

# Технологія MPLS як транспорта технологія для входу до ГІІ

Лекція №6

Навчальні питання:

1. Введення до MPLS
2. Принцип роботи технології MPLS.
3. Мітки та стек міток.
4. Режими комутації по міткам

# Історична довідка

- В 1996 году Ipsilon, Cisco, IBM и несколько других компаний объединили свои фирменные разработки и создали **технологію многопротокольной коммутации на основе меток (MPLS – Multiprotocol Label Switching)**.
- **Основная идея разработки** - реализовать возможность передачи трафика по наименее загруженным маршрутам IP-сети и обеспечить легкость конфигурирования VPN с одновременной поддержкой гарантии качества передачи, а также присвоения приоритетов различным видам трафика.

# 1. Введение в MPLS

- **Многопротокольная коммутация по меткам MPLS**
- Это новая архитектура построения магистральных сетей, которая значительно расширяет имеющиеся перспективы масштабирования, повышает скорость обработки трафика и предоставляет огромные возможности для организации дополнительных услуг.
- сочетает в себе возможности управления трафиком, присущие технологиям канального уровня, и масштабируемость и гибкость протоколов, характерные для сетевого уровня.
- Достоинства MPLS:
  - соединила в себе надежность АТМ, удобные и мощные средства доставки и обеспечения гарантированного качества обслуживания IP-сетей,
  - позволяет совместного использования протоколов IP и АТМ.

# Особенности технологии MPLS

- отделение процесса коммутации пакета от анализа IP-адреса в его заголовке, что позволяет осуществлять коммутацию пакетов значительно быстрее.
- маршрутизаторы и коммутаторы присваивают на каждой точке входа в таблицу маршрутизации особую **метку** и сообщают эту метку соседним устройствам.

# Основные области применения MPLS:

- управление трафиком;
- поддержка классов обслуживания (QoS);
- организация виртуальных частных сетей (VPN).

# Расположение технологии MPLS в семиуровневой модели ВОС



Плоскость пересылки данных MPLS не образует полноценного уровня, она "вклинивается" в сети IP, ATM или Frame Relay между 2-м и 3-м уровнями модели OSI, оставаясь независимой от этих уровней, приводит к образованию так **называемого уровня 2.5**, где, собственно, и выполняется коммутация по меткам.

- **Сетевой уровень** – это комплексный уровень, который обеспечивает возможность соединения и выбор маршрута между двумя конечными системами, подключенными к разным "подсетям", которые могут находиться в разных географических пунктах. В данном случае "подсеть" – это, по сути, независимый сетевой кабель (иногда называемый сегментом). Так как две конечные системы, желающие организовать связь, может разделять значительное географическое расстояние и множество подсетей, сетевой уровень является доменом маршрутизации. Протоколы маршрутизации выбирают оптимальные маршруты через последовательность соединенных между собой подсетей. Традиционные протоколы сетевого уровня передают информацию вдоль этих маршрутов.
- 
- **Канальный уровень** (формально называемый информационно-канальным уровнем) обеспечивает надежный транзит данных через физический канал. Выполняя эту задачу, канальный уровень решает вопросы физической адресации (в противоположность сетевой или логической адресации), топологии сети, линейной дисциплины (каким образом конечной системе использовать сетевой канал), уведомления о неисправностях, а также вопросы упорядоченной доставки блоков данных и управления потоком информации.
- 
- **Физический уровень определяет** электротехнические, механические, процедурные и функциональные характеристики активации, поддержания и дезактивации физического канала между конечными системами. Спецификации физического уровня определяют такие характеристики, как уровни напряжений, синхронизацию изменения напряжений, скорость передачи физической информации, максимальные расстояния передачи информации, физические соединители и другие аналогичные характеристики.

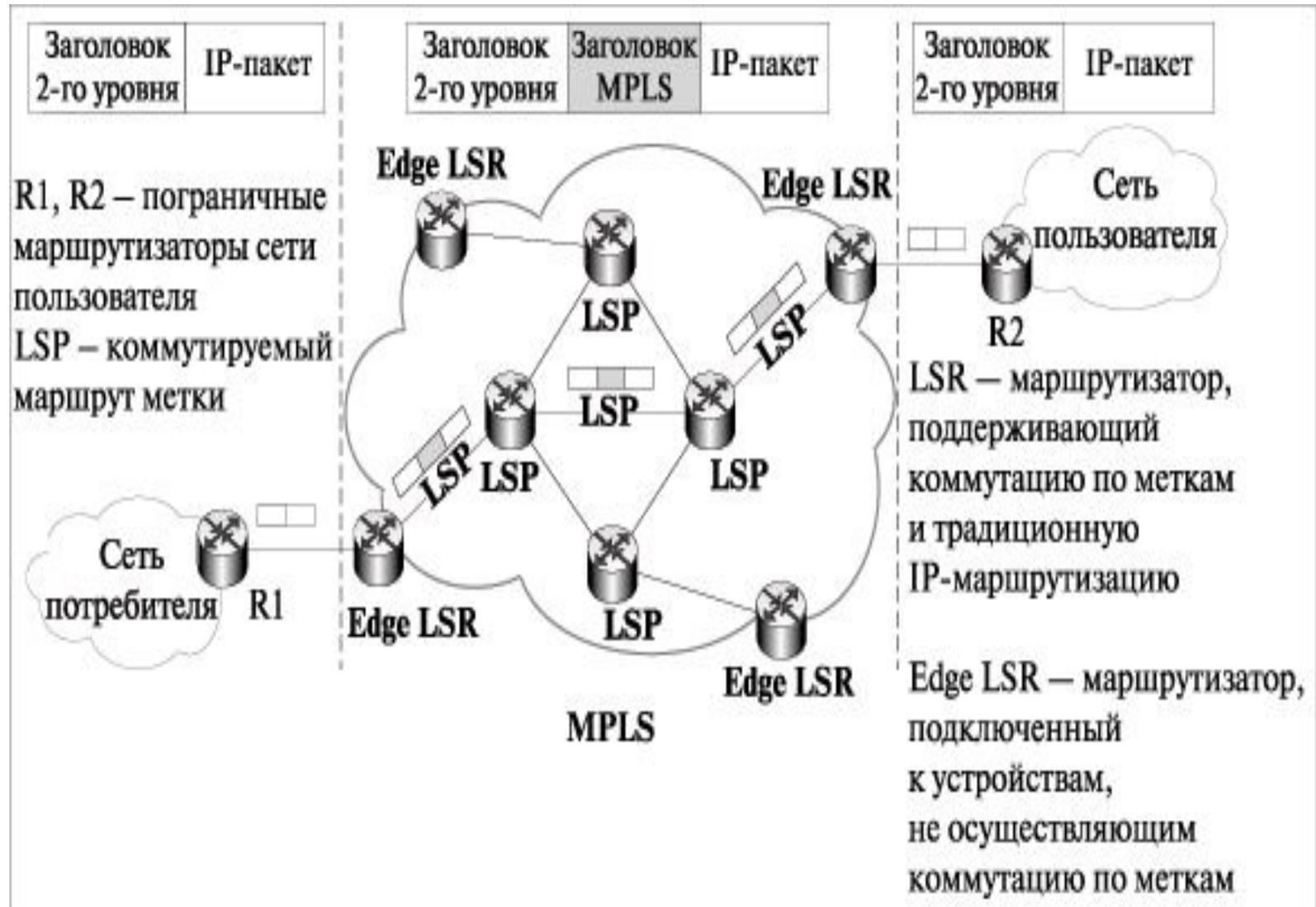
# Основные термины MPLS

- **FEC** – Forwarding Equivalence Class –класс эквивалентности пересылки Множество пакетов, которые пересылаются одинаково, например, с целью обеспечить заданное QoS
- **Label** – метка Короткий идентификатор фиксированной длины, определяющий принадлежность пакета тому или иному FEC
- **Label swapping** – замена меток Замена метки принятого узлом сети MPLS пакета новой меткой, связанной с тем же FEC, при пересылке этого пакета к нижестоящему узлу
- **LER** – MPLS edge router – пограничный узел сети MPLS. Пограничный узел сети MPLS, который соединяет домен MPLS с узлом, находящимся вне этого домена
- **Loop detection** - выявление закольцованных

# Основные термины MPLS

- **Loop prevention** – предотвращение образования закольцованных маршрутов. Метод, позволяющий обнаружить, что пакет прошел через узел более одного раза
- **LSP** – Label Switched Path – коммутируемый по меткам тракт. Приходящий через один или более LSR тракт, по которому следуют пакеты одного и того же FEC
- **ER-LSP** – explicitly routed LSP – LSP с явно заданным маршрутом. Тракт LSP, который организован способом, отличным от традиционной маршрутизации пакетов IP
- **LSR** – label switching router – маршрутизатор коммутации по меткам. Маршрутизатор, способный пересылать пакеты по технологии MPLS
- **MPLS domain** – домен MPLS. Совокупность узлов MPLS, между которыми существуют непрерывные LSP
- **MPLS egress node** – выходной узел сети MPLS. Последний MPLS-узел в LSP, направляющий исходный пакет к адресату, который находится вне MPLS-сети
- **MPLS ingress node** – Первый MPLS-узел в LSP, принимающий исходный пакет и помещающий в него метку MPLS

# 2 Принцип работы



**Этап 1.** Сеть автоматически формирует таблицы маршрутизации. В этом процессе участвуют маршрутизаторы или коммутаторы IP+ATM, установленные в сети сервис-провайдера. При этом применяются внутренние протоколы маршрутизации, такие как OSPF или IS-IS.

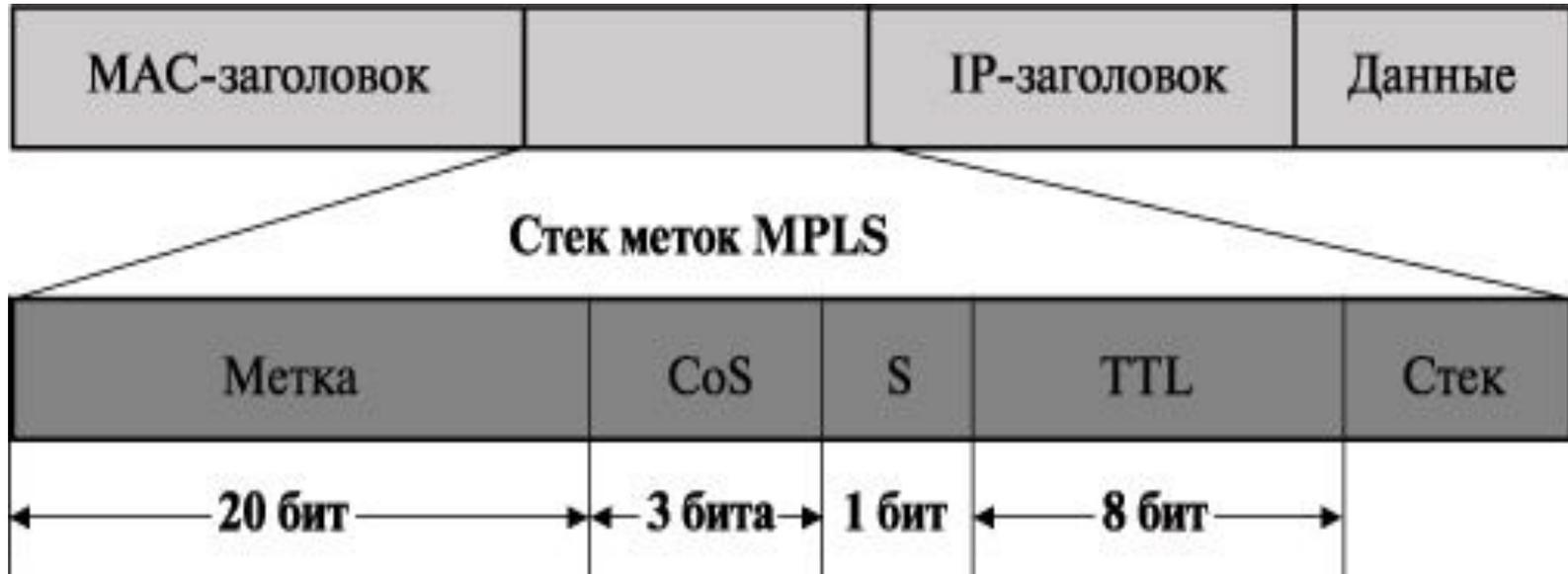
**Этап 2.** Протокол распределения меток (Label Distribution Protocol — LDP) использует отраженную в таблицах топологию маршрутизации для определения значений меток, указывающих на соседние устройства. В результате этой операции формируются маршруты с коммутацией по меткам (Label Switched Paths – LSP). Автоматическое присвоение меток MPLS выгодно отличает эту технологию от технологии частных виртуальных каналов ATM PVC, требующих ручного присвоения VCI/VPI.

**Этап 3.** Входящий пакет поступает на пограничный Label Switch Router (LSR), который определяет, какие услуги 3-го уровня необходимы этому пакету (например QoS или управление полосой пропускания). На основе учета всех требований маршрутизации и правил высокого уровня (policies), пограничный LSR выбирает и присваивает метку, которая записывается в заголовок пакета, после чего пакет передается дальше.

**Этап 4.** Устройство LSR, находящееся в опорной сети, считывает метки каждого пакета, заменяет старые метки новыми (новые метки определяются по локальной таблице) и передает пакет дальше. Эта операция повторяется в каждой точке передачи пакета по опорной сети.

**Этап 5.** На выходе пакет попадает в пограничный LSR, который удаляет метку, считывает заголовок пакета и передает его по месту назначения. В магистральных LSR метка MPLS сравнивается с заранее рассчитанными таблицами коммутации и содержит информацию 3-го уровня. Это позволяет каждому устройству LSR автоматически оказывать каждому пакету необходимые IP-услуги. Таблицы рассчитываются заранее, что снимает необходимость повторной обработки пакетов в каждой точке передачи. Такая схема не только позволяет разделить разные типы трафика (например, отделить неприоритетный трафик от критически важного); она делает решения MPLS хорошо масштабируемыми. Поскольку для присвоения меток технология MPLS использует разные наборы правил (policy mechanisms), она отделяет передачу пакетов от содержания заголовков IP. Метки имеют только локальное значение и многократно переиспользуются в крупных сетях, поэтому исчерпать запас меток практически невозможно. В рамках предоставления корпоративных IP-услуг самое главное преимущество MPLS заключается в способности присваивать метки, имеющие специальное значение. Наборы меток определяют не только место назначения, но и тип приложения и класс обслуживания.

# Метка и стек меток



Метка представляет собой последовательность записей. Каждая запись в стек **имеет длину 4 октета**.

Запись меток размещается после заголовка канального уровня, и перед заголовком сетевого уровня (например между Ethernet- и IP-заголовком). Верх стека записывается первым, а дно – последним. Сетевой заголовок следует сразу за записью стека меток с битом  $S=1$ . Каждая запись стека меток содержит в себе следующие поля.

## **Дно стека (S)**

Является средством поддержки иерархической структуры стека меток MPLS. В заголовке последней (т. е. самой глубокой или нижней) метки бит  $S=1$ , а во всех остальных метках в стеке бит  $S=0$ .

## **Время жизни (TTL)**

Это 8-битовое поле служит для представления значения времени жизни пакета. Данное поле является механизмом, предотвращающим возможность бесконечной циркуляции пакетов по сети вследствие образования закольцованных маршрутов. Байт TTL находится в конце заголовка метки.

## **Экспериментальное поле (CoS)**

Это 3-битовое поле зарезервировано для экспериментальных целей (QoS). В настоящее время проводится работа на создание согласованного стандарта использования этих битов для поддержки дифференцированного обслуживания разнотипного трафика и идентификации класса обслуживания. Первоначально это поле так и называлось – "Класс обслуживания" (CoS), и это название до сих пор широко распространено. При предоставлении дифференцированных услуг MPLS-сети это поле может указывать определенный класс обслуживания.

## **Значение метки**

Это 20-битовое поле несет в себе код метки. Может быть любым числом в диапазоне от 0 до 220- 1, за исключением резервных значений (0, 1, 2, 3 и др.), определением использования которых занимается рабочая группа MPLS в составе комитета IETF.

## **Анализ значения метки пакета:**

- следующий шаг, куда должен быть переадресован пакет;
- операция, которая должна быть выполнена со стеком меток до переадресации. Эта операция может быть заменой метки, удалением записи из стека, замещение верхней позиции в стеке и занесением туда затем одной или более новых записей.
- получить данные об инкапсуляции выходной информации и возможно другие данные, которые необходимы для того, чтобы корректно переадресовывать пакеты

# Значения зарезервированных меток.

- **Значение 0** представляет "IPv4 Explicit NULL Label". Это значение метки является единственно допустимым для дна стека меток. Оно указывает, что стек должен быть очищен и переадресация пакета должна основываться на IPv4-заголовке.
- **Значение 1** представляет "Router Alert Label". Это значение метки является легальным в любом месте стека меток, за исключением дна. Когда полученный пакет содержит такую метку на вершине стека, он доставляется локальному модулю для обработки. Действительная переадресация пакета определяется меткой в его стеке. Однако если пакет переадресовывается дальше, еще до переадресации в стек должна быть занесена метка "Router Alert". Использование этой метки сходно с применением опции "Router Alert" в IP-пакетах. Так как эта метка не может лежать на дне стека, она не ассоциируется с определенным протоколом сетевого уровня.
- **Значение 2** представляет "IPv6 Explicit NULL Label". Это значение метки является единственно допустимым для записи на дне стека. Оно указывает, что стек должен быть очищен, а переадресация пакетов должна после этого основываться на заголовке IPv6.
- **Значение 3** представляет "Implicit NULL Label". Это метка, которую LSR может присваивать и рассылать, но которая в действительности никогда не используется при инкапсуляции. Когда LSR замещает метку на верху стека на новую и эта новая метка является "Implicit NULL", LSR очистит стек, вместо того чтобы осуществить замену. Хотя это значение не может появиться при инкапсуляции, оно должно быть специфицировано в протоколе рассылки меток, так что значение может считаться зарезервированным.

• Значения 4-15 зарезервированы

**Стек меток MPLS - возможность иметь в пакете более одной метки в виде стека**

**MPLS может выполнить со стеком следующие операции:**

- помещать метку в стек;
- удалять метку из стека;
- и заменять метку.

**MPLS начинает работу с верхнего уровня стека, этот стек используется по принципу "последним пришел, первым ушел".**

# Четырехуровневый стек меток



- Заголовок MPLS № 1 был первым заголовком MPLS, помещенным в пакет, затем в него были помещены заголовки № 2, № 3 и, наконец, заголовок № 4. Коммутация по меткам всегда использует верхнюю метку стека, и метки удаляются из пакета так, как это определено выходным узлом для каждого LSP, по которому следует пакет. Бит S имеет значение 1 в нижней метке стека и 0 – во всех остальных метках. Это позволяет привязывать префикс к нескольким меткам, другими словами – к стеку меток (Label Stack). Каждая метка стека имеет собственные значения поля EXP, S-бита и поля TTL.

# Принцип инкапсуляции заголовка MPLS



- модель инкапсуляции подразумевает, что инкапсуляции MPLS может быть использована с любой технологией уровня 2. Метка MPLS может быть помещена в существующий формат заголовка уровня 2, как в случае ATM или FR, или вписана в специальный заголовок MPLS, как в случае Ethernet или PPP. Во всех случаях любые дополнительные метки находятся между верхней меткой стека и IP-заголовком уровня 3.
- Одной из самых сильных сторон технологии MPLS является то, что она может использоваться совместно с различными протоколами уровня 2. Среди этих протоколов – ATM, Frame Relay, PPP и Ethernet, FDDI и другие, предусмотренные документами по MPLS.

# Таблицы пересылки LIB (Label Information Base) –предназначены чтобы принять решение о дальнейшей обработке пакета.

Основные функции:

- операция, которую надо произвести со стеком меток пакета (заменить верхнюю метку стека, удалить верхнюю метку, поместить поверх стека новую метку);
- следующий маршрутизатор в LSP, причем "следующим" может быть тот же самый LSR;
- используемая при передаче пакета инкапсуляция на уровне звена данных;
- способ кодирования стека меток при передаче пакета;
- другая информация, относящаяся к пересылке пакета.

Входящая метка	Первая подзапись	Вторая подзапись
Значение входящей метки	Исходящая метка	Исходящая метка
	Выходной интерфейс	Выходной интерфейс
	Адрес следующего LSR	Адрес следующего LSR

Разные подзаписи внутри одной записи могут иметь либо одинаковые, либо разные значения исходящих меток.

Более одной подзаписи бывает нужно для поддержки многоадресной рассылки пакета, когда пакет, который поступил к одному входящему интерфейсу, должен затем рассылаться через несколько исходящих интерфейсов.

Запись в таблице может также содержать информацию, указывающую, какие ресурсы имеет возможность использовать пакет, например, определенную выходную очередь.

LSR может поддерживать либо одну общую таблицу, либо отдельные таблицы для каждого из своих интерфейсов. В первом варианте обработка пакета определяется исключительно меткой, переносимой в пакете. Во втором варианте обработка пакета определяется не только меткой, но и интерфейсом, к которому поступил пакет. LSR может использовать либо первый вариант, либо второй, либо их сочетание.

# Привязка "метка-FEC"

Каждая запись в таблице пересылки, которую ведет LSR, содержит одну входящую метку и одну или более исходящих меток. В соответствии с этими двумя типами меток обеспечивается **два типа привязки меток к FEC:**

- первый тип – **локальная привязка** (метка для привязки выбирается и назначается в LSR локально).
- второй тип – **удаленная привязка** (LSR получает от некоторого другого LSR информацию о привязке метки, которая соответствует привязке, созданной на этом другом LSR).

# Два варианта привязки меток к FEC

- **привязка метки к FEC "снизу"** (downstream label binding), потому что в этом случае привязка переносимой пакетом метки к тому FEC, которому принадлежит этот пакет, создается нижестоящим LSR, т.е. LSR, расположенным ближе к адресату пакета, чем LSR, который помещает метку в пакет. При привязке "снизу" пакеты, которые переносят определенную метку, передаются в направлении, противоположном направлению передачи информации о привязке этой метки к FEC.
- **метки к FEC "сверху"** (upstream label binding), потому что в этом случае привязка переносимой пакетом метки к тому FEC, которому принадлежит этот пакет, создается тем же LSR, который помещает метку в пакет; т.е. создатель привязки расположен "выше" (ближе к отправителю пакета), чем LSR, к которому пересылается этот пакет. При привязке "сверху" пакеты, которые переносят определенную метку, передаются в том же направлении, что и информация о привязке этой метки к FEC.

# Режимы операций с метками

## 1. Режим назначения меток:

- **независимое назначение** (т.е. независимое создание трактов LSP) - при таком назначении меток каждый LSR сам, независимо от других событий, принимает решение о привязке метки к обнаруженному FEC и об уведомлении вышестоящего LSR об этой привязке. Такая ситуация аналогична традиционной маршрутизации, выполняемой в обычных IP-сетях, когда обнаруживаются новые маршруты.
- **упорядоченное назначение** (т.е. упорядоченное создание трактов LSP) - этот способ назначения меток имеет больше ограничений, чем предыдущий, в том смысле, что привязка метки к определенному FEC происходит только тогда, когда LSR либо выступает в роли выходного узла для этого FEC, либо уже получил информацию о привязке "метка-FEC" от нижестоящего маршрутизатора.

## 2. Режимы сохранения меток.

- **либеральный режим**- вышестоящий LSR сохраняет у себя любую привязку к FEC меток, которые он получил от несмежных нижестоящих LSR (т.е. метки пришли к нему транзитом).
- **консервативный режим** - вышестоящий LSR отказывается от таких меток, т.е. отбрасывает их.

### 3. Распределение меток

- **нижестоящим LSR по запросу вышестоящего** - используется для создания трактов LSP по участкам. Он позволяет вышестоящему LSR в явном виде запрашивать привязку метки к определенному FEC у соседнего с ним нижестоящего LSR.
- **нижестоящим LSR по собственной инициативе** - когда нижестоящему LSR нужно "раздать" метки вышестоящим, хотя те не передавали запроса в явном виде.