

# Troubleshooting de DLSw com Comandos debug

## Índice

[Introdução](#)

[Pré-requisitos](#)

[Requisitos](#)

[Componentes Utilizados](#)

[Convenções](#)

[Debugs](#)

[Tradução de mídia DLSw](#)

[DLSw administrando a tradução de mídia reversa](#)

[Tradução de mídia DLSw local](#)

[Problemas de desempenho](#)

[Informações Relacionadas](#)

## [Introdução](#)

Este documento fornece a informação em como pesquisar defeitos o switching de link de dados (DLSw) com **comandos debug**.

## [Pré-requisitos](#)

### [Requisitos](#)

Não existem requisitos específicos para este documento.

### [Componentes Utilizados](#)

Este documento não é restrito a versões de software ou hardware específicas.

As informações neste documento foram criadas a partir de dispositivos em um ambiente de laboratório específico. Todos os dispositivos utilizados neste documento foram iniciados com uma configuração (padrão) inicial. Se a sua rede estiver ativa, certifique-se de que entende o impacto potencial de qualquer comando.

### [Convenções](#)

Para obter mais informações sobre convenções de documento, consulte as [Convenções de dicas técnicas Cisco](#).

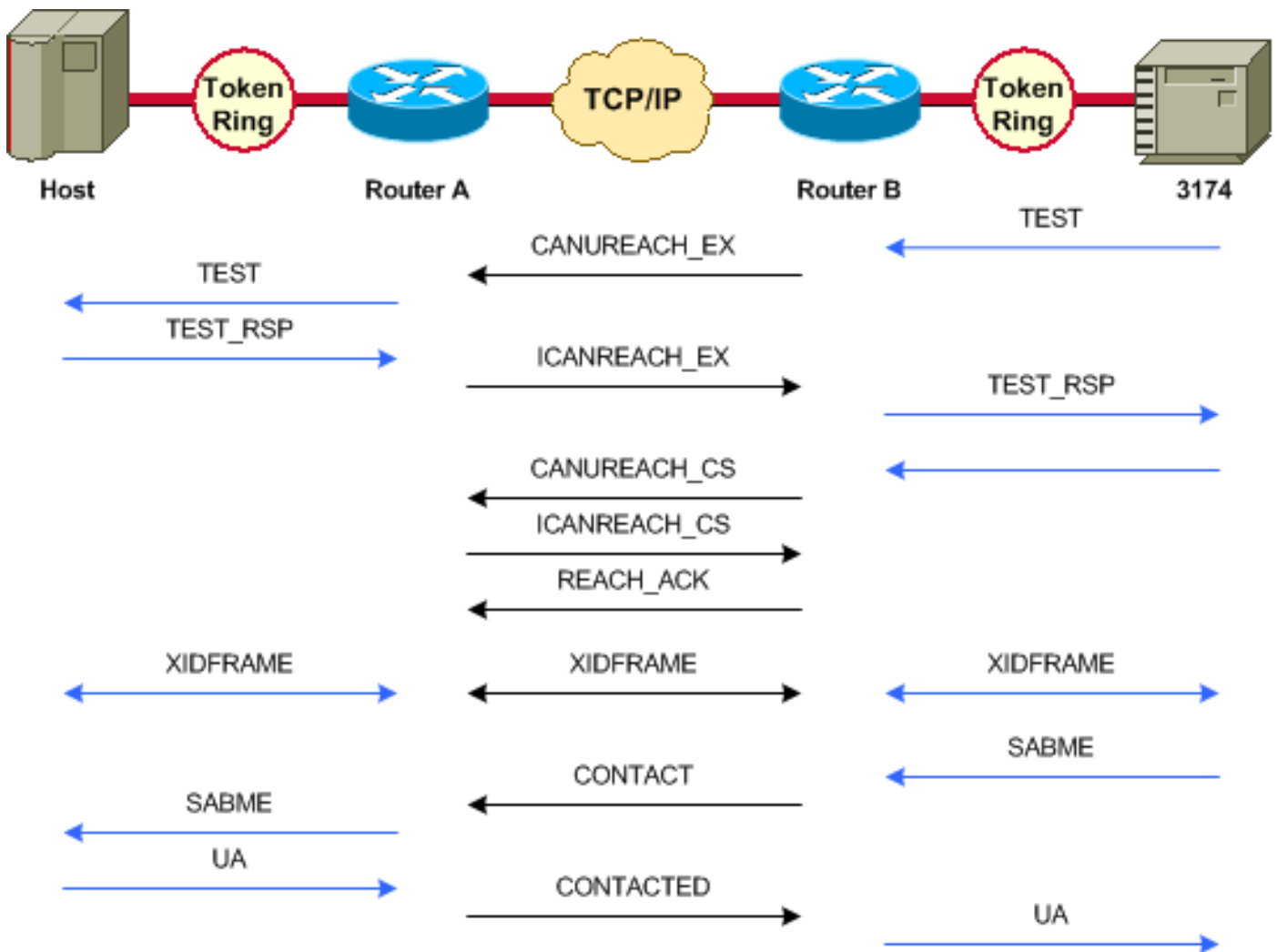
# Debugs

Antes que você tente alguns dos **comandos debug** neste documento, refira a [informação importante em comandos Debug](#).

Quando você está pesquisando defeitos a partida de uma sessão, emita o **comando debug dlsw** e observe-o:

- A instalação inicial da sessão
- Se o circuito está vindo acima

Este diagrama mostra o fluxo para um controlador de comunicações de Cisco 3174 ao host com o Data-Link Switching Plus (DLSw+):



O exemplo seguinte do **comando debug dlsw** mostra o fluxo de uma sessão correta enquanto é trazido acima.

**Caution:** O **comando debug dlsw** pode causar a degradação séria do desempenho, especialmente quando executado em um roteador que tenha circuitos múltiplos conecte com os peer múltiplos configurados.

```
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
```

Observe o frame para teste que está vindo do LAN (localmente) da estação c001.68ff.0001 ao MAC address de 4000.0000.0001. Cada vez que isso você vê a. Ind, é um pacote que esteja vindo do LAN. Quando isso um pacote é enviado ao LAN, você deve ver um .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
```

Observe a transmissão que é enviada ao peer remoto e à resposta ICR (eu posso alcançar). Isto significa que o roteador remoto identificou a estação como alcançável. Observe então o TEST\_STN.Rsp, que é o roteador??? resposta de teste s à estação.

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

Depois que a estação recebe a resposta de teste, a primeira identificação de intercâmbio (XID) está enviada ao roteador Cisco; isto pode ser visto com o ID\_STN.Ind. O roteador pára nesse quadro até que os detalhes sejam limpos entre os dois roteadores 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
```

Observe o fluxo interno de DLSw entre os dois pares. Esses pacotes são normais para toda inicialização de sessão.

A primeira etapa neste processo é mover-se de um estado desconectado para um estado CKT\_ESTABLISHED; esta sequência ocorre:

1. Os dois roteadores transmitem um quadro CUR para o próprio circuito, chamado de CUR\_cs (configuração de circuito Can You Reach).
2. Quando o correspondente que inicia o quadro CUR\_cs recebe um quadro ICR\_cs, o correspondente envia um reconhecimento e se movimenta para estabelecer um circuito.
3. Os dois roteadores DLSw já estão prontos para o processamento da 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
```

Depois que um XID é recebido, a resposta de teste está enviada à estação e mantida pelo roteador. O roteador transmite então este XID a seu par através deste circuito, assim que significa que os pacotes estão sendo enviados a e do par com o circuit id etiquetado a ele.

Desta maneira, DLSw conhece o que está indo em duas estações no meio, porque DLSw termina a sessão LLC2 em cada lado da nuvem.

```
gnb%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
```

Inicialmente, há uma resposta ao primeiro XID que tinha sido enviado antes.

Observe que, em ID.Rsp, o XID está enviado à estação, que responde para trás com um ID.Ind (que é um outro XID que seja enviado transversalmente ao par de 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
```

A estação no outro lado responde com um SABME (CONQ) ao XID; conseqüentemente, a negociação de XID terminou e a sessão está pronta para começar.

```
DLSw: core: dlsw_action_i()
DISP Sent : CLSI Msg : CONNECT.Req dlen: 16
!--- CONNECT.Req means that a SABME has been sent. 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
```

O roteador recebe agora o UA da estação, e este pode ser visto na mensagem do `CONNECT.Cfm`. Isto é enviado ao peer remoto através de um `CONR`.

```
%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
```

O `DATA.Req` é uma indicação que um Eu-quadro esteve transmitido; `DATA.Ind` é uma indicação que um Eu-quadro esteve recebido. Estes são muito úteis determinar que pacotes estão fluindo através dos roteadores DLSw.

```
%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
```

Essa saída mostra um `DISCONNECT.Ind`; como visto antes. O `Ind` está vindo do LAN. Isso significa que a estação enviou uma desconexão. Isto faz com que o roteador comece rasgar para baixo o circuito.

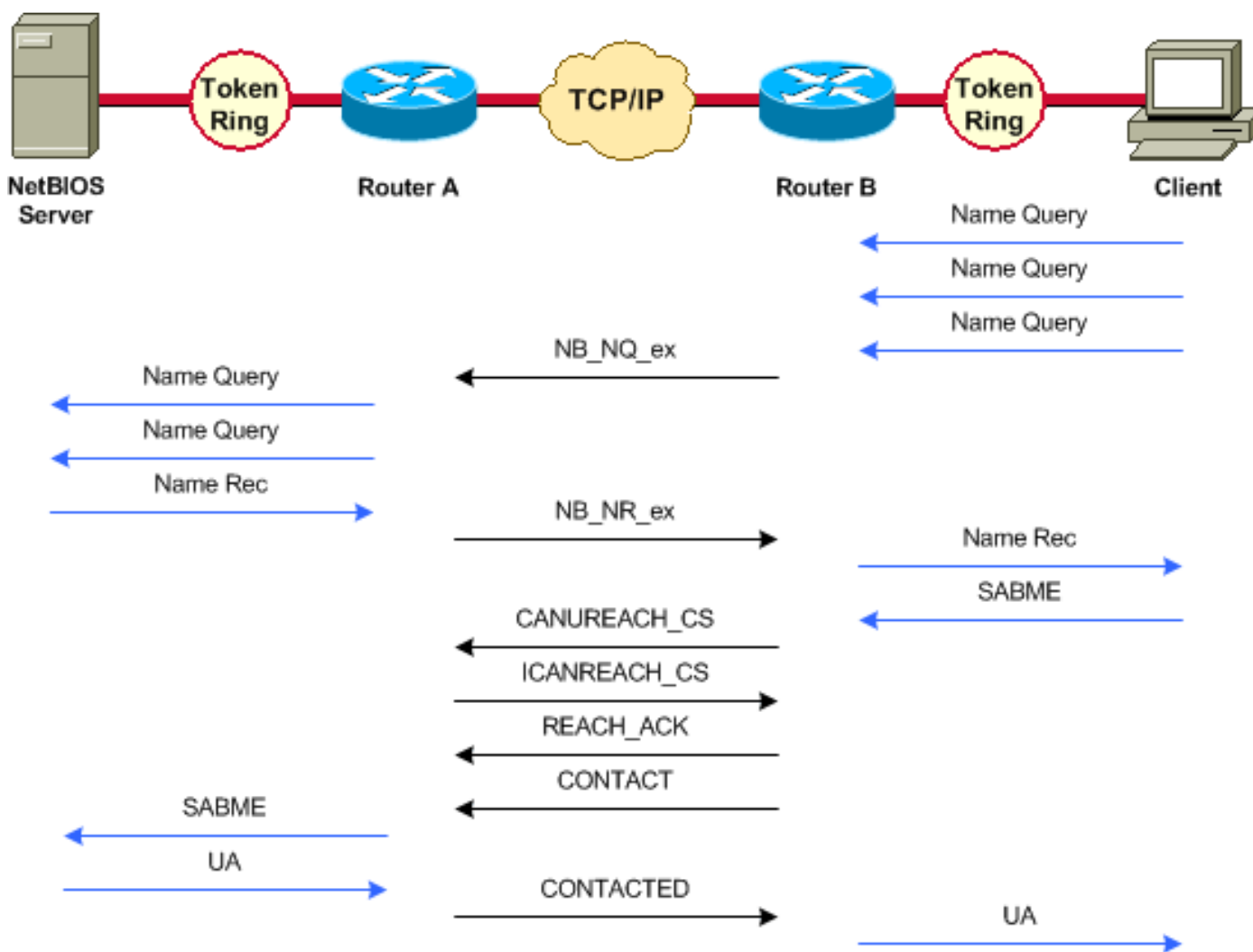
```
DLSw: core: dlsw_action_n()
%DLSWC-3-SENDSSP: SSP OP = 14( HLTQ ) to peer 5.5.5.1(2065) success
DLSw: END-FSM (1622182940): state:CONNECTED->DISC_PENDING
%DLSWC-3-RECVSSP: SSP OP = 15( HLTR ) from peer 5.5.5.1(2065)
DLSw: START-FSM (1622182940): event:WAN-HLTR state:DISC_PENDING
```

Após o recebimento da desconexão, o roteador envia um `HALT` para o peer remoto e aguarda

sua resposta. Após o recebimento da resposta, o roteador envia um UA para a estação e fecha o circuito. Isto é mostrado como o `DISCONNECT.Rsp`:

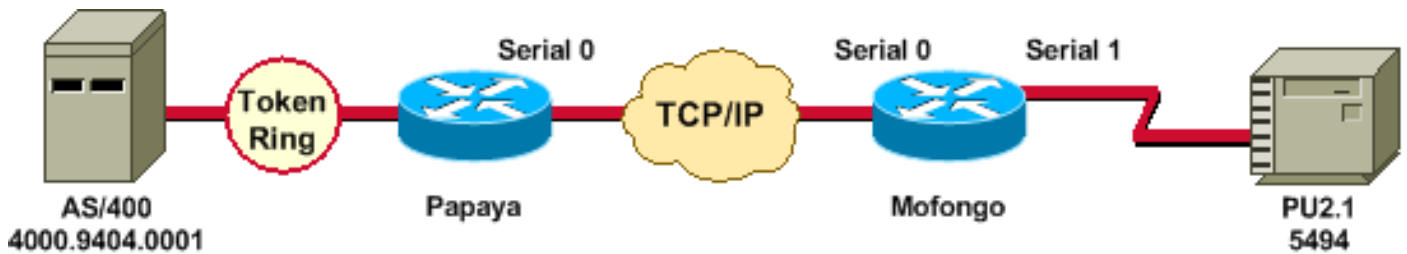
```
DLSw: core: dlsw_action_q()
DISP Sent : CLSI Msg : DISCONNECT.Rsp  dlen: 4
DISP Sent : CLSI Msg : CLOSE_STN.Req  dlen: 4
DLSw: END-FSM (1622182940): state:DISC_PENDING->CLOSE_PEND
DLSW Received-ctlQ : CLSI Msg : CLOSE_STN.Cfm CLS_OK dlen: 8
DLSw: START-FSM (1622182940): event:DLC-CloseStn.Cnf state:CLOSE_PEND
DLSw: core: dlsw_action_y()
DLSw: 1622182940 to dead queue
DLSw: END-FSM (1622182940): state:CLOSE_PEND->DISCONNECTED
```

Em seguida, o DLSw coloca o circuito na fila inativa. Na fila inativa, os ponteiros são apagados e estão prontos para um novo circuito.



Para uma sessão com NetBIOS, há umas mudanças na maneira como o DLSw lida com negociação; mas, debuga são muito similar. A única diferença entre a SNA e o NetBIOS é que as XIDs não fluem para as estações NetBios, e os roteadores DLSw trocam os quadros de consulta de nome e de nome identificado do NetBIOS.

## [Tradução de mídia DLSw](#)



Depois que a relação vem acima, o roteador começa o processo: determina o lugar do controlador remoto.

```

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

```

Depois que o quadro ICR é recebido, o DLsw liga a máquina de estado finito (FSM) para esta sessão. Isto é feito pelo REQ\_OPNSTN.Req e REQ\_OPNSTN.Cfm que vão entre DLsw e os Cisco Link Service connecte (CLSI).

```

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

```

Após se comunicar com a CLSI, o roteador envia quadros CUR de início de sessão para o roteador remoto. Esses quadros CUR estão somente entre os dois roteadores.

```

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

```

Depois que o circuito é estabelecido, envia a troca armazenada XID e de começo XID.

É muito importante compreender de onde os XID estão vindo. Nesta situação, há duas saídas importantes:

- DLC-identificação??? Significa que o XID veio da estação local DLC.
- WAN-XID??? Significa que o XID veio do roteador remoto (a estação remota).

```
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: dls_action_y()
DLsw: 1622182940 to dead queue
DLsw: END-FSM (1622182940): state:CLOSE_PEND->DISCONNECTED

```

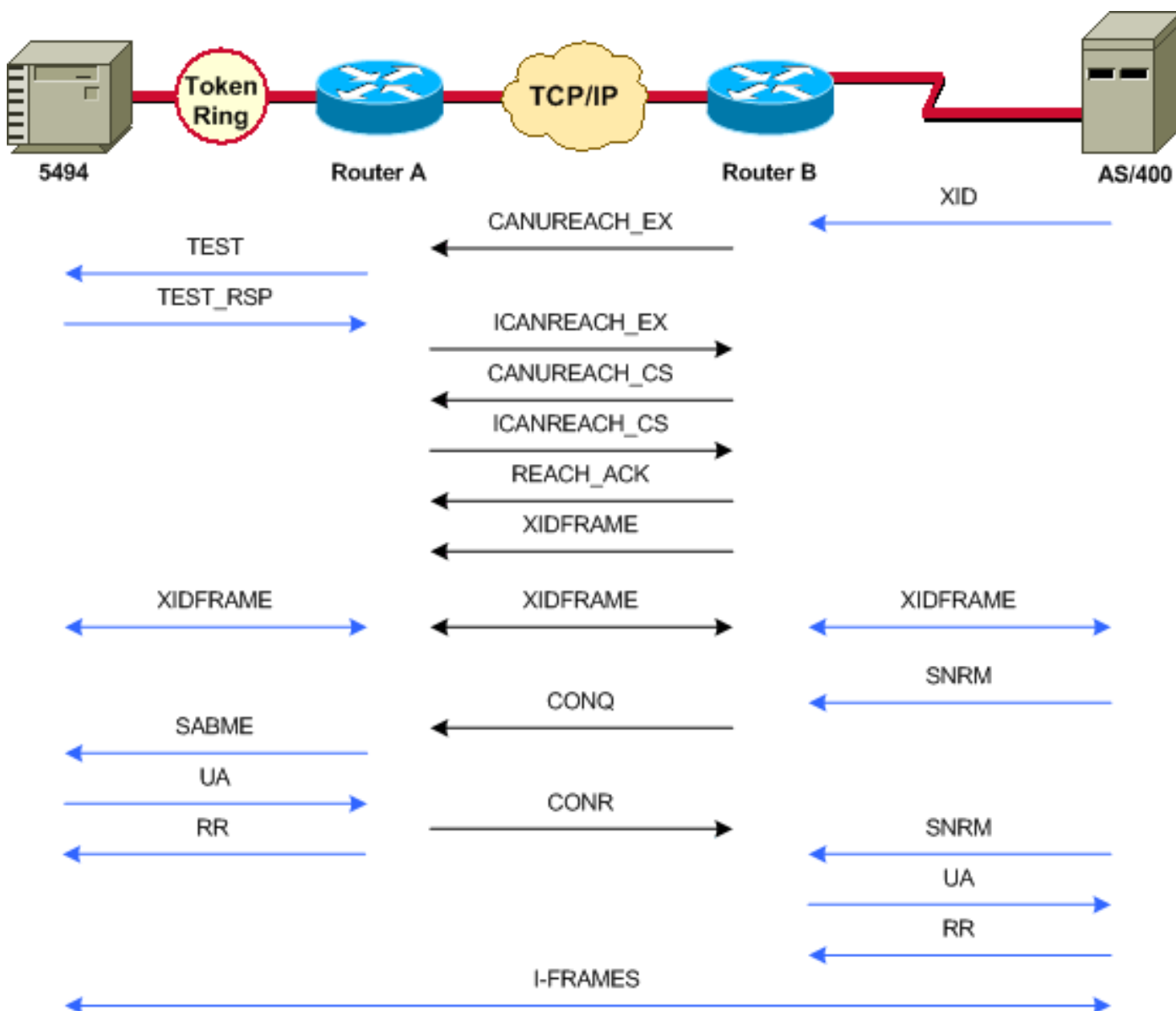
O roteador recebe então o CONQ do AS/400 (SABME) que é traduzido à linha de série como uma resposta normal do grupo (SNRM). Quando o UA aparecer na linha de série (CONNECT.Cfm), o roteador envia o CONR ao outro lado e move a sessão para CONECTADO.

```

DLsw: core: dls_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: dls_action_y()
DLsw: 1622182940 to dead queue
DLsw: END-FSM (1622182940): state:CLOSE_PEND->DISCONNECTED

```

### DLSw administrando a tradução de mídia reversa





Uma outra instalação comum é chamada o Logical Link Control do controle de link de dados de reverso-Síncrono (SDLC) (LLC; SDLLC), que é quando a estação principal for anexada ao roteador através de uma linha SDLC. Isto é visto geralmente nos ambientes do host que migram o host a um acessório do Token Ring. Esta instalação muda a maneira que DLSw segura a linha SDLC, porque há geralmente um grau elevado de incerteza se o PU remoto é ativo ou não.

Porque o AS/400 é preliminar ou é ajustado a negociável no papel, precisa de começar a sessão. Quando isso acontece, esta sequência ocorre:

1. A linha serial torna-se operacional.
2. O AS/400 envia o primeiro XID.
3. O processo de pesquisa para os começos do controlador remoto.
4. A instalação para o circuito termina.
5. A negociação de XID é iniciada na linha.

### Tradução de mídia DLSw local

Quando a negociação de XID termina, o AS/400 envia um SNRM ao roteador Cisco. Isto faz com que o roteador envie um CONQ e espere um CONR do roteador remoto. Mas o UA não é enviado até depois do recibo do CONR.



```
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
```

Porque este é um DLSw local, o comportamento é um pouco de diferente. A primeira coisa vista é o XID do lado serial. O XID do lado de série precisa de ser armazenado até os frames para teste LLC e as respostas são terminadas.

```
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
```

A estação de teste sae do roteador, e a resposta volta do AS/400. O FS local pode agora ser criado. (Recorde que esta é uma sessão local.)

```
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
```

Depois que a confirmação local que o FS está pronto, o roteador envia o XID (ID.Req) ao sócio, que é o AS/400 nesta encenação.

```
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
```

Um XID vem do Token Ring. Esse ID.Ind tem um comprimento de 108 e deve ser encaminhado ao parceiro nesse cenário, que é a linha SDLC. Isso pode ser visto com o ID.Req que foi enviado. Observe que, cada vez que aquele um pacote é recebido, um LFS deve ser começado.

```
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
```

Observe a resposta de XID da linha serial e como ela é encaminhada para o parceiro (nesse caso, a estação Token Ring). Isto vai sobre por um tempo, até que a troca XID para este dispositivo PU2.1 esteja terminada.

```
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
```

Depois que a troca XID, um SABME é recebida do AS/400 com o CONNECT.Ind. Isso instrui o roteador a enviar um CONNECT.Req à linha de SDLC, que é o SNRM. Um CONNECT.Cfm (UA) é recebido então da linha de série, que faz com que o código de DLSw envie um CONNECT.Rsp (UA) ao AS/400.

```
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
```

Esta é a sessão que ocorre quando o controlador (SDLC) é fechado:

```
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
```

Aqui, um DISC é enviado ao AS/400 (**DISCONNECT.Rsp**). O circuito local é, em seguida, eliminado.

```
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
```

Depois que o DISCONNECT.Ind (UA) é recebido do AS/400, a sessão está limpada, e é movida para um estado da desconexão.

## [Problemas de desempenho](#)

Para obter mais informações sobre dos problemas de desempenho, refira a seção do [gerenciamento de largura de banda e do Enfileiramento no Data-Link Switching Plus \(DLSw+\)](#), ou refira [técnicas de filtração do DLSw+ SAP/MAC](#).

## [Informações Relacionadas](#)

- [Troubleshooting de DLSw](#)
- [Apoio de DLSw e de DLSw+](#)
- [Suporte de tecnologia](#)
- [Suporte de Produto](#)
- [Suporte Técnico - Cisco Systems](#)