

Transmisja bezpołączeniowa i połączeniowa

Mikołaj Leszczuk

Spis treści wykładu

- Komunikacja bezpołączeniowa
- Komunikacja połączeniowa
- Protokół UDP
- Protokół TCP
- Literatura

Komunikacja bezpołączeniowa – wprowadzenie

- Ang. *Datagram Support Services*
- **Datagram** = niezależny pakiet
- „Zawodne” przesyłanie krótkich wiadomości
- „Zawodność” - brak gwarancji, że adresat odbierze daną wiadomość
- Brak potwierdzeń – brak informowania strony wysyłającej o tym co stało się z wysłaną wiadomością

Komunikacja bezpołączeniowa – adresowanie datagramów

- Za pomocą konkretnego (unikalnego) adresu pojedynczego adresata
- Za pomocą trybu rozgłaszania w sieci (*broadcast*) – wiadomość wysyłana do wszystkich odbiorców w sieci, wpisanych na listę rozgłoszeniową

Komunikacja bezpołączeniowa – zastosowania

- Transmisje o niewielkiej ważności (czas)
- Transmisje w sieciach idealnych (2 komputery w LAN)
- Transmisje multimedialne (video na żądanie)
- Transmisje między uproszczonymi urządzeniami (wgrywanie nowych wersji oprogramowania)
- Transmisje rozgłoszeniowe (ang. *multicast*) w sieci (radio internetowe)

Komunikacja bezpołączeniowa (4/4)

- Należy podkreślić, że wiadomość zostanie odebrana tylko wtedy, gdy adresat oczekuje na odbiór datagramu
- W przeciwnym wypadku wiadomość jest ignorowana
- Limit przesyłanej informacji w pojedynczym datagramie wynosi zwykle około 500 bajtów

Komunikacja połączeniowa (1)

- *Session Support Services*
- Komunikacja połączeniowa polega na nawiązaniu sesji (logicznego połączenia) pomiędzy dwiema utworzonymi wcześniej nazwami.
- Przy czym nie muszą one znajdować się na dwóch różnych stacjach.
- Można także dokonywać połączeń w ramach jednej maszyny, a nawet w ramach jednego procesu

Komunikacja połączeniowa (2)

- Po nawiązaniu sesji, przydzielany jest jej unikatowy numer.
- W ramach sesji wiadomości przesyłane są niezawodnie.
- Oznacza to, że ich zawartość oraz kolejność odbierania nie zostają naruszone.
- Jeśli adresat nie odbierze wiadomości, strona wysyłająca jest informowana o zaistniałym błędzie.

Komunikacja połączeniowa (3)

- Pojedyncza wiadomość może mieć zwykle do około 64kB długości.
- Jeśli wystąpi błąd podczas przesyłania wiadomości, to następuje zamknięcie sesji.
- Protokoły zwykle same dbają o podział długich wiadomości na małe pakiety, w zależności od fizycznych wymagań sieci.

Protokoły połączeniowe

Mechanizm okna (1)

- Proces odbierający dane wysyła wraz z potwierdzeniami rozmiar bufora, który może przeznaczyć na dodatkowe dane.
- Ta ilość bajtów, nazywana oknem, to maksymalna ilość bajtów, jakie strona wysyłająca może wysłać dopóki nie dostanie nowego okna.
- Czasami odbiorca nie ma miejsca na przyjęcie danych - wtedy powinien odesłać wartość zero.
- W takiej sytuacji nadawca powinien co jakiś czas wysłać niedużą porcję danych by sprawdzić, czy można wysłać więcej danych.

Protokoły połączeniowe

Mechanizm okna (2)

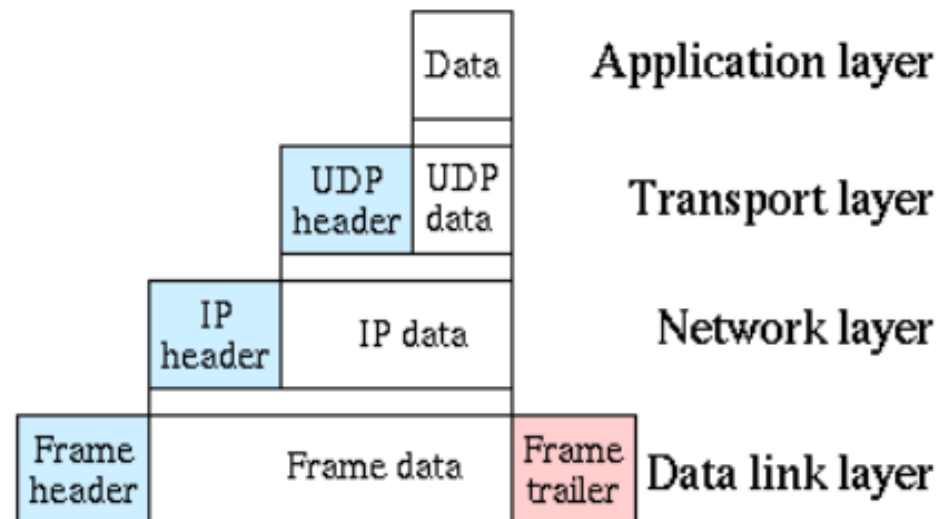
- Jeżeli okno ma przez dłuższy czas wartość zero, a nadawca nie może doczekać się żadnej komunikacji ze strony odbiorcy powinien uznać, że odbiorca "nie żyje".
- Musimy tu zauważyć jeszcze jedną bardzo ważną sytuację: odbiorca odsyłając "okno" określa, ile może przyjąć danych powyżej bajtu, który właśnie potwierdza.
- Taką wartość nazwiemy proponowanym oknem.
- Nadawca oblicza dla siebie wartość okna: jako proponowane okno minus ilość nie potwierdzonych bajtów.

UDP

Wprowadzenie

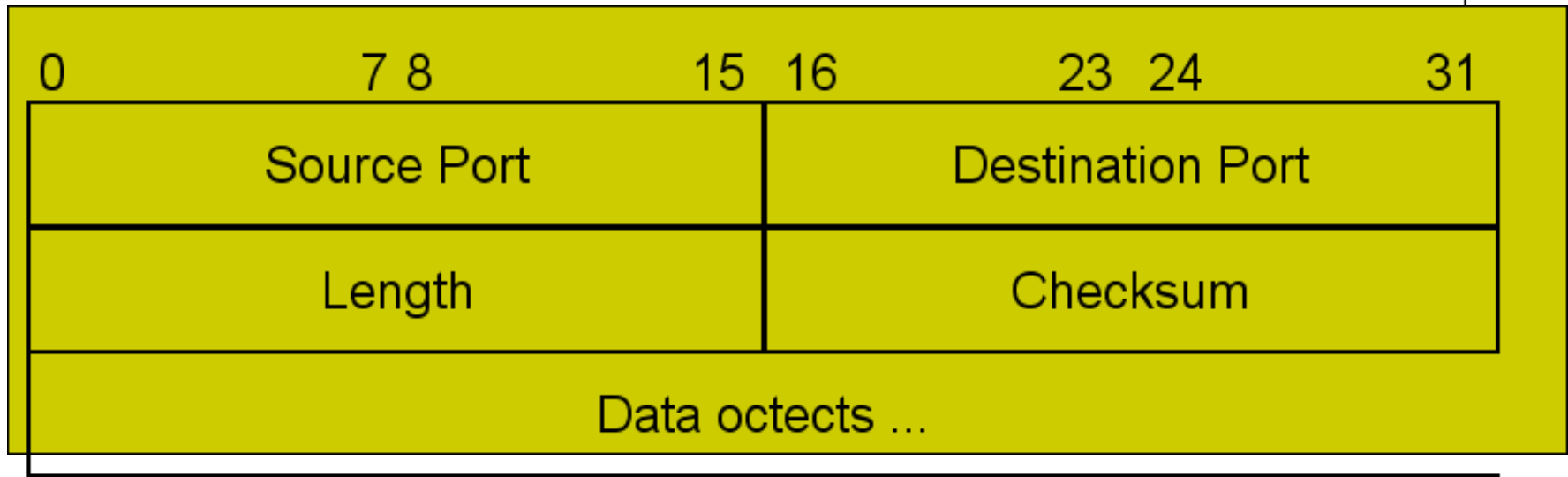
- User Datagram Protocol (UDP) został stworzony aby móc przesyłać datagramy pomiędzy komputerami podłączonymi do sieci komunikacyjnej
- Protokół ten zakłada występowanie IP w niższej warstwie
- Protokół zapewni maksymalnie uproszczoną procedurę przesyłania wiadomości pomiędzy aplikacjami
- Protokół jest zorientowany transakcyjnie, a dostarczenie wiadomości nie jest gwarantowane
- Aplikacje wymagające uporządkowanej i gwarantowanej transmisji strumieni danych powinny używać TCP
- Aplikacje wymagające uporządkowanej transmisji strumieni danych powinny używać RTP (Real-Time Transport Protocol)

Enkapsulacja UDP



Protokół UDP

Format nagłówka (1)



- **Source Port** jest polem opcjonalnym, jeśli podane, oznacza ono port procesu wysyłającego i można założyć, że jest to port, na który zaadresowana powinna być odpowiedź, jeśli brak jakiegokolwiek innej informacji.
- Jeśli nie używane, wstawiana jest wartość zero.

Protokół UDP

Format nagłówka (2)

- Destination Port ma znaczenie w kontekście konkretnego adresu IP celu.
- Length jest długością w oktetach datagramu wliczając w to nagłówek i dane.
- To oznacza, że najmniejszą wartością długości jest osiem.
- Checksum to suma kontrolna nagłówka i danych.

UDP

Interfejs użytkownika

- Interfejs użytkownika powinien pozwalać na:
- Tworzenie nowych portów odbiorczych,
- Operacje odbioru na portach odbiorczych:
 - Zwracające oktety danych
 - Określające:
 - Port źródłowy
 - Adres źródłowy
- Operacje umożliwiające wysłanie datagramu, przy określeniu:
 - Danych
 - Portu źródłowego i docelowego
 - Adresu docelowego

UDP

Interfejs IP

- Moduł UDP musi być w stanie określić źródłowy i docelowy adres IP oraz pole Protocol z nagłówka IP.
- W szczególności, interfejs UDP/IP może zwracać cały datagram IP wraz z pełnym nagłówkiem jako odpowiedź operacji odbioru.
- Taki interfejs UDP może także pozwolić a przekazanie całego datagramu IP z pełnym nagłówkiem do warstwy IP, która wyśle datagram w sieć.
- Warstwa IP weryfikuje wtedy zgodność konkretnych pól i oblicza sumę kontrolną nagłówka IP.

UDP

Aplikacje i numer protokołu

- Aplikacje najczęściej korzystające z UDP to:
 - Internet Name Server
 - Trivial File Transfer
- Numer protokołu do wpisania w odpowiednie pola nagłówka IP to 17 (21 oktalnie).

Protokół TCP

Wprowadzenie (1)

- TCP (ang. *Transmission Control Protocol*) to strumieniowy protokół komunikacji między dwoma komputerami.
- TCP zapewnia:
 - wiarygodne połączenie dla wyższych warstw komunikacyjnych przy pomocy sum kontrolnych
 - numerowanie pakietów w celu weryfikacji:
 - wysyłki
 - odbioru
- TCP porządkuje pakiety, które na przykład na skutek podróży różnymi ścieżkami, przybywają w nieprawidłowej kolejności.

Protokół TCP

Wprowadzenie (2)

- Brakujące pakiety są obsługiwane przez żądania retransmisji.
- TCP wymaga (w porównaniu do innych protokołów) większych:
 - mocy procesora
 - przepustowości łącz
- W związku z tym aplikacje, w których czas jest krytyczny (media strumieniowe, gry sieciowe) poświęcają wiarygodność na rzecz szybkości i nie używają TCP.
- Z kolei protokoły takie jak HTTP i FTP używają wiarygodnego TCP.

Protokół TCP

Przyczyny powstania

- Początkowo w celu sprostania wymaganiom transmisji dla potrzeb wojska
- Zwłaszcza:
 - pewność transmisji w obliczu niepewnego działania sieci
 - Dostępność transmisji w obliczu możliwych natłoków w sieci
- Oczywiście wiele z tych problemów występuje także w sieci publicznej – stąd późniejsza popularność TCP w Internecie
- Później więc protokół TCP został użyty w celu sprzęgania komputerów wielu różnych typów znajdujących się:
 - w instytutach badawczych,
 - na uniwersytetach,
 - w agencjach rządowych.

Protokół TCP

Niezawodna wymiana danych

- TCP jest protokołem end-to-end, stworzonym do współpracy z innymi protokołami.
- Protokół TCP realizuje usługę niezawodnej wymiany danych w łączach zorientowanych na połączenie.
- Oznacza to, że przy pisaniu aplikacji sieciowych wymieniających dane i wiadomości z innymi komputerami (a tym samym komunikujących się z niższymi warstwami protokołów sieciowych w celu przesyłania tych danych) nie ma potrzeby tworzenia funkcji weryfikujących, czy dane zostały odebrane.
- To zadanie realizuje niezawodny protokół TCP.

TCP

Warstwy protokołów



- Część większej całości określanej jako stos TCP-IP. W modelu OSI TCP – w warstwie Sesji. Ponad protokołem IP.
- Internet Protocol dostarcza protokołowi TCP mechanizmy wysyłki i odbierania zmiennej długości segmentów danych “zapakowanych” w datagramy IP
- Datagramy przejmują kwestie adresacji

Protokół TCP

Użycie protokołu IP

- Protokół TCP wykorzystuje protokół IP do przesyłania informacji poprzez sieć.
- Zaskakujące jest jednak to, że IP jest bezpołączeniowym protokołem sieciowym, nie gwarantującym niezawodnego przekazu danych.
- W zamian za to IP oferuje efektywny mechanizm transmisji.
- Jego wady związane z niezawodnością eliminuje nadrzędny protokół TCP.
- Dane i wiadomości TCP (oficjalnie nazwane segmentami) są umieszczane w datagramach IP i przesyłane przez ten protokół poprzez sieć.

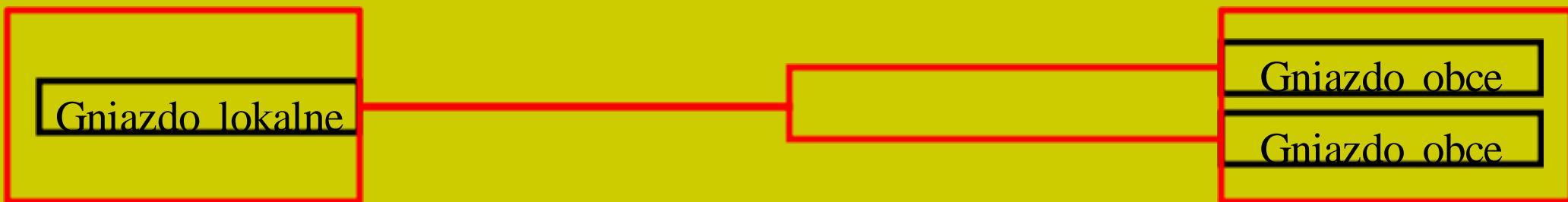
Hermetyzacja realizowana przez router

- Projektanci opracowali technikę hermetyzacji, w celu wyeliminowania konieczności dokonywania zmian w wewnętrznych protokołach sieciowych użytkowników, (co byłoby niezbędne w przypadku łączenia różnych typów sieci).
- Założono, że każda sieć może korzystać z własnego protokołu komunikacyjnego.
- Funkcję hermetyzacji realizują routery (oryginalnie nazywane bramami (gateways)).

Cechy TCP

Nawiązywanie połączenia (1)

- Identyfikator portu TCP: w celu identyfikacji niezależnych strumieni obsługiwanych przez TCP
- Porty wybierane niezależnie przez każdą ze stron i przez każde z połączeń => mogą nie być unikalne
- “Doklejane” do adresu IP: **AAA.BBB.CCC.DDD:EEEE**
- Połączenie specyfikowane przez parę gniazd (*sockets*) na obu końcach
- Transmisja danych w obu kierunkach (“*full-duplex*”)



Cechy TCP

Nawiązywanie połączenia (2)

- Dowolność przyporządkowywania portów TCP do procesów
- Jednakże w każdej implementacji – pewne podstawowe konwencje
- Grupa powszechnie znanych portów do których przyporządkowywane są tylko “odpowiednie” procesy
- Procesy mogą “posiadać” porty i inicjalizować połączenia tylko na portach które posiadają.
- Znane porty TCP (wybór):
 - 21 ftp
 - 22 ssh
 - 80 http
 - 139 netbios-ssn
 - 445 microsoft-ds
 - 8080 http-proxy

Cechy TCP

Nawiązywanie połączenia (3)

- Sposób implementacji nabywania portów w posiadanie – lokalny, według standardu:
 - Polecenie użytkownika (Request Port)
 - Powiązanie grupy portów identyfikowanych wyższymi bitami numerów z numerem procesu
- Połączenie specyfikowane wywołaniem OPEN
- W czasie nawiązywania połączenia – konieczność zapamiętania wielu informacji
- Struktura danych Transmission Control Block (TCB)
- W trakcie nawiązywania połączenia – wymieniane trzy komunikaty

Protokół TCP

Format nagłówka (1)

0		1								2								3													
0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1
Source Port																Destination Port															
Sequence Number																															
Acknowledgment Number																															
Data Offset		Reserved						U	A	P	R	S	F	Window																	
								R	C	S	S	Y	I																		
								G	K	H	T	N	N																		
Checksum																Urgent Pointer															
Options																								Padding							
Data																															

Protokół TCP

Format nagłówka (2)

- Source Port: 16 bitów
Numer portu źródłowego
- Destination Port: 16 bitów
Numer portu docelowego
- Sequence Number: 32 bity
Numer sekwencyjny pierwszego oktetu w tym segmencie (chyba, że ustawiono SYN). Jeśli ustawiono SYN, to numer sekwencyjny to initial sequence number (ISN)

Protokół TCP

Format nagłówka (3)

- Acknowledgment Number: 32 bity
Jeśli ustawiono bit ACK, to pole zawiera numer sekwencyjny, jakiego oczekuje nadawca tego segmentu. Zawsze wysyłany od chwili nawiązania połączenia.
- Data Offset: 4 bity
Liczba 32-bitowych słów w Nagłówku TCP, czyli wskaźnik na początek danych. Nagłówek TCP (nawet zawierający opcje) to liczba całkowita o długości 32 bitów.
- Zarezerwowane: 6 bitów
Zarezerwowane dla przyszłego użycia

Protokół TCP

Format nagłówka (4)

- Control bits: 6 bitów (od lewej do prawej)
 - URG: Urgent Pointer field significant
 - ACK: Acknowledgment field significant
 - PSH: Push Function
 - RST: Reset the connection
 - SYN: Synchronize sequence numbers
 - FIN: No more data from sender
- Window: 16 bitów
Liczba oktetów danych, poczynając od oktetu wskazanego w polu potwierdzenia, które zaakceptuje nadawca tego segmentu

Protokół TCP

Format nagłówka (5)

- Checksum: 16 bitów
Pole sumy kontrolnej to 16-bitowe dopełnienie do 16 bitów sumy kontrolnej każdego z bitów nagłówka i danych. Gdy segment nie jest wielokrotnością 16 bitów – wypełnienie zerami do prawa (tylko dla celów sumy kontrolnej). Pole sumy kontrolnej w trakcie jej liczenia równe zero.
- Urgent Pointer: 16 bitów
Pole określające aktualną wartość urgent pointer jako dodatniego przesunięcia od numeru sekwencyjnego w segmencie. Urgent pointer wskazuje na numer sekwencyjny oktetu zaraz po pilnych danych. Pole interpretowane tylko ustawiony jest bit kontrolny URG.

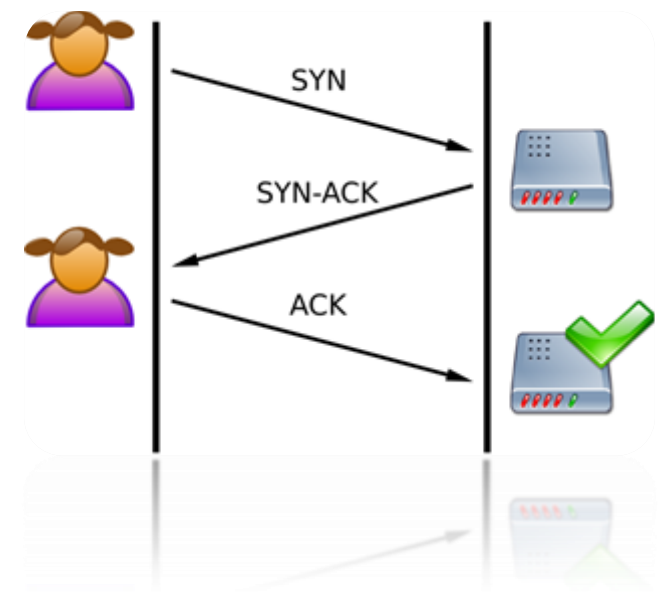
Protokół TCP

Format nagłówka (6)

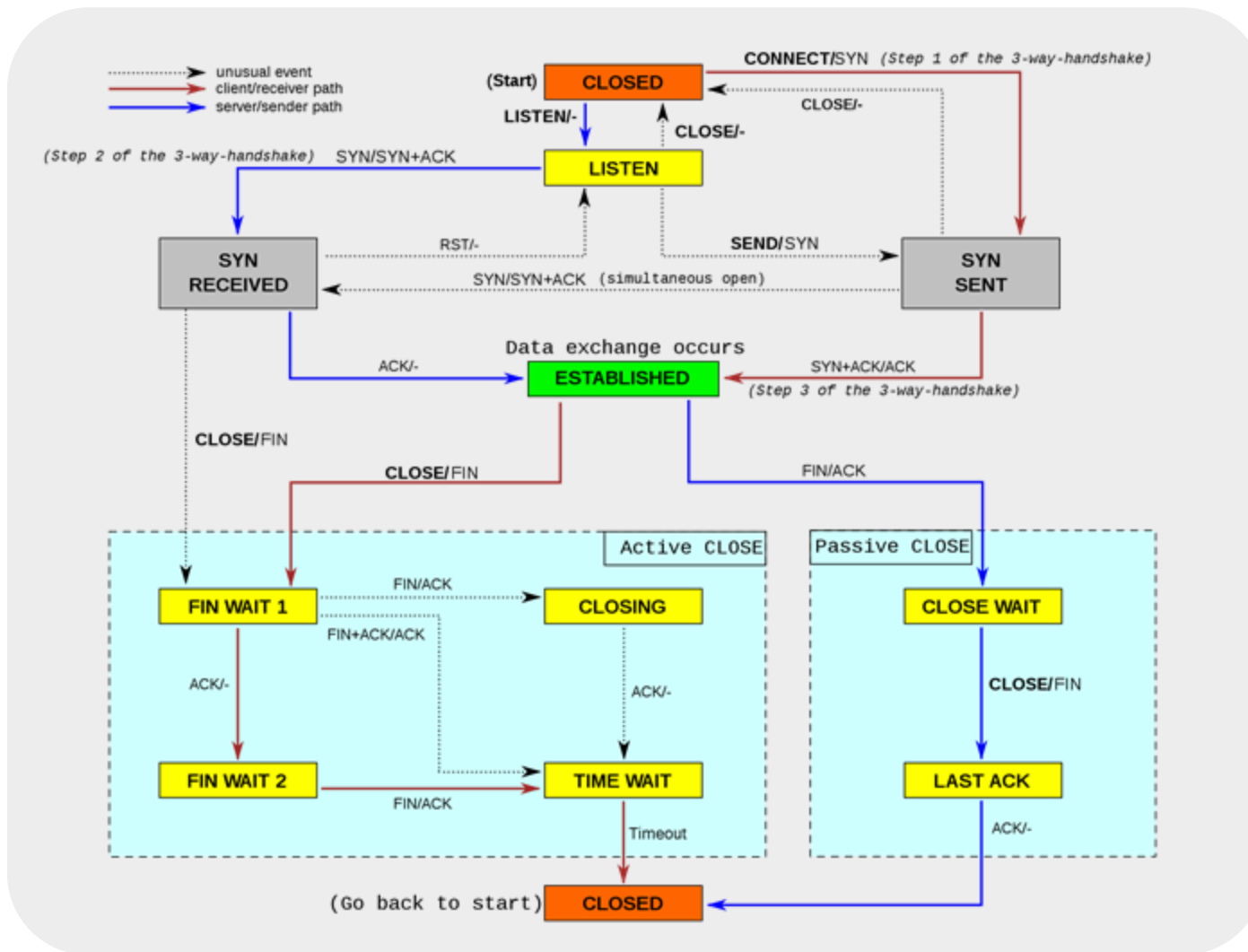
- Options: zmienne
Możliwe zajęcie przestrzeni aż do końca nagłówka TCP, wielokrotność długości 8 bitów. Wszystkie opcje są wliczane do sumy kontrolnej. Opcja może rozpoczynać się od pozycji wielokrotności oktetu. Dwie możliwości formatowania opcji:
 - Przypadek 1: opcja o długości jednego oktetu
 - Przypadek 2: opcja o długości wielu oktetów:
indykator opcji, długość, właściwe dane
- Padding: zmienne
Zapewnia podzielność długości nagłówka przez 32 bity, wartość zerowa

Three-Way Handshake

- TCP characteristic connection point
- Host initiating connection sending packet containing TCP segment with SYN flag set (synchronize)
- Host receiving call (if wanting to handle them) sending package with flags SYN and ACK (acknowledge – confirmation)
- Initiating host sending now first chunk of data, just by setting the ACK flag set (and putting out SYN)
- If receiving host does not want or cannot take call, then responding with packet set to RST (reset)



Simplified TCP State Diagram



Literatura i źródła (1/3)

- ***“RFC: 793; Transmission Control Protocol; DARPA Internet Program; Protocol Specification”***
<http://www.rfc-editor.org/rfc/rfc793.txt>
- ***“RFC 768 - User Datagram Protocol”***
<http://www.rfc-editor.org/rfc/rfc768.txt>

Literatura i źródła (2/3)

- ***“Wikipedia”***
<http://pl.wikipedia.org>
- ***“TCP, Transmission Control Protocol”***
<http://www.networksorcery.com/enp/protocol/tcp.htm>
- ***Tomasz Czwarono, “Działanie i konfiguracja TCP”***
<http://www.owad.civ.pl/index.php?howto=5>

Literatura i źródła (3/3)

- “Trivial TCP; BARDZO małe wprowadzenie do mechanizmów zawartych w protokole TCP”
<http://rainbow.mimuw.edu.pl/public/inf/SIK/zadania/zad1.txt>
- “Pakiet 1: Sieci komputerowe”
http://medtech.eti.pg.gda.pl/pakiet1/pkt_1_40.html