

DLSw+ (Data Link Switching Plus)

Contenu

[Introduction](#)

[Conditions préalables](#)

[Conditions requises](#)

[Components Used](#)

[Conventions](#)

[Commandes générales](#)

[source-bridge ring-group](#)

[Définir l'identification des homologues locaux](#)

[Définir l'homologue distant](#)

[Minuteurs utilisés dans DLSw](#)

[Commandes DLSw supplémentaires](#)

[Commandes show](#)

[show dlsw peer](#)

[show dlsw capabilities](#)

[show dlsw reachability](#)

[show dlsw circuit](#)

[Dépannage](#)

[Boucles](#)

[Homologues de sauvegarde/coût](#)

[Homologues périphériques](#)

[débogage](#)

[Sessions NetBIOS](#)

[Informations connexes](#)

[Introduction](#)

La commutation de liaisons de données (DLSw) est une norme mise en oeuvre par IBM qui prend en charge le transport du contrôle de liaison logique (LLC) sur les réseaux étendus. DLSw est une forme plus élaborée de pontage source-route distant (RSRB) et est plus spécifique quant à ce qu'il peut ou ne peut pas ponter. DLSw nécessite que le routeur transporte une session LLC2 valide ou une session NetBIOS.

Les routeurs Cisco implémentent les normes RFC 1795 (DSLw) et 2166 (DLSw version 2). En outre, DLSw met en oeuvre plus de fonctions de contrôle de diffusion et transporte moins d'informations sur le WAN que les autres méthodes.

[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](#).

Commandes générales

Cette section traite des commandes DLSw importantes, des commandes de configuration de DLSw et des commandes de dépannage de DLSw.

source-bridge ring-group

La première étape de la configuration de DLSw consiste à ajouter la commande **source-bridge ring-group**. Elle connecte les interfaces Token Ring exécutant le pontage SRB (Source-Route Bridging).

Tâche	Commande
Définissez un groupe de sonneries.	source-bridge ring-group <i>ring-group</i> [adresse-mac-virtuelle]

Remarque : lorsque vous exécutez DLSw sur un routeur qui ne comporte que des interfaces Ethernet, il n'est pas nécessaire de configurer un **groupe d'anneau**.

Définir l'identification des homologues locaux

L'option suivante consiste à définir l'identification de l'homologue local. Il s'agit d'une adresse IP dans la même zone. Cette opération démarre DLSw dans le routeur.

Tâche	Commande
Définissez l'homologue local DLSw+.	dlsw local-peer [peer-id <i>adresse-ip</i>] [group <i>group</i>] [border] [cost <i>costf</i>] [lf size] [keepalive <i>seconds</i>] [passive] [promiscuous] [biu-segment]

L'option la plus basique dans la configuration de DLSw est d'établir l'**adresse IP d'ID d'homologue local**. Voici les descriptions des paramètres de commande :

- **group** and **border** : ces commandes sont exécutées conjointement pour créer des homologues de périphérie dans le réseau.

- **cost** : cette commande est exécutée lorsqu'il existe plusieurs chemins vers le même emplacement. Cette commande indique au routeur comment atteindre ces sites distants en utilisant d'abord le chemin le moins coûteux.
- **if** : cette commande détermine la plus grande taille de trame que cet homologue peut gérer. Les tailles de trame peuvent être : Taille de trame maximale de 516 à 516 octets
Taille de trame maximale : 1 470 à 1 470 octets
Taille de trame maximale de 1 500 à 1 500 octets
Taille de trame maximale de 2 052 à 2 052 octets
4 472 à 4 472 octets de taille de trame maximale
8 144 à 8 144 octets de taille de trame maximale
11407-11407 octets taille de trame maximale
11454-11454 octets taille de trame maximale
17800-17800 octets taille de trame maximale
- **keepalive** : cette commande définit l'intervalle entre les paquets keepalive. L'intervalle peut être compris entre 0 et 1 200 secondes. Il est généralement défini sur 0 lors de la configuration de DLSw pour le routage à établissement de connexion à la demande (DDR).
- **passive** : cette commande configure le routeur pour qu'il ne démarre pas un homologue à partir du routeur.
- **promiscuous** : cette commande signifie que le routeur accepte les connexions de tout homologue distant demandant le démarrage d'un homologue. Cette commande est utile dans les sites de grande taille qui ont de nombreux homologues, car vous n'avez pas besoin de définir tous les homologues distants dans le routeur principal.
- **biu-segment** - Cette commande est une option pour DLSw qui permet à DLSw de contrôler la taille de segment plus élevée dans les couches SNA (System Network Architecture). Cette commande permet aux stations d'extrémité de croire qu'elles peuvent envoyer de plus grandes quantités de données.

Définir l'homologue distant

Après avoir défini l'homologue local, vous définissez l'homologue distant. Vous pouvez définir trois types d'homologues : TCP, FST (Fast-Sequenced Transport), HDLC (High-Level Link Control) et Frame Relay. Voici les explications des commandes émises pour définir l'homologue distant :

Tâche	Commande
Encapsulation directe sur Frame Relay	dlsw remote-peer list-number frame-relay interface numéro de série dci-number [backup-peer ip-address] [bytes-netbios-out bytes-list-name] [cost cost] [dest-mac mac-address] [dmac-output-list access-list-number] [host-netbios-out host-list-name] [keepalive seconds] [if size] [linger minutes] [lsap-output-list list] [pass-thru]
Encapsulation directe sur HDLC	dlsw remote-peer list-number interface numéro de série [backup-peer ip-address] [bytes-netbios-out bytes-list-name] [coût] [dest-mac mac-address] [dmac-output-list access-list-number] [host-netbios-out host-list-name] [keepalive seconds] [if size] [linger minutes] [lsap-output-list list list] [pass-thru]
FST	dlsw remote-peer list-number fst ip-address [backup-peer ip-address] [bytes-netbios-out bytes-

	list-name] [cost cost] [dest-mac mac-address] [dmac-output-list access-list-number] [host-netbios-out host-list-name] [keepalive seconds] [lf size] [linger minutes] [lsap-output-list list list] [pass-thru]
TCP	dls w remote-peer list-number tcp ip-address [backup-peer ip-address] [bytes-netbios-out bytes-list-name] [cost cost] [dest-mac mac-address] [dmac-output-list access-list-number] [dynamic] [host-netbios-out host-list-name] [inactivity minutes] [keepalive seconds] [lf size] [linger minutes] [lsap-output-list list] [no-llc minutes] [priority] [tcp-queue] taille max] [délai d'attente en secondes][v2-single-tcp]

Voici les descriptions des options de commande :

- **backup peer** : cette option de commande définit l'homologue qui sauvegarde cet homologue en cas de défaillance du premier homologue.
- **cost** : cette option de commande définit le coût de cet homologue. Cette commande est utilisée lorsqu'il existe plusieurs chemins vers une destination et lorsque vous avez besoin d'un scénario de préférence.
- **dest-mac**, **dynamic**, **no-llc** et **inactivity** : ces options de commande sont décrites dans la section **Peer de sauvegarde/coût de ce document**.
- **dmac-output-list** — Cette option de commande est émise pour définir une liste d'accès qui indique au routeur quelles adresses MAC de destination distante vous autorisez ou refusez le trafic de l'explorateur.
- **host-netbios-out** — Cette option de commande est émise pour appliquer les noms de filtres hôtes NetBIOS.
- **keepalive** : cette option de commande est émise pour déterminer l'intervalle en secondes entre les keepalives. Il est principalement utilisé pour les configurations DDR.
- **lf** : cette option de commande spécifie la taille maximale autorisée pour l'homologue.
- **linger** : cette option de commande spécifie la durée pendant laquelle le routeur laisse l'homologue de secours ouvert qui devient actif (en raison d'une défaillance principale) après que la liaison principale redevienne active.
- **priority** - Cette option de commande crée plusieurs homologues pour la hiérarchisation du trafic DLSw.
- **tcp-queue-max** — Cette option de commande modifie la valeur par défaut de 200 pour les files d'attente TCP.
- **timeout** : cette option de commande indique le nombre de secondes pendant lesquelles le protocole TCP attend un accusé de réception avant de couper la connexion.
- **V2-single-tcpM** — Cette option de commande est conçue pour une utilisation dans les environnements de traduction d'adresses de réseau (NAT). Chaque homologue pense avoir l'adresse IP la plus élevée pour empêcher chaque homologue d'interrompre une des connexions TCP.

[Minuteurs utilisés dans DLSw](#)

Voici des explications des compteurs utilisés dans DLSw :

Paramètre	Description
icannotreach-block-time	Durée de vie du cache d'une ressource inaccessible, pendant laquelle les recherches de cette ressource sont bloquées. La plage valide est comprise entre 1 et 86400 secondes. La valeur par défaut est 0 (désactivé)
netbios-cache-timeout	Durée de vie du cache de l'emplacement du nom NetBIOS pour le cache d'accessibilité local et distant. La plage valide est comprise entre 1 et 86400 secondes. La valeur par défaut est 16 minutes.
netbios-explorer-timeout	Durée pendant laquelle le logiciel IOS® attend une réponse de l'explorateur avant de marquer une ressource inaccessible (LAN et WAN). La plage valide est comprise entre 1 et 86400 secondes. 6 secondes sont établies par défaut.
netbios-retry-interval	Intervalle entre deux tentatives de l'explorateur NetBIOS (LAN uniquement). La plage valide est comprise entre 1 et 86400 secondes. La valeur par défaut est 1 seconde.
netbios-verify-interval	Intervalle entre la création d'une entrée de cache et le moment où l'entrée est marquée comme obsolète. Si une requête de recherche arrive pour une entrée de cache périmée, une requête de vérification dirigée est envoyée pour s'assurer qu'elle existe toujours. La plage valide est comprise entre 1 et 86400 secondes. La valeur par défaut est 4 minutes.
sna-cache-timeout	Durée pendant laquelle une entrée de cache d'emplacement SNA MAC/SAP (Service Access Point) existe avant d'être supprimée (locale et distante). La plage valide est comprise entre 1 et 86400 secondes. La valeur par défaut est 16 minutes.
sna-explorer-timeout	Durée pendant laquelle le logiciel IOS attend une réponse de l'explorateur avant de marquer une ressource inaccessible (LAN et WAN). La plage valide est comprise entre 1 et 86400 secondes. La valeur par défaut est 3 minutes.
sna-retry-interval	Intervalle entre les tentatives d'exploration SNA (LAN). La plage valide est comprise entre 1 et 86400 secondes. 30 secondes sont établies par défaut.
sna-verify-interval	Intervalle entre la création d'une entrée de cache et le moment où l'entrée est marquée

	comme obsolète. Si une requête de recherche arrive pour une entrée de cache périmée, une requête de vérification dirigée est envoyée pour s'assurer qu'elle existe toujours. La plage valide est comprise entre 1 et 86400 secondes. La valeur par défaut est 4 minutes.
explorer-wait-time	Durée, en secondes, pendant laquelle le routeur attend le retour de tous les explorateurs avant de déterminer quel homologue utiliser.

Ces paramètres sont très utiles. Par exemple, vous pouvez modifier l'intervalle en secondes pendant lequel le routeur envoie un explorateur. Cela permet de réduire le nombre d'explorateurs sur le réseau en augmentant le temps entre eux. Vous pouvez également modifier les valeurs auxquelles le routeur expire les entrées du cache.

Commandes DLSw supplémentaires

Voici d'autres commandes DLSw importantes :

- **dlsw allroute-sna/netbios** : cette commande est exécutée pour modifier le comportement de DLSw afin que tous les explorateurs de route soient utilisés à la place des explorateurs de route unique.
- **dlsw bridge-group** — Cette commande est exécutée pour lier de manière transparente les domaines pontés avec DLSw. Il est largement utilisé lors de la configuration de NetBIOS avec Ethernet.
- **dlsw explorerq-depth** — Cette commande définit la valeur de la file d'attente de l'explorateur DLSw. Cette commande est émise après la commande normale **source-bridge explorer-queue**, mais elle fait référence à toutes les trames CANUREACH (CUR) qui doivent être traitées. Cette commande est importante parce qu'elle couvre les paquets de l'Ethernet, même si elle n'est pas couverte dans la commande **source-bridge explorerq-depth**. Référez-vous à [Compréhension et dépannage du pontage Source-Route](#) pour plus d'informations sur cette commande.

Commandes show

Les commandes **show** et les résultats décrits dans cette section sont utiles lors du dépannage de DLSw.

show dlsw peer

Cette commande fournit des informations sur les homologues. Chaque homologue distant configuré est affiché ici, y compris la quantité de paquets transmis et reçus.

```
Peers:                state      pkts_rx  pkts_tx  type  drops  ckts  TCP  uptime
TCP 5.5.5.1           CONNECT   2         2    conf    0     0    0  00:00:06
```

Voici les états possibles :

- **CONNECT** : cet état signifie que l'homologue DLSw est actif et en cours d'exécution.
- **DISCONNECT** : cet état signifie que l'homologue est hors service ou n'est pas connecté.
- **CAP_EXG** : cet état signifie que DLSw est en échange de capacités avec l'homologue distant.
- **WAIT_RD** : cet état est la dernière étape du démarrage de l'homologue. Cet homologue attend que l'homologue distant ouvre le port de lecture. Référez-vous à la section [débogage](#) de ce document pour plus d'informations sur quand l'homologue démarre et émet la commande **debug dlsw peer**.
- **WAN_BUSY** : cet état signifie que la file d'attente TCP sortante est pleine et que le paquet ne peut pas être transmis.

La commande **show dlsw peer** montre également le nombre de pertes, la quantité de circuits sur l'homologue spécifique, la file d'attente TCP et le temps de fonctionnement. Le compteur d'abandon augmente pour les raisons suivantes :

- L'interface WAN n'est pas active pour un homologue direct.
- DLSw tente d'envoyer un paquet avant que l'homologue ne soit entièrement connecté (en attente d'un événement TCP ou d'un événement de capacités). File d'attente TCP sortante saturée.
- Non-concordance du nombre de numéros de séquence FST.
- Impossible d'obtenir le tampon pour ralentir le paquet FST du commutateur.
- Défaillance du contrôleur CiscoBus sur le haut de gamme ; impossible de déplacer le paquet du tampon de réception au tampon de transmission, ou vice versa.
- L'adresse IP de destination du paquet FST ne correspond pas à l'ID d'homologue local.
- L'interface WAN n'est pas active pour un homologue FST.
- Aucune commande SRB route cache configurée.
- La mémoire tampon en anneau Madge est pleine sur les systèmes bas de gamme : Le WAN alimente le LAN trop rapidement.

[show dlsw capabilities](#)

```
DLSw: Capabilities for peer 5.5.5.1(2065)
 vendor id (OUI)           : '00C' (cisco)
 version number            : 1
 release number            : 0
 init pacing window        : 20
 unsupported saps           : none
 num of tcp sessions        : 1
 loop prevent support       : no
 icanreach mac-exclusive   : no
 icanreach netbios-excl.   : no
 reachable mac addresses    : none
 reachable netbios names    : none
 cisco version number      : 1
 peer group number         : 0
 border peer capable        : no
 peer cost                  : 3
 biu-segment configured    : no
 local-ack configured       : yes
 priority configured        : no
 version string             :
Cisco Internetwork Operating System Software
```

show dlsw reachability

```
DLSw MAC address reachability cache list
Mac Addr      status      Loc.      peer/port      rif
0800.5a0a.c51d FOUND      LOCAL    TokenRing3/0  06B0.0021.00F0
0800.5a49.1e38 FOUND      LOCAL    TokenRing3/0  06B0.0021.00F0
0800.5a95.3a13 FOUND      REMOTE   5.5.5.1(2065)
```

```
DLSw NetBIOS Name reachability cache list
NetBIOS Name  status      Loc.      peer/port      rif
PIN-PIN       FOUND      LOCAL    TokenRing3/0  06B0.0021.00F0
QUENEPA      FOUND      LOCAL    TokenRing3/0  06B0.0021.00F0
WIN95        FOUND      REMOTE   5.5.5.1(2065)
```

Le champ **status** est la partie la plus importante de la commande **show dlsw reach**. Voici les états possibles :

- **TROUVÉ** : le routeur a localisé le périphérique.
- **SEARCHING** : le routeur recherche la ressource.
- **NOT_FOUND** : la mise en cache négative est activée et la station n'a pas répondu aux requêtes.
- **UNCONFIRMED** : la station est configurée, mais DLSw ne l'a pas vérifiée.
- **VERIFY** : vérification des informations du cache en raison de son obsolescence ou de la vérification de la configuration utilisateur.

show dlsw circuit

```
Index          local addr(lsap)  remote addr(dsap)  state
1622193728    4001.68ff.0001(04) 4000.0000.0001(04) CONNECTED
PCEP: 60A545B4  UCEP: 60B0B640
Port:To3/0      peer 5.5.5.1(2065)
Flow-Control-Tx CW:20, Permitted:32; Rx CW:20, Granted:32
RIF = 06B0.0021.00F0
```

Lorsque vous émettez la commande **show dlsw circuit**, faites attention au contrôle de flux. Le contrôle de flux est effectué circuit par circuit. Il s'agit d'une communication qui se produit pendant que les deux homologues DLSw attribuent au circuit une fenêtre de transfert possible. Cette valeur augmente et diminue en fonction de la quantité de trafic que le circuit tente de traverser. La valeur peut varier en fonction de l'encombrement du cloud.

La commande **show dlsw circuit** est plus étendue depuis IOS 11.1. La commande vous permet maintenant d'examiner le circuit DLSw sur une valeur de point d'accès de service (SAP) ou une valeur MAC, ce qui simplifie la localisation des circuits lors du dépannage. Voici est un exemple de sortie :

```

Index          local addr(lsap)    remote addr(dsap)  state
1622193728    4001.68ff.0001(04) 4000.0000.0001(04) CONNECTED
ibu-7206#sh dls cir det ?
<0-4294967295> Circuit ID for a specific remote circuit
mac-address    Display all remote circuits using a specific MAC
sap-value     Display all remote circuits using a specific SAP
<cr>

```

```

ibu-7206#show dls circuit detail mac 4000.0000.0001
Index          local addr(lsap)    remote addr(dsap)  state
1622193728    4001.68ff.0001(04) 4000.0000.0001(04) CONNECTED
PCEP: 60A545B4  UCEP: 60B0B640
Port:To3/0      peer 5.5.5.1(2065)
Flow-Control-Tx CW:20, Permitted:29; Rx CW:20, Granted:29
RIF = 06B0.0021.00F0
241-00 4000.0000.0001(04) 4001.68ff.0000(04) CONNECTED
Port:To0        peer 5.5.7.1(2065)
Flow-Control-Tx CW:20, Permitted:27; Rx CW:20, Granted:27
RIF = 0630.00F1.0010

```

```

s5e#sh cls
DLU user: DLSWDLU
  SSap:0x63      type: llc0    class:0
  DTE:0800.5a95.3a13 0800.5a0a.c51d F0 F0
  T1 timer:0      T2 timer:0      Inact timer:0
  max out:0       max in:0         retry count:0
  XID retry:0     XID timer:0     I-Frame:0

  DTE:4000.0000.0001 4001.68ff.0000 04 04
  T1 timer:0      T2 timer:0      Inact timer:0
  max out:0       max in:0         retry count:0
  XID retry:0     XID timer:0     I-Frame:0
TokenRing0 DTE: 4000.0000.0001 4001.68ff.0000 04 04 state NORMAL
  V(S)=23, V(R)=23, Last N(R)=22, Local window=7, Remote Window=127
  akmax=3, n2=8, Next timer in 1240
  xid-retry timer 0/0      ack timer 1240/1000
  p timer          0/1000   idle timer 10224/10000
  rej timer        0/3200   busy timer 0/9600
  akdelay timer    0/100    txQ count 0/200

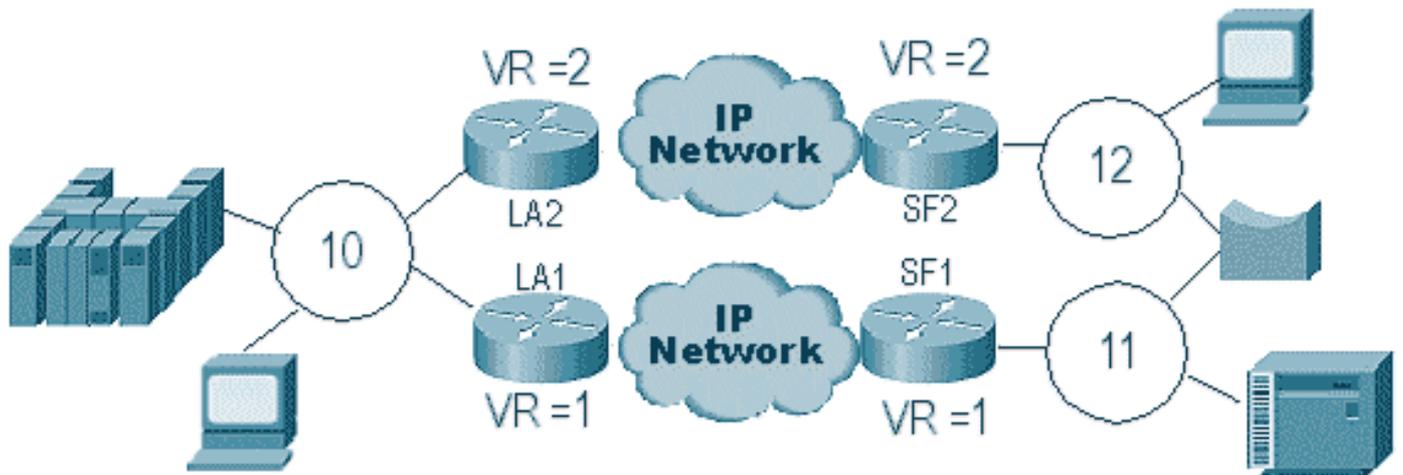
```

Dépannage

Par défaut, DLSw met fin aux sessions LLC sur les routeurs (local-ack). En outre, comme il termine le champ d'informations de routage (RIF), d'autres problèmes de conception doivent être pris en compte. Les problèmes DLSw les plus courants sont décrits dans cette section.

Boucles

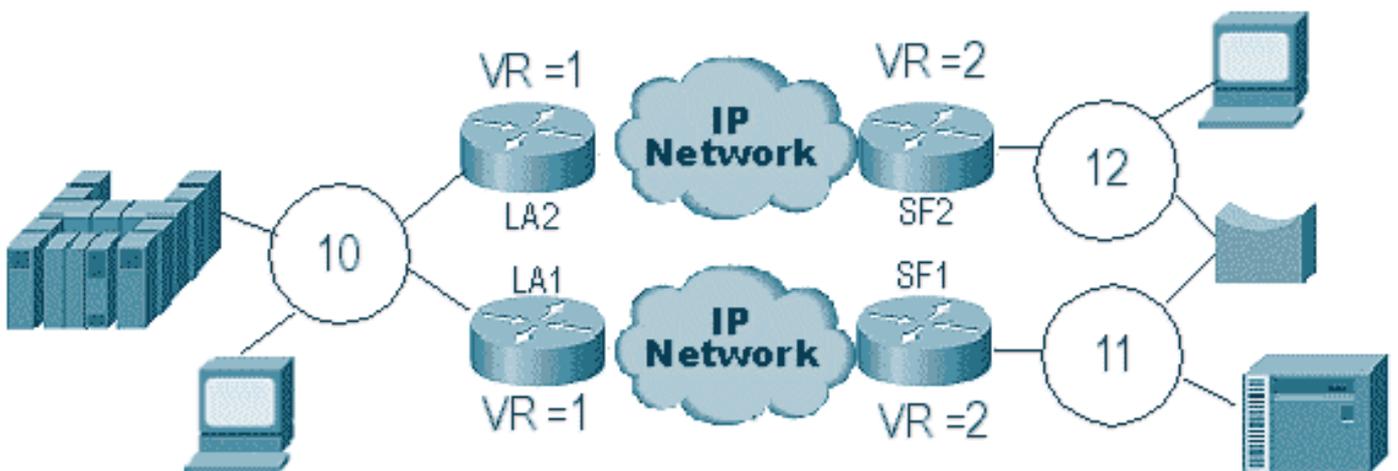
L'une des choses les plus importantes à retenir au sujet de DLSw est la terminaison RIF. Il s'agit d'un problème car il est facile de créer des boucles importantes dans le réseau. Ce schéma illustre une boucle :



Dans ce cas, puisque DLSw termine le RIF, le paquet circule indéfiniment. En effet, chaque fois qu'une trame CUR est envoyée d'un homologue à l'autre, l'homologue récepteur crée un nouvel explorateur (NO RIF) et l'envoie. Les étapes de l'explorateur sont décrites :

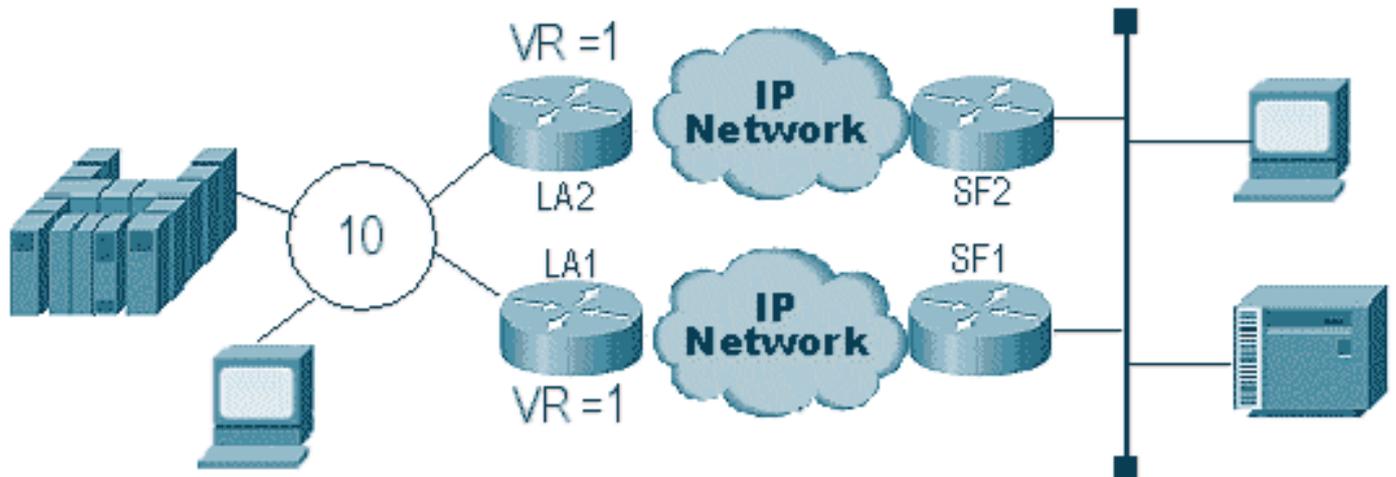
1. Le 3174 de l'anneau 11 envoie un explorateur pour atteindre l'hôte.
2. SF1 et le pont copient la trame.
3. SF1 crée une trame CUR vers LA1 (son homologue) pour indiquer à LA1 que le routeur 3174 veut atteindre l'hôte.
4. SF2 reçoit le paquet et fait la même chose.
5. Maintenant, LA1 et LA2 créent l'explorateur et l'envoient à l'anneau.
6. LA1 et LA2 reçoivent un explorateur que l'un et l'autre ont créé.
7. Maintenant, il y a un dilemme, parce que chaque côté croit que le 3174 est localement attaché.
8. Chaque routeur dispose du 3174, local et distant.
9. Maintenant, ils envoient une trame Icanreach à SF1 et SF2, respectivement, ce qui crée une réponse de l'hôte vers le 3174.
10. SF1 et SF2 placent la réponse de l'explorateur sur le Token Ring et chacun apprend que l'adresse MAC de l'hôte est accessible localement et à distance.
11. L'accessibilité DLSw pare-feu efficacement contre l'explorateur en boucle indéfinie. Cependant, avec les trames d'informations non numérotées (UI), cela peut créer une boucle, puis augmenter l'utilisation du CPU et de la ligne jusqu'à 100 %.

Dans ce cas, vérifiez que l'anneau virtuel des routeurs est exactement le même de chaque côté du cloud, comme illustré dans le schéma suivant :



Les routeurs de chaque côté de ce cloud ont exactement le même numéro de sonnerie virtuelle. Ainsi, l'un des routeurs envoie un explorateur qui a déjà traversé l'anneau, puis le routeur l'abandonne. Lorsque LA1 génère un explorateur pour une trame CUR reçue par SF1, LA2 l'abandonne parce que l'explorateur a déjà traversé l'anneau 1. Dans ce scénario, il est important que le routeur ait un pont différent configuré si le paquet est dirigé vers le même anneau, ce qui est le cas du côté L.A. du réseau.

Dans une version Ethernet du même scénario, vous devez désactiver un homologue. Un exemple est affiché dans ce diagramme :



Étant donné qu'un paquet sur Ethernet ne possède pas de RIF, le routeur ne peut pas déterminer si la diffusion, créée par l'autre routeur sur le LAN, provient de l'autre routeur ou d'une station d'origine. Avec l'architecture SNA, le paquet provient localement ou est distant. Comme les explorateurs d'un environnement Token Ring possèdent en fait à la fois des adresses MAC source et de destination, ils ne constituent pas une diffusion sur Ethernet, mais une trame dirigée vers une station à partir d'une autre.

Ce qui se produit dans le schéma précédent est expliqué dans ces étapes :

1. Un explorateur est envoyé du 3174 à l'hôte.
2. Cet explorateur est accepté par SF1 et SF2.
3. SF1 et SF2 génèrent chacun une CUR de l'autre côté LA1 et LA2.
4. Ils génèrent un explorateur auquel l'hôte répond ; étant donné qu'il s'agit d'un explorateur de route unique, il reçoit une réponse avec un explorateur de route.
5. La1 et LA2 créent une trame CUR vers SF1 et SF2, qui créent le paquet pour le 3174.
6. SF1 entend l'adresse MAC de l'HÔTE provenant de l'Ethernet et croit maintenant que l'HÔTE se trouve sur le LAN local. Mais dans le cache de SF1, l'ID d'HÔTE répond d'un homologue distant.
7. Cela force le routeur à avoir l'HÔTE à la fois local et distant, ce qui signifie que DLSw est cassé.

[Homologues de sauvegarde/coût](#)

Les homologues de sauvegarde ajoutent une tolérance aux pannes à DLSw en cas de perte d'un homologue. Cette option est généralement configurée dans les environnements principaux de sorte que lorsqu'un routeur principal tombe en panne, un autre routeur peut accepter le routeur

défaillant. Les configurations et le schéma de cette section illustrent une configuration homologue de sauvegarde.

D3B

```
Current configuration:
!
version 11.1
service udp-small-servers
service tcp-small-servers
!
hostname d3b
!
!
source-bridge ring-group 2
dlsw local-peer peer-id 1.1.14.1
    cost 2 promiscuous
!
interface Loopback0
    ip address 1.1.14.1 255.255.255.0
!
interface Serial0
    ip address 1.1.6.2 255.255.255.0
    bandwidth 125000
    clockrate 125000
!
interface TokenRing0
    ip address 1.1.5.1 255.255.255.0
    ring-speed 16
    source-bridge 3 1 2
    source-bridge spanning
!
```

D3C

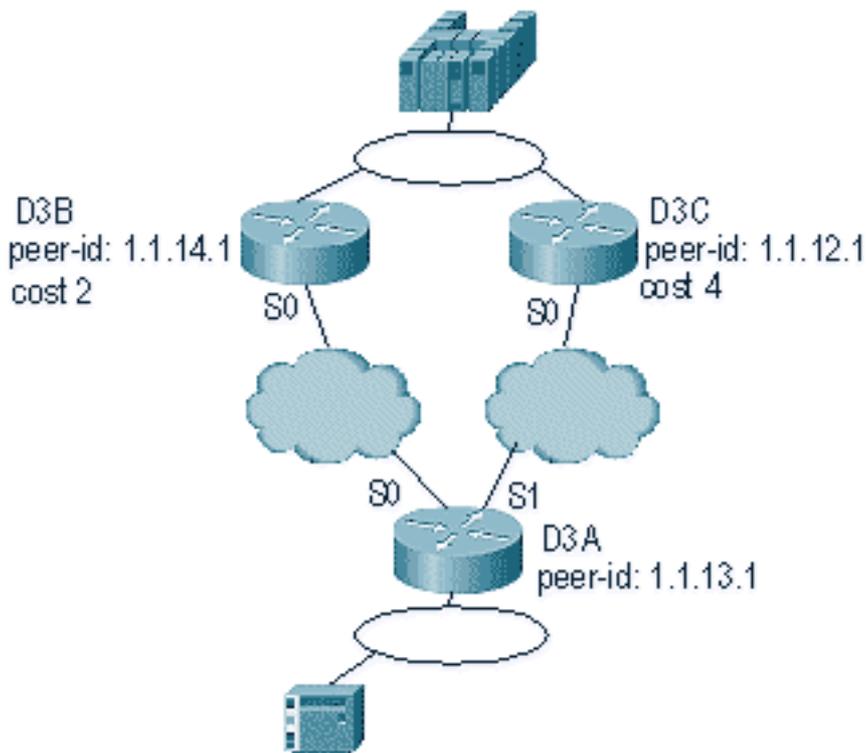
```
Current configuration:
!
version 11.1
service udp-small-servers
service tcp-small-servers
!
hostname d3c
!
!
source-bridge ring-group 2
dlsw local-peer peer-id 1.1.12.1
    cost 4 promiscuous
!
interface Loopback0
    ip address 1.1.12.1 255.255.255.0
!
interface Serial0
    ip address 1.1.4.1 255.255.255.0
    bandwidth 500000
    clockrate 500000
!
interface TokenRing0
    ip address 1.1.5.2 255.255.255.0
    ring-speed 16
```

```
source-bridge 3 2 2
source-bridge spanning
!
```

D3A

Current configuration:

```
!
version 11.1
service udp-small-servers
service tcp-small-servers
!
hostname d3a
!
!
source-bridge ring-group 2
dlsw local-peer peer-id 1.1.13.1
dlsw remote-peer 0 tcp 1.1.14.1
dlsw remote-peer 0 tcp 1.1.12.1
dlsw timer explorer-wait-time 2
!
interface Loopback0
 ip address 1.1.13.1 255.255.255.0
!
interface Serial0
 ip address 1.1.6.1 255.255.255.0
 bandwidth 500000
!
interface Serial1
 ip address 1.1.4.2 255.255.255.0
 bandwidth 125000
!
interface TokenRing0
 ip address 1.1.1.1 255.255.255.0
 ring-speed 16
 source-bridge 3 1 2
 source-bridge spanning
!
```



La première chose à retenir à propos des homologues de coût DLSw est que les deux homologues sont actifs. Le routeur ne gère qu'un homologue de secours. Il peut en avoir deux à la fois si **linger** est configuré. Voici ce qui s'est produit dans le schéma précédent :

1. D3a reçoit un explorateur et lance le processus en envoyant une trame CUR à chaque homologue distant.
2. D3B et D3C reçoivent les trames CUR. Chacun génère un explorateur vers l'hôte, qui répond à D3B et D3C.
3. D3B et D3C répondent tous deux à D3A avec lcanreach.
4. D3A envoie la réponse de l'explorateur à la station d'extrémité.
5. La station distante démarre le circuit dlsw, avec l'identification d'échange (XID) pour SNA et le mode équilibré étendu asynchrone (SABME) pour NetBIOS.
6. D3A sélectionne un coût inférieur dans l'accessibilité.

Un minuteur dans D3A peut être défini pour indiquer au routeur combien de temps il faut attendre que tous les explorateurs reviennent à D3A. Cela évite les problèmes de coûts qui peuvent survenir lorsque le routeur utilise le premier explorateur qui lui revient. Émettez la commande **dlsw timer explorer-wait-time <seconds>** pour définir ce minuteur.

En outre, lors de l'exécution d'**homologues en limite**, DLSw envoie une seule trame CUR à l'homologue de moindre coût. Il se comporte différemment de ce qu'il fait en effectuant des coûts sans homologues en limite.

Les homologues de sauvegarde fonctionnent un peu différemment. Vous spécifiez l'homologue de sauvegarde dans l'homologue qui sera sauvegardé pour l'homologue spécifié. Cela signifie que l'homologue qui a l'instruction de sauvegarde est l'homologue de sauvegarde lui-même.

Spécifiez l'option **linger** de sorte que lorsque l'homologue principal redevient opérationnel, les circuits ne puissent pas se démonter immédiatement. Cela est utile si l'homologue principal varie de haut en bas, car vous ne voulez pas utiliser l'homologue défectueux.

Ceci démontre la configuration des homologues de sauvegarde :

D3B

Current configuration:

```
!  
version 11.1  
service udp-small-servers  
service tcp-small-servers  
!  
hostname d3b  
!  
!  
source-bridge ring-group 2  
dlsw local-peer peer-id 1.1.14.1  
    promiscuous  
!  
interface Loopback0  
    ip address 1.1.14.1 255.255.255.0  
!  
interface Serial0  
    ip address 1.1.6.2 255.255.255.0  
    bandwidth 125000  
    clockrate 125000  
!  
interface TokenRing0  
    ip address 1.1.5.1 255.255.255.0  
    ring-speed 16  
    source-bridge 3 1 2  
    source-bridge spanning  
!
```

D3C

Current configuration:

```
!  
version 11.1  
service udp-small-servers  
service tcp-small-servers  
!  
hostname d3c  
!  
!  
source-bridge ring-group 2  
dlsw local-peer peer-id 1.1.12.1  
    promiscuous  
!  
interface Loopback0  
    ip address 1.1.12.1 255.255.255.0  
!  
interface Serial0  
    ip address 1.1.4.1 255.255.255.0  
    bandwidth 500000  
    clockrate 500000  
!  
interface TokenRing0  
    ip address 1.1.5.2 255.255.255.0  
    ring-speed 16  
    source-bridge 3 2 2  
    source-bridge spanning  
!
```

D3A

```
Current configuration:
!
version 11.1
service udp-small-servers
service tcp-small-servers
!
hostname d3a
!
!
source-bridge ring-group 2
dlsw local-peer peer-id 1.1.13.1
dlsw remote-peer 0 tcp 1.1.14.1
dlsw remote-peer 0 tcp 1.1.12.1 backup-peer 1.1.14.1
linger 5
dlsw timer explorer-wait-time 2
!
interface Loopback0
 ip address 1.1.13.1 255.255.255.0
!
interface Serial0
 ip address 1.1.6.1 255.255.255.0
 bandwidth 500000
!
interface Serial1
 ip address 1.1.4.2 255.255.255.0
 bandwidth 125000
!
interface TokenRing0
 ip address 1.1.1.1 255.255.255.0
 ring-speed 16
 source-bridge 3 1 2
 source-bridge spanning
!
```

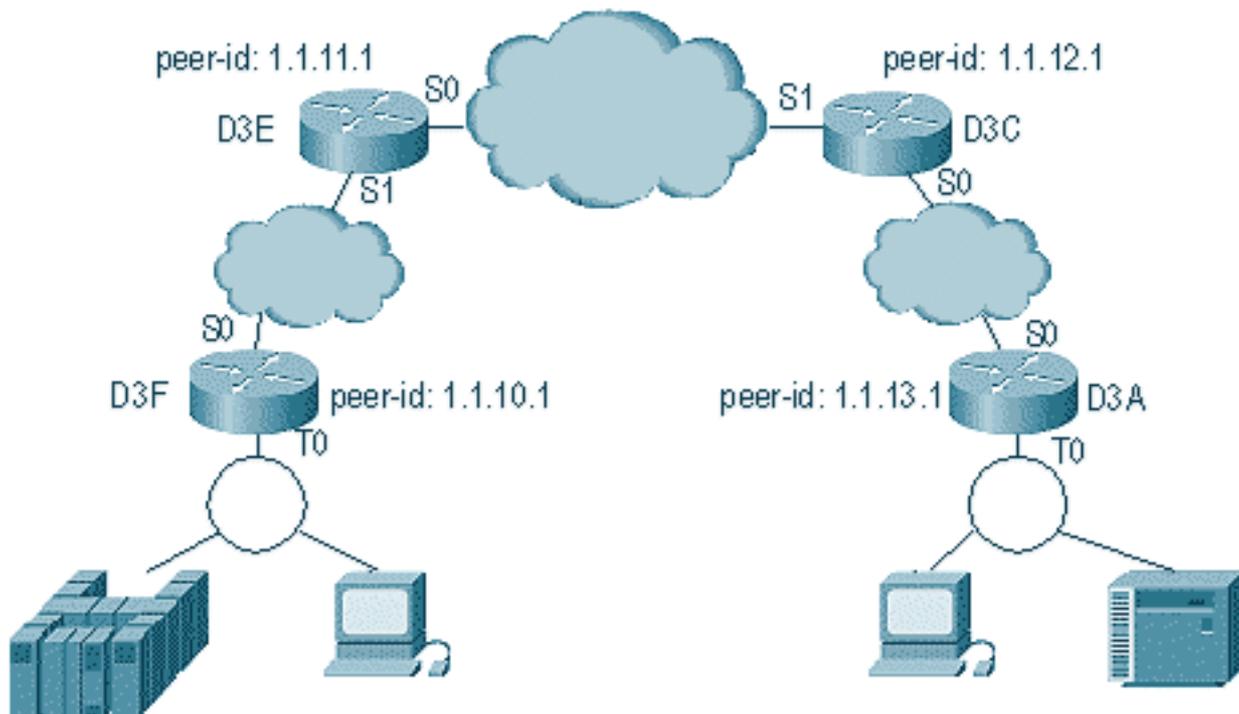
L'homologue est déconnecté en émettant la commande **show dlsw peer** :

```
d3a#sh dls peer
```

Peers:	state	pkts_rx	pkts_tx	type	drops	ckts	TCP	uptime
TCP 1.1.14.1	CONNECT	464	1286	conf	0	0	0	03:17:02
TCP 1.1.12.1	DISCONN	0	0	conf	0	0	-	-

Homologues périphériques

Les homologues en limite sont une fonctionnalité DLSw importante, car ils résolvent le problème du contrôle de diffusion dans un réseau. Cet exemple illustre la configuration des homologues en limite et ce qui se produit lorsqu'une session s'ouvre :



D3E

Current configuration:

```

!
version 11.1
service udp-small-servers
service tcp-small-servers
!
hostname d3e
!
!
dlsw local-peer peer-id 1.1.11.1 group 1
border promiscuous
dlsw remote-peer 0 tcp 1.1.12.1
!
interface Loopback0
ip address 1.1.11.1 255.255.255.0
!
interface Serial0
ip address 1.1.3.1 255.255.255.0
!
interface Serial1
ip address 1.1.2.2 255.255.255.0
clockrate 500000
!
interface TokenRing0
ip address 10.17.1.189 255.255.255.0
ring-speed 16
!
router ospf 100
network 1.0.0.0 0.255.255.255 area 0
!

```

D3C

Current configuration:

```
!  
version 11.1  
  
service udp-small-servers  
service tcp-small-servers  
!  
hostname d3c  
!  
!  
dlsw local-peer peer-id 1.1.12.1 group 2  
    border promiscuous  
dlsw remote-peer 0 tcp 1.1.11.1  
!  
interface Loopback0  
    ip address 1.1.12.1 255.255.255.0  
!  
interface Serial0  
    ip address 1.1.4.1 255.255.255.0  
    no fair-queue  
    clockrate 500000  
!  
interface Serial1  
    ip address 1.1.3.2 255.255.255.0  
    clockrate 500000  
!  
interface TokenRing0  
    no ip address  
    shutdown  
    ring-speed 16  
!  
router ospf 100  
    network 1.0.0.0 0.255.255.255 area 0  
!
```

D3F

Current configuration:

```
!  
version 11.1  
service udp-small-servers  
service tcp-small-servers  
!  
hostname d3f  
!  
!  
source-bridge ring-group 2  
dlsw local-peer peer-id 1.1.10.1 group 1  
    promiscuous  
dlsw remote-peer 0 tcp 1.1.11.1  
dlsw peer-on-demand-defaults inactivity 1  
!  
interface Loopback0  
    ip address 1.1.10.1 255.255.255.0  
!  
interface Serial0  
    ip address 1.1.2.1 255.255.255.0  
    no fair-queue  
!!  
interface TokenRing0  
    ip address 1.1.1.1 255.255.255.0  
    ring-speed 16  
    source-bridge 1 1 2
```

```
source-bridge spanning
!
router ospf 100
 network 1.0.0.0 0.255.255.255 area 0
```

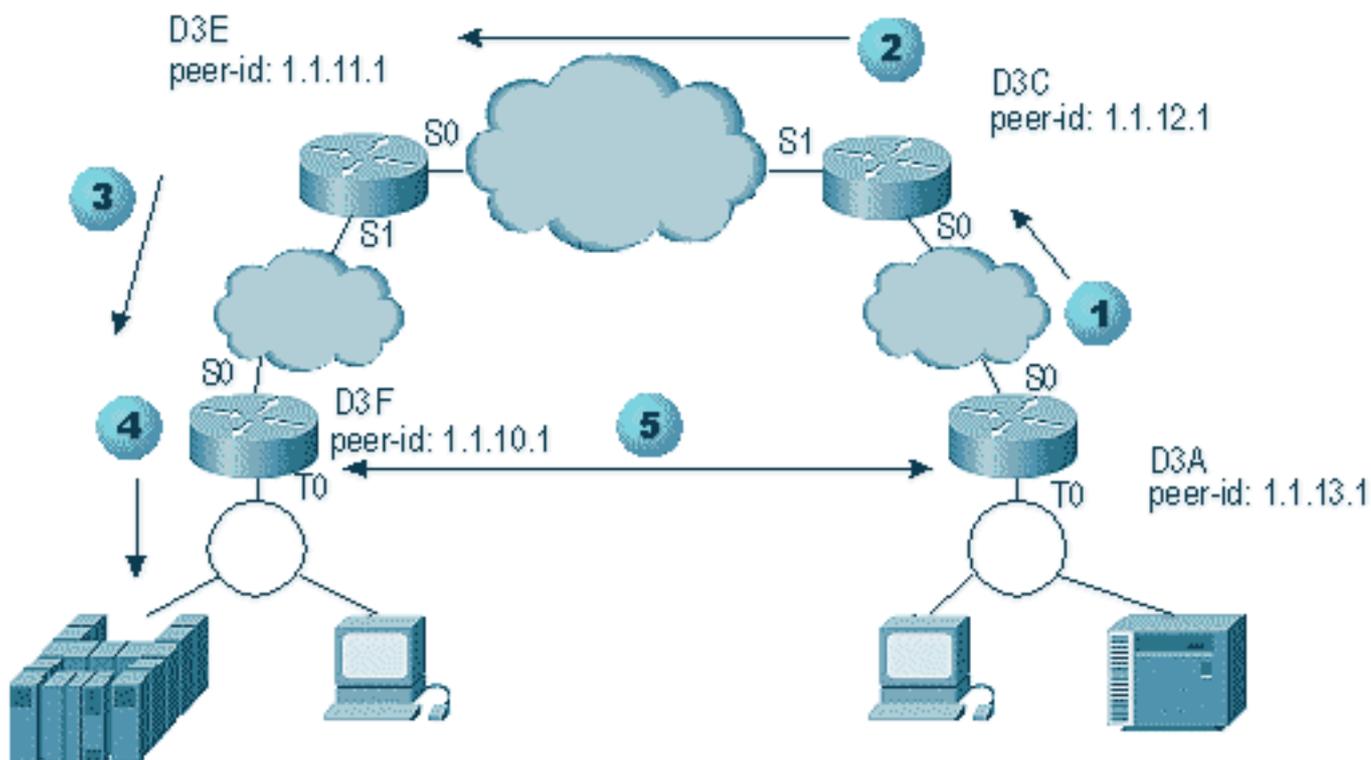
D3A

Current configuration:

```
!
version 11.1
service udp-small-servers
service tcp-small-servers
!
hostname d3a
!
!
source-bridge ring-group 2
dlsw local-peer peer-id 1.1.13.1 group 2
 promiscuous
dlsw remote-peer 0 tcp 1.1.12.1
dlsw peer-on-demand-defaults inactivity 1
!
interface Loopback0
 ip address 1.1.13.1 255.255.255.0
!
interface Serial0
 ip address 1.1.4.2 255.255.255.0
!
interface TokenRing0
 ip address 1.1.5.1 255.255.255.0
 ring-speed 16
 source-bridge 3 1 2
 source-bridge spanning
!
router ospf 100
 network 1.0.0.0 0.255.255.255 area 0
!
```

La première partie de la configuration des homologues périphériques consiste à créer des homologues proches. Les homologues proches acceptent les connexions de tout routeur DLSw qui tente d'ouvrir un homologue avec ce routeur. Par exemple, dans le schéma précédent, vous voulez que D3A ouvre un homologue avec D3F. S'il n'y a pas d'homologues en limite, vous devez configurer des homologues statiques dans le réseau. Cela fonctionne bien, mais lorsque vous avez des centaines de sites et que vous utilisez des homologues statiques lorsqu'un routeur doit trouver une station à distance, le routeur doit envoyer une trame CUR à chaque homologue. Cela peut entraîner une surcharge importante.

D'autre part, lorsque vous utilisez des homologues périphériques, ce routeur distant n'a besoin d'envoyer qu'une seule requête à l'homologue périphérique. Cette requête est ensuite propagée à travers les groupes, et le routeur distant ouvre un homologue avec l'autre routeur distant pour démarrer un circuit et établir une connexion. Ce processus est expliqué dans ce schéma :



1. Lorsque D3A reçoit l'explorateur, il envoie une diffusion à D3C. D3C est l'homologue de bordure auquel D3A est attaché.
2. Lorsque D3C reçoit la trame CUR, il envoie la trame CUR à tous les homologues du groupe. D3C envoie également une trame de test à toutes les interfaces locales qui sont configurées pour ce faire, et envoie une trame CUR aux homologues périphériques de l'autre groupe.
3. D3E reçoit le CUR de D3C dans un autre groupe. Ensuite, D3E fait la même chose en envoyant le CUR à tous les homologues du groupe et à toutes les interfaces locales.
4. D3F reçoit la trame CUR et envoie un sondage test à l'interface locale. Si D3F a un homologue pointant vers un autre routeur, il ne peut pas transmettre cette trame CUR à un autre routeur.
5. Lorsque le D3F reçoit une réponse pour la station d'extrémité, il renvoie la trame lcanreach à D3E.
6. D3E l'envoie à D3C, qui le transmet à D3A. D3A envoie une réponse de test au périphérique.
7. Lorsque la station d'extrémité démarre un circuit dlsw, avec XID pour SNA et SABME pour NetBIOS, D3A initie une connexion homologue avec D3F et démarre la session.

Voici le débogage à la fois de D3C et D3A au cours de ce processus :

```
d3a#
DLSw Received-ctlQ : CLSI Msg : TEST_STN.Ind  dlen: 40
CSM: Received CLSI Msg : TEST_STN.Ind  dlen: 40 from TokenRing0
CSM:  smac c001.68ff.0000, dmac 4000.0000.0001, ssap 4 , dsap 0
DLSw: sending bcast to BP peer 1.1.12.1(2065)
```

La trame de test qui entre dans le routeur est visible. Ensuite, le routeur génère une trame CUR vers D3C. L'activité D3C affiche ce résultat :

```
DLSw: Pak from peer 1.1.13.1(2065) with op DLX_MEMBER_TO_BP
DLSw: rcv_member_to_border() from peer 1.1.13.1(2065)
```

```
DLSw: passing pak to core originally from 1.1.13.1 in group 2
%DLSWC-3-RECVSSP: SSP OP = 3( CUR ) -explorer from peer 1.1.13.1(2065)
DLSw: Pak from peer 1.1.11.1(2065) with op DLX_RELAY_RSP
DLSW: relaying pak to member 1.1.13.1 in group 2
```

Lorsque D3C reçoit le paquet de D3A, il le transfère au coeur. Plus tard, vous voyez la réponse de l'homologue distant qui est relayée vers D3A. Ensuite, D3A démarre la connexion (homologue à la demande) avec l'homologue distant D3F dans ce débogage :

```
DLSw: Pak from peer 1.1.12.1(2065) with op DLX_RELAY_RSP
DLSW: creating a peer-on-demand for 1.1.10.1
DLSw: passing pak to core originally from 1.1.10.1 in group 1
%DLSWC-3-RECVSSP: SSP OP = 4( ICR ) -explorer from peer 1.1.10.1(2065)
DISP Sent : CLSI Msg : TEST_STN.Rsp dlen: 44
DLSW Received-ctlQ : CLSI Msg : ID_STN.Ind dlen: 54
CSM: Received CLSI Msg : ID_STN.Ind dlen: 54 from TokenRing0
CSM: smac c001.68ff.0000, dmac 4000.0000.0001, ssap 4 , dsap 4
DLSw: new_ckt_from_clsi(): TokenRing0 4001.68ff.0000:4->4000.0000.0001:4
DLSw: action_a() attempting to connect peer 1.1.10.1(2065)
DLSw: action_a(): Write pipe opened for peer 1.1.10.1(2065)
DLSw: peer 1.1.10.1(2065), old state DISCONN, new state WAIT_RD
DLSw: passive open 1.1.10.1(11003) -> 2065
DLSw: action_c(): for peer 1.1.10.1(2065)
DLSw: peer 1.1.10.1(2065), old state WAIT_RD, new state CAP_EXG
DLSw: CapExId Msg sent to peer 1.1.10.1(2065)
DLSw: Recv CapExId Msg from peer 1.1.10.1(2065)
DLSw: Pos CapExResp sent to peer 1.1.10.1(2065)
DLSw: action_e(): for peer 1.1.10.1(2065)
DLSw: Recv CapExPosRsp Msg from peer 1.1.10.1(2065)
DLSw: action_e(): for peer 1.1.10.1(2065)
DLSw: peer 1.1.10.1(2065), old state CAP_EXG, new state CONNECT
DLSw: peer_act_on_capabilities() for peer 1.1.10.1(2065)
DLSw: action_f(): for peer 1.1.10.1(2065)
DLSw: closing read pipe tcp connection for peer 1.1.10.1(2065)
DLSw: new_ckt_from_clsi(): TokenRing0 4001.68ff.0000:4->4000.0000.0001:4
DLSw: START-FSM (1474380): event:DLC-Id state:DISCONNECTED
DLSw: core: dlsw_action_a()
DISP Sent : CLSI Msg : REQ_OPNSTN.Req dlen: 106
DLSw: END-FSM (1474380): state:DISCONNECTED->LOCAL_RESOLVE
```

Une fois que le routeur a reçu le paquet relayé de l'homologue périphérique, il ouvre un homologue à la demande avec l'homologue distant D3F (1.1.10.1) et démarre le circuit.

débogage

La première étape d'un réseau DLSw consiste à activer les homologues. Sans les homologues, il n'y a pas d'échange de données. La plupart des détails de ce qui se produit entre les homologues DLSw sont expliqués dans la RFC 1795.

Remarque : si vous vous adressez à un équipement non Cisco via DLSw, utilisez DLSw. Cependant, entre les routeurs Cisco, utilisez DLSw+.

Ce résultat provient de l'émission de **debug dlsw peers** et de l'activation des homologues entre deux routeurs Cisco :

```

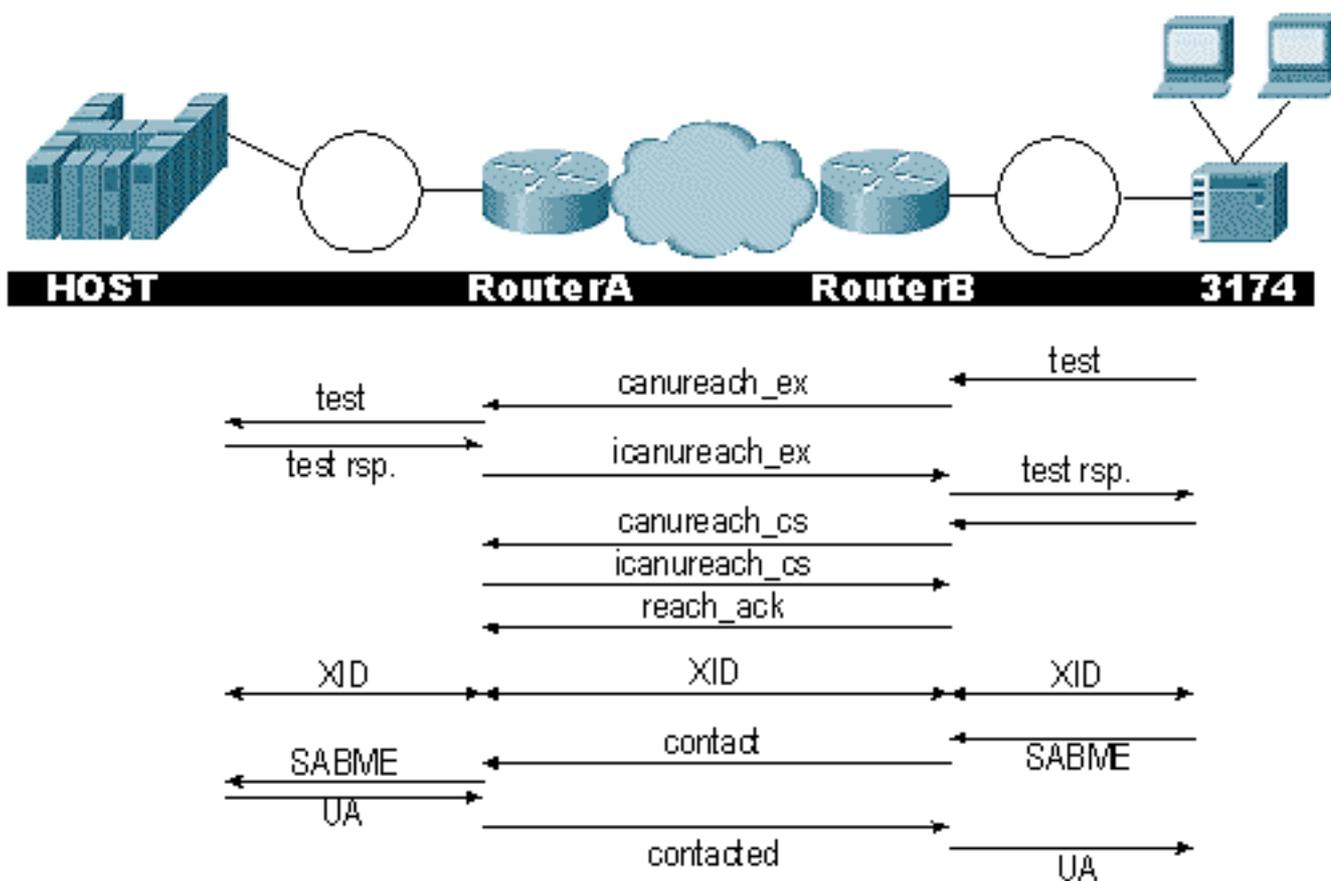
DLsw: passive open 5.5.5.1(11010) -> 2065
DLsw: action_b(): opening write pipe for peer 5.5.5.1(2065)
DLsw: peer 5.5.5.1(2065), old state DISCONN, new state CAP_EXG
DLsw: CapExId Msg sent to peer 5.5.5.1(2065)
DLsw: Recv CapExId Msg from peer 5.5.5.1(2065)
DLsw: Pos CapExResp sent to peer 5.5.5.1(2065)
DLsw: action_e(): for peer 5.5.5.1(2065)
DLsw: Recv CapExPosRsp Msg from peer 5.5.5.1(2065)
DLsw: action_e(): for peer 5.5.5.1(2065)
shSw: peer 5.5.5.1(2065), old state CAP_EXG, new state CONNECT
DLsw: peer_act_on_capabilities() for peer 5.5.5.1(2065)
DLsw: action_f(): for peer 5.5.5.1(2065)
DLsw: closing read pipe tcp connection for peer 5.5.5.1(2065)

```

Ce résultat montre le routeur démarrant l'homologue et ouvrant une session TCP avec l'autre routeur. Puis il commence à échanger des capacités. Après un échange positif de capacités, l'homologue est connecté. Contrairement à RSRB, DLsw ne déplace pas l'homologue vers un état fermé lorsqu'il n'y a aucune activité, telle que le trafic. Ils restent toujours connectés. Si les homologues sont déconnectés, émettez **debug dlsw peer** pour déterminer pourquoi ils ne sont pas capables de s'ouvrir.

Lors du dépannage d'une session en cours d'ouverture, émettez **debug dlsw core** pour observer l'échec de la session et pour vérifier si le circuit s'active.

Voici le flux d'un contrôleur de communications 3174 vers l'hôte via DLsw+ :



Le résultat de **debug dlsw** affiche le flux de la session qui est allumée correctement :

```

ibu-7206#debug dlsw
DLsw reachability debugging is on at event level for all protocol traffic
DLsw peer debugging is on
DLsw local circuit debugging is on
DLsw core message debugging is on
DLsw core state debugging is on
DLsw core flow control debugging is on
DLsw core xid debugging is on
ibu-7206#
DLsw Received-ctlQ : CLSI Msg : UDATA_STN.Ind   dlen: 208
CSM: Received CLSI Msg : UDATA_STN.Ind   dlen: 208 from TokenRing3/0
CSM:   smac 8800.5a49.1e38, dmac c000.0000.0080, ssap F0, dsap F0
CSM: Received frame type NETBIOS DATAGRAM from 0800.5a49.1e38, To3/0
DLsw: peer_put_bcast() to non-grouped peer 5.5.5.1(2065)
DLsw: Keepalive Request sent to peer 5.5.5.1(2065)
DLsw: Keepalive Response from peer 5.5.5.1(2065)
DLsw Received-ctlQ : CLSI Msg : TEST_STN.Ind   dlen: 41
CSM: Received CLSI Msg : TEST_STN.Ind   dlen: 41 from TokenRing3/0
CSM:   smac c001.68ff.0001, dmac 4000.0000.0001, ssap 4 , dsap 0

```

Notez la trame de test provenant du réseau local (localement) de la station c001.68ff.0001 à l'adresse MAC 4000.0000.0001. Chaque **.Ind** indique qu'un paquet arrive du réseau local. Lorsque le routeur envoie un paquet au réseau local, vous voyez un **.RSP**.

```

DLsw: peer_put_bcast() to non-grouped peer 5.5.5.1(2065)
%DLSWC-3-RECVSSP: SSP OP = 4( ICR ) -explorer from peer 5.5.5.1(2065)
DISP Sent : CLSI Msg : TEST_STN.Rsp   dlen: 44

```

Vous pouvez maintenant voir la diffusion envoyée à l'homologue distant et la réponse ICR (débit de cellules initial). Cela signifie que le routeur distant a identifié la station comme étant accessible. TEST_STN.Rsp est le routeur qui envoie une réponse de test à la station.

```

DLsw Received-ctlQ : CLSI Msg : ID_STN.Ind   dlen: 54
CSM: Received CLSI Msg : ID_STN.Ind   dlen: 54 from TokenRing3/0
CSM:   smac c001.68ff.0001, dmac 4000.0000.0001, ssap 4 , dsap 4

```

Une fois que la station reçoit la réponse de test, elle envoie le premier XID. Vous pouvez le remarquer avec IS_STN.Ind. Le routeur doit maintenant conserver cette trame temporairement jusqu'à ce qu'il efface quelques détails entre les deux routeurs DLsw.

```

DLsw: new_ckt_from_clsi(): TokenRing3/0 4001.68ff.0001:4->4000.0000.0001:4
DLsw: START-FSM (1622182940): event:DLC-Id state:DISCONNECTED
DLsw: core: dlsw_action_a()
DISP Sent : CLSI Msg : REQ_OPNSTN.Req   dlen: 108
DLsw: END-FSM (1622182940): state:DISCONNECTED->LOCAL_RESOLVE

DLsw Received-ctlQ : CLSI Msg : REQ_OPNSTN.Cfm CLS_OK dlen: 108
DLsw: START-FSM (1622182940): event:DLC-ReqOpnStn.Cnf state:LOCAL_RESOLVE
DLsw: core: dlsw_action_b()
CORE: Setting lf size to 30
%DLSWC-3-SENDSSP: SSP OP = 3(CUR) to peer 5.5.5.1(2065) success
DLsw: END-FSM (1622182940): state:LOCAL_RESOLVE->CKT_START

%DLSWC-3-RECVSSP: SSP OP = 4(ICR) from peer 5.5.5.1(2065)

```

```

DLSw: 1622182940 recv FCI 0 - s:0 so:0 r:0 ro:0
DLSw: recv RWO
DLSw: START-FSM (1622182940): event:WAN-ICR state:CKT_START
DLSw: core: dlsw_action_e()
DLSw: sent RWO
DLSw: 1622182940 sent FCI 80 on ACK - s:20 so:1 r:20 ro:1
%DLSWC-3-SENDSSP: SSP OP = 5(ACK) to peer 5.5.5.1(2065) success
DLSw: END-FSM (1622182940): state:CKT_START->CKT_ESTABLISHED

```

Vous pouvez remarquer ici le flux interne de DLSw entre les deux homologues. Ces paquets sont normaux pour chaque démarrage de session. La première étape consiste à passer d'un état déconnecté à un état CKT_ESTABLISHED. Les deux routeurs transmettent une trame CUR pour le circuit lui-même. Il s'agit de la commande can u reach circuit setup (CURCS). Lorsque l'homologue qui initie la trame CURCS reçoit une trame ICRCS, il envoie un accusé de réception et passe à l'état Circuit established. Les deux routeurs DLSw sont désormais prêts pour le traitement XID.

```

DLSw: START-FSM (1622182940): event:DLC-Id state:CKT_ESTABLISHED
DLSw: core: dlsw_action_f()
DLSw: 1622182940 sent FCA on XID
%DLSWC-3-SENDSSP: SSP OP = 7(XID) to peer 5.5.5.1(2065) success
DLSw: END-FSM (1622182940): state:CKT_ESTABLISHED->CKT_ESTABLISHED

```

Le routeur a reçu un XID après avoir envoyé la réponse de test à la station. Il enregistre ce XID pendant un instant, puis le transmet à l'homologue à travers le circuit. Cela signifie que vous envoyez des paquets vers/depuis l'homologue avec l'ID de circuit qui lui est associé. De cette façon, DLSw comprend l'activité entre les deux stations. N'oubliez pas que DLSw met fin à la session LLC2 (Logical Link Control, type 2) de chaque côté du cloud.

```

%DLSWC-3-RECVSSP: SSP OP = 7(XID) from peer 5.5.5.1(2065)
DLSw: 1622182940 recv FCA on XID - s:20 so:0 r:20 ro:0
DLSw: START-FSM (1622182940): event:WAN-XID state:CKT_ESTABLISHED
DLSw: core: dlsw_action_g()
DISP Sent : CLSI Msg : ID.Rsp dlen: 12
DLSw: END-FSM (1622182940): state:CKT_ESTABLISHED->CKT_ESTABLISHED

```

```

DLSw Received-ctlQ : CLSI Msg : ID.Ind dlen: 39
DLSw: START-FSM (1622182940): event:DLC-Id state:CKT_ESTABLISHED
DLSw: core: dlsw_action_f()
%DLSWC-3-SENDSSP: SSP OP = 7(XID) to peer 5.5.5.1(2065) success
DLSw: END-FSM (1622182940): state:CKT_ESTABLISHED->CKT_ESTABLISHED

```

Vous remarquez d'abord une réponse au premier XID envoyé précédemment. Dans ID.Rsp, vous voyez que le XID a été envoyé à la station, à laquelle la station a répondu avec un ID.Ind. Il s'agit d'un autre XID envoyé à l'homologue DLSw.

```

%DLSWC-3-RECVSSP: SSP OP = 8(CONQ) from peer 5.5.5.1(2065)
DLSw: START-FSM (1622182940): event:WAN-CONQ state:CKT_ESTABLISHED

```

Cette partie nous montre que la station de l'autre côté a répondu avec un SABME (**CONQ**) au XID. La négociation XID a pris fin et le routeur est prêt à démarrer la session.

```

DLSw: core: dlsw_action_i()

```

DISP Sent : CLSI Msg : **CONNECT.Req** dlen: 16

Ensuite, le routeur envoie le SABME à la station dans **CONNECT.Req**.

DLsw: END-FSM (1622182940): state:CKT_ESTABLISHED->CONTACT_PENDING

DLsw Received-ctlQ : CLSI Msg : CONNECT.Cfm CLS_OK dlen: 8

DLsw: START-FSM (1622182940): event:DLC-Connect.Cnf state:CONTACT_PENDING

DLsw: core: dlsw_action_j()

%DLSWC-3-SENDSSP: SSP OP = 9(CONR) to peer 5.5.5.1(2065) success

DISP Sent : CLSI Msg : FLOW.Req dlen: 0

DLsw: END-FSM (1622182940): state:CONTACT_PENDING->CONNECTED

Vous recevez ensuite l'accusé de réception non numéroté (UA) de la station, qui est affiché dans le message **CONNECT.Cfm**. Il est envoyé à l'homologue distant via un **CONR**. Ensuite, le processus de débit relatif (RR) est lancé avec **FLOW.Req**.

%DLSWC-3-RECVSSP: SSP OP = 10(INFO) from peer 5.5.5.1(2065)

DLsw: 1622182940 decr r - s:20 so:0 r:19 ro:0

DLsw: START-FSM (1622182940): event:WAN-INFO state:CONNECTED

DLsw: core: dlsw_action_m()

DISP Sent : CLSI Msg : **DATA.Req** dlen: 34

DLsw: END-FSM (1622182940): state:CONNECTED->CONNECTED

DLsw: 1622182940 decr s - s:19 so:0 r:19 ro:0

DLsw Received-disp : CLSI Msg : **DATA.Ind** dlen: 35

DLsw: sent RWO

DLsw: 1622182940 sent FCI 80 on INFO - s:19 so:0 r:39 ro:1

%DLSWC-3-SENDSSP: SSP OP = 10(INFO) to peer 5.5.5.1(2065) success

%DLSWC-3-RECVSSP: SSP OP = 10(INFO) from peer 5.5.5.1(2065)

DLsw: 1622182940 decr r - s:19 so:0 r:38 ro:1

DLsw: 1622182940 recv FCA on INFO - s:19 so:0 r:38 ro:0

DLsw: 1622182940 recv FCI 0 - s:19 so:0 r:38 ro:0

DLsw: recv RWO

DLsw: START-FSM (1622182940): event:WAN-INFO state:CONNECTED

DLsw: core: dlsw_action_m()

DISP Sent : CLSI Msg : DATA.Req dlen: 28

DLsw: END-FSM (1622182940): state:CONNECTED->CONNECTED

Le message **DATA.Req** indique que le routeur a transmis une trame I. **Data.Ind** indique que le routeur a reçu une trame I. Vous pouvez utiliser ces informations pour déterminer le flux de paquets sur les routeurs DLsw.

DLsw Received-ctlQ : CLSI Msg : DISCONNECT.Ind dlen: 8

DLsw: START-FSM (1622182940): event:DLC-Disc.Ind state:CONNECTED

Cette partie contient un **DISCONNECT.Ind**. Un **.Ind** indique un paquet provenant du LAN. Dans ce cas, la station envoie un message **DISCONNECT**, ce qui entraîne le démarrage du démontage du circuit par le routeur.

DLsw: core: dlsw_action_n()

%DLSWC-3-SENDSSP: SSP OP = 14(HLTQ) to peer 5.5.5.1(2065) success

DLsw: END-FSM (1622182940): state:CONNECTED->DISC_PENDING

```
%DLSWC-3-RECVSSP: SSP OP = 15( HLTR ) from peer 5.5.5.1(2065)
DLSw: START-FSM (1622182940): event:WAN-HLTR state:DISC_PENDING
```

Une fois que le routeur a reçu le message DISCONNECT, il envoie un message HALT à l'homologue distant et attend la réponse. Il ne reste plus qu'à envoyer un UA à la station et à fermer le circuit, ce qui est montré dans le débogage suivant avec DISCONNECT.Rsp :

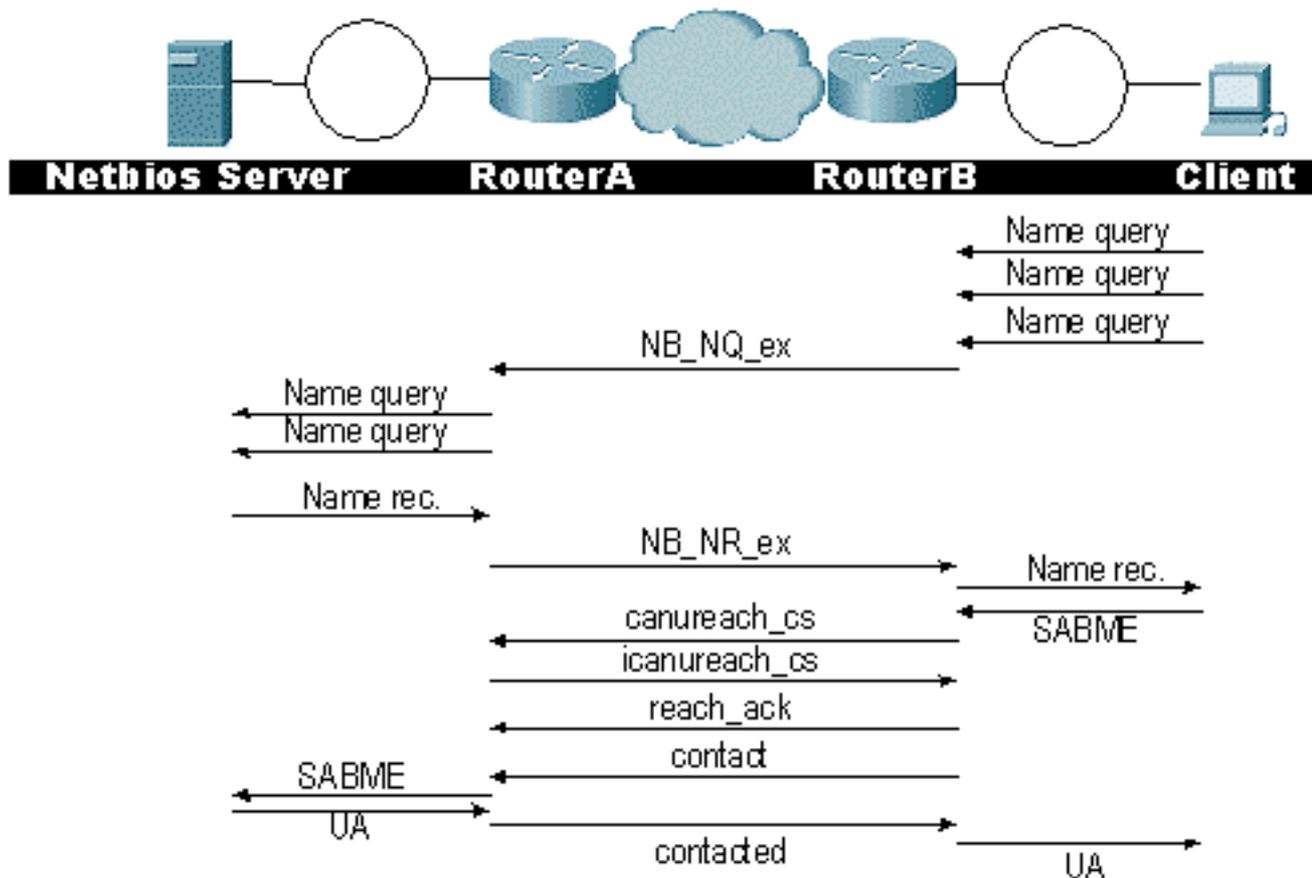
```
DLSw: core: dlsw_action_q()
DISP Sent : CLSI Msg : DISCONNECT.Rsp dlen: 4
DISP Sent : CLSI Msg : CLOSE_STN.Req dlen: 4
DLSw: END-FSM (1622182940): state:DISC_PENDING->CLOSE_PEND

DLSW Received-ctlQ : CLSI Msg : CLOSE_STN.Cfm CLS_OK dlen: 8
DLSw: START-FSM (1622182940): event:DLC-CloseStn.Cnf state:CLOSE_PEND
DLSw: core: dlsw_action_y()
DLSw: 1622182940 to dead queue
DLSw: END-FSM (1622182940): state:CLOSE_PEND->DISCONNECTED
```

La dernière chose que DLSw effectue est de mettre le circuit dans la file d'attente morte. À partir de là, les pointeurs sont nettoyés et prêts pour un nouveau circuit.

Sessions NetBIOS

DLSw gère les sessions NetBIOS différemment, mais les débogages sont très similaires.



Remarque : n'oubliez pas que les XID ne circulent pas pour les stations NetBIOS et que les routeurs DLSw échangent des trames de processeur de commutation système (SSP) de requête de nom NetBIOS et que le nom NetBIOS est reconnu. C'est la principale différence.

Informations connexes

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

À propos de cette traduction

Cisco a traduit ce document en traduction automatisée vérifiée par une personne dans le cadre d'un service mondial permettant à nos utilisateurs d'obtenir le contenu d'assistance dans leur propre langue.

Il convient cependant de noter que même la meilleure traduction automatisée ne sera pas aussi précise que celle fournie par un traducteur professionnel.