

Маршрутизация Протокол OSPF

Адаптивная маршрутизация

	Внутренняя маршрутизация (IGP)				Внешняя маршрутизация (EGP)
	Дистанционно-векторные пр.	Протоколы состояния связей			
Classful	RIP	IGRP			EGP
Classless	RIPv2	EIGRP	OSPF v2	IS-IS	BGPv4
IPv6	RIPng	EIGRP for IPv6	OSPF v3	IS-IS for IPv6	BGPv4 for IPv6

- **Адаптивная маршрутизация** (adaptive routing) – это совокупность методов маршрутизации, при которых маршрутизаторы могут в процессе работы изменять таблицы маршрутов, подстраиваясь под изменения в сети

Протоколы состояния связей

- В процессе инициализации маршрутизаторы собирают полную информацию о топологии сети (топологическая БД) и рассчитывают маршруты
- В процессе работы маршрутизаторы обмениваются короткими сообщениями для проверки состояния связей
- Повторный обмен маршрутной информацией только при изменении топологии (экономия ресурсов)

Общий принцип работы LSP

- Каждый маршрутизатор распознаёт непосредственно подключенные сети
- Каждый маршрутизатор распознаёт соседние маршрутизаторы, имеющие интерфейсы в непосредственно подключенных сетях
- Каждый маршрутизатор формирует пакет LSP (link state packet) с информацией о всех подключенных к нему сетях и рассылает соседям
- Каждый маршрутизатор рассылает свой LSP соседям. При получении LSP от соседей, маршрутизатор копирует его и рассылает далее, не внося изменений
- Каждый маршрутизатор использует построенную по LSP топологическую БД для расчёта кратчайших маршрутов и заполнения таблиц маршрутизации

Open Shortest Path First - OSPF

- Область применения – внутренний протокол маршрутизации
- Тип алгоритма – link-state (протокол «состояния связей»), т.е. все маршрутизаторы имеют одинаковое представление о топологии сети и рассчитывают наилучшие пути по алгоритму SPF (алгоритм Дейкстры)
- Разрешает балансировку трафика по равноценным путям
- Версии:
 - OSPFv2 – для IPv4 – RFC 2328
 - OSPFv3 – для IPv6 – RFC 5340
- Протокол работает поверх IP (номер 89) и использует для рассылки два мультикаст-адреса:
 - 224.0.0.5 – all-ospf-routers
 - 224.0.0.6 – all-ospf-dr-bdr
- Метрика: $Cost = 100 / Int.BW$
- AD = 110 (по умолчанию)

OSPF – структуры данных

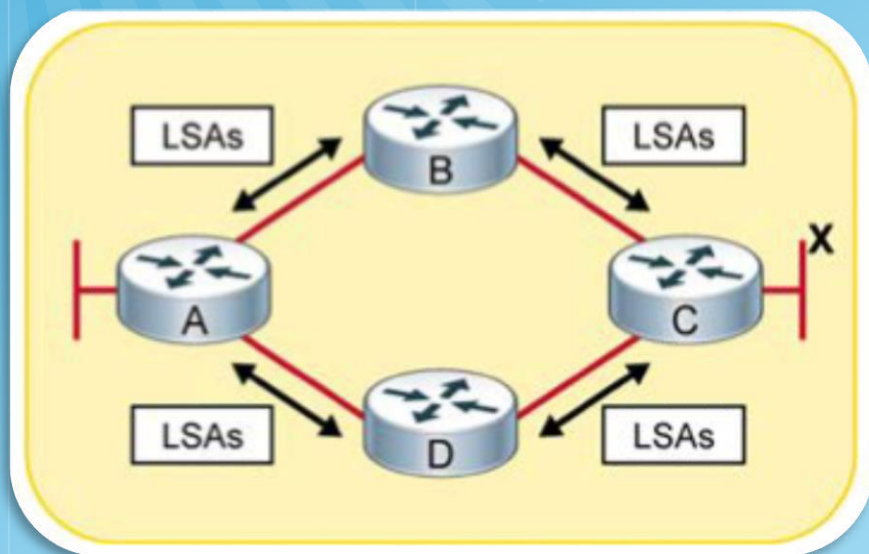
A: Neighbors Table
B
D

LSA flooding

A: Topology Table
A: 3 сети
D: 2 сети
...

Dijkstra SPF

A: Routing Table
Сеть X через D
...

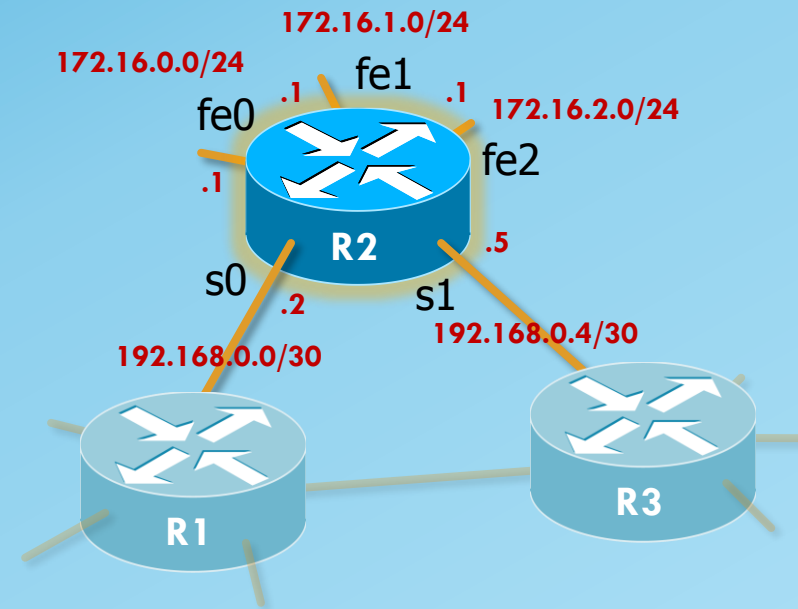


- **БД соседей** – содержит информацию о соседних М.
- **Топологическая БД** – содержит информацию о всех интерфейсах всех М.
- **Таблица маршрутов** – содержит информацию о лучших маршрутах

OSPF шаг 1 – распознавание подключенных сетей

■ R2 – 5 интерфейсов:

- fe0 = 172.16.0.1
в сети 172.16.0.0/24
- fe1 = 172.16.1.1
в сети 172.16.1.0/24
- fe2 = 172.16.3.1/24
в сети 172.16.3.0/24
- s0 = 192.168.0.2/30
в сети 192.168.0.0/30
- s1 = 192.168.0.5/30
в сети 192.168.0.4/30

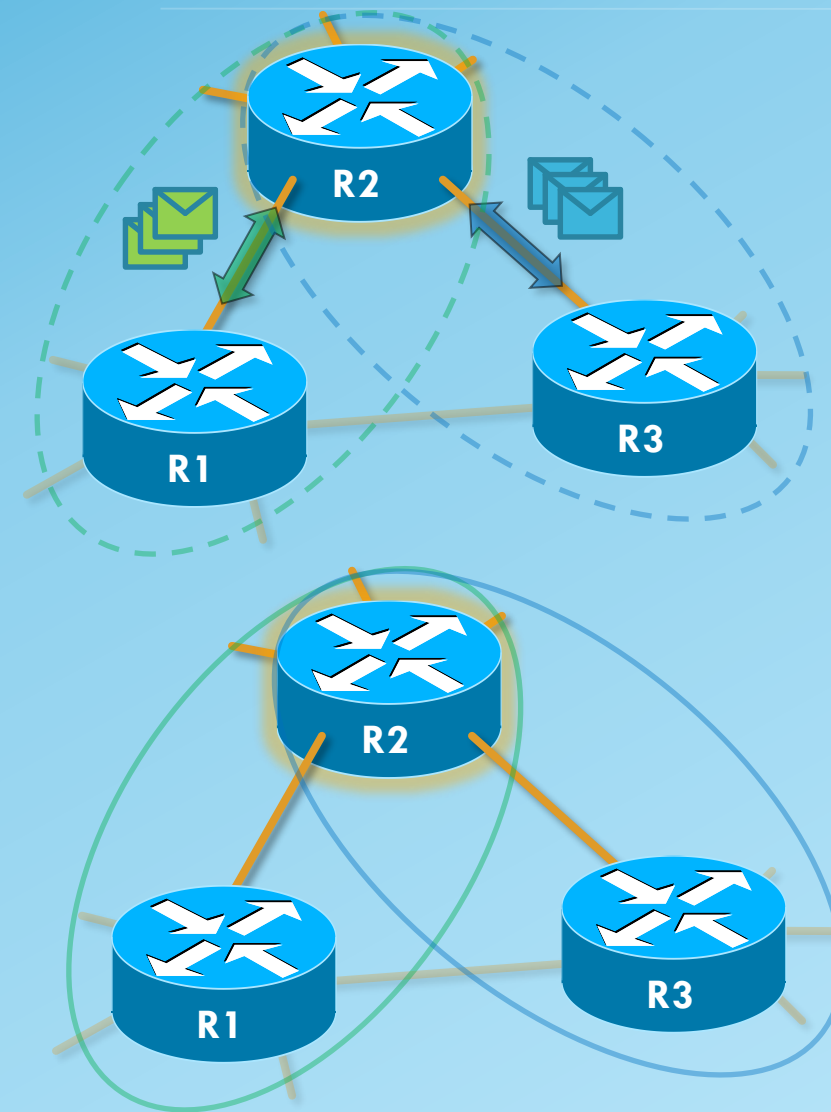


OSPF шаг 2 – отношения соседства

- R2 посылает пакеты HELLO со всех интерфейсов, где включен OSPF, на 224.0.0.5
- R2 получает HELLO от R1 и R3
- В случае совпадения настроек OSPF (зона, способ расчёта метрики, таймеры синхронизации и т.д.) пары R2/R1 и R2/R3 устанавливают отношения соседства
- Формируется таблица соседей

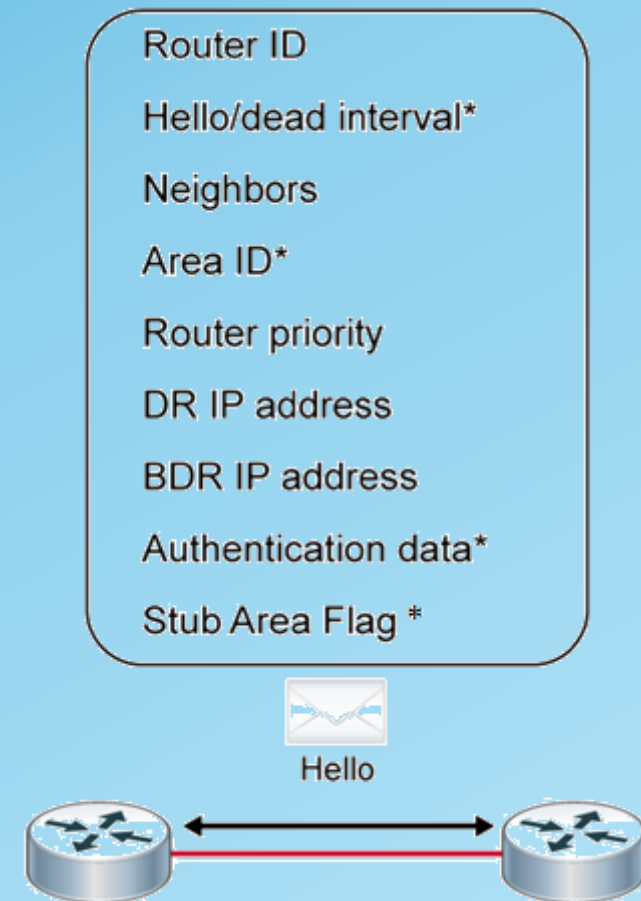
```
router#show ip ospf neighbors
```

- посмотреть сведения о соседях



OSPF – сообщения HELLO

- Поля в Hello пакете, отмеченные *, должны совпадать.
- Router ID (RID) – уникальный идентификатор (32 бита), представляется в формате X.X.X.X. Если RID не назначен вручную, то выбирается следующим образом:
 - берется самый большой IP-адрес с виртуального интерфейса типа Loopback;
 - если нет ни одного интерфейса Loopback, то берется самый большой IP-адрес с реального физического интерфейса.
- Area ID = 32 бита, может быть представлен в формате X.X.X.X.

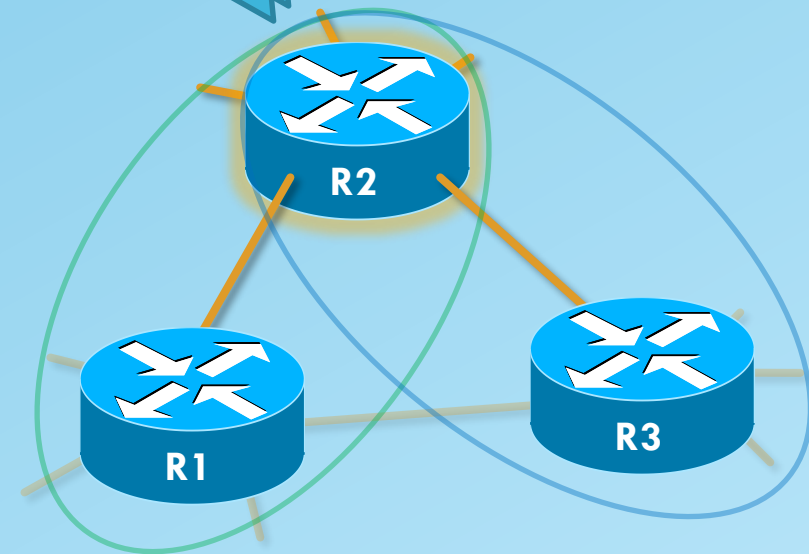


OSPF – Link State Packet (LSP)

- R2 формирует LSP – link state packet – извещение о состоянии известных ему каналов
- Сведения о состоянии канала включают (для каждого интерфейса):
 - IP-адрес сети (с маской)
 - IP-адрес собственного интерфейса в этой сети
 - Тип сети (Ethernet, Serial, frame relay и т.д.)
 - «Стоимость» канала – метрику перехода
 - Соседи на этом канале

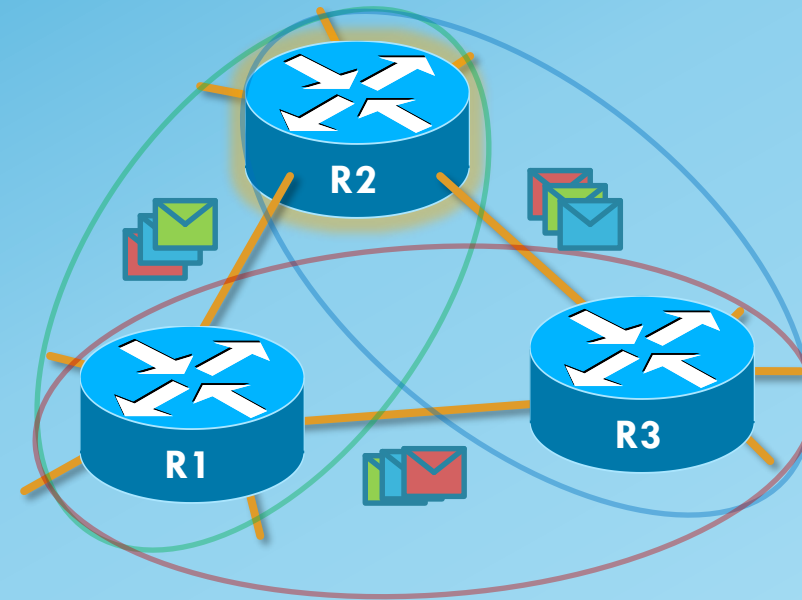
Link 1:

- network 172.16.0.0/24
- IP 172.16.0.1
- network type: Ethernet
- link cost: 2
- neighbors: none



OSPF шаг 3 – LSP Flooding, LSDB

- R2 посылает свой LSP всем маршрутизаторам, с которыми он находится в состоянии соседства (R1 и R3)
- R2 получает LSP от R1 и от R3. При получении LSP он копирует их в свою топологическую БД (Link state database) и пересылает дальше без изменений
- Таким образом все маршрутизаторы получают все LSP – формируются полные LSDB у всех роутеров



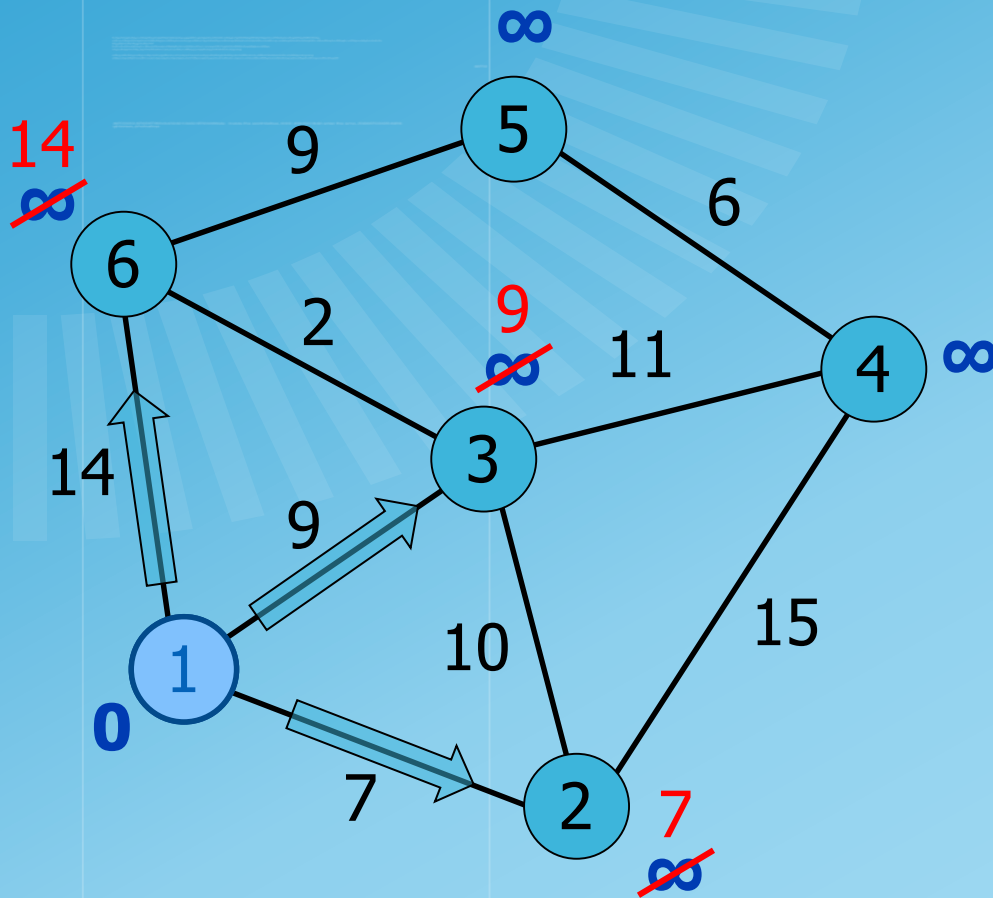
- `router#show ip ospf database`
 - посмотреть содержимое топологической базы данных

- **База данных состояния каналов** (link state database – LSDB, **топологическая база данных** – topological database) – список всех записей о состоянии каналов

Алгоритм Дейкстры

- **Алгоритм Дейкстры** (SPF, shortest path first) – алгоритм на графах, позволяющий найти кратчайшее расстояние от одной из вершин графа до всех остальных (работает только для графов без рёбер отрицательного веса)
- Пусть $D(v)$ равно сумме весов связей для данного пути
- Пусть $c(i,j)$ равно весу связи между узлами с номерами i и j
- 1. Устанавливаем множество проверенных узлов $N = \emptyset$
- 2. Устанавливаем $D(1) = 0$, все другие $D(v) = \infty$
- 3. Для каждого шага находим узел w не из множества N , для которого $D(w)$ минимально
- 4. Актуализируем $D(v)$ для всех узлов не из множества N :
 $D(v) = \min \{ D(v), D(w) + c(w,v) \}$
- 5. Добавляем узел w в множество N
- 6. Повторяем шаги 3-5, пока все узлы не окажутся в множестве N

Алгоритм Дийкстры – пример (1)



$$N = \emptyset$$

$$D(1) = 0 \quad D(4) = \infty$$

$$D(2) = \infty \quad D(5) = \infty$$

$$D(3) = \infty \quad D(6) = \infty$$

$$w=1, D(w)=0$$

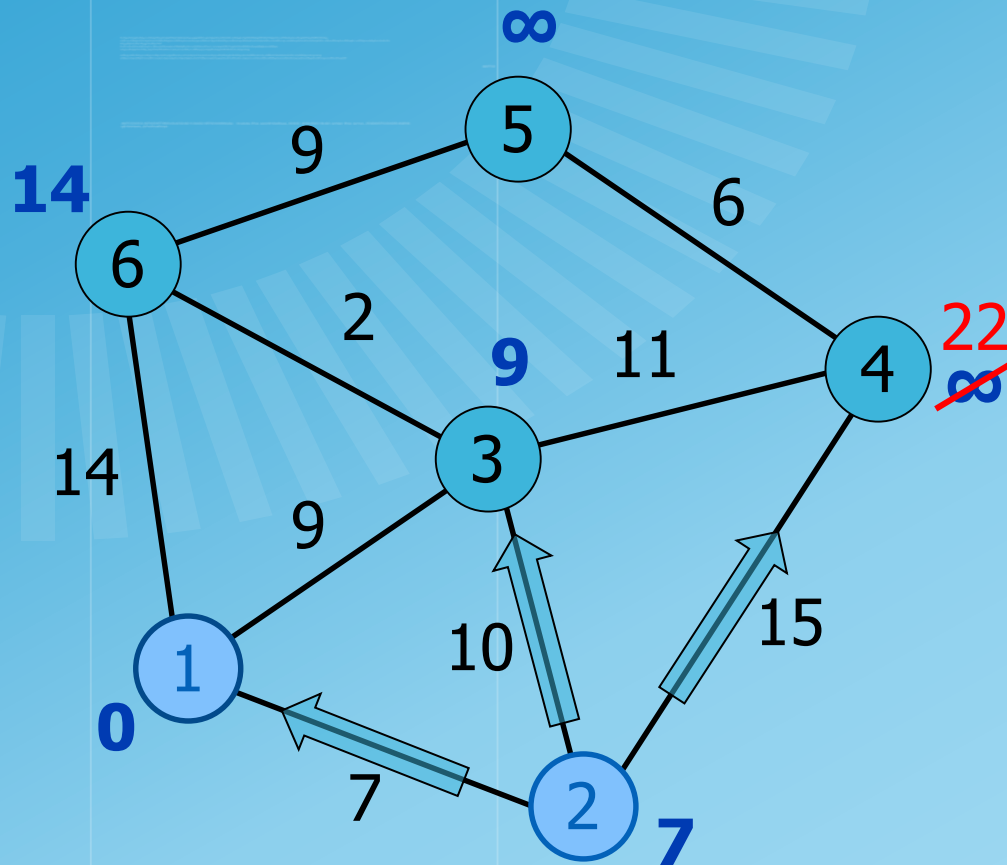
$$D(2) = \min\{0+7, \infty\} = 7$$

$$D(3) = \min\{0+9, \infty\} = 9$$

$$D(6) = \min\{0+14, \infty\} = 14$$

$$N = \{1\}$$

Алгоритм Дийкстры – пример (2)



$N = \{1\}$

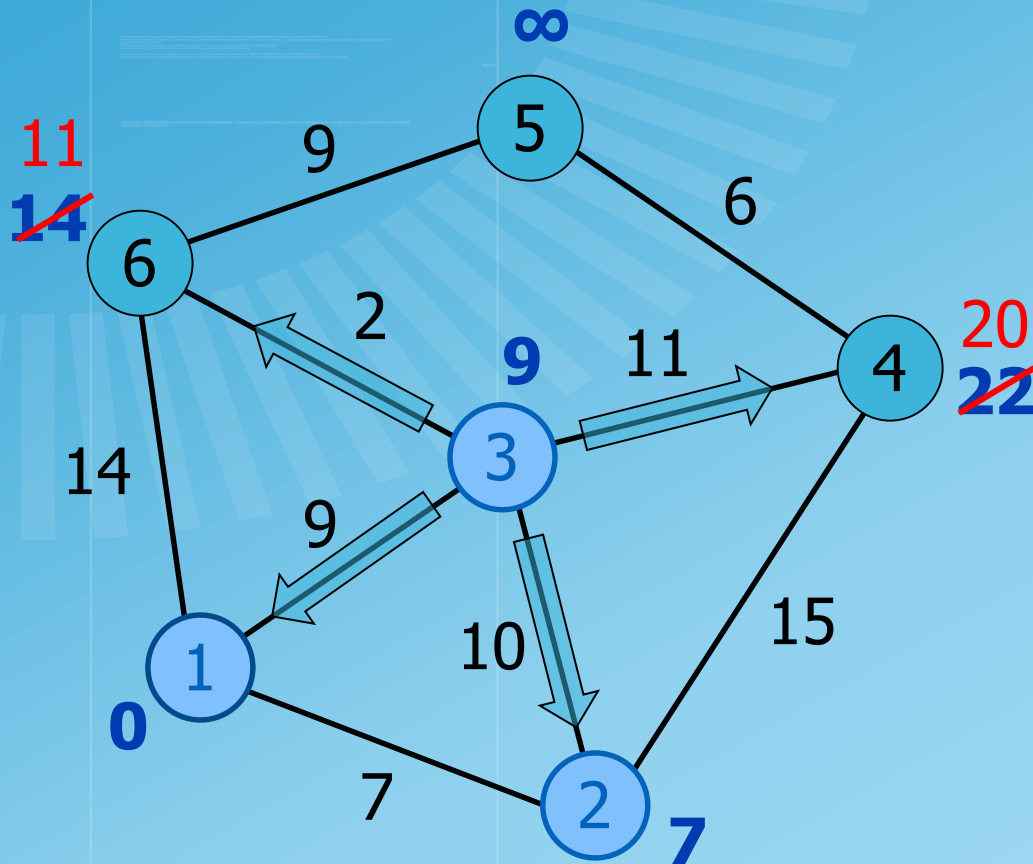
$D(1) = 0$ $D(4) = \infty$
 $D(2) = 7$ $D(5) = \infty$
 $D(3) = 9$ $D(6) = 14$

$w=2, D(w)=7$

$D(1) = \min\{7+7, 0\} = 0$
 $D(3) = \min\{7+10, 9\} = 9$
 $D(4) = \min\{7+15, \infty\} = 22$

$N = \{1, 2\}$

Алгоритм Дийкстры – пример (3)



N = {1,2}

D(1) = 0 D(4) = 22

D(2) = 7 D(5) = ∞

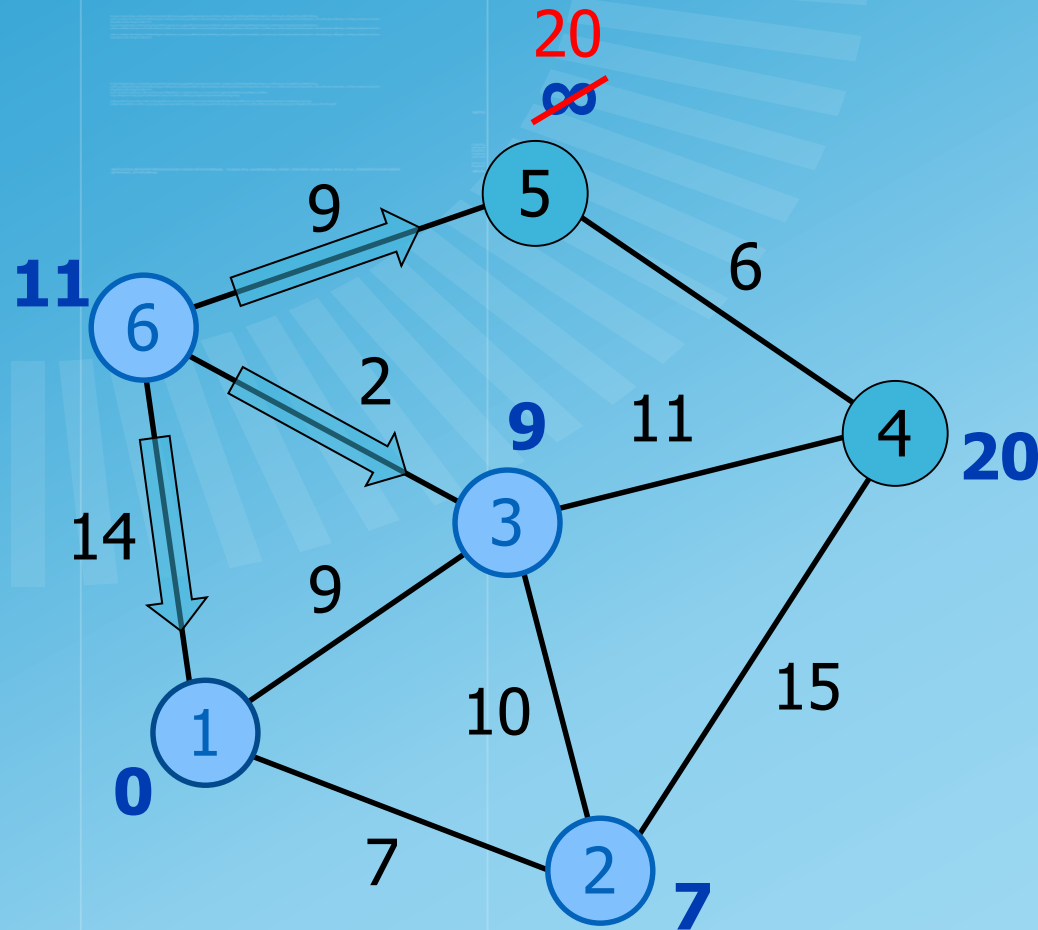
D(3) = 9 D(6) = 14

$w=3, D(w)=9$

$$D(1) = \min\{9+9, 0\} = 0$$
$$D(2) = \min\{9+10, 7\} = 7$$
$$D(4) = \min\{9+11, 22\} = 20$$
$$D(6) = \min\{9+2, 14\} = 11$$

$N = \{1, 2, 3\}$

Алгоритм Дийкстры – пример (4)



$N = \{1, 2, 3\}$

$D(1) = 0 \quad D(4) = 20$

$D(2) = 7 \quad D(5) = \infty$

$D(3) = 9 \quad D(6) = 11$

$w=6, D(w)=11$

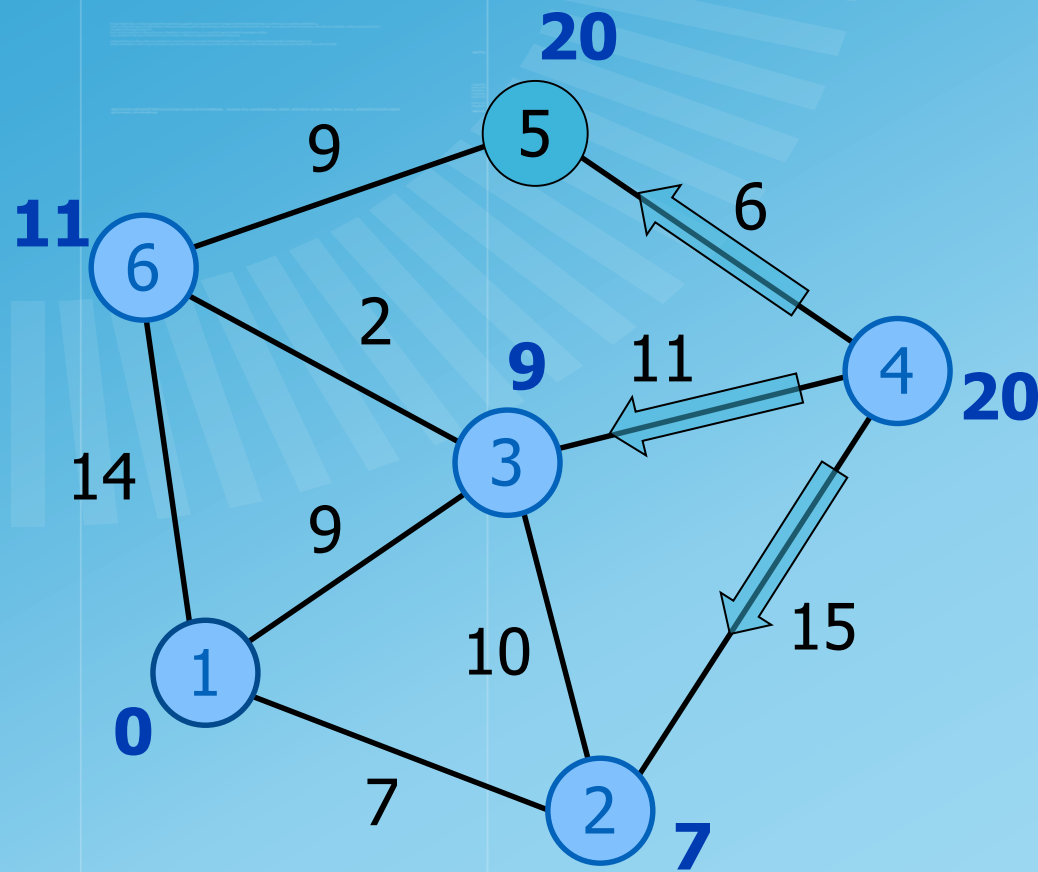
$D(1) = \min\{11+14, 0\} = 0$

$D(3) = \min\{11+2, 9\} = 9$

$D(5) = \min\{11+9, \infty\} = 20$

$N = \{1, 2, 3, 6\}$

Алгоритм Дijkstra – пример (5)



$N = \{1, 2, 3, 6\}$

$D(1) = 0$ $D(4) = 20$

$D(2) = 7$ $D(5) = 20$

$D(3) = 9$ $D(6) = 11$

$W=4, D(w)=20$

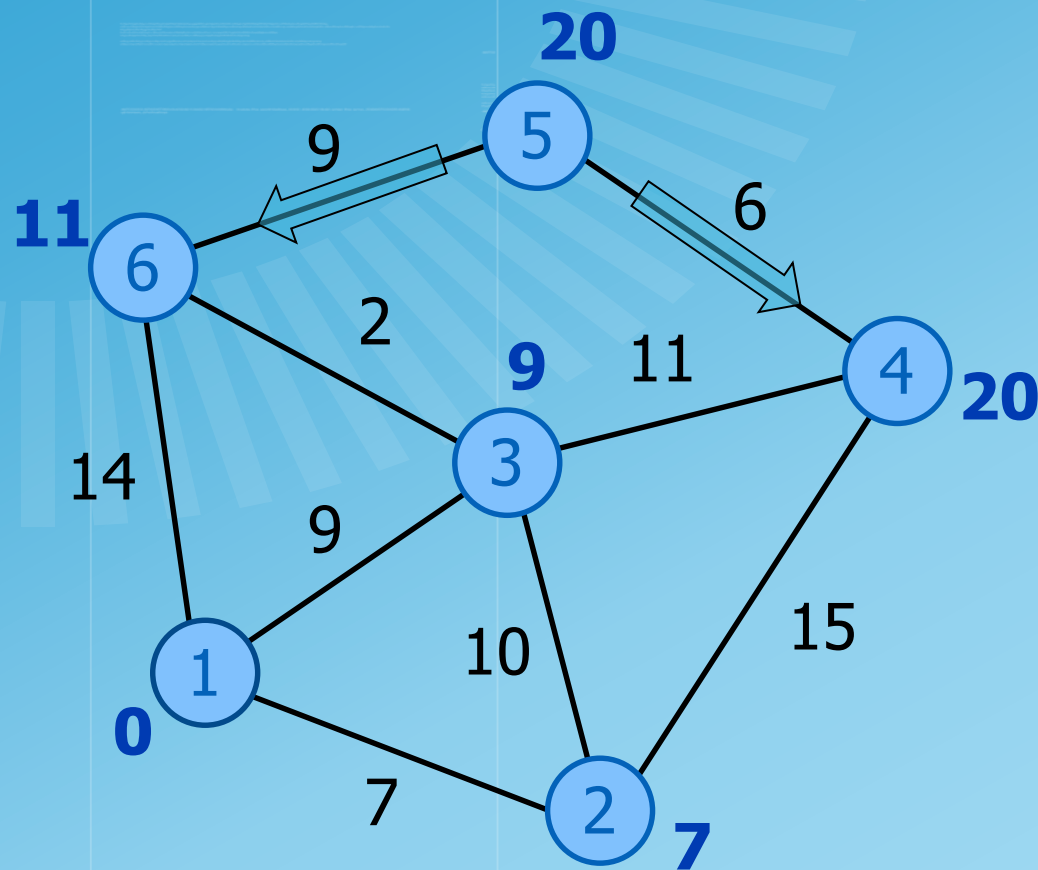
$D(2) = \min\{20+15, 7\} = 7$

$D(3) = \min\{20+11, 9\} = 9$

$D(5) = \min\{20+6, 10\} = 20$

$N = \{1, 2, 3, 4, 6\}$

Алгоритм Дийкстры – пример (6)



$N = \{1, 2, 3, 4, 6\}$

$D(1) = 0$ $D(4) = 20$

$D(2) = 7$ $D(5) = 20$

$D(3) = 9$ $D(6) = 11$

$W=5, D(w)=20$

$D(4) = \min\{20+6, 20\} = 20$

$D(6) = \min\{20+9, 11\} = 11$

$N=\{1, 2, 3, 4, 5, 6\}$

Алгоритм завершен

OSPF – механизмы

- TTL записей топологической базы, актуализация: Link-State Request, Link-State Update
- Периодическая инициализация топологических баз
- Исключена вероятность распространения устаревшей ложной информации о доступности сетей
- Поддержка QoS на уровне метрики
- Поддержка балансировки нагрузки (хранит несколько маршрутов с одинаковой метрикой)
- Обладает высокой вычислительной сложностью
 - Сеть разбивается на т.н. области сети (зоны)
 - OSPF работает на мощных аппаратных маршрутизаторах

Настройка OSPF – пример (1)

- **router#show ip ospf [...]**
 - сведения о конфигурации и работе протокола OSPF
- **router(config)#router ospf 1**
 - переход в режим конфигурирования протокола OSPF, номер процесса 1
 - номер процесса имеет локальное значение для данного маршрутизатора
- **router(config-router)#no auto-summary**
 - отключает автосуммирование маршрутов (по умолчанию вкл.), рекомендуется
- **router(config-router)#network 10.0.0.0 0.0.0.255 area 0**
 - включает данный процесс протокола OSPF на всех интерфейсах, входящих в указанную сеть, и помещает их в выбранную зону (маска – инверсная)
- **router(config-router)#passive-interface f0/0**
 - переводит интерфейс f0/0 в пассивный режим; пассивные интерфейсы не осуществляют рассылку маршрутных сообщений
- **router(config-router)#default-information originate**
 - включить анонсирование маршрута по умолчанию

Настройка OSPF – пример (2)



□ `router(config-router)#area 0 authentication message-digest`

- включает md5-аутентификацию для всех интерфейсов данной зоны
- метод аутентификации также может быть задан отдельно для каждого интерфейса

□ `router(config-if)#ip ospf authentication message-digest`

- включает md5-аутентификацию для протокола OSPF на текущем интерфейсе
- это необходимо лишь в том случае, если такой метод аутентификации уже не выбран для всех для всех интерфейсов данной зоны

□ `ip ospf message-digest-key 123 md5 qwerty`

- создает md5-ключ с порядковым номером **123** и паролем **qwerty**
- порядковый номер необходим для управления версиями ключей – при смене ключа каждый следующий номер должен быть больше предыдущего
- номер ключа и пароль должны совпадать на соседних маршрутизаторах, но не рекомендуется использовать одинаковые значения во всей зоне

Функции маршрутизатора

