
Wstęp

Sieci komputerowe

Wykład 1

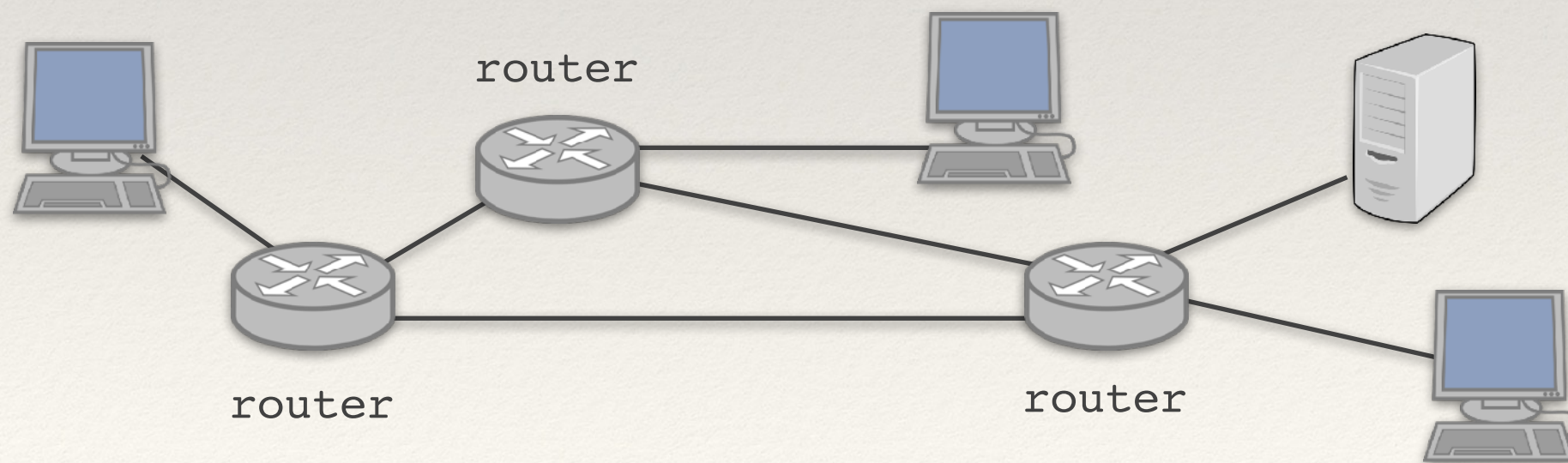
Marcin Bieńkowski

Cel przedmiotu

- ❖ Przedstawienie koncepcji leżących u podstaw **sieci komputerowych...**
- ❖ ... na przykładzie **Internetu**.

Sieć komputerowa?

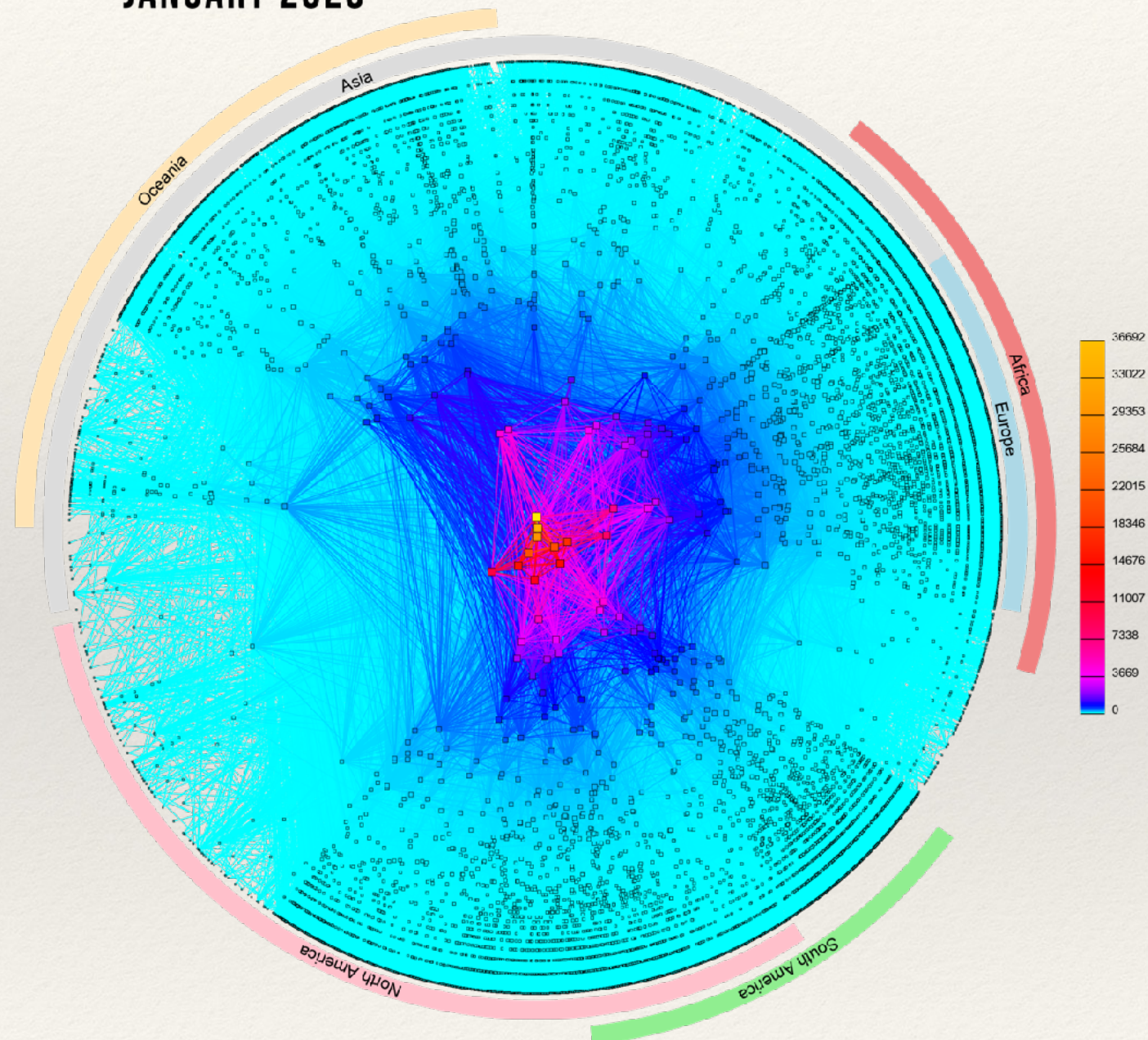
- ❖ **Urządzenia końcowe (*hosts*):** komputery, telefony, tablety, konsole, sprzęt AGD, zegarki, samochody, sensory, ...
- ❖ **Infrastruktura umożliwiająca przesyłanie danych pomiędzy nimi:**
 - ♦ **kanały komunikacyjne:** kable, światłowody, fale radiowe, ...
 - ♦ **urządzenia pośredniczące:** routery, przełączniki, ...



Internet?

Internet w 2020 roku:

CAIDA'S IPV4 AS CORE GRAPH
JANUARY 2020



Obrazek ze strony

<https://www.caida.org/projects/cartography/as-core/2020/>

COPYRIGHT © 2020 UC REGENTS

Internet 2024

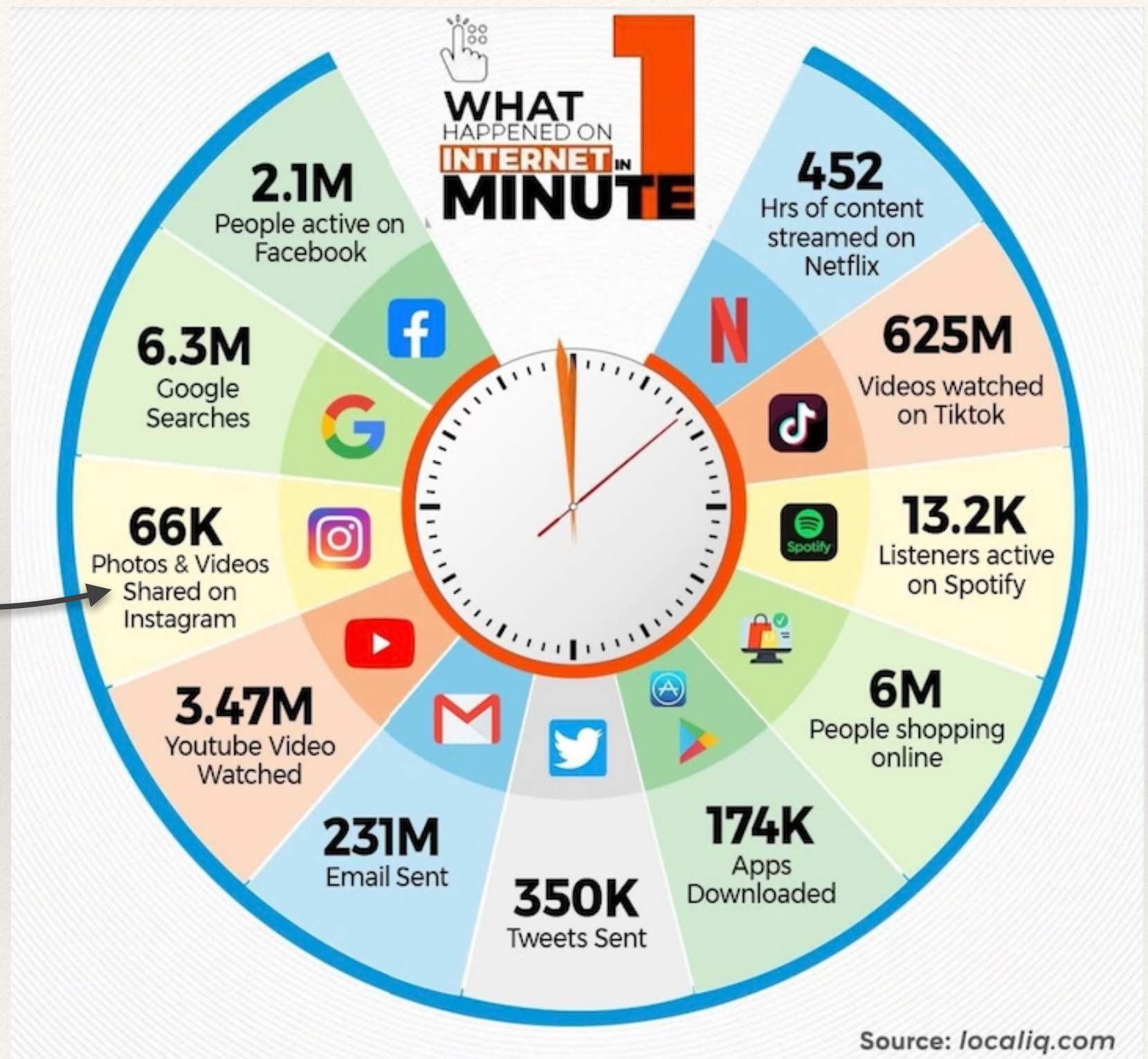
Trochę liczb i nie tylko:

- ❖ 5,5 mld użytkowników (68% populacji).
 - ❖ 5,2 mld używa sieci społecznościowych.
- ❖ 30 mld podłączonych urządzeń.
 - ❖ Internet rzeczy (IoT).
- ❖ 350 mln domen, 4-6 mld stron WWW.
 - ❖ Niedostępnych dla wyszukiwarek: 100-1000 mld stron WWW.
- ❖ 10-20 mln włamań rocznie.
- ❖ Łączy szkieletowe przez które można przesyłać ~100 terabitów / sek.
- ❖ Wszystko jako usługa (XaaS).

Jedna minuta w Internecie



Jedna minuta w Internecie



~ 10h nowych
filmików z kotami

Różnorodność

- ❖ Od sensorów do superkomputerów.
- ❖ Przepustowości od 1 kbit/s do 100 Gbit/s.
- ❖ Aplikacje: telewizja, transfer plików, wideokonferencje, gry, sieci społecznościowe, komunikatory.
- ❖ Technologie: bezprzewodowe, satelitarne, miedziane, optyczne.

Decentralizacja i dynamika

Internet = sieć łącząca różne sieci

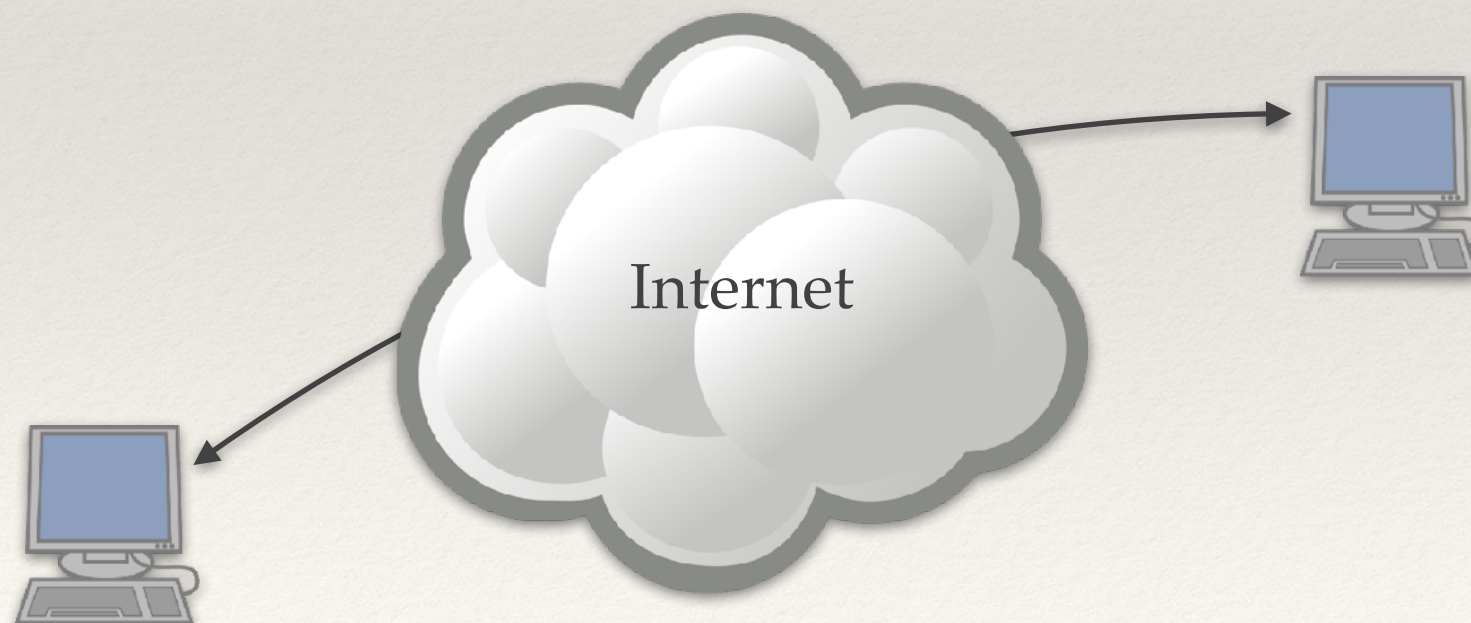
- ❖ Różne technologie (WIFI, 2G / 3G / 4G / 5G, kable, światłowody, łącza satelitarne, Bluetooth, ...).
- ❖ Brak centralnej kontroli.
- ❖ Obecnie ~20 tys. ISP (dostawców internetu), ~100 tys. tzw. systemów autonomicznych (AS).
- ❖ Ciągłe zmiany (hardware i software).
- ❖ Błędy i ataki.



Główne zastosowanie sieci

Dwukierunkowy niezawodny kanał:

- ❖ Umożliwia komunikację między aplikacjami na różnych komputerach.
- ❖ Aplikacje definiują **protokół komunikacji** określający, jak wygląda przesyłany **strumień danych**.
- ❖ Na razie abstrahujemy tutaj od tego, jak sieć to przesyła; chcemy mieć potok pomiędzy dwoma procesami.



Przykład: protokół HTTP (przeglądarka i serwer WWW)

Przeglądarka (klient WWW) wysyła do serwera:

GET / HTTP/1.1

Host: example.org

Accept: text/html,application/xhtml+xml

Accept-Language: en-US,en;q=0.9,pl;q=0.8

User-Agent: Mozilla/5.0 (Macintosh)



Przykład: protokół HTTP (przeglądarka i serwer WWW)

Przeglądarka (klient WWW) wysyła do serwera:

GET / HTTP/1.1

Host: example.org

Accept: text/html,application/xhtml+xml

Accept-Language: en-US,en;q=0.9,pl;q=0.8

User-Agent: Mozilla/5.0 (Macintosh)



Serwer (serwer WWW) wysyła do przeglądarki:

HTTP/1.1 200 OK

Content-Length: 648

Content-Type: text/html

Date: Mon, 01 Mar 2021 10:36:19 GMT

Last-Modified: Thu, 17 Oct 2019 07:18:26 GMT

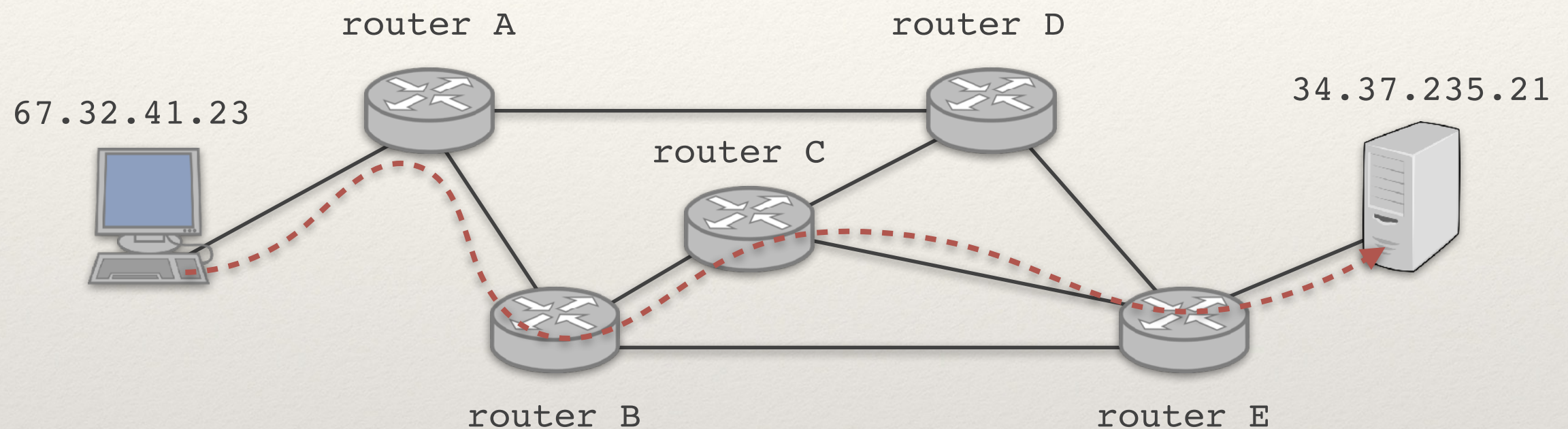
Server: Apache/2.2.22 (Debian)



demonstracja

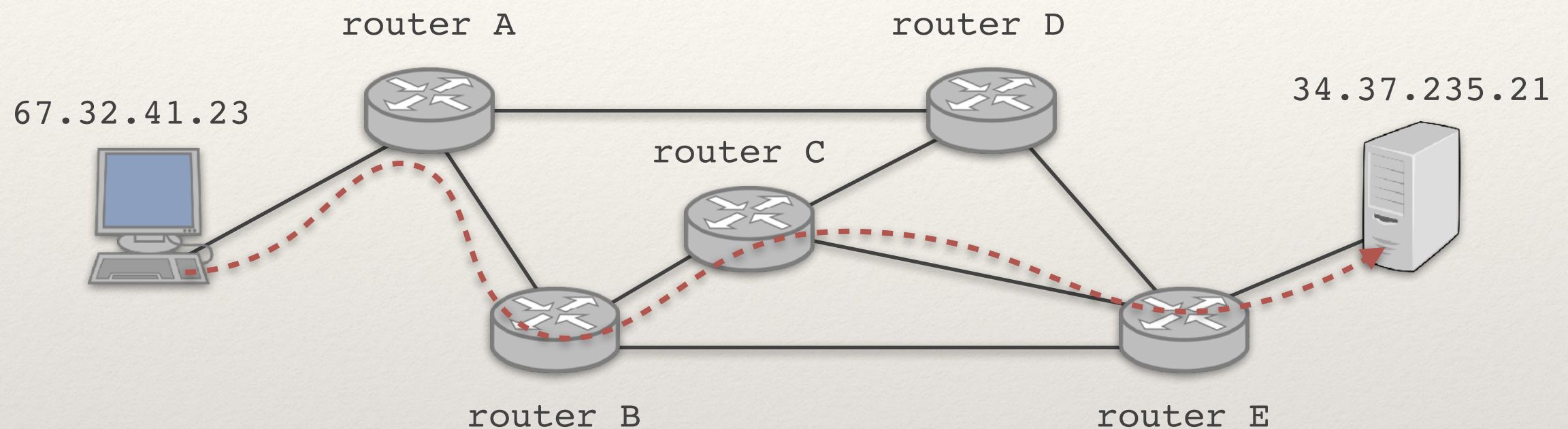
Jak przesyłać dane przez sieć

- ❖ **Globalne adresowanie:** w Internecie każde urządzenie (właściwie każda karta sieciowa) ma unikatowy 4-bajtowy adres IP.



Jak przesyłać dane przez sieć

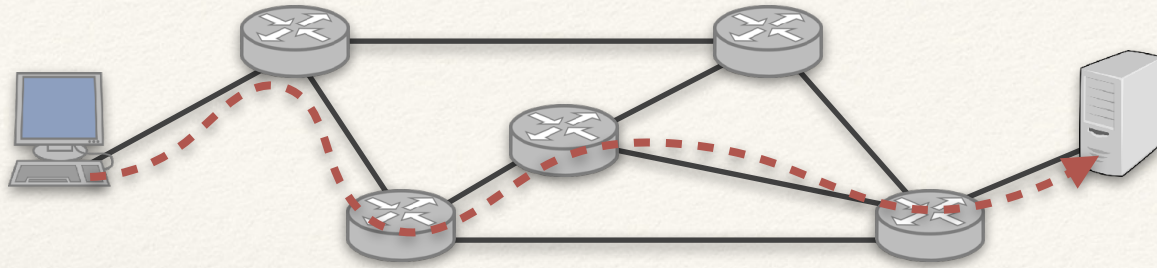
- ❖ **Globalne adresowanie:** w Internecie każde urządzenie (właściwie każda karta sieciowa) ma unikatowy 4-bajtowy adres IP.



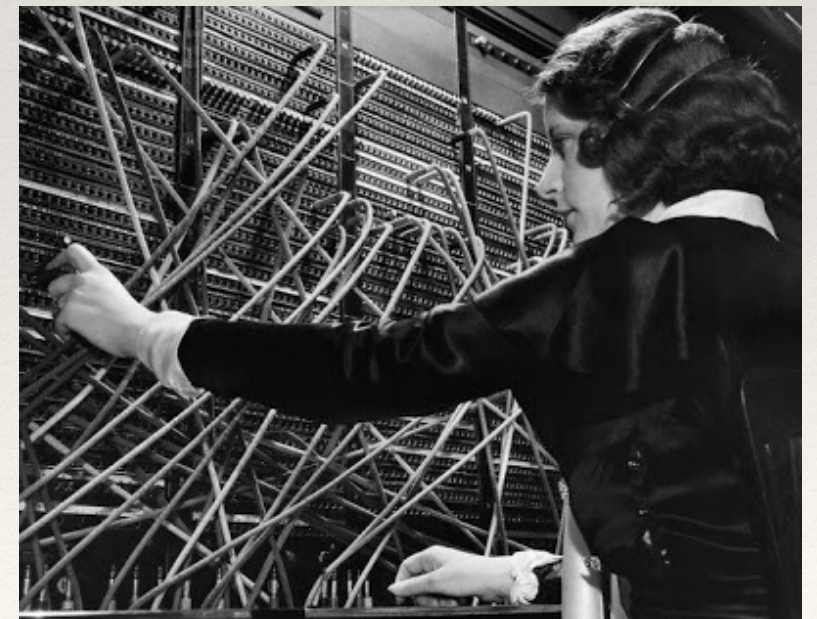
- ❖ Dwa paradygmaty przesyłania danych:
 - ♦ przełączanie obwodów,
 - ♦ przełączanie pakietów.

Przełączanie obwodów i pakietów

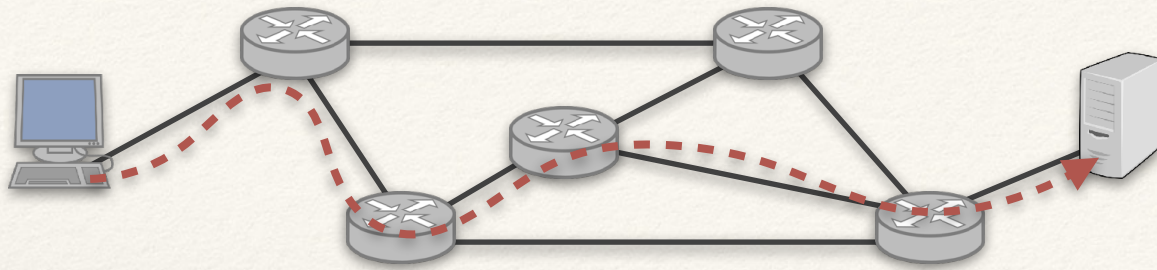
Przełączanie obwodów (1)



- ❖ Koncepcja jak w analogowych centralach telefonicznych.
- ❖ Ścieżka dla strumienia danych **ustalana na cały czas komunikacji**.
- ❖ Zasoby wymagane przez ścieżkę (bufory routerów, szybkość łącza) **rezerwowane** na czas komunikacji.
- ❖ **Gwarantowana stała szybkość.**

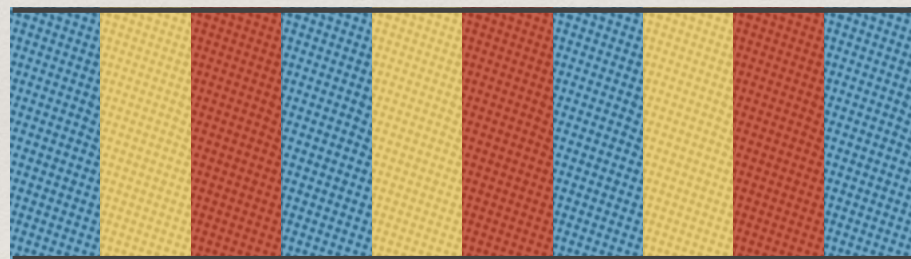


Przełączanie obwodów (2)



Multipleksowanie: rezerwujemy tylko fragment łącza dla jednego strumienia danych.

z podziałem czasu (TDM)



z podziałem częstotliwości (FDM)

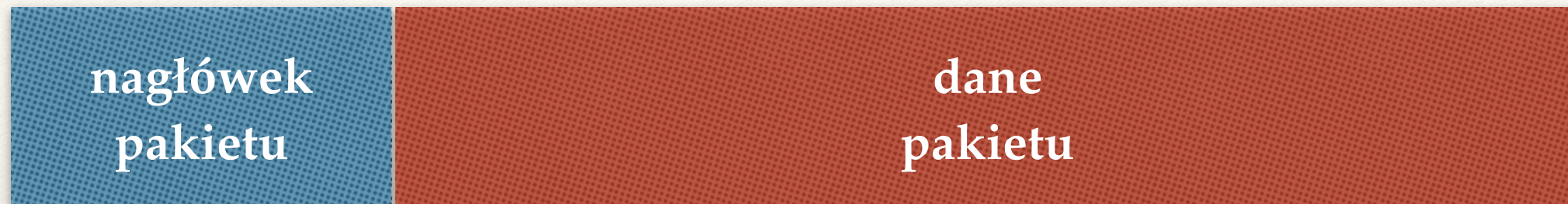


czas

- ❖ Gwarantowana stała szybkość.
- ❖ Ale marnotrawstwo łącza jeśli akurat nic nie wysyłamy.

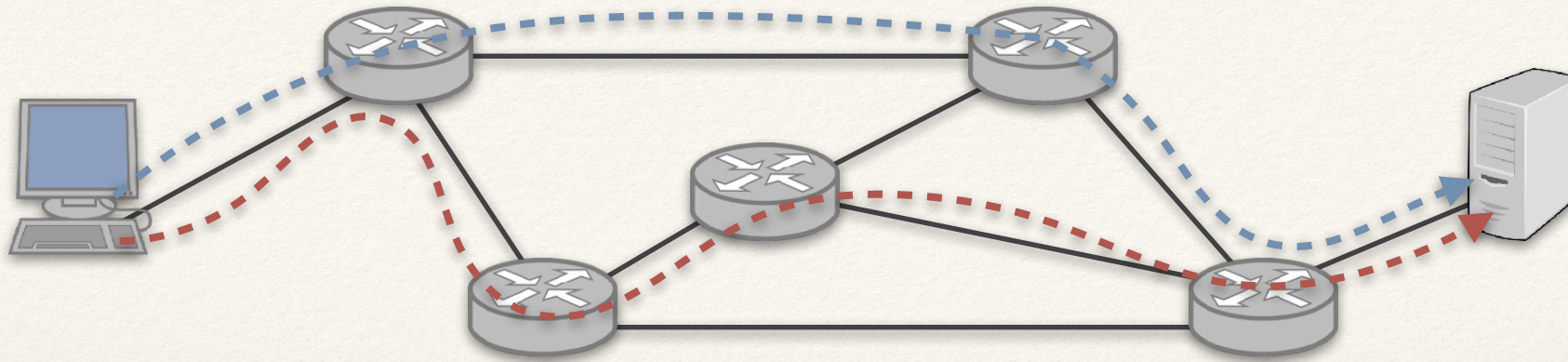
Przełączanie pakietów (1)

- ❖ Wysyłany strumień danych dzielimy na małe porcje: **pakiety**.



- ❖ Dane pakietu = fragment strumienia danych.
- ❖ Nagłówek pakietu = informacje kontrolne, m.in. adres źródłowy i docelowy.
- ❖ Każdy pakiet jest **przesyłany niezależnie**.
- ❖ Dla każdego pakietu router podejmuje niezależną decyzję gdzie przesłać go dalej (na podstawie informacji z nagłówka).

Przełączanie pakietów (2)



- ❖ **Routing (trasowanie) = wybór trasy dla danego pakietu.**
- ❖ Wysyłane pakiety mogą docierać do celu różnymi drogami.
- ❖ Router: tylko przekazuje pakiet dalej.
 - ♦ Nie musi wiedzieć nic o oryginalnym strumieniu danych.
 - ♦ Prostota → niski koszt, łatwa implementacja.
 - ♦ Brak stanu → łatwość odtwarzania po awarii.

Rodzaje routingu

- ❖ **Routing źródłowy**
 - ♦ Nagłówek pakietu zawiera całą trasę do celu.

Rodzaje routingu

❖ Routing źródłowy

- ♦ Nagłówek pakietu zawiera całą trasę do celu.

❖ Wykorzystujący tablice routingu

- ♦ Router utrzymuje pewien stan nazywany **tablicą routingu**.
 - Zbiór reguł typu „jeśli adres docelowy pasuje do wzorca A, przekaz pakiet do sąsiedniego routera X”.
- ♦ Jak ustawiać tablice routingu?

Rodzaje routingu

- ❖ **Routing źródłowy**
 - ♦ Nagłówek pakietu zawiera całą trasę do celu.
- ❖ **Wykorzystujący tablice routingu**
 - ♦ Router utrzymuje pewien stan nazywany **tablicą routingu**.
 - Zbiór reguł typu „jeśli adres docelowy pasuje do wzorca A, przekaz pakiet do sąsiedniego routera X”.
 - ♦ Jak ustawiać tablice routingu?
- ❖ **Wirtualne przełączanie obwodów**
 - ♦ Nadawca najpierw wysyła pakiet(y) kontrolny ustanawiający ścieżkę do celu i konfiguruje routery, czasem rezerwując część łącza.
 - ♦ Kolejne pakiety przesyłane są tą samą ścieżką.

Przykładowa trasa

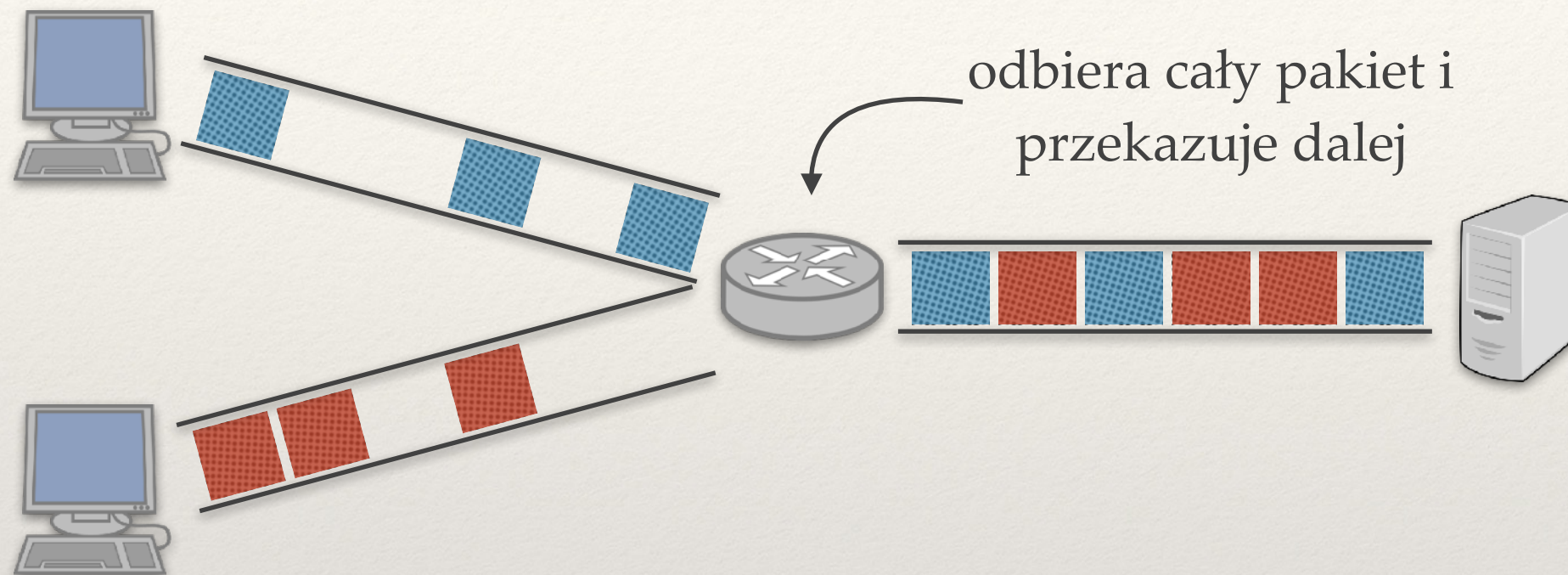
traceroute to www.ii.uni.wroc.pl (156.17.4.3), 64 hops max, 52 byte packets

```
1  192.168.1.1 (192.168.1.1)  1.525 ms
2  wro-bng2.tpnet.pl (80.50.18.74)  31.965 ms
3  wro-r2.tpnet.pl (80.50.122.73)  31.870 ms
4  lodz-ar3.tpnet.pl (213.25.5.206)  62.835 ms
5  z-tpnetu.lodz-gw.rtr.pionier.gov.pl (80.50.231.26)  37.103 ms
6  lodz-gw2.z-lodz-gw.rtr.pionier.gov.pl (212.191.126.77)  37.606 ms
7  z-lodz-gw.wroclaw.10gb.rtr.pionier.gov.pl (212.191.225.34)  44.687 ms
8  rolnik-karkonosz.wask.wroc.pl (156.17.254.112)  46.707 ms
9  archi-rolnik.wask.wroc.pl (156.17.254.108)  47.936 ms
10 matchem-archi.wask.wroc.pl (156.17.254.142)  47.986 ms
11 gwuwrmatchem.uni.wroc.pl (156.17.252.33)  49.342 ms
12 www.ii.uni.wroc.pl (156.17.4.3)  48.511 ms
```

demonstracja

Przełączanie pakietów: multipleksowanie (1)

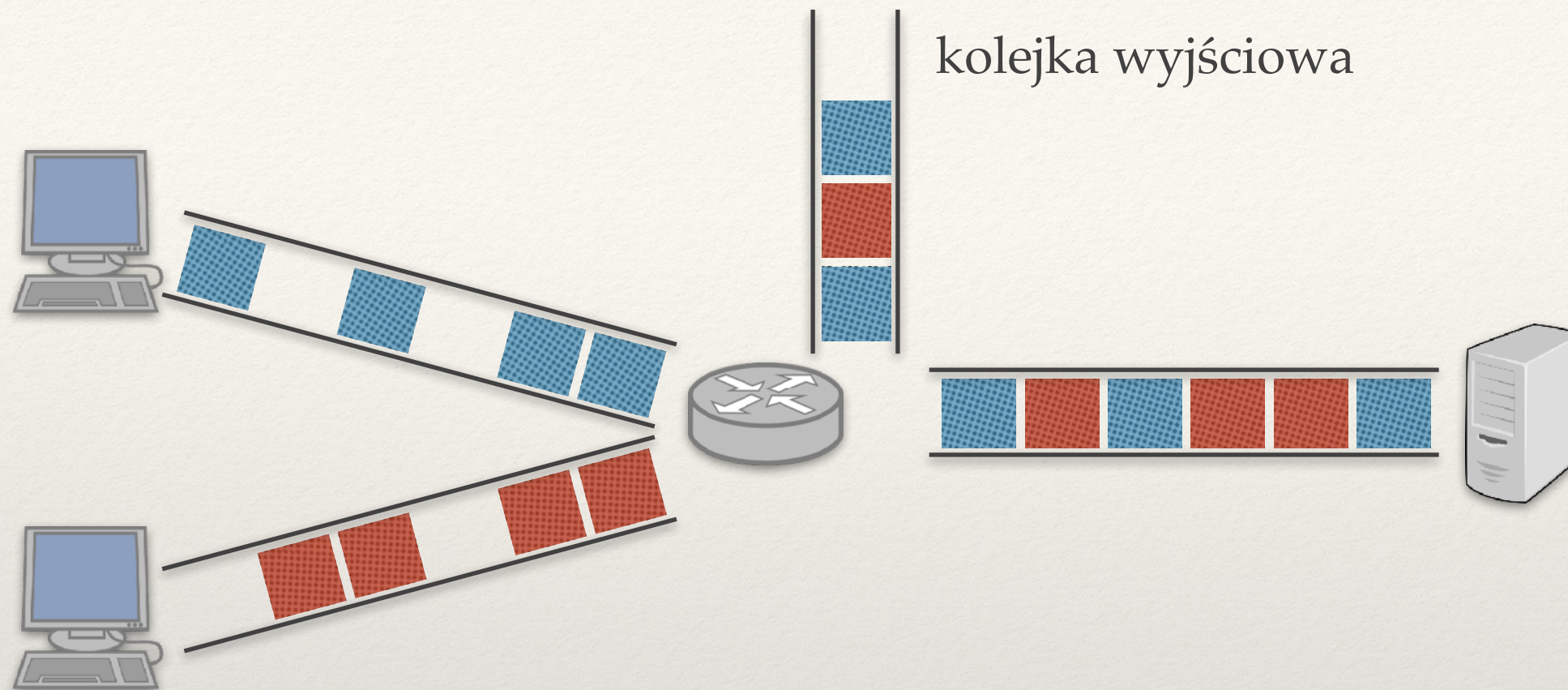
Pakiety należące do różnych strumieni danych mogą w naturalny sposób korzystać z tego samego łącza.



Multipleksowanie statystyczne:

- ❖ Założenie: różne komputery wykorzystują łącze w innych momentach czasu → lepsze wykorzystanie łącza.
- ❖ Jak radzić sobie z przejściowymi zwiększeniami ilości danych ponad możliwości łącza wyjściowego?

Przełączanie pakietów: multipleksowanie (2)



Kolejki przy łączach wyjściowych:

- ❖ Pomagają przy przejściowym przeładowaniu pakietami.
- ❖ Jeśli kolejka się przepełni (**przeciążenie**) → pakiety są odrzucane
 - ♦ główna przyczyna utraty pakietów w sieci!

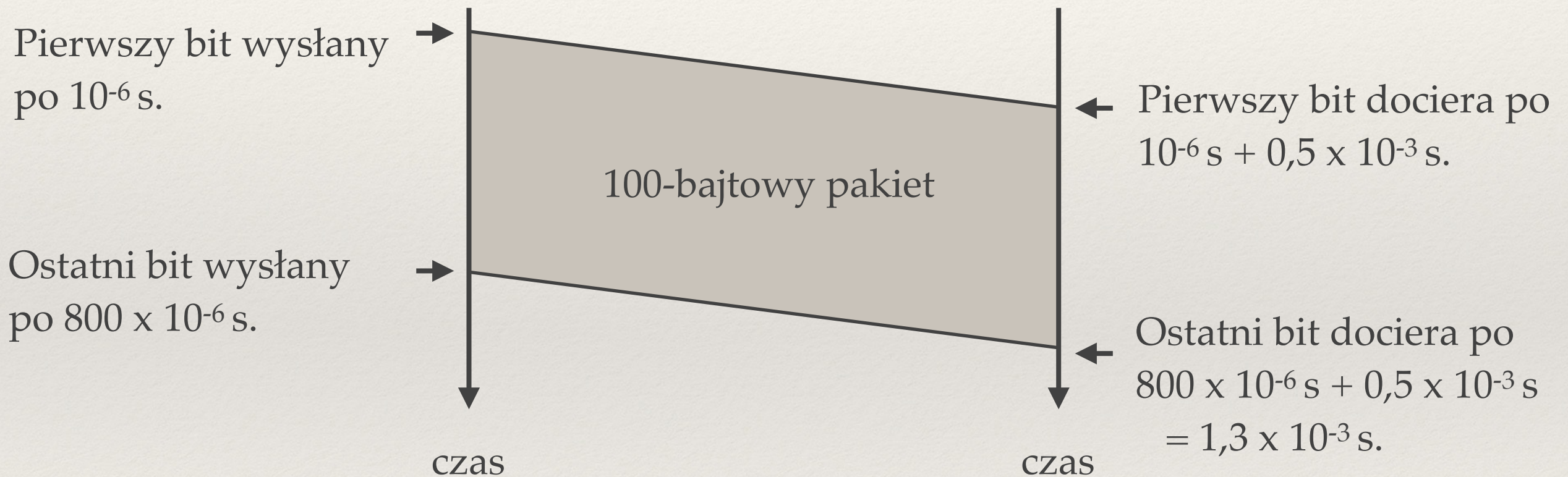
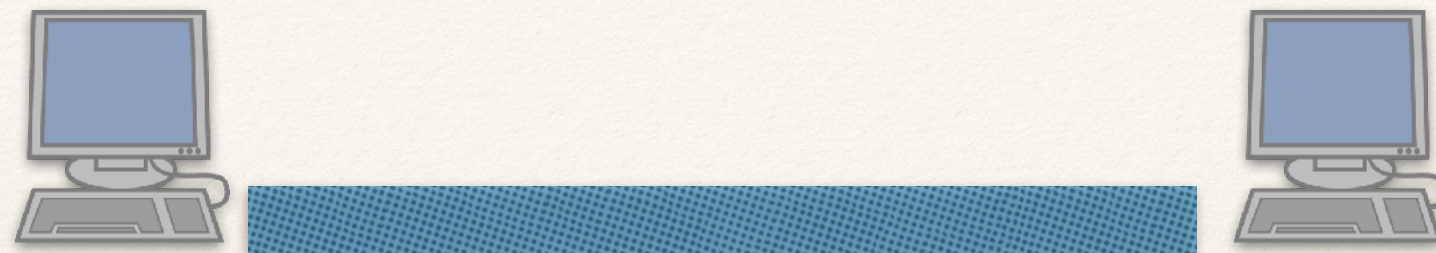
Własności kanału komunikacyjnego



- ❖ **Przepustowość:** ile możemy zapisać do kanału na jednostkę czasu.
- ❖ **Czas propagacji:** ile czasu podróżuje sygnał między końcami kanału.
- ❖ **RTT (Round Trip Time):** $2 \times$ czas propagacji.
- ❖ **BDP (bandwidth-delay product):** iloczyn przepustowości i RTT = „ile możemy wysłać zanim dostaniemy odpowiedź od odbiorcy”.
- ❖ **Duplex:** pełny, półduplex, simplex.

Opóźnienie pakietu

Przepustowość: 1 Mbit/s, czas propagacji: 0,5 ms.



**Opóźnienie pakietu na łączy = czas oczekiwania pakietu w kolejce
+ rozmiar pakietu / przepustowość + czas propagacji.**

Przełączanie obwodów vs. przełączanie pakietów

przełączanie obwodów	przełączanie pakietów
gwarancja przepustowości	brak gwarancji
narzut czasowy na nawiązanie połączenia	narzut czasowy dla każdego pakietu (nagłówek)
marnowanie łącza jeśli są przerwy w strumieniu danych	efektywne wykorzystanie łącza (statystyczny multipleksing)
wolne odtwarzanie w przypadku awarii	odporne na awarie: wybierana inna ścieżka routingu
skomplikowane	prostsze

Komunikacja oparta na pakietach

Przełączanie pakietów w Internecie: protokół IP

❖ Podstawowe cechy:

- ♦ Zaimplementowany na każdym urządzeniu.
- ♦ Definiuje adresy urządzeń (4-bajtowe identyfikatory).
- ♦ Definiuje **zawodną, bezpołączeniową** usługę umożliwiającą przesłanie pakietu między dwoma dowolnymi urządzeniami w sieci.

Przełączanie pakietów w Internecie: protokół IP

- ❖ **Podstawowe cechy:**
 - ♦ Zaimplementowany na każdym urządzeniu.
 - ♦ Definiuje adresy urządzeń (4-bajtowe identyfikatory).
 - ♦ Definiuje **zawodną, bezpołączeniową** usługę umożliwiającą przesłanie pakietu między dwoma dowolnymi urządzeniami w sieci.

- ❖ **Zasada dołożenia wszelkich starań (*best effort*):**
 - ♦ Brak gwarancji (pakiety mogą gubić się lub przychodzić w innej kolejności), ale nie są gubione celowo.

Przełączanie pakietów w Internecie: protokół IP

❖ Podstawowe cechy:

- ♦ Zaimplementowany na każdym urządzeniu.
- ♦ Definiuje adresy urządzeń (4-bajtowe identyfikatory).
- ♦ Definiuje **zawodną, bezpołączeniową** usługę umożliwiającą przesłanie pakietu między dwoma dowolnymi urządzeniami w sieci.

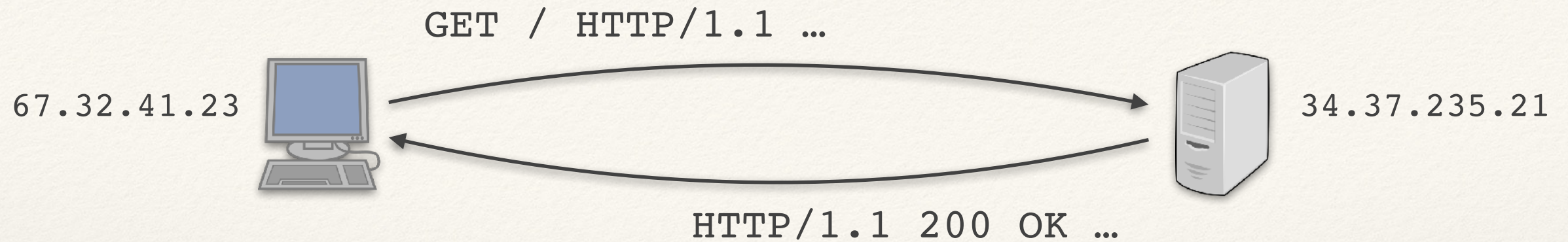
❖ Zasada dołożenia wszelkich starań (*best effort*):

- ♦ Brak gwarancji (pakiety mogą gubić się lub przychodzić w innej kolejności), ale nie są gubione celowo.

❖ Zasada end-to-end:

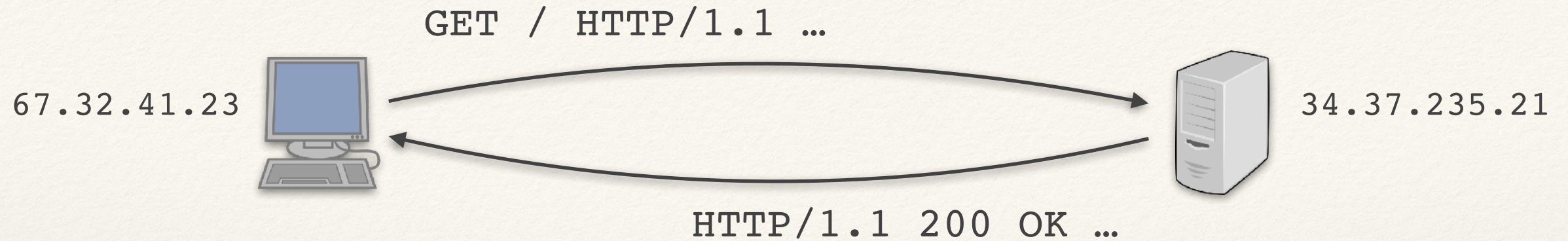
- ♦ Wszystkie dodatkowe cechy (np. niezawodne przesyłanie danych) implementowane w urządzeniach końcowych (komputerach) → łatwa ewolucja, niski koszt innowacyjności.

HTTP a pakiety IP



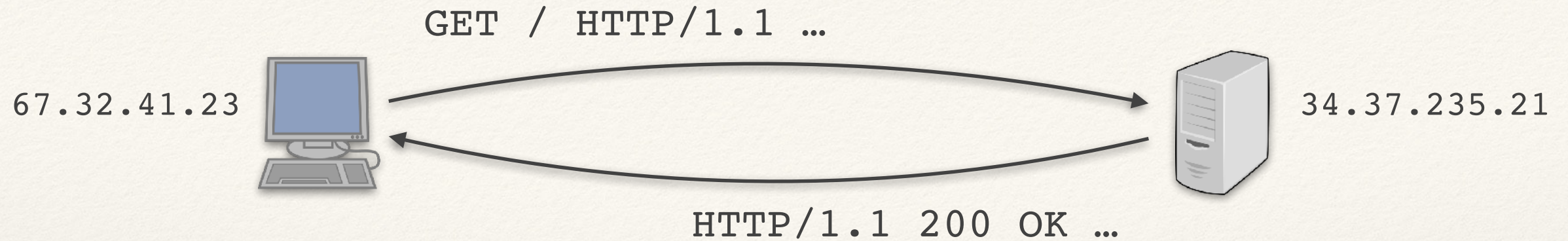
- ❖ Mamy do dyspozycji protokół IP (umiemy wysyłać pakiety).
Co potrzebne jest do scenariusza z HTTP?

HTTP a pakiety IP



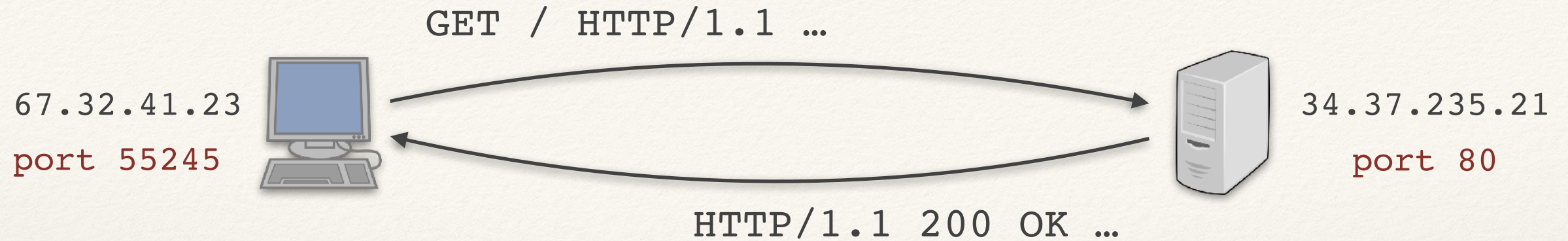
- ❖ Mamy do dyspozycji protokół IP (umiemy wysyłać pakiety).
Co potrzebne jest do scenariusza z HTTP?
 - ♦ Niezawodność przesyłania danych.

HTTP a pakiety IP



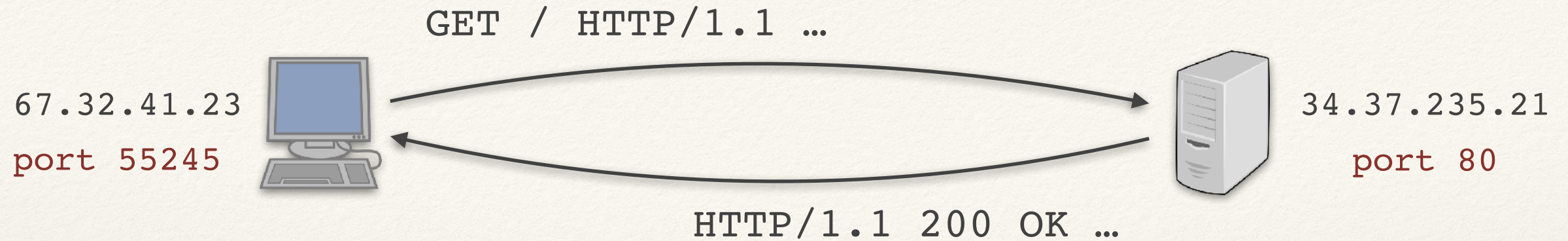
- ❖ Mamy do dyspozycji protokół IP (umiemy wysyłać pakiety).
Co potrzebne jest do scenariusza z HTTP?
 - ♦ Niezawodność przesyłania danych.
 - ♦ Konieczność dzielenia na pakiety.

HTTP a pakiety IP



- ❖ Mamy do dyspozycji protokół IP (umiemy wysyłać pakiety).
Co potrzebne jest do scenariusza z HTTP?
 - ♦ Niezawodność przesyłania danych.
 - ♦ Konieczność dzielenia na pakiety.
 - ♦ Jak rozróżnić różne przeglądarki działające na tym samym komputerze? → identyfikatory aplikacji (**porty**).

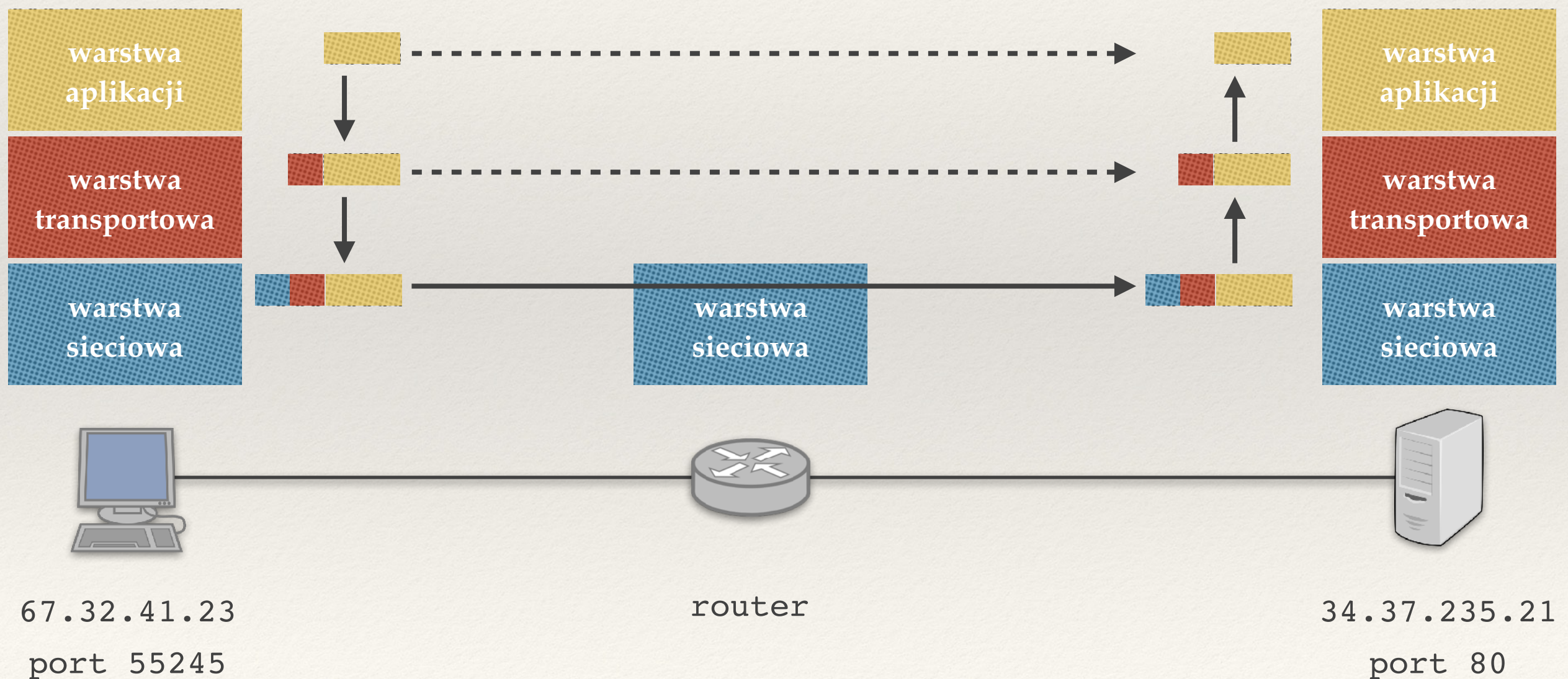
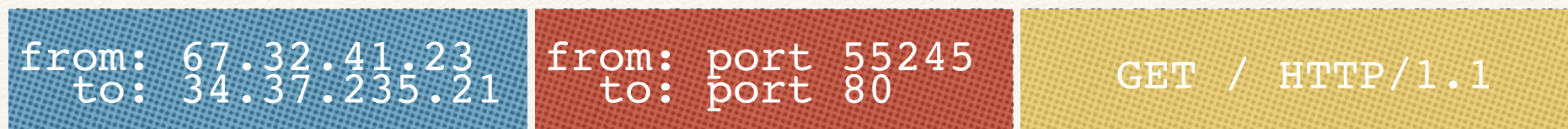
HTTP a pakiety IP



- ❖ Mamy do dyspozycji protokół IP (umiemy wysyłać pakiety).
Co potrzebne jest do scenariusza z HTTP?
 - ♦ Niezawodność przesyłania danych.
 - ♦ Konieczność dzielenia na pakiety.
 - ♦ Jak rozróżnić różne przeglądarki działające na tym samym komputerze? → identyfikatory aplikacji (**porty**).
- ❖ Typowe problemy → warto wydzielić osobny komponent a nie wbudowywać tych funkcji w protokół HTTP.

Model warstwowy (1)

Przesyłany pakiet:



Model warstwowy (1)

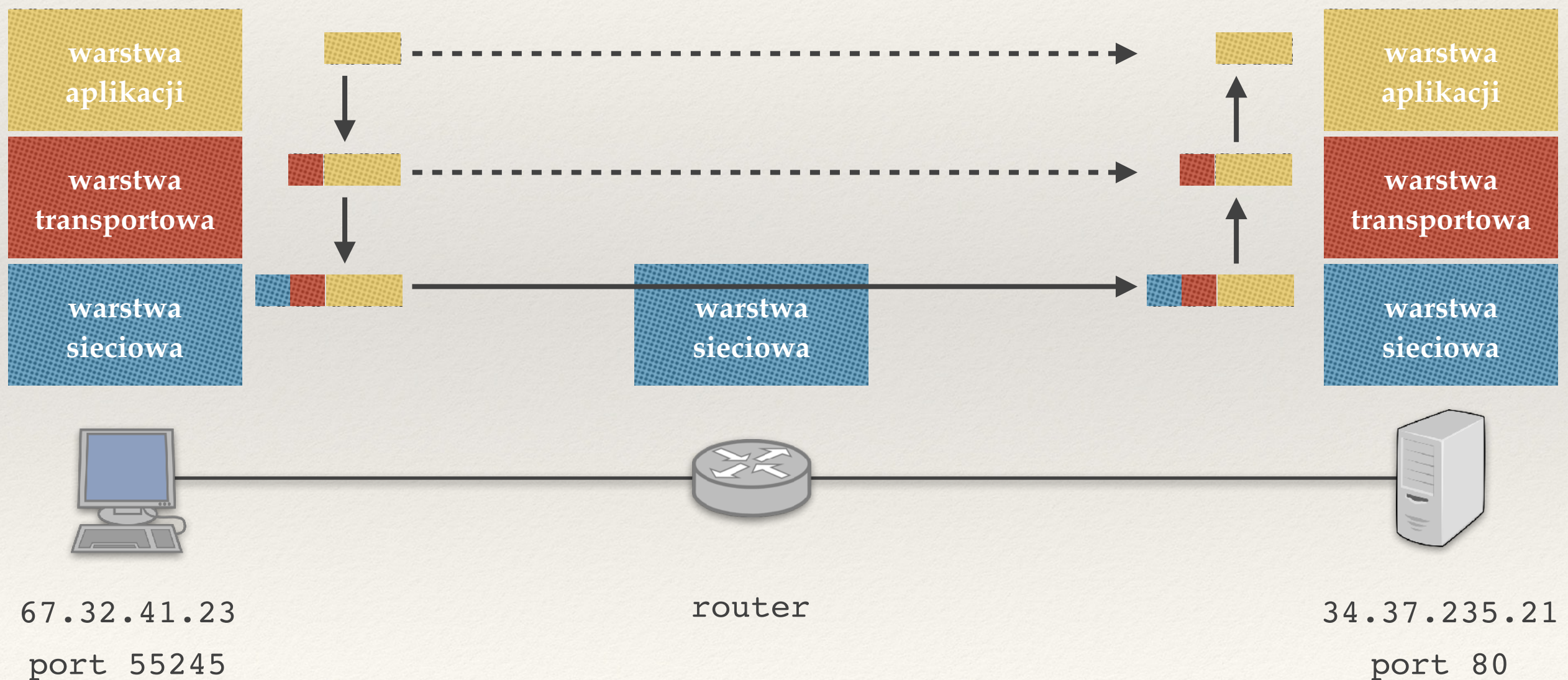
Przesyłany pakiet:

demonstracja

from: 67.32.41.23
to: 34.37.235.21

from: port 55245
to: port 80

GET / HTTP/1.1



Model warstwowy (2)

- ❖ **Warstwa definiuje:**
 - ♦ swoje zadania,
 - ♦ interfejs do sąsiednich warstw.

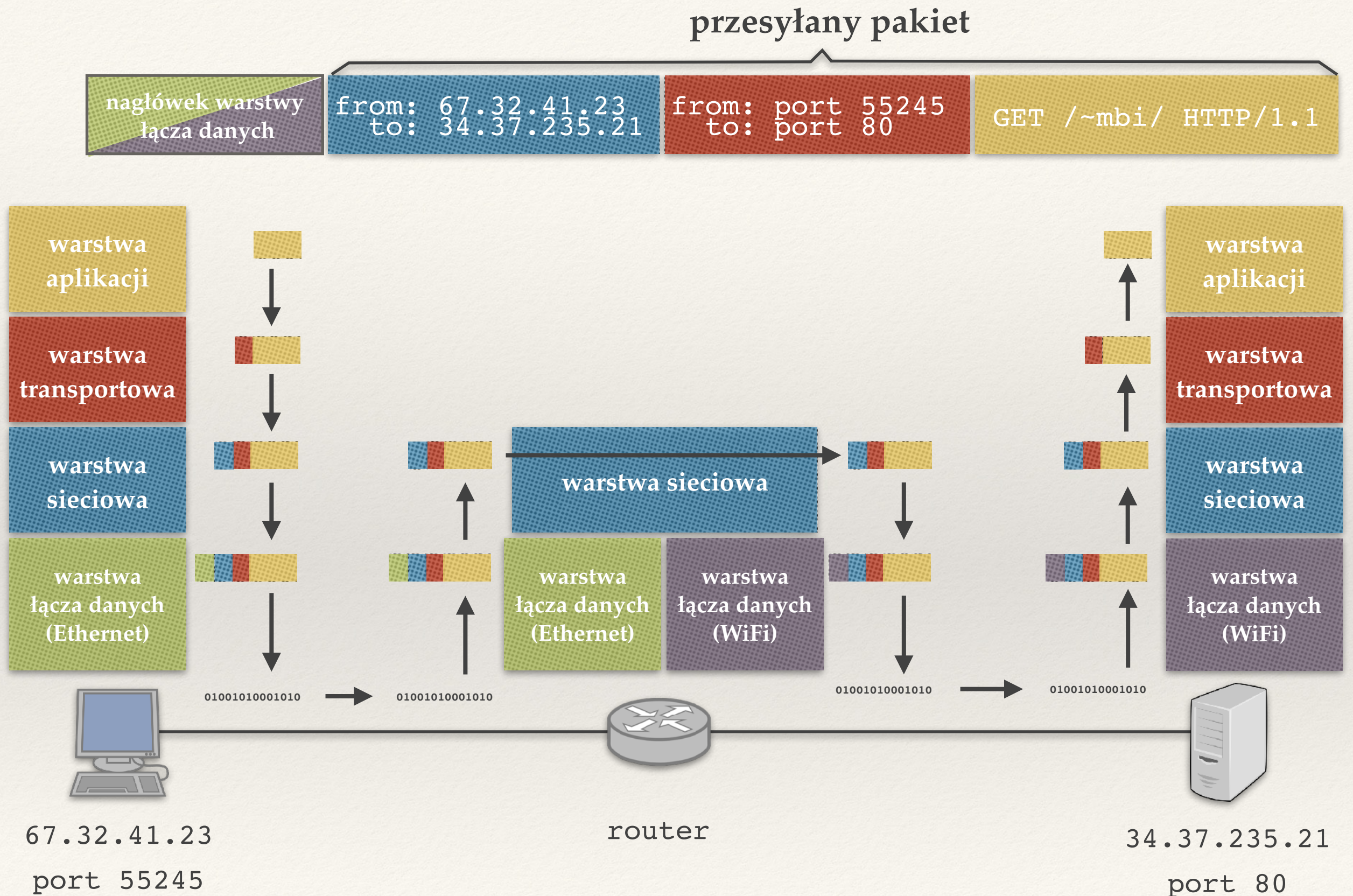
- ❖ **Modularność:** możliwość wymiany jednej implementacji warstwy na inną.

- ❖ **Enkapsulacja/dekapsulacja:** proces dodawania / usuwania nagłówka przy przechodzeniu przez kolejną warstwę.

Lokalne dostarczanie

- ❖ Pakiety IP są niezależne od architektury sprzętowej.
- ❖ Warstwa sieciowa potrzebuje procedury (zawodnego) dostarczania pakietu do sąsiedniego (bezpośrednio połączonego) komputera.
- ❖ Architektura sprzętowa definiuje kolejne dwie warstwy: łącza danych i fizyczną.

Internetowy model warstwowy (1)



Internetowy model warstwowy (2)



zapewnia **globalne** dostarczanie danych pomiędzy **aplikacjami**

zapewnia **globalne** dostarczanie danych pomiędzy **komputerami**

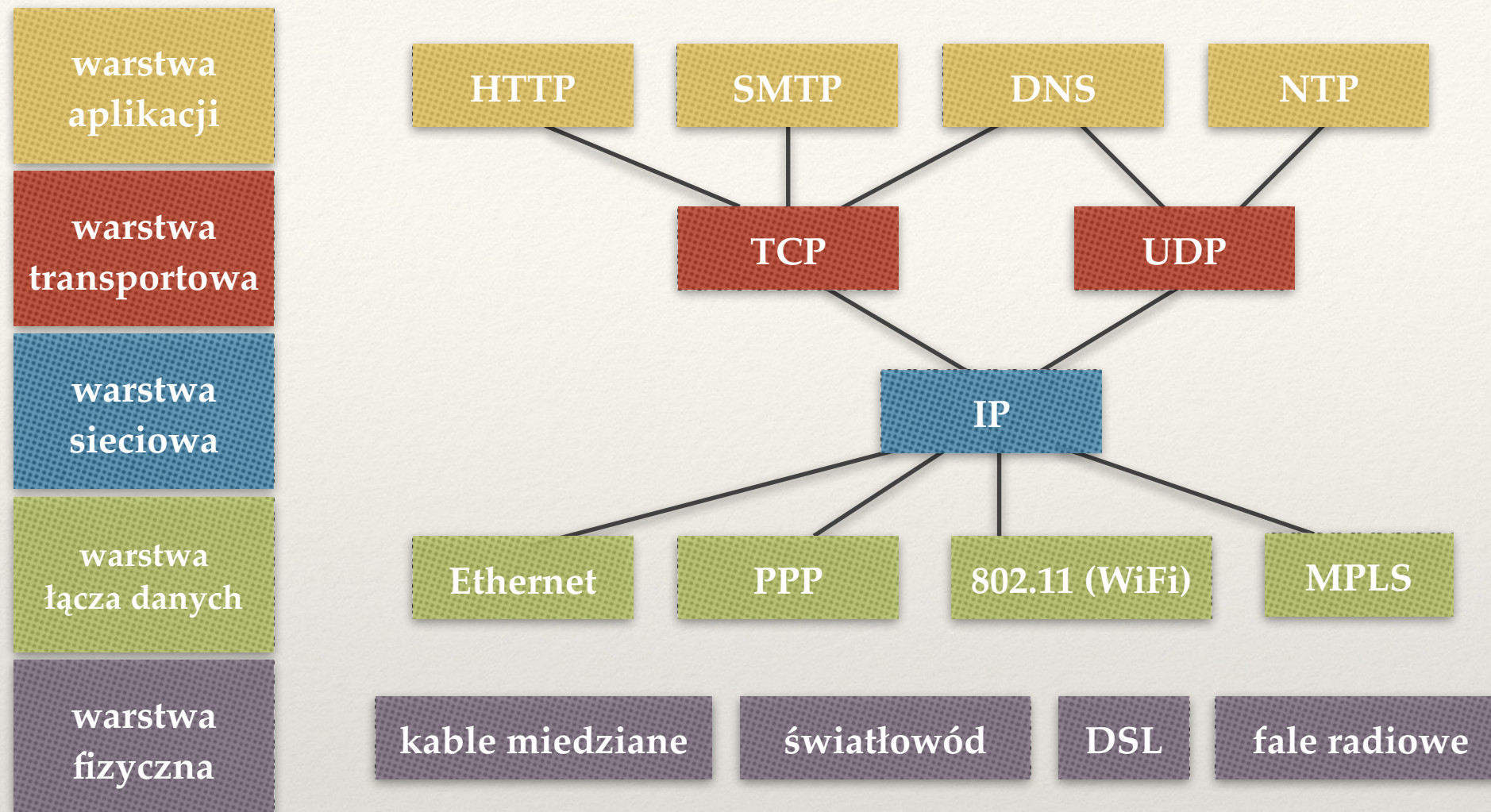
zapewnia **lokalne** dostarczanie danych pomiędzy **komputerami**

Internetowy model warstwowy (3)

	warstwa	protokoły	zadanie
5	aplikacji	HTTP, SMTP,	Protokoły użytkowników.
4	transportowa	TCP, UDP	Wprowadza porty, dzieli strumień danych na pakiety, zapewnia że pakiety dotrą, składa je w strumień danych po stronie odbiorcy.
3	sieci	IP	Routuje pakiety.
2	łącza danych	Ethernet, WiFi, ...	Przesyła ramki z pakietami, zapewnia dostęp do współdzielonego kanału.
1	fizyczna		Przesyła bity.

- ❖ Model TCP / IP: sklejona warstwa 1 i 2.
- ❖ Model OSI: dodatkowe warstwy sesji i prezentacji pomiędzy warstwą 4 i 5.

Protokoły w Internecie



- ❖ Warstwa sieciowa: tylko jeden protokół!
 - ✦ Pozwala na współpracę dowolnych sieci.
 - ✦ Umożliwia niezależny rozwój we wszystkich innych warstwach.

Lektura dodatkowa

- ❖ Kurose & Ross: rozdział 1.
- ❖ Tanenbaum: rozdział 1.
- ❖ Historia Internetu:
 - ♦ <https://www.youtube.com/watch?v=9hIQjrMHTv4>
 - ♦ <https://www.youtube.com/watch?v=h8K49dD52WA>
 - ♦ <http://www.internetsociety.org/internet/what-internet/history-internet/brief-history-internet>

Sprawy organizacyjne

Zajęcia dodatkowe (ćwiczeniopracownia)

Zajęcia dodatkowe (100 pkt. do zdobycia, zaliczenie od 40 pkt.)

- ❖ Algorytmy, kodowanie, kryptografia → ćwiczenia (20 pkt.)
- ❖ Praktyka (np. konfiguracja routingu) → warsztaty (40 pkt.)
- ❖ Zadania programistyczne → pracownia (40 pkt.)
 - ♦ Do zaliczenia: minimum 10 pkt. z zadań programistycznych

Więcej informacji na stronie przedmiotu

<https://skos.ii.uni.wroc.pl/>

- ❖ **Test wielokrotnego wyboru.**
 - ♦ Materiał z wykładu przerabiany na ćwiczeniach i warsztatach.
 - ♦ Zagadnienia i pojęcia na stronie przedmiotu.

- ❖ **Ocena na podstawie $\max \{ E, E/2 + C/2 \}$, gdzie:**
 - ♦ E = procent punktów uzyskanych z egzaminu,
 - ♦ C = procent punktów uzyskanych z zajęć dodatkowych w roku egzaminu

- ❖ **Zaliczenie od 60%.**

Wymagania wstępne

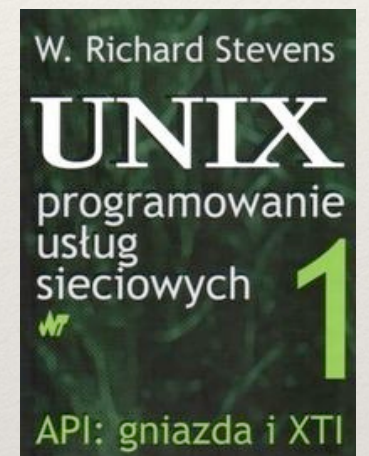
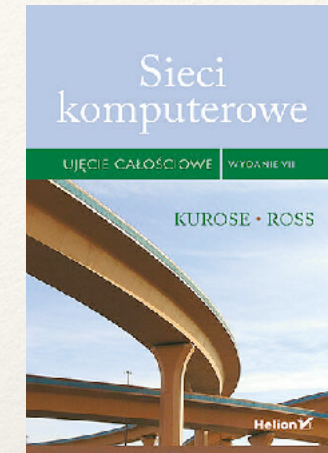
Co powinniście umieć już teraz:

- ❖ Obsługa Linuksa (wiersz poleceń, proste skrypty, automatyzacja).
 - ✦ Linux command line: <http://linuxcommand.org/tlcl.php>
- ❖ Systemy operacyjne (procesy, sygnały, IPC).
- ❖ Programowanie w C lub C++ lub Rust.
- ❖ Algebra lub Matematyka dyskretna (wielomiany, pierścienie, operacje na grupach).

Ten przedmiot nie jest przewidziany dla studentów 1. roku!

Literatura

- ❖ James F. Kurose, Keith W. Ross,
Sieci komputerowe, ujęcie całościowe
Helion, 2018, wydanie VII
(wydania V i VI też ok.)
- ❖ W. Richard Stevens
UNIX programowanie usług sieciowych, tom 1
Wydawnictwo naukowo techniczne, 2002
(ten rok to nie pomyłka)
- ❖ Andrew S. Tanenbaum,
Sieci komputerowe
Helion, 2012, wydanie V
- ❖ **Internet!**



Zagadnienia

- ❖ Co to jest protokół komunikacyjny? Dlaczego wprowadza się warsty protokołów?
- ❖ Wymień warstwy internetowego modelu warstwowego. Jakie są zadania każdej z nich?
- ❖ Jakie warstwy są zaimplementowane na komputerach a jakie na routerach?
- ❖ Czym różni się model warstwowy TCP/IP od OSI?
- ❖ Co jest potrzebne do zbudowania dwukierunkowego niezawodnego kanału?
- ❖ Porównaj wady i zalety przełączania obwodów i przełączania pakietów.
- ❖ Jakie znasz rodzaje multipleksowania? Po co i kiedy się je stosuje?
- ❖ Porównaj ze sobą rodzaje routingu.
- ❖ Co to jest komunikacja pełnoduplexowa, półduplexowa i simpleksowa?
- ❖ Do czego służy polecenie `traceroute`? Co pokazuje?
- ❖ Po co stosuje się bufor w routerach? Co to jest przeciążenie?
- ❖ Jakie są przyczyny opóźnień pakietów?
- ❖ Co to jest BDP? Co to jest czas propagacji?
- ❖ Wyjaśnij pojęcia: komunikacja simpleksowa, półduplexowa, pełnoduplexowa.
- ❖ Co umożliwia protokół IP? Co to znaczy, że protokół realizuje zasadę *best effort*?
- ❖ Jakie są zalety i wady zasady end-to-end?
- ❖ Po co wprowadza się porty?
- ❖ Wyjaśnij pojęcie enkapsulacji i deenkapsulacji.