

**МСС-MSN**  
**Тема 6. Лекция 2**

**Технологии МСС/L3**

**Маршрутизация**  
**Алгоритмы, протоколы**

**Костюкович А.Е.**  
**Каф.АЭС, СибГУТИ**  
**[www.aek-54.ru](http://www.aek-54.ru)**

# ПЛАН Лекции

- 1. Общие положения**
- 2. Задачи управления трафиком**
- 3. Задачи маршрутизации**
- 4. Маршрутная таблица**
- 5. Алгоритмы выбора маршрутов**
- 6. Классификация алгоритмов**
- 7. Критерии выбора маршрутов (метрики)**

# 1. Общие положения

В лекции рассмотрены подходы к классификации алгоритмов выбора маршрутов, применяемые для формирования маршрутных таблиц в каждом узле.

Максимальный коммерческий эффект от сети не может быть получен без оптимального использования всех сетевых ресурсов – в первую очередь пропускной способности каналов связи.

# 1. Общие положения

Работу пакетной сети можно считать эффективной, когда каждый ресурс загружен, но не перегружен.

Это значит, что коэффициент использования ресурса должен приближаться к единице, но не настолько, чтобы очереди пакетов к ресурсу (неизбежное явление в пакетных сетях) приводили к недопустимым задержкам и потерям из-за переполнения внутренних буферов в маршрутизаторах.

Оптимизация сетевых ресурсов за счет использования динамических методов маршрутизации позволяет выжать из сети максимум возможного.

## 2. Управление сетевым трафиком

Увеличение количества пользователей в сети и появление трафика от разных приложений заставляет оператора:

- более тщательно изучать характеристики трафика,
- более эффективно использовать полосу пропускания магистральных интерфейсов
- использовать более совершенные методы управления трафиком.

Управление трафиком является основным инструментом для управления качеством оказываемых услуг(гарантии QoS).

К управляемым ресурсам и средствам управления относятся:

- управление очередями
- управление пропускной способностью
- управление маршрутизацией
- управление тарифами

### 3. Задачи маршрутизации

Основной функцией уровня L3 является Функция Маршрутизации.

Оптимизация выбора маршрута позволяет увеличить объем пропускаемого трафика, а это и есть цель любого оператора – передать по сети максимальный объем трафика при допустимом качестве обслуживания.

Маршрутизация сводится к выбору интерфейса и следующего транзитного узла при продвижении пакета между сетями.

Маршрутизацию выполняют специальные устройства – сетевые маршрутизаторы (Router).

## 4. Маршрутная таблица – RT

Основным результатом работы алгоритма маршрутизации является создание и поддержка таблицы маршрутизации (RT), в которую записывается вся маршрутная информация.

Содержание таблицы маршрутизации зависит от типа сети, используемого протокола маршрутизации и типа ОС.

Записи в **RT** точно определяют:

- место назначения (сетевой адрес/маска сети)
- следующий маршрутизатор (шлюз)
- интерфейс
- метрику (вес) маршрута

# Пример RT



У маршрутизатора R1 есть 2 интерфейса:

- LAN Порт Ethernet с IP-адресом **192.168.0.4** (к LAN).
- Модем с интерфейсом PPP и IP-адресом **203.96.10.51** (WAN).

## Пример RT в R1:

Network Address Адрес Сети	Netmask Маска	Gateway Address Адрес шлюза	Interface Интерфейс	Metric Метрика
<b>127.0.0.0</b>	<b>255.0.0.0</b>	<b>127.0.0.1</b>	<b>127.0.0.1</b>	<b>1</b>
<b>192.168.0.0</b>	<b>255.255.255.0</b>	<b>192.168.0.1</b>	<b>192.168.0.4</b>	<b>2</b>
<b>192.168.0.255</b>	<b>255.255.255.255</b>	<b>192.168.0.1</b>	<b>192.168.0.4</b>	<b>2</b>
<b>224.0.0.0</b>	<b>224.0.0.0</b>	<b>192.168.0.1</b>	<b>192.168.0.4</b>	<b>2</b>
<b>203.96.10.0</b>	<b>255.255.255.0</b>	<b>203.96.10.1</b>	<b>203.96.10.51</b>	<b>1</b>
<b>203.96.10.255</b>	<b>255.255.255.255</b>	<b>203.96.10.1</b>	<b>203.96.10.51</b>	<b>1</b>
<b>224.0.0.0</b>	<b>224.0.0.0</b>	<b>203.96.10.1</b>	<b>203.96.10.51</b>	<b>1</b>
<b>0.0.0.0</b>	<b>0.0.0.0</b>	<b>203.96.10.1</b>	<b>203.96.10.51</b>	<b>1</b>

**0.0.0.0** – адрес шлюза по умолчанию (**default gateway**)

**127.0.0.0** – Localhost (петля – Loopback)

Метрика имеет значение для альтернативных маршрутов !!!

# Процедура вычисления адреса следующего хопа

1. Из поступившего пакета берется IP-адрес назначения

Пусть DA = 192.168.0.7

2. Последовательно просматриваются все записи в RT

Порядок просмотра записей в таблице маршрутизации следующий: сначала просматриваются маршруты к существующим сетям, если такие маршруты не найдены – просматривается маршрут по умолчанию.

3. Для каждой записи вычисляется **IP-адрес & Маска**

4. По результату выбирается подходящий адрес сети

5. Для выбранной сети пакет направляется в соответствующий интерфейс

## 5. Алгоритмы выбора маршрутов

В больших составных сетях всегда существует несколько альтернативных маршрутов между двумя конечными узлами, поэтому встает задача определения лучшего (оптимального) маршрута из нескольких возможных.

Выбор маршрута осуществляется на основании анализа следующей информации:

- сетевое адреса в заголовке IP-дейтаграммы (текущая, динамическая информация),
- информации о конфигурации сети (информация в маршрутных таблицах),
- критерия выбора (метрики маршрута).

**Лучший маршрут** – это маршрут с наименьшей метрикой.

## 5. Алгоритмы выбора маршрутов

Определение маршрута передачи информации происходит программно (средствами операционной системы и специализированных приложений).

В решении задач маршрутизации программными средствами есть недостатки:

1. Маршрутизаторы с ПО более дороги, чем аппаратные
2. Программное решение функций – более медленно, чем аппаратное

Однако, преимущества программных средств более значительны:

1. Гибкость настройки параметров маршрутизатора
2. Гибкость модификации

Соответствующие программные средства носят названия протоколов маршрутизации.

Логика их работы основана на алгоритмах маршрутизации.

Алгоритмы маршрутизации вычисляют стоимость доставки и выбирают путь с меньшей стоимостью (метрикой).

## 5. Алгоритмы выбора маршрутов

### **Основные требования к алгоритму маршрутизации:**

- 1. Оптимальность выбора маршрута.**
- 2. Простота реализации**
- 3. Низкие непроизводительные затраты (малый объем служебных затрат).**
- 4. Устойчивость.**
- 5. Быстрая сходимость.**
- 6. Гибкость.**

## 5. Алгоритмы выбора маршрутов

**1. Оптимальность выбора маршрута** является основной целью разработки алгоритма. Она характеризует способность алгоритма маршрутизации выбирать "наилучший" маршрут с точки зрения какого-либо критерия (метрики).

**2-3. Простота реализации и низкие служебные затраты.** Простой алгоритм маршрутизации должен обеспечивать свои возможности, **с минимальными вычислительными затратами, определяющими** стоимость ПО, стоимость процессоров, производительность (быстродействие) маршрутизатора как сетевого узла, работающего в реальном времени, а **также с минимальными затратами на служебный трафик.**

## 5. Алгоритмы выбора маршрутов

4. Алгоритмы маршрутизации должны обладать устойчивостью, т.е. они должны четко работать в случаях отказов аппаратуры, в условиях высокой нагрузки и ошибок в построении сети.

Часто наилучшими алгоритмами маршрутизации оказываются именно те, которые выдержали испытание временем и доказали свою надежность в различных условиях работы сети.

## 5. Алгоритмы выбора маршрутов

5. Быстрая сходимость является важной характеристикой алгоритмов маршрутизации. Сходимость определяется **возможностью** и **временем процесса согласования таблиц** (RT) между сетевыми узлами.

Когда событие в сети приводит к тому, что маршруты или блокируются, или обратно – становятся доступными, сетевые узлы рассылают **сообщения об обновлении RT**.

Сообщения об обновлении RT запускают процесс пересчета маршрутов, вынуждая все узлы прийти к соглашению по этим маршрутам.

Есть ситуации, когда процессы обновления могут либо вообще не закончиться, либо продолжаться долго.

Алгоритмы маршрутизации, **которые сходятся медленно**, могут привести к образованию **петель маршрутизации** или выходу из строя сети.

## 5. Алгоритмы выбора маршрутов

6. Алгоритмы маршрутизации должны обладать гибкостью, т.е. должны быстро и точно адаптироваться к изменениям в сети, в частности, к изменениям:

- полосы пропускания,
- размеров очереди к узлу,
- величины задержки и к другим влияниям

## 5. Алгоритмы выбора маршрутов

### Алгоритмы маршрутизации могут быть:

- статическими или динамическими;
- одномаршрутными или многомаршрутными;
- одноуровневыми или иерархическими;
- централизованные или распределенные;
- внутридоменными и междоменными;
- одноадресными или групповыми (многоадресными);
- алгоритмами состояния канала или вектора расстояний.

# Показатели алгоритмов (метрики)

Основные критерии выбора (метрики), используемые в алгоритмах маршрутизации:

- Длина маршрута.
- Надежность.
- Задержка.
- Ширина полосы пропускания.
- Нагрузка.
- Стоимость маршрута (тариф).

## Алгоритмы выбора маршрутов

В больших сетях главная проблема состоит в том, чтобы **оперативно получать** текущее состояние каждого маршрута.

Т.к. состояние маршрутов является величиной случайной и постоянно меняющейся, то количество модификаций RT может быть очень большим и алгоритм вычисления оптимального маршрута никогда не сойдется.

# Алгоритмы выбора маршрутов

Методы маршрутизации делятся на :

- Централизованные (от источника), когда решающие функции закреплены за одним узлом, который посылает соответствующие команды другим узлам (например – в Softswitch);
- Децентрализованные (от узла к узлу), когда каждый узел самостоятельно выбирает маршрут на основе собственной информации (**основной метод в Интернете**).

# Алгоритмы выбора маршрутов

В зависимости от способа формирования таблиц маршрутизации алгоритмы маршрутизации делятся на два класса :

- алгоритмы **фиксированной или статической** маршрутизации;
- алгоритмы **адаптивной (динамической)** маршрутизации.

В алгоритмах фиксированной или статической маршрутизации RT строятся и обновляются администратором вручную без участия протоколов маршрутизации.

Используются в примитивных сетях.

# Алгоритмы выбора маршрутов

Адаптивные алгоритмы обмена маршрутной информацией делятся на три группы:

- дистанционно-векторные алгоритмы (**DVA** – Distance Vector Algorithms) – основаны на алгоритме Беллмана-Форда
- алгоритмы состояния связей (**LSA** – Link State Algorithms) – основаны на алгоритме Дейкстры
- **алгоритмы на основе политик** (правил) – наиболее оптимальные алгоритмы, учитывающие ряд ограничений, накладываемых администратором сети

## Алгоритмы выбора маршрутов

Алгоритм DVA прост и надежен, но работает только в сетях с ограниченным количеством узлов.

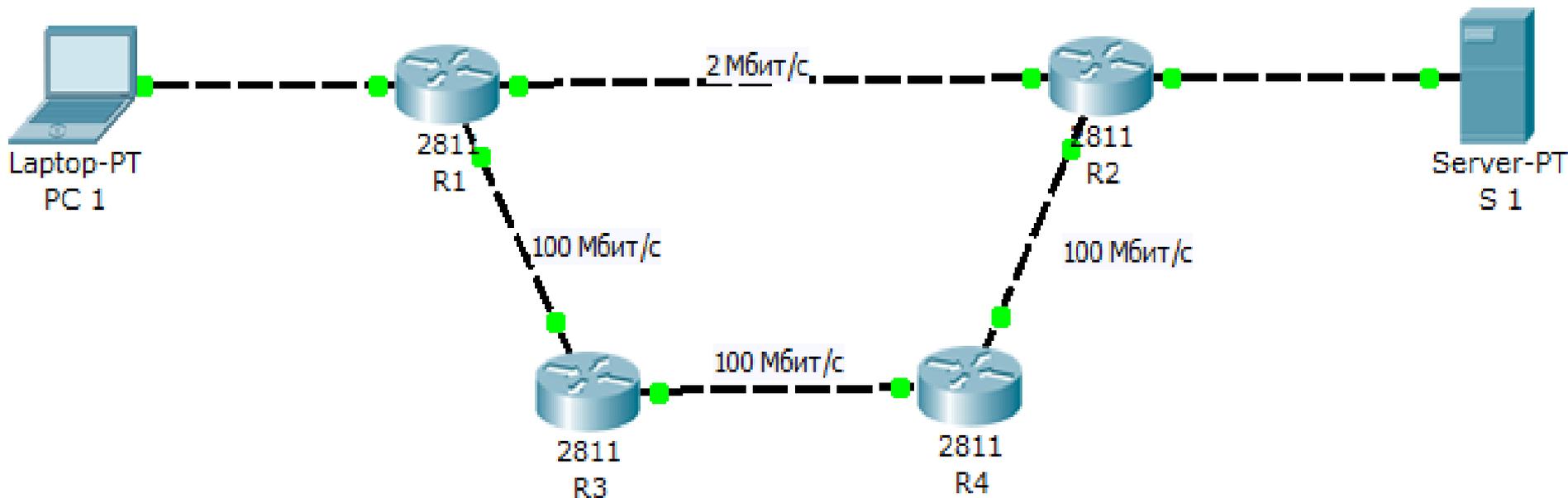
Наиболее распространенный протокол класса **DVA** – **RIP** (Routing Information Protocol).

В качестве метрики при выборе маршрута RIP использует количество переходов (хопов)

В протоколе RIP много времени затрачивается на перенос изменений топологии сети в маршрутную базу данных (**проблема медленной сходимости**), поэтому применение данного протокола в сети более 15-ти узлов не рекомендуется.

# Алгоритмы выбора маршрутов

Очевидный недостаток метрики по числу хопов наглядно демонстрирует схема, на которой из всех возможных маршрутов между R1 и R2, протокол RIPv1 выберет маршрут 2 Мбит/с, т.е. с одним ХОПОМ.



# Алгоритмы выбора маршрутов

Недостатки RIP (1-я версия):

1. RIP – не поддерживает сетевые маски CIDR/VLSM
2. RIP рассылает полную RT каждые 30 сек.
3. RIP считает оптимальность пути только по числу хопов.

Cisco разработала усовершенствованный алгоритм – IGRP, который поддерживает сетевые маски, работает по состоянию, хранит "веса" линков, не требует больших ресурсов, но реализован только на CISCO-роутерах.

В версии RIP-2 также устранены недостатки RIP-1.

# Алгоритмы выбора маршрутов

**RIP-2** поддерживает:

- маски подсетей;
- механизм аутентификации для обеспечения безопасного обновления таблиц маршрутизации;
- способность переносить дополнительную информацию о маршрутизации пакетов;

Протокол **RIP-2** предотвращает появление петель в маршрутизации, устанавливая максимально допустимое количество хопов на маршруте.

Стандартное максимальное значение количества хопов равно 15.

# Алгоритмы выбора маршрутов

При получении маршрутизатором информации обновления маршрутов, содержащей новую запись, он увеличивает значение метрики на единицу.

Если при этом значение метрики превышает 15, то считается бесконечно большим, и сеть-получатель считается недостижимой.

Протокол **RIP-2** позволяет также использовать механизмы расщепления горизонта (**split horizon** – один из методов предотвращения петель) и таймеры удержания информации для предотвращения распространения некорректных сведений о маршрутах.

Другой известный протокол класса **DVA – BGP**, используемый между автономными системами.

# Алгоритмы выбора маршрутов

Крупные сети не могут обойтись без периодического обмена сообщениями для описания сети, однако, большинство этих сообщений избыточны, что замедляет обновление RT, а процесс обновления может занимать много времени (десятки секунд).

Поэтому при выходе линий связи из строя несуществующие маршруты могут оставаться в **RT** в течение длительного периода времени.

Трафик, направленный по такому маршруту, не достигнет своего адресата.

Данная проблема решаема, но ни одно из этих решений не является простым.

# Алгоритмы выбора маршрутов

Более совершенны, чем DVA алгоритмы класса LSA.  
В них единицей обмена служит **описание состояния связи**, а **RT** строится в каждом узле на основании полученного множества таких описаний.

При изменении топологии сети **измененные или новые описания** состояний связи быстро распространяются по всей сети, и сходимость получается выше.

Однако, реализация таких протоколов требует дополнительной памяти в маршрутизаторах (для хранения множества описаний состояния связей) и большого быстродействия (для быстрого построения графов).

Примером протокола класса LSA может служить протокол **OSPF** (Open Shortest Path First).

## Алгоритмы выбора маршрутов

Протокол OSPF (RFC 2328) обеспечивает:

- отсутствие ограничений на размер сети (AS), иерархическая структура сети
- **несколько маршрутов** в сторону одного узла (балансировка трафика)
- поддержку безклассовых сетей (CIDR/VLSM)
- передачу обновлений маршрутов с использованием **multicast**-адресов
- быструю сходимость
- использование процедуры authentication при передаче и получении обновлений

# Алгоритмы выбора маршрутов

Для вычисления маршрутов в **RT OSPF** использует алгоритм выбора кратчайшего пути (SPF), который вычисляет путь, имеющий наименьшую стоимость (сумму метрик) между маршрутизаторами.

Метрика маршрута в протоколе OSPF формируется по специальному алгоритму и учитывает следующие параметры:

- **пропускная способность канала**
- величина задержки распространения в канале
- надежность канала
- загруженность канала
- размер MTU интерфейса

**Содержимое RT OSPF будет различное для разных метрик !**

# Алгоритмы выбора маршрутов

Вместо того чтобы обмениваться всеми записями **RT**, как это делают RIP-маршрутизаторы, OSPF-маршрутизаторы хранят **схему** объединенной сети, обновляя ее после каждого изменения топологии сети.

Эта **схема**, называемая **базой данных состояния связей (link state database, LSDB)**, синхронизируется между всеми OSPF-маршрутизаторами и используется для вычисления маршрутов в **RT**.

Для передачи информации состояния маршрута OSPF использует широковещательные сообщения Hello. Для повышения безопасности предусмотрена авторизация процедур.

OSPF-протокол требует резервирования двух мультикастинг-адресов:

224.0.0.5 – для обращения ко всем маршрутизаторам

224.0.0.6 – для обращения к выделенному маршрутизатору (DR)

# Алгоритмы выбора маршрутов

OSPF использует IP непосредственно, т.е. не привлекает протоколы UDP или TCP.

OSPF имеет свой код (**89**) в поле Protocol IP-заголовка.

Код TOS (type of service) в IP-пакетах, содержащих OSPF-сообщения, равен нулю, значение TOS задается в самих пакетах OSPF.

Формат сообщений OSPF:

<b>1</b>			<b>32</b>
<b>Версия</b>	<b>Тип</b>	<b>Длина сообщения</b>	
<b>IP-адрес маршрутизатора отправителя</b>			
<b>Идентификатор области (Area ID)</b>			
<b>Контрольная сумма</b>		<b>Тип аутентификации (0, 1, 2)</b>	
<b>Аутентификация (октеты 0 – 3)</b>			
<b>Аутентификация (октеты 4 – 7)</b>			

# Алгоритмы выбора маршрутов

Типы сообщений OSPF:

<b>Тип</b>	<b>Значение</b>
<b>1</b>	<b>Hello (используется для проверки доступности маршрутизатора).</b>
<b>2</b>	<b>Описание базы данных (топология).</b>
<b>3</b>	<b>Запрос состояния канала.</b>
<b>4</b>	<b>Изменение состояния канала.</b>
<b>5</b>	<b>Подтверждение получения сообщения о статусе канала.</b>

# Алгоритмы выбора маршрутов

Формат сообщения Hello:

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32
0— 3	<b>Version</b>						<b>Type = 1</b>						<b>Packet length</b>																			
4— 7	<b>Router ID</b>																															
8— 11	<b>Area ID</b>																															
12— 15	<b>Checksum</b>																<b>Authentication type</b>															
16— 23	<b>Authentication</b>																															
24— 27	<b>Network mask</b>																															
28— 31	<b>Hello interval</b>																<b>Options</b>								<b>Router priority</b>							
32— 35	<b>время отключения маршрутизатора</b>																															
36— 39	<b>Designated router (IP-адрес DR)</b>																															
40— 43	<b>Backup designated router (IP-адрес BDR)</b>																															
44— 47	<b>Neighbor ID (IP-адрес соседа k )</b>																															
...	<b>...</b>																															

# Метрики протокола OSPF

Наиболее часто в качестве метрики в OSPF используется – пропускная способность интерфейса – BW. За эталонную – берется BW=100 Мбит/с.

Метрика конкретного интерфейса вычисляется:

$$M_{if} = 100 \text{ Мбит/с} / BW_{if}$$

Тип интерфейса	BW (Мбит/с)	M
FE	100	1
10 BT	10	10
E1	2,048	49
DS0	64 кбит/с	1563

# Таймеры протокола OSPF

- **HelloInterval** — Интервал времени в секундах по истечению которого маршрутизатор отправляет следующий hello-пакет с интерфейса. Для широковещательных сетей и сетей точка-точка значение по умолчанию равно 10 секунд. Для нешироковещательных сетей со множественным доступом значение по умолчанию 30 секунд.
- **RouterDeadInterval** — Интервал времени в секундах по истечению которого сосед будет считаться «мертвым». Этот интервал должен быть кратным значению HelloInterval. Как правило, RouterDeadInterval равен 4 интервалам отправки hello-пакетов, то есть 40 секунд.
- **Wait Timer** — Интервал времени в секундах по истечению которого маршрутизатор выберет DR в сети. Его значение равно значению интервала RouterDeadInterval.
- **RxmtInterval** — Интервал времени в секундах по истечению которого маршрутизатор повторно отправит пакет на который не получил подтверждения о получении (например, Database Description пакет или Link State Request пакеты). Это интервал называется также Retransmit interval. Значение интервала 5 секунд.

# Алгоритм работы OSPF

1. Маршрутизаторы обмениваются hello-пакетами через все интерфейсы на которых активирован **OSPF**.
2. Маршрутизаторы становятся соседями, когда они приходят к договоренности об определенных параметрах указанных в их hello-пакетах.
3. Маршрутизаторы пытаются перейти в состояние соседства со своими соседями. Переход в состояние соседства определяется типом маршрутизаторов обменивающихся hello-пакетами и типом сети по которой передаются hello-пакеты.
4. **OSPF** определяет несколько типов сетей и несколько типов маршрутизаторов. Пара маршрутизаторов, находящихся в состоянии соседства синхронизирует между собой базу данных состояния каналов.
5. Каждый маршрутизатор в дальнейшем посылает объявление о состоянии канала (LSA) только маршрутизаторам с которыми он находится в состоянии соседства.

# Алгоритм работы OSPF

6. Каждый роутер получивший LSA от соседа записывает информацию передаваемую в нем в базу данных **LSDB** и рассылает копию объявления всем другим своим соседям.
7. Рассылая объявления через зону, все роутеры строят идентичную базу данных **LSDB**.
8. Когда база данных построена, каждый роутер использует алгоритм кратчайший путь для вычисления графа без петель, который будет описывать кратчайший путь к каждому известному пункту назначения с собой в качестве корня.
9. Этот граф и будет деревом кратчайшего пути для данного роутера.
10. Каждый роутер строит RT из своего дерева кратчайшего пути.

# Алгоритм работы OSPF

В сетях, в которых маршрутизатор через одно соединение подключается больше чем к одному устройству (например, Ethernet) во время работы протокола OSPF для уменьшения трафика, связанного с OSPF сообщениями, выбирается Designated Router (DR), который принимает обновления от всех маршрутизаторов сети и пересылает эти обновления всем другим.

Также выбирается Backup Designated Router (BDR), который становится DR в случае, если основной DR недоступен.

BDR также собирает обновления, которые предназначены для DR. Все остальные маршрутизаторы в сети называются DROTHER.

Сбор обновлений происходит по мультикастному адресу 224.0.0.6, а рассылка обновлений от DR к другим маршрутизаторам - по 224.0.0.5.

# Первоначальный выбор DR и BDR:

1. DR становится маршрутизатор с наивысшим OSPF приоритетом интерфейса, подключенного к сети. Изменить приоритет можно командой

**Router(config-if)#ip ospf priority {0 - 255}.**

2. BDR становится маршрутизатор со вторым наивысшим OSPF приоритетом интерфейса, подключенного к сети.

3. Если OSPF приоритеты интерфейсов одинаковые, DR становится маршрутизатор с наивысшим **router-id**.

# Выбор DR и BDR во время работы:

1. Если в процессе работы текущий DR становится недоступным, роль DR переходит к BDR и происходит новый выбор BDR.
2. Если в сеть подключается новый маршрутизатор, когда выборы DR/BDR закончены, он никогда сразу не становится DR или BDR, даже если имеет больший приоритет или *router-id*. Он может стать BDR на следующих выборах, если станет недоступным текущий DR или BDR
3. Если в процессе работы старый DR восстанавливает работоспособность, он становится DROTHER. Он может стать BDR на следующих выборах, если станет недоступным текущий DR или BDR.
4. Если в процессе работы текущий BDR становится недоступным, происходят выборы нового BDR.

Поэтому, если есть необходимость в назначении DR и BDR конкретных маршрутизаторов, надо:

- или выключить все маршрутизаторы, затем включать сначала DR, затем BDR, затем остальные
- или погасить необходимые интерфейсы на всех маршрутизаторах, затем поднимать сначала DR, затем BDR, затем остальные

# Преимущества OSPF

OSPF имеет следующие преимущества перед RIP:

- маршруты, вычисленные OSPF, не могут быть циклическими
- OSPF обеспечивает масштабируемость для очень больших объединенных сетей
- более быстрая перестройка RT при изменении топологии сети
- Служебный трафик OSPF в разы меньше чем у RIP
- OSPF изначально поддерживал CIDR/VLSM, в RIP – только RIP-2

# Недостатки OSPF

1. Сложность и высокие требования к памяти.
2. OSPF труден в реализации и нуждается в значительном объеме памяти для хранения объявлений о состоянии каналов.
3. По мере увеличения размера базы данных состояния связей увеличивается и требуемый объем памяти, а также время, необходимое для пересчета.

Для решения этой проблемы масштабирования OSPF делит объединенную сеть на области (совокупности соединенных сетей), которые связаны друг с другом через магистральную область.

Каждый маршрутизатор хранит базу данных состояния связей только для тех областей, которые к нему подключены.

# СПАСИБО за ВНИМАНИЕ

