Динамично маршрутизиране с протоколи, следящи състояние на връзките. Протокол OSPF

проф. д-р инж. Венета Алексиева

Основни моменти

- Маршрутизиране със следене на състоянието на връзките
- OSPF
- Hello протокол
- Алгоритъм SPF
- Работа в една или няколко области

Link-State маршрутизиране

- Маршрутизирането със следене на състоянието на връзките използват алгоритъма “shortest path first”(SPF).
- Link-state протоколите са създадени по алгоритъма на Dijkstra SPF .

Classification of Routing Protocols

	Interior Gateway Protocols				Exterior Gateway Protocols
	Distance Vector Routing Protocols		Link State Routing Protocols		Path Vector
Classful	RIP	IGRP			EGP
Classless	RIPv2	EIGRP	OSPFv2	IS-IS	BGPv4
IPv6	RIPng	EIGRP for IPv6	OSPFv3	IS-IS for IPv6	BGPv4 for IPv6

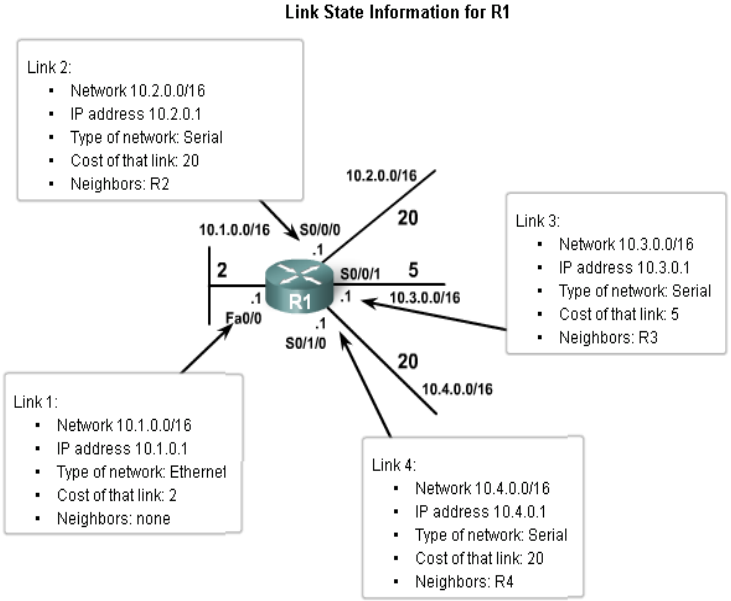
История

- Първият link-state routing protocol е създаден още за ARPANET и той е предшественикът на сегашните.
- По-късно DEC (Digital Equipment Corporation) изобретяват link-state routing protocol за ISO мрежи, който се нарича IS-IS (Intermediate System-to-Intermediate System).
- След това IS-IS е разширен от IETF за IP маршрутизиране.
- **Работна група на IETF измисля протокол, който е специализиран за** IP маршрутизиране – това е OSPF (Open Shortest Path First).
- Използва се алгоритъма на **Dijkstra за изграждане на** SPT *(Shortest Path Tree)*

Постигане на конвергенция

- Всеки Link-State маршрутизатор научава първо директно свързаните си мрежи.
- Link state маршрутизаторите обменят hello пакети с други директно свързани link state маршрутизатори.
- Всеки маршрутизатор създава собствен Link State Packet (LSP), който включва информация за съседите - neighbor ID, link type, bandwidth.
- Процесът на изпращане на LSP продължава, докато всички маршрутизатори вече имат една и съща информация.
- След като всички маршрутизатори получат всички LSP, всеки от тях изгражда собствена карта на топологията на мрежата, която използва за определяне на най-добрия път до всяка от дестинациите.

Понятия



- Директно свързани мрежи
- Link – това е интерфейс на маршрутизатора
- Link state – това е информация за състоянието на интерфейса и неговите връзки със съседните маршрутизатори, като:
  - IP address/mask на интерфейса,
  - тип на мрежата, към която е свързан
  - маршрутизаторите, свързани към тази мрежа
  - metric (cost) на тази връзка
- Link-state database - наборът от всички link-states.

Link-State протоколи

Изисквания:

- Те използват повече памет от distance-vector протоколите
- Те изискват по-мощен процесор от distance-vector протоколите
- Те използват голяма част от bandwidth в момента на началната обмяна на съобщения

За маршрутизиране на IP се използват 2 link-state протокола:

- Open Shortest Path First (OSPFv2; OSPFv3)
- Intermediate System-Intermediate System (IS-IS)

Предимства на link-state протоколите

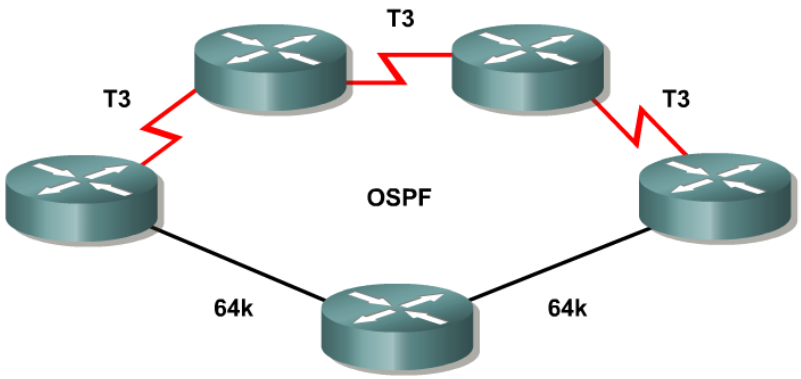
Протокол	Създава ли цялостна “карта” на мрежата	Маршрутизаторите могат сами да определят най-добрия път до всяка мрежа	Конвергенция	Периодични/тригерирани updates	Използва LSP
Distance vector	не	не	бавна	Като цяло - да	не
Link State	да	да	бърза	Като цяло -да	да

OSPF- версии

- За IPv4
  - OSPFv1- RFC1131 отменена, несъвместима с OSPFv2, разликите са описани в RFC1247 от 1991г.
  - OSPFv2- RFC1247 от 1991г., доразвита в RFC2178, RFC 1583 и RFC 2328 от 1998г.
- За IPv6
  - OSPFv3- RFC2740 от 1999г., доразвита в RFC5340 от 2008г.

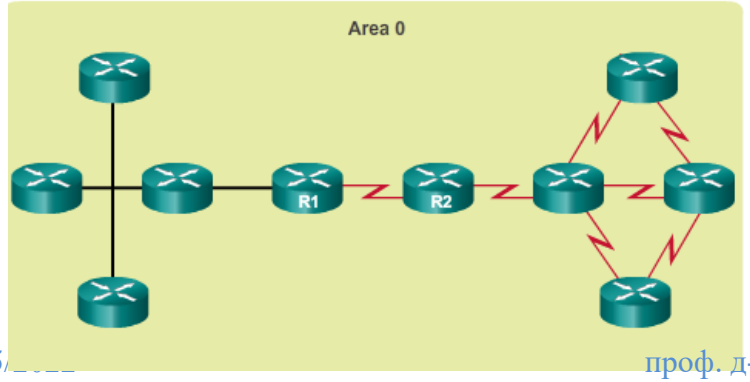
Предимства на OSPF

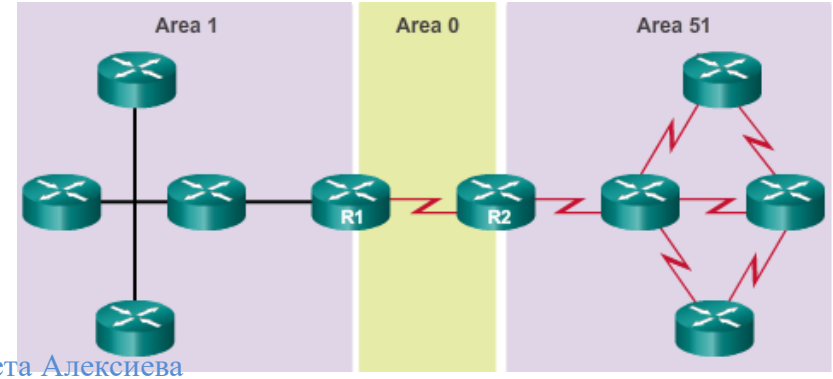
- OSPF е link-state routing protocol
  - RIP, IGRP, EIGRP са distance-vector протоколи, податливи са на routing loops, бавна конвергенция и др.п.
- OSPF постига бърза конвергенция
  - RIP , IGRP работят с hold-down timers и постигат бавна конвергенция.
- OSPF поддържа VLSM и CIDR
  - RIPv1, IGRP не поддържат



Предимства на OSPF

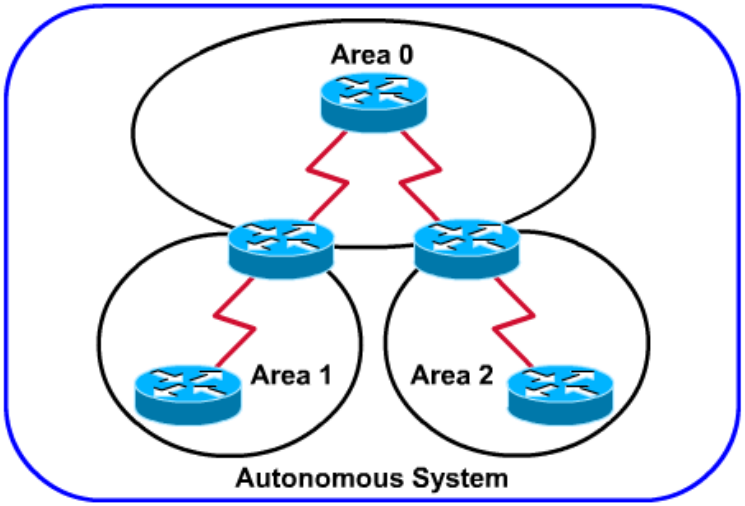
- OSPF метриката е bandwidth
  - на RIP е hop count
  - на IGRP/EIGRP е bandwidth, delay, reliability, load
- OSPF само изпраща промените, ако има такива
  - RIP изпраща routing table на всеки 30s, IGRP -на 90s
  - При OSPF маршрутизаторът изпраща до всички свои LSAs, когато те престоят 30 min без промени
- OSPF работи в области (area) за реализиране на йерархично маршрутизиране





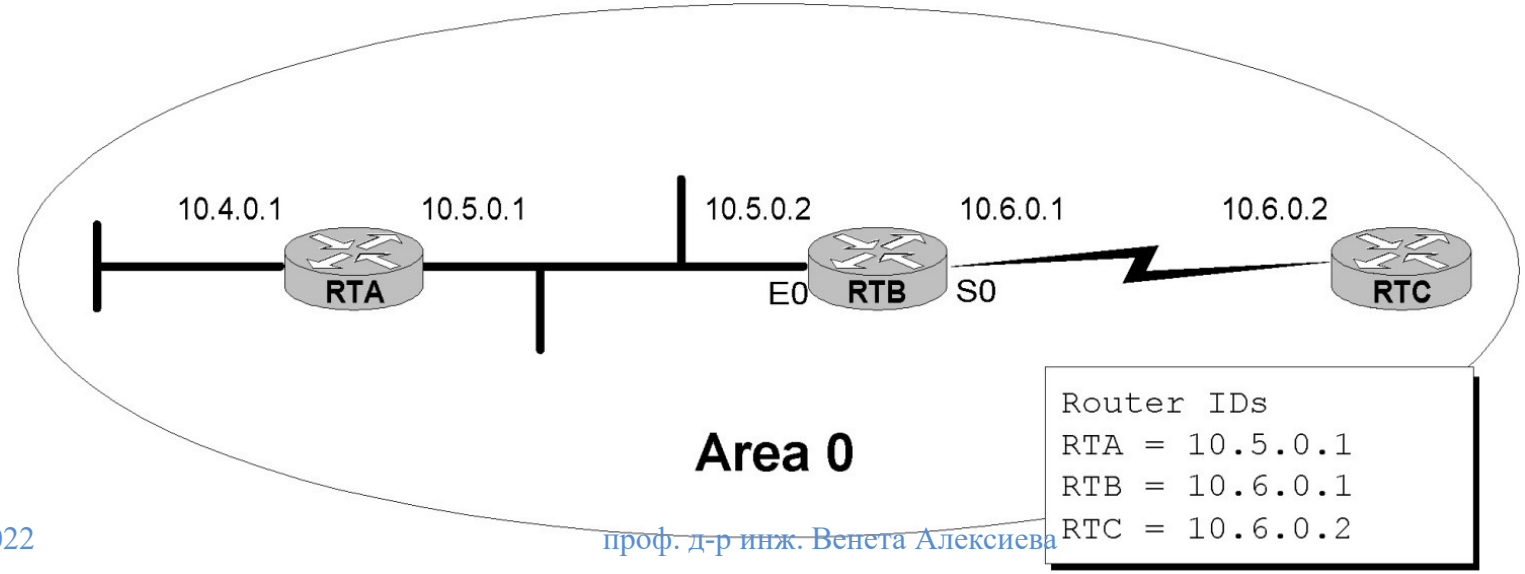
Single Area OSPF и Multi-Area OSPF

- Всички маршрутизатори са в една област - single area, препоръката е това да бъде area 0. Нарича се още backbone area.
- Ако има обаче повече от една област, OSPF задължително трябва да има area 0, към която са свързани останалите области.

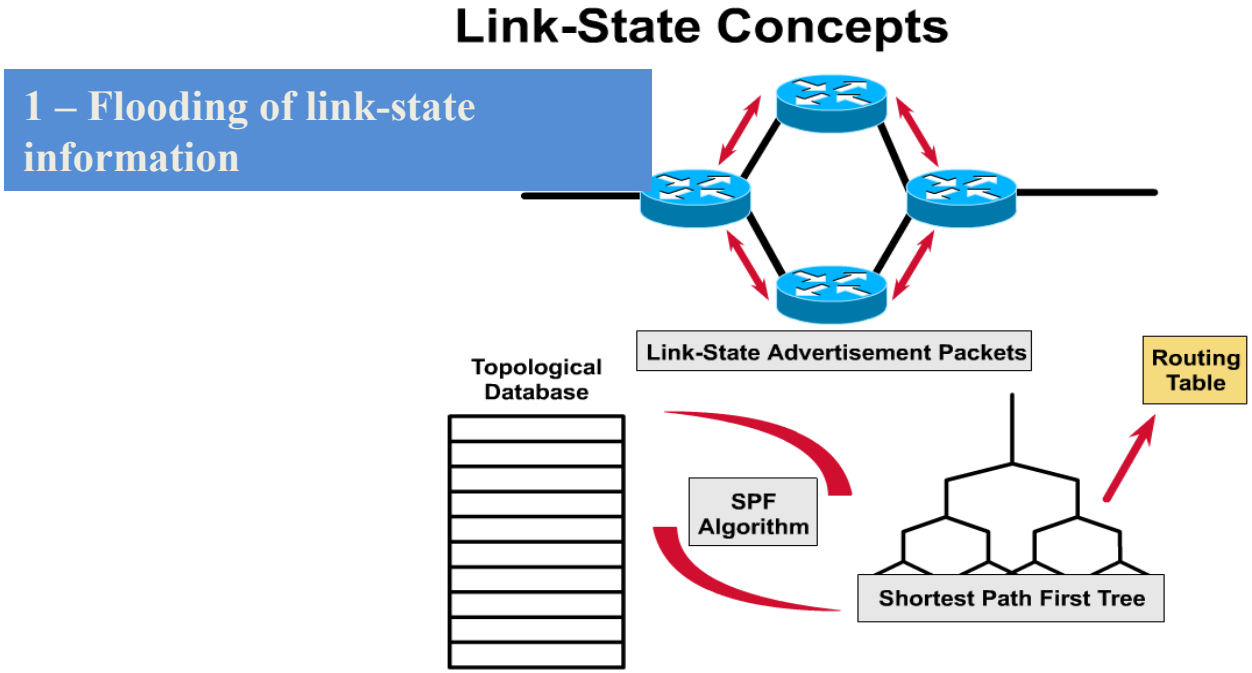


Определяне на Router ID

- Router ID – идентифицира маршрутизаторите в мрежата – това е IP адресът с най-голяма стойност (който и да е IP address - IP на активен интерфейс или loopback)
- Loopback адресът има предимството, че е винаги активен, което осигурява да не се създават често наново съседства

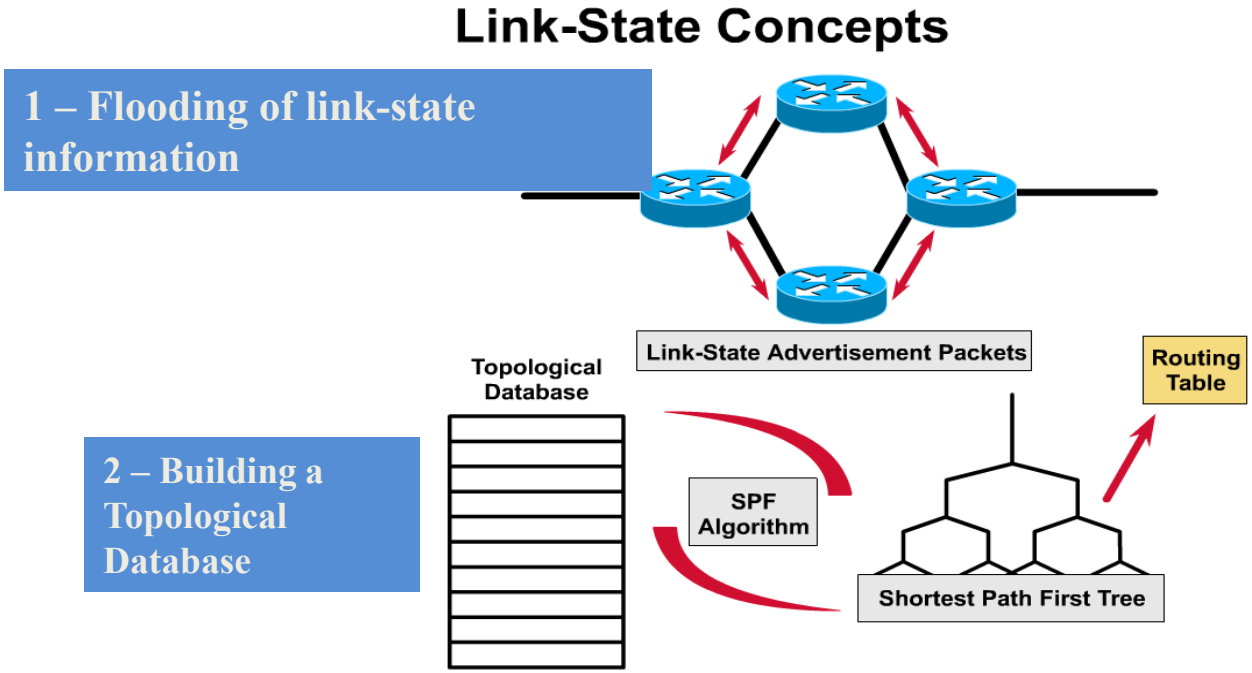


Стъпки на OSPF



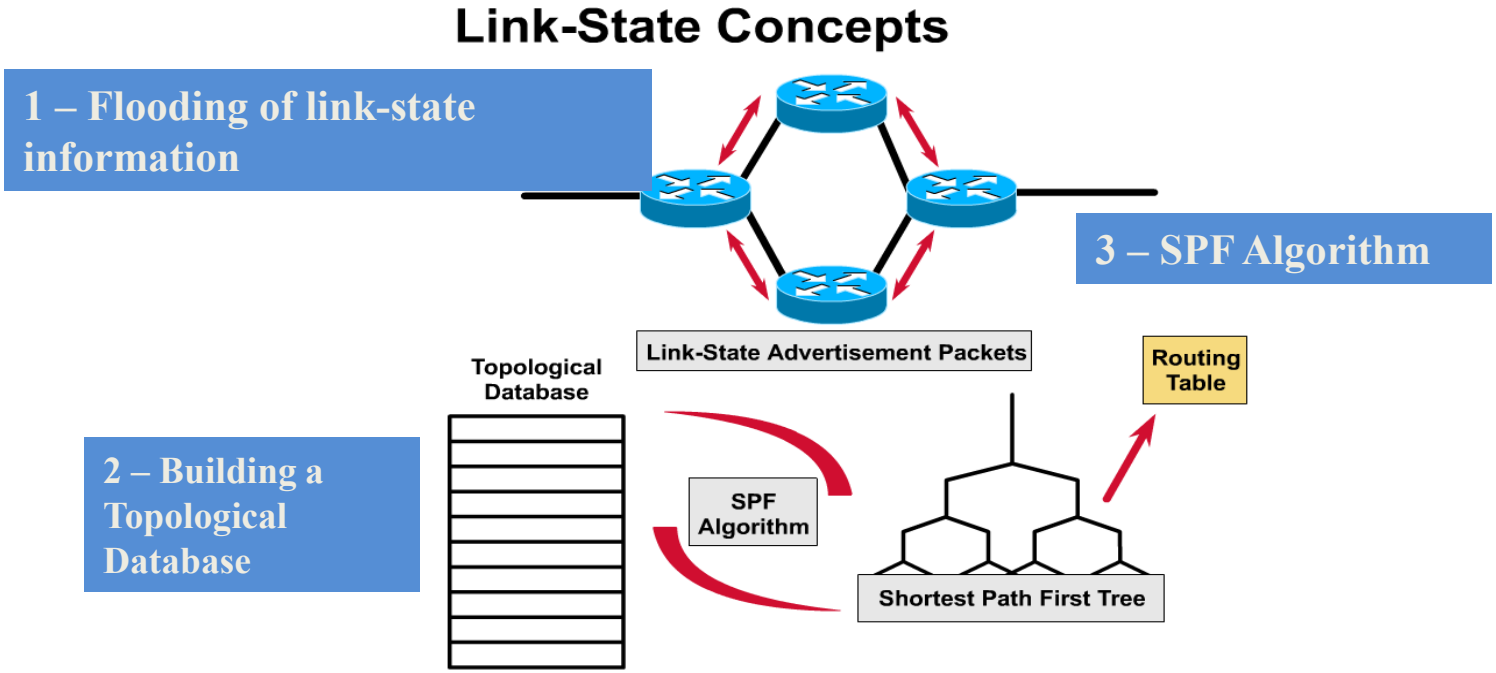
Първо: <u>всеки маршрутизатор</u> в мрежата праща своя порция link-state information <u>до всички други маршрутизатори</u> в мрежата, в която включва кои са съседните му маршрутизатори и cost на връзката с тях.

Стъпки на OSPF



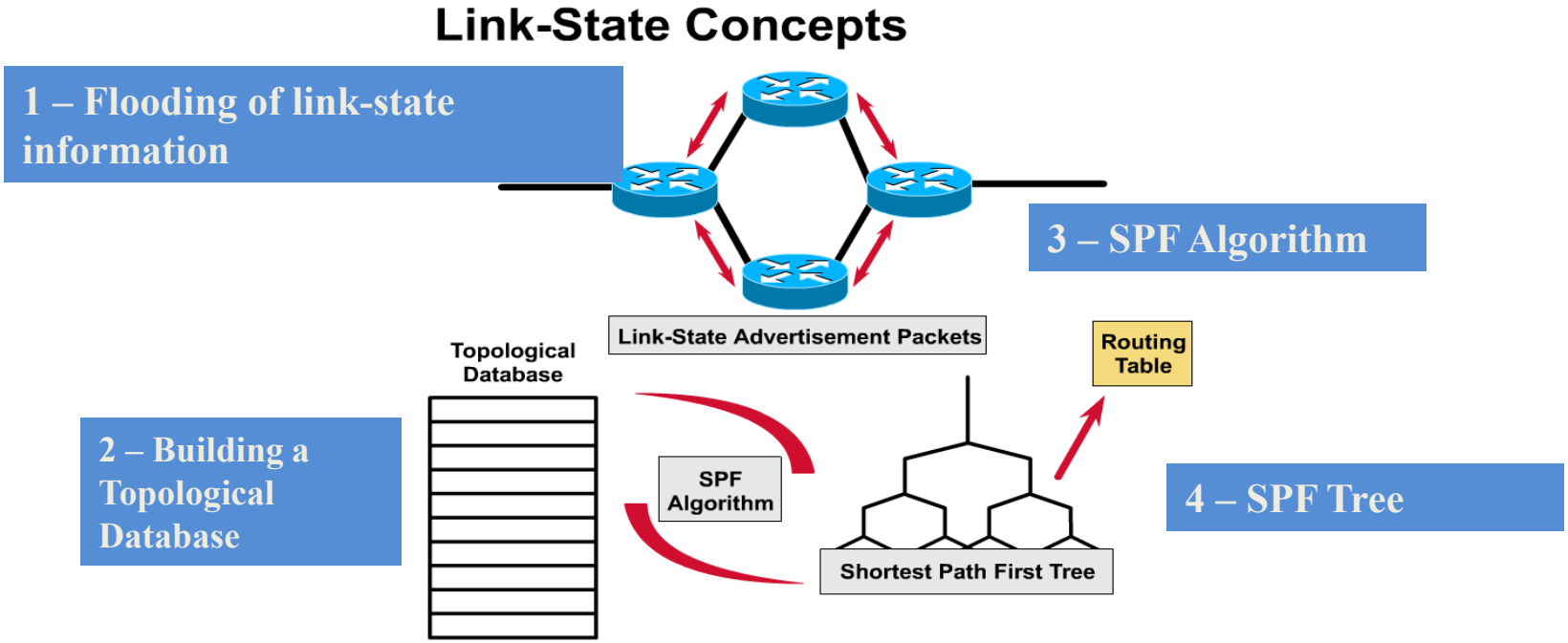
Второ: всеки маршрутизатор събира получените LSA и ги записва в своята topological database.

Стъпки на OSPF



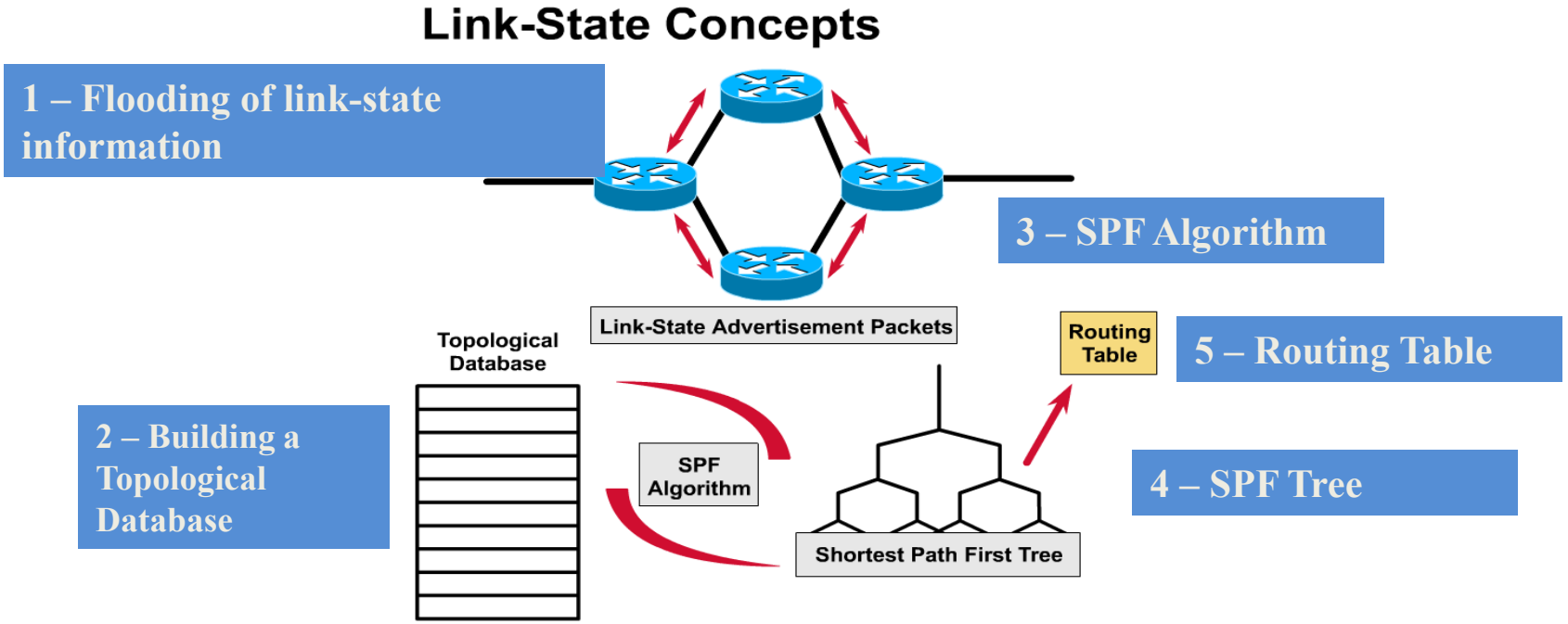
Трето: от получените LSA се прави граф на топологията на мрежата по алгоритъма на Dijkstra – SPF.

Стъпки на OSPF



Четвърто: изгражда се SPF дърво, като маршрутизаторът се поставя в корена му, а другите маршрутизатори и връзките към тях са клоновете му.

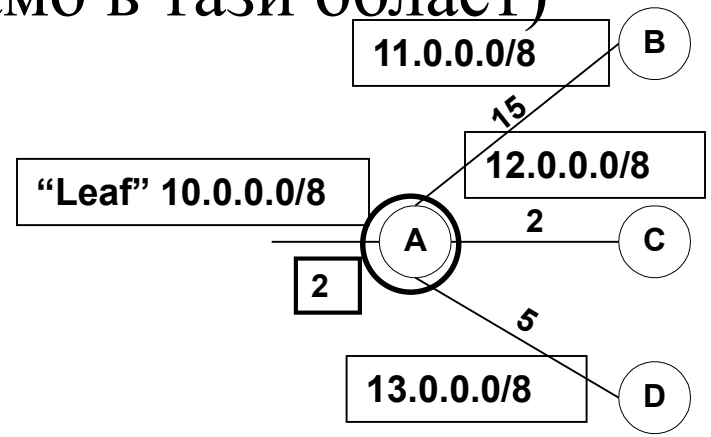
Стъпки на OSPF



Пето: на база на SPF дървото, маршрутизаторът изгражда routing таблицата.

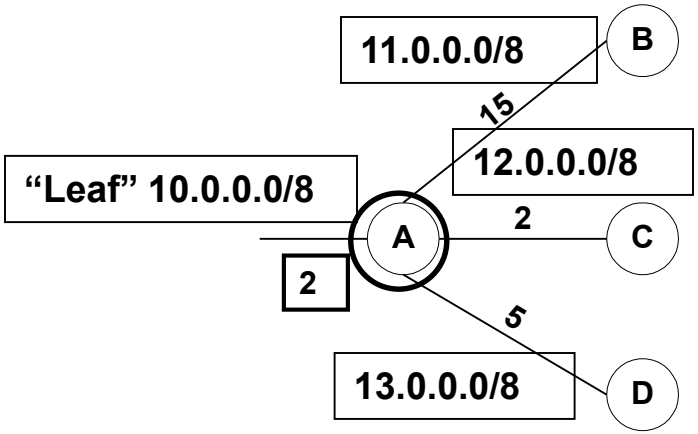
Пример

- **RouterA** обменя “Hellos” с:
  - RouterB по обща мрежа 11.0.0.0/8 с cost 15,
  - RouterC по обща мрежа 12.0.0.0/8 с cost 2
  - RouterD по обща мрежа 13.0.0.0/8 с cost 5
  - Директно свързана мрежа 10.0.0.0/8 с cost 2
- Тази информация той праща на всички други маршрутизатори
- Те също пращат тяхната link state информация. (OSPF: само в тази област)



Пример

Link State Database на маршрутизатор A



RouterB:

- **Свързан с RouterA по 11.0.0.0/8, с cost 15**
- **Свързан с RouterE по 15.0.0.0/8, cost 2**
- **"leaf" network 14.0.0.0/8, cost of 2**

RouterC:

- **Свързан с RouterA по 12.0.0.0/8, cost 2**
- **Свързан с RouterD по 16.0.0.0/8, cost 2**
- **"leaf" network 17.0.0.0/8, cost of 2**

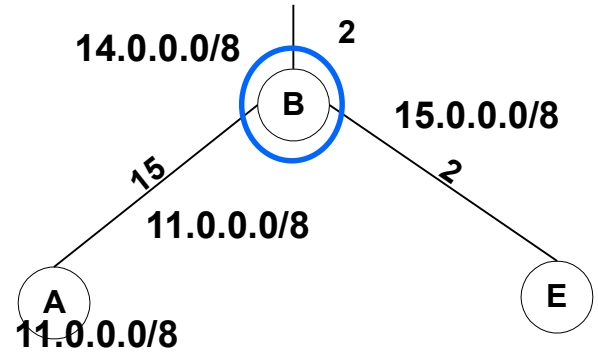
RouterD:

- **Свързан с RouterA по 13.0.0.0/8, cost 5**
- **Свързан с RouterC по 16.0.0.0/8, cost 2**
- **Свързан с RouterE по 18.0.0.0/8, cost 10**
- **"leaf" network 19.0.0.0/8, cost of 2**

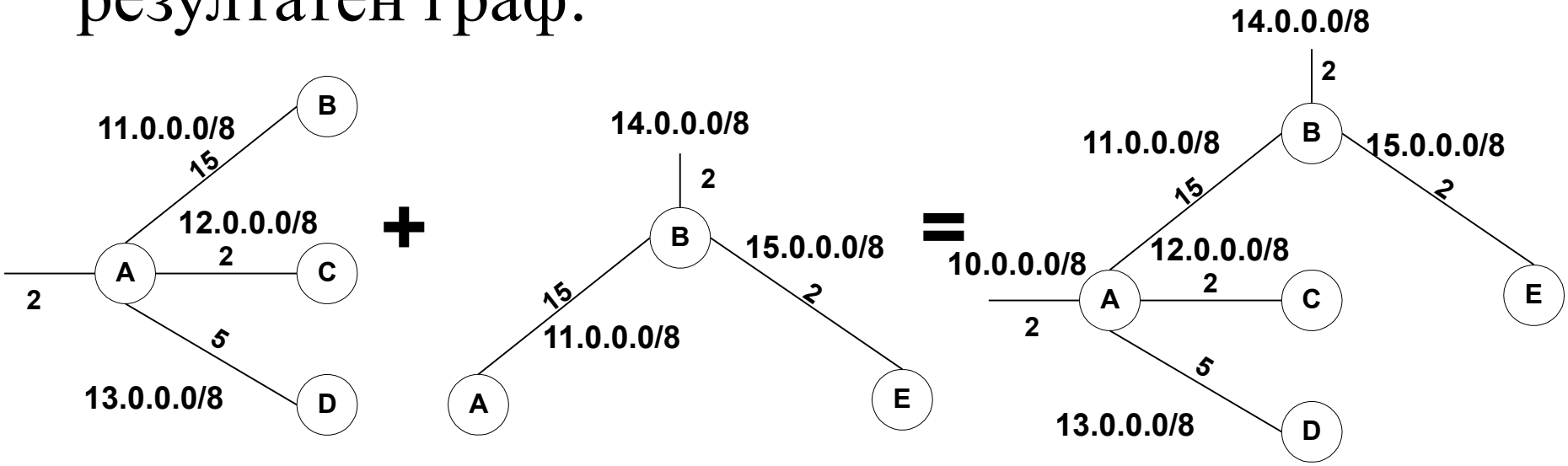
RouterE:

- **Свързан с RouterB по 15.0.0.0/8, cost 2**
- **Свързан с RouterD по 18.0.0.0/8, cost 10**
- **"leaf" network 20.0.0.0/8, cost of 2**

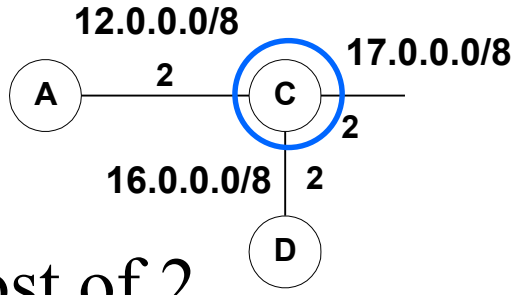
Пример



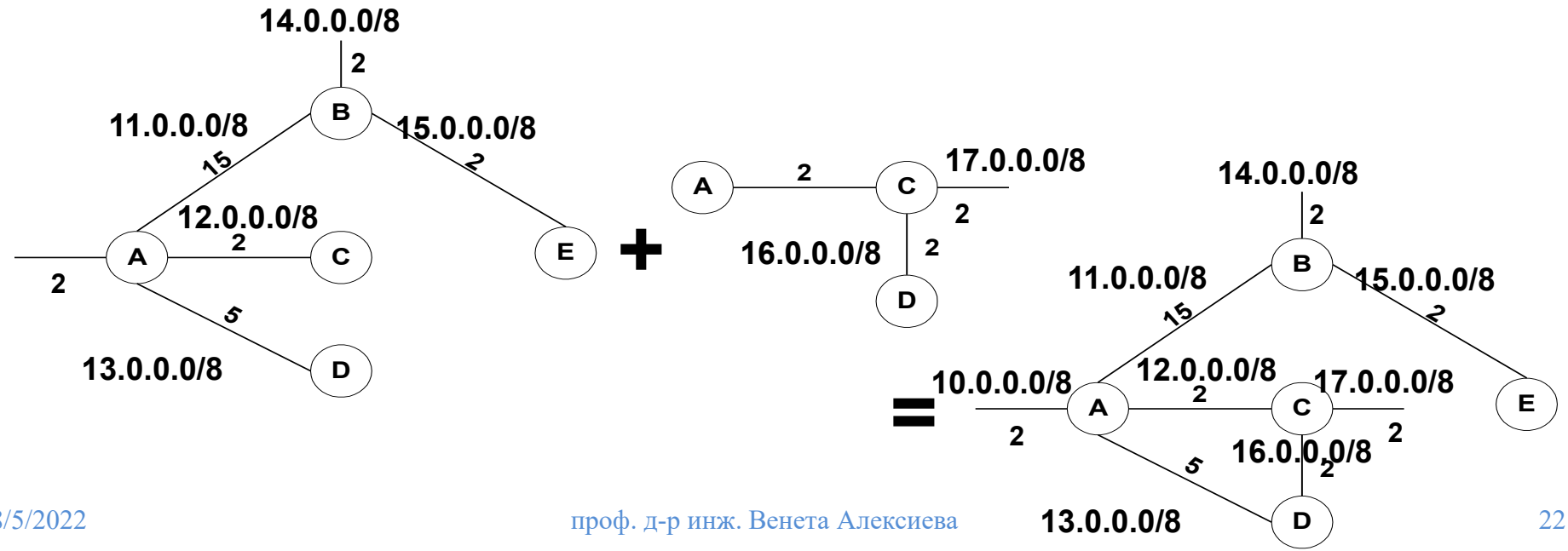
- Информация от RouterB
  - Свързан с RouterA по 11.0.0.0/8, cost of 15
  - Свързан с RouterE по 15.0.0.0/8, cost of 2
  - “leaf” network 14.0.0.0/8, cost of 2
- Така RouterA събира 2-та графа и получава резултатен граф:



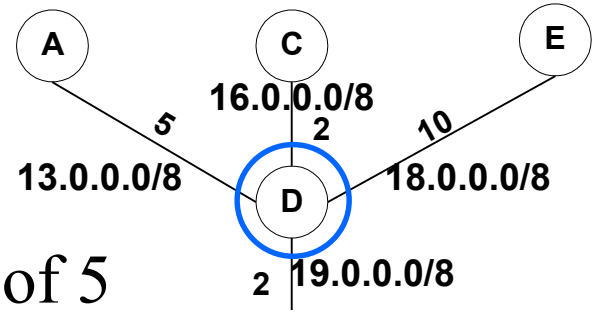
Пример



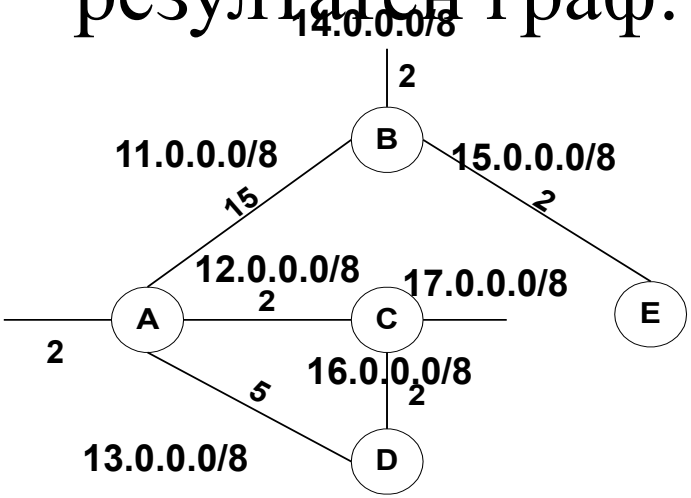
- Информация от RouterC
  - Свързан с RouterA по 12.0.0.0/8, cost of 2
  - Свързан с RouterD по 16.0.0.0/8, cost of 2
  - “leaf” network 17.0.0.0/8, cost of 2
- Така RouterA събира 2-та графа и получава резултатен граф:



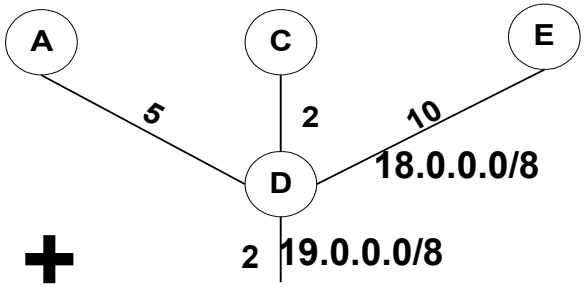
Пример



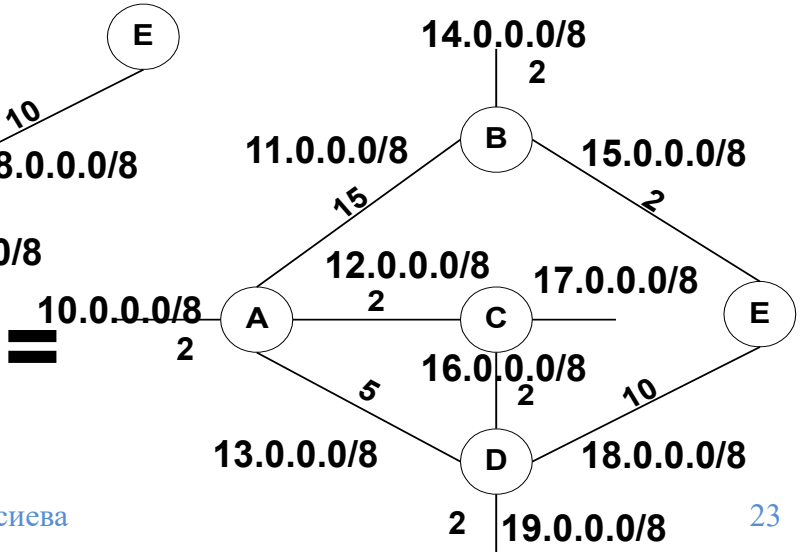
- Информация от RouterD
  - Свързан с RouterA по 13.0.0.0/8, cost of 5
  - Свързан с RouterC по 16.0.0.0/8, cost of 2
  - Свързан с RouterE по 18.0.0.0/8, cost of 10
  - “leaf” network 19.0.0.0/8, cost of 2
- Така RouterA събира 2-та графа и получава резултатен граф:



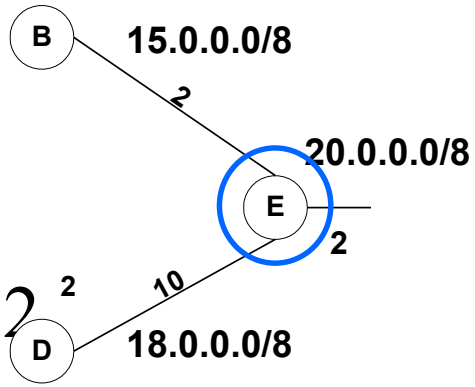
+



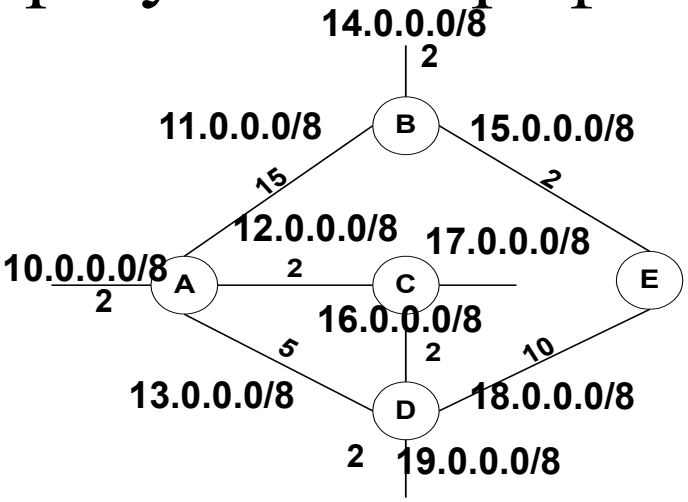
=

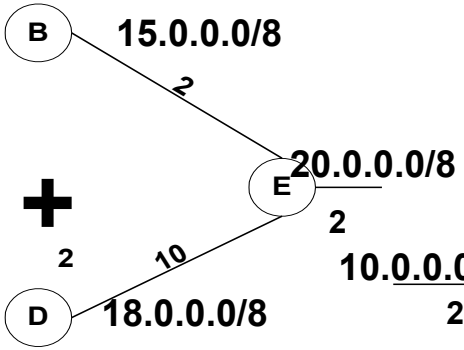


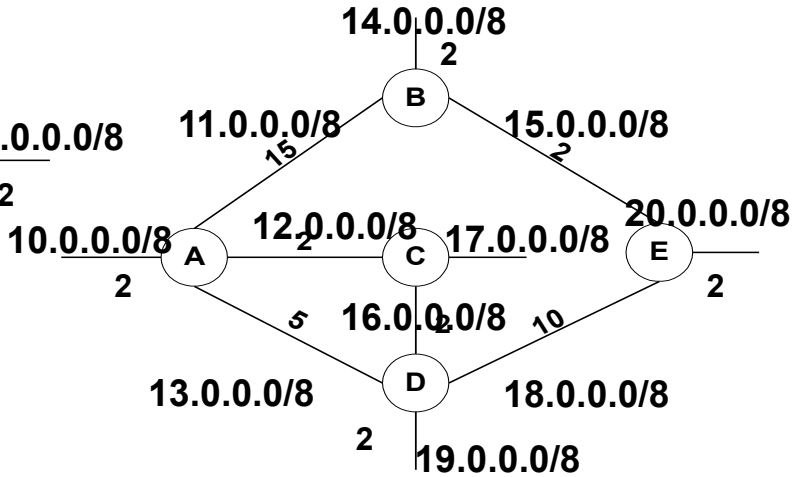
Пример



- Информация от RouterE
  - Свързан с RouterB по 15.0.0.0/8, cost of 2
  - Свързан с RouterD по 18.0.0.0/8, cost of 10
  - “leaf” network 20.0.0.0/8, cost of 2
- Така RouterA събира 2-та графа и получава резултатен граф:

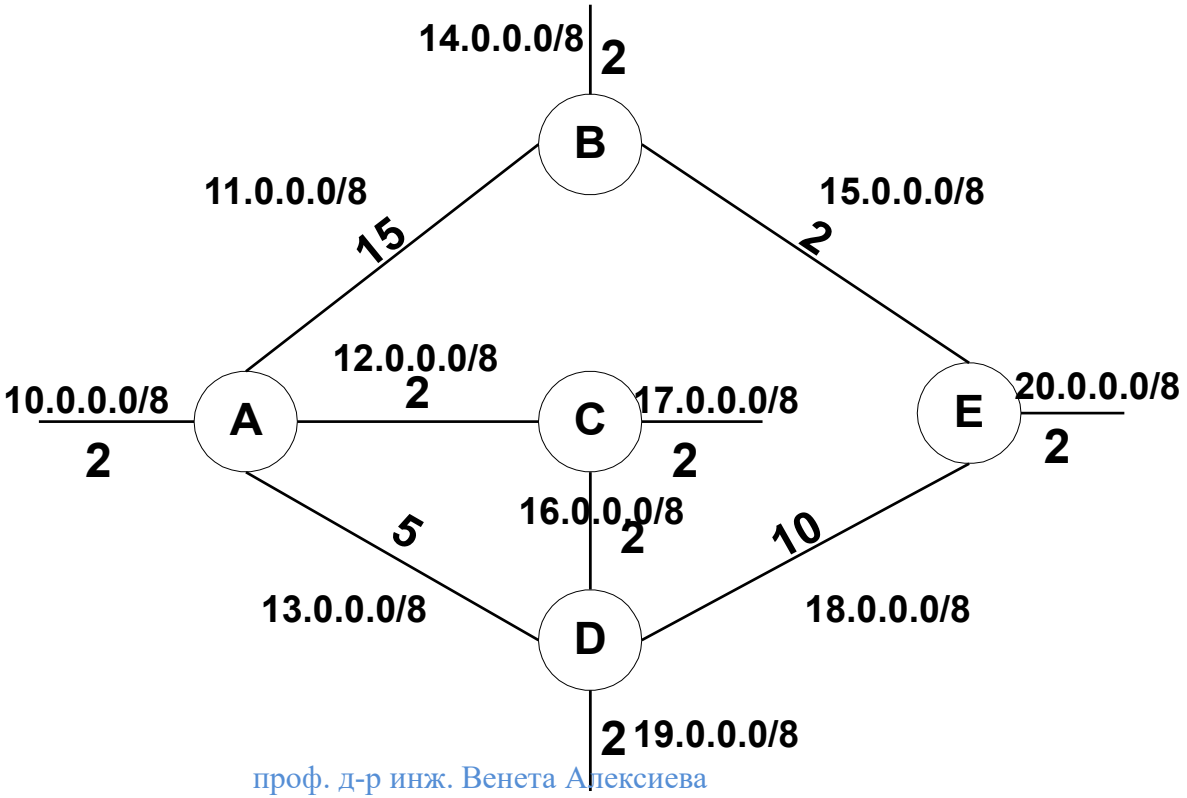






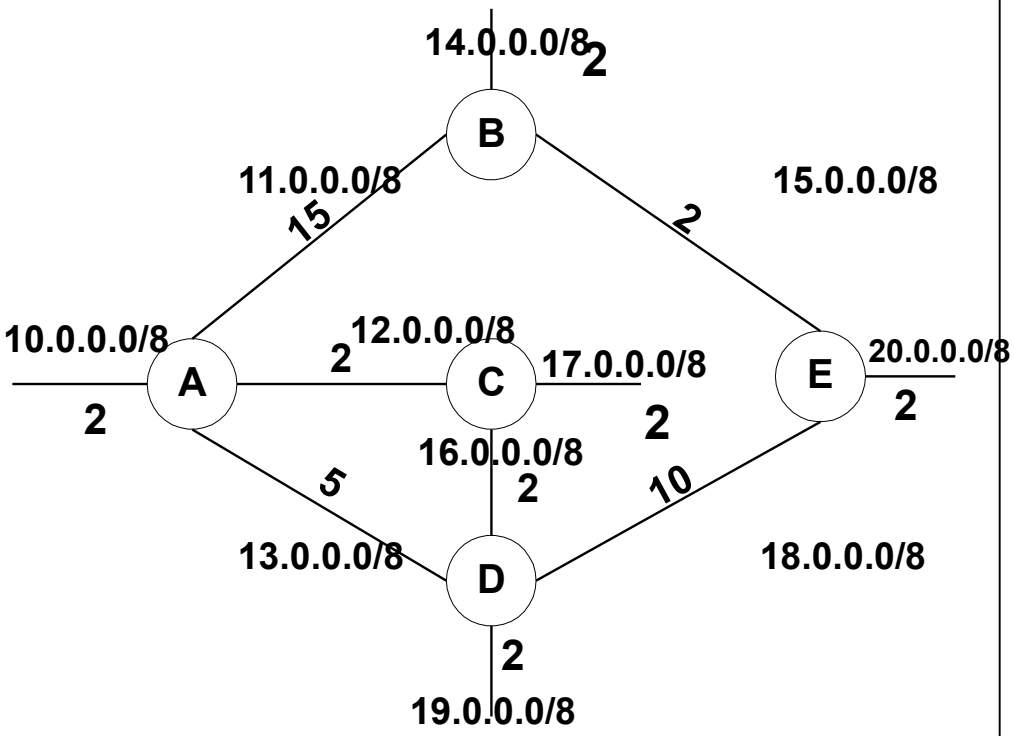
Пример

- Така RouterA е изградил пълна топология на мрежата
- Следващата стъпка е да се открие най-добър път до всяка точка в тази мрежа.



Пример

- Избор на най-добър път



RouterB:
- Свързан с RouterA по 11.0.0.0/8, с cost 15
- Свързан с RouterE по 15.0.0.0/8, cost 2
- “leaf” network 14.0.0.0/8, cost of 2

RouterC:
- Свързан с RouterA по 12.0.0.0/8, cost 2
- Свързан с RouterD по 16.0.0.0/8, cost 2
- “leaf” network 17.0.0.0/8, cost of 2

RouterD:
- Свързан с RouterA по 13.0.0.0/8, cost 5
- Свързан с RouterC по 16.0.0.0/8, cost 2
- Свързан с RouterE по 18.0.0.0/8, cost 10
- “leaf” network 19.0.0.0/8, cost of 2

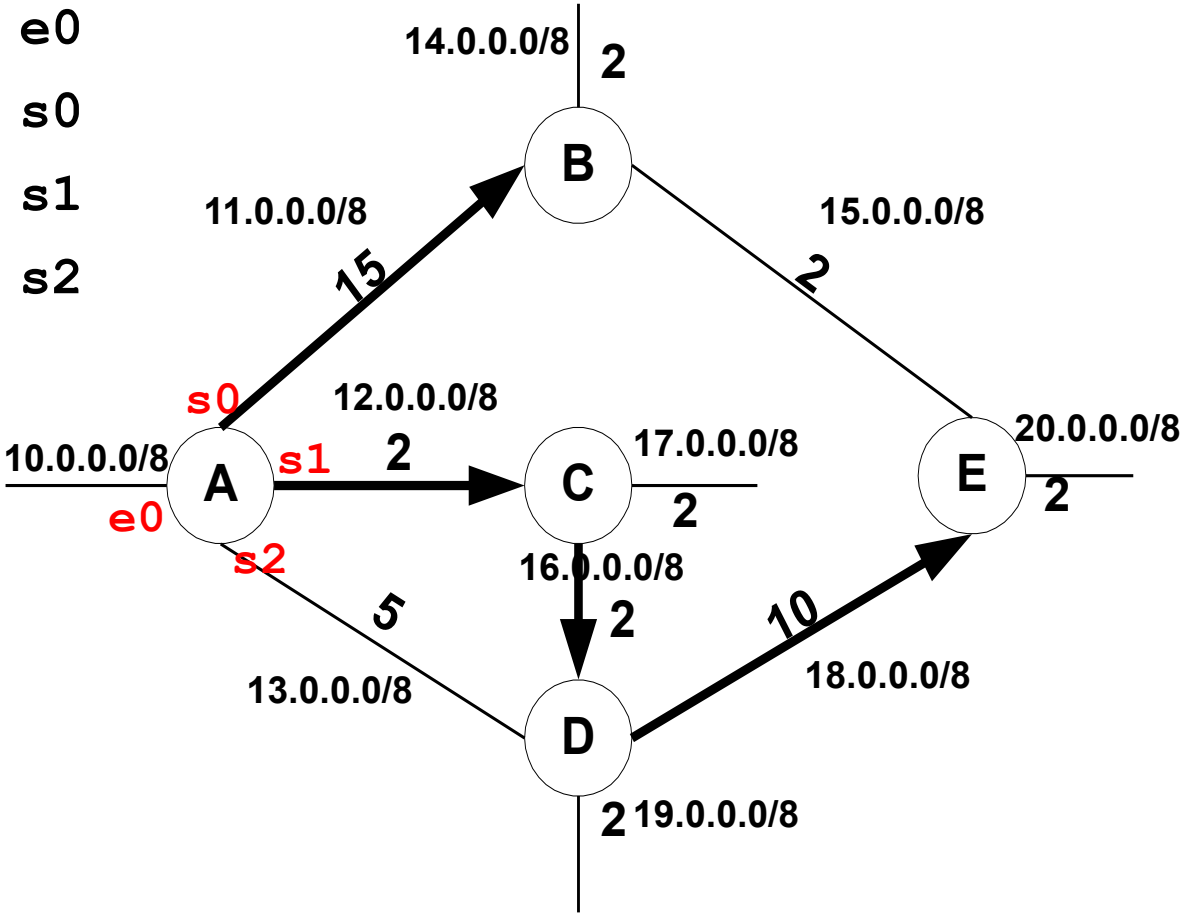
RouterE:
- Свързан с RouterB по 15.0.0.0/8, cost 2
- Свързан с RouterD по 18.0.0.0/8, cost 10
- “leaf” network 20.0.0.0/8, cost of 2

Пример

- SPT дървото се разполага в маршрутната таблица

RouterA's Routing Table

```
10.0.0.0/8 connected e0
11.0.0.0/8 connected s0
12.0.0.0/8 connected s1
13.0.0.0/8 connected s2
14.0.0.0/8  17 s0
15.0.0.0/8  16 s1
16.0.0.0/8   4 s1
17.0.0.0/8   4 s1
18.0.0.0/8  14 s1
19.0.0.0/8   6 s1
20.0.0.0/8  16 s1
```



OSPF метриката е Cost

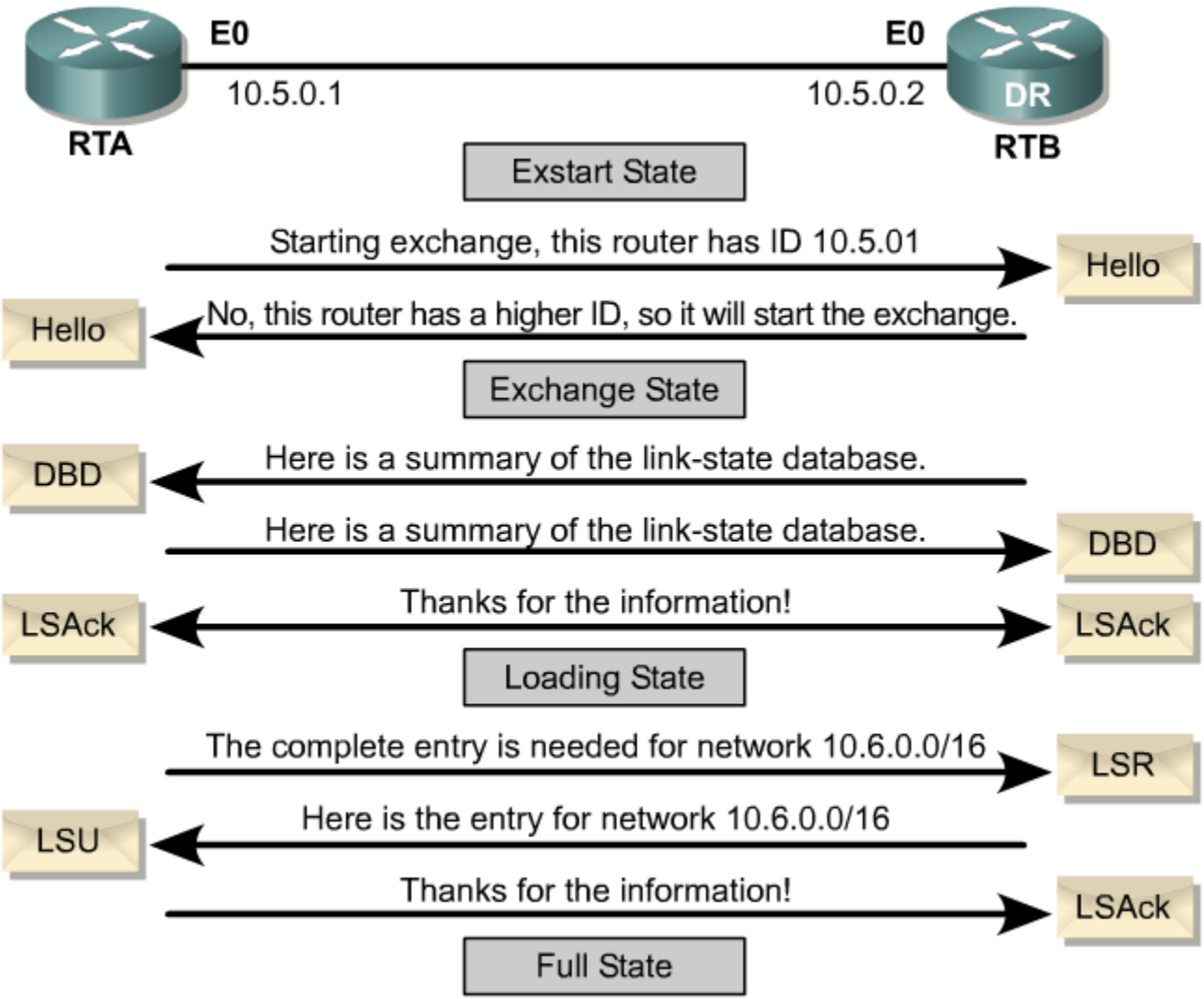
- Метриката за OSPFv2 е описана в RFC 2328, но там не се посочват никакви стойности за “cost”.
- Cost е свързана с изхода на всеки интерфейс на маршрутизатора.
- Тя може да се конфигурира от системния администратор.
- Колкото по-ниска е Cost, по-вероятно е интерфейсът да се използва за прехвърляне на данните.
- Cost на пътя е сумата от costs на изходящите интерфейси от този маршрутизатор към мрежата.
- Някои производители използват default cost = 1 за всички интерфейси (по асоциация с hop count)
- В Linux: Cost се определя от Bandwidth

default cost = 10^8/bandwidth

OSPF типове съобщения

- 1- Hello
  - Установява съседства и обменя информация със съседите
- 2- Database description packet
  - Описва съдържанието на link-state database
- 3- link-state request (LSR)
  - Изисква конкретна информация от link-state database
- 4- link-state update (LSU)
  - Транспортира LSA (advertisments) до съседните маршрутизатори
- 5- link-state acknowledgement (LSAck)
  - Потвърждава получаването на LSA

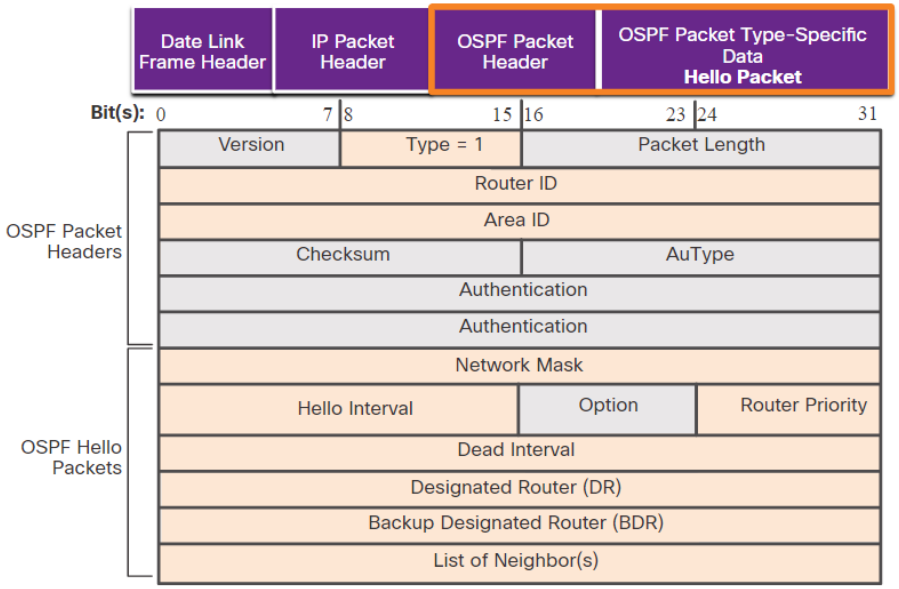
OSPF процес на комуникация



Синхронизиране на OSPF базите данни

- След състоянието Two-Way state, маршрутизаторите синхронизират OSPF базите си с данни (DBD) в 3 стъпки:
  - Избор на първи маршрутизатор: маршрутизаторът с най-голямо ID изпраща своята DBD първи.
  - Обмяна на DBDs: Другият рутер трябва да потвърди всека получена DBD с пакет LSAck .
  - Изпращане на LSR: Всеки рутер сравнява DBD информацията с локалния LSDB. Ако DBD има по-актуална информация за връзката, маршрутизаторът преминава в състояние loading state.
- Изпращат се актуализации (LSU):
  - Когато се възприеме промяна (incremental updates)
  - На всеки 30 минути

OSPF Hello Protocol



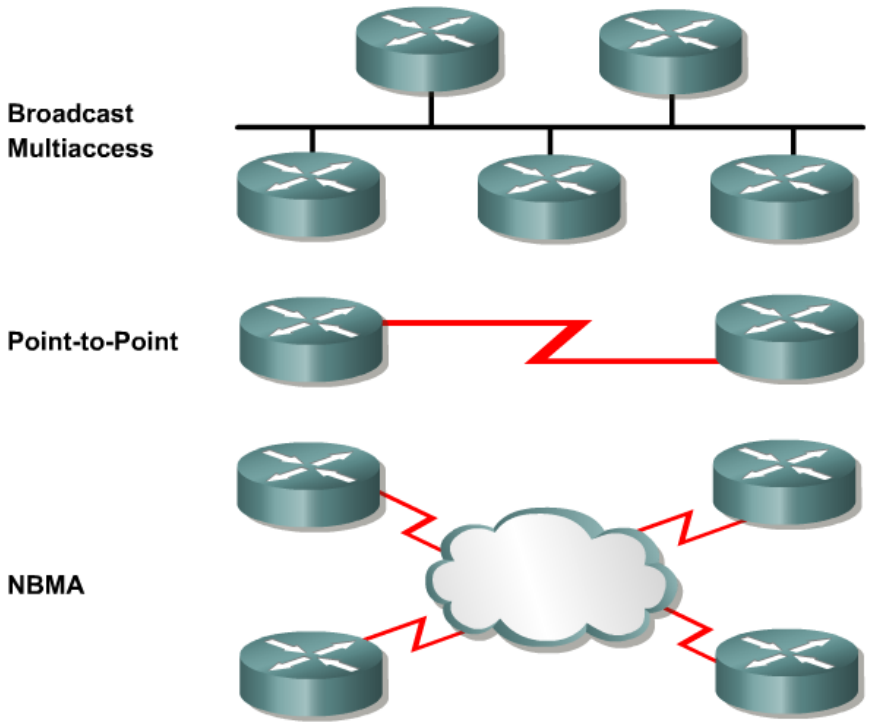
- Динамично открива съседи
- Открива недостижими съседи
- Осигурява two-way комуникация между съседи
- Осигурява коректност на основните параметри на интерфейса между съседи
- Осигурява необходимата информация за избор на Designated and Backup Designated routers (DR & BDR) в един LAN сегмент

Настройки

- OSPF изпраща Hello пакети до всички маршрутизатори по 224.0.0.5
- Hello Interval:
  - 10 сек.(multi-access and point-to-point segments)
  - 30 сек.(NBMA- Frame Relay, X.25, ATM).
- Router Dead Interval – Време в секунди, което маршрутизатора ще чака да получи Hello от съседа, преди да обяви, че той е down.
  - Изчислява се като 4x Hello Interval
- За да създадат съседства, маршрутизаторите трябва да имат еднакви стойности за Hello, Dead Interval и тип на мрежата, в противен случай Hello packets ще бъдат отхвърлени!

Типове мрежи

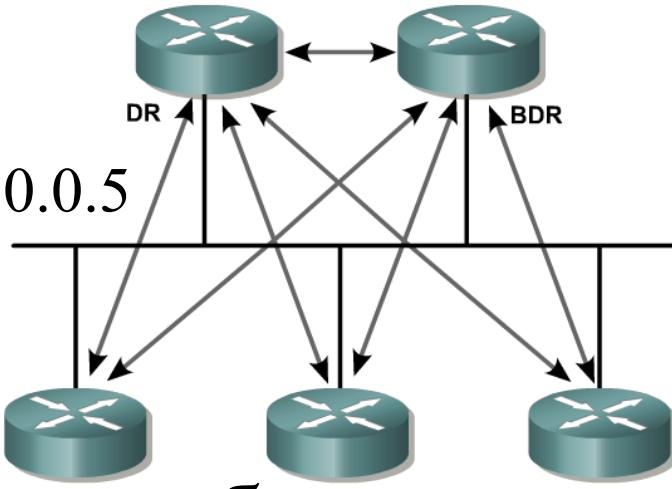
Network Type	Table Title
Broadcast multiaccess	Ethernet, Token Ring, or FDDI
Nonbroadcast multiaccess	Frame Relay, X.25, SMDS
Point-to-point	PPP, HDLC
Point-to-multipoint	Configured by an administrator



Стъпки на OSPF

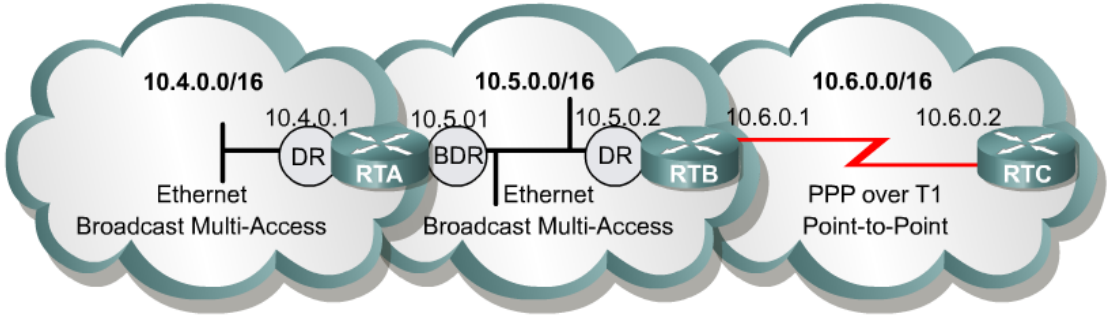
1. Установяване на съседства
2. Избор на Designated Router(DR) и (Backup Designated Router)BDR – само в Multi-access (broadcast) сегменти

- Обменят multicast LSAs по 224.0.0.6
- DR им изпраща до всички LSA по 224.0.0.5



2. Откриване на маршрути
3. Попълване на LSDB
4. Попълване на маршрутната таблица

Пример



- маршрутизаторът с най-голямо **Router ID** става DR, а следващия - BDR.
- Но подобно на други избори, този избор може да бъде нагласен.

  Приоритет на маршрутизатора може да се настрои така, че да се гарантира, че ще се избере за DR или да осигури, че няма да бъде избран.

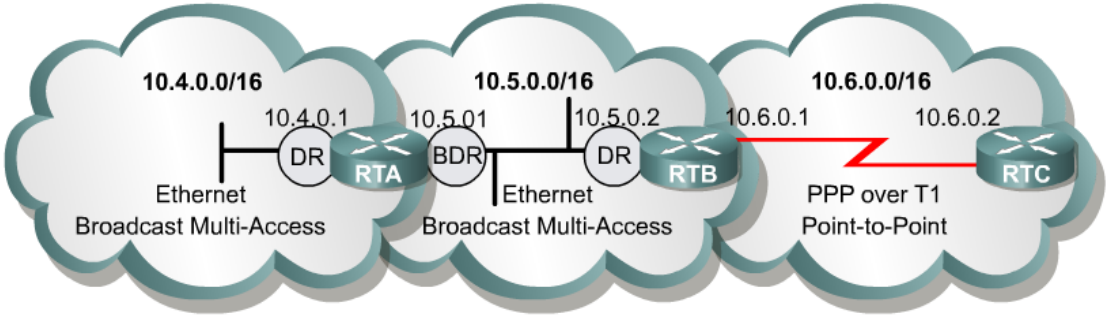
  `Rtr(config-if)#` **`ip ospf priority <0-255>`**

  – С по-високия приоритет става DR/BDR

  – Default = 1

  – 0 = никога няма да стане DR/BDR

Роля на BDR



- Слуша, но не действа
- Щом се изпрати LSA, BDR установява таймер. Ако той изтече преди DR да отговори, BDR се превръща в DR и поема неговата работа.
- Стартира се процес на избор на нов BDR.

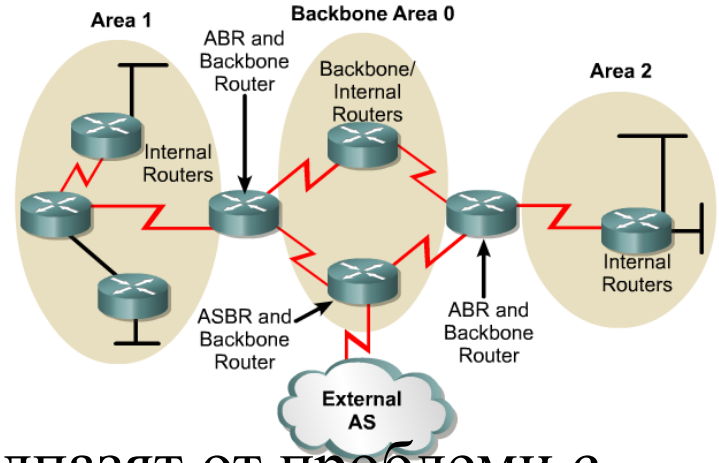
Добавяне на нов маршрутизатор

- След като веднъж е избран DR, дори да се добави нов маршрутизатор с по-висок приоритет или Router ID, този новият няма да стане DR или BDR.
- Ако DR се повреди, BDR става DR и се стартира процес на избор на нов BDR.

И още...

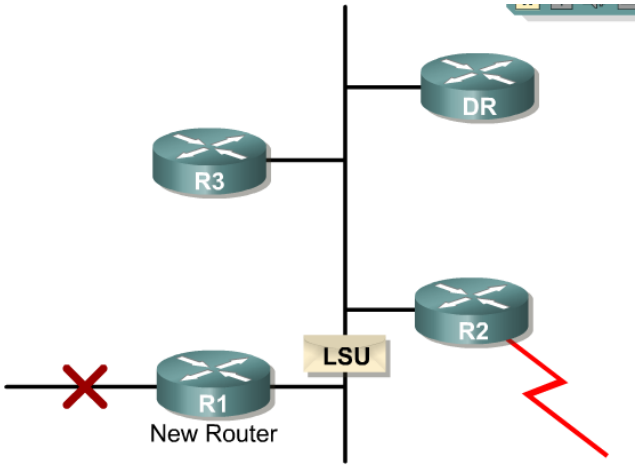
- Времето за изчисление на SPF зависи от:
  - Мощността на CPU
  - Размера на областта (area)
  - Броя на маршрутизаторите
  - Размера на link state database.
- “flapping” връзките могат да предизвикат постоянни преизчисления и мрежата никога да не се конвергира, затова:
  - SPF изчисленията се забавят с **5 seconds** след получаването на **LSU** (Link State Update) – това време (*spf-delay)* може да се промени
  - **Изчакването между последователни SPF изчисления е 10 seconds** – това време (*spf-holdtime)* може да се промени

Защо multi-area

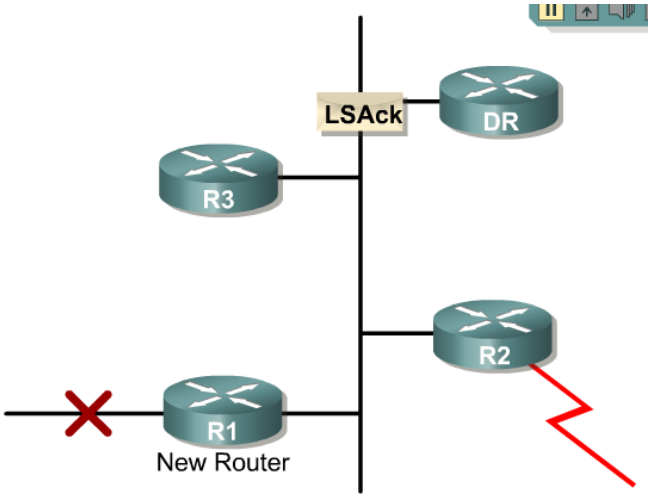


- OSPF areas се създават, за да се предпазят от проблеми с “flapping” връзки вътре в областта.
- SPF не се преизчислява когато се сменя топологията в друга област.
- OSPF дистрибутира информацията между областите през **ABR** с distance-vector метод.
- **OSPF използва link-state метод само вътре в областта**.
- Добре е да се направи route summarization между областите, за да се скрият промените от една област и да не се отразят върху маршрутните таблици в друга област.

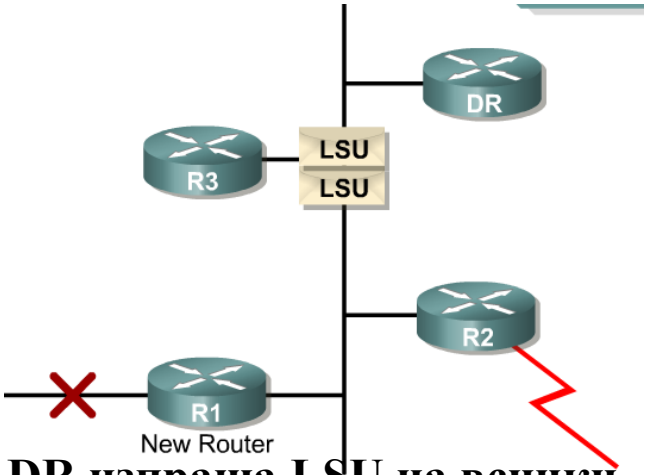
При отпадане на връзка



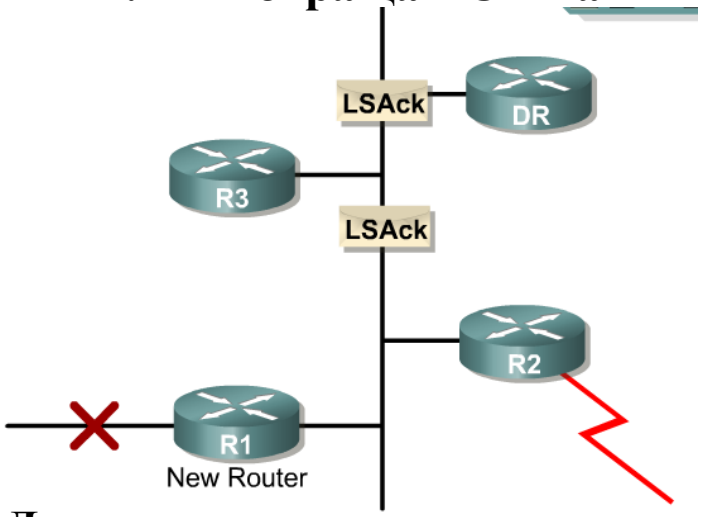
1. R1 изпраща LSU на DR



2. DR изпраща ACK на R1



3. DR изпраща LSU на всички маршрутизатори



4. Другите маршрутизатори отговарят с ACK на DR

Време за конвергенция

Определя се от:

- Link-failure - физическия или data link слой директно изпраща информация за промяната по директно свързаните интерфейси.
- Сигнал за недостижим Neighbor – подпомагащия Hello протокол не получава пакет от съседа в RouterDeadInterval (40 seconds или 120 seconds по NBMA links).
- Получаване на нов LSA
- Наводняване с нова версия на LSA до всички маршрутизатори
- SPF изчисления на всички маршрутизатори

- Изтриване или добавяне на път в routing table може да предизвика изпращане на LSAs до други области.

Въпроси ?

Благодаря за вниманието !