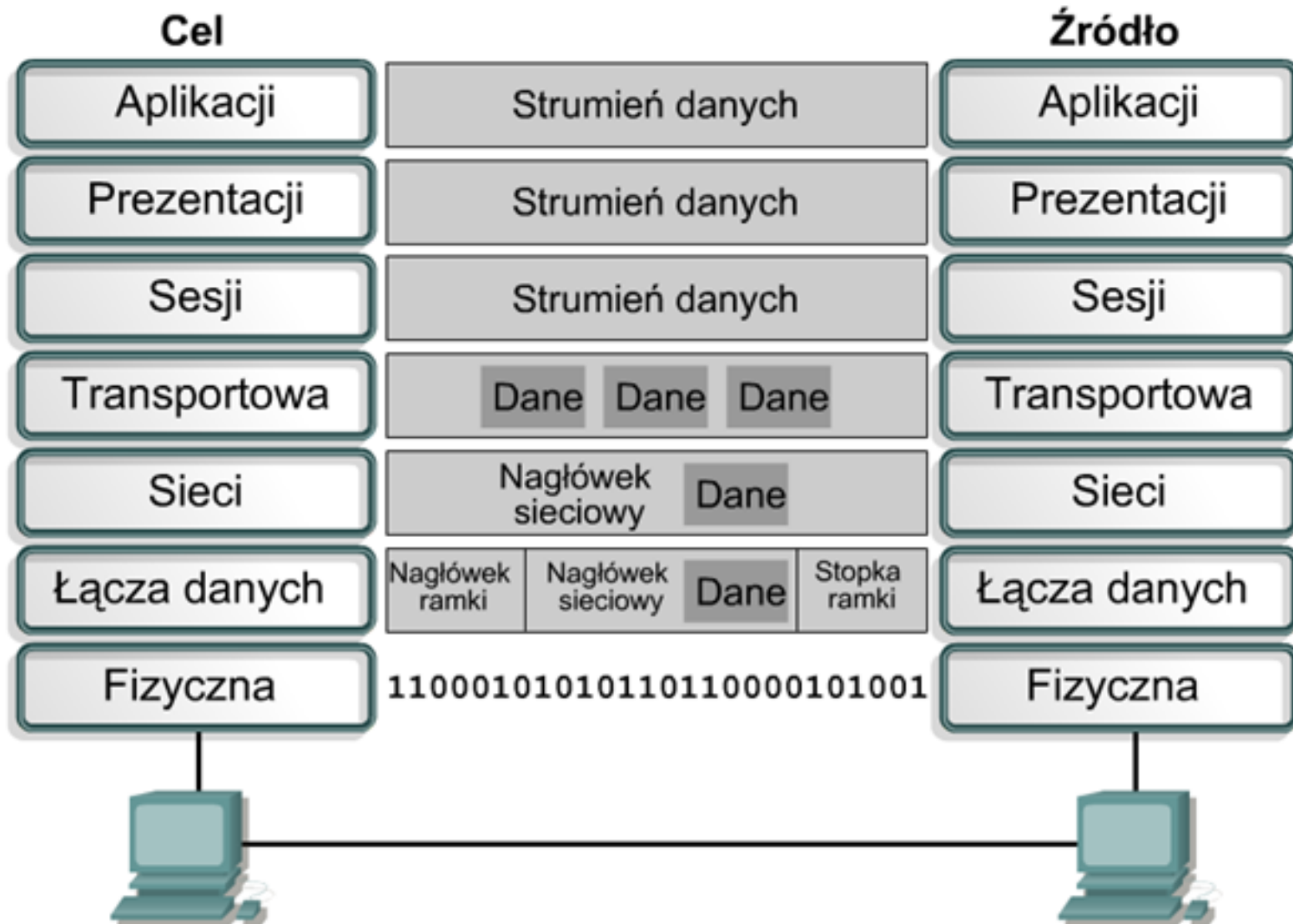
# Stos protokołów TCP/IP / protokoły dostępu



# Mechanizm enkapsulacji



# Mechanizm enkapsulacji



# Mechanizm enkapsulacji

