NORTEL NETWORKS

Nortel Networks Multiservice Switch

7400/15000/20000

Frame Relay Configuration Management

NN10600-901

Nortel Networks Multiservice Switch 7400/15000/20000
Frame Relay Configuration Management

Publication: NN10600-901
Document status: Standard
Document version: 6.1S1
Document date: August 2004

Publication history

August 2004

6.1S1 Standard
General availability. Contains information on Nortel Networks Multiservice
Switch 7400, 15000, and 20000 for the PCR6.1 release.

Contents

Chapter 10
Module-wide QOS monitoring   175

Chapter 11
Multilink frame relay monitoring   189

List of figures

List of tables

About this document

This document describes the configuration of Nortel Networks Multiservice Switch node frame relay service.

The term frame relay in this guide refers to the Multiservice Switch node's frame relay service unless otherwise specified.

The term network in the context of this document refers in general to any switching network, and often the Multiservice Switch network in particular. The term user refers to the customer equipment connected to the network. For instance, a router is a user.

This document also describes the configuration of interworking between a network element running hyperstream software (BNX) and a Multiservice Switch 15000 or Multiservice Switch 20000 node, in which interworking supports frame relay over IP. This functionality is known throughout this document as BNX interworking function (BNX-Iwf).

The configuration of Multiservice Switch frame relay ISDN switched access service Multiservice Switch 7400 nodes is described in this document. The term frame relay ISDN switched access refers to the Multiservice Switch frame relay ISDN switched access service unless otherwise specified.

Who should read this document and why

This document is for persons responsible for performing the following tasks for frame relay:

- planning

- engineering

- installing and configuring

- provisioning

- operating and maintaining

- troubleshooting

What you need to know

This guide assumes that you know and understand frame relay and the Nortel Networks Multiservice Switch network architecture. For BNX-Iwf, it is also assumed that you understand the network element running hyperstream software (BNX).

To fully understand the information in this guide, you should also be familiar with the Open Systems Interconnection (OSI) model.

What's new in this document

There were no new features added to this document.

Other changes made to this document include the following:

- The terms Passport and PVG have been rebranded in conjunction with the new Nortel Networks' brand simplified naming format. Passport is now referred to as the Nortel Networks Multiservice Switch, and PVG is now Media Gateway 7480/15000. For more information on the product rebranding, refer to NN10600-000 *Nortel Networks Multiservice Switch 7400/15000/20000 What's New in PCR6.1*.

- Changes made throughout the document to enhance compliance with Nortel Networks documentation standards (for example, Modular Task Based Information standards).

Chapter 1
Frame relay configuration

Use the frame relay configuration work flow to perform the tasks needed to create the frame relay service, create frame relay connections and configure additional frame relay features.

Prerequisites to frame relay configuration

- Install and configure appropriate hardware for frame relay.

- Install base software.

- Install the frame relay software on the node and define a logical processor type.

- Create a network plan with the values needed for configuration.

Frame relay configuration tasks

This task flow shows you the sequence of procedures you perform to configure frame relay. To link to any task, go to "Frame relay configuration task navigation" (page 19).

Figure 1
Frame relay configuration tasks
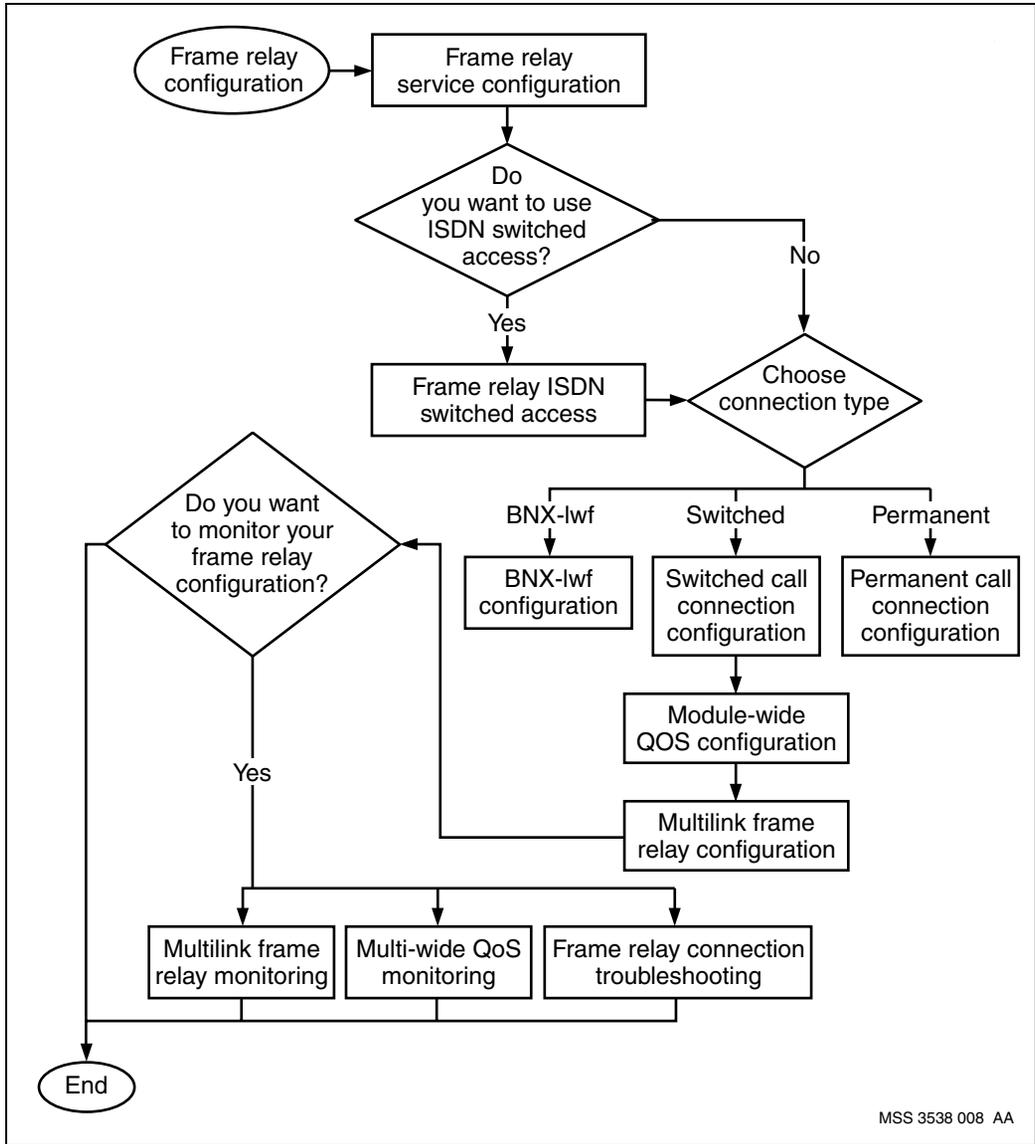
MSS 3538 008 AA

Frame relay configuration task navigation

Chapter 2
Frame relay service configuration

Configure frame relay to establish FR UNI or FR NNI services for your network and manipulate how the services are managed.

Frame relay service configuration procedures

This task flow shows you the sequence of procedures you perform to configure frame relay. To link to any procedure, go to "Frame relay service configuration procedure navigation" (page 23).

Figure 2
Frame relay service configuration procedures
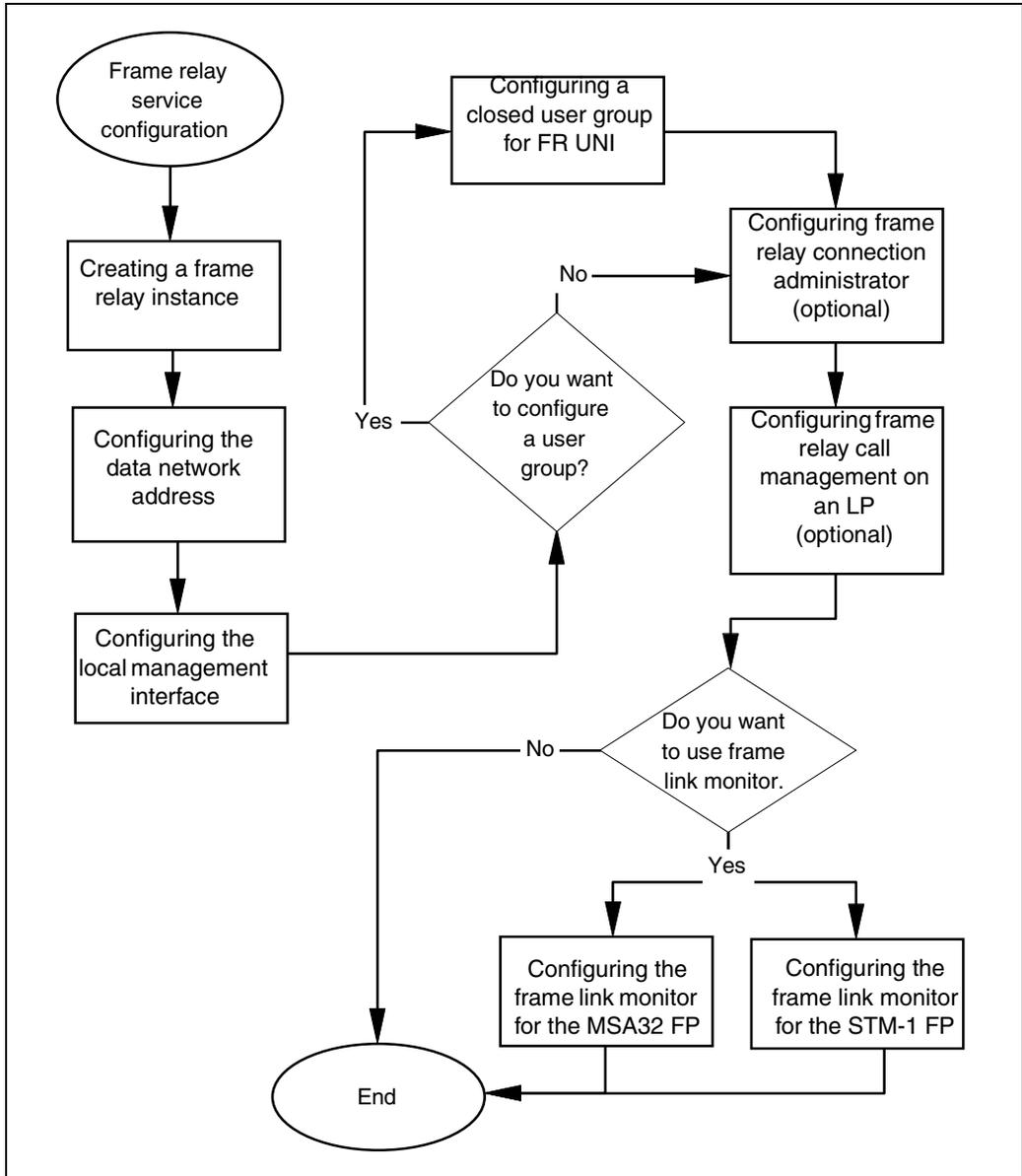
Frame relay service configuration procedure navigation

Creating a frame relay instance

Create a frame relay instance to establish the FR UNI or FR NNI service on the Nortel Networks Multiservice Switch node.

Prerequisites

- If you are using the virtual framer, define it in the feature list of the logical processor type.

Procedure steps

1 Create an instance of the frame relay service.

add FrUni/<FrUni_inst>

add FrNni/<FrNni_inst>

2 If you are using a physical access link, associate the *Framer* component to a physical port.

set FrUni/<FrUni_inst> Framer interfaceName Lp/<lp_no> <port_type>/<port_no>

set FrNni/<FrNni_inst> Framer interfaceName Lp/<lp_no> <port_type>/<port_no>

3 If you are using a logical access link, delete the *Framer* component.

del FrUni/<FrUni_inst> Framer

del FrNni/<FrNni_inst> Framer

4 If you are using a logical access link, add a *virtualFramer* component.

add FrUni/<FrUni_inst> virtualFramer

add FrNni/<FrNni_inst> virtualFramer

5 Associate the virtual framer to a logical processor at the local end.

set FrUni/<FrUni_inst> virtualFramer logicalProcessor Lp/<lp_no>

set FrNni/<FrNni_inst> virtualFramer lp Lp/<lp_no>

6 Associate the virtual framer to the *otherVirtualFramer* at the far end of the connection.

set FrUni/<FrUni_inst> virtualFramer otherVirtualFramer <far_end_connection> virtualFramer

```
set FrNni/<FrNni_inst> virtualFramer
otherVirtualFramer <far_end_connection> virtualFramer
```

If the virtual framer is configured on the 4-port DS3Ch or 1-port STM-1Ch FPs, the pair of virtual framers must be on the same LP.

7   Set the *numberingPlanIndicator* for the frame relay instance.

```
set FrUni/<FrUni_inst> Dna npi <plan>
```

```
set FrNni/<FrNni_inst> Dna npi <plan>
```

8   Specify the address for the frame relay instance. The data network address (DNA) identifies the FR UNI or FR NNI in the network and must be unique.

```
set FrUni/<FrUni_inst> Dna dna <address>
```

```
set FrNni/<FrNni_inst> Dna dna <address>
```

9   To spare the FR UNI, add a *spareDna* to the FR UNI instance.

```
add FrUni/<FrUni_inst> dna spareDna
```

10   For sparing the FR UNI, configure the DNA of the *spareDna*.

```
set FrUni/<FrUni_inst> dna spareDna dna <spareaddress>
```

11   Optionally, specify the number of egress emission priority queues for the frame relay instances.

```
set FrUni/<FrUni_inst> numberOfEmissionQs <number>
```

```
set FrNni/<FrNni_inst> numberOfEmissionQs <number>
```

Variable definitions

Variable	Value
<address>	is the data network address for the FR UNI instance.
<far_end_connection>	is the type of connection and the instance that the virtual framer is connecting to at the far end. For example, the far end could be a FR UNI, FR NNI, or FRATM connection.
<FrNni_inst>	is the instance number of the FR NNI. The instance value you assign must be unique.
<FrUni_inst>	is the instance number of the FR UNI. The instance value you assign must be unique.
(Sheet 1 of 2)	

Variable	Value
<lp_no>	is the LP instance associated with the frame relay instance.
<number>	is the number of emission queues.
<plan>	is the numbering plan indicator (either X.121 or E.164).
<port_type>	is the type of port associated with the frame relay instance.
<port_no>	is the port number associated with the frame relay instance.
<spareaddress>	is the spare data network address.
(Sheet 2 of 2)	

Procedure job aid

Figure 3
Frame relay instance FR UNI component hierarchy

```
EM
   └── FrameRelayUni (FrUni)
          numberOfEmissionQs

          ── Framer
                  interfaceName

          ── VirtualFramer (VFramer)
                  otherVirtualFramer (ovf)
                  logicalProcessor (lp)

          ── DataNetworkAddress (Dna)
                  numberingPlanIndicator (npi)
                  dataNetworkAddress (dna)
          ── LocalManagementInterface (Lmi)
```
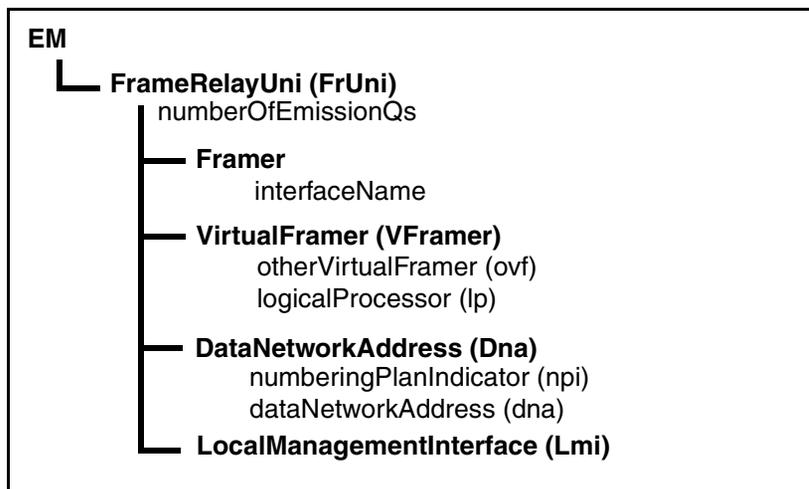
Figure 4
Frame relay instance FR NNI component hierarchy

EM
        FrameRelayNni (FrNni)
            numberOfEmissionQs
            Framer
                  interfaceName

            VirtualFramer (VFramer)
                  otherVirtualFramer (ovf)
                  logicalProcessor (lp)
            DataNetworkAddress (Dna)
                  numberingPlanIndicator (npi)
                  dataNetworkAddress (dna)
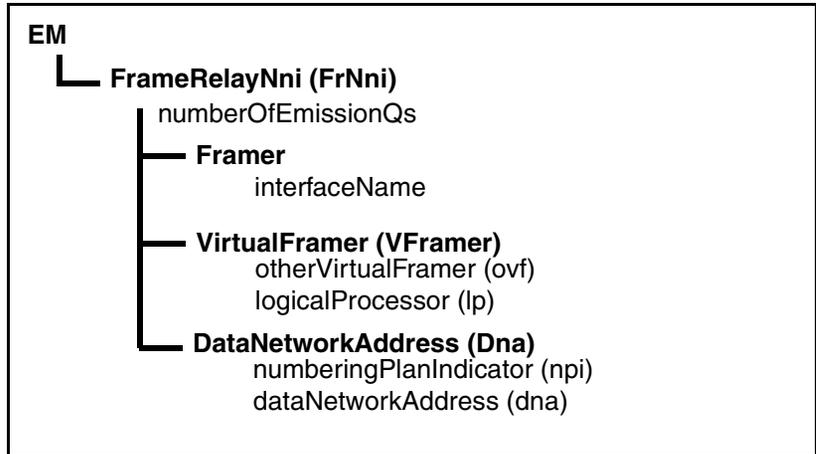
Configuring the data network address

Configure the data network address to change the values of outgoing call options; discard priority, routing reliability, and transfer priority.

Procedure steps

1 Specify the discard priority for outgoing calls on the interface.

```
set FrUni/<FrUni_inst> Dna odf <gl_dp>
```

```
set FrNni/<FrNni_inst> Dna odf <gl_dp>
```

2 Specify the class of routing reliability for outgoing calls on the interface.

```
set FrUni/<FrUni_inst> Dna opr <gl_pr>
```

```
set FrNni/<FrNni_inst> Dna opr <gl_pr>
```

3 Specify the default transfer priority for outgoing calls on the interface.

```
set FrUni/<FrUni_inst> Dna dtp <gl_tp>
```

```
set FrNni/<FrNni_inst> Dna dtp <gl_tp>
```

4 Specify whether the interface overrides its configured transfer priority (when receiving a call) with the transfer priority configured at the calling end.

```
set FrUni/<FrUni_inst> Dna tpor <tpor_on>
```

```
set FrNni/<FrNni_inst> Dna tpor <tpor_on>
```

Variable definitions

Variable	Value
<gl_dp>	is the discard priority setting of normal or high.
<gl_pr>	is the routing reliability setting of normal or high.
<gl_tp>	is the transfer priority instance between 0 and 15.
<FrNni_inst>	is the instance number of the FR NNI.
<FrUni_inst>	is the instance number of the FR UNI.
<tpor_on>	is a value of yes or no that allows the interface to override its configured transfer priority.

Procedure job aid

Figure 5
DNA FR UNI component hierarchy

```
EM
  └─ FrameRelayUni (FrUni)
        └─ DataNetworkAddress (Dna)
              ....
              outDefaultPriority (odf)
              outDefaultPathReliability (opr)
              defaultTransferPriority (dtp)
              transferPriorityOverride (tpor)
```
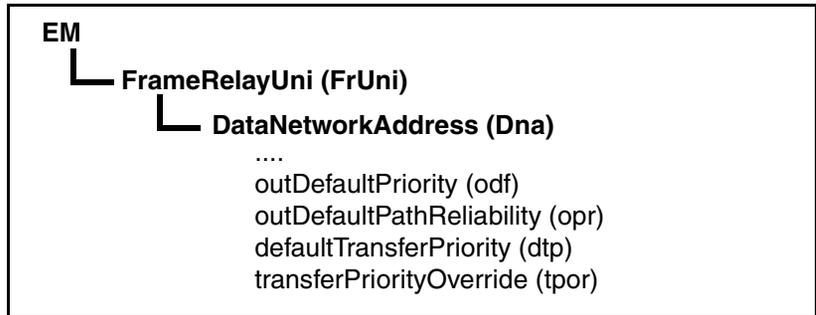
Figure 6
DNA FR NNI component hierarchy

```
EM
  └─ FrameRelayNni (FrNni)
        └─ DataNetworkAddress (Dna)
              ....
              outDefaultPriority (odf)
              outDefaultPathReliability (opr)
              defaultTransferPriority (dtp)
              transferPriorityOverride (tpor)
```
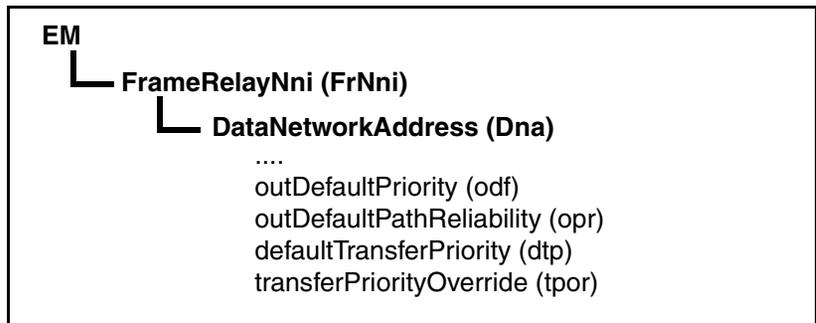
Configuring the local management interface

Configure the local management interface (LMI) to provision the settings for connection status management.

Procedure steps

1 Specify the type of LMI procedures to be used for permanent connection status management on the FR UNI.

```
set FrUni/<FrUni_inst> Lmi proc <type>
```

```
set FrNni/<FrNni_inst> Lmi proc <type>
```

On a 4-port DS3 channelized frame relay FP, if the timeslot of a channel associated with a framer interface is provisioned with the value of none, the type of LMI procedure on the frame relay interface also needs to be provisioned as none. Provisioning the timeslot value of none prevents the DLCI component from being provisionable.

2 Specify the number of error events that must occur before a PVC segment or end-to-end SPVC connection is considered inactive.

```
set FrUni/<FrUni_inst> Lmi n392 <error_count>
```

```
set FrNni/<FrNni_inst> Lmi n392 <error_count>
```

3 Specify the number of events within which no more than the specified number of error events (as configured above) may occur.

```
set FrUni/<FrUni_inst> Lmi n393 <count>
```

```
set FrNni/<FrNni_inst> Lmi n393 <count>
```

4 Specify whether the FR UNI generates asynchronous status reports.

```
set FrUni/<FrUni_inst> Lmi asr <asr_on>
```

```
set FrNni/<FrNni_inst> Lmi asr <asr_on>
```

5 Specify the frequency that status enquiry messages are initiated for link integrity verification.

```
set FrUni/<FrUni_inst> Lmi t391 <chk_interval>
```

```
set FrNni/<FrNni_inst> Lmi t391 <chk_interval>
```

6 Specify the number of polling cycles that must complete before requesting a full status report from the other network.

```
set FrUni/<FrUni_inst> Lmi n391 <cycles>
```

```
set FrNni/<FrNni_inst> Lmi n391 <cycles>
```

7   Specify the time interval to expect a status enquiry message from the other network.

```
set FrUni/<FrUni_inst> Lmi t392 <chk_interval>
```

```
set FrNni/<FrNni_inst> Lmi t392 <chk_interval>
```

8   Specify whether A-bit signaling affects data transfer on a PVC.

```
set FrUni/<FrUni_inst> Lmi inab <inab_on>
```

```
set FrNni/<FrNni_inst> Lmi inab <inab_on>
```

9   Specify whether the LMI includes traffic parameter and QoS information in full status messages for the FR UNI.

```
set FrUni/<FrUni_inst> Lmi pvcConfig <pvcconfig_on>
```

10 Select the type of *pvcAlarmsReporting* attribute warning messages you want to display.

```
set FrUni/<FrUni_inst> Lmi pvcAlarmsReporting
<alarm_type>
```

```
set FrNni/<FrNni_inst> Lmi pvcAlarmsReporting
<when_alarm>
```

Variable definitions

Variable	Value
<asr_on>	is a setting of on or off for enabling asynchronous status reporting.
<chk_interval>	is a length of time in seconds.
<count>	is the number of monitored events.
<cycles>	is the number of polling cycles.
<error_count>	is the number of error events.
<FrNni_inst>	is the instance number of the FR NNI.
<FrUni_inst>	is the instance number of the FR UNI.
<inab_on>	is a setting of on or off to determine whether A-bit signaling is ignored.
<pvcconfig_on>	is a setting of on or off to allow the full status message to include additional Q.933 information elements.
(Sheet 1 of 2)	

Variable	Value
<type>	is the type of LMI procedure used (either none, vendorForum, ansi, itu, or autoConfigure).
<when_alarm>	is a time when internal and external networks display pvcAlarmsReporting warning messages.
(Sheet 2 of 2)	

Procedure job aid

Figure 7
LMI FR UNI component hierarchy

```
EM
    └── FrameRelayUni (FrUni)
            └── LocalManagementInterface (Lmi)
                    procedures (proc)
                    asyncStatusReport (asr)
                    errorEventThreshold (n392)
                    eventCount (n393)
                    checkPointTimer (t392)
                    ignoreActiveBit (inab)
                    fullStatusPollingCycles (n391)
                    linkVerificationTimer (t391)
                    pvcConfigParmsInFsr (pvcConfig)
                    pvcAlarmsReporting (pvcReport)
```
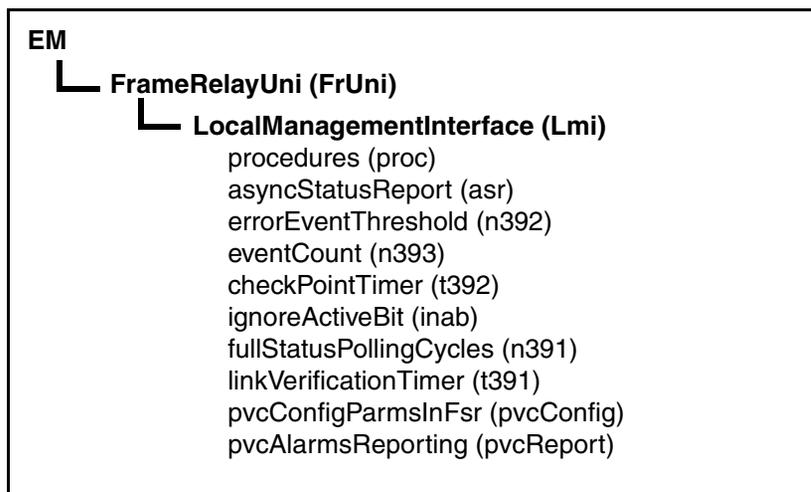
Figure 8
LMI FR NNI component hierarchy

```
EM
      └── FrameRelayNni (FrNni)
              └── LocalManagementInterface (Lmi)
                      procedures (proc)
                      errorEventThreshold (n392)
                      eventCount (n393)
                      asyncStatusReport (asr)
                      linkVerificationTimer (t391)
                      fullStatusPollingCycles (n391)
                      checkPointTimer (t392)
                      ignoreActiveBit (inab)
                      pvcAlarmsReporting (pvcReporting)
```
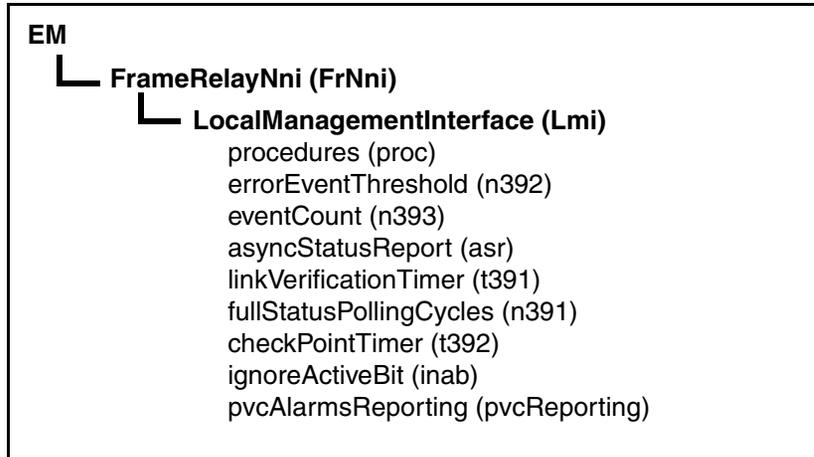
Configuring a closed user group for FR UNI

Configure a closed user group (CUG) for FR UNI to subscribe a FR UNI to a CUG and define its call options.

Procedure steps

1 Subscribe the FR UNI to a CUG, by creating an instance of the *ClosedUserGroup* component under the address component of the interface.

```
add FrUni/<FrUni_inst> Dna Cug/<cg>
```

2 Define the CUG type.

```
set FrUni/<FrUni_inst> Dna Cug/<cg> type national
```

3 If you have configured an international CUG, assign the data network id code (dnic).

```
set FrUni/<FrUni_inst> Dna Cug/<cg> dnic <dnic_code>
```

4 Assign the CUG identifier.

```
set FrUni/<FrUni_inst> Dna Cug/<cg> ilc <id_code>
```

5 Specify whether CUG members can receive calls from non-CUG members.

```
set FrUni/<FrUni_inst> Dna Cug/<cg> inc <inc_access>
```

6 Specify whether CUG members can make calls to non-CUG members.

```
set FrUni/<FrUni_inst> Dna Cug/<cg> ouc <out_access>
```

7 Optionally, configure the CUG as a simple CUG by setting the *preferential* attribute.

```
set FrUni/<FrUni_inst> Dna Cug/<cg> preferential
<pref_on>
```

8 To remove subscription from a CUG, remove the *Cug* component from the *Dna* component.

```
delete FrUni/<FrUni_inst> Dna Cug/<cg>
```

Variable definitions

Variable	Value
<cg>	is the instance number of the CUG.
<dnic_code>	is an assigned data network ID code.
<FrUni_inst>	is the instance number of the FR UNI.
<id_code>	is a CUG identifier of a national or international CUG call.
<inc_access>	is a setting of allowed or disallowed that permits incoming calls to be made using the particular CUG.
<out_access>	is a setting of allowed or disallowed that permits outgoing calls to be made using the particular CUG.
<pref_on>	is a setting of yes or no to make the CUG a SIMPLE CUG or SELECTABLE CUG.
	You can only configure a simple CUG if the associated interface subscribes to a single CUG. The *incCalls* and *outCalls* attribute values must be set to allowed in order to change the *preferential* attribute.

Procedure job aid

Figure 9
CUG for FR UNI component hierarchy

```
EM
    └── FrameRelayUni (FrUni)
            └── DataNetworkAddress (Dna)
                    └── ClosedUserGroup (Cug)
                            type
                            dnic
                            interlockCode (ilc)
                            preferential (pre)
                            outCalls (ouc)
                            incCalls (inc)
```
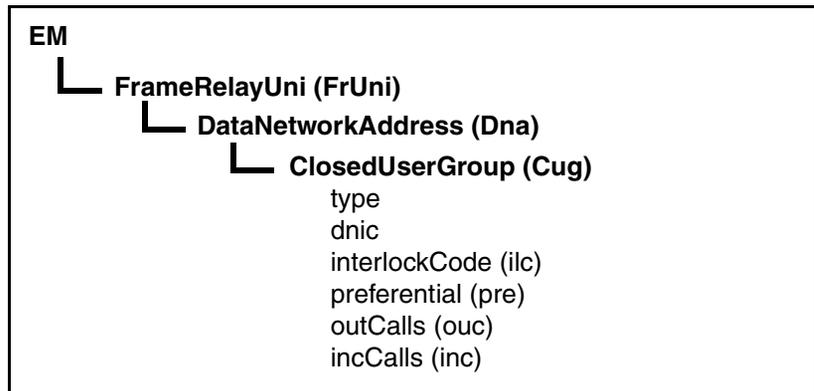
Configuring frame relay connection administrator

Configure frame relay connection administrator to define ingress and egress call admission control parameters for the frame relay interface.

Procedure steps

1 Turn call admission control (CAC) on for traffic in the ingress direction.

```
set FrUni/<FrUni_inst> Ca ingCos <cac_intraffic>
```

```
set FrNni/<FrNni_inst> Ca ingCos <cac_intraffic>
```

2 Turn call admission control (CAC) on for the FR UNI for traffic the egress direction.

```
set FrUni/<FrUni_inst> Ca egCos <cac_egtraffic>
```

```
set FrNni/<FrNni_inst> Ca egCos <cac_egtraffic>
```

3 Specify the maximum number of simultaneous EIR-only connections for traffic in the ingress direction.

```
set FrUni/<FrUni_inst> Ca ingressMaximumEirOnlyCalls
<eir_max>
```

```
set FrNni/<FrNni_inst> Ca ingressMaximumEirOnlyCalls
<eir_max>
```

4 Specify the maximum number of simultaneous EIR-only connections for traffic in the egress direction.

```
set FrUni/<FrUni_inst> Ca egressMaximumEirOnlyCalls
<eir_max>
```

```
set FrNni/<FrNni_inst> Ca egressMaximumEirOnlyCalls
<eir_max>
```

5 Specify the port capacity assigned to each bandwidth pool for the interface for CIR in the ingress direction.

```
set FrUni/<FrUni_inst> Ca ingCirBP <pool_no> <per0>
<pool_no> <per1>
```

```
set FrNni/<FrNni_inst> Ca ingCirBP <pool_no> <per0>
<pool_no> <per1>
```

6 Specify the port capacity assigned to each bandwidth pool for the interface for EIR in the ingress direction.

```
set FrUni/<FrUni_inst> Ca ingEirBP <pool_no> <per0>
<pool_no> <per1>
```

```
set FrNni/<FrNni_inst> Ca ingEirBP <pool_no> <per0>
<pool_no> <per1>
```

7 Specify the port capacity assigned to each bandwidth pool for the
interface for CIR in the egress direction.

```
set FrUni/<FrUni_inst> Ca egCirBP <pool_no> <per0>
<pool_no> <per1>
```

```
set FrNni/<FrNni_inst> Ca egCirBP <pool_no> <per0>
<pool_no> <per1>
```

8 Specify the port capacity assigned to each bandwidth pool for the
interface for EIR in the egress direction.

```
set FrUni/<FrUni_inst> Ca egEirBP <pool_no> <per0>
<pool_no> <per1>
```

```
set FrNni/<FrNni_inst> Ca egEirBP <pool_no> <per0>
<pool_no> <per1>
```

9 Specify the bandwidth pool to be used for ingress direction traffic with
each transfer priority. You configure this parameter at the module-wide
level.

```
set Mod Frs DprsNet Tpm/<tp> asgInBwPool <pool>
```

```
set Mod Frs DprsNet Tpm/<tp> asgInBwPool <pool>
```

10 Specify the bandwidth pool to be used for egress direction traffic with
each transfer priority. You configure this parameter at the module-wide
level.

```
set Mod Frs DprsNet Tpm/<tp> asgEgBwPool <pool>
```

```
set Mod Frs DprsNet Tpm/<tp> asgEgBwPool <pool>
```

11 Optionally, specify the bandwidth pool to be used for each ingress
direction transfer priority on the interface.

```
set FrUni/<FrUni_inst> Ca Tpm/<tp> asgInBwPool <pool>
```

```
set FrNni/<FrNni_inst> Ca Tpm/<tp> asgInBwPool <pool>
```

12 Optionally, specify the bandwidth pool to be used for each egress
direction transfer priority on the interface.

```
set FrUni/<FrUni_inst> Ca Tpm/<tp> asgEgBwPool <pool>
```

```
set FrNni/<FrNni_inst> Ca Tpm/<tp> asgEgBwPool <pool>
```

13 Specify the maximum amount of bandwidth permitted for a connection on the interface.

```
set FrUni/<FrUni_inst> Ca maxBwCall <bw>
```

```
set FrNni/<FrNni_inst> Ca maxBwCall <bw>
```

14 Specify a link rate other than the physical link rate for the interface. If you are using a virtual framer, you must specify an equivalent link rate.

```
set FrUni/<FrUni_inst> Ca ovLinkRate <rate>
```

```
set FrNni/<FrNni_inst> Ca ovLinkRate <rate>
```

Variable definitions

Variable	Value
<bw>	is a bandwidth in bits/s.
<cac_egtraffic>	is the egress class of service traffic that has CAC enabled for switched connections.
<cac_intraffic>	is the ingress class of service traffic that has CAC enabled for switched connections.
<eir_max>	is the number of calls.
<FrNni_inst>	is the instance number of the FR NNI.
<FrUni_inst>	is the instance number of the FR UNI.
<per0>	is the percentage of port capacity assigned to bandwidth pool 0.
	To partition port capacity among bandwidth pools, separate the percentage values with a space; the values are applied in order to pools 0 through 15.
<per1>	is the percentage of port capacity assigned to bandwidth pool 1.
	To partition port capacity among bandwidth pools, separate the percentage values with a space; the values are applied in order to pools 0 through 15.
<pool>	is the bandwidth pool used for traffic with this transfer priority.
	When you set this parameter, you override the module-wide transfer priority mapping for the FR UNI.
(Sheet 1 of 2)	

Variable	Value
<pool_no>	is a pool number that you are setting.
<rate>	is the equivalent link rate in bits/s.
<tp>	is the transfer priority instance.
(Sheet 2 of 2)	

Procedure job aid

Figure 10
Frame relay connection administrator FR UNI component hierarchy

```
EM
    ModuleData (Mod)
        FrameRelaySystem (Frs)
            DprsNetworking (DprsNet)
                TransferPriorityMapping (Tpm)
                    assignedIngressBandwidthPool (asgInBwPool)
                    assignedEgressBandwidthPool (asgEgBwPool)

    FrameRelayUni (FrUni)
        ConnectionAdministrator (Ca)
            ingressApplyToCos (ingCos)
            ingressMaximumEirOnlyCalls (maxEirIng)
            ingressCirBandwidthPool (ingCirBP)
            ingressEirBandwidthPool (ingEirBP)

            egressApplyToCos (egCos)
            egressMaximumEirOnlyCalls (maxEirEg)
            egressCirBandwidthPool (egCirBP)
            egressEirBandwidthPool (egEirBP)

            maximumBandwidthPerCall (maxBwCall)
            overrideLinkRate (ovLinkRate)


            TransferPriorityMapping (Tpm)
                assignedIngressBandwidthPool (asgInBwPool)
                assignedEgressBandwidthPool (asgEgBwPool)
```

Figure 11
Frame relay connection administrator FR NNI component hierarchy

```
EM
    ModuleData (Mod)
        FrameRelaySystem (Frs)
            DprsNetworking (DprsNet)
                TransferPriorityMapping (Tpm)
                    assignedIngressBandwidthPool (asgInBwPool)
                    assignedEgressBandwidthPool (asgEgBwPool)


    FrameRelayNni (FrNni)
        ConnectionAdministrator (Ca)
            ingressApplyToCos (ingCos)
            ingressMaximumEirOnlyCalls (maxEirIng)
            ingressCirBandwidthPool (ingCirBP)
            ingressEirBandwidthPool (ingEirBP)

            egressApplyToCos (egCos)
            egressMaximumEirOnlyCalls (maxEirEg)
            egressCirBandwidthPool (egCirBP)
            egressEirBandwidthPool (egEirBP)
            maximumBandwidthPerCall (maxBwCall)
            overrideLinkRate (ovLinkRate)


            TransferPriorityMapping (Tpm)
                assignedIngressBandwidthPool (asgInBwPool)
                assignedEgressBandwidthPool (asgEgBwPool)
```

Configuring frame relay call management on an LP

Configure frame relay call management on an LP to set engineering parameters on an LP basis for the entire frame relay system.

Procedure steps

1   Create a component to override default frame relay call management values on an LP.

    `add Lp/<lp_inst> Eng Frs Ov`

2   Specify the maximum number of calls permitted on the LP.

    `set Lp/<lp_inst> Eng Frs Ov maxCalls <max>`

3   Specify the number of calls accepted when the LP has reached overload conditions, that is, when the number of current calls and the number of queued clear records equals the *maxCalls* attribute.

    `set Lp/<lp_inst> Eng Frs Ov headroomCalls <hdroom>`

Variable definitions

Variable	Value
<hdroom>	is the maximum number of calls accepted on the LP under overload conditions.
<lp_inst>	is the instance number of the logical processor (LP).
<max>	is the maximum number of current frame relay calls of any type allowed on the LP.

Procedure job aid

Figure 12
Frame relay call management on an LP FR UNI Component hierarchy

```
EM
  └── LogicalProcessor (Lp)
        └── Engineering (Eng)
              └── FrameRelaySystem(Frs)
                   currentCalls
                   callsRefused
                   maxCalls
                   headroomCalls
                   └── Override (Ov)
                          maxCalls
                          headroomCalls
```
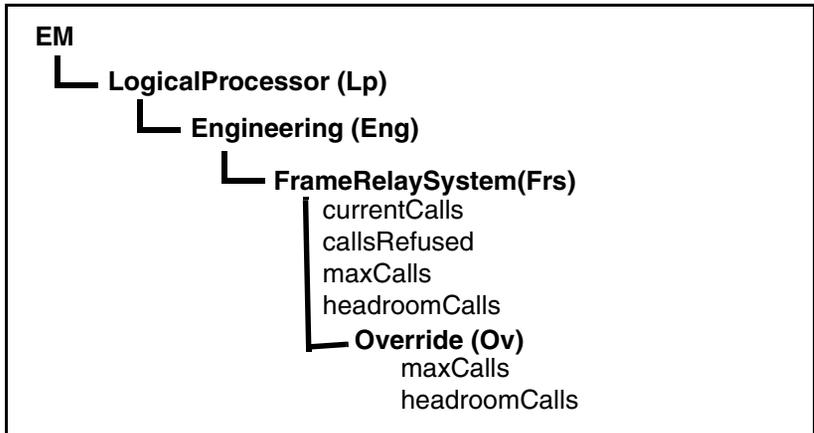
Figure 13
Frame relay call management on an LP FR NNI Component hierarchy

```
EM
  └── LogicalProcessor (Lp)
        └── Engineering (Eng)
              └── FrameRelaySystem(Frs)
                   currentCalls
                   callsRefused
                   maxCalls
                   headroomCalls
                   └── Override (Ov)
                          maxCalls
                          headroomCalls
```
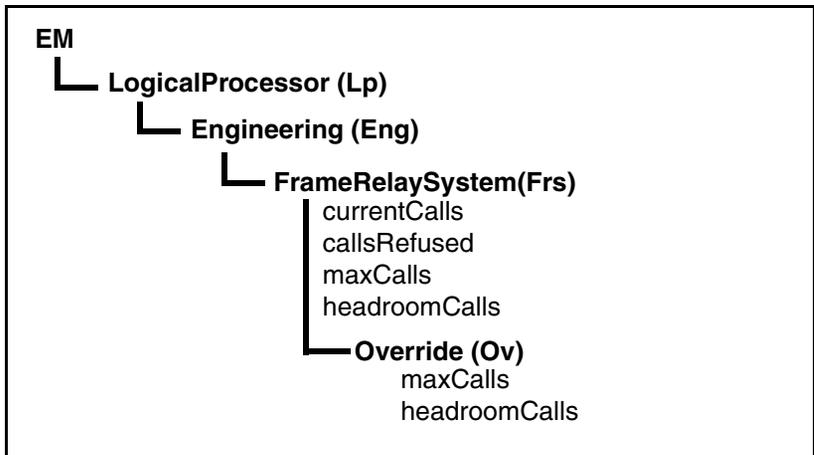
Configuring the frame link monitor for the MSA32 FP

Configure the frame link monitor for the MSA32 FP in order to set up automatic HDLC or E1 CAS A-bit monitoring on a frame relay channel of an E1 port.

Prerequisites

- Before provisioning an FLM, the service to be monitored must already be provisioned and operational.

Procedure steps

1 Add a *FrameLinkMonitoring (Flm)* subcomponent to the *channel* component of the service to be monitored.

   ```
   add Lp/<lp_no> E1/<port_num> Ch/<E1_chan> Flm
   ```

   By default, both operational attributes of the *Flm* component are enabled, so monitoring begins immediately.

2 If you want to disable HDLC monitoring and monitor only the E1 CAS A-bit status, set the *hdlcMonitoring* attribute to the disabled condition.

   ```
   set Lp/<lp_no> E1/<port_num> Chan/<E1_chan> Flm
   hdlcMonitoring disabled
   ```

3 If you want to disable E1 CAS A-bit monitoring and monitor only the HDLC status, set the *aBitMonitoring* attribute to the disabled condition.

   ```
   set Lp/<lp_no> E1/<port_num> Chan/<E1_Chan> Flm
   aBitMonitoring disabled
   ```

Variable definitions

Variable	Value
<E1_chan>	is the instance number of the Chan component, having a value between 0 and 31.
<lp_no>	is the instance number of the *Lp* component for the FP providing the service to be monitored. Because the 32-port DS1/E1 MSA FP is a double-width card, the instance number has a value between 1 and 14 inclusive.
<port_num>	is the instance number of the E1 component, having a value between 0 and 31.

Procedure job aid

Figure 14
Frame link monitor component hierarchy

```
EM
  └─ LogicalProcessor (Lp)
        └─ E1
              └─ Channel (Chan)
                    └─ FrameLinkMonitoring (Flm)
                          Provisioned (Prov) group
                            aBitMonitoring
                            hdlcMonitoring

                          Operational (Op) group
                            flmStatus
```
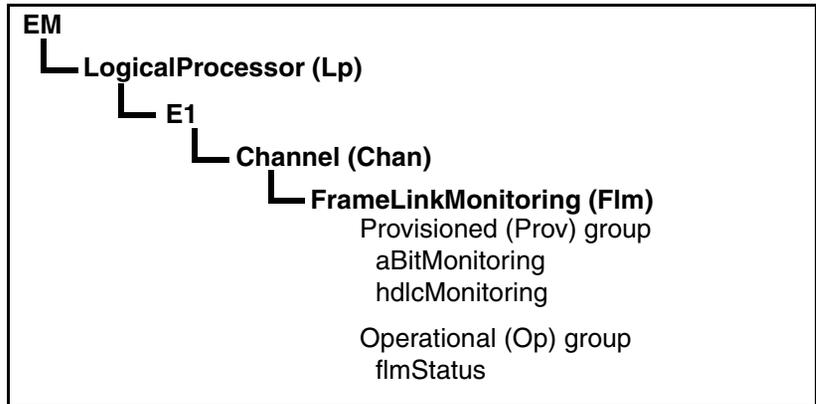
Configuring the frame link monitor for the STM-1 FP

Configure the frame link monitor for the STM-1 FP in order to set up automatic HDLC or E1 CAS A-bit monitoring of a frame relay channel.

Prerequisites

- Before provisioning an FLM, the service to be monitored must already be provisioned and operational.

Procedure steps

1 Add a *FrameLinkMonitoring (Flm)* subcomponent to the *channel* component of the service to be monitored.

```
add Lp/<lp_no> sdh/0 vc4/0 vc12/<k,l,m> E1 Ch/
<E1_chan> Flm
```

By default, both operational attributes of the *Flm* component are enabled, so monitoring begins immediately.

2 If you want to disable HDLC monitoring and monitor only the E1 CAS A-bit status, set the *hdlcMonitoring* attribute to the disabled condition.

```
set Lp/<lp_no> sdh/0 vc4/0 vc12/<k,l,m> E1 Ch/
<E1_chan> Flm hdlcMonitoring disabled
```

hdlcMonitoring always needs to be set to disabled for the STM-1 card, since this card cannot monitor for HDLC.

3 If you want to disable E1 CAS A-bit monitoring and monitor only the HDLC status, set the *aBitMonitoring* attribute to the disabled condition.

```
set Lp/<lp_no> sdh/0 vc4/0 vc12/<k,l,m> E1 Ch/
<E1_chan> Flm abitMonitoring disabled
```

Variable definitions

Variable	Value
<E1_chan>	is the instance number of the Chan component, having a value between 0 and 31.
(Sheet 1 of 2)	

Variable	Value
<k,l,m>	is the instance number for the vc12 component.
<lp_no>	is the instance number of the *Lp* component for the FP providing the service to be monitored. Because the 32-port DS1/E1 MSA FP is a double-width card, the instance number has a value between 1 and 14 inclusive.
(Sheet 2 of 2)	

Procedure job aid

Figure 15
Frame link monitor component hierarchy

Em
    └── **LogicalProcessor (Lp)**
           └── **E1**
               └── **Channel (Chan)**
                    └── **FrameLinkMonitoring (Flm)**
                        Provisioned (Prov) group
                          aBitMonitoring
                          hdlcMonitoring

                        Operational (Op) group
                          flmStatus

PPT 3439 024 AA
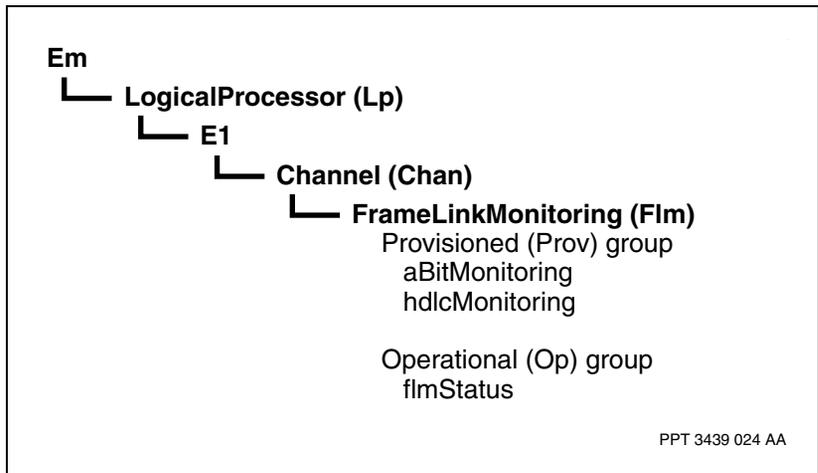
Chapter 3
Frame relay ISDN switched access configuration

Configure frame relay ISDN switched access to create a digital switched access path through an ISDN network.

Prerequisites to frame relay ISDN switched access configuration

- Frame relay ISDN switched access is available on Nortel Networks Multiservice Switch 7400 nodes.

- Before installing and configuring frame relay ISDN switched access, ensure that all the network node candidates for frame relay ISDN switched access traffic run the same version of frame relay ISDN switched access-compatible software. See NN10600-270 *Nortel Networks Multiservice Switch 7400/15000/20000 Software Installation*, which describes how to install the software for this service.

Frame relay ISDN switched access configuration procedures

This task flow shows you the sequence of procedures you perform to configure frame relay ISDN switched access. To link to any procedure, go to "Frame relay ISDN switched access configuration procedure navigation" (page 50).

Figure 16
Frame relay ISDN switched access configuration procedures
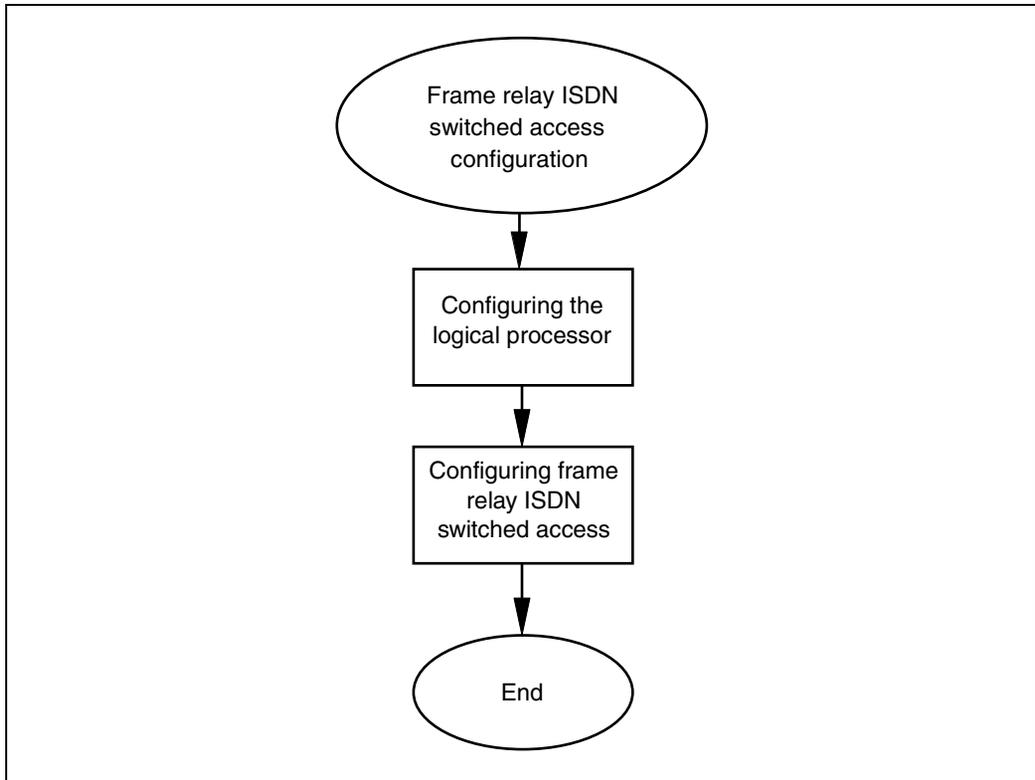
Frame relay ISDN switched access configuration procedure navigation

- "Configuring the logical processor" (page 51)

- "Configuring frame relay ISDN switched access" (page 52)

Configuring the logical processor

Configure the logical processor to set the LPT and Lp, and link the LP to the card.

Procedure steps

1   Add a *LogicalProcessorType* component.

```
add sw lpt/<lpt_name>
```

2   Set the *featureList* attribute of the *LogicalProcessorType* component to include the ISDN protocol, the frame relay ISDN feature, and the frame relay SVC software.

```
set sw lpt/<lpt_name> featureList <Q.931 variant>
frameRelayIsdn frameRelayUniPvcSvc
```

3   Set the *logicalProcessorType* attribute of the *logicalProcessor* component to use the frame relay ISDN package.

```
set lp/<y> logicalProcessorType sw lpt/<lpt_name>
```

Variable definitions

Variable	Value
<lpt_name>	is any mnemonic (for example, FPFrIsdn).
<Q.931 variant>	is the ISDN protocol (for example, TS014, Etsi, JapanIns, NationalIsdn2 or Nis).
<y>	is the card slot number of the logical processor.

Configuring frame relay ISDN switched access

Configure frame relay ISDN switched access to create a digital switched access path through an ISDN network.

Prerequisites

- To provision frame relay ISDN switched access, use either the default settings that come with the package, or provision the service to meet your specific requirements. If you decide to override the attributes' default values, be sure you understand what each attribute does and the implications of the change, see NN10600-060 *Nortel Networks Multiservice Switch 7400/15000/20000 Component Reference*.

- Each protocol that frame relay ISDN switched access supports has unique parameters that affect how you provision the service. See "Procedure job aid" (page 58) for additional information.

- For additional information about Australian TS041 protocol provisioning, see NN10600-900 *Nortel Networks Multiservice Switch 7400/15000/20000 Frame Relay Technology Fundamentals* for additional information.

- See the following sections for assistance with frame relay ISDN switched access configuration:
  - "Frame relay ISDN switched access component hierarchy" (page 58)
  - "Australian TS014 provisioning details" (page 59)
  - "European ISDN provisioning details" (page 60)
  - "Japanese ISDN provisioning details" (page 61)
  - "North American ISDN provisioning details" (page 61)

Procedure steps

1 Add the *DataSigChan* component to the root.

    add DataSigChan/<x>

2 Set the customer identifier for the *DataSigChan* component.

    set DataSigChan/<x> customerId <u>

3   Add the specific Q.931 standard that the PSTN network uses to define interpretation of the D-channel protocol information.

```
add DataSigChan/<x> <Q.931 variant>
```

4   Set the physical channel for the *Framer* component.

```
set DataSigChan/<x> <Q.931 variant> framer
interfaceName lp/<y> <ds1 or e1>/<z> chan/<w>
```

5   Add a *ChannelGroup* component.

```
add DataSigChan/<x> callControl channelGroup/<k>
```

6   Set the channelList arguments as required.

```
set DataSigChan/<x> callControl
channelGroup/<k> channelList ! a b
```

7   Set the ChannelGroup arguments as required.

```
set DataSigChan/<x> callControl
channelGroup/<k> ScreeningMode ! upvp np
```

```
set DataSigChan/<x> callControl channelGroup/<k>
ChannelAssignmentOrder ascending
```

```
add DataSigChan/<x> callControl ChannelGroup/<k>
callingPartyNumber/<party_number1>
```

```
add DataSigChan/<x> callControl ChannelGroup/<k>
callingPartyNumber/<party_number2>
```

8   Add the *Isdn* component to the *FrUni* component.

```
add Fruni/<m> Isdn
```

```
add Fruni/<n> Isdn
```

9   Set the physical channel for the *FrUni* component.

```
set Fruni/<m> framer interfaceName lp/<y> <ds1 or e1>/
<z> chan/<a>
```

```
set Fruni/<n> framer interfaceName lp/<y> <ds1 or e1>/
<z> chan/<b>
```

Variable definitions

Variable	Value
<k>	is a channel group number.
<m> and <n>	are the FR UNI instance numbers.
<party_number1>	is a calling party number.
<party_number2>	is a calling party number.
<Q.931 variant>	is the ISDN protocol (for example, TS014, Etsi, JapanIns, NationalIsdn2 or Nis).
<u>	is number unique to this customer.
<w>	is a channel number.
<x>	is the channel number for the ISDN PRI.
<y>	is the card slot number of the logical processor.
<z>	is a port number. When provisioning a 32-port DS1 or E1MSA FP, port numbers 0 to 3 are still the only valid choices.

Example of configuring frame relay ISDN switched access

This example describes how to provision frame relay ISDN switched access to run the TS014 protocol on an E1C function processor with the D-channel on timeslot 16.

Note: The parameters in this section are examples only. Each network has unique requirements. This section provides general guidelines to demonstrate a typical application.

In this example, three channel groups control access to five B-channels that run frame relay ISDN switched access service on timeslots 1, 2, 3, 4, and 5. The B-channels are numbered 1 to 5. The calling party number provisioning controls access to these B-channels. The B-channel assignment order is set to ascending. So, when a call requests access to the B-channel the network assigns the B-channel with the lowest unused channel number.

The first groups can use channels 1, 2, and 3. Calls can originate from phone numbers: 1111111, 2222222, 3333333, 4444444, and 5555555.

The second group can use channel 4 only. Calls can originate from phone numbers:1231230, 1231231, and 1231232.

The last group can use channel 5 only. Calls can originate from phone number 2222220 only.

1   Associate a *logicalProcessor* component to an E1C function processor (FP) on the Nortel Networks Multiservice Switch shelf. In this example, an E1C FP is in card slot 9.

```
add lp/9
```

```
set lp/9 mainCard shelf card/9
```

2   Add the ISDN software to the *logicalProcessorType* component's feature list, and set the *logicalProcessor* component to use this feature list.

```
add sw lpt/FPFrIsdn featureList
```

```
set sw lpt/FPFrIsdn
featureList Ts014 frameRelayIsdn frameRelayUni
```

```
set lp/9 logicalProcessorType sw lpt/FPFrIsdn
```

3   Add an E1C port to the *logicalProcessor* component. Then, add six *Channel* components to this port. Also associate each *Channel* component with a timeslot on the E1 frame.

```
add lp/9 e1/0
```

```
add lp/9 e1/0 chan/16
```

```
add lp/9 e1/0 chan/1
```

```
add lp/9 e1/0 chan/2
```

```
add lp/9 e1/0 chan/3
```

```
add lp/9 e1/0 chan/4
```

```
add lp/9 e1/0 chan/5
```

```
set lp/9 e1/0 chan/16 timeslots 16
```

```
set lp/9 e1/0 chan/1 timeslots 1
```

```
set lp/9 e1/0 chan/2 timeslots 2
```

```
set lp/9 e1/0 chan/3 timeslots 3
```

```
set lp/9 e1/0 chan/4 timeslots 4
```

```
set lp/9 e1/0 chan/5 timeslots 5
```

4 Add an *Isdn* component to those *FrUni* components you want for ISDN and assign a different Channel Number to each FR UNI.

```
add Fruni/901 Isdn

add Fruni/902 Isdn

add Fruni/903 Isdn

add Fruni/904 Isdn

add Fruni/905 Isdn

set Fruni/901 framer interfaceName lp/9 e1/0 chan/1

set Fruni/902 framer interfaceName lp/9 e1/0 chan/2

set Fruni/903 framer interfaceName lp/9 e1/0 chan/3

set Fruni/904 framer interfaceName lp/9 e1/0 chan/4

set Fruni/905 framer interfaceName lp/9 e1/0 chan/5
```

5 Add the *DataSigChan* and *Ts014* components and set the signaling channel framer to use the appropriate physical interface. Also set the PSTN interface number.

```
add DataSigChan/90

add DataSigChan/90 Ts014

set DataSigChan/90 Ts014
framer interface lp/9 e1/0 chan/16
```

6 Add the first *channelGroup* component.

```
add DataSigChan/90 callControl channelGroup/1

set DataSigChan/90 callControl
channelGroup/1 channelList 1 2 3

add DataSigChan/90 callControl channelGroup/1
CallingPartyNumber/1111111

add DataSigChan/90 callControl channelGroup/1
CallingPartyNumber/2222222

add DataSigChan/90 callControl channelGroup/1
CallingPartyNumber/3333333

add DataSigChan/90 callControl channelGroup/1
CallingPartyNumber/4444444
```

```
add DataSigChan/90 callControl channelGroup/1
CallingPartyNumber/5555555
```

7   Add the second *channelGroup* component.

```
add DataSigChan/90 callControl channelGroup/2
```

```
set DataSigChan/90 callControl channelGroup/2
channelList 4
```

```
add DataSigChan/90 callControl channelGroup/2
CallingPartyNumber/1231230
```

```
add DataSigChan/90 callControl channelGroup/2
CallingPartyNumber/1231231
```

```
add DataSigChan/90 callControl channelGroup/2
CallingPartyNumber/1231232
```

8   Add the last *channelGroup* component.

```
add DataSigChan/90 callControl channelGroup/3
```

```
set DataSigChan/90 callControl channelGroup/3
channelList 5
```

```
add DataSigChan/90 callControl channelGroup/3
CallingPartyNumber/2222220
```

Procedure job aid

Figure 17
Frame relay ISDN switched access component hierarchy
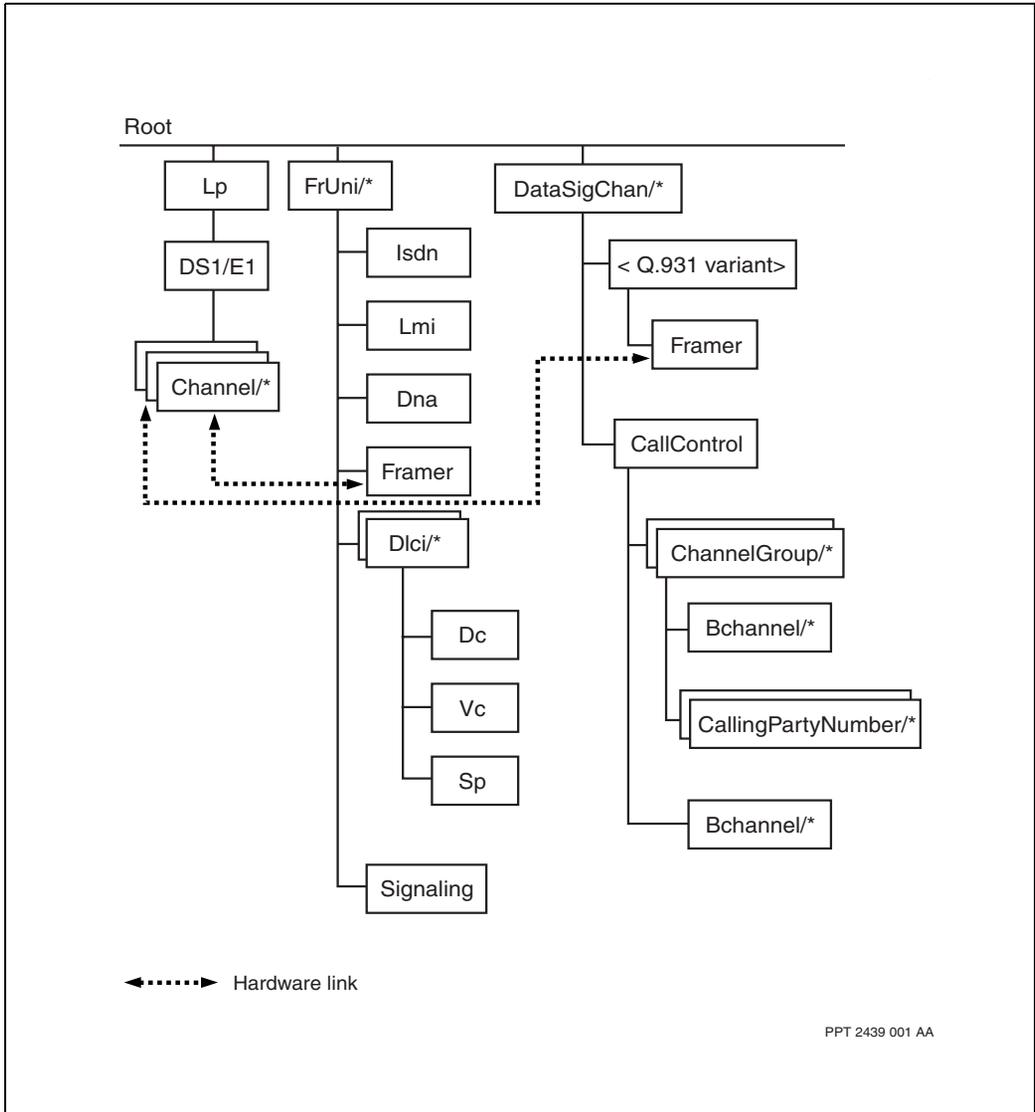
Australian TS014 provisioning details

The following elements determine how to provision frame relay ISDN
switched access to support the Australian TS014 protocol:

- screening indicator values

    — user-provided, verified and passed

    — network provided

- channel mapping

The table "TS014 to Multiservice Switch node channel mapping" (page 59)
shows how the TS014 protocol's channel numbers map to a node's channels
and timeslots.

Table 1
TS014 to Multiservice Switch node channel mapping

Channel number provisioned on a node	Corresponding timeslot numbers provisioned on the node	Corresponding channel numbers used in the TS014 protocol
1	1	1
.	.	.
.	.	.
.	.	.
15	15	15
0 or 16	16	D-channel
17	17	16
.	.	.
.	.	.
.	.	.
31	31	30

European ISDN provisioning details

The European ISDN network uses the CRC-4 procedure. To ensure the PRI link operates properly, set the *crc4Mode* attribute to on.

For example:

```
set Lp/x E1/y crc4Mode on
```

Channel mapping determines how to provision frame relay ISDN switched access to support the European ISDN protocol. The table "European ISDN to Multiservice Switch node channel mapping" (page 60) shows how the European ISDN protocol's channel numbers map to a node's channels and timeslots.

Table 2
European ISDN to Multiservice Switch node channel mapping

Channel number provisioned on a node	Corresponding timeslot numbers provisioned on the node	Corresponding channel numbers used in the European ISDN protocol
1	1	1
.	.	.
.	.	.
.	.	.
15	15	15
0	16	D-channel
17	17	17
.	.	.
.	.	.
.	.	.
31	31	31

Japanese ISDN provisioning details

Channel mapping determines how to provision frame relay ISDN switched access to support the Japanese ISDN protocol. The table "Japanese ISDN to Multiservice Switch node channel mapping" (page 61) shows how the Japanese ISDN protocol's channel numbers map to a node's channels and timeslots.

Table 3
Japanese ISDN to Multiservice Switch node channel mapping

Channel number provisioned on a node	Corresponding timeslot numbers provisioned on the node	Corresponding channel numbers used in the Japanese ISDN protocol
1	1	1
.	.	.
.	.	.
.	.	.
15	15	15
16	16	16
17	17	17
.	.	.
.	.	.
23	23	23
0	24	D-channel

North American ISDN provisioning details

Channel mapping determines how to provision frame relay ISDN switched access to support the North American NI-2 and NIS A211-1 ISDN protocol.

The table "North American ISDN to Multiservice Switch node channel mapping" (page 62) shows how the North American ISDN protocol's channel numbers map to a node's channels and timeslots.

Table 4
North American ISDN to Multiservice Switch node channel mapping

Channel numbers provisioned on a node	Corresponding timeslot numbers provisioned on the node	Corresponding channel numbers used in the Ni2/Nis ISDN protocol
1	1	1
.	.	.
.	.	.
.	.	.
15	15	15
16	16	16
17	17	17
.	.	.
.	.	.
23	23	23
0	24	D-channel

Chapter 4
Switched call connection configuration

Configure switched call connections to configure the options for switched calls on FR UNI and FR NNI.

Switched call connection configuration procedures

This task flow shows you the sequence of procedures you perform to configure switched call connections. To link to any procedure, go to "Switched call connection configuration procedure navigation" (page 64).

Figure 18
Switched call connection configuration procedures

PPT 3348 003 AA
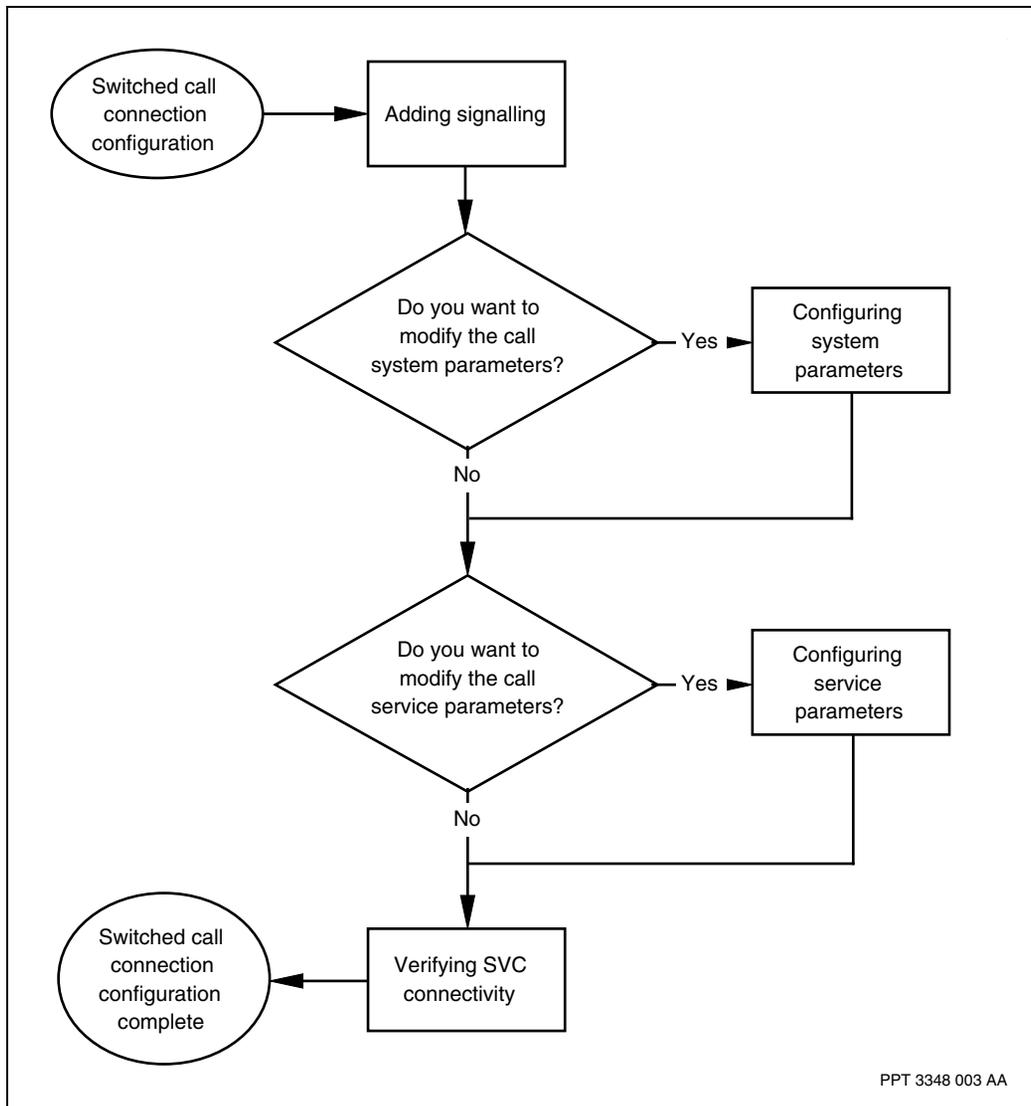
Switched call connection configuration procedure navigation

- "Adding signalling" (page 66)

Adding signalling

Add signalling to configure the FR UNI or FR NNI to support switched call connections.

Procedure steps

1 Add the *Signaling* component.

add FrUni/<FrUni_inst> Sig

add FrNni/<FrNni_inst> Sig

2 Set the DLCI range for permanent and switched calls.

set FrUni/<FrUni_inst> Sig hpd <dlci>

set FrNni/<FrNni_inst> Sig hpd <dlci>

3 Set the DLCI allocation method for switched connections on the FR NNI.

set FrNni/<FrNni_inst> Sig dlciAllocation <point>

4 Optionally, turn accounting on or off for switched calls.

set FrUni/<FrUni_inst> Sig defAcc <on_off>

set FrNni/<FrNni_inst> Sig defAcc <on_off>

Variable definitions

Variable	Value
<dlci>	is the highest DLCI value permitted for a permanent connection.Only PVCs and P-SPVCs with a DLCI value between 0 and <highestPvcDlci> are accepted on the interface. All other values are implicitly reserved for switched connections.
<FrNni_inst>	is the instance number of the FR NNI.
<FrUni_inst>	is the instance number of the FR UNI.
<on_off>	is a setting of on or off, indicating whether accounting is enabled or disabled.
<point>	is the initiation point for dynamic DLCI assignment. The initiation point can be either the high or low end of the configured range.

Procedure job aid

Figure 19
Signalling FR UNI component hierarchy

```
EM
        Software (Sw)
                LogicalProcessorType (Lpt)
                    featurelist (fl)
        FrameRelayUni (FrUni)
            Signaling (Sig)
                    ....
                    highestPvcDlci (hpd)
                    defaultAccounting (defAcc)
```
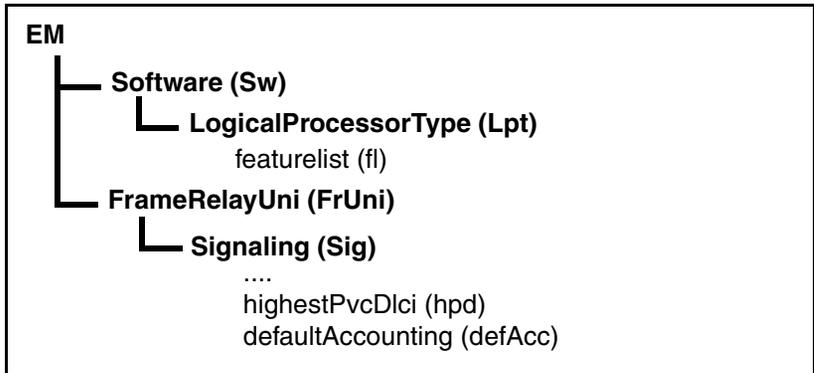
Figure 20
Signalling FR NNI component hierarchy

```
EM
        Software (Sw)
                LogicalProcessorType (Lpt)
                    featureList (fl)
        FrameRelayNni (FrNni)
            Signaling (Sig)
                    ....
                    highestPvcDlci (hpd)
                    dlciAllocation
                    defaultAccounting (defAcc)
```
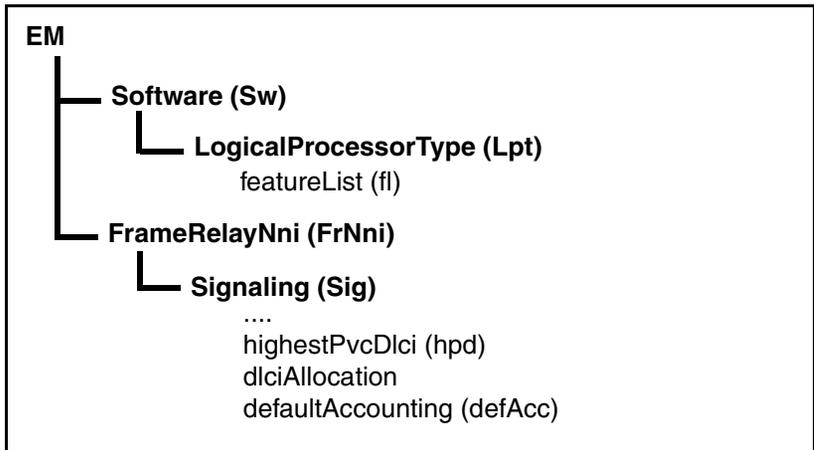
Configuring system parameters

Configure system parameters to define data link layer management for switched calls.

Procedure steps

1   Specify the maximum number of unacknowledged sequenced frames that may be outstanding at one time.

    ```
    set FrUni/<FrUni_inst> Sig ws <wndw_size>
    ```

    ```
    set FrNni/<FrNni_inst> Sig ws <wndw_size>
    ```

2   Specify the number of times a frame can be retransmitted by the SVC data link layer before remedial action is taken.

    ```
    set FrUni/<FrUni_inst> Sig rtl <retran_limit>
    ```

    ```
    set FrNni/<FrNni_inst> Sig rtl <retran_limit>
    ```

3   Specify the length of time to wait for an acknowledgment from the other network for a transmitted frame.

    ```
    set FrUni/<FrUni_inst> Sig at <milli_interval>
    ```

    ```
    set FrNni/<FrNni_inst> Sig at <milli_interval>
    ```

4   Specify the length of time to wait before sending an acknowledgment for a received frame.

    ```
    set FrUni/<FrUni_inst> Sig adt <milli_interval>
    ```

    ```
    set FrNni/<FrNni_inst> Sig adt <milli_interval>
    ```

5   Specify the length of time that the data link can remain idle before polling its peer for status.

    ```
    set FrUni/<FrUni_inst> Sig ipt <milli_interval>
    ```

    ```
    set FrNni/<FrNni_inst> Sig ipt <milli_interval>
    ```

6   Specify the length of time to wait for a response to a call setup message before retransmitting the message.

    ```
    set FrUni/<FrUni_inst> Sig t303 <sec_interval>
    ```

    ```
    set FrNni/<FrNni_inst> Sig t303 <sec_interval>
    ```

7   Specify the length of time the FR UNI waits for a response to a disconnect message before it retransmits the message.

    ```
    set FrUni/<FrUni_inst> Sig t305 <sec_interval>
    ```

8   Specify the length of time to wait for a response to a call release message before retransmitting the message.

```
set FrUni/<FrUni_inst> Sig t308 <sec_interval>
```

```
set FrNni/<FrNni_inst> Sig t308 <sec_interval>
```

9   Specify the length of time to wait for a connect message after receiving a call proceeding message from the other network.

```
set FrUni/<FrUni_inst> Sig t310 <sec_interval>
```

```
set FrNni/<FrNni_inst> Sig t310 <sec_interval>
```

10   Specify the length of time to wait for an acknowledgment of a restart message before the message to the other network.

```
set FrUni/<FrUni_inst> Sig t316 <sec_interval>
```

```
set FrNni/<FrNni_inst> Sig t316 <sec_interval>
```

11   Specify the length of time the local network waits to process a received restart message before generating an alarm.

```
set FrUni/<FrUni_inst> Sig t317 <sec_interval>
```

```
set FrNni/<FrNni_inst> Sig t317 <sec_interval>
```

12   Specify the length of time to wait for a status report from the other network before retransmitting a status enquiry message.

```
set FrUni/<FrUni_inst> Sig t322 <sec_interval>
```

```
set FrNni/<FrNni_inst> Sig t322 <sec_interval>
```

13   Optionally, specify either a public or private network.

```
set FrUni/<FrUni_inst> Sig type <type>
```

```
set FrNni/<FrNni_inst> Sig type <type>
```

Variable definitions

Variable	Value
<FrNni_inst>	is the instance number of the FR NNI.
<FrUni_inst>	is the instance number of the FR UNI.
<milli_interval>	is a time interval in milliseconds.
(Sheet 1 of 2)	

Variable	Value
<retran_limit>	is the maximum number of times a frame can be retransmitted.
<sec_interval>	is a time interval in seconds.
<type>	is the type of network (either public or private).
<wndw_size>	is the maximum number of outstanding unacknowledged sequenced frames allowed at one time.
(Sheet 2 of 2)	

Procedure job aid

Figure 21
System parameters FR UNI component hierarchy

```
EM
    └── FrameRelayUni (FrUni)
            └── Signaling (Sig)
                    ....
                    windowSize (ws)
                    retransmitLimit (rtl)
                    ackTimer (at)
                    ackDelayTimer (adt)
                    idleProbeTimer (ipt)
                    callSetupTimer (t303)
                    disconnectTimer (t305)
                    releaseTimer (t308)
                    callProceedingTimer (t310)
                    restartReqTimer (t316)
                    restartRcvTimer (t317)
                    statusEnqTimer (t322)
                    networkType (type)
```
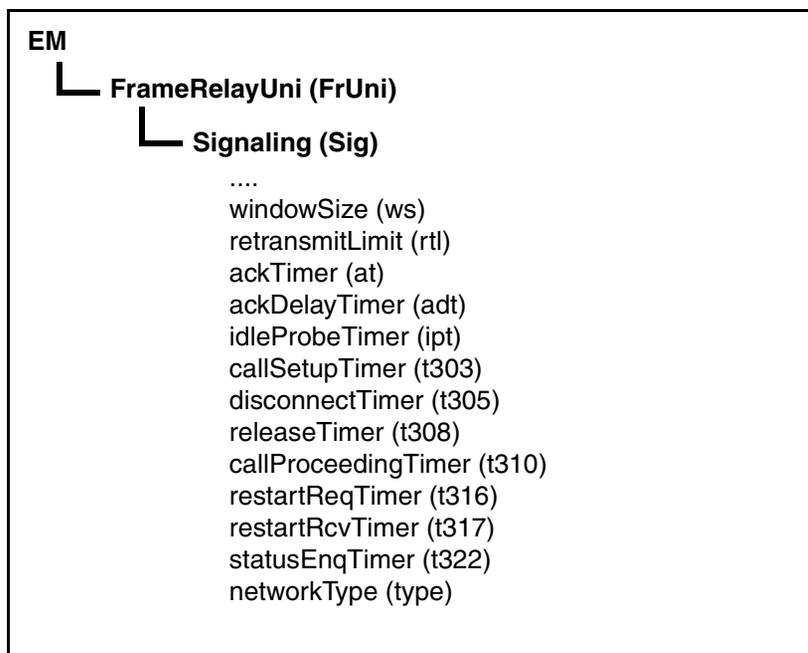
Figure 22
System parameters FR NNI component hierarchy

```
EM
    └─ FrameRelayNni (FrNni)
          └─ Signaling (Sig)
                 ....
                 windowSize (ws)
                 retransmitLimit (rtl)
                 ackTimer (at)
                 ackDelayTimer (adt)
                 idleProbeTimer (ipt)
                 callSetupTimer (t303)
                 releaseTimer (t308)
                 callProceedingTimer (t310)
                 restartReqTimer (t316)
                 restartRcvTimer (t317)
                 statusEnqTimer (t322)
                 networkType (type)
```
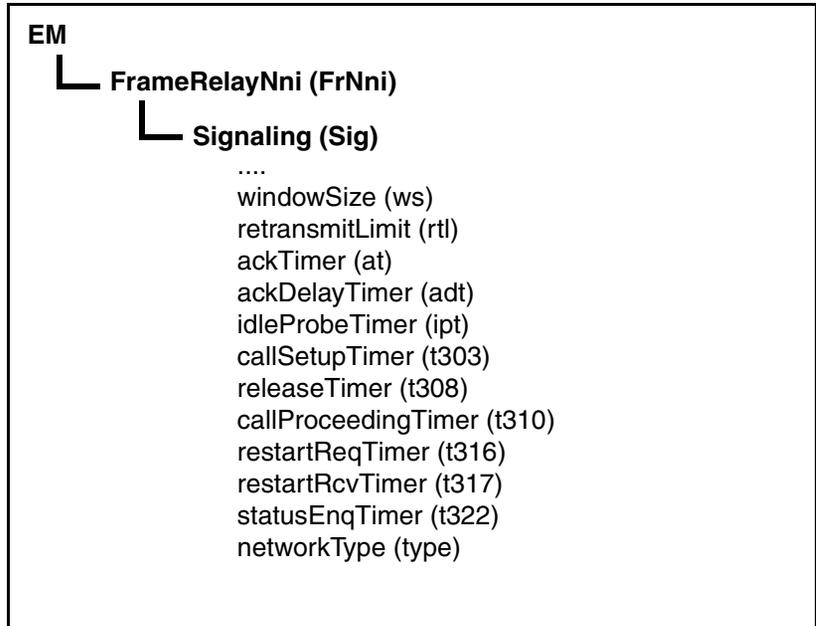
Configuring service parameters

Configure service parameters to define call signalling options for switched calls.

Procedure steps

1 Specify the maximum number of octets permitted in the DLCI information field for switched connections on this interface.

`set FrUni/<FrUni_inst> Sig dn1 <size>`

`set FrNni/<FrNni_inst> Sig dn1 <size>`

2 Specify the default maximum number of octets permitted in the DLCI information field for switched connections on this interface.

`set FrUni/<FrUni_inst> Sig defaultdn1 <def_size>`

`set FrNni/<FrNni_inst> Sig defaultdn1 <def_size>`

3 Specify the default committed information rate (CIR) for the user to network direction.

`set FrUni/<FrUni_inst> Sig defaultCir <def_cir>`

`set FrNni/<FrNni_inst> Sig defaultCir <def_cir>`

4 Specify the default committed burst size (Bc) for the user to network direction.

`set FrUni/<FrUni_inst> Sig defaultBc <def_bc>`

`set FrNni/<FrNni_inst> Sig defaultBc <def_bc>`

5 Specify the default excess burst size (Be) for the user to network direction.

`set FrUni/<FrUni_inst> Sig defaultBe <def_be>`

`set FrNni/<FrNni_inst> Sig defaultBe <def_be>`

6 Specify whether rate enforcement is on or off for switched connections.

`set FrUni/<FrUni_inst> Sig re <re_on>`

`set FrNni/<FrNni_inst> Sig re <re_on>`

7 Specify whether rate adaptation is on or off for switched connections.

`set FrUni/<FrUni_inst> Sig ra <ra_on>`

`set FrNni/<FrNni_inst> Sig ra <ra_on>`

8   If you have enabled rate adaptation, specify the rate adaptation sensitivity index for switched connections.

```
set FrUni/<FrUni_inst> Sig rasens <ra_sens>

set FrNni/<FrNni_inst> Sig rasens <ra_sens>
```

9   Specify whether BCI bits are set to report external congestion to the Nortel Networks Multiservice Switch network, based on incoming BECN bits.

```
set FrUni/<FrUni_inst> Sig updBci <updbci_on>

set FrNni/<FrNni_inst> Sig updBci <updbci_on>
```

Variable definitions

Variable	Value
<def_bc>	is the committed burst size in bits used for call parameter negotiation when the user's setup message does not specify the Bc.
<def_be>	is the excess burst size in bits used for call parameter negotiation when the user's setup message does not specify the Be.
<def_cir>	is the committed information rate in bits/s used for call parameter negotiation when the user's setup message does not specify the CIR.
<def_size>	is the maximum size in octets used for call parameter negotiation when the user's setup message does not specify the maximum frame size.
<FrUni_inst>	is the instance number of the FR UNI.
<FrNni_inst>	is the instance number of the FR NNI.
<ra_sens>	is the sensitivity of the rate adaptation system.
<ra_on>	is a setting of on or off, indicating that rate adaptation is enabled or disabled.
<re_on>	is a setting of on or off, indicating that rate enforcement is enabled or disabled.
(Sheet 1 of 2)	

Variable	Value
<size>	is the maximum size in octets that may be included in the information field of the user DLCI for any switched connections using the interface. Its value may not be smaller than the defaultMaximumFrameSize.
<updbci_on>	is a setting of on or off, indicating that backward congestion indication (BCI) is enabled or disabled.
(Sheet 2 of 2)	

Procedure job aid

Figure 23
Service parameters FR UNI component hierarchy

```
EM
   └── FrameRelayUni (FrUni)

         └── Signaling (Sig)
                  ....
                  maximumFrameSize (dn1)
                  defaultMaximumFrameSize (defaultdn1)
                  defaultCommittedInformationRate (defaultCir)
                  defaultCommittedBurstSize (defaultBc)
                  defaultExcessBurstSize (defaultBe)
                  rateEnforcement (re)
                  rateAdaptation (ra)
                  raSensitivity (rasens)
                  updateBCI (updBci)
```
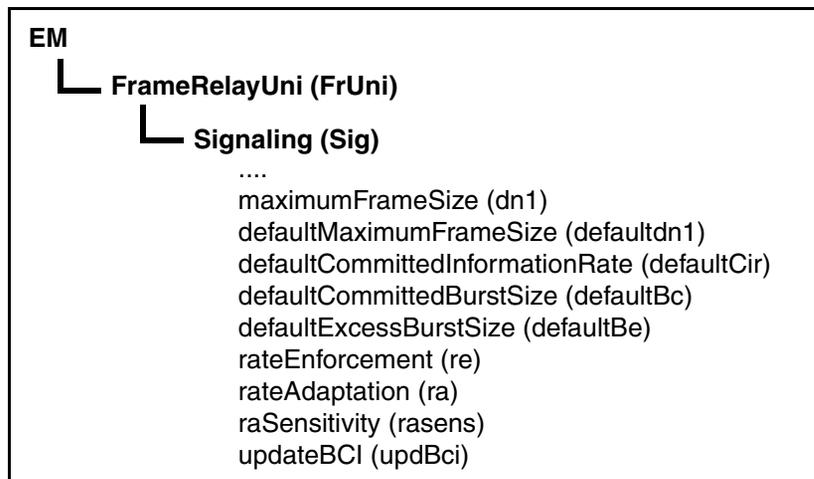
Figure 24
Service parameters FR NNI component hierarchy

```
EM
    └── FrameRelayNni (FrNni)
            └── Signaling (Sig)
                    ....
                    maximumFrameSize (dn1)
                    defaultMaximumFrameSize (defaultdn1)
                    defaultCommittedInformationRate (defaultCir)
                    defaultCommittedBurstSize (defaultBc)
                    defaultExcessBurstSize (defaultBe)
                    rateEnforcement (re)
                    rateAdaptation (ra)
                    raSensitivity (rasens)
                    updateBCI (updBci)
```
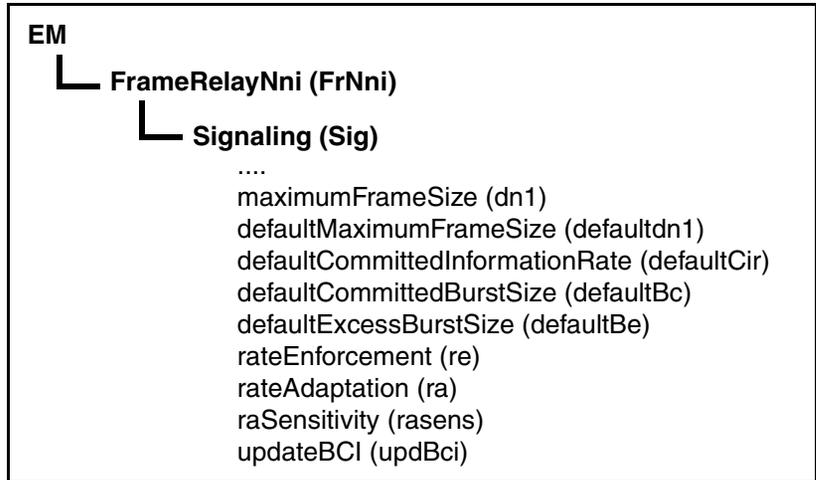
Verifying SVC connectivity

Verify SVC connectivity to confirm that the SVC has been configured correctly and is operating normally.

Procedure steps

1   Verify the current state of the SVC data link layer for each interface.

   display FrUni/<FrUni_inst> Sig currentState

   display FrNni/<FrNni_inst> Sig currentState

   A value of informationTransfer indicates that the link is up.

2   Verify that the interface is sending and receiving call setup messages.

   display FrUni/<FrUni_inst> Sig Statistics

   display FrNni/<FrNni_inst> Sig Statistics

   The *inCalls* and *outCalls* attributes indicate the number of incoming and outgoing SVC calls on the interface.

3   Verify that the SVC is up at both ends of the connection.

   display FrUni/<FrUni_inst> Sig LapfStatistics

   display FrNni/<FrNni_inst> Sig LapfStatistics

   The *iFramesTransmitted* and *iFramesReceived* attributes indicate the number of incoming and outgoing I-frames on the data link layer.

4   Verify that a dynamic *Dlci* component has been created under the interface for the SVC.

   list FrUni/<FrUni_inst>

   list FrNni/<FrNni_inst>

5   Verify the current state of the virtual circuit for the interface.

   display FrUni/<FrUni_inst> Dlci/<dlci_no> Vc state

   display FrNni/<FrNni_inst> Dlci/<dlci_no> Vc state

   A value of dataTransferP4 indicates that the connection is up and in data transfer phase.

6   Ensure that the calling and called ends are properly indicated in the call data statistics at the interface.

   display FrUni/<FrUni_inst> Dlci/<dlci_no> Vc callData

```
display FrNni/<FrNni_inst> Dlci/<dlci_no> Vc callData
```

The *callingNpi* and *callingDna* attributes indicate the calling interface. The *calledNpi* and *calledDna* attributes indicate the called interface.

Variable definitions

Variable	Value
<dlci_no>	is the local instance number of the dynamic DLCI.
<FrNni_inst>	is the instance number of the FR NNI.
<FrUni_inst>	is the instance number of the FR UNI.

Procedure job aid

Figure 25
SVC connectivity FR UNI component hierarchy
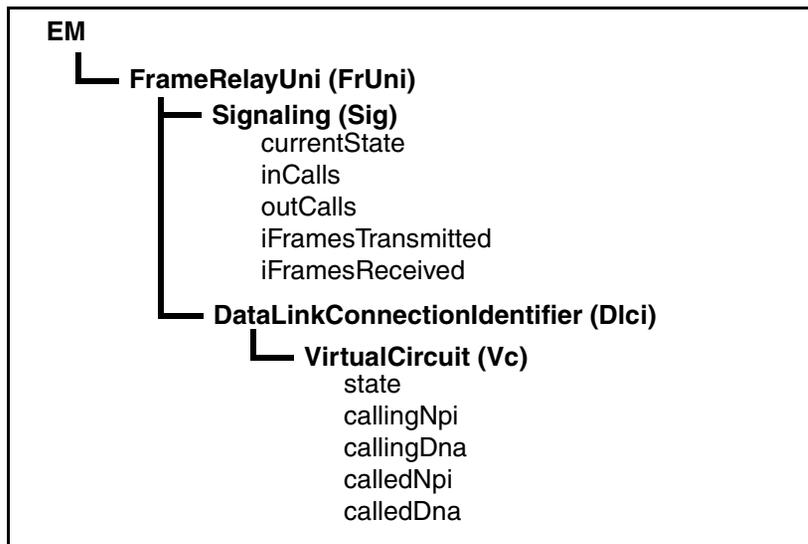
Figure 26
SVC connectivity FR NNI component hierarchy

```
EM
    └─ FrameRelayNni (FrNni)
            ├─ Signaling (Sig)
            │       currentState
            │       inCalls
            │       outCalls
            │       iFramesTransmitted
            │       iFramesReceived
            │
            └─ DataLinkConnectionIdentifier (Dlci)
                    └─ VirtualCircuit (Vc)
                            state
                            callingNpi
                            callingDna
                            calledNpi
                            calledDna
```
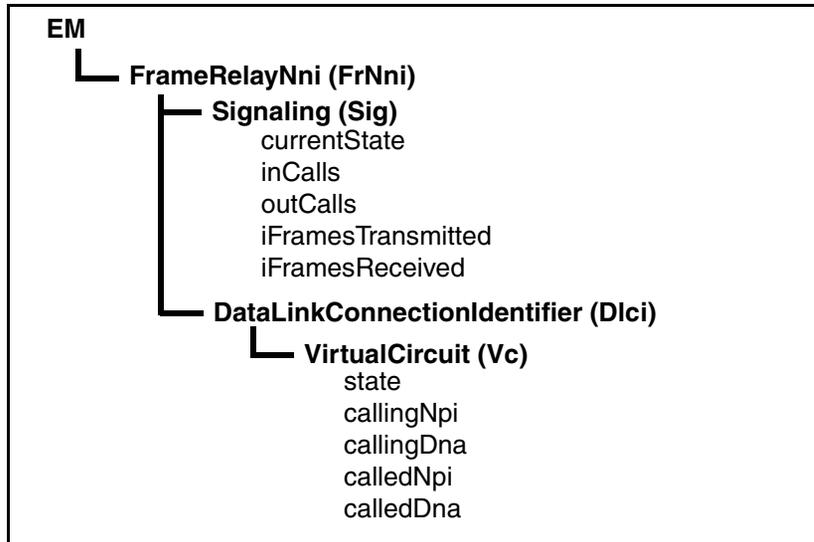
Chapter 5
Permanent call connection configuration

Configure permanent call connections to create connections and configure call options for permanent call connections for FR UNI and FR NNI.

Permanent call connection configuration procedures

This task flow shows you the sequence of procedures you perform to configure permanent call connections. To link to any procedure, go to "Permanent call connection configuration procedure navigation" (page 81).

Figure 27
Permanent call connection configuration procedures
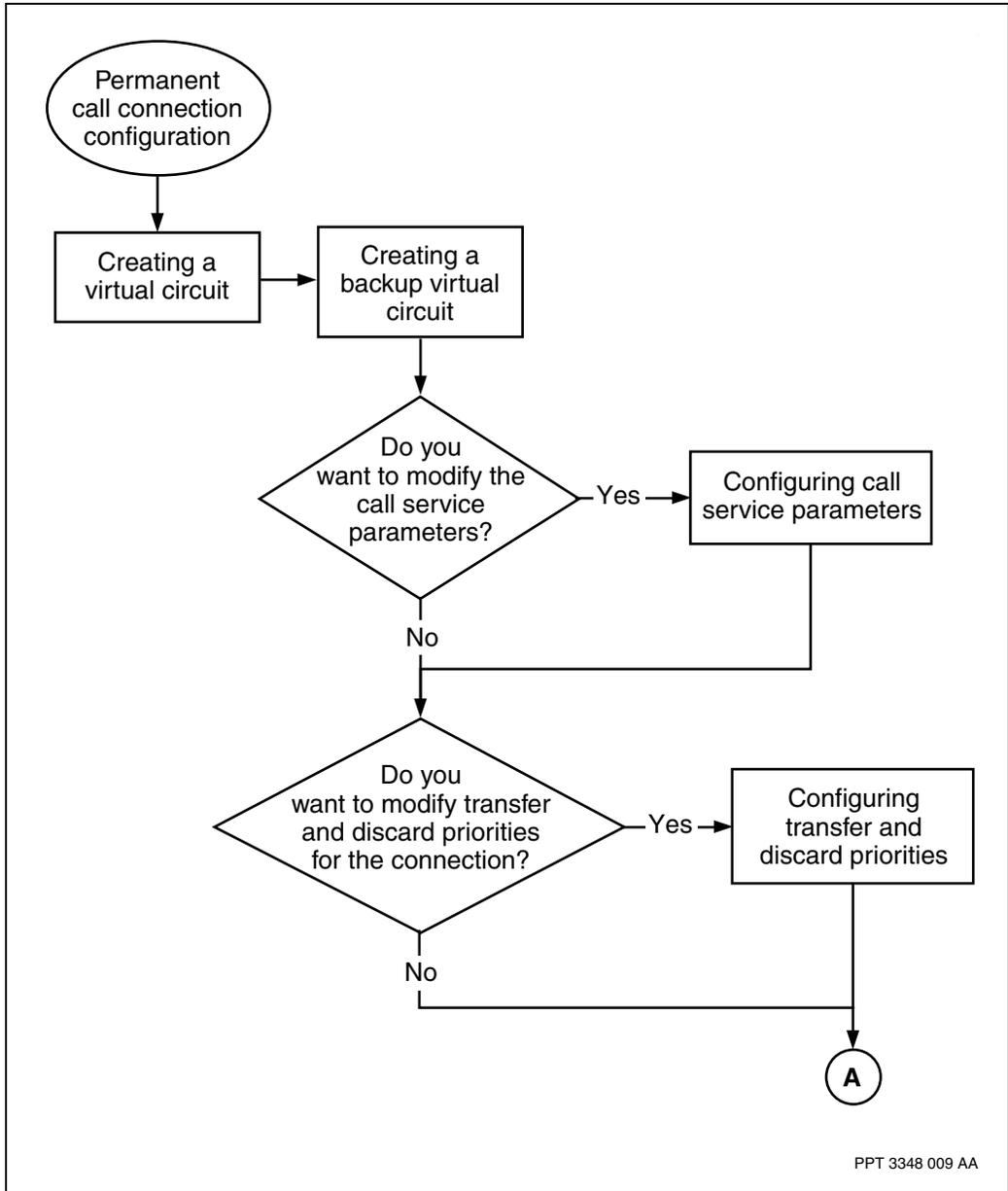
Figure 28
Permanent call connection configuration procedures (continued)

MSS 3348 009 AB
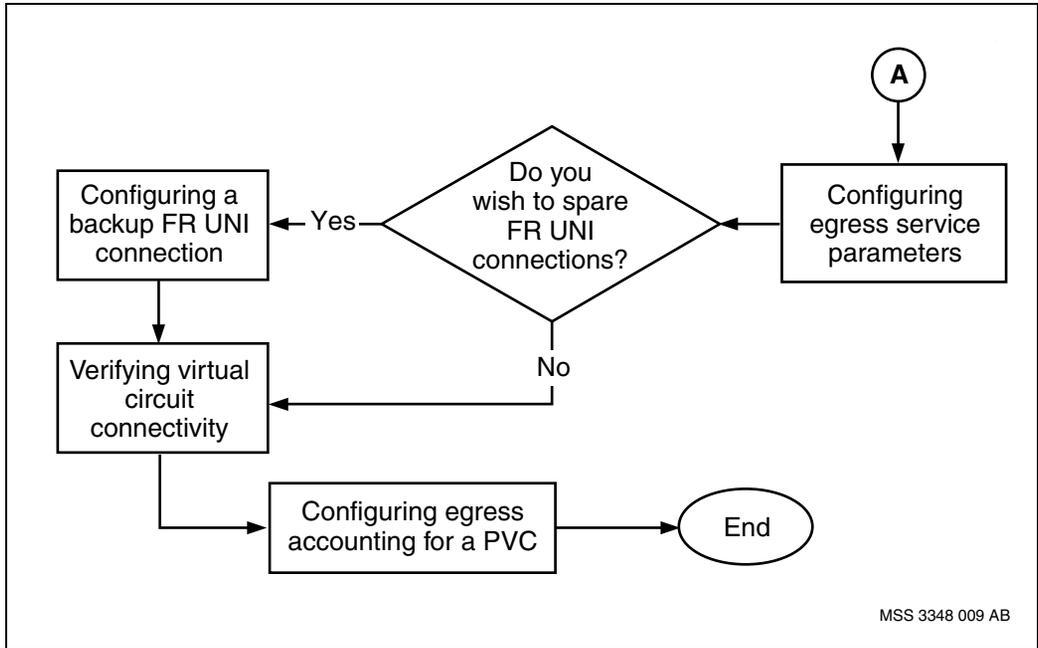
Permanent call connection configuration procedure navigation

Creating a virtual circuit

Create a virtual circuit to establish a switched (SPVC) or non-switched (PVC) permanent call connection.

Prerequisites

- Ensure that the local and remote DNA and DLCI values are consistent at each end of the connection.

Procedure steps

1   Create a DLCI with a unique instance value.

```
add FrUni/<FrUni_inst> Dlci/<dlci_no>

add FrNni/<FrNni_inst> Dlci/<dlci_no>
```

2   Set the numbering plan indicator of the remote end of the connection.

```
set FrUni/<FrUni_inst> dlci/<dlci_no> Dc rnpi
<rem_npi>

set FrNni/<FrNni_inst> dlci/<dlci_no> Dc rnpi
<rem_npi>
```

3   Set the data network address of the remote end of the connection.

```
set FrUni/<FrUni_inst> dlci/<dlci_no> Dc rdna
<rem_dna>

set FrNni/<FrNni_inst> dlci/<dlci_no> Dc rdna
<rem_dna>
```

4   Set the DLCI of the remote end of the connection.

```
set FrUni/<FrUni_inst> dlci/<dlci_no> Dc rdlci
<rem_dlci>

set FrNni/<FrNni_inst> dlci/<dlci_no> Dc rdlci
<rem_dlci>
```

5   Define the type of VC call.

```
set FrUni/<FrUni_inst> dlci/<dlci_no> Dc type
<vc_calltype>

set FrNni/<FrNni_inst> dlci/<dlci_no> Dc type
<vc_calltype>
```

If you define the VC as a PVC or SPVC slave, the remote end must be defined as a PVC or SPVC master.

Variable definitions

Variable	Value
<dlci_no>	is the instance number for the DLCI.
<FrNni_inst>	is the instance number of the FR NNI.
<FrUni_inst>	is the instance number of the FR UNI.
<rem_dlci>	is the destination DLCI for the connection.
<rem_dna>	is the address of the destination for the connection.
<rem_npi>	is the numbering plan indicator for the destination address.
<vc_calltype>	is the type of VC call based on whether this is the source or destination DLCI for a PVC or SPVC.

Procedure job aid

Figure 29
Virtual circuit FR UNI component hierarchy

```
EM
    └── FrameRelayUni (FrUni)
            └── DataLinkConnectionIdentifier (Dlci)
                    ├── DirectCall (Dc)
                    │       remoteNpi (rnpi)
                    │       remoteDna (rdna)
                    │       remoteDlci (rdlci)
                    │       type
                    │       ....
                    ├── ServiceParameters (Sp)
                    └── EgressServiceParameters (EgressSp)
```
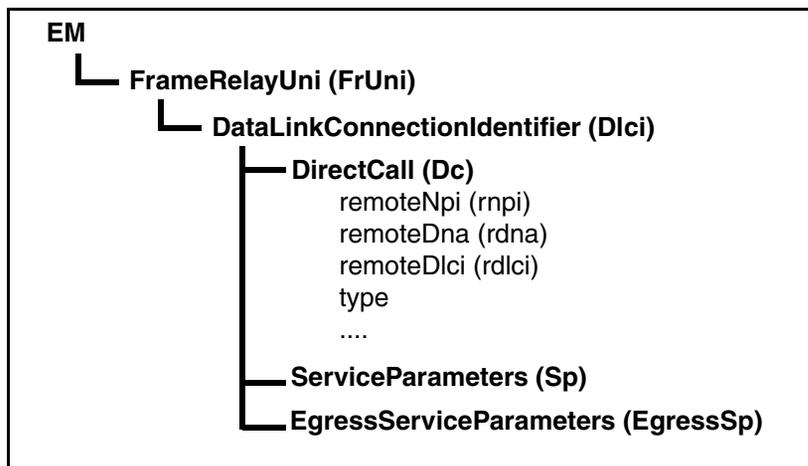
Figure 30
Virtual circuit FR NNI component hierarchy

```
EM
    └── FrameRelayNni (FrNni)
            └── DataLinkConnectionIdentifier (Dlci)
                    ├── DirectCall (Dc)
                    │       remoteNpi (rnpi)
                    │       remoteDna (rdna)
                    │       remoteDlci (rdlci)
                    │       type
                    │       ....
                    ├── ServiceParameters (Sp)
                    └── EgressServiceParameters (EgressSp)
```
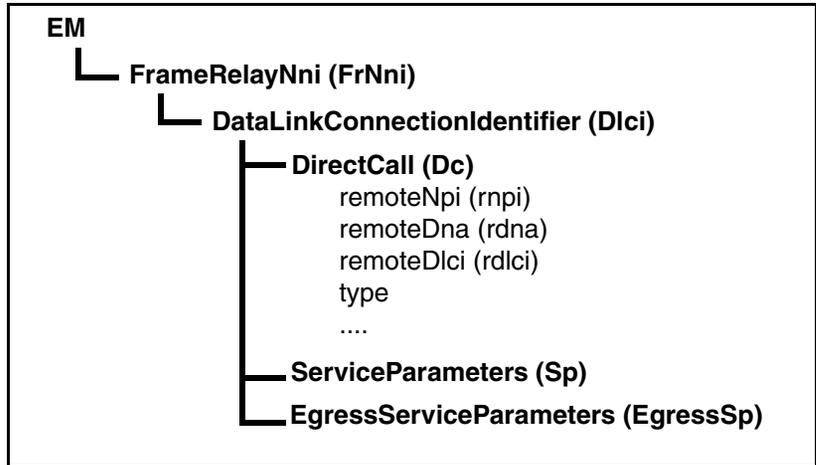
Creating a backup virtual circuit

Create a backup virtual circuit to configure a backup connection for permanent connection resiliency.

Prerequisites

- Ensure that the local and remote DNA and DLCI values are consistent at each end of the connection.

Procedure steps

1   Create a DLCI with a unique instance value.

    add FrUni/<FrUni_back> Dlci/<dlci_no>

    add FrNni/<FrNni_back> Dlci/<dlci_no>

2   Specify the numbering plan indicator for the remote end of the connection.

    **set FrUni/<FrUni_back> dlci/<dlci_no> Dc rnpi
    <rem_npi>**

    **set FrNni/<FrNni_back> dlci/<dlci_no> Dc rnpi
    <rem_npi>**

3   Specify the data network address of the remote end of the connection.

    **set FrUni/<FrUni_back> dlci/<dlci_no> Dc rdna
    <rem_dna>**

    **set FrNni/<FrNni_back> dlci/<dlci_no> Dc rdna
    <rem_dna>**

4   Set the DLCI of the remote end of the PVC connection.

    **set FrUni/<FrUni_back> dlci/<dlci_no> Dc rdlci
    <rem_dlci>**

    **set FrNni/<FrNni_back> dlci/<dlci_no> Dc rdlci
    <rem_dlci>**

5   Define the connection as a backup VC. Make sure that the backup's configured parameters match those of the end it is backing up.

    **set FrUni/<FrUni_back> dlci/<dlci_no> Dc type
    <backuppvc_type>**

    **set FrNni/<FrNni_back> dlci/<dlci_no> Dc type
    <backuppvc_type>**

Variable definitions

Variable	Value
<backupvc_type>	is the type of backup VC call depending if this is a PVC or SPVC connection.
<dlci_no>	is the instance number for the DLCI.
<FrNni_back>	is the instance number of the backup FR NNI.
<FrUni_back>	is the instance number of the backup FR UNI.
<rem_dlci>	is the destination DLCI for the PVC connection.
<rem_dna>	is the address of the destination for the connection.
<rem_npi>	is the numbering plan indicator for the destination address.

Procedure job aid

Figure 31
Backup virtual circuit FR UNI component hierarchy

```
EM
  └─ FrameRelayUni (FrUni)
        └─ DataLinkConnectionIdentifier (Dlci)
              ├─ DirectCall (Dc)
              │      remoteNpi (rnpi)
              │      remoteDna (rdna)
              │      remoteDlci (rdlci)
              │      type
              │      ....
              ├─ ServiceParameters (Sp)
              └─ EgressServiceParameters (EgressSp)
```
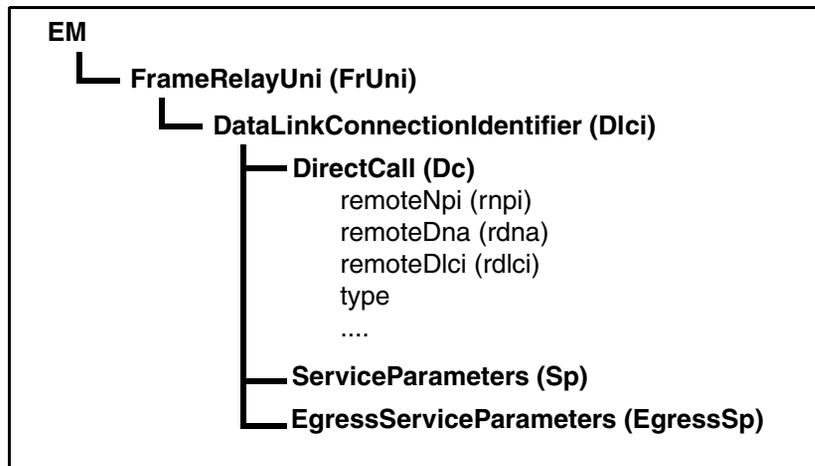
Figure 32
Backup virtual circuit FR NNI component hierarchy

```
EM
    └── FrameRelayNni (FrNni)
            └── DataLinkConnectionIdentifier (Dlci)
                    ├── DirectCall (Dc)
                    │       remoteNpi (rnpi)
                    │       remoteDna (rdna)
                    │       remoteDlci (rdlci)
                    │       type
                    │       ....
                    ├── ServiceParameters (Sp)
                    └── EgressServiceParameters (EgressSp)
```
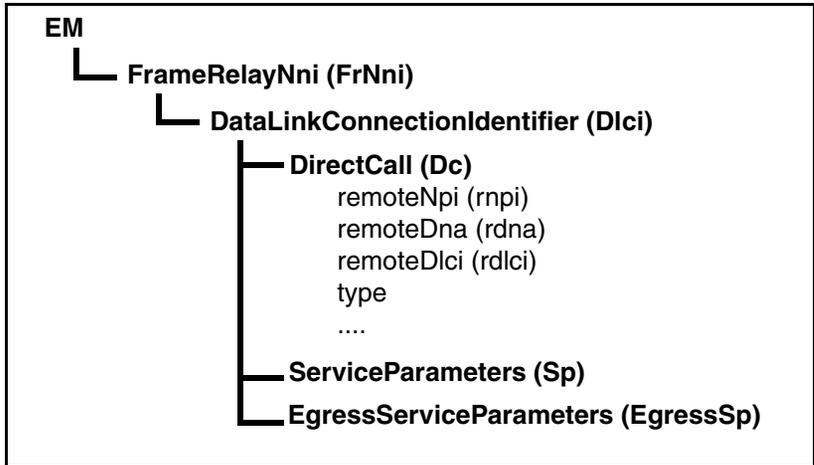
Configuring call service parameters

Configure call service parameters to change and specify how call features are defined for a permanent connection.

Procedure steps

1 Specify the connection's maximum frame size for ingress traffic.

   `set FrUni/<FrUni_inst> dlci/<dlci_no> Sp n203 <size>`

   `set FrNni/<FrNni_inst> dlci/<dlci_no> Sp n203 <size>`

2 Turn rate enforcement either on or off.

   `set FrUni/<FrUni_inst> dlci/<dlci_no> Sp re <rate_enfon>`

   `set FrNni/<FrNni_inst> dlci/<dlci_no> Sp re <rate_enfon>`

3 Specify the connection's committed information rate.

   `set FrUni/<FrUni_inst> dlci/<dlci_no> Sp cir <sp_cir>`

   `set FrNni/<FrNni_inst> dlci/<dlci_no> Sp cir <sp_cir>`

4 Specify the connection's committed burst size.

   `set FrUni/<FrUni_inst> dlci/<dlci_no> Sp bc <sp_bc>`

   `set FrNni/<FrNni_inst> dlci/<dlci_no> Sp bc <sp_bc>`

5 Specify the connection's excess burst size.

   `set FrUni/<FrUni_inst> dlci/<dlci_no> Sp be <sp_be>`

   `set FrNni/<FrNni_inst> dlci/<dlci_no> Sp be <sp_be>`

6 Turn rate adaptation either on or off.

   `set FrUni/<FrUni_inst> dlci/<dlci_no> Sp ra <rate_adapton>`

   `set FrNni/<FrNni_inst> dlci/<dlci_no> Sp ra <rate_adapton>`

7 If you have enabled rate adaptation, set the rate adaptation sensitivity index.

   `set FrUni/<FrUni_inst> dlci/<dlci_no> Sp rasens <ra_val>`

```
set FrNni/<FrNni_inst> dlci/<dlci_no> Sp rasens
<ra_val>
```

8   Turn accounting either on or off for the connection.

```
set FrUni/<FrUni_inst> dlci/<dlci_no> Sp acc
<accnt_on>
```

```
set FrNni/<FrNni_inst> dlci/<dlci_no> Sp acc
<accnt_on>
```

9   Specify whether BCI bits are set to report external congestion to the Nortel Networks Multiservice Switch network, based on incoming BECN bits.

```
set FrUni/<FrUni_inst> dlci/<dlci_no> Sp updBci
<BCI_report>
```

```
set FrNni/<FrNni_inst> dlci/<dlci_no> Sp updBci
<BCI_report>
```

10   Specify the value of the correlation tag you want on a specific DLCI.

```
set FrUni/<FrUni_inst> dlci/<dlci_no> Sp
correlationTag <value_string>
```

```
set FrNni/<FrNni_inst> dlci/<dlci_no> Sp
correlationTag <value_string>
```

The default value for a correlation tag identifier is an empty string. If you do not want to use the circuit id functionality, do not change the default value.

11   If necessary, delete a previously provisioned correlation tag value.

```
set FrUni/<FrUni_inst> dlci/<dlci_no> Sp
correlationTag ""
```

```
set FrNni/<FrNni_inst> dlci/<dlci_no> Sp
correlationTag <"">
```

If required for correlation purposes, you need to manually provision a unique correlation tag with the same value at both endpoints of the connection. This provisioning task is not system enforced.

12   For FR UNI, specify whether a DLCI operates as an FRF.1 DTE or as an FRF.12 DTE.

```
set FrUni/<FrUni_inst> dlci/<dlci_no> Sp frf12EndToEnd
<frf12endtoend_on>
```

13 Specify the maximum size of data fragments to be transmitted if
frf12EndToEnd has been enabled.

```
set FrUni/<FrUni_inst> dlci/<dlci_no> Sp
frf12FragmentSize <frf12fragmentsize>
```

Variable definitions

Variable	Value
<"">	is a pair of double quotation marks used to indicate an empty string.
<acnt_on>	is a setting of on or off for accounting on the DLCI.
<BCI_report>	indicates whether BCI bits are set to reflect external congestion for the DLCI.
	If you have turned rate adaptation on, leave this attribute off. When the *updateBCI* attribute is on, rate adaptation is assumed to occur at the ingress of a connected FR UNI or FR NNI.
<dlci_no>	is the instance number of the DLCI.
<frf12endtoend_on>	is a setting of off to specify that the DLCI act as an FRF.1 DTE, or on to specify that the DLCI act as an end-to-end FRF.12 DTE. The default setting is off.
<FrNni_inst>	is the instance number of the FR NNI.
<FrUni_inst>	is the instance number of the FR UNI.
<frf12fragmentsize>	is the maximum data fragment size.
	When *frf12EndToEnd* is turned on, you must determine the maximum size of the data fragments that will be transmitted by Nortel Networks Multiservice Switch frame relay. Available values in bytes are 40, 48, 60, 80, 120, 240, and 480.
<ra_val>	is the sensitivity of the rate adaptation system.
<rate_adapton>	is a setting of on or off indicating that rate adaptation for the DLCI is enabled or disabled.
<rate_enfonf>	is a setting of on or off indicating that rate enforcement for the DLCI is enabled or disabled.
<size>	is the maximum frame size in bytes.
<sp_bc>	is the committed burst size for the DLCI in bits.
(Sheet 1 of 2)	

Variable	Value
<sp_be>	is the excess burst size for the DLCI in bits.
<sp_cir>	is the committed information rate for the DLCI in bits/s.
<value_string>	is an alphanumeric string of up to 32 ASCII digits that identifies the connection.
(Sheet 2 of 2)	

Procedure job aid

Figure 33
Call service parameters FR UNI component hierarchy

Em

   └── **FrameRelayUni (FrUni)**

       └── **DataLinkConnectionIdentifier**

          └── **ServiceParameters (Sp)**
             maximumFrameSize (n203)
             rateEnforcement (re)
             committedInformationRate (cir)
             committedBurstSize (bc)
             excessBurstSize (be)
             measurementInterval (t)
             rateAdaption (ra)
             accounting (ac)
             raSensitivity (rasens)
             udpateBCI (updBCI)
             correlationTag (ctag)
             frf12EndtoEnd (frf12)
             frf12FragmentSize (fragSize)

PPT 3311 002 AA
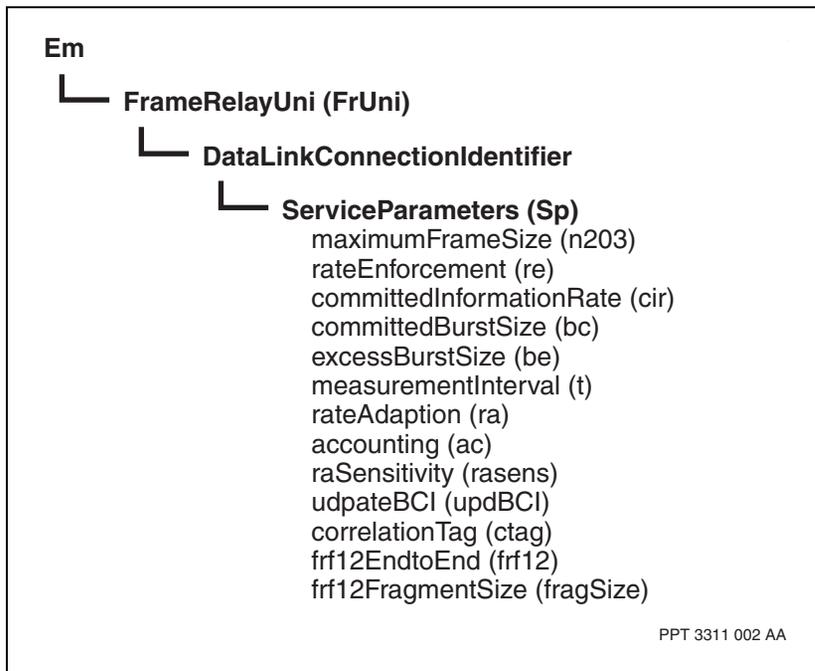
Figure 34
Call service parameters FR NNI component hierarchy

```
EM
     └── FrameRelayNni (FrNni)
              └── DataLinkConnectionIdentifier (Dlci)
                       └── ServiceParameters (Sp)
                               maximumFrameSize (n203)
                               rateEnforcement (re)
                               committedInformationRate (cir)
                               committedBurstSize (bc)
                               excessBurstSize (be)
                               measurementInterval (t)
                               rateAdaptation (ra)
                               raSensitivity (rasens)
                               accounting (ac)
                               updateBCI (updBci)
                               correlationTag (ctag)
```
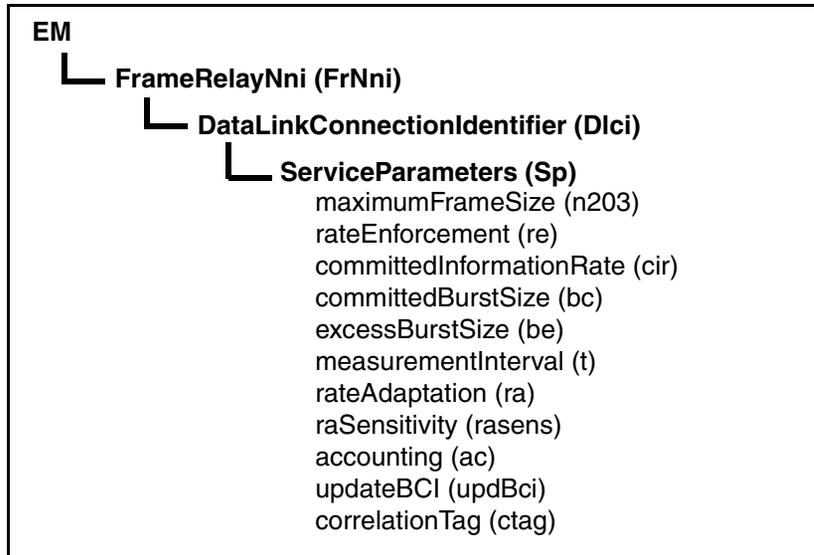
Configuring transfer and discard priorities

Configure transfer and discard priorities to define how the discard and transfer priorities are set and how they are applied to frames on a permanent connection.

Procedure steps

1   Set the transfer priority for outgoing frames on this DLCI.

    ```
    set FrUni/<FrUni_inst> dlci/<dlci_no> Dc tp <tp>
    ```

    ```
    set FrNni/<FrNni_inst> dlci/<dlci_no> Dc tp <tp>
    ```

2   Set the discard priority for outgoing frames on this DLCI.

    ```
    set FrUni/<FrUni_inst> dlci/<dlci_no> Dc dp <dp>
    ```

    ```
    set FrNni/<FrNni_inst> dlci/<dlci_no> Dc dp <dp>
    ```

3   Specify whether the discard priority applies to outgoing frames with the DE bit set on this DLCI.

    ```
    set FrUni/<FrUni_inst> dlci/<dlci_no> Dc ddp <de>
    ```

    ```
    set FrNni/<FrNni_inst> dlci/<dlci_no> Dc ddp <de>
    ```

4   Specify whether the IP TOS values in IP packets are mapped to predefined frame relay discard and emission priorities for traffic on this DLCI.

    ```
    set FrUni/<FrUni_inst> dlci/<dlci_no> Dc frIpTm
    <mapto_QOS>
    ```

    ```
    set FrNni/<FrNni_inst> dlci/<dlci_no> Dc frIpTm
    <mapto_QOS>
    ```

    This attribute applies to PVC connections only.

5   Specify the data path to be used by virtual circuits on this DLCI.

    ```
    set FrUni/<FrUni_inst> dlci/<dlci_no> Dc dpa <path>
    ```

    ```
    set FrNni/<FrNni_inst> dlci/<dlci_no> Dc dpa <path>
    ```

Variable definitions

Variable	Value
<de>	indicates whether the discard priority applies to DE bit frames.
<dlci_no>	is the instance number for the DLCI.
<dp>	is the discard priority setting for outgoing calls using the DLCI.
<FrNni_inst>	is the instance number of the FR NNI.
<FrUni_inst>	is the instance number of the FR UNI.
<mapto_QOS>	indicates whether the FR NNI maps the IP TOS field in IP packets to specific QoS parameters.
<path>	is the data path to be used by virtual circuits on this DLCI.
<tp>	is the transfer priority to the local network for outgoing calls on this DLCI.

Procedure job aid

Figure 35
Transfer and discard priorities FR UNI component hierarchy

```
EM
    └─ FrameRelayUni (FrUni)
           └─ DataLinkConnectionIdentifier (Dlci)
                   └─ DirectCall (Dc)
                          ....
                          transferPriority (tp)
                          discardPriority (dp)
                          deDiscardPriority (ddp)
                          frIpTrafficManagement (frIpTm)
                          dataPath (dpa)
```
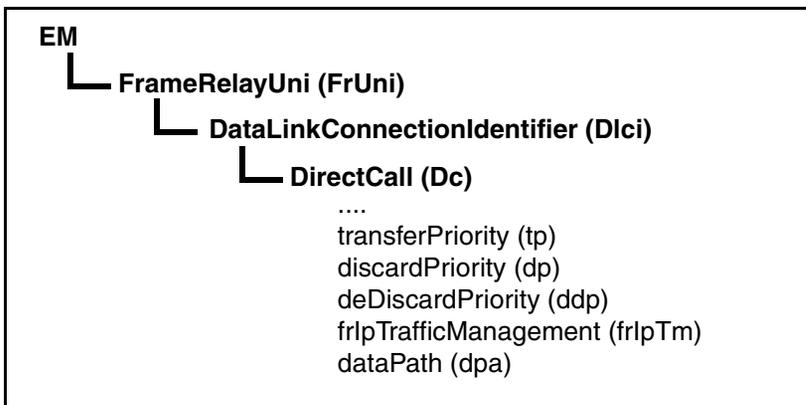
Figure 36
Transfer and discard priorities FR NNI component hierarchy

```
EM
    └── FrameRelayNni (FrNni)
            └── DataLinkConnectionIdentifier (Dlci)
                    └── DirectCall (Dc)
                            ....
                            transferPriority (tp)
                            discardPriority (dp)
                            deDiscardPriority (ddp)
                            frIpTrafficManagement (frIpTm)
                            dataPath (dpa)
```
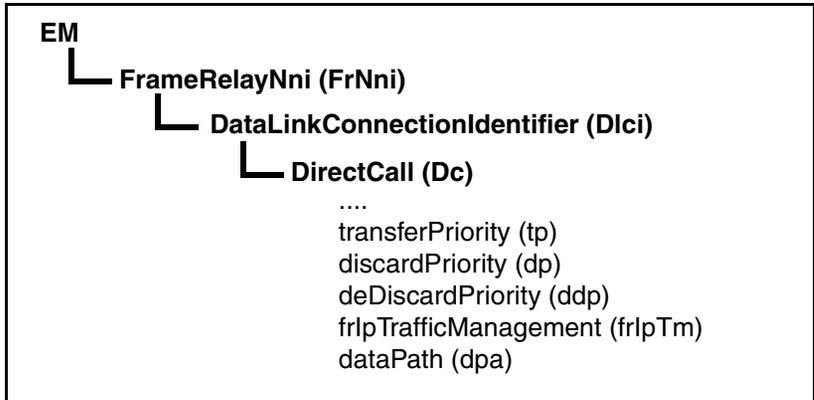
Configuring egress service parameters

Configure egress service parameters to define egress call features differently than the ingress services for a permanent connection.

Procedure steps

1 Set the egress service parameters to be different from the ingress service parameters.

```
set FrUni/<FrUni_inst> Dlci/<dlci_no> EgressSp
sameAsSP <no>
```

```
set FrNni/<FrNni_inst> Dlci/<dlci_no> EgressSp
sameAsSP <no>
```

2 Specify the connection's committed information rate (CIR) in the egress direction.

```
set FrUni/<FrUni_inst> Dlci/<dlci_no> EgressSp cir
<egsp_cir>
```

```
set FrNni/<FrNni_inst> Dlci/<dlci_no> EgressSp cir
<egsp_cir>
```

3 Specify the connection's committed burst size (Bc) in the egress direction.

```
set FrUni/<FrUni_inst> Dlci/<dlci_no> EgressSp bc
<egsp_bc>
```

```
set FrNni/<FrNni_inst> Dlci/<dlci_no> EgressSp bc
<egsp_bc>
```

4 Specify the connection's excess burst size in the egress direction.

```
set FrUni/<FrUni_inst> Dlci/<dlci_no> EgressSp be
<egsp_be>
```

```
set FrNni/<FrNni_inst> Dlci/<dlci_no> EgressSp be
<egsp_be>
```

5 Specify the time interval for rate and burst measurements in the egress direction.

```
set FrUni/<FrUni_inst> dlci/<dlci_no> EgressSp t
<interval>
```

```
set FrNni/<FrNni_inst> dlci/<dlci_no> EgressSp t
<interval>
```

Variable definitions

Variable	Value
<n>	is the instance number of the FR UNI.
<dlci_no>	is the instance number of the DLCI.
<no>	indicates that service parameters for the DLCI are different for egress direction.
<egsp_cir>	is the committed information rate for the DLCI in bits/s.
<egsp_bc>	is the committed burst size for the DLCI in bits.
<egsp_be>	is the excess burst size for the DLCI in bits.
<interval>	is a time interval in milliseconds.

Procedure job aid

Figure 37
Egress service parameters FR UNI component hierarchy
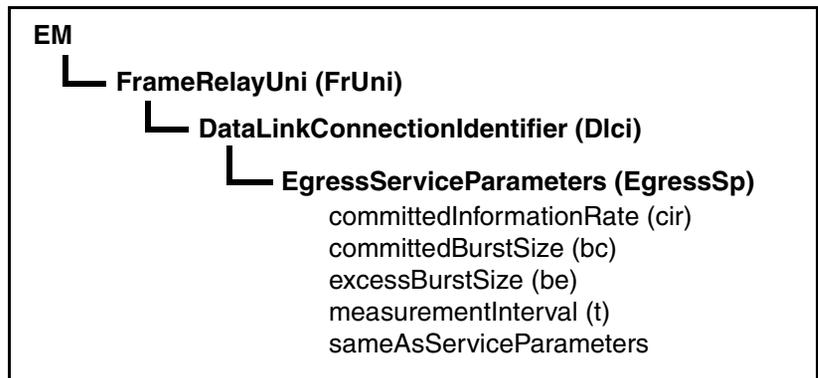
Figure 38
Egress service parameters FR NNI component hierarchy

```
EM
    └── FrameRelayNni (FrNni)
            └── DataLinkConnectionIdentifier (Dlci)
                    └── EgressServiceParameters (EgressSp)
                        committedInformationRate (cir)
                        committedBurstSize (bc)
                        excessBurstSize (be)
                        measurementInterval (t)
                        sameAsServiceParameters
```
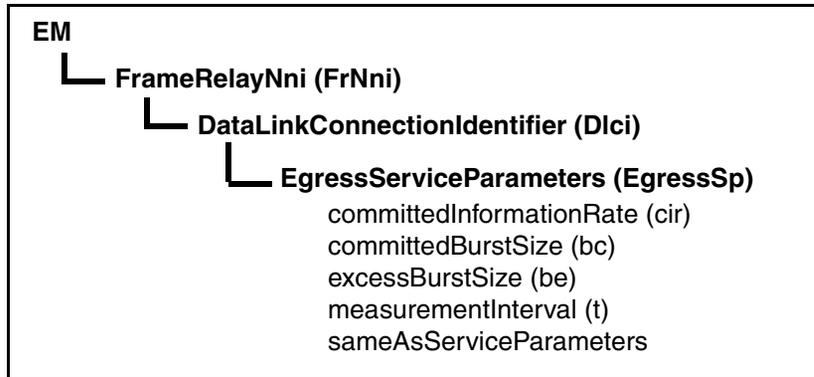
Configuring a backup FR UNI connection

Configure a backup FR UNI connection to provide redundancy for the permanent FR UNI connection.

Procedure steps

1   Add a SpareDc component to the DLCI you want to spare.

add FrUni/<FrUni_inst> Dlci/<dlci_no> SpareDc

2   Set the numbering plan indicator of the remote end of the connection.

set FrUni/<FrUni_inst> dlci/<dlci_no> SpareDc srnpi<spare_rem_npi)

3   Set the data network address of the remote end of the connection.

set FrUni/<FrUni_inst> dlci/<dlci_no> SpareDc srdna <spare_rem_dna>

Variable definitions

Variable	Value
<dlci_no>	is the instance of the DLCI that you are creating a spare for.
<FrUni_inst>	is the instance number of the FR UNI.
<spare_rem_npi>	is the numbering plan indicator for the destination spare address.
<spare_rem_dna>	is the spare address of the destination for the connection.

Procedure job aid

Figure 39
Backup FR UNI connection component hierarchy

```
EM
    └── FrameRelayUni (FrUni)
            └── DataLinkConnectionIdentifier (Dlci)
                    └── SpareDirectCall (SpareDc)
                            spareRemoteNpi (srnpi)
                            spareRemoteDna (srdna)
```
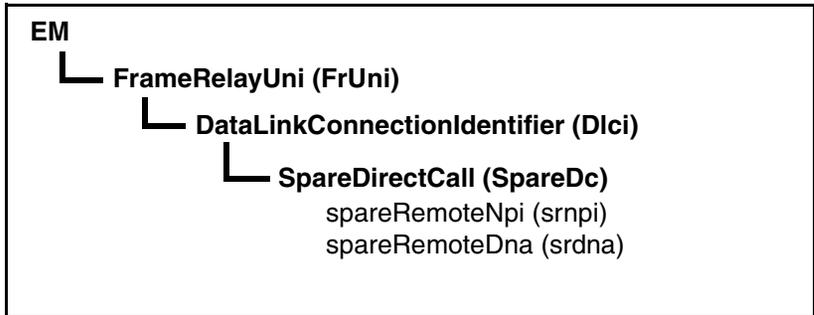
Verifying virtual circuit connectivity

Verify virtual circuit connectivity to confirm that the permanent connection has been provisioned correctly and it is operating without errors.

Procedure steps

1   Display the A-bit status to check that the *aBitStatusToIf* and *aBitStatusFromIf* attributes are active. If either attribute is inactive on an interface, the connection cannot transmit data.

```
display FrUni/<FrUni_inst> Dlci/<dlci_no> aBitStatus

display FrNni/<FrNni_inst> Dlci/<dlci_no> aBitStatus
```

2   Check the VC state for the connection. If the state attribute is not dataTransferP4, the connection is down.

```
display FrUni/<FrUni_inst> Dlci/<dlci_no> Vc state

display FrNni/<FrNni_inst> Dlci/<dlci_no> Vc state
```

3   Verify that the *remoteDna* and *remoteDlci* attributes have been properly configured with the DNA and DLCI of the interface that terminates the end-to-end connection in the other network.

```
display FrUni/<FrUni_inst> Dlci/<dlci_no> Dc

display FrNni/<FrNni_inst> Dlci/<dlci_no> Dc
```

4   If you have configured a backup FR UNI connection, verify that the *remoteDna* and *remoteDlci* attributes have been properly configured with the DNA and DLCI of the interface that terminates the end-to-end connection in the other network.

```
display FrUni/<FrUni_inst> Dlci/<dlci_no> spareDc
```

5   Ensure the *Dc type* attribute configuration has the correct corresponding configuration at the remote end.

```
display FrUni/<FrUni_rem> Dlci/<dlci_no> Dc type

display FrNni/<FrNni_rem> Dlci/<dlci_no> Dc type
```

6   For a Nortel Networks Multiservice Switch node-only network, ensure that the prefix-DNAs for each endpoint have been configured under the *CallRouter* (*Cr*) component, along with their corresponding routing ID (RID) and module ID (MID). If routing has been configured correctly, determine if any of the DNAs are already in use.

```
display -p Cr npi/<x121_e164> dna/*
```

7 For a mixed Nortel Networks Multiservice Switch 7400-DPN network, check the configuration of the Call Server Resource Module (CSRM).

8 If the *abitStatusFromIf* attribute is inactive, an error on the local link is indicated. Check the state of the local port by displaying the *interfaceName* attribute of the *Framer* component. The *adminState* attribute should be unlocked, the *operationalState* attribute should be enabled, and the *usageState* attribute is should be either active or busy.

```
display -p FrUni/<FrUni_inst> Framer interfaceName
```

```
display -p FrNni/<FrNni_inst> Framer interfaceName
```

```
display Lp/<lp_no> <port_type>/<pt>
```

9 Verify the LMI protocol configuration for the connection.

```
display FrUni/<FrUni_inst> Lmi
```

```
display FrNni/<FrNni_inst> Lmi
```

10 If a local LMI error is indicated, check the LMI protocol configuration for the local interface and ensure that the procedures attribute at the local end is configured with the same LMI procedures as the remote end.

```
display -p FrUni/<FrUni_inst> Lmi
```

```
display -p FrNni/<FrNni_inst> Lmi
```

11 If a remote LMI error is indicated, check the LMI protocol configuration for the remote interface and ensure that the procedures attribute at the remote end is configured with the same LMI procedures as the local end.

```
display -p FrUni/<FrUni_rem> Lmi
```

```
display -p FrNni/<FrNni_rem> Lmi
```

On a 4-port DS3 channelized frame relay FP, if the timeslot of a channel associated with a framer interface is provisioned with the value of none, the type of LMI procedure on the FrUni interface also needs to be provisioned as none.

Variable definitions

Variable	Values
<dlci_no>	is the local instance number of the DLCI.
<lp_no>	is the instance number of the associated logical processor.
(Sheet 1 of 2)	

Variable	Values
<FrNni_inst>	is the instance number of the FR NNI terminating the connection.
<FrNni_rem>	is the instance number of the remote FR NNI terminating the connection.
<FrUni_inst>	is the instance number of the FR UNI terminating the connection.
<FrUni_rem>	is the instance number of the remote FR UNI terminating the connection.
<port_type>	is the physical interface on the port.
<pt>	is the instance number of the physical interface.
<x121_e164>	is the numbering plan indicator of the prefix-DNA.
(Sheet 2 of 2)	

Procedure job aid

Figure 40
Virtual circuit connectivity FR UNI component hierarchy

```
EM
    └─ FrameRelayNni (FrUni)
            └─ DataLinkConnectionIdentifier (Dlci)
                    aBitStatusToIf
                    aBitReasonToIf
                    aBitStatusFromIf
                    aBitReasonFromIf
                    ├─ Vc
                    │       state
                    └─ DirectCall (Dc)
                            remoteDna (rdna)
                            remoteDlci (rdlci)
                            type
```
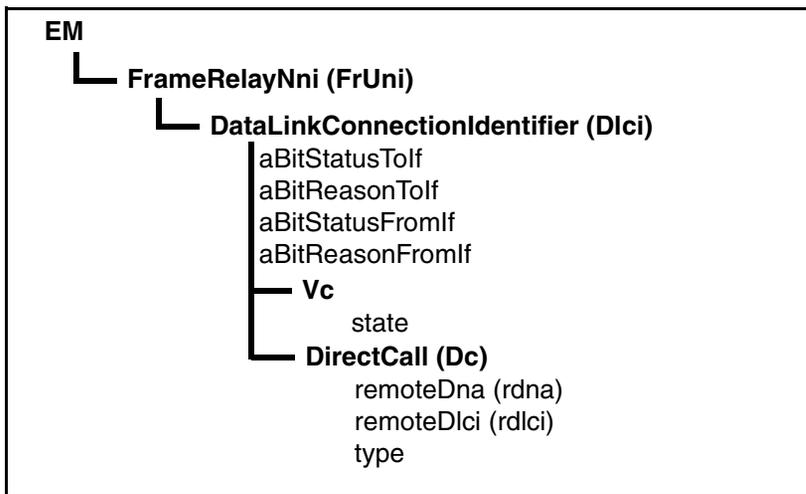
Figure 41
Virtual circuit connectivity FR NNI component hierarchy

```
EM
    └── FrameRelayNni (FrNni)
            └── DataLinkConnectionIdentifier (Dlci)
                    aBitStatusToIf
                    aBitReasonToIf
                    aBitStatusFromIf
                    aBitReasonFromIf
                    ── Vc
                    │       state
                    └── DirectCall (Dc)
                            remoteDna (rdna)
                            remoteDlci (rdlci)
                            type
```
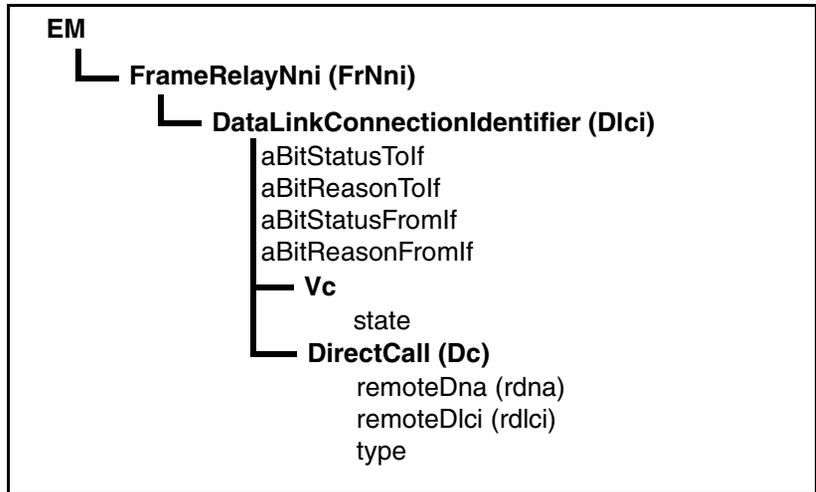
Configuring egress accounting

Configure egress accounting to set module-wide parameters for egress accounting on permanent connections.

Prerequisites

- Configure basic Nortel Networks Multiservice Switch accounting by following the procedures in NN10600-560 *Nortel Networks Multiservice Switch 7400/15000/20000 Accounting*.

Procedure steps

1   Specify the segment size for national call accounting. This attribute is a module-wide parameter.

    ```
    set Mod Vcs segmentSize <size>
    ```

2   Specify the type of unit to be counted. This attribute is a module-wide parameter.

    ```
    set Mod Vcs unitsCounted <unit>
    ```

3   Specify the mode of accounting record generation.This attribute is a module-wide parameter.

    ```
    set Mod Vcs genMod <mode>
    ```

4   Turn egress accounting on for the FR UNI or FR NNI port.

    ```
    set FrUni/<FrUni_inst> Dna ega <accnt_on>
    ```

    ```
    set FrNni/<FrNni_inst> Dna ega <accnt_on>
    ```

5   Specify the purpose for record collection. If you do not specify a reason, accounting record generation does not occur.

    ```
    set FrUni/<FrUni_inst> Dna aco <category>
    ```

    ```
    set FrNni/<FrNni_inst> Dna aco <category>
    ```

6   Optionally, enable or disable the accounting feature for a specific DLCI.

    ```
    set FrUni/<FrUni_inst> Dlci Sp ac <dlciacct_on>
    ```

    ```
    set FrNni/<FrNni_inst> Dlci Sp ac <dlciacct_on>
    ```

7   Specify the value of the correlation tag you want on a specific DLCI.

    ```
    set FrUni/<FrUni_inst> dlci/<dlci_no> Sp
    correlationTag <value_string>
    ```

```
set FrNni/<FrNni_inst> dlci/<dlci_no> Sp
correlationTag <value_string>
```

8   You can also delete a previously provisioned correlation tag value.

```
set FrUni/<FrUni_inst> dlci/<dlci_no> Sp
correlationTag ""
```

```
set FrNni/<FrNni_inst> dlci/<dlci_no> Sp
correlationTag ""
```

9   If required for correlation purposes, manually provision a unique
correlation tag.

Variable definitions

Variable	Value
""	is a pair of double quotation marks used to indicate an empty string. The default value for a correlation tag identifier is an empty string. If you do not want to use the circuit id functionality, do not change the default value.
	If you do not provision a value for the correlation tag on a specific DLCI then no correlation tag attribute displays under that DLCI component in the operational view.
<accnt_on>	is a setting that enables egress accounting.
<category>	is a category assigned for the purpose for record collection.
<dlciacct_on>	is a setting that enables accounting for the DLCI.
<FrNni_inst>	is the instance number of the FR NNI.
<FrUni_inst>	is the instance number of the FR UNI.
<mode>	is the accounting mode, either double- or single-ended.
<size>	is the size of segment to be counted.
<value_string>	is an alphanumeric string of up to 32 ASCII digits that identifies the connection.
<unit>	is the type of unit to be counted, either frames or segments.

Procedure job aid

Figure 42
Egress accounting FR UNI component hierarchy

```
EM

    ModuleData (Mod)

        VirtualCircuitSystem (Vcs)
            segmentSize
            unitsCounted
            generationMode (genMod)


    FrameRelayUni (FrUni)

        DataNetworkAddress (Dna)
            accountCollection (aco)
            egressAccounting (ega)

        DataLinkConnectionIdentifier (Dlci)

            ServiceParameters (Sp)
                ....
                accounting (ac)
                correlationTag (ctag)
```
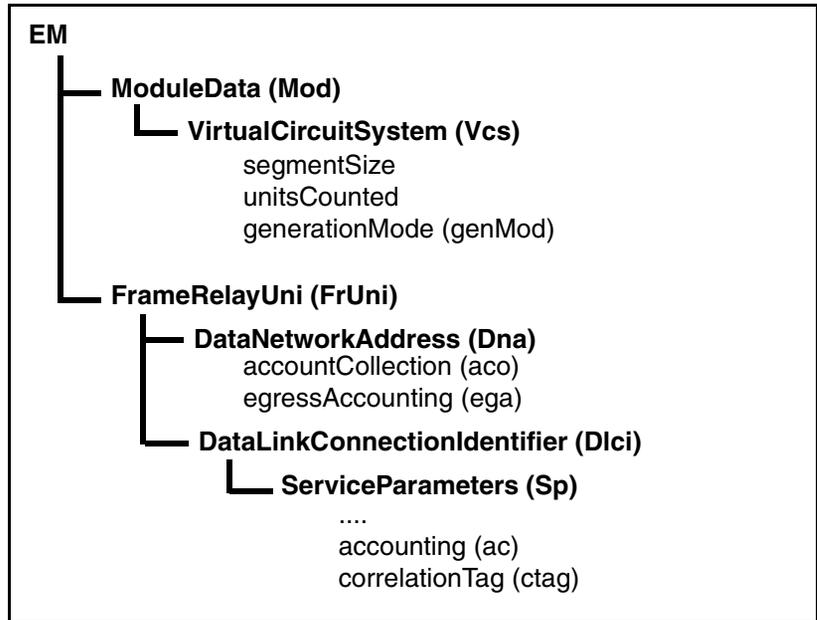
Figure 43
Egress accounting FR NNI component hierarchy

```
EM
  └── ModuleData (Mod)
         └── VirtualCircuitSystem (Vcs)
                  segmentSize
                  unitsCounted
                  generationMode (genMod)

  └── FrameRelayNni (FrNni)
         └── DataNetworkAddress (Dna)
                  accountCollection (aco)
                  egressAccounting (ega)
         └── DataLinkConnectionIdentifier (Dlci)
                  └── ServiceParameters (Sp)
                           ....
                           accounting (ac)
                           correlationTag (ctag)
```
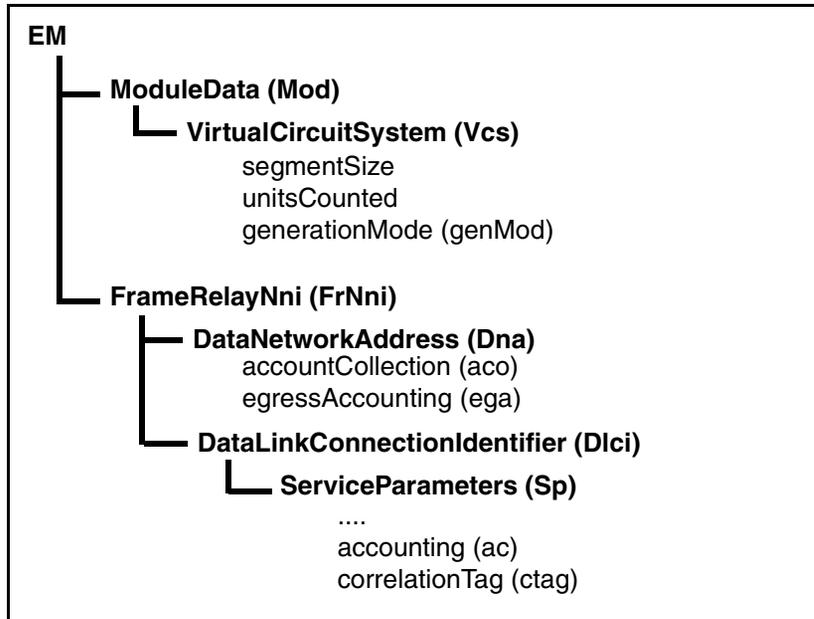
Chapter 6
BNX-Iwf configuration

Configure BNX-Iwf to interwork between a network element running hyperstream software (BNX) and a Nortel Networks Multiservice Switch 15000 or Multiservice Switch 20000 node, in which interworking supports frame relay over IP.

Prerequisites to BNX-Iwf configuration

- Use the procedures in NN10600-270 *Nortel Networks Multiservice Switch 7400/15000/20000 Software Installation* to install Frame Relay software. The applications required for the Frame Relay over IP-BNX Interworking function are named frameRelay, ip, and wanDte.

BNX-Iwf configuration procedures

This task flow shows you the sequence of procedures you perform to configure BNX-Iwf. To link to any procedure, go to "BNX-lwf configuration procedure navigation" (page 110).

Figure 44
BNX-Iwf configuration procedures

MSS 3538 009 AA
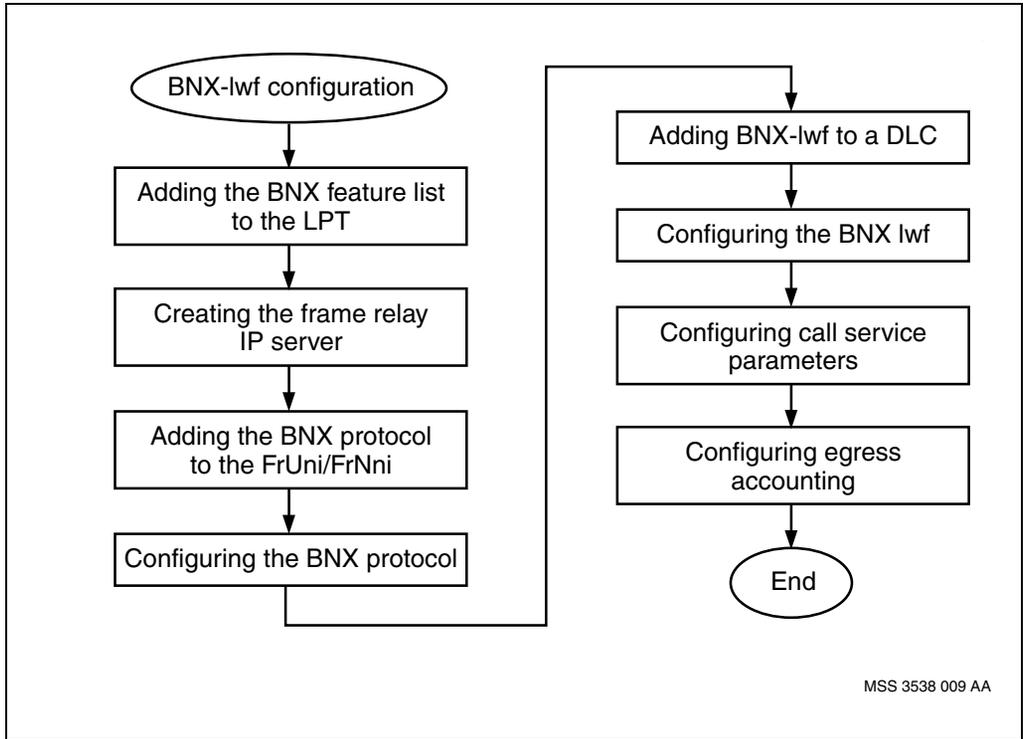
BNX-Iwf configuration procedure navigation

Adding the BNX feature list to the LPT

Add the BNX feature list to the logical processor type (LPT) before you configure the FrUni or Nni service. You can add the features to an existing FrUni or Nni LPT, or even create a new LPT.

Prerequisites

- Configure a logical processor (LP) for the Frame Relay over IP-BNX service. See NN10600-550 *Nortel Networks Multiservice Switch 7400/15000/20000 Common Configuration Procedures*.

Procedure steps

1   Define an instance of a LPT. You can use any mnemonic for the name of the LPT instance.

   `add Software lpt/<n>`

2   Determine which features you want to enable for the interface.

   `display Software av/frameRelay_<version> feature/*`

   `display Software av/ip_<version> feature/*`

   `display Software av/wanDte_<version> feature/*`

3   Add the FrUni or FrNni service to the LPT's feature list, if you want support for the Frame Relay over IP-BNX interworking feature.

   `set Software lpt/<n> featurelist <feature name>`

4   Link the LPT to a logical processor.

   `set lp/<m> logicalProcessorType Software Lpt/<n>`

5   Check and activate your configuration changes.

Variable definitions

Variable	Definition
<feature name>	is the name of the software feature for the Frame Relay over IP-BNX service. The features required for FrUni over IP-BNX service are frUniPvcBnx, IP, and AtmMpe. The features required for FrNni BNX service are frNniPvcBnx, IP, and AtmMpe.
<m>	is the instance of the logical processor defined for the frame relay BNX service.
<n>	is the name of the LPT instance.
<version>	is the version number of the application software.

Creating the frame relay IP server

Create the frame relay IP server to establish the link between the frame relay application and IP.

Prerequisites

- Ensure that a protocol port is available on a virtual router in your IP network. For information about configuring virtual routers see NN10600-801 *Nortel Networks Multiservice Switch 7400/15000/20000 IP Configuration Management*.

Procedure steps

1   Create the FRIP.

```
add FrIpServer/<f>
```

2   Establish the link between the protocol port and the FRIP.

```
set FrIpServer/<f> linkToProtocolPort VirtualRouter/
<v_str> ProtocolPort/<p_str>
```

3   Establish the link between the FRIP and the logical processor that supports the FRIP.

```
set FrIpServer/<f> logicalProcessor LogicalProcessor/
<lp_no>
```

Variable definitions

Variable	Definition
<f>	is the instance number of the frame relay IP server and is a decimal between 0 and 1024.
<lp_no>	is the instance number of the logical processor and is a decimal between 0 and 15.
<p_str>	is the name of the protocol port. an ASCII string of 1 to 20 characters.
<v_str>	is the name of the virtual router. an ASCII string of 1 to 8 characters.

Procedure job aid

Figure 45
Frame relay IP server component hierarchy

Em

　　├─── **FrIpServer**
　　　　　linkToProtocol Port
　　　　　logicalProcessor

PPT 3085 015 AA
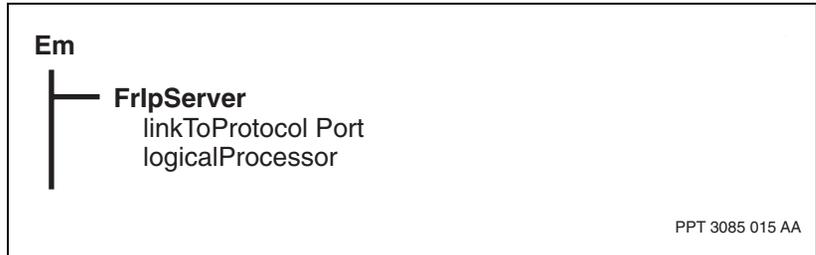
Adding the BNX protocol to the FrUni/FrNni

Add the BNX protocol to the FrUni/FrNni in order to add the BNX protocol port to the FrUni/FrNni and link it to the FRIP.

Procedure steps

1   Add the BNX protocol to the FrUni or FrNni.

```
add <fruni_frnni>/<fr_uni> BnxProtocol
```

2   Link the BNX protocol to the FRIP.

```
set <fruni_frnni>/<fr_uni> BnxProtocol linkToIpServer
FrIpServer/<f>
```

3   Display the attributes of the *FrUni BnxProtocol* component.

```
display <fruni_frnni>/<fr_uni> BnxProtocol
```

Variable definitions

Variable	Definition
<f>	is the instance number of the frame relay IP server to which the protocol is linked and is a decimal between 0 and 1024.
<fr_uni>	is the instance number of the FrUni and is a decimal between 1 and 4 294 967 295.
<fruni_frnni>	is either FrUni, for configuration of the FrUni, or FrNni, for configuration of the FrNni .

Procedure job aid

Figure 46
BNX protocol component hierarchy

```
Em
    ──── FrameRelayUni/FrameRelayNni
              └── BnxProtocol
                    linkToIpServer
                    ipAddress
                    crossNetEnabled
                    crossNetPollingInterval
                    crossNetErrorThreshold
                    crossNetAsyncUpdate
                    crossNetListenEnabled
                    trafficPriorityEnabled
                    discardLevelEnabled
                    trafficDiscardLevel
                    defaultTrafficPriority

                                            PPT 3085 012 AA
```
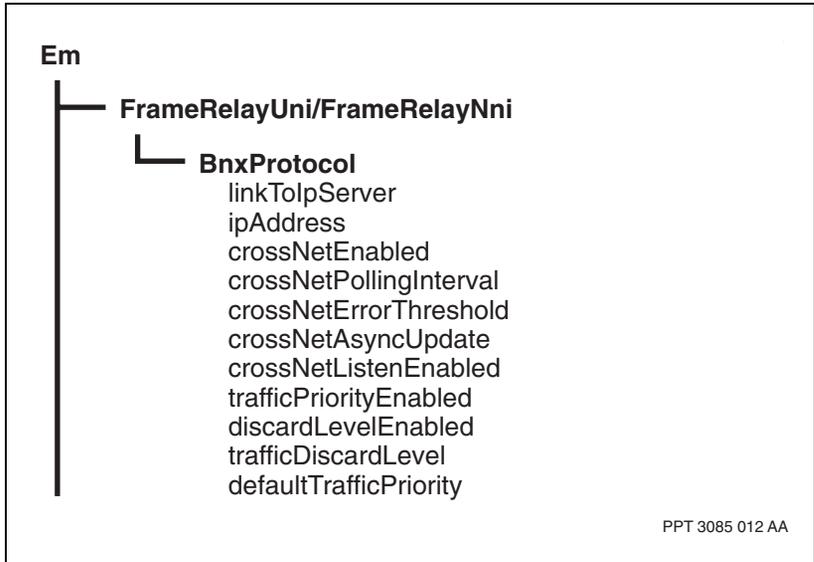
Configuring the BNX protocol

Configure the BNX protocol in order to support the BNX protocol port used to link to the FRIP.

Procedure steps

1   Associate an IP address with the frame relay port.

```
set <fruni_frnni>/<fr_uni> BnxProtocol ipAddress
<ip_addr>
```

2   Enable or disable BNX cross-net signaling.

```
set <fruni_frnni>/<fr_uni> BnxProtocol crossNetEnabled
<crossNet_yes_no>
```

3   Set the time interval between cross-net signals.

```
set <fruni_frnni>/<fr_uni> BnxProtocol
crossNetPollingInterval <xnet_poll>
```

4   Define the number of cross-net polling intervals that must elapse without a signal from the remote end before the near end declares the PVC out of service.

```
set <fruni_frnni>/<fr_uni> BnxProtocol
crossNetErrorThreshold <xnet_err>
```

5   Enable or disable the cross-net update for data link connection identifier (DLCI) state changes.

```
set <fruni_frnni>/<fr_uni> BnxProtocol
crossNetAsyncUpdate <xnet_asyn>
```

6   Enable or disable the capacity of the interface to listen for periodic cross-net signaling from the remote end as a way of declaring the PVC either in or out of service.

```
set <fruni_frnni>/<fr_uni> BnxProtocol
crossNetListenEnabled <listen_yes_no>
```

7   Determine how traffic through this interface is tagged for priority.

```
set <fruni_frnni>/<fr_uni> BnxProtocol
trafficPriorityEnabled <traffic_yes_no>
```

8   Determine how traffic through this interface is tagged for discard level.

```
set <fruni_frnni>/<fr_uni> BnxProtocol
discardLevelEnabled <discard_yes_no>
```

9   Set the traffic discard level.

```
set <fruni_frnni>/<fr_uni> BnxProtocol
trafficDiscardLevel <level>
```

10   Set the default traffic priority which is used on all BNX-Iwf DLCIs.

```
set <fruni_frnni>/<fr_uni> BnxProtocol
defaultTrafficPriority <priority>
```

Variable definitions

Variable	Definition
<crossNet_yes_no>	is either yes or no. The default value is yes. If you enable crossNetSignaling, the BNX cross-net signal is sent for each time interval (which is set through *crossNetPollingInterval* attribute in step 3).
<discard_yes_no>	is either yes or no. The default value is no. When set to no, the system tags incoming DE=1 frames with discard level 2 and DE=0 frames with discard level 4. When set to yes, the system tags all incoming frames with the discard level set to trafficDiscardLevel.
<fr_uni>	is the instance number of the FrUni and is a decimal between 1 and 4 294 967 295.
<fruni_frnni>	is either FrUni, for configuration of the FrUni, or FrNni, for configuration of the FrNni.
<ip_addr>	is a valid IP address. This address must be in the same subnet as the associated protocol port.
<level>	is a decimal between 1 and 4. The default value is 4. Through this attribute, the system tags ingress frames with the configured discard level. The setting of this attribute is meaningful only if the *discardLevelEnabled* attribute is set to yes.
<listen_yes_no>	is either yes or no. The default value is yes. If you enable cross-net listening, the DLCI expects a cross-net signal for each time interval (which is set through the *crossNetPollingInterval* attribute in step 3).
(Sheet 1 of 2)	

Variable	Definition
<priority>	is a decimal between 1 and 3. The default value is 2. This setting can be overwritten at the DLCI level by setting the trafficPriority attribute under *BnxInterworkingFunction* component.
<traffic_yes_no>	is either yes or no. The default value is no. When set to no, the switch sets the trafficPriority field set to 2. When set to yes, the switch tags all ingress traffic according to the provisioned value of the *trafficPriority* attribute. Traffic priority can be set at the FrUni level or overridden at the DLCI level.
<xnet_asyn>	is either enabled or disabled. The default value is enabled. If enabled, the system sends a cross-net signal as soon as the DLCI changes state. Otherwise, the cross-net signal is sent only at the next cross-net polling interval.
<xnet_err>	is a number of intervals, and is a decimal in the range of 1 to 30. The default value is 3.
<xnet_poll>	is the number of seconds between transmission of cross-net signaling datagrams, and is a decimal in the range of 5 to 86 400. The default value is 120.
(Sheet 2 of 2)	

Adding BNX-Iwf to a DLCI

Add BNX-Iwf to a DLCI to provide interworking of frame relay applications between a Nortel Networks Multiservice Switch node and a BNX.

Prerequisites

- BNX-Iwf cannot co-exist with the direct call function, which is represented by the DirectCall component under the FrUni Dlci and FrNni Dlci components.

Procedure steps

1 Add the DLCI to the FrUni or FrNni.

```
add <fruni_frnni>/<fr_uni> Dlci/<dlci>
```

DPRS and BNX DLCIs share the PVC DLCI range. This range is configured through the *FrUni Signaling highestPvcDlci* attribute, though which the range of DLCIs reserved for PVCs is defined as 16 to the value of the *highestPvcDlci* attribute.

2 Add BNX-Iwf to the DLCI.

```
add <fruni_frnni>/<fr_uni> Dlci/<dlci>
BnxInterworkingFunction
```

3 Specify whether a DLCI operates as an FRF.1 DTE or as an FRF.12 DTE:

```
set <fruni_frnni>/<FrUni_inst> dlci/<dlci_no> Sp
frf12EndToEnd <frf12endtoend_on>
```

4 Specify the maximum size of data fragments to be transmitted if *frf12EndToEnd* has been enabled.

```
set <fruni_frnni>/<FrUni_inst> dlci/<dlci_no> Sp
frf12FragmentSize <frf12fragmentsize>
```

5 Optionally, display the attributes of the *FrUni Dlci BnxInterworkingFunction* component or *FrNni Dlci BnxInterworkingFunction* component.

```
display <fruni_frnni>/<fr_uni> Dlci/<dlci>
BnxInterworkingFunction
```

Variable definitions

Variable	Definition
<dlci>	is the instance number of the DLCI (a decimal between 16 and 1007).
	All the DLCIs at this interface, as long as accounting is enabled for the DLCI itself, generate the local accounting record.
<frf12endtoend_on>	is a setting that specifies whether or not DLCIs will fragment data frames.
<frf12fragmentsize>	is the maximum size of the data fragments to be transmitted by Multiservice Switch frame relay. Available values in bytes are 40, 48, 60, 80, 120, 240, and 480.
<fr_uni>	is the instance number of the FrUni (a decimal between 1 and 4 294 967 295).
<fruni_frnni>	is either FrUni, for configuration of the FrUni, or FrNni, for configuration of the FrNni.
<reason>	is either bill, study, test, audit, or force. The default value is bill.

Procedure job aid

Figure 47
BNX-Iwf to a DLCI component hierarchy

```
Em
    └── FrameRelayUni/FrameRelayNni
            └── DataLinkConnectionIdentifier
                    ├── BnxInterworkingFunction
                    │       calledIpAddress
                    │       calledDlci
                    │       trafficPriority
                    │       backupCalledIpAddress
                    │       backupCalledDlci
                    │       redirectType
                    │       minActiveIntervals
                    │       defaultRedirectAction
                    └── ServiceParameters (Sp)
                            frf12EndtoEnd
                            frt12FragmentSize

                                              PPT 3085 013 AA
```
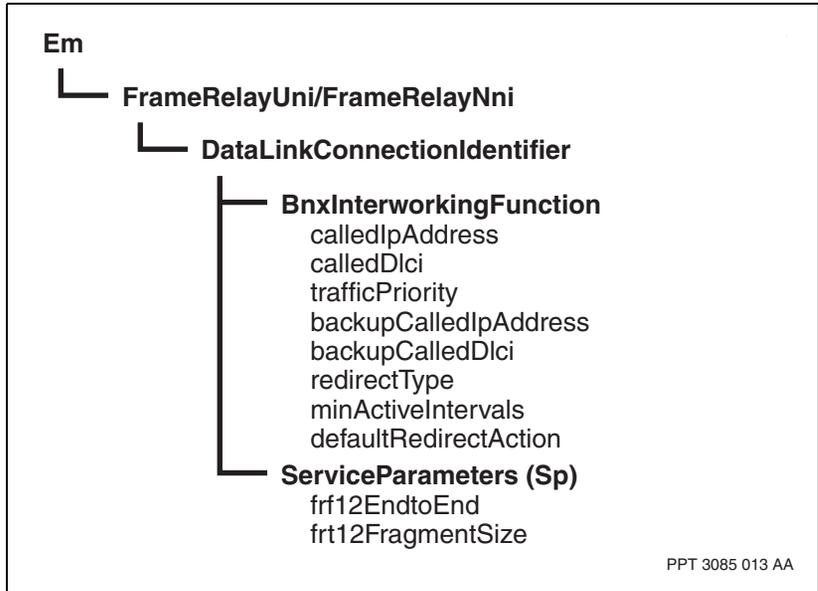
Configuring the BNX-Iwf

Configure the BNX-Iwf to set the attributes of the *BnxInterworkingFunction* component.

Prerequisites

- BNX-Iwf cannot co-exist with the direct call function, which is represented by the DirectCall component under the FrUni Dlci and FrNni Dlci components.

Procedure steps

1   Set the IP address of the destination.

```
set <fruni_frnni>/<fr_uni> Dlci/<dlci>
BnxInterworkingFunction calledIpAddress <ip_addr>
```

2   Set the DLCI of the destination.

```
set <fruni_frnni>/<fr_uni> Dlci/<dlci>
BnxInterworkingFunction calledDlci <dest_dlci>
```

3   Set the traffic priority for BNX-Iwf on this connection.

```
set <fruni_frnni>/<fr_uni> Dlci/<dlci>
BnxInterworkingFunction trafficPriority <priority>
```

4   Define the backup called IP address (the backup destination). The IP address of the backup must not be the same as that of the primary.

```
set <fruni_frnni>/<fr_uni> Dlci/<dlci>
BnxInterworkingFunction backupCalledIpAddress
<ip_addr>
```

5   Define the backup called DLCI (the backup destination).

```
set <fruni_frnni>/<fr_uni> Dlci/<dlci>
BnxInterworkingFunction backupCalledDlci <dest_dlci>
```

The value of the *backupCalledDlci* attribute must not be undefined if the value of the *backupCalledIpAddress* attribute is not 0.0.0.0. By extension, the value of the *backupCalledDlci* attribute must be undefined when the *backupCalledIpAddress* attribute is 0.0.0.0.

6   Define the type of DTE notify method for PVC redirection.

```
set <fruni_frnni>/<fr_uni> Dlci/<dlci>
BnxInterworkingFunction redirectType <redir_type>
```

7 Define the minimum number of consecutive time intervals for which the primary destination is active before the PVC is redirected back to primary in switch-on-demand mode.

```
set <fruni_frnni>/<fr_uni> Dlci/<dlci>
BnxInterworkingFunction minActiveIntervals <int_no>
```

A value of 0 for the *minActiveIntervals* attribute configures the redirect to occur immediately on receiving an active cross-net. The length of the time interval is defined by the *crossNetPollingInterval* attribute under the *BnxProtocol* component.

8 Define the default redirection action for the PVC.

```
set <fruni_frnni>/<fr_uni> Dlci/<dlci>
BnxInterworkingFunction defaultRedirectAction
<dra_opt>
```

A value of 0 for the *minActiveIntervals* attribute configures the redirect to occur immediately on receiving an active cross-net. The length of the time interval is defined by the *crossNetPollingInterval* attribute under the *BnxProtocol* component.

Variable definitions

Variable	Definition
<dest_dlci>	is the instance number of the DLCI assigned to the far end of the connection, and is a decimal between 16 and 1007.
<dlci>	is the instance number of the DLCI and is a decimal between 16 and 1007.
<dra_opt>	is either goPrimary, goOnDemand, or goBackup. The default value is goPrimary.
<fr_uni>	is the instance number of the FrUni and is a decimal between 1 and 4 294 967 295.
<fruni_frnni>	is either FrUni, for configuration of the FrUni, or FrNni, for configuration of the FrNni.
<int_no>	is a decimal from 0 to 4. The default value is 0.
<ip_addr>	valid IP address for the far end of the connection.
(Sheet 1 of 2)	

Variable	Definition
<priority>	is a decimal between 1 and 3 or the value useDefaultTrafficPriority. The default value is useDefaultTrafficPriority. The default traffic priority is derived from the value configured for the *defaultTrafficPriority* attribute of *BnxProcotol* component.
<redir_type>	is either transparent, nIntrusive, or aIntrusive. The default value is aIntrusive.
(Sheet 2 of 2)	

Chapter 7
Module-wide QOS configuration

Configure module-wide QOS to define transfer, emission and routing priority mapping on a module-wide or a per-connection basis.

Module-wide QOS configuration procedures

This task flow shows you the sequence of procedures you perform to configure module-wide QOS. To link to any procedure, go to "Module-wide QOS configuration procedure navigation" (page 128).

Figure 48
Module-wide QOS configuration procedures


```
        ┌─────────────────────────┐
        │   Module-wide QoS       │
        │   configuration         │
        └─────────────────────────┘
                    │
                    ▼
    ┌─────────────────────────────────┐
    │ Configuring flexible QoS mapping │
    └─────────────────────────────────┘
                    │
                    ▼
    ┌─────────────────────────────────┐
    │   Configuring IP CoS mapping     │
    └─────────────────────────────────┘
                    │
                    ▼
    ┌─────────────────────────────────┐
    │      Configuring MMTC            │
    └─────────────────────────────────┘
                    │
                    ▼
              ┌──────────┐
              │   End    │
              └──────────┘
```

MSS 3348 004 AA
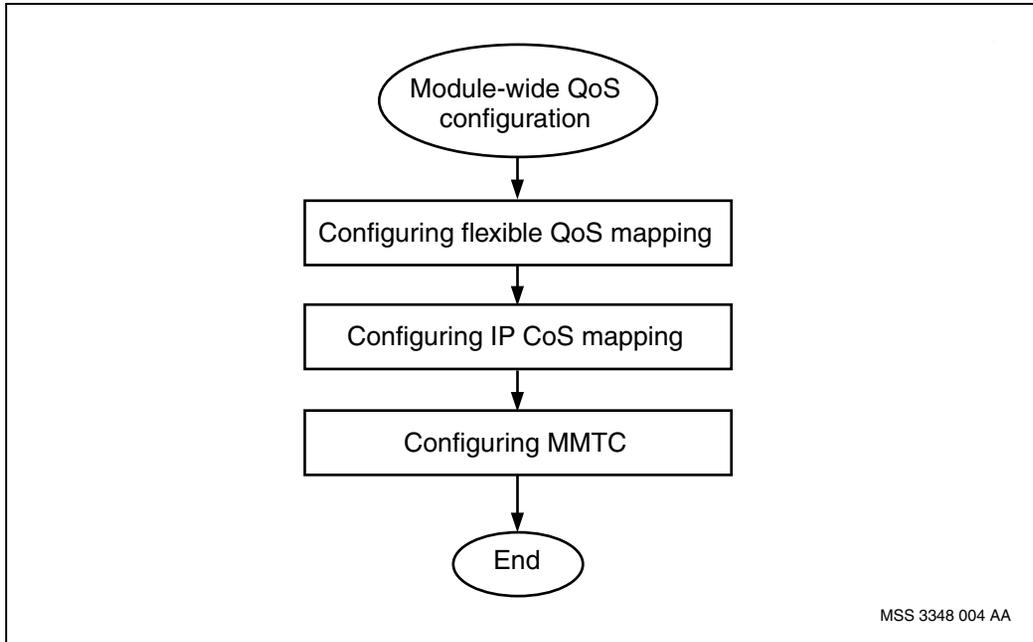
Module-wide QOS configuration procedure navigation

- "Configuring flexible QoS mapping" (page 129)

- "Configuring IP CoS mapping" (page 130)

- "Configuring MMTC" (page 132)

Configuring flexible QoS mapping

Configure flexible QoS mapping to map a transfer priority to a specific emission priority and routing class of service (RCOS) value at the module level.

Procedure steps

1   Specify an emission priority for each transfer priority instance.

    set Mod Frs DprsNet Tpm/<tp> ep <em_pr>

2   Specify a routing class of service for each transfer priority instance.

    set Mod Frs DprsNet Tpm/<tp> rcos <rcos>

Variable definitions

Variable	Value
<em_pr>	is the egress emission priority used for this transfer priority.
<rcos>	is the routing class of service value used for this transfer priority.
<tp>	is the transfer priority instance.

Procedure job aid

Figure 49
Flexible QoS mapping component hierarchy

```
EM
  └─ ModuleData (Mod)
        └─ FrameRelaySystem (Frs)
              └─ DprsNetworking (DprsNet)
                    └─ TransferPriorityMappingDprsNet (Tpm)
                         emissionPriority (ep)
                         routingClassOfService (rcos)
```
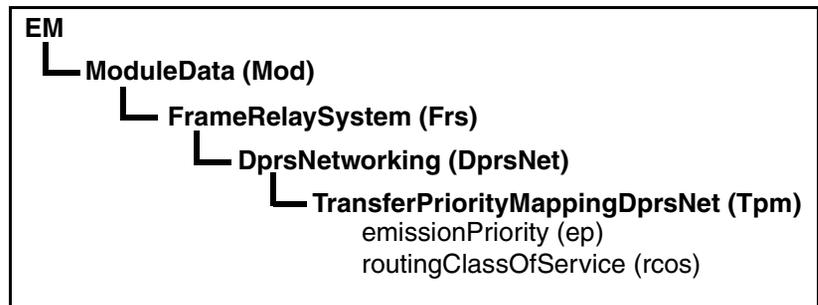
Configuring IP CoS mapping

Configure IP CoS mapping to map an IP class of service (CoS) to frame relay QoS parameters at the module level and to map the IP CoS to frame relay QoS egress policy on individual DLCIs for PVC connections.

Procedure steps

1 Specify a discard priority for each IP type of service (TOS) value.

   ```
   set Mod Frs IpCosToFrQos Tosm/<tos> dp <dp>
   ```

2 Specify an emission priority for each IP type of service (TOS) value.

   ```
   set Mod Frs IpCosToFrQos Tosm/<tos> ep <em_pr>
   ```

3 For FR UNI set the CoS to QoS mapping feature.

   ```
   set FrUni/<FrUni_inst> DLCI/<dlci_inst> Dc mapctof
   <mapCos_on>
   ```

Variable definitions

Variable	Value
<dlci_inst>	is the instance number for the DLCI.
<dp>	is the discard priority for this IP CoS.
<em_pr>	is the egress emission priority used for this IP CoS.
<mapCos_on>	is a setting of yes, no, or protocol dependent that enables overriding the configured flexible QoS mappings for transfer priorities.
<tos>	is the IP type of this service instance.

Procedure job aid

Figure 50
IP CoS mapping component hierarchy

```
EM
  └── ModuleData (Mod)
        └── FrameRelaySystem (Frs)
              └── IpCosToFrQos
                    └── TosMapping (Tosm)
                          discardPriority (dp)
                          emissionPriority (ep)
```
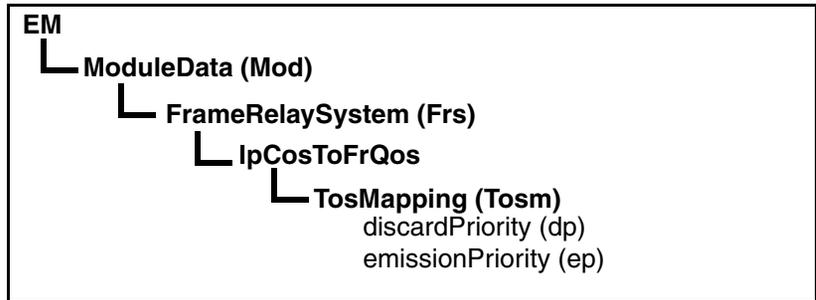
Configuring MMTC

Configure multimedia traffic classes (MMTC) in order to configure subscriber options that let you meet your applications' requirements by selecting traffic handling preferences from the network. You can configure the transfer priority (TP) for each port, each DLCI, or each direction.

Procedure steps

1   Configure the default transfer priority for the calling DCE.

```
set FrUni/<FrUni_inst> Dna dtp <def_tp>
```

```
set FrNni/<FrNni_inst> Dna dtp <def_tp>
```

2   Configure the default transfer priority for the called end DCE.

```
set FrUni/<called_FrUni> Dna dtp <def_tp>
```

```
set FrNni/<called_FrNni> Dna dtp <def_tp>
```

3   Specify a transfer priority for the connection. You can override the value configured for the interface and specify a different transfer priority at the DLCI level.

```
set FrUni/<called_FrUni> Dlci/<dlci_no> Dc tp <tp>
```

```
set FrNni/<called_FrNni> Dlci/<dlci_no> Dc tp <tp>
```

4   Optionally, use the interrupting capability of the Nortel Networks Multiservice Switch frame-cell trunk.

```
set Trunk/<trnk_inst> Unacked Framer framingType
<fmtype_prtcl>
```

Variable definitions

Variable	Value
<called_FrUni>	is the instance number of the called FR UNI.
<called_FrNni>	is the instance number of the called FR NNI.
<def_tp>	is the transfer priority, in the range of 0 to15.
<dlci_no>	is the instance number for the DLCI.
(Sheet 1 of 2)	

Variable	Value
<fmtype_prtcl>	is the protocol used for the framingType. To use the multimedia RCOS and EP3 (TP11 by default), you must set the *framingType* attribute to interrupting for all Multiservice Switch trunks. If multimedia traffic traverses an HDLC trunk, it is downgraded to the delay RCOS (TP10 by default).
<FrNni_inst>	is the instance number of the calling FR NNI.
<FrUni_inst>	is the instance number of the calling FR UNI.
<tp>	is the transfer priority for the DLCI, in the range of 0 to 15.
<trnk_inst>	is the instance number of the Multiservice Switch frame-cell trunk.
(Sheet 2 of 2)	

Procedure job aid

Figure 51
MMTC Fr Uni component hierarchy
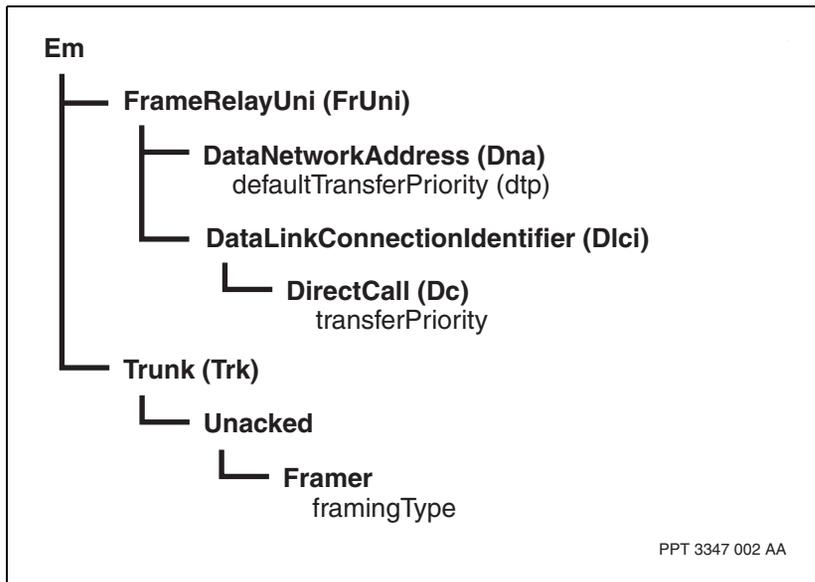
PPT 3347 002 AA

Figure 52
MMTC Fr Nni component hierarchy
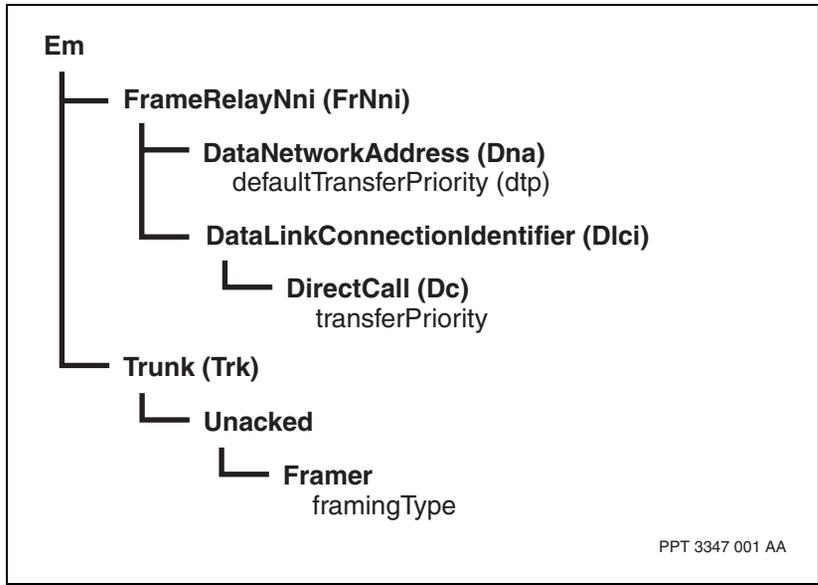
```
Em
 │
 ├── FrameRelayNni (FrNni)
 │        │
 │        ├── DataNetworkAddress (Dna)
 │        │    defaultTransferPriority (dtp)
 │        │
 │        └── DataLinkConnectionIdentifier (Dlci)
 │                 │
 │                 └── DirectCall (Dc)
 │                      transferPriority
 │
 └── Trunk (Trk)
          │
          └── Unacked
                   │
                   └── Framer
                        framingType

                                        PPT 3347 001 AA
```

Chapter 8
Multilink frame relay configuration

Configure multilink frame relay to create MLFR bundles that operate as a logical interface emulating a single physical interface.

Multilink frame relay configuration procedures

This task flow shows you the sequence of procedures you perform to configure multilink frame relay. To link to any procedure, go to "Multilink frame relay configuration procedure navigation" (page 136).

Figure 53
Multilink frame relay configuration procedures
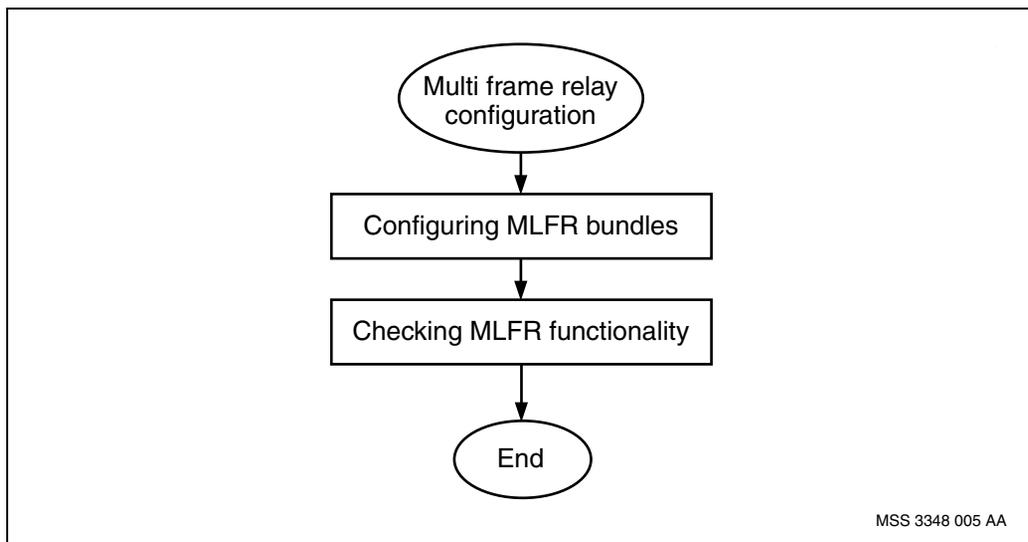
Multilink frame relay configuration procedure navigation

Configuring MLFR bundles

Configure MLFR bundles to add an MLFR service and link it to a frame relay application.

Procedure steps

1   Add the *MultiLinkFrameRelay* component to the logical processor.

```
add LogicalProcessor/<lp_no> MultiLinkFrameRelay/
<mlfr_no>
```

2   Add another link to the *MultiLinkFrameRelay* component.

```
add LogicalProcessor/<lp_no> MultiLinkFrameRelay/
<mlfr_no> Link/<link_no>
```

3   Associate each MLFR link with the *Channel* subcomponent.

```
set LogicalProcessor/<lp_no> MultiLinkFrameRelay/
<mlfr_no> Link/<link_no> interfaceName Lp/<lp_no> Ds3/
<ds3_no> Ds1/<ds1_no> Channel/<chan>
```

4   Add a frame relay application. The Framer is created automatically.

```
add FrUni/<FrUni_inst>
```

```
add FrNni/<FrNni_inst>
```

5   Specify the numbering plan indicator.

```
set FrUni/<FrUni_inst> Dna npi <plan>
```

```
set FrNni/<FrNni_inst> Dna npi <plan>
```

6   Specify the unique data network address (DNA).

```
set FrUni/<FrUni_inst> Dna dna <address>
```

```
set FrNni/<FrNni_inst> Dna dna <address>
```

7   Link the Framer to an MLFR instance.

```
set FrUni/<FrUni_inst> Framer interfaceName
LogicalProcessor/<lp_no> MultiLinkFrameRelay/
<mlfr_no>
```

```
set FrNni/<FrNni_inst> Framer interfaceName
LogicalProcessor/<lp_no> MultiLinkFrameRelay/
<mlfr_no>
```

Variable definitions

Variable	Value
<address>	is a data network address.
<chan>	is the instance number of the *Channel* component.
<ds1_no>	is the instance of the DS1.
<ds3_no>	is the instance of the DS3.
<fp_no>	is the instance number of the function processor.
<FrNni_inst>	is the instance of an FR NNI application.
<FrUni_inst>	is the instance of an FR UNI application.
<link_no>	is a link number.
<lp_no>	is the instance of the logical processor.
<plan>	is a numbering plan indicator (either X.121 or E.164).

Procedure job aid

Figure 54
MLFR bundles FRUNI component hierarchy

PPT 3347 003 AA
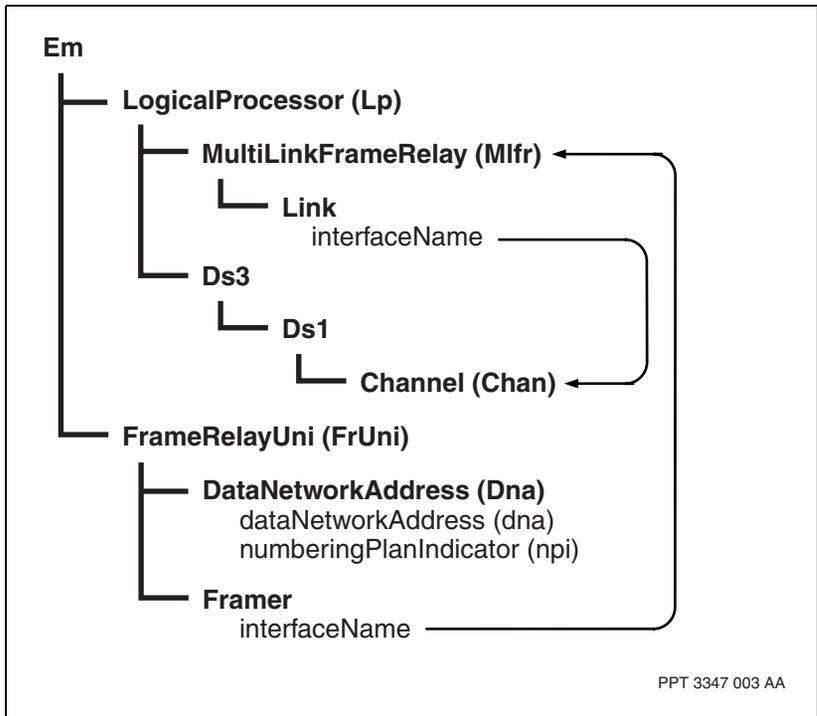
Figure 55
MLFR bundles FRNNI component hierarchy
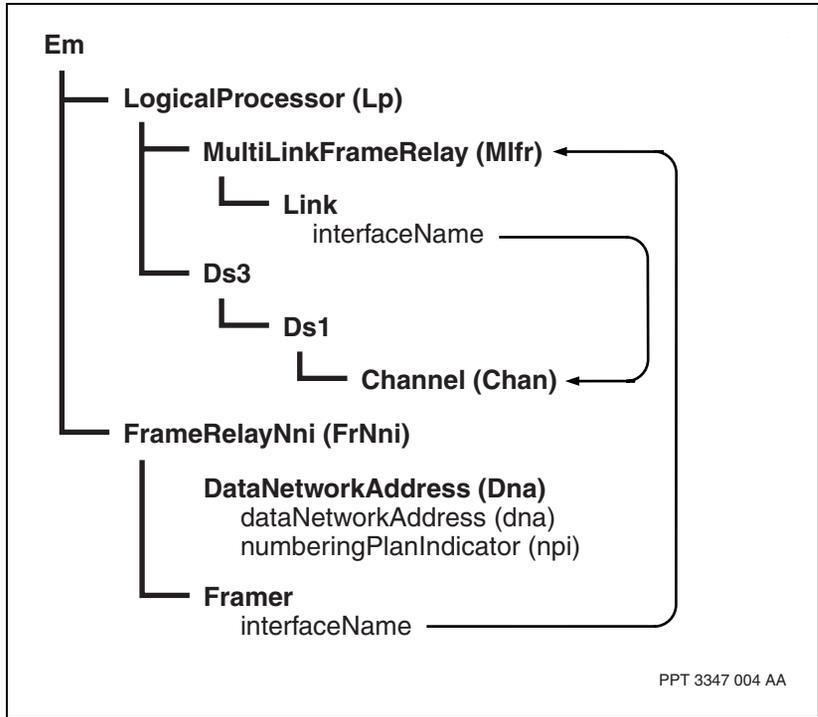
Checking MLFR functionality

Check MLFR functionality to verify that the MLFR bundles have been created properly and are working.

Procedure steps

> **WARNING**
> **Risk of traffic loss**
> This procedure affects traffic on the MLFR bundles associated with the DS3 port. Use this procedure on inactive ports only.

1   Create a loopback at the DS3 port that is associated with the bundle to be tested. Use a physical loopback plug to create the loopback.

2   Display all operational and statistical attributes for the bundle and note the values.

```
display LogicalProcessor/<lp_no> MultiLinkFrameRelay/
<mlfr_no> Link/<link_no>
```

3   Display all operational and statistical attributes for the bundle again.

```
display LogicalProcessor/<lp_no> MultiLinkFrameRelay/
<mlfr_no> Link/<link_no>
```

4   Compare the value of *errorCause* (displays the value loopback Detected) with the previous value and look for changes.

5   Compare the value of *remoteName* with the previous value and look for changes.

6   Compare the value of *runningTime* (shows increasing values) with the previous value and look for changes.

7   Compare the value of *framesLipTx* (shows increasing values) with the previous value and look for changes.

8   Compare the value of *framesLipRx* (shows increasing values) with the previous value and look for changes.

9   Compare the value of *loopback* (shows increasing values) with the previous value and look for changes.

Variable definitions

Variable	Value
<link_no>	is the link number.
<lp_no>	is the instance number of the logical processor.
<mlfr_no>	is the instance number of the multilink frame relay component.

Procedure job aid

Figure 56
MLFR functionality component hierarchy

Em

   └── **LogicalProcessor (Lp)**

          └── **MultiLinkFrameRelay (Mlfr)**
                  Link (Lk)
                  ErrorCause (cause)
                  remoteName (remName)
                  runningTime (running)
                  framesLipTx (lipTx)
                  framesLipRx (lipRx)
                  loopback (loop)

PPT 3347 005 AA
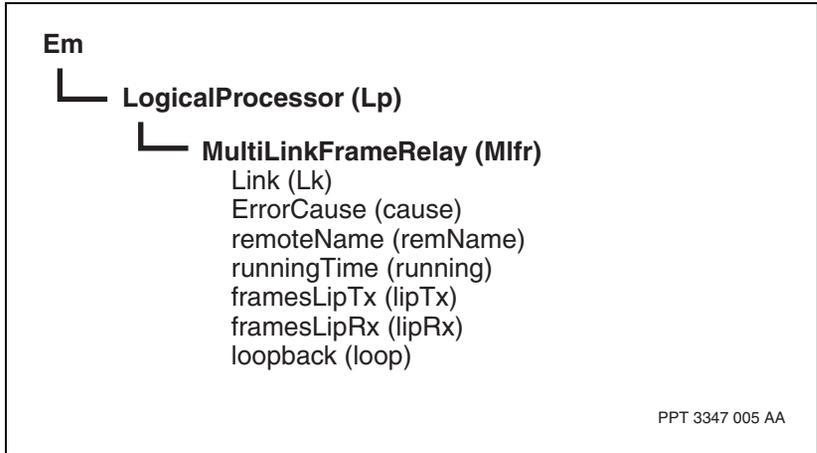
Chapter 9
Frame relay connection troubleshooting

Troubleshoot frame relay connections to discover and isolate failures for frame relay connections and interfaces using the following procedures:

Verifying hardware availability for standby FR UNI services

Verify the hardware availability for standby FR UNI services to make sure the status of the standby card and the *SparedServices* component are in the required state for providing backup service to the FR UNI.

Procedure steps

1 Display the standby status of the spared services.

```
display shelf card/<card_slot> SparedServices
standbyStatus
```

If the *standbyStatus* is not *hotStandby* then an unsupported feature is provisioned in the Lp's Lpt.

2 Display the availability status of the spared services.

```
display shelf card/<card_slot> SparedServices
availabilityStatus
```

If the *availabilityStatus* is degraded then at least one of the ports or channels on the standby FP failed its internal diagnostics due to hardware failure or at least one of the *standbyVc* is not in a *dataTransferP4* state.

Variable definitions

Variable	Value
<card_slot>	is the shelf slot number of the shelf in which the card is physically installed.

Procedure job aid

Figure 57
Standby FR UNI service component hierarchy

```
EM
  └─ Shelf
        └─ card
              └─ SparedServices (SpServ)
                     standbyStatus
                     availabilityStatus
```
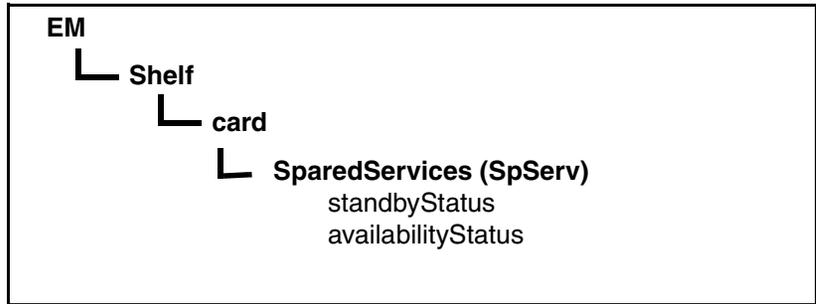
Verifying data flow for subnet connections

Verify data flow for subnet connections to verify that data is being transferred in the subnet.

Procedure steps

1 Verify that the framer on the remote FR UNI is receiving data from the FR NNI.

```
display FrUni/<FrUni_inst> Framer
```

Ensure that the *frmFromIf* attribute is increasing.

On a 4-port DS3 channelized frame relay FP, if the timeslot value of none is provisioned, then the channel associated with the framer component is in a ready, enabled, idle state and does not transfer data. A provisioned timeslot value of none prevents the DLCI component from being provisionable.

2 Verify that the remote FR UNI DLCI is receiving data from the FR NNI and the *frmFromIf* and *bytesFromIf* attributes are increasing.

```
display FrUni/<FrUni_inst> Dlci/<dlci_no> Stats
```

3 Verify that the remote FR UNI is transmitting data into the subnet and the *sendSequenceNumber* attribute is incrementing in relation to data flow and reflects the send sequence number of the last transmitted frame.

```
display FrUni/<FrUni_inst> Dlci/<dlci_no> Vc
```

4 Verify that the framer is receiving data and the *frmFromIf* attribute is increasing.

```
display FrUni/<FrUni_inst> Framer
```

5 Verify that the remote FR UNI DLCI is receiving data and the *frmFromIf* and *bytesFromIf* attributes are increasing.

```
display FrUni/<FrUni_inst> Dlci/<dlci_no> Stats
```

6 For a connection segment across networks, verify that the framer on the adjacent FR NNI is receiving data from the FR NNI and the *frmToIf* attribute is increasing.

```
display FrNni/<FrNni_inst> Framer
```

On a 4-port DS3 channelized frame relay FP, if the timeslot value of none is provisioned, then the channel associated with the framer component is in a ready, enabled, idle state and does not transfer data. A provisioned

timeslot value of none prevents the DLCI component from being provisionable.

7   Verify that the FR NNI DLCI in the adjacent network is receiving data from the FR NNI and the *frmToIf* and *bytesToIf* attributes are increasing.

```
display FrNni/<FrNni_inst> Dlci/<dlci_no> Stats
```

8   Verify that the adjacent FR NNI is transmitting data into its subnet and the *sendSequenceNumber* attribute is incrementing in relation to data flow and reflects the send sequence number of the last transmitted frame.

```
display FrNni/<FrNni_inst> Dlci/<dlci_no> Vc
```

Variable definitions

Variable	Value
<dlci_no>	is the instance number of the DLCI.
<FrUni_inst>	is the instance number of the FR UNI.
<FrNni_inst>	is the instance number of the FR NNI in the adjacent network.

Procedure job aid

Figure 58
Data flow for subnet connections component hierarchy

```
EM
   └── FrameRelayUni (FrUni)
          └── Framer
                 frmFromIf
          └── DataLinkConnectionIdentifier (Dlci)
                 frmFromIf
                 bytesFromIf
                    └── DirectCall (Dc)
                           remoteDlci=  ◄─────┐
                    └── VirtualCircuit (Vc)   │
                           sendSequenceNumber │
   └── FrameRelayNni (FrNni)                  │
          └── Framer                          │
                 frmToIf                      │
          └── DataLinkConnectionIdentifier (Dlci)
                 frmToIf                      │
                 bytesToIf                    │
                    └── DirectCall (Dc)       │
                           remoteDlci=  ◄─────┘
```
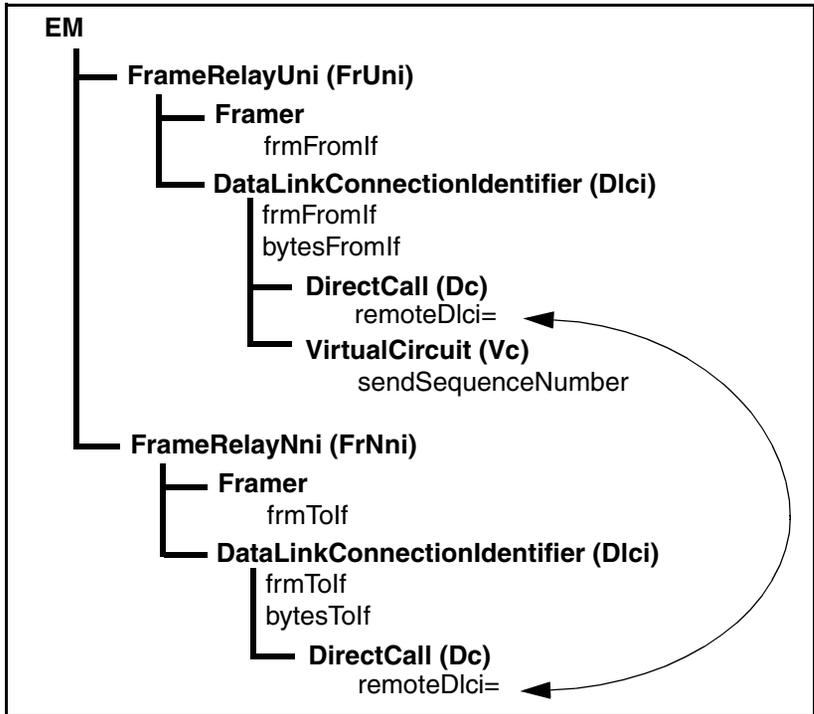
Verifying data flow across network connections

Verify data flow across network connections in order to verify that data is being transferred between adjacent networks.

Procedure steps

1   For a connection segment across networks, verify that the framer on the adjacent FR NNI is receiving data from the FR NNI and the *frmFromIf* attribute is increasing.

```
display FrNni/<FrNni_inst> Framer
```

On a 4-port DS3 channelized frame relay FP, if the timeslot value of none is provisioned, then the channel associated with the framer component is in a ready, enabled, idle state and does not transfer data. A provisioned timeslot value of none prevents the DLCI component from being provisionable.

2   Verify that the FR NNI DLCI in the adjacent network is receiving data from the FR NNI and the *frmFromIf* and *bytesFromIf* attributes are increasing.

```
display FrNni/<FrNni_adj> Dlci/<dlci_no> Stats
```

3   Verify that the adjacent FR NNI is transmitting data into its subnet and the *sendSequenceNumber* attribute is incrementing in relation to data flow and reflects the send sequence number of the last transmitted frame.

```
display FrNni/<FrNni_inst> Dlci/<dlci_no> Vc
```

Variable definitions

Variable	Value
<dlci_no>	is the instance number of the DLCI.
<FrNni_adj>	is the instance number of the FR NNI in the adjacent network.
<FrUni_inst>	is the instance number of the FR UNI.

Procedure job aid

Figure 59
Data flow across network connections component hierarchy

```
EM
 │
 ├── FrameRelayUni (FrUni)
 │    │
 │    ├── Framer
 │    │      frmFromIf
 │    │
 │    └── DataLinkConnectionIdentifier (Dlci)
 │           frmFromIf
 │           bytesFromIf
 │           │
 │           ├── DirectCall (Dc)
 │           │      remoteDlci=        ◄──────┐
 │           │                                 │
 │           └── VirtualCircuit (Vc)           │
 │                  sendSequenceNumber         │
 │                                             │
 └── FrameRelayNni (FrNni)                     │
      │                                        │
      ├── Framer                               │
      │      frmToIf                           │
      │                                        │
      └── DataLinkConnectionIdentifier (Dlci)  │
             frmToIf                           │
             bytesToIf                         │
             │                                 │
             └── DirectCall (Dc)               │
                    remoteDlci=  ◄─────────────┘
```
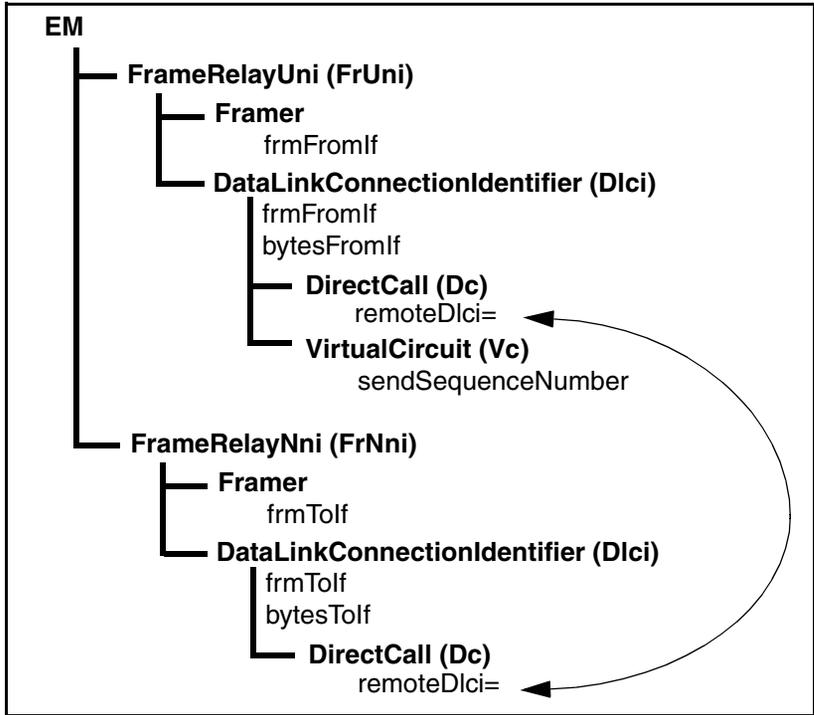
Isolating points of congestion

Isolate points of congestion to discover the point at which data is discarded for each end of the connection.

Procedure steps

1 Determine the resource utilization levels of the FP providing frame relay service.

`display Lp/<lp_no> Util`

If the CPU utilization is at 100%, the FP may be congested and losing data.

2 Display the statistics for the interface.

`display FrUni/<FrUni_inst> Dlci/<dlci_no> Stats`

`display FrNni/<FrNni_inst> Dlci/<dlci_no> Stats`

3 Record the values of the attributes *fecnFrmFromIf* and *fciToSubnet* (outgoing frames to the subnet encounter congestion).

4 Record the values of the attributes *becnFrmToIf* and *bciFromSubnet* (incoming frames warn interface of congestion encountered toward the subnet).

5 Record the value of the attribute *discCongestedFromIf* (outgoing frames discarded due to local congestion toward subnet).

6 Repeat the display command.

`display FrUni/<FrUni_inst> Dlci/<dlci_no> Stats`

`display FrNni/<FrNni_inst> Dlci/<dlci_no> Stats`

7 Check for increases in the values of the attributes *fecnFrmFromIf* and *fciToSubnet* (outgoing frames to the subnet encounter congestion), which indicate congestion in the forward direction (toward the network).

8 Check for increases in the values of the attributes *becnFrmToIf* and *bciFromSubnet* (incoming frames warn interface of congestion encountered toward the subnet), which indicate congestion in the forward direction (toward the network).

9 Check for an increase in the value of the attribute *discCongestedFromIf* (outgoing frames discarded due to local congestion toward subnet), which indicates congestion in the forward direction (toward the network).

10 Display the interface statistics.

```
display FrUni/<FrUni_inst> Dlci/<dlci_no> Stats
```

```
display FrNni/<FrNni_inst> Dlci/<dlci_no> Stats
```

11 Check the attributes *fecnFrmToIf* and *fciFromSubnet* (incoming frames to the interface encountered congestion) for congestion in the backward direction (toward the link).

12 Check the attributes *becnFrmFromIf* and *bciToSubnet* (outgoing frames warn connected device of congestion encountered toward interface) for congestion in the backward direction (toward the link).

13 Check the attribute *discCongestedToIf* (incoming frames discarded due to local congestion toward interface) for congestion in the backward direction (toward the link).

14 Repeat the display command.

```
display FrUni/<FrUni_inst> Dlci/<dlci_no> Stats
```

```
display FrNni/<FrNni_inst> Dlci/<dlci_no> Stats
```

15 Check for an increase in the attributes *fecnFrmToIf*, *fciFromSubnet*, *becnFrmFromIf*, *bciToSubnet*, and *discCongestedToIf*.

If there is congestion in the backward direction, one or more of the attribute values will increase.

Variable definitions

Variable	Value
<dlci_no>	is the instance number of the DLCI.
<FrNni_inst>	is the instance number of the FR NNI.
<FrUni_inst>	is the instance number of the FR UNI.
<lp_no>	is the instance number of the logical processor associated with the FP.

Procedure job aid

Figure 60
Points of congestion FR UNI component hierarchy

```
EM
    └── FrameRelayUni (FrUni)
            └── DataLinkConnectionIdentifier (DIci)
                    fecnFrmToIf
                    becnFrmToIf
                    fciToSubnet
                    bciToSubnet
                    discCongestedToIf
                    fecnFrmFromIf
                    becnFrmFromIf
                    fciFromSubnet
                    bciFromSubnet
                    discCongestedFromIf
```
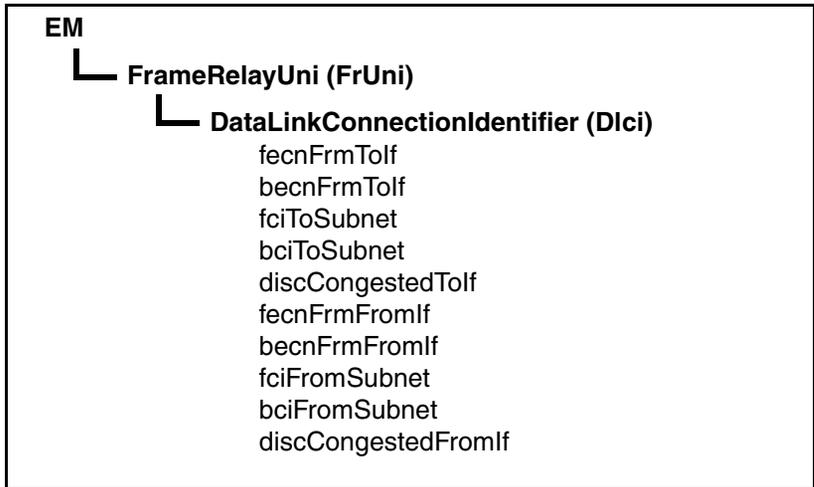
Figure 61
Points of congestion FR NNI component hierarchy

```
EM
    └── FrameRelayNni (FrNni)
            └── DataLinkConnectionIdentifier (DIci)
                    fecnFrmToIf
                    becnFrmToIf
                    fciToSubnet
                    bciToSubnet
                    discCongestedToIf
                    fecnFrmFromIf
                    becnFrmFromIf
                    fciFromSubnet
                    bciFromSubnet
                    discCongestedFromIf
```
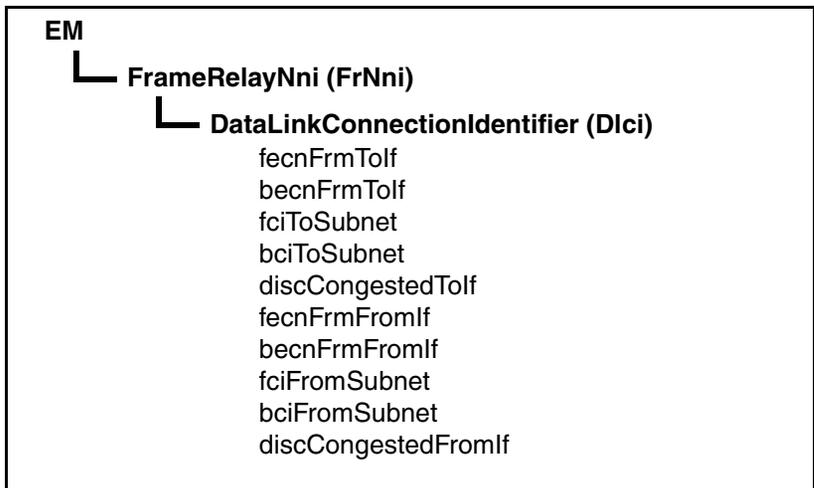
Isolating data loss at the interface

Isolate data loss at the interface to determine if the transmission problem occurs at the interface.

Procedure steps

1 Display the interface's framer attributes.

```
display FrUni/<FrUni_inst> Framer
```

```
display FrNni/<FrNni_inst> Framer
```

2 Record the value of the *aborts* attribute.

3 Record the value of the *crcErrors* attribute.

4 Record the value of the *overruns* attribute.

5 Record the value of the *underruns* attribute.

6 Repeat the display command.

```
display FrUni/<FrUni_inst> Framer
```

```
display FrNni/<FrNni_inst> Framer
```

7 Check the *aborts* attribute. Any increase indicates data loss caused by a bad cable or incorrect hardware configuration.

8 Check the *crcErrors* attribute. Any increase indicates data loss caused by a bad cable or incorrect hardware configuration.

9 Check the *overruns* attribute. Any increase indicates data loss caused by a bad cable or incorrect hardware configuration.

10 Check the *underruns* attribute. Any increase indicates data loss caused by a bad cable or incorrect hardware configuration.

11 Display the interface's framer attributes.

```
display FrUni/<FrUni_inst> Framer
```

```
display FrNni/<FrNni_inst> Framer
```

12 Record the value of the *largeFrmErrors* attribute.

13 Record the value of the*nonOctetErrors* attribute.

14 Repeat the display command to check for an increase in these attributes.

```
display FrUni/<FrUni_inst> Framer
```

```
display FrNni/<FrNni_inst> Framer
```

15   Check the *largeFrmErrors* attribute. Any increase may indicate data loss due to corrupted data.

16   Check the *nonOctetErrors* attribute. Any increase may indicate data loss due to corrupted data.

17   Display statistics for the interface.

```
display FrUni/<FrUni_inst> Stats
```

```
display FrNni/<FrNni_inst> Stats
```

18   Record the value of the *lastUnknownDlci* attribute

19   Record the value of the *unknownDlciFramesFromIf* attribute

20   Record the value of the *nvalidHeaderFramesFromIf* attribute

21   Repeat the display command.

```
display FrUni/<FrUni_inst> Stats
```

```
display FrNni/<FrNni_inst> Stats
```

22   Check the *lastUnknownDlci* attribute. An increase may indicate data loss caused due to incorrect service configuration.

23   Check the *unknownDlciFramesFromIf* attribute. An increase may indicate data loss caused due to incorrect service configuration.

24   Check the *nvalidHeaderFramesFromIf* attribute. An increase may indicate data loss caused due to incorrect service configuration.

Variable definitions

Variable	Value
<FrNni_inst>	is the instance number of the FR NNI interface.
<FrUni_inst>	is the instance number of the FR UNI interface.

Procedure job aid

Figure 62
Data loss at the interface FR UNI component hierarchy

```
EM
    └── FrameRelayUni (FrUni)
            lastUnknownDlci
            unknownDlciFramesFromIf
            invalidHeaderFramesToIf

            └── Framer
                    ifAborts
                    crcErrors
                    overruns
                    underruns
                    largeFrmErrors
                    nonOctetErrors
```
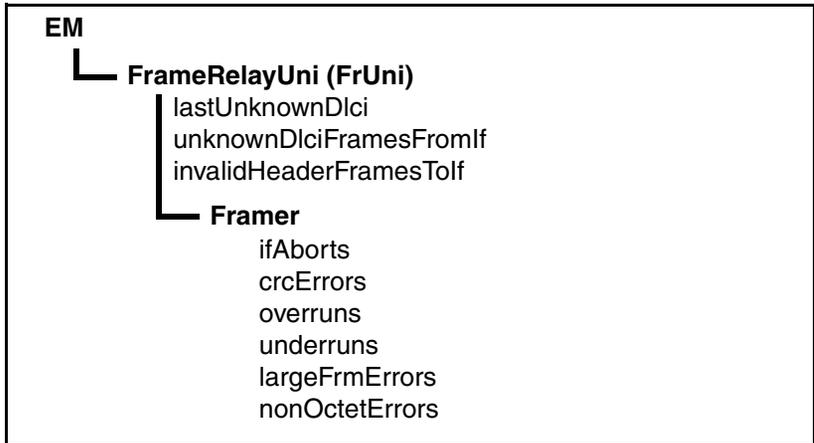
Figure 63
Data loss at the interface FR NNI component hierarchy

```
EM
    └── FrameRelayNni (FrNni)
            lastUnknownDlci
            unknownDlciFramesFromIf
            invalidHeaderFramesToIf

            └── Framer
                    ifAborts
                    crcErrors
                    overruns
                    underruns
                    largeFrmErrors
                    nonOctetErrors
```
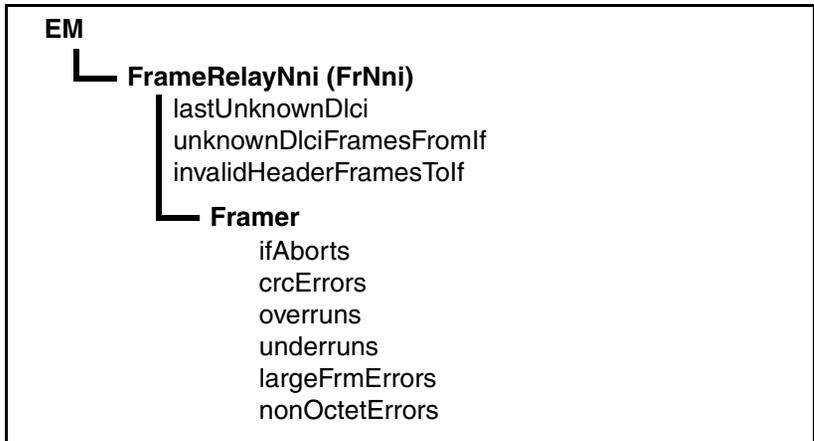
Isolating data loss on a connection

Isolate data loss on a connection to determine if the transmission problem occurs on the connection.

Procedure steps

1 Display frame statistics for the connection.

   display FrUni/<FrUni_inst> Dlci/<dlci_no> Stats

   display FrNni/<FrNni_inst> Dlci/<dlci_no> Stats

2 Record the value of the *discFrameAbit* attribute.

3 Record the value of the *discExcessFromIf* attribute.

4 Record the value of the *errorLongFrmFromIf* attribute.

5 Record the value of the *errorShortBytesFromIf* attribute.

6 Repeat the display command.

   display FrUni/<FrUni_inst> Dlci/<dlci_no> Stats

   display FrNni/<FrNni_inst> Dlci/<dlci_no> Stats

7 Check the *discFrameAbit* attribute. An increase indicates data loss toward the network due to inactive PVC status.

8 Check the *discExcessFromIf* attribute. An increase indicates data loss due to rate enforcement.

9 Check the *errorLongFrmFromIf* attribute. An increase indicates data loss due to frame length that exceeds the size configured in the *maximumFrameSize* attribute under the *FrUni Dlci Sp* component.

10 Check the *errorShortBytesFromIf* attribute. An increase indicates data loss due to frame length that has a zero byte payload.

11 Display frame statistics for the virtual circuit.

   display FrUni/<FrUni_inst> Dlci/<dlci_no> Vc

   display FrNni/<FrNni_inst> Dlci/<dlci_no> Vc

12 Record the value of the *notDataXferToSubnet* attribute.

13 Record the value of the *notDataXferFromSubnet* attribute.

14 Record the value of the *combErrorsFromSubnet* attribute.

15 Record the value of the *outOfRangeFrmFromSubnet* attribute.

16 Record the value of the *ooSeqPktCntExceeded* attribute. The *ooSeqPktCntExceeded* value indicates the number of times the out of sequence packet threshold is exceeded.

17 Record the value of the *peakOoSeqPktCount* attribute. The *peakOoSeqPktCount* value displays the peak queue length of the out of sequence queue.

18 Repeat the display command.

```
display FrUni/<FrUni_inst> Dlci/<dlci_no> Vc

display FrNni/<FrNni_inst> Dlci/<dlci_no> Vc
```

19 Check the *notDataXferToSubnet* attribute. An increase indicates data discarded from the link during VC recovery from internal path failure.

20 Check the *notDataXferFromSubnet* attribute. An increase indicates data discarded from the subnet during VC recovery from internal path failure.

21 Check the *combErrorsFromSubnet* attribute. An increase indicates data loss due to segmentation and reassembly errors.

22 Check the *outOfRangeFrmFromSubnet* attribute. An increase indicates data loss of frames arriving at the interface too late to be within the permitted out of sequence range.

23 Check the *ooSeqPktCntExceeded* attribute. An increase indicates that you may need to re-engineer the out of sequence threshold.

24 Check the *peakOoSeqPktCount* attribute. An increase indicates that you may need to re-engineer the out of sequence threshold.

Variable definitions

Variable	Value
<dlci_no>	is the instance number of the DLCI.
<FrNni_inst>	is the instance number of the FR NNI.
<FrUni_inst>	is the instance number of the FR UNI.

Procedure job aid

Figure 64
Data loss on a connection FR UNI component hierarchy

```
EM
    └── FrameRelayUni (FrUni)
            └── DataLinkConnectionIdentifier (Dlci)
                    discFrameAbit
                    discExcessFromIf
                    errorLongFrmFromIf
                    errorShortBytesFromIf

                    └── VirtualCircuit (Vc)
                            notDataXerToSubnet
                            notDataXerFromSubnet
                            combErrorsFromSubnet
                            outOfRangeFrmFromSubnet
                            ooSeqPktCntExceeded
                            peakOoSeqByteCount
```
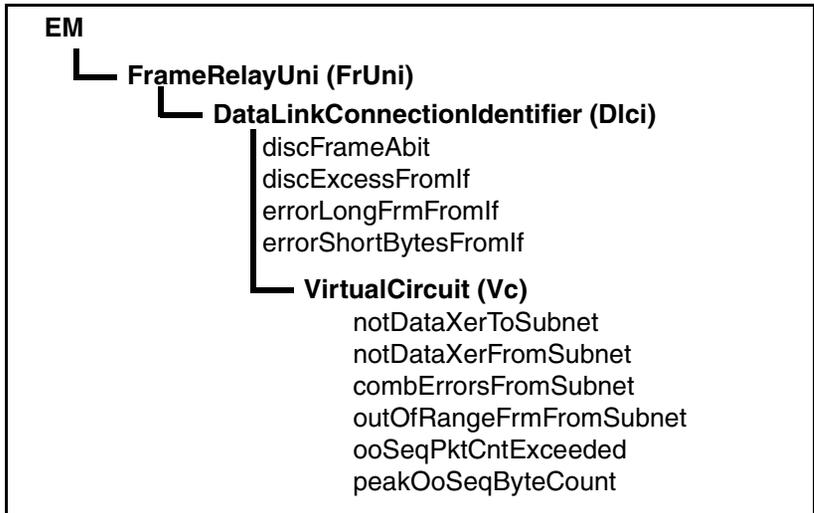
Figure 65
Data loss on a connection FR NNI component hierarchy

```
EM
    └── FrameRelayNni (FrNni)
            └── DataLinkConnectionIdentifier (Dlci)
                    discFrameAbit
                    discExcessFromIf
                    errorLongFrmFromIf
                    errorShortBytesFromIf

                    └── VirtualCircuit (Vc)
                            notDataXerToSubnet
                            notDataXerFromSubnet
                            combErrorsFromSubnet
                            outOfRangeFrmFromSubnet
                            ooSeqPktCntExceeded
                            peakOoSeqByteCount
```
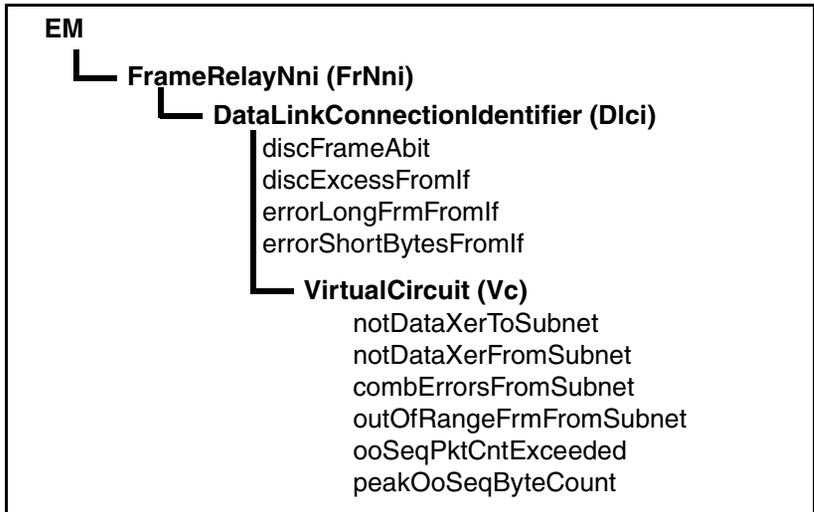
Isolating SVC connection failure

Isolate SVC connection failure to help determine if calls are rejected on a connection because of protocol errors, insufficient available bandwidth or quality of service (QoS), or improper configuration.

Procedure steps

1 Display signaling statistics for the interface.

   display FrUni/<FrUni_inst> Sig Statistics

   display FrNni/<FrNni_inst> Sig Statistics

2 Check the *inCallsRefused* attribute for the number of incoming call requests rejected by the interface.

   The interface rejects calls if it is unable to provide the requested QoS or bandwidth, or if it detects protocol errors.

3 Check the *outCallsFailed* attribute for the number of outgoing call requests rejected by the interface, the network side, or the called user.

   The network side or called user rejects calls if it is unable to provide the requested QoS or bandwidth. Call rejection may also be due to network congestion or data corruption.

4 Check the *protocolErrors* attribute for the number of call setup failures due to unexpected or corrupted messages received by the interface.

5 Check the *qualityOfServiceNotAvailable* attribute for the number of call setup failures due to the interface's inability to provide the requested QoS (as signaled in the setup message).

6 Display all configured hunt group addresses for the FR UNI.

   display -p FrUni/<FrUni_inst> Dna Hgm HgA

   display -p FrNni/<FrNni_inst> Dna Hgm HgA

7 Ensure that a hunt group address is configured on the interface for each hunt group that includes the FR UNI as a member.

8 Ensure that the hunt group address information is accurate.

9 Display all configured hunt groups.

   display -p Hg/*

10 Ensure that the FR NNI's or FR UNI's address is configured as a hunt group member address on each hunt group that includes it.

11   Ensure that the hunt group member address information is accurate.

Variable definitions

Variable	Value
<FrNni_inst>	is the instance number of the FR NNI.
<FrUni_inst>	is the instance number of the FR UNI.

Procedure job aid

Figure 66
SVC connection failure FR UNI component hierarchy
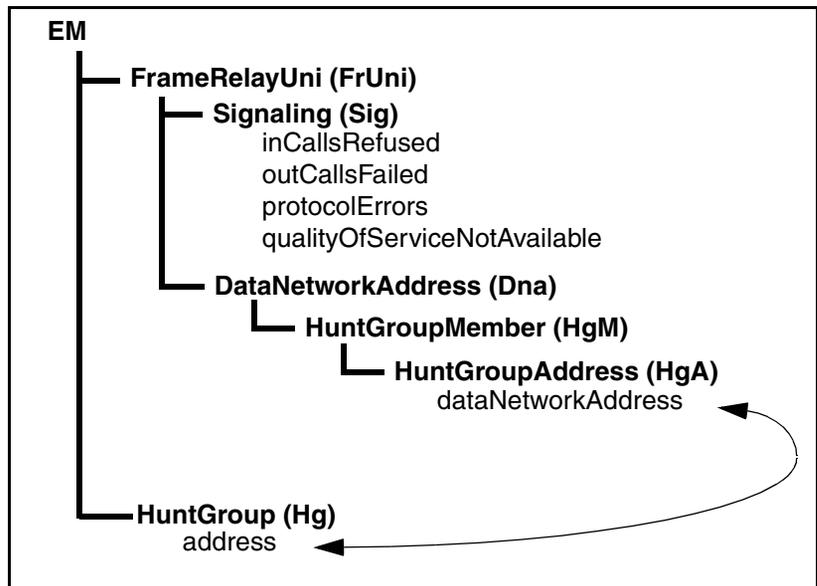
Figure 67
SVC connection failure FR NNI component hierarchy

```
EM
   ┣━ FrameRelayNni (FrNni)
   ┃     ┣━ Signaling (Sig)
   ┃     ┃        inCallsRefused
   ┃     ┃        outCallsFailed
   ┃     ┃        protocolErrors
   ┃     ┃        qualityOfServiceNotAvailable
   ┃     ┃        bandwidthNotAvailable
   ┃     ┃        dlciCollision
   ┃     ┃
   ┃     ┗━ DataNetworkAddress (Dna)
   ┃              ┗━ HuntGroupMember (HgM)
   ┃                       ┗━ HuntGroupAddress (HgA)
   ┃                                dataNetworkAddress
   ┃
   ┗━ HuntGroup (Hg)
            address
```
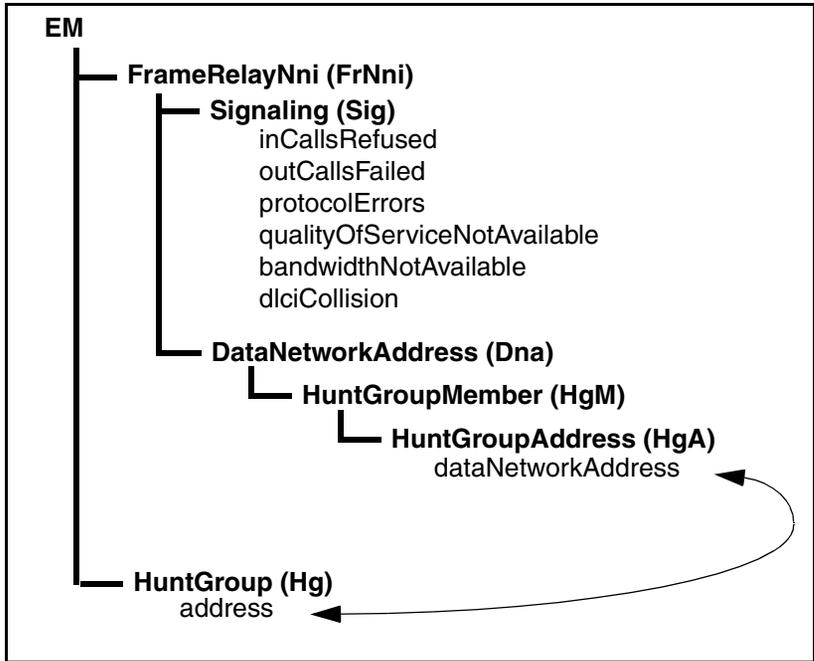
Monitoring rate adaptation

Monitor rate adaptation to display information about rate adaptation on a connection.

Procedure steps

1   Display the number of completed rate reductions on a specific connection.

```
display FrUni/<FrUni_inst> Dlci/<dlci_no>
rateAdaptReduct
```

```
display FrNni/<FrNni_inst> Dlci/<dlci_no>
rateAdaptReduct
```

2   Display in milliseconds how long rate adaptation procedures have been in effect.

```
display FrUni/<FrUni_inst> Dlci/<dlci_no>
rateAdaptReductPeriod
```

```
display FrNni/<FrNni_inst> Dlci/<dlci_no>
rateAdaptReductPeriod
```

3   Display the actual committed information rate (CIR) for the connection.

```
display FrUni/<FrUni_inst> Dlci/<dlci_no> cirPresent
```

```
display FrNni/<FrNni_inst> Dlci/<dlci_no> cirPresent
```

If rate enforcement is off, the *cirPresent* attribute matches the configured CIR for the connection.

If rate adaptation is enabled, the *cirPresent* attribute displays the allowed CIR. This parameter may be less than its configured value.

4   Display the actual excess information rate (EIR) for the connection.

```
display FrUni/<FrUni_inst> Dlci/<dlci_no> eirPresent
```

```
display FrNni/<FrNni_inst> Dlci/<dlci_no> eirPresent
```

If rate enforcement is off, the *eirPresent* attribute matches the configured EIR value.

If rate adaptation is enabled, the *eirPresent* attribute displays the allowed EIR. This parameter may be less than its configured values.

Variable definitions

Variable	Value
<dlci_no>	is the instance number of the DLCI.
<FrNni_inst>	is the instance number of the FR NNI.
<FrUni_inst>	is the instance number of the FR UNI.

Procedure job aid

Figure 68
Rate adaptation FR UNI component hierarchy

EM

    └ **FrameRelayUni (FrUni)**

        └ **DataLinkConnectionIdentifier (Dlci)**
            rateAdaptReduct
            rateAdaptReductPeriod
            cirPresent
            eirPresent
            bciFromSubnet
            becnFrmFromIf
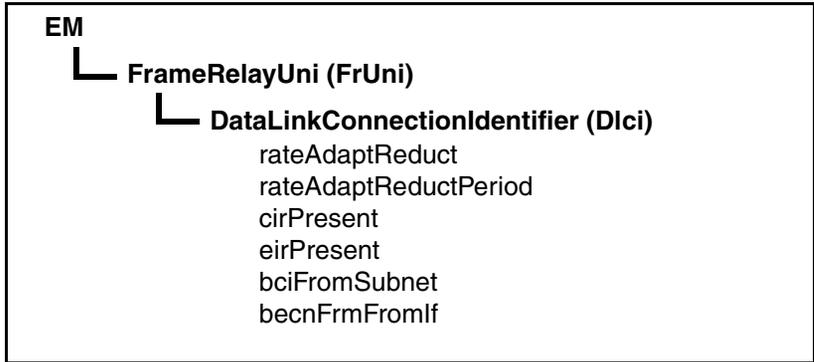
Figure 69
Rate adaptation FR NNI component hierarchy

EM

    └ **FrameRelayNni (FrNni)**

        └ **DataLinkConnectionIdentifier (Dlci)**
            rateAdaptReduct (rar)
            rateAdaptReductPeriod (rarp)
            cirPresent
            eirPresent
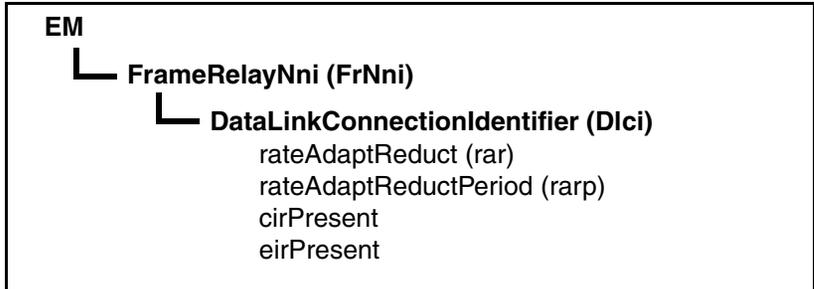
Monitoring FR UNI FRF.12 statistics

Monitor FR UNI FRF.12 statistics to view the operational statistics related to end-to-end FRF.12 fragmentation and reassembly for a DLCI.

Prerequisites

- The *frfEndToEnd* attribute must be turned on.

Procedure steps

1   Display FRF.12 statistics.

```
d FrUni/<FrUni_inst> Dlci/<dlci_no> frf12Stats
```

Variable definitions

Variable	Definition
<dlci_no>	is the instance number of the DLCI.
<FrUni_inst>	is the instance number of the FR UNI.

Procedure job aid

Figure 70
FRF.12 statistics component hierarchy

```
Em
  └── FrameRelayUni (FrUni)
        └── DataLinkConnectionIdentifier (Dlci)
              └── Frf12Stats
                    fragFromIf
                    peakFragSizeFromIf
                    expectedSequenceEvent
                    outOfSequenceEvent
                    headerErrorEvent
                    cpuExhaustionFromIfEvent
                    discFragFromIf
                    fragToIf
                    sendSequenceNumber
                    cpuExhaustionToIfEvent
                    longFrameToIfEvent
                                          PPT 3311 003 AA
```
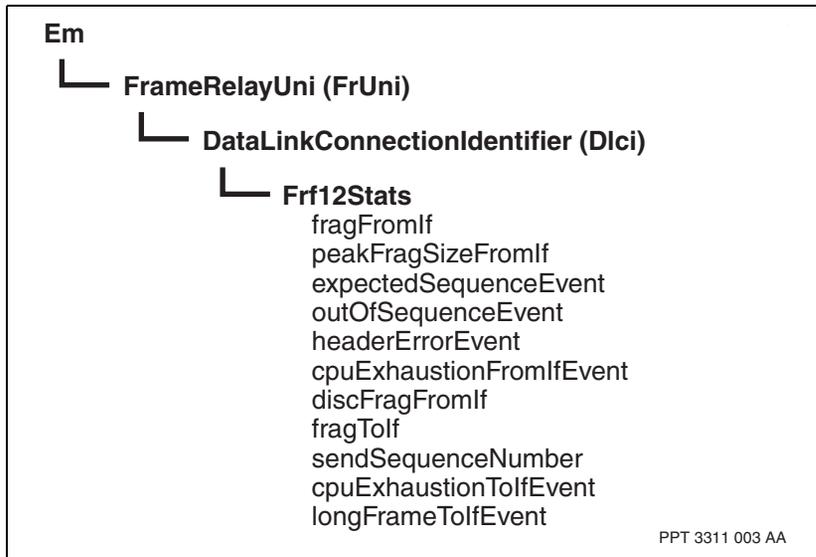
Monitoring hardware resource availability

Monitor hardware resource availability to verify that the number of DLCIs and sub-connections provisioned on the logical processor are within capacity.

Procedure steps

1   Display the number of FrUni/FrNni DLCIs that are currently under the Lp.

```
d lp/<v> eng frs currentCalls
```

2   Display the number of sub-connections on the Lp.

```
d lp/<v> eng fcrc
```

IP-optimized DLCI requires two sub-connections.

3   Delete one or more DLCIs to free up hardware resources.

```
del FrUni/<w> Dlci/<x>
```

```
del FrNni/<y> Dlci/<x>
```

```
del FrAtm/<u> Dlci/<x>
```

Deleting any type of DLCI will free up resources.

4   Unlock one or more DLCI after hardware resources become available.

```
unlock FrUni/<w> Dlci/<z>
```

```
unlock FrNni/<y> Dlci/<z>
```

Variable definitions

Variable	Value
<u>	is the instance number of the FrAtm.
<v>	is the instance number of the Lp.
<w>	is the instance number of the FrUni.
<x>	is the instance number of the DLCI to be deleted.
<y>	is the instance number of the FrNni.
<z>	is the instance number of the DLCI to be unlocked.

Procedure job aid

Figure 71
Hardware resource availability component hierarchy

```
Em
    ── LogicalProcessor (Lp)
            └── Engineering (Eng)
                    ── FrameRelaySystem (Frs)
                         currentCalls
                    └── FrameCoreResourceControl (Fcrc)
    ── FrAtm
            └── DataLinkConnectionIdentifier (Dlci)
    ── FrUni
            └── DataLinkConnectionIdentifier (Dlci)
    └── FrNni
            └── DataLinkConnectionIdentifier (Dlci)

                                            PPT 3331 001 AA
```
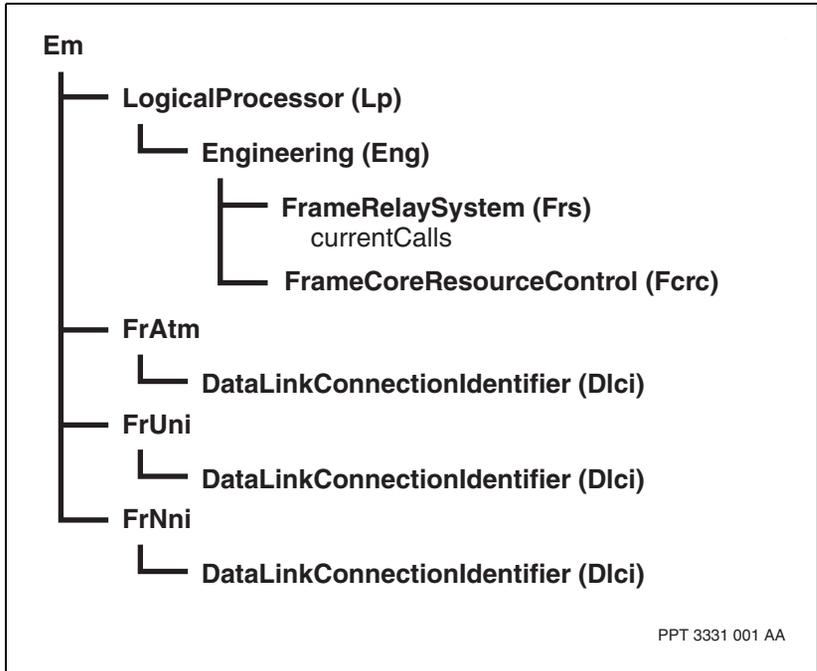
Using a frame link monitor to determine connection status

Use a frame link monitor to determine connection status to discover the state of a HDLC connection.

Procedure steps

1   Display the value of the *flmStatus* attribute.

```
display Lp/<lp_no> E1/<port_num> Chan/<E1_chan> Flm
flmStatus
```

Variable definitions

Variable	Value
<E1_chan>	is the instance number of the *Chan* component, having a value between 0 and 31.
<Lp_no>	is the instance number of the *Lp* component for the FP providing the service to be monitored. Because the 32-port DS1/E1 MSA FP is a double-width card, the instance number has a value between 1 and 14 inclusive.
<port_num>	is the instance number of the E1 component, having a value between 0 and 31.

Procedure job aid

Table 5
Interpreting the values of the *flmStatus* attribute for the 32-port DS1/E1 MSA FP on Multiservice Switch 7400 nodes

flmStatus value	A-bit status	HDLC flag status
frameLinkUp	valid	present
lossOfHdlc	valid	not present
lossOfAbit	invalid	present
frameLinkDown	invalid	not present
(Sheet 1 of 2)		

Table 5 (continued)
Interpreting the values of the *flmStatus* attribute for the 32-port DS1/E1 MSA FP on Multiservice Switch 7400 nodes

flmStatus value	A-bit status	HDLC flag status
NotMonitoring	monitoring disabled	monitoring disabled
Note: The NotMonitoring value can also occur if the channel is locked or in testing mode.		
(Sheet 2 of 2)		

Table 6
Interpreting the values of the *flmStatus* attribute for the 1-port STM-1 FP on Multiservice Switch 15000 nodes

flmStatus value	A-bit status	Link status
LinkUp	unset (0)	The link is okay. The status indicates LinkUp.
AbitLoss	set (1)	The link is not working properly. The status indicates AbitLoss.
frameLinkDown	not available	The physical connection is incorrectly setup or not available. No signaling is available. The status indicates FrameLinkDown.
NotMonitoring	monitoring disabled	monitoring disabled
Note: The NotMonitoring value can also occur if the channel is locked or in testing mode.		

Initiating and terminating loopback mode

Initiate and terminate loopback mode to turn the data loopback tool on and off for a connection.

Prerequisites

- To start and terminate loopback mode, you must have a minimum service level of operator.

Procedure steps

1   Initiate loopback on a specific connection.

    start FrUni/<FrUni_inst> Dlci/<dlci_no> loopback

    start FrNni/<FrNni_inst> Dlci/<dlci_no> loopback

2   Display the loopback statistics.

    display FrNni/<FrNni_inst> Dlci/<dlci_no> loopback stats

    display FrUni/<FrUni_inst> Dlci/<dlci_no> loopback stats

3   Terminate loopback on a specific connection.

    stop FrUni/<FrUni_inst> Dlci/<dlci_no> loopback

    stop FrNni/<FrNni_inst> Dlci/<dlci_no> loopback

When you issue a command to terminate loopback mode, the operational attributes that contain the loopback statistics display automatically.

Variable definitions

Variable	Value
<dlci_no>	is the instance number of the DLCI.
<FrNni_inst>	is the instance number of the FR NNI.
<FrUni_inst>	is the instance number of the FR UNI.

Procedure job aid

Figure 72
Loopback mode FR UNI component hierarchy

```
EM
    └── FrameRelayUni (FrUni)
            └── DataLinkConnectionIdentifier (DIci)
                    └── LoopBack (Lb)
                            localTotalFrm (ltf)
                            localTotalBytes (ltb)
                            localFecnFrm (lff)
                            localBecnFrm (lbf)
                            localDeFrm (ldf)
                            localDeBytes (ldb)
                            remoteTotalFrm (rtf)
                            remoteTotalBytes (rtb)
                            remoteFecnFrm (rff)
                            remoteBecnFrm (rbf)
                            remoteDeFrm (rdf)
                            remoteDeBytes (rdb)
```
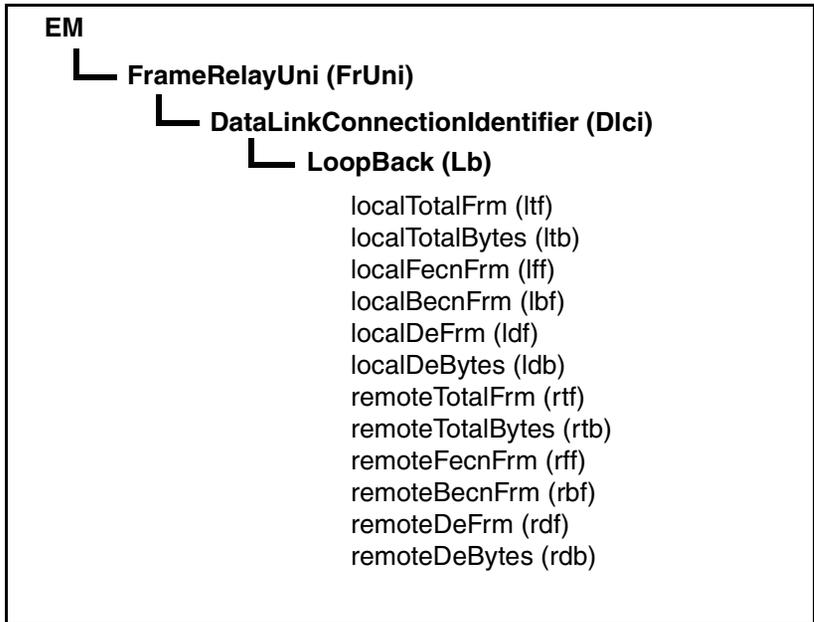
Figure 73
Loopback mode FR NNI component hierarchy

```
EM
    └── FrameRelayNni (FrNni)
            └── DataLinkConnectionIdentifier (Dlci)
                    └── LoopBack (Lb)
                                localTotalFrm (ltf)
                                localTotalBytes (ltb)
                                localFecnFrm (lff)
                                localBecnFrm (lbf)
                                localDeFrm (ldf)
                                localDeBytes (ldb)
                                remoteTotalFrm (rtf)
                                remoteTotalBytes (rtb)
                                remoteFecnFrm (rff)
                                remoteBecnFrm (rbf)
                                remoteDeFrm (rdf)
                                remoteDeBytes (rdb)
```
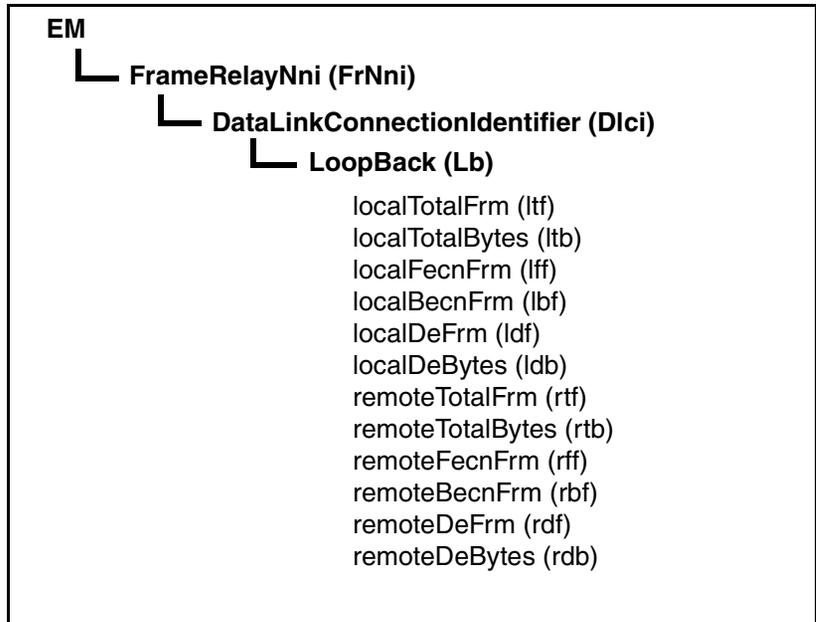
Chapter 10
Module-wide QOS monitoring

Monitor module-wide quality of service (QOS) to display and verify transfer, emission and routing priority mappings on a module-wide or a per-connection basis.

Module-wide QOS monitoring procedures

This task flow show you the procedures you can perform to monitor module-wide QOS. To link to any procedure, go to "Module-wide QOS monitoring procedure navigation" (page 176).

Figure 74
Module-wide QOS monitoring procedures

MSS 3538 006 AA
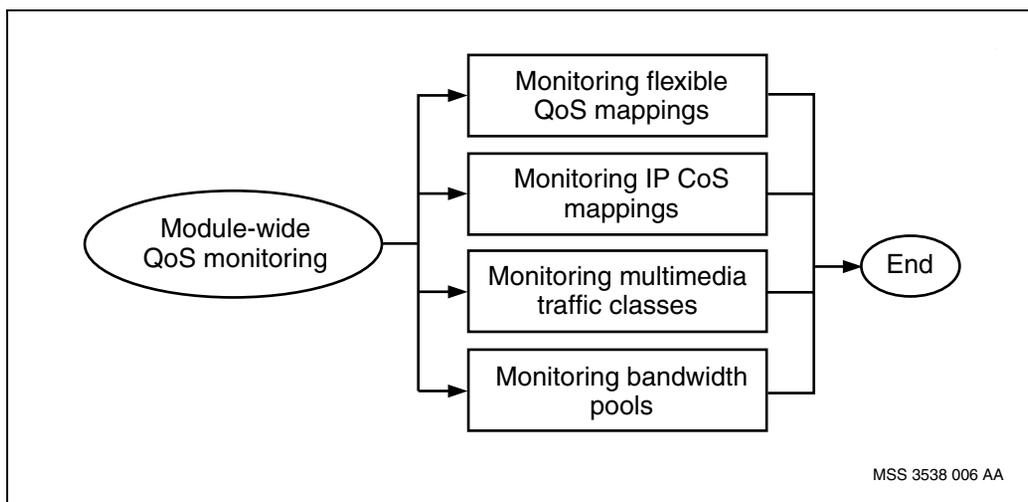
Module-wide QOS monitoring procedure navigation

Monitoring flexible QoS mappings

Monitor flexible QoS mappings to check the assignment of routing class of service (RCOS) and emission priority values to specific transfer priorities.

Procedure steps

1   Determine the QoS mappings for all transfer priorities on the node.

```
display -p Mod Frs dprsNet Tpm/*
```

2   Determine the routing class of service and emission priority mappings for a specific transfer priority.

```
display Mod Frs dprsNet Tpm/<tp> rcos, ep
```

3   Determine the transfer priority of frames transmitted to and from the network on a specific connection.

```
display FrUni/<FrUni_inst> Dlci/<dlci_no> tpToNwk,
tpFromNwk
```

```
display FrNni/<FrNni_inst> Dlci/<dlci_no> tpToNwk,
tpFromNwk
```

4   Determine the routing class of service for frames transmitted to and from the network on a specific connection.

```
display FrUni/<FrUni_inst> Dlci/<dlci_no> Vc rcosTo,
rcosFrom
```

```
display FrNni/<FrNni_inst> Dlci/<dlci_no> Vc rcosTo,
rcosFrom
```

5   Determine the internal emission priority for frames transmitted to and from the network on a specific connection.

```
display FrUni/<FrUni_inst> Dlci/<dlci_no> Vc epTo,
epFrom
```

```
display FrNni/<FrNni_inst> Dlci/<dlci_no> Vc epTo,
epFrom
```

6   Verify the Nortel Networks Multiservice Switch trunking queue on which the frames are forwarded.

```
display Rtg Dpn lp/* interruptingPackets,
throughputPackets, delayPackets
```

Ensure that the correct attribute is incremented (determined by attribute *rcosToNetwork*).

---

7 Verify the egress emission queue on which the frames are forwarded.

```
display FrUni/<FrUni_inst> frmToIfByQueue
```

```
display FrNni/<FrNni_inst> frmToIfByQueue
```

The *frmToIfByQueue* attribute is an array with indices 0 to 3. Ensure that the correct index is incremented (determined by the combination of attributes *numberOfEmissionQs* and *emissionPriorityToIf*).

The *frmToIfByQueue* only returns a value if the *numberOfEmissionQs* attribute (under the frame relay interface) is set to 4.

Variable definitions

Variable	Value
<dlci_no>	is the instance number of the DLCI.
<FrNni_inst>	is the instance number of the FR NNI interface.
<FrUni_inst>	is the instance number of the FR UNI interface.
<tp>	is the instance number the transfer priority.

Procedure job aid

Figure 75
Flexible QoS mapping FR UNI component hierarchy

```
EM
    ModuleData (Mod)
        FrameRelaySystem (Frs)
            DprsNetworking (DprsNet)
                TransferPriorityMapping (Tpm)
                    routingClassOfService
                    emissionPriority

    Routing (Rtg)
        DpnAddressPlan (Dpn)
            interruptingPackets
            throughputPackets
            delayPackets

    FrameRelayUni (FrUni)
        DataNetworkAddress (Dna)
            transferPriorityOverride
        DataLinkConnectionIdentifier (Dlci)
            transferPriorityToNetwork
            transferPriorityFromNetwork

                VirtualCircuit (Vc)
                    rcosToNetwork
                    rcosFromNetwork
                    emissionPriorityToNetwork
                    emissionPriorityFromNetwork
```

Figure 76
Flexible QoS mapping FR NNI component hierarchy

```
EM
   ├── ModuleData (Mod)
   │      └── FrameRelaySystem (Frs)
   │             └── DprsNetworking (DprsNet)
   │                    └── TransferPriorityMapping (Tpm)
   │                           routingClassOfService (rcos)
   │                           emissionPriority (ep)
   ├── Routing (Rtg)
   │      └── DpnAddressPlan (Dpn)
   │             interruptingPackets
   │             throughputPackets
   │             delayPackets
   └── FrameRelayNni (FrNni)
          ├── DataNetworkAddress (Dna)
          │      transferPriorityOverride (tpor)
          └── DataLinkConnectionIdentifier (Dlci)
                 transferPriorityToNetwork
                 transferPriorityFromNetwork
                 └── VirtualCircuit (Vc)
                        rcosToNetwork
                        rcosFromNetwork
                        emissionPriorityToNetwork
                        emissionPriorityFromNetwork
```

Monitoring IP CoS mappings

Monitor IP CoS mappings to check the assignment of discard and emission priority values to specific IP classes of service for PVC connections.

Procedure steps

1 Determine the QoS mappings for all IP type of service (TOS) values on the node.

```
display Mod Frs IpCosToFrQos Tosm/*
```

2 Determine the discard priority assigned to a specific IP TOS.

```
display Mod Frs IpCosToFrQos Tosm/<tos> dp
```

3 Determine the emission priority assigned to a specific IP TOS.

```
display Mod Frs IpCosToFrQos Tosm/<tos> ep
```

4 Determine the internal emission priority for frames transmitted to the interface on a specific connection.

```
display FrUni/<FrUni_inst> Dlci/<dlci_no> epToIf
```

```
display FrNni/<FrNni_inst> Dlci/<dlci_no> epToIf
```

If the IP CoS mapping feature is enabled on the connection, the emission priority could be different for each packet. In this case, the system displays the configured emission priority.

5 Determine the discard priority for frames transmitted to the interface on a specific connection.

```
display FrUni/<FrUni_inst> Dlci/<dlci_no> Vc priority
```

```
display FrNni/<FrNni_inst> Dlci/<dlci_no> Vc priority
```

If the IP CoS mapping feature is enabled on the connection, the discard priority could be different for each packet. In this case, the system displays the configured discard priority (normal).

6 Verify the egress emission queue on which the frames are forwarded.

```
display FrUni/<FrUni_inst> frmToIfByQueue
```

```
display FrNni/<FrNni_inst> frmToIfByQueue
```

The *frmToIfByQueue* only returns a value if the *numberOfEmissionQs* attribute is set to 4.

The *frmToIfByQueue* attribute is an array with indices 0 to 3. Ensure that the correct index is incremented (determined by the combination of attributes *numberOfEmissionQs* and *emissionPriorityToIf*).

Variable definitions

Variable	Value
<dlci_no>	is the instance number of the DLCI.
<FrNni_inst>	is the instance number of the FR NNI.
<FrUni_inst>	is the instance number of the FR UNI.
<tos>	is the IP TOS instance.

Procedure job aid

Figure 77
IP COS mappings FR UNI component hierarchy
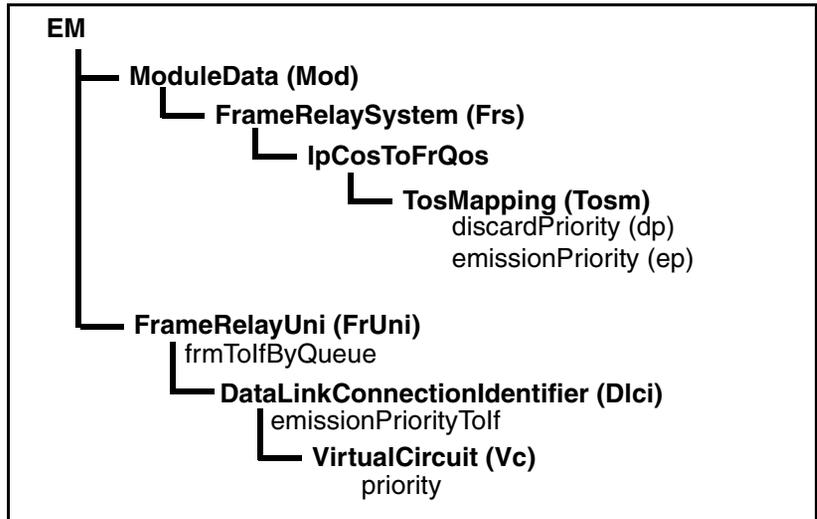
Figure 78
IP COS mappings FR NNI component hierarchy

```
EM
    ModuleData (Mod)
        FrameRelaySystem (Frs)
            IpCosToFrQos
                TosMapping (Tosm)
                    discardPriority (dp)
                    emissionPriority (ep)

    FrameRelayNni (FrNni)
    frmToIfByQueue
        DataLinkConnectionIdentifier (Dlci)
        emissionPriorityToIf
            VirtualCircuit (Vc)
                priority
```
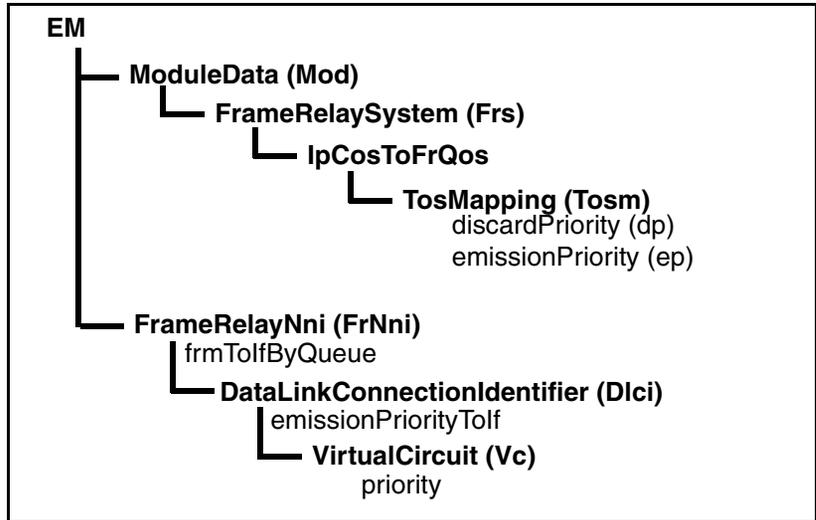
Monitoring multimedia traffic classes

Monitor multimedia traffic classes (MMTC) to display the traffic's transfer priority and the number of multimedia packets handled by an interface.

Procedure steps

1 Determine the transfer priority of frames transmitted to the network on a specific DLCI.

   display FrUni/<FrUni_inst> Dlci/<dlci_no> tpToNwk

   display FrNni/<FrNni_inst> Dlci/<dlci_no> tpToNwk

2 Determine the transfer priority of frames transmitted from network on a specific DLCI.

   display FrUni/<FrUni_inst> Dlci/<dlci_no> tpFromNwk

   display FrNni/<FrNni_inst> Dlci/<dlci_no> tpFromNwk

3 Determine the total number of multimedia packets handled by DPN packet forwarding on the module.

   display Rtg Dpn interruptingPackets

4 Determine the number of multimedia packets handled by DPN packet forwarding on a specific port.

   display Rtg Dpn LpStats/<y> interruptingPackets

Variable definitions

Variable	Value
<dlci_no>	is the instance number of the DLCI.
<FrNni_inst>	is the instance number of the FR NNI.
<FrUni_inst>	is the instance number of the FR UNI.
<y>	is the instance number of the logical processor.

Procedure job aid

Figure 79
MMTC FR UNI component hierarchy

```
Em

    ┕━━ FrameRelayUni (FrUni)

            ┕━━ DataLinkConnectionIdentifier (Dlci)
                    transferPriorityToNetwork (tpToNwk)
                    transferPriorityFromNetwork (tpFromNwk)

    ┕━━ Routing (Rtg)

            ┕━━ DpnAddressPlan (Dpn)
                 ┃ interruptingPackets
                    ┕━━ LpStatistics (LpStats)
                            interruptingPackets
```

PPT 3347 007 AA
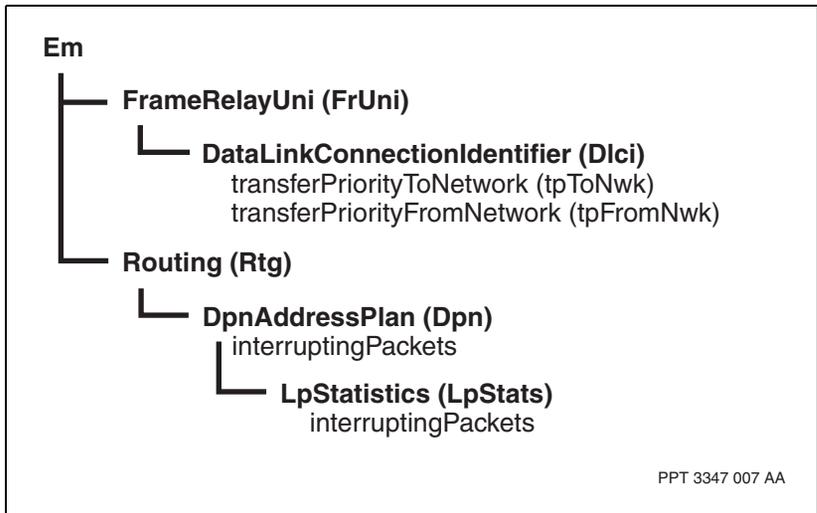
Figure 80
MMTC FR NNI component hierarchy

```
Em

    ┕━━ FrameRelayNni (FrNni)

            ┕━━ DataLinkConnectionIdentifier (Dlci)
                    transferPriorityToNetwork (tpToNwk)
                    transferPriorityFromNetwork (tpFromNwk)

    ┕━━ Routing (Rtg)

            ┕━━ DpnAddressPlan (Dpn)
                 ┃ interruptingPackets
                    ┕━━ LpStatistics (LpStats)
                            interruptingPackets
```

PPT 3347 006 AA
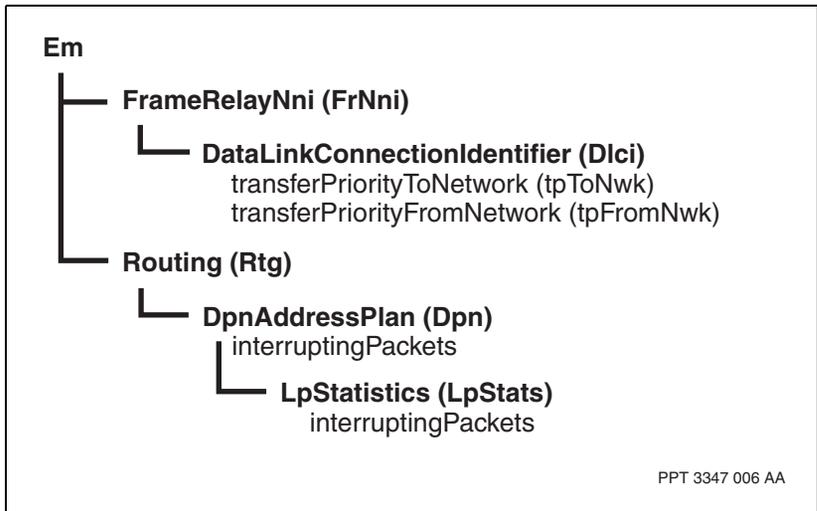
Monitoring bandwidth pools

Monitor bandwidth pools to monitor admitted and available bandwidth for ingress and egress calls by the connection administrator.

Procedure steps

1 Verify the link rate.

```
display FrUni/<FrUni_inst> Ca linkRate
```

```
display FrNni/<FrNni_inst> Ca linkRate
```

2 Determine the admitted CIR bandwidth for the ingress bandwidth pools. Each pool corresponds to a transfer priority.

```
display FrUni/<FrUni_inst> Ca ingCirPoolAdmitBw
```

```
display FrNni/<FrNni_inst> Ca ingCirPoolAdmitBw
```

3 Determine the admitted CIR bandwidth for the egress bandwidth pools. Each pool corresponds to a transfer priority.

```
display FrUni/<FrUni_inst> Ca egCirPoolAdmitBw
```

```
display FrNni/<FrNni_inst> Ca egCirPoolAdmitBw
```

4 Determine the admitted EIR bandwidth for the ingress bandwidth pools. Each pool corresponds to a transfer priority.

```
display FrUni/<FrUni_inst> Ca ingEirPoolAdmitBw
```

```
display FrNni/<FrNni_inst> Ca ingEirPoolAdmitBw
```

5 Determine the admitted EIR bandwidth for the egress bandwidth pools. Each pool corresponds to a transfer priority.

```
display FrUni/<FrUni_inst> Ca egEirPoolAdmitBw
```

```
display FrNni/<FrNni_inst> Ca egEirPoolAdmitBw
```

6 Check the available CIR bandwidth for the ingress bandwidth pools. Each pool corresponds to a transfer priority.

```
display FrUni/<FrUni_inst> Ca ingCirPoolAvailBw
```
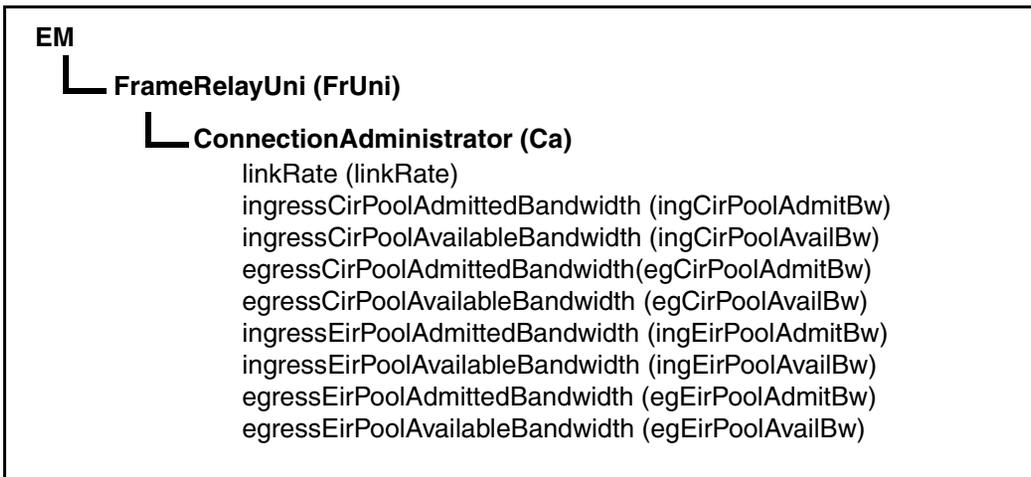
```
display FrNni/<FrNni_inst> Ca ingCirPoolAvailBw
```

7 Check the available CIR bandwidth for the egress bandwidth pools. Each pool corresponds to a transfer priority.

```
display FrUni/<FrUni_inst> Ca egCirPoolAvailBw
```

```
display FrNni/<FrNni_inst> Ca egCirPoolAvailBw
```

8   Check the available EIR bandwidth for the ingress bandwidth pools. Each pool corresponds to a transfer priority.

```
display FrUni/<FrUni_inst> Ca ingEirPoolAvailBw
```

```
display FrNni/<FrNni_inst> Ca ingEirPoolAvailBw
```

9   Check the available EIR bandwidth for the egress bandwidth pools. Each pool corresponds to a transfer priority.

```
display FrUni/<FrUni_inst> Ca egEirPoolAvailBw
```

```
display FrNni/<FrNni_inst> Ca egEirPoolAvailBw
```
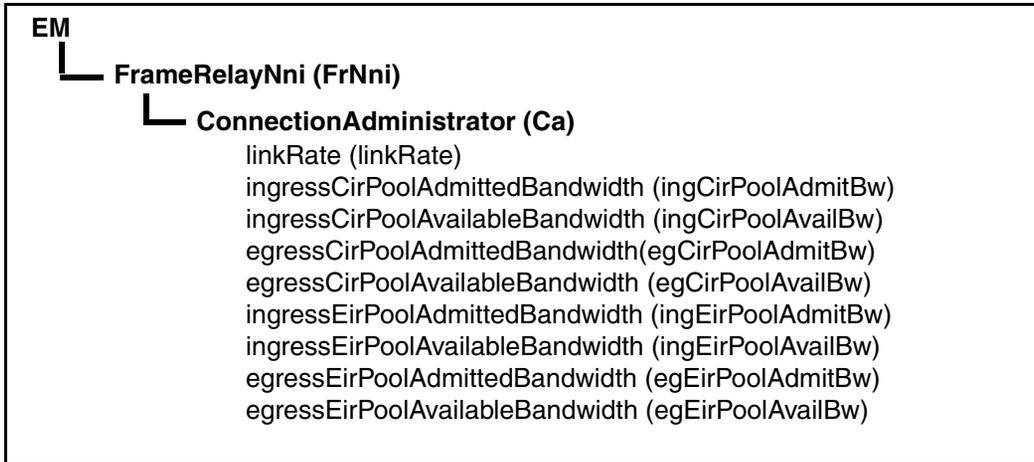
Variable definitions

Variable	Value
<FrNni_inst>	is the instance number of the FR NNI interface.
<FrUni_inst>	is the instance number of the FR UNI interface.

Procedure job aid

Figure 81
Bandwidth pools FR UNI component hierarchy

```
EM
    └─ FrameRelayUni (FrUni)
        └─ ConnectionAdministrator (Ca)
               linkRate (linkRate)
               ingressCirPoolAdmittedBandwidth (ingCirPoolAdmitBw)
               ingressCirPoolAvailableBandwidth (ingCirPoolAvailBw)
               egressCirPoolAdmittedBandwidth(egCirPoolAdmitBw)
               egressCirPoolAvailableBandwidth (egCirPoolAvailBw)
               ingressEirPoolAdmittedBandwidth (ingEirPoolAdmitBw)
               ingressEirPoolAvailableBandwidth (ingEirPoolAvailBw)
               egressEirPoolAdmittedBandwidth (egEirPoolAdmitBw)
               egressEirPoolAvailableBandwidth (egEirPoolAvailBw)
```

Figure 82
Bandwidth pools FR NNI component hierarchy

```
EM
    ├── FrameRelayNni (FrNni)
            └── ConnectionAdministrator (Ca)
                    linkRate (linkRate)
                    ingressCirPoolAdmittedBandwidth (ingCirPoolAdmitBw)
                    ingressCirPoolAvailableBandwidth (ingCirPoolAvailBw)
                    egressCirPoolAdmittedBandwidth(egCirPoolAdmitBw)
                    egressCirPoolAvailableBandwidth (egCirPoolAvailBw)
                    ingressEirPoolAdmittedBandwidth (ingEirPoolAdmitBw)
                    ingressEirPoolAvailableBandwidth (ingEirPoolAvailBw)
                    egressEirPoolAdmittedBandwidth (egEirPoolAdmitBw)
                    egressEirPoolAvailableBandwidth (egEirPoolAvailBw)
```

Chapter 11
Multilink frame relay monitoring

Monitor multilink frame relay (MLFR) to verify that the MLFR service is functioning correctly.

Multilink frame relay monitoring procedures

This task flow show you the procedures you can perform to monitor multilink frame relay. To link to any procedure, go to "Multilink frame relay monitoring procedure navigation" (page 189).

Figure 83
Multilink frame relay monitoring procedures
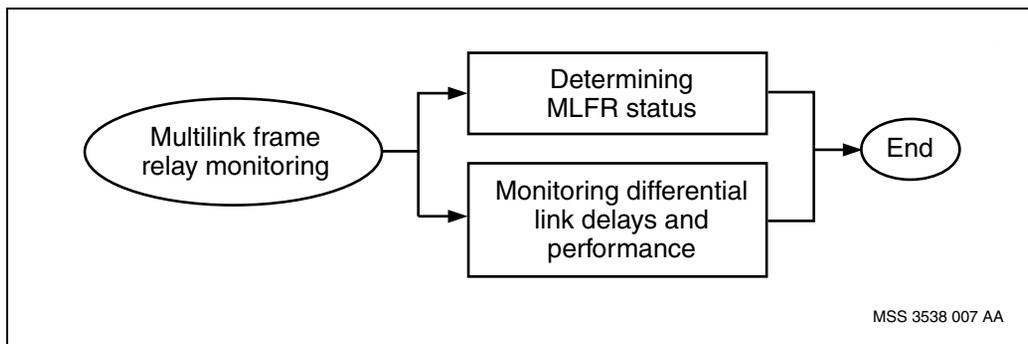
Multilink frame relay monitoring procedure navigation

- "Determining MLFR status" (page 190)

- "Monitoring differential link delays and performance" (page 192)

Determining MLFR status

Determine MLFR status to check the OSI state of the *MultiLinkFrameRelay* component, to determine the name of the remote *MultiLinkFrameRelay* component, and to determine which links are up within a MLFR bundle.

Procedure steps

1   Display the current OSI state of the *MultiLinkFrameRelay* component.

```
display LogicalProcessor/<lp_no> MultiLinkFrameRelay/
<mlfr_no> OsiState
```

When the operational and administrative states of the *Mlfr* component are enabled and unlocked respectively, then the component is in an up state.

2   Display the value of the *remoteName* attribute of the *MultiLinkFrameRelay* component to determine the name of the link received from far end.

```
display LogicalProcessor/<lp_no> MultiLinkFrameRelay/
<mlfr_no> remoteName
```

3   Display the current OSI state of the *Link* subcomponents.

```
display LogicalProcessor/<lp_no> MultiLinkFrameRelay/
<mlfr_no> link/* OsiState
```

When the operational and administrative states of a *Link* subcomponent of a *MultiLinkFrameRelay* component are enabled and unlocked respectively, then the subcomponent is in an up state.

4   If a specific link is not in an up state, examine the error cause values for the error conditions indicated by the *Link* component.

```
display LogicalProcessor/<lp_no> MultiLinkFrameRelay/
<mlfr_no> link/<link_no> errorCause
```

5   If a specific link is not in an up state, examine the remote cause values for the error conditions indicated by the *Link* component.
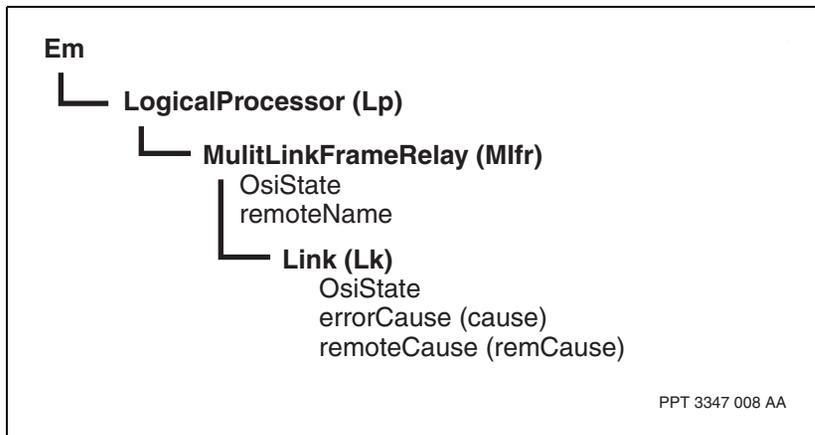
```
display LogicalProcessor/<lp_no> MultiLinkFrameRelay/
<mlfr_no> link/<link_no> remoteCause
```

Variable definitions

Variable	Value
asterisk (*)	is a wildcard character that allows you to display the status of all the *Link* subcomponents.
<lp_no>	is the instance number of the logical processor.
<mlfr_no>	is the instance number of the multilink frame relay component.
<link_no>	is the instance number of a specific *Link* subcomponent.

Procedure job aid

Figure 84
MLFR status component hierarchy


```
Em
   └── LogicalProcessor (Lp)
          └── MulitLinkFrameRelay (Mlfr)
             │ OsiState
             │ remoteName
             └── Link (Lk)
                    OsiState
                    errorCause (cause)
                    remoteCause (remCause)
```

PPT 3347 008 AA

Monitoring differential link delays and performance

Monitor differential link delays and performance to determine the current differential delay in increments of ten milliseconds of a MLFR link and to display various performance monitoring statistics for the links of a MLFR.

Procedure steps

1   Display the current differential delay in milliseconds for all the links in the bundle.

    ```
    display LogicalProcessor/<lp_no> MultiLinkFrameRelay/
    <mlfr_no> Link/* diffDelay
    ```

    A system response of notMeasured indicates an inability to measure delay due to physical layer problems or to no MLFR link integrity protocol (LIP) messages being received on the link.

2   Display performance monitoring statistics for all the links in the bundle.

    ```
    display LogicalProcessor/<lp_no> MultiLinkFrameRelay/
    <mlfr_no> Link/* statistics
    ```
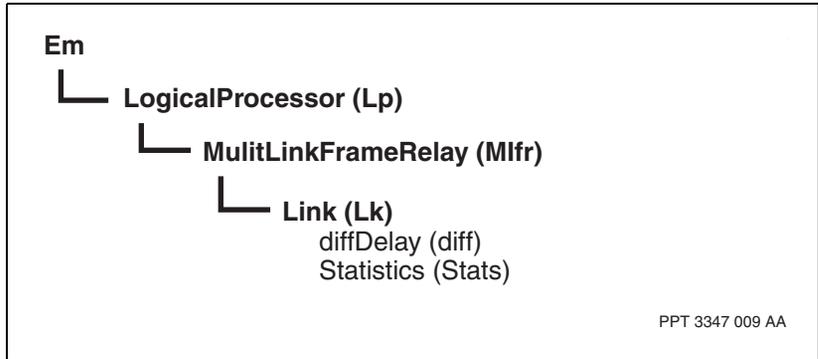
Variable definitions

Variable	Value
asterisk (*)	is a wildcard character that allows you to display the status of all the *Link* components.
<lp_no>	is the instance number of the logical processor.
<mlfr_no>	is the instance number of the multilink frame relay component.

Procedure job aid

Figure 85
Differential link delays and link performance component hierarchy

Em

　　LogicalProcessor (Lp)

　　　　MulitLinkFrameRelay (Mlfr)

　　　　　　Link (Lk)
　　　　　　　diffDelay (diff)
　　　　　　　Statistics (Stats)

PPT 3347 009 AA

Nortel Networks Multiservice Switch 7400/15000/20000
Frame Relay Configuration Management

Release 6.1

**NØRTEL
NETWORKS**