

Sieci komputerowe

Wykład 7 — Warstwa transportowa

Marta Szarmach
Zakład Telekomunikacji Morskiej
Wydział Elektryczny
Uniwersytet Morski w Gdyni

04.2022

Plan prezentacji

1 Warstwa transportowa modelu OSI

- Przypomnienie
- Pojęcie portu

2 Protokół TCP

- Cechy
- Nagłówek
- Działanie

3 Protokół UDP

- Cechy
- Nagłówek

1. Warstwa transportowa

Rola, protokoły, pojęcie portu

1.1 Warstwa transportowa. Przypomnienie

Warstwa transportowa — przypomnienie:

Rola

- Tworzenie całościowego połączenia pomiędzy usługami na konkretnych urządzeniach
- Segmentowanie danych — dane przed wysłaniem są buforowane, w jednym pakiecie mogą być umieszczone dane od kilku aplikacji i odwrotnie
- W przypadku TCP — zapewnienie niezawodności komunikacji i właściwej kolejności odbieranych danych

Protokoły

- TCP
- UDP

Jednostka danych

- dla TCP: segment
- dla UDP: datagram

Identyfikacja poprzez

Numer portu

1.2 Warstwa transportowa. Pojęcie portu

Definicja

Portem sieciowym nazywamy identyfikator konkretnej usługi (procesu) na zdalnym urządzeniu.

Rodzaje portów:

- **Dobrze znane** — od 0 do 1023, przypisane konkretnym usługom
- **Zarejestrowane** — od 1024 do 49151, z których zazwyczaj korzystają pewne usługi (ale nie muszą)
- **Prywatne (dynamiczne)** — od 49152 do 65535, przydzielane dynamicznie klientom według potrzeby

1.2 Warstwa transportowa. Pojęcie portu

Przykłady portów dobrze znanych:

- 20 i 21 — **FTP** (serwer plików)
- 22 — **SSH** (bezpieczne połączenie z urządzeniem)
- 23 — **telnet** (połączenie z urządzeniem)
- 25 — **SMTP** (serwer poczty wychodzącej)
- 53 — **DNS** (tłumaczenie adresów IP na mnemoniczne)
- 80 — **HTTP** (przesył stron WWW)
- 110 — **POP3** (serwer poczty przychodzącej)
- 143 — **IMAP** (serwer poczty przychodzącej)
- 443 — **HTTPS** (bezpieczny przesył stron WWW)

1.2 Warstwa transportowa. Pojęcie portu

Definicja

Gniazdem (ang. *socket*) nazywamy kombinację adresu IP urządzenia źródłowego/docelowego oraz numeru portu źródłowego/docelowego.

Gniazda umożliwiają jednoznaczną identyfikację konkretnej usługi na konkretnym urządzeniu.

Przykład

192.168.0.10:80

Usługa HTTP (port 80) na urządzeniu o adresie IP 192.168.0.10

2. Protokół TCP

Cechy i działanie

2.1 Protokół TCP. Cechy

Cechy protokołu TCP (ang. *Transmission Control Protocol*):

- **Połączeniowość** — pomiędzy komunikującymi się urządzeniami tworzona jest sesja, która jest kontrolowana
- **Działanie w trybie klient-serwer**
- **Niezawodność** — TCP stosuje mechanikę potwierdzeń (ACK), aby być pewnym, że wysłane dane zostały poprawnie odebrane (i we właściwej kolejności), w razie potrzeby dokonuje retransmisji utraconych danych
- **Umiejętność sterowania przepływem** — komunikujące się urządzenia potrafią regulować ilość przesyłanych danych w czasie, aby nie doszło do przeciążenia

2.1 Protokół TCP. Cechy

Zalety i wady protokołu TCP:

Zalety:

- Niezawodność — dane na pewno dotrą w nieuszkodzonej formie i kolejności
- Możliwość sterowania przepływem

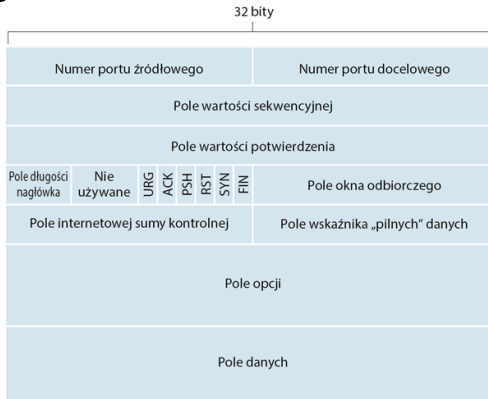
Wady:

- Wysyłanie potwierdzeń i retransmisji dodatkowo obciąża łącze

W związku z tym, TCP najczęściej przenosi informacje pochodzące z usług nieakceptujących błędów w transmisji: przesył zawartości stron internetowych (HTTP), poczty elektronicznej (POP3, IMAP, SMTP), plików (FTP), telnet oraz SSH.

2.2 Protokół TCP. Nagłówek

Budowa nagłówka TCP:



Grafika: strefainzyniera.pl

2.2 Protokół TCP. Nagłówek

Budowa nagłówka TCP:

- **Port źródłowy** — identyfikator usługi (aplikacji) źródłowej
- **Port docelowy** — identyfikator usługi (aplikacji) docelowej
- **Numer sekwencyjny** — numer identyfikujący konkretny segment w całym strumieniu danych
- **Numer potwierdzenia** — numer używany do potwierdzenia odebrania segmentu; informuje, którego bajtu danych oczekuje teraz odbiorca
- **Suma kontrolna** — umożliwia wykrycie błędów w nagłówku TCP
- **Rozmiar okna odbiorczego** — zawiera informacje o tym, ile bajtów odbiorca jest w stanie przyjąć, wykorzystywane przy sterowaniu przepływem

2.2 Protokół TCP. Nagłówek

Budowa nagłówka TCP:

- **Flagi** — wskazują cel segmentu:
 - URG — dane prorytetowe (wymusza natychmiastowe przesłanie segmentu, bez jego buforowania)
 - ACK — segment stanowi potwierdzenie odebrania porcji danych
 - PSH — podobnie jak URG wymusza przesłanie danych (np. kiedy wiadomo, że są to ostatnie dane i nie ma sensu czekać na następne)
 - RST — informacja o resecie/odrzućeniu połączenia (np. przy próbie rozpoczęcia sesji)
 - SYN — sygnalizuje chęć rozpoczęcia sesji i synchronizuje numery sekwencyjne w jej ramach
 - FIN — sygnalizuje chęć zakończenia sesji

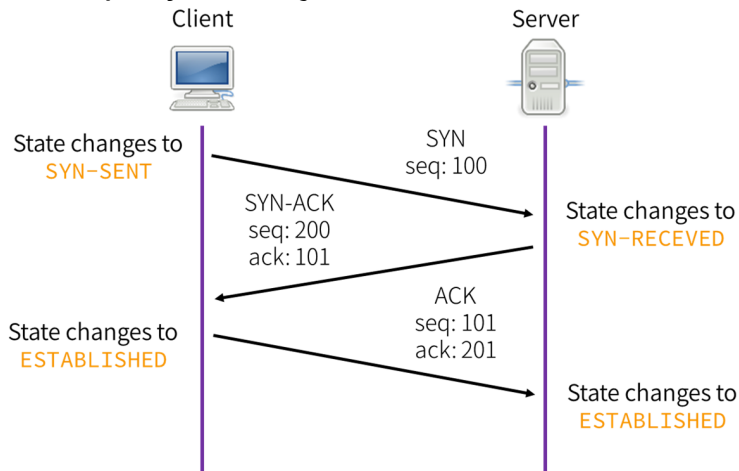
2.3 Protokół TCP. Działanie

Proces rozpoczynania sesji — *three way handshaking*:

1. Serwer oczekuje na chętnych do otwarcia połączenia.
Serwer w stanie LISTEN
2. Stacja wyrażająca chęć nawiązania sesji (klient) wysyła do serwera segment TCP z flagą SYN.
Klient w stanie SYN-SENT
3. Jeśli serwer godzi się na rozpoczęcie sesji, odsyła segment z flagą SYN i ACK. Oczekuje na potwierdzenie od klienta.
Serwer w stanie SYN-RECEIVED
4. Klient odsyła ACK. Sesja zostaje ustanowiona.
Klient i serwer w stanie ESTABLISHED

2.3 Protokół UDP. Działanie

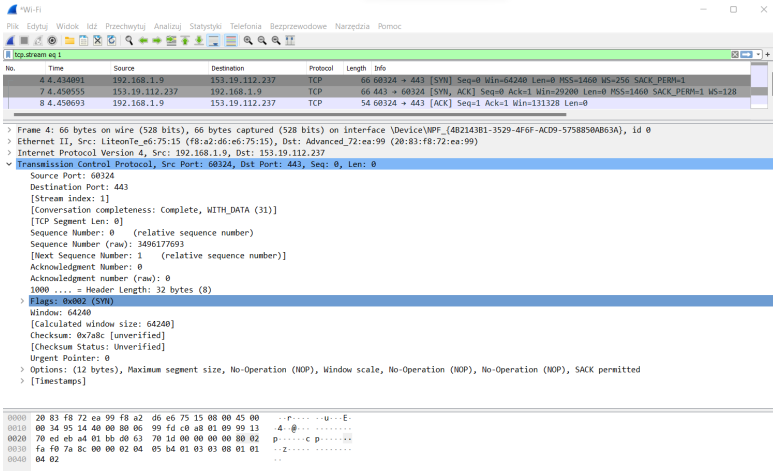
Proces rozpoczynania sesji:



Grafika: python.astrotech.io

2.3 Protokół TCP. Działanie

Proces rozpoczynania sesji:



The image shows a Wireshark packet capture of a TCP SYN packet. The packet list pane shows three packets: a SYN packet (66 bytes), a SYN-ACK packet (66 bytes), and an ACK packet (54 bytes). The packet details pane for the first packet (Frame 4) shows the following information:

- Frame 4: 66 bytes on wire (528 bits), 66 bytes captured (528 bits) on interface \Device\NPF_{4B2143B1-3529-4F6F-ACD9-5758850AB63A}, id 0
- Ethernet II, Src: LiteonTe_e6:75:15 (f8:a2:d6:e6:75:15), Dst: Advanced_72:ea:99 (20:83:f8:72:ea:99)
- Internet Protocol Version 4, Src: 192.168.1.9, Dst: 153.19.112.237
- Transmission Control Protocol, Src Port: 60324, Dst Port: 443, Seq: 0, Len: 0
 - Source Port: 60324
 - Destination Port: 443
 - [Stream index: 1]
 - [Conversation completeness: Complete, WITH_DATA (31)]
 - [TCP Segment Len: 0]
 - Sequence Number: 0 (relative sequence number)
 - Sequence Number (raw): 3496177693
 - [Next Sequence Number: 1 (relative sequence number)]
 - Acknowledgment Number: 0
 - Acknowledgment number (raw): 0
 - 1000 = Header Length: 32 bytes (8)
 - Flags: 0x002 (SYN)
 - Window: 64240
 - [Calculated window size: 64240]
 - Checksum: 0x7a8c [unverified]
 - [Checksum Status: Unverified]
 - Urgent Pointer: 0
 - Options: (12 bytes), Maximum segment size, No-Operation (NOP), Window scale, No-Operation (NOP), No-Operation (NOP), SACK permitted
 - [Timestamps]

The packet bytes pane shows the raw data of the packet, including the Ethernet II header, IP header, and TCP header.

2.3 Protokół TCP. Działanie

W trakcie trwania sesji — proces potwierdzania:

1. Stacja wysyłająca rozpoczyna wysyłanie danych — pierwszy segment posiada numer sekwencyjny 0, a dane mają X bajtów.
2. Stacja odbierająca odebrała segment o numerze sekwencyjnym 0. Potwierdza odebranie tych danych, wysyłając segment z ustawioną flagą ACK z numerem potwierdzenia informującym, którego bajtu danych teraz spodziewa się odebrać: $0+X+1$
3. Jeśli stacja wysyłająca nie otrzyma potwierdzenia w określonym czasie, dokonuje retransmisji tego fragmentu danych.

2.3 Protokół TCP. Działanie

W trakcie trwania sesji — sterowanie przepływem:

1. Podczas nawiązywania sesji, urządzenia uzgodniły rozmiar okna odbiorczego — ilość bajtów, jaka zmieści się w ich buforach odbiorczych — przykładowo, 10 000 bajtów.
2. Stacja wysyłająca wysłała przykładowo 2000 bajtów. Wie, że teraz może wysłać jedynie 8000 bajtów — o ile nie otrzyma potwierdzenia, że wysłane przez nią dane zostały poprawnie przetworzone.
3. W sytuacji, gdy urządzenie odbierające nie nadąża z odbieraniem i przetwarzaniem danych, może w polu Rozmiar okna podać mniejszą wartość, wymuszając na nadawcy zwolnienie transmisji (częstsze oczekiwanie na potwierdzenie).

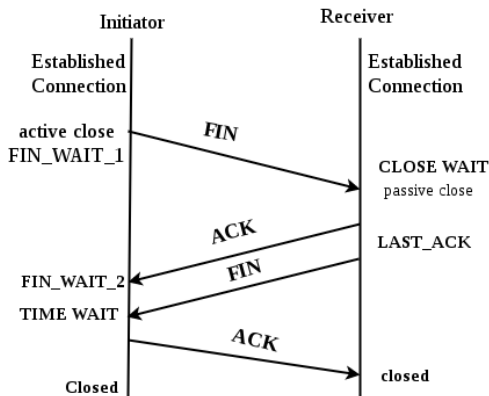
2.3 Protokół TCP. Działanie

Proces kończenia sesji:

1. Stacja inicjująca zakończenie sesji wysyła segment TCP z ustawioną flagą FIN.
Stacja inicjująca w stanie FIN-WAIT1
2. Stacja odbierająca odpowiada na FIN wysyłając ACK.
Stacja odbierająca w stanie CLOSE-WAIT, stacja inicjująca w FIN-WAIT2
3. Stacja odbierająca potwierdza chęć zakończenia sesji, wysyłając swój własny FIN.
Stacja odbierająca w stanie LAST-ACK, stacja inicjująca w TIME-WAIT.
4. Stacja inicjująca potwierdza (ACK) otrzymanie FIN od drugiej strony. Sesja jest zakończona po upływie czasu trwania stanu TIME-WAIT (max. 4 min).

2.3 Protokół TCP. Działanie

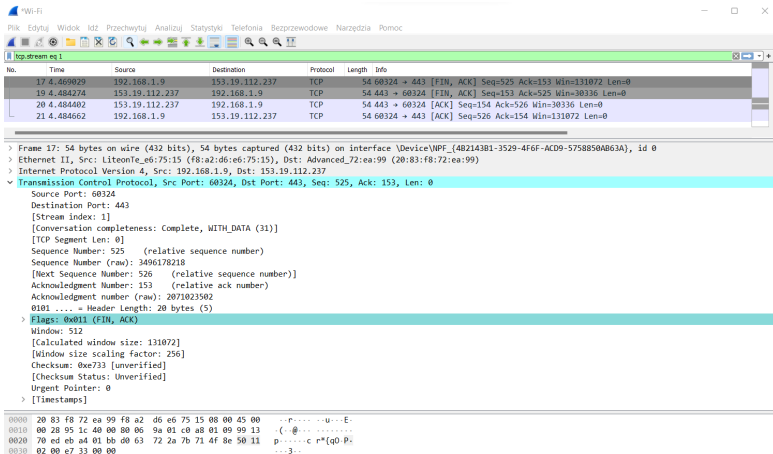
Proces kończenia sesji:



Grafika: geeksforgeeks.org

2.3 Protokół TCP. Działanie

Proces kończenia sesji:



The image shows a Wireshark packet capture of a TCP session termination. The packet list pane shows four packets. The selected packet is packet 17, a FIN, ACK from 192.168.1.9 to 153.19.112.237. The packet details pane shows the following information:

- Frame 17: 54 bytes on wire (432 bits), 54 bytes captured (432 bits) on interface \Device\NPF_{482143B1-3529-4F6F-ACD9-5758850AB63A}, id 0
- Ethernet II, Src: LiteonTe_e6:75:15 (f8:a2:d6:e6:75:15), Dst: Advanced_72:ea:99 (20:83:f8:72:ea:99)
- Internet Protocol Version 4, Src: 192.168.1.9, Dst: 153.19.112.237
- Transmission Control Protocol, Src Port: 60324, Dst Port: 443, Seq: 525, Ack: 153, Len: 0
 - Source Port: 60324
 - Destination Port: 443
 - [Stream index: 1]
 - [Conversation completeness: Complete, WITH_DATA (31)]
 - [TCP Segment Len: 0]
 - Sequence Number: 525 (relative sequence number)
 - Sequence Number (raw): 3496178218
 - [Next Sequence Number: 526 (relative sequence number)]
 - Acknowledgment Number: 153 (relative ack number)
 - Acknowledgment number (raw): 2071023502
 - 0101 = Header Length: 20 bytes (5)
 - Flags: 0x011 (FIN, ACK)
 - Window: 512
 - [Calculated window size: 131072]
 - [Window size scaling factor: 256]
 - Checksum: 0xe733 [unverified]
 - [Checksum Status: Unverified]
 - Urgent Pointer: 0
 - [Timestamps]

The packet bytes pane shows the raw data of the packet:

```
0000  20 83 f8 72 ea 99 f8 a2 d6 e6 75 15 08 00 45 00  ..P....-u...E-
0010  00 28 95 1c 40 00 80 06 9a 01 c0 a8 01 09 99 13  -(..@.....
0020  70 ed eb a4 01 bb 00 63 72 2a 7b 71 4f 8e 50 11  p.....c r*[q0.P
0030  02 00 e7 33 00 00                                     ...3..
```

2.3 Protokół TCP. Działanie

Podsumowanie — stany transmisji w TCP rozpoczynające sesję:

- **LISTEN** — nasłuchiwanie, czy nikt nie chce skorzystać z usługi; gotowość do nawiązania sesji
- **SYN-SENT** — początek nawiązywania połączenia (wysłano segment z flagą SYN, oczekiwanie na SYN ACK)
- **SYN-RECEIVED** — otrzymano SYN, wysłano SYN ACK w odpowiedzi, oczekiwanie na ACK
- **ESTABLISHED** — sesja nawiązana (odebrano ACK), można normalnie wymieniać dane

2.3 Protokół TCP. Działanie

Podsumowanie — stany transmisji w TCP kończące sesję:

- **FIN-WAIT1** — stacja inicjująca zakończenie sesji wysłała FIN, oczekuje na ACK
- **CLOSE-WAIT** — stacja odbiorcza odebrała FIN i odpowiedziała na niego, wysyłając ACK; oczekuje, aż sama wyśle FIN
- **FIN-WAIT2** — stacja inicjująca otrzymała potwierdzenie ACK na wysłany przez siebie FIN, oczekiwanie na FIN z drugiej strony
- **LAST-ACK** — stacja odbiorcza wysłała swój FIN, oczekuje na jego potwierdzenie
- **TIME-WAIT** — stacja inicjująca odebrała FIN od drugiej strony i odesłała na niego potwierdzenie, oczekuje, czy druga strona je odebrała przez maksymalnie 4 minuty

3. Protokół UDP

Cechy i działanie

3.1 Protokół UDP. Cechy

Cechy protokołu UDP (ang. *User Datagram Protocol*):

- **Bezpołączeniowość** — UDP nie tworzy sesji
- **Działanie *best effort*** — brak gwarancji niezawodności transmisji
- **Brak mechanizmów sterowania przepływem ani kontroli kolejności otrzymywanych datagramów**

Zalety:

- Brak obciążania łącza dodatkowymi informacjami kontrolno-sterującymi

Wady:

- Brak gwarancji niezawodności transmisji

3.1 Protokół UDP. Cechy

UDP sprawdza się tam, gdzie zalety TCP stają się jego wadami:

- tam, gdzie ważniejsza jest szybkość transmisji (VoIP, streaming)
- w komunikacji typu zapytanie-odpowiedź, gdzie brak odpowiedzi samoczynnie odbierane jest jako zachęta do retransmisji (np. DNS, DHCP),
- tam, gdzie zapewnienie potwierdzenia odbioru jest zaimplementowane na wyższych warstwach (TFTP).

3.1 Protokół UDP. Cechy

Przykładowe połączenie za pomocą UDP:

*Wi-Fi

Plik Edytuj Widok Idź Przechwytyj Analizuj Statystyki Telefon Bezprzewodowe Narzędzia Pomoc

udp

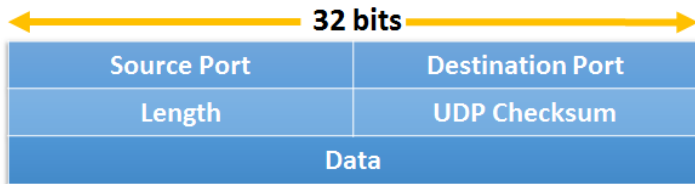
No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	192.168.1.9	192.168.1.1	DNS	70	Standard query 0x0004 A umg.edu.pl
2	0.006196	192.168.1.1	192.168.1.9	DNS	86	Standard query response 0x0004 A umg.edu.pl A 153.19.111.231
3	0.007516	192.168.1.9	192.168.1.1	DNS	70	Standard query 0x0005 AAAA umg.edu.pl
4	0.020765	192.168.1.1	192.168.1.9	DNS	117	Standard query response 0x0005 AAAA umg.edu.pl SOA dns1.umg.edu.pl

```
> Frame 1: 70 bytes on wire (560 bits), 70 bytes captured (560 bits) on interface \Device\NPF_{4B2143B1-3529-4F6F-ACD9-5758850AB63A}, id 0
> Ethernet II, Src: LiteonTe_e6:75:15 (f8:a2:d6:e6:75:15), Dst: Advanced_72:ea:99 (20:83:f8:72:ea:99)
> Internet Protocol Version 4, Src: 192.168.1.9, Dst: 192.168.1.1
√ User Datagram Protocol, Src Port: 52468, Dst Port: 53
  Source Port: 52468
  Destination Port: 53
  Length: 36
  Checksum: 0x66f3 [unverified]
  [Checksum Status: Unverified]
  [Stream index: 0]
  > [Timestamps]
  UDP payload (28 bytes)
> Domain Name System (query)
```

```
0000 20 83 f8 72 ea 99 f8 a2 d6 e6 75 15 08 00 45 00  ..-...-u...E:
0010 00 38 4b a7 00 00 80 11 6b b3 c0 a8 01 09 c0 a8  ..8K....k.....
0020 01 01 cc f4 00 35 00 24 66 f3 00 04 01 00 00 01  ....5.$f.....
0030 00 00 00 00 00 00 03 75 6d 67 03 65 64 75 02 70  ....u mg-edu.p
0040 6c 00 00 01 00 01 1...x
```

3.2 Protokół UDP. Nagłówek

Budowa nagłówka UDP:



UDP Segment Structure

Grafika: ipwithease.com

3.2 Protokół UDP. Nagłówek

Budowa nagłówka UDP:

- **Port źródłowy** — identyfikator usługi (aplikacji) źródłowej
- **Port docelowy** — identyfikator usługi (aplikacji) docelowej
- **Długość** — długość nagłówka UDP
- **Suma kontrolna** — umożliwia wykrycie błędów w nagłówku UDP

Można zauważyć, że nagłówek UDP jest bardzo uproszczony w porównaniu z nagłówkiem TCP