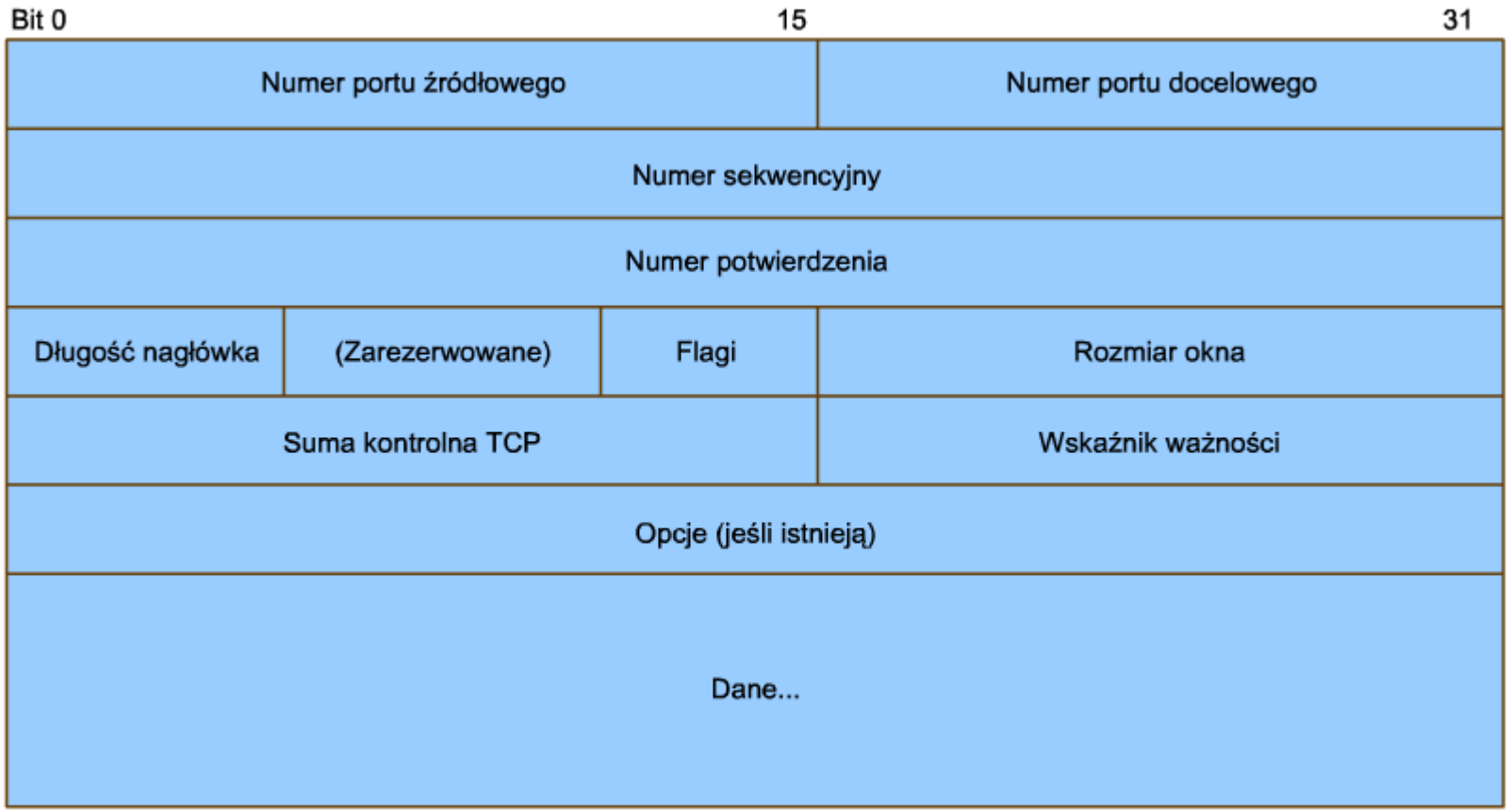
TCP i UDP

TCP

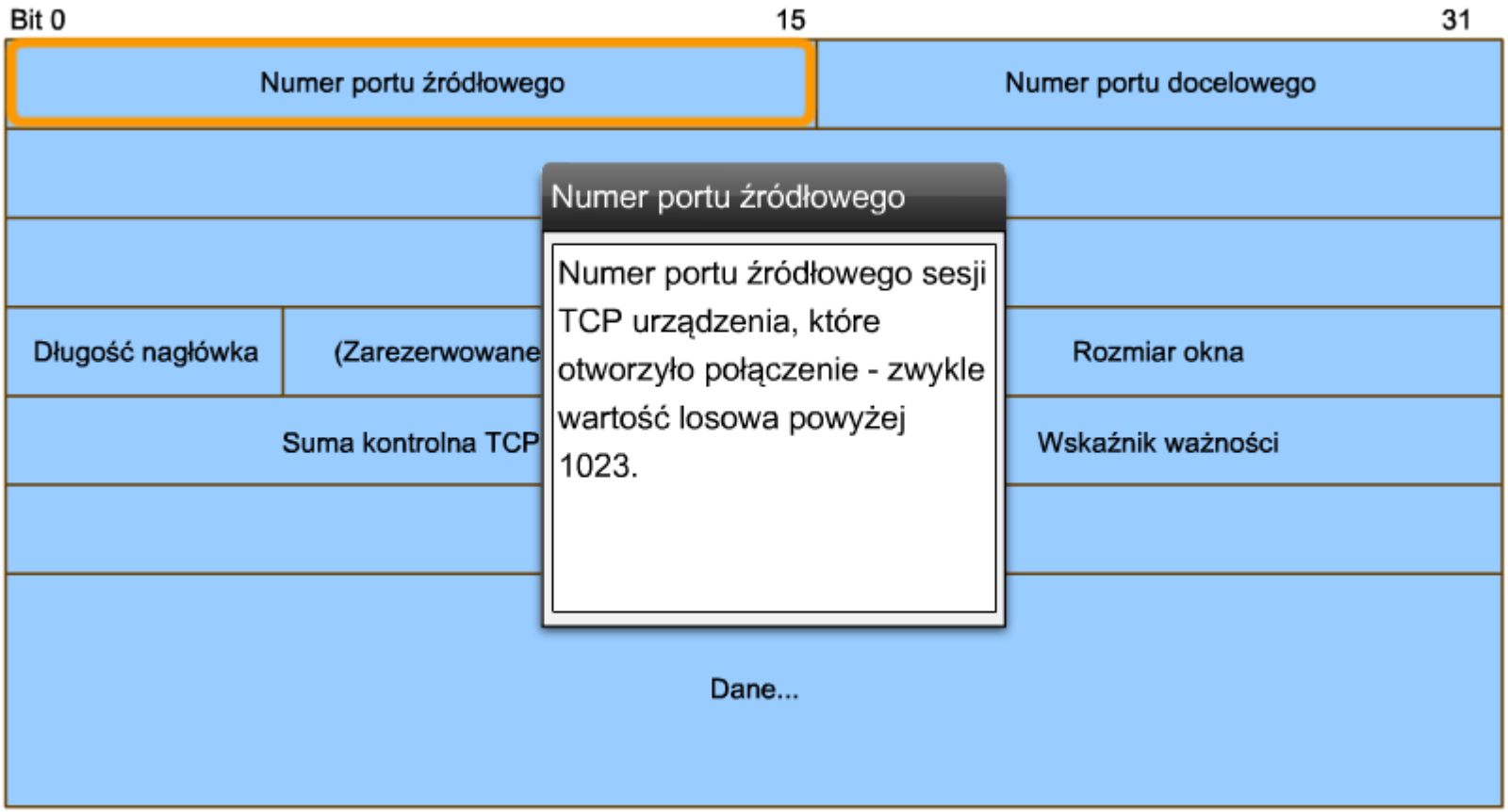
- Niezawodność komunikacji TCP dokonywana jest przez zorientowane połączeniowo sesje. Połączenie to umożliwia śledzenie sesji lub strumienia komunikacyjnego pomiędzy hostami. Dzięki temu procesowi obie strony są świadome komunikacji i są do niej odpowiednio przygotowane.
- Niezawodność komunikacji TCP opiera się na potwierdzeniach odebranych segmentów!!

Nagłówek TCP



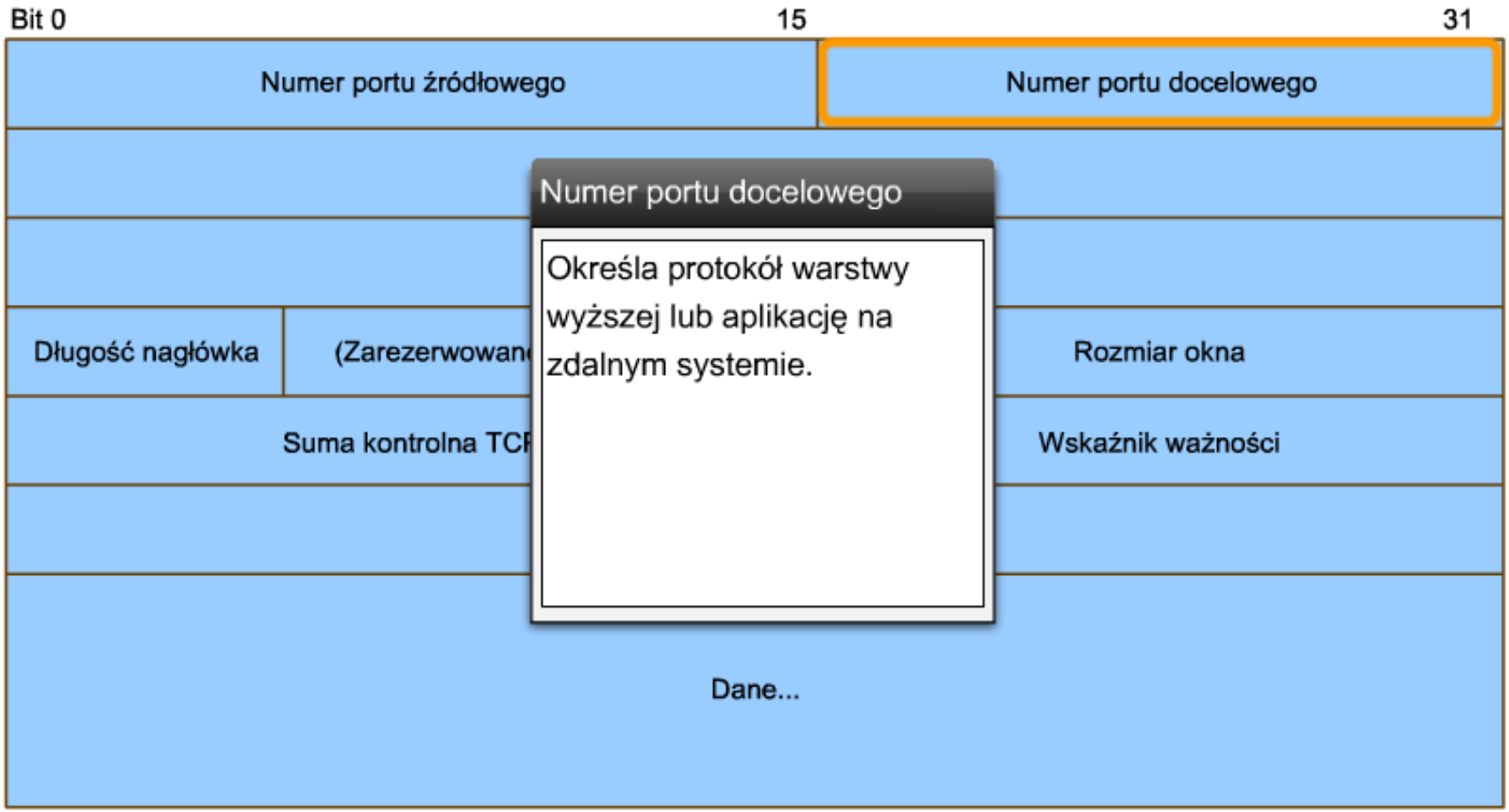
Pola w nagłówku TCP umożliwiają zorientowaną połączeniowo, niezawodną wymianę danych.

Nagłówek TCP



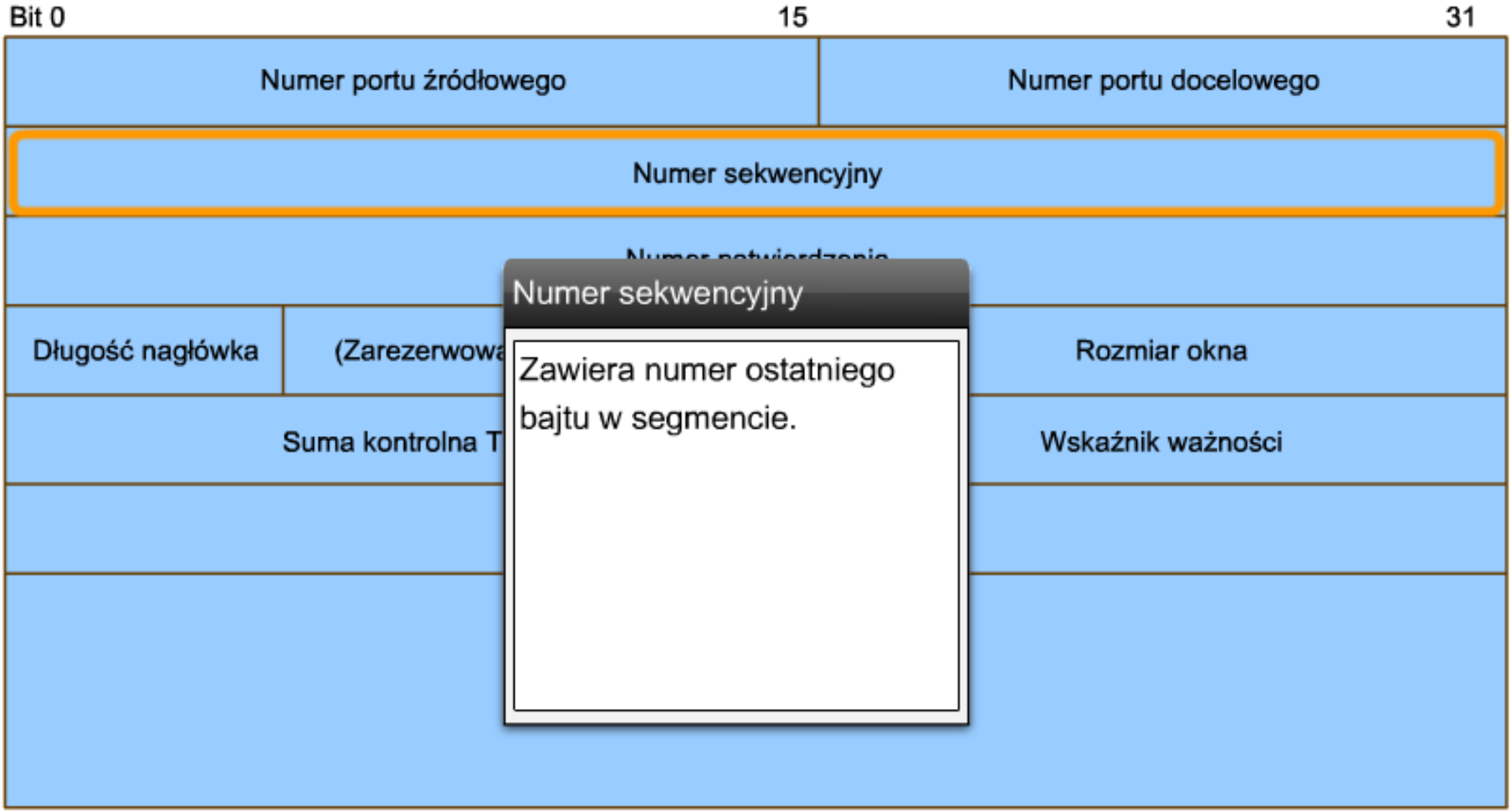
Pola w nagłówku TCP umożliwiają zorientowaną połączeniowo, niezawodną wymianę danych.

Nagłówek TCP



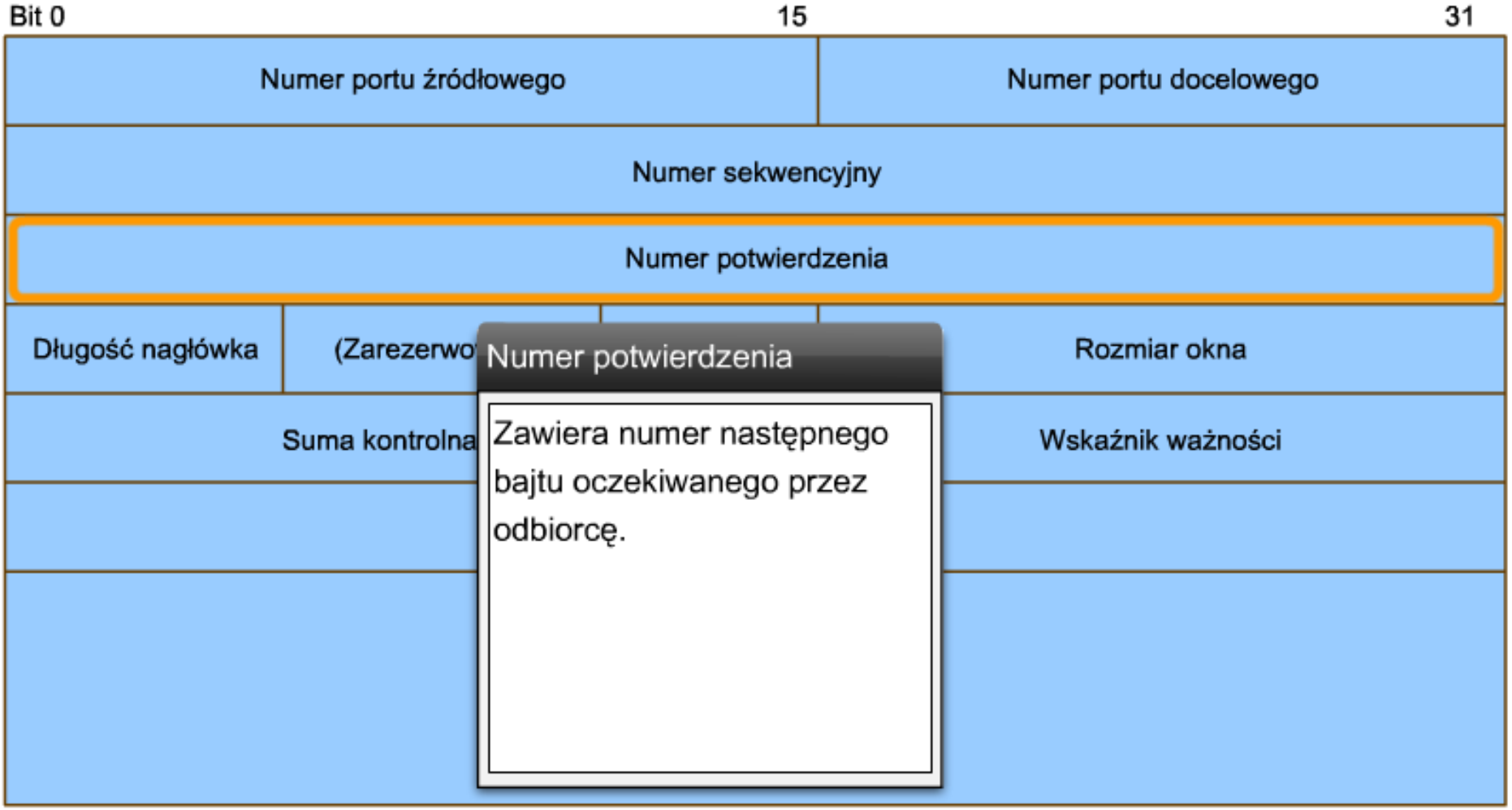
Pola w nagłówku TCP umożliwiają zorientowaną połączeniowo, niezawodną wymianę danych.

Nagłówek TCP



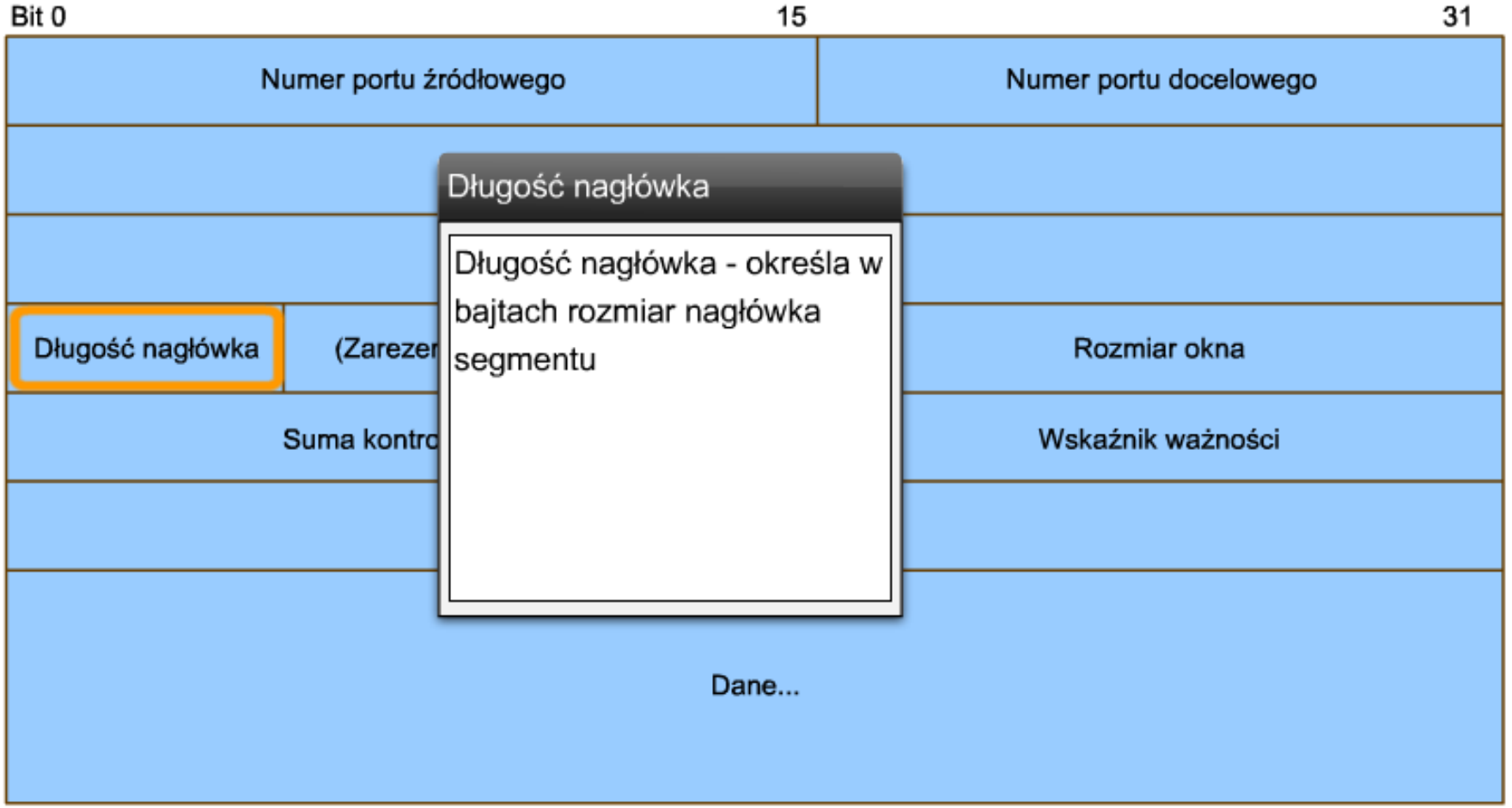
Pola w nagłówku TCP umożliwiają zorientowaną połączeniowo, niezawodną wymianę danych.

Nagłówek TCP



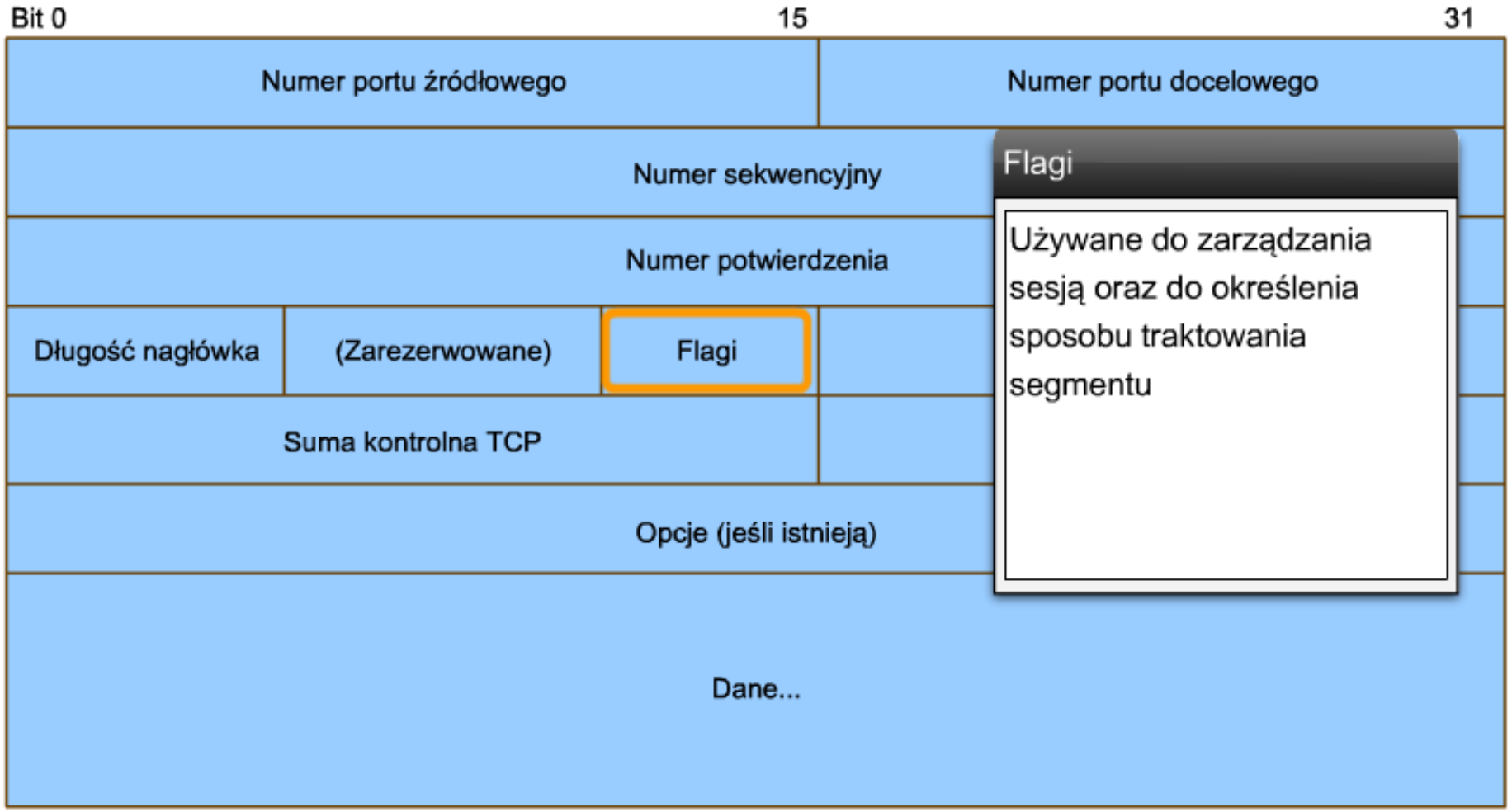
Pola w nagłówku TCP umożliwiają zorientowaną połączeniowo, niezawodną wymianę danych.

Nagłówek TCP



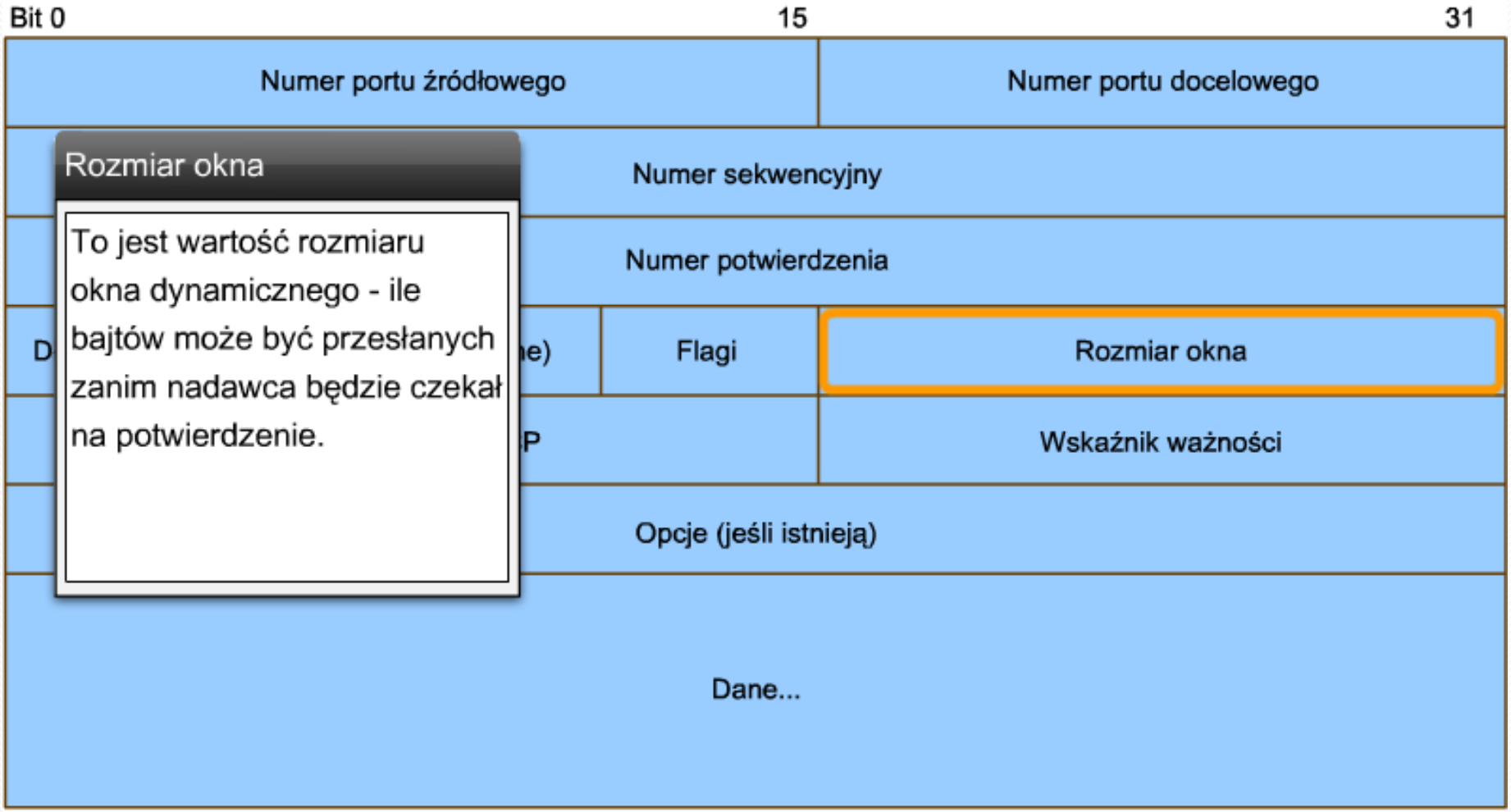
Pola w nagłówku TCP umożliwiają zorientowaną połączeniowo, niezawodną wymianę danych.

Nagłówek TCP



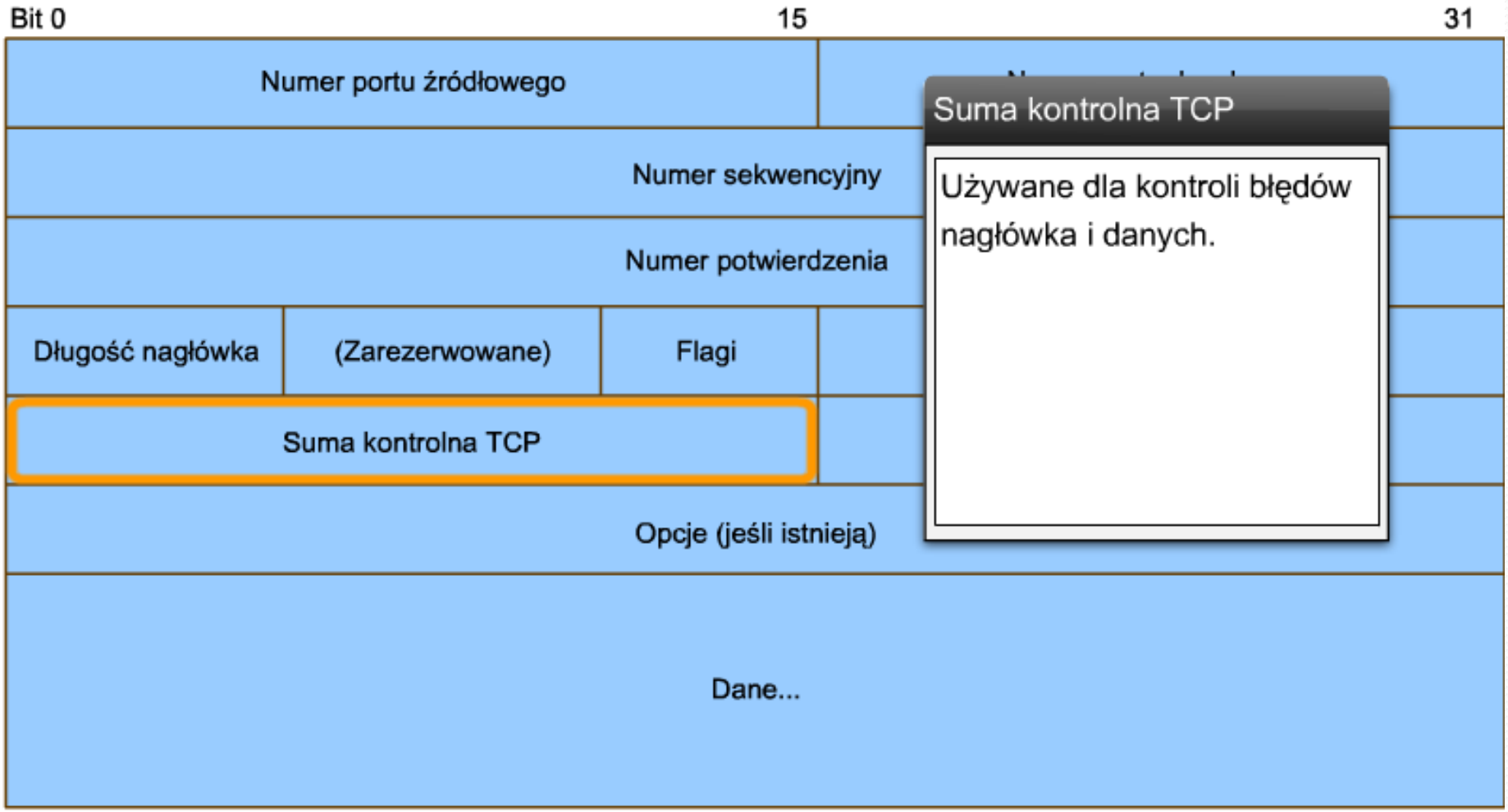
Pola w nagłówku TCP umożliwiają zorientowaną połączeniowo, niezawodną wymianę danych.

Nagłówek TCP



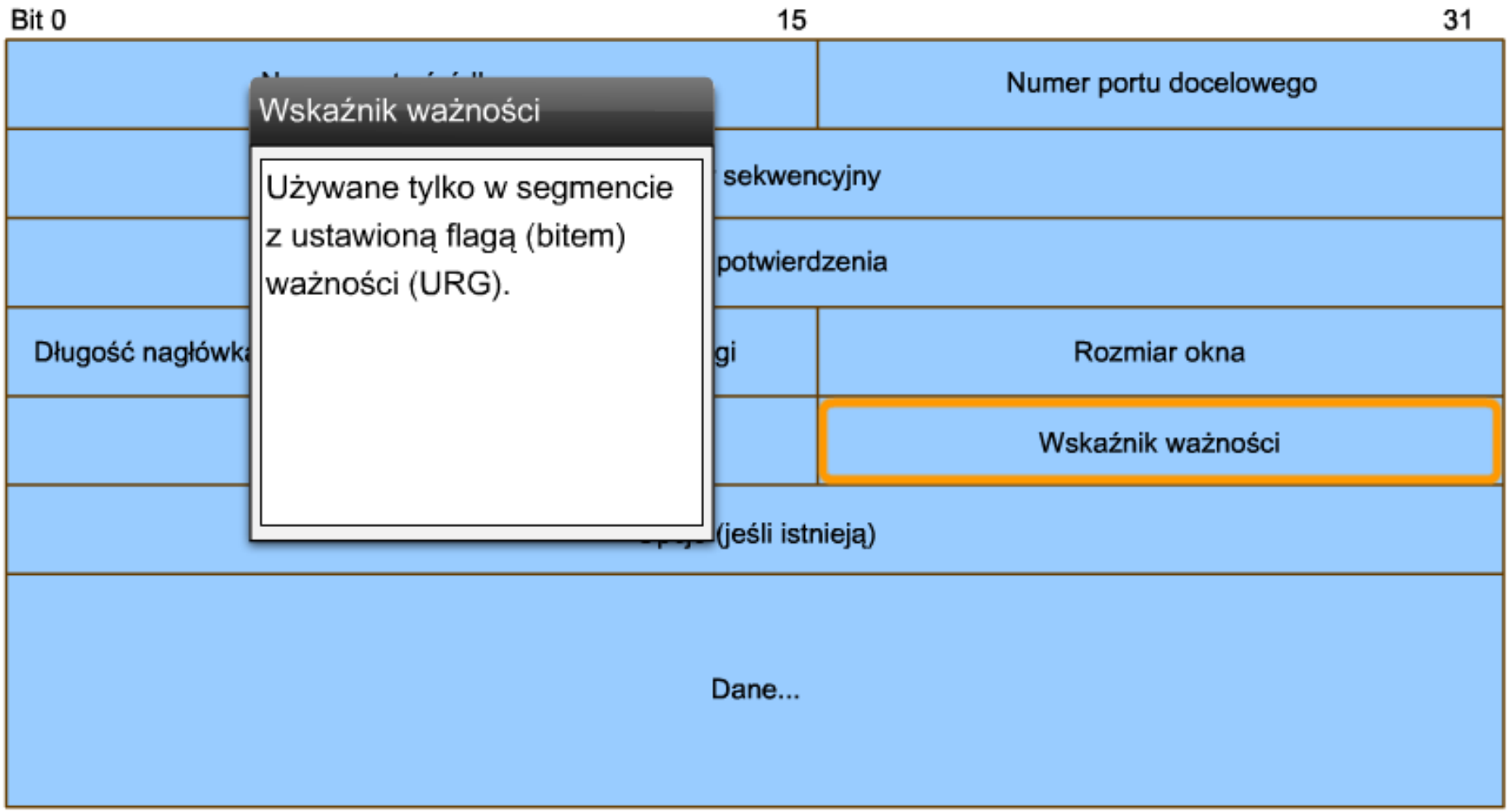
Pola w nagłówku TCP umożliwiają zorientowaną połączeniowo, niezawodną wymianę danych.

Nagłówek TCP



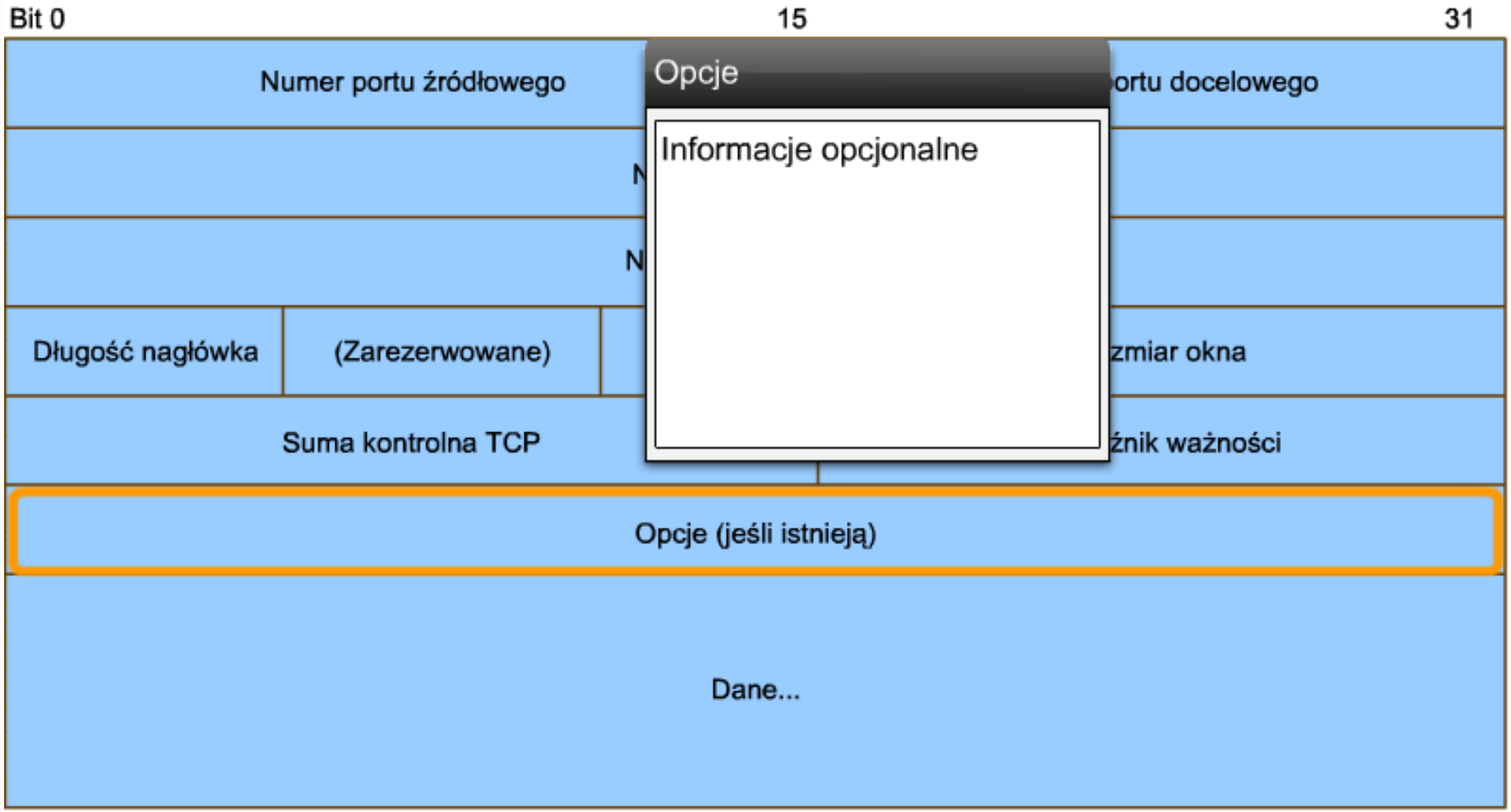
Pola w nagłówku TCP umożliwiają zorientowaną połączeniowo, niezawodną wymianę danych.

Nagłówek TCP



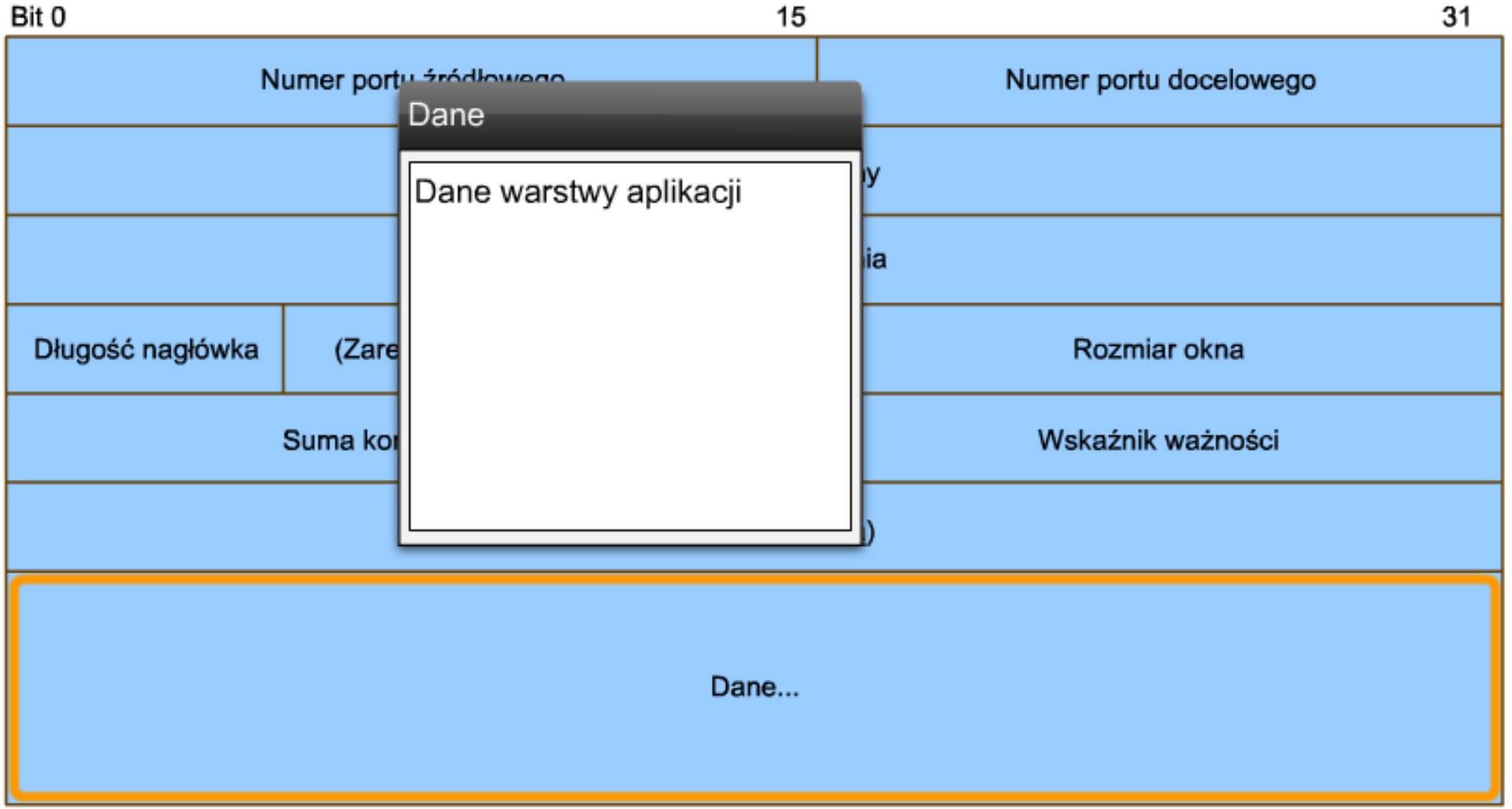
Pola w nagłówku TCP umożliwiają zorientowaną połączeniowo, niezawodną wymianę danych.

Nagłówek TCP



Pola w nagłówku TCP umożliwiają zorientowaną połączeniowo, niezawodną wymianę danych.

Nagłówek TCP



Pola w nagłówku TCP umożliwiają zorientowaną połączeniowo, niezawodną wymianę danych.

Nawiązywanie połączenia TCP

- Zanim dane mogą zostać przesłane w komunikacji między hostami przy użyciu protokołu TCP, musi być ustanowione między nimi połączenie. Aby nawiązać połączenie, host używa uzgadniania trójetapowego:

Nawiązywanie połączenia TCP

Trzy etapy podczas ustanawiania połączenia to:

- 1. Klient (host inicjujący połączenie) wysyła segment (SYN) zawierający początkową wartość synchronizacyjną (numer początkowy ISN).

Nawiązywanie połączenia TCP

Trzy etapy podczas ustanawiania połączenia to:

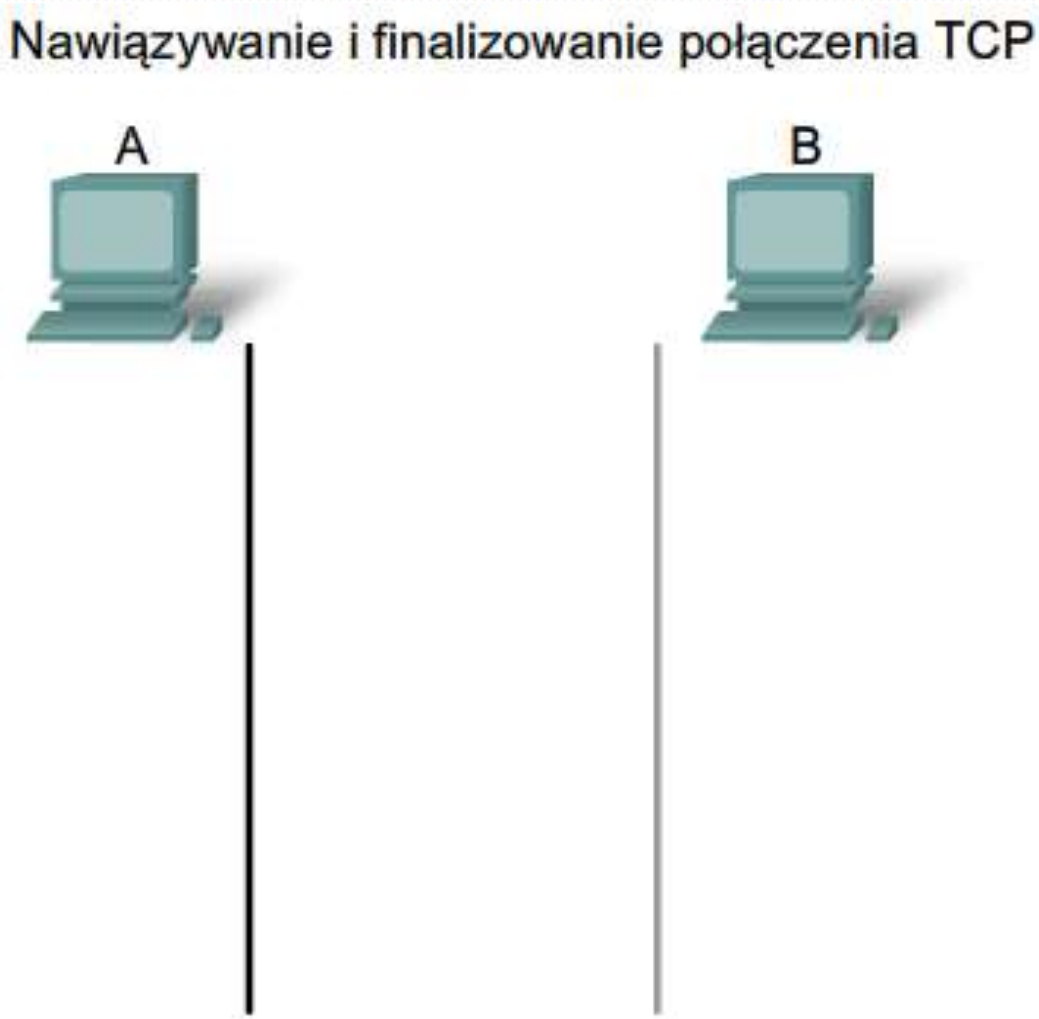
- 2. Serwer odpowiada, wysyłając segment zawierający wartość potwierdzenia (która jest równa przysłanej przez host wartości synchronizacyjnej powiększonej o 1) oraz wysyła swoją własną wartość synchronizacyjną (swój własny numer początkowy ISN). Wartość potwierdzenia jest zawsze większa od numeru sekwencyjnego ponieważ oznacza ona numer następnego spodziewanego bajtu (oktetu). Ta wartość potwierdzenia umożliwia klientowi dopasowanie odpowiedzi do numeru segmentu wysłanego do serwera.

Nawiązywanie połączenia TCP

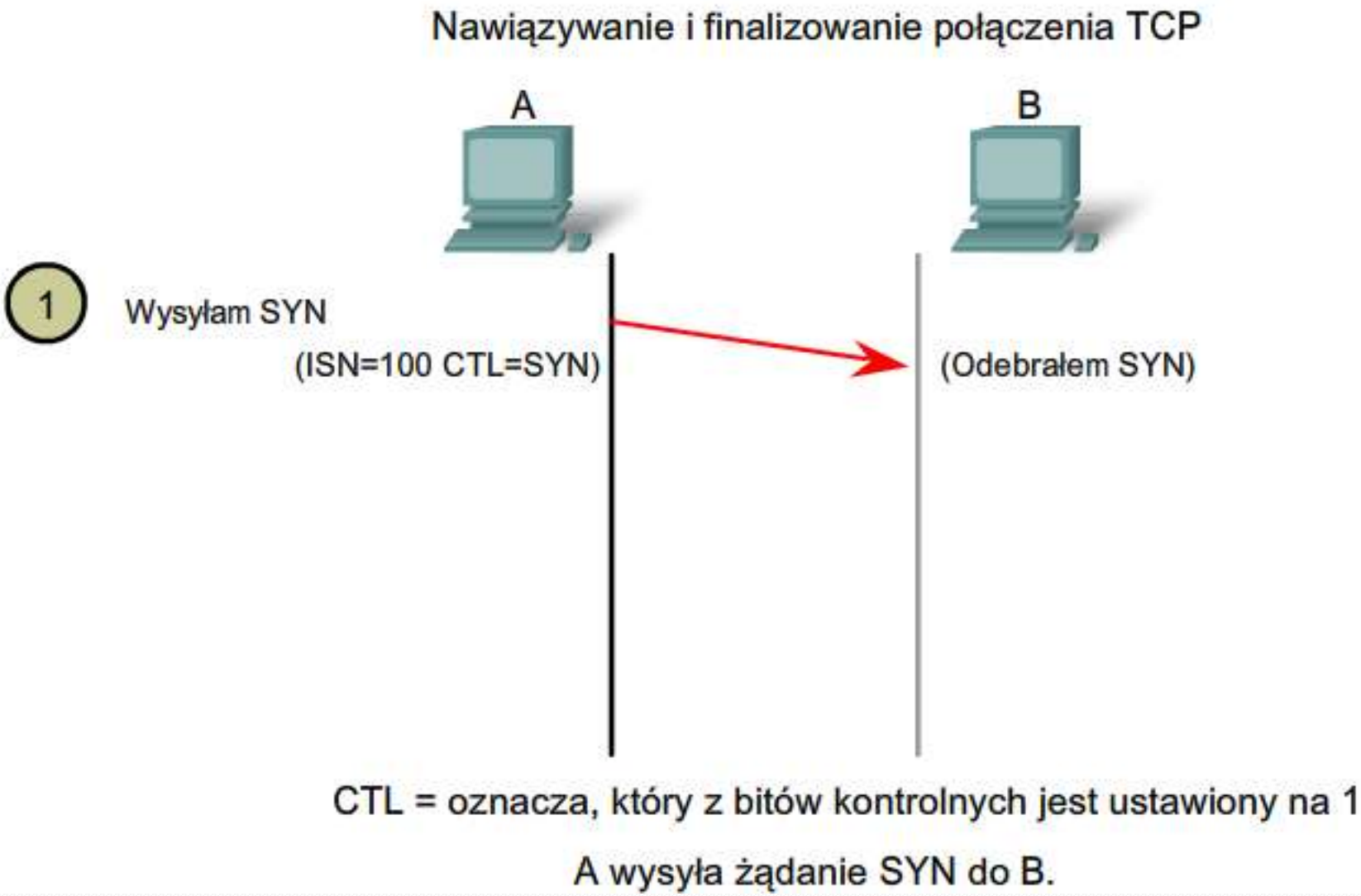
Trzy etapy podczas ustanawiania połączenia to:

- 3. Inicjujący połączenie klient odpowiada potwierdzeniem o wartości równej numerowi sekwencyjnemu serwera powiększonemu o 1. To kończy proces nawiązywania połączenia.

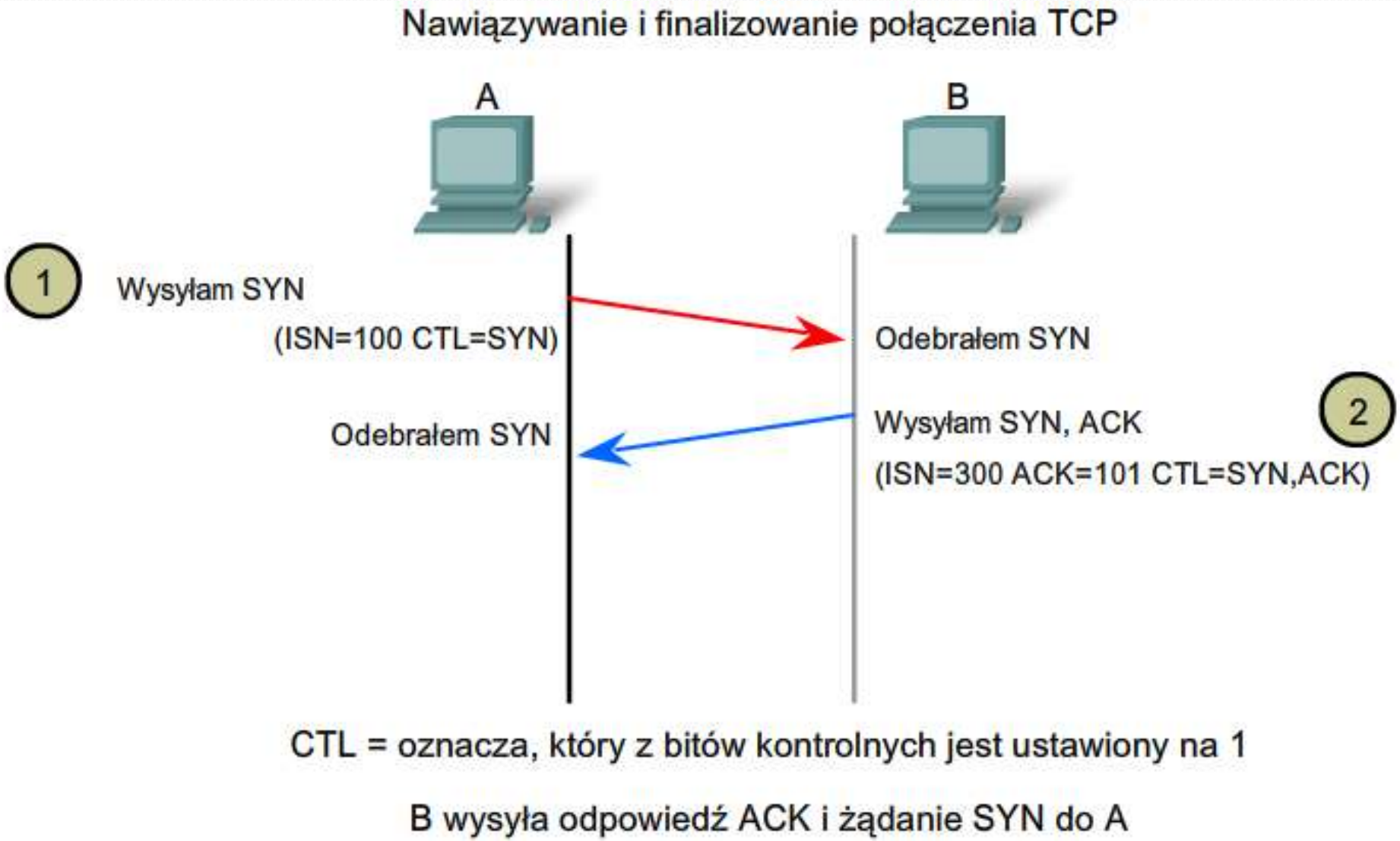
Nawiązywanie połączenia TCP



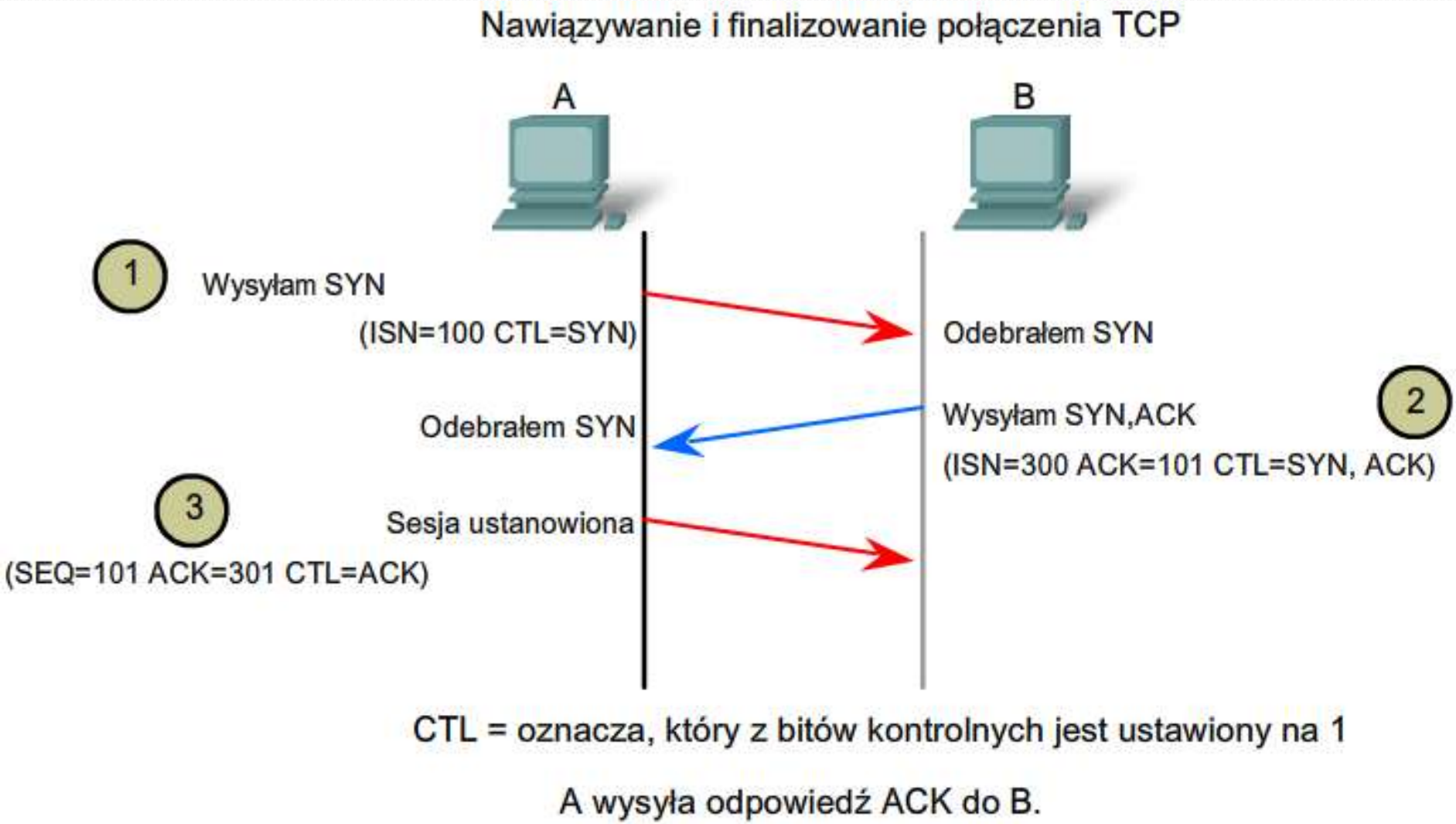
Nawiązywanie połączenia TCP



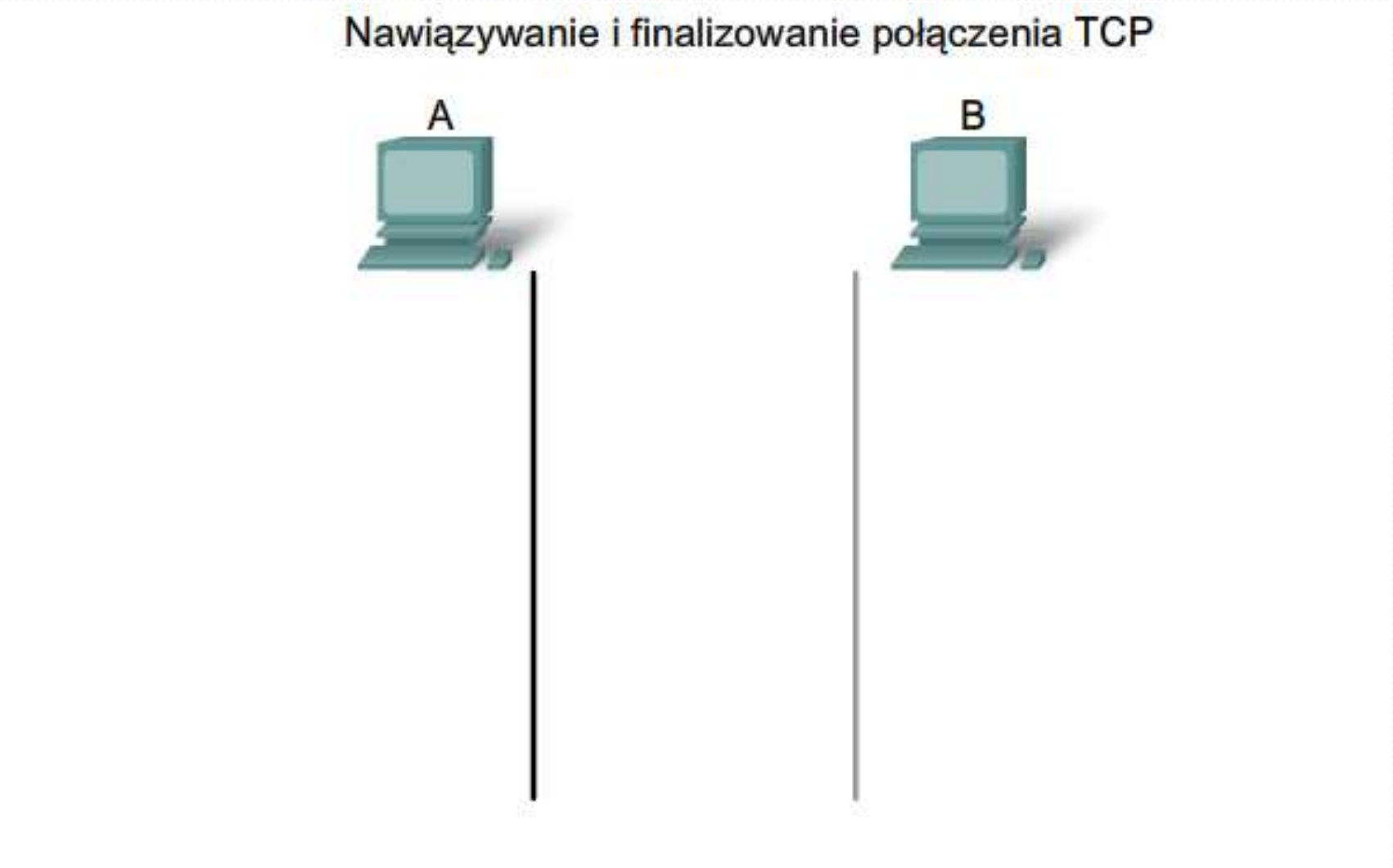
Nawiązywanie połączenia TCP



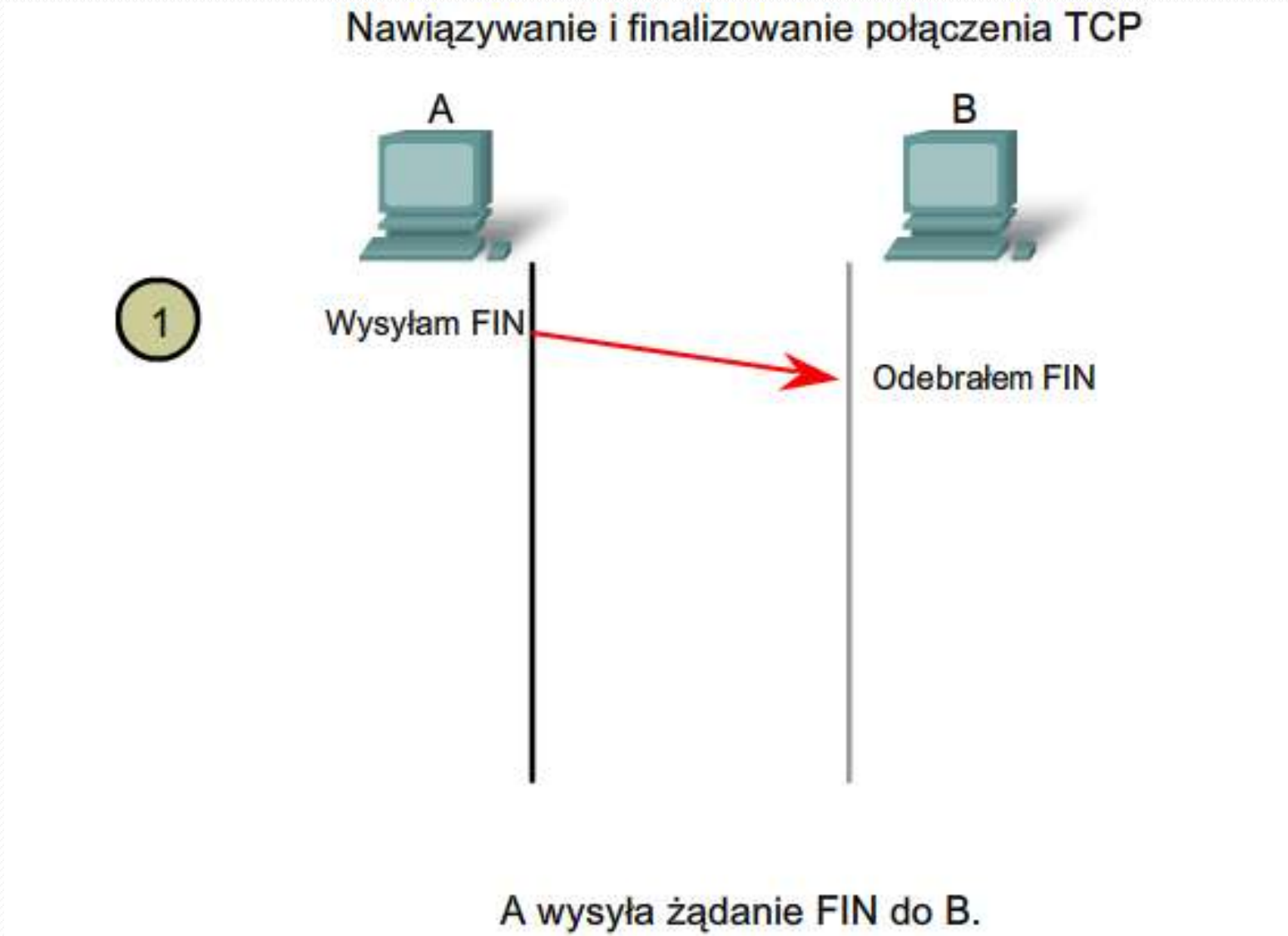
Nawiązywanie połączenia TCP



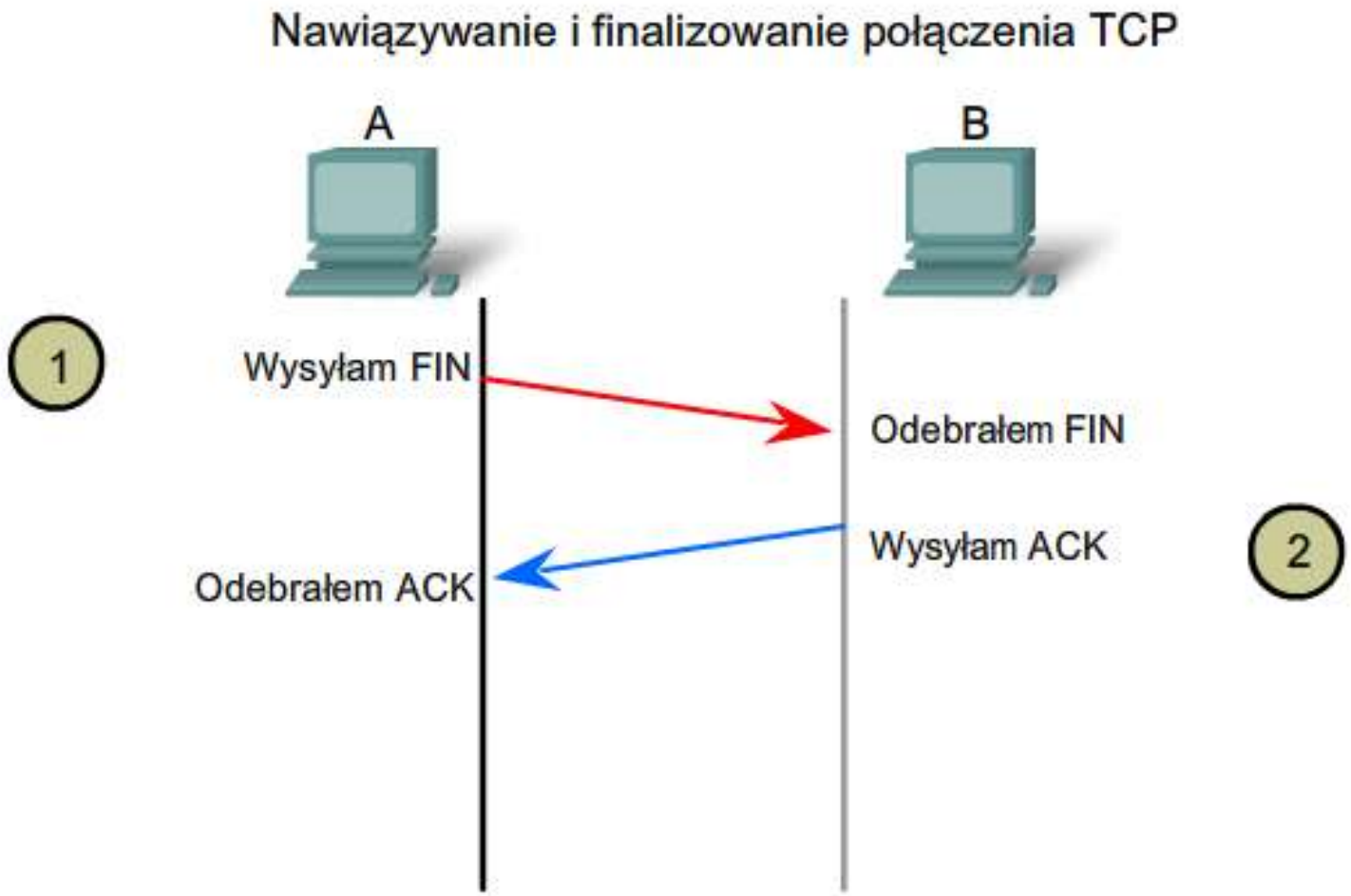
Zakończenie połączenia TCP



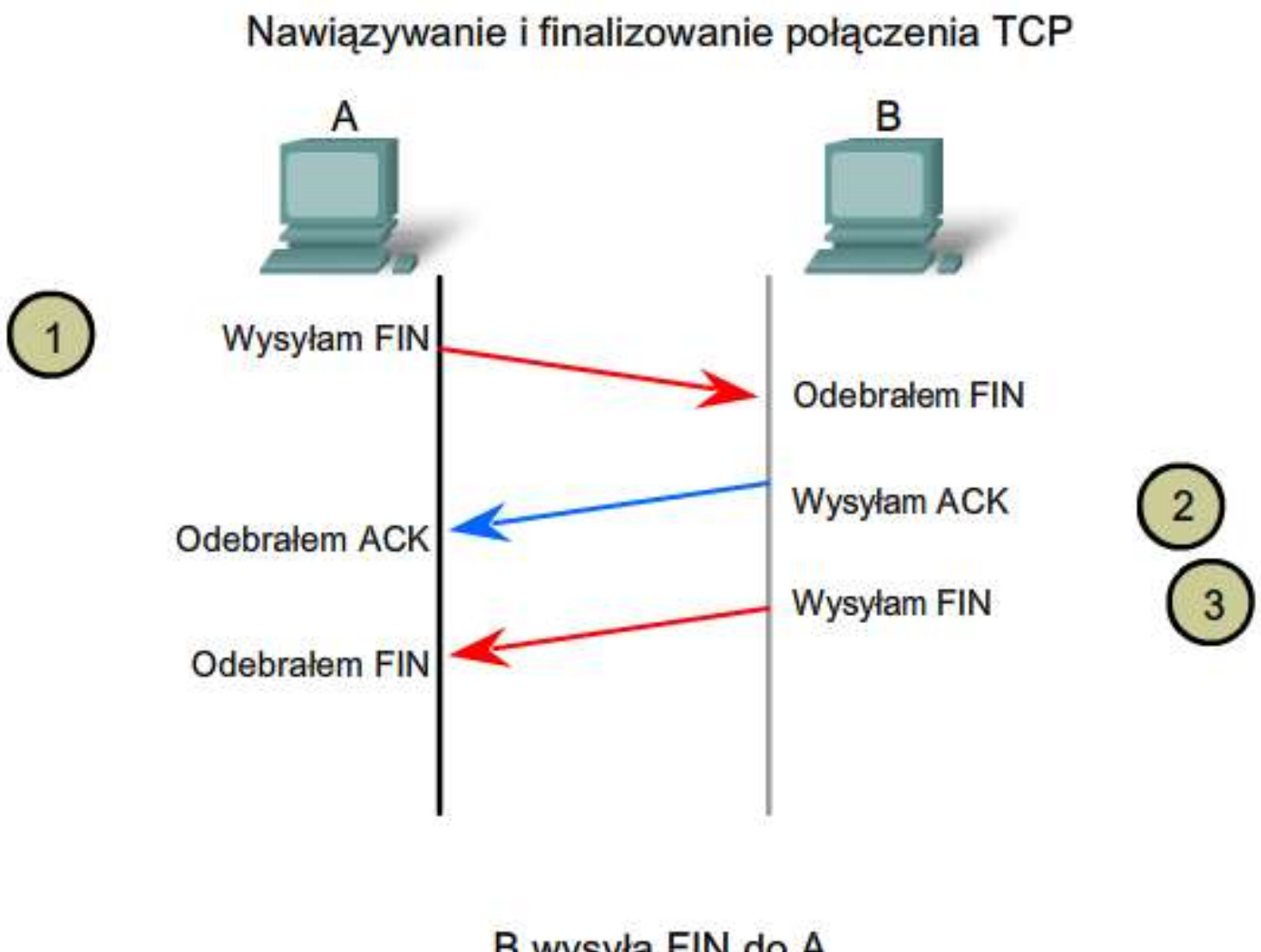
Zakończenie połączenia TCP



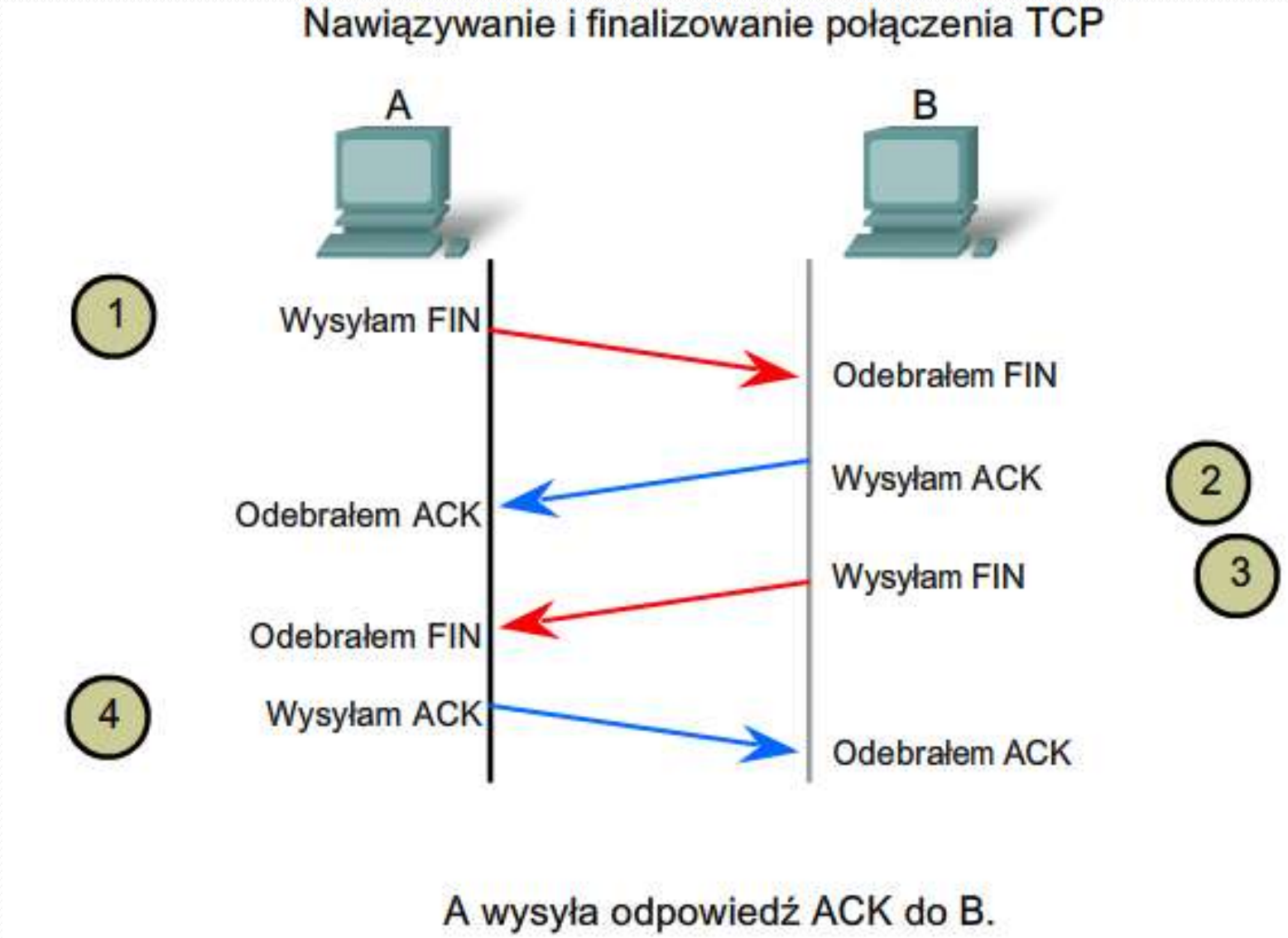
Zakończenie połączenia TCP



Zakończenie połączenia TCP



Zakończenie połączenia TCP

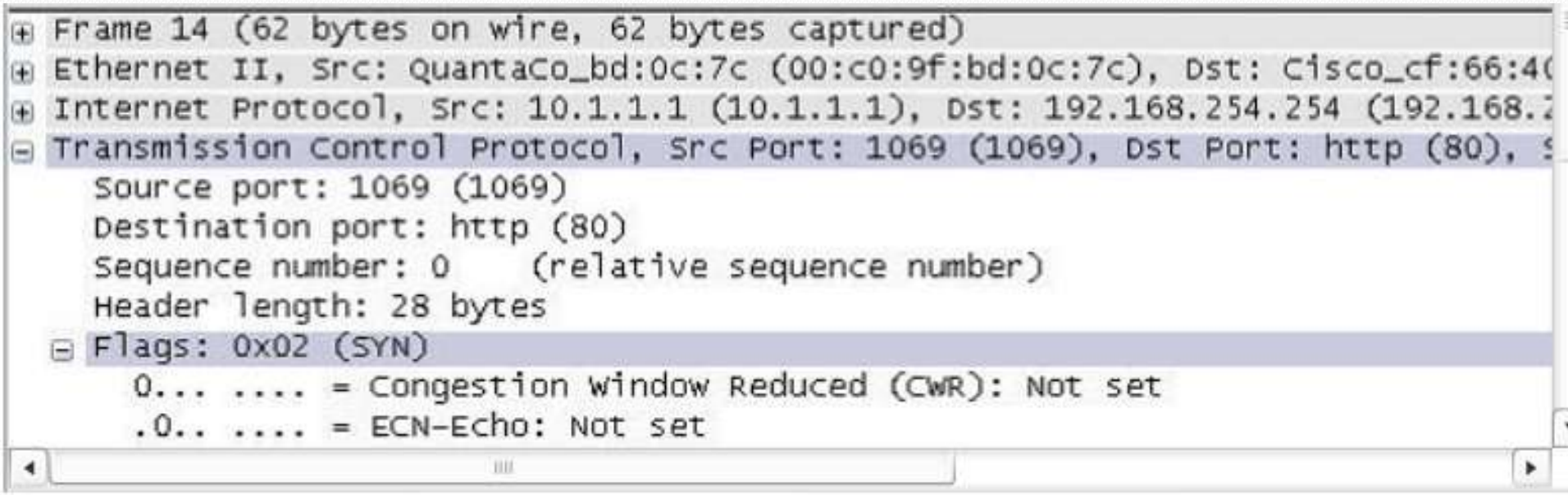


Flagi TCP

W segmencie TCP jest 6 jednobitowych pól (flag), które zawierają informacje kontrolne używane podczas zarządzania komunikacją TCP. Tymi polami są:

- URG- flaga, która wskazuje na ważność pola "Pilny”.
- ACK - flaga, która oznacza ważność pola "Potwierdzenie„.
- PSH - flaga, która oznacza wykorzystanie funkcji PUSH;
- RST - flaga, która używana jest do kończenia połączenia;
- SYN - flaga, która wskazuje na Synchronizację numerów sekwencyjnych;
- FIN - flaga, która wskazuje koniec danych od nadawcy.

Etap 1



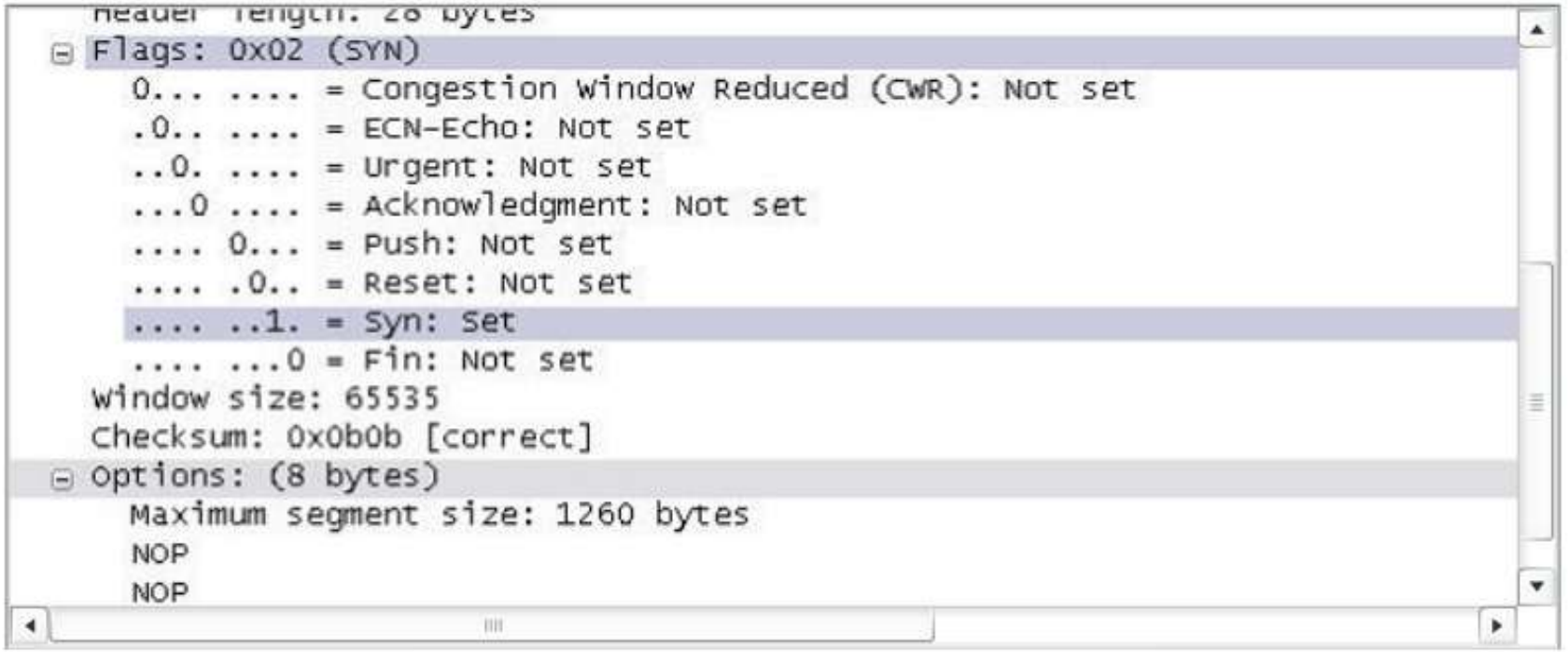
Analizator pakietów pokazuje w ramce 14 początkowe żądanie klienta do ustanowienia sesji

Segment TCP w tej ramce zawiera:

- Ustawiony bit synchronizacji (SYN) potwierdza zawartość początkowy numer sekwencyjny (ISN)
- Losowy numer sekwencyjny (dla ułatwienia na rysunku przedstawiono relatywną wartość 0)
- Losowo wygenerowany port źródłowy 1069
- Port docelowy z grupy dobrze znanych portów (well known ports) równy 80 (port HTTP) wskazujący na serwer WWW

Etap 1

Uzgadnianie trójetapowe TCP (SYN)



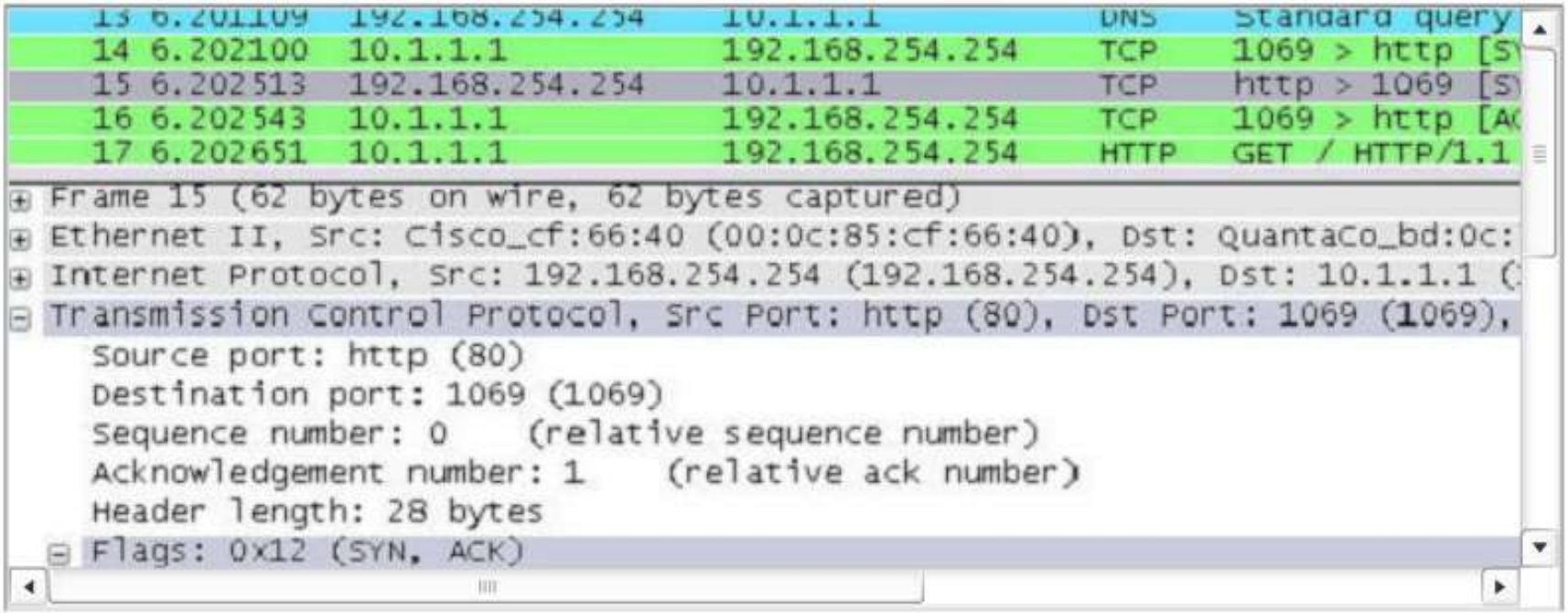
Analizator pakietów pokazuje w ramce 14 początkowe żądanie klienta do ustanowienia sesji

Segment TCP w tej ramce zawiera:

- Ustawiony bit synchronizacji (SYN) potwierdza zawartość początkowy numer sekwencyjny (ISN)
- Losowy numer sekwencyjny (dla ułatwienia na rysunku przedstawiono relatywną wartość 0)
- Losowo wygenerowany port źródłowy 1069
- Port docelowy z grupy dobrze znanych portów (well known ports) równy 80 (port HTTP) wskazujący na serwer WWW

Etap 2

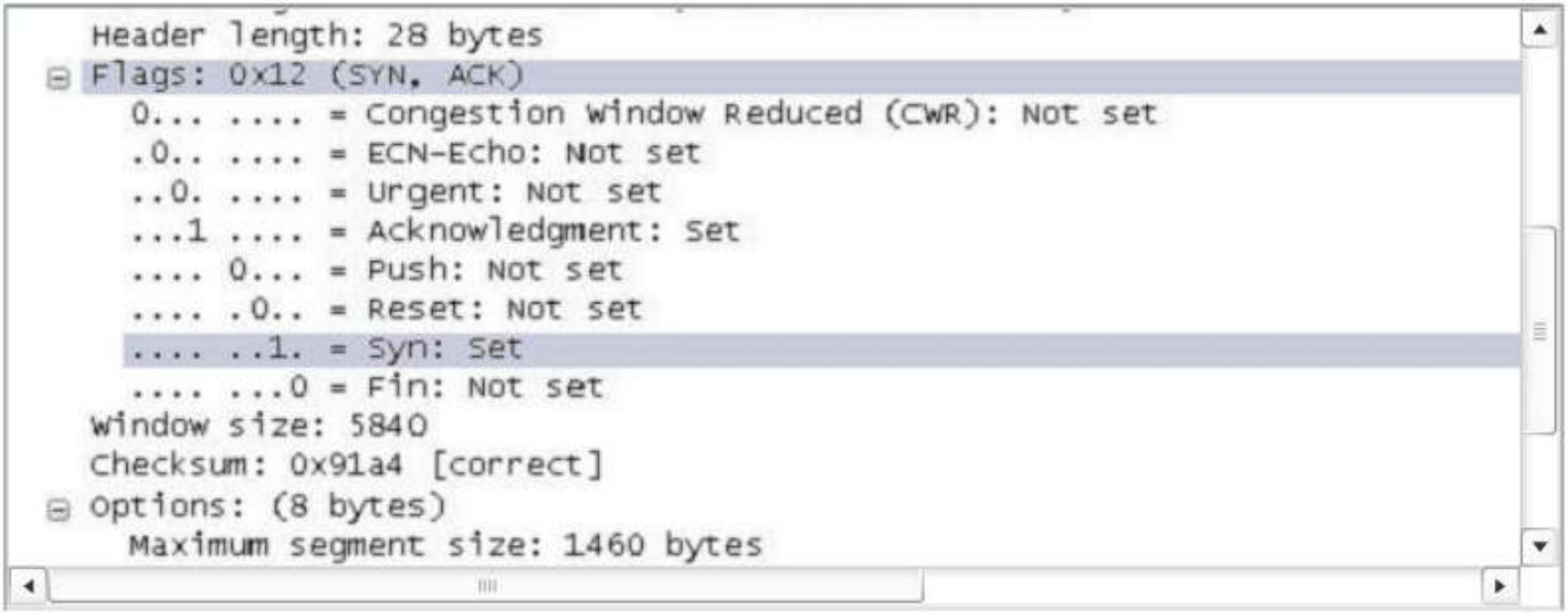
Uzgadnianie trójetapowe TCP (SYN, ACK)



Analizator pakietów pokazuje odpowiedź serwera w ramce nr 15

- Flaga potwierdzenia (ACK) wskazuje, że segment powinien zawierać prawidłowy numer potwierdzenia
- Numer potwierdzenia (ACK number) odpowiedzi, odniesiony relatywnie do startowego numeru sekwencyjnego, wynoszący 1
- Flaga synchronizacji (SYN) ustawiona na 1 wskazuje, że segment zawiera początkowy numer sekwencyjny ze strony serwera
- Port docelowy, którego wartość 1069 odpowiada portowi źródłowemu klienta
- Port źródłowy, którego wartość 80 (HTTP) wskazuje, że odpowiedź pochodzi od serwera WWW

Etap 2

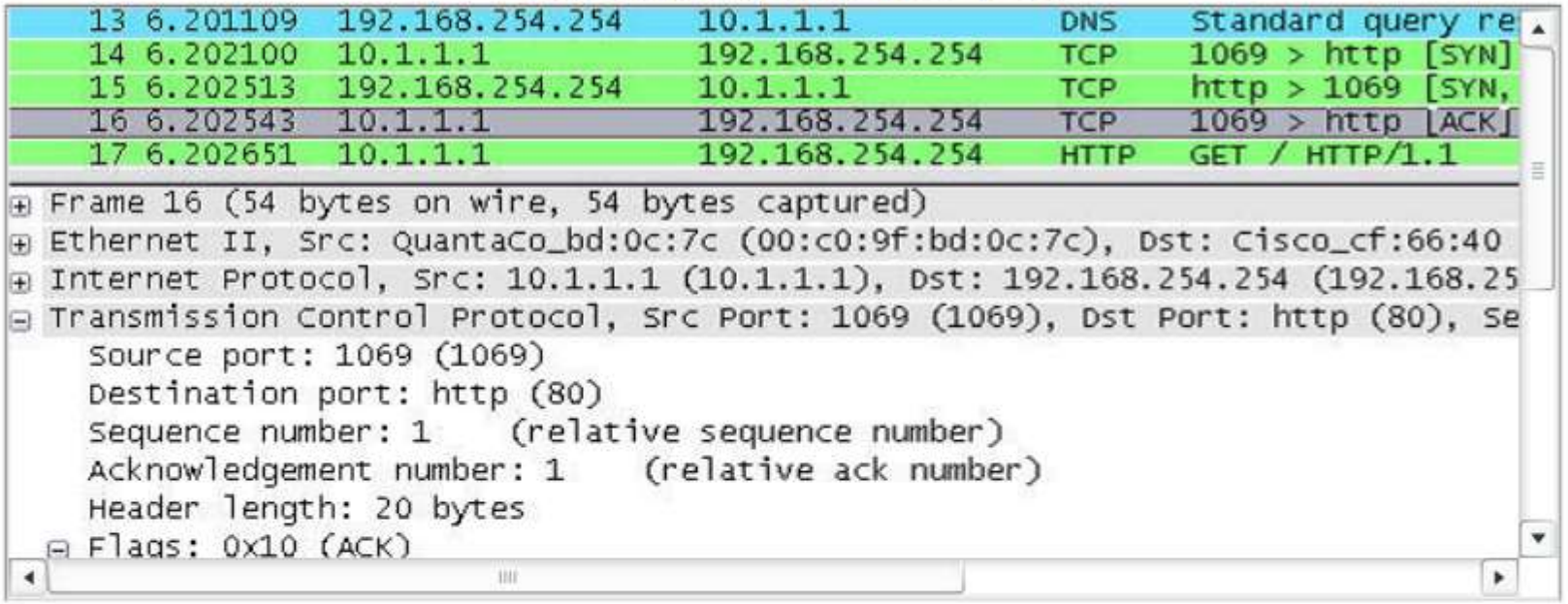


Analizator pakietów pokazuje odpowiedź serwera w ramce nr 15

- Flaga potwierdzenia (ACK) wskazuje, że segment powinien zawierać prawidłowy numer potwierdzenia
- Numer potwierdzenia (ACK number) odpowiedzi, odniesiony relatywnie do startowego numeru sekwencyjnego, wynoszący 1
- Flaga synchronizacji (SYN) ustawiona na 1 wskazuje, że segment zawiera początkowy numer sekwencyjny ze strony serwera
- Port docelowy, którego wartość 1069 odpowiada portowi źródłowemu klienta
- Port źródłowy, którego wartość 80 (HTTP) wskazuje, że odpowiedź pochodzi od serwera WWW

Etap 3

Uzgadnianie trójetapowe TCP (ACK)



Analizator pakietów pokazuje odpowiedź klienta w ramce numer 16

Segment TCP w tej ramce zawiera:

- Flagę ACK wskazującą, że segment powinien zawierać prawidłowy numer potwierdzenia
- Numer potwierdzenia (ACK number) odpowiedzi, odniesiony relatywnie do startowego numeru sekwencyjnego, wynoszący 1
- Numer portu źródłowego 1069
- Numer portu docelowego 80 (HTTP) wskazujący na usługę WWW (httpd)

Etap 3

Uzgadnianie trójetapowe TCP (ACK)

```
⊟ Flags: 0x10 (ACK)
    0... .... = Congestion Window Reduced (CWR): Not set
    .0.. .... = ECN-Echo: Not set
    ..0. .... = Urgent: Not set
    ...1 .... = Acknowledgment: Set
    .... 0... = Push: Not set
    .... .0.. = Reset: Not set
    .... ..0. = Syn: Not set
    .... ...0 = Fin: Not set
  Window size: 65535
  Checksum: 0xd538 [correct]
⊟ [SEQ/ACK analysis]
    [This is an ACK to the segment in frame: 15]
    [The RTT to ACK the segment was: 0.000030000 seconds]
```

Analizator pakietów pokazuje odpowiedź klienta w ramce numer 16

Segment TCP w tej ramce zawiera:

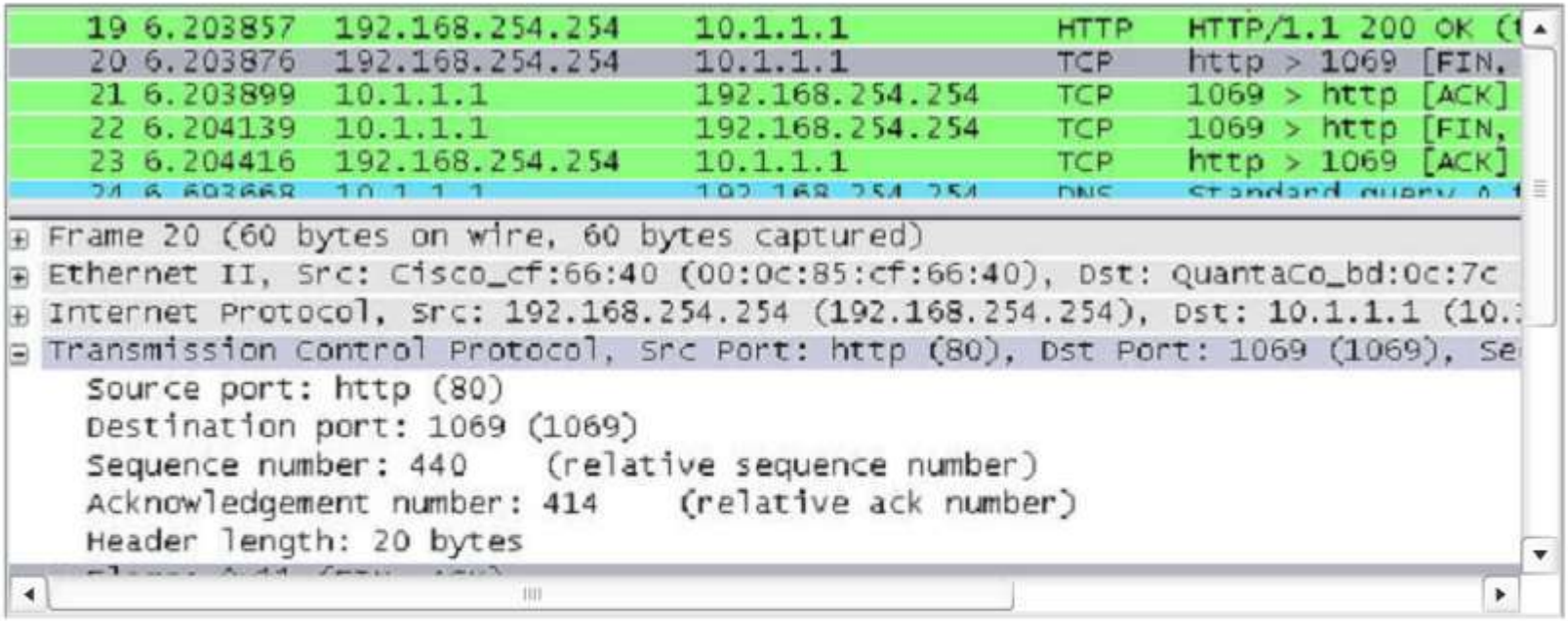
- Flagę ACK wskazującą, że segment powinien zawierać prawidłowy numer potwierdzenia
- Numer potwierdzenia (ACK number) odpowiedzi, odniesiony relatywnie do startowego numeru sekwencyjnego, wynoszący 1
- Numer portu źródłowego 1069
- Numer portu docelowego 80 (HTTP) wskazujący na usługę WWW (httpd)

Zakończenie sesji

- Aby zakończyć połączenie, w nagłówku segmentu musi być ustawiona flaga FIN. W celu zakończenia każdej jednokierunkowej sesji TCP, używane jest podwójne uzgadnianie, składające się z segmentów FIN oraz ACK. Tak więc, aby zakończyć pojedynczą konwersację TCP, dla obu sesji (klient -> serwer oraz serwer -> klient) potrzebne są dwa takie uzgadniania.

Zakończenie sesji

Zakończenie sesji TCP (FIN)



Analizator pakietów pokazuje szczegóły ramki 20, żądanie TCP FIN.

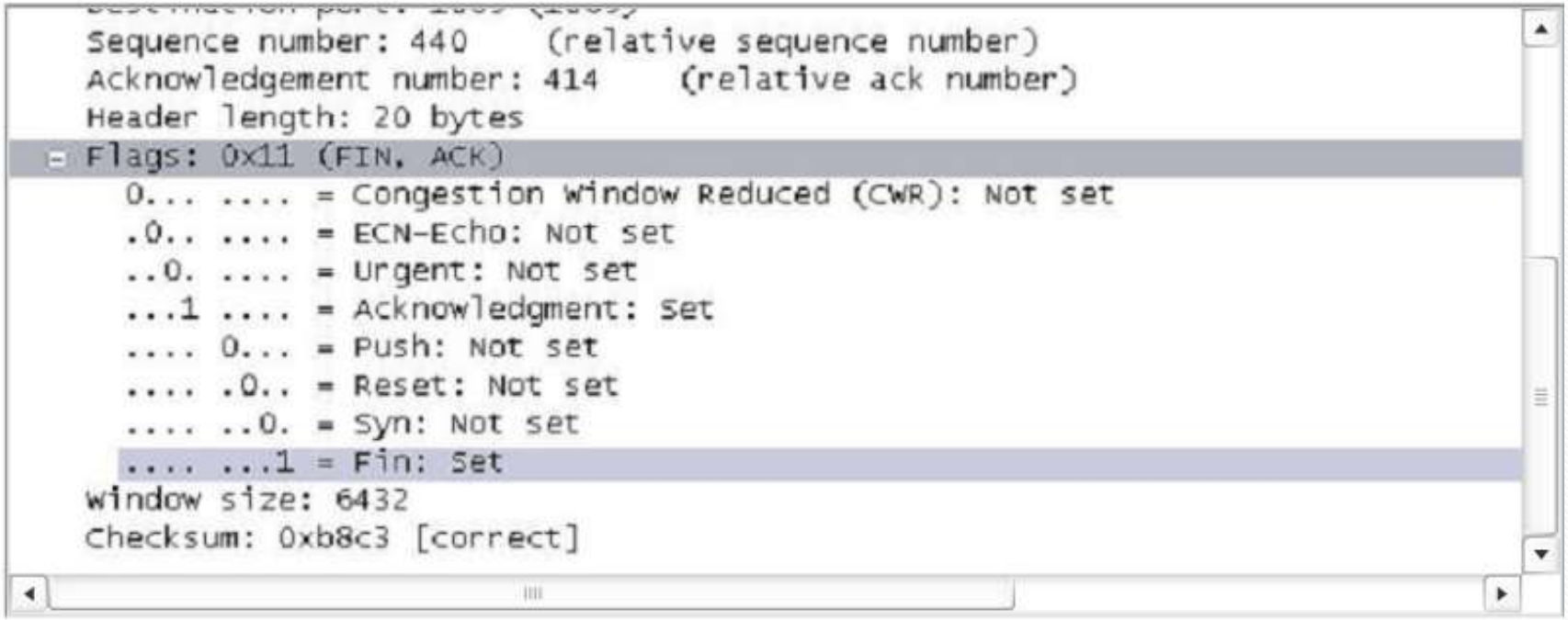
Porty docelowe i źródłowe
Zawartość i wartości pola nagłówka

FIN

ACK

Zakończenie sesji

Zakończenie sesji TCP (FIN)



Analizator pakietów pokazuje szczegóły ramki 20, żądanie TCP FIN.

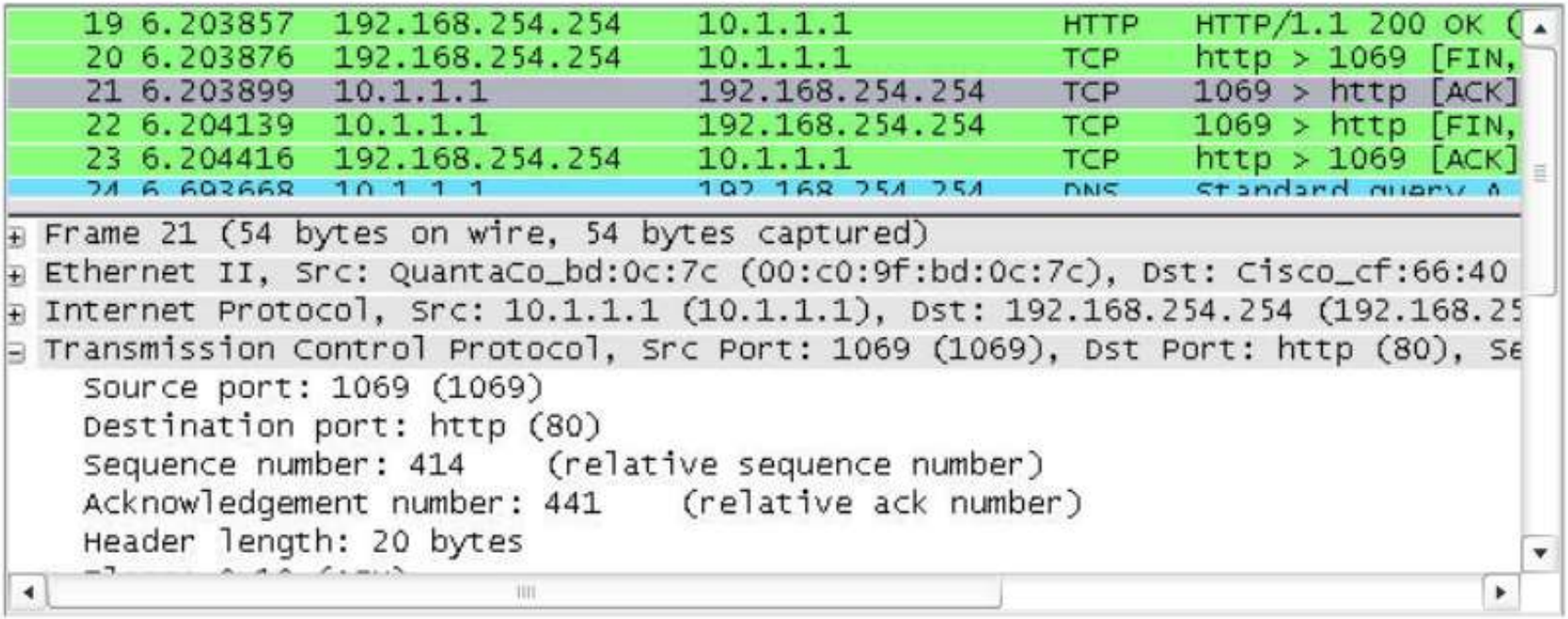
Porty docelowe i źródłowe
Zawartość i wartości pola nagłówka

FIN

ACK

Zakończenie sesji

Zakończenie sesji TCP (ACK)



Analizator pakietów pokazuje szczegóły ramki 21, żądanie TCP ACK.

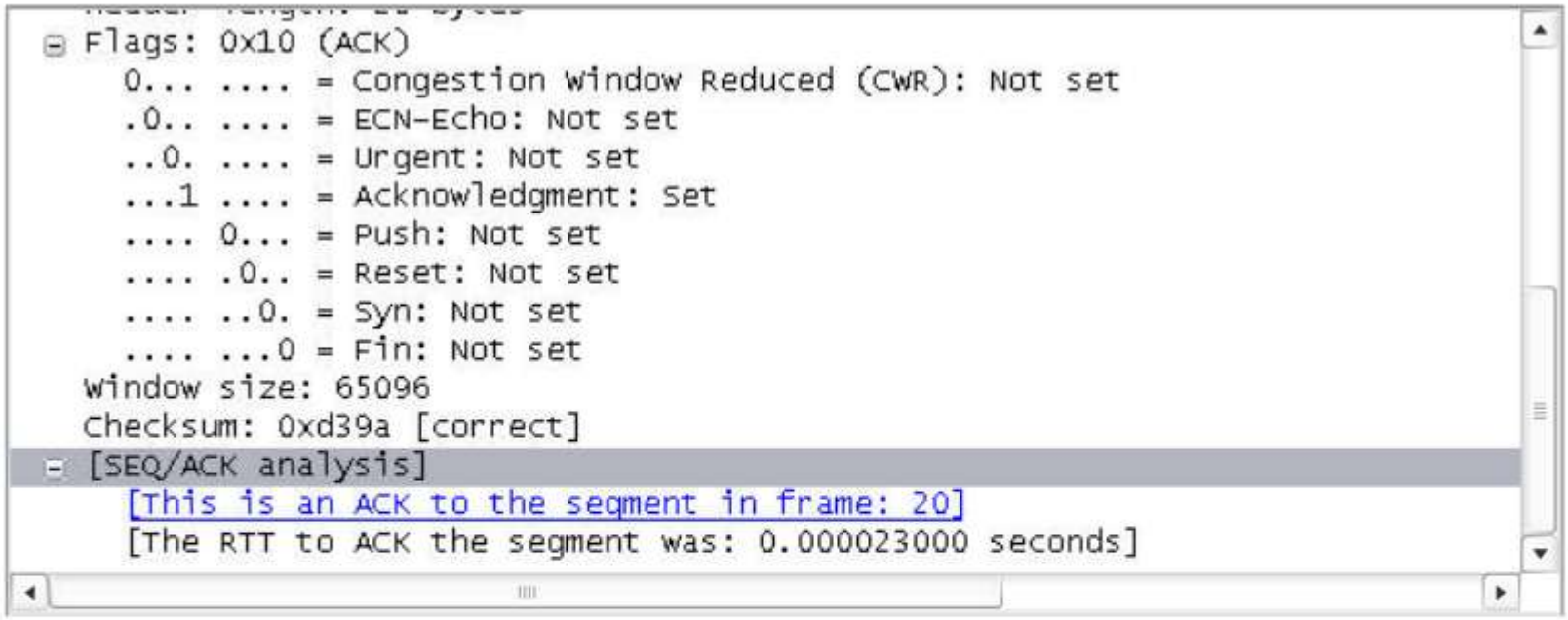
Porty docelowe i źródłowe
Zawartość i wartości pola nagłówka

FIN

ACK

Zakończenie sesji

Zakończenie sesji TCP (ACK)



Analizator pakietów pokazuje szczegóły ramki 21, żądanie TCP ACK.

Porty docelowe i źródłowe
Zawartość i wartości pola nagłówka

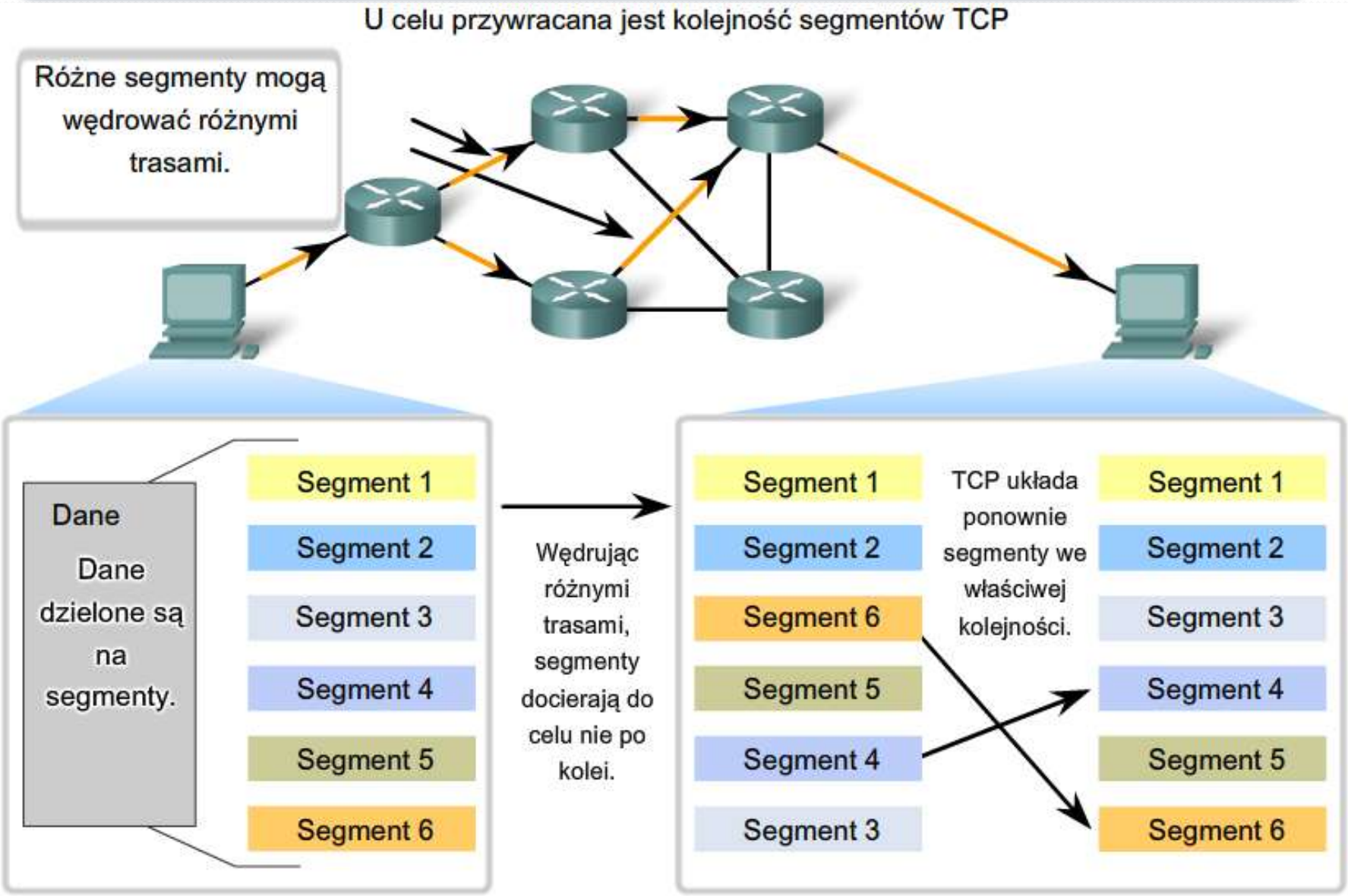
FIN

ACK

Scalanie segmentów TCP

- Aby odbiorca właściwie zrozumiał nadawaną wiadomość, dane w odebranych segmentach są scalane w kolejności w jakiej były wysyłane. Do tego celu używane są numery sekwencyjne znajdujące się w nagłówku każdego segmentu.

Scalanie segmentów TCP



Scalanie segmentów TCP

- Odbierający proces TCP umieszcza dane z segmentu w buforze odbiorczym. Segmenty są ustawiane w kolejności zgodnej z numerami sekwencyjnymi i następnie przekazywane do warstwy aplikacji. Jeśli odebrane zostaną segmenty, których numery sekwencyjne nie zachowują ciągłości, są one zatrzymywane (w buforze) w celu późniejszego przetworzenia. Po otrzymaniu brakujących segmentów dane zostają niezwłocznie przetworzone.

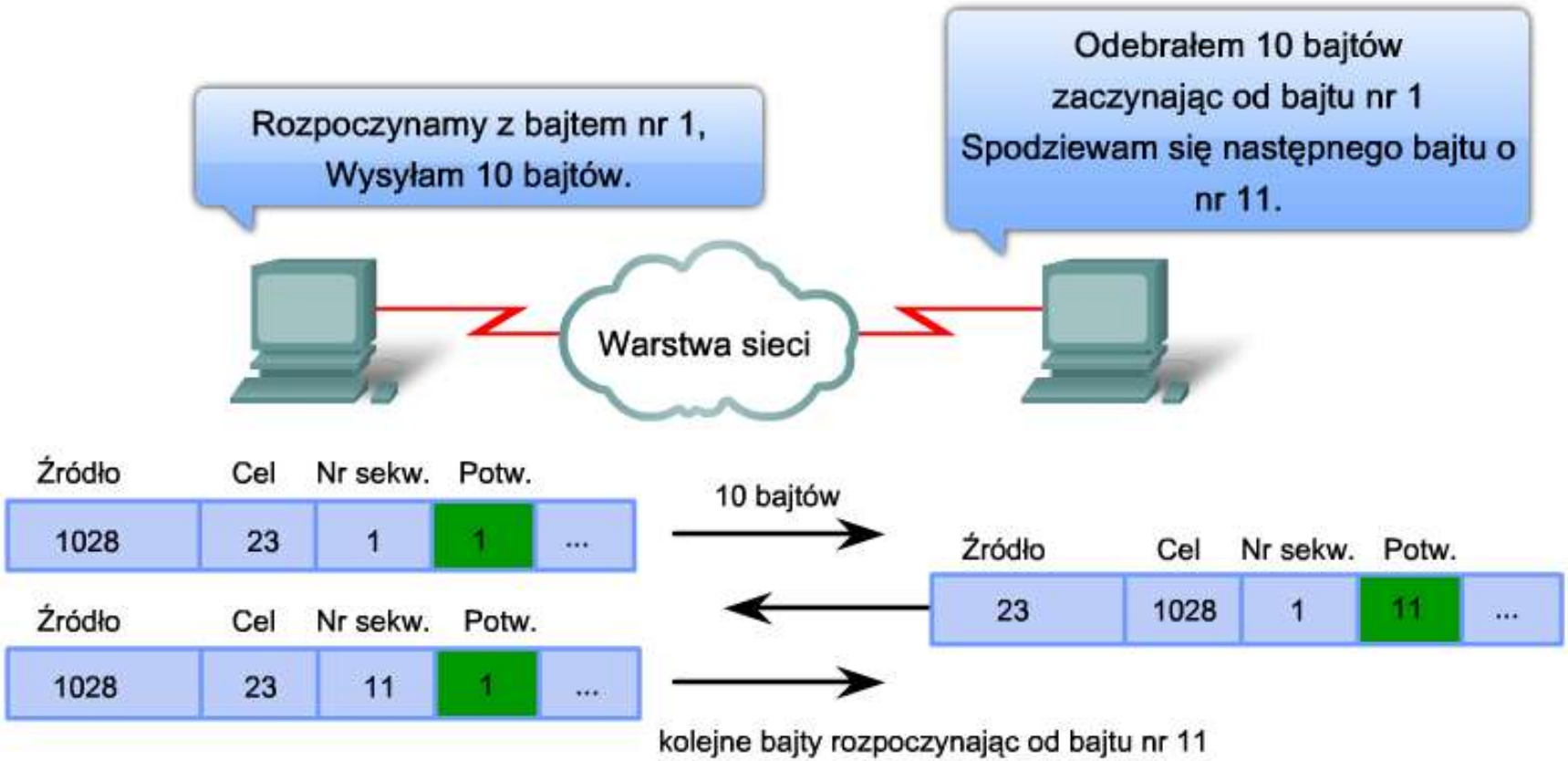
Potwierdzenie TCP

- Odbiorca po prawej otrzymuje segment (warstwa 4) i ustala, że numer sekwencyjny wynosi 1, a segment zawiera 10 bajtów danych. Odsyła więc segment do hosta po lewej z potwierdzeniem otrzymania tych danych. W tym odsyłanym segmencie wartość potwierdzenia ustawiana jest na 11, aby wskazać numer następnego bajtu, który host po prawej spodziewa się otrzymać w tej sesji.

Potwierdzenie TCP

Potwierdzenia segmentów TCP

Port źródłowy	Port docelowy	Numer sekwencyjny	Numer potwierdzenia	...

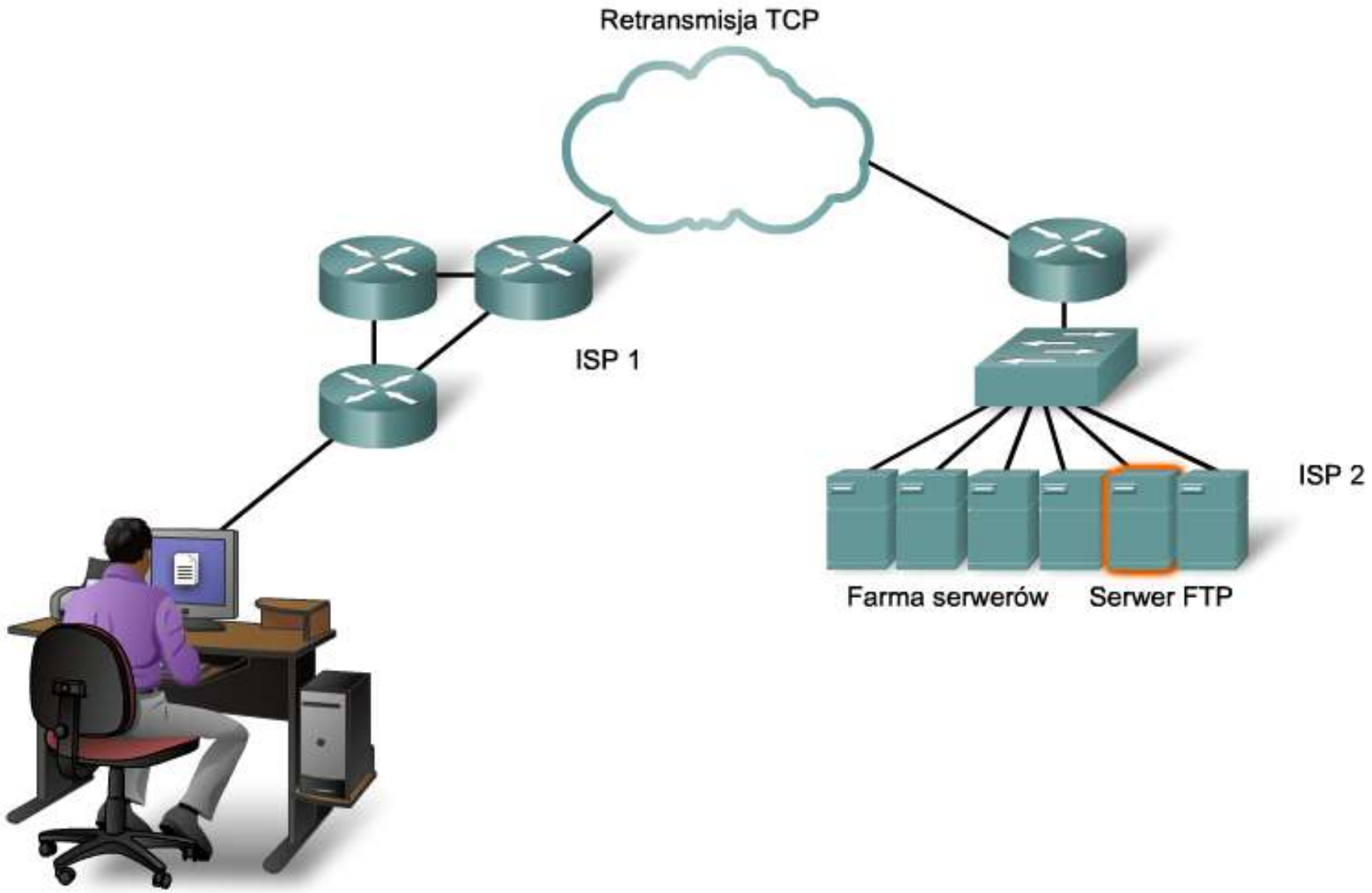


Zarządzanie utraconymi segmentami

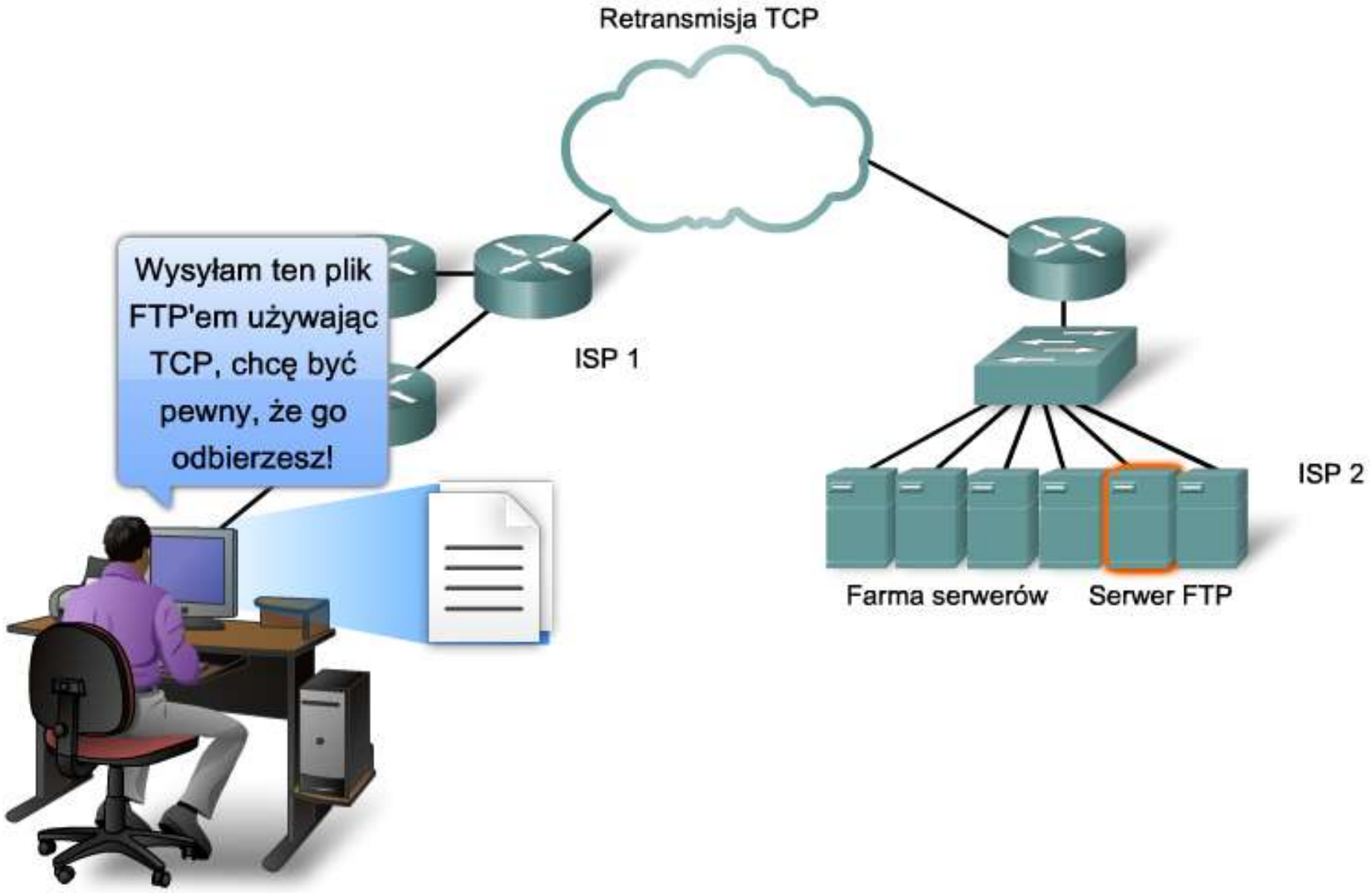
- Usługa TCP na hoście docelowym zwykle potwierdza ciągłą partię danych. Jeśli brakuje jednego lub więcej segmentów, to potwierdzane są tylko te dane w segmencie, które poprzedzają pierwszy brakujący fragment.

- Na przykład, jeśli adresat otrzymał segmenty z numerami sekwencyjnymi od 1500 do 3000 i od 3400 do 3500, to numer potwierdzenia wyniesie 3001. Oczywiście dlatego, że segmnty z numerami sekwencyjnymi od 3001 do 3399 nie zostały odebrane.

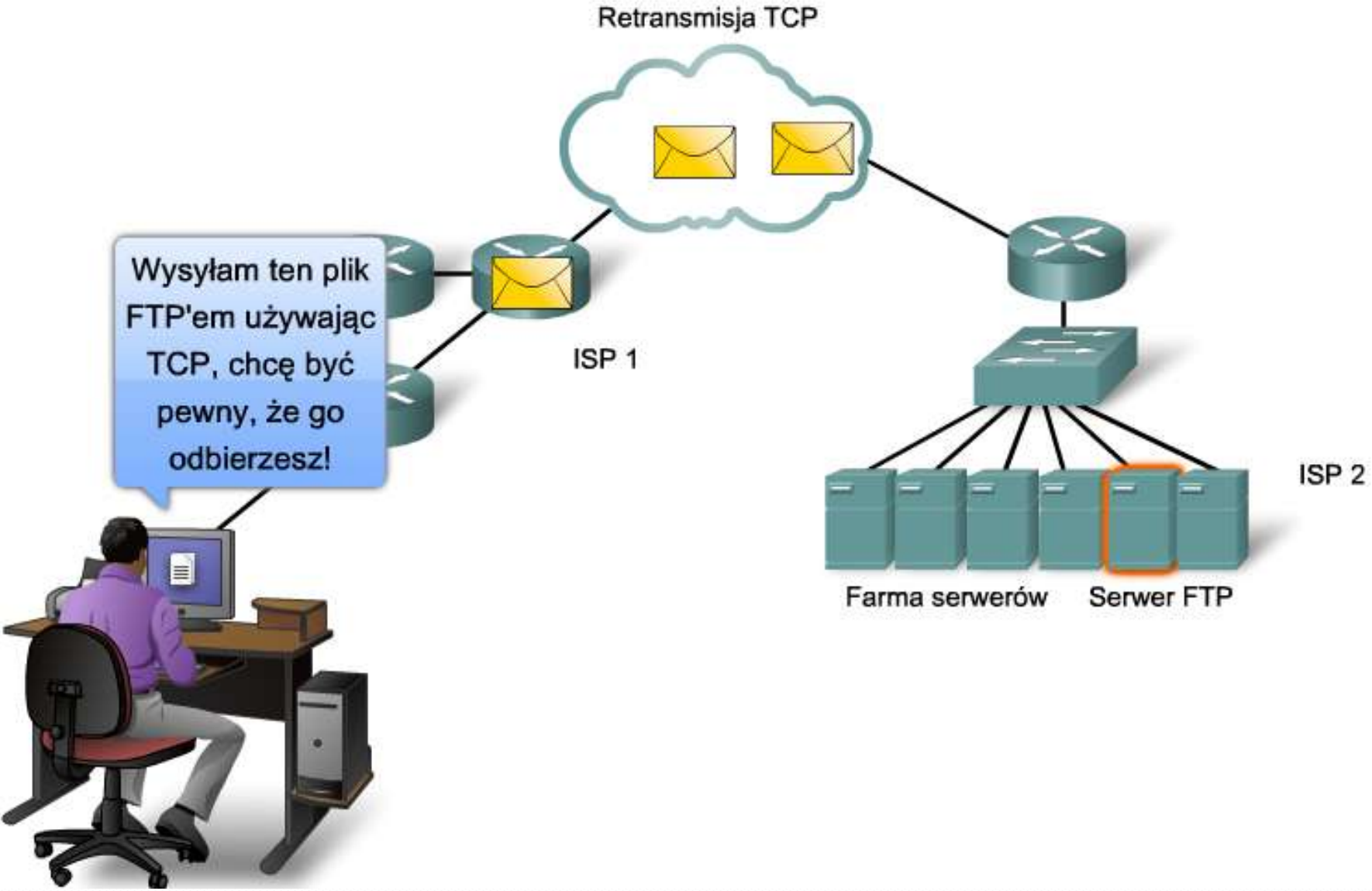
Zarządzanie utraconymi segmentami



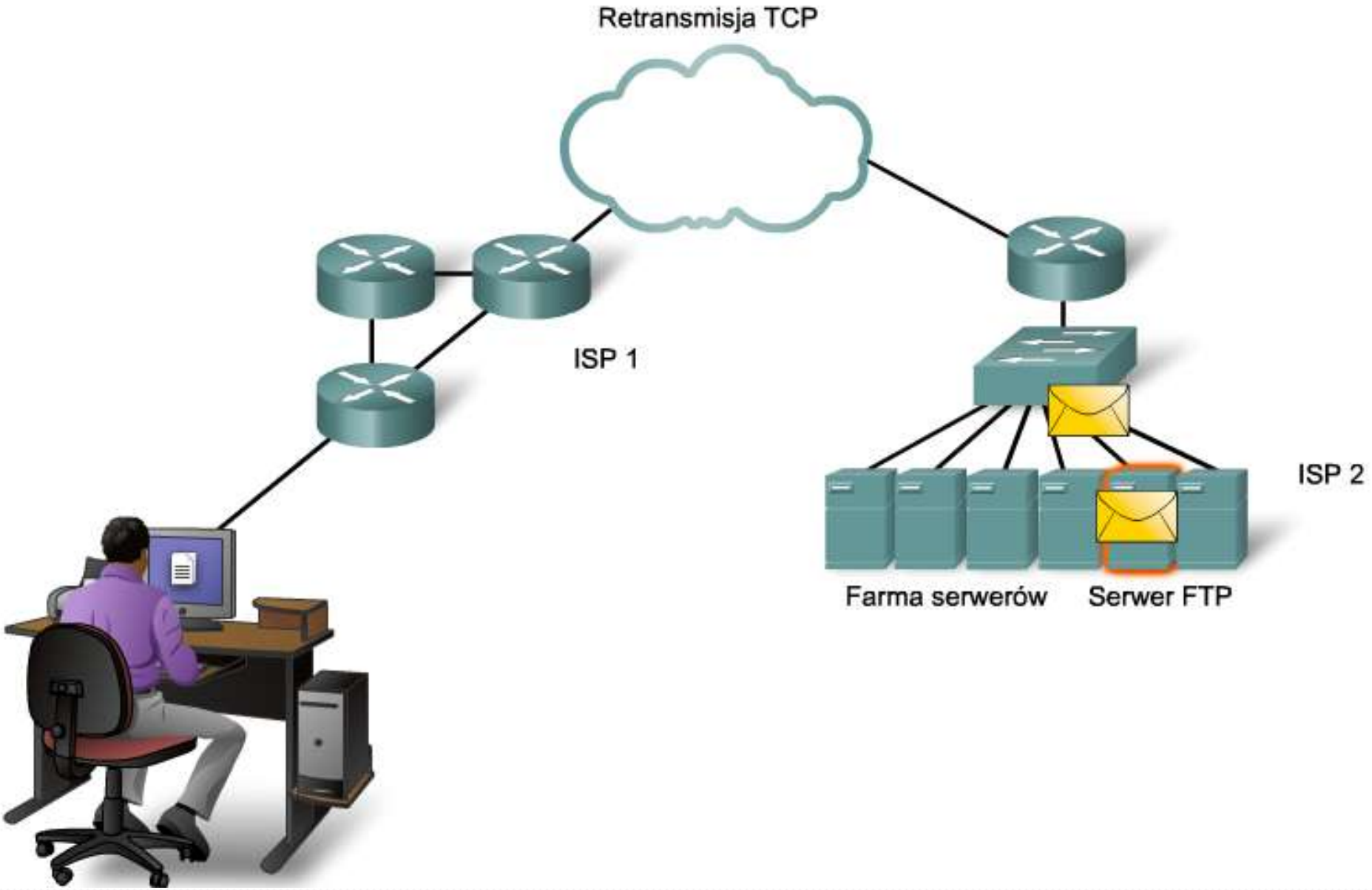
Zarządzanie utraconymi segmentami



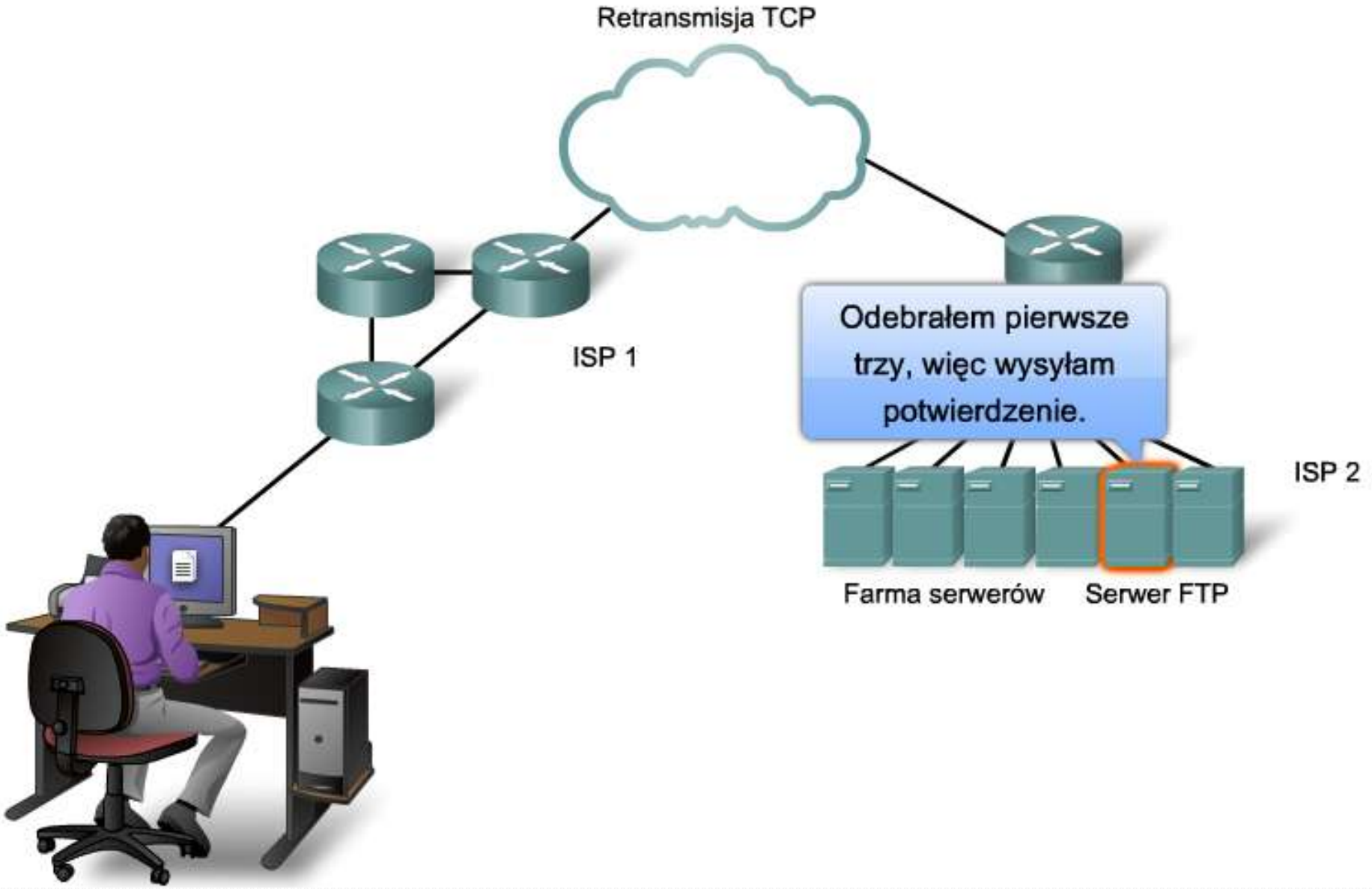
Zarządzanie utraconymi segmentami



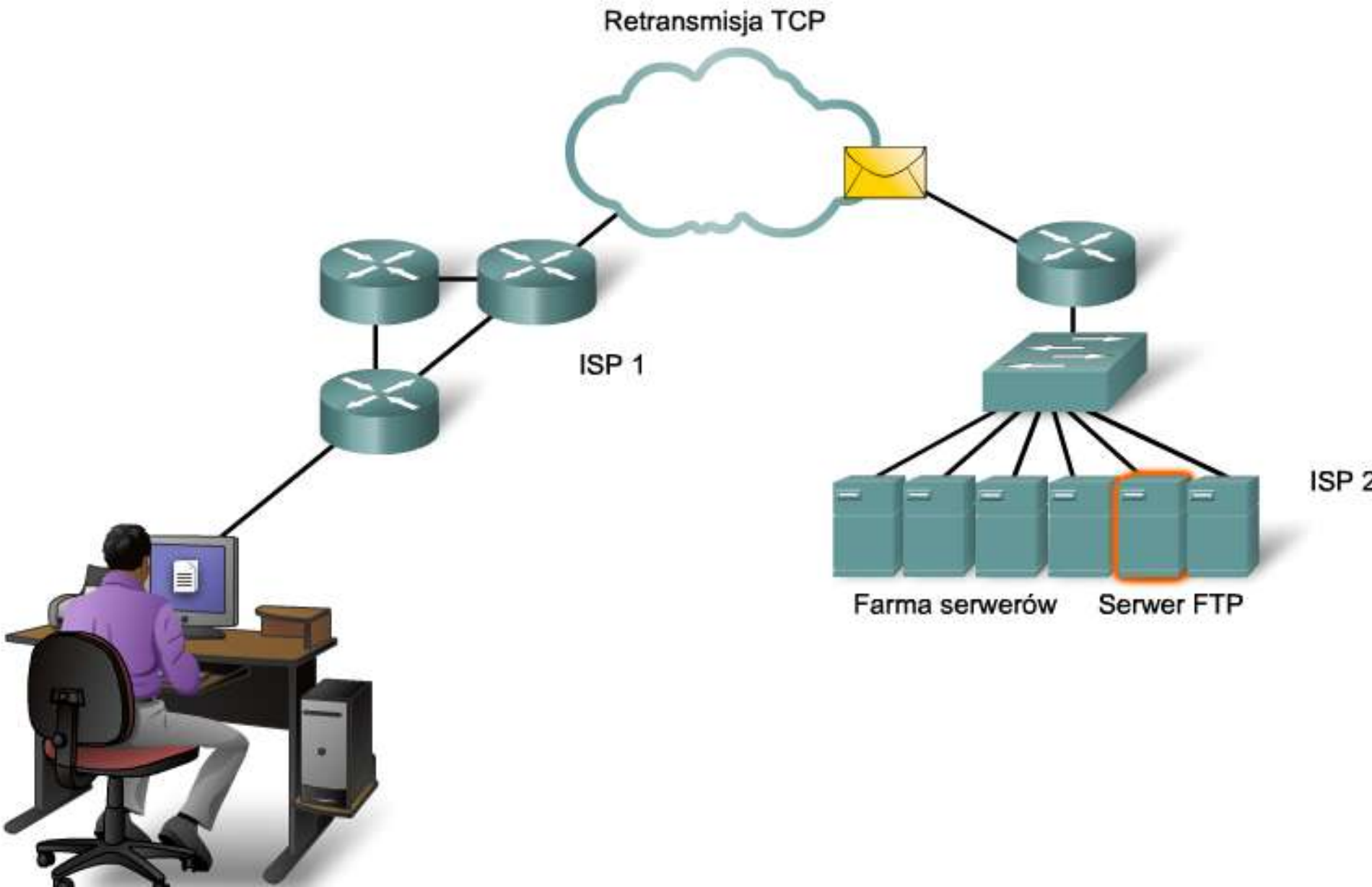
Zarządzanie utraconymi segmentami



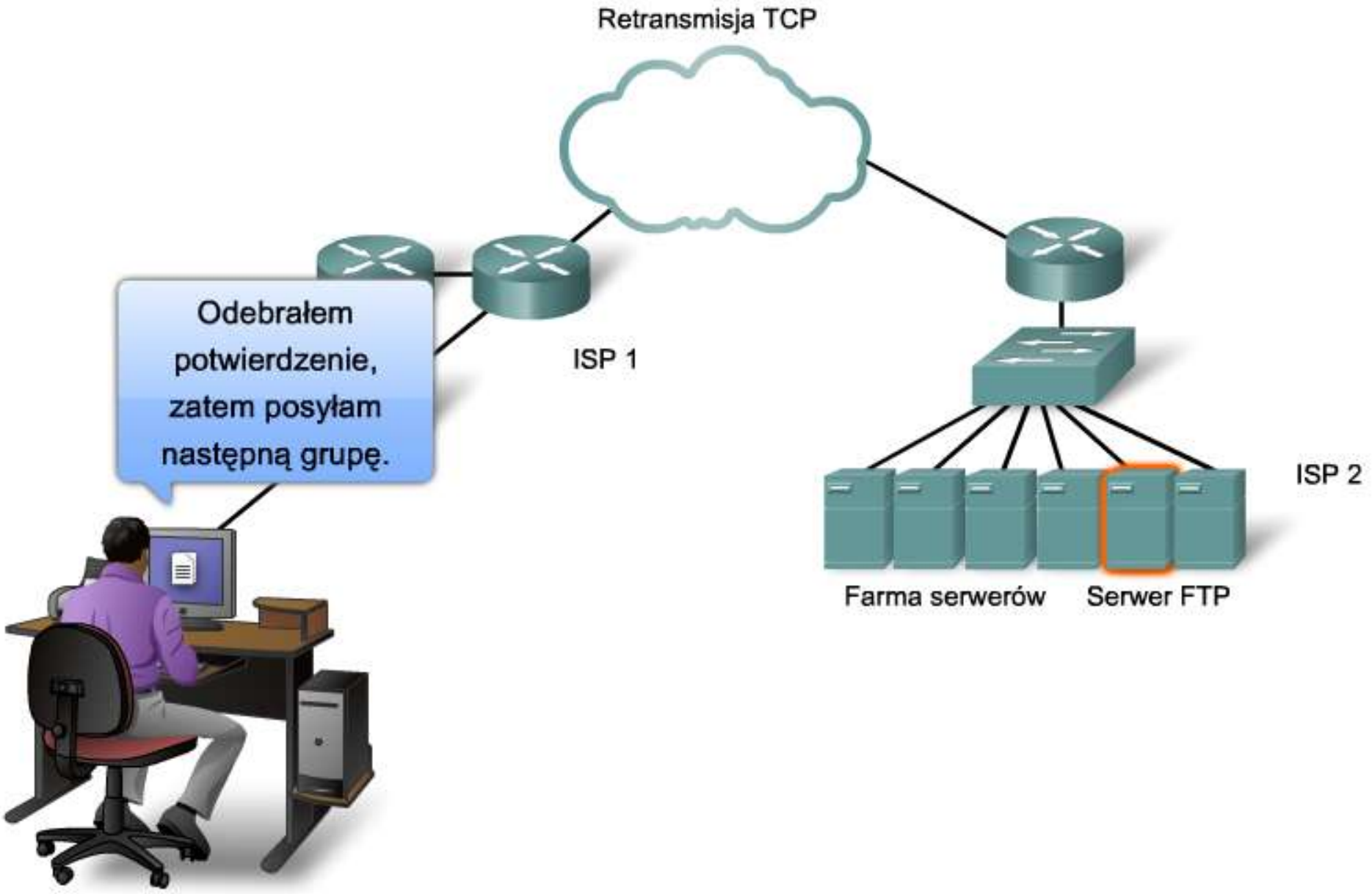
Zarządzanie utraconymi segmentami



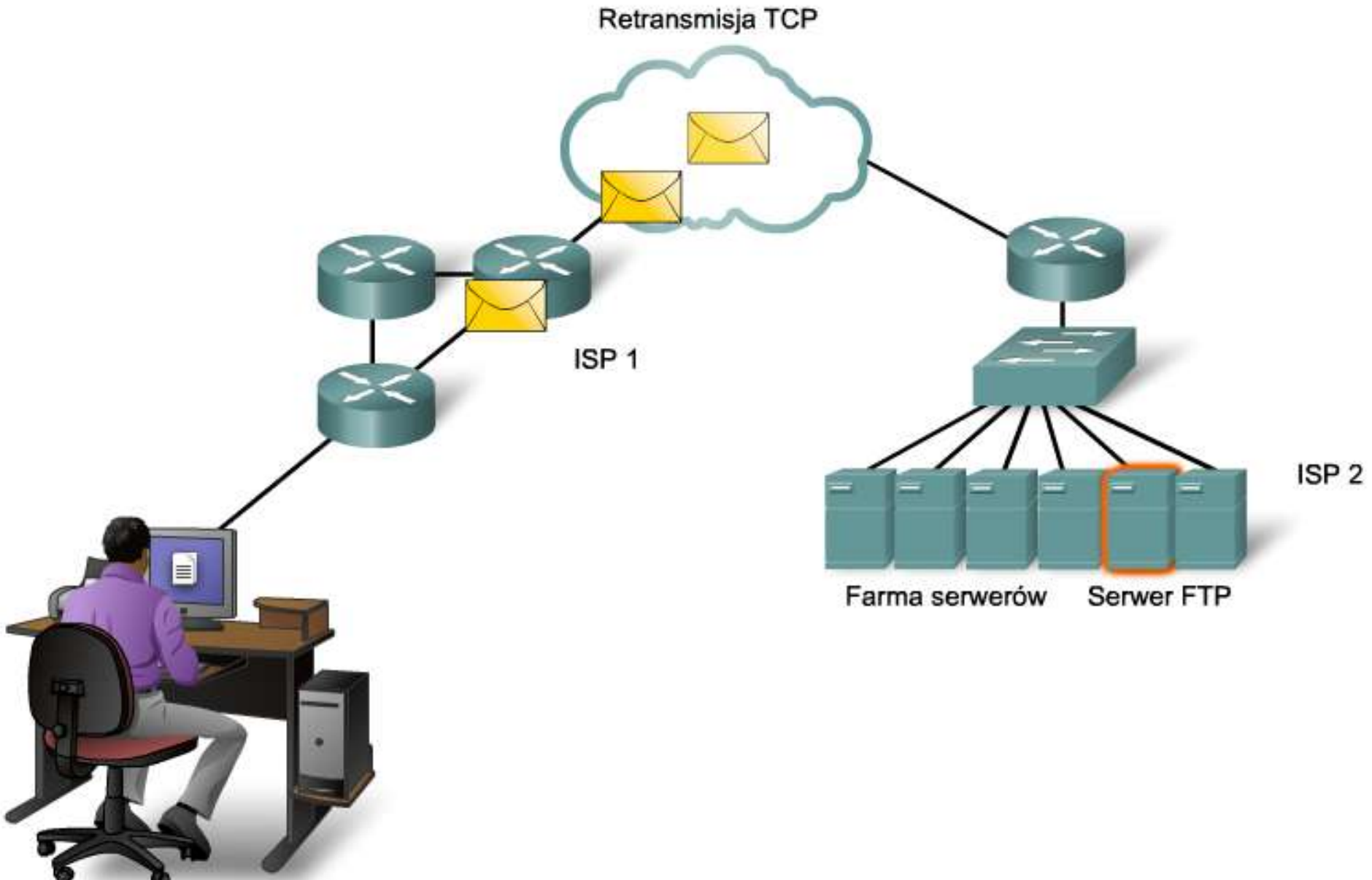
Zarządzanie utraconymi segmentami



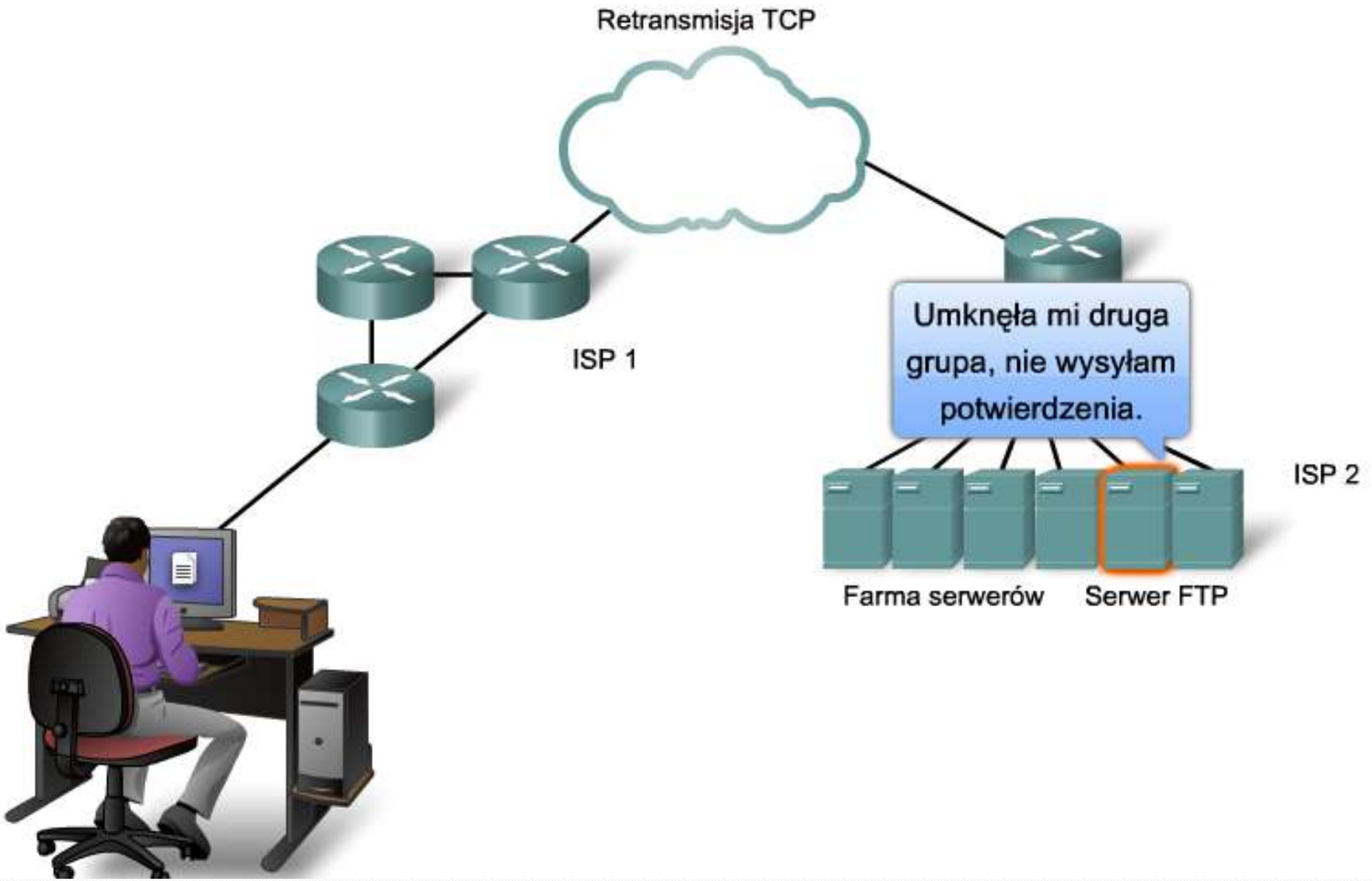
Zarządzanie utraconymi segmentami



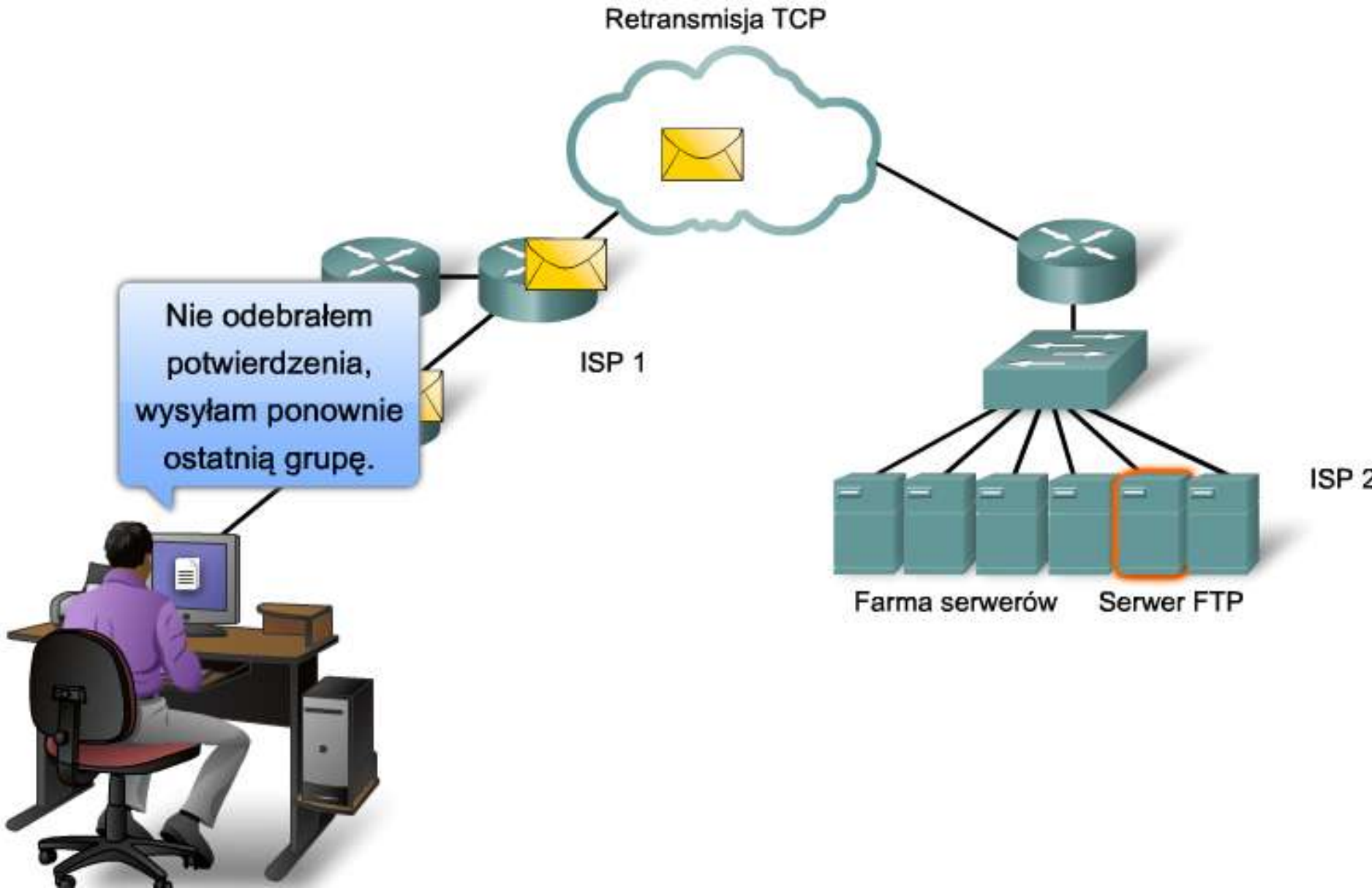
Zarządzanie utraconymi segmentami



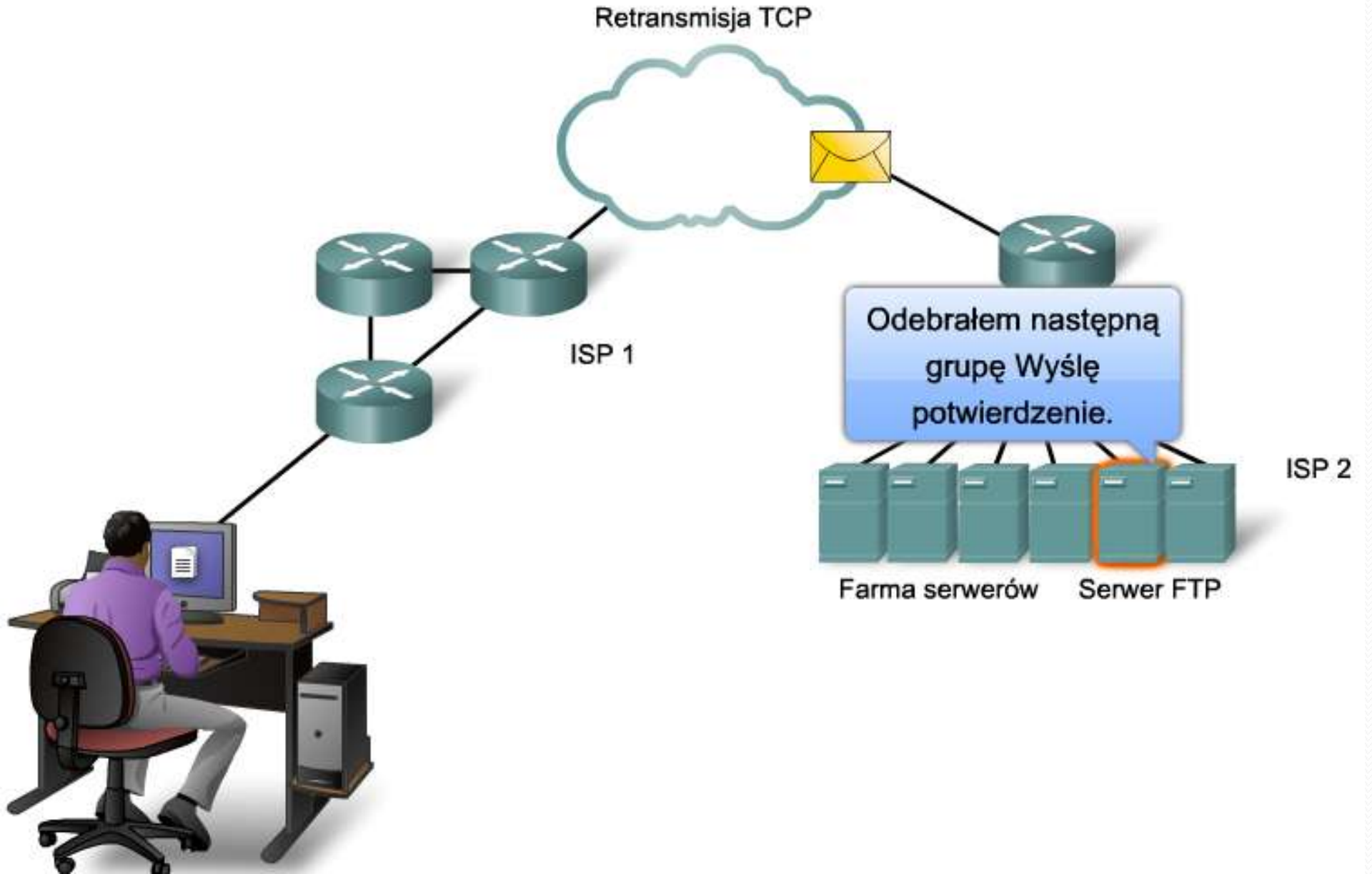
Zarządzanie utraconymi segmentami



Zarządzanie utraconymi segmentami



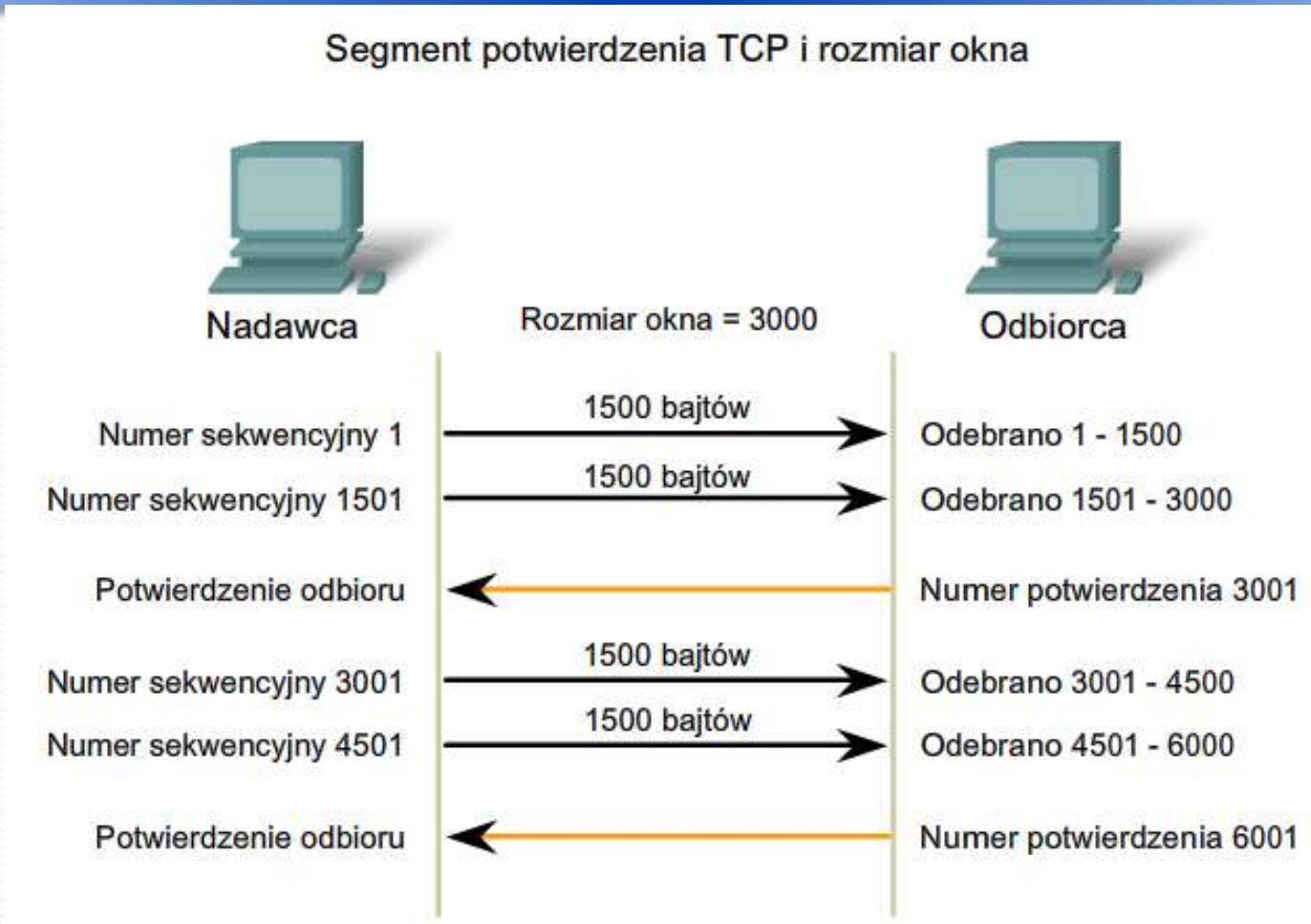
Zarządzanie utraconymi segmentami



Kontrola przepływu

- Protokół TCP zapewnia również mechanizm kontroli przepływu (ang. flow control). Kontrola przepływu wspiera mechanizm zapewnienia niezawodności transmisji TCP przez dostosowanie tempa przepływu danych między dwoma usługami w sesji. Jeśli nadawca dowiaduje się, że określona ilość danych zawarta w segmencie została otrzymana, może kontynuować nadawanie ze zwiększoną ilością danych dla tej sesji.

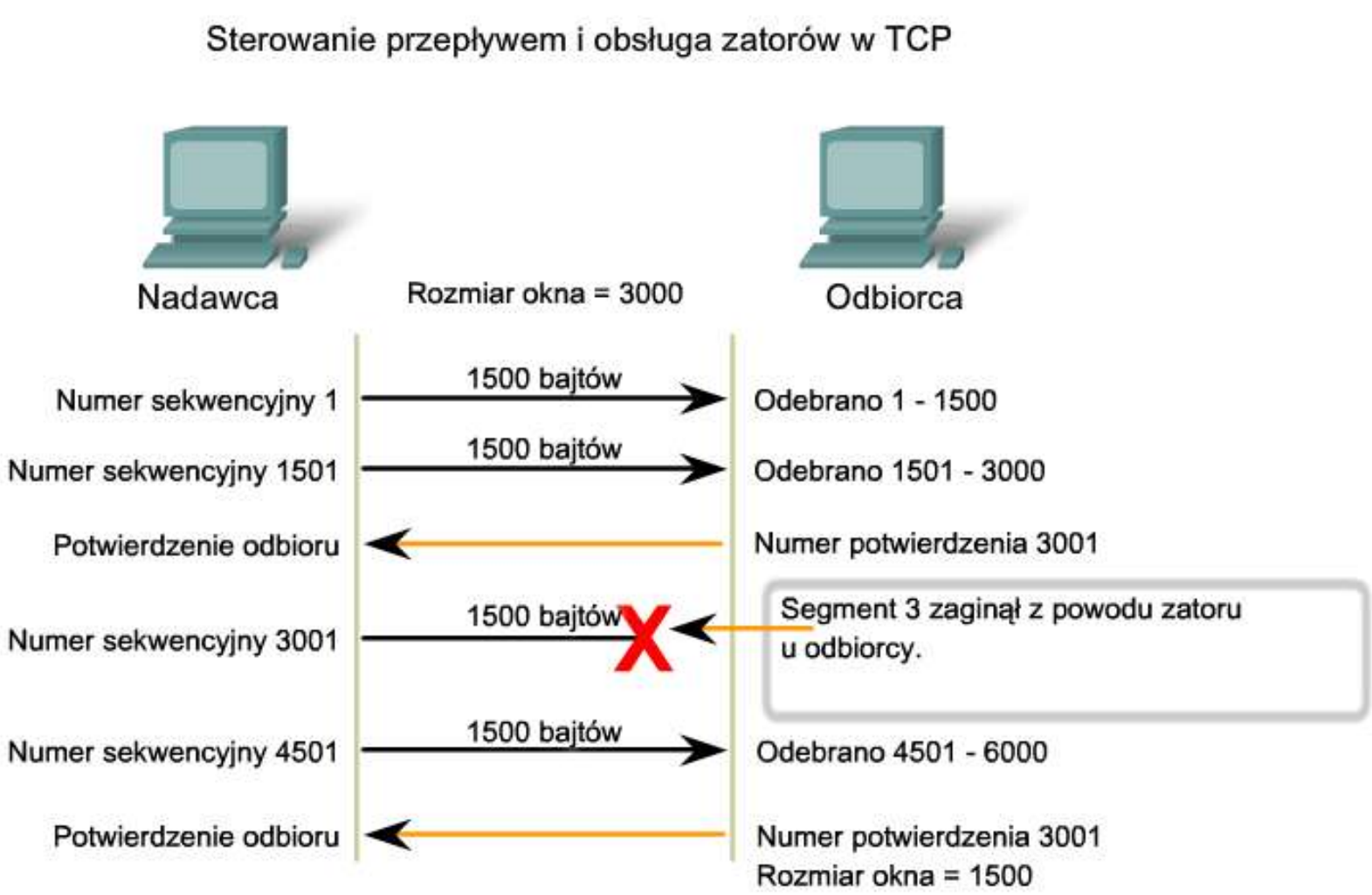
Kontrola przepływu



Rozmiar okna określa liczbę bajtów wysłanych zanim nadawca będzie oczekiwał na ich potwierdzenie.

Wartość potwierdzenia jest numerem następnego spodziewanego bajtu.

Kontrola przepływu



Jeśli segment zostanie zgubiony z powodu zatoru, odbiorca potwierdzi ostatni prawidłowo odebrany segment, lecz ze zmniejszonym rozmiarem okna.

UDP

- Protokół UDP (*User Datagram Protocol)* jest prostym protokołem, który zapewnia podstawowe funkcje warstwy transportowej. Ma on mniejszą ilość informacji kontrolnych niż TCP gdyż nie jest zorientowany połączeniowo, co oznacza, że nie zapewnia wyszukanych mechanizmów retransmisji, porządkowania odebranych danych czy kontroli przepływu.

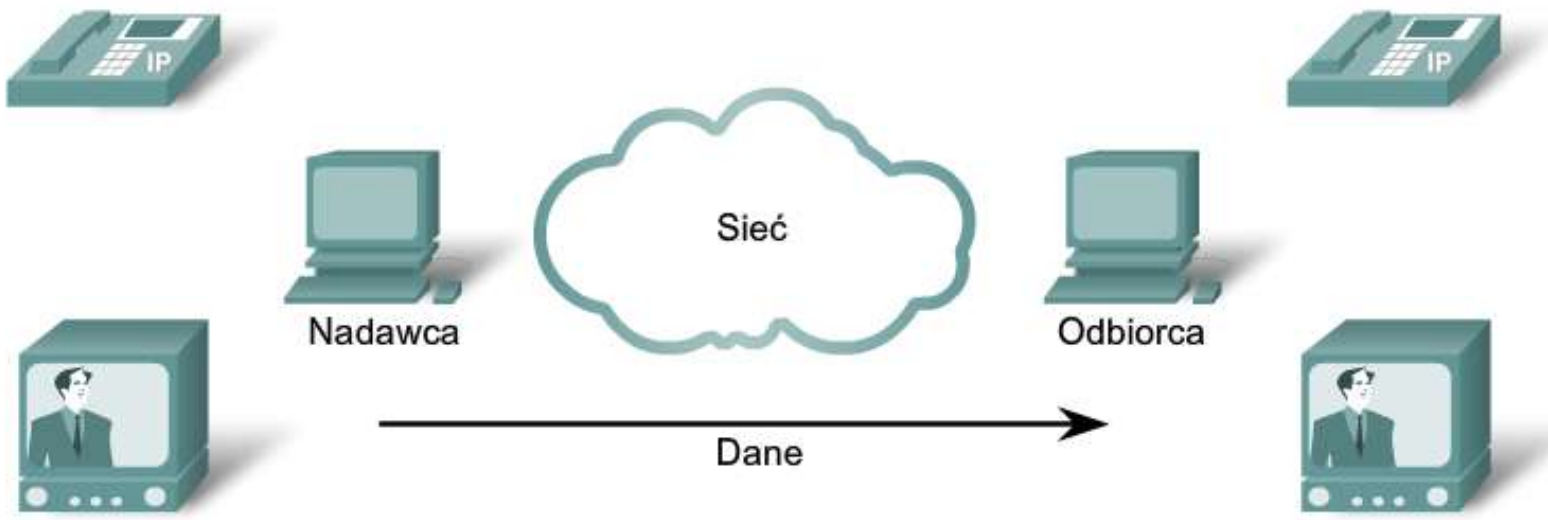
UDP

Do głównych aplikacji oraz protokołów warstwy aplikacji korzystających z tego protokołu zaliczamy:

- protokół DNS (ang. Domain Name System, system nazw domenowych),
- protokół SNMP (ang. Simple Network Management Protocol),
- protokół DHCP (ang. Dynamic Host Configuration Protocol),
- protokół RIP (ang. Routing Information Protocol),
- protokół TFTP (ang. Trivial File Transfer Protocol),
- gry online.

UDP

Transport danych UDP z małym narzutem



UDP nie ustanawia połączenia przed wysłaniem danych.

UDP zapewnia mały narzut z powodu niewielkiego nagłówka oraz braku zarządzania ruchem w sieci.

UDP – protokół bezpołączeniowy

- Ponieważ protokół UDP nie jest połączeniowy, sesje nie są nawiązywane przed rozpoczęciem transmisji między hostami, jak to ma miejsce w przypadku użycia protokołu TCP. Mówi się, że protokół UDP jest protokołem transakcyjnym. Innymi słowy, kiedy aplikacja ma dane do wysłania, po prostu je wysyła.

PDU warstwy transportowej

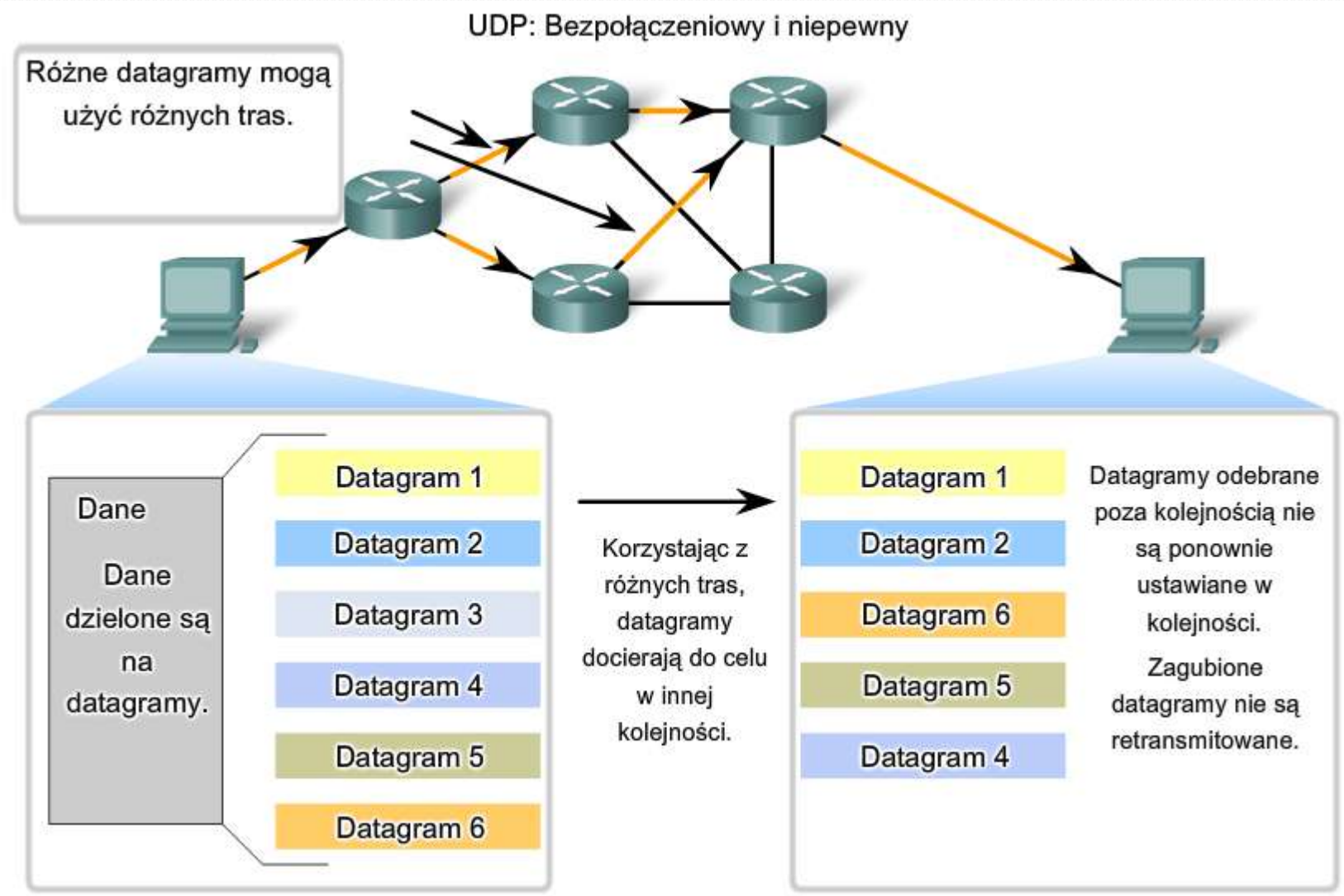
- Datagram – jednostka danych PDU protokołu UDP
- Segment – jednostka danych PDU protokołu TCP

Terminy segment i datagram czasami używane są zamiennie przy opisywaniu jednostki danych protokołów warstwy transportowej.

Scalanie datagramów

- Kiedy kilka datagramów jest wysyłanych do odbiorcy, mogą one być przesyłane różnymi ścieżkami i przybyć do odbiorcy w niewłaściwej kolejności. Protokół UDP nie znakuje segmentów przy użyciu numerów sekwencyjnych tak jak robi to TCP. Stąd protokół UDP nie ma możliwości uporządkowania odebranych datagramów według kolejności ich nadawania.

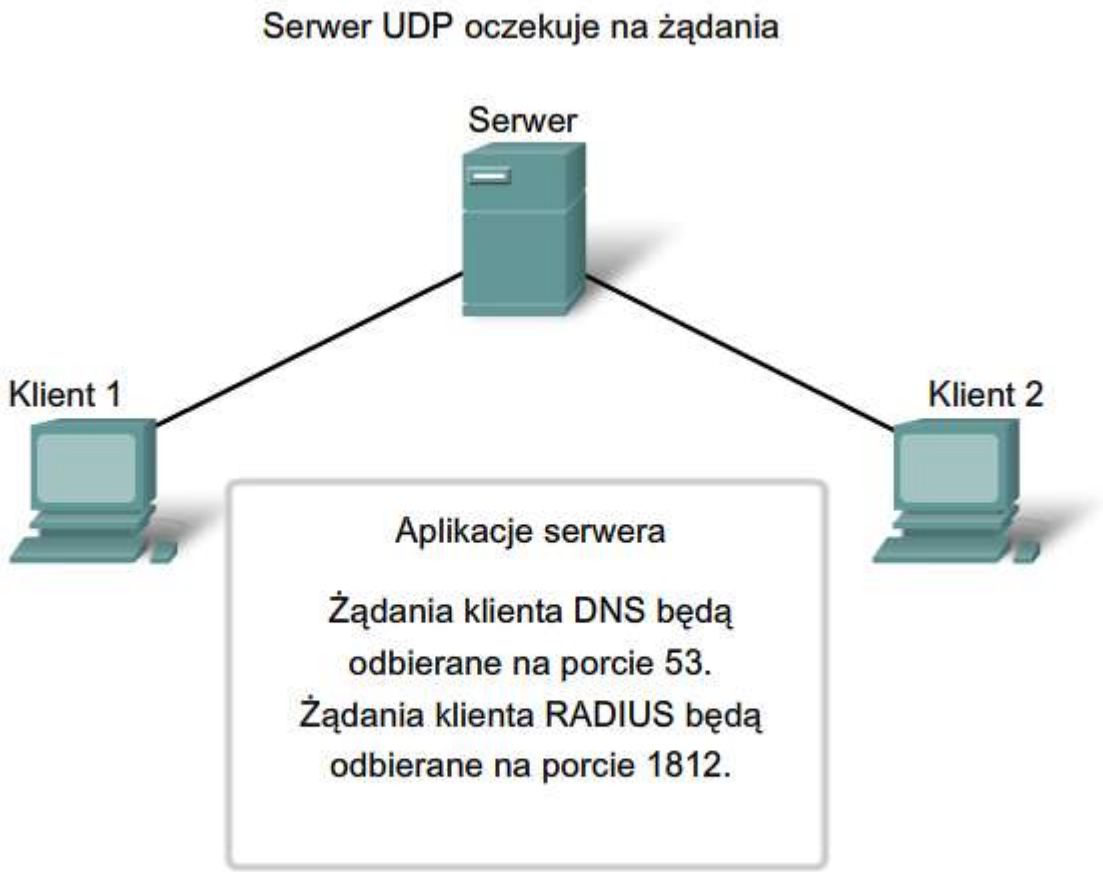
Scalanie datagramów



Porty adresem aplikacji

- Tak jak w aplikacjach wykorzystujących protokół TCP, aplikacje serwerowe wykorzystujące protokół UDP mają przypisane numery portów. Kiedy protokół UDP otrzymuje datagram adresowany do jednego z tych portów, to na jego podstawie przekazuje odebrane dane do odpowiedniej aplikacji.

Porty adresem aplikacji



Żądania klienckie do serwera używają tzw. dobrze znanych numerów portów jako docelowych.

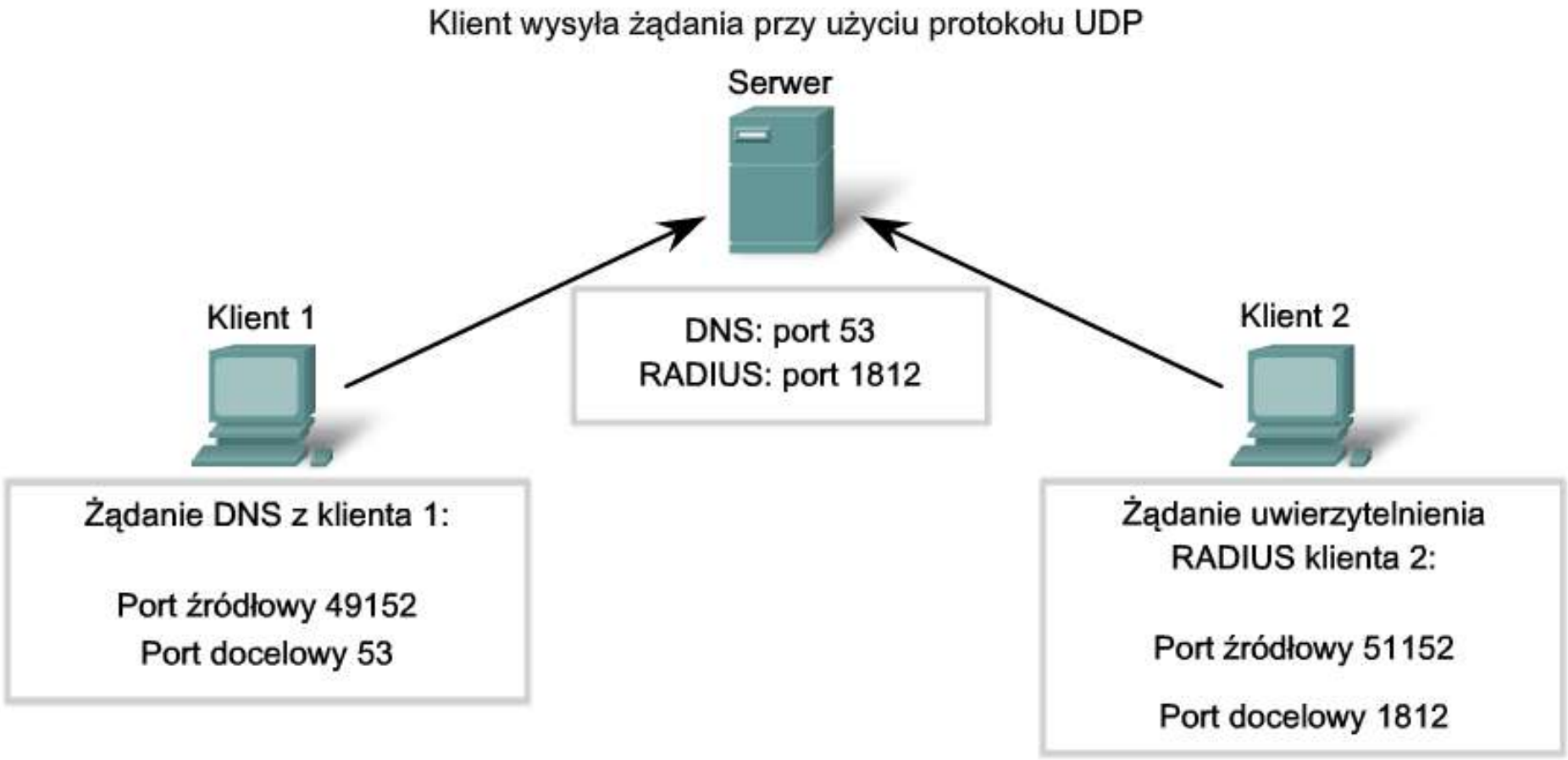
Klient protokołu UDP

- Komunikacja typu klient/serwer jest inicjowana przez aplikację po stronie klienta, która to żąda jakichś danych od jakiejś usługi serwera. Aplikacja klienta, losowo wybiera numer portu z zakresu portów dynamicznych i używa tego numeru jako portu źródłowego dla tej komunikacji. Docelowy port będzie zwykle z grupy dobrze znanych lub zarejestrowanych portów, który z kolei przyporządkowany jest usłudze na serwerze.

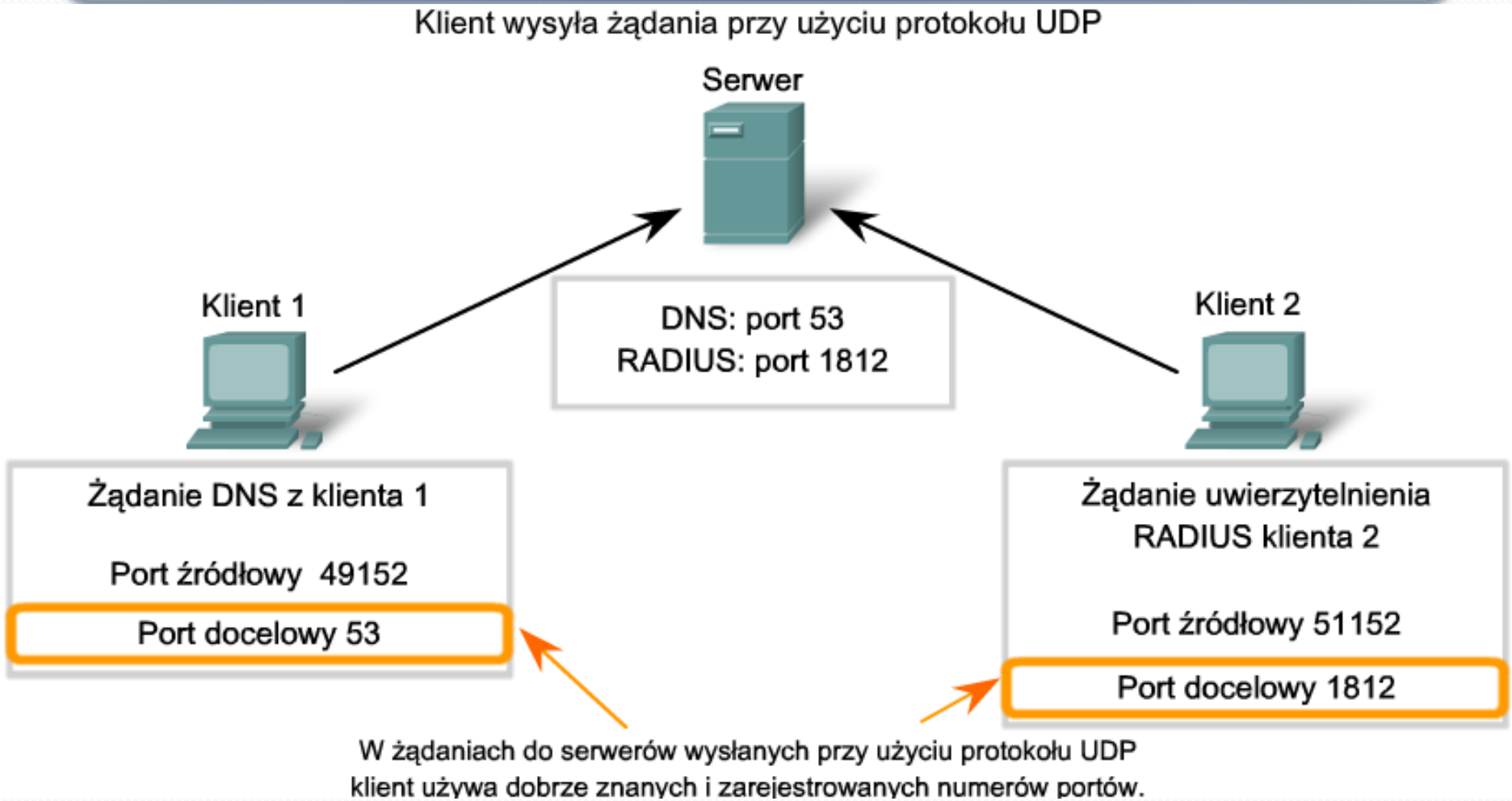
Bezpieczeństwo PDU

- Losowe numery portów zwiększają bezpieczeństwo. Gdybyśmy mieli przewidywalne porty źródłowe, intruz (włamywacz) mógłby łatwiej uzyskać dostęp do klienta poprzez połączenie z nim na numer portu, który najprawdopodobniej byłby użyty i otwarty (na kliencie) dla komunikacji z serwerem.

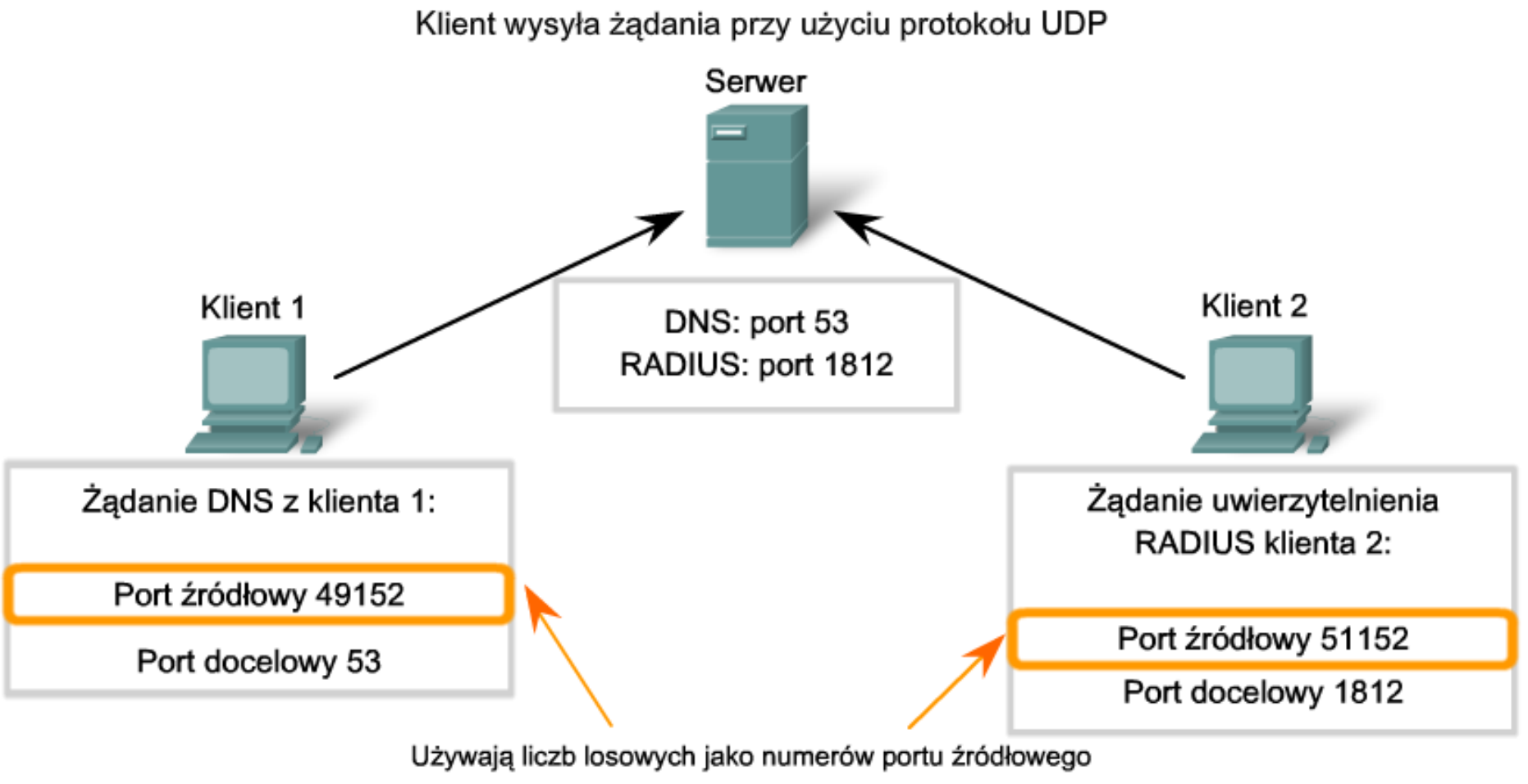
Klient protokołu UDP



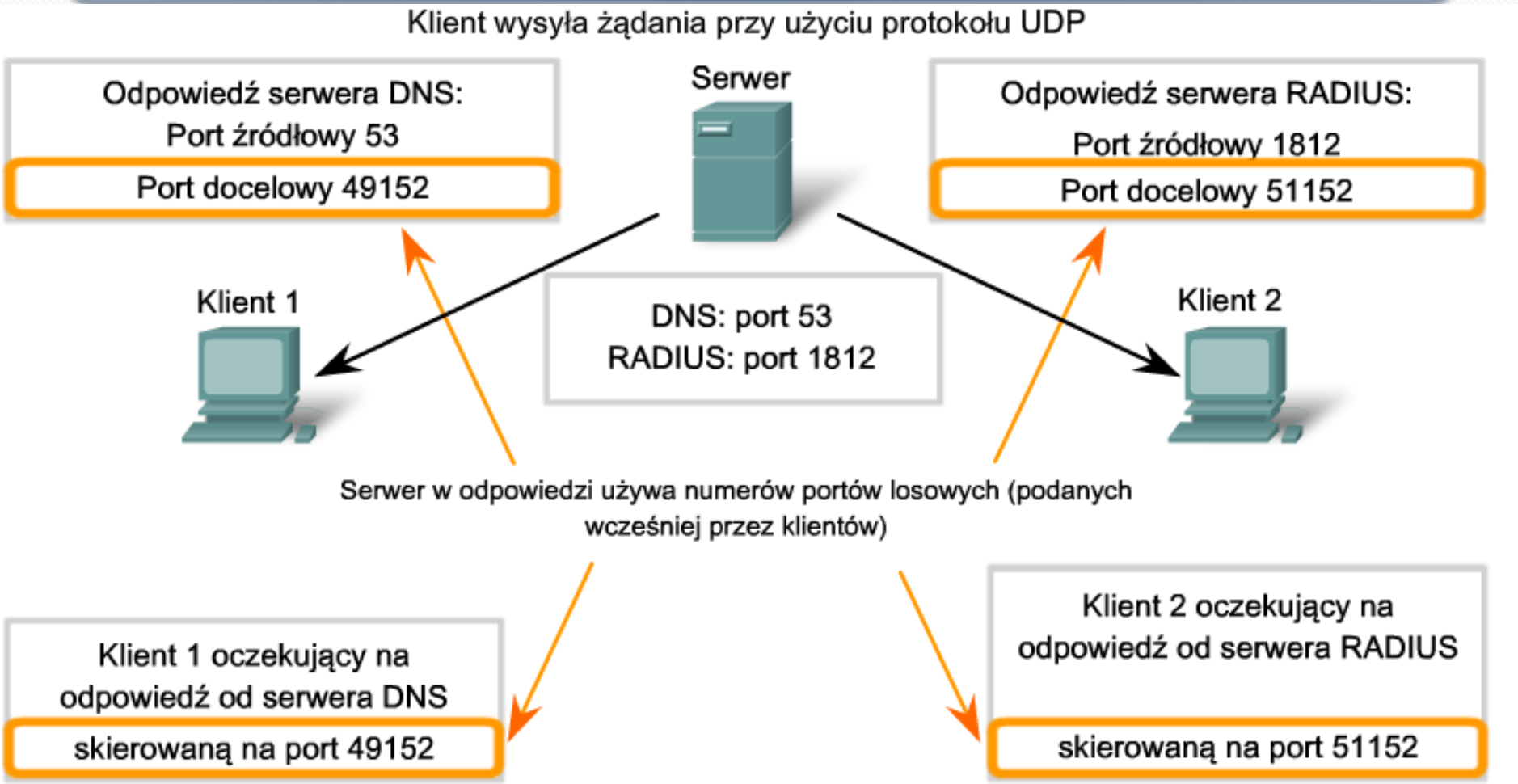
Klient protokołu UDP



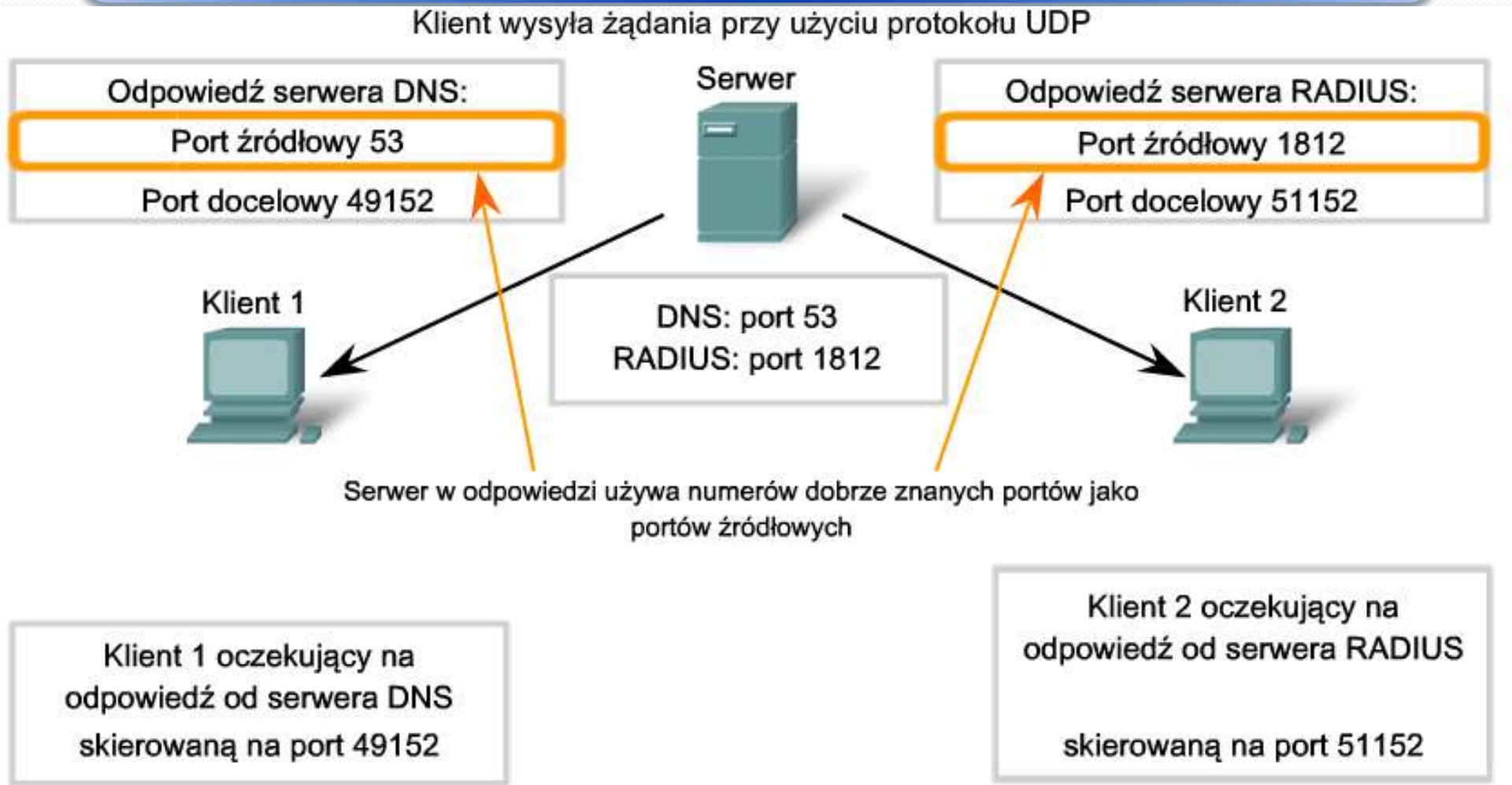
Klient protokołu UDP



Klient protokołu UDP



Klient protokołu UDP



Adresacja portów

- Dzięki usługom TCP i UDP hosty są w stanie śledzić na bieżąco te aplikacje. By rozróżniać segmenty i datagramy każdej z aplikacji, zarówno TCP jak i UDP mają w nagłówkach pola, dzięki którym można jednoznacznie zidentyfikować te aplikacje. Te unikalne identyfikatory to numery portów.

Adresacja portów

- Dzięki usługom TCP i UDP hosty są w stanie śledzić na bieżąco te aplikacje. By rozróżniać segmenty i datagramy każdej z aplikacji, zarówno TCP jak i UDP mają w nagłówkach pola, dzięki którym można jednoznacznie zidentyfikować te aplikacje. Te unikalne identyfikatory to numery portów.

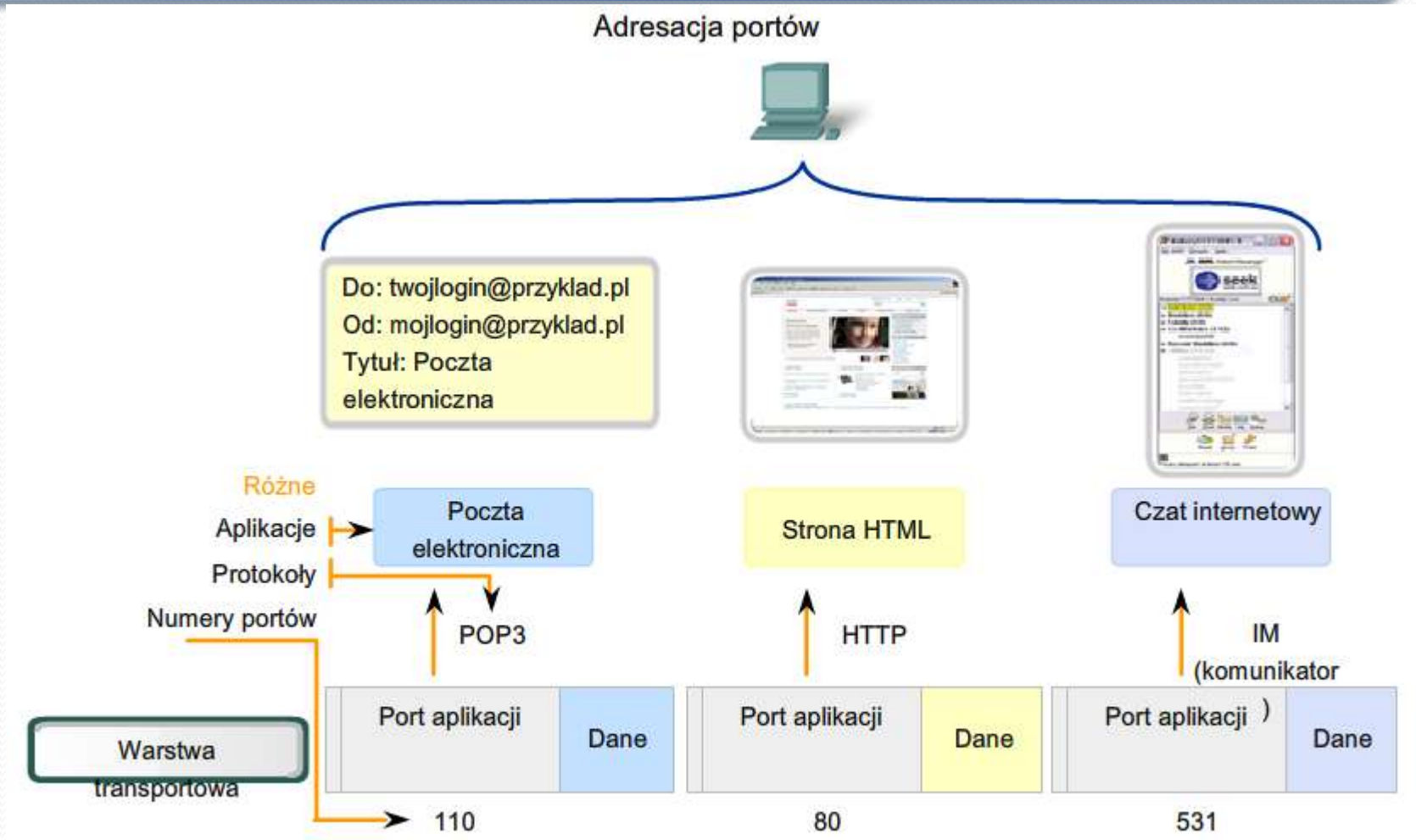
Adresacja portów

- Numery portów mogą być przypisane na różne sposoby, w zależności czy dana wiadomość to zapytanie czy odpowiedź. Serwery posiadają statyczne numery portów przypisane do aplikacji, z kolei klienci dynamicznie wybierają porty dla każdej konwersacji.

Adresacja portów

- Połączenie numeru portu i adresu IP jednoznacznie identyfikuje konkretny proces na konkretnym urządzeniu. Taka kombinacja zwana jest gniazdem (ang. socket). Czasem zdarza się, że terminy numer portu i gniazdo używane są wymiennie. Para gniazd składająca się ze źródłowych i docelowych adresów IP i numerów portów identyfikuje jednoznacznie konkretną konwersację pomiędzy hostami.

Adresacja portów



Dane związane z różnymi aplikacjami kierowane są tylko do odpowiedniej aplikacji dzięki zastosowaniu unikalnych numerów portów.

Rodzaje portów

Istnieją następujące typy numerów portów:

- **Dobrze znane porty** (numery od 0 do 1023) - te numery są zarezerwowane dla usług i aplikacji.
- **Zarejestrowane porty** (numery od 1024 do 49151) - te numery są zarezerwowane dla aplikacji i procesów użytkownika.
- **Dynamiczne lub prywatne numery portów** (numery od 49152 do 65535) - znane również pod nazwą "ephemeral ports", to numery portów, które są dynamicznie losowane przez aplikacje klienckie podczas inicjowania połączeń.

Rodzaje portów

Numery portów

Zakres portów	Grupa portów
od 0 do 1023	Dobrze znane porty
od 1024 do 49151	Porty zarejestrowane
od 49152 do 65535	Prywatne i/lub dynamiczne porty

Rodzaje portów (TCP)

Numery portów

Zakres portów	Grupa portów
od 0 do 1023	Dobrze znane porty
od 1024 do 49151	Porty zarejestrowane
od 49152 do 65535	Prywatne i/lub dynamiczne porty

Zarejestrowane porty TCP:
1863 MSN Messenger
8008 Dodatkowy port dla HTTP
8080 Dodatkowy port dla HTTP

Dobrze znane porty TCP:
21 FTP
23 Telnet
25 SMTP
80 HTTP
110 POP3
194 Protokół IRC
443 Bezpieczny HTTP (HTTPS)

Rodzaje portów (UDP)

Numery portów

Zakres portów	Grupa portów
od 0 do 1023	Dobrze znane porty
od 1024 do 49151	Porty zarejestrowane
od 49152 do 65535	Prywatne i/lub dynamiczne porty

Zarejestrowane porty UDP:
1812 Protokół uwierzytelniania RADIUS
2000 Cisco SCCP (VoIP)
5004 RTP (Protokoły dla głosu i video)
5060 SIP (VoIP)

Dobrze znane porty UDP:
69 TFTP
520 RIP

Rodzaje portów (TCP i UDP)

Numery portów

Zakres portów	Grupa portów
od 0 do 1023	Dobrze znane porty
od 1024 do 49151	Porty zarejestrowane
od 49152 do 65535	Prywatne i/lub dynamiczne porty

Porty zarejestrowane, wspólne dla TCP i UDP:
1433 MS SQL
2948 WAP (MMS)

Dobrze znane porty, wspólne dla TCP i UDP:
53 DNS
161 SNMP
531 Komunikator AOL, IRC

NETSTAT

- Netstat to ważne narzędzie sieciowe umożliwiające ich zweryfikowanie. Netstat wyświetla używane protokoły, lokalne adresy i numery portów, zdalne adresy i numery portów oraz status każdego połączenia.

NETSTAT

Wyjście polecenia netstat

```
C:\>netstat

Active Connections

  Proto  Local Address          Foreign Address        State
  TCP    kenpc:3126             192.168.0.2:netbios-ssn    ESTABLISHED
  TCP    kenpc:3158             207.138.126.152:http       ESTABLISHED
  TCP    kenpc:3159             207.138.126.169:http       ESTABLISHED
  TCP    kenpc:3160             207.138.126.169:http       ESTABLISHED
  TCP    kenpc:3161             sc.msn.com:http            ESTABLISHED
  TCP    kenpc:3166             www.cisco.com:http         ESTABLISHED

C:\>
```

Użyte protokoły

NETSTAT

Wyjście polecenia netstat

```
C:\>netstat

Active Connections

Proto    Local Address          Foreign Address        State
TCP      kenpc:3126             192.168.0.2:netbios-ssn    ESTABLISHED
TCP      kenpc:3158             207.138.126.152:http       ESTABLISHED
TCP      kenpc:3159             207.138.126.169:http       ESTABLISHED
TCP      kenpc:3160             207.138.126.169:http       ESTABLISHED
TCP      kenpc:3161             sc.msn.com:http            ESTABLISHED
TCP      kenpc:3166             www.cisco.com:http         ESTABLISHED

C:\>
```

Port źródłowy

NETSTAT

Wyjście polecenia netstat

```
C:\>netstat

Active Connections

Proto    Local Address          Foreign Address        State
TCP      kenpc:3126             192.168.0.2:netbios-ssn    ESTABLISHED
TCP      kenpc:3158             207.138.126.152:http       ESTABLISHED
TCP      kenpc:3159             207.138.126.169:http       ESTABLISHED
TCP      kenpc:3160             207.138.126.169:http       ESTABLISHED
TCP      kenpc:3161             sc.msn.com:http            ESTABLISHED
TCP      kenpc:3166             www.cisco.com:http         ESTABLISHED

C:\>
```

Adres lub nazwa zdalnego hosta

NETSTAT

Wyjście polecenia netstat

```
C:\>netstat

Active Connections

Proto    Local Address          Foreign Address        State
TCP      kenpc:3126             192.168.0.2:netbios-ssn    ESTABLISHED
TCP      kenpc:3158             207.138.126.152:http       ESTABLISHED
TCP      kenpc:3159             207.138.126.169:http       ESTABLISHED
TCP      kenpc:3160             207.138.126.169:http       ESTABLISHED
TCP      kenpc:3161             sc.msn.com:http            ESTABLISHED
TCP      kenpc:3166             www.cisco.com:http         ESTABLISHED

C:\>
```

Port docelowy

NETSTAT

Wyjście polecenia netstat

```
C:\>netstat

Active Connections

Proto    Local Address          Foreign Address        State
TCP      kenpc:3126             192.168.0.2:netbios-ssn    ESTABLISHED
TCP      kenpc:3158             207.138.126.152:http       ESTABLISHED
TCP      kenpc:3159             207.138.126.169:http       ESTABLISHED
TCP      kenpc:3160             207.138.126.169:http       ESTABLISHED
TCP      kenpc:3161             sc.msn.com:http            ESTABLISHED
TCP      kenpc:3166             www.cisco.com:http         ESTABLISHED

C:\>
```

Stan połączenia