

Exemplo da configuração de redundância do DLSw Ethernet

Índice

[Introdução](#)

[Pré-requisitos](#)

[Requisitos](#)

[Componentes Utilizados](#)

[Convenções](#)

[Configurar](#)

[Diagrama de Rede](#)

[Notas de configuração](#)

[Configurações](#)

[Verificar](#)

[Troubleshooting](#)

[Comandos para Troubleshooting](#)

[Informações Relacionadas](#)

[Introdução](#)

Esta configuração de exemplo executa os recursos de redundância de Ethernet do software de Cisco IOS® em uma rede do switching de link de dados (DLSw). Igualmente são incluídas neste documento as técnicas para debugar esta característica. O recurso de redundância de Ethernet foi adicionado ao Cisco IOS Software versão 12.0(5)T.

Esta configuração de exemplo mostra como executar os recursos de redundância de Ethernet e os comandos usados para monitorar o estado das conexões. Três encenações são simuladas em que a falha ocorre quando os comandos debug forem executado a fim mostrar o comportamento de cada roteador.

[Pré-requisitos](#)

[Requisitos](#)

Não existem requisitos específicos para este documento.

[Componentes Utilizados](#)

Este documento não se restringe a versões de software e hardware específicas.

[Convenções](#)

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

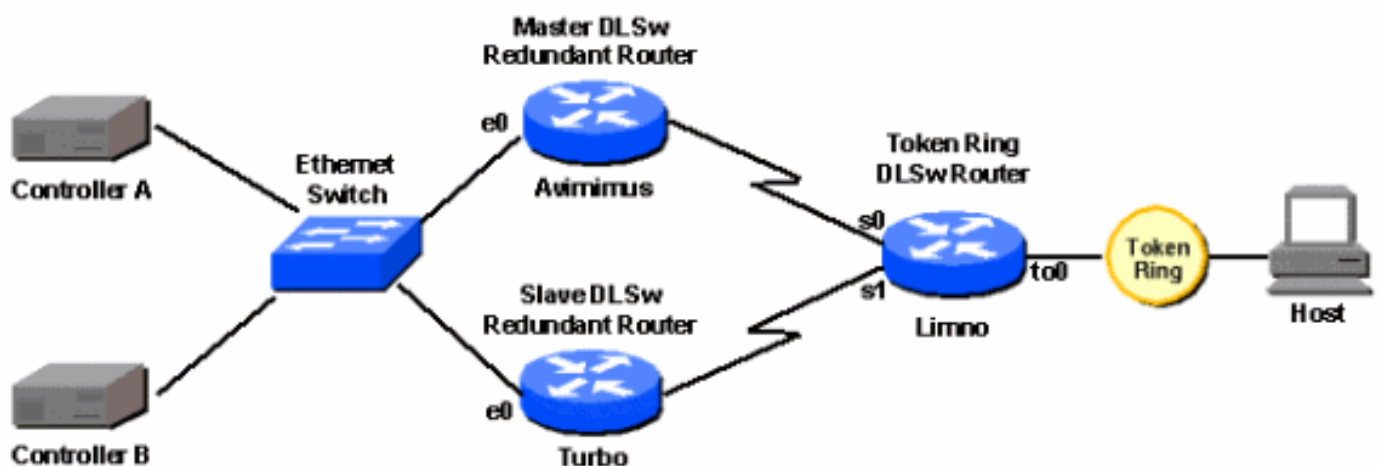
Configurar

Nesta seção, você encontrará informações para configurar os recursos descritos neste documento.

Nota: Para localizar informações adicionais sobre os comandos usados neste documento, utilize a Ferramenta Command Lookup (somente clientes [registrados](#)).

Diagrama de Rede

Este documento utiliza a seguinte configuração de rede:



Notas de configuração

- O controlador A é configurado para estabelecer duas sessões de controle de link lógico (LLC) com o host; usa um valor do Access point do serviço local (LSAP) de 0x04 e de 0x08, respectivamente. O controlador B tenta estabelecer somente uma sessão de LLC, e o host usa um valor lsap de 0x04.
- Para esta configuração de exemplo, quando ambos os roteadores DLSw estão acima, o controlador A conecta ao host com o roteador avimimus, e o controlador B conecta ao host com o turbocompressor do roteador. O Balanceamento de carga é realizado quando ambo o Roteadores é ascendente e trabalho corretamente.
- Estão aqui os endereços MAC dos controladores e das configurações: Controlador A smac = 0000.2222.0000 (cann), dmac = 0000.6666.0000 (cann)
lsap = 0x04 and lsap = 0x08, dsap = 0x04 for both connections Para a referência fácil, estes são os valores de permuta de bit deste MAC address: smac = 0000.4444.0000 (non-cann), dmac = 0000.6666.0000 (non-cann) Controlador B smac = 0000.1111.0000 (cann), dmac = 0000.7777.0000 (cann)
lsap = 0x08, dsap = 0x04 Para a referência fácil, estes são os valores de permuta de bit deste MAC address: smac = 0000.8888.0000 (non-cann), dmac = 0000.EEEE.0000 (non-can)
- O host tem este MAC address: smac = 4000.3745.0000 (non-can)
lsap = 0x04 Para a referência fácil, este é o valor de permuta de bit do MAC address do host: smac = 0200.eca2.0000 (cann)

- Avimimus é configurado para ser o roteador DLSw mestre, e o turbocompressor é configurado para ser o roteador DLSw do escravo. Avimimus é atribuído um valor da baixa prioridade para conseguir esta configuração. A prioridade padrão é 100. São alistados aqui os comandos usados para especificar o valor de prioridade para cada roteador: Avimimus — o **dlsw transparente Redundância-permite 9999.9999.9999 a mestre-prioridade 10** Turbocompressor — o **dlsw transparente Redundância-permite 9999.9999.9999** Ambos os Roteadores deve ser configurado para usar o mesmo endereço MAC de transmissão múltipla para uma comunicação (9999.9999.9999).
- Avimimus tem este MAC address configurado: Ethernet 0:
smac = 0000.3333.0000 (cann), 0000.CCCC.0000 (non-cann)
- O turbocompressor tem este MAC address configurado: Ethernet 0:
smac = 0000.5555.0000 (cann), 0000.AAAA.0000 (non-cann)
- Avimimus é configurado para fazer este mapeamento do MAC address: local mac:
0000.6666.0000 --> remote mac: 0200.ECA2.0000 (cann) Para a referência fácil, estes são os valores de permuta de bit deste MAC address: local mac: 0000.6666.0000 --> remote mac: 4000.3745.0000 (non-can)
- O turbocompressor é configurado para fazer este mapeamento do MAC address: local mac:
0000.7777.0000 --> remote mac: 0200.ECA2.0000 (cann) Para a referência fácil, estes são os valores de permuta de bit do MAC address: local mac: 0000.EEEE.0000 --> remote mac: 4000.3745.0000 (non-can)
- São alistados aqui os comandos especificar o mapeamento do MAC address nos roteadores DLSw: Avimimus — **vizinho transparente 0000.5555.0000 do MAC remoto 0200.eca2.0000 do mapa local-MAC 0000.6666.0000 do dlsw** Turbocompressor — **vizinho transparente 0000.3333.0000 do MAC remoto 0200.eca2.0000 do mapa local-MAC 0000.7777.0000 do dlsw**
Nota: Todos os endereços MAC configurados nestes comandos devem estar no formato canônico. O valor " mac local " representa o MAC de destino (DMAC) a que as estações final dos Ethernet apontam (0000.6666.0000 e 0000.7777.0000, neste caso). O valor " mac remoto " é a representação canônica do host remoto que você tenta alcançar com DLSw. O valor vizinho representa o endereço MAC da interface Ethernet do outro roteador redundante DLSw.

Configurações

Este documento utiliza as seguintes configurações:

- [Roteador redundante DLSW mestre](#)
- [Slave DLSw Redundant Router](#)
- [Roteador DLSw de Token Ring](#)

Roteador redundante DLSW mestre

```
avimimus# show run Building configuration... ! hostname
avimimus ! dlsw local-peer peer-id 1.1.1.1 dlsw remote-
peer 0 tcp 3.3.3.1 dlsw transparent switch-support ! !
interface Ethernet0 mac-address 0000.3333.0000 ip
address 5.5.5.3 255.255.255.0 no ip redirects no ip
directed-broadcast media-type 10BaseT standby timers 3
10 standby priority 150 preempt standby mac-address
0000.3333.3333 standby ip 5.5.5.4 dlsw transparent
redundancy-enable 9999.9999.9999 master-priority 10 !---
This is the command used to specify the priority value
of Avimimus. dlsw transparent map local-mac
0000.6666.0000 remote-mac 0200.eca2.0000 neighbor
```

```
0000.5555.0000 !--- This is the command used to specify
the MAC address mapping at the DLSw router. ! interface
Serial0 ip address 1.1.1.1 255.255.255.0 no ip directed-
broadcast no ip mroute-cache no fair-queue clockrate
4000000 ! end
```

Slave DLSw Redundant Router

```
turbo# show run Building configuration... ! hostname
turbo ! dlsw local-peer peer-id 2.2.2.1 dlsw remote-peer
0 tcp 3.3.3.1 dlsw transparent switch-support !
interface Ethernet0 mac-address 0000.5555.0000 ip
address 5.5.5.1 255.255.255.0 no ip redirects no ip
directed-broadcast no ip route-cache no ip mroute-cache
standby timers 3 10 standby priority 100 standby mac-
address 0000.3333.3333 standby ip 5.5.5.4 dlsw
transparent redundancy-enable 9999.9999.9999 !--- This
is the command used to specify the priority value of
Turbo. dlsw transparent map local-mac 0000.7777.0000
remote-mac 0200.eca2.0000 neighbor 0000.3333.0000 !---
This is the command used to specify the MAC address
mapping at the DLSw router. ! interface Serial1 ip
address 2.2.2.1 255.255.255.0 no ip directed-broadcast
no ip route-cache no ip mroute-cache ! end
```

Roteador DLSw de Token Ring

```
limno# show run ! hostname limno ! source-bridge ring-
group 100 ! dlsw local-peer peer-id 3.3.3.1 dlsw remote-
peer 0 tcp 2.2.2.1 dlsw remote-peer 0 tcp 1.1.1.1 !
interface TokenRing0 ring-speed 16 source-bridge 10 1
100 source-bridge spanning ! interface Serial0 ip
address 1.1.1.2 255.255.255.0 no ip directed-broadcast
no ip mroute-cache no fair-queue ! interface Serial1 ip
address 2.2.2.2 255.255.255.0 no ip directed-broadcast
no ip route-cache no ip mroute-cache clockrate 4000000 !
end
```

Verificar

Esta seção fornece informações que você pode usar para confirmar se sua configuração funciona adequadamente. Todo o **show command output (resultado do comando show)** desta configuração é apresentado. Há um comentário com a saída do comando para ajudá-lo a encontrar seções importantes.

A [Output Interpreter Tool \(somente clientes registrados\)](#) oferece suporte a determinados comandos show, o que permite exibir uma análise da saída do comando show.

Esta seção é dividida nestas subseções:

- [Saída do comando show dlsw circuits and show llc](#)
- [Saída do comando show dlsw transparent](#)

Saída do comando show dlsw circuits and show llc

Nesta seção, a saída do **comando show dlsw circuits and show llc** é apresentada.

Roteador redundante DLSW mestre

```
avimimus# show dlsw circuits Index local addr(lsap)
```

```
remote addr(dsap) state uptime 738197594
0000.4444.0000(04) 4000.3745.0000(04) CONNECTED 3d02h
335544411 0000.4444.0000(08) 4000.3745.0000(04)
CONNECTED 3d02h Total number of circuits connected: 2
```

Os circuitos DLSw são estabelecidos entre o MAC de estação final (0000.4444.0000) e o MAC de host. A estação final não está realmente apontando para 4000.3745.0000, mas o roteador está executando o mapeamento de endereço MAC especificado no comando `dlsw transparent map`.

Roteador redundante DLSW mestre

```
avimimus# show llc brief LLC2 Connections: total of 3
connections Et0 NORMAL 0000.aaaa.0000 0000.cccc.0000 94
94 Et0 NORMAL 0000.4444.0000 0000.6666.0000 04 04 Et0
NORMAL 0000.4444.0000 0000.6666.0000 04 08
```

A primeira sessão de LLC é estabelecida entre os roteadores redundantes do DLSw Ethernet usando um valor de SAP de 0x94. Os endereços MAC para os roteadores DLSw aparecem no formato não canônico. Para mais informação, refira a [seção de emissor do comando show dlsw transparent](#) deste documento.

Outras duas sessões de LLC referem quando os dispositivos finais se comunicam com o MAC address configurado no roteador. Isto é, 0000.6666.0000 são o endereço DMAC configurado nas estações final. Todas as sessões de LLC parecem originar do interface ethernet 0 (Et0); contudo, os endereços MAC alistados mais tarde são indicados no formato não canônico. São alistados esta maneira para a referência fácil aos **comandos show dlsw**, que indicam todos os endereços MAC no formato não canônico.

Roteador redundante DLSW mestre

```
avimimus# show llc LLC2 Connections: total of 3
connections Ethernet0 DTE: 0000.aaaa.0000 0000.cccc.0000
94 94 state NORMAL V(S)=5, V(R)=7, Last N(R)=5, Local
window=7, Remote Window=127 akmax=3, n2=8, xid-retry
timer 0/0 ack timer 0/1000 p timer 0/1000 idle timer
9480/10000 rej timer 0/3200 busy timer 0/9600 akdelay
timer 0/100 txQ count 0/200 Ethernet0 DTE:
0000.4444.0000 0000.6666.0000 04 04 state NORMAL V(S)=1,
V(R)=1, Last N(R)=1, Local window=7, Remote Window=127
akmax=3, n2=8, xid-retry timer 0/0 ack timer 0/1000 p
timer 0/1000 idle timer 8920/10000 rej timer 0/3200 busy
timer 0/9600 akdelay timer 0/100 txQ count 0/200
Ethernet0 DTE: 0000.4444.0000 0000.6666.0000 04 08 state
NORMAL V(S)=1, V(R)=1, Last N(R)=1, Local window=7,
Remote Window=127 akmax=3, n2=8, xid-retry timer 0/0 ack
timer 0/1000 p timer 0/1000 idle timer 9100/10000 rej
timer 0/3200 busy timer 0/9600 akdelay timer 0/100 txQ
count 0/200
```

Slave DLSw Redundant Router

```
turbo# show dlsw circuits Index local addr(lsap) remote
addr(dsap) state uptime 2634022913 0000.8888.0000(08)
4000.3745.0000(04) CONNECTED 4d00h Total number of
circuits connected: 1 turbo# show llc brief LLC2
Connections: total of 2 connections Et0 NORMAL
0000.cccc.0000 0000.aaaa.0000 94 94 Et0 NORMAL
0000.8888.0000 0000.eeee.0000 04 08 turbo# show llc LLC2
Connections: total of 2 connections Ethernet0 DTE:
0000.cccc.0000 0000.aaaa.0000 94 94 state NORMAL V(S)=7,
V(R)=5, Last N(R)=7, Local window=7, Remote Window=127
akmax=3, n2=8, xid-retry timer 0/0 ack timer 0/1000 p
```

```
timer 0/1000 idle timer 7480/10000 rej timer 0/3200 busy
timer 0/9600 akdelay timer 0/100 txQ count 0/200
Ethernet0 DTE: 0000.8888.0000 0000.eeee.0000 04 08 state
NORMAL V(S)=1, V(R)=1, Last N(R)=1, Local window=7,
Remote Window=127 akmax=3, n2=8, xid-retry timer 0/0 ack
timer 0/1000 p timer 0/1000 idle timer 2410/10000 rej
timer 0/3200 busy timer 0/9600 akdelay timer 0/100 txQ
count 0/200
```

Saída do comando show dlsw transparent

Nesta seção, a saída do comando `show dlsw transparent` é apresentada.

Roteador redundante DLSW mestre

```
avimimus# show dlsw transparent neighbor Interface
Ethernet0 0000.cccc.0000 SELF Master 0000.aaaa.0000 Rcvd
Master-Accepted VALID
```

O comando `show dlsw transparent neighbor` indica os vizinhos DLSw em um domínio interligado transparente. Todos os endereços MAC alistados são mostrados no formato não canônico. Contudo, quando você configura o roteador vizinho (com o comando `dlsw transparent map`), use o formato canônico. O roteador avimimus tem a mais baixa prioridade e assim que é o roteador mestre.

Roteador redundante DLSW mestre

```
avimimus# show dlsw transparent cache Interface
Ethernet0 Circuit Cache local addr(lsap) remote
addr(dsap) state Owner 0000.4444.0000(04)
4000.3745.0000(04) POSITIVE SELF 0000.4444.0000(08)
4000.3745.0000(04) POSITIVE SELF 0000.8888.0000(08)
4000.3745.0000(04) NEGATIVE 0000.aaaa.0000 Total number
of circuits in the Cache: 3
```

O comando `show dlsw transparent cache` indica o cache de circuito fechado para o domínio de transmissão. Este roteador está ciente da conexão que é estabelecida no segundo roteador; isto corresponde com a entrada identificada por meio de um estado `NEGATIVO`.

Roteador redundante DLSW mestre

```
avimimus# show dlsw transparent map Interface Ethernet0
LOCAL Mac REMOTE MAC BACKUP -----
0000.6666.0000 4000.3745.0000 0000.aaaa.0000 STATIC
0000.eeee.0000 4000.3745.0000 0000.aaaa.0000
DYNAMIC(Passive)
```

Na saída precedente, o roteador está ciente da tradução especificada em sua configuração (veja a `ESTÁTICA` marcada entrada) e igualmente da entrada que é configurada no segundo roteador (`DINÂMICO` marcado da entrada). O endereço MAC LOCAL é o endereço MAC que deve ser usado pelas estações finais como DMAC.

Em seguida, o roteador executa a tradução para endereços MAC que aparecem na coluna `REMOTE MAC`. O `MAC REMOTO` deve combinar o MAC real do destino remoto a que DLSw fornece a Conectividade.

A segunda entrada está em um estado `passivo`, assim que significa que o segundo roteador está vivo e fornece sua própria tradução. No caso de uma falha, Avimimus toma sobre essa

responsabilidade.

Slave DLSw Redundant Router

```
turbo# show dlsw transparent neighbor Interface
Ethernet0 0000.aaaa.0000 SELF Slave 0000.cccc.0000
Connected MASTER. turbo# show dlsw transparent cache
Interface Ethernet0 Circuit Cache local addr(lsap)
remote addr(dsap) state Owner 0000.8888.0000(08)
4000.3745.0000(04) POSITIVE SELF Total number of
circuits in the Cache: 1 turbo# show dlsw transparent
map Interface Ethernet0 LOCAL Mac REMOTE Mac BACKUP ----
-----
0000.eeee.0000 4000.3745.0000
0000.cccc.0000 STATIC 0000.6666.0000 4000.3745.0000
0000.cccc.0000 DYNAMIC(Passive)
```

Troubleshooting

Esta seção fornece informações que podem ser usadas para o troubleshooting da sua configuração.

Comandos para Troubleshooting

Esta seção apresenta todas as saídas de comandos debug desta configuração. O comentário é incluído com a saída do comando para ajudá-lo a encontrar seções importantes.

Nota: Antes de emitir **comandos debug**, refira a [informação importante em comandos Debug](#).

Esta seção é dividida nestas subseções:

- [Saída do comando debug dlsw transparent durante falha simulada do roteador mestre](#)
- [Saída do comando debug dlsw transparent durante a recuperação do roteador mestre](#)
- [Saída do comando debug dlsw transparent durante o estabelecimento de um circuito DLSw do roteador slave](#)

Saída do comando debug dlsw transparent durante falha simulada do roteador mestre

Esta seção apresenta a saída de comando debug dlsw transparent que é gerada durante falhas simuladas do roteador mestre.

- **debug dlsw transparent master** — Os indicadores debugam as mensagens relativas aos eventos mestre-escravos.
- **debug dlsw transparent circuits** — Os indicadores debugam as mensagens relativas ao cache de circuito fechado para o domínio interligado transparente.
- **debug dlsw transparent address-map** — Os indicadores debugam as mensagens relativas aos mapeamentos do MAC address no roteador local e a todos os mapeamentos para que o roteador local atuar como o backup para um par vizinho.

Roteador redundante DLSW mestre

```
avimimus# show debug DLSw: DLSw Ethernet Redundancy -
Master debugging is on DLSw Ethernet Redundancy -
Circuits debugging is on DLSw Ethernet Redundancy -
```

```
Address Map debugging is on Mar 5 18:07:10.039: DLSW-ER:
Sending MP Frame Mar 5 18:07:30.039: DLSW-ER: Sending MP
Frame Mar 5 18:07:40.039: DLSW-ER: Sending MP Frame Mar
5 18:07:50.039: DLSW-ER: Sending MP Frame Mar 5
18:08:00.047: DLSW-ER: Sending MP Frame Mar 5
18:08:10.051: DLSW-ER: Sending MP Frame Mar 5
18:08:20.051: DLSW-ER: Sending MP Frame Mar 5
18:08:30.051: DLSW-ER: Sending MP Frame Mar 5
18:08:40.051: DLSW-ER: Sending MP Frame Mar 5
18:08:50.051: DLSW-ER: Sending MP Frame
```

O roteador redundante DLSW mestre envia a um quadro do master present (MP) os segundos cada 10.

Aqui, uma falha é introduzida quando a interface Ethernet é fechada:

Roteador redundante DLSW mestre

```
avimimus# configure terminal Enter configuration
commands, one per line. End with CNTL/Z.
avimimus(config)# interface ethernet0 avimimus(config-
if)# shut Mar 5 18:09:00.951: %STANDBY-6-STATECHANGE:
Standby: 0: Ethernet0 state Active -> Init Mar 5
18:09:02.951: %LINK-5-CHANGED: Interface Ethernet0,
changed state to administratively down Mar 5
18:09:02.951: DLSW-ER:dm_action_b: LLC2 session dead
freeing neighbor 0000.aaaa.0000 Mar 5 18:09:02.951:
DLSW-ER:Sourcing a TestFrame 0000.6666.0000 -->
0000.aaaa.0000 on Ethernet0 Mar 5 18:09:02.951: DLSW-
ER:Sourcing a TestFrame 0000.eeee.0000 -->
0000.aaaa.0000 on Ethernet0 Mar 5 18:09:02.971: DLSW-
ER:CSM->MS: CG:OK: 4000.3745.0000:4 0000.4444.0000:4 Mar
5 18:09:02.979: DLSW-ER:CSM->MS: CG:OK: 4000.3745.0000:4
0000.4444.0000:8 Mar 5 18:09:03.951: %LINEPROTO-5-
UPDOWN: Line protocol on Interface Ethernet0, changed
state to down
```

Seja aqui debuga tomado do roteador slave durante a falha:

Slave DLSw Redundant Router

```
turbo#
Mar 5 18:09:10.251: %STANDBY-6-STATECHANGE: Standby: 0:
Ethernet0 state Standby -> Active
Mar 5 18:09:10.583: DLSW-ER:dm_action_r: LLC2 session
dead to neighbor 0000.cccc.0000
Mar 5 18:09:10.587: DLSW-ER:Sourcing a TestFrame
0000.eeee.0000 --> 0000.cccc.0000 on Ethernet0
Mar 5 18:09:10.591: DLSW-ER:Sourcing a TestFrame
0000.6666.0000 --> 0000.cccc.0000 on Ethernet0
Mar 5 18:09:10.595: DLSW-ER: Sending MP Frame
Mar 5 18:09:10.595: DLSW-ER:dm_action_u: Freeing current
master 0000.cccc.0000
Mar 5 18:09:10.599: DLSW-ER:dm_action_u: Changing state
to Master
Mar 5 18:09:20.595: DLSW-ER: Sending MP Frame
Mar 5 18:09:30.595: DLSW-ER: Sending MP Frame
```

Na sintaxe anterior, o turbocompressor transforma-se o roteador mestre e igualmente toma-se sobre a tradução que foi configurada no roteador Avimimus.

Seja aqui debuga no turbocompressor quando as estações final que são conectadas através do reattempt de Avimimus a conexão ao host:

```
Slave DLSw Redundant Router
turbo#
Mar 5 18:09:40.595: DLSW-ER: Sending MP Frame
Mar 5 18:09:40.927: DLSW-ER:Replacing dmac
0000.6666.0000 with 4000.3745.0000 on a frame
from Ethernet0
Mar 5 18:09:40.939: DLSW-ER:Replacing dmac
0000.6666.0000 with 4000.3745.0000 on a frame
from Ethernet0
Mar 5 18:09:40.995: DLSW-ER:Replacing dmac
0000.6666.0000 with 4000.3745.0000 on a frame
from Ethernet0
Mar 5 18:09:40.995: DLSW-ER:CSM->MS: C_INQ:NEW:
4000.3745.0000:4 0000.4444.0000:4
Mar 5 18:09:40.999: DLSW-ER:CSM->MS: IW:PENDING:
4000.3745.0000:4 0000.4444.0000:4
```

Na sintaxe anterior, você vê que o turbocompressor termina as traduções requerida (de que foi possuído previamente por Avimimus) para estabelecer a primeira sessão de LLC do controlador A. Essa sessão usa LSAP 0x04 e RSAP 0x04.

```
Slave DLSw Redundant Router
Mar 5 18:09:41.963: DLSW-ER:Replacing dmac
0000.6666.0000 with 4000.3745.0000 on a frame
from Ethernet0
Mar 5 18:09:41.975: DLSW-ER:Replacing dmac
0000.6666.0000 with 4000.3745.0000 on a frame
from Ethernet0
Mar 5 18:09:41.979: DLSW-ER:CSM->MS: C_INQ:NEW:
4000.3745.0000:4 0000.4444.0000:8
Mar 5 18:09:41.983: DLSW-ER:CSM->MS: IW:PENDING:
4000.3745.0000:4 0000.4444.0000:8
```

Na sintaxe anterior, você vê que o turbocompressor termina as traduções requerida (de que foi possuído previamente por Avimimus) para estabelecer a segunda sessão de LLC do controlador A. Esta sessão usa o LSAP 0x08 e o RSAP 0x04.

```
Slave DLSw Redundant Router
Mar 5 18:09:41.991: DLSW-ER:Replacing dmac
0000.6666.0000 with 4000.3745.0000 on a frame
from Ethernet0
Mar 5 18:09:41.995: DLSW-ER:CSM->MS: C_INQ:PENDING:
4000.3745.0000:4 0000.4444.0000:4
Mar 5 18:09:41.999: DLSW-ER:MS->CSM:UGotIt
4000.3745.0000:4 0000.4444.0000:4
```

O primeiro circuito de DLSw foi estabelecido, indicado pela frase de UGotIt.

```
Slave DLSw Redundant Router
Mar 5 18:09:42.003: DLSW-ER:action_a(): target mapped
from (wan) 4000.3745.0000 --->
0000.6666.0000
Mar 5 18:09:42.971: DLSW-ER:Replacing dmac
0000.6666.0000 with 4000.3745.0000 on a frame
from Ethernet0
```

```
Mar 5 18:09:42.975: DLSW-ER:CSM->MS: C_INQ:PENDING:
4000.3745.0000:4 0000.4444.0000:8
Mar 5 18:09:42.983: DLSW-ER:MS->CSM:UGotIt
4000.3745.0000:4 0000.4444.0000:8
```

O segundo circuito de DLSw é terminado.

Slave DLSw Redundant Router

```
Mar 5 18:09:42.987: DLSW-ER:action_a(): target mapped
from (wan) 4000.3745.0000 --->
0000.6666.0000
Mar 5 18:09:50.595: DLSW-ER: Sending MP Frame
```

Estão aqui todos os comandos dlsw que indicam que o turbocompressor tomou sobre Avimimus e que todos os circuitos estão conectados com o turbocompressor:

Slave DLSw Redundant Router

```
turbo# show dlsw transparent cache Interface Ethernet0
Circuit Cache local addr(lsap) remote addr(dsap) state
Owner 0000.4444.0000(04) 4000.3745.0000(04) POSITIVE
SELF 0000.4444.0000(08) 4000.3745.0000(04) POSITIVE SELF
0000.8888.0000(08) 4000.3745.0000(04) POSITIVE SELF
Total number of circuits in the Cache: 3 turbo# show
dlsw transparent map Interface Ethernet0 LOCAL Mac
REMOTE Mac BACKUP -----
0000.eeee.0000 4000.3745.0000 0000.cccc.0000 STATIC
0000.6666.0000 4000.3745.0000 0000.cccc.0000
DYNAMIC(Active)
```

Na sintaxe anterior, você pode ver que o estado da segunda tradução é agora `DYNAMIC(Active)`, que indica que Avimimus deve estar para baixo.

Slave DLSw Redundant Router

```
turbo# show dlsw circuits Index local addr(lsap) remote
addr(dsap) state uptime 4009754676 0000.4444.0000(04)
4000.3745.0000(04) CONNECTED 00:01:05 1610612789
0000.4444.0000(08) 4000.3745.0000(04) CONNECTED 00:01:04
2634022913 0000.8888.0000(08) 4000.3745.0000(04)
CONNECTED 4d01h Total number of circuits connected: 3
```

[Saída do comando debug dlsw transparent durante a recuperação do roteador mestre](#)

Esta seção apresenta o resultado do comando debug dlsw transparent command gerado durante a recuperação do roteador mestre.

Roteador redundante DLSW mestre

```
avimimus# configure terminal Enter configuration
commands, one per line. End with CNTL/Z.
avimimus(config)# interface ethernet0 avimimus(config-
if)# no shut Mar 5 18:12:00.087: DLSW-ER: Sending MP
Frame Mar 5 18:12:03.127: %LINK-3-UPDOWN: Interface
Ethernet0, changed state to up Mar 5 18:12:03.595:
%STANDBY-6-STATECHANGE: Standby: 0: Ethernet0 state
Listen -> Active Mar 5 18:12:04.127: %LINEPROTO-5-
UPDOWN: Line protocol on Interface Ethernet0, changed
state to up Mar 5 18:12:10.087: DLSW-ER: Sending MP
Frame Mar 5 18:12:10.599: DLSW-ER: New neighbor: master
```

```
0000.cccc.0000, neighbor 0000.aaaa.0000 Mar 5
18:12:10.599: DLSW-ER:dm_action_a: Rcvd MP with worse
priority from 0000.aaaa.0000 Mar 5 18:12:10.607: DLSW-
ER:dm_action_l: LLC2 up for neighbor 0000.aaaa.0000 Mar
5 18:12:10.607: DLSW-ER: Sending MC to 0000.aaaa.0000
Mar 5 18:12:10.615: DLSW-ER:dm_action_d: Received MA
from neighbor 0000.aaaa.0000
```

Na sintaxe anterior, Avimimus recebeu master accept (MA) uma mensagem do turbocompressor, que confirma que Avimimus é agora o mestre novo.

Aqui, os dois Roteadores trocam BACKMEUP_REQ pelos mapeamentos para que é configurado. Cada pedido deve ser seguido por um ACK.

Slave DLSw Redundant Router

```
Mar 5 18:12:10.615: DLSW-ER: Sending BACKMEUP_REQ
0000.6666.0000 --> 4000.3745.0000 to
neighbor 0000.aaaa.0000 (617321C8)
Mar 5 18:12:10.615: DLSW-ER: Sending DN to
0000.aaaa.0000
Mar 5 18:12:10.623: DLSW-ER:Rcvd BACKMEUP_REQ from
0000.aaaa.0000 for map entry 0000.eeee.0000
--> 4000.3745.0000
Mar 5 18:12:10.623: DLSW-ER: Sending BACKMEUP_ACK
0000.eeee.0000 --> 4000.3745.0000 to
neighbor 0000.aaaa.0000 (617321C8)
Mar 5 18:12:10.651: DLSW-ER:dm_action_f: Rcvd CO from
0000.aaaa.0000
Mar 5 18:12:10.667: DLSW-ER:Rcvd BACKMEUP_REQ from
0000.aaaa.0000 for mapentry 0000.eeee.0000
--> 4000.3745.0000
Mar 5 18:12:10.667: DLSW-ER: Sending BACKMEUP_ACK
0000.eeee.0000 --> 4000.3745.0000 to
neighbor 0000.aaaa.0000 (617321C8)
Mar 5 18:12:19.731: DLSW-ER:dm_action_h: Rcvd CG <-
0000.aaaa.0000 4000.3745.0000:4 0000.4444.0000:4
Mar 5 18:12:19.735: DLSW-ER:dm_action_h: Rcvd CG <-
0000.aaaa.0000 4000.3745.0000:4 0000.4444.0000:8
Mar 5 18:12:20.087: DLSW-ER: Sending MP Frame
Mar 5 18:12:20.647: DLSW-ER:Rcvd BACKMEUP_ACK from
0000.aaaa.0000 for mapentry 0000.6666.0000
--> 4000.3745.0000
Mar 5 18:12:20.647: DLSW-ER:Sourcing a TestFrame
0000.6666.0000 --> 0000.aaaa.0000 on Ethernet0
Mar 5 18:12:30.087: DLSW-ER: Sending MP Frame
```

Seja aqui debuga tomado do turbocompressor quando o roteador mestre Avimimus vem acima:

Slave DLSw Redundant Router

```
turbo#
Mar 5 18:12:00.595: DLSW-ER: Sending MP Frame
Mar 5 18:12:03.603: %STANDBY-6-STATECHANGE: Standby: 0:
Ethernet0 state Active -> Speak
Mar 5 18:12:10.087: DLSW-ER: New neighbor: master
0000.aaaa.0000, neighbor 0000.cccc.0000
Mar 5 18:12:10.091: DLSW-ER:dm_action_o: Rcvd MP with
better priority: 0000.cccc.0000
Mar 5 18:12:10.595: DLSW-ER: Sending MP Frame
Mar 5 18:12:10.611: DLSW-ER:dm_action_l: LLC2 up for
neighbor 0000.cccc.0000
```

```
Mar 5 18:12:10.611: DLSW-ER: Sending MA to
0000.cccc.0000
```

O turbocompressor reconhece a mensagem de Avimimus, e então o turbocompressor transforma-se o escravo.

Slave DLSw Redundant Router

```
Mar 5 18:12:10.615: DLSW-ER: Sending BACKMEUP_REQ
0000.eeee.0000 --> 4000.3745.0000
to neighbor 0000.cccc.0000 (45B47C)
Mar 5 18:12:10.623: DLSW-ER:dm_action_c: Rcvd MC with
better priority from 0000.cccc.0000
Mar 5 18:12:10.627: DLSW-ER:dm_action_c: Changing state:
Master to Slave
```

O turbocompressor transforma-se o roteador slave.

Slave DLSw Redundant Router

```
Mar 5 18:12:10.627: DLSW-ER:Rcvd BACKMEUP_REQ from
0000.cccc.0000 for mapentry 0000.6666.0000
--> 4000.3745.0000
Mar 5 18:12:10.635: DLSW-ER:calling admin_stop for
ckt(0000.4444.0000(4) 4000.3745.0000(4)) with
lmac 0000.6666.0000
Mar 5 18:12:10.643: DLSW-ER:calling admin_stop for
ckt(0000.4444.0000(8) 4000.3745.0000(4)) with
lmac 0000.6666.0000
```

Na sintaxe anterior, o turbocompressor termina os circuitos de DLSw que devem ser possuídos por Avimimus. (Os circuitos devem ser rasgados abaixo de quando o mestre vier apoio.) Esta terminação acontece depois que o turbocompressor recebe o BACKMEUP_REQ de Avimimus.

Slave DLSw Redundant Router

```
Mar 5 18:12:10.643: DLSW-ER: dm_action_n: Rcvd DN frame
from 0000.cccc.0000
Mar 5 18:12:10.647: DLSW-ER:Sending CO frame # 0 to
0000.cccc.0000
Mar 5 18:12:10.651: DLSW-ER:Rcvd BACKMEUP_ACK from
0000.cccc.0000 for mapentry 0000.eeee.0000
--> 4000.3745.0000
Mar 5 18:12:10.655: DLSW-ER:Sourcing a TestFrame
0000.eeee.0000 --> 0000.cccc.0000 on Ethernet0
Mar 5 18:12:10.659: DLSW-ER:dm_action_s: LLC2 session up
to neighbor 0000.cccc.0000
Mar 5 18:12:10.659: DLSW-ER: Sending BACKMEUP_REQ
0000.eeee.0000 --> 4000.3745.0000
to neighbor 0000.cccc.0000 (45B47C)
Mar 5 18:12:10.671: DLSW-ER:Rcvd BACKMEUP_ACK from
0000.cccc.0000 for mapentry 0000.eeee.0000
--> 4000.3745.0000
Mar 5 18:12:10.675: DLSW-ER:Sourcing a TestFrame
0000.eeee.0000 --> 0000.cccc.0000 on Ethernet0
Mar 5 18:12:13.603: %STANDBY-6-STATECHANGE: Standby: 0:
Ethernet0 state Speak -> Standby
Mar 5 18:12:19.723: DLSW-ER:CG -> 0000.cccc.0000:
4000.3745.0000:4 0000.4444.0000:4
Mar 5 18:12:19.727: DLSW-ER:CSM->MS: CG:OK:
4000.3745.0000:4 0000.4444.0000:4
Mar 5 18:12:19.731: DLSW-ER:CG -> 0000.cccc.0000:
```

```

4000.3745.0000:4 0000.4444.0000:8
Mar 5 18:12:19.735: DLSW-ER:CSM->MS: CG:OK:
4000.3745.0000:4 0000.4444.0000:8
Mar 5 18:12:20.643: DLSW-ER: Sending BACKMEUP_ACK
0000.6666.0000 --> 4000.3745.0000
  to neighbor 0000.cccc.0000 (45B47C)

turbo# show dlsw circuits Index local addr(lsap) remote
addr(dsap) state uptime 2634022913 0000.8888.0000(08)
4000.3745.0000(04) CONNECTED 4d01h Total number of
circuits connected: 1

```

São apresentados na sintaxe aqui debuga que ocorre quando as estações final tentam restabelecer a conexão. Avimimus é alternativo servir como o roteador mestre.

Roteador redundante DLSW mestre

```

avimimus#
Mar 5 18:12:40.071: DLSW-ER:Replacing dmac
0000.6666.0000 with 4000.3745.0000 on a frame
  from Ethernet0
Mar 5 18:12:40.071: DLSW-ER:Replacing dmac
0000.6666.0000 with 4000.3745.0000 on a frame
  from Ethernet0
Mar 5 18:12:40.079: DLSW-ER:Replacing dmac
0000.6666.0000 with 4000.3745.0000 on a frame
  from Ethernet0
Mar 5 18:12:40.079: DLSW-ER:CSM->MS: C_INQ:NEW:
4000.3745.0000:4 0000.4444.0000:4
Mar 5 18:12:40.079: DLSW-ER:CSM->MS: IW:PENDING:
4000.3745.0000:4 0000.4444.0000:4
Mar 5 18:12:40.087: DLSW-ER: Sending MP Frame
Mar 5 18:12:41.071: DLSW-ER:Replacing dmac
0000.6666.0000 with 4000.3745.0000 on a frame
  from Ethernet0
Mar 5 18:12:41.075: DLSW-ER:Replacing dmac
0000.6666.0000 with 4000.3745.0000 on a frame
  from Ethernet0
Mar 5 18:12:41.075: DLSW-ER:CSM->MS: C_INQ:NEW:
4000.3745.0000:4 0000.4444.0000:8
Mar 5 18:12:41.075: DLSW-ER:CSM->MS: IW:PENDING:
4000.3745.0000:4 0000.4444.0000:8
Mar 5 18:12:41.079: DLSW-ER:MS->CSM:UGotIt
4000.3745.0000:4 0000.4444.0000:4
Mar 5 18:12:41.079: DLSW-ER:action_a(): target mapped
from (wan) 4000.3745.0000 --->
  0000.6666.0000
Mar 5 18:12:42.075: DLSW-ER:MS->CSM:UGotIt
4000.3745.0000:4 0000.4444.0000:8
Mar 5 18:12:42.075: DLSW-ER:action_a(): target mapped
from (wan) 4000.3745.0000 --->
  0000.6666.0000

avimimus# show dlsw circuits Index local addr(lsap)
remote addr(dsap) state uptime 3070230625
0000.4444.0000(04) 4000.3745.0000(04) CONNECTED 00:00:08
4194304098 0000.4444.0000(08) 4000.3745.0000(04)
CONNECTED 00:00:08 Total number of circuits connected: 2

```

[Saída do comando debug dlsw transparent durante o estabelecimento de um circuito DLSw do roteador slave](#)

Esta seção apresenta o **comando debug dlsw transparent** output que é gerado quando o roteador slave tenta trazer acima um circuito de DLSw.

É útil examinar a comunicação entre o escravo e o mestre antes que o escravo receba a permissão aceitar o circuito.

A encenação é simulada quando o circuito existente de DLSw é cancelado no roteador slave.

```
Slave DLSw Redundant Router
turbo# show dlsw circuits Index local addr(lsap) remote
addr(dsap) state uptime 2634022913 0000.8888.0000(08)
4000.3745.0000(04) CONNECTED 4d02h Total number of
circuits connected: 1 turbo# clear dlsw circuits
2634022913 turbo# Mar 5 20:02:37.426: DLSW-ER:CG ->
0000.cccc.0000: 4000.3745.0000:4 0000.8888.0000:8 Mar 5
20:02:37.430: DLSW-ER:CSM->MS: CG:OK: 4000.3745.0000:4
0000.8888.0000:8
```

O turbocompressor diz o roteador mestre para cancelar sua entrada de cache para o circuito apenas-apagado.

```
Slave DLSw Redundant Router
Mar 5 20:03:07.398: DLSW-ER:Replacing dmac
0000.eeee.0000 with 4000.3745.0000 on a frame
from Ethernet0
Mar 5 20:03:07.462: DLSW-ER:Replacing dmac
0000.eeee.0000 with 4000.3745.0000 on a frame
from Ethernet0
Mar 5 20:03:07.466: DLSW-ER:CSM->MS: C_INQ:NEW:
4000.3745.0000:4 0000.8888.0000:8
Mar 5 20:03:07.470: DLSW-ER:IW -> 0000.cccc.0000:
4000.3745.0000:4 0000.8888.0000:8
```

O turbocompressor recebe uma requisição recebida do dispositivo final conectar ao host final. Em 20:03:07.470, o turbocompressor notifica o roteador mestre deste pedido.

```
Slave DLSw Redundant Router
Mar 5 20:03:07.474: DLSW-ER:CSM->MS: IW:PENDING:
4000.3745.0000:4 0000.8888.0000:8
Mar 5 20:03:08.458: DLSW-ER:Replacing dmac
0000.eeee.0000 with 4000.3745.0000 on a frame
from Ethernet0
Mar 5 20:03:08.462: DLSW-ER:CSM->MS: C_INQ:PENDING:
4000.3745.0000:4 0000.8888.0000:8
Mar 5 20:03:08.474: DLSW-ER:dm_action_k: Rcvd UG for
4000.3745.0000:4 0000.8888.0000:8
Mar 5 20:03:08.478: DLSW-ER:action_a(): target mapped
from (wan) 4000.3745.0000 --->
0000.eeee.0000
```

O turbocompressor obtém um UG de seu mestre, e o circuito é trazido acima.

```
Slave DLSw Redundant Router
turbo# show dlsw circuits Index local addr(lsap) remote
addr(dsap) state uptime 385876023 0000.8888.0000(08)
4000.3745.0000(04) CONNECTED 00:00:33 Total number of
circuits connected: 1
```

Estão aqui os dados tomados no roteador mestre então o escravo tenta trazer acima o circuito.

Roteador redundante DLSW mestre

```
avimimus# show dlsw circuits Index local addr(lsap)
remote addr(dsap) state uptime 3070230625
0000.4444.0000(04) 4000.3745.0000(04) CONNECTED 01:49:13
4194304098 0000.4444.0000(08) 4000.3745.0000(04)
CONNECTED 01:49:13 Total number of circuits connected: 2
avimimus# show dlsw transparent cache Interface
Ethernet0 Circuit Cache local addr(lsap) remote
addr(dsap) state Owner 0000.4444.0000(04)
4000.3745.0000(04) POSITIVE SELF 0000.4444.0000(08)
4000.3745.0000(04) POSITIVE SELF 0000.8888.0000(08)
4000.3745.0000(04) NEGATIVE 0000.aaaa.0000 Total number
of circuits in the Cache: 3 Mar 5 20:02:37.433: DLSW-
ER:dm_action_h: Rcvd CG <- 0000.aaaa.0000
4000.3745.0000:4 0000.8888.0000:8 Mar 5 20:02:41.409:
DLSW-ER: Sending MP Frame Mar 5 20:02:51.409: DLSW-ER:
Sending MP Frame Mar 5 20:03:01.417: DLSW-ER: Sending MP
Frame Mar 5 20:03:07.473: DLSW-ER:dm_action_j: Rcvd IW
<- 0000.aaaa.0000 4000.3745.0000:4 0000.8888.0000:8 Mar
5 20:03:08.473: DLSW-ER:UG -> 0000.aaaa.0000:
4000.3745.0000:4 0000.8888.0000:8 Mar 5 20:03:11.421:
DLSW-ER: Sending MP Frame Mar 5 20:03:21.421: DLSW-ER:
Sending MP Frame Mar 5 20:03:31.421: DLSW-ER: Sending MP
Frame Mar 5 20:03:41.421: DLSW-ER: Sending MP Frame Mar
5 20:03:51.421: DLSW-ER: Sending MP Frame avimimus#un
all
```

Informações Relacionadas

- [Redundância Ethernet DLSw+](#)
- [Suporte Técnico - Cisco Systems](#)