Warstwa aplikacji

część 2

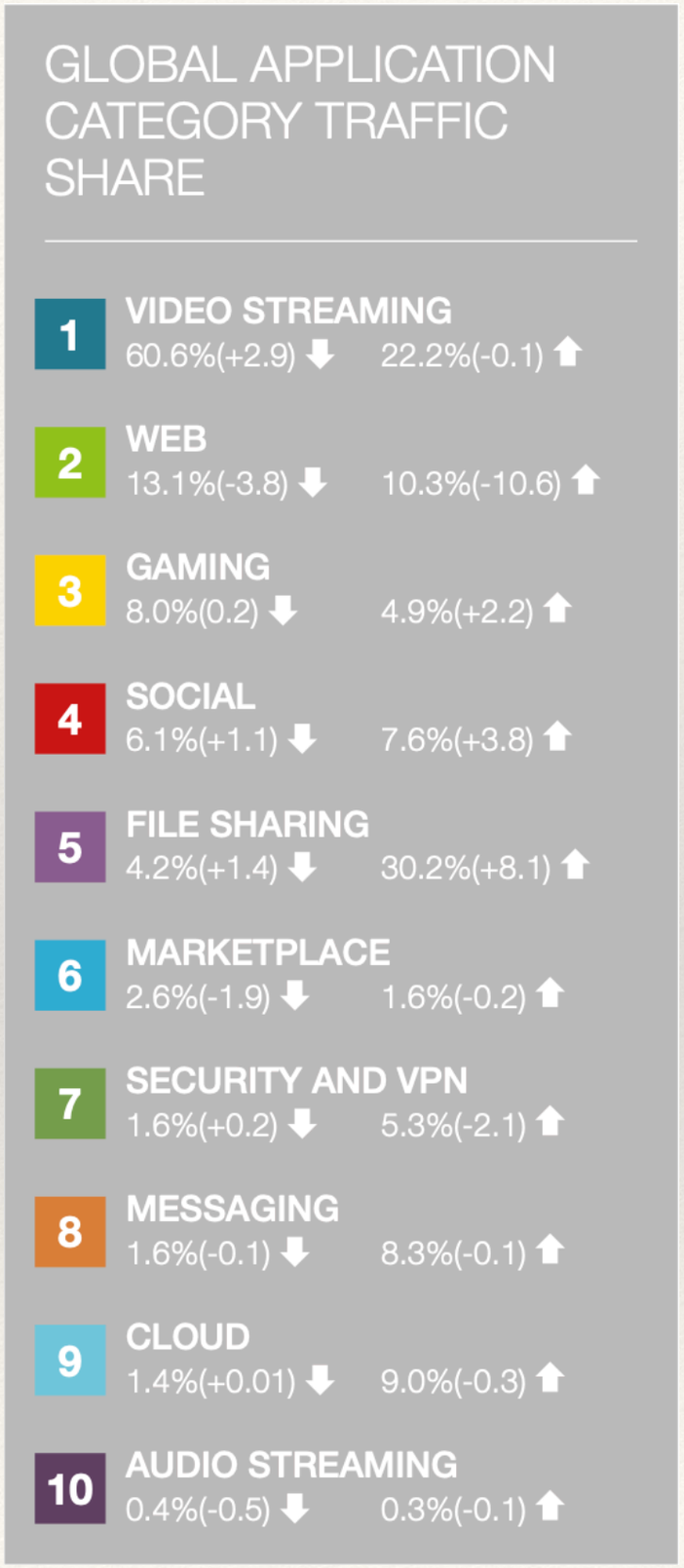
Sieci komputerowe

Wykład 10

Marcin Bieńkowski

W dzisiejszym odcinku

- Sieci peer-to-peer.
- NAT (*network address translation*) a warstwa aplikacji.
- Wydajność HTTP.
- Poczta elektroniczna.

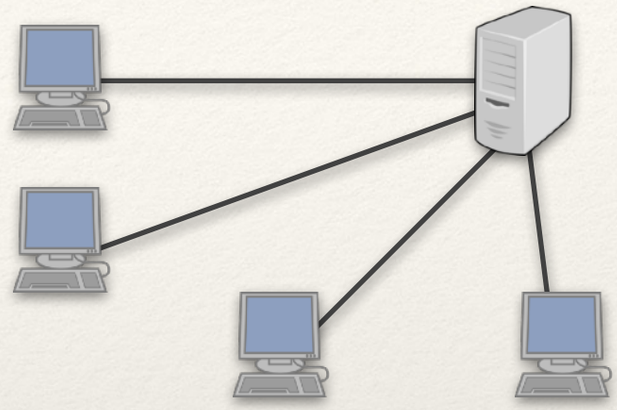


Obrazek ze strony https://www.sandvine.com/global-internet-phenomena-report-2019

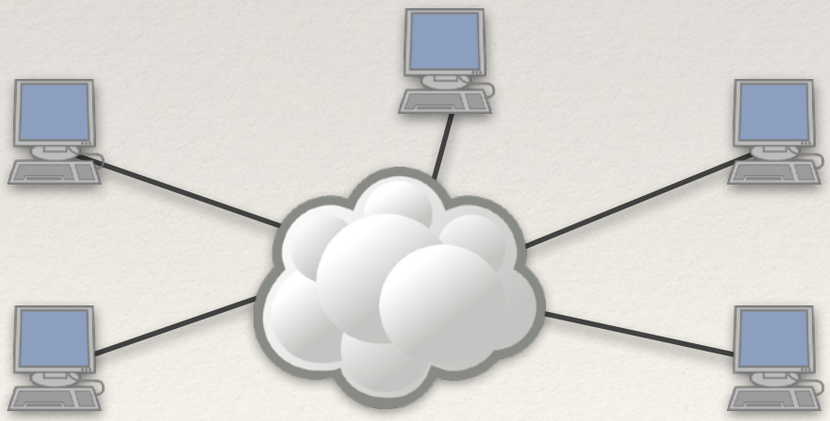
Peer-to-peer

Klient-serwer a peer-to-peer

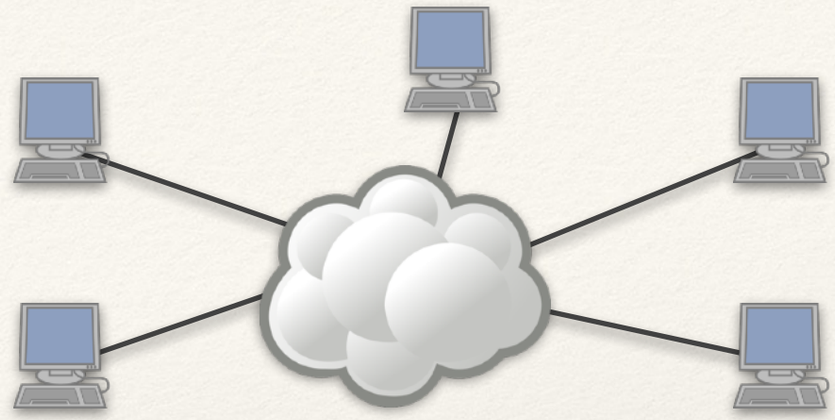
- Do tej pory mówiliśmy o **architekturze klient-serwer** (HTTP, DNS, poczta elektroniczna,…)



- Duża część rozwiązań pracuje w **architekturze peer-to-peer** (BitTorrent, Skype, …)



Peer-to-peer

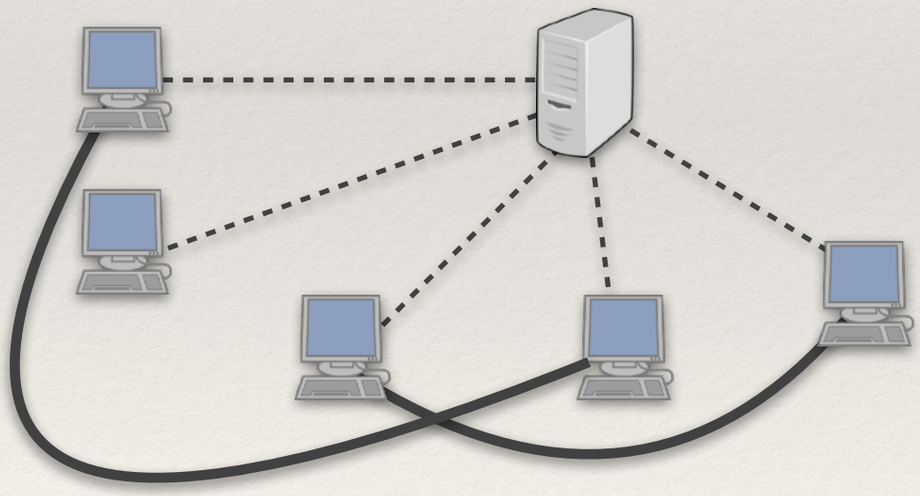


Architektura P2P:

- Wszystkie komputery są jednocześnie klientami i serwerami.
- Każdy komputer może nawiązywać połączenia z innymi.
- Brak centralnego miejsca z danymi:
  - Lepsza skalowalność i niezawodność
  - Autonomia ale trudniejsze zagwarantowanie współpracy całości.

„Niepełne” peer-to-peer

- W większości architektur peer-to-peer istnieją wyróżnione komputery:
  - Przechowują np. bazę użytkowników (Skype)
  - Pomagają w podłączeniu (punkt pierwszego kontaktu + później)



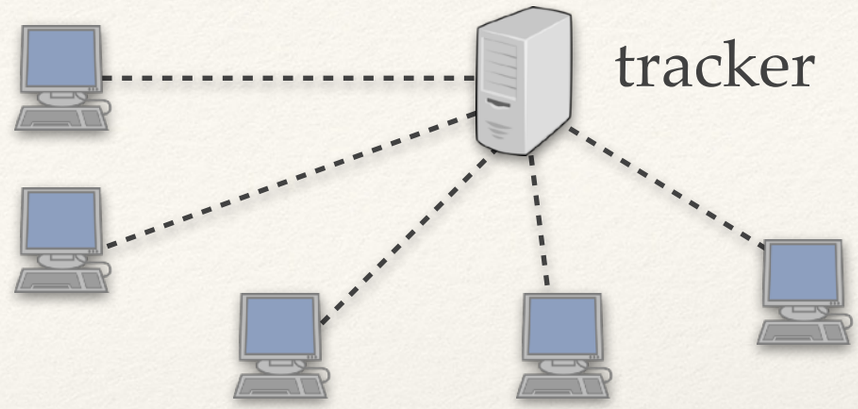
komunikaty kontrolne

właściwe dane

Warstwa aplikacji a warstwa transportowa

- Peer-to-peer = określenie logiki warstwy aplikacji.
- Dwa punkty sieci peer-to-peer chcą wymieniać dane.
  - Wykorzystują w tym celu warstwę transportową (TCP lub UDP).
  - Z punktu widzenia TCP lub UDP jeden z nich (serwer) oczekuje na połączenie a drugi (klient) łączy się z nim.

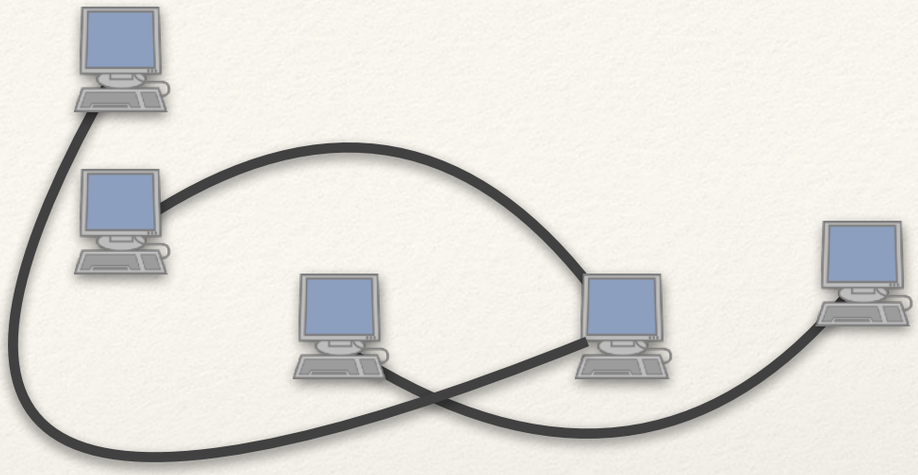
Przykład: BitTorrent



Podłączanie się do sieci

- Łączymy się z trackerem.
- Tracker zna adresy członków sieci i udostępnia adresy niektórych (50-100).
- Po jakimś czasie możemy prosić o kolejne adresy.

BitTorrent: przesyłanie pliku (1)

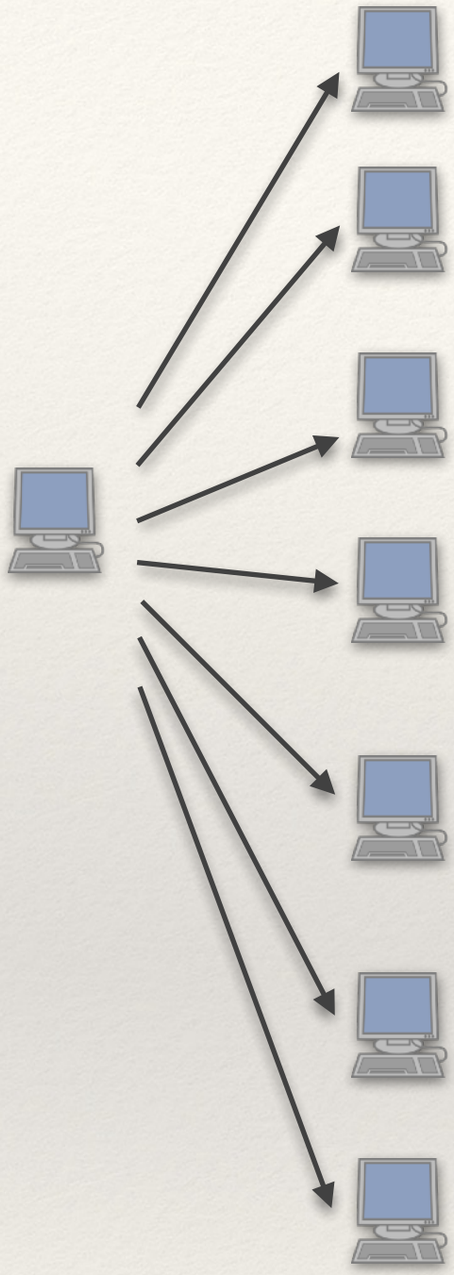


Plik dzielony na kawałki

- Każdy kawałek pobierany niezależnie
- Rozmiar kawałka ok. 256 KB - 16 MB
  - Duży → żeby okno TCP miało czas urosnąć.
  - Mały → żeby plik miał wiele kawałków (urównoleglenie).

- *Seeder* = ma wszystkie kawałki.
- *Leecher* = ma niektóre kawałki.

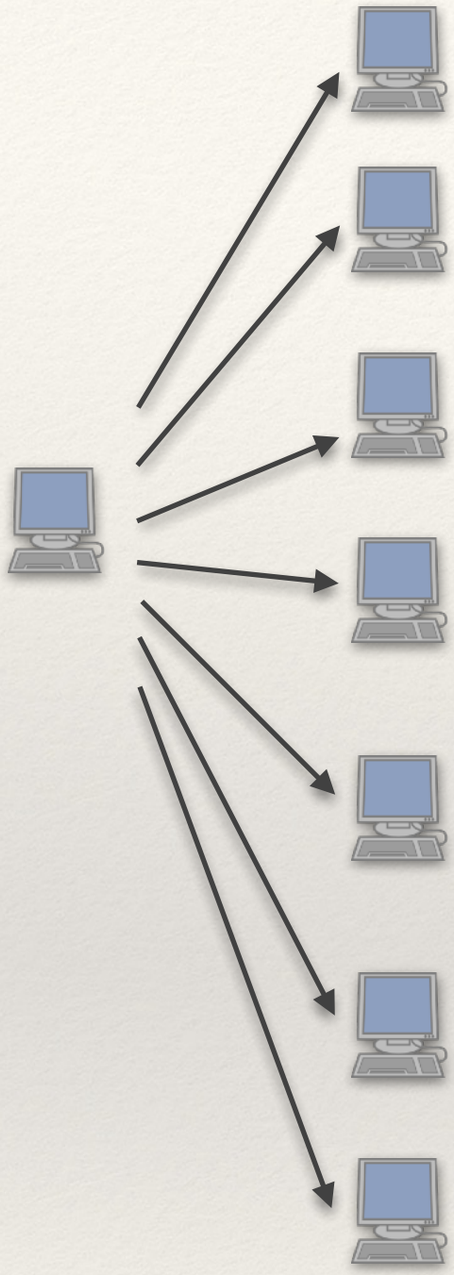
Jak przesłać jeden kawałek, żeby wszyscy go mieli? (1)



1. Minimalne opóźnienie, maksymalne obciążenie pojedynczego członka sieci

(jak w modelu klient-server)

Jak przesłać jeden kawałek, żeby wszyscy go mieli? (1)



1. Minimalne opóźnienie, maksymalne obciążenie pojedynczego członka sieci

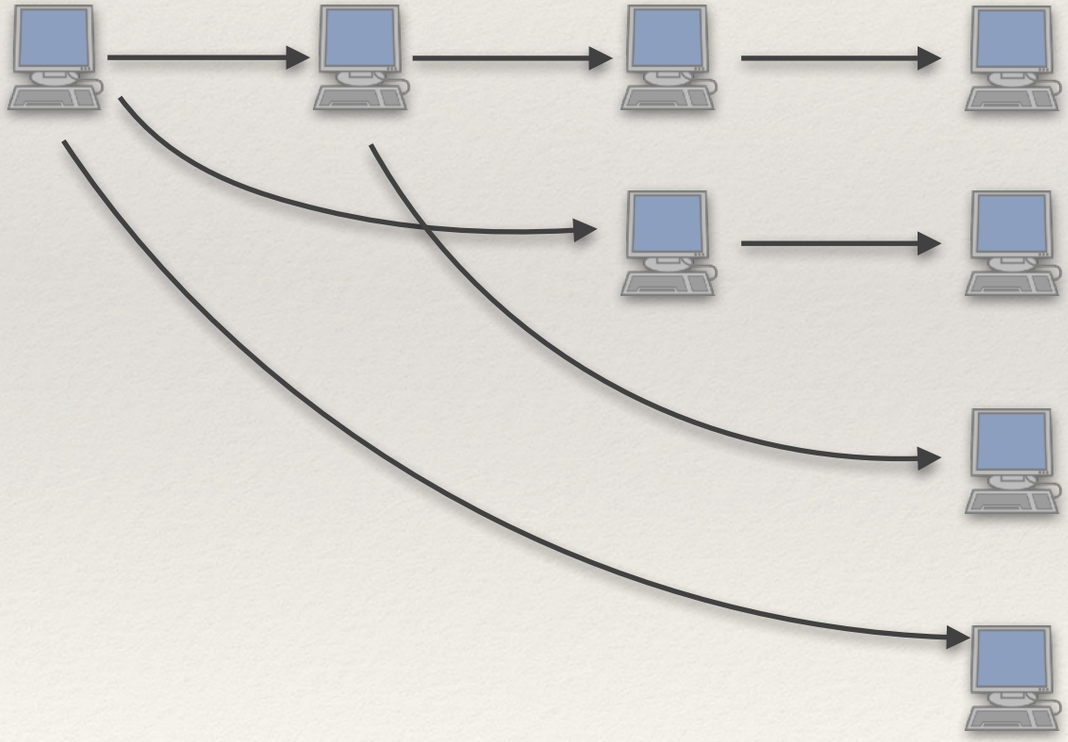
(jak w modelu klient-server)

2. Minimalne obciążenie pojedynczego członka sieci, duże opóźnienie

Jak przesłać jeden kawałek, żeby wszyscy go mieli? (2)

3. Rozwiązanie pośrednie:

- logarytmiczna głębokość
- logarytmiczne obciążenie pojedynczych wierzchołków
- duża odporność na opuszczanie sieci przez komputery



BitTorrent: przesyłanie pliku (2)

- **Typowo**: każdy klient ma mnóstwo chętnych na posiadane kawałki.
  - Klienci zazwyczaj chcą kawałki, które występują najrzadziej w sieci.
- ***Seeder*** wybiera kandydatów wśród chętnych po kolei (co najwyżej Q naraz).
- ***Leecher*** ma listę P klientów, którym udostępnia.
  - Tylko pod warunkiem, że *dostał od nich coś w zamian*.
  - Na liście jest P takich klientów, którzy najszybciej wysyłają mu swoje kawałki.
  - Eksploracja: czasem dajemy kawałek losowemu członkowi sieci
    - może odeśle nam jakiś kawałek odpowiednio szybko?
  - Nowi klienci: jeśli klient mówi, że jest nowy, to dostaje kawałek za darmo.

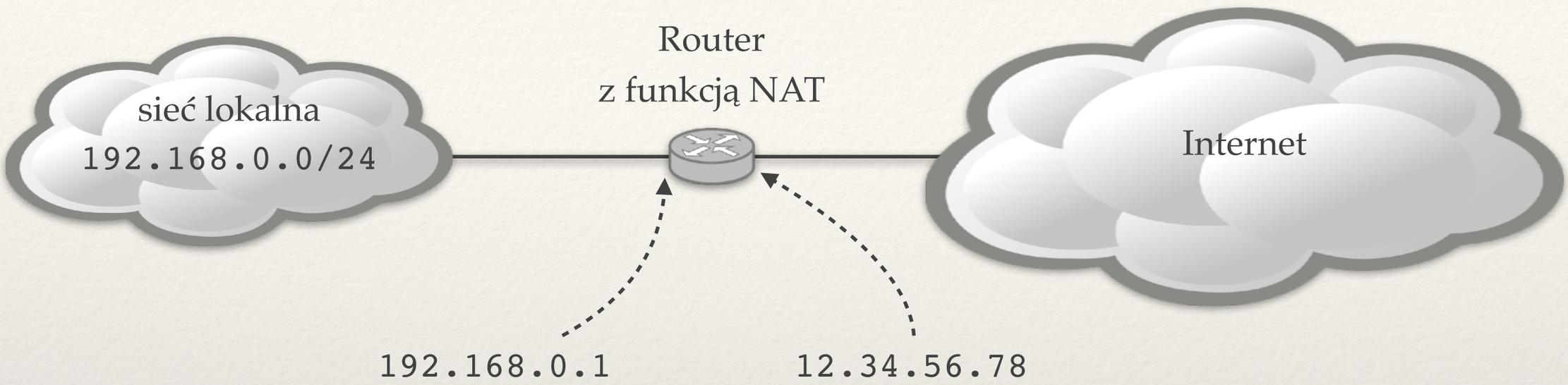
animacja

BitTorrent: metadane

- Z plikiem X związany jest plik `.torrent`, umieszczany na WWW.
- Zawiera IP trackera.
- Zawiera funkcje skrótu dla wszystkich kawałków → umożliwia sprawdzenie, czy pobraliśmy dobry kawałek.

NAT vs. warstwa aplikacji

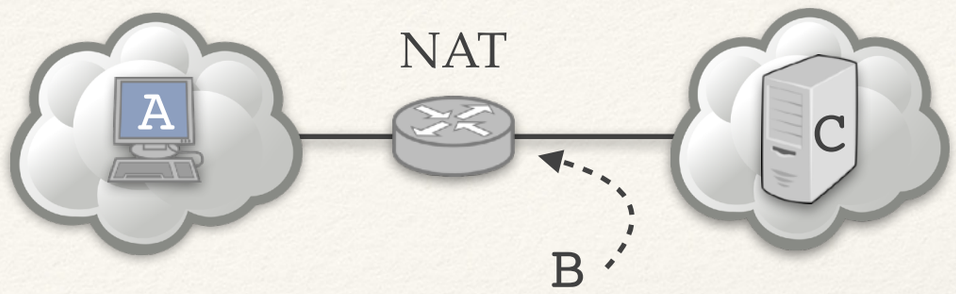
NAT



- Bardzo powszechne rozwiązanie.
- Z reszty Internetu cała sieć lokalna wygląda tak samo, jak pojedynczy komputer z adresem `12.34.56.78`.

Co robi router z funkcją NAT?

- **Pakiet**
  - Z adresu i portu (A, P_A).
  - Do adresu i portu (C, P_C).
  - NAT na podstawie krotki (A, P_A, C, P_C) wybiera port P_B.
  - Adres i port źródłowy pakietu podmienione na (B, P_B).

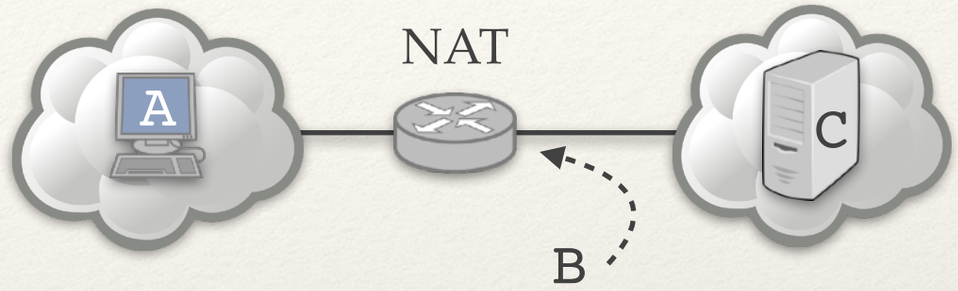


- **Tablica NAT:**
  - Przechowuje przez pewien czas przypisanie (A, P_A, C, P_C) → P_B.
  - **Dla kolejnych podobnych pakietów przypisanie będzie takie samo.**
  - Jeśli przychodzi pakiet **z Internetu** do (B, P_B) to jego adres i port docelowy zostanie podmieniony na (A, P_A).

NAT a P2P

Kiedyś:

- Komunikacja zawsze w modelu klient-serwer.
- Serwery nie są za routerami z NAT.
- Klienci mogą być za routerami z NAT

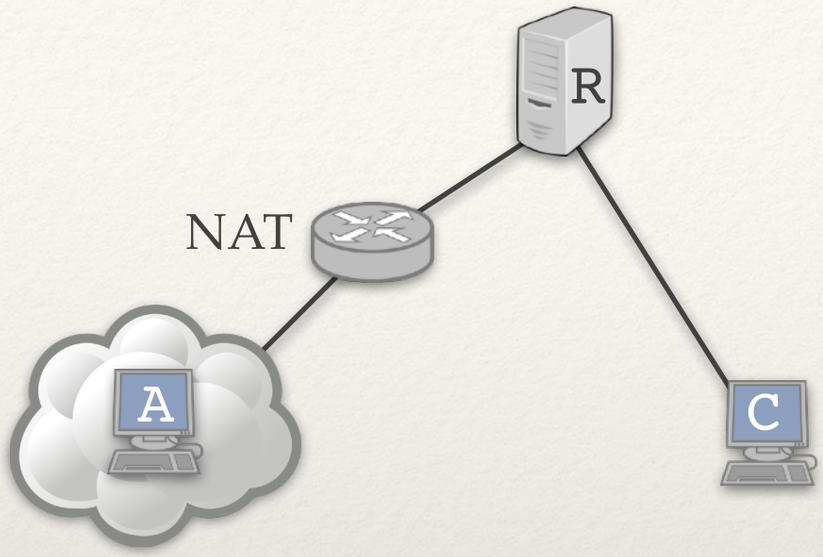


- Początkowa transmisja (np. TCP SYN) od klienta do serwera tworzy przypisanie $(A, P_A, C, P_C) \rightarrow P_B$, dzięki któremu pakiety z odpowiedziami serwera mogą wracać do klienta.

Obecnie:

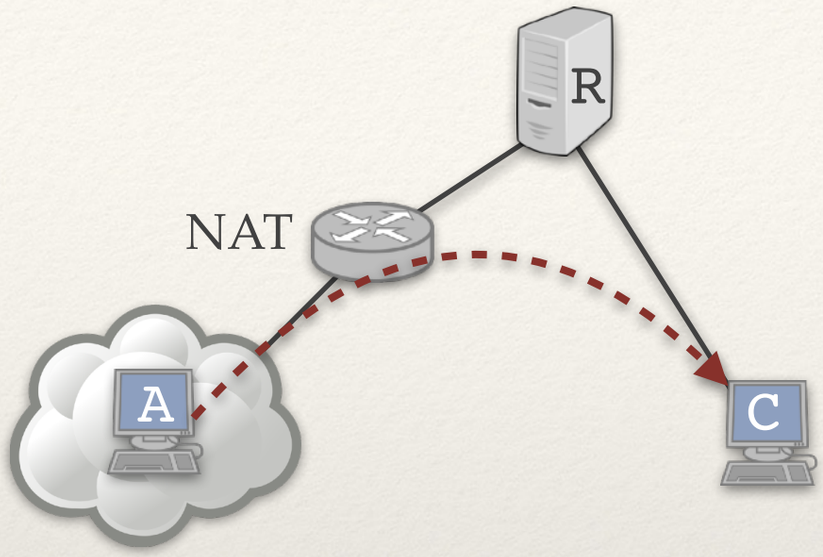
- Chcemy często przesyłać dane w modelu peer-to-peer (Bittorrent, Skype, …)
- Obie strony są często za NAT!
- Brak naturalnej możliwości zainicjowania połączenia.

Odwrócone połączenie



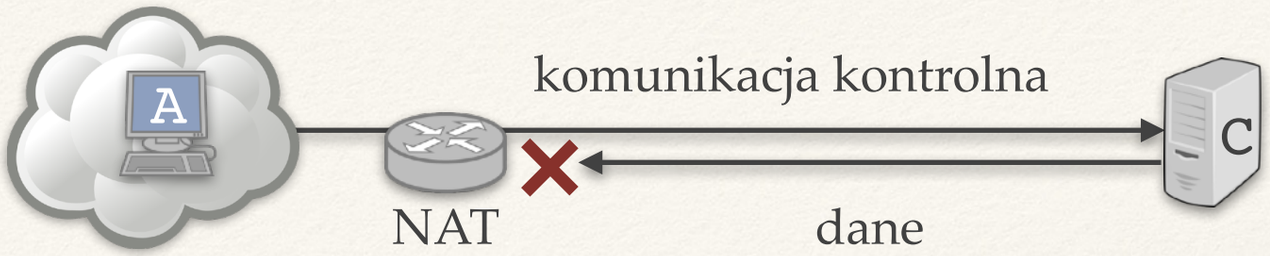
- C chce nawiązać połączenie z A, ale A jest za NAT.
- Jeśli oba utrzymują kontakt z R, to C może poprosić (przez R) komputer A o nawiązanie bezpośredniego połączenia z C.
- Stosowane np. w Skype.

Odwrócone połączenie



- C chce nawiązać połączenie z A, ale A jest za NAT.
- Jeśli oba utrzymują kontakt z R, to C może poprosić (przez R) komputer A o nawiązanie bezpośredniego połączenia z C.
- Stosowane np. w Skype.

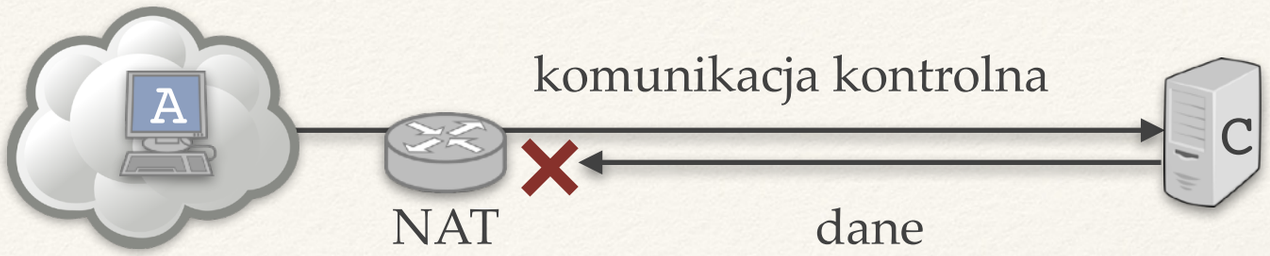
Odwrócone połączenie w protokole FTP



FTP: protokół przesyłania plików.

- Na początku klient A łączy się z serwerem C na porcie 21 (połączenie na komunikaty kontrolne).
- A wysyła polecenie „chcę pobrać plik i słucham na porcie X“
  - C łączy się z portem X klienta A i wysyła plik (odrębne połączenie TCP).
  - Połączenie odrzucane przez NAT.

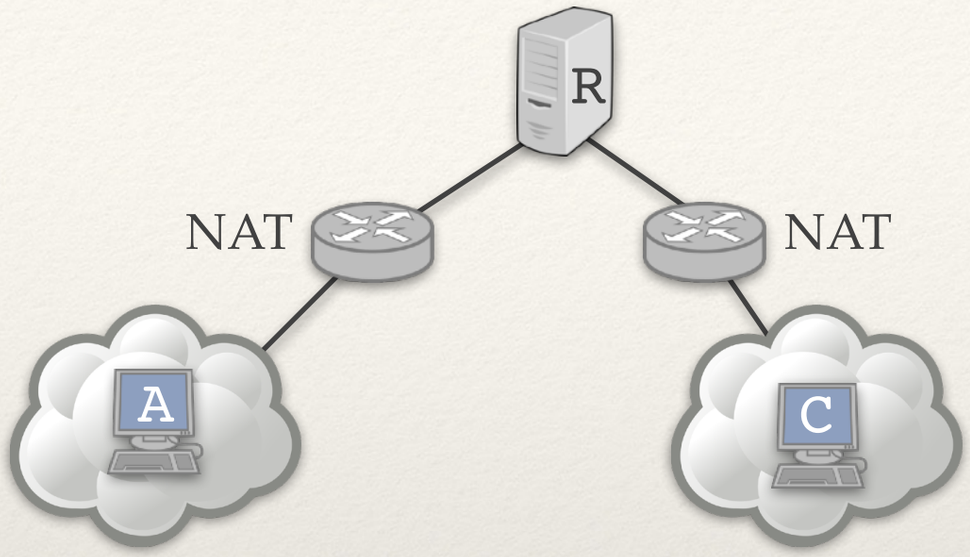
Odwrócone połączenie w protokole FTP



FTP: protokół przesyłania plików.

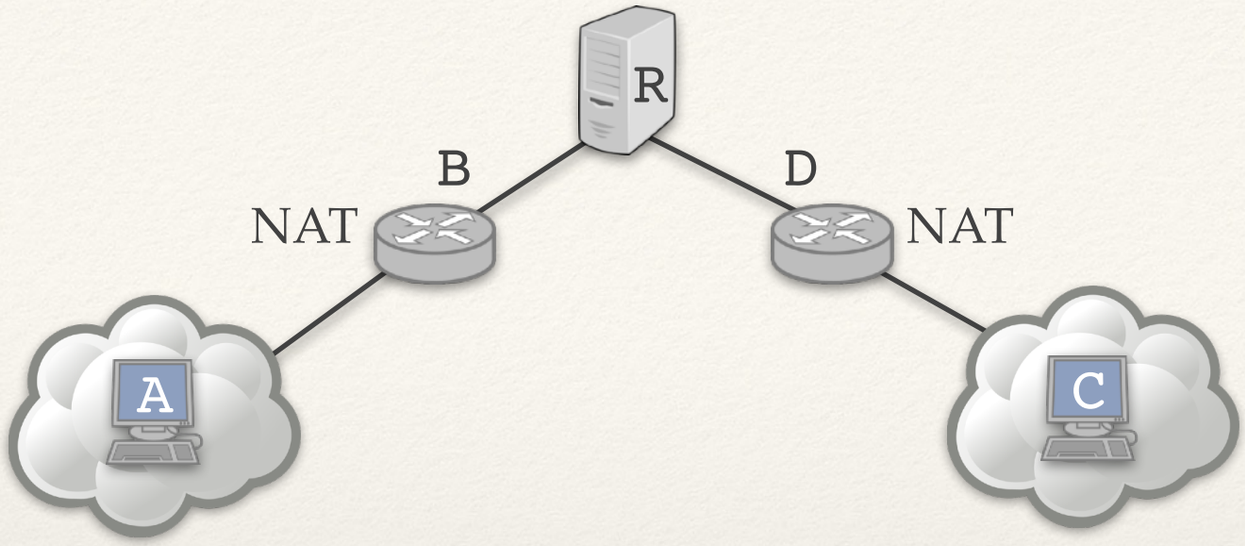
- Na początku klient A łączy się z serwerem C na porcie 21 (połączenie na komunikaty kontrolne).
- A wysyła polecenie „chcę pobrać plik i słucham na porcie X“
  - C łączy się z portem X klienta A i wysyła plik (odrębne połączenie TCP).
  - Połączenie odrzucane przez NAT.
- Tryb pasywny FTP: A wysyła polecenie „chce pobrać plik w trybie pasywnym“
  - C zaczyna słuchać na porcie Y i wysyła komunikat „słucham na porcie Y“.
  - A łączy się z portem Y serwera C i pobiera plik.

Przekaźniki

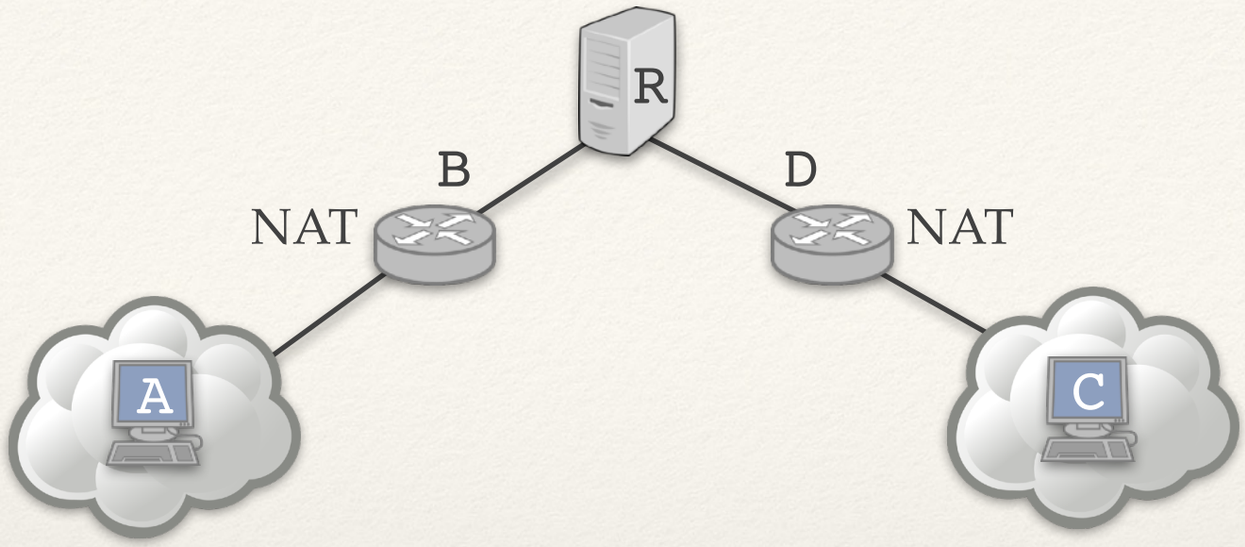


- Jeśli A i C utrzymują kontakt z R, to oba mogą nawiązać połączenie z R i R może przekazywać między nimi dane.
- Stosowane np. w Skype (jeśli wszystko inne zawiedzie).

Przechodzenie przez NAT (1)

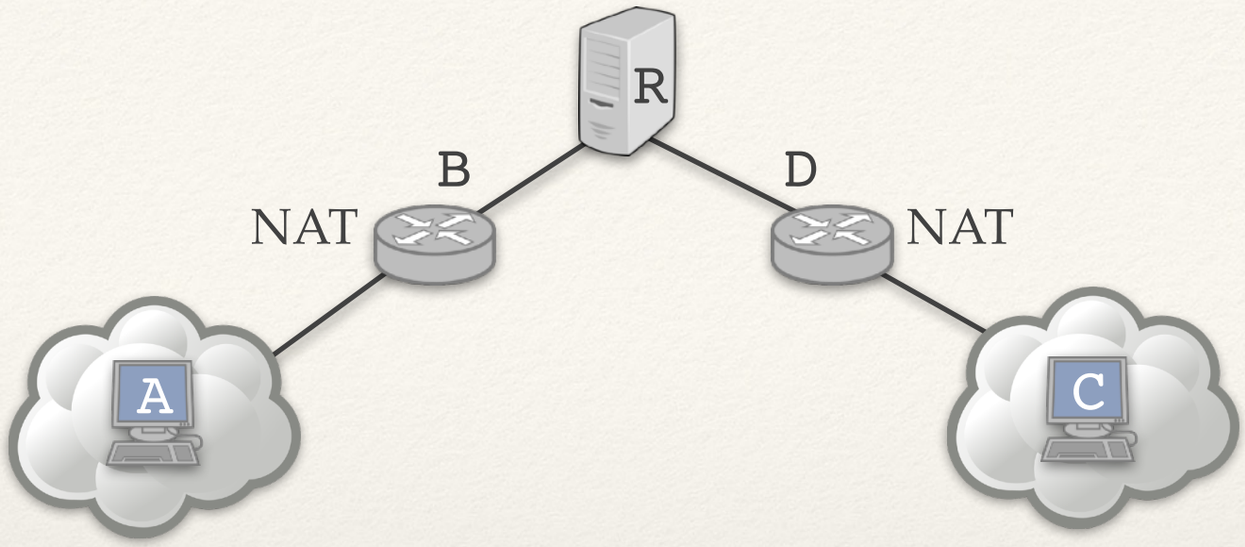


Przechodzenie przez NAT (1)



- A wysyła z portu P_A pakiet do R o treści „**(A,P_A)**“.
- Na routerze NAT zostaje utworzone przypisanie **(A,P_A) → (B,P_B)**.
- R widzi pakiet o treści „**(A,P_A)**“ od **(B, P_B)**, tj. poznaje przypisanie wygenerowane przez B.

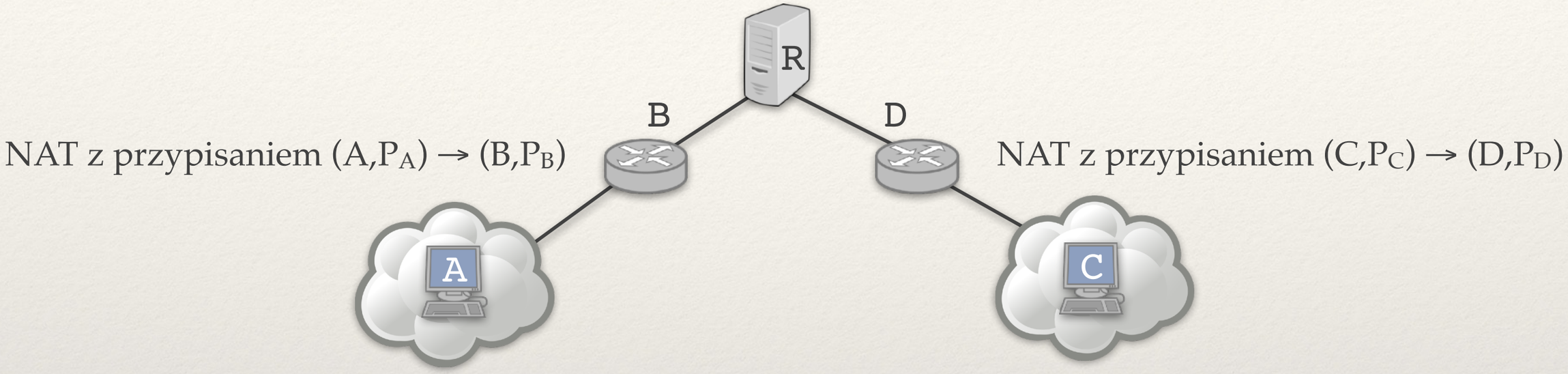
Przechodzenie przez NAT (1)



- A wysyła z portu P_A pakiet do R o treści „**(A,P_A)**“.
- Na routerze NAT zostaje utworzone przypisanie **(A,P_A) → (B,P_B)**.
- R widzi pakiet o treści „**(A,P_A)**“ od **(B, P_B)**, tj. poznaje przypisanie wygenerowane przez B.
- W taki sam sposób R poznaje przypisanie (C,P_C) → (D,P_D).
- R odsyła poznane przypisania do A i C.

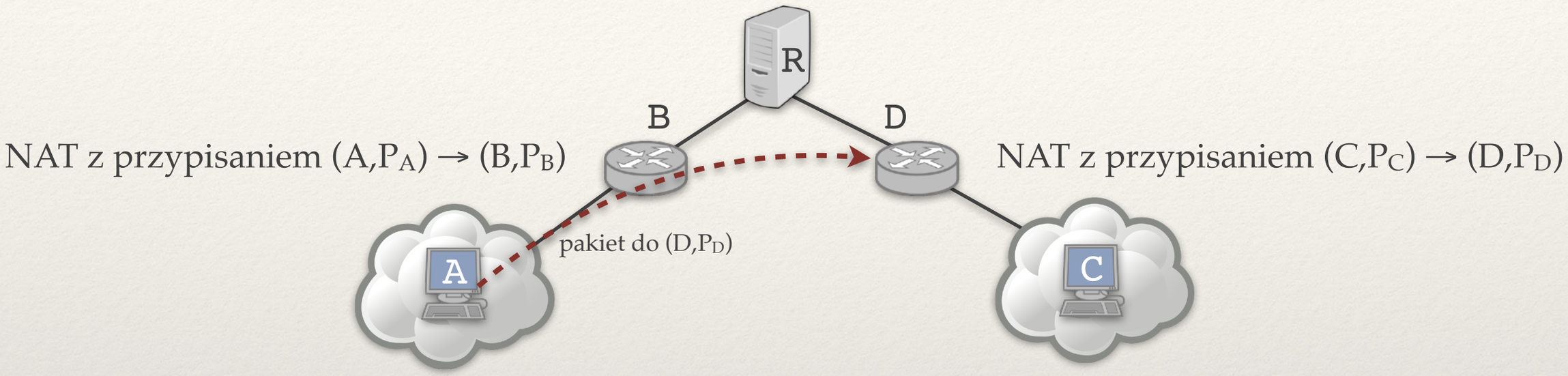
Przechodzenie przez NAT (2)

Fakt: Jeśli R wyśle dane do (D,P_D) to zostaną przesłane do (C,P_C).



Przechodzenie przez NAT (2)

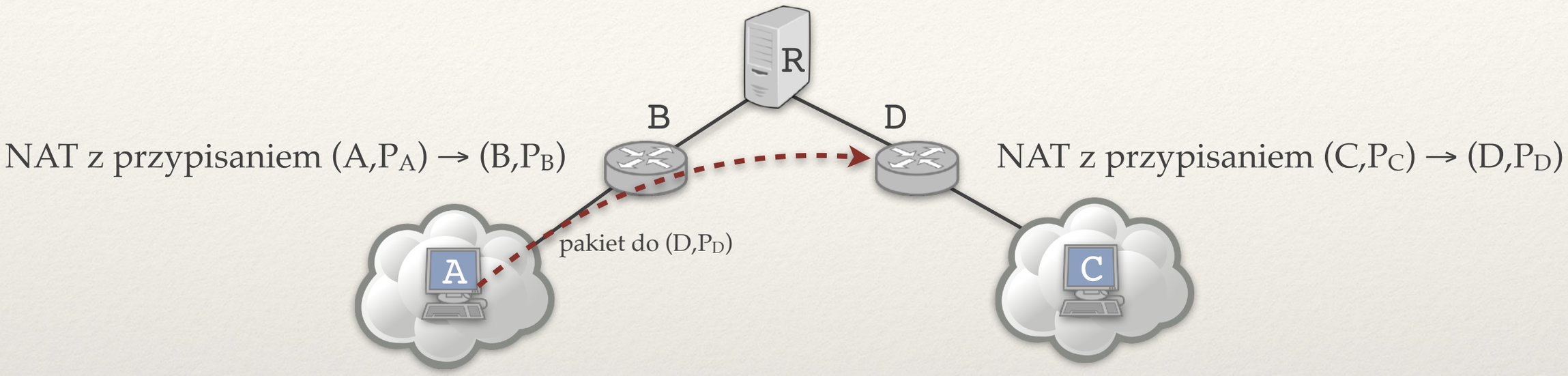
Fakt: Jeśli R wyśle dane do (D,P_D) to zostaną przesłane do (C,P_C).



Pytanie: Czy jeśli A wyśle dane do (D,P_D) to zostaną one przesłane do (C,P_C)?

Przechodzenie przez NAT (2)

Fakt: Jeśli R wyśle dane do (D,P_D) to zostaną przesłane do (C,P_C).

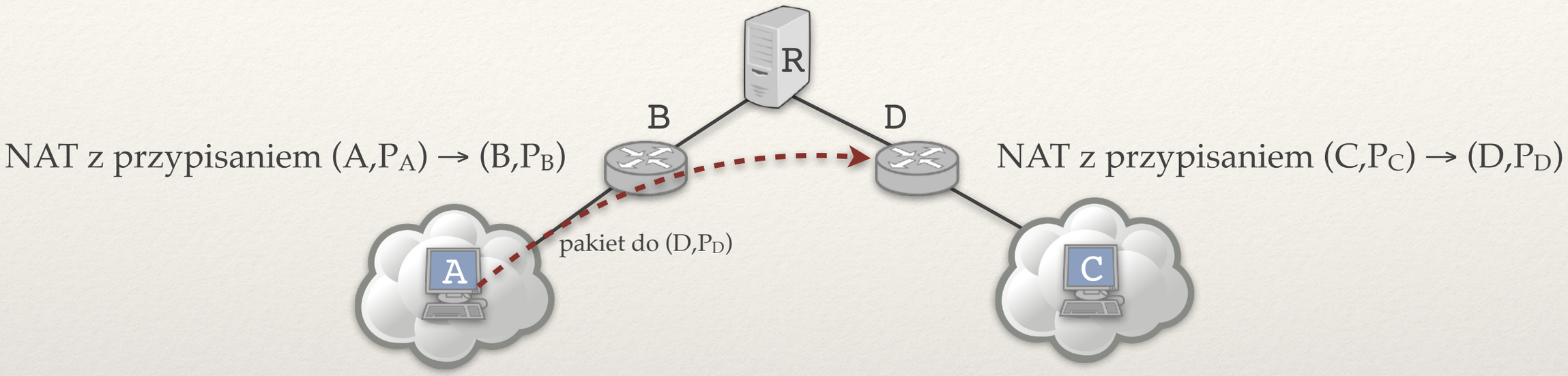


Pytanie: Czy jeśli A wyśle dane do (D,P_D) to zostaną one przesłane do (C,P_C)?

- Niestety nie zawsze!

Przechodzenie przez NAT (2)

Fakt: Jeśli R wyśle dane do (D,P_D) to zostaną przesłane do (C,P_C).

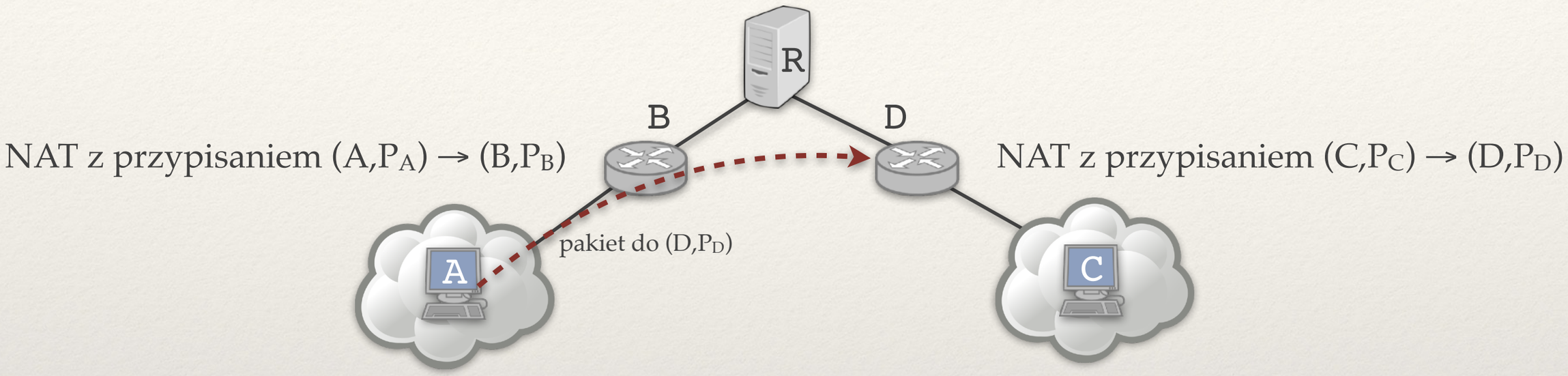


Pytanie: Czy jeśli A wyśle dane do (D,P_D) to zostaną one przesłane do (C,P_C)?

- Niestety nie zawsze!
- Ale jeśli tak jest, to możliwa jest komunikacja:
  - A adresuje pakiety do (D,P_D), przychodzą one do C jako pakiety od (B,P_B).
  - C adresuje pakiety do (B,P_B), przychodzą one do A jako pakiety od (D,P_D).

Przechodzenie przez NAT (2)

Fakt: Jeśli R wyśle dane do (D,P_D) to zostaną przesłane do (C,P_C).



Pytanie: Czy jeśli A wyśle dane do (D,P_D) to zostaną one przesłane do (C,P_C)?

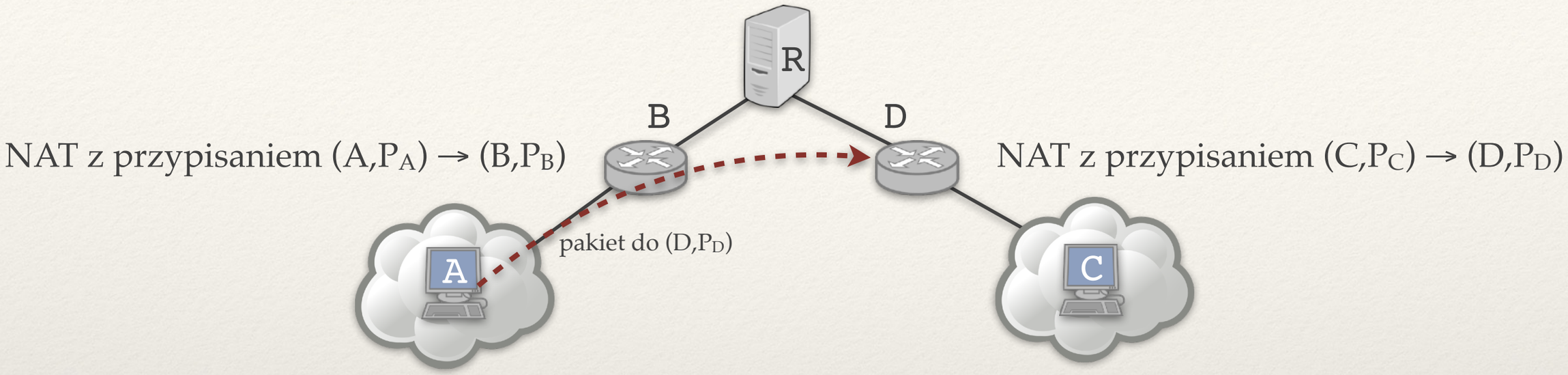
- Niestety nie zawsze!

dlaczego nie?

- Ale jeśli tak jest, to możliwa jest komunikacja:
  - A adresuje pakiety do (D,P_D), przychodzą one do C jako pakiety od (B,P_B).
  - C adresuje pakiety do (B,P_B), przychodzą one do A jako pakiety od (D,P_D).

Przechodzenie przez NAT (3)

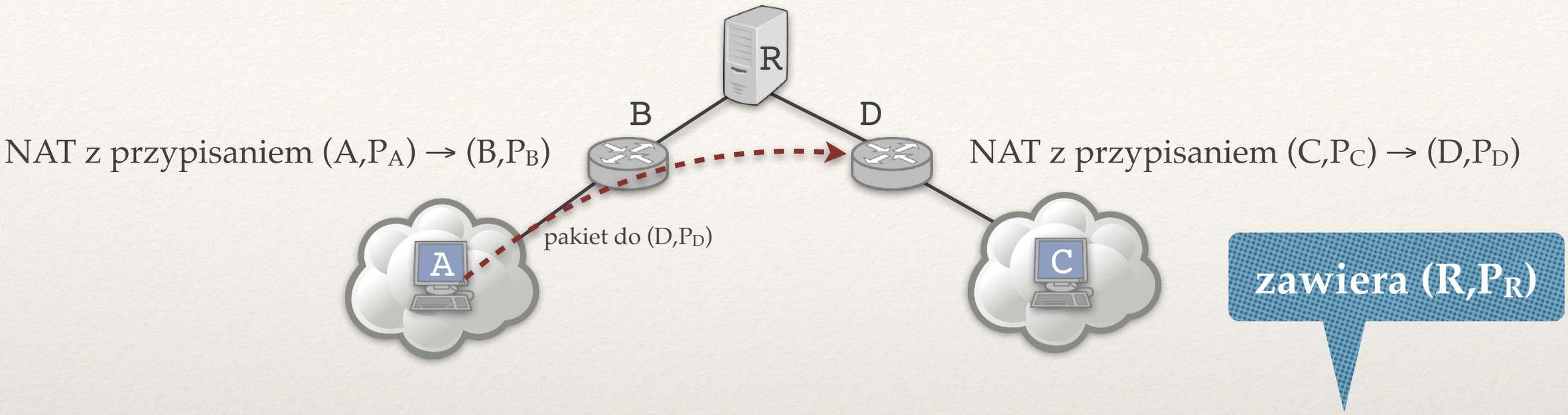
A wysyła pakiet do (D,P_D)…



Poza przypisaniem (C,P_C) → (D,P_D) router D pamięta **listę odbiorców** pakietów, które wychodziły przez (D,P_D).

Przechodzenie przez NAT (3)

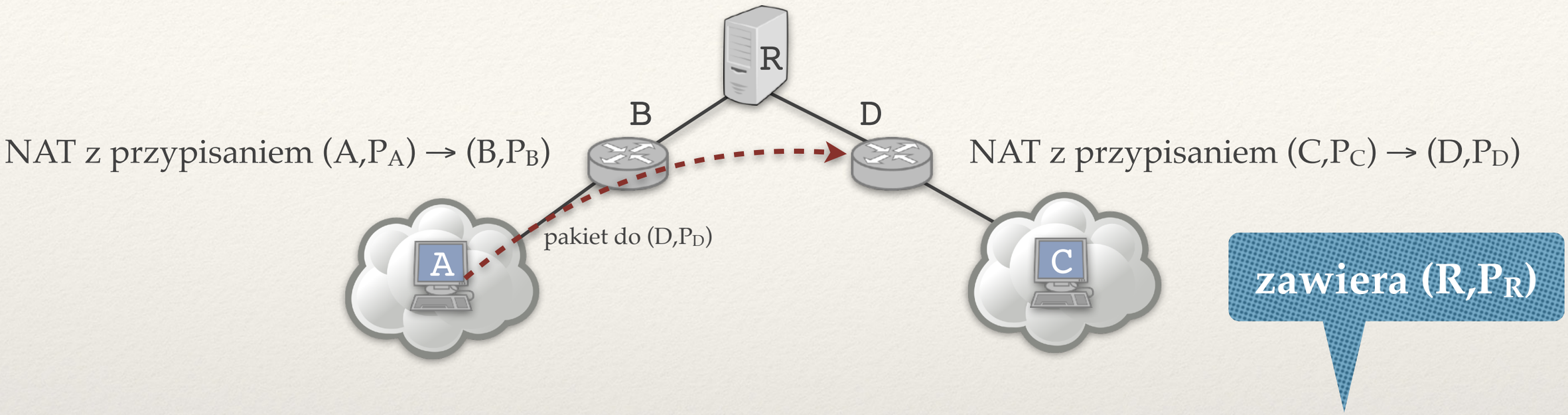
A wysyła pakiet do (D,P_D)…



Poza przypisaniem (C,P_C) → (D,P_D) router D pamięta **listę odbiorców** pakietów, które wychodziły przez (D,P_D).

Przechodzenie przez NAT (3)

A wysyła pakiet do (D,P_D)…



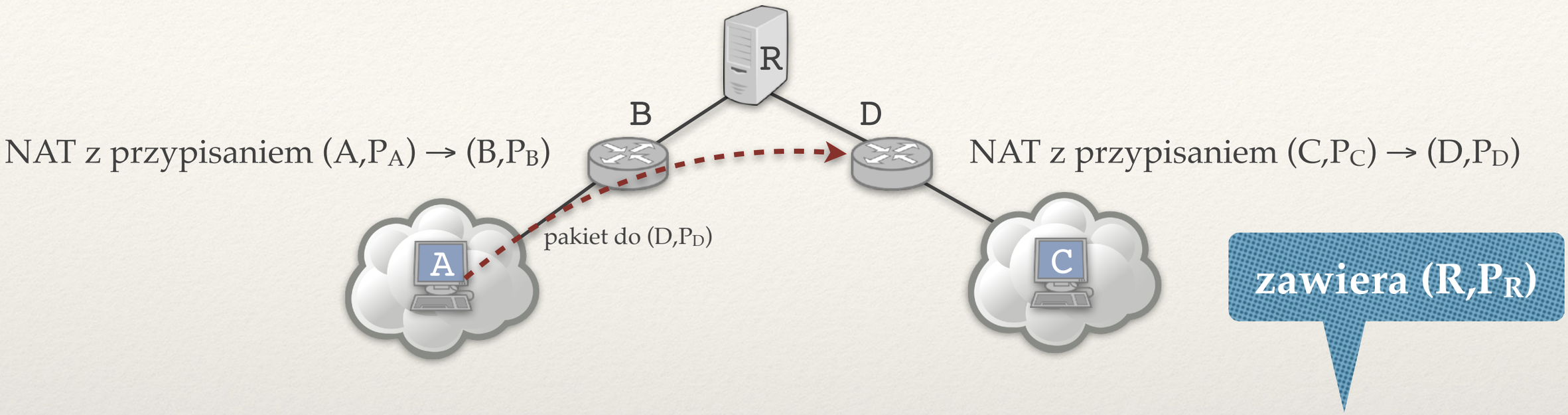
Poza przypisaniem (C,P_C) → (D,P_D) router D pamięta **listę odbiorców** pakietów, które wychodziły przez (D,P_D).

D przekaże do (C,P_C):

- wszystkie pakiety (**pełny asymetryczny NAT**).
- pakiety tylko od IP z listy (**ograniczony as. NAT**).
- pakiety tylko od par (IP,port) z listy (**ogranicz. portowo as. NAT**).

Przechodzenie przez NAT (3)

A wysyła pakiet do (D,P_D)…



Poza przypisaniem (C,P_C) → (D,P_D) router D pamięta **listę odbiorców** pakietów, które wychodziły przez (D,P_D).

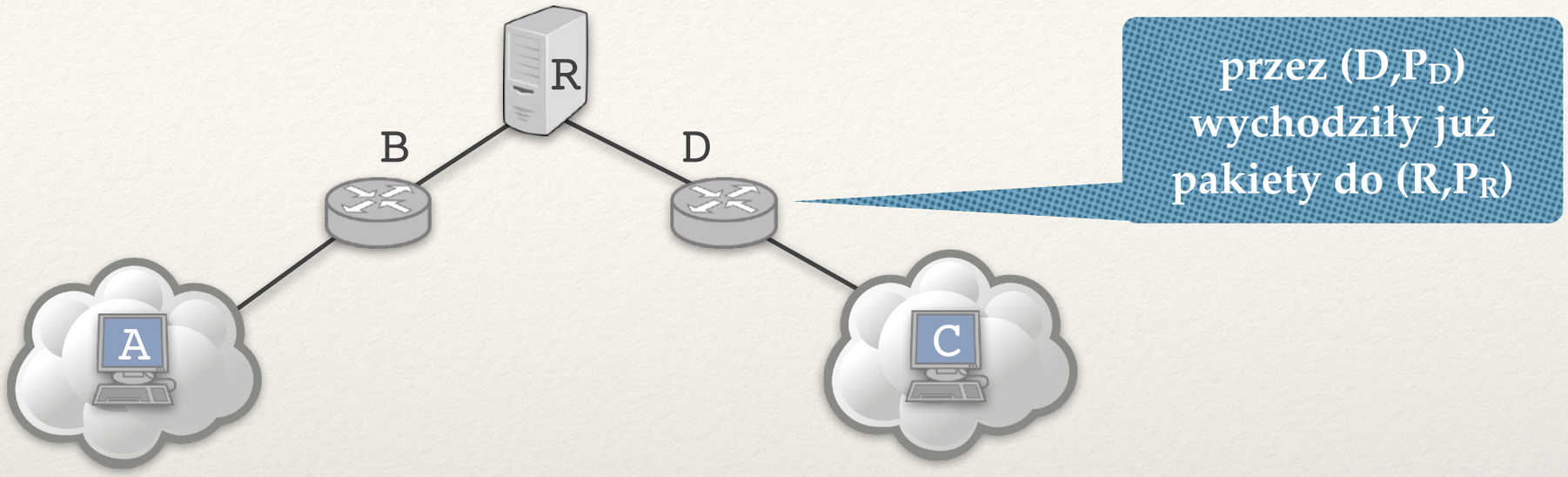
D przekaże do (C,P_C):

- wszystkie pakiety (**pełny asymetryczny NAT**).
- pakiety tylko od IP z listy (**ograniczony as. NAT**).
- pakiety tylko od par (IP,port) z listy (**ogranicz. portowo as. NAT**).

pakiet od (A,P_A) przejdzie przez D

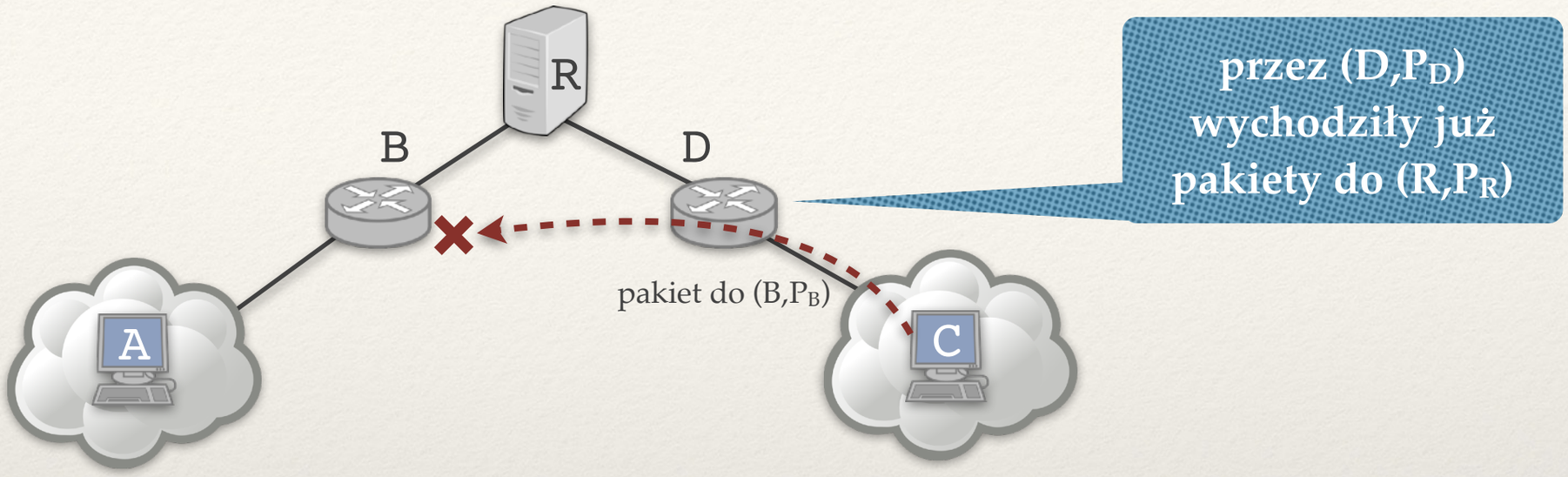
Wybijanie dziur (*hole punching*)

Co z asymetrycznymi ograniczonymi (portowo) NAT?



Wybijanie dziur (*hole punching*)

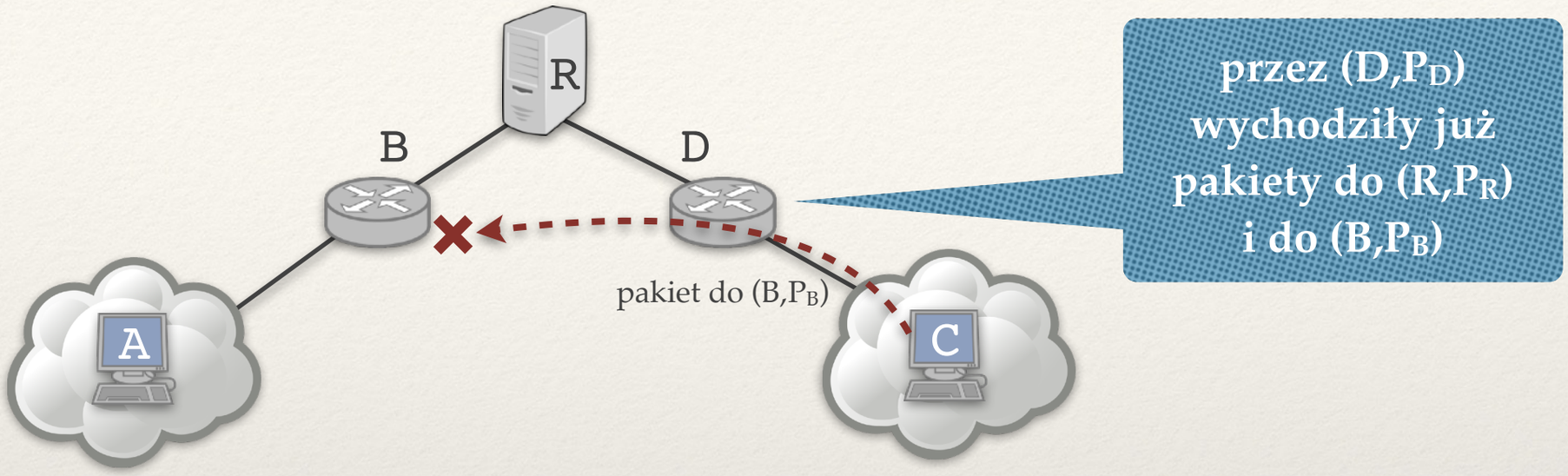
Co z asymetrycznymi ograniczonymi (portowo) NAT?



- (C,P_C) wysyła pakiet do (B,P_B). B odrzuca ten pakiet.

Wybijanie dziur (*hole punching*)

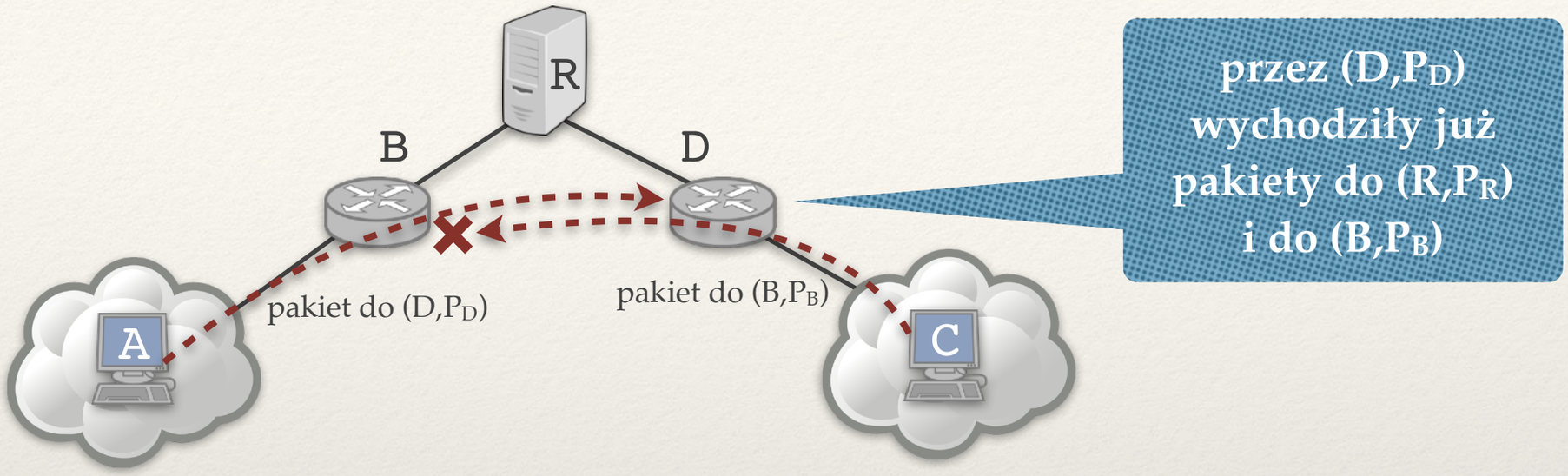
Co z asymetrycznymi ograniczonymi (portowo) NAT?



- (C,P_C) wysyła pakiet do (B,P_B). B odrzuca ten pakiet.
- Na routerze D: (B,P_B) dodany do listy odbiórców pakietów wychodzących przez (D,P_D)!

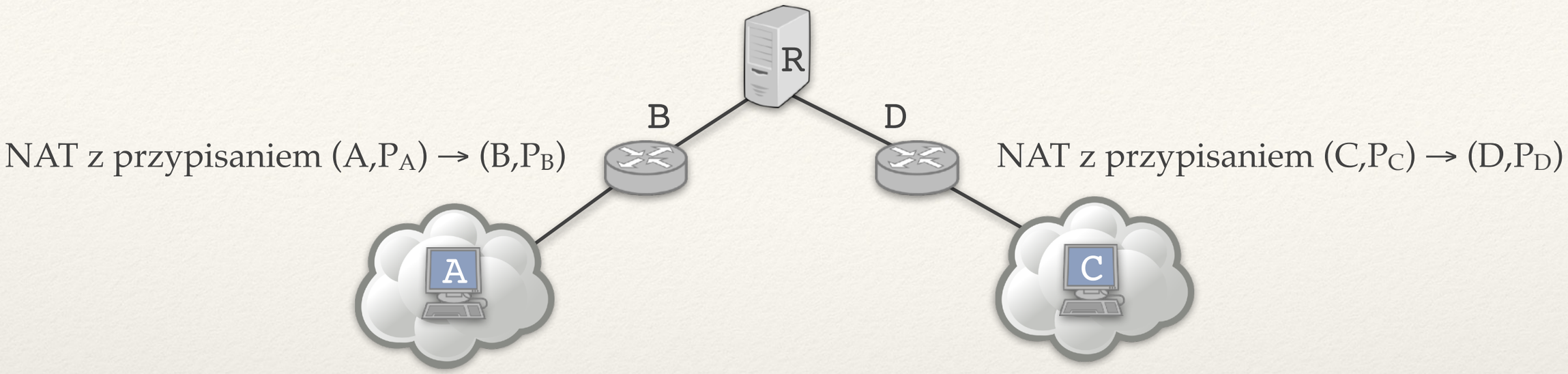
Wybijanie dziur (*hole punching*)

Co z asymetrycznymi ograniczonymi (portowo) NAT?



- (C,P_C) wysyła pakiet do (B,P_B). B odrzuca ten pakiet.
- Na routerze D: (B,P_B) dodany do listy odbiórców pakietów wychodzących przez (D,P_D)!
- (A,P_A) wysyła pakiet do (D,P_D):
  - adres źródłowy zostaje podmieniony na (B,P_B)
  - D przepuszcza pakiet „od (B,P_B)“ do (C,P_C).

NAT symetryczny



- Milcząco założyliśmy, że jeśli (A,P_A) wysyłało pakiet do (R,P_R) i potem do (D,P_D), to w obu przypadkach B wybierze port P_B.

- **NAT asymetryczny:** P_B zależy tylko od adresu i portu nadawcy.
- **NAT symetryczny:** P_B zależy od adresu i portu nadawcy i odbiorcy. Wybijanie dziur nie działa, pomagają tylko przekaźniki.

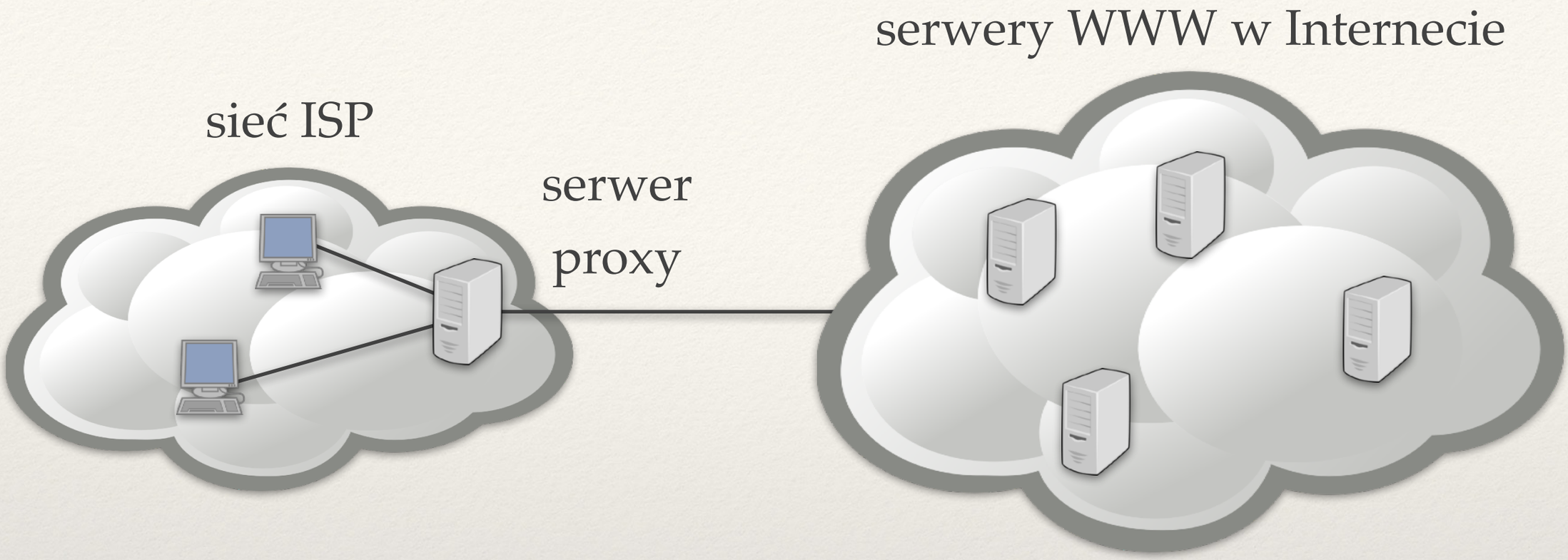
Wydajność HTTP

Poprawianie wydajności HTTP

- **Połączenia trwałe:**
  - Wiele żądań i odpowiedzi HTTP w jednym połączeniu TCP (standardowe zachowanie HTTP 1.1).

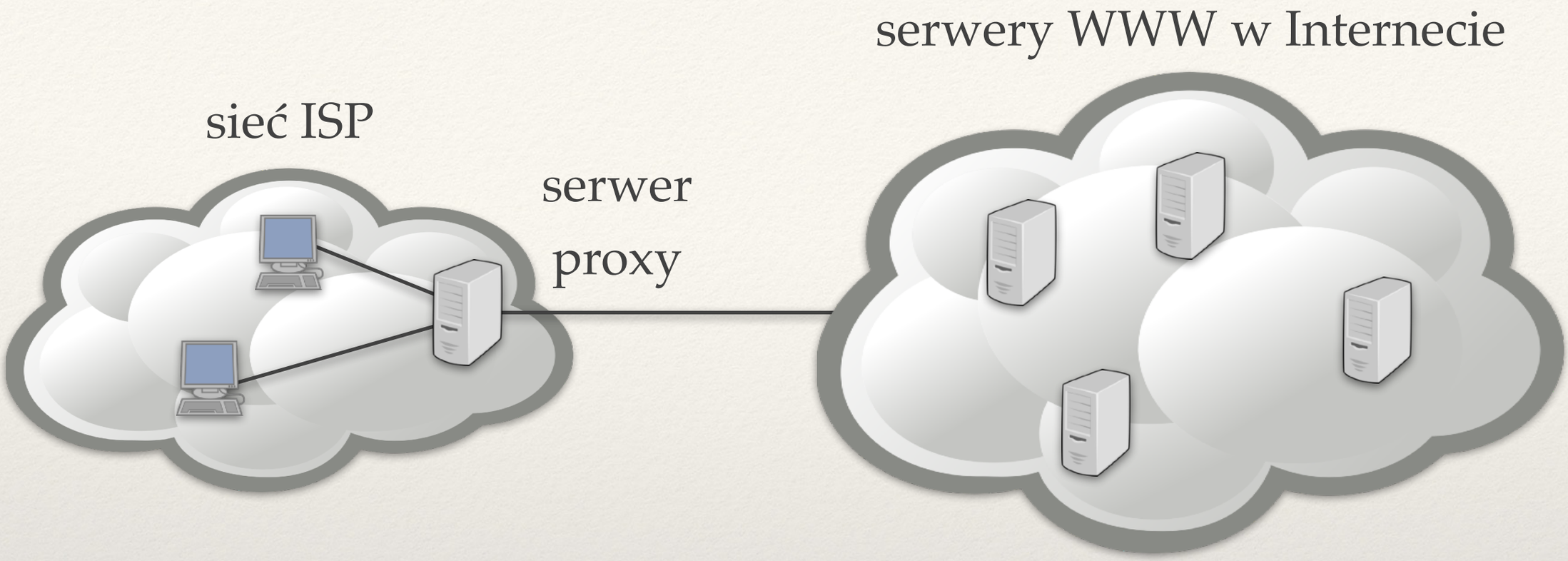
- **Pamięć podręczna w przeglądarce WWW:**
  - Zapytanie GET z polem `If-Modified-Since:`
  - Serwer może umieszczać w nagłówku odpowiedzi pola:
    - `Expires:` (do kiedy można trzymać dokument w pamięci podręcznej) → można całkowicie pominąć żądanie strony.
    - `Cache-Control: no-cache` (nigdy nie trzymaj w pamięci podręcznej)

Serwery proxy (1)



- Przeglądarka wysyła zapytanie HTTP do serwera proxy.
- Proxy w razie potrzeby łączy się z serwerem HTTP.
- Serwer proxy odpowiada używając stron przechowywanych w swojej pamięci podręcznej.
- W razie potrzeby przeglądarka może wymusić pominięcie proxy.

Serwery proxy (2)



Serwer proxy

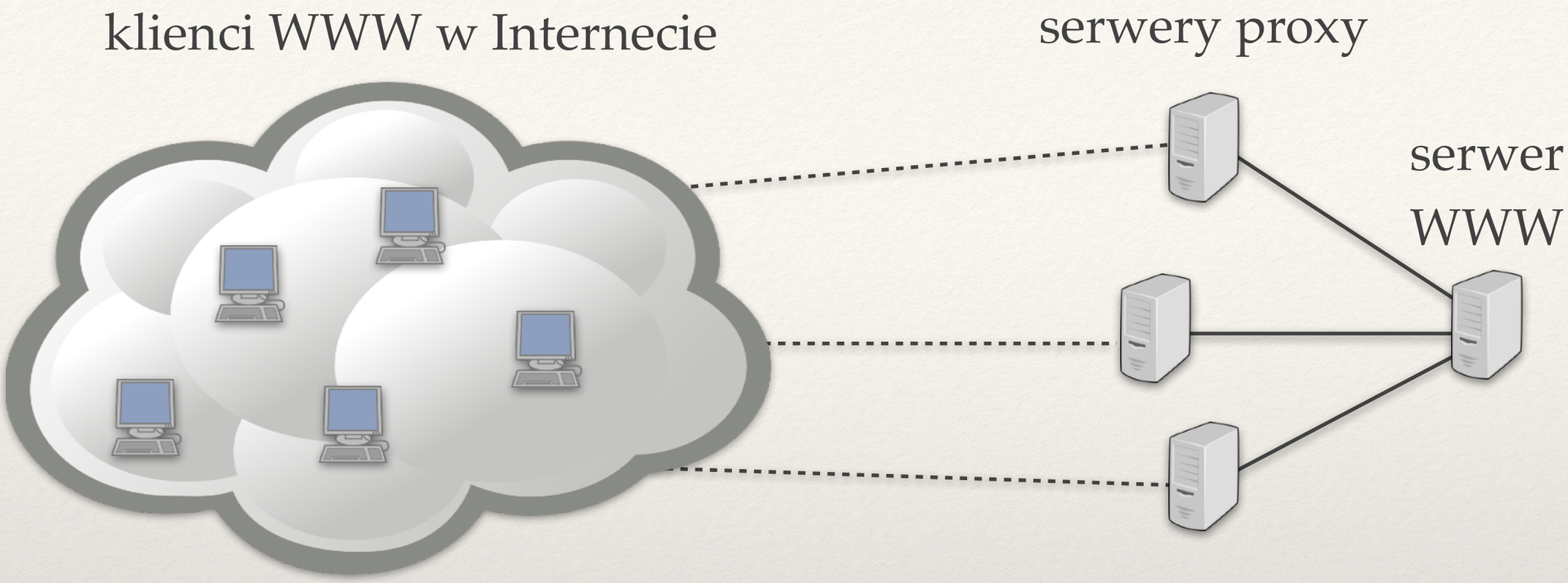
- Wpisywany w ustawieniach przeglądarki HTTP
- Czasem wymuszany przez ISP (integrowany z routerem obsługującym ruch z danej sieci).
- Korzyści: głównie dla ISP (ograniczenie ilości danych).

Anonimizujące serwery proxy

- **Serwer proxy dodaje do żądania HTTP dodatkowe pola.**
  - `X-Forwarded-For:` adres IP.
  - `Via:` adres IP proxy.

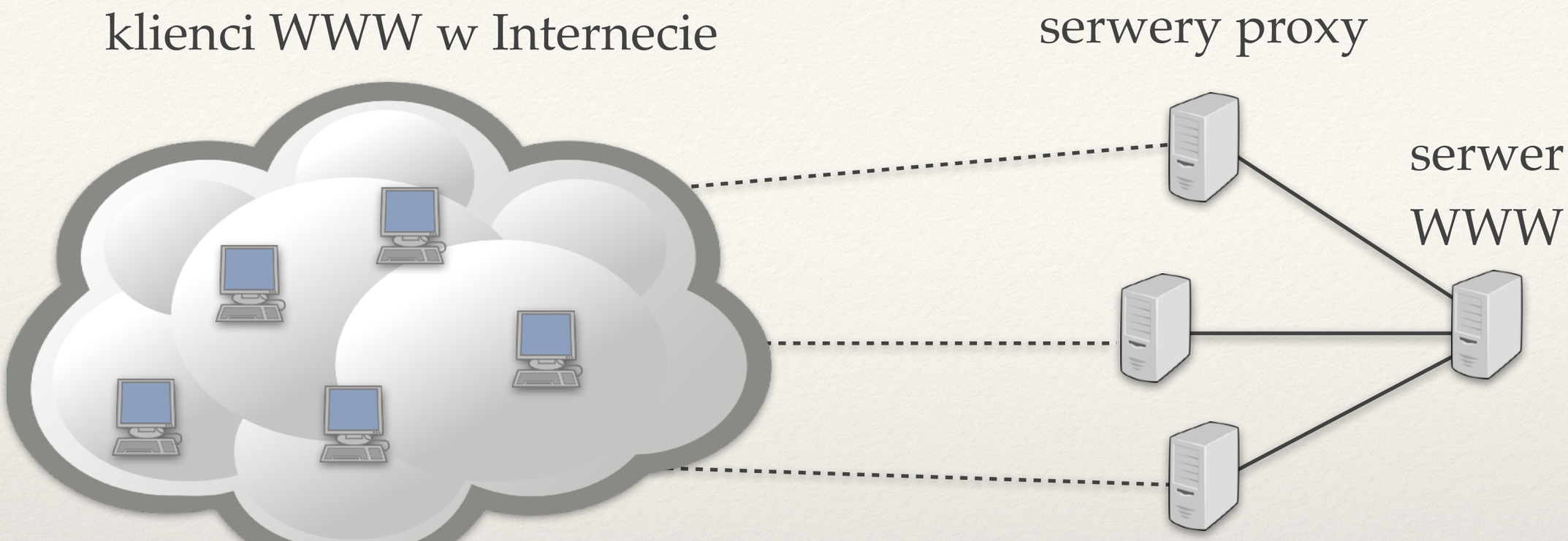
- **Anonimizujące serwery proxy:**
  - Nie dodają takich nagłówków.
  - Zwykle płatne.

Odwrotne proxy (1)



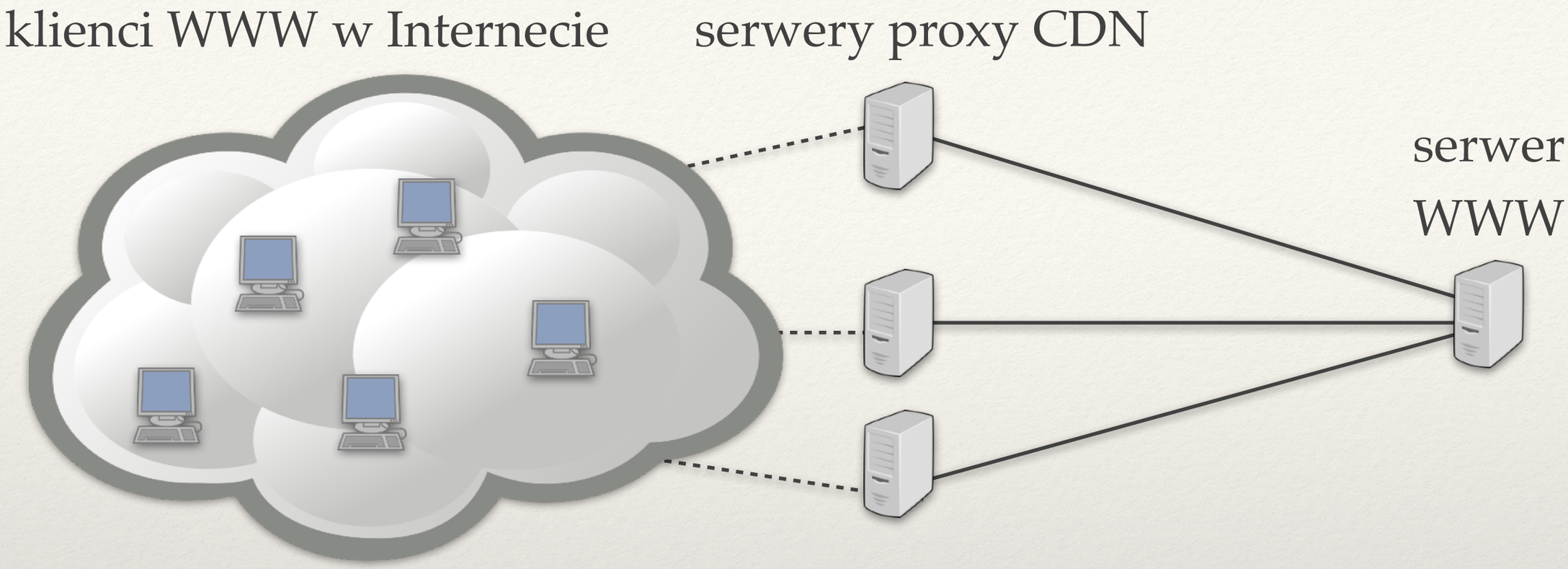
- Wykorzystywane przez dostawców treści.
- Zmniejszają obciążenie samego serwera WWW.
- Adresy IP serwerów proxy podawane zazwyczaj przez DNS jako adresy IP przy rozwiązywaniu nazwy serwera WWW.
  - Serwery DNS zazwyczaj zwracają listę adresów IP w losowej albo cyklicznej kolejności.

Odwrotne proxy (2)



- Zysk dla klienta i dostawcy treści.
- Ale wciąż duże opóźnienie w przesyłaniu pakietów pomiędzy klientami i serwerami proxy.
- Jak opłacalnie przysunąć serwery proxy do klientów?

CDN (Content Distribution Networks)



- Serwery proxy obsługiwane przez osobną organizację (obsługuje wiele serwerów WWW).
  - Akamai, Limelight, …
  - Setki tysięcy serwerów proxy.
- CDN utrzymuje również serwery DNS: umożliwiają wybieranie bliskiego serwera proxy.

Poczta elektroniczna

Protokół SMTP

- Protokół przekazywania poczty.
- Protokół tekstowy, serwer nasłuchuje na porcie 25.

Wysyłanie bezpośrednie (1)

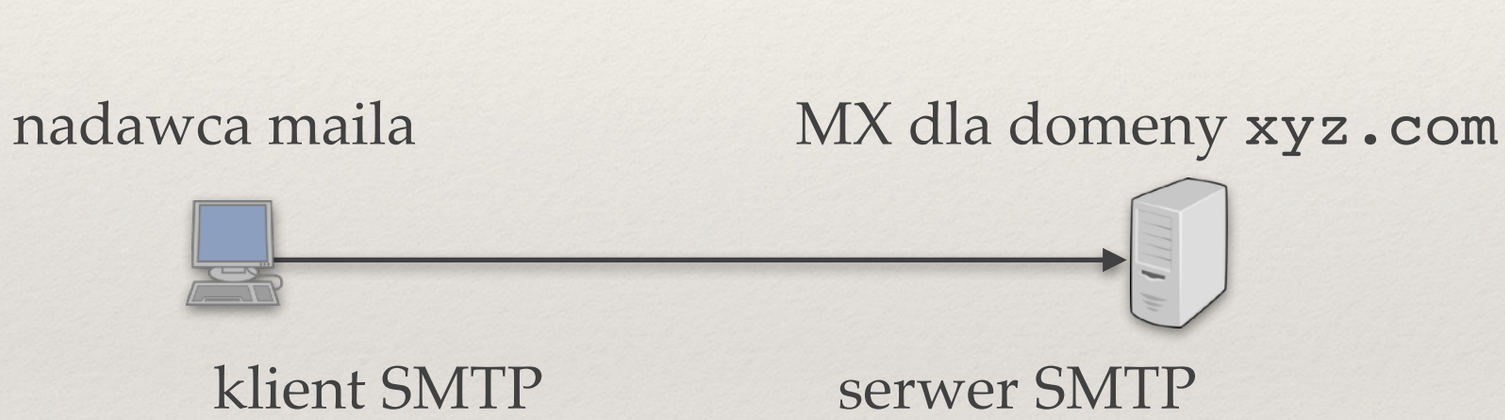
Chcemy wysłać pocztę do adresu `abc@xyx.com`.

- Łączymy się z adresem IP serwera odpowiedzialnego za odbieranie i przechowywanie pocztę dla domeny `xyx.com`.

- Rekord MX (*mail exchange*) w DNS a rekord A:
  - A, `ii.uni.wroc.pl` → **`156.17.4.11`**
  - MX, `ii.uni.wroc.pl` → `aspmx.l.google.com`
  - A, `aspmx.l.google.com` → **`66.102.1.26`**
  - Wysyłając pocztę do `abc@ii.uni.wroc.pl` łączymy się z `66.102.1.26` (nie z `156.17.4.11`).

Wysyłanie bezpośrednie (2)

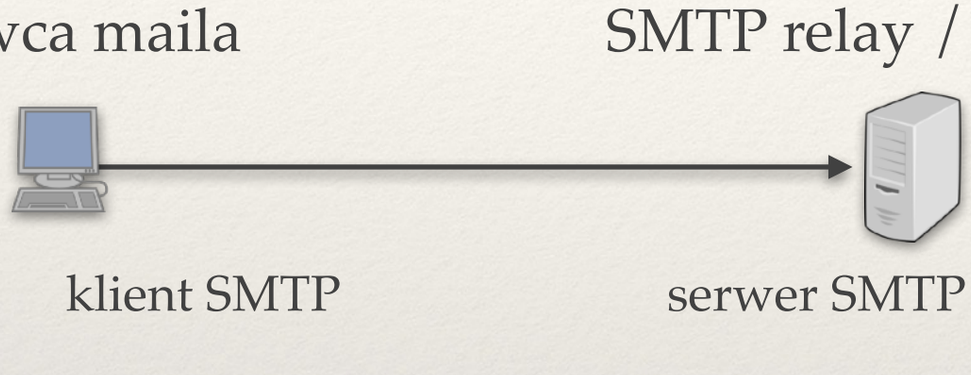
Chcemy wysłać pocztę do adresu `abc@xyx.com`.



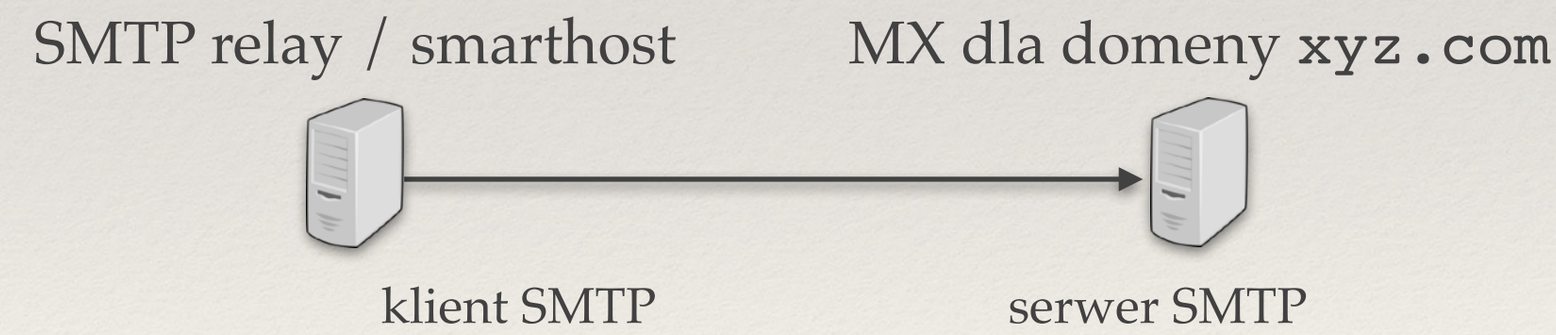
Wysyłanie pośrednie

Chcemy wysłać pocztę do adresu `abc@xyx.com`.

Krok 1.



Krok 2.



(Kroków może być więcej).

Wysyłanie pośrednie

Chcemy wysłać pocztę do adresu `abc@xyx.com`.

Krok 1.

nadawca maila — SMTP relay / smarthost

klient SMTP — serwer SMTP

= to co program pocztowy klienta ma ustawione jako „serwer SMTP“

Krok 2.

SMTP relay / smarthost — MX dla domeny `xyz.com`

klient SMTP — serwer SMTP

(Kroków może być więcej).

Przekazywanie pośrednie

- Zazwyczaj wymaga autoryzacji nadawcy u SMTP relay
  - Różne mechanizmy autoryzacji są elementem protokołu SMTP.
  - Zabezpieczenie przed rozsyłaniem niechcianej poczty (spamu).
  - Czasem autoryzacja na podstawie adresu IP klienta.

Przykładowy email (otrzymany)

```
Delivered-To: marcin.bienkowski@cs.uni.wroc.pl
Received: by 10.64.232.142 with SMTP id to14csp146725iec;
        Sat, 23 Apr 2016 08:41:37 -0700 (PDT)
Received: from aisd.ii.uni.wroc.pl (156.17.4.30)
        by mx.google.com with ESMTP id l199si7484936lfl.24.2016.04.23.08.41.36
        for <marcin.bienkowski@cs.uni.wroc.pl>;
        Sat, 23 Apr 2016 08:41:36 -0700 (PDT)
Received: by aisd.ii.uni.wroc.pl (Postfix, from userid 1000) id E6BCD5F84D;
        Sat, 23 Apr 2016 17:41:35 +0200 (CEST)
Date: Sat, 23 Apr 2016 17:41:35 +0200
From: mbi <mbi@ii.uni.wroc.pl>
To: marcin.bienkowski@cs.uni.wroc.pl
Subject: Testowy email
Message-ID: <20160423154135.GA11834@aisd.ii.uni.wroc.pl>
MIME-Version: 1.0
Content-Type: text/plain; charset=utf-8
Content-Disposition: inline
Content-Transfer-Encoding: 8bit
User-Agent: Mutt/1.5.23 (2014-03-12)


Jakaś treść maila.
M.
```

pola ustawiane przez odbiorcę

pola ustawiane przez serwery pośredniczące

pola ustawiane przez nadawcę

Pola nagłówka ustawiane przez klienta

- From:
- To:
- Subject:
- Cc:
- Bcc: („ślepa kopia“)
- Message-ID: (unikatowy identyfikator wiadomości)
- Date: (data wysłania)
- In-Reply-To: (ID maila, na którego odpowiadamy)
- References:

Typ zawartości

Pole `Content-Type:` nagłówka określa:

- czym jest treść maila (w standardzie MIME)
  - czysty tekst (`text/plain`)
  - HTML (`text/html`)
- kodowanie znaków

```
Content-Type: text/plain; charset=utf-8
Content-Transfer-Encoding: 8bit
```

Załączniki pocztowe

```
Content-Type: multipart/mixed; boundary=„--UNIKATOWY-CIĄG-ZNAKÓW“
Content-Transfer-Encoding: 8bit

--UNIKATOWY-CIĄG-ZNAKÓW
Content-Type: text/plain; charset=utf-8
Content-Disposition: inline
Content-Transfer-Encoding: 8bit

Wiadomość testowa
M.

--UNIKATOWY-CIĄG-ZNAKÓW
Content-Type: image/jpeg
Content-Disposition: attachment; filename="obrazek.jpg"
Content-Transfer-Encoding: base64

ZAWARTOŚĆ-PLIKU-ZAKODOWANA-W-BASE64.

--UNIKATOWY-CIĄG-ZNAKÓW
```

treść tekstowego maila w UTF-8

załącznik obrazek.jpg

Dostarczanie poczty do użytkownika

- Protokół POP3.
- Protokół IMAP.
- Klienty pocztowe jako aplikacje WWW.

Spam: niechciane wiadomości pocztowe

Sposoby wykrywania i usuwania spamu:

- metody statystyczne, uczenie maszynowe
- greylisting
- SPF
- …

Greylisting

Wolne wysyłanie spamu jest nieopłacalne.

- Pomysł: każemy klientowi SMTP wysłać wiadomość ponownie za jakiś czas:

```
 -> MAIL FROM: <nadawca@jakasdomena.pl>
<- 250 2.1.0 Sender ok
-> RCPT TO: <odbiorca@innadomena.pl>
<- 451 4.7.1 Please try again later
```

- Ustawiamy okno czasowe (np. „nie wcześniej niż za 10 min. i nie później niż po godzinie“).
  - Jeśli klient ponowi w danym oknie czasowym, to akceptujemy jego email.
  - Problemy z poprawnym ustawieniem okna.
  - Dostarczanie poczty przestaje być szybkim procesem.

- Stosowany wariant: zamiast odrzucać, odbieraj wolniej z okna TCP.

Spam: SPF (Sender Policy Framework)

- Rekord SPF (o typie TXT) w DNS dla danej domeny:
  - TXT, `ii.uni.wroc.pl` → `„v=spf1 ip4:156.17.4.0/24 mx:ii.uni.wroc.pl mx:gmail.com mx:google.com -all"`
- Definiuje jakie komputery są uprawnione do wysyłania poczty z polem `From:` równym *`adres`*`@ii.uni.wroc.pl`.
  - komputery z adresów `156.17.4.0/24`.
  - serwery SMTP obsługujące pocztę dla domen `ii.uni.wroc.pl`, `gmail.com` i `google.com`.
- Rekord sprawdzany przez odbiorcę.

Lektura dodatkowa

- Kurose & Ross: rozdział 2.
- Tanenbaum: rozdział 7.
- `https://www.sandvine.com/global-internet-phenomena-report-2019`

Zagadnienia

- Jaka jest rola trackera w sieci Bittorrent?
- Po co w plikach `.torrent` stosuje się funkcje skrótu?
- Jakie są różnice w postępowaniu *seedera* i *leechera* w sieci BitTorrent?
- Na czym polegają połączenia odwrócone? Jak stosuje się je w protokole FTP?
- Opisz podobieństwa i różnice asymetrycznych (*cone)* NAT (pełnego, ograniczonego i ograniczonego portowo) i symetrycznych NAT.
- Opisz technikę wybijania dziur (*hole punching*) w NAT. Po co konieczny jest serwer pośredniczący?
- Do czego służą serwery proxy?
- Co to jest odwrotne proxy? Co to jest CDN?
- Jak skłonić klienta, żeby łączył się z serwerem proxy a nie bezpośrednio ze stroną WWW?
- Jakie informacje dołączane są przez serwer proxy do zapytania?
- Co to są anonimowe serwery proxy?
- Do czego służy protokół SMTP a do czego POP3?
- Co to jest przekazywanie poczty (*relaying*)? Co to jest *smarthost*?
- Jaki rekord DNS jest sprawdzany przed wysłaniem poczty do danej domeny?
- Wymień parę popularnych pól w nagłówku maila. Do czego służą pola `Received` i `Bcc`?
- Jakie pola w nagłówku są używane do tworzenia wątków z wiadomościami?
- Co umożliwia standard MIME?
- Co to jest spam? Jakie znasz metody walki ze spamem?
- Na czym polega greylisting?
- Na czym polega mechanizm SPF?