

Présentation et dépannage de la traduction de supports réseau SDLC en LLC

Contenu

[Introduction](#)

[Conditions préalables](#)

[Conditions requises](#)

[Components Used](#)

[Conventions](#)

[SDLLC](#)

[Configuration SDLC](#)

[Configuration SDLLC](#)

[Débogage de SDLLC](#)

[Traduction multimédia DLSw](#)

[Commandes show](#)

[Débogage de paquets SDLC pendant DLSw/SDLC pour PU2.1](#)

[Exemple de traduction multimédia DLSw](#)

[DLSw Exécution de la traduction de support inverse](#)

[Traduction multimédia DLSw locale](#)

[Informations connexes](#)

Introduction

Ce document fournit des informations permettant de comprendre et de dépanner une traduction de support réseau SDLC (Synchronous Data Link Control) vers LLC (Logical Link Control).

Conditions préalables

Conditions requises

Aucune spécification déterminée n'est requise pour ce document.

Components Used

Ce document n'est pas limité à des versions de matériel et de logiciel spécifiques.

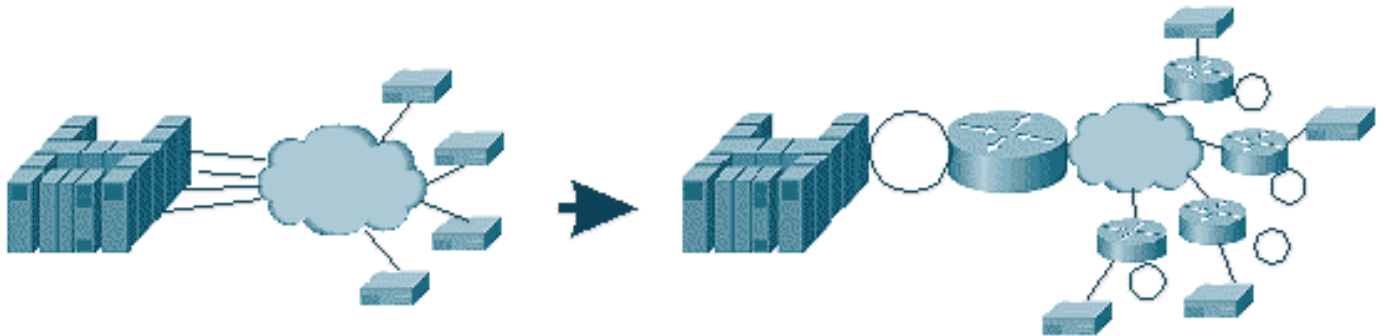
Conventions

Pour plus d'informations sur les conventions utilisées dans ce document, reportez-vous à [Conventions relatives aux conseils techniques Cisco](#).

SDLLC

La conversion SDLC-LAN (SDLLC) permet de convertir une session SDLC pour un périphérique PU2.0 (Physical Unit 2) en session LLC2 (Logical Link Control). Cela est très utile si vous avez un grand nombre de contrôleurs distants alimentés en un seul port Token Ring sur un processeur frontal (FEP).

La partie gauche de ce diagramme affiche un FEP avec de nombreuses lignes SDLC quittant les emplacements distants. Le côté droit de ce schéma présente le même scénario avec les routeurs Cisco.



Les routeurs permettent au FEP de n'avoir que l'interface Token Ring. À partir de ce point, plusieurs emplacements distants exécutent SDLLC vers l'hôte, ainsi que le trafic SRB (Source-Route Bridge) régulier.

Remarque : l'utilisation de SDLLC pour la conversion LLC vers SDLC s'applique uniquement aux périphériques PU2.0, et non pas aux unités physiques de type 2.1 (PU2.1). PU2.1 est pris en charge dans la commutation de liaison de données (DLSw).

Pour configurer SDLLC, vous avez besoin d'un SRB dans le routeur. Référez-vous à [Compréhension et dépannage du pontage source-route local](#) pour plus d'informations sur la façon de configurer un SRB.

Configuration SDLC

Puisque SDLLC convertit à partir d'une interface SDLC, vous devez d'abord configurer SDLC correctement. Effectuez les étapes suivantes pour configurer SDLC :

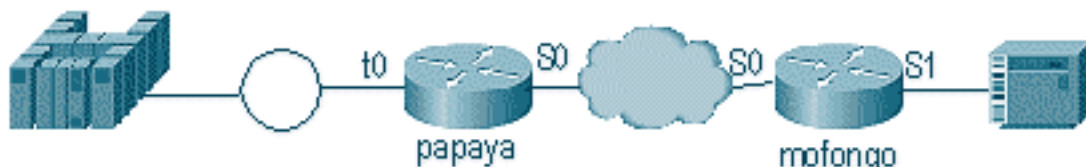
1. Exécutez la commande **encapsulation sdlc** pour remplacer l'encapsulation série par SDLC.
2. Émettez la commande **sdlc role primary** pour changer le rôle du routeur en principal dans la ligne SDLC. **Remarque :** dans les environnements STUN (Serial Tunneling), il existe des rôles principaux et secondaires. Référez-vous à [Configuration et dépannage de la tunnellation série \(STUN\)](#) pour plus d'informations.
3. Émettez la commande **sdlc address xx** pour configurer l'adresse d'interrogation SDLC.

Configuration SDLLC

Pour configurer SDLLC, la première commande émise est **traddr**. Cette commande définit ce vers quoi le SDLC convertit dans l'environnement LLC2. Suivez ces étapes pour configurer SDLLC :

- Émettez la commande **sdllc traddr xxxx.xxxx.xx00 lr bn tr** pour activer la traduction de support SDLLC sur une interface série. Cette commande indique au routeur l'adresse MAC virtuelle de la station SDLC. Ensuite, la commande spécifie le numéro d'anneau local (**lr**), le numéro de pont (**bn**) et le numéro d'anneau cible (**tr**). Le **lr** doit être unique dans le réseau. La **valeur bn** peut être comprise entre 1 et 15. Le **trn** doit être l'anneau virtuel dans le routeur. Si vous configurez le SDLLC local, vous pouvez faire pointer ce point sur un anneau virtuel ou sur une interface (anneau physique connecté à l'interface Token Ring) du routeur. **Remarque** : Les deux derniers chiffres de l'adresse MAC dans cette commande sont **00**. Vous ne pouvez pas définir les deux derniers chiffres de l'échange car le routeur utilise ces chiffres pour insérer l'adresse SDLC de cette ligne. Si vous spécifiez les deux derniers chiffres, le routeur les remplace par l'adresse SDLC. Ensuite, l'hôte ne répond pas pour cette adresse MAC. Par exemple, si l'adresse MAC de la transaction est configurée comme 4000.1234.5678 et que l'adresse SDLC est 0x01, le routeur utilise l'adresse MAC 4000.1234.5601 pour représenter le périphérique SDLC dans le domaine LLC. En outre, l'adresse MAC de l'échange est au format non canonique, qui est le même format que la trame Token Ring.
- Émettez la commande **sdllc xid address xxxxxxxx** pour spécifier la valeur XID (exchange identification) appropriée pour que la station SDLC corresponde aux valeurs VTAM (Virtual Telecommunications Access Method). Ceci est déterminé à partir des IDBLK et IDNUM dans le noeud principal du commutateur dans VTAM. Si cela ne correspond pas, l'échange XID échoue.
- Émettez la commande **sdllc partner mac-address sdllc-address pour activer les connexions pour SDLLC**. Indique l'adresse MAC du partenaire, qui est généralement l'hôte.

Un exemple simple de configuration SDLLC s'affiche. Le contrôleur SDLC connecté apparaît comme un périphérique Token Ring local connecté au FEP.

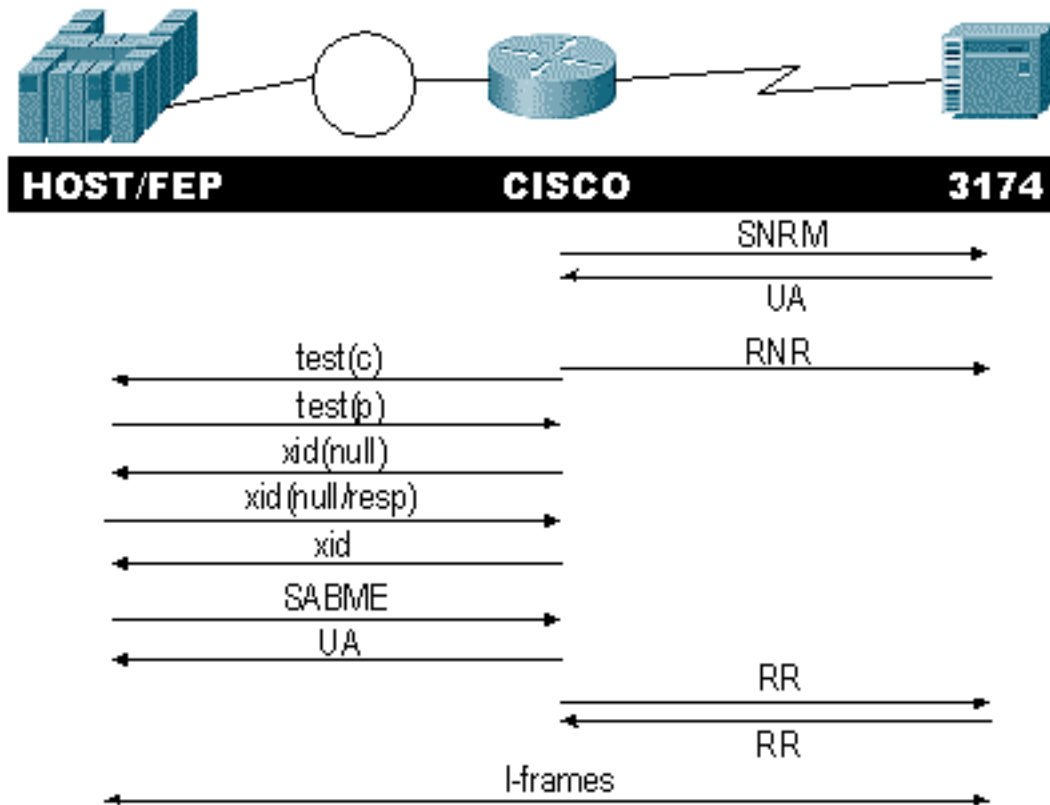


Papaya	Mofongo
<pre>source-bridge ring-group 100 source-bridge remote- peer 100 tcp 1.1.1.1 source-bridge remote- peer 100 tcp 1.1.2.1 local-ack interface tokenring 0 ip address 1.1.3.1 255.255.255.0 source-bridge 33 2 100 source-bridge spanning interface loopback 0 ip address 1.1.1.1 255.255.255.0</pre>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

Débogage de SDLLC

Un problème SDLLC nécessite le dépannage de deux environnements différents : le monde SDLC et le monde de contrôle de liaison logique de type 2 (LLC2) à l'endroit où vous traduisez les trames. Comme vous ne pouvez avoir qu'un seul type de contrôleur, le débogage de SDLLC est plus facile à comprendre que la commutation de liaison de données (DLSw)/SDLC.

Tout d'abord, notez les flux pour ce démarrage de session spécifique :



Vérifiez la réponse SNRM (Set Normal Response Mode) du contrôleur. Le routeur ne démarre pas la partie LLC tant que la partie SDLC n'est pas opérationnelle.

Émettez ces commandes pour vérifier la réponse SNRM :

- `état_sdlic`
- `état_sdllc`

Dans cet exemple, le SNRM est envoyé au contrôleur, qui change l'état de la ligne en SNRMSSENT. Si le routeur reste dans cet état, il n'a pas reçu l'accusé de réception non numéroté (UA) du contrôleur. Cela peut signifier que quelque chose ne va pas avec la ligne SDLC. Si cela se produit, le débogage s'affiche comme suit :

```
%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 SNRMSENT
-> CONNECT
SDLLC_STATE: Serial1 C6 SDLC PRI WAIT
-> NET UP WAIT
SDLC_STATE: (5235700) Serial1 C6 CONNECT
-> USBUSY
```

Si le routeur reçoit l'UA, le **sdlc_state** passe de SNRM_SENT à CONNECT. Ensuite, l'état SDLLC passe de SDLC_PRI_WAIT à NET_UP_WAIT. Dans ce cas, le routeur peut commencer à activer le côté LLC de la connexion. L'action finale consiste à commencer à envoyer des RNR (réception not ready(s)) à la ligne SDLC. Cela empêche le contrôleur d'envoyer des informations jusqu'à ce que le côté LLC soit opérationnel.

Ensuite, le routeur envoie un explorateur pour trouver l'emplacement de son partenaire.

```
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
```

La sortie précédente affiche le sondage de test envoyé et reçu. Comme cet exemple comporte un contrôleur connecté localement et une Token Ring, le sondage de test laisse le routeur rechercher l'adresse du partenaire. Une fois que le routeur a reçu la trame de test, il commence l'échange XID. Le routeur met en cache le champ RIF (Routing Information Field) pour cette session, que vous pouvez vérifier à l'aide de la commande **show rif**. Comme il s'agit d'une PU2.0, le routeur envoie un XID de type 2 de format 0 à l'hôte après la réponse à la valeur Null de 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
```

Après l'échange XID, le routeur reçoit le SABME (Set Asynchronous Balancing Mode Extended) de l'hôte. Ceci finalise la procédure de démarrage et le routeur répond avec un UA à l'hôte. Maintenant, l'état de la ligne SDLC passe de USBUSY à CONNECT, et les trames I peuvent passer par le routeur.

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

[Traduction multimédia DLSw](#)

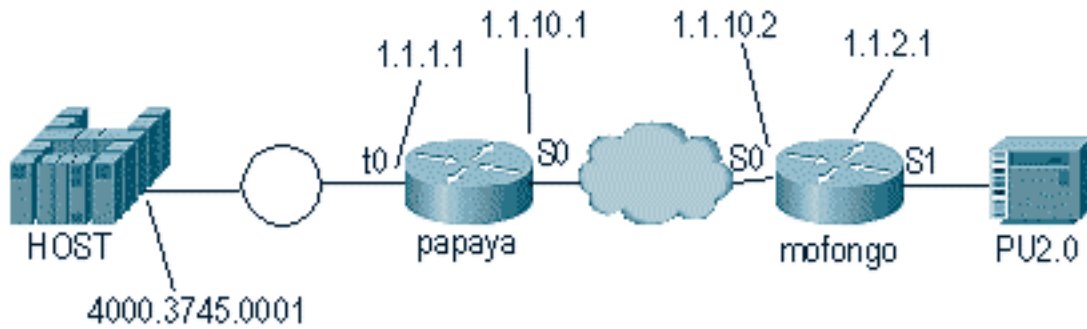
DLSw fournit une amélioration majeure à la traduction de support car il prend en charge PU2.1. Cela lui permet de convertir SDLLC en LLC2 pour les contrôleurs, tels que les modèles 5494 et 5394 (avec option de mise à niveau vers PU2.1 - IBM RPQ 8Q0775) en AS/400. Cela élimine le

besoin de lignes multipoints STUN et AS/400 défectueuses.

Les paramètres de configuration de la traduction de support DLSw sont légèrement différents des paramètres SDLLC. Il y a une commande DLSw qui est ajoutée, les autres sont des commandes SDLC. Suivez ces étapes pour configurer la traduction de support DLSw :

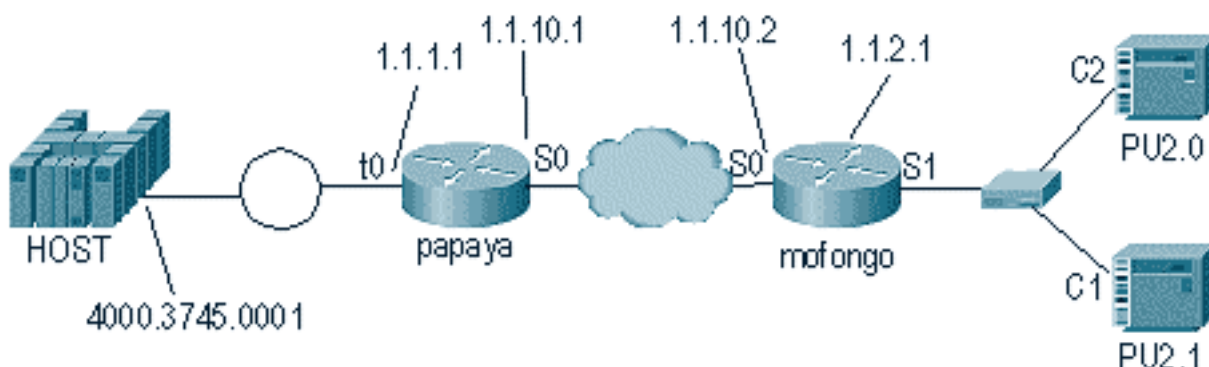
1. Exécutez la commande **encapsulation sdlc** pour remplacer l'encapsulation série par SDLC. Puisque vous allez terminer la ligne SDLC dans le routeur, le routeur doit agir en tant que routeur principal à des fins d'interrogation. Ceci est différent de STUN, car le principal sera l'hôte ou l'AS/400.
2. Émettez la commande **sdlc role primary** pour changer le rôle du routeur en principal dans la ligne SDLC.
3. Émettez la commande **sdlc address xx** pour configurer l'adresse d'interrogation SDLC. C'est là que DLSw diffère de SDLLC. Dans SDLLC, vous spécifiez des commandes avec le mot clé **sdllc**. Dans DLSw, spécifiez des commandes avec le mot clé **sdlc**.
4. Émettez la commande **sdlc vmac xxx.xxxx.xx00** pour configurer l'adresse MAC virtuelle pour le contrôleur SDLC. Ce paramètre indique au routeur l'adresse MAC virtuelle de ce contrôleur SDLC dans l'environnement LLC2. N'oubliez pas de laisser le dernier octet sur **00** car l'adresse d'interrogation y est ajoutée (**adresse sdlc**).
5. Exécutez la commande **sdlc xid nn xxxxxxxx** pour configurer le XID pour ce PU 2.0. Dans cette commande, **nn** est l'adresse d'interrogation du contrôleur et **xxxxxx** est le XID de ce PU2.0 (IDBLOCK et IDNUM codés dans le noeud principal du commutateur dans VTAM). **Remarque** : si vous avez un PU2.1, il y a une négociation de XID. Ainsi, la commande change.
6. Émettez la commande **sdlc xid nn xid-poll** pour configurer le XID pour ce PU 2.1. Dans cette commande, **nn** est l'adresse d'interrogation de la station.
7. Exécutez la commande **sdlc partner xxx.xxxx.xxxx nn** pour configurer l'adresse MAC du partenaire de routeur. Dans cette commande, **nn** est l'adresse d'interrogation du contrôleur en question. Il est important de spécifier l'adresse du contrôleur, car dans les lignes multipoints il peut y avoir un contrôleur dirigé vers un hôte et un autre contrôleur dirigé vers un autre hôte.
8. Émettez la commande **sdlc dlsw nn** pour configurer DLSw pour le contrôleur spécifique. Dans cette commande, **nn** est l'adresse d'interrogation du contrôleur ou des contrôleurs dans la multidiffusion. Cette commande vous permet de spécifier plusieurs adresses d'interrogation dans une seule commande. **Note** : Attention au bogue #CSCdi75481. Référez-vous à [Bug Toolkit](#) (clients [enregistrés](#) uniquement) pour plus d'informations. Si la commande **sdlc dlsw nn** n'est pas supprimée avant de modifier l'adresse SDLC du routeur, le code CLS ne peut pas communiquer correctement DLSw avec l'interface SDLC. L'interface se comporte ainsi comme si rien n'avait été configuré. Ce bogue a été corrigé dans le logiciel Cisco IOS® Versions 11.1(8.1) 11.1(8.1)AA01(01.03) 11.1(8.1)AA01(01.02) et ultérieures.

Un exemple de configuration pour un contrôleur DLSw SDLC PU2.0 s'affiche.



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>

Lorsque vous codez un multipoint, n'oubliez pas que les PU2.1 sont plus intelligents et ont plus d'informations à échanger qu'un périphérique PU2.0 standard. Ceci est important lors de la configuration d'un environnement multipoint, car vous devez coder la ligne en tant que ligne principale pour le périphérique PU2.0. Vous devez également ajouter le **sondage xid** pour l'adresse SDLC du périphérique PU2.1 afin que le code comprenne quoi faire avec chacun des contrôleurs. Voici un exemple de configuration.



Papaya	Mofongo
<pre> source-bridge ring- group 100 dlsw local-peer peer-id 1.1.1.1 dlsw remote-peer 0 tcp </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 </pre>

1.1.2.1	!
!	interface serial 0
interface serial 0	ip address 1.1.10.2
ip address 1.1.10.1	255.255.255.0
255.255.255.0	!
!	interface serial 1
interface tokenring 0	no ip address
ip address 1.1.1.1	encapsulation sdhc
255.255.255.0	sdhc role primary
ring-speed 16	sdhc vmac 4000.3174.0000
source-bridge 1 1 100	sdhc address c1 xid-poll
source-bridge spanning	sdhc partner 4000.9404.0001 c1
	sdhc address c2 01767890
	sdhc partner 4000.9404.0001 c2
	sdhc dlsw c1 c2

Commandes show

Référez-vous à [Data-Link Switching Plus](#) pour plus d'informations sur les commandes show utilisées pour la traduction multimédia DLSw.

Débogage de paquets SDLC pendant DLSw/SDLC pour PU2.1

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

La première chose à se produire est un XID, ou **BF** à l'adresse de diffusion SDLC de **FF**.

```
Serial2 SDLC output      FFBF
```

Ensuite, un XID est reçu du 5494. Il s'agit d'un format XID de type 2 type 3, qui est affiché dans la sortie de commande **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
```

Voici des explications de plusieurs champs de cette commande :

- **073000DD** — Ce champ correspond à l'ID de bloc et au numéro d'ID configuré dans le 5494. L'ID de bloc et le numéro d'ID sont appelés XID et sont envoyés par le 5494 à l'homologue lors de la négociation de session.
- **NETA** - Ce champ est l'identificateur réseau APPN (Advanced Peer-to-Peer Networking) utilisé. Normalement, ce champ correspond au NETID configuré dans l'homologue. Dans ce cas, l'homologue est un AS/400.
- **CP5494** — Ce champ correspond au nom du point de contrôle (CP) du 5494.
- **DD** — Ce champ est l'adresse SDLC.

Ensuite, le XID est reçu de l'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** — Ce champ correspond à l'ID de bloc et au numéro d'ID de l'AS/400.
- **RTP400A** — Ce champ correspond au nom CP de l'AS/400. Le nom CP se trouve dans le fichier Display Network Attributes (DSPNETA) de l'AS/400.

Ensuite, le SNRM (93) et l'UA (73) sont affichés sur la ligne. Avant le SNRM, le routeur utilise toujours l'adresse de diffusion. À partir de ce point, le routeur utilise toujours l'adresse d'interrogation réelle de DD.

```

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

```

À ce stade, la connexion est interrompue en raison de l'état RR (Receiver Ready) fixe entre le routeur et le 5494.

Remarque : Si le routeur sur lequel vous devez exécuter le débogage a d'autres interfaces SDLC et que vous ne enregistrez pas en mémoire tampon, le routeur peut suspendre. Comprendre quand vous pouvez exécuter un débogage sur le terminal par rapport à la journalisation est une expérience. Si vous n'êtes pas sûr, utilisez toujours logging buffered et la commande **show log** pour afficher les débogages SDLC

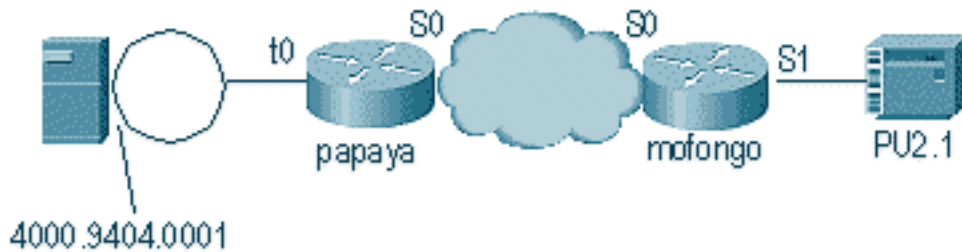
Mettez le contrôleur hors tension sur l'AS/400. Vous pouvez ainsi voir DISK (53) et UA (73) qui résultent du côté SDLC de la session.

```

Serial2 SDLC output      DD53
Serial2 SDLC input      DD73

```

[Exemple de traduction multimédia DLSw](#)



Une fois l'interface activée, le routeur commence le processus en déterminant l'emplacement du contrôleur distant.

```
%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
```

Après réception de la trame ICR, DLSw démarre la machine à état fini (FSM) pour cette session. Ceci est effectué par les messages **REQ_OPNSTN.Req** et **REQ_OPNSTN.Cfm** qui se trouvent entre DLSw et 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
```

Après la conversation avec CLSI, DLSw envoie des trames **CUR de démarrage de session** au routeur distant. Celles-ci se produisent uniquement entre les deux routeurs.

```
%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
```

Une fois le circuit établi, le routeur envoie le **XID** stocké et démarre l'échange **XID**. Il est important de comprendre où se trouvent les **XID**. Dans cet exemple, l'**ID** de contrôle de liaison de données (DLC) signifie que le **XID** provient de la station DLC locale et que le **WAN-XID** provient du routeur distant ou de la station distante.

```
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: dls_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: 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-RECVSSP: 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-RECVSSP: 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

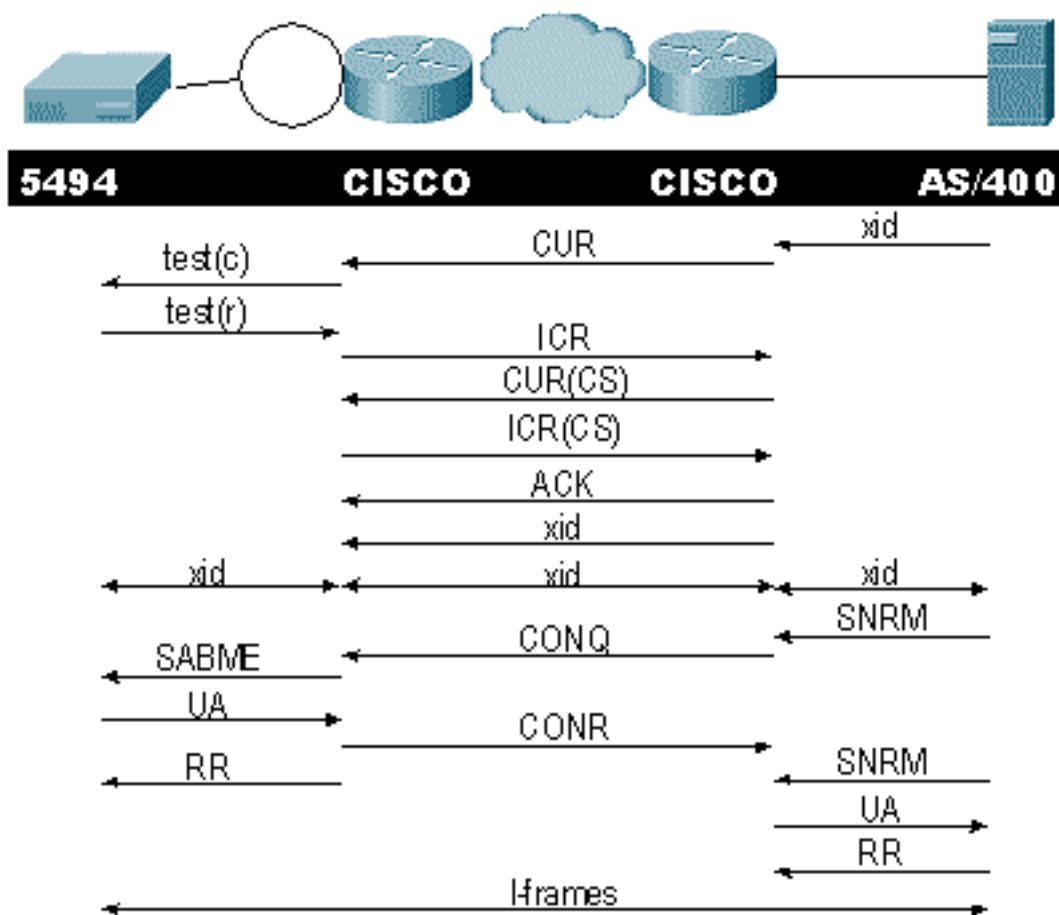
Le routeur reçoit la **CONQ** de l'AS/400 (SABME). Ceci est traduit en ligne série en tant que **SNRM**. Ensuite, le routeur attend l'UA sur la ligne série (**CONNECT.Cfm**) et envoie le **CONR** à l'autre extrémité. L'état de la session devient **CONNECTED**.

%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: 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

DLSw Exécution de la traduction de support inverse

Une autre configuration courante est **inverse-sdlc**. Dans SDLLC inverse, la station principale est connectée au routeur via une ligne SDLC. Ceci est généralement observé dans les environnements hôtes lorsque les utilisateurs souhaitent migrer l'hôte vers une pièce jointe Token Ring. Reverse SDLLC modifie la façon dont DLSw gère la ligne SDLC, car il n'est souvent pas clair si l'unité de données distante est active ou non.



Tout d'abord, parce que l'AS/400 est principal dans ce cas, ou configuré pour être négociable dans le rôle, il doit démarrer la session. Lorsque l'AS/400 envoie le premier XID après que la ligne série soit opérationnelle, le routeur lance le processus de recherche pour le contrôleur distant. Une fois le circuit configuré, la négociation XID peut commencer dans la ligne.

Une fois la négociation XID terminée, l'AS/400 envoie le SNRM au routeur. Cela entraîne l'envoi de la requête CONQ par le routeur et l'attente de la requête CONR du routeur distant. Le routeur ne peut pas répondre avec l'UA tant qu'il n'a pas vu un SNRM et après avoir reçu le CONR. Dans presque toutes les versions du code, le routeur attend 30 secondes jusqu'à ce qu'il expire la session. Il s'agit de la réception des SNRM du périphérique principal une fois que le périphérique principal reçoit la CONR de l'hôte distant.

Dans le dernier code Cisco IOS 11.1, les valeurs par défaut sont passées à une minute au lieu de 30 secondes. Dans l'AS/400, ce délai d'attente est appelé **compteur de réponse non productif** et prend par défaut 32 secondes.

Traduction multimédia DLSw locale



```

%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 première chose que vous remarquez dans DLSw local est le XID du côté série. Ce XID doit être stocké jusqu'à ce que le routeur envoie les trames/réponses de test LLC par le biais de.

```

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

```

Ensuite, la station de test quitte le routeur et la réponse revient de l'AS/400. Maintenant, le routeur peut créer le FSM local.

Remarque : N'oubliez pas qu'il s'agit d'une session 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

```

Une fois que le routeur a confirmé localement que le FSM est prêt, il peut envoyer le XID au partenaire. Dans cet exemple, le partenaire est 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

Ensuite, un XID est reçu de la Token Ring. L'**ID.Ind** a une longueur de 108. Le routeur transfère ce XID au partenaire dans ce scénario, qui est la ligne SDLC. Ceci est indiqué par l'**ID.Req** qui a été envoyé. Chaque fois que le routeur reçoit un paquet, il doit démarrer la machine à état fini linéaire (LFSM). C'est la clé pour comprendre ce débogage, parce qu'il vous informe où il commence et où il va.

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

Ensuite, la réponse XID est reçue de la ligne série et est transmise au partenaire (la station Token Ring dans cet exemple). Cette opération se poursuit jusqu'à ce que l'échange XID soit terminé pour ce périphérique 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

Après l'échange XID, le routeur reçoit un SABME de l'AS/400 via **CONNECT.Ind**. Ceci indique au routeur d'envoyer un **CONNECT.Req** à la ligne SDLC, qui est le SNRM. Ensuite, un message **CONNECT.Cfm** (UA) est reçu de la ligne série, ce qui fait que le code DLSw envoie un **CONNECT.Rsp** (UA) à l'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.Reg dlen: 0
DLSw: END-LFSM (4000.5494.00dd->4000.9404.0001): state:CONN_OUT_PEND->CONNECTED

```

La session lorsque le contrôleur (SDLC) s'arrête s'affiche.

```

%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

```

Ensuite, le routeur envoie un DISQUE à l'AS/400 (DISCONNECT.Rsp). Ensuite, il commence à détruire le circuit local.

```

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.Reg dlen: 4
DLSw: END-LFSM (4000.9404.0001->4000.5494.00dd): state:ESTABLISHED ->CLOSE_PEND

DISP Sent : CLSI Msg : CLOSE_STN.Reg 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

```

Une fois que le routeur a reçu le fichier DISCONNECT.Ind (UA) de l'AS/400, il termine la suppression de la session et passe à un état de déconnexion.

Informations connexes

- [Technologies IBM](#)
- [Support et documentation techniques - Cisco Systems](#)