

# TECHNIKI ROUTINGU W SIECIACH KOMPUTEROWYCH

Routing z wykorzystaniem stanu łącza, OSPF  
część 1

opracowanie na podstawie materiałów Cisco

Marcin Raniszewski

Roman Krzeszewski

Łukasz Sturgulewski

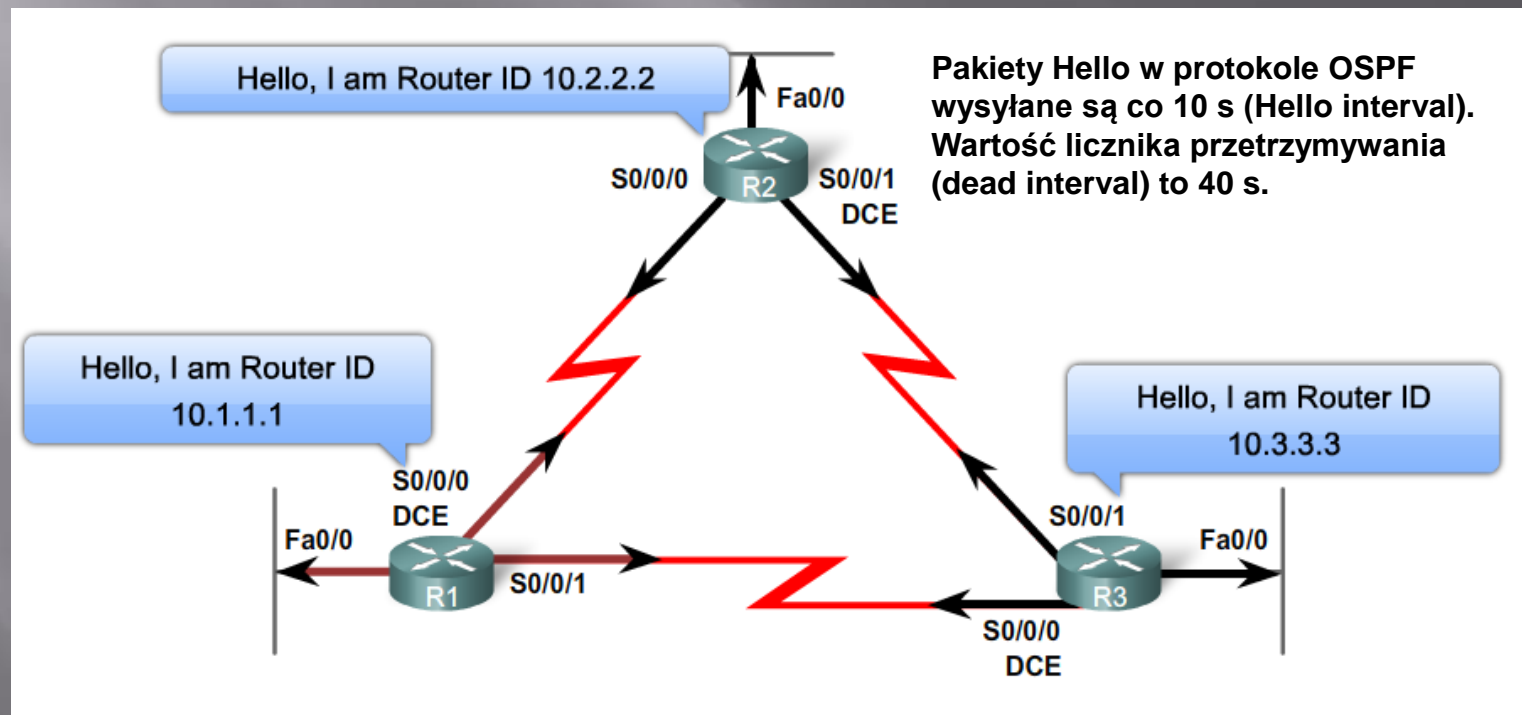
Grzegorz Nowak

# Plan wykładu

- ▣ Tworzenie relacji sąsiedzkich, typy pakietów OSPF
- ▣ Komunikat OSPF Hello
- ▣ Sposoby otrzymywania informacji o routingu
- ▣ Zalety i wady routingu według stanu łącza
- ▣ Protokół OSPF — główne cechy
- ▣ Algorytm stanu łącza
- ▣ Pojęcie kosztu – metryka w OSPF
- ▣ Algorytm SPF (Shortest Path First)
- ▣ Pojęcia obszarów i routerów brzegowych w OSPF
- ▣ Sąsiedzi w OSPF

# Tworzenie relacji przylegania

Określanie sąsiedztwa pomiędzy routerami odbywa się za pomocą rozsyłania pakietów *Hello*. Każdy router wysyła okresowo pakiety *Hello* do przyległych sąsiadów, informując je o swoim istnieniu i sprawnym działaniu. Mechanizm ten umożliwia wykrycie, że dany router przestał odpowiadać, jeśli brak jest pakietu *Hello* w okresie *czasu przetrzymywania* (*dead interval*).



# Typy pakietów OSPF

Rozróżnia się 5 podstawowych typów pakietów LSP (*ang. link-state packet*):

1. **Hello** – pakiety hello służą do tworzenia i podtrzymywania przyległości z innymi routerami OSPF.
2. **DBD** – opis bazy danych (*database description*), pakiet zawiera skróconą listę bazy danych łącze-stan routera wysyłającego.
3. **LSR** – routery mogą zażądać dodatkowej informacji o dowolnym wpisie DBD, wysyłając żądanie LSR (*link-state request*).
4. **LSU** – pakiety aktualizacji (*link-state update*) są używane do odpowiadania na LSR i do ogłaszania nowych informacji. Pakiety LSU zawiera jedno lub więcej ogłoszeń **LSA** (*ang. link state advertisement*). Niekiedy określeń LSU i LSA używa się zamiennie).
5. **LSAck** – po odebraniu pakietu LSU router wysyła potwierdzenie LSAck (*link-state acknowledge*), aby potwierdzić odebranie pakietu.

0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3

**Wersja:** Numer wersji OSPF.

**Typ:** Typ pakietu OSPF. W OSPF wyróżnia się 5 typów pakietów, takich jak poniżej:

- Typ 1 Hello - Odkrywanie/zarządzanie sąsiadami
- Typ 2 Database Description - Sumaryzacja zawartości bazy danych
- Typ 3 Link State Request - Download bazy danych
- Typ 4 Link State Update - Aktualizacja bazy danych
- Typ 5 Link State Ack - Potwierdzenie rozplywowe

**Długość pakietu:** Długość pakietu OSPF w bajtach, łącznie z długością nagłówka OSPF.

**Router ID:** Identyfikator routera OSPF, który wysłał ten pakiet.

**Area ID:** 32 bitowy numer identyfikacyjny obszaru OSPF (AREA), do której należy ten pakiet. Wszystkie pakiety OSPF są skojarzone z jedną area.

**Suma kontrolna:** standardowa suma kontrolna taka jak w IP, liczona z całego pakietu OSPF łącznie z główkiem, ale bez 64-bitów pola Autoryzacji.

**AuType:** Identyfikuje procedurę autoryzacji pakietu OSPF.

- 0 - brak autoryzacji
- 1 - proste hasło
- 2 - autoryzacja kryptograficzna (algorytm MD5)

**Autoryzacja:** 64-bitowe pole schematu autoryzacji.

# Nagłówek komunikatu OSPF (24oktety)

**Maska sieci:** maska sieci przypisana do interfejsu, z którego wysłano pakiet Hello.

**Opcje:** dodatkowe funkcjonalności obsługiwane przez router.

0	1	2	3	4	5	6	7
*	*	DC	EA	N/P	MC	E	*

- E-bit - rozgłaszanie (floodowanie) AS-external-LSA
- MC-bit - opisuje czy datagram multicastowy IP jest przekazywany zgodnie z RFC1584 "Multicast Extensions to OSPF".
- N/P-bit - opisuje obsługę Type-7 LSA wg. RFC3101 "The OSPF NSSA Option".
- EA-bit - wyraża chęć do otrzymywania i przekazywania External-Attributes-LSA.
- DC-bit - obsługa żądania łączy wg. RFC1793 "Extending OSPF to Support Demand Circuits".

**HelloInterval:** ilość sekund między pakietami Hello.

**Rtr Pri:** Router Priority - priorytet routera, wykorzystywany przy wyborze routera DR i BDR [(Backup) Designated Router]. Wartość 0 oznacza wyłączenie z elekcji do DR i BDR.

**RouterDeadInterval:** ilość sekund przed deklaracją, że router jest nieosiągalny.

**Designated Router:** router desygnowany dla tej sieci. DR jest identyfikowany przez swój adres IP interfejsu w tej sieci. 0.0.0.0 oznacza, że brak jest DR.

**Backup Designated Router:** zapasowy router desygnowany dla tej sieci. BDR jest identyfikowany przez swój adres IP interfejsu w tej sieci. 0.0.0.0 oznacza, że brak jest BDR.

**Neighbor:** kolejne pola z RouterID sąsiadów, od których ten router ostatnio usłyszał poprawne pakiety Hello. Ostatnio oznacza, że w ciągu sekund jakie wyznacza RouterDeadInterval.

# Pakiet OSPF Hello

0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3



# Sposoby otrzymywania informacji o routingu

Protokoły routingu według stanu łącza korzystają z następujących elementów:

- ogłoszeń **LSA** (*Link-State Advertisement*); wymiana aktualizacji LSA (oprócz pakietów *Hello*) jest **wyzwalana przez wystąpienie zdarzenia w sieci**, nie jest natomiast aktualizacją cykliczną.
- **bazy danych o topologii**;
- **algorytmu SPF** (*Shortest Path First*);
- **drzewa SPF**;
- **tablicy routingu**.

# Sposoby otrzymywania informacji o routingu

- ▣ Łącze jest równoważne **interfejsowi** routera.
- ▣ Stan łącza jest opisem interfejsu oraz **relacji z sąsiednimi routerami**. Opis interfejsu może na przykład zawierać adres IP interfejsu, maskę podsieci, typ sieci, do której jest przyłączony, routery dołączone do danej sieci itp.
- ▣ Zbiór stanów poszczególnych łączy stanowi bazę danych stanów łączy, która jest nazywana **bazą danych o topologii**.
- ▣ Baza danych stanów łączy służy od obliczania najlepszych tras w sieci. Routery wykorzystują w tym celu **algorytm Dijkstry** (*Shortest Path First*). Algorytm ten tworzy drzewo SPF, którego korzeniem jest lokalny router. Następnie w drzewie SPF są wyszukiwane ścieżki do dostępnych sieci, które zostają umieszczone w **tablicy routingu**.



# Zalety routingu według stanu łącza

- ▣ Przy wyborze tras przez sieć protokoły routingu według stanu łącza używają metryki kosztu. **Metryka kosztu** odzwierciedla przepustowość łączy na tych trasach.
- ▣ Protokoły routingu według stanu łącza używają **wyzwalanych aktualizacji** oraz rozpływowego przekazywania pakietów LSA, aby móc natychmiast powiadamiać wszystkie routery w sieci o zmianach jej topologii. Prowadzi to do **szybkiej zbieżności**;
- ▣ Każdy router dysponuje pełnym i zsynchronizowanym **obrazem sieci**.
- ▣ Protokoły routingu według stanu łącza obsługują notacje **CIDR i VLSM**.

# Wady routingu według stanu łącza

- ▣ Wymagają większej ilości **pamięci i mocy obliczeniowej** niż protokoły działające na podstawie wektora odległości.
- ▣ Wymagają ściśle **hierarchicznego projektu sieci**, gdzie sieć jest podzielona na mniejsze obszary w celu zmniejszenia tablic topologii;
- ▣ Wymagają pracy **administratora** dobrze rozumiejącego działanie tych protokołów;
- ▣ Podczas początkowego procesu wykrywania sieć jest **zalewana pakietami LSA**. Proces ten może znacząco zmniejszyć możliwość przesyłania danych w sieci. Może to w widoczny sposób obniżyć wydajność sieci.

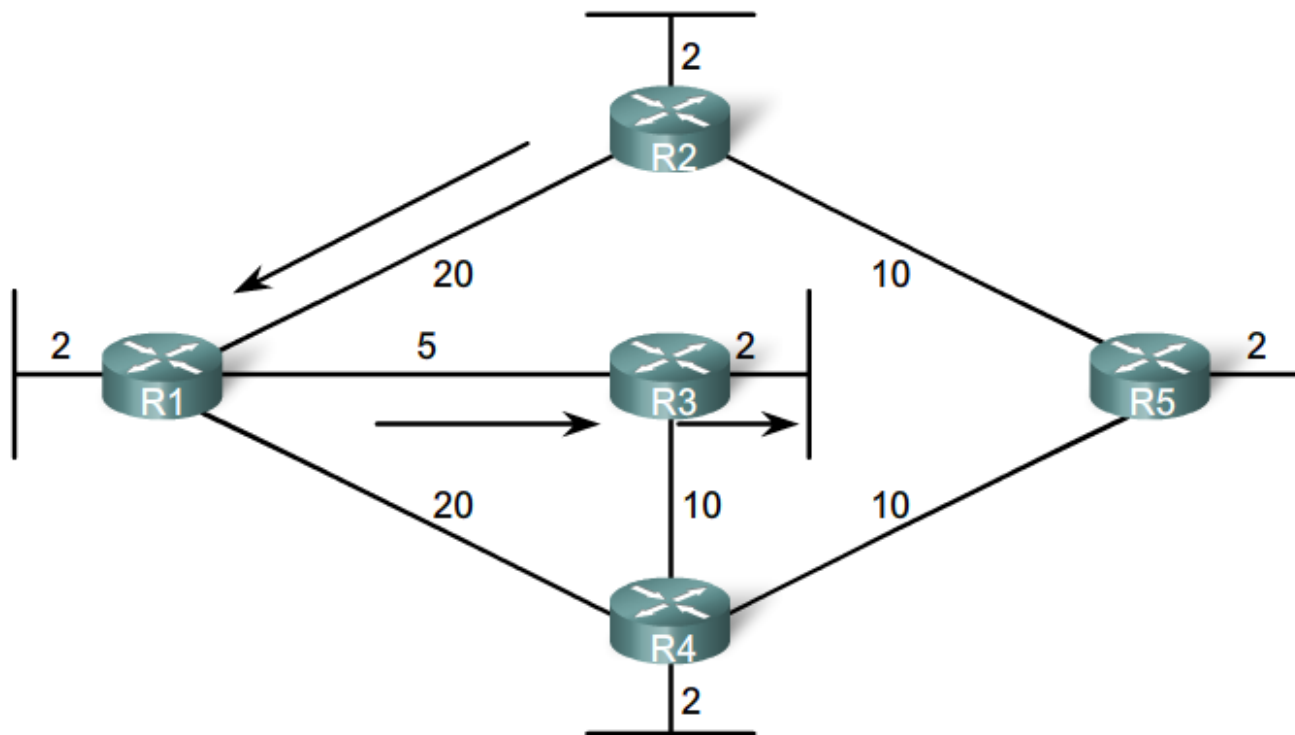
# Protokół OSPF — główne cechy

- ▣ Protokół wyszukiwania najkrótszej ścieżki **OSPF** (*Open Shortest Path First*) jest protokołem **IGP** (routing w obrębie jednego systemu autonomicznego).
- ▣ Został on dokładnie opisany w dokumencie **RFC 2328** (druga wersja OSPF).
- ▣ Jest rozwiązaniem bezklasowym, w pełni obsługującym techniki **CIDR** i **VLSM**.
- ▣ Gwarantuje **krótki czas zbieżności sieci i brak pętli**. Nie ma zdefiniowanej maksymalnej liczby przeskoków.
- ▣ Udostępnia **mechanizm uogólniania tras**, podobnie jak EIGRP.
- ▣ W sieciach, w których wymagany jest dodatkowy poziom zabezpieczeń, można w taki sposób skonfigurować routery OSPF, aby w trakcie komunikacji z innymi urządzeniami tego typu korzystały z **uwierzytelnienia**.
- ▣ Komunikaty w OSPF korzystają z **TCP/IP**.
- ▣ Przy wymianie komunikatów LSA, OSPF korzysta z **rozgłaszania grupowego**.
- ▣ Umożliwia także **podział sieci na obszary**, ograniczając rozgłaszanie pakietów LSA tylko do nich.
- ▣ Jego obsługa jest uwzględniana przez niemal wszystkich dostawców urządzeń sieciowych. Z tego względu doskonale **nadaje się do zastosowania** jako protokół routingu **w sieciach zbudowanych z urządzeń różnych firm**.

# Wprowadzenie do algorytmu SPF (Shortest Path First)

- ▣ Algorytm ten został opracowany przez holenderskiego informatyka Edsgera **Dijkstrę** i został opublikowany w roku 1959.
- ▣ W przypadku algorytmu SPF **najlepszą trasą jest trasa o najniższym koszcie.**
- ▣ Algorytm przedstawia **sieć jako zbiór węzłów** połączonych przez łącza punkt-punkt.
- ▣ Każdemu **łączu przypisany jest koszt.**
- ▣ Każdy węzeł ma **nazwę.**
- ▣ Każdy węzeł dysponuje pełną bazą danych wszystkich łączy, tak więc jest mu znana **pełna informacja o topologii** fizycznej.
- ▣ Wszystkie bazy danych stanów łączy znajdujące się w danym obszarze są **identyczne.**

# Wprowadzenie do algorytmu SPF



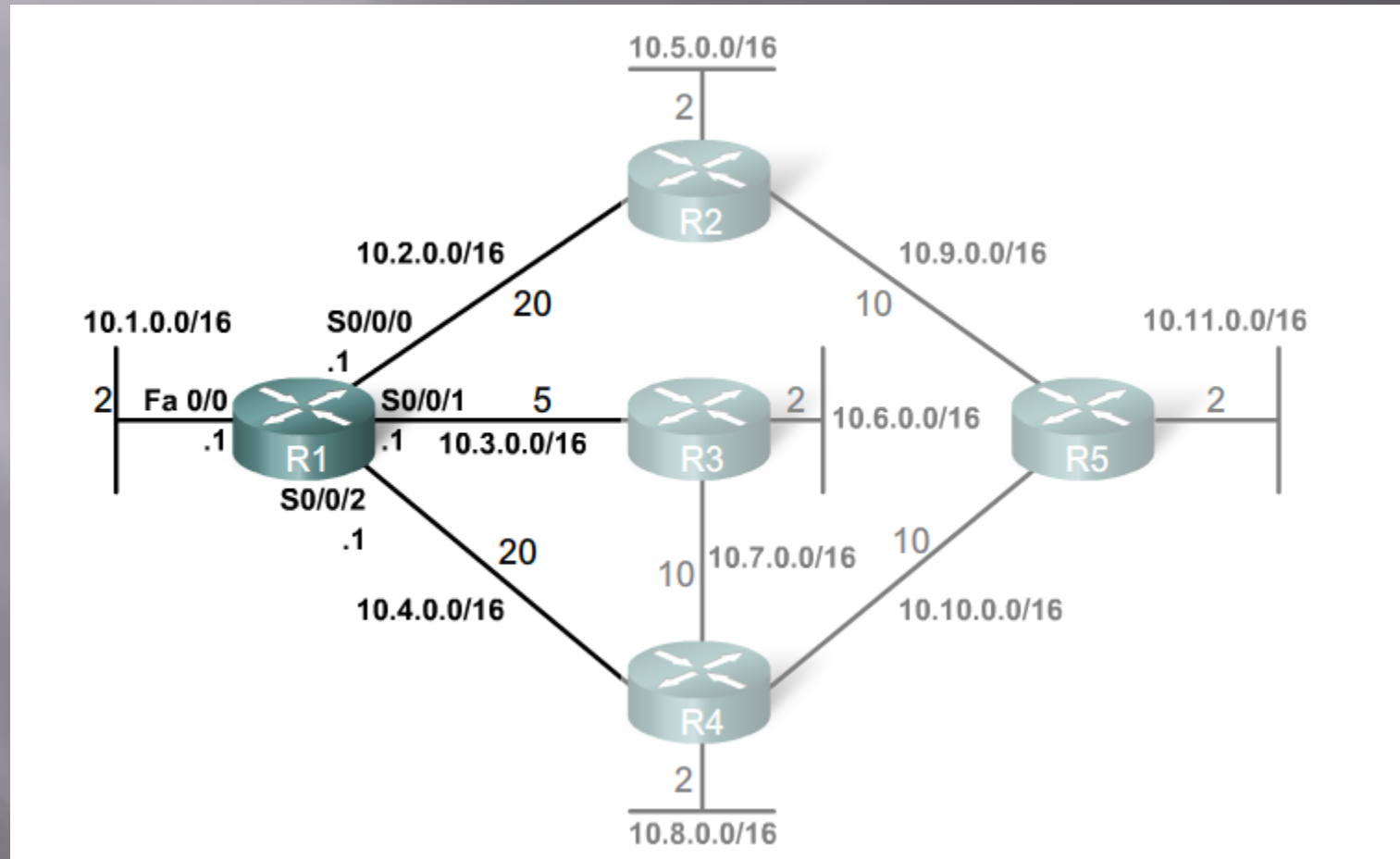


# Wprowadzenie do algorytmu SPF

1. Każdy router dowiaduje się o własnych **sieciach podłączonych bezpośrednio**.
2. Każdy router jest odpowiedzialny za wysłanie pakietów **Hello** do swoich sąsiadów w sieciach podłączonych bezpośrednio.
3. Każdy router **buduje pakiet LSP** (link-state packet) zawierający stan każdego łącza podłączonego bezpośrednio.
4. Każdy router zalewowo **wysyła pakiety LSP** do swoich sąsiadów. Zapisuje otrzymane dane w bazie danych.
5. Każdy router używa bazy danych do skonstruowania kompletnej **mapy topologii** i oblicza **najlepszą trasę** do każdej sieci docelowej.

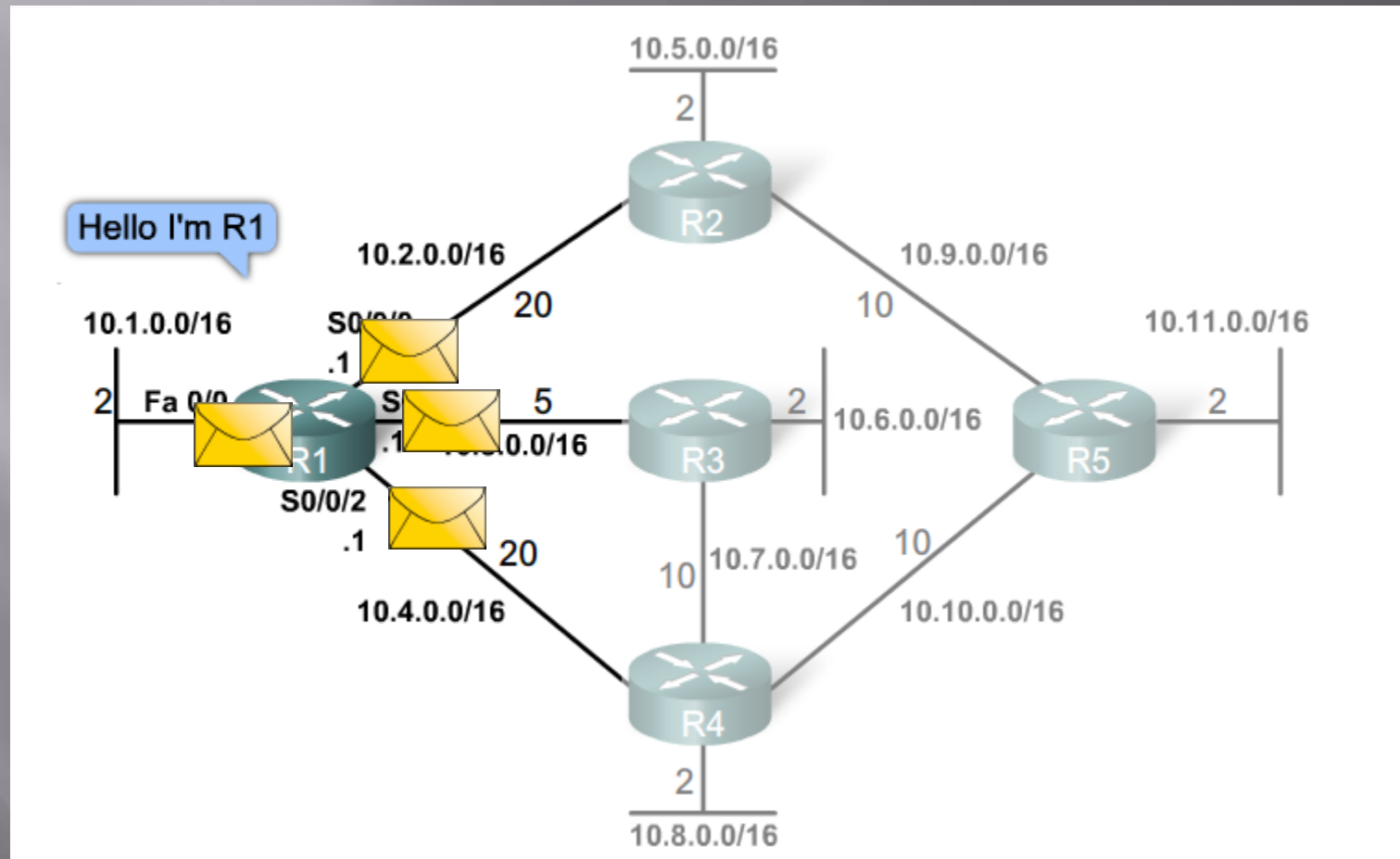
# Wprowadzenie do algorytmu SPF

1. Każdy router dowiaduje się o własnych sieciach podłączonych bezpośrednio.



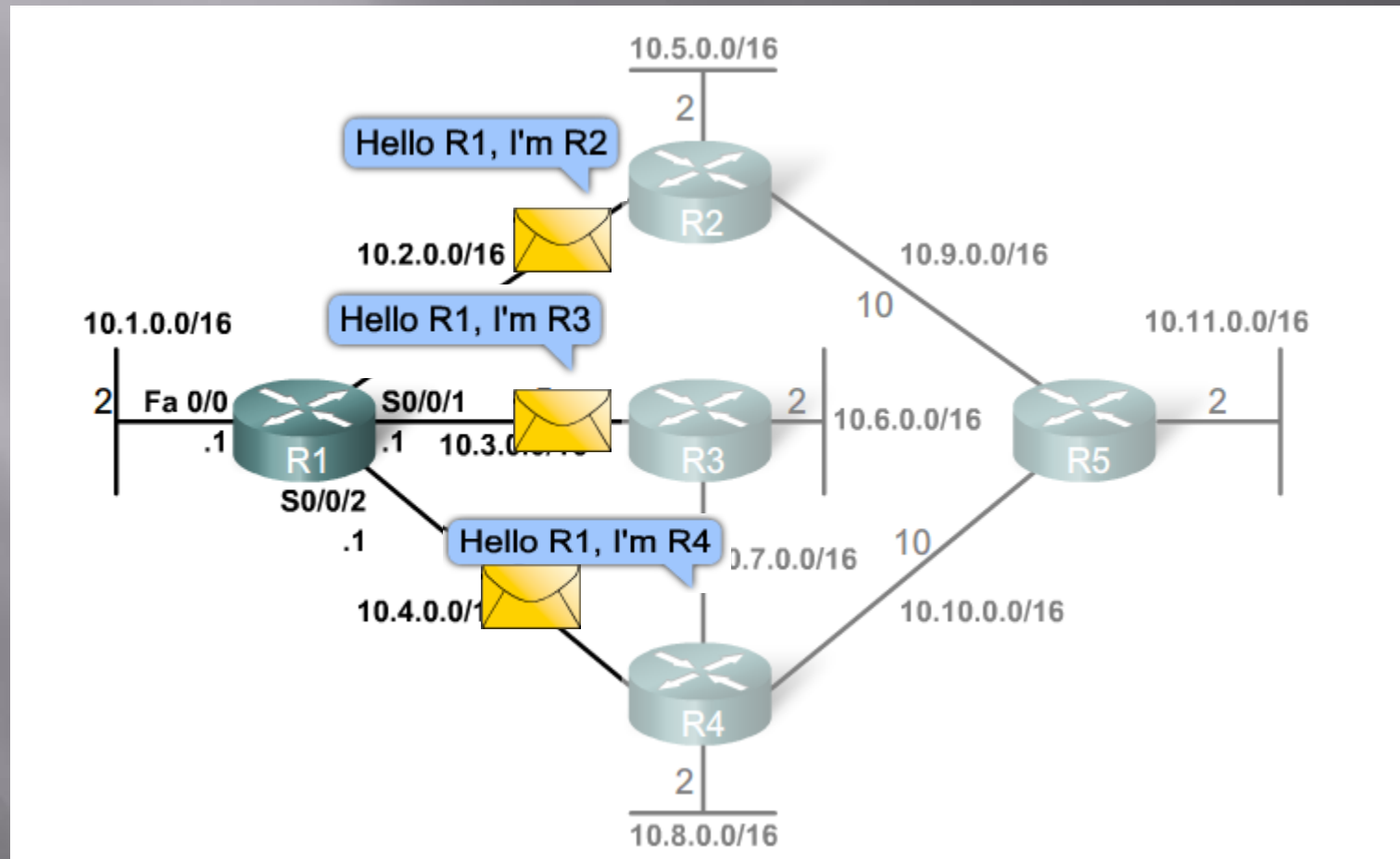
# Wprowadzenie do algorytmu SPF

2. Każdy router jest odpowiedzialny za wysłanie pakietów *Hello* do swoich sąsiadów w sieciach podłączonych bezpośrednio.



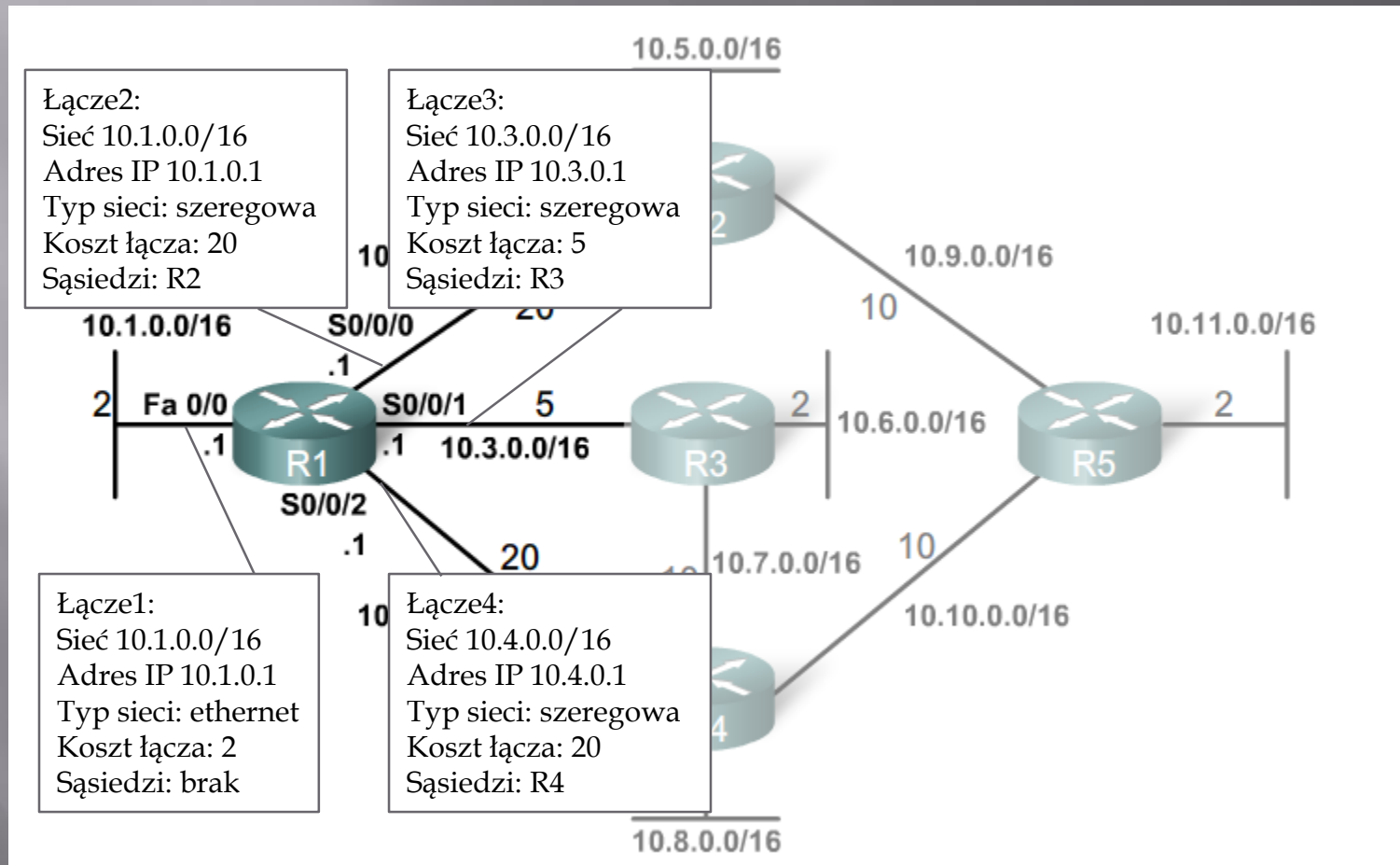
# Wprowadzenie do algorytmu SPF

2. Każdy router jest odpowiedzialny za wysłanie pakietów *Hello* do swoich sąsiadów w sieciach podłączonych bezpośrednio.



# Wprowadzenie do algorytmu SPF

3. Każdy router buduje pakiet LSP (link-state packet) zawierający stan każdego łącza podłączonego bezpośrednio.

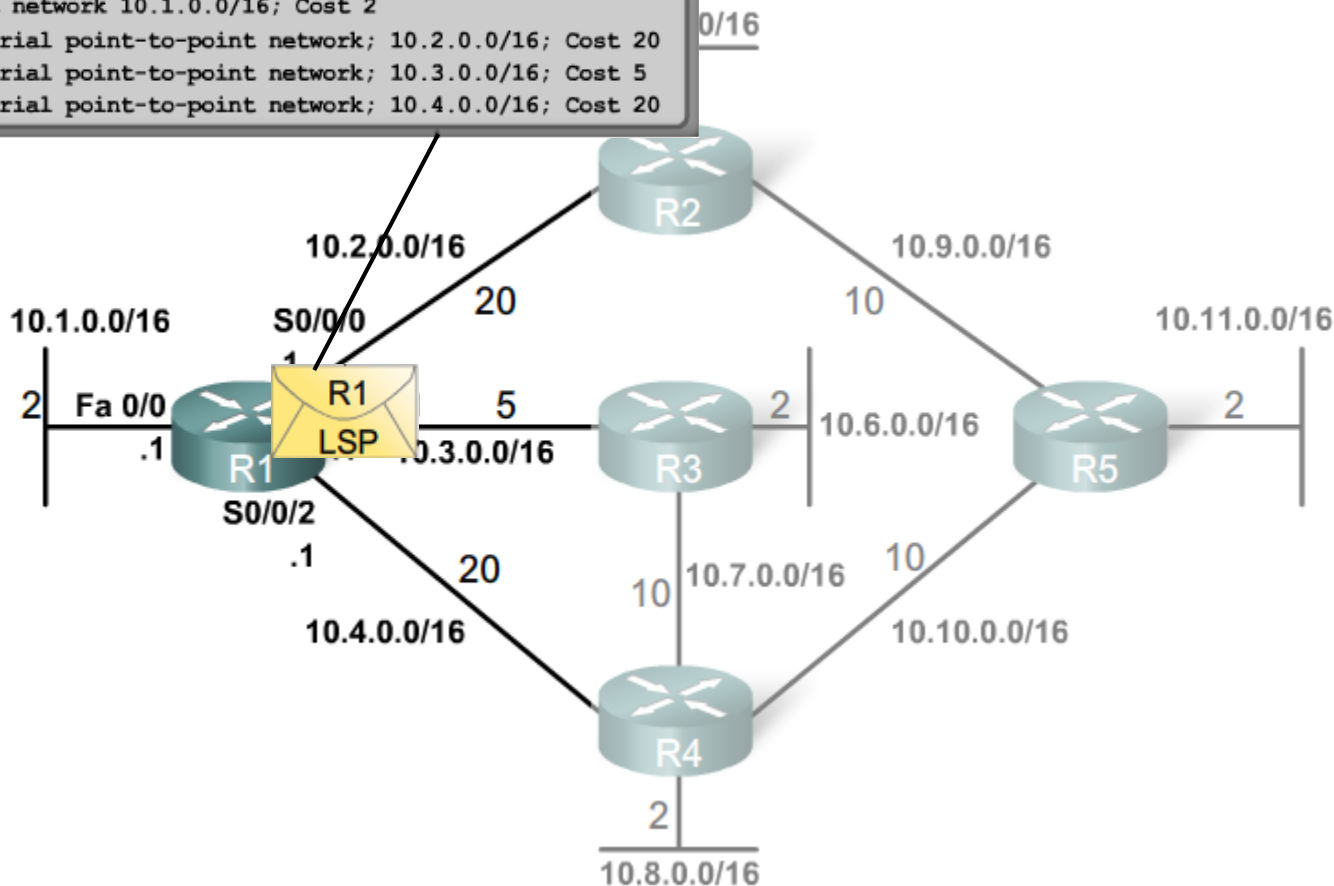




# Wprowadzenie do algorytmu SPF

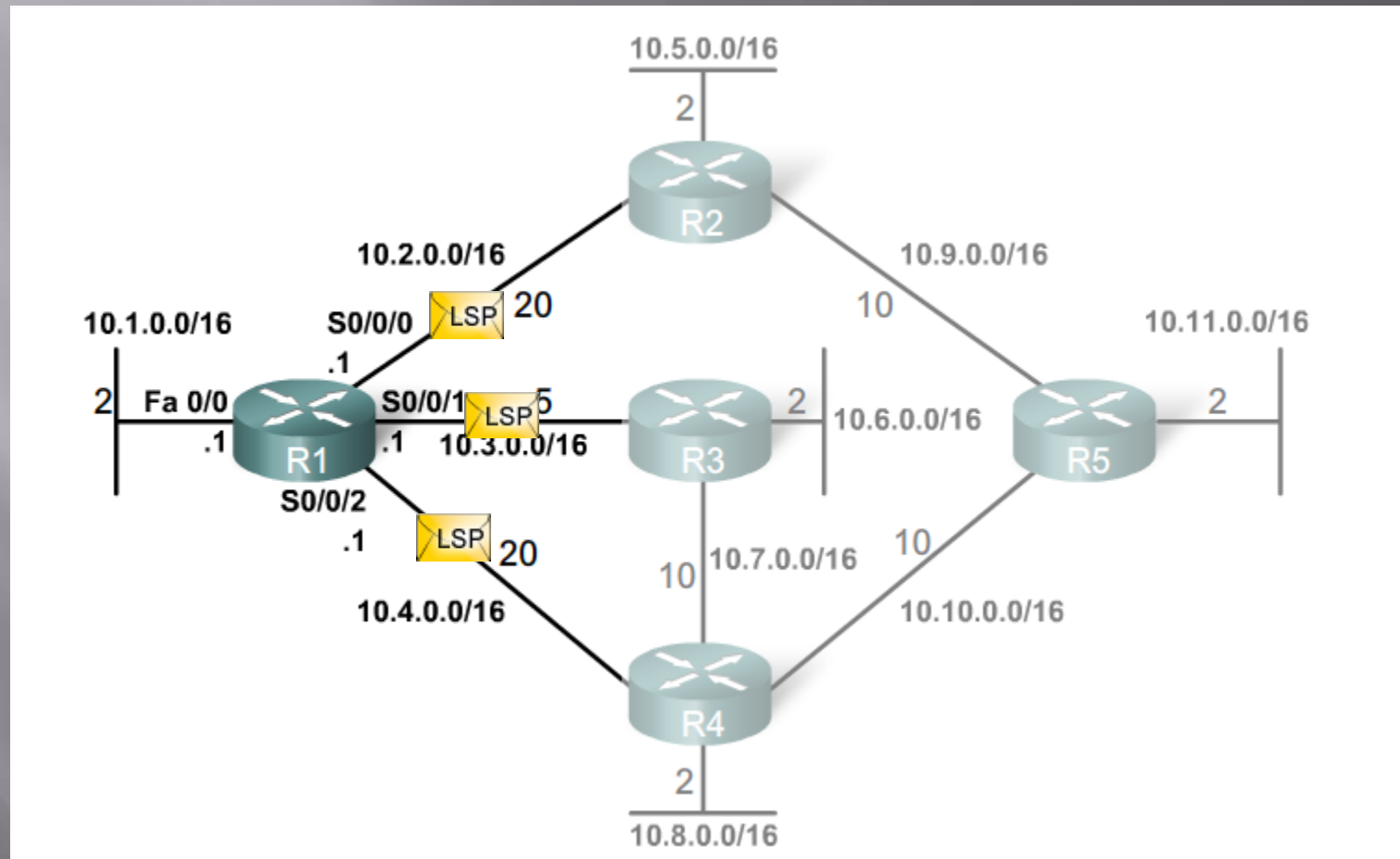
3. Każdy router buduje pakiet LSP (link-state packet) zawierający stan każdego łącza podłączonego bezpośrednio.

1. R1; Ethernet network 10.1.0.0/16; Cost 2
2. R1 -> R2; Serial point-to-point network; 10.2.0.0/16; Cost 20
3. R1 -> R3; Serial point-to-point network; 10.3.0.0/16; Cost 5
4. R1 -> R4; Serial point-to-point network; 10.4.0.0/16; Cost 20



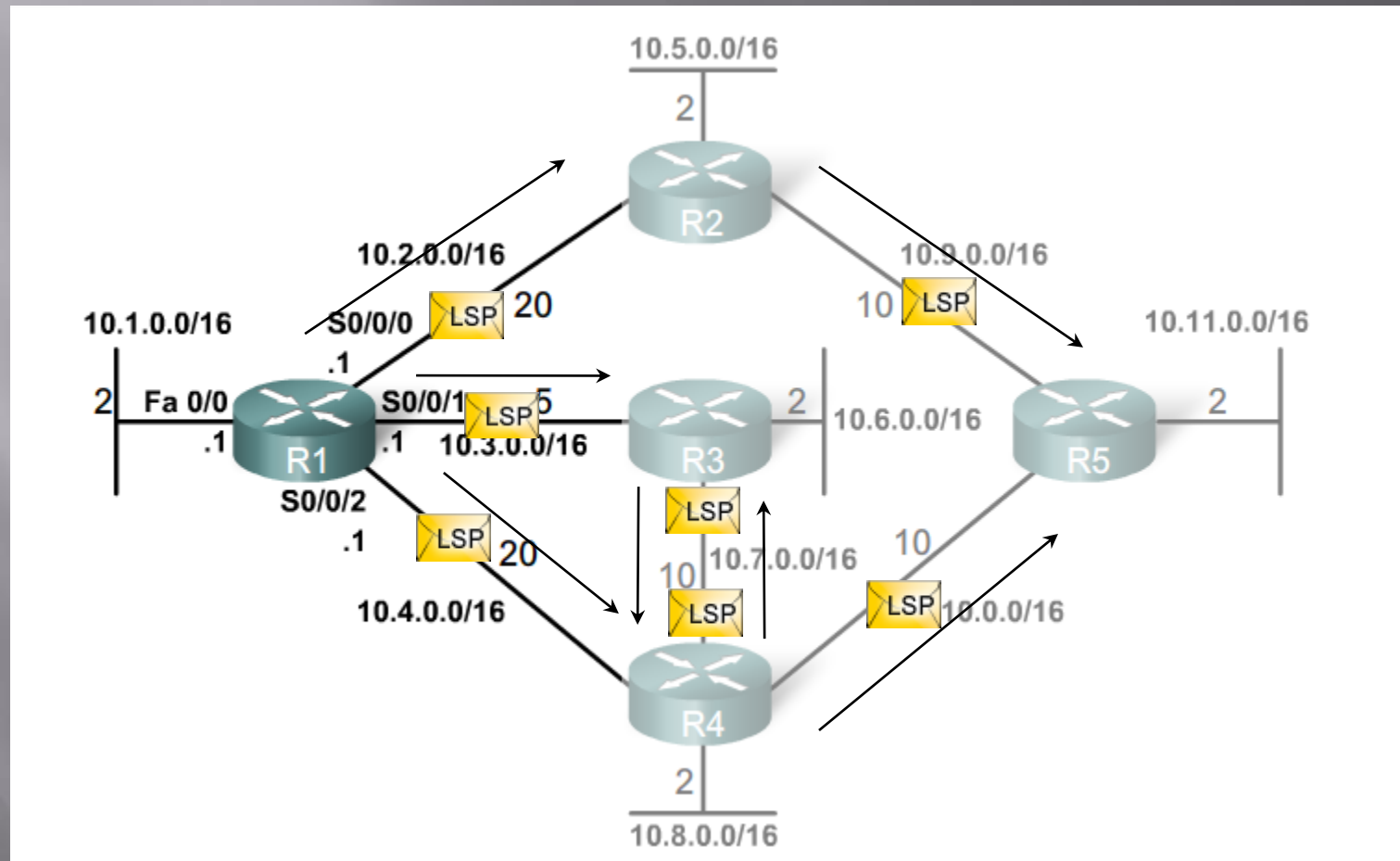
# Wprowadzenie do algorytmu SPF

4. Każdy router zalewowo wysyła pakiety LSP do swoich sąsiadów. Otrzymane dane zapisuje w bazie danych.



# Wprowadzenie do algorytmu SPF

4. Każdy router zalewowo wysyła pakiety LSP do swoich sąsiadów. Otrzymane dane zapisuje w bazie danych.

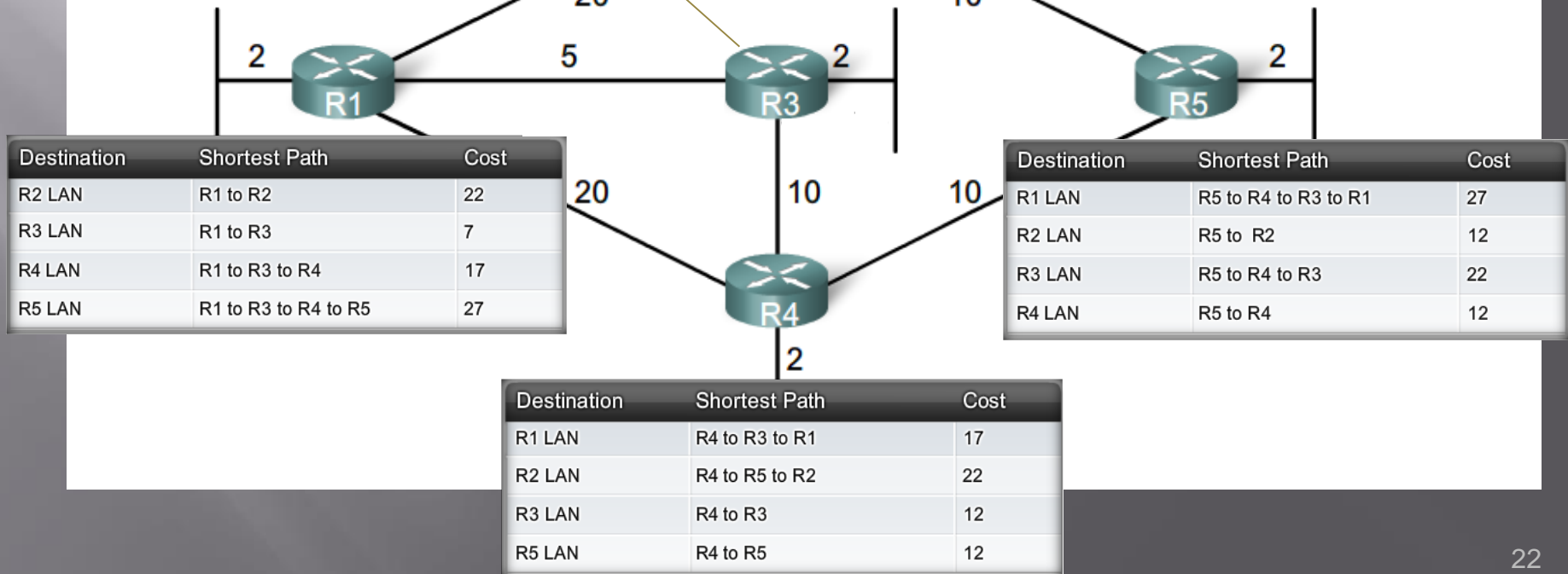


# Wprowadzenie do algorytmu SPF

5. Każdy router używa bazy danych do skonstruowania kompletnej mapy topologii i oblicza najlepszą trasę do każdej sieci.

Destination	Shortest Path	Cost
R1 LAN	R3 to R1	7
R2 LAN	R3 to R1 to R2	27
R4 LAN	R3 to R4	12
R5 LAN	R3 to R4 to R5	22

Destination	Shortest Path	Cost
R1 LAN	R2 to R1	22
R3 LAN	R2 to R1 to R3	27
R4 LAN	R2 to R5 to R4	22
R5 LAN	R2 to R5	12



# Wprowadzenie do algorytmu SPF

## R1 Link State Database

### R1 Links-states:

- Connected to neighbor R2 on network 10.2.0.0/16, cost of 20
- Connected to neighbor R3 on network 10.3.0.0/16, cost of 5
- Connected to neighbor R4 on network 10.4.0.0/16, cost of 20
- Has a network 10.1.0.0/16, cost of 2

### LSPs from R2:

- Connected to neighbor R1 on network 10.2.0.0/16, cost of 20
- Connected to neighbor R5 on network 10.9.0.0/16, cost of 10
- Has a network 10.5.0.0/16, cost of 2

### LSPs from R3:

- Connected to neighbor R1 on network 10.3.0.0/16, cost of 5
- Connected to neighbor R4 on network 10.7.0.0/16, cost of 10
- Has a network 10.6.0.0/16, cost of 2

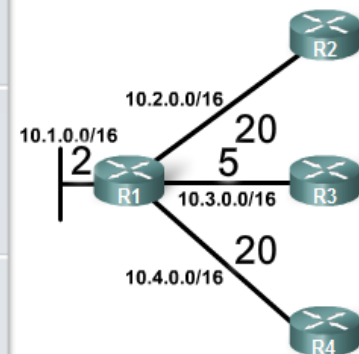
### LSPs from R4:

- Connected to neighbor R1 on network 10.4.0.0/16, cost of 20
- Connected to neighbor R3 on network 10.7.0.0/16, cost of 10
- Connected to neighbor R5 on network 10.10.0.0/16, cost of 10
- Has a network 10.8.0.0/16, cost of 2

### LSPs from R5:

- Connected to neighbor R2 on network 10.9.0.0/16, cost of 10
- Connected to neighbor R4 on network 10.10.0.0/16, cost of 10
- Has a network 10.11.0.0/16, cost of 2

## R1 Link States





# Wprowadzenie do algorytmu SPF

## R1 Link State Database

### R1 Links-states:

- Connected to neighbor R2 on network 10.2.0.0/16, cost of 20
- Connected to neighbor R3 on network 10.3.0.0/16, cost of 5
- Connected to neighbor R4 on network 10.4.0.0/16, cost of 20
- Has a network 10.1.0.0/16, cost of 2

### LSPs from R2:

- Connected to neighbor R1 on network 10.2.0.0/16, cost of 20
- Connected to neighbor R5 on network 10.9.0.0/16, cost of 10
- Has a network 10.5.0.0/16, cost of 2

### LSPs from R3:

- Connected to neighbor R1 on network 10.3.0.0/16, cost of 5
- Connected to neighbor R4 on network 10.7.0.0/16, cost of 10
- Has a network 10.6.0.0/16, cost of 2

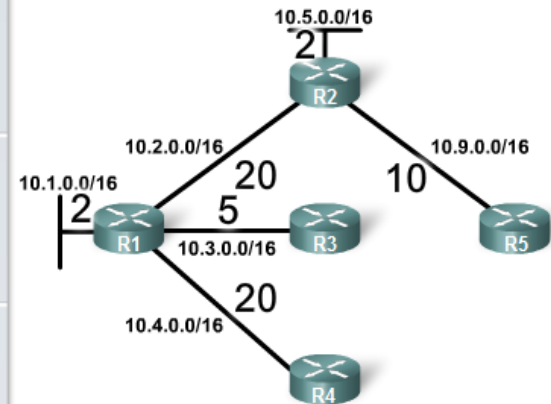
### LSPs from R4:

- Connected to neighbor R1 on network 10.4.0.0/16, cost of 20
- Connected to neighbor R3 on network 10.7.0.0/16, cost of 10
- Connected to neighbor R5 on network 10.10.0.0/16, cost of 10
- Has a network 10.8.0.0/16, cost of 2

### LSPs from R5:

- Connected to neighbor R2 on network 10.9.0.0/16, cost of 10
- Connected to neighbor R4 on network 10.10.0.0/16, cost of 10
- Has a network 10.11.0.0/16, cost of 2

## Processing the R2 LSPs



# Wprowadzenie do algorytmu SPF

## R1 Link State Database

### R1 Links-states:

- Connected to neighbor R2 on network 10.2.0.0/16, cost of 20
- Connected to neighbor R3 on network 10.3.0.0/16, cost of 5
- Connected to neighbor R4 on network 10.4.0.0/16, cost of 20
- Has a network 10.1.0.0/16, cost of 2

### LSPs from R2:

- Connected to neighbor R1 on network 10.2.0.0/16, cost of 20
- Connected to neighbor R5 on network 10.9.0.0/16, cost of 10
- Has a network 10.5.0.0/16, cost of 2

### LSPs from R3:

- Connected to neighbor R1 on network 10.3.0.0/16, cost of 5
- Connected to neighbor R4 on network 10.7.0.0/16, cost of 10
- Has a network 10.6.0.0/16, cost of 2

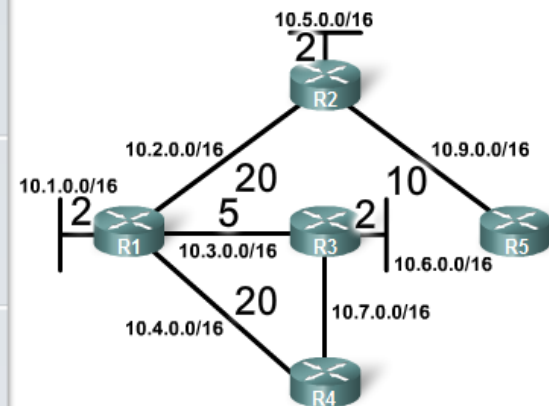
### LSPs from R4:

- Connected to neighbor R1 on network 10.4.0.0/16, cost of 20
- Connected to neighbor R3 on network 10.7.0.0/16, cost of 10
- Connected to neighbor R5 on network 10.10.0.0/16, cost of 10
- Has a network 10.8.0.0/16, cost of 2

### LSPs from R5:

- Connected to neighbor R2 on network 10.9.0.0/16, cost of 10
- Connected to neighbor R4 on network 10.10.0.0/16, cost of 10
- Has a network 10.11.0.0/16, cost of 2

## Processing the R3 LSPs



# Wprowadzenie do algorytmu SPF

## R1 Link State Database

### R1 Links-states:

- Connected to neighbor R2 on network 10.2.0.0/16, cost of 20
- Connected to neighbor R3 on network 10.3.0.0/16, cost of 5
- Connected to neighbor R4 on network 10.4.0.0/16, cost of 20
- Has a network 10.1.0.0/16, cost of 2

### LSPs from R2:

- Connected to neighbor R1 on network 10.2.0.0/16, cost of 20
- Connected to neighbor R5 on network 10.9.0.0/16, cost of 10
- Has a network 10.5.0.0/16, cost of 2

### LSPs from R3:

- Connected to neighbor R1 on network 10.3.0.0/16, cost of 5
- Connected to neighbor R4 on network 10.7.0.0/16, cost of 10
- Has a network 10.6.0.0/16, cost of 2

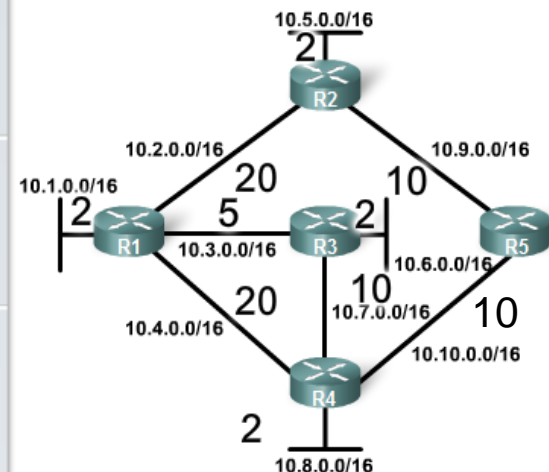
### LSPs from R4:

- Connected to neighbor R1 on network 10.4.0.0/16, cost of 20
- Connected to neighbor R3 on network 10.7.0.0/16, cost of 10
- Connected to neighbor R5 on network 10.10.0.0/16, cost of 10
- Has a network 10.8.0.0/16, cost of 2

### LSPs from R5:

- Connected to neighbor R2 on network 10.9.0.0/16, cost of 10
- Connected to neighbor R4 on network 10.10.0.0/16, cost of 10
- Has a network 10.11.0.0/16, cost of 2

## Processing the R4 LSPs



# Wprowadzenie do algorytmu SPF

## R1 Link State Database

### R1 Links-states:

- Connected to neighbor R2 on network 10.2.0.0/16, cost of 20
- Connected to neighbor R3 on network 10.3.0.0/16, cost of 5
- Connected to neighbor R4 on network 10.4.0.0/16, cost of 20
- Has a network 10.1.0.0/16, cost of 2

### LSPs from R2:

- Connected to neighbor R1 on network 10.2.0.0/16, cost of 20
- Connected to neighbor R5 on network 10.9.0.0/16, cost of 10
- Has a network 10.5.0.0/16, cost of 2

### LSPs from R3:

- Connected to neighbor R1 on network 10.3.0.0/16, cost of 5
- Connected to neighbor R4 on network 10.7.0.0/16, cost of 10
- Has a network 10.6.0.0/16, cost of 2

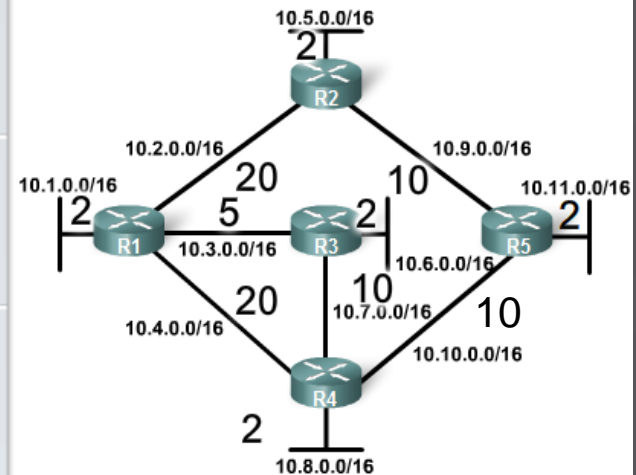
### LSPs from R4:

- Connected to neighbor R1 on network 10.4.0.0/16, cost of 20
- Connected to neighbor R3 on network 10.7.0.0/16, cost of 10
- Connected to neighbor R5 on network 10.10.0.0/16, cost of 10
- Has a network 10.8.0.0/16, cost of 2

### LSPs from R5:

- Connected to neighbor R2 on network 10.9.0.0/16, cost of 10
- Connected to neighbor R4 on network 10.10.0.0/16, cost of 10
- Has a network 10.11.0.0/16, cost of 2

## Processing the R5 LSPs



# Generowanie drzewa SPF i tablicy routingu

R1 Routing Table

## SPF Information

- Network 10.5.0.0/16 via R2 serial 0/0/0 at a cost of 22
- Network 10.6.0.0/16 via R3 serial 0/0/1 at a cost of 7
- Network 10.7.0.0/16 via R3 serial 0/0/1 at a cost of 15
- Network 10.8.0.0/16 via R3 serial 0/0/1 at a cost of 17
- Network 10.9.0.0/16 via R2 serial 0/0/0 at a cost of 30
- Network 10.10.0.0/16 via R3 serial 0/0/1 at a cost of 25
- Network 10.11.0.0/16 via R3 serial 0/0/1 at a cost of 27

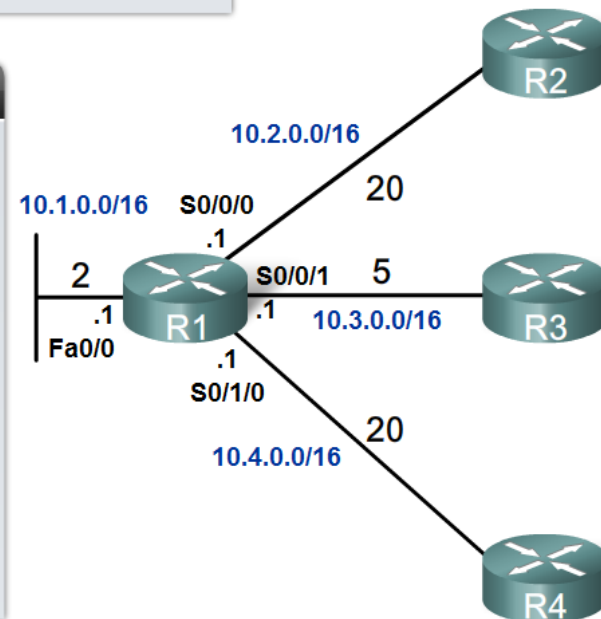
## R1 Routing Table

### Directly Connected Networks

- 10.1.0.0/16 Directly Connected Network
- 10.2.0.0/16 Directly Connected Network
- 10.3.0.0/16 Directly Connected Network
- 10.4.0.0/16 Directly Connected Network

### Remote Networks

- 10.5.0.0/16 via R2 serial 0/0/0, cost = 22
- 10.6.0.0/16 via R3 serial 0/0/1, cost = 7
- 10.7.0.0/16 via R3 serial 0/0/1, cost = 15
- 10.8.0.0/16 via R3 serial 0/0/1, cost = 17
- 10.9.0.0/16 via R2 serial 0/0/0, cost = 30
- 10.10.0.0/16 via R3 serial 0/0/1, cost = 25
- 10.11.0.0/16 via R3 serial 0/0/1, cost = 27



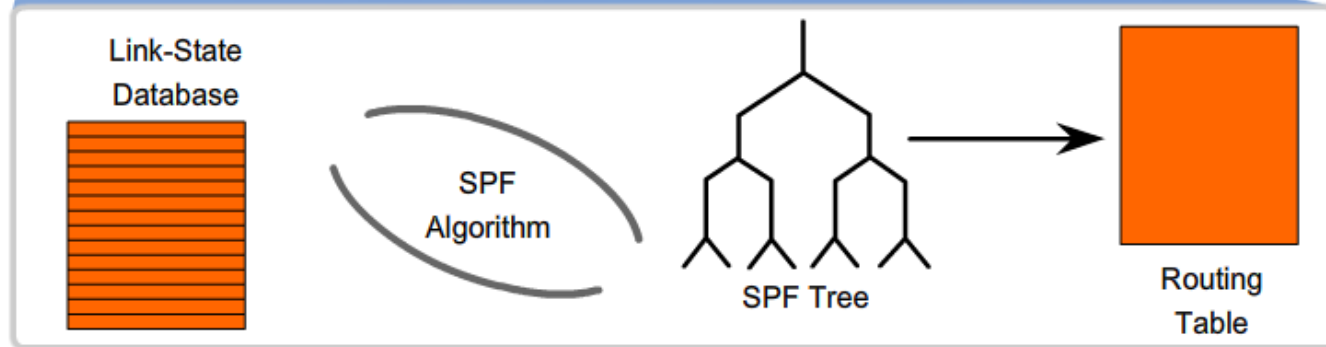
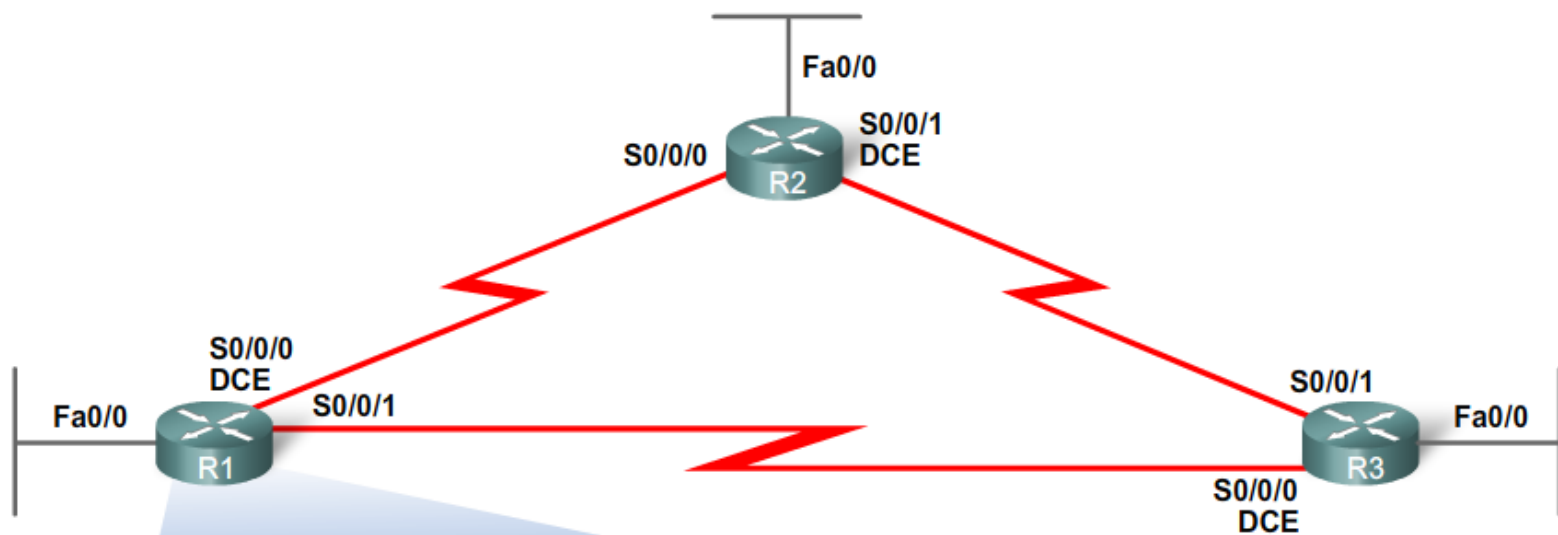


# Algorytm SPF (Shortest Path First)

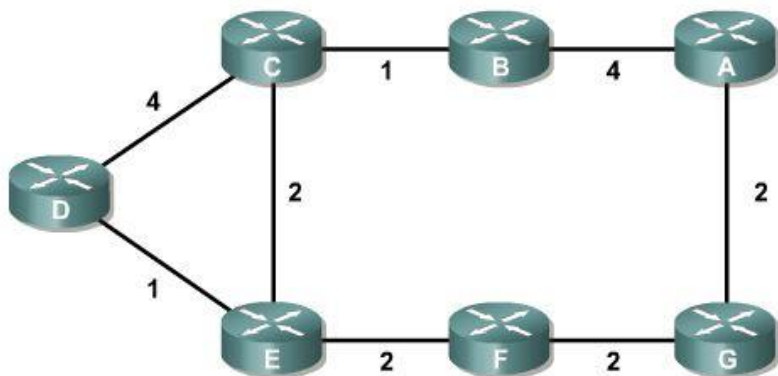
OSPF używa algorytmu SPF by zbudować i wyliczyć najkrótsze trasy do wszystkich znanych sieci.

- Podczas inicjalizacji lub podczas jakiegokolwiek zmiany w sieci, router **generuje pakiet LSA**, który zawiera informacje o wszystkich łączach routera.
- Każdy router, który otrzymuje taki pakiet, wykorzystuje go do uaktualnienia informacji w swojej bazie danych topologii i **przesyła ten pakiet do pozostałych routerów** (*flooding*).
- Na podstawie bazy danych topologii **budowane jest drzewo najkrótszych ścieżek** we wszystkich możliwych dostępnych kierunkach. Na podstawie danych z tego drzewa **konstruowana jest tablica routingu**.
- W przypadku gdy żadna zmiana w sieci się nie pojawia, OSPF praktycznie się nie komunikuje (oprócz wysyłanych regularnie małych pakietów Hello).

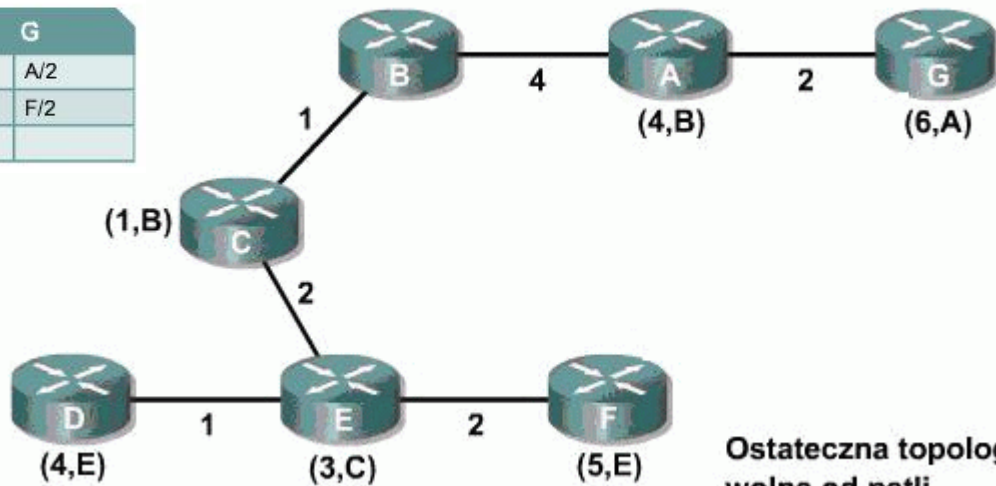
# Algoytm SPF (Shortest Path First)



# Algorytm SPF (Shortest Path First)



A	B	C	D	E	F	G
B/4	A/4	B/1	C/4	C/2	E/2	A/2
G/2	C/1	D/4	E/1	D/1	G/2	F/2
		E/2		F/2		



Ostateczna topologia wolna od pętli

# Pojęcia obszarów i routerów brzegowych w OSPF

- Jednym z najważniejszych elementów standardu OSPF jest koncepcja **obszaru** (*area*) stanowiąca jednocześnie największą trudność przy konfigurowaniu routerów OSPF (wykorzystanie protokołu OSPF w dużych, skalowalnych sieciach wymaga uwzględnienia tego faktu już na etapie ich projektowania).
- Sieć OSPF można podzielić na obszary, które są łączone za pomocą **routerów brzegowych** (ABR - *Area Border Router*).
- Zadanie routerów brzegowych polega na uogólnieniu tras, które są przekazywane poza dany obszar.
- Dzięki takiemu rozwiązaniu routery z jednego obszaru nie muszą przetwarzać danych LSA istotnych dla routerów innego obszaru, co znacznie zwiększa **stabilność** pracy sieci i **skraca czas konwergencji**. Dodatkowo ograniczeniu ulega wykorzystanie pamięci oraz procesora, niezbędnych do obsługi protokołu OSPF w routerze.

# Pojęcia obszarów i routerów brzegowych w OSPF

- ▣ Poprawna praca protokołu OSPF zależy od **właściwego przydziału adresów IP** dla poszczególnych obszarów sieci.
- ▣ Konieczne jest bowiem **uogólnienie tras** do danego obszaru przy przekazywaniu ich do innego obszaru sieci.
- ▣ Wynikiem procedury uogólniania nie musi być pojedyncza trasa do danego obszaru, ale należy pamiętać, że im mniej komunikatów LSA trzeba rozesłać między obszarami, tym lepszy będzie efekt skalowania OSPF.



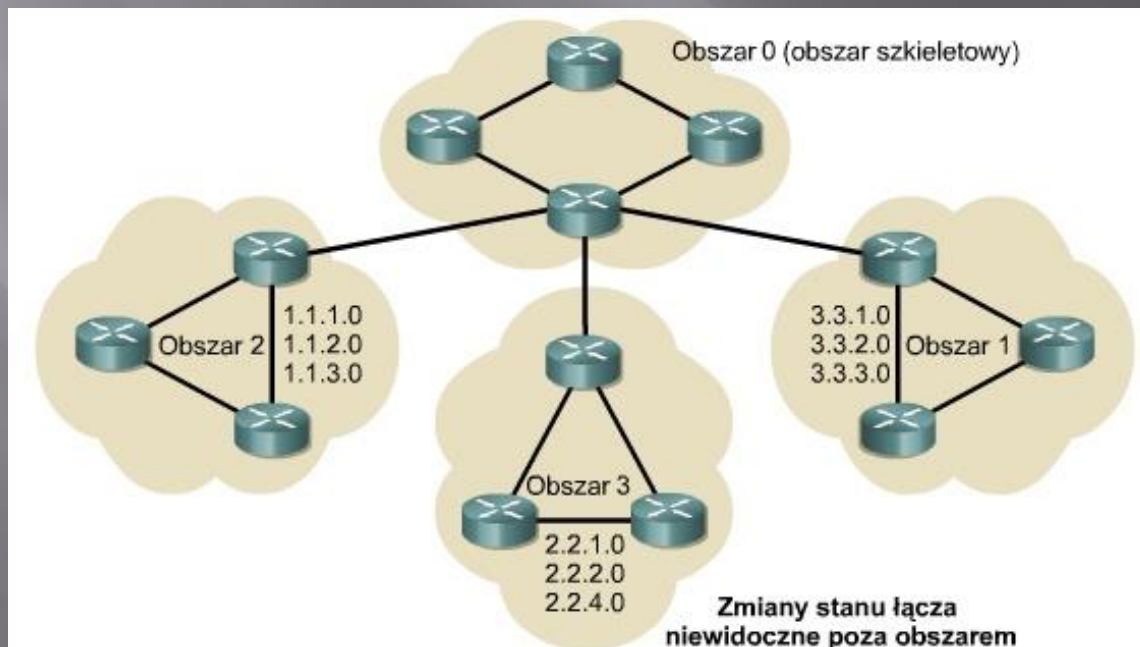
# Pojęcia obszarów i routerów brzegowych w OSPF

- ❑ Każdy obszar ma **32-bitowy identyfikator**, który często jest zapisywany w postaci dziesiętnej z poszczególnymi bitami rozdzielonymi kropkami (tak samo jak adresy IP).
- ❑ Każda sieć OSPF musi zawierać obszar o identyfikatorze 0 (*Area 0*) lub 0.0.0.0, tzw. **obszar szkieletowy**, a każdy router ABR musi być należeć do obszaru 0.
- ❑ Rozwiązanie to wymusza **hierarchiczny** projekt sieci OSPF.
- ❑ Jedyne odstępstwo od tej grupy dotyczy sieci złożonych z jednego obszaru. W takim wypadku obszar ten może mieć dowolny identyfikator, choć nie zaleca się nadawania mu wartości innych niż zero, gdyż może to utrudnić ewentualne późniejsze wydzielenie innych obszarów sieci.
- ❑ Wszystkie routery w ramach jednego obszaru mają wspólną bazę danych topologii. **Rozgłaszanie** pakietów LSA i wyliczanie tras za pomocą algorytmu SPF **jest ograniczone do obszarów**.



# Pojęcia obszarów i routerów brzegowych w OSPF

- Można wprowadzić następujący podział routerów:
  - routery wewnętrzne** (IR – *Internal Router*) – to routery, których wszystkie interfejsy należą do jednego obszaru;
  - routery brzegowe** (ABR) – to routery, których interfejsy należą do różnych obszarów;
  - routery brzegowe systemu autonomicznego** (ASBR – *Autonomous System Boundary Router*) – to routery, które pełnią funkcje bram (redystrybucja) pomiędzy OSPF a innymi protokołami routingu (IGRP, EIGRP, RIP, BGP, statyczny) lub innymi instancjami routingu OSPF.



# Sąsiedzi w OSPF

- ▣ Routery, które dzielą wspólny segment sieci mogą stać się sąsiadami. Pakiety Hello pomagają wykryć sąsiednie routery. Routery stają się sąsiadami jeżeli informacje o nich są wymieniane w pakietach Hello.
- ▣ Następujące warunki muszą być spełnione by dwa routery mogły zostać sąsiadami:
  - dwa routery muszą mieć **ten sam identyfikator obszaru** na sąsiadujących interfejsach. Oczywiście interfejsy muszą należeć do tej samej podsieci i mieć tą samą maskę;
  - jeśli włączone jest uwierzytelnianie, interfejsy muszą wymieniać **to samo hasło**;
  - **liczniki czasowe muszą się zgadzać**: czas pomiędzy wysłaniem dwóch kolejnych pakietów Hello oraz czas przetrzymania, czyli czas, po którego upływie zgłaszana jest niedostępność routera o ile nie napływa od niego żaden komunikat Hello.

# Pojęcie kosztu – metryka w OSPF

- OSPF korzysta z tak zwanych kosztów, przypisanych do każdego łącza (czyli interfejsu) o maksymalnych wartościach z przedziału: 1...65535. **Koszt** ten domyślnie jest odwrotnie proporcjonalny do szerokości pasma na danym łączu - jest obliczany na podstawie wzoru  $10^8 / \text{przepustowość}$ , gdzie przepustowość jest wyrażona w b/s.

Typ łącza a przepustowość	Koszt
56-kbps Łącze szeregowo	1785
T1 1.544-Mbps Łącze szeregowo	64
E1 2.048-Mbps Łącze szeregowo	48
4-Mbps Token Ring	25
Ethernet 10 Mb/s	10
16-Mbps Token Ring	6
100-Mbps Fast Ethernet , FDDI	1

- W protokole OSPF **rutery rozpowszechniają koszty pojedynczych łączy**, a nie całych tras, tak jak to miało miejsce w RIP czy EIGRP - co za tym idzie nie ma sprecyzowanej maksymalnej wartości kosztu całej trasy.

# Podsumowanie

- ▣ Tworzenie relacji sąsiedzkich, typy pakietów OSPF
- ▣ Komunikat OSPF Hello
- ▣ Sposoby otrzymywania informacji o routingu
- ▣ Zalety i wady routingu według stanu łącza
- ▣ Protokół OSPF — główne cechy
- ▣ Algorytm stanu łącza
- ▣ Pojęcie kosztu – metryka w OSPF
- ▣ Algorytm SPF (Shortest Path First)
- ▣ Pojęcia obszarów i routerów brzegowych w OSPF
- ▣ Sąsiedzi w OSPF