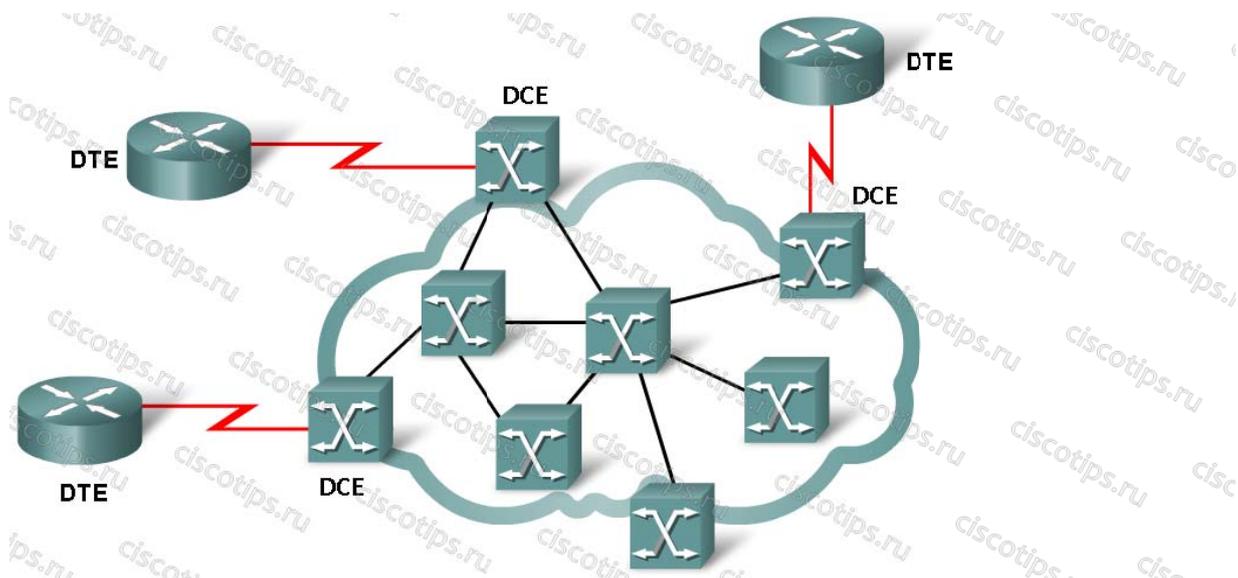


Frame Relay — пример настройки и описание

Frame Relay – WAN-протокол, работающий на втором уровне [модели OSI](#), то есть, там же, где работают Ethernet, PPP, HDLC и др. Frame Relay пришёл на смену протокола X.25, в России сравнительно широкого распространения не получил, а сейчас – и вовсе его время давно прошло. Знания этого протокола могут потребоваться, если вы работаете у провайдера, у которого по старой памяти остались какие-то абоненты, работающие по FR. Кроме того, знание Frame Relay помогает в понимании MPLS.

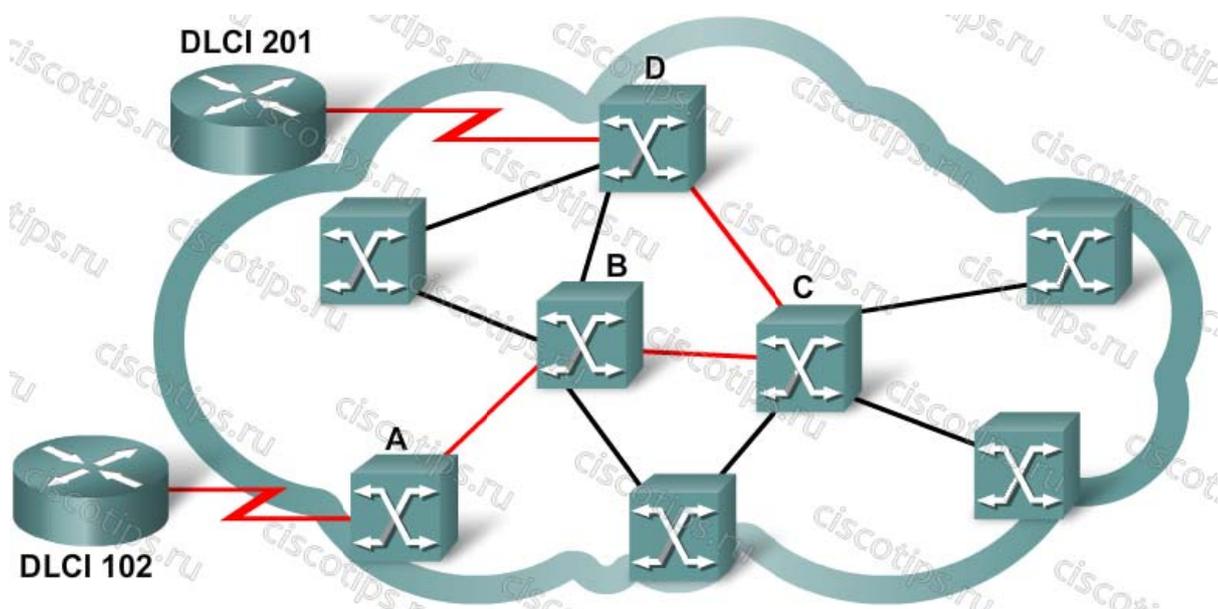
Принципы работы Frame Relay

Аналогично другим WAN протоколам мы, как клиент, настраиваем маршрутизатор как DTE. В качестве DCE выступает провайдерское оборудование, а именно, коммутатор FrameRelay switch. В качестве такого коммутатора может выступать обычный маршрутизатор cisco и пусть слово «коммутатор» вас не смущает.



Как используется Frame Relay? Мы арендуем у провайдера виртуальную сеть для соединения двух удалённых подразделений, он даёт нам кабель и говорит: настройте у себя на маршрутизаторе Frame Relay и добавляет: «Чтобы попасть из офиса А в офис Б используется DLCI 102, чтобы попасть из офиса Б в офис А используется DLCI 201»

DLCI – это «Data Link Connection Identifier», идентификатор соединения. У провайдера есть большая сеть, через которую проходит множество разных соединений (Virtual Circuit-ов), каждое направление по каждому из них имеет свой идентификатор – DLCI. Причём, DLCI имеет локальное значение, так что, если смотреть на приведённый рисунок, то первый же Frame Relay коммутатор «А», получив фрейм с DLCI 201 вполне может поменять его на другое значение, так как само число 201 имеет смысл только в контексте маленького участка сети между двумя соседними коммутаторами. Благодаря этому одни и те же номера DLCI можно использовать в разных частях сети, главное, чтобы все настройки на коммутаторах были согласованы между собой. DLCI Фактически это адрес канального уровня, то есть для Frame Relay DLCI – это способ идентификации общающихся устройств, как, например, MAC адрес для Ethernet. Только тут речь идёт скорее не об устройствах, а о каналах. В общем всё это для общего развития, а практически надо знать одно, с каждого из наших устройств, на границе облака, до каждого другого (до которого мы арендовали сеть от этого) ведёт один конкретный DLCI, который нам надо настроить.



Структура кадра Frame Relay

Кадр FR имеет следующий формат:



Назначение полей:

1. FLAG – признак начала фрейма, специальная последовательность нулей и единиц («01111110»), которая говорит получателю, что далее будет следовать тело фрейма.
2. ADDRESS – поле, содержащее DLCI получателя и отправителя, а так же некоторые флаги для управления потоком при возникновении заторов в сети.
3. DATA – поле с полезными данными вышестоящих протоколов.
4. FCS – контрольная сумма.
5. FLAG – признак окончания фрейма, содержит то же значение что и признак начала фрейма.

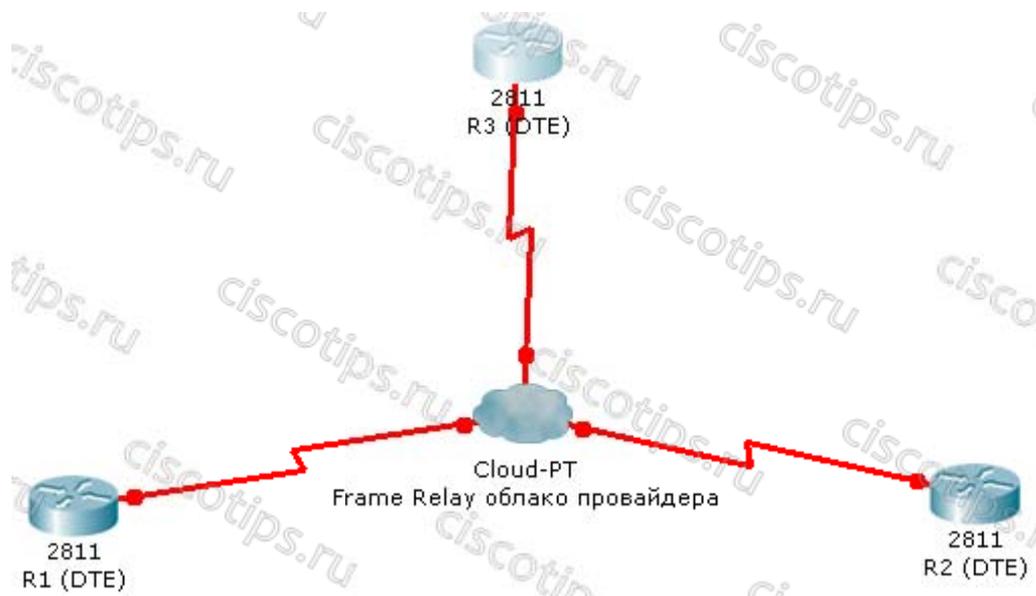
Для того, чтобы отправить что-то через сеть, надо, чтобы маршрутизатор знал, что его ждёт за тем или иным DLCI-ем, на том конце линии. Иными словами, необходима таблица соответствия DLCI-ев и IP адресов роутеров, находящимися за ними. Такая таблица называется frame relay map, она может строиться автоматически или настраиваться вручную администратором. С некоторой натяжкой можно провести аналогию между frame relay map и arp таблицей в Ethernet, так как в обоих случаях таблица нужна для установления соответствия между адресами протоколов второго и третьего уровней. Для автоматического создания таблицы есть протокол Inverse ARP – он позволяет, зная номер DLCI, узнать ip адрес маршрутизатора, находящегося на том конце соответствующего этому DLCI маршрута.

Frame Relay сам по себе занимается только доставкой данных. Однако, у него есть расширение **LMI** – Local management interface, которое позволяет, например, автоматически получать от провайдера список открытых для нас DLCI, через него же работает и Inverse ARP. Технически LMI помещает служебную информацию в поля ADDRESS и DATA обычного Frame Relay фрейма.

Настройка Frame Relay на маршрутизаторе

В CCNA рассматривается два способа настройки Frame Relay: с сабинтерфейсами и без. Второй способ проще, но через него не будет работать динамическая маршрутизация. Поэтому он может использоваться либо при соединениях точка-точка, либо, если вы знаете, что делаете.

Имеется топология, давайте попробуем настроить двумя способами.



Настройка без сабинтерфейсов

Пусть провайдер нам сообщил номера DLCI:

1. R1 – R2 – 102
2. R2 – R1 – 201
3. R1 – R3 – 103
4. R3 – R1 – 301
5. R2 – R3 – 203
6. R3 – R2 – 302

IP адреса маршрутизаторов:

1. R1 – 192.168.0.1
2. R2 – 192.168.0.2
3. R3 – 192.168.0.3

Настройка без сабинтерфейсов – все DLCI-и выходят наружу с нашего маршрутизатора через один общий serial интерфейс.

Настройка R1:

```
R1#
R1#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
R1(config)#interface serial0/0/0
R1(config-if)#no shutdown
%LINK-5-CHANGED: Interface Serial0/0/0, changed state to up
R1(config-if)#encapsulation frame-relay
%LINEPROTO-5-UPDOWN: Line protocol on Interface Serial0/0/0, changed state to up
R1(config-if)#ip address 192.168.0.1 255.255.255.0
R1(config-if)#bandwidth 64
R1(config-if)#frame-relay map ip 192.168.0.2 102 broadcast
R1(config-if)#frame-relay map ip 192.168.0.3 103 broadcast
```

Настройка достаточно простая: мы указали включили интерфейс, настроили инкапсуляцию и ip адрес, с помощью bandwidth сообщили маршрутизатору реальную скорость канала (чтобы потом,

например, метрика правильно считалась) и добавили две строчки в карту framerelay, сообщив, что 192.168.0.2 находится за DLCI-ем 102, а 192.168.0.3 – за 103. Слово broadcast означает в данном контексте вот что: Сам FR не поддерживает широковещательный трафик, но если мы задаём это слово, то при необходимости отправки брудкаста, он будет заменён на множество юникастовых фреймов – для всех получателей. Без этого слова не будут работать, например, протоколы динамической маршрутизации.

Настройка R2:

```
R2#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
R2(config)#interface serial 0/0/0
R2(config-if)#no shutdown
%LINK-5-CHANGED: Interface Serial0/0/0, changed state to up
R2(config-if)#encapsulation frame-relay
%LINEPROTO-5-UPDOWN: Line protocol on Interface Serial0/0/0, changed state to up
R2(config-if)#ip address 192.168.0.2 255.255.255.0
R2(config-if)#frame-relay map ip 192.168.0.1 201 broadcast
R2(config-if)#frame-relay map ip 192.168.0.3 203 broadcast
R2(config-if)#bandwidth 64
```

Аналогично настраиваем R3:

```
R3#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
R3(config)#interface serial 0/0/0
R3(config-if)#no shutdown
%LINK-5-CHANGED: Interface Serial0/0/0, changed state to up
R3(config-if)#encapsulation frame-relay
%LINEPROTO-5-UPDOWN: Line protocol on Interface Serial0/0/0, changed state to up
R3(config-if)#ip address 192.168.0.3 255.255.255.0
R3(config-if)#frame-relay map ip 192.168.0.1 301 broadcast
R3(config-if)#frame-relay map ip 192.168.0.2 302 broadcast
R3(config-if)#bandwidth 64
```

Настройка с сабинтерфейсами

Теперь попробуем реализовать то же самое, но с сабинтерфейсами. Отличие этого способа в том, что для каждого DLCI создаётся отдельный сабинтерфейс. Это нужно для работы протоколов динамической маршрутизации. Дело в том, что в динамической маршрутизации есть правило Split horizon, которое означает, что маршрутизатор не сообщает про некую сеть через тот же интерфейс, откуда он про неё узнал. То есть, если настраивать без сабинтерфейсов и R2 сообщит про какую-то сеть для R1, то R1 не сможет про неё рассказать дальше роутеру R3, так как он узнал про эту сеть с интерфейса s0/0/0, он не может через него же про неё сообщить. В случае использования сабинтерфейсов проблема решается, так как R1 узнал про эту сеть через интерфейс s0/0/0.102, а сообщил через s0/0/0.103. В этой топологии разные DLCI находятся на разных сабинтерфейсах, а значит надо поменять и ip адресацию чтобы не получилось, что у маршрутизатора два интерфейса в одной и той же сети.

Пусть сети будут такие:

1. R1 – R2 – 192.168.0.0/30
2. R1 – R3 – 192.168.0.4/30
3. R2 – R3 – 192.168.0.8/30

```
R1#
R1#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
R1(config)#interface serial0/0/0
R1(config-if)#encapsulation frame-relay
R1(config-if)#no shutdown
%LINK-5-CHANGED: Interface Serial0/0/0, changed state to up
R1(config-if)# interface serial0/0/0.102 point-to-point
%LINK-5-CHANGED: Interface Serial0/0/0.102, changed state to up
%LINEPROTO-5-UPDOWN: Line protocol on Interface Serial0/0/0.102, changed state to up
R1(config-subif)#frame-relay interface-dlci 102
R1(config-subif)#bandwidth 64
R1(config-subif)#ip address 192.168.0.1 255.255.255.252
R1(config-subif)#interface serial0/0/0.103 point-to-point
%LINK-5-CHANGED: Interface Serial0/0/0.103, changed state to up
%LINEPROTO-5-UPDOWN: Line protocol on Interface Serial0/0/0.103, changed state to up
R1(config-subif)#ip address 192.168.0.5 255.255.255.252
R1(config-subif)#frame-relay interface-dlci 103
R1(config-subif)#bandwidth 64
```

Настраиваем R2:

```
R2#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
R2(config)#interface serial0/0/0
R2(config-if)#encapsulation frame-relay
R2(config-if)#no shutdown

%LINK-5-CHANGED: Interface Serial0/0/0, changed state to up
%LINEPROTO-5-UPDOWN: Line protocol on Interface Serial0/0/0, changed state to up
R2(config-if)#interface serial0/0/0.201 point-to-point
%LINK-5-CHANGED: Interface Serial0/0/0.201, changed state to up
%LINEPROTO-5-UPDOWN: Line protocol on Interface Serial0/0/0.201, changed state to up
R2(config-subif)#frame-relay interface-dlci 201
R2(config-subif)#ip address 192.168.0.2 255.255.255.252
R2(config-subif)#interface serial0/0/0.203 point-to-point
%LINK-5-CHANGED: Interface Serial0/0/0.203, changed state to up
%LINEPROTO-5-UPDOWN: Line protocol on Interface Serial0/0/0.203, changed state to up
R2(config-subif)#frame-relay interface-dlci 203
R2(config-subif)#ip address 192.168.0.9 255.255.255.252
```

И R3:

```
R3#
R3#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
R3(config)#interface serial0/0/0
R3(config-if)#encapsulation frame-relay
R3(config-if)#no shutdown
%LINK-5-CHANGED: Interface Serial0/0/0, changed state to up
%LINEPROTO-5-UPDOWN: Line protocol on Interface Serial0/0/0, changed state to up
R3(config)#interface serial0/0/0.301 point-to-point
%LINK-5-CHANGED: Interface Serial0/0/0.301, changed state to up
%LINEPROTO-5-UPDOWN: Line protocol on Interface Serial0/0/0.301, changed state to up
R3(config-subif)#frame-relay interface-dlci 301
R3(config-subif)#ip address 192.168.0.6 255.255.255.252
R3(config-subif)#interface serial0/0/0.302 point-to-point
%LINK-5-CHANGED: Interface Serial0/0/0.302, changed state to up
%LINEPROTO-5-UPDOWN: Line protocol on Interface Serial0/0/0.302, changed state to up
R3(config-subif)#frame-relay interface-dlci 302
R3(config-subif)#ip address 192.168.0.10 255.255.255.252
```

Как и при создании сабинтерфейсов для работы с VLAN, сам номер сабинтерфейса не говорит ничего о номере DLCI для связывания с DLCI используется команда `frame-relay interface-dlci`. Но с точки зрения удобства и понятности, лучше делать номер сабинтерфейса совпадающим с номером DLCI.