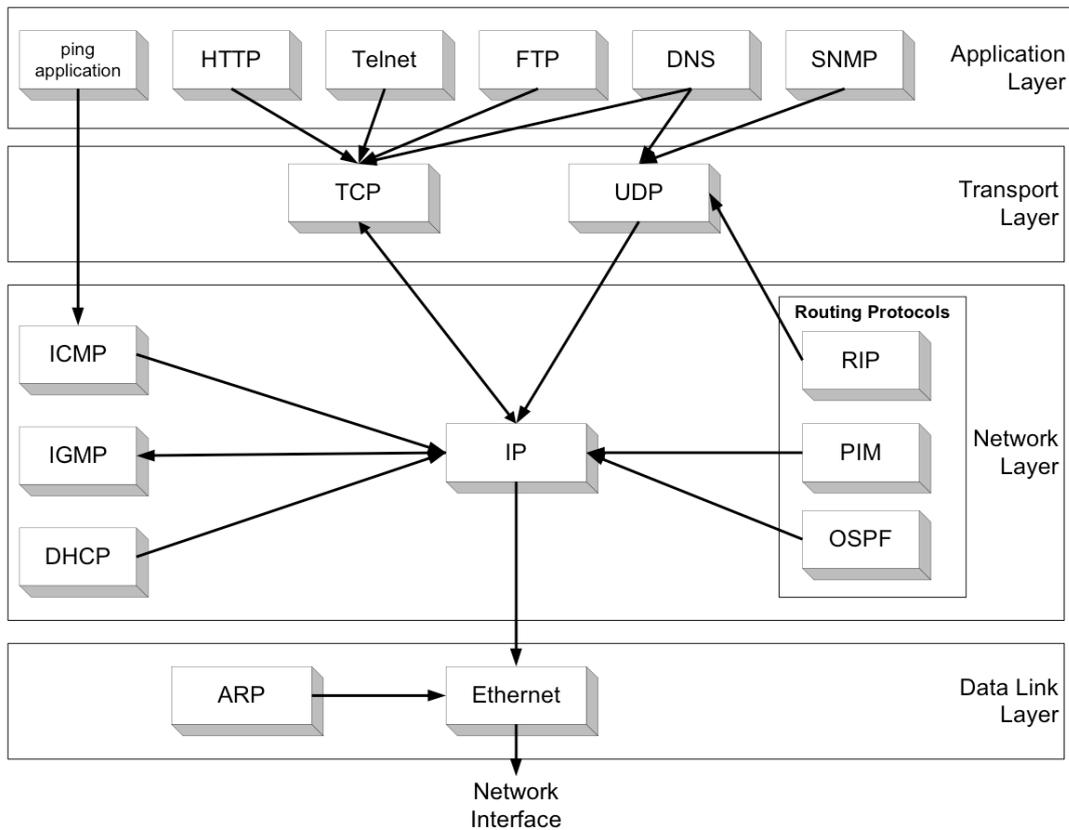


Протокол OSPF.



Введение.

Open Shortest Path First - открыть первым наикратчайший путь.

Протокол является открытым (Open), т.е. является общественным достоянием. Спецификация OSPF опубликована в форме Запроса для Комментария RFC 1247 (– Request for Comments) в 1991 г.

Протокол базируется на алгоритме SPF (Shortest Path First). Алгоритм SPF иногда называют алгоритмом Dijkstra по имени автора, который его разработал в 1959 г.



OSPF непосредственно инкапсулируется в IP без TCP/UDP. Номер протокола 89. Для передачи пакетов OSPF может использовать мультикаст адреса:

- 224.0.0.5 все маршрутизаторы OSPF;
- 224.0.0.6 все DR и BDR (Designated Router - выделенный маршрутизатор).

OSPF это протокол внутренних маршрутизаторов, является альтернативой RIP.

В OSPF маршрутизатор принимает информацию о состоянии каналов **непосредственно** от передающих маршрутизаторов и на её основании сам строит полную таблицу маршрутизации. В RIP маршрутизатор обновляет свою таблицу маршрутизации на основании векторов расстояний, которые он получает **косвенно** от своих соседей.

1. Преимущества OSPF в сравнении с RIP.

- LSA значительно быстрее, чем DVA осуществляется сходимость сети. Под сходимостью (converge) подразумевают стабилизацию сети после каких-либо изменений в ней.
- Каждому интерфейсу назначается цена (метрика). Она может быть назначена на основании пропускной способности, времени RTT, надежности или по какому-либо другому параметру.
- OSPF может рассчитать отдельный маршрут для каждого TOS IP (type-of-service), т.е. для любого пункта назначения в ТМ может быть несколько записей, по типам сервиса.
- Если к цели есть маршруты с близкой метрикой, то OSPF может осуществлять балансировку загруженности (распределять трафик обратно пропорционально метрике).
- OSPF поддерживает подсети: маска подсети приписана каждому объявленному маршруту. Маршруты к хостам объявляются с маской подсети, из всех единичных бит.
- OSPF маршрутизаторы могут поддерживать point-to-point и point-to-multipoint (broadcast и multicast сообщения) каналы связи.
- Каналы точка-точка между роутерами не имеют IP адресов на концах - это называется сетями без адреса (unnumbered), позволяет сэкономить IP адреса и работу админов.
- OSPF могут использовать групповую адресацию вместо широковещательной, что уменьшает загруженность систем в сети, которые не работают с OSPF.
- Используется простая схема аутентификации. Может быть указан пароль в виде открытого текста, так же как это делается в схеме RIP-2.

Заключение. В OSPF сняты все ограничения, присущие для RIP, что делают его более предпочтительным. Так как большинство поставщиков маршрутизаторов поддерживают OSPF, он начинает постепенно замещать собой RIP в большинстве сетей.

2. Формат заголовка пакета OSPF.

Пакеты OSPF инкапсулируются в IP-пакет с номером протокола 89. Существует пять типов пакетов OSPF. У всех пакетов OSPF одинаковый заголовок.

| Octet | 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 |
|-------|----------------|---|---|---|---|---|------|---|---|---|----|----|---------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| 00 | Version # | | | | | | Type | | | | | | Packet Length | | | | | | | | | | | | | | | | | | | |
| 04 | Router ID | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 08 | Area ID | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 12 | Checksum | | | | | | | | | | | | AuthType | | | | | | | | | | | | | | | | | | | |
| 16 | Authentication | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 20 | Authentication | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 24 | Data | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

1. Версия протокола (Version #) — версия протокола OSPF. Текущая версия для IPv4 = 2.
2. Тип пакета (Type) — указывает какой тип пакета OSPF передается:
 - 1 — Hello
 - 2 — Database Description (DBD)
 - 3 — Link State Request (LSR)
 - 4 — Link State Update (LSU)
 - 5 — Link State Acknowledgment (LSAck)

3. Длина пакета (Packet length) — длина пакета OSPF в байтах. Длина включает в себя и заголовок.
4. Идентификатор маршрутизатора (Router ID) — определяет какой маршрутизатор отправил пакет.
5. Идентификатор зоны (Area ID) — определяет в какой зоне сгенерирован пакет.
6. Контрольная сумма (Checksum) — используется для проверки целостности пакета OSPF, для обнаружения ошибок при передаче.
7. Тип аутентификации (Authentication type) — тип аутентификации, который используется между маршрутизаторами:
 - 0 — аутентификация не используется,
 - 1 — аутентификация открытым текстом,
 - 2 — MD5-аутентификация.
8. Данные аутентификации (Authentication) — используется при аутентификации маршрутизаторов.
9. Поле Данные отличается для различных типов пакетов OSPF:
 - Hello — список известных соседей
 - DBD — содержит суммарную информацию базы данных состояний каналов, которая включает в себя все известные идентификаторы маршрутизаторов и их последние номера последовательностей (sequence number) и другую информацию.
 - LSR — содержит тип необходимого LSU и идентификатор маршрутизатора, у которого есть этот LSU.
 - LSU — содержит полные записи объявления о состоянии канала. Несколько LSA могут передаваться в одном пакете обновлений.
 - LSAck — поле данных пустое.

3. Протокол OSPF.

3.1. Терминология протокола OSPF.

Базовые термины:

- Канал/интерфейс (link/interface) — соединение маршрутизатора и одной из подключенных к нему сетей, термины канал и интерфейс часто употребляются как синонимы.
- Метрика (metric) — условный показатель расстояния до сети назначения.
- Стоимость (cost) — условный показатель "стоимости" пересылки данных по каналу.
- Автономная система (autonomous system) — группа маршрутизаторов, обменивающаяся маршрутизирующей информацией с помощью одного протокола маршрутизации.

Базовые термины OSPF:

- Идентификатор маршрутизатора (RID) — уникальное 32-битовое число, которое уникально идентифицирует маршрутизатор в пределах одной AS.
- Зона (area) — совокупность сетей и маршрутизаторов, имеющих один и тот же идентификатор зоны.
- Объявление о состоянии канала (link-state advertisement, LSA) — единица данных, которая описывает локальное состояние маршрутизатора или сети. Например, для маршрутизатора LSA1 включает описание состояния каналов и отношений соседства.

- База данных состояния каналов (link state database, LSDB) — список всех записей о состоянии каналов (LSA) описывающих маршрутизаторы и сети. Аналогичный термин топологическая база данных (topological database), употребляется как синоним LSDB.

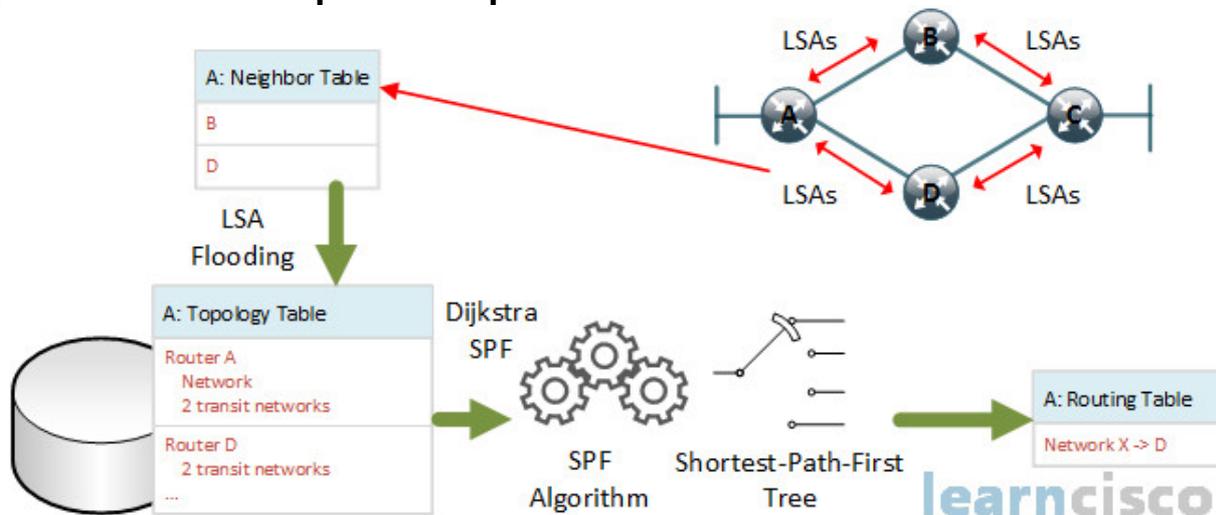
Соседи OSPF:

- Соседи (neighbours) — два маршрутизатора, интерфейсы которых находятся в одном широковещательном сегменте (и на которых включен OSPF на этих интерфейсах).
- Отношения соседства (adjacency) — взаимосвязь между соседними маршрутизаторами, установленная с целью синхронизации информации.
- Hello-протокол (hello protocol) — протокол, использующийся для установки и поддержания соседских отношений.
- База данных соседей (neighbours database) — список всех соседей (также используется термин neighbour table).

Пакеты OSPF:

- Hello пакеты используются для обнаружения соседей, установки отношений соседства и мониторинга их доступности (keepalive)
- DBD пакеты описывают содержание LSDB
- LSR пакеты (request), с помощью которых запрашивается у других полная информация об LSA, которых недостает в LSDB локального маршрутизатора.
- LSU пакеты (update), которые передают полную информацию, которая содержится в LSA.
- LSAck пакеты, с помощью которых подтверждается получение других пакетов

3.2. Краткое описание работы протокола OSPF.



1. Идентификатор. Каждый маршрутизатор имеет уникальный в AS идентификатор (32-битовое число), который назначается админом или определяется автоматически (например, как наименьший или наибольший из собственных IP маршрутизатора).

2. HELLO. Для выяснения состояния связей маршрутизаторы-соседи каждые 10-30 сек обмениваются короткими сообщениями HELLO. При появлении нового соседа, маршрутизатор узнает об этом из новых пакетов HELLO.

3. LSA. Для распространения по сети объявлений о состоянии связей, маршрутизаторы обмениваются ретранслируемыми сообщениями типа LSA (Link State Advertisement), получая в конце концов информацию о топологии всей сети. Есть 6 типов LSA объявлений. LSA извещения рассылаются всем маршрутизаторам OSPF, что сокращает время сходимости. Для предупреждения лавинных рассылок LSA (для n маршрутизаторов сеть может сгенерировать n^2 сообщений) могут использоваться выделенные маршрутизаторы (DR) и запасные BDR (Backup DR) (уменьшают рассылки до величины $n+2$ сообщений).

4. Изменения в сети. Поддерживаются через обновления LS и Flooding protocol.

5. LSDB. На основе HELLO, LSA, LSU информации формируется и поддерживается **база данных связей**, одинаковая для всех маршрутизаторов сети.

6. Орграф связей и SPF Tree. Далее строится орграф - вершинами графа являются маршрутизаторы, а рёбрами – стоимости связи. Каждый маршрутизатор на основе орграфа строит по алгоритму Dijkstra дерево наикратчайших путей. Маршрутизатор повторно запускает алгоритм SPF Tree если возникли изменения в LSDB.

7. Таблица маршрутизации. Хотя маршрутизатор и вычисляет весь оптимальный маршрут до каждой адресуемой сети, но использует (заносит в свою ТМ) только первый промежуточный маршрутизатор из каждого маршрута.

3.3. Изменения, LS-record и Flooding protocol.

Каждый маршрутизатор отвечает за те записи в LSDB, которые описывают связи, исходящие от данного маршрутизатора. Это значит, что при образовании новой связи, изменении в состоянии связи или ее исчезновении (обрыве), маршрутизатор должен изменить свою копию LSDB и **немедленно известить все остальные маршрутизаторы** OSPF-системы о произошедших изменениях, чтобы они также внесли исправления в свои копии базы данных.

Эту задачу в OSPF выполняет **Flooding protocol (затопление)** - всеми маршрутизаторами пересылаются (ретранслируются) сообщения типа "Обновление состояния связей (Link State Update)", получение которых подтверждается сообщениями типа "Link State Acknowledgment".

Каждая запись в LSDB о состоянии связей LS имеет свой **номер версии**. Каждая новая версия записи имеет больший номер, что предотвращает попадание в базы устаревших версий.

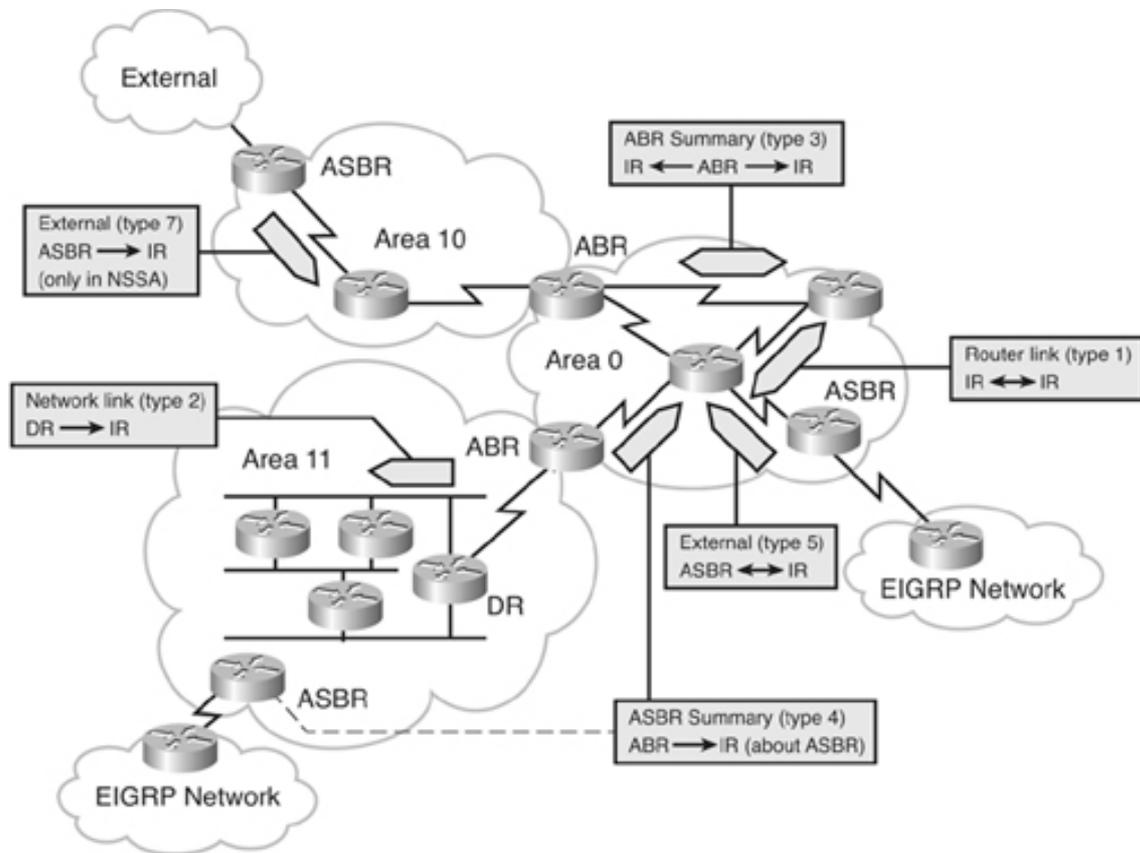
Протокол OSPF поддерживает **возраст LS-записи** в LSDB. При создании (подтверждении) записи возраст устанавливается в ноль. Каждый маршрутизатор, который ретранслирует сообщение, увеличивает возраст записи на определенную величину. Кроме этого, возраст увеличивается на единицу каждую секунду, а также есть рассинхронизация таймеров.

При достижении записью **Max возраста** =60 мин, она удаляется из LSDB (и SPF Tree) и обнаруживший маршрутизатором инициирует протокол затопления об устаревшей записи. Чтобы записи в LSDB не устаревали, ответственные маршрутизаторы каждые 30 минут затопливают систему сообщениями об обновлении записей. Содержимое записей в этих сообщениях неизменно, но номер версии больше, а возраст равен нулю.

3.4. Таймеры протокола OSPF.

1. **HelloInterval** — интервал времени по истечении которого маршрутизатор отправляет следующий hello-пакет с интерфейса. Для широковещательных сетей и сетей точка-точка значение по умолчанию, как правило, =10 сек. Для нешироковещательных сетей со множественным доступом значение по умолчанию =30 сек.
2. **RouterDeadInterval** — интервал времени по истечении которого сосед будет считаться "мертвым" (dead). Этот интервал должен быть кратным значению HelloInterval. Как правило, RouterDeadInterval равен 4 HelloInterval, то есть =40 сек.
3. **WaitTimer** — интервал времени по истечении которого маршрутизатор выберет DR (маршрутизатор с самым малым RID) в сети. Его значение равно значению интервала RouterDeadInterval, то есть =40 сек.
4. **LSttl** – время устаревания LS записи в LSDB, значение по умолчанию =60 мин.
5. **LSttlNew** - обновление LS записей в LSDB, значение по умолчанию =30 мин.
6. **RxmtInterval** — интервал времени по истечении 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.

3.5. Области OSPF и типы LSA.



3.5.1. Типы областей и маршрутизаторов OSPF.

OSPF работает лишь в пределах автономной системы (**AS**). AS может быть поделена на отдельные области (**Area**), каждая из областей может стать объектом маршрутизации, а внутренняя структура снаружи не будет видна.

В пределах выделенной тупиковой области может работать свой протокол маршрутизации, не обязательно OSPF, например, RIP или EIGRP.

В OSPF используется термин опорной сети (**Backbone**) для коммуникаций между областями. Маршрутизаторы областей обмениваются сообщениями LSA разных типов.

Существуют и другие типы областей и LSA, например, полностью тупиковая (totally stubby) и тупиковая область (stubby area) – хождение LSA5 в них запрещено; но, в NSSA (not so stubby area – не совсем тупиковая область) передавать внешние маршруты LSA7 разрешено, они будут конвертированы на ABR в LSA5.

Иерархическая структура областей и **Address Summarization** уменьшают перегрузки в сети, связанные с поддержкой огромных баз данных и с пересчётом таблиц маршрутизации при изменениях маршрутов.

С точки зрения протокола OSPF, имеются маршрутизаторы нескольких типов:

1. IR (Internal Router) - внутренний маршрутизатор, все интерфейсы которого находятся внутри одной области OSPF;

2. BR (Backbone Router) - маршрутизатор опорной области, все интерфейсы которого находятся внутри опорной области (Area0);
3. ABR (Area Border Router) - пограничный маршрутизатор области, располагающийся на границе двух областей OSPF;
4. ASBR (Autonomous System Boundary Router) - пограничный маршрутизатор AS, который располагается на границе двух автономных систем с поддержкой OSPF.
5. DR (Designated Router) - назначенный маршрутизатор и резервный назначенный маршрутизатор BDR (Backup), которые являются центральными узлами сбора всех сообщений о корректировках от (для) подчиненных маршрутизаторов области.

Один и тот же маршрутизатор может выполнять в системе несколько функций одновременно.

3.5.2. Суммарная информация о типах LSA.

| Тип LSA | Название LSA | Link-State ID | Кто отправляет | Область распространения |
|------------------|------------------------|-------------------------|---------------------------|-------------------------|
| LSA1 | Router LSA | Router ID отправителя | Все маршрутизаторы | Внутри (IntraArea) |
| LSA2 | Network LSA | IP-адрес интерфейса DR | DR в сети с multi доступ. | Внутри (IntraArea) |
| LSA3 | Network Summary LSA | Сети назначения и маски | ABR | AS (InterArea) |
| LSA4 | ASBR Summary LSA | Router ID ASBR | ABR | AS (InterArea) |
| LSA5 | AS External LSA | Внешняя сеть и маска | ASBR | AS (InterArea) |
| LSA6 | Multicast OSPF LSA | | | |
| LSA7 | AS Exter. LSA for NSSA | Внешняя сеть и маска | ASBR только в NSSA | NSSA |
| LSA6, 8,9, 10,11 | Opaque LSAs | | | |

3.6. Метрики OSPF и их стоимость.

В OSPF метрика представляет собой оценку эффективности связи в канале связи: **чем меньше стоимость метрики, тем эффективнее организация связи**. Стоимость полного составного маршрута равна сумме стоимости метрик всех связей, входящих в маршрут.

Для того чтобы обозначить недоступную сеть, OSPF использует стоимость метрики равную **(2²⁴-1=16 777 215)**, которая считается **недостижимой метрикой** для OSPF.

Для каждой из метрик протокол OSPF строит отдельную таблицу маршрутизации. Чаще всего в OSPF выбирают маршрут на основании полосы пропускания канала.

Часть метрик имеет стандартизованный порядок расчёта стоимости, но, для метрик, оценивающих надёжность, задержку и цену передачи, порядок пока не определён. Эти вопросы решаются администратором сети.

3.6.1. Метрика – полоса пропускания канала.

Стоимость метрики, оценивающая пропускную способность канала вычисляется по формуле:

$$\text{cost} = \text{reference bandwidth} / \text{link bandwidth}$$

Reference bandwidth (эталонная полоса пропускания) может определяться производителями маршрутизаторов различным образом. Например, компания CISCO, определяет эту метрику как количество секунд, нужное для передачи 100 Мбит информации через канал сети.

Например, по этой формуле вычислена стоимость метрики следующих каналов связи:

- канал со скоростью 100 Мбит/с соответствует метрике 1;
- сеть Ethernet / 802.3 (10 Мбит/с) соответствует метрике 10;
- тракт E1 2,048 Мбит/с соответствует метрике 48;
- тракт T1 1,544 Мбит/с соответствует метрике 65;
- канал 64 Кбит/с соответствует метрике 1562;
- канал 56 Кбит/с соответствует метрике 1785;
- канал 19,2 Кбит/с соответствует метрике 5208;
- канал 9,6 Кбит/с соответствует метрике 10416.

3.6.2. Метрика - хопы.

В простейшем случае стоимость метрики маршрута может равняться его длине в пересылках (hops), как это происходит в протоколе RIP.

3.6.3. Метрика – загрузка канала.

Загрузка канала изменяется в зависимости от интенсивности использования канала, и поэтому при маршрутизации бывает целесообразно выбирать менее нагруженные каналы. В случае, когда имеется несколько маршрутов с одинаковым значением метрики, маршрутизаторы могут использовать для передачи пакетов все эти маршруты, обеспечивая балансировку нагрузки. Маршрутизатор OSPF помещает в таблицу маршрутизации все маршруты с одинаковыми (или близкими) метриками, и балансировка нагрузки между маршрутами происходит автоматически.

3.6.4. Метрика – задержка.

Задержка – определяет время распространения пакета в микросекундах, которое требуется маршрутизатору для обработки, установки в очередь и передачи пакетов.

3.6.5. Метрика – цена.

Определяет цену передачи единицы данных через данный канал, возможно, в зависимости от времени суток.

3.6.6. Метрика – тип услуг.

Протокол OSPF позволяет определить для любой сети значения метрики в зависимости от типа услуги ToS (Type of Service). В OSPF последних версиях ToS не используется.

3.6.7. Другие метрики.

- надёжность связи;
- число дейтаграмм, стоящих в очереди для передачи;
- требования безопасности;
- число шагов до цели;
- возможностями промежуточных связей (многовариантность достижения адресата).

3.7. Заключение.

OSPF представляет собой протокол, определяющий отношения соседства и смежности с другими маршрутизаторами, осуществляющий контроль состояния каналов, распространяющий эту информацию и определяющий на ее основе маршруты наименьшей стоимости в заданной метрике (алгоритм SPF).

SPF-алгоритм работает на основе взвешенного орграфа сети, представляющего некоторую совокупности маршрутизаторов (область), где для каждой пары смежных вершин графа (маршрутизаторов) указано ребро (канал), их соединяющее, и метрика (вес) этого ребра. Граф считается ориентированным, т.е. ребро, соединяющее R1 с R2, и ребро, соединяющее R2 с R1, могут быть разными, или это может быть одно и то же ребро, но с разными метриками в разных направлениях.

Все вычисления OSPF производит локально по известной базе данных состояния связей LSDB, а потому - быстро по сравнению с дистанционно-векторными протоколами (RIP), при этом результаты получаются на основе полной, а не частичной информации о графе связей системы сетей, что уменьшает время реакции на изменения в сети (сходимость).

Более подробное описание OSPF см. на <http://xgu.ru/wiki/ospf>

Для практического закрепления материала учащимся предлагается освоить программу GraphTea и выполнить лабораторную работу, см. методические указания «Алгоритмы на графах для компьютерных сетей: Поиск вершин, Минимальное остовое дерево, Дерево минимальных путей, Максимальный поток».

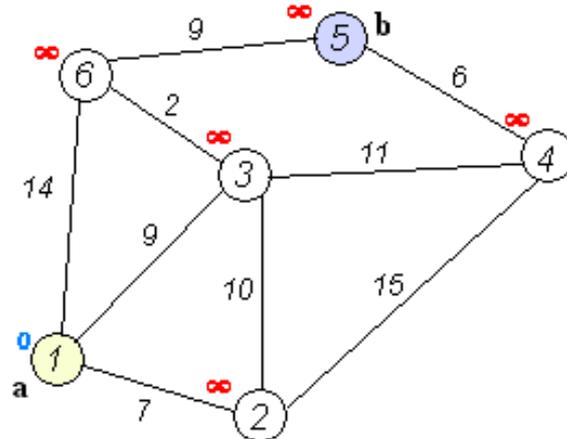
4. Алгоритм SPF Tree (Дейкстры).

Алгоритм SPF был предложен Е.В. Дейкстрой (Edsger Wybe Dijkstra), известным программистом и физиком-теоретиком. Прочитайте статью о жизни Дейкстры на веб-сайте Ассоциации по вычислительной технике (ACM) по адресу:

http://amturing.acm.org/award_winners/dijkstra_1053701.cfm.

Посмотрите анимацию Дейкстры (Dijkstra) о поиске кратчайшего пути, которую можно найти по адресу:

http://upload.wikimedia.org/wikipedia/commons/5/57/Dijkstra_Animation.gif.



Образное описание алгоритма SPF Tree. Представим изображенную на рис. сеть как набор из 6 фишек, лежащих на поверхности стола и соединенных между собой нитями разной длины. Пусть, например, алгоритм Дейкстры выполняется в маршрутизаторе 1. Постепенно поднимаем со стола фишку 1. Нити, связывающие эту фишку с другими, начинают натягиваться, и следующей со стола будет поднята фишка 2, связанная с 1 самой короткой нитью. При дальнейшем подъеме фишки 1 мы поднимем с поверхности стола фишку 3. Далее 3 поднимет 6. Кратчайший путь между a и b представит составной путь из нитей между фишками 1, 3, 6 и 5. Продолжая процедуру подъема фишки 1, мы шаг за шагом поднимем все фишки, находя каждый раз кратчайший путь между 1 и тем маршрутизатором, которому соответствует очередная поднимаемая нить фишка.

4.1. Неформальное описание алгоритма SPF Tree.

Алгоритм Дейкстры решает задачу о кратчайших путях из одной исходной вершины во все остальные вершины для взвешенного ориентированного графа. На каждом шаге алгоритм «посещает» одну вершину и пытается уменьшать метки стоимости путей.

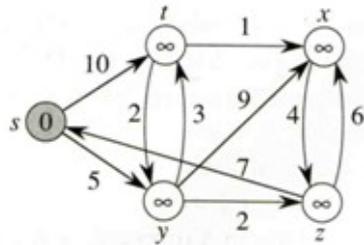
Инициализация. Все вершины графа помечаются как непосещённые. Метка стоимости исходной вершины к себе полагается равной 0, метки остальных вершин — бесконечности (∞). Это отражает то, что расстояния до других вершин пока неизвестны. Работа алгоритма завершается, когда все вершины посещены.

Шаг алгоритма. Из ещё не посещенных вершин выбирается вершина u , имеющая минимальную метку. Рассматриваются всевозможные маршруты, в которых u является предпоследним пунктом. Вершины, соединенные с вершиной u ребрами, называют соседями этой вершины. Для каждого соседа рассматривают новую метку (длину) пути, равную сумме текущей метки u и длины ребра, соединяющего u с этим соседом. Если полученная длина меньше метки соседа, заменяют метку этой длиной. Рассмотрев всех соседей, помечают вершину u как посещенную и повторяют шаг.

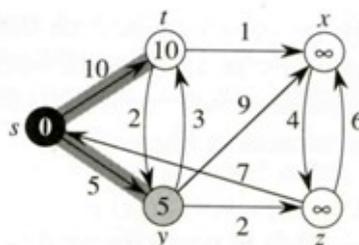
Результатом работы алгоритма является список наикратчайших путей, соединяющих заданную вершину с каждой вершиной. Для другой исходных вершины, таблица и топология выбранных путей будут другими.

Пример пошаговой работы алгоритма отражена на рисунке и в таблице ниже. В (скобках) записана стоимость кратчайшего отобранного маршрута к вершине на данном шаге.

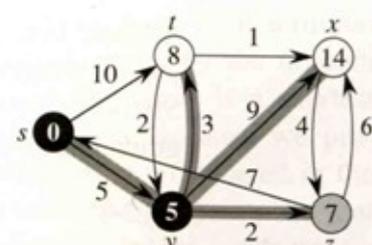
4.2. Пример 1 работы алгоритма SPF Tree для взвешенного орграфа.



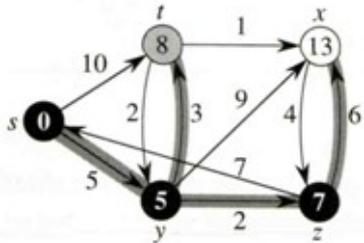
(a)



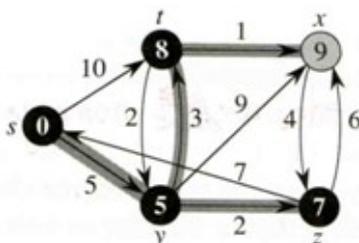
(b)



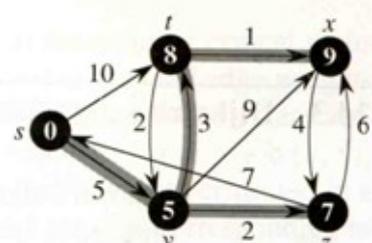
(c)



(d)



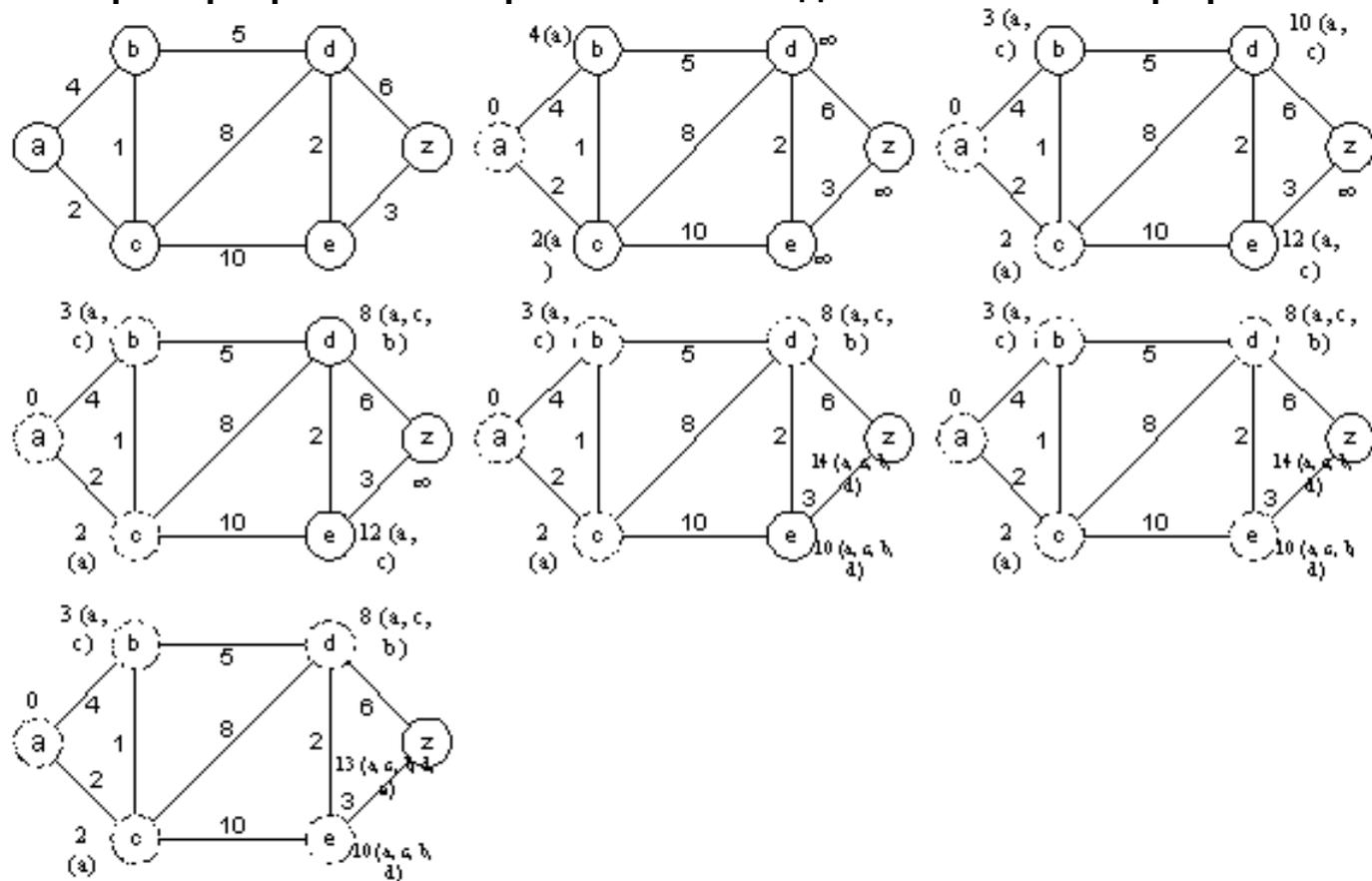
(e)



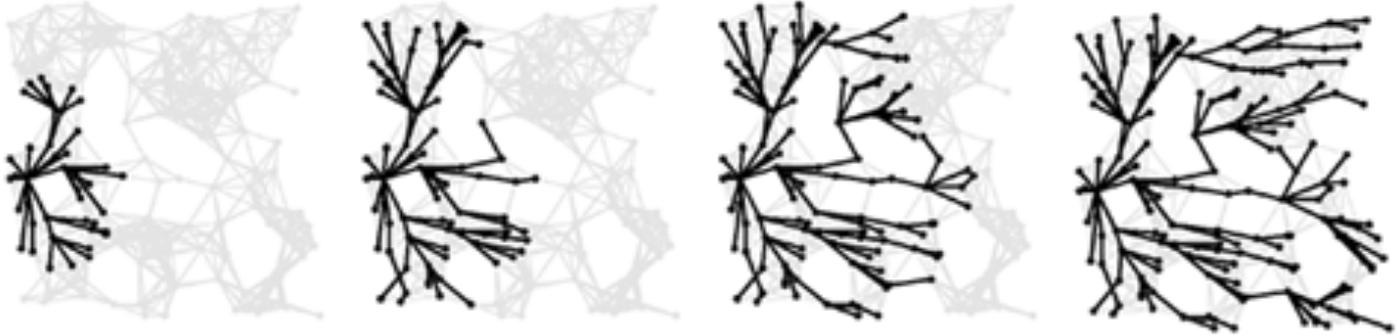
(f)

| Шаг i | Множество посещённых вершин | Метки стоимости и путь связи вершины s с вершинами | | | | |
|----------|--------------------------------|--|---------|-------------|--------|-------------|
| | | s | t | x | y | z |
| 0 | {s} | 0-s | 10-st | ∞ -s | 5-sy | ∞ -s |
| 1 | {s,y} | (0)-s | 8-syt | 14-syx | (5)-sy | 7-sy |
| 2 | {s,y,z} | | 8-syt | 13-syzx | | (7)-sy |
| 3 | {s,y,z,t} | | (8)-syt | 9-sytx | | |
| 4 | {s,y,z,t,x} | | | (9)-sytx | | |

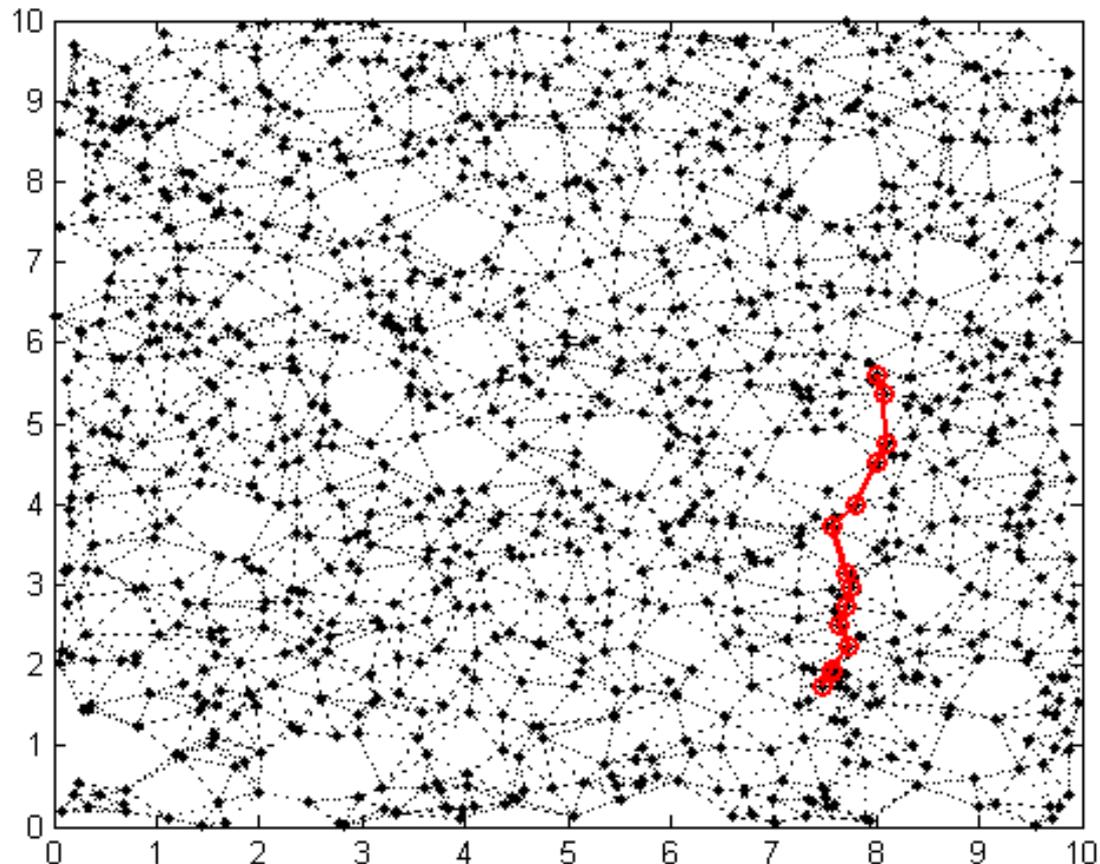
4.3. Пример 2 работы алгоритма SPF Tree для взвешенного графа.



4.5. Размеры реальных сетей.



Distance from 1 to 1000 = 4.118



Path = [1 249 544 689 239 793 128 485 721 217 777 747 174 1000]

5. Traffic Engineering.

5.1. Об MPLS.

MPLS (Multiprotocol Label Switching) - многопротокольная коммутация по меткам – это технология, разработанная рабочей группой по созданию интегрированных услуг IETF (в 1996г.) и объединяющая фирменные разработки Ipsilon, Cisco, IBM и других компаний.

MPLS - это архитектура построения магистральных сетей реализующая возможность передачи трафика по наименее загруженным маршрутам IP-сети и обеспечивающая лёгкость конфигурирования VPN с одновременной поддержкой гарантии качества передачи, а также присвоения приоритетов различным видам трафика.

MPLS является масштабируемым и независимым от каких-либо протоколов механизмом передачи данных.

На сегодняшний день существуют три основные области применения MPLS:

1. управление трафиком (TE - Traffic Engineering);
2. поддержка классов обслуживания (CoS);
3. организация виртуальных частных сетей (VPN).

Рассмотрим лишь первую область применения MPLS, а именно Traffic Engineering.

5.2. Недостатки традиционной маршрутизации IP-трафика.

При традиционной маршрутизации IP-трафик маршрутизируется посредством его передачи от одной точки назначения к другой и следует до пункта назначения по пути, имеющему наименьшую суммарную метрику сетевого уровня.

1. Следует заметить, что при наличии в сети нескольких равноценных альтернативных маршрутов трафик может делиться между ними, и нагрузка на маршрутизаторы и каналы связи распределяется более сбалансированно. Но, в целом, **так как маршруты не являются полностью равноценными, распределение трафика между ними не происходит.**

2. Вторым существенным недостатком традиционных методов маршрутизации трафика в сетях IP заключается в том, что **пути выбираются без учета текущей загрузки ресурсов сети.** Если кратчайший путь уже перегружен, то пакеты все равно будут посылаться по этому пути.

Налицо явная ущербность методов распределения ресурсов сети – одни из них работают с перегрузкой, а другие не используются вовсе. Никакие методы QoS данную проблему решить не могут: **нужны качественно иные механизмы.**

Достаточно эффективный механизм использования ресурсов сети является **Технология управления трафиком (Traffic Engineering)** - это методы и механизмы сбалансированной загрузки всех ресурсов сети за счет рационального выбора пути прохождения трафика через сеть. Механизм ТЕ предоставляет возможность **устанавливать явный путь, по которому будут передаваться потоки данных.**

5.3. Traffic Engineering и OSPF.

Для решения задачи Traffic Engineering (TE) технология MPLS использует расширения протоколов маршрутизации, работающих на основе алгоритма состояния связей.

Сегодня такие расширения стандартизованы для протоколов OSPF и IS-IS. Данные протоколы, в отличие от дистанционно-векторных протоколов, к которым относится, например, RIP, дают маршрутизатору полную топологическую информацию о сети. Их объявления содержат информацию о маршрутизаторах и сетях, а также о физических связях между ними. Каждая связь характеризуется текущим состоянием работоспособности и метрикой, в качестве которой используется величина, обратная пропускной способности канала.

Для решения задачи TE в протоколы OSPF и IS-IS включены **новые типы объявлений для распространения по сети информации о номинальной и незарезервированной (доступной для потоков TE) пропускной способности каждой связи.**

Таким образом, ребра результирующего графа сети, создаваемого в топологической базе каждого маршрутизатора, будут маркированы этими двумя дополнительными параметрами (см. рис. ниже).

Располагая таким графом, а также параметрами потоков, для которых нужно определить пути TE, маршрутизатор может найти **рациональное решение**, удовлетворяющее, например, некоторому ограничению на коэффициенты использования ресурсов сети, обеспечив тем самым ее сбалансированную загрузку.

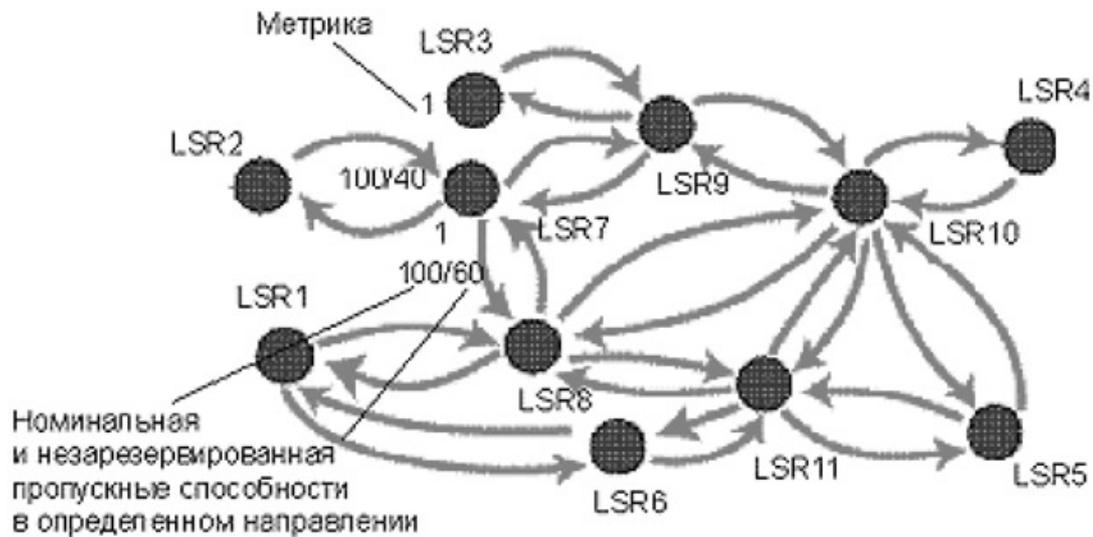


Рис. Граф сети с дополнительными показателями загруженности каналов.

5.3.1. Последовательное рассмотрение потоков.

Для упрощения задачи оптимизации выбор путей для некоторого набора потоков может **осуществляется по очереди**, при этом в качестве ограничения выступает суммарная загрузка каждого ресурса сети.

Обычно считается, что внутренней производительности маршрутизатора достаточно (в среднем) для обслуживания любого трафика, который способны принять интерфейсы маршрутизатора. Поэтому в качестве ограничений выступают только максимально допустимые значения коэффициентов загрузки каналов связи, устанавливаемые индивидуально или же имеющее общее значение. Решение задачи определения маршрута с учетом ограничений получило название **Constrained-based Routing**, а протокол OSPF с соответствующими расширениями – Constrained SPF, или CSPF.

Пример.

В примере, показанном на рис. ниже, ограничением является максимально допустимое значение коэффициента использования ресурсов, равное 0,65.

В варианте 1 на рис. решение было найдено при очередности рассмотрения потоков 1→2→3.

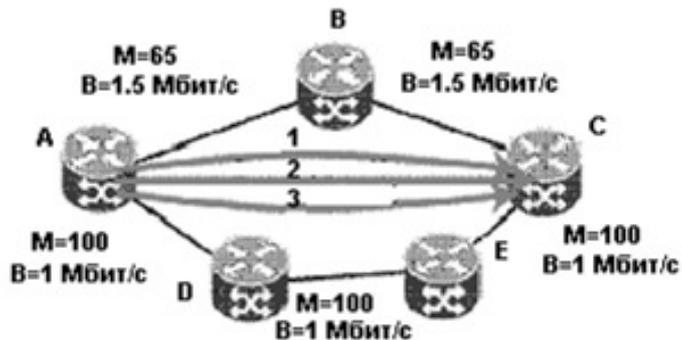
Для первого потока был выбран путь А-В-С, так как в этом случае он, с одной стороны, удовлетворяет ограничению (все ресурсы вдоль пути – каналы А-В, А-С и соответствующие интерфейсы маршрутизаторов оказываются загруженными на $0,5/1,5 = 0,33$), а с другой – обладает минимальной метрикой ($65 + 65 = 130$).

Для второго потока также был выбран путь A-B-C, так как и в этом случае ограничение удовлетворяется – результирующий коэффициент использования оказывается равным $(0,5 + 0,4)/1,5 = 0,6$.

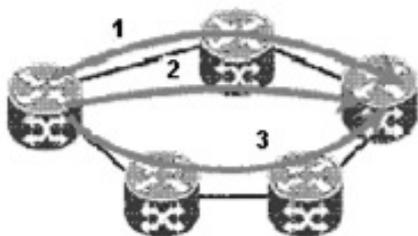
Третий поток направляется по пути A-D-E-C и загружает ресурсы каналов A-D, D-E и E-C на 0,3 (метод расчета метрик канала был описан в предыдущих пунктах).

Решение 1 можно назвать удовлетворительным, так как коэффициент использования любого ресурса в сети не превышает 0,6.

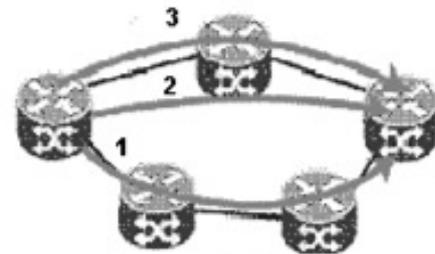
Очевидно, что поиск путей TE по очереди снижает качество решения – при одновременном рассмотрении всех потоков можно найти более рациональную загрузку ресурсов.



1 поток, $V=0.5$ Мбит/с
 2 поток, $V=0.4$ Мбит/с
 3 поток, $V=0.3$ Мбит/с



Вариант 1: 1-2-3
 $K_{\max} = 0,6$



Вариант 2: 2-3-1
 $K_{\max}=0,5$

Рис. Варианты загрузки ресурсов.

5.3.2. Одновременное рассмотрение потоков.

Существует лучший способ, представленный в варианте 2.

Здесь по верхнему пути A-B-C были направлены потоки 2 и 3, а поток 1 – по нижнему пути A-D-E-C.

Ресурсы верхнего пути оказываются загружены на 0,46, а нижнего – на 0,5, т. е. налицо более равномерная загрузка ресурсов, а максимальный коэффициент использования по всем ресурсам сети не превышает 0,5.

Этот вариант может быть получен при одновременном рассмотрении всех трех потоков с учетом ограничения $\min(\max K_i)$ или же при рассмотрении потоков по очереди в последовательности 2 → 3 → 1.

Тем не менее в производимом сегодня оборудовании применяется вариант MPLS TE с последовательным рассмотрением потоков. Он проще в реализации и ближе к стандартным для протоколов OSPF и IS-IS процедурам нахождения кратчайшего пути для одной сети назначения.