

## Услуга (сервис) RESET уровня звена данных DL

### Исполнители:

**Главный архитектор** (принимает решения о поставке сервиса функциональными уровнями сети)

**Архитектор уровня** (задает функции уровня при поставке сервиса)

**Программист** (определяет принципы реализации сервиса)

**Инженер** (аппаратная поддержка)

### Главный архитектор:

Решение о поставке уровнем DL сервиса сброса (RESET) сетевому уровню.

Сервис состоит в сбросе текущего состояния уровня и его очистке от всех занятых в нем объектов без остановки работы по передаче данных (в отличие от DISCONNECT)

### Архитектор уровня:

Определяет инициаторов сервиса:

один из пользователи сервиса на сетевом уровне (user) или

поставщик сервиса (service provider) на уровне DL.

*Далее вариант с поставщиком (слайд 4.3).*

В поставке сервиса заняты три объекта: user A, user B, service provider;

поставщик сервиса на уровне DL (service provider) действует как единый объект

Принцип поставки сервиса показан на слайде 4.3 (две очереди и объекты в очередях, расположенные в зоне поставщика); в состав объектов в очередях входят все сервисные примитивы, поступающие от сетевого уровня:

поставщик сервиса сообщает о ситуации, требующей сброса (по его заключению), посылая примитив RESET-indication каждому из пользователей пользователям,

каждый пользователь отвечает согласием, открывая посылкой примитива RESET-response каждому пользователю.

Фактическая очистка очереди начинается после согласия на поставку сервиса сброса от пользователя с сетевого уровня: соответственно, примитив RESET- response от пользователя уничтожает все объекты, предшествующие ему в очереди.

Ввиду того, что между сигналом от поставщика о необходимости сброса и реакцией пользователя (посылкой от сетевого уровня примитива RESET-response) проходит определенное время, за которое в очередь могут поступить некоторые (возможно искаженные) объекты, необходим служебный объект, который делает очередь невидимой пользователю. Это синхронизационный маркер (synchronization marker).

Архитектор, программист и инженер решают, что сервис будут реализовать контроллеры, расположенные на границе DL-уровня в точках доступа к услугам SAPs. Таким образом, функции поставщика совместно реализуются контроллерами DLC на сторонах А и В.

В качестве протокола уровня DL выбирается HDLC.

### **Программист:**

Программист создает программное обеспечение контроллеров DLC.

Он определяет фактические команды и ответы из протокола HDLC, с использованием которых будут реализованы функции уровня DL в части сброса по инициативе поставщика:

очистка очереди после согласия от пользователя с сетевого уровня, переданного примитивом RESET-respose (r-r) -

### **команда RESET;**

сообщение в рамках поставщика на «тот конец» о ситуации сброса RESET-indication (r-i) и «скрытие очереди» от пользователя синхронизационным маркером (s) -

### **команда и ответ UNI (ненумерованная информация)**

Процедуры сброса, выполняемые контроллерами DLC на сторонах А и В состоят в следующем (четыре строчки на слайде 4.3):

1. Поставщик в целом должен действовать как единый объект, поэтому поставщик на стороне А, обнаружив условие сброса, сообщает об этом поставщику на стороне В, посылая логическую пару **s r-i** ( r-i впереди) в команде UNI.
2. Поставщик на стороне В отвечает, подтверждает готовность к совместному действию, посылая ответ UNI с логической парой **s r-i**.
3. Теперь обе стороны поставщика знают о ситуации, требующей сброса, поэтому поставщик на стороне В от имени «всего поставщика» сообщает о ней своему пользователю, посылая ему примитив RESET-indication (r-i). Одновременно, чтобы не допустить поступления «мусора», он закрывает свою очередь от своего пользователя (отрабатывает логическое условие **s**). Пользователь на стороне В посылает на уровень DL примитив RESET-response (**r-r**). Получив его, контроллер на стороне В очищает у себя очередь к А и посылает по каналу протокольную команду RESET с функцией **r-r** , которая должна очистить очередь от В на стороне А. После этого контроллер сможет снова передавать «чистые» протокольные единицы данных (незачерненные квадратики).
4. Получив протокольный ответ UNI со стороны В, поставщик на стороне А посылает примитив RESET-indication (r-i) к пользователю А. Очередь от В блокируется логикой **s**, оставшиеся в очереди объекты и блокировка **s** уничтожаются объектом **r-r**, поступившим от В в составе протокольной команды RESET.  
Получив RESET-indication, пользователь А отвечает примитивом RESET-response, по которому контроллер очищает очередь к В на стороне А и посылает по каналу протокольную команду RESET с функцией **r-r**, которая

должна вычистить очистить очередь и снять блокировку **s** на стороне В.  
Сразу за **r-r** можно посылать «чистые» протокольные единицы данных.

**Инженер:**

Проектирует контроллеры DLC, на которых можно запрограммировать и эффективно исполнить описанные операции.