

Понимание и устранение неполадок преобразования передающей среды SDLC – LLC

Содержание

[Введение](#)

[Предварительные условия](#)

[Требования](#)

[Используемые компоненты](#)

[Условные обозначения](#)

[SDLLC](#)

[Конфигурация SDLC](#)

[Конфигурация SDLLC](#)

[Отладка SDLLC](#)

[Преобразование носителей DLSw](#)

[команды "show"](#)

[Отладка пакетов SDLC во время DLSw/SDLC для PU2.1](#)

[Пример трансляции с носителя DLSw](#)

[Обратное преобразование информации выполнения DLSw](#)

[Преобразование информации с помощью локальной коммутации каналов передачи данных \(DLSw\)](#)

[Дополнительные сведения](#)

Введение

Этот документ предоставляет сведения, чтобы понять и устранить неполадки Протокола SDLC к преобразованию информации в сети Протокола LLC.

Предварительные условия

Требования

Для этого документа отсутствуют особые требования.

Используемые компоненты

Настоящий документ не имеет жесткой привязки к каким-либо конкретным версиям программного обеспечения и оборудования.

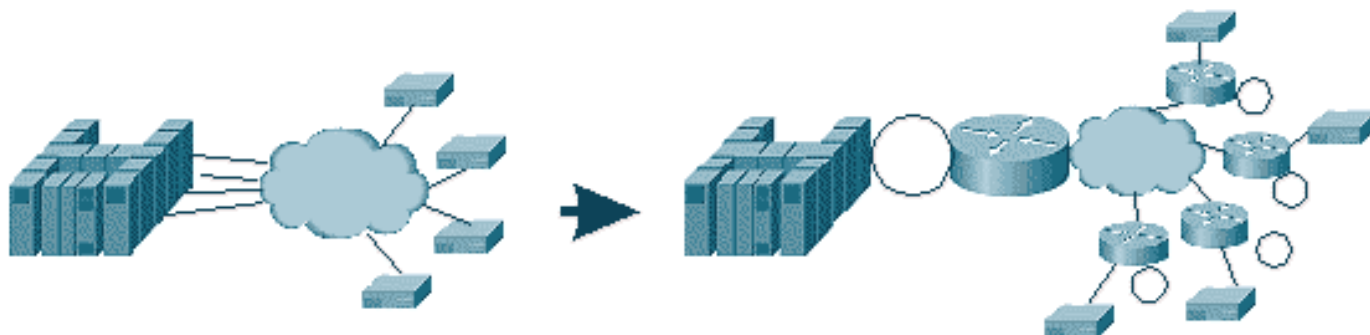
Условные обозначения

[Дополнительные сведения об условных обозначениях см. в документе Условные обозначения технических терминов Cisco.](#)

SDLLC

SDLC-to-LAN conversion (SDLLC) используется для преобразования сеанса SDLC для Physical Units 2 (PU2.0) устройство к Управлению логическим Каналом (LLC), сеансу типа 2 (LLC2). Если у вас есть большое количество удаленных контроллеров, питаемых в одиночный Порт Token Ring на препроцессоре (FEP), это очень полезно.

Левая часть этой схемы отображает FEP со многими линиями SDLC, уезжающими к удаленным местоположениям. Правая часть этой схемы отображает тот же сценарий с маршрутизаторами Cisco.



Маршрутизаторы позволяют FEP иметь только Интерфейс Token Ring. От той точки существуют несколько удаленных местоположения, выполняющие SDLLC к хосту, а также обычный трафик маршрутизации от источника (SRB).

Примечание: Использование SDLLC для LLC к преобразованию SDLC только просит устройства PU2.0, не для Типа физического устройства 2.1 (PU2.1). PU2.1 поддерживается в коммутации соединения передачи данных (DLSw).

Для настройки SDLLC вам нужен SRB в маршрутизаторе. См. [Понимание и Устранение проблем Local Source-Route Bridging](#) для получения информации о том, как настроить SRB.

Конфигурация SDLC

Поскольку SDLLC преобразовывает из интерфейса SDLC, вам сначала нужен SDLC, правильно настроенный. Выполните эти шаги для настройки SDLC:

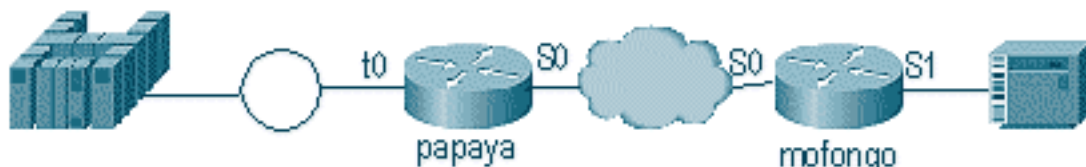
1. Выполните команду **encapsulation sdlc** для изменения серийной инкапсуляции на SDLC.
2. Выполните основную команду **sdlc - роли** для изменения роли маршрутизатора к основному в линии SDLC. **Примечание:** В Последовательном туннелировании (STUN) среды существует основной и дополнительные роли. См. [Настройку и Последовательное туннелирование Устранения проблем \(STUN\)](#) для получения дополнительной информации.
3. Выполните команду **sdlc address xx** для настройки опрашиваемого адреса SDLC.

Конфигурация SDLLC

Для настройки SDLLC первая выполненная команда является **traddr**. Эта команда определяет то, во что преобразовывает SDLC в среде LLC2. Выполните эти шаги для настройки SDLLC:

1. Выполните команду **sdllc traddr xxxx.xxxx.xx00 lr bn tr** для включения преобразования форматов разных сред передачи SDLLC на последовательном интерфейсе. Эта команда говорит маршрутизатору виртуальный MAC - адрес Станции SDLC. Затем команда задает номер местного кольца (**lr**), номер моста (**мллард**) и номер конечного вызова (**кнцерн**). **lr** должен быть уникальным в сети. **Мллард** может быть значением от 1 до 15. **trn** должен быть виртуальным кольцом в маршрутизаторе. При настройке локального SDLLC можно высказать это мнение к виртуальному кольцу или к интерфейсу (физический вызов, связанный с Интерфейсом Token Ring) в маршрутизаторе. **Примечание:** Последние две цифры MAC-адреса в этой команде **00**. Вы не можете установить последние две цифры **traddr**, потому что маршрутизатор использует эти цифры для вставки адреса SDLC этой линии. При определении последних двух цифр маршрутизатор заменяет их адресом SDLC. Затем хост не отвечает для того MAC-адреса. Например, если **traddr** MAC настроен как 4000.1234.5678, и адрес SDLC является 0x01, маршрутизатор использует MAC 4000.1234.5601 для представления устройства SDLC в домене LLC. Кроме того, **traddr** MAC находится в неканонический формате, который является тем же форматом как Фрейм Token Ring.
2. Выполните команду **sdllc xid address xxxxxxxx** для определения значения eXchange IDentification (XID), соответствующего Станции SDLC совпадать со значениями Virtual Telecommunications Access Method (VTAM). Это определено от IDBLK и IDNUM в главном узле коммутатора в VTAM. Если это не совпадает, сбои обмена XID.
3. Выполните команду **sdlc-address mac-address sdllc partner** для включения соединений для SDLLC. Это задает MAC-адрес партнера, который обычно является хостом.

Отображен простой пример конфигурации SDLLC. Подключенный к SDLC контроллер появляется как локальное подключенное устройство Token Ring к FEP.



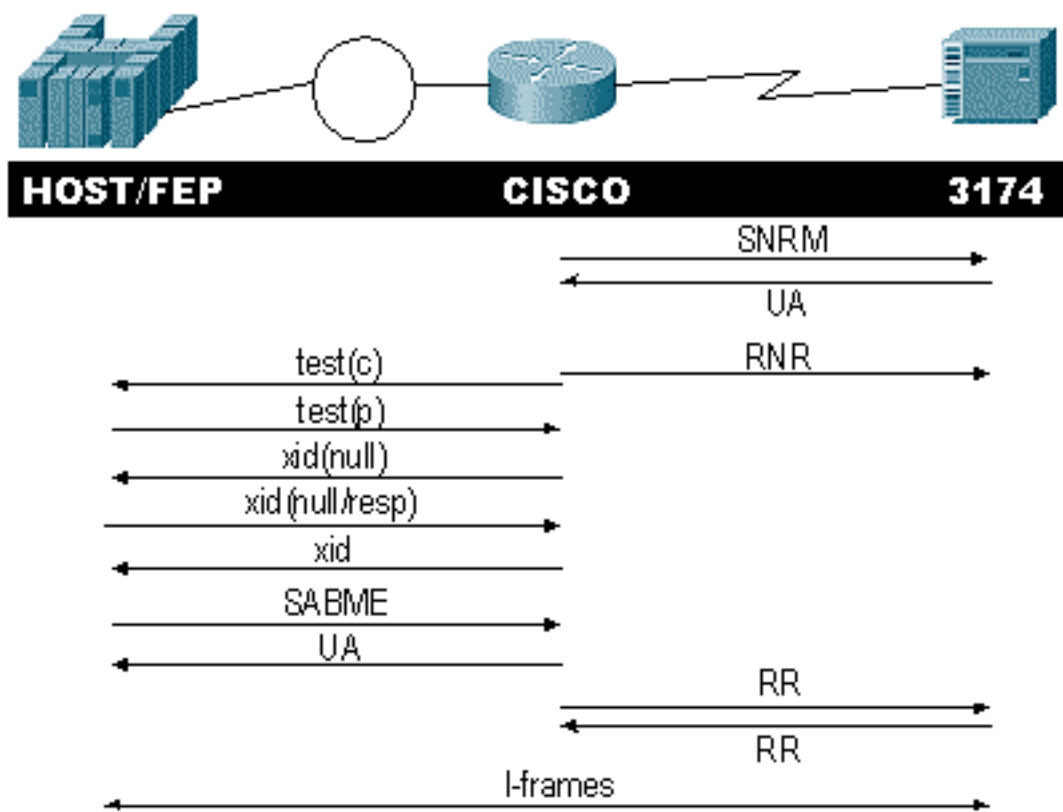
Папайя	Mofongo
<pre>source-bridge ring-group 100 source-bridge remote- peer 100 tcp 1.1.1.1 source-bridge remote- peer 100 tcp 1.1.2.1 local-ack interface tokenring 0 ip address 1.1.3.1 255.255.255.0 source-bridge 33 2 100 source-bridge spanning</pre>	<pre>source-bridge ring group 100 source-bridge remote-peer 100 tcp 1.1.2.1 source-bridge remote-peer 100 tcp 1.1.1.1 local-ack source-bridge sdllc local-ack interface serial 0 encapsulation sdllc-primary sdllc address c6 sdllc traddr 4000.3174.1100 333 3 100 sdllc partner 4000.1111.1111 c1 sdllc xid c1 17200c6</pre>

<pre>interface loopback 0 ip address 1.1.1.1 255.255.255.0</pre>	<pre>interface loopback 0 ip address 1.1.2.1 255.255.255.0</pre>
--	--

Отладка SDLLC

Проблема SDLLC требует, чтобы вы устранили неполадки двух других сред: слово SDLC и Управление логическим Каналом (LLC), мир типа 2 (LLC2) туда, где вы преобразовываете кадры. Поскольку у вас может только быть один тип контроллера, отлаживание SDLLC легче понять, чем коммутация соединения передачи данных (DLSw) / SDLC.

Во-первых, заметьте потоки для этого определенного запуска сеанса:



Проверьте для ответа Обычного режима ответа набора (SNRM) от контроллера. Маршрутизатор не запускает часть LLC, пока часть SDLC не в порядке.

Выполните эти команды для проверки ответа SNRM:

- `sdhc_state`
- `sdllc_state`

В данном примере SNRM передается контроллеру, который изменяет состояние линии к SNRMSSENT. Если маршрутизатор остается в этом состоянии, то это не получило нумерованное подтверждение (UA) от контроллера. Это может означать, что что-то неправильно с линией SDLC. Если это происходит, отладка отображена как:

```
%LINEPROTO-5-UPDOWN: Line protocol on Interface Serial1, changed state to up
s4f#
SDLLC_STATE: Serial1 C6 DISCONNECT
-> SDLC PRI WAIT
SDLC_STATE: (5234984) Serial1 C6 DISCONNECT -> SNRMSSENT %SYS-5-CONFIG_I: Configured from console
by console %LINK-3-UPDOWN: Interface Serial1, changed state to up Serial1 SDLC output C693
```

```
Serial1 SDLC input C673 SDLC_STATE: (5235700) Serial1 C6 SNRMSSENT -> CONNECT SDLLC_STATE:
Serial1 C6 SDLC PRI WAIT -> NET UP WAIT SDLC_STATE: (5235700) Serial1 C6 CONNECT -> USBUSY
```

Если маршрутизатор получает UA, шаги **sdlc_state** от SNRM_SENT до ПОДКЛЮЧЕНИЯ. Затем, SDLLC сообщает шаги от SDLC_PRI_WAIT до NET_UP_WAIT. Когда это происходит, маршрутизатор может начать поднимать сторону LLC соединения. Заключительное действие должно начать передавать Receive (Receive) Not Ready (RNR) к линии SDLC. Это отключает контроллер от передачи любой информации, пока сторона LLC не в рабочем состоянии.

Затем, маршрутизатор передает проводник для обнаружения местоположения его партнера.

```
SDLLC: O TEST, dst 4000.1111.1111 src 4000.3174.11c6 dsap 0 ssap 0 To0: out: MAC: acfc: 0x8040
Dst: 4000.1111.1111 Src: c000.3174.11c6 bf: 0x82 0x304A210 To0: out: RIF: 8800.14D3.0642.0210
To0: out: LLC: 0000F300 00800000 000C3BF0 7D000000 00800000 000C3BF0 ln: 25 SDLLC: NET UP WAIT
recv FORWARD TEST P/F(F3) 4000.3174.11c6 c000.1111.1111 00 01 -> Serial1 C6 caching rif
```

Предыдущие выходные данные отображают тестовый опрос, передаваемый и полученный. Поскольку данный пример имеет локально подключенный контроллер и Token Ring, тестовый опрос оставляет маршрутизатор, ищущий адрес партнера. После того, как маршрутизатор получает тестовый фрейм, он начинает обмен XID. Маршрутизатор кэширует Поле маршрутной информации (RIF) для этого сеанса, который можно проверить с командой **show rif**. Поскольку это - PU2.0, маршрутизатор передает XID Типа 2 Формата 0 к хосту после ответа на пустой указатель XID.

```
SDLLC: O xid(null), 4000.1111.1111 4000.3174.11c6 4 4 [1000.14D3.0641.0051.12C2.0194.01F1.02C0]
SDLLC: NET UP WAIT recv FORWARD XID P/F(BF) 4000.3174.11c6 c000.1111.1111 04 05 -> Serial1 C6
SDLLC: O xid(0T2), 4000.1111.1111 4000.3174.11c6 4 4 [1000.14D3.0641.0051.12C2.0194.01F1.02C0]
SDLLC: NET UP WAIT recv FORWARD SABME P/F(7F) 4000.3174.11c6 c000.1111.1111 04 04 -> Serial1 C6
SDLLC: SABME for Serial1 C6 in NET UP WAIT %SDLLC-5-ACT_LINK: SDLLC: Serial1 LINK address C6
ACTIVATED: Net connect SDLLC_STATE: Serial1 C6 NET UP WAIT -> CONNECT
```

После обмена XID маршрутизатор получает установку расширенного асинхронного сбалансированного режима (SABME) от хоста. Это завершает процедуру запуска, и маршрутизатор отвечает UA к хосту. Теперь, состояние изменений линии SDLC от USBUSY для СОЕДИНЕНИЙ, и I-кадры может пройти через маршрутизатор.

```
SDLC_STATE: (5235944) Serial1 C6 USBUSY
-> CONNECT
Serial1 SDLC output      C611
Serial1 SDLC input      C611
s4f#
```

Преобразование носителей DLSw

DLSw предоставляет главный enhancement преобразованию форматов разных сред передачи, потому что это поддерживает PU2.1. Это позволяет ему иметь SDLLC к преобразованию для контроллеров LLC2, такому как 5494 и 5394 (с параметром обновления к PU2.1 - RPQ IBM 8Q0775) к AS/400s. Это устраняет необходимость STUN и плохих многоточечных линий AS/400.

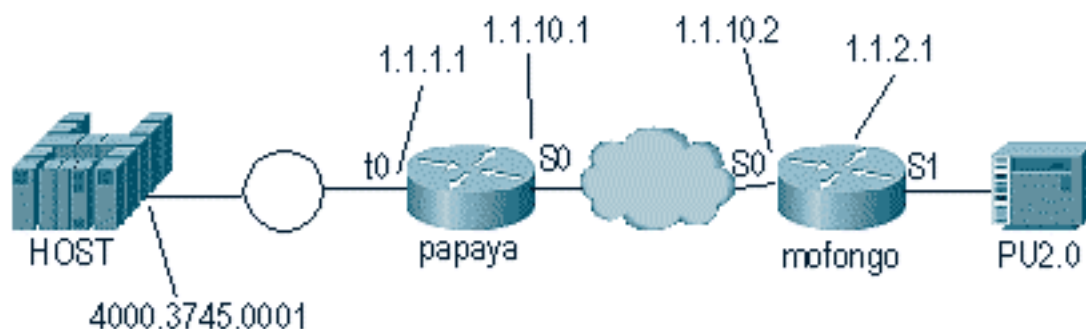
Параметры конфигурации для преобразования форматов разных сред передачи DLSw немного отличаются от параметров SDLLC. Существует одна команда DLSw, которая добавлена, остальные - команды SDLC. Выполните эти шаги для настройки преобразования форматов разных сред передачи DLSw:

1. Выполните команду **encapsulation sdhc** для изменения серийной инкапсуляции на

SDLC. Поскольку вы переходите оконечный линия SDLC в маршрутизаторе, маршрутизатор должен действовать как основной для целей опроса. Это отличается от STUN, потому что основной будет ХОСТОМ или AS/400.

2. Выполните **основную команду sdhc - роли** для изменения роли маршрутизатора к основному в линии SDLC.
3. Выполните **команду sdhc address xx** для настройки опрашиваемого адреса SDLC. Это - то, где DLSw отличается от SDLLC. В SDLLC вы задаете команды с **ключевым словом sdllc**. В DLSw задайте команды с **ключевым словом sdhc**.
4. Выполните **команду sdhc vmac xxxx.xxxx.xx00** для настройки виртуального MAC - адреса для Контроллера SDLC. Этот параметр говорит маршрутизатору виртуальный MAC - адрес для этого Контроллера SDLC в среде LLC2. Не забудьте оставлять последний набор байта **00**, потому что опрашиваемый адрес добавлен там (**sdhc address**).
5. Выполните **команду sdhc xid nn xxxxxxxx** для настройки XID для этого PU 2.0. В этой команде **nn** является опрашиваемым адресом контроллера, и **xxxxxxx** является XID для этого PU2.0 (IDBLOCK и IDNUM, который закодирован в главном узле коммутатора в VTAM). **Примечание:** Если у вас есть PU2.1, существует согласование XID. Таким образом, изменения команды.
6. Выполните **команду sdhc xid nn xid-poll** для настройки XID для этого PU 2.1. В этой команде **nn** является опрашиваемым адресом станции.
7. Выполните **команду sdhc partner xxxx.xxxx.xxxx nn** для настройки MAC-адреса партнера маршрутизатора. В этой команде **nn** является опрашиваемым адресом для рассматриваемого контроллера. Важно задать адрес контроллера, потому что в многоточечных линиях может быть контроллер, двигался к одному хосту, и другой контроллер двигался к другому хосту.
8. Выполните **команду sdhc dlsw nn** для настройки DLSw для определенного контроллера. В этой команде **nn** является опрашиваемым адресом контроллера или контроллеров в многоточечном. Эта команда позволяет вам задавать множественные опрашиваемые адреса в одной команде. **Примечание:** Остерегайтесь дефекта #CSCdi75481. См. [Bug Toolkit \(только зарегистрированные клиенты\)](#) для получения дополнительной информации. Если команда **sdhc dlsw nn** не удалена прежде, чем изменить адрес SDLC маршрутизатора, код CLS не может правильно передать DLSw с интерфейсом SDLC. Это заставляет интерфейс вести себя, как будто ничто не было настроено. Эта ошибка была исправлена в Версиях программного обеспечения Cisco IOS 11.1 (8.1) 11.1 (8.1) AA01 (01.03) 11.1 (8.1) AA01 (01.02) и позже.

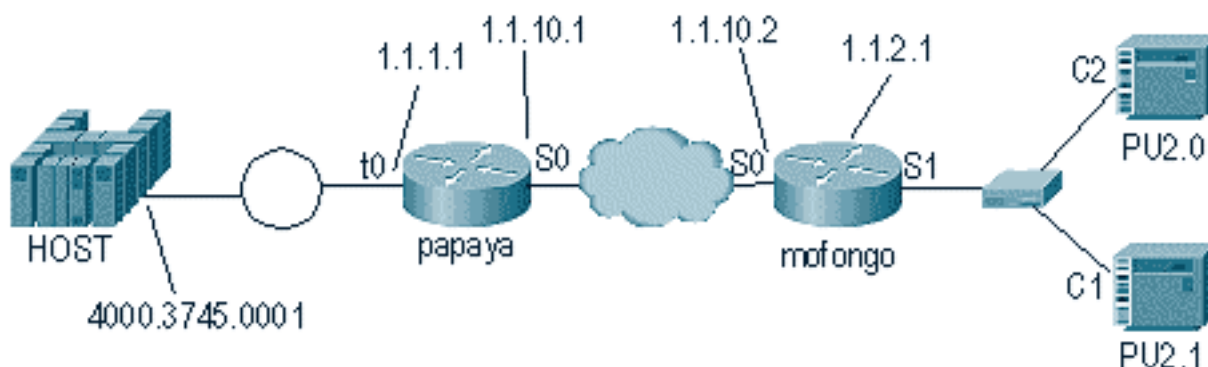
Пример конфигурации для контроллера PU2.0 SDLC DLSw отображен.



Папайя	Mofongo
source-bridge ring-group	dlsw local-peer peer-id 1.1.2.1

<pre> 100 dlsw local-peer peer-id 1.1.1.1 dlsw remote-peer 0 tcp 1.1.2.1 ! interface serial 0 ip address 1.1.10.1 255.255.255.0 ! interface tokenring 0 ip address 1.1.1.1 255.255.255.0 ring-speed 16 source-bridge 1 1 100 source-bridge spanning </pre>	<pre> dlsw remote-peer 0 tcp 1.1.1.1 ! interface loopback 0 ip address 1.1.2.1 ! interface serial 0 ip address 1.1.10.2 255.255.255.0 ! interface serial 1 no ip address encapsulation sdslc sdslc role primary sdslc vmac 4000.3174.0000 sdslc address c1 sdslc xid c1 01767890 sdslc partner 4000.3745.0001 c1 sdslc dlsw c1 </pre>
--	---

При кодировании многоточечного помните, что PU2.1 более интеллектуальны и имеют дополнительные сведения для обмена, чем обычное устройство PU2.0. Это важно при настройке многоточечной среды, потому что необходимо закодировать линию как основную для устройства PU2.0. Также необходимо добавить **xid-опрос** для адреса SDLC устройства PU2.1, таким образом, код понимает, что сделать с каждым из контроллеров. Это - пример конфигурации.



Папайя	Mofongo
<pre> source-bridge ring- group 100 dlsw local-peer peer-id 1.1.1.1 dlsw remote-peer 0 tcp 1.1.2.1 ! interface serial 0 ip address 1.1.10.1 255.255.255.0 ! interface tokenring 0 ip address 1.1.1.1 255.255.255.0 ring-speed 16 source-bridge 1 1 100 source-bridge spanning </pre>	<pre> dlsw local-peer peer-id 1.1.2.1 dlsw remote-peer 0 tcp 1.1.1.1 ! interface loopback 0 ip address 1.1.2.1 ! interface serial 0 ip address 1.1.10.2 255.255.255.0 ! interface serial 1 no ip address encapsulation sdslc sdslc role primary sdslc vmac 4000.3174.0000 sdslc address c1 xid-poll sdslc partner 4000.9404.0001 c1 sdslc address c2 01767890 sdslc partner 4000.9404.0001 c2 sdslc dlsw c1 c2 </pre>

команды "show"

См. [Data-Link Switching Plus](#) для получения дополнительной информации о командах показа, используемых для преобразования форматов разных сред передачи DLSw.

Отладка пакетов SDLC во время DLSw/SDLC для PU2.1

```
%LINK-3-UPDOWN: Interface Serial2, changed state to up
```

Первой вещью произойти является XID или BF к широковещательному адресу SDLC И следующие

```
Serial2 SDLC output      FFBF
```

Затем, XID получен от 5494. Это - XID format 2 type 3, который отображен в этих выходных данных команды `debug sdhc packet`:

```
Serial2 SDLC input
```

```
0046C930: DDBF3244 073000DD 0000B084 00000000 .....d.... 0046C940: 00000001 0B000004  
09000000 00070010 ..... 0046C950: 17001611 01130012 F5F4F9F4 F0F0F2F0  
.....54940020 0046C960: F0F0F0F0 F0F0F0F0 0E0CF4D5 C5E3C14B 00000000..4NETA. 0046C970:  
C3D7F5F4 F9F4 CP5494
```

Это пояснения нескольких из полей от этой команды:

- **073000DD** — Это поле является Блочным ID и ID Цифра, которая настроена в 5494. Блочный ID и ID Цифра упоминаются как XID и передаются 5494 узлу во время согласования в процессе сеанса.
- **NETA** — Это поле является Идентификатором сети (NETID) Усовершенствованной одноранговой сети (APPN), который используется. Обычно, это поле совпадает с NETID, который настроен в узле. В этом случае точкой вызова будет AS/400.
- **CP5494** — Это поле является названием Контрольной точки (CP) 5494.
- **DD** — Это поле является адресом SDLC.

Затем, XID получен от AS/400:

```
Serial2 SDLC output
```

```
004BC070:      FFBF 324C0564 52530000 000A0800 ...<..... 004BC080: 00000000 00010B30  
0005BA00 00000007 ..... 004BC090: 000E0DF4 D5C5E3C1 4BD9E3D7 F4F0F0C1  
..4NETA.RTP400A 004BC0A0: 1017F116 11011300 11F9F4F0 F4C6F2F5 ..1.....9404F25 004BC0B0:  
F1F0F0F0 F4F5F2F5 F3460505 80000000 100045253..... 004BC0C0: Serial2 SDLC input 0046C270:  
DDBF3244 073000DD ..... 0046C280: 0000B084 00000000 00000001 0B000004 ...d.....  
0046C290: 09000000 00070010 17001611 01130012 ..... 0046C2A0: F5F4F9F4 F0F0F2F0  
F0F0F0F0 F0F0F0F0 5494002000000000 0046C2B0: 0E0CF4D5 C5E3C14B C3D7F5F4 F9F4 ..4NETA.CP5494  
Serial2 SDLC output 004C0B10: FFBF 324C0564 52530000 00F6C800 ...<.....6H. 004C0B20: 00000080  
15010B10 0005BA00 00000007 ..... 004C0B30: 000E0DF4 D5C5E3C1 4BD9E3D7 F4F0F0C1  
..4NETA.RTP400A 004C0B40: 1017F116 11011300 11F9F4F0 F4C6F2F5 ..1.....9404F25 004C0B50:  
F1F0F0F0 F4F5F2F5 F3460505 80150000 100045253..... 004C0B60: Serial2 SDLC input 0046BBC0:  
DDBF3244 073000DD 0000B084 00000000 .....d.... 0046BBD0: 00000001 0B000004 09000000  
00070010 ..... 0046BBE0: 17001611 01130012 F5F4F9F4 F0F0F2F0 .....54940020  
0046BBF0: F0F0F0F0 F0F0F0F0 0E0CF4D5 C5E3C14B 00000000..4NETA. 0046BC00: C3D7F5F4 F9F4 CP5494
```

- **05645253** — Это поле является Блочной Цифрой Идентификатора и Идентификатора AS/400.
- **RTP400A** — Это поле является названием CP AS/400. Название CP найдено в Сетевых атрибутах Показа (DSPNETA) файлом на AS/400.

Затем SNRM (93) и UA (73) отображен на линии. До SNRM маршрутизатор всегда использует широковещательный адрес. С этого момента маршрутизатор всегда использует

фактический опрашиваемый адрес DD.

```
Serial2 SDLC output      DD93 Serial2 SDLC input DD73 Serial2 SDLC output DD11 Serial2 SDLC input DD11
```

На этом этапе соединение приостанавливает из-за устойчивого состояния Готового Reciever (RR) между маршрутизатором и 5494.

Примечание: Если маршрутизатор, на котором необходимо выполнить отладку, имеет другие интерфейсы SDLC, и вы не logging buffered, маршрутизатор может приостановить. Понимание, когда можно выполнить отладку к терминалу по сравнению с регистрацией, идет с опытом. Если вы не уверены, всегда используйте logging buffered и команду **show log** для отображения отладок SDLC

Варьируйте контроллер прочь на AS/400. Это позволяет вам видеть DISC (53) и UA (73) что результат на Стороне SDLC сеанса.

```
Serial2 SDLC output      DD53 Serial2 SDLC input DD73
```

Пример трансляции с носителя DLSw



После того, как интерфейс подходит и, маршрутизатор начинает процесс путем определения местоположения удаленного контроллера.

```
%LINEPROTO-5-UPDOWN: Line protocol on Interface Serial4, changed state to up
DLSw Received-ctlQ : CLSI Msg : ID_STN.Ind  dlen: 46
CSM: Received CLSI Msg : ID_STN.Ind  dlen: 46 from Serial4
CSM: smac 4000.5494.00dd, dmac 4000.9404.0001, ssap 4 , dsap 4
%DLSWC-3-RECVSSP: SSP OP = 4( ICR ) -explorer from peer 10.17.2.198(2065)
DLSw: new_ckt_from_clsi(): Serial4 4000.5494.00dd:4->4000.9404.0001:4
```

После приема кадра ICR DLSw запускает блок конечных состояний (FSM) для этого сеанса. Это выполнено **REQ_OPNSTN.Req** и сообщениями **REQ_OPNSTN.Cfm**, которые являются между DLSw и Интерфейсом сервисов соединения Cisco (CLSI).

```
DLSw: START-FSM (488636): event:DLC-Id state:DISCONNECTED
DLSw: core: dlsw_action_a()
DISP Sent : CLSI Msg : REQ_OPNSTN.Req dlen: 106 DLSw: END-FSM (488636): state:DISCONNECTED->LOCAL_RESOLVE
DLSw Received-ctlQ : CLSI Msg : REQ_OPNSTN.Cfm CLS_OK dlen: 106 DLSw: START-FSM (488636): event:DLC-ReqOpnStn.Cnf state:LOCAL_RESOLVE
DLSw: core: dlsw_action_b() CORE: Setting lf size to FF
```

После диалога с CLSI DLSw передает Фреймы **CUR** запуска сеанса к удаленному маршрутизатору. Они происходят между этими двумя маршрутизаторами только.

```
%DLSWC-3-SENDSSP: SSP OP = 3( CUR ) to peer 10.17.2.198(2065) success DLSw: END-FSM (488636): state:LOCAL_RESOLVE->CKT_START
%DLSWC-3-RECVSSP: SSP OP = 4( ICR ) from peer 10.17.2.198(2065)
DLSw: 488636 recv FCI 0 - s:0 so:0 r:0 ro:0 DLSw: recv RWO DLSw: START-FSM (488636): event:WAN-ICR state:CKT_START
DLSw: core: dlsw_action_e() DLSw: sent RWO DLSw: 488636 sent FCI 80 on ACK - s:20 so:1 r:20 ro:1
%DLSWC-3-SENDSSP: SSP OP = 5( ACK ) to peer 10.17.2.198(2065) success DLSw: END-FSM (488636): state:CKT_START->CKT_ESTABLISHED
```

Как только канал установлен, маршрутизатор передает XID, который был сохранен и запускает обмен XID. Важно понять, где XID. В данном примере управлении соединения передачи данных (DLC) - Идентификатор означает, что XID прибыл из локальной станции DLC, и WAN-XID прибыл из удаленного маршрутизатора или удаленной станции.

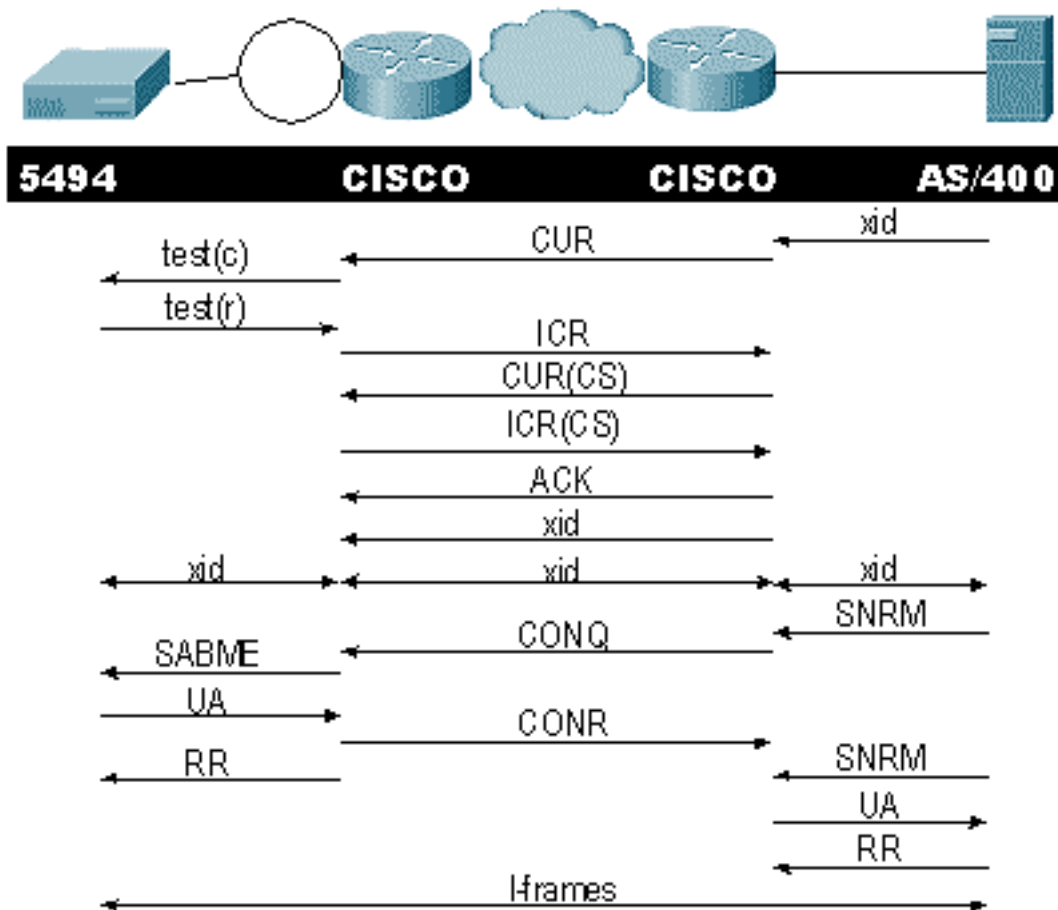
```
DLsw: START-FSM (488636): event:DLC-Id state:CKT_ESTABLISHED DLsw: core: dls_action_f() DLsw: 488636 sent FCA on XID %DLSWC-3-SENDSSP: SSP OP = 7( XID ) to peer 10.17.2.198(2065) success DLsw: END-FSM (488636): state:CKT_ESTABLISHED->CKT_ESTABLISHED %DLSWC-3-RCVSSP: SSP OP = 7( XID ) from peer 10.17.2.198(2065) DLsw: 488636 recv FCA on XID - s:20 so:0 r:20 ro:0 DLsw: START-FSM (488636): event:WAN-XID state:CKT_ESTABLISHED DLsw: core: dls_action_g() DISP Sent : CLSI Msg : ID.Rsp dlen: 12 DLsw: END-FSM (488636): state:CKT_ESTABLISHED->CKT_ESTABLISHED %DLSWC-3-RCVSSP: SSP OP = 7( XID ) from peer 10.17.2.198(2065) DLsw: START-FSM (488636): event:WAN-XID state:CKT_ESTABLISHED DLsw: core: dls_action_g() DISP Sent : CLSI Msg : ID.Req dlen: 88 DLsw: END-FSM (488636): state:CKT_ESTABLISHED->CKT_ESTABLISHED DLSW Received-ctlQ : CLSI Msg : ID.Ind dlen: 82 DLsw: START-FSM (488636): event:DLC-Id state:CKT_ESTABLISHED DLsw: core: dls_action_f() %DLSWC-3-SENDSSP: SSP OP = 7( XID ) to peer 10.17.2.198(2065) success DLsw: END-FSM (488636): state:CKT_ESTABLISHED->CKT_ESTABLISHED %DLSWC-3-RCVSSP: SSP OP = 7( XID ) from peer 10.17.2.198(2065) DLsw: START-FSM (488636): event:WAN-XID state:CKT_ESTABLISHED DLsw: core: dls_action_g() DISP Sent : CLSI Msg : ID.Rsp dlen: 88 DLsw: END-FSM (488636): state:CKT_ESTABLISHED->CKT_ESTABLISHED DLSW Received-ctlQ : CLSI Msg : ID.Ind dlen: 82 DLsw: START-FSM (488636): event:DLC-Id state:CKT_ESTABLISHED DLsw: core: dls_action_f() %DLSWC-3-SENDSSP: SSP OP = 7( XID ) to peer 10.17.2.198(2065) success DLsw: END-FSM (488636): state:CKT_ESTABLISHED->CKT_ESTABLISHED %DLSWC-3-RCVSSP: SSP OP = 7( XID ) from peer 10.17.2.198(2065) DLsw: START-FSM (488636): event:WAN-XID state:CKT_ESTABLISHED DLsw: core: dls_action_g() DISP Sent : CLSI Msg : ID.Rsp dlen: 88 DLsw: END-FSM (488636): state:CKT_ESTABLISHED->CKT_ESTABLISHED DLSW Received-ctlQ : CLSI Msg : ID.Ind dlen: 82 DLsw: START-FSM (488636): event:DLC-Id state:CKT_ESTABLISHED DLsw: core: dls_action_f() %DLSWC-3-SENDSSP: SSP OP = 7( XID ) to peer 10.17.2.198(2065) success DLsw: END-FSM (488636): state:CKT_ESTABLISHED->CKT_ESTABLISHED
```

Маршрутизатор получает CONQ от AS/400 (SABME). Это преобразовано в последовательную линию как SNRM. Затем маршрутизатор ждет UA на последовательной линии (CONNECT.Cfm) и передает CONR к другой стороне. Это изменяет состояние сеанса на СВЯЗАННЫЙ.

```
%DLSWC-3-RCVSSP: SSP OP = 8( CONQ ) from peer 10.17.2.198(2065) DLsw: START-FSM (488636): event:WAN-CONQ state:CKT_ESTABLISHED DLsw: core: dls_action_i() DISP Sent : CLSI Msg : CONNECT.Req dlen: 16 DLsw: END-FSM (488636): state:CKT_ESTABLISHED->CONTACT_PENDING DLSW Received-ctlQ : CLSI Msg : CONNECT.Cfm CLS_OK dlen: 8 DLsw: START-FSM (488636): event:DLC-Connect.Cnf state:CONTACT_PENDING DLsw: core: dls_action_j() %DLSWC-3-SENDSSP: SSP OP = 9( CONR ) to peer 10.17.2.198(2065) success DISP Sent : CLSI Msg : FLOW.Req dlen: 0 DLsw: END-FSM (488636): state:CONTACT_PENDING->CONNECTED
```

[Обратное преобразование информации выполнения DLsw](#)

Другая общая настройка является обратной-sdlc. В обратном SDLLC основная станция подключена через линию SDLC к маршрутизатору. Когда пользователи хотят переместить хост прикрепления Token Ring, это обычно замечается в серверных средах. Обратите изменения SDLLC путем, DLsw обрабатывает линию SDLC, потому что часто не ясно, если удаленный PU активен или нет.



Во-первых, потому что AS/400 является основным в этом случае, или набор, чтобы быть договорным в роли, это должно начать сеанс. Когда AS/400 передает первый XID после того, как последовательная линия становится в рабочем состоянии, маршрутизатор запускает процесс поиска для удаленного контроллера. После того, как канал установлен, согласование XID может начаться в линии.

Когда согласование XID заканчивается, AS/400 передает SNRM маршрутизатор. Это заставляет маршрутизатор передавать CONQ и ожидать CONR от удаленного маршрутизатора. Маршрутизатор не может ответить UA, пока это не видит SNRM, и после того, как это получает CONR. В почти всех версиях кода маршрутизатор ждет 30 секунд, пока это не вызывает таймаут сеанса. Это в отношении получения SNRM от основного устройства, как только основное устройство получает CONR от удаленного хоста.

В последнем коде Cisco IOS 11.1 настройки по умолчанию изменились на одну минуту вместо 30 секунд. В AS/400 этот таймаут называют таймером непродуктивного ответа pop и настройками по умолчанию к 32 секундам.

[Преобразование информации с помощью локальной коммутации каналов передачи данных \(DLSw\)](#)



```

%LINEPROTO-5-UPDOWN: Line protocol on Interface Serial2, changed state to up
%SYS-5-CONFIG_I: Configured from console by console
DLSW Received-ctlQ : CLSI Msg : ID_STN.Ind dlen: 46
CSM: Received CLSI Msg : ID_STN.Ind dlen: 46 from Serial2
  
```

Первой вещью, которую вы замечаете в локальном DLSw, является XID с последовательной стороны. Этот XID должен быть сохранен, пока маршрутизатор не передает тестовые кадры/ответы LLC через.

```

CSM: smac 4000.5494.00dd, dmac 4000.9404.0001, ssap 4 , dsap 4
DISP Sent : CLSI Msg : TEST_STN.Req dlen: 46 DISP Sent : CLSI Msg : TEST_STN.Req dlen: 46 DISP
Sent : CLSI Msg : TEST_STN.Req dlen: 46 CSM: Write to all peers not ok - PEER_NO_CONNECTIONS
DLSW Received-ctlQ : CLSI Msg : TEST_STN.Ind dlen: 43 CSM: Received CLSI Msg : TEST_STN.Ind
dlen: 43 from TokenRing0 CSM: smac c000.9404.0001, dmac 4000.5494.00dd, ssap 0 , dsap 4
  
```

Затем, тестовая станция оставляет маршрутизатор, и ответ возвращается из AS/400. Теперь, маршрутизатор может создать локальный FSM.

Примечание: Помните, что это - локальный сеанс.

```

DLSw: csm_to_local(): Serial2-->TokenRing0 4000.5494.00dd:4->4000.9404.0001:4
DLSw: START-LFSM TokenRing0 (4000.9404.0001->4000.5494.00dd) event:ADMIN-START
DLSw: LFSM-A: Opening DLC station
DISP Sent : CLSI Msg : REQ_OPNSTN.Req dlen: 106 DLSw: END-LFSM (4000.9404.0001->4000.5494.00dd):
state:DISCONNECTED ->OPN_STN_PEND DLSw: START-LFSM Serial2 (4000.5494.00dd->4000.9404.0001)
event:ADMIN-START DLSw: LFSM-A: Opening DLC station DISP Sent : CLSI Msg : REQ_OPNSTN.Req dlen:
106 DLSw: END-LFSM (4000.5494.00dd->4000.9404.0001): state:DISCONNECTED ->OPN_STN_PEND DLSW
Received-ctlQ : CLSI Msg : REQ_OPNSTN.Cfm CLS_OK dlen: 106 DLSw: START-LFSM TokenRing0
(4000.9404.0001->4000.5494.00dd) event:DLC-ReqOpnStn.Cnf DLSw: LFSM-B: DLC station opened DLSw:
END-LFSM (4000.9404.0001->4000.5494.00dd): state:OPN_STN_PEND ->ESTABLISHED DLSW Received-ctlQ :
CLSI Msg : REQ_OPNSTN.Cfm CLS_OK dlen: 106 DLSw: START-LFSM Serial2 (4000.5494.00dd->
4000.9404.0001) event:DLC-ReqOpnStn.Cnf DLSw: LFSM-B: DLC station opened DLSw: processing saved
clsi message
  
```

После того, как маршрутизатор подтвердил локально, что FSM готов, это может передать XID к партнеру. В данном примере партнер является AS/400 (ID.Req).

```

DLSw: START-LFSM Serial2 (4000.5494.00dd->4000.9404.0001) event:DLC-Id
DLSw: LFSM-X: forward XID to partner
DISP Sent : CLSI Msg : ID.Req dlen: 12 DLSw: END-LFSM (4000.5494.00dd->4000.9404.0001):
state:ESTABLISHED ->ESTABLISHED DLSw: END-LFSM (4000.5494.00dd->4000.9404.0001):
state:OPN_STN_PEND ->ESTABLISHED DLSW Received-ctlQ : CLSI Msg : ID.Cfm CLS_OK dlen: 32 DLSw:
START-LFSM TokenRing0 (4000.9404.0001->4000.5494.00dd) event:DLC-Id DLSw: LFSM-X: forward XID to
partner DISP Sent : CLSI Msg : ID.Rsp dlen: 12 DLSw: END-LFSM (4000.9404.0001->4000.5494.00dd):
state:ESTABLISHED ->ESTABLISHED
  
```

Затем XID получен от Token Ring. ID.Ind имеет длину 108. Маршрутизатор вперед этот XID к партнеру в этом сценарии, который является линией SDLC. Это обозначено ID.Req, который передавался. Каждый раз, когда маршрутизатор получает пакет, он должен запустить линейный блок конечных состояний (LFSM). Это - ключ к пониманию этой отладки, потому что это сообщает вам, где это запускается и который указывает, это идет.

```

DLSW Received-ctlQ : CLSI Msg : ID.Ind dlen: 108 DLSw: START-LFSM TokenRing0 (4000.9404.0001->
4000.5494.00dd) event:DLC-Id DLSw: LFSM-X: forward XID to partner DISP Sent : CLSI Msg : ID.Req
  
```

dlen: 88 DLsw: END-LFSM (4000.9404.0001->4000.5494.00dd): state:ESTABLISHED ->ESTABLISHED

Затем, ответ XID получен от последовательной линии и передан партнеру (Станция Token Ring в данном примере). Это продолжается, пока обмен XID не закончен для этого устройства PU2.1.

```
DLSW Received-ctlQ : CLSI Msg : ID.Ind dlen: 82 DLsw: START-LFSM Serial2 (4000.5494.00dd->4000.9404.0001) event:DLC-Id DLsw: LFSM-X: forward XID to partner DISP Sent : CLSI Msg : ID.Rsp
dlen: 80 DLsw: END-LFSM (4000.5494.00dd->4000.9404.0001): state:ESTABLISHED ->ESTABLISHED DLSW
Received-ctlQ : CLSI Msg : ID.Ind dlen: 108 DLsw: START-LFSM TokenRing0 (4000.9404.0001->4000.5494.00dd) event:DLC-Id DLsw: LFSM-X: forward XID to partner DISP Sent : CLSI Msg : ID.Rsp
dlen: 88 DLsw: END-LFSM (4000.9404.0001->4000.5494.00dd): state:ESTABLISHED ->ESTABLISHED DLSW
Received-ctlQ : CLSI Msg : ID.Ind dlen: 82 DLsw: START-LFSM Serial2 (4000.5494.00dd->4000.9404.0001) event:DLC-Id DLsw: LFSM-X: forward XID to partner DISP Sent : CLSI Msg : ID.Rsp
dlen: 80 DLsw: END-LFSM (4000.5494.00dd->4000.9404.0001): state:ESTABLISHED ->ESTABLISHED DLSW
Received-ctlQ : CLSI Msg : ID.Ind dlen: 108 DLsw: START-LFSM TokenRing0 (4000.9404.0001->4000.5494.00dd) event:DLC-Id DLsw: LFSM-X: forward XID to partner DISP Sent : CLSI Msg : ID.Rsp
dlen: 88 DLsw: END-LFSM (4000.9404.0001->4000.5494.00dd): state:ESTABLISHED ->ESTABLISHED %LINK-3-UPDOWN: Interface Serial2, changed state to up DLSW Received-ctlQ : CLSI Msg : ID.Ind dlen: 82
DLsw: START-LFSM Serial2 (4000.5494.00dd->4000.9404.0001) event:DLC-Id DLsw: LFSM-X: forward XID to partner DISP Sent : CLSI Msg : ID.Rsp dlen: 80 DLsw: END-LFSM (4000.5494.00dd->4000.9404.0001): state:ESTABLISHED ->ESTABLISHED
```

После обмена XID маршрутизатор получает SABME от AS/400 через **CONNECT.Ind**.
Маршрутизатор получает инструкцию отправить **CONNECT.Req** на линию SDLC, что является **SNRM**. Затем **CONNECT.Cfm (UA)**, сообщение получено от последовательной линии, которая заставляет код DLsw передавать **CONNECT.Rsp (UA)** к AS/400.

```
DLSW Received-ctlQ : CLSI Msg : CONNECT.Ind dlen: 8 DLsw: START-LFSM TokenRing0 (4000.9404.0001->4000.5494.00dd) event:DLC-Connect.Ind DLsw: LFSM-C: starting local partner DLsw: START-LFSM
Serial2 (4000.5494.00dd->4000.9404.0001) event:ADMIN-CONN DLsw: LFSM-D: sending connect request
to station DISP Sent : CLSI Msg : CONNECT.Req dlen: 16 DLsw: END-LFSM (4000.5494.00dd->4000.9404.0001): state:ESTABLISHED ->CONN_OUT_PEND DLsw: END-LFSM (4000.9404.0001->4000.5494.00dd): state:ESTABLISHED ->CONN_IN_PEND DLSW Received-ctlQ : CLSI Msg : CONNECT.Cfm
CLS_OK dlen: 8 DLsw: START-LFSM Serial2 (4000.5494.00dd->4000.9404.0001) event:DLC-Connect.Cnf
DLsw: LFSM-E: station accepted the connection DLsw: START-LFSM TokenRing0 (4000.9404.0001->4000.5494.00dd) event:ADMIN-CONN DLsw: LFSM-F: accept incoming connection DISP Sent : CLSI Msg
: CONNECT.Rsp dlen: 20 DLsw: END-LFSM (4000.9404.0001->4000.5494.00dd): state:CONN_IN_PEND ->CONNECTED DISP Sent : CLSI Msg : FLOW.Req dlen: 0 DLsw: END-LFSM (4000.5494.00dd->4000.9404.0001): state:CONN_OUT_PEND->CONNECTED
```

Сеанс, когда контроллер (SDLC) завершает работу, отображен.

```
%LINEPROTO-5-UPDOWN: Line protocol on Interface Serial2, changed state to down
%LINK-5-CHANGED: Interface Serial2, changed state to administratively down
DLSW Received-ctlQ : CLSI Msg : DISCONNECT.Ind dlen: 8 DLsw: START-LFSM Serial2 (4000.5494.00dd->4000.9404.0001) event:DLC-Disc.Ind DLsw: LFSM-Q: acknowledge disconnect DISP Sent : CLSI Msg :
DISCONNECT.Rsp dlen: 4
```

Затем, маршрутизатор передает DISC к AS/400 (**DISCONNECT.Rsp**). Затем это начинает разъединять локальную цепь.

```
DLsw: START-LFSM TokenRing0 (4000.9404.0001->4000.5494.00dd) event:ADMIN-STOP
DLsw: LFSM-Z: close dlc station request
DISP Sent : CLSI Msg : CLOSE_STN.Req dlen: 4
DLsw: END-LFSM (4000.9404.0001->4000.5494.00dd): state:ESTABLISHED ->CLOSE_PEND
```

```
DISP Sent : CLSI Msg : CLOSE_STN.Req dlen: 4
DLsw: END-LFSM (4000.5494.00dd->4000.9404.0001): state:ESTABLISHED ->CLOSE_PEND
```

```
DLSW Received-ctlQ : CLSI Msg : CLOSE_STN.Cfm CLS_OK dlen: 8
DLsw: START-LFSM TokenRing0 (4000.9404.0001->4000.5494.00dd) event:DLC-CloseStn.Cnf
DLsw: LFSM-Y: driving partner to close circuit
DLsw: START-LFSM Serial2 (4000.5494.00dd->4000.9404.0001) event:ADMIN-STOP
```

DLSw: END-LFSM (4000.5494.00dd->4000.9404.0001): state:CLOSE_PEND ->CLOSE_PEND

DLSw: END-LFSM (4000.9404.0001->4000.5494.00dd): state:CLOSE_PEND ->DISCONNECTED

DLSW Received-ctlQ : CLSI Msg : **DISCONNECT.Ind** dlen: 8 DLSw: START-LFSM Serial2 (4000.5494.00dd->4000.9404.0001) event:DLC-Disc.Ind DLSw: END-LFSM (4000.5494.00dd->4000.9404.0001): state:CLOSE_PEND ->CLOSE_PEND DLSW Received-ctlQ : CLSI Msg : CLOSE_STN.Cfm CLS_OK dlen: 8 DLSw: START-LFSM Serial2 (4000.5494.00dd->4000.9404.0001) event:DLC-CloseStn.Cnf DLSw: LFSM-Y: removing local switch entity DLSw: END-LFSM (4000.5494.00dd->4000.9404.0001): state:CLOSE_PEND ->DISCONNECTED

После того, как маршрутизатор получает **DISCONNECT.Ind (UA)** от AS/400, это заканчивает очищать сеанс и переходит в состояние разъединения.

[Дополнительные сведения](#)

- [Технологии IBM](#)
- [Cisco Systems – техническая поддержка и документация](#)