

Descrizione e risoluzione dei problemi di SDLC to LLC Network Media Translation

Sommario

[Introduzione](#)

[Prerequisiti](#)

[Requisiti](#)

[Componenti usati](#)

[Convenzioni](#)

[SDLLC](#)

[Configurazione SDLC](#)

[Configurazione SDLLC](#)

[Debug di SDLLC](#)

[DLSw Media Translation](#)

[Comandi show](#)

[Debug dei pacchetti SDLC durante DLSw/SDLC per PU2.1](#)

[Esempio di traduzione di supporti DLSw](#)

[DLSw: esecuzione della traduzione inversa dei supporti](#)

[Local DLSw Media Translation](#)

[Informazioni correlate](#)

[Introduzione](#)

In questo documento vengono fornite informazioni per la comprensione e la risoluzione dei problemi relativi a un SDLC (Synchronous Data Link Control) per la traduzione di supporti di rete LLC (Logical Link Control).

[Prerequisiti](#)

[Requisiti](#)

Nessun requisito specifico previsto per questo documento.

[Componenti usati](#)

Il documento può essere consultato per tutte le versioni software o hardware.

[Convenzioni](#)

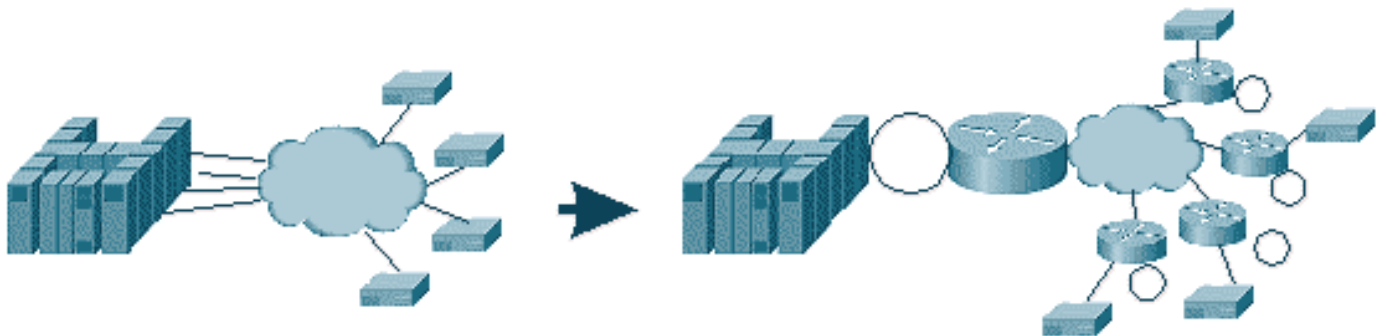
Per ulteriori informazioni sulle convenzioni usate, consultare il documento [Cisco sulle convenzioni](#)

[nei suggerimenti tecnici.](#)

SDLLC

SDLC-to-LAN conversion (SDLC) viene utilizzata per convertire una sessione SDLC per un dispositivo unità fisica 2 (PU2.0) in una sessione LLC2 (Logical Link Control). Ciò è molto utile se si dispone di una grande quantità di telecomandi alimentati in una singola porta Token Ring su un processore front-end (FEP).

La parte sinistra del diagramma mostra un FEP con molte linee SDLC che lasciano a postazioni remote. Il lato destro del diagramma mostra lo stesso scenario con i router Cisco.



I router permettono al FEP di avere solo l'interfaccia Token Ring. Da quel momento, sono presenti più postazioni remote che eseguono SDLLC sull'host, oltre al normale traffico source-route bridge (SRB).

Nota: l'uso di SDLLC per la conversione da LLC a SDLC si applica solo ai dispositivi PU2.0 e non al tipo di unità fisica 2.1 (PU2.1). PU2.1 è supportato nella commutazione DLSw (Data-Link Switching).

Per configurare SDLLC, è necessario un SRB nel router. Per informazioni su come configurare un SRB, fare riferimento a [Descrizione e risoluzione dei problemi di Local Source-Route Bridging](#).

Configurazione SDLC

Poiché SDLC viene convertito da un'interfaccia SDLC, è necessario che SDLC sia configurato correttamente. Completare la procedura seguente per configurare il protocollo SDLC:

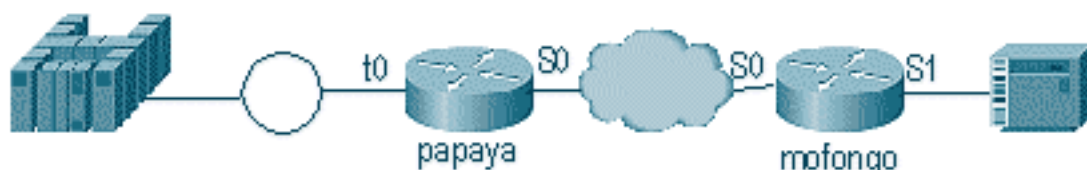
1. Utilizzare il comando **encapsulation sdlc** per modificare l'incapsulamento seriale in SDLC.
2. Utilizzare il comando **sdlc role primary** per modificare il ruolo del router in primario nella riga SDLC. **Nota:** negli ambienti STUN (Serial Tunneling) sono presenti ruoli principali e secondari. per ulteriori informazioni, fare riferimento a [Configurazione e risoluzione dei problemi di tunneling seriale \(STUN\)](#).
3. Utilizzare il comando **sdlc address xx** per configurare l'indirizzo di polling SDLC.

Configurazione SDLLC

Per configurare SDLLC, il primo comando emesso è **traddr**. Questo comando definisce in che modo il file SDLC viene convertito nell'ambiente LLC2. Completare la procedura seguente per configurare SDLLC:

1. Utilizzare il comando **sdllc traddr xxxx.xxxx.xx00 lr bn tr** per abilitare la traduzione dei supporti SDLLC su un'interfaccia seriale. Questo comando indica al router l'indirizzo MAC virtuale della stazione SDLC. Il comando specifica quindi il numero di anello locale (**lr**), il numero di ponte (**bn**) e il numero di anello di destinazione (**tr**). La **lr** deve essere unica nella rete. Il valore **bn** può essere compreso tra 1 e 15. Il **trn** deve essere l'anello virtuale del router. Se si sta configurando SDLC locale, è possibile puntare a un anello virtuale o a un'interfaccia (anello fisico collegato all'interfaccia Token Ring) nel router. **Nota:** le ultime due cifre dell'indirizzo MAC in questo comando sono **00**. Non è possibile impostare le ultime due cifre del **trader** perché il router usa queste cifre per inserire l'indirizzo SDLC di questa riga. Se si specificano le ultime due cifre, il router le sostituisce con l'indirizzo SDLC. L'host non risponde quindi a tale indirizzo MAC. Ad esempio, se l'indirizzo MAC del trader è configurato come 4000.1234.5678 e l'indirizzo SDLC è 0x01, il router utilizza l'indirizzo MAC 4000.1234.5601 per rappresentare il dispositivo SDLC nel dominio LLC. Inoltre, l'indirizzo MAC del trader è in formato non canonico, che è lo stesso formato del frame Token Ring.
2. Eseguire il comando **sdllc xid address xxxxxxxx** per specificare il valore di identificazione dello scambio (XID) appropriato affinché la stazione SDLC corrisponda ai valori VTAM (Virtual Telecommunications Access Method). Questa condizione viene determinata a partire da IDBLK e IDNUM nel nodo principale dello switch in VTAM. Se non corrisponde, lo scambio XID non riesce.
3. Utilizzare il comando **sdllc partner mac-address sdllc-address** per abilitare le connessioni per SDLLC. Specifica l'indirizzo MAC del partner, in genere l'host.

Viene visualizzata una semplice configurazione di esempio SDLLC. Il controller collegato SDLC viene visualizzato come dispositivo locale collegato a Token Ring al FEP.

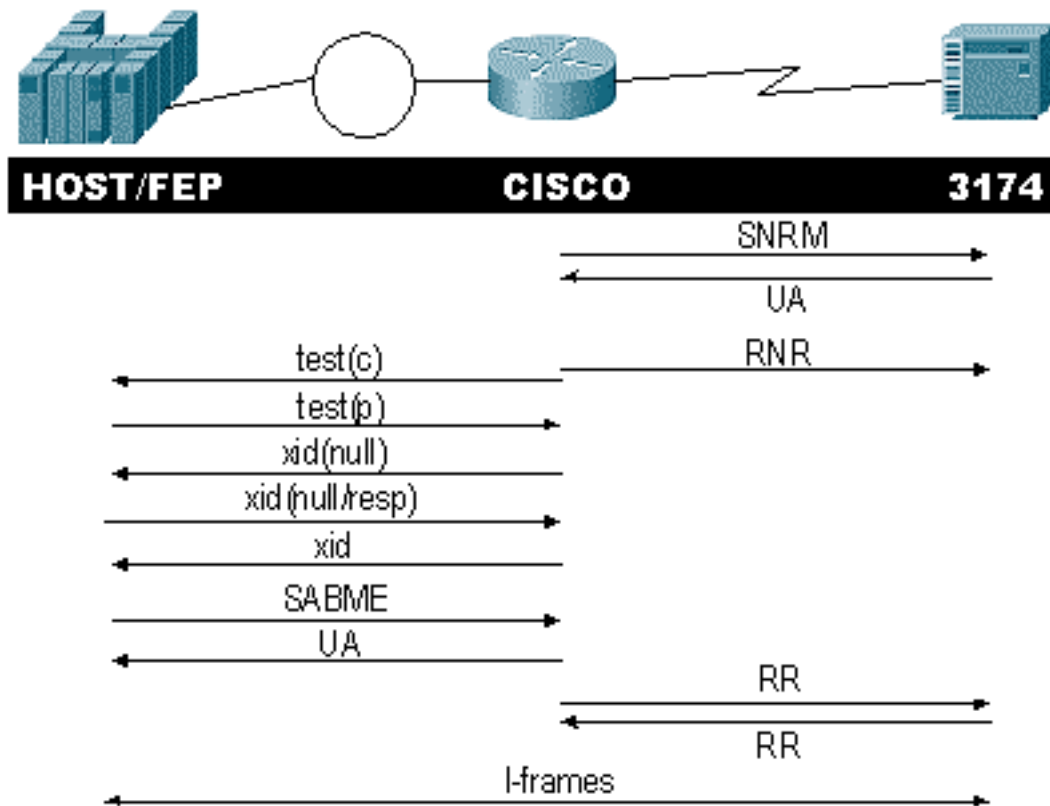


Papaya	Mofongo
<pre>source-bridge ring-group 100 source-bridge remote- peer 100 tcp 1.1.1.1 source-bridge remote- peer 100 tcp 1.1.2.1 local-ack interface tokenring 0 ip address 1.1.3.1 255.255.255.0 source-bridge 33 2 100 source-bridge spanning interface loopback 0 ip address 1.1.1.1 255.255.255.0</pre>	<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 interface loopback 0 ip address 1.1.2.1 255.255.255.0</pre>

[Debug di SDLLC](#)

Un problema SDLLC richiede la risoluzione dei problemi in due ambienti diversi: il mondo SDLC e il mondo LLC2 (Logical Link Control) in cui vengono tradotti i frame. Poiché è possibile utilizzare un solo tipo di controller, il debug di SDLC è più comprensibile rispetto al debug DLSw/SDLC.

Notare innanzitutto i flussi per l'avvio della sessione specifica:



Controllare la risposta del controller Set Normal Response Mode (SNRM). Il router non avvia la parte LLC finché la parte SDLC non è attiva e in esecuzione.

Utilizzare i seguenti comandi per verificare la risposta SNRM:

- `stato_sdlic`
- `stato_sdllc`

In questo esempio, il protocollo SNRM viene inviato al controller, che modifica lo stato della linea in SNRMSSENT. Se il router rimane in questo stato, non ha ricevuto l'avviso di ricezione senza numero (UA) dal controller. Ciò può significare che c'è qualcosa di sbagliato nella linea SDLC. In questo caso, il messaggio debug viene visualizzato come:

```
%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
```

Se il router riceve l'agente utente, **sdlc_state** si sposta da SNRM_SENT a CONNECT. Successivamente, lo stato SDLC si sposta da SDLC_PRI_WAIT a NET_UP_WAIT. In questo caso, il router può iniziare a sollevare il lato LLC della connessione. L'azione finale consiste nell'iniziare a inviare i messaggi ricevuti non pronti (RNR) alla linea SDLC. In questo modo il controller non potrà inviare informazioni finché il lato LLC non sarà operativo.

Successivamente, il router invia un messaggio di ricerca per individuare la posizione del partner.

```
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 rcv FORWARD TEST P/F(F3) 4000.3174.11c6 c000.1111.1111 00 01 -> Serial1
C6
caching rif
```

L'output precedente visualizza il polling di test inviato e ricevuto. Poiché questo esempio ha un controller collegato localmente e Token Ring, il polling di test lascia il router alla ricerca dell'indirizzo del partner. Dopo aver ricevuto il frame di test, il router inizia lo scambio XID. Il router memorizza nella cache il campo RIF (Routing Information Field) per questa sessione, che è possibile verificare con il comando **show rif**. Poiché si tratta di una PU2.0, il router invia un XID Format 0 Type 2 all'host dopo la risposta allo XID null.

```
SDLLC: O xid(null), 4000.1111.1111 4000.3174.11c6 4 4 [1000.14D3.0641.0051.12C2.0194.01F1.02C0]
SDLLC: NET UP WAIT rcv 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 rcv 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
```

Dopo lo scambio XID, il router riceve il comando Set Asynchronous Balanced Mode Extended (SABME) dall'host. In questo modo la procedura di avvio viene completata e il router risponde all'host con un'interfaccia utente. Ora, lo stato della linea SDLC passa da BUSY a CONNECT e gli I-frame possono passare attraverso il router.

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

[DLSw Media Translation](#)

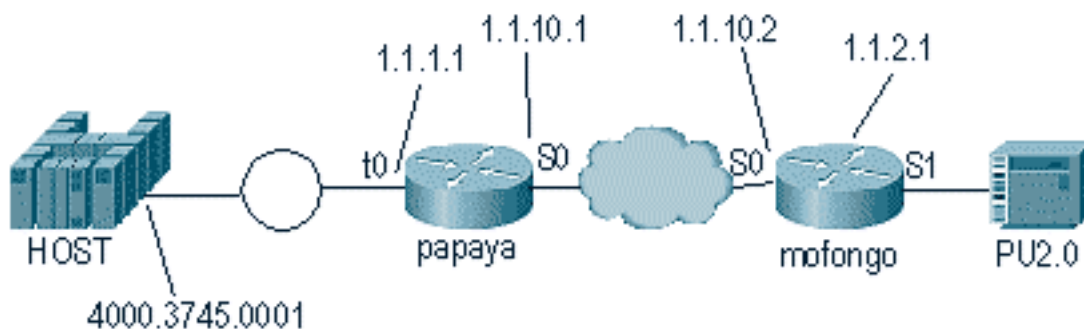
DLSw fornisce un importante miglioramento alla conversione dei supporti perché supporta PU2.1. Ciò consente di avere la conversione da SDLLC a LLC2 per i controller, come il 5494 e il 5394 (con opzione di aggiornamento a PU2.1 - IBM RPQ 8Q0775) a AS/400s. In questo modo si elimina la necessità di linee multipunto AS/400 STUN e cattive.

I parametri di configurazione per la traduzione multimediale DLsw sono leggermente diversi dai

parametri SDLLC. Viene aggiunto un comando DLSw, gli altri sono comandi SDLC. Completare la procedura seguente per configurare DLSw media translation:

1. Utilizzare il comando **encapsulation sdlc** per modificare l'incapsulamento seriale in SDLC. Poiché si sta per terminare la linea SDLC nel router, il router deve agire come router primario a scopo di polling. Questo comportamento è diverso da STUN in quanto il principale sarà HOST o AS/400.
2. Utilizzare il comando **sdlc role primary** per modificare il ruolo del router in primario nella riga SDLC.
3. Utilizzare il comando **sdlc address xx** per configurare l'indirizzo di polling SDLC. Questo è il punto in cui DLSw differisce da SDLLC. In SDLLC i comandi vengono specificati con la parola chiave **sdllc**. In DLSw, specificare i comandi con la parola chiave **sdlc**.
4. Eseguire il comando **sdlc vmac xxxx.xxxx.xx00** per configurare l'indirizzo MAC virtuale del controller SDLC. Questo parametro indica al router l'indirizzo MAC virtuale per questo controller SDLC nell'ambiente LLC2. Ricordarsi di lasciare l'ultimo byte impostato su **00** perché l'indirizzo di polling è stato aggiunto qui (**indirizzo sdlc**).
5. Eseguire il comando **sdlc xid nn xxxxxxxx** per configurare l'XID per questa unità di elaborazione 2.0. In questo comando, **nn** è l'indirizzo di polling del controller e **xxxxxxxx** è l'XID per questa PU2.0 (IDBLOCK e IDNUM codificati nel nodo principale dello switch in VTAM). **Nota:** se si dispone di una PU2.1, è in corso la negoziazione di XID. Di conseguenza, il comando cambia.
6. Eseguire il comando **sdlc xid nn xid-poll** per configurare l'XID per questa unità di elaborazione 2.1. In questo comando, **nn** è l'indirizzo di polling della stazione.
7. Eseguire il comando **sdlc partner xxxx.xxxx.xxxx.nn** per configurare l'indirizzo MAC del partner del router. In questo comando, **nn** è l'indirizzo di polling per il controller in questione. È importante specificare l'indirizzo del controller, in quanto nelle righe multipunto è possibile che un controller sia diretto a un host e un altro controller a un host diverso.
8. Eseguire il comando **sdlc dlsw nn** per configurare DLSw per il controller specifico. In questo comando, **nn** è l'indirizzo di polling del controller o dei controller nel multidrop. Questo comando consente di specificare più indirizzi di polling in un unico comando. **Nota:** prestare attenzione al bug #CSCdi75481. Per ulteriori informazioni, fare riferimento a [Bug Toolkit](#) (solo utenti [registrati](#)). se il comando **sdlc dlsw nn** non viene rimosso prima di modificare l'indirizzo SDLC del router, il codice CLS non è in grado di comunicare correttamente il DLSw con l'interfaccia SDLC. In questo modo, l'interfaccia si comporta come se non fosse stato configurato nulla. Questo bug è stato risolto nel software Cisco IOS® versione 11.1(8.1) 11.1(8.1)A01(01.03) 11.1(8.1)A01(01.02) e successive.

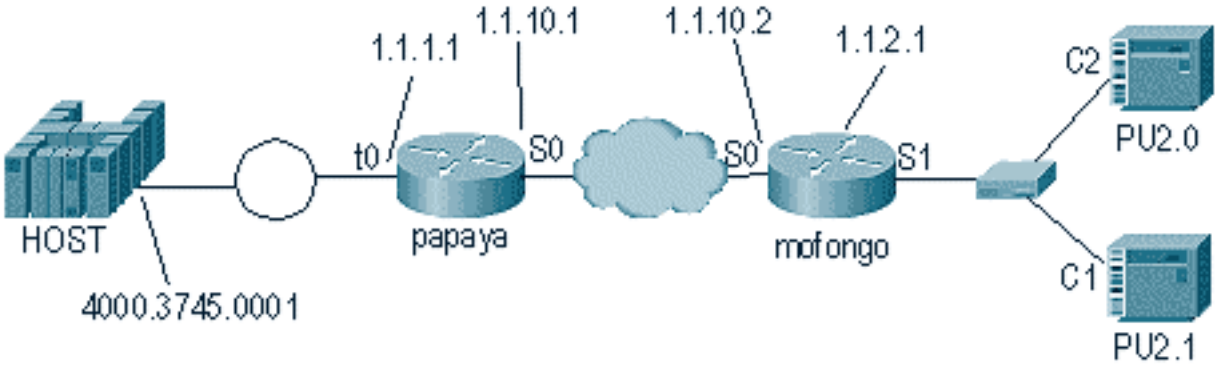
Viene visualizzata una configurazione di esempio per un controller DLSw SDLC PU2.0.



Papaya	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 sdhc sdhc role primary sdhc vmac 4000.3174.0000 sdhc address c1 sdhc xid c1 01767890 sdhc partner 4000.3745.0001 c1 sdhc dlsw c1 </pre>
---	--

Quando si codifica un multidrop, tenere presente che le PU2.1 sono più intelligenti e hanno più informazioni da scambiare rispetto a un normale dispositivo PU2.0. Questo è importante quando si configura un ambiente multidrop, perché è necessario codificare la linea come primaria per il dispositivo PU2.0. È inoltre necessario aggiungere il **poll xid** per l'indirizzo SDLC del dispositivo PU2.1 in modo che il codice comprenda cosa fare con ogni controller. Questo è un esempio della configurazione.



Papaya	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 sdhc sdhc role primary sdhc vmac 4000.3174.0000 sdhc address c1 xid-poll sdhc partner 4000.9404.0001 c1 sdhc address c2 01767890 </pre>

	<pre>sdlc partner 4000.9404.0001 c2 sdlc dlsw c1 c2</pre>
--	---

Comandi show

Per ulteriori informazioni sui comandi show usati per la traduzione del supporto DLSw, consultare il documento [Data-Link Switching Plus](#).

Debug dei pacchetti SDLC durante DLSw/SDLC per PU2.1

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

La prima cosa che si verifica è un **XID**, o **BF**, per l'indirizzo di broadcast SDLC di **FF**.

```
Serial2 SDLC output      FFBF
```

Successivamente, viene ricevuto un **XID** dal modello 5494. Questo è un formato **XID 2** di tipo 3, visualizzato nell'output del comando **debug sdlc 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
```

Di seguito sono riportate le spiegazioni di diversi campi di questo comando:

- **073000DD**: questo campo è l'ID blocco e il numero ID configurati in 5494. L'ID blocco e il numero ID sono denominati **XID** e vengono inviati dal 5494 al peer durante la negoziazione della sessione.
- **NETA**: questo campo è il **NETID** (Advanced Peer-to-Peer Networking) utilizzato. In genere, questo campo corrisponde al **NETID** configurato nel peer. In questo caso, il peer è AS/400.
- **CP5494** — Questo campo è il nome del punto di controllo (**CP**) del modello 5494.
- **DD**: questo campo è l'indirizzo SDLC.

Successivamente, l'**XID** viene ricevuto da 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** — Questo campo è l'ID blocco e il numero ID di AS/400.
- **RTP400A**: questo campo è il nome TCP di AS/400. Il nome CP si trova nel file DSPNETA (Display Network Attributes) in AS/400.

Quindi, SNRM (93) e UA (73) vengono visualizzati sulla linea. Prima del protocollo SNRM, il router usa sempre l'indirizzo di broadcast. Da questo punto in poi, il router usa sempre l'indirizzo di polling effettivo di DD.

```

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

```

A questo punto, la connessione viene sospesa a causa dello stato di ricevitore pronto (RR) stabile tra il router e lo switch 5494.

Nota: se il router su cui deve essere eseguito il debug ha altre interfacce SDLC e non si sta eseguendo il log nel buffer, il router può sospendere l'esecuzione. La verifica del momento in cui è possibile eseguire il debug sul terminale rispetto alla registrazione è fondamentale. In caso di dubbio, usare sempre il log nel buffer e il comando **show log** per visualizzare i debug SDLC

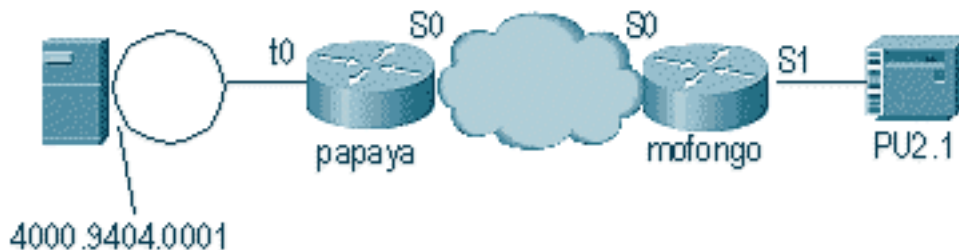
Spegnere il controller su AS/400. Ciò consente di visualizzare DISC (53) e UA (73) sul lato SDLC della sessione.

```

Serial2 SDLC output      DD53
Serial2 SDLC input      DD73

```

Esempio di traduzione di supporti DLSw



Dopo l'accensione dell'interfaccia, il router inizia il processo determinando la posizione del controller remoto.

```

%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
```

Dopo aver ricevuto il frame ICR, DLsw avvia la macchina a stati finiti (FSM) per questa sessione. Questa operazione viene eseguita dai messaggi **REQ_OPNSTN.Req** e **REQ_OPNSTN.Cfm** scambiati tra DLsw e Cisco Link Services Interface (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
```

Dopo la conversazione con CLSI, DLsw invia i frame **CUR di avvio sessione** al router remoto. Queste situazioni si verificano solo tra i due router.

```
%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
```

Una volta stabilito il circuito, il router invia l'XID memorizzato e avvia lo scambio XID. È importante capire dove si trovano gli XID. In questo esempio, l'ID-DLC (Data-Link Control) indica che l'XID proviene dalla stazione DLC locale, mentre l'ID-WAN proviene dal router remoto o dalla stazione remota.

```
DLsw: START-FSM (488636): event:DLC-Id state:CKT_ESTABLISHED
DLsw: core: dlsw_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-RECVSSP: 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: dlsw_action_g()
DISP Sent : CLSI Msg : ID.Rsp  dlen: 12
DLsw: END-FSM (488636): state:CKT_ESTABLISHED->CKT_ESTABLISHED
```

```
%DLSWC-3-RECVSSP: SSP OP = 7( XID ) from peer 10.17.2.198(2065)
DLsw: START-FSM (488636): event:WAN-XID state:CKT_ESTABLISHED
DLsw: core: dlsw_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: dlsw_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-RECVSSP: SSP OP = 7( XID ) from peer 10.17.2.198(2065)
DLSw: START-FSM (488636): event:WAN-XID state:CKT_ESTABLISHED
DLSw: core: dlsw_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: dlsw_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-RECVSSP: SSP OP = 7( XID ) from peer 10.17.2.198(2065)
DLSw: START-FSM (488636): event:WAN-XID state:CKT_ESTABLISHED
DLSw: core: dlsw_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: dlsw_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
```

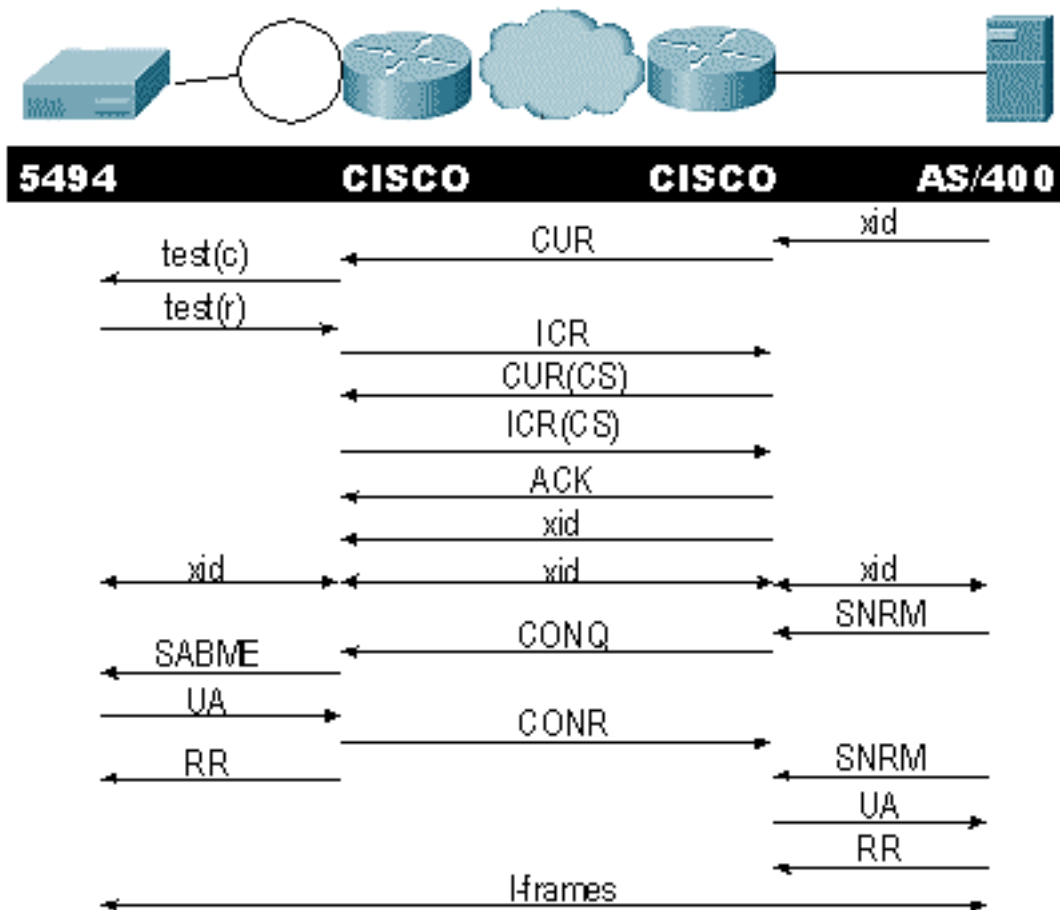
Il router riceve il messaggio **CONQ** da AS/400 (SABME). Questo viene convertito nella linea seriale come **SNRM**. Quindi, il router attende l'UA sulla linea seriale (**CONNECT.Cfm**) e invia il comando **CONR** all'altro lato. In questo modo lo stato della sessione viene impostato su **CONNECTED**.

```
%DLSWC-3-RECVSSP: SSP OP = 8( CONQ ) from peer 10.17.2.198(2065)
DLSw: START-FSM (488636): event:WAN-CONQ state:CKT_ESTABLISHED
DLSw: core: dlsw_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: dlsw_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
```

[DLSw: esecuzione della traduzione inversa dei supporti](#)

Un'altra impostazione comune è **reverse-sdlc**. In modalità reverse SDLC, la stazione principale è collegata al router tramite una linea SDLC. Questa condizione si verifica in genere negli ambienti host in cui gli utenti desiderano migrare l'host a un allegato Token Ring. Reverse SDLC cambia il modo in cui DLSw gestisce la linea SDLC, in quanto spesso non è chiaro se l'unità di elaborazione remota è attiva o meno.



Innanzitutto, poiché in questo caso AS/400 è primario o è impostato per essere negoziabile nel ruolo, deve avviare la sessione. Quando l'AS/400 invia il primo XID dopo che la linea seriale è diventata operativa, il router avvia il processo di ricerca per il controller remoto. Una volta impostato il circuito, può iniziare la negoziazione XID nella linea.

Al termine della negoziazione XID, AS/400 invia il messaggio SNRM al router. In questo modo, il router invia il messaggio CONQ e si aspetta che il messaggio CONR venga inviato dal router remoto. Il router non può rispondere con l'interfaccia utente finché non rileva un messaggio SNRM e dopo aver ricevuto il messaggio CONR. In quasi tutte le versioni del codice, il router attende 30 secondi fino al timeout della sessione. Ciò riguarda la ricezione di SNRM dal dispositivo primario quando il dispositivo primario riceve il CONR dall'host remoto.

Nell'ultimo codice Cisco IOS versione 11.1, le impostazioni predefinite sono state modificate a un minuto anziché a 30 secondi. In AS/400, questo timeout è denominato **timer di risposta non produttivo** e viene impostato per impostazione predefinita su 32 secondi.

[Local DLSw Media Translation](#)



```
%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
```

La prima cosa che si nota in DLSw local è lo XID del lato seriale. Questo XID deve essere archiviato finché il router non invia i frame/le risposte di test LLC tramite.

```
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
```

Successivamente, la stazione di test lascia il router e la risposta viene restituita da AS/400. A questo punto, il router può creare l'FSM locale.

Nota: ricordare che si tratta di una sessione locale.

```
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
```

Dopo aver confermato localmente che l'FSM è pronto, il router può inviare l'XID al partner. Nell'esempio, il partner è 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
```

Quindi, viene ricevuto un XID dal Token Ring. La lunghezza di ID.Ind è 108. Il router inoltra questo

XID al partner in questo scenario, che è la linea SDLC. È indicato dal **file ID.Reg** inviato. Ogni volta che il router riceve un pacchetto, deve avviare la macchina a stati finiti lineari (LFSM). Questa è la chiave per comprendere questo debug, in quanto indica l'inizio e i punti in cui sta andando.

```
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.Reg   dlen: 88
DLSw: END-LFSM (4000.9404.0001->4000.5494.00dd): state:ESTABLISHED ->ESTABLISHED
```

Successivamente, la risposta XID viene ricevuta dalla linea seriale e inoltrata al partner (la stazione Token Ring in questo esempio). Questa operazione continua fino al termine dello scambio XID per questo dispositivo 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
```

Dopo lo scambio XID, il router riceve una SABME da AS/400 tramite **CONNECT.Ind**. In questo modo il router invia un messaggio **CONNECT.Reg** alla linea SDLC, ossia il protocollo SNRM. Quindi, un messaggio **CONNECT.Cfm** (UA) viene ricevuto dalla linea seriale, che fa sì che il codice DLSw invii un messaggio **CONNECT.Rsp** (UA) all'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.Reg   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

Viene visualizzata la sessione alla chiusura del controller (SDLC).

%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

Successivamente, il router invia un messaggio DISC all'AS/400 (DISCONNECT.Rsp). Poi inizia a distruggere il circuito locale.

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

Dopo aver ricevuto il file DISCONNECT.Ind (UA) da AS/400, il router completa la cancellazione della sessione e passa allo stato di disconnessione.

[Informazioni correlate](#)

- [IBM Technologies](#)
- [Documentazione e supporto tecnico – Cisco Systems](#)