

# Troubleshooting DLSw: Ethernet and Qualified Logical Link Control

Document ID: 12490

---

## Introduction

### Prerequisites

- Requirements
- Components Used
- Conventions

### Ethernet

### QLLC

- QLLC Implementation Overview and Message Flows
- PU 2.0 Normal QLLC Connection Initiated by X.25 Device
- PU 2.0 Normal QLLC Connection Initiated by a LAN PU 2.0 Device to FEP Running NCP Packet Switching Interface
- PU 2.1 Normal QLLC Connection Initiated by an X.25 Device
- PU 2.1 QLLC Connection Initiated by the LAN Device
- DLSw/SDLC over QLLC Sample Configuration and Debug
- Troubleshooting Steps
- QLLC Debugs

### NetPro Discussion Forums – Featured Conversations

### Related Information

---

## Introduction

This document explains how to implement Qualified Logical Link Control (QLLC) in Cisco routers and message flows, for a call connection in a topology where a front-end processor (FEP) is connected through Ethernet and where remote devices (either physical unit [PU] type 2.0 or PU type 2.1) are connected to the X.25 network. It also covers appropriate steps to troubleshoot this type of call connection.

## Prerequisites

### Requirements

There are no specific requirements for this document.

### Components Used

This document is not restricted to specific software or hardware versions.

### Conventions

For more information on document conventions, refer to the Cisco Technical Tips Conventions.

## Ethernet

When you are troubleshooting an Ethernet-attached device that communicates through data-link switching (DLSw), the first thing that you need to verify is that **dlsw bridge-group *x*** exists, where *x* refers to the bridge

number that is configured in the **bridge-group** command on the Ethernet interface. To verify your configuration, refer to Basic DLSw+ Configurations for sample configurations on Ethernet-attached devices.

Another useful troubleshooting command is **show bridge**, which verifies that the transparent bridge knows about the MAC address of the device, both local and remote. Ethernet MAC addresses appear in canonical format, unlike Token Ring addresses, which have a non-canonical format. Use the following guideline to translate MAC addresses:

Ethernet MAC Address (canonical format)	0 1 2 3 4 5 6 7 8 9 A B C D E F
becomes	
Token Ring Address (non-canonical format)	0 8 4 C 2 A 6 E 1 9 5 D 3 B 7 F

This is an example, on Ethernet, that follows that rule:

1. Ethernet MAC Address (canonical format)	0200.4556.1140
2. Intermediate step	0400.2AA6.8820
3. Final Token Ring Address (non-canonical format)	4000.A26A.8802

**Note:** To arrive at the final, non-canonical address, you swap around each bit within a byte. For more information on bitswapping, refer to the [Bitswapping Tool](#) (registered customers only).

Compare entries that are found in the **show bridge** command output with entries that are found in the **show dlsw reachability** command output. Remember that entries in the **show dlsw reachability** command output appear in non-canonical format, as opposed to canonical format as on Ethernet or in the **show bridge** command output.

For general Ethernet troubleshooting, refer to [Troubleshooting Ethernet](#).

## QLLC

**Note:** The Document Contents section of this document series shows all of the sections of the series, to aid navigation.

### QLLC Implementation Overview and Message Flows

QLLC commands are implemented in X.25 packets with the use of the Q-bit. X.25 packets that contain QLLC primitives are typically five bytes, or the length of the X.25 packet header plus two bytes of QLLC control information.

**Note:** X.25 data packets that contain Systems Network Architecture (SNA) data do not use the Q-bit.

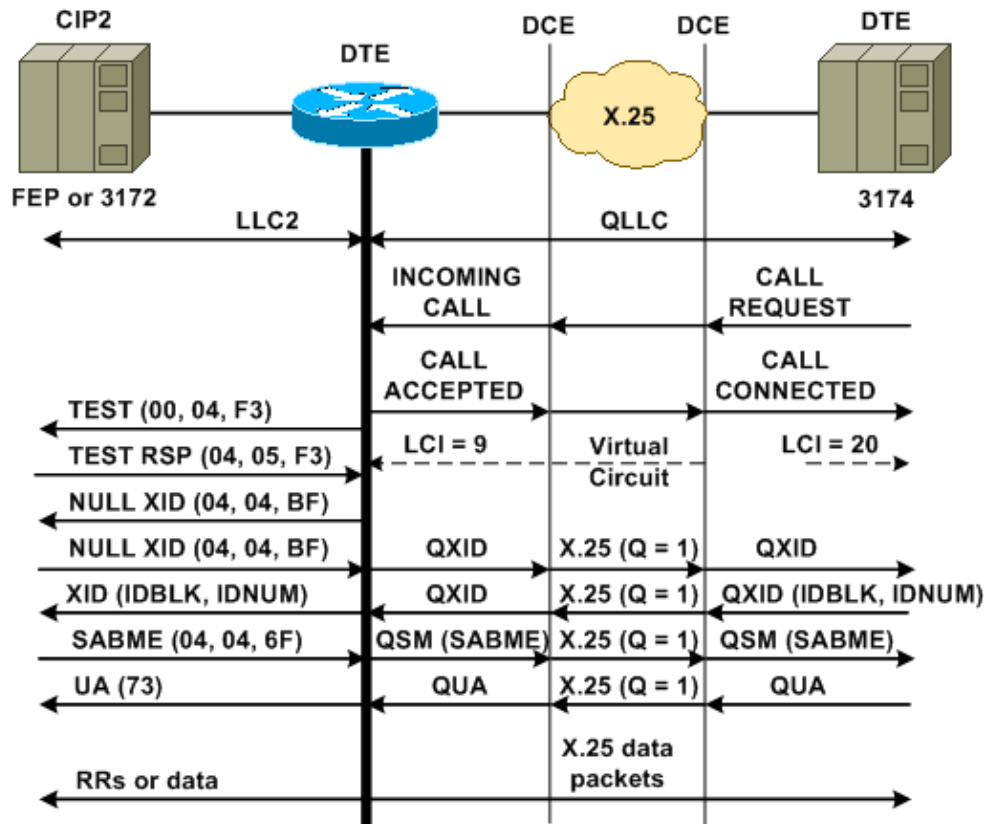
After the QLLC connection is established, the unique virtual circuit of the X.25 connection is used to forward data traffic. Logical Link Control (LLC) is a subset of High-Level Data Link Control (HDLC). Synchronous Data Link Control (SDLC) and QLLC are also subsets of HDLC. Cisco converts these QLLC primitives to LLC primitives, and vice versa:

QLLC	LLC
------	-----

QSM	SABME
QXID	XID
QDISC	DISC
QUA	UA
X.25 DATA PACKET	I-FRAME

## PU 2.0 Normal QLLC Connection Initiated by X.25 Device

Figure 1 QLLC Flows for PU 2.0



A normal QLLC/LLC connection is initiated with the receipt of an X.25 INCOMING CALL, which contains the QLLC Call User Data (CUD) (0xc3). A reverse QLLC connection is a QLLC/LLC connection initiated by a LAN.

**Note:** For a QLLC/LLC connection, there is a QLLC connection between the QLLC device and the router, and an LLC connection between the LAN-attached device and the router.

Figure 1 shows this sequence:

1. An X.25 QLLC incoming call is answered with an X.25 CALL CONNECTED by the router.
2. The router then sends a TEST frame (or explorer) to the LAN device, to initiate the LAN connection.
3. If the LAN partner can be located, the LAN partner sends an explorer response with a Routing Information Field (RIF) that explains how the LAN partner can be found.
4. The router then sends a null exchange identification (XID) to the LAN partner, under the assumption that the QLLC device can perform XID negotiation. (Most SNA devices can perform XID negotiation.) If the QLLC device can not perform the negotiation by itself, the router offers an XID proxy utility.

5. The QLLC device sends an XID with an IDBLK and IDNUM that is compared against the IDNUM and IDBLK that are configured on the host (switched major node PU).
6. If the IDs match, then the host sends a Set Asynchronous Balanced Mode Extended (SABME).
7. The SABME is converted into a Qualified Setresponse Mode (QSM), and the QLLC device sends a Qualified Unnumbered Acknowledgement (QUA).
8. This QUA is converted into an LLC Unnumbered Acknowledgement (UA) and it is sent to the LAN partner.

At this point, a QLLC connection exists between the QLLC device and the router, an LLC connection exists between the router and the LAN device, and an active QLLC/LLC connection exists on the router.

## **PU 2.0 Normal QLLC Connection Initiated by a LAN PU 2.0 Device to FEP Running NCP Packet Switching Interface**

In a Token Ring or remote source–route bridging (RSRB) environment, this sequence occurs:

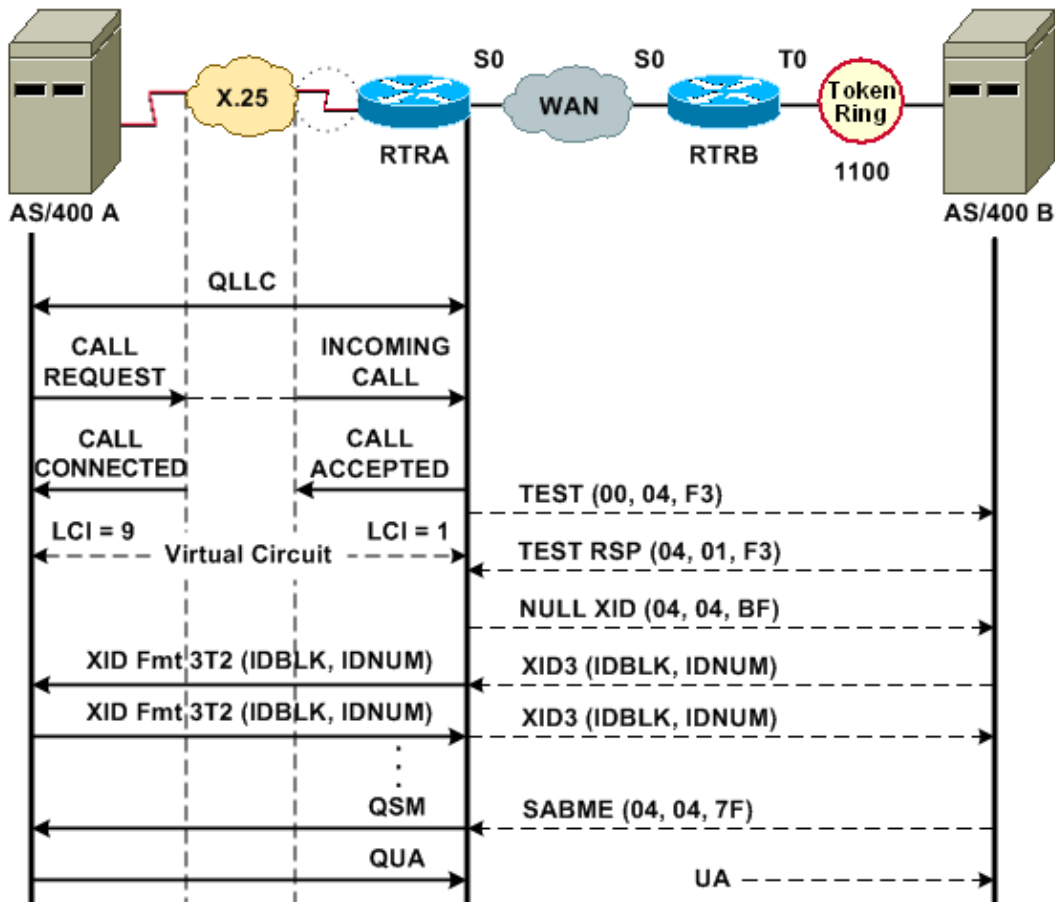
1. The LAN–attached device starts up and sends a test upstream. Then, it sends a null XID packet upstream.
2. If QLLC forwards this null XID to an X.25–attached FEP, the FEP responds as if it were connecting to a PU 2.1 device and aborts the connection, when the PU 2.0 device next sends an XID Format 0 Type 2.
3. The **qlc npsi–poll** command intercepts any null XID packet that the Cisco IOS® software receives on the LAN interface, and it returns a null XID response to the downstream device. The **qlc npsi–poll** command continues to allow XID Format 3 and XID Format 0 packets through the X.25 device.
4. The router sends a CALL REQUEST packet to initiate the X.25 connection, and it receives the CALL ACCEPTED packet in response.
5. The PU 2.0 SNA device sends an XID with an IDBLK and IDNUM that is compared against the IDBLK and IDNUM that is configured on the host (switched major node PU).
6. If the IDs match, the host sends a QSM. The QSM is converted into a SABME.
7. The LAN device responds with a UA, which is converted into a QUA and sent to the FEP.

At this point, there is:

- A QLLC connection between the QLLC device and the router
- An LLC connection between the router and the LAN device
- An active QLLC/LLC connection on the router

## **PU 2.1 Normal QLLC Connection Initiated by an X.25 Device**

**Figure 2 QLLC Flows for PU 2.1**



A normal QLLC/LLC connection is initiated with the receipt of an X.25 INCOMING CALL that contains the QLLC CUD (0xc3). A reverse QLLC connection is a QLLC/LLC connection that is initiated by a LAN.

Figure 2 shows this sequence:

1. An X.25 QLLC incoming call is answered with an X.25 CALL CONNECTED by the router.
2. The router sends a TEST frame (or explorer) to the LAN device, to initiate the LAN connection.
3. If the LAN partner can be located, the LAN partner sends an explorer response, with a RIF that explains how it can be found.
4. The router then sends a null XID to the LAN partner, under the assumption that the QLLC device can perform XID negotiation. (Most SNA devices can perform XID negotiation.) If the QLLC device can not perform the negotiation by itself, the router offers an XID proxy utility.
5. The PU 2.1 devices exchange XID3s until they agree on the primary and secondary roles and other PU 2.1 parameters.
6. The PU 2.1 node that becomes the primary establishes the link-level connection with its PU 2.1 partner.
7. SABME is converted into a QSM, and QUA to a UA.

## PU 2.1 QLLC Connection Initiated by the LAN Device

1. The PU 2.1 LAN starts up and sends a test frame. When it receives a test response from the router, it starts to send an XID3 (or a null XID followed by an XID3).
2. The router sends a CALL REQUEST packet to establish the X.25 connection. From this point on, it translates all of the messages that are exchanged between the two PU 2.1 nodes from LLC2 into X.25.
3. The PU 2.1 devices exchange XID3s until they agree on the primary and secondary roles and other PU 2.1 parameters.
4. The PU 2.1 node that becomes the primary establishes the link-level connection with its PU 2.1

partner.

5. SABME is converted into a QSM, and QUA to a UA.

At this point, there is:

- A QLLC connection between the QLLC device and the router
- An LLC connection between the router and the LAN device
- An active QLLC/LLC connection on the router

## DLSw/SDLC over QLLC Sample Configuration and Debug

There are major differences between RSRB over QLLC and DLSw over QLLC. Perhaps the most important one is that there is a uniform interface (Cisco Link Services [CLS]) between DLSw and the various Data-Link Controls (DLCs) that are available.

Before you attempt any of the **debug** commands in this document, refer to Important Information on Debug Commands.

When you are troubleshooting on the QLLC router, output from these **debug** commands is recommended:

- **debug dlsw core message**
- **debug cls message**
- **debug x25 event**
- **debug qlc state**
- **debug qlc packet**

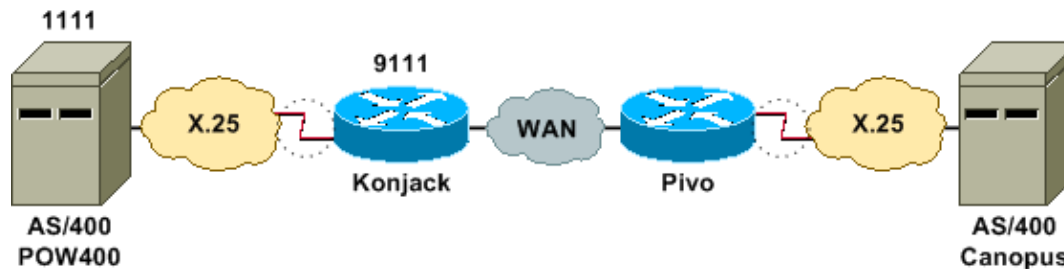
Output from these **show** commands is also useful:

- **show cls**
- **show qlc**

On the SDLC/DLSw Peer Router, these **debug** commands are useful:

- **debug dlsw core message**
- **debug cls message**

Figure 3 QLLC/DLSw Configuration and Debugs



This network diagram uses these configurations:

- Konjack
- Pivo

Konjack

```

X25 routing
dlsw local-peer peer-id 10.3.2.7
dlsw remote-peer 0 tcp 10.3.2.8
!
interface Serial3
 encapsulation x25 dce
 x25 address 9111
 x25 ltc 10
 x25 htc 4095
 x25 map qllc 4000.0000.1111 1111
 clockrate 19200
 qllc dlsw vmacaddr 4000.0000.1111 partner 4000.0000.2222

```

### Pivo

```

x25 routing
!
dlsw local-peer peer-id 10.3.2.8
dlsw remote-peer 0 tcp 10.3.2.7
!
interface serial 0
 no ip address
 encapsulation x25 dce
 x25 address 4444
 x25 map qllc 4000.0000.2222 4444
 qllc dlsw vmac 4000.0000.2222 partner 4000.0000.1111

```

Figure 3 illustrates how two IBM AS/400 Servers can communicate through QLLC/DLSw. vmacaddr 4000.0000.1111 is the MAC address associated to the AS/400 (POW400), and partner 4000.0000.2222 is the MAC address associated to the remote AS/400 (Canopus).

For more information on the **qlc dlsw** command, refer to the DLSw+ Configuration Commands.

The TEST.STN REQ from DLSw to QLLC should result in a TEST.STN.IND packet, and the REQ OPEN STN REQ packet should result in a CALL REQUEST.

The next sample output shows the debug output with annotation. These **debug** commands were issued:

- **debug dlsw core message**
- **debug cls message**
- **debug qlc state**
- **debug qlc packet**
- **debug x25 event**

Konjack#

```
%DLSWC-3-RECVSSP: SSP OP = 3( CUR ) -explorer from peer 10.3.2.8(2065)
```

```
!--- CUR_ex [Can You Reach (explorer)] is received from the peer.
!--- (Note the -explorer.) DLSw starts to explore.
```

```

00:27:26: DLSW: DISP Sent : CLSI Msg : TEST_STN.Req  dlen: 46
00:27:26: (DLSWDLU:DLU-->SAP):
00:27:26: TEST_STN.Req to pSAP: 0x5C733C sel: LLC hlen: 40, dlen: 46
00:27:26: DLSW: DISP Sent : CLSI Msg : TEST_STN.Req  dlen: 46
00:27:26: (DLSWDLU:DLU-->SAP):
00:27:26: TEST_STN.Req to pSAP: 0x5C74A0 sel: LLC hlen: 40, dlen: 46
00:27:26: DLSW: DISP Sent : CLSI Msg : TEST_STN.Req  dlen: 46
00:27:26: (DLSWDLU:DLU-->SAP):
00:27:26: TEST_STN.Req to pSAP: 0x5C7924 sel: LLC hlen: 40, dlen: 46

```

```

!--- There is a match on the destination MAC address in QLLC.

00:27:26: (DLSWDLU:CLS-->DLU):
00:27:26: TEST_STN.Ind to uSAP: 0x5C78BC sel: LLC hlen: 36, dlen: 35
00:27:26: DLSW Received-ctlQ : CLSI Msg : TEST_STN.Ind dlen: 35

!--- DLSw sends an ICR_ex [I Can Reach (explorer)] to the peer.

%DLSWC-3-RECVSSP: SSP OP = 3( CUR ) from peer 10.3.2.8(2065)

!--- CUR_cs [Can You Reach (circuit setup)] is received from the peer.

00:27:26: DISP Sent : CLSI Msg : REQ_OPNSTN.Reg dlen: 102

!--- DLSw sends the CLS message Request Open Station Request to QLLC.

00:27:26: (DLSWDLU:DLU-->SAP):
00:27:26: REQ_OPNSTN.Reg to pSAP: 0x5C7924 sel: LLC hlen: 48, dlen: 102

!--- QLLC places the call to the AS/400.

00:27:26: Serial3: X25 O P3 CALL REQUEST (13) 8 lci 10
00:27:26: From(4): 9111 To(4): 1111
00:27:26: Facilities: (0)
00:27:26: Call User Data (4): 0xC3000000 (qllc)

!--- QLLC X.25 FSM handling Request Open Station Request
!--- Output: Issues CALL REQUEST (see above),
!--- Nothing to CLS/DLSw
!--- Starts a 10000 msec timer
!--- Enters State P2 (see X.25 standard)

00:27:26: QLLC-XFSM state P1, input QX25ReqOpenStnReq: (CallReq,-,XGo 10000) ->P2/D2

!--- QLLC receives CALL ACCEPT from the AS/400.

00:27:26: Serial3: X25 I P3 CALL CONNECTED (9) 8 lci 10
00:27:26: From(4): 9111 To(4): 1111
00:27:26: Facilities: (0)

!--- QLLC X.25 FSM handling CALL ACCEPT
!--- Output: Nothing to X.25
!--- Request Open Station Confirm to CLS/DLSw
!--- Stops Timer
!--- Enters State P4/D1

00:27:26: QLLC-XFSM state P2/D2, input QX25CallConfirm:
(-,ReqOpenStnConf,xStop) ->P4/D1
00:27:26: QLLC: Serial3 I: QXID-CMD 0 bytes

!--- QLLC Logical FSM Receives XID, send ID Indication to DLSw

00:27:26: QLLC-LFSM state QLClosed, input QLXID: (-,IdInd,LGo 3000)
00:27:26: (DLSWDLU:CLS-->DLU):
00:27:26: REQ_OPNSTN.Cfm(CLS_OK) to uCEP: 0x5CA310 sel: LLC hlen: 48, dlen: 102
00:27:26: (DLSWDLU:CLS-->DLU):
00:27:26: ID.Ind to uCEP: 0x5CA310 sel: LLC hlen: 40, dlen: 15
00:27:26: DLSW Received-ctlQ : CLSI Msg : REQ_OPNSTN.Cfm CLS_OK dlen: 102

!--- DLSw receives Request Open Station Confirm from QLLC.

%DLSWC-3-SENDSSP: SSP OP = 4( ICR ) to peer 10.3.2.8(2065) success

!--- DLSw sends ICR_cs [I Can Reach (circuit setup)] to the peer.

%DLSWC-3-SENDSSP: SSP OP = 4( ICR ) to peer 10.3.2.8(2065) success

```

```
!--- DLSw receives ID.Ind from QLLC.

00:27:26: DLSW Received-ctlQ : CLSI Msg : ID.Ind dlen: 15

!--- DLSw receives Reach ACK from the peer.

%DLSWC-3-RECVSSP: SSP OP = 5( ACK ) from peer 10.3.2.8(2065)

!--- DLSw receives XID from the peer.

%DLSWC-3-RECVSSP: SSP OP = 7( XID ) from peer 10.3.2.8(2065)

!--- DLSw sends ID.Reg to QLLC.

00:27:26: DISP Sent : CLSI Msg : ID.Reg dlen: 12
00:27:26: (DLSWDLU:DLU-->CEP):
00:27:26: ID.Reg to pCEP: 0x4C51CC sel: LLC hlen: 40, dlen: 12
00:27:26: QLLC: Serial3 O: QXID-RSP 0 bytes

!--- QLLC Logical FSM Handling ID.Reg from CLS/DLSw.
!--- Output: QLLC XID to X.25
!--- Nothing to CLS
!--- No Timer Action

00:27:26: QLLC-LFSM state QLClosed, input CLSXID: (XId,-,-)

!--- QLLC Receives XID from X.25

00:27:26: QLLC: Serial3 I: QXID-CMD 77 bytes Fmt 3T2: 056B4532
00:27:26: QLLC-LFSM state QLClosed, input QLXID: (-,IdInd,LGo 3000)
00:27:26: (DLSWDLU:CLS-->DLU):
00:27:26: ID.Cfm(CLS_OK) to uCEP: 0x5CA310 sel: LLC hlen: 40, dlen: 92

!--- DLSw receives ID Confirm from QLLC.

00:27:26: DLSW Received-ctlQ : CLSI Msg : ID.Cfm CLS_OK dlen: 92

!--- DLSw sends XID to the peer.

%DLSWC-3-SENDSSP: SSP OP = 7( XID ) to peer 10.3.2.8(2065) success

!--- DLSw receives XID from the peer.

%DLSWC-3-RECVSSP: SSP OP = 7( XID ) from peer 10.3.2.8(2065)
00:27:27: DISP Sent : CLSI Msg : ID.Reg dlen: 89
00:27:27: (DLSWDLU:DLU-->CEP):
00:27:27: ID.Reg to pCEP: 0x4C51CC sel: LLC hlen: 40, dlen: 89
00:27:27: QLLC: Serial3 O: QXID-RSP 77 bytes Fmt 3T2: 05627844
00:27:27: QLLC-LFSM state QLClosed, input CLSXID: (XId,-,-)
00:27:27: QLLC: Serial3 I: QXID-CMD 77 bytes Fmt 3T2: 056B4532

!--- QLLC Logical FSM Handling ID.Reg from CLS.
!--- Output: Nothing to CLS
!--- QLLC XID to X.25
!--- Timer started for 3000 msec

00:27:27: QLLC-LFSM state QLClosed, input QLXID: (-,IdInd,LGo 3000)

!--- More XID negotiation.

00:27:27: (DLSWDLU:CLS-->DLU):
00:27:27: ID.Cfm(CLS_OK) to uCEP: 0x5CA310 sel: LLC hlen: 40, dlen: 92
00:27:27: DLSW Received-ctlQ : CLSI Msg : ID.Cfm CLS_OK dlen: 92
%DLSWC-3-SENDSSP: SSP OP = 7( XID ) to peer 10.3.2.8(2065) success
%DLSWC-3-RECVSSP: SSP OP = 7( XID ) from peer 10.3.2.8(2065)
```

```
00:27:30: DISP Sent : CLSI Msg : ID.Req  dlen: 12
00:27:30: (DLSWDLU:DLU-->CEP):
00:27:30: ID.Req to pCEP: 0x4C51CC sel: LLC hlen: 40, dlen: 12
00:27:30: QLLC: Serial3 O: QXID-RSP  0 bytes
00:27:30: QLLC-LFSM state QLClosed, input CLSXID: (XId,-,-)
00:27:30: QLLC: Serial3 I: QXID-CMD  77 bytes Fmt 3T2: 056B4532
00:27:30: QLLC-LFSM state QLClosed, input QLXID: (-,IdInd,LGo 3000)
00:27:30: (DLSWDLU:CLS-->DLU):
00:27:30: ID.Cfm(CLS_OK) to uCEP: 0x5CA310 sel: LLC hlen: 40, dlen: 92
00:27:30: DLSW Received-ctlQ : CLSI Msg : ID.Cfm CLS_OK dlen: 92
%DLSWC-3-SENDSSP: SSP OP = 7( XID ) to peer 10.3.2.8(2065) success
%DLSWC-3-RECVSSP: SSP OP = 7( XID ) from peer 10.3.2.8(2065)
00:27:30: DISP Sent : CLSI Msg : ID.Req  dlen: 89
00:27:30: (DLSWDLU:DLU-->CEP):
00:27:30: ID.Req to pCEP: 0x4C51CC sel: LLC hlen: 40, dlen: 89
00:27:30: QLLC: Serial3 O: QXID-RSP  77 bytes Fmt 3T2: 05627844
00:27:30: QLLC-LFSM state QLClosed, input CLSXID: (XId,-,-)
00:27:30: QLLC: Serial3 I: QXID-CMD  77 bytes Fmt 3T2: 056B4532
00:27:30: QLLC-LFSM state QLClosed, input QLXID: (-,IdInd,LGo 3000)
00:27:30: (DLSWDLU:CLS-->DLU):
00:27:30: ID.Cfm(CLS_OK) to uCEP: 0x5CA310 sel: LLC hlen: 40, dlen: 92
00:27:30: DLSW Received-ctlQ : CLSI Msg : ID.Cfm CLS_OK dlen: 92
%DLSWC-3-SENDSSP: SSP OP = 7( XID ) to peer 10.3.2.8(2065) success
%DLSWC-3-RECVSSP: SSP OP = 7( XID ) from peer 10.3.2.8(2065)
00:27:30: DISP Sent : CLSI Msg : ID.Req  dlen: 89
00:27:30: (DLSWDLU:DLU-->CEP):
00:27:30: ID.Req to pCEP: 0x4C51CC sel: LLC hlen: 40, dlen: 89
00:27:30: QLLC: Serial3 O: QXID-RSP  77 bytes Fmt 3T2: 05627844
00:27:30: QLLC-LFSM state QLClosed, input CLSXID: (XId,-,-)
```

*!--- AS/400 becomes primary and sends QSM to QLLC.*

```
00:27:30: QLLC: Serial3 I: QSM
```

*!--- QLLC Logical FSM Handling QSM.*

*!--- Output: Nothing*

*!--- Connect.Ind to CLS/DLSw*

*!--- Start Timer for 3000 msec*

*!--- State QLogical Remote Opening*

```
00:27:30: QLLC-LFSM state QLClosed, input QLSM:
(-,ConnInd,LGo 3000) ->QLRemoteOpening
00:27:30: (DLSWDLU:CLS-->DLU):
00:27:30: CONNECT.Ind to uCEP: 0x5CA310 sel: LLC hlen: 40, dlen: 8
```

*!--- DLSw receives CONNECT.Ind from QLLC and sends CON.Reg to the peer.*

```
00:27:30: DLSW Received-ctlQ : CLSI Msg : CONNECT.Ind  dlen: 8
%DLSWC-3-SENDSSP: SSP OP = 8( CONQ ) to peer 10.3.2.8(2065) success
```

*!--- DLSw receives CON.Response from the peer and sends Connect Response to QLLC.*

```
%DLSWC-3-RECVSSP: SSP OP = 9( CONR ) from peer 10.3.2.8(2065)
00:27:30: DISP Sent : CLSI Msg : CONNECT.Rsp  dlen: 20
00:27:30: (DLSWDLU:DLU-->CEP):
00:27:30: CONNECT.Rsp to pCEP: 0x4C51CC sel: LLC hlen: 42, dlen: 20
```

*!--- QLLC Handling Connect Response from CLS/DLSw.*

*!--- Output: QUA to X.25*

*!--- Conected.Ind to CLS/DLSw*

*!--- State to QLOpened*

```
00:27:30: QLLC: Serial3 O: QUA
00:27:30: QLLC-LFSM state QLRemoteOpening,
input ConnectResponse: (UA,ConnectedInd,lStop) ->QLOpened
00:27:30: (DLSWDLU:CLS-->DLU):
```

```
00:27:30:      CONNECTED.Ind to uCEP: 0x5CA310 sel: LLC hlen: 40, dlen: 8
00:27:30:  DLSW Received-ctlQ : CLSI Msg : CONNECTED.Ind  dlen: 8
```

```
Konjack# show dls reach
```

```
DLSw MAC address reachability cache list
Mac Addr      status      Loc.      peer/port      rif
4000.0000.1111 FOUND      LOCAL    P003-S000      --no rif--
4000.0000.2222 FOUND      REMOTE   10.3.2.8(2065)
```

```
!--- 4000.0000.2222 was the partner.
```

## Troubleshooting Steps

This section details some of the **show** commands that can be executed on the router that is running QLLC/DLSw.

To eliminate the possibility that the problem is hardware-related, issue these commands:

- **show interface serial 0**
- **show controllers serial 0**
- **show controllers cbus**

Check the router configuration: X.121 address, packet size, modulo number, permanent virtual circuits (PVCs), switched virtual circuits (SVCs), and Link Access Protocol Balanced (LAPB) parameters (such as window size and modulo).

- Issue the **show interface serial** command on the X.25 line to look at the status of the line and the protocol. Line down, protocol down (DTR is down).
- Issue the **show controller serial** command and look at the top of the output. Does it show the correct cable?

You should see DCE-RS-232 or DCE-V.35 for DCE routers (the router emulates a modem with the **clockrate** command).

You should see DTE-RS-232 or DTE-V.35 for DTE routers (the router connects to a DCE device, such as a modem or a router that emulates a modem).

Check the connected equipment, including the serial board, modems, remote device, and cabling. When you check cabling, make sure of these points:

- The cable provided by Cisco is connected to the correct interface on the remote device.
- If the router is the DCE, the cable from the router is connected to the cable of the DTE device.
- If the line is up and the protocol is down, determine if the router interface is a DCE or DTE. The DCE provides the clock.
- If the router interface is a DCE, do you have the **clock rate** command configured?
- Have you configured for X.25 encapsulation?
- Issue the **show interface serial 0** command. Is the LAPB state CONNECT?
- Are both sides configured for either half duplex or full duplex?
- If the line is up and the protocol is up, are the X.25 and LAPB configuration parameters correct? These parameters need to match those defined for the X.25 provider.
- Ensure that these X.25 parameters are correct:

- ◆ X.121 address specification
- ◆ Input and output packet sizes (x25 ips and x25 ops) the default is 128 bytes.

- ◆ Window sizes (x25 wout and x25 win) the default is 2.
- ◆ X.25 modulo the default is 8.
- ◆ Check QLLC largest–packet value (the default is 256). This value agrees with the value that is configured in the remote SNA device. The valid range is 0 through 1024.
- Ensure that these LAPB parameters are correct:
  - ◆ LAPB window size (k)
  - ◆ LAPB acknowledgment timer (T1)
  - ◆ LAPB modulo
  - ◆ QLLC VMACs (virtual MAC addresses) are mapped correctly to X.121 addresses

Is the number in the Set Asynchronous Balance Mode (SABM) field higher than ten? Check the **show interface serial** command output for the SABM requests field. There should always be at least one SABM, but not more than ten. If there are more than ten SABMs, the packet switch is probably not responding.

Check the modems, cables, and connections to the X.25 node. Call the X.25 provider to check the configuration and status of the X.25 node. You can use `loopback mode` to check for a connection problem.

Issue the **show interface serial** command several times. In any of the next fields, are the numbers incrementing or large? Consider the number large if it represents more than 0.5 percent of the number of information frames. Large numbers in these fields indicate that there is a possible problem somewhere in the X.25 network provider (in which case, line quality needs to be checked):

- Number of rejects (REJs)
- Number of Receive Not Ready (RNR) events
- Number of protocol frame errors (FRMRs)
- Number of restarts (RESTARTs)
- Number of disconnects (DISCs)

If subaddresses are used, ensure that these configuration statements are included:

```
x25 routing x25 route ^xxx.*alias serial 0 - ?

!--- Your interface number could be different.

!
x25 routing

!--- Enables x25 switching.

!
x25 route

!--- Add an entry to the X.25 routing table.

!
interface serial y
  x25 alias ^xxx.*
```

The *xxx* indicates the interface serial 0 address of the X.25 router.

If you are using reversed QLLC where a PU 2.0 LAN device communicates with an IBM FEP that is running NCP Packet Switching Interface (NPSI) X.25 software then add these configuration parameter to serial 0:

1. The **npsi–poll** command does not allow null XIDs to be sent to the FEP. It enables a connection between a PU 2.0 on the LAN side and a FEP that is running NPSI. This command is necessary because, in a Token Ring or RSRB environment, the LAN–attached devices start up by sending a null

XID packet upstream. If the Cisco IOS software forwards this null XID to an X.25–attached FEP, then the FEP responds as if it were connecting to a PU 2.1 device and breaks the connection when the PU 2.0 next sends an XID Format 0 Type 2.

2. The **qlc npsi-poll** command intercepts any null XID packet that the software receives on the LAN interface and returns a null XID response to the downstream device. It continues to allow XID Format 3 and XID Format 0 packets through the X.25 device.

Are you using PVCs and SVCs? The PVC channel specifications need to be lower than any SVC range. The default is a two–way range between 1 and 1024, so the lowest two–way circuit (LTC) value needs to be raised, to define any PVCs. Check with your X.25 provider, and reconfigure the virtual circuits to match requirements.

Are the X.25 SVCs configured in this order?

1. All one–way incoming circuits.
2. All two–way circuits.
3. All one–way outgoing circuits.

You can issue these commands to verify the parameters and the status of the connection:

- **show llc2**
- **show x25 map**
- **show x25 vc**
- **show qlc**

## QLLC Debugs

Before you attempt any of the **debug** commands in this document, refer to Important Information on Debug Commands.

If the X.25 Layer 2 protocol LAPB in the output of the **show interface serial** command is not in CONNECT status, then issue this command:

- **debug lapb**

When you are troubleshooting QLLC, issue these **debug** commands:

- **debug qlc error**
- **debug qlc event**
- **debug qlc packet**
- **debug qlc state**
- **debug qlc timer**
- **debug qlc x25**
- **debug x25 all**
- **debug x25 events**

The **debug x25 vc** command displays information on traffic for a particular virtual circuit. It modifies the operation of the **debug x25 all** or **debug x25 events** commands, so one of those commands must be issued with **debug x25 vc**, to produce output.

For the DLSw peer router, these **debug** commands are useful:

- **debug dlsw core message**
- **debug cls message**

Output from these **show** commands is also useful:

- **show cls**
- **show qllc**

The next, short example output is of a QLLC startup under these circumstances:

- A dumb PU 2.0 is coaxially attached to an IBM 3174 Establishment Controller.
- The 3174 has a QLLC connection to a router.
- The LAN partner is an IBM 3745 Communications Controller, and the PU is performing 3270 emulation.

**Note:** For a more detailed explanation of X.25 parameters and states, refer to X.25 international standards specifications at the Protocol Directory .

```
Serial0: I X25 P1 CALL REQUEST (11) 8 lci 20

From(8): 06431743 To(2): 64
Facilities (0)
Call User Data (1): 0xC3 (qllc)
Serial 0: X25 O P4 CALL CONNECTED (5) 8 lci 20
From(0): To(0):
Facilities: (0)
QLLC: allocating new qllc lci 20
QLLC: tx POLLING TEST, da 4000.3172.0002,sa 4000.011c.3174
QLLC: rx explorer response, da 4000.011c.3174, sa c000.3172.0002,
rif 08B0.1A91.1901.A040
QLLC: gen NULL XID, da c000.3172.0002, sa 4000.011c.3174,
rif 0830.1A91.1901.A040, dsap 4, ssap 4
QLLC: rx XID response, da 4000.011c.3174, sa c000.3172.0002,
rif 08B0.1A91.1901.A040
Serial0 QLLC O: ADM XID
Serial0: X25 O P4 DATA (5) Q 8 lci 20 PS 0 PR 0
Serial0: X25 I P4 RR (3) 8 lci 20 PR 1
Serial0: X25 I D1 DATA (25) Q 8 lci 20 PS 0 PR 1
Serial0 QLLC I: QXID-RSPQLLC: addr 01, ctl BF
QLLC: Fmt 1T2: 01731743
QLLC: 4000.011c.3174DISCONNECT net <-SABME (NONE)6F
QLLC: QLLC_OPEN : VMAC 4000.011C.3174
SERIAL0 QLLC O: QSM-CMD
SERIAL0: X25 O D1 DATA (5) Q 8 LCI 20 PS 1 PR 1
```

These are some explanations of that output:

- I An input packet.
- P1 An X.25 state.
- CALL REQUEST An X.25 DTE to DCE packet that starts the X.25 connection.
- (11) The length of the packet, in bytes.
- 8 Indicates modulo 8.
- lci 20 The X.25 logical channel number used by this connection.
- From(8): 06431743 The eight-byte calling address.
- To(2): 64 The two-byte called address.
- (0) Indicates that no facilities are used.
- (1): 0xC3 One byte of X.25 user data, which indicates a QLLC connection

## NetPro Discussion Forums – Featured Conversations

Networking Professionals Connection is a forum for networking professionals to share questions, suggestions,

and information about networking solutions, products, and technologies. The featured links are some of the most recent conversations available in this technology.

NetPro Discussion Forums – Featured Conversations for IBM
---

Network Infrastructure: Enterprise Data Centers
---

---

## Related Information

- **Troubleshooting DLSw**
  - **DLSw and DLSw+ Support**
  - **Technology Support**
  - **Product Support**
  - **Technical Support – Cisco Systems**
- 

[Contacts & Feedback](#) | [Help](#) | [Site Map](#)

© 2008 – 2009 Cisco Systems, Inc. All rights reserved. [Terms & Conditions](#) | [Privacy Statement](#) | [Cookie Policy](#) | [Trademarks of Cisco Systems, Inc.](#)

---

Updated: Jan 28, 2008

Document ID: 12490

---