

SiST

część 3

Małgorzata Langer

Główne typy protokołów

- „PROTOKÓŁ” – co to jest?
- Połączeniowy (np. **TCP/IP**)
- Beipołączeniowy (np. **UDP/IP**)

Ping

- Oprogramowanie używane przez TCP/IP (np. w internecie) to diagnozowania połączeń sieciowych. Pozwala na testowanie połączenia pomiędzy częścią testującą i testowanymi (dotyczy jakości, opóźnienia...)

- Zakres ważnych adresów hosta zawiera się *pomiędzy* numerem podsieci (pole host wyzerowane) oraz adresem nadawania (broadcast) – pole hosta zawiera same jedyńki
- Tak więc np. zakres adresów hosta dla podsieci 172.16.8.0/22 wynosi od:

10101100.00010000.00001000.00000001=172.16.8.1

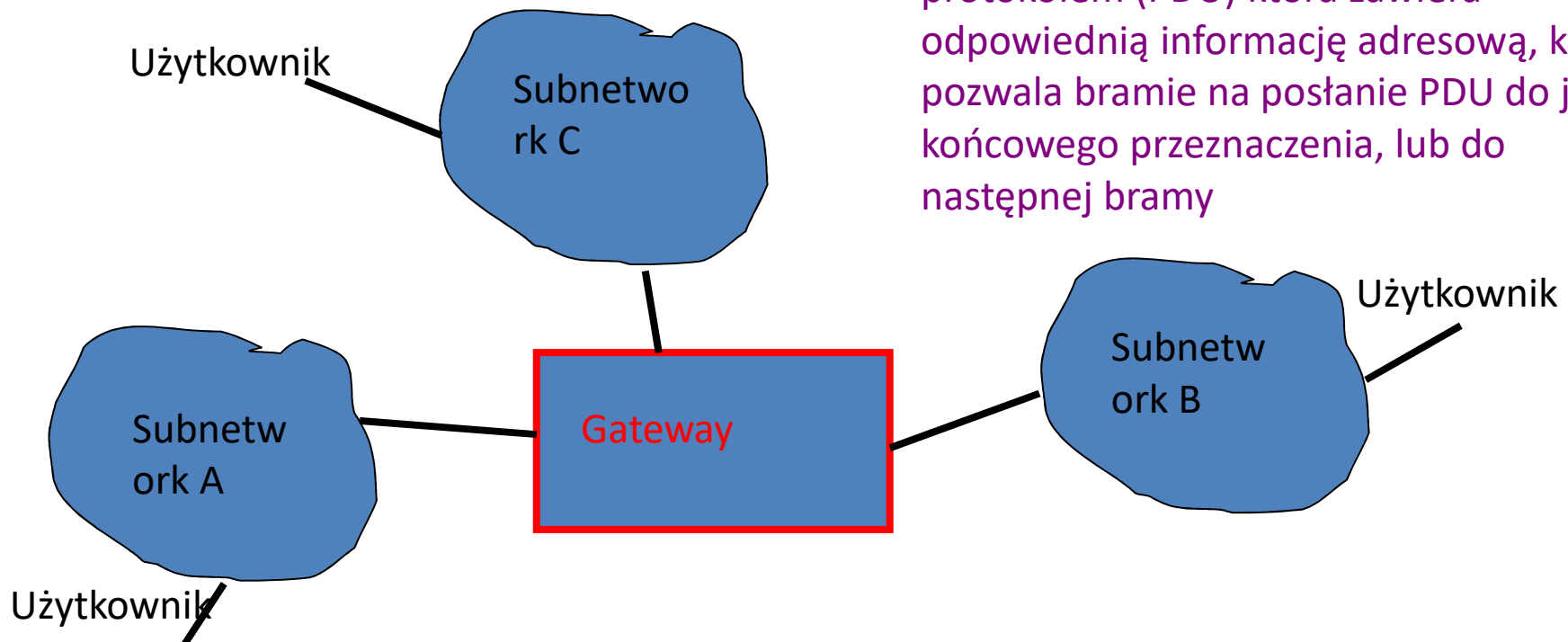
do

10101100.00010000.00001011.11111110=172.16.11.254

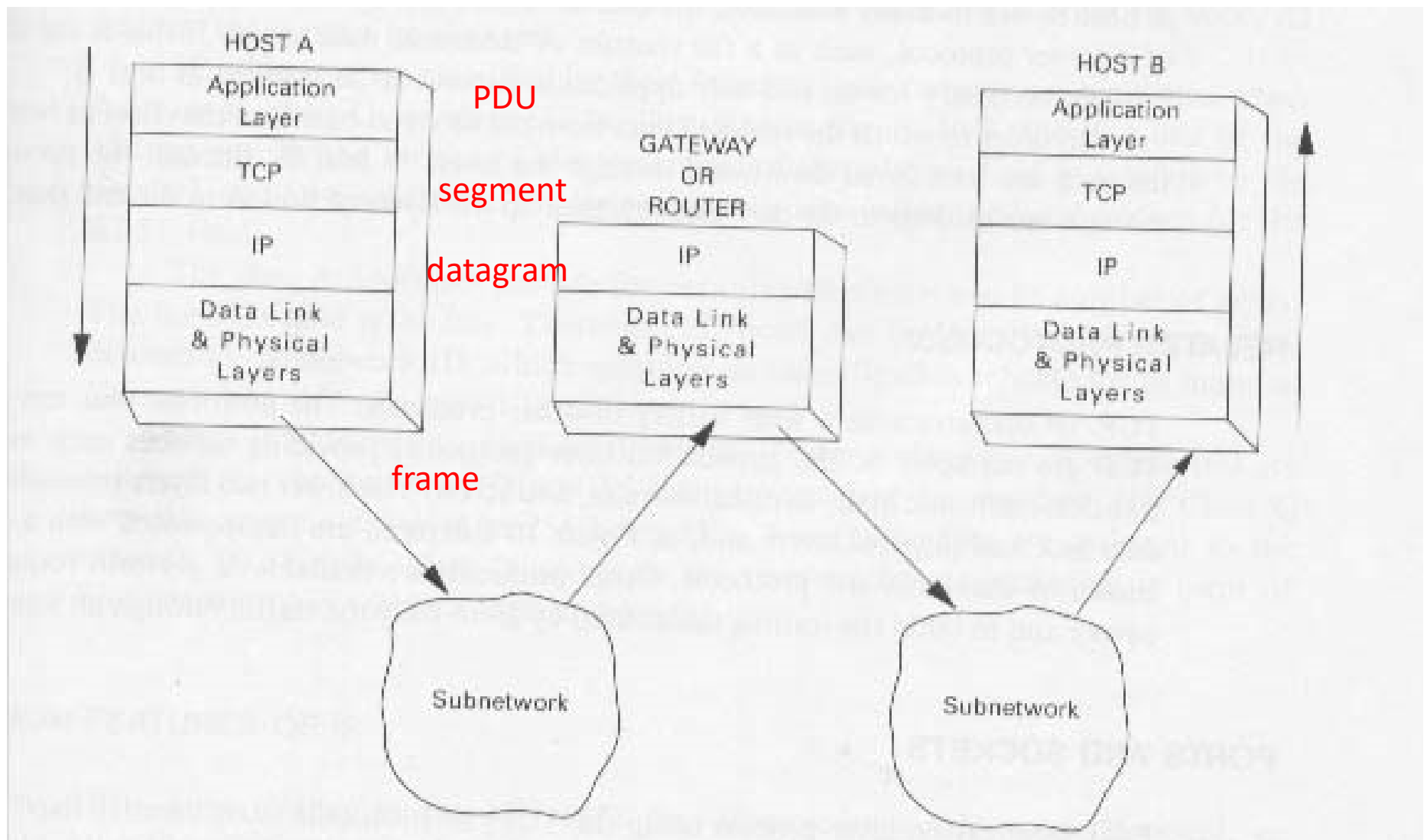
TCP/IP

- The Transmission Control Protocol/Internet Protocol
- Brama, router

Zasadniczym przeznaczeniem bramy jest otrzymanie jednostki danych zg, z protokołem (PDU) która zawiera odpowiednią informację adresową, która pozwala bramie na posłanie PDU do jej końcowego przeznaczenia, lub do następnej bramy



Przykład operacji TCP/IP



- TCP/IP jest nieświadome, co dzieje się wewnątrz sieci. Manager sieci (administrator) ma swobodę manipulacji i zarządzania PDU w dowolny niezbędny sposób. W większości przypadków Internet PDU (dane i nagłówki) pozostaje niezmienną w całym czasie transmisji w podsieci.

W bramie...

- PDU jest przetwarzane przez niższe warstwy i przechodzi do warstwy IP (network - sieciowej). Tutaj podejmowane są decyzje wyznaczenia trasy w oparciu o adres dostarczony przez hosta (host computer)
- Datagram jest przesłany do połączenia komunikacyjnego, które podłączone jest do odpowiedniej podsieci
- Datagram jest ponownie kapsułowany (PDU) do warstwy data link (jako ramka) i przesyłany do następnej podsieci.

Porty i Gniazda

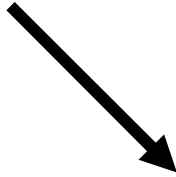
- Każdy proces warstwy aplikacji, używający protokołów TCP/IP musi się identyfikować numerem *portu*.
- *Gniazdo* identyfikuje końcowy proces komunikacji (logiczna suma numeru portu i adresu sieci)
- Pewne numery portów są wstępnie definiowane (od 0 do 255) i nazywane *well-known-ports*

IP Datagram

WERSJA (4)		DŁUGOŚĆ NAGŁÓWKA (4)	
RODZAJ USŁUGI - TOS (8)			
DŁUGOŚĆ CAŁKOWITA (16)			
FLAGI (3)		FRAGMENT OFFSET (13)	
TTL – CZAS ŻYCIA (8)			
PROTOKÓŁ (8)			



IP Datagram – C.D.



SUMA KONTROLNA NAGŁÓWKA (16)
ADRES ŹRÓDŁOWY (32 lub 128)
ADRES PRZEZNACZENIA (32 lub 128)
OPCJE I PADDING (różne)
DANE (różne)

IP Datagram

- **Pole wersji** – identyfikuje używaną wersję IP (niektóre węzły sieci mogą nie mieć tej samej wersji protokołu) – aktualna wersja IP to 6 (128 bitów) lub wciąż jeszcze 4 (32 bity)
- **Długość nagłówka** zawiera 4 bity, które ustawione są na wartość wskazującą długość nagłówka datagramu, mierzoną w słowach 32-bitowych. Typowo nagłówek zawiera 20 ośmiobitowych ciągów (bajtów) a wartość zazwyczaj ustawiona jest na 5

- **Rodzaj usługi** (TOS) – pierwsze 2 bity są używane do wskazania kilku funkcji, zależnie od administratora

Bit 3 –bit opóźnienia (D bit); ustawiony na 1 żąda jak najmniejszego opóźnienia w usługach internetu

Bit 4 – bit przepływności (T bit); ustawiony na 1 żąda jak największej przepływności w usługach internetu

Bit 5 –bit niezawodności (R bit) pozwala użytkownikowi zażądać wysokiej niezawodności dla datagramu

Bity 6 i 7 – nie są zdefiniowane

- Pole całkowitej długości podaje całkowitą długość datagramu IP; mierzoną w bajtach; zawiera długość nagłówka i danych. Maksymalna możliwa długość datagramu wynosi 65.535 bajtów (2^{16})

Wszystkie bramki muszą „dać sobie radę” z datagramami o całkowitej długości 576 bajtów (TO JEST MINIMUM)

IP odejmuje długość nagłówka od pola długości całkowitej i oblicza rozmiar pola danych.

Trzy pola w nagłówku służą kontroli fragmentacji datagramu i jego ponownego zestawienia

- **Identyfikator** – podaje adres źródłowy w hoście odbierającym, aby mógł zidentyfikować fragment
- **Flagi** – zawierają bity do określenia czy datagram może być cięty i jeżeli tak, jeden z bitów może być ustawiony do określenia czy jest to ostatni fragment datagramu
- **Fragmentation offset** – zawiera wartość, która określa relatywną pozycję fragmentu w stosunku do oryginalnego datagramu; wartość ustawiona jest początkowo na 0 i kolejno ustawiana na właściwą wartość, jeżeli brama dokonuje fragmentacji danych; mierzona w jednostkach ośmiu bajtów

TTL – czas życia

- Parametr jest stosowany do pomiaru czasu, w jakim datagram przebywa w internecie. Każda brama musi sprawdzić to pole i zmniejszyć jego wartość w każdym przetwarzanym datagramie – **zapobiega to zapętleniu przesyłania;**
może być również wykorzystywany dla celów diagnostycznych

- **Pole protokołu** – identyfikuje protokół warstwy następnej powyżej IP, który ma odebrać datagram w hoście końcowego przeznaczenia; najczęściej używane liczby, to: 6=TCP, 20=OSI transport layer...)
- **Suma kontrolna nagłówka** stosowana jest by wykryć zakłócenie, które może nastąpić w zakresie nagłówka. Kontrole nie są wykonywane dla strumienia danych użytkownika (dane mają być „przezroczyste”).
- **Adresy źródłowy i przeznaczenia** – wartość nie jest zmieniona w ciągu całego życia datagramu; są to adresy IP; 128 to bardzo długi ciąg; istnieją metody do jego minimalizacji

- **Pole opcji** do identyfikacji kilku dodatkowych usług, nie jest wykorzystywane zawsze, często służy potrzebom administratora sieci i diagnostyki
- **Padding field** – może być wykorzystane do uzupełnienia dokładnie kolejnych 32 bitów
- **Data field** (pole danych) zawiera dane użytkownika, suma długości pola danych i nagłówka nie może przekroczyć 65.535 bajtów.

Główne usługi IP

- Source routing (wyznaczenie trasy zgodnie ze źródłami)
- Operacje wyznaczania trasy (Routing)
- Loose and strict routing (swobodny i dokładnie wyznaczony)
- Opcja Route-Recording
- Opcja zapisywania czasu (Timestamp)
- Moduł ICMP

Source routing

- Pozwala protokołowi warstwy wyższej (upper-layer protocol ULP) określić, w jaki sposób bramy IP mają wyznaczać trasę datagramów. ULP posiada opcję przesłania liste adresów internetowych do modułu IP – pośrednie węzły IP przez które powinny być przesyłane datagramy do końcowego przeznaczenia, które jest ostatnim adresem na liście.

Source routing – c.d.

- Gdy IP otrzymuje datagram, wykorzystuje adres w polu adresu źródłowego, aby określić następny pośredni skok. IP używa pola wskaźnikowego aby dowiedzieć się, jaki jest następny adres IP. Wtedy IP zamienia wartość w liście source routing swoim własnym adresem (i zwiększa wskaźnik o jeden adres (4 bajty w IPv4), aby ustawić następny skok do kolejnego adresu IP na trasie. W ten sposób datagram wędruje zgodnie z trasą (source route) wyznaczoną przez ULP i równocześnie zapisuje się cała jego trasa.

Operacje Routingu

- Brama IP podejmuje decyzje dot. Routingu w oparciu o routing list. Jeżeli host przeznaczenia jest w innej sieci, brama IP musi zdecydować, jak wyznaczyć trasę do tej innej sieci.
- Każda brama utrzymuje tabelę routingu (statyczną lub dynamiczną) która zawiera wejście do każdej osiągalnej sieci (adres sieci i jedną z sąsiadujących bram.

Brama sąsiadująca

- Daje najkrótszą drogę do sieci przeznaczenia

Miara odległości

Liczba skoków pomiędzy bramą i końcowym przeznaczeniem

Jeżeli nie udało się niczego dopasować....

- Brama buduje komunikat o błędzie, który zostanie przesłany wstecz do źródła IP przez protokół ICMP (*Internet Control Message Protocol*), który zawiera kod 'destination unreachable'.

Loose & Strict Routing

- **Routing swobodny** – moduły IP korzystają z pośrednich skoków, do osiągnięcia adresów otrzymanych na liście źródłowej, jak długo datagram przechodzi przez podane węzły
- **Strict source routing** – datagram podróżuje wyłącznie przez wskazane adresy z listy źródłowej. Jeżeli nie można wykonać takiego ruchu, host wysyłający jest powiadamiany komunikatem o błędzie.

Opcja zapisywania trasy

- Działa jak source routing z dodatkową cechą zapisywania trasy

Opcja zapisywania czasu (Timestamp)

Każdy moduł IP daje swoją pieczęć – czas jest oparty na milisekundach, z wykorzystaniem czasu uniwersalnego (Greenvich)

ICMP – the internet control message protocol

- **Dlaczego?** –IP jest protokołem pracującym w trybie bezpołączeniowym; jako taki nie ma żadnych mechanizmów raportujących i/lub poprawiających błędy
- **Po co?** –ICMP raportuje błędy przy przetwarzaniu datagramu i wydaje kilka administracyjnych komunikatów oraz komunikatów o stanie

FORMAT komunikatu ICMP

IP Nagłówek
TYP (8)
KOD (8)
SUA KONTROLNA (16)
PARAMETERY, może nie być uży
INFORMACJA (długość zmienna)

Pole protokołu w nagłówku IP ustawione jest na 1 aby wskazać

Aby zdefiniować typ komunikatu

Aby opisać typ błędu lub informację o statusie

Wylicza się dopełnienie „1” do 16 bitów w komunikacie ICMP

komunikaty ICMP są przenoszone przez użytkownika datagramów

Komunikat ICMP o błędach zawiera również główkę internetową i pierwsze 64 bity z pola użytkownika, co jest użyteczne przy analizie problemu czy awarii

ICMP – Procedury raportowania błędu i statusu

- Przekroczony czas życia wyznaczony dla datagramu
- Nieczytelny parameter
- Nieosiągalne miejsce przeznaczenia
- Nieważność źródła
- Powtarzalne echo
- Przekierowanie
- Timestamp oraz odpowiedź na timestamp
- Żądanie informacji lub odpowiedź
- Żądanie maski adresu i odpowiedź

TCP

- TCP rezyduje **w warstwie transportu** (ponad IP a poniżej warstw górnych)
- TCP realizuje zadania niezawodności, kontroli przepływu, kolejności, otwierania i zamykania
- Używany jest jako połączenia TCP/IP ale może również wspierać inne protokoły (ISO, FTP, itd...)
- TCP jest protokołem połączeniowym

TCP

- Odpowiada za transfer end-to-end danych przez jedną sieć lub wiele sieci
- Odpowiada za niezawodny transfer każdego znaku (bajtu, słowa) przekazanego mu z warstwy górnej – **tworzy obwód wirtualny**
- Zwraca potwierdzenie (ACK), lub negatywne potwierdzenie (NACK) do modułu wysyłającego TCP

Protokół kierowany strumieniowo

- Przesyła **indywidualne znaki**, a *nie bloki, ramki, itp.*
- Gdy w warstwie TCP pojawiają się bajty – zostają pogrupowane w *segmenty*
- Długość segmentu określana jest przez TCP (tzn. administrator określa sposób podejmowania tej decyzji przez TCP)

Inne funkcje TCP

- TCP sprawdza, czy nie dublują się dane
- Wspiera funkcję 'push'
- Może wykorzystywać dla ACK numerów segmentów, ale może też zmieniać kolejność segmentów w końcowym miejscu przeznaczenia
- Eliminuje duplikaty segmentów
- Tworzy sesje wielu użytkowników (funkcja multiplexu w ramach pojedynczego hosta na ULP (skrypty) (współdzielenie portów i gniazd)
- Daje pełną dwukierunkową (duplex) transmisję pomiędzy dwoma jednostkami TCP
- Daje możliwość podania poziomów bezpieczeństwa i priorytetów
- Wykonuje 'graceful close' (zamyka w sposób bezpieczny) obwód wirtualny

Otwarcia pasywne i aktywne

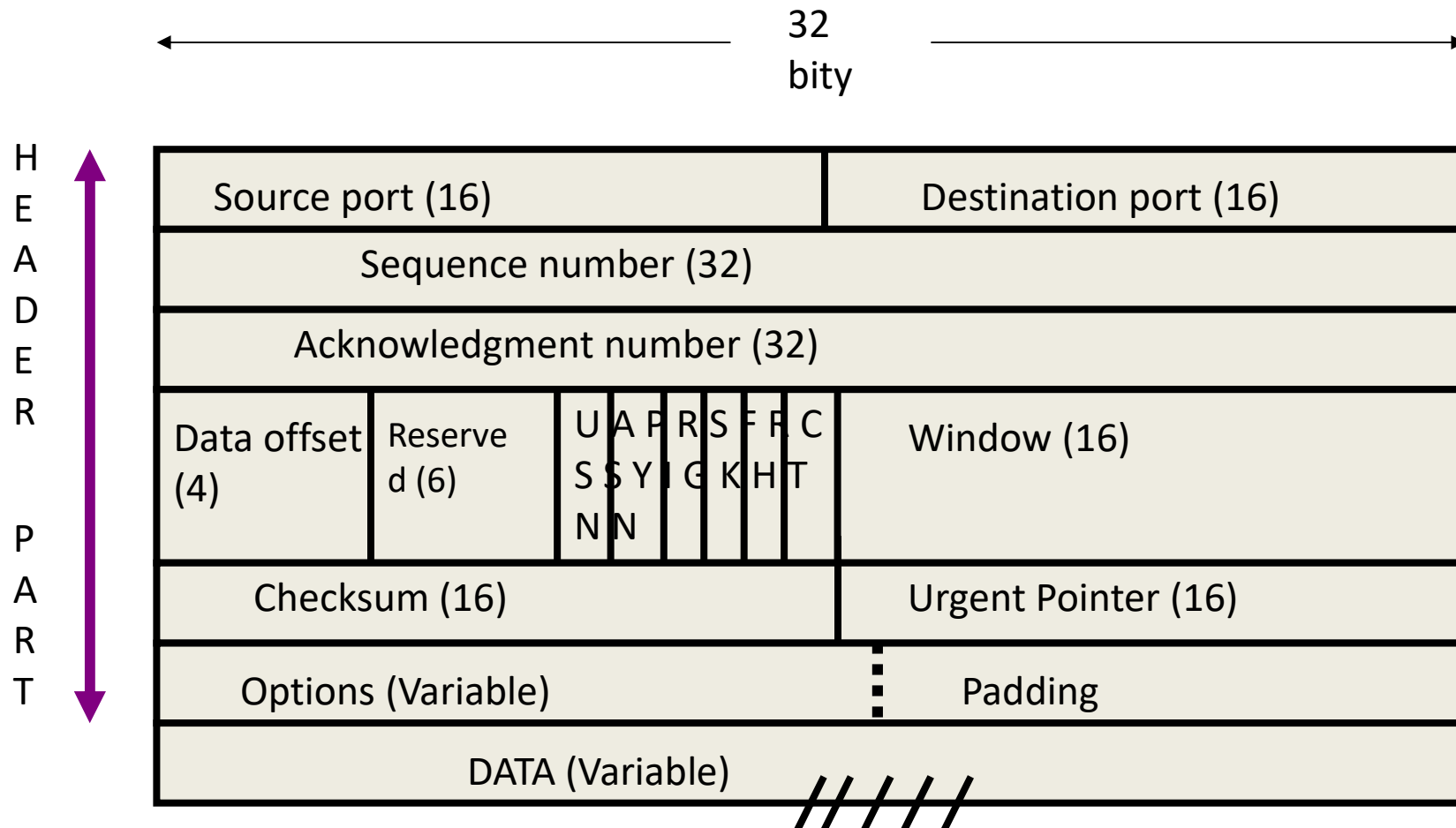
- Otwarcie pasywne – server nakazuje TCP oczekiwanie na nadejście żądania połączenia. Gdy ono nadejdzie, nie będzie żadnego opóźnienia.
- Otwarcie aktywne – ULP (server) wskazuje inne gniazdo, poprzez które ustalone jest połączenie (otwarcie aktywne wydawane jest na port z pasywnym otwarciem, aby ustalić wirtualny obwód)

The Transmission Control Block (TCB) – blok kontroli transmisji

- Przechowuje informację o każdym wirtualnym połączeniu.

Zawiera również kilka zmiennych związanych z numerami kolejności wysyłania i otrzymywania.

TCP Segment (PDU)



- **Port źródła i przeznaczenia** – identyfikuje oprogramowanie aplikacyjne w górnych warstwach, które używają połączenia TCP
- **Numer kolejny** – kolejny numer pierwszego bajtu w polu użytkownika (podaje pozycję w strumieniu transmitującym bajty)
- ***Podczas operacji zarządzania*** – podaje numer **initial send sequence** (ISS) – początkowa wysłana sekwencja, który będzie wykorzystany do kolejnej numeracji danych użytkownika

- **Numer potwierdzenia** ustawiany jest na wartość, która potwierdza dane uprzednio otrzymane (wszystkie bajty do i włącznie z niniejszym numerem minus jeden 1)
- **Data offset** – pole podaje ilość słów uzupełniających 32-bity nagłówek TCP; używane jest do wskazania, gdzie rozpoczyna się pole danych
- **Pole zarezerwowane** – 6 bitów, które MUSZĄ być ustawione na 0

Flagi

- **URG** wskazuje, czy ważny jest wskaźnik pola „pilne”
- **ACK** wskazuje czy ważne jest pole potwierdzeń
- **PSH** wskazuje, że moduł będzie podlegał przyciskowi (push button)
- **RST** wskazuje, że połączenie musi być zresetowane

Flagi -

- **SYN** wskazuje konieczność synchronizacji numerów kolejnych
- **FIN** wskazuje, że wysyłający nie ma już więcej danych do wysłania (podobnie, jak EOT)

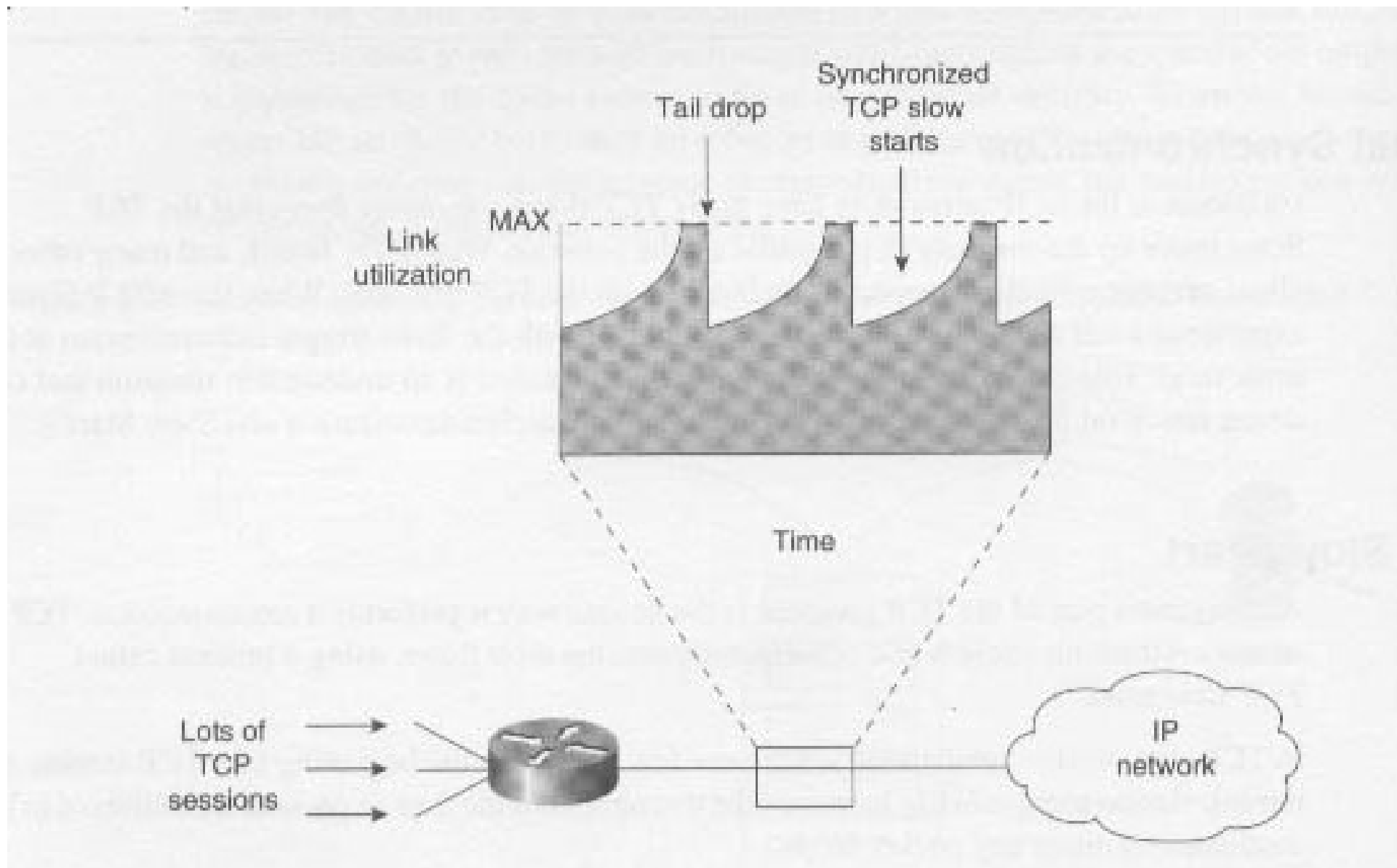
- **Window** – ustawiony jest na wartość wskazującą ile bajtów odbiornik może zaakceptować
- **Checksum** – dokłada 16-bitowe uzupełnienie jedynekami do wszystkich 16-bitowych słów w segmencie (włącznie z nagłówkiem i tekstem)
- **Urgent pointer** – używany, jeśli ustawiona jest flaga URG. Celem jest wskazanie bajtu danych, po którym płyną dane pilne (*out-of-band*)

- **Options** – pole przygotowane dla dalszych opcji TCP; trzy z nich są zdefiniowane:
0: end-of-option list
1: no operation
2: maximum segment size
- **Padding** - pole które zapewnia, że cały nagłówek TCP jest wielokrotnością 32 bitów.

TCP Slow Start – powolny start

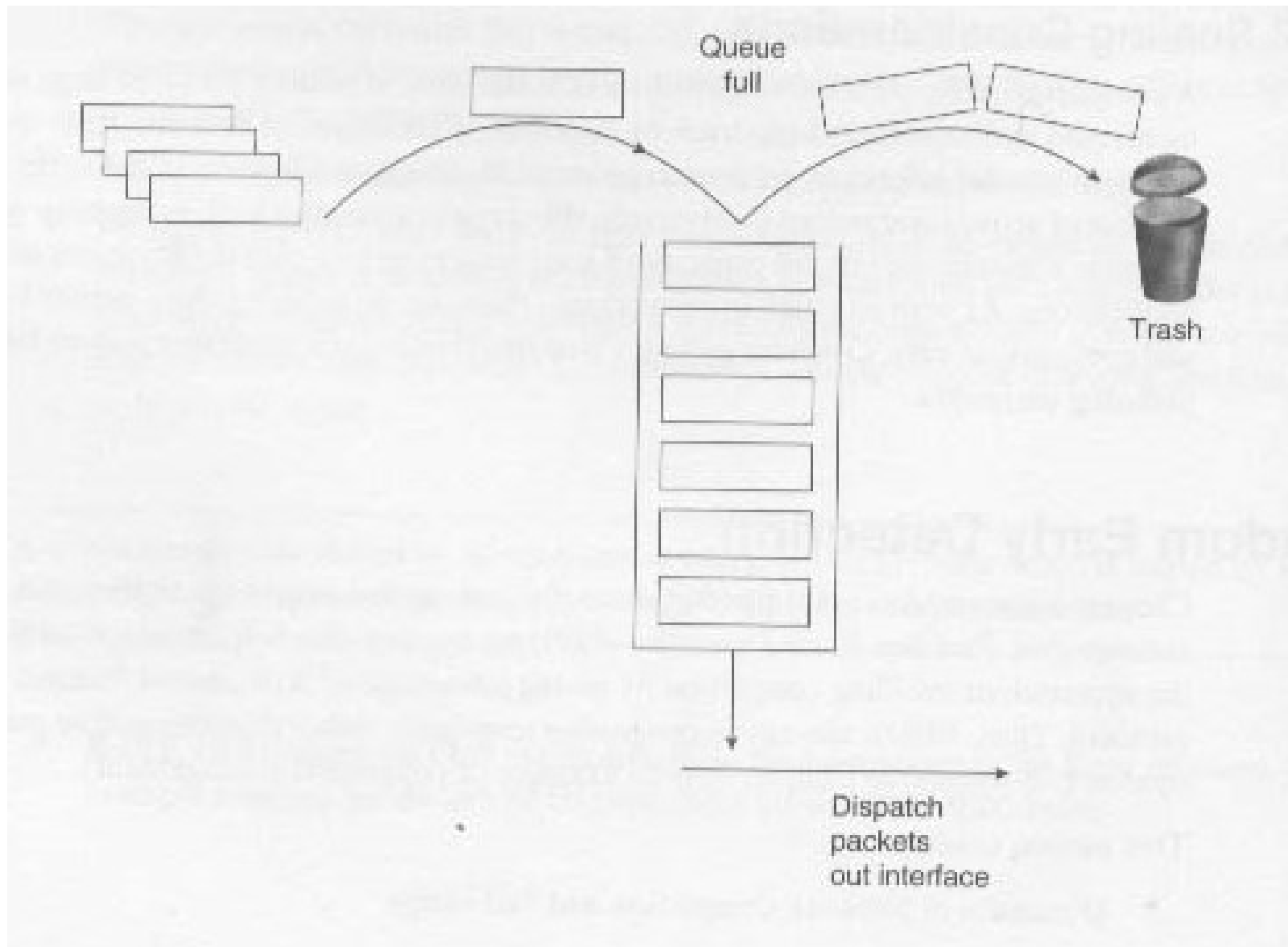
- Przy TCP slow start, terminal na początku sesji TCP lub retransmisji przesyła bardzo mało pakietów, stopniowo zwiększając, gdy pakiety są prawidłowo doręczane do miejsca przeznaczenia bez odrzucania pakietów drops

Jeżeli wiele strumieni TCP realizuje TCP slow start w tym samym czasie, ruch w sieci przybiera kształt zębów piły (sawtooth pattern)



Cykl ruchu, który wzrasta do nasycenia i gwałtownie opada

Główna przyczyna – kiedy bufor kolejki w systemie się wypełnia, ruter może jedynie wykonywać odrzucanie kolejnych (TAIL DROP)



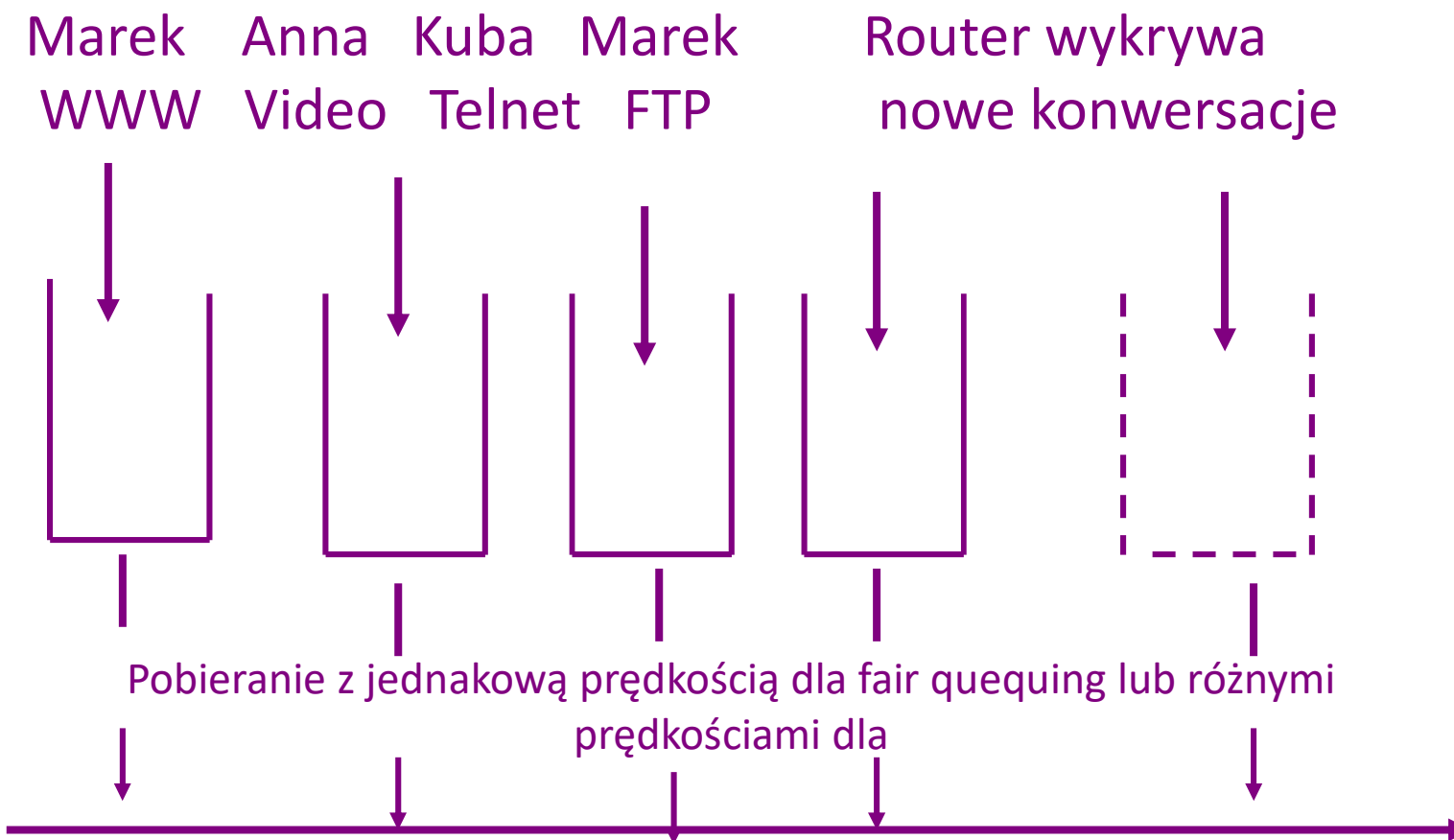
- Tail drop – nadchodzące na końcu pakiety są odrzucane
- TCP dokona retransmisji pakietów, które zostały odrzucone, (ale na przykład UDP – nie)
- Kolejki mogą być różnie konfigurowane; najczęściej jest to: FIFO (first-in, first-out); inne systemy kolejkowe również mogą podlegać odrzucaniu pakietów

Weighted Fair Queuing (WFQ)

- Dynamicznie wykrywa strumień pakietów od różnych aplikacji i automatycznie tworzy oddzielne kolejki dla tych strumieni
- W logice WFQ te strumienie nazywa się *konwersacjami*. Może to być np. sesja Telnet, FTP transfer, transmisja video przez IP, połączenia ze stronami internetowymi

- WFQ zarządza wieloma kolejkami konwersacji, jedna dla każdej konwersacji oraz sortuje pakiety do odpowiednich kolejek w oparciu o te konwerscje
- WFQ jest wywoływana tylko wtedy, gdy połączenie zewnętrzne jest przeciążone
- **fair-queue** ustawia WFQ, **no fair-queue** nie pozwala na WFQ i ustawia kolejkowanie na FIFO
- Niektóre interfejsy posiadają WFQ jako podstawę w swojej konfiguracji, nie będą więc pokazywały komendy **running-config**

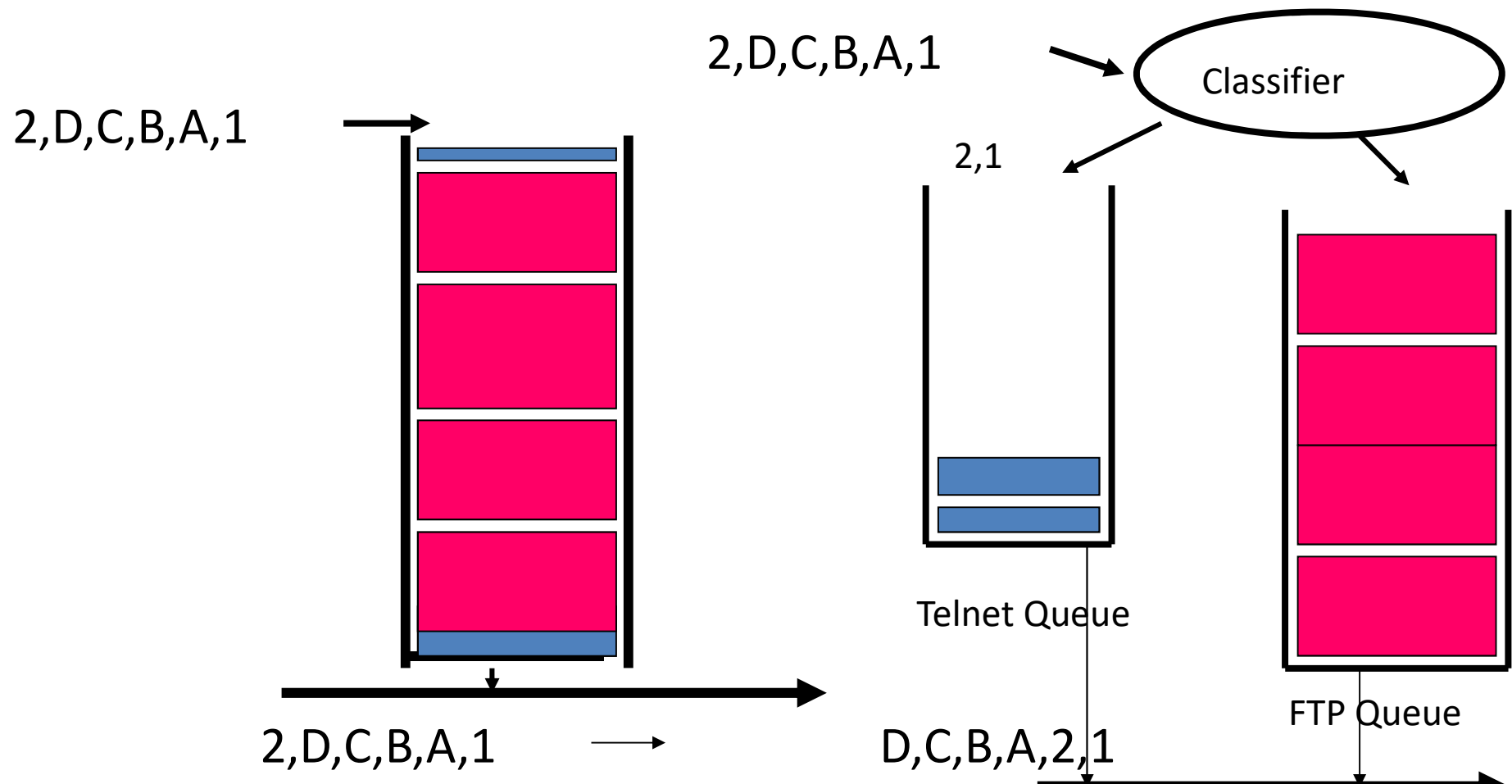
Fair queuing tworzy oddzielną kolejkę dla każdej konwersacji



Fair Queuing Versus FIFO

A FIFO Queuing Example

Fair Queuing Example



- Jeżeli pakiety 1 i 2 są pakietami niewielkimi w porównaniu z olbrzymimi pakietami od A do D, wtedy pobieranie jednakowej liczby bajtów z każdej kolejki oznacza, że na jeden pobrany pakiet długi będzie przypadać kilka pakietów niewielkich.
- Nawet, jeśli konwersacje o dużej objętości są pobierane z większą prędkością, nie wpływają na kolejki z pakietami krótkimi. Wszystkie kolejki są oddzielone i ruter uczciwie pobiera z każdej.

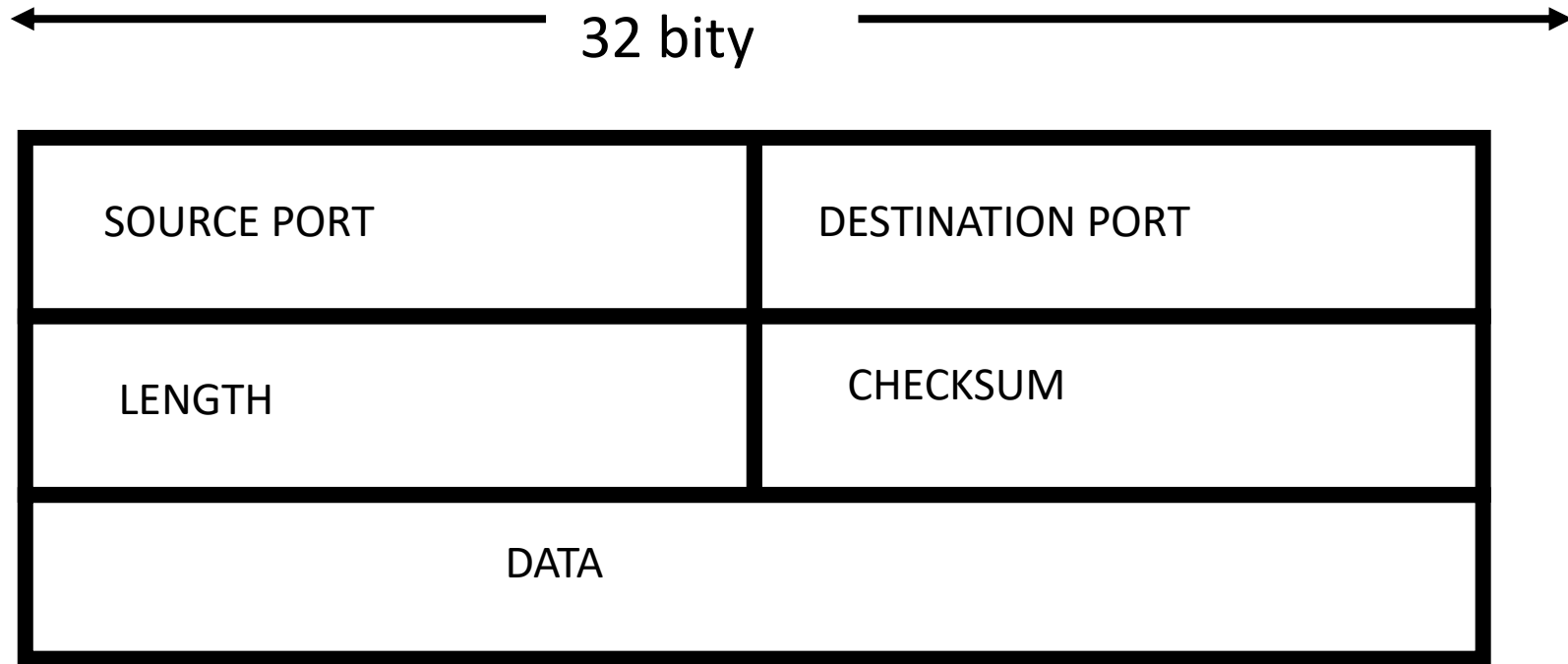
fair-queue 64 128

- Ustawia maksimum na 128 kolejek; 64 jest graniczną liczbą pakietów w jednej kolejce
- Weighting oraz IP precedence gives weighted fair queuing
- WFQ kontroluje wartość IP precedence w pakiecie, aby obliczyć liczbę nazywaną *wagą* dla tego pakietu; później używa tej wagi do określenia szybkości pobierania tego pakietu z kolejki konwersacji.

Jeżeli nie istnieje potrzeba wykorzystania wszystkich usług TCP , można zastosować USER DATAGRAM PROTOCOL (UDP)

- UDP jest klasyfikowany jako protokół bezpołączeniowy – nie zapewnia żadnej niezawodności ani mechanizmów kontrolujących przepływ, nie posiada żadnych procedur naprawiających błędy
- UDP jest prostym interfejsem aplikacji do IP – pracuje, jak multiplexer / demultiplexer przy otrzymywaniu i wysyłaniu ruchu IP

Format dlaUDP



Port Źródła: identyfikuje port procesu aplikacji wysyłającej. Pole jest OPCJONALNE. Jeżeli nie jest używane, wypełnia się je zerami.

- **Port przeznaczenia:** identyfikuje proces otrzymujący na hoście przeznaczenia
- **Długość** – podaje długość datagramu użytkownika włączając nagłówek i dane. Wartość jako minimum może być 8 bajtów
- **Suma kontrolna** – dokładanie jedynek do 16-bitów - pseudonagłówek IP, nagłówek UDP oraz dane

Random Early Detection (RED)

- RED losowo odrzuca pakiety w oparciu o liczbę pakietów w kolejce interfejsu; gdy kolejka zapełnia się do maksymalnej pojemności, RED odrzuca pakiety bardziej agresywnie ale pozwala to uniknąć tail drop
- RED jest rozwiązaniem dla piłokształtnego wzoru wywołanego przez TCP Slow Start

Problemy piłokształtnego wzoru:

- W całym czasie połączenie nie jest używane w 100% a całkowity przepływ nie osiąga wartości optymalnej
- Prędkości transmisji mają powtarzalne start i stop; przepływ jest niejednakowy
- Trudno jest określić możliwość przydziału zasobów i uaktualnień sieci

RED jest podobne do wlewania
wody do lejka – coraz wolniej, gdy
lejek zaczyna się wypełniać

WRED – Ważony RED

- Kombinacja RED i poziomu pierwszeństwa IP: aplikacje o wysokim priorytecie z mniejszym prawdopodobieństwem doświadczają odrzucania pakietów (oraz TCP slow start) niż te o priorytecie niższym

Committed Access Rate (CAR)

- Używana jest do kontroli przepustowości wchodzącej lub wychodzącej z interfejsu.
(również: ograniczanie prędkości, ustawianie właściwej polityki...)
- CAR może być wykorzystane do ograniczenia przepustowości od danego źródła lub aplikacji
- Ruch przekraczający podany próg może być odrzucony lub przeklasyfikowany (z wykorzystaniem priorytetu IP)

Protokoły warstwy aplikacji

- TELNET dla usług terminalu
- TFTP (Trivial File Transfer Protocol) dla prostych usług transferu plików
- FTP (File Transfer Protocol) dla bardziej złożonych usług transferu plików
- SMTP (Simple Mail Transfer Protocol) dla usług transferu komunikatów
- Outlook, Netscape