

---

# Wstęp

Sieci komputerowe

Wykład 1

---

*Marcin Bieńkowski*

# Cel przedmiotu

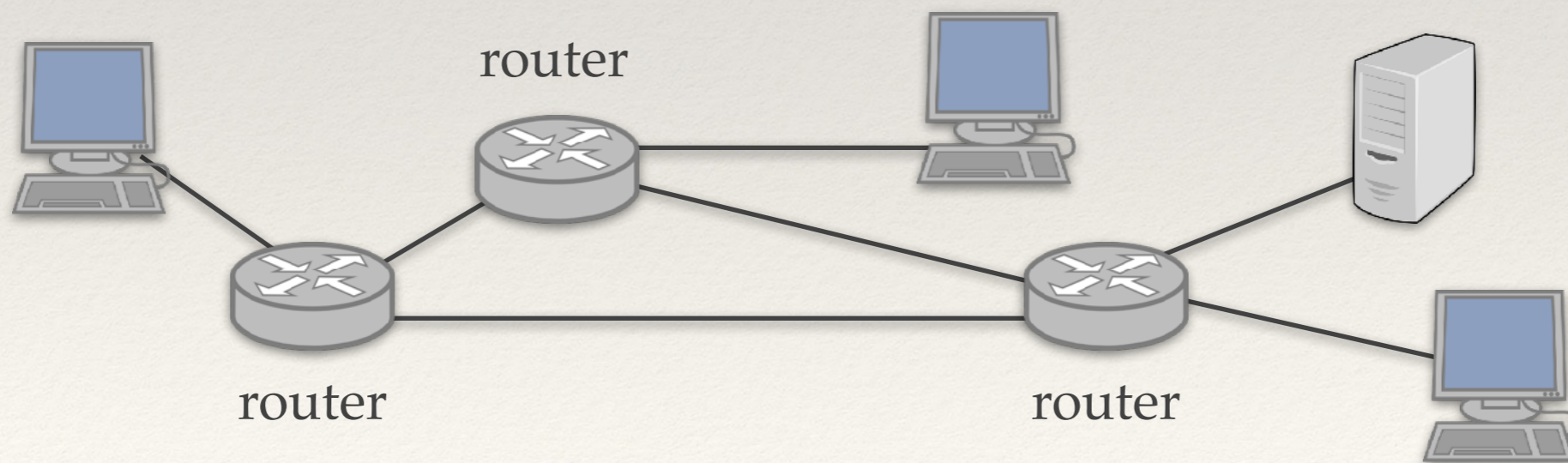
---

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

# Sieć komputerowa?

---

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



# Internet 2025 (trochę liczb)

---

- ❖ 6 mld użytkowników (73% populacji).
  - ◆ 5,7 mld tożsamości w sieciach społecznościowych.
- ❖ 31 mld podłączonych urządzeń.
  - ◆ Internet rzeczy (IoT, 21 mld urządzeń).
- ❖ 350 mln domen, 1 mld stron WWW.
  - ◆ Niedostępnych dla wyszukiwarek: 100-1000 mld stron WWW.
- ❖ 10-20 mln włamań rocznie.
- ❖ Łączy przez które można przesyłać wiele 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 10 Tbit/s.
- ❖ Aplikacje: telewizja, transfer plików, wideokonferencje, gry, sieci społecznościowe, komunikatory.
- ❖ Technologie: bezprzewodowe (Wi-Fi, 2G/3G/4G/5G), satelitarne, kable miedziane, optyczne (światłowody).

# Decentralizacja i dynamika

---

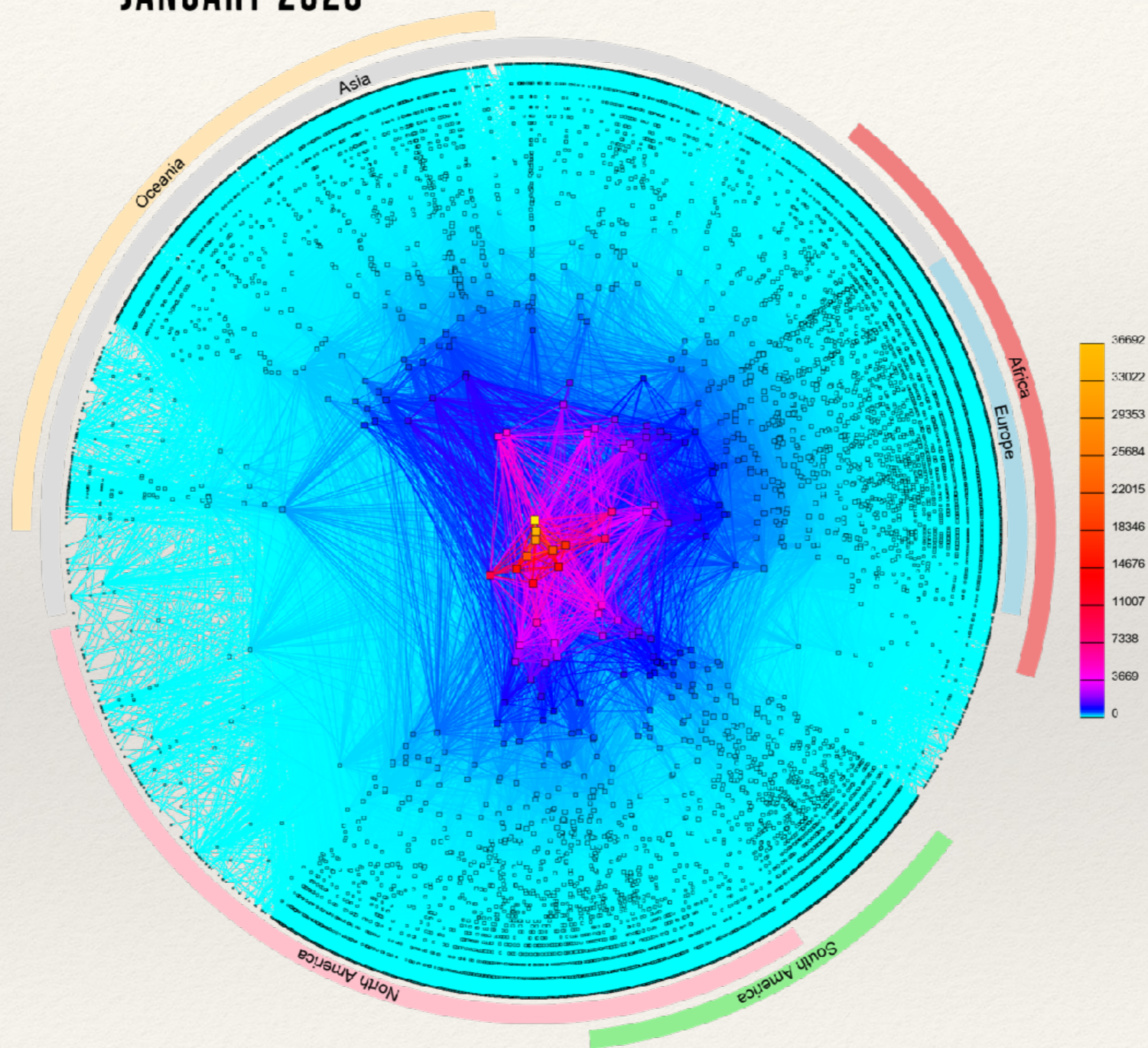
**Internet: sieć łącząca różne sieci.**

- ❖ Brak centralnej kontroli.
- ❖ Obecnie ~20 tys. ISP (dostawców internetu),  
~100 tys. tzw. systemów autonomicznych (AS).
- ❖ Ciągłe zmiany (hardware i software).
- ❖ Błędy i ataki.



# Rdzeń internetu w 2020

## CAIDA'S IPV4 AS CORE GRAPH JANUARY 2020



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

COPYRIGHT © 2020 UC REGENTS

---

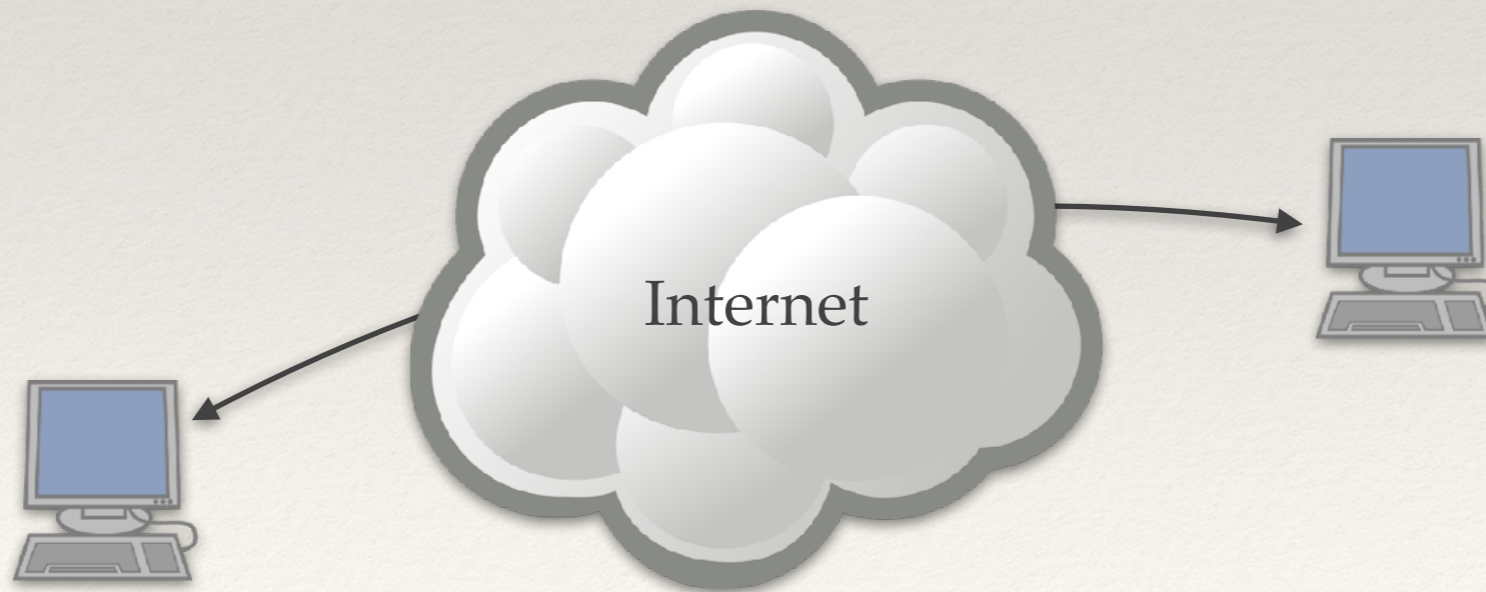
# Przesyłanie danych

---

# Dwukierunkowa niezawodna komunikacja

---

- ❖ Główne zastosowanie sieci.
- ❖ Umożliwia komunikację między aplikacjami na różnych komputerach.
- ❖ Wymaga zdefiniowania **protokołu komunikacji** określającego, jak wygląda przesyłany **strumień danych**.
- ❖ Chwilowo nie wnikamy, jak to zaimplementować.



# Przykład: protokół HTTP

---

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 (klient WWW) wysyła do serwera:

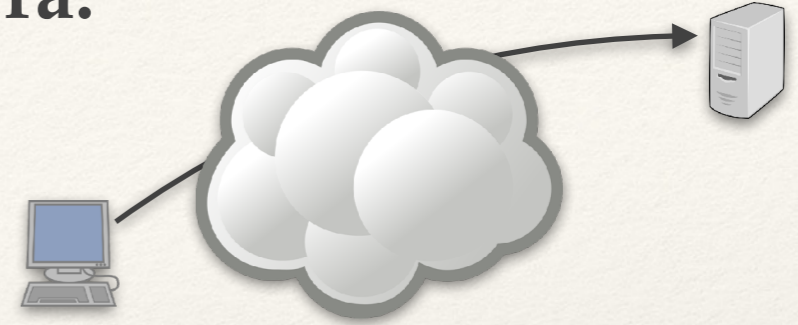
```
GET / HTTP/1.1
```

```
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
```

```
Last-Modified: Wed, 18 Feb 2026 05:33:48 GMT
```

```
Date: Tue, 24 Feb 2026 20:12:21 GMT
```



# Przykład: protokół HTTP

---

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
```

```
Last-Modified: Wed, 18 Feb 2026 05:33:48 GMT
```

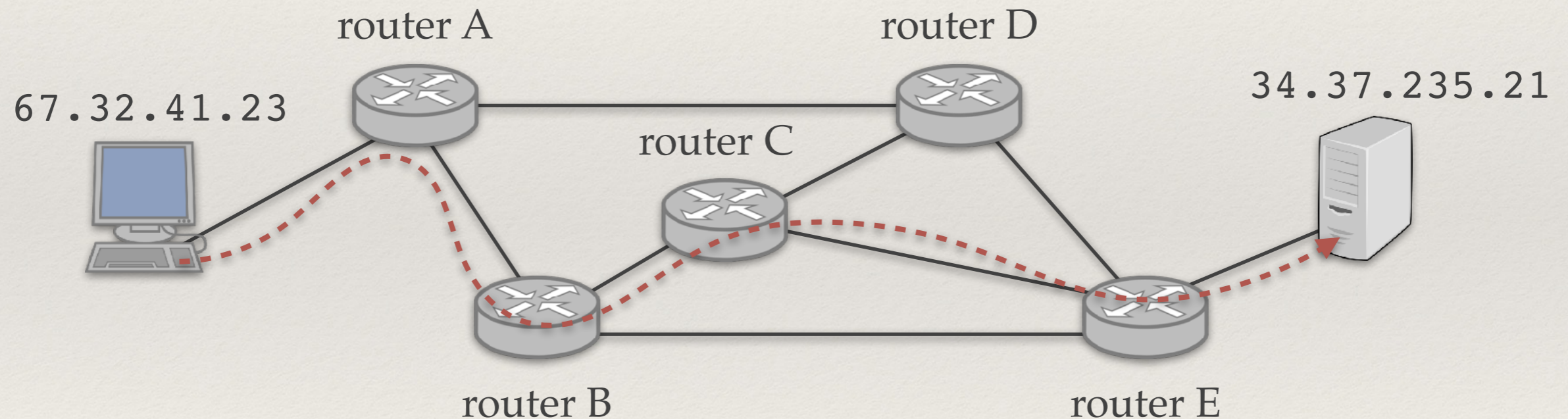
```
Date: Tue, 24 Feb 2026 20:12:21 GMT
```



demonstracja

# Globalne adresowanie

- ❖ Założenie projektowe Internetu: każde urządzenie (a właściwie każda karta sieciowa) ma **unikatowy 4-bajtowy adres IP**.
- ❖ Współcześnie to nieprawda.

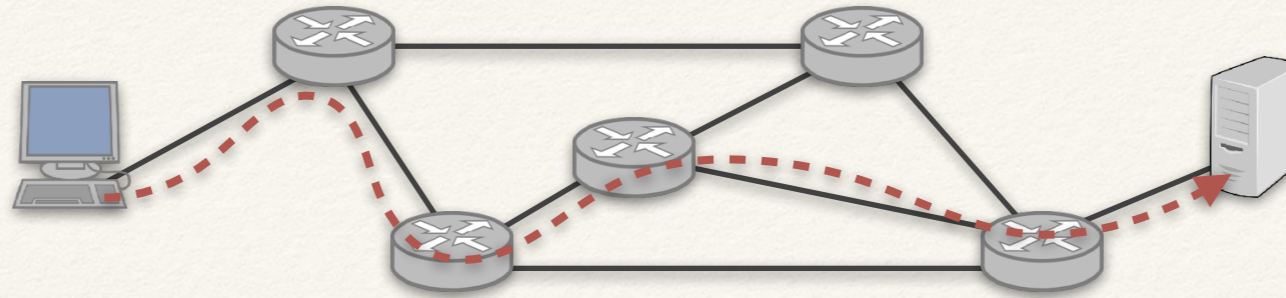


---

# Implementacja # 1: przełączanie obwodów

---

# Przełączanie obwodów



- ❖ 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 cały czas komunikacji.
- ❖ Gwarantowana stała szybkość.



Obrazek ze strony

<http://unusualhistoricals.blogspot.com/2011/02/ordinary-day-in-life-of-telephone.html>

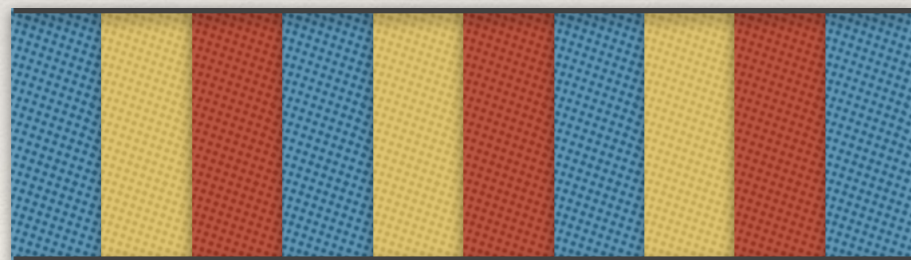
# Multipleksowanie

---

**Multipleksowanie:** rezerwujemy tylko fragment łącza dla jednego strumienia danych.

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

z podziałem czasu  
(2G, 4G, ...)



z podziałem częstotliwości  
(Wi-Fi, DSL, ...)



czas

---

# Implementacja #2: przełączanie pakietów

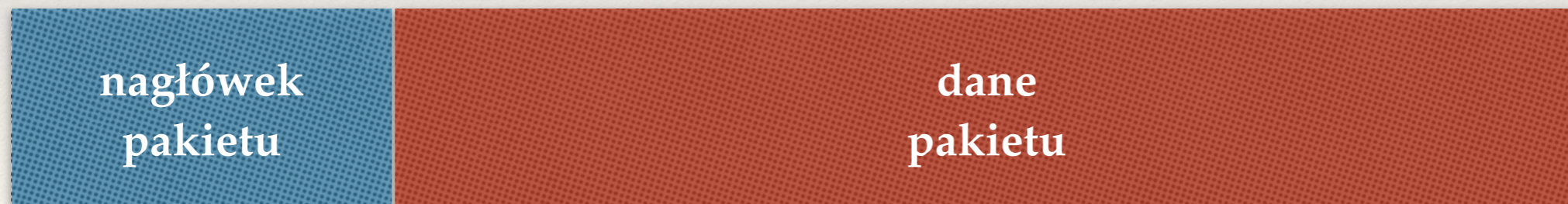
---

# Przełączanie pakietów

---

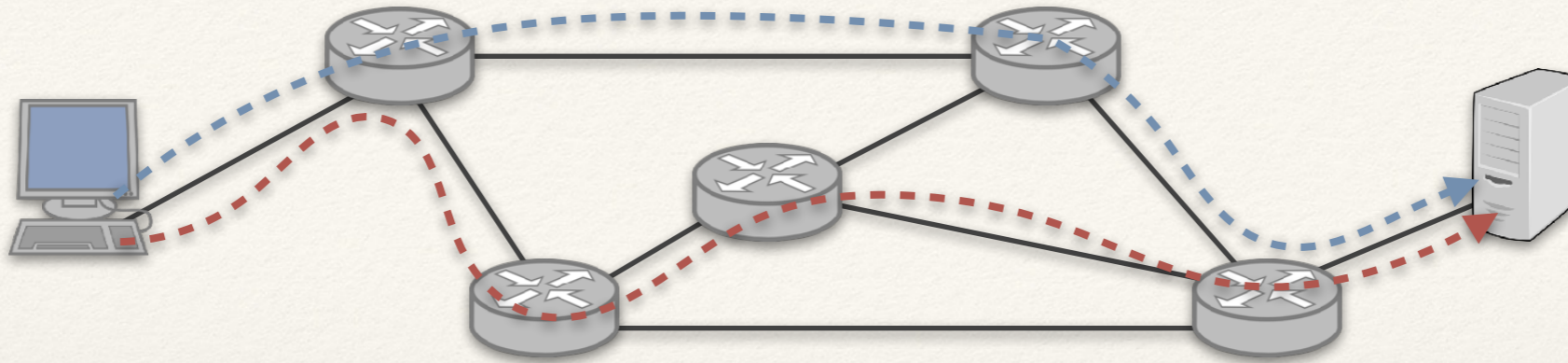
Wysyłany strumień danych dzielimy na małe porcje: **pakiety**.

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



# Routing (1)

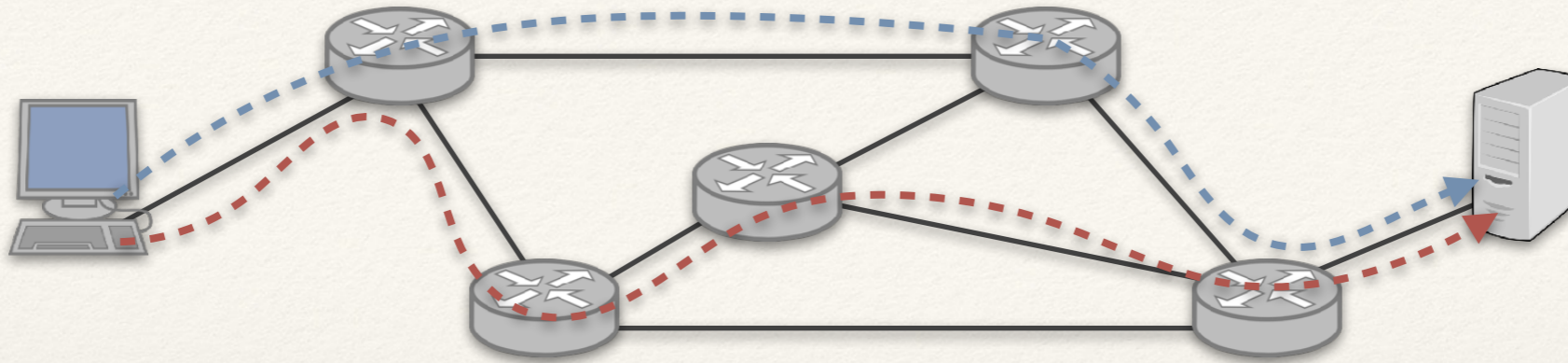
---



- ❖ **Routing (trasowanie)** = wybór trasy dla danego pakietu.
- ❖ Każdy router utrzymuje pewien stan nazywany **tablicą routingu**.
  - ◆ Zbiór reguł „jeśli adres **docelowy** pasuje do wzorca A, to przekaż pakiet do sąsiedniego routera X”.
  - ◆ Pakiet przekazywany tylko o krok dalej.

# Routing (2)

---



- ❖ Router nie musi nic wiedzieć o oryginalnym strumieniu danych.
- ❖ Brak stanu poza tablicą routingu. → Łatwość odtwarzania po awarii.
- ❖ Tablice mogą zmieniać się w czasie. → Pakiety z tego samego strumienia danych mogą docierać do celu różnymi drogami.

# Przykładowa trasa

---

```
traceroute to example.org (104.18.2.24)
 1  _gateway (172.16.16.254)  0.160 ms
 2  info.wask.wroc.pl (156.17.4.254)  1.003 ms
 3  matchem-vprn509-curie-uni.wask.wroc.pl (156.17.252.26)  0.406
 4  uwrvprn509-unir2.wask.wroc.pl (156.17.252.37)  0.610 ms
 5  unir2-uwrvprn509.wask.wroc.pl (156.17.252.36)  0.620 ms
 6  archi3-uni2.wask.wroc.pl (156.17.254.66)  2.411 ms
 7  156.17.254.116 (156.17.254.116)  1.000 ms
 8  156.17.254.119 (156.17.254.119)  0.800 ms
 9  212.191.238.214 (212.191.238.214)  1.105 ms
10  13335-wa1-ix-02.equinix.com (195.182.218.66)  15.714 ms
11  162.158.100.27 (162.158.100.27)  7.508 ms
12  104.18.2.24 (104.18.2.24)  6.678 ms
```

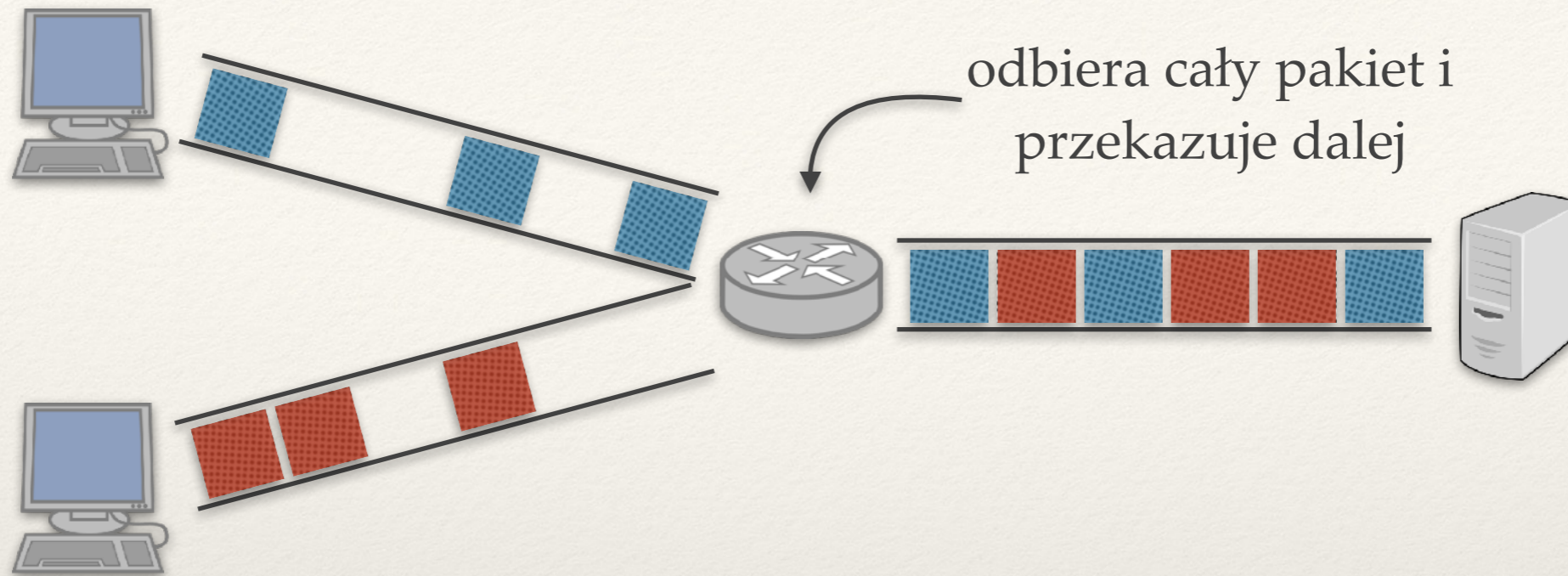
# Przykładowa trasa

---

```
traceroute to example.org (104.18.2.24)
 1  _gateway (172.16.16.254)  0.160 ms
 2  info.wask.wroc.pl (156.17.4.254)  1.003 ms
 3  matchem-vprn509-curie-uni.wask.wroc.pl (156.17.252.26)  0.406
 4  uwrvprn509-unir2.wask.wroc.pl (156.17.252.37)  0.610 ms
 5  unir2-uwrvprn509.wask.wroc.pl (156.17.252.36)  0.620 ms
 6  archi3-uni2.wask.wroc.pl (156.17.254.66)  2.411 ms
 7  156.17.254.116 (156.17.254.116)  1.000 ms
 8  156.17.254.119 (156.17.254.119)  0.800 ms
 9  212.191.238.214 (212.191.238.214)  1.105 ms
10  13335-wa1-ix-02.equinix.com (195.182.218.66)  15.714 ms
11  162.158.100.27 (162.158.100.27)  7.508 ms
12  104.18.2.24 (104.18.2.24)  6.678 ms
```

demonstracja

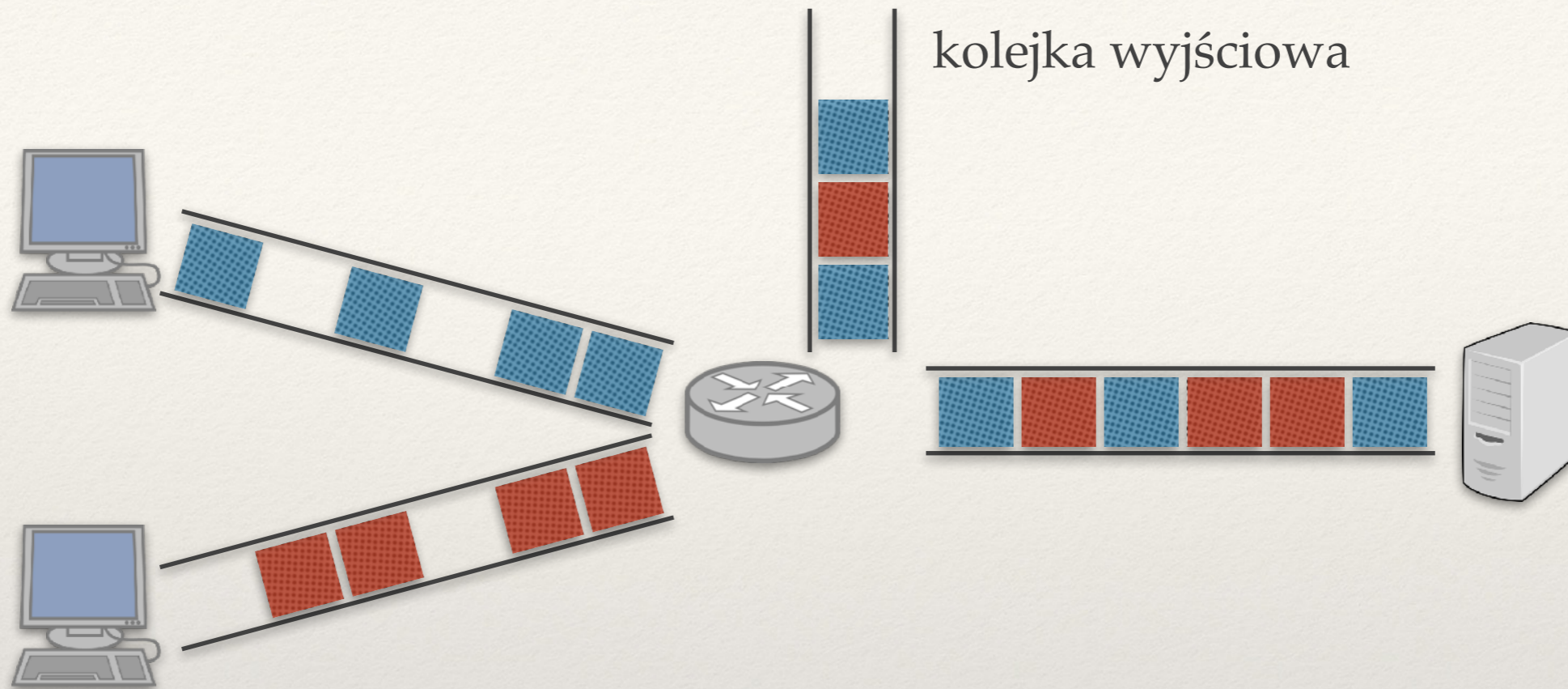
# Multipleksowanie



## Multipleksowanie statystyczne:

- ❖ Jedno łącze wykorzystywane przez pakiety należące do różnych strumieni danych.
- ❖ Lepsze wykorzystanie łącza niż odgórna rezerwacja.
- ❖ **Co jeśli pakiety nadchodzą szybciej niż jesteśmy w stanie wysłać?**

# Kolejki pakietów



- ❖ Pomagają przy przejściowym nadmiarze pakietów.
- ❖ Jeśli kolejka się przepełni (**przeciążenie**) → Pakiety są odrzucane.
  - ◆ Główna przyczyna utraty pakietów w sieci.

# Przełączanie obwodów vs. pakietów

Przełączanie obwodów	Przełączanie pakietów
gwarancja przepustowości	
narzut czasowy na nawiązanie połączenia	narzut czasowy dla każdego pakietu (nagłówek)
marnowanie łącza, jeśli są przerwy w strumieniu danych	efektywne wykorzystanie łącza (statystyczny multipleksing)
	odporne na awarie: wybierana inna ścieżka routingu
	prostsze

---

# Komunikacja oparta na pakietach

---

# 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, tj. „ile możemy wysłać zanim dostaniemy odpowiedź od odbiorcy“.
- ❖ **Duplex**: pełny, półduplex, simplex.

# Opóźnienie pakietu

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



Pierwszy bit wysłany  
po  $10^{-6}$  s.

Ostatni bit wysłany  
po  $800 \times 10^{-6}$  s.



czas

Pierwszy bit dociera  
po  $10^{-6}$  s +  $0,5 \times 10^{-3}$  s.

Ostatni bit dociera po  
 $800 \times 10^{-6}$  s +  $0,5 \times 10^{-3}$  s  
=  $1,3 \times 10^{-3}$  s.

czas

**Opóźnienie pakietu na łączu** = czas oczekiwania pakietu w kolejce  
+ (rozmiar pakietu / przepustowość) + czas propagacji.

# Protokół IP (*Internet Protocol*)

---

- ❖ Zaimplementowany w każdym urządzeniu.
- ❖ Określa adresy urządzeń (4-bajtowe identyfikatory).
- ❖ **Zawodna, bezpołączeniowa** usługa umożliwiająca przesłanie pakietu między dwoma dowolnymi urządzeniami w sieci.

# Protokół IP (*Internet Protocol*)

---

- ❖ Zaimplementowany w każdym urządzeniu.
- ❖ Określa adresy urządzeń (4-bajtowe identyfikatory).
- ❖ **Zawodna, bezpołączeniowa** usługa umożliwiająca 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

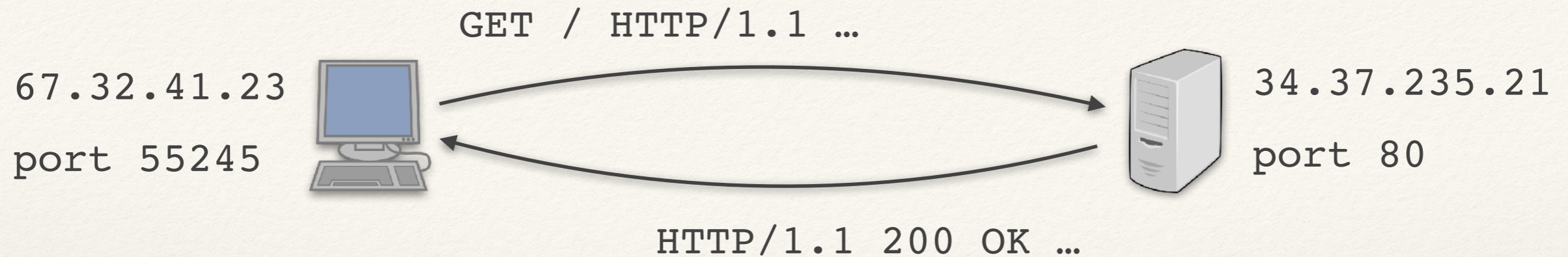
---



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

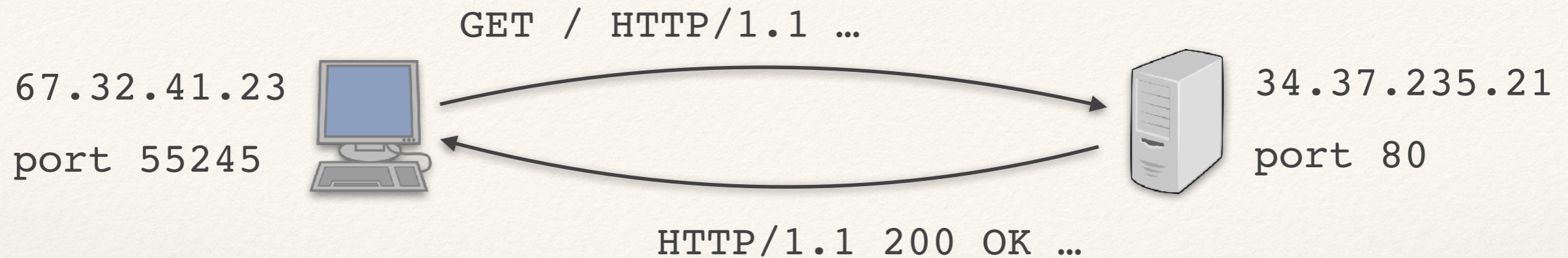
# HTTP a pakiety IP

---



- ❖ Mamy protokół IP (umiemy wysyłać pakiety). Co potrzebne jest do konstrukcji HTTP?
  - ♦ Niezawodność przesyłania danych.
  - ♦ Dzielenie na pakiety.
  - ♦ Rozróżnianie różnych przeglądarek na tym samym komputerze → identyfikatory aplikacji (**porty**).

# HTTP a pakiety IP



- ❖ Mamy protokół IP (umiemy wysyłać pakiety). Co potrzebne jest do konstrukcji HTTP?
  - ♦ Niezawodność przesyłania danych.
  - ♦ Dzielenie na pakiety.
  - ♦ Rozróżnianie różnych przeglądarek na tym samym komputerze → identyfikatory aplikacji (**porty**).
- ❖ Typowe problemy → warto wydzielić osobny komponent.

---

# Model warstwowy

---

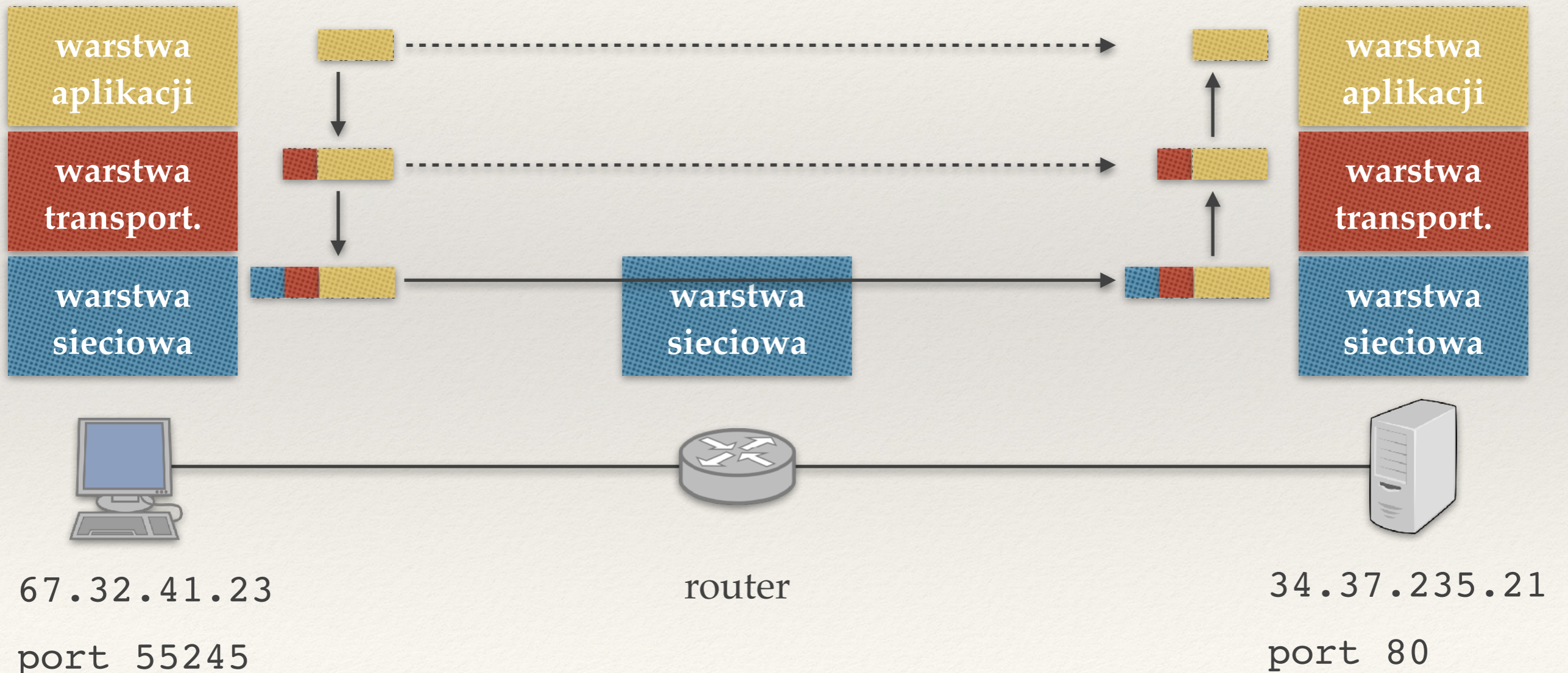
# Warstwa aplikacji, transportowa i sieciowa

od: 67.32.41.23  
do: 34.37.235.21

od: port 55245  
do: port 80

GET / HTTP/1.1

przesyłany pakiet



# Warstwa aplikacji, transportowa i sieciowa

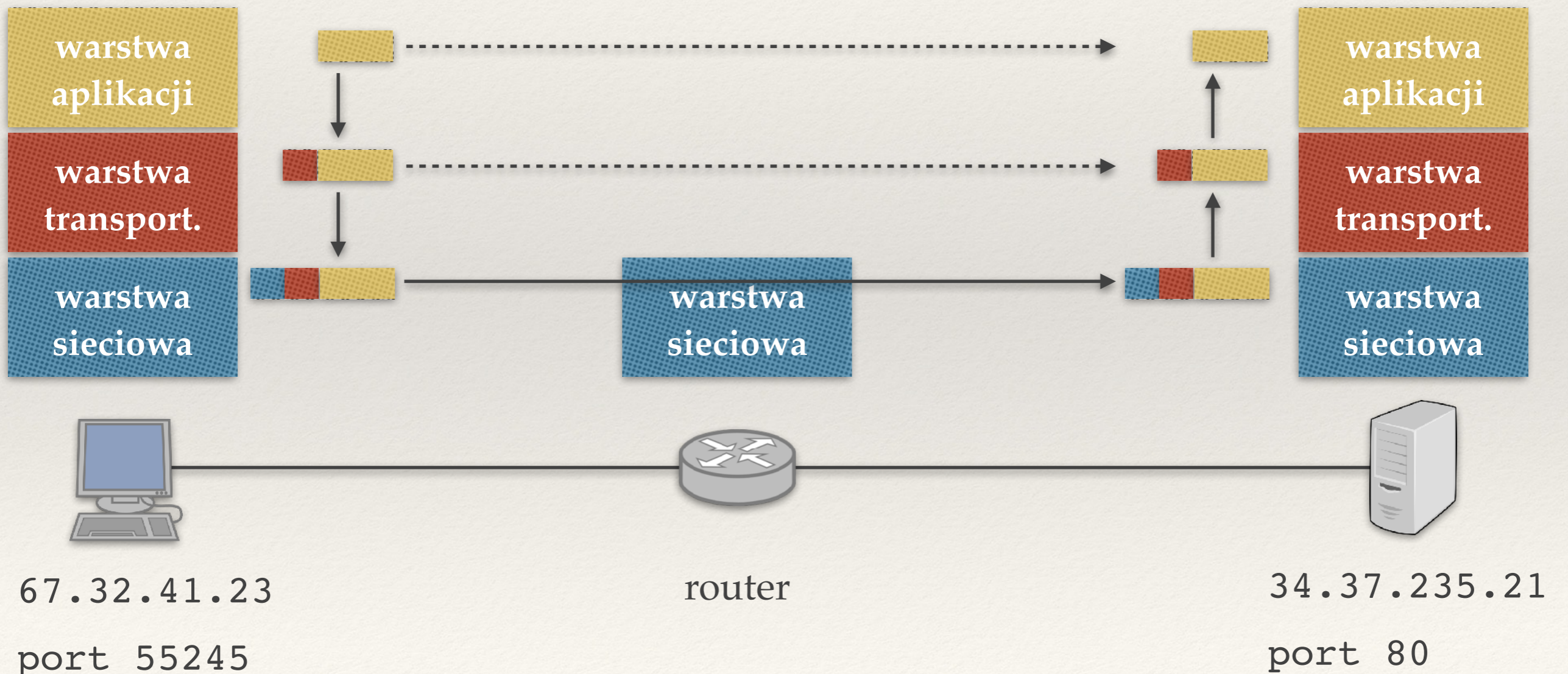
od: 67.32.41.23  
do: 34.37.235.21

od: port 55245  
do: port 80

GET / HTTP/1.1

demonstracja

przesyłany pakiet



# Model warstwowy

---

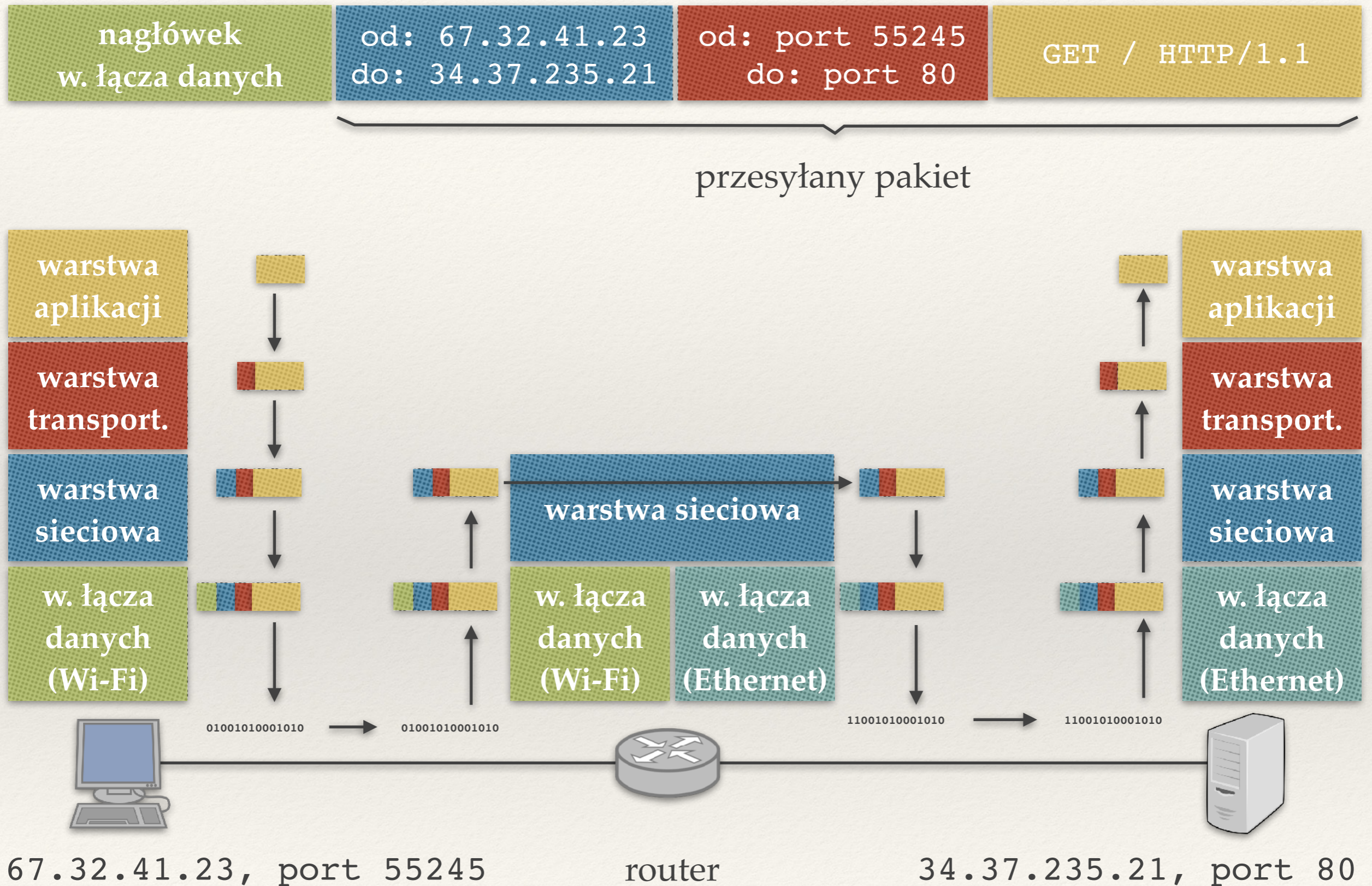
- ❖ **Warstwa definiuje:**
  - ◆ swoje zadania,
  - ◆ interfejs do sąsiednich warstw.
- ❖ **Modularność:** możliwość wymiany implementacji warstwy.
- ❖ **Enkapsulacja/dekapsulacja:** proces dodawania / usuwania nagłówka przy przechodzeniu przez kolejną warstwę.

# Lokalne dostarczanie

---

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

# Internetowy model warstwowy (1)



# Internetowy model warstwowy (2)

---



**Globalne dostarczanie danych pomiędzy aplikacjami.**

**Globalne dostarczanie danych pomiędzy komputerami.**

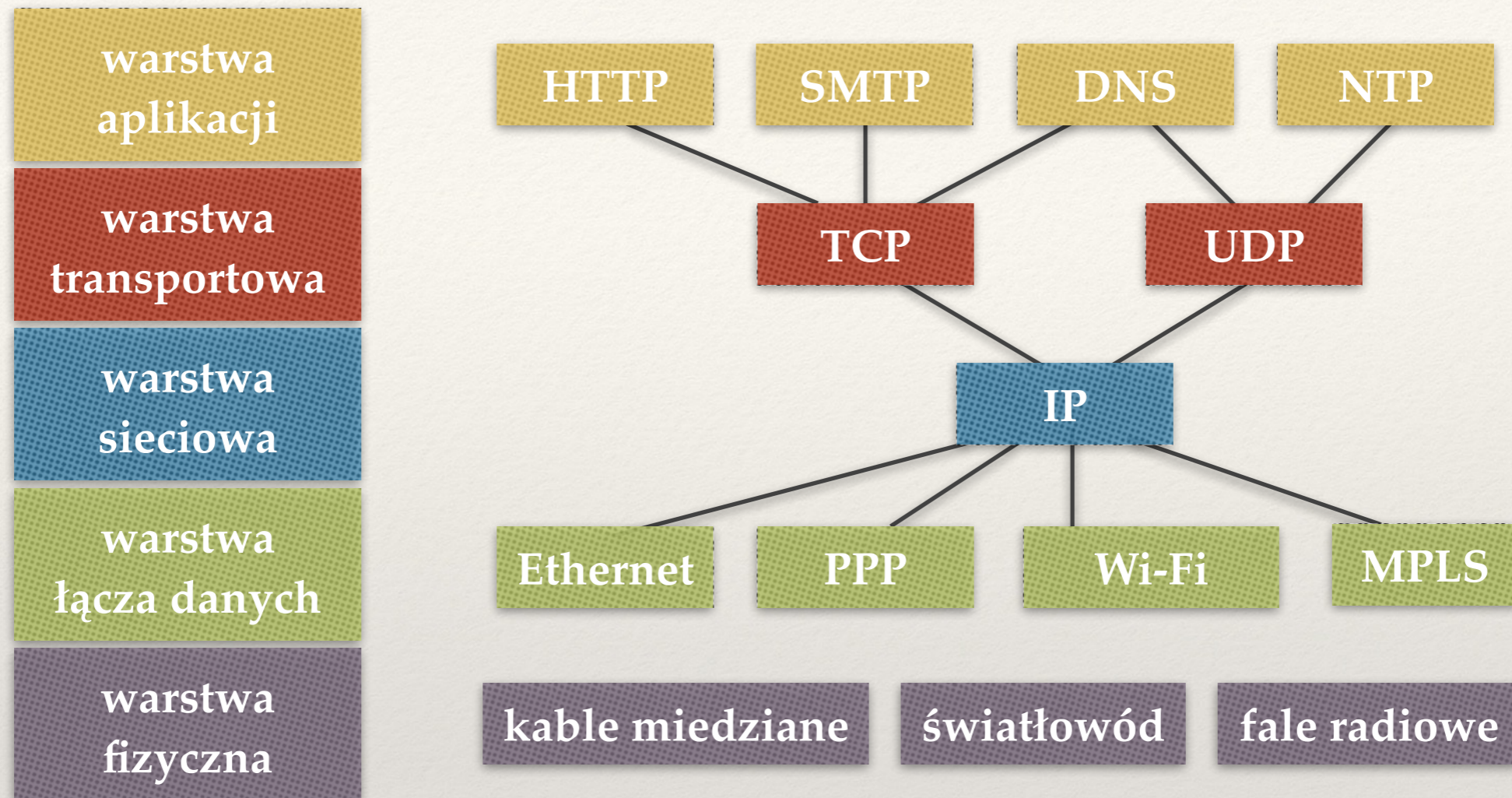
**Lokalne dostarczanie danych pomiędzy komputerami.**

# 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	Wybiera trasy dla pakietów.
2 łącza danych	Ethernet, Wi-Fi, ...	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.
- ❖ Umożliwi(ł) 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)

---

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

# Egzamin

---

- ❖ **Test wielokrotnego wyboru.**
  - ♦ Materiał z wykładu, ćwiczeń i warsztatów.
  - ♦ Zagadnienia i pojęcia.
- ❖ 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 egzaminu: od 60%.**

# Wymagania wstępne

---

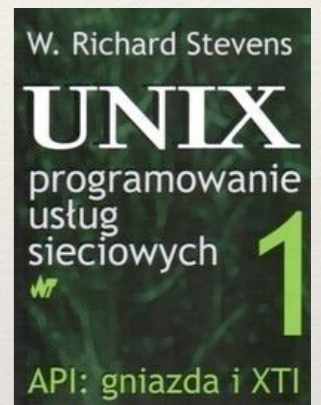
- ❖ Obsługa Linuksa (wiersz poleceń, proste skrypty, automatyzacja).
  - ✦ Linux command line: <http://linuxcommand.org/tlcl.php>
- ❖ Systemy operacyjne (procesy, sygnały, IPC).
- ❖ Programowanie w C lub C++ lub Rust.
- ❖ Algebra lub Matematyka dyskretna (wielomiany, pierścienie, operacje na grupach).

Uwaga: ten przedmiot nie jest przewidziany dla studentów 1. roku.

# Literatura

---

- ❖ J. Kurose, K. Ross: Sieci komputerowe, ujęcie całościowe Helion, 2018, wydanie VII (lub V lub VI).
- ❖ W. R. Stevens: UNIX programowanie usług sieciowych, tom 1, WNT, 2002.
- ❖ Andrew S. Tanenbaum, Sieci komputerowe Helion, 2012, wydanie V.
- ❖ Internet.
- ❖ <https://www.rfc-editor.org/>



# Zagadnienia

---

- ❖ Co jest potrzebne do zbudowania dwukierunkowego niezawodnego kanału?
- ❖ Co to jest protokół komunikacyjny?
- ❖ Porównaj wady i zalety przełączania obwodów i przełączania pakietów.
- ❖ Jakie znasz rodzaje multipleksowania? Po co i kiedy się je stosuje?
- ❖ Do czego służy polecenie `traceroute`? Co pokazuje?
- ❖ Po co stosuje się bufony w routerach? Co to jest przeciążenie?
- ❖ Co to jest BDP? Co to jest czas propagacji?
- ❖ Co oznaczają pojęcia pełny duplex, półduplex, simplex?
- ❖ Co wpływa na opóźnienie pakietu?
- ❖ Co umożliwia protokół IP? Co to znaczy, że protokół realizuje zasadę dołożenia wszelkich starań?
- ❖ Jakie są zalety i wady zasady end-to-end?
- ❖ Po co wprowadza się porty?
- ❖ Wyjaśnij pojęcie enkapsulacji i dekapulacji.
- ❖ Dlaczego wprowadza się warstwy protokołów?
- ❖ Wymień warstwy internetowego modelu warstwowego. Jakie są zadania każdej z nich?
- ❖ Jakie warstwy są zaimplementowane na komputerach a jakie na routerach?