WYDZIAŁ INFORMATYKI I TELEKOMUNIKACJI

POLITECHNIKA POZNAŃSKA

Praca dyplomowa - Inżynierska

Stworzenie protokołu komunikacyjnego do wymiany danych w grze sieciowej - przykład edukacyjny

Jan Biały, 144334
Piotr Stawski, 144336

Promotor
prof. dr hab. inż. Grzegorz Danilewicz

POZNAŃ 2022

Spis Treści

1. Wstęp

1.1. Kontekst

1.2. Cel Pracy

2. Protokoły komunikacyjne

Modele Warstwowe

Pierwszym i zarazem najistotniejszym efektem prac normalizacyjnych w systemach otwartych jest wypracowanie jednolitego modelu sieciowego, przetwarzającego dane w rozproszonych systemach sieciowych od poziomu procesu aplikacji do przesłania danych poprzez linie transmisyjną.

2.1. ISO/OSI

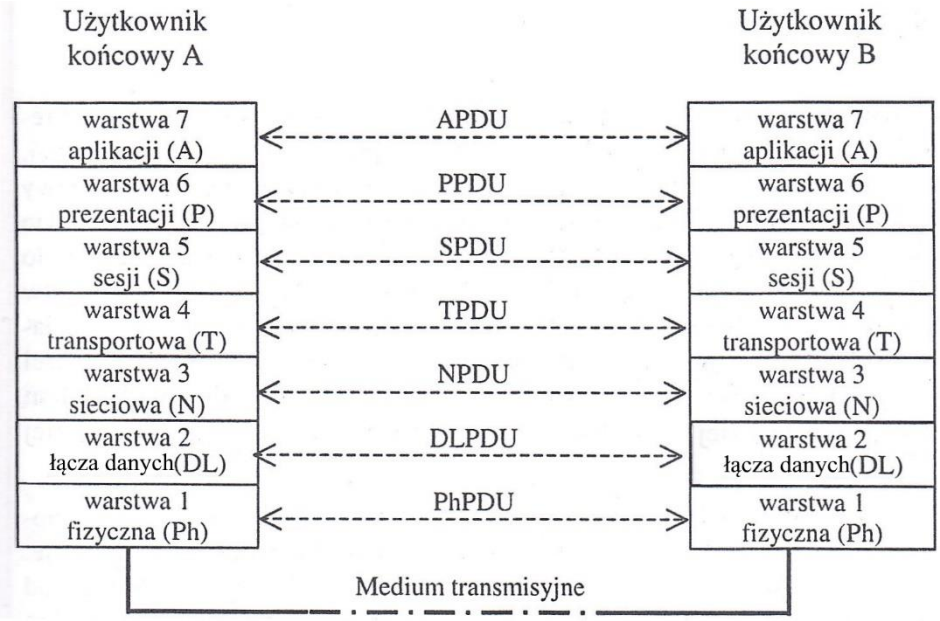
Jednym z takich modeli jest model OSI (pełna nazwa **ISO OSI RM** ang. *ISO Open Systems Interconnection Reference Model*), zwany **modelem odniesienia współdziałania systemów otwartych ISO/OSI**. Został on zaprezentowany w 1983 roku przez przedstawicieli największych firm komputerowych i telekomunikacyjnych. Jako międzynarodowy standard został przyjęty w 1984 roku. Model ten definiuje architekturę i ogólne zasady przetwarzania danych w sieci oraz w środowisku przetwarzania otwartego, to oznacza, że do takiego systemu może podłączyć się każdy kto spełnia ustandaryzowane ogólne zasady.

Głównym pojęciem, które przewija się w strukturze modelu OSI jest **warstwa.** Architektura Systemu OSI składa się z podsystemów tego samego poziomu. Jako podsystem rozumiemy hierarchiczny element systemu otwartego, który współpracuje jedynie z sąsiednim wyższym, bądź niższym poziomem hierarchii modelu sieciowego.

Model ISO/OSI składa się z siedmiu warstw o dokładnie określonych funkcjach i interfejsach łączących poszczególne warstwy. Każda warstwa posiada swoją nazwę oraz numer (od 1 do 7) liczoną od warstwy fizycznej (1) do warstwy aplikacji (7). Warstwowość modelu pozwala na grupowanie poszczególnych funkcjonalności w moduły, co czyni strukturę sieciową przejrzystą i dobrze zdefiniowaną.

Każdy system otwarty podłączony do sieci, funkcjonuje w niej jako zbiór stacji.

Stacja jest reprezentacją procesu (lub zestawu procesów) w obrębie systemu otwartego świadczącego usługę w sieci. Stacja jest to aktywny, programowy lub sprzętowy moduł, który może występować pod różnymi postaciami. Każda taka stacja może reagować na otrzymywane wiadomości od innych jednostek, bądź sama wysyłać wiadomości do innych stacji. Każda warstwa modelu może być rozpatrywana jako zbiór dowolnej ilości stacji znajdujących się w jednym systemie bądź w wielu systemach rozproszonych w sieci. Stacje ulokowane w najwyższej warstwie modelują procesy aplikacyjne. Wszystko co znajduje się poniżej tej warstwy modeluje procesy odpowiadające za dostarczenie usług OSI, najniższa warstwa modelu umożliwia podłączenie systemu otwartego do medium transmisyjnego.



Rysunek 2.1 Model odniesienia ISO/OSI

Źródło: Wymiana informacji w heterogenicznych systemach sieciwoych. E. Kosmulska-Bochenek

Usługa

Usługą określamy zbiór funkcji realizowanych przez określoną warstwę. Każda warstwa posiada swój zbiór funkcji, które świadczy warstwie wyższej. Każda warstwa świadczy usługi warstwie wyższej korzystając z usług dostarczonych jej poprzez warstwę bezpośrednio niższą, a co za tym idzie pośrednio świadczy usługi wszystkich warstw znajdujących się poniżej. Każda usługa jest odpowiedzialna za implementowanie własnej jasno określonej funkcjonalności.

Protokół

Komunikacja między usługami tego samego poziomu, jeżeli obie usługi znajdujące się na tym samym systemie otwartym (np. komputerze) jest stosunkowo prosta, ponieważ mogą bezpośrednio się ze sobą komunikować. Sytuacja zmienia się diametralnie, kiedy chcemy połączyć ze sobą usługi znajdujące się w różnych systemach otwartych niejednokrotnie oddalonych w stopniu znaczącym. Komunikacja między takimi systemami otwartymi odbywa się z wykorzystaniem ustanowionego zbioru reguł i formantów. Taki zbiór reguł i formatów (semantyka, składnia i zależności czasowe) obowiązujących w danej warstwie nazywa się **protokołem**. Każda stacja, aby świadczyć usługi i implementować określone funkcjonalności musi działać wedle tych reguł. Stacje wymieniają informację pomiędzy sobą za pomocą jednostek danych protokołów (PDU – *Protocol Data Unit)*, dostosowanych do warstwy w których funkcjonują (rys 2.1). PDU można podzielić na dwa zbiory informacji:

- Pierwszym z nich są *informacje sterujące protokołem* (PCI – *Protocol Control Information*), ten zbiór informacji podczas implementowania protokołu nazywamy nagłówkiem protokołu. Nagłówek ten jest tylko i wyłącznie interpretowany przez protokół, do którego należy, którego jest częścią. To on informuje jaka akcja powinna zostać podjęta w momencie, w którym otrzymaliśmy określone PDU.

- Drugim zbiorem informacji są *dane* które zostają przekazane do wyższej warstwy w celu obsłużenia usług znajdujących się wyżej w modelu systemu otwartego.

2.1.1. Warstwa aplikacji (7)

Warstwa aplikacji (*ang. Application layer*) jest najwyższą warstwą modelu, a co za tym idzie jest zawsze najbliżej użytkownika. Dzieje się tak, ponieważ to właśnie w tej warstwie działają aplikacje, programy, z których GUI (*ang. Graphical User Interface*) użytkownik ma bezpośredni wpływ na to co i na jakich warunkach zostanie przesłane poprzez zhierarchizowaną strukturę systemu otwartego. Może on chcieć wyświetlić stronę WWW (*ang. World Wide Web*) co będzie przyczyną wybrania w dalszym segmencie modelu protokołu TCP (*ang. Transmission Control Protocol*) oraz portu 80 w docelowym systemie otwartym. Poprzez programy działające w warstwie aplikacji użytkownik może wywołać akcje, które będą potrzebować różnorodnego obsłużenia na poziomie poszczególnych warstw modelu OSI.

W warstwie aplikacji występuję wiele protokołów, jednakże często zdarza się tak, że protokół nie działa indywidualnie tylko potrzebuje do prawidłowego działania innych protokołów (*ang. Interaction between network protocols*), przykładem takiego protokołu jest protokół HTTP.

Interakcja pomiędzy protokołami często polega na wspomożeniu protokołu wyższej warstwy o funkcjonalność jakiej ten nie posiada. Przykładem takiego połączenia jest para protokołów HTTP i TCP.

HTTP może być zaimplementowany zarówno po stronie klienckiej jak i serwerowej, ale HTTP nie posiada funkcjonalności dotarcia z jednego punktu do drugiego. Z pomocą przychodzą więc protokoły warstwy transportowej takie jak np. TCP, który zapewnia mechanizm wymiany informacji docierający do miejsca docelowego. Dodatkowo pojawia się protokół adresowania IP z pomocą którego informacje w postaci PDU będą mogły być przenoszone pomiędzy podsieciami i zarazem będą wiedziały z jakiego, i do jakiego miejsca chcą dotrzeć. W procesie poprawnej komunikacji interakcja pomiędzy protokołami jest bardzo ważna do prawidłowego funkcjonowania systemu komunikacji.

Do podstawowych protokołów warstwy aplikacji można zaliczyć takie protokoły jak:

- HTTP (*ang. Hypertext Tranfer Protocol*) – dzięki współpracy z siecią WWW (ang. World Wide Web) umożliwia przeglądanie stron internetowych. Klient w warstwie aplikacji wyposażony w przeglądarkę internetową dzięki protokołowi HTTP może połączyć się z serwerem udostępniającym treść i przeglądać je na własnym komputerze. W praktyce proces ten wygląda następująco:
  - użytkownik wysyła żądanie PDU w nagłówku, którego znajduje się polecenie *GET* co oznacza dla serwera docelowego, że użytkownik zażądał danych znajdujących się na jego serwerze.
  - Następnie serwer odsyła użytkownikowi informację najczęściej z kodem 200, który oznacza, że żądanie użytkownika zostało obsłużone w sposób prawidłowy. Serwer przesyła także zawartość jaką użytkownik zechciał otrzymać wysyłając polecenie *GET*.
- DHCP (ang. Dynamic Host Configuration Protocol) – protokół służący do automatycznego przydzielania adresu IP urządzeniowi sieciowemu oraz innych adresów niezbędnych do prawidłowego funkcjonowania urządzenia w sieci. Takimi adresami są adres bramy domyślnej czy adres serwerów DNS (ang. Domain Name System).

- SMTP (*ang. Simple Mail Transport Protocol*) – protokół służący do przesyłania poczty elektronicznej, wyłącznie w postaci tekstowej, najczęściej współpracuje z protokołem POP3.
- POP3 (*ang. Post Office Protocol*) – odpowiada za odbiór poczty z serwera, współpracuje z protokołem TCP.
- SSL (*ang. Secure Sockets Layer*) – wprowadza funkcjonalność szyfrowania komunikacji pomiędzy urządzeniami końcowymi.
- DNS (ang. Domain Name System – wprowadza funkcjonalność odwzorowywania nazw domenowych na adresy IP.
- FTP (ang. File Transfer Protocol) – wprowadza funkcjonalność przesyłania i odbierania plików z urządzeń znajdujących się w sieci.

Warstwa aplikacji jest więc, początkiem, ale też i końcem komunikacji pomiędzy aplikacjami użytkowanych przez użytkowników końcowych.

2.1.2. Warstwa Prezentacji

Warstwa prezentacji (*ang. Presentation layer*) odpowiada za prezentowanie danych w sieci. Warstwa aplikacji przesyła dane warstwie prezentacji, która jest odpowiedzialna za przetłumaczenie tych danych do postaci zgodnej ze specyfikacją OSI RM. Następnie jeżeli jest to wymagane szyfruje te dane lub kompresuje je. Celem warstwy prezentacji jest doprowadzenie do sytuacji, aby dane przekazywane w komunikacji miały wspólny format.

2.1.3. Warstwa Sesji

Warstwa sesji (*ang. Session layer*) jest odpowiedzialna za zestawianie sesji pomiędzy urządzeniami końcowymi oraz jej zarządzaniem, jeżeli sesja jest już nawiązana. Warstwa sesji wykorzystując odpowiednie protokoły zapewnia dwa rodzaje komunikacji: *połączeniową* i *bezpołączeniową*. W warstwie sesji dochodzi też do zabezpieczenia przed ponowną transmisją poprzez wprowadzenie do przesyłanych danych swego rodzaju punktów kontrolnych. Kiedy sesja zostaje zerwana, dane nie muszą być retransmitowane w całości, ale od miejsca ostatniego punktu kontrolnego.

2.1.4. Warstwa transportowa

Warstwa transportowa (*ang. Transport layer*) należy do najważniejszych warstw modelu ISO/OSI. To ona odpowiada za sposób w jaki dane są przesyłane. Dane wysyłane przez sieć nigdy nie są wysyłane w całości. Taka transmisja byłaby nieefektywna, ponieważ w przypadku zerwania połączenia dane musiały by być transmitowane od początku. Dlatego też dane dzielone są na mniejsze kawałki. Proces ten nazywamy *segmentowaniem (ang. Segmenting).* Dzięki temu mechanizmowi znacznie poprawia się szybkość przesyłania danych w sieci, poprawia się wydajność całego systemu oraz zwiększa się bezpieczeństwo

przesyłanych danych. Podczas stosowania mechanizmu segmentacji, pakiety powinny być po stronie odbiorczej odpowiednio odebrane a następnie przetworzone.

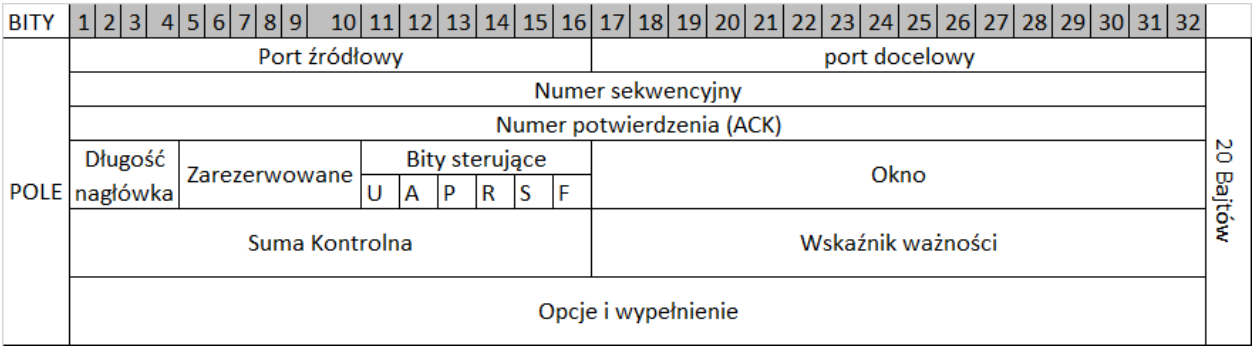
W sieciach komputerowych a zwłaszcza w sieci Internet, gdzie występuję bardzo duża złożoność połączeń, może się zdarzyć tak, że pakiety docierają do odbiorcy w kolejności innej niż zostały wysłane przez nadawcę. Zjawisko takie jest dość powszechne, dlatego też stworzono mechanizm porządkowania pakietów po stronie odbiorczej. Dlatego też każdy pakiet wysyłany przez nadawcę jest oznaczany numerem sekwencyjnym (*ang. Sequence number*). Warstwa transportowa jest odpowiedzialna ta to, aby każdy pakiet wysłany przez nadawcę został dostarczony do miejsca docelowego w odpowiedniej kolejności. W warstwie transportowej występują numery portów są to swego rodzaju adresy, na których pracują aplikacje. To właśnie w tej warstwie zostaje podjęta decyzja którego protokołu wybrać TCP (ang. *Transmission Control Protocol*) czy UDP (ang. *User Datagram Protocol*).

2.1.4.1. TCP

Priorytetowym zadaniem protokołu TCP (ang. *Transmission Control Protocol*) jest poprawne dostarczenie pakietu do miejsca docelowego. TCP jest uznawany za protokół pewny. Oznacza to, że TCP ma zaimplementowane mechanizmy zapewniające dotarcie danych do odbiorcy. Podczas segmentowania każdy segment dostaje indywidualny numer sekwencyjny wykorzystywany do prawidłowego złączenia ich po stronie odbiorcy. TCP na samym początku nawiązuje połączenie z odbiorcą, a co za tym idzie jest właśnie protokołem połączeniowym, który najpierw zestawia połączenie i uzgodnienie wszystkich parametrów transmisji np. prędkości wysyłania dostosowanej do prędkości akceptowalnej dla odbiorcy.

Podczas przesyłania danych przy pomocy protokołu TCP dane po każdym wysłaniu muszą zostać potwierdzone. Jeżeli zostały potwierdzone, oznacza to, że zostały prawidłowo dostarczone do adresata. Protokół ten zapewnia mechanizm retransmisji danych w przypadku, jeżeli nie zostanie przesłane potwierdzenie od adresata. Mechanizm ten jest bardzo ważny podczas przesyłania wrażliwych danych np. komunikacji email, korespondencji bankowej. Podczas komunikacji TCP dwa urządzenia końcowe zestawiają między sobą sesję, dzięki której oba urządzenia przygotowują się do wymiany informacji między sobą. W momencie przekazywania danych, jeżeli jedno z urządzeń nie nadąża z przetwarzaniem odbieranych danych zostaje aktywowany mechanizm kontroli przepływu (ang. Flow control) w momencie działania mechanizmu urządzenia dostosowują nowe warunki połączenia.

TCP jest tzw. protokołem stanowym (ang. stateful) ustanawiając sesje, może on kontrolować które informacje zostały już przez niego wysłane oraz które zostały już potwierdzone. TCP posiada 20-bajtowy nagłówek, w którym znajdują się informację niezbędne do prawidłowego funkcjonowania komunikacji.



Rysunek 2.2 Nagłówek TCP

Pole *port źródłowy* (*ang. Source port*) ma długość 16 bitów. W tym polu znajduje się informacja dotycząca portu aplikacji z którego zostały wysłane dane.

Kolejnym polem jest *port docelowy (ang. Destination port),* w tym polu znajduje się port urządzenia odbiorcy, na który mają zostać wysłane dane. Dzięki temu polu warstwa transportowa stacji docelowej będzie wiedziała, gdzie przekazać odebrane dane.

Pole *numer sekwencyjny (ang. sequence number)* ma długość 32 bitów i jest to jedno z najważniejszych pól nagłówka TCP, ponieważ odgrywa istotną rolę w porządkowaniu przychodzących pakietów do odbiorcy. Numer ten określa numer porządkowy pierwszego odebranego bajtu każdego odebranego segmentu danych.

Kolejnym polem jest 32 bitowy *Numer potwierdzenia (ang. ACK acknowledgement number)*. To pole zawiera informację na temat kolejnego bajtu jaki jest oczekiwany przez odbiorcę. Dzięki tym dwóm polom cały mechanizm porządkowania segmentów odbywa się w sposób niemylący się.

Długość nagłówka (ang. header length) jest to pole 4 bitowe, które zawiera informację o długości nagłówka TCP z tego powodu, iż nagłówek może mieć zmienną długość, dlatego to pole informuję odbiorcę jakiej długości pakietu ma się spodziewać. Długość nagłówka TCP zależy od ewentualnych dodatkowych danych w polu *Opcje*.

Zarezerwowane bity (ang. reserved bits) to pole o długości 6 bitów które nie jest używane.

Pole Bity sterujące (flag) (ang. control bits – flags) zawiera tak naprawdę sześć pól, które mogą zajmować wartości 0 lub 1. Znaczenie kolejnych flag są następujące:

- U (URG) oznacza ważność (ang. urgent). Oznacza to czy pole ma większy priorytet w kolejności przetwarzania pakietów przez odbiorcę.

- A (ACK) oznacza potwierdzenie (ang. acknowledgment) otrzymanego segmentu. Po wysłanym segmencie bądź grupie segmentów następuje potwierdzenie ich otrzymania.

- P (PSH) informuje, że odbiorca powinien przekazać dane *(ang. push)* od razu do warstwy wyższej.

- R (RST) to flaga resetująca *(ang. reset)*, która jest wykorzystywana do natychmiastowego resetowania danej sesji.

- S (SYN) to flaga, która występuje podczas nawiązywania połączenia i rozpoczyna proces synchronizacji *(ang. synchronization)* klienta i serwera. Jeżeli klient chce rozpocząć połączenie, wysyła segment z ustawioną flagą SYN, informując drugą stronę, że chce rozpocząć nawiązywanie sesji za pomocą TCP.

- F (FIN) to flaga zakończenia (*ang. finish*) która jest ustawiana w momencie jak klient kończy połączenie TCP. Druga strona też odpowiada w tym momencie pakietem FIN i połączenie pomiędzy nimi zostaje zakończone.

Pole *Okno* (ang. window) informuje o ilości danych jakie mogą zostać przesłane bez konieczności ich potwierdzenia. Oznacza to więc również ilość danych jaka może być przesłana do odbiorcy.

Pole *suma kontrolna (ang. checksum)* jest polem kontrolnym 16-bitowym w którym TCP umieszcza obliczoną wartość podsumowującą segment. Strona odbierająca w momencie odebrania segmentu również wykonuje obliczenia, aby zweryfikować czy przesyłana wiadomość nie zawiera błędów.

Wskaźnik *ważność (ang. urgent pointer)* to pole wykorzystywane do nadania priorytetu danych wysyłanych za pomocą TCP.

Ostatnie pole nagłówka TCP to pole, *Opcje i wypełnienie (ang. options and completion)*, służy do umieszczania dodatkowych informacji związanych z przesyłanymi danymi.

Kiedy jest ustanawiana sesja TCP ustawiany jest także początkowy *numer sekwencyjny (ang. Initial Sequence Number - ISN).* Jest to liczba losowa, reprezentująca wartość początkową bajtów przesyłanych do aplikacji odbierającej dane. W momencie przyjścia danych do odbiorcy, liczba ta jest zwiększana o liczbę przesyłanych bajtów. Dane następnie są składowane w buforze odbiorczym a następnie przekazywane do warstwy aplikacji. Żaden protokół nie jest w stanie zagwarantować, że dane dotrą do odbiorcy. W tym celu TCP ma mechanizmy, dzięki którymi jest w stanie zarządzać utraconymi pakietami. Jeżeli segment nie dotarł do celu to TCP ponownie przesyła je do odbiorcy.

2.1.4.2. UDP

UDP *(ang. User Datagram Protocol)* jest protokołem zawodnym i niepewnym, gdyż jest protokołem bezpołączeniowym co oznacza, że w momencie jak dane nie dotrą do adresata a nadawca nie dowie się, że dane które wysłał nigdy nie dotarły. W przypadku UDP nie jest to jednak wadą, ponieważ niektóre rodzaje komunikacji nie mogą otrzymywać potwierdzenia po każdorazowym wysłaniu wiadomości. Przykładem takiego rozwiązania jest telefonia IP. Jeżeli podczas rozmowy telefonicznej, któraś z danych zostanie utracona, protokół je pominie. UDP nie prześle ich ponownie do odbiorcy. Odbiorca może odczuć ich brak, ale jeżeli problem ten pojawia się rzadko, to w zasadzie nic drożnego się nie dzieje. W momencie wykorzystania protokołu TCP do takiego celu utrata niewielkiej ilości

danych spowodowałaby konieczność przesłania powtórnie całej transmisji. Wywołałoby to wieli zamęt w rozmowie a nawet uniemożliwiałoby to rozmowę. Typowym przykładem wykorzystania protokołu UDP to TFTP (ang. *Trivial File Transfer Protocol)*, SNMP (*ang. Simple Network Management Protocol*), DNS (ang. Domain Name System).

Protokół UDP nie zapewnia niezawodności i kontroli przepływu jak TCP. Nie zapewnia dostarczenia pakietów do odbiorcy, nie wymaga potwierdzenia otrzymania danych. Jest więc protokołem bezpołączeniowym. Jest jednak szybszy i bardziej wydajny od TCP, dlatego jest przeznaczony do innych zastosowań.

BITY	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 32	
POLE	Port źródłowy		port docelowy		8-bajtów
	Długość		Suma kontrolna		
	Dane warstwy aplikacji				

Rysunek 2.3 Nagłówek UDP

Pole *port źródłowy* i *port docelowy* pełnią identyczne funkcje jak ich odpowiedniki w protokole TCP.

Pole Długość informuję o długości całego segmentu UDP. Mówi ono o ilości bajtów jakie zawiera segment.

Pole suma kontrolna podobnie jak w TCP służy do określania czy wystąpił błąd podczas przesyłania segmentu.

O tym którego protokołu użyć – TCP czy UDP – decyduje warstwa wyższa. Wybór ten jak widać może mieć decydujące znaczenie dla jakości transmisji.

Ponieważ urządzenie jednocześnie korzysta z wielu usług pracujących na różnych portach stąd właśnie protokół transportowy potrafi rozróżnić do jakiej aplikacji przekazać odebrane dane.

2.1.5. Warstwa sieci

Warstwa sieci (*ang. Network layer*) odbiera segmenty z warstwy transportowej, a następnie są przeprowadzane działania przygotowujące wysłanie danych do adresata. W warstwie sieci spotykamy się z innym typem adresowania niż w warstwie transportowej. W warstwie sieci występuję m.in. protokół IP, który działa z wykorzystaniem adresów IP nadawcy oraz odbiorcy. Adresy te są wykorzystywane podczas przesyłania pakietów pomiędzy sieciami i podsieciami w celu dotarcia pakietu do miejsca docelowego. W warstwie wyższej został przygotowany segment określający numery portów z jakiego i na jaki ma dotrzeć segment. Aby segment mógł zostać prawidłowo dostarczony do odbiorcy potrzebuje dodatkowych informacji adresowych.

Najważniejszymi protokołami służącymi do adresowania segmentów w sieci są IPv4 oraz IPv6 (*ang. Internet Protocol*). Nie są to jednak jedyne protokoły działające w warstwie sieciowej. Aby urządzenia sieciowe znajdujące się w różnych podsieciach mogły komunikować się między sobą wykorzystują one inne protokoły, np. protokoły trasowania (ang. Routing protocols). Do takich protokołów należą m.in. OSPF, EIGRP. Rozwiązaniem wspierającym cała komunikację jest również ICMP (*ang. Internet Control Message Protocol*).

Każde urządzenie chcące się komunikować z urządzeniami znajdującymi się w innych sieciach czy też podsieciach potrzebuje adresu IP urządzenia docelowego. Informacje te znajdują się w nagłówku IP.

Protokół IP został zaprojektowany do adresowania urządzeń i jest on protokołem bezpołączeniowym (*ang. connectionless*) co oznacza, że nie gwarantuje dostarczenia pakietu do miejsca docelowego. Pakiet IP zawiera w sobie zasadniczo dwa pola nagłówek (*ang. header*) oraz ładunek (*ang. payload*).

Nagłówek IP jest bardzo istotnym elementem w kontekście komunikacji w sieci. To właśnie on zawiera adres źródłowy nadawcy i docelowy adres odbiorcy oraz informację jak warstwa wyższa ma zareagować na otrzymane dane.

2.1.5.1. IPv4

Adresowanie IPv4 jest adresowaniem wykorzystującym 32-bitowy adres IP składający się z czterech oktetów dzielących adres na mniejsze fragmenty przyjmujące wartości od 0 do 255 oddzielone od siebie kropkami. Każde urządzenie znajdujące się w sieci komputerowej potrzebuje adresu IP który je identyfikuje. Adres IPv4 składa się z dwóch części których na pierwszy rzut oka nie widać. Dopiero w momencie, gdy adres IP zostanie zestawiony z maską podsieci można zauważyć część adresu określającą fragment sieci oraz fragment adresu identyfikujący konkretne urządzenie znajdujące się w danej podsieci. Maska podsieci jest to ciąg jedynek który może mieć maksymalnie 32bity rozpoczynający się od lewej strony maski.

<table>
<tr><th>BITY</th><th>0</th><th>1</th><th>2</th><th>3</th><th>4</th><th>5</th><th>6</th><th>7</th><th>8</th><th>9</th><th>10</th><th>11</th><th>12</th><th>13</th><th>14</th><th>15</th><th>16</th><th>17</th><th>18</th><th>19</th><th>20</th><th>21</th><th>22</th><th>23</th><th>24</th><th>25</th><th>26</th><th>27</th><th>28</th><th>29</th><th>30</th><th>31</th></tr>
<tr><td>0</td><td colspan="4">Wersja</td><td colspan="4">Długość nagłówka</td><td colspan="6">Usługi zróżnicowane</td><td colspan="2">ECN</td><td colspan="16">Całkowita długość</td></tr>
<tr><td>32</td><td colspan="16">Numer identyfikacyjny</td><td colspan="3">Flagi</td><td colspan="13">Przesunięcie</td></tr>
<tr><td>64</td><td colspan="8">Czas życia</td><td colspan="8">Protokół warstwy wyższej</td><td colspan="16">Suma kontrolna nagłówka</td></tr>
<tr><td>96</td><td colspan="32">Adres źródłowy IP</td></tr>
<tr><td>128</td><td colspan="32">Adres docelowy IP</td></tr>
<tr><td>160</td><td colspan="19">Opcje IP</td><td colspan="13">Wypełnienie</td></tr>
<tr><td>192</td><td colspan="32">Dane</td></tr>
</table>

Rysunek 2.4 Nagłówek pakietu IPv4

Pierwszym polem nagłówka IP jest *Wersja (ang. Version)* zawiera ono informację o tym która wersja protokołu IP jest wykorzystywana.

Pole *długość nagłówka* (*ang. Header length*) określa długość nagłówka IPv4.

Ośmiobitowe pole *usługi zróżnicowane DS* (ang. Differentiated services) wykorzystywane jest do obsługi mechanizmu QoS (ang. Quality of service).

Szesnastobitowe pole *Długość pakietu (ang. Total length)* określa całkowitą długość pakietu liczoną łącznie z nagłówkiem i danymi znajdującymi się w pakiecie. Dzięki temu polu odbiorca wie, kiedy skończyć przetwarzanie otrzymanego pakietu.

Pole *Flagi (ang. flags)* określa, czy pakiet mógł zostać poddany fragmentacji. Jeżeli bit DF (ang. Don't Fragment) jest ustawione, oznacza to, że pakiet został odebrany w całości. Jeżeli pole MF (ang. More Fragment) jest ustawione oznacza to, że odebrany fragment nie jest ostatnią częścią odbieranego pakietu.

Pole *Przesunięcie (ang. Fragment Offset)* określa miejsce odebranego fragmentu w momencie, gdy odebrany fragment jest elementem pakietu podzielonego na mniejsze fragmenty.

Pole *TTL* (ang. Time To Live) oznacza maksymalną liczbę urządzeń, przez które może zostać przesłany pakiet. Przy każdym przesłaniu pakietu przez router pole to jest pomniejszane o 1. W momencie jak to pole osiągnie wartość 0 pakiet jest odrzucany i nie jest przesyłany po sieci w nieskończoność.

Pole *protokół* (ang. Protocol) oznacza charakter przesyłanych danych, dzięki tej wiadomości warstwa wyższa zostaje poinformowana w jaki sposób ma obsłużyć odebrane dane podczas procesu dekapsulacji.

Pole *Suma kontrolna nagłówka (ang. header checksum)* pozwala stwierdzić, czy nagłówek został przesłany poprawnie, jest sprawdzany przy każdorazowym analizowaniu nagłówka.

Pole *Adres źródłowy (ang. Source IP Address)* jest polem 32 bitowym określający adres IPv4 nadawcy.

Pole *Adres docelowy (ang. Destination IP Address)* jest polem 32 bitowym zawierający adres IPv4 odbiorcy.

Pole *Opcje (ang. Options)* nieobowiązkowe pole charakteryzujące dodatkowe zachowanie względem odebranego pakietu IP.

Pole wypełnienie (ang. Padding) jest polem opcjonalnym wypełniającym nagłówek tak aby jego wielkość była wielokrotnością 32, w tym celu wypełniane jest zerami.

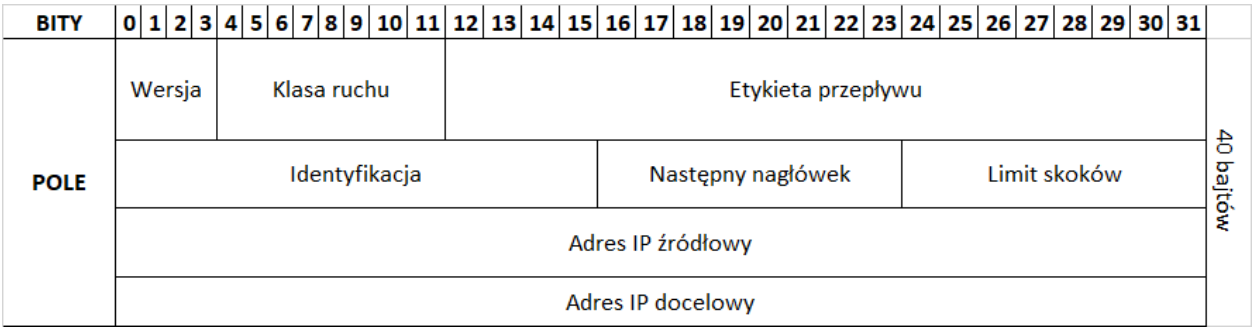
2.1.5.2. IPv6

Protokół IPv4 jest nadal najpowszechniejszym protokołem na świecie, jednakże specjaliści prorokowali już wiele razy jego koniec, ze względu na kończące się możliwości adresowania nowych urządzeń dodawanych do sieci. Ze względu na brak nowych adresów IPv4 konieczne jest omijanie tego problemu poprzez rozwiązania programowe takie jak *NAT (ang. Network Address Translation)* lub poprzez przekierowywanie portów. Zastosowanie tych dwóch mechanizmów zmniejsza szybkość transferu informacji, gdyż każdy pakiet musi zostać przeanalizowany przed przesłaniem dalej.

W związku z tym problemem między innymi został wynaleziony protokół IPv6, aby zabezpieczyć dostępność i możliwość dalszego adresowania urządzeń. Mechanizm NAT opóźnił przejęcie sieci przez nowy standard adresacji IPv6, ale przez to mechanizm ten uniemożliwia bezpośredniego komunikowania się urządzeń między sobą. Mechanizm ten polega na „ukryciu" adresów znajdujących się w danej podsieci pod jednym adresem publicznym.

Adresacja IPv6 została wynaleziona już w latach dziewięćdziesiątych przez *IETF (Internet Engineering Task Forces)*. Wielkim atutem adresacji IPv6 nad IPv4 jest zwiększenie pola adresu z 32 do 128 bitów.

Razem z wprowadzeniem adresacji IPv6 zmieniła się również forma nagłówka pakietu IP jest ona zdecydowanie prostsza niż w przypadku IPv4.



Rysunek 2.5 Nagłówek IPv6

Pole *wersja (ang. Version)* identyfikuje jakiej wersji protokołu IP należy użyć podczas analizowania pakietu

Pole *Klasa ruchu (ang. Traffic class)* pełni identyczną funkcję co pole DS w nagłówku IPv4, jest więc związane z mechanizmem QoS.

Pole *następny nagłówek (ang. next header)* określa rodzaj danych jakie są przesyłane w pakiecie IP

Pole *limit skoków (ang. Hop limit)* jest to pole 8-bitowe zawierające informację o ilości skoków jakie może wykonać pakiet. Przy każdorazowym przeskoku pole to jest pomniejszane o 1. W momencie, gdy to pole osiągnie wartość 0 pakiet zostaje odrzucony.

Pole *adres źródłowy (ang. Source Address)* jest w polem 128 bitowym przechowującym adres nadawcy pakietu.

Pole *adres docelowy (and. Destination Address)* jest polem 128 bitowym przechowującym adres odbiorcy pakietu.

2.1.6. Warstwa łącza danych

Warstwa *łącza danych (ang. data link layer)* umieszcza pakiety w ramkach, które wykorzystują do adresacji adresy *MAC (ang. Medium Access Control)* które są adresami fizycznymi urządzeń. Warstwę łącza danych dzielimy na dwie podwarstwy: *MAC (ang. Media Access Control)* oraz *LLC (ang. Logical Link Control).*

Podwarstwa MAC określa sposób w jaki zostają przesyłane dane przez medium sieciowe i jest oparta na adresacji fizycznej natomiast podwarstwa LLC identyfikuje protokoły i występującą w nich enkapsulację danych.

Proces Enkapsulacji polega na dodaniu do fragmentu danych nagłówka oraz pola końcowego. W warstwie łącza danych dodawane są pola przechowujące źródłowe i docelowe adresy fizyczne (MAC).

Podwarstwa MAC odpowiedzialna jest za dostęp do medium transmisyjnego tak aby w tym samym momencie korzystało z niego tylko jedno urządzenie. Mechanizmem sterującym jest tutaj *CSMA\CD (ang. Carrier Sense Multiple Access Collision Detection).*

64 BITY	48 BITÓW	48 BITÓW	16 BITÓW	46 - 1500 BAJTÓW	32 BITY
PREAMBUŁA	ADRES DOCELOWY	ADRES ŹRÓDŁOWY	TYP/DŁUGOŚĆ	DANE	CRC

Rysunek 2.6 Ramka Ethernet

Preambuła (ang. preambule) jest to pole odpowiedzialne za wykrycie przez urządzenie rozpoczęcia komunikacji

Pole *Adres Docelowy (ang. Destination Address)* pole zawierające 48-bitowy adres fizyczny urządzenia, do którego jest wysyłana ramka

Pole Adres Źródłowy (ang. Source Address) pole zawierające 48-bitowy adres fizyczny urządzenia wysyłającego ramkę

Pole *typ/długość (ang. type/length)* ma długość 16 bitów odpowiedzialne jest za poinformowanie urządzenia, do którego jest wysyłana ramka, jakie dane są w niej przesyłane.

Pole *dane (ang. data)* ma minimum 46 bajtów a maksymalnie 1500 bajtów. W tym miejscu umieszczę są informację przesłane z warstwy wyższej modelu.

Ostatnie pole ramki CRC *(ang. Cyclic Redundancy Check)* jest to pole określające czy ramka została przesłana w sposób niezmieniony. Zapewnia ono sprawdzenie integralności przesyłanych danych.

2.1.7. Warstwa fizyczna

Warstwa fizyczna (ang. Physical layer) jest ostatnią najniższą warstwą modelu OSI, która przesyła dane przez medium transmisyjne.

Warstwa ta jest odpowiedzialna jedynie za przesyłanie informacji przez dane medium transmisyjne. Warstwa ta obsługuje różne typy medium transmisyjnego, takie jak: *kabel miedziany*, *światłowód* czy *też fale radiowe.* Ponieważ transmisja odbywa się za pośrednictwem przesyłania zer i jedynek warstwa fizyczna konwertuje otrzymane dane do postaci strumieni binarnych. W momencie odbioru musi ona zamienić otrzymane impulsy czy to elektryczne, optyczne czy też radiowe na postać danych obsługiwanych przez warstwę wyższą modelu.

2.2. TCP/IP

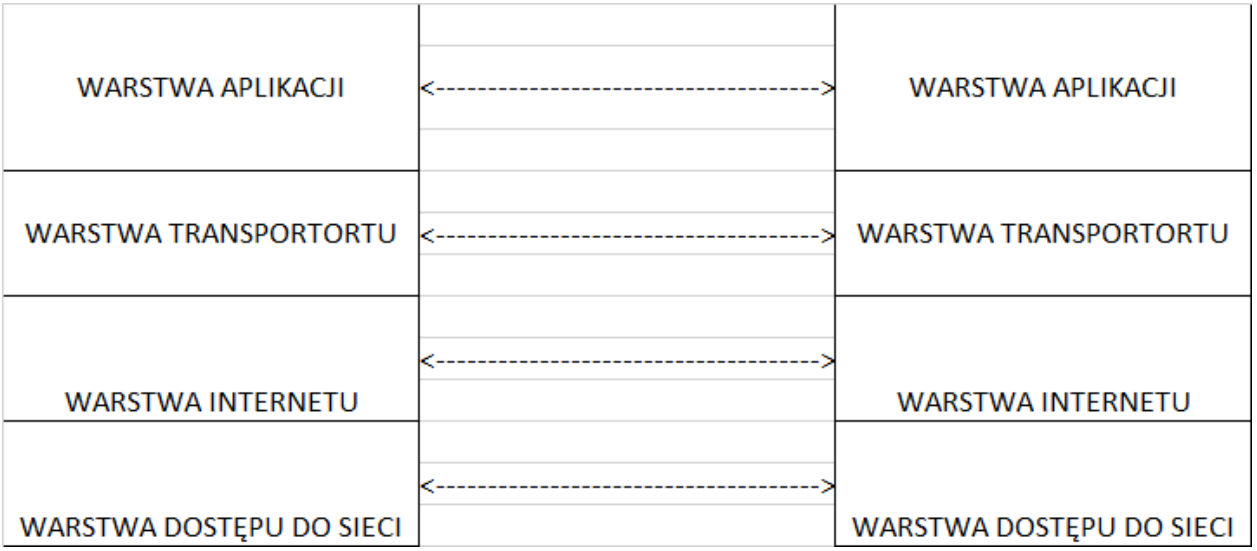
Model TCP/IP jest swego rodzaju odzwierciedleniem sieci komputerowej oraz umiejscowienia danych urządzeń na odpowiednim poziomie modelu.

TCP/IP złożony jest z 4 warstw, a każda warstwa opowiada co dzieje się na konkretnym etapie wymiany informacji pomiędzy dwoma urządzeniami. Dzięki takiemu pogrupowaniu informacji można szybko przyporządkować dany problem w komunikacji do konkretnej warstwy i w tym miejscu szukać jego rozwiązania.

Model TCP/IP jest modelem otwartym, który został opracowany we wczesnych latach siedemdziesiątych XX w. dla Departamentu Obrony Stanów Zjednoczonych.

Model ten zawiera cztery warstwy, którymi są:

- Warstwa aplikacji (ang. appliacation layer)
- Warstwa transportu (ang. transport layer)
- Warstwa internetowa (ang. internet layer)
- Warstwa dostępu do sieci (ang. network access layer)



Rysunek 2.7 Model systemu otwartego TCP/IP

2.2.1. Warstwa aplikacji

Warstwa aplikacji (*ang. Application layer*) jest najwyższą warstwą modelu a co za tym idzie jest zawsze najbliżej użytkownika. Dzieje się tak ponieważ to właśnie w tej warstwie działają aplikacje, programy, z których GUI (*ang. Graphical User Interface*) użytkownik ma bezpośredni wpływ na to co i na jakich warunkach zostanie przesłane poprzez zhierarchizowaną strukturę systemu otwartego. Wybór aplikacji jest uzależniony od wyboru użytkownika czego on w danym momencie potrzebuje, może to być klient poczty w tym celu wykorzysta on protokół *POP3* (*ang. Post Office Protocol*) bądź też *IMAP (ang.* Internet Message Access Protocol*)* do odebrania wiadomości. Może on też

użyć najpowszechniejszego protokołu tej warstwy protokołu *HTTP (ang. Hypertext Transfer Protocol)* w celu nawiązania komunikacji z serwerem *WWW (ang. World Wide Web).* Pobranie zasobu odbywa się po podaniu prawidłowego adresu strony zwanego *URL (ang. Uniform Resource Loader).* Następnie po wybraniu odpowiedniego protokołu warstwa aplikacji przekazuje dane warstwie niższej w celu kontynuowania procesu komunikacji z urządzeniem docelowym.

2.2.2. Warstwa transportu

Warstwa ta jest odpowiedzialna za transport danych pomiędzy urządzeniami końcowymi i wszystkim co jest związane z prawidłowym dostarczeniem danych do odbiorcy. Na tym poziomie jest wybierany prawidłowy rodzaj komunikacji oraz protokół z jakiego będzie korzystać usługa wybrana w warstwie aplikacji. Do dwóch najpopularniejszych protokołów warstwy transportu zaliczają się *TCP (ang. Transmission Control Protocol)* i *UDP (*ang. *User Datagram Protocol).* Oba te protokoły zostały opisane w *podrozdziale 2.1.4.*

2.2.3. Warstwa internetowa

Warstwa internetowa (ang. internet layer) jest to warstwa odpowiedzialna za tworzenie pakietów i adresowanie ich oraz podejmowania decyzji jaką drogą pakiet zostanie dostarczony do miejsca docelowego. Kontrolę nad tymi zadaniami sprawuje protokół IP. Warstwa internetowa w modelu TCP/IP odpowiada warstwie sieciowej (ang. network layer) w modelu odniesienia ISO/OSI.

W celu dostarczenia pakietu z jednego miejsca do drugiego należy zaadresować dane, służy do tego protokół IP w wersji 4 bądź też w wersji 6.

To właśnie w tej warstwie podejmowana jest decyzja jaką ścieżką pakiet zostanie dostarczony do miejsca docelowego, służy do tego mechanizm Routingu.

Proces Routingu jest to proces budowania trasy od jednego routera do drugiego. Można wyróżnić dwa typy routingu: statyczny (ang. static routing) i dynamiczny (ang. dynamic routing). Routingiem statycznym nazywamy proces, w którym to administrator decyduje jaką trasą zostanie przesłany pakiet przez sieć. Natomiast routingiem dynamicznym nazywamy sytuacje, gdzie to routery same podejmują decyzje jak pakiet zostanie przesłany przez sieć.

2.2.4. Warstwa dostępu do sieci

Warstwa dostępu do sieci (ang. network access layer) jest najniższą warstwą modelu TCP/IP. Odpowiada ona za prawidłowe zaadresowanie i przesłanie na poziomie fizycznym danych. Warstwa ta odpowiada także za prawidłowe dobranie technologii, aby dane zostały prawidłowo przesłane przez medium transmisyjne. To właśnie w tej warstwie dane zostają zamienione na bity.

Definicja protokołu

Definicja jednostek danych protokołu

Definicja zasad wymiany jednostek danych protokołu

Specyfikacja ASN.1

Notacja abstrakcyjna typów danych

Zasady kodowania na przykładzie BER

Bibliografia

Józefiok, A. (2020). *CCNA 200-301 Zostań administratorem sieci komputerowych CISCO.* Gliwice: Helion.

Kosmulska-Bochenek, E. (2002). *Wymiana informacji w heterogenicznych systemach sieciowych.* Wrocław: Oficyna Wydawnicza Politechniki Wrocławskiej.

Spis Ilustracji

Spis listingów

****

BLUBRY

****

WSTĘP

Cel pracy
Zakres pracy
Struktura pracy
    Jan Biały
    Piotr Stawski

Architektura i specyfikacja Gry Sieciowej

Klient
- Wymagania funkcjonalne
- Wymagania niefunkcjonalne
- Przypadki użycia
- Diagram Klas

Serwer
- Opis
- Implementacja
- Dziennik pracy
- 3.3. Protokół sieciowy
- Opis protokołu
- Elementy protokołu
- Implementacja protokołu

Implementacja gry
- Środowisko i narzędzia programistyczne
- Struktura plików i katalogów
- Techniki programistyczne wykorzystane w aplikacji
- Biblioteki
- Fragmenty kodu źródłowego
- Implementacja protokołu sieciowego
- Problemy implementacyjne?????

****
PROMOTOR
****

Spis treści

Spis zagadnień