Using Serial Ports to Connect to ATM with DXI Encapsulation

Document ID: 10418

Contents

Introduction

On serial interfaces, we normally change the layer−2 protocol, known as the encapsulation, with a configuration command. On a standard serial interface, the default encapsulation is High−Level Data Link Control (HDLC). We can change this encapsulation with the **encapsulation ppp** or **encapsulation frame−relay** commands. Other examples of layer−2 encapsulations on a serial interface include HDLC, Synchronous Data Link Control (SDLC), and X.25.

In contrast, if we want to connect to an ATM circuit from a telephone company, we cannot simply change the encapsulation on our serial interface to something like **encapsulation atm**. (Note: The only exception is the MC3810's multiflex trunk module, which uses a software−based SAR.) This is because a "native" ATM interface, such as the PA−A3 port adapter for the Cisco 7x00 router series, consists of special hardware and a segmentation and reassembly (SAR) chip for chopping up variable−length IP or other data frames into fixed 53−byte cells. Instead, what we can do is configure the serial interface with the **encapsulation atm−dxi** command. Data exchange interface (DXI) encapsulates your data inside HDLC−like frames and carries these frames to an ATM data service unit (DSU).

In this sample output of the **show interface serial** command, the encapsulation has been set to ATM−DXI:

```
Serial0 is up, line protocol is up
Hardware is MCI Serial
Internet address is 131.108.177.159, subnet mask is 255.255.255.0
MTU 1500 bytes, BW 1544 Kbit, DLY 20000 usec, rely 255/255, load 1/255
Encapsulation ATM−DXI, loopback not set, keepalive not set
Last input 0:00:02, output 0:00:01, output hang never
Last clearing of "show interface" counters never
Output queue 0/40, 0 drops; input queue 0/75, 0 drops
5 minute input rate 0 bits/sec, 0 packets/sec
5 minute output rate 1000 bits/sec, 0 packets/sec
15246 packets input, 14468957 bytes, 0 no buffer
Received 0 broadcasts, 0 runts, 0 giants
0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
15313 packets output, 14445489 bytes, 0 underruns
```

```
     0 output errors, 0 collisions, 4 interface resets, 0 restarts
     1 carrier transitions RTS up, CTS down, DTR up, DSR down
```

This document describes ATM−DXI encapsulation, how to configure it, and how to troubleshoot it.

Prerequisites

Requirements

There are no specific requirements for this document.

Components Used

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

Conventions

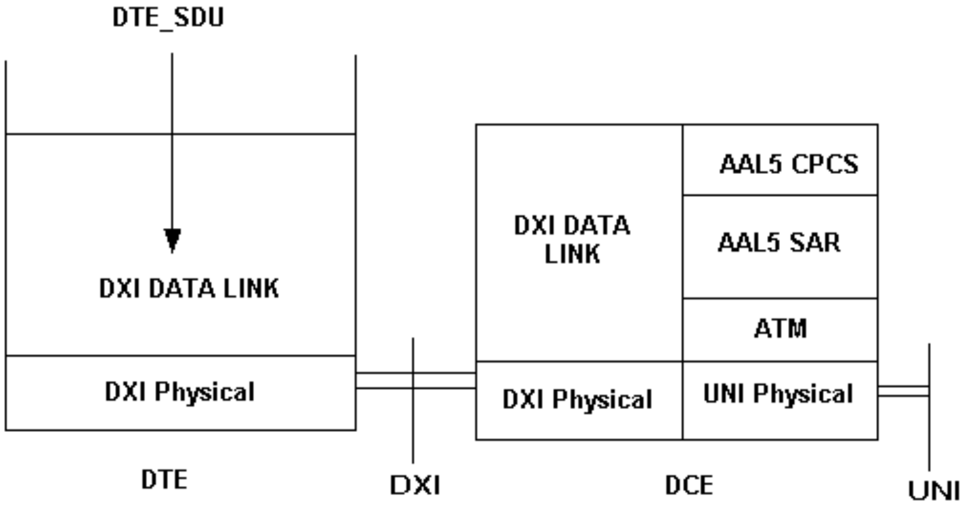
For more information on document conventions, refer to the Cisco Technical Tips Conventions.

Physical Setup

ATM−DXI creates an interface or connection between a data terminal equipment (DTE) and a data circuit−terminating equipment (DCE). In the case of ATM−DXI, the router's serial interface is the DTE, and an ATM data service unit (ADSU) is the DCE. An ADSU is a special DSU that is capable of converting outgoing packets into ATM cells and reassembling incoming ATM cells into packets. Both serial and High−Speed Serial Interfaces (HSSIs) can be configured with ATM−DXI encapsulation.

With ATM−DXI encapsulation, both the router and the ADSU are responsible for processing the packet in some way and adding overhead bytes to the packet. Specifically, transmission to the ATM network uses this process:

1. The router's serial interface prepends a variable−length frame with a DXI frame header and (optionally) an Logical Link Control(LLC)/Subnetwork Access Protocol (SNAP) or network layer protocol identification (NLPID) header, and creates a DXI frame.
2. The serial interface transmits the DXI frame out to the ADSU.
3. The ADSU removes the DXI header and retains any LLC/SNAP or NLPID header.
4. The ADSU performs the ATM−level processing by appending an ATM adaptation layer 5 (AAL5) trailer and then segments the packet into ATM cells.
5. The ADSU analyzes the DXI frame address (DFA) and maps the VPI/VCI contained in the DFA to the virtual path identifier or virtual channel identifier (VPI/VCI) fields in a standard ATM 5−byte cell header.
6. The cells are transmitted onto the ATM network.



The important part about this setup is that an ADSU is required to convert from frames into ATM cells. Manufacturers of standard DSU/CSUs also offer special ADSUs. Contact your telco provider for recommended ADSUs. Kentrox is one manufacturer of ADSUs.

ATM−DXI Modes

ATM−DXI supports three modes, which can differ in these four ways:

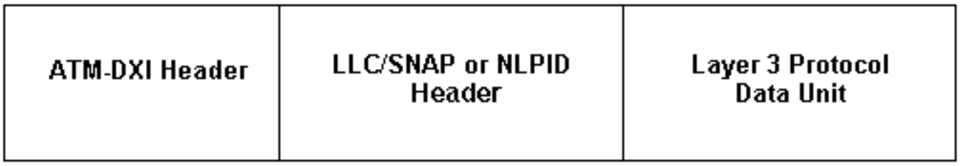
- Number of supported virtual circuits.
- Length of the protocol data unit (PDU) or data frame.
- Supported ATM adaptation layer (AAL) encapsulations.
- 16−bit or 32−bit frame check sequence (FCS).

Cisco uses mode 1a for the DXI header format.

ATM−DXI Headers

Depending on the configuration, ATM−DXI encapsulates your packets inside two headers at layer−2 of the OSI reference model. These two headers are the DXI header and, optionally, an LLC/SNAP or NLPID header. The following sections describe these headers.

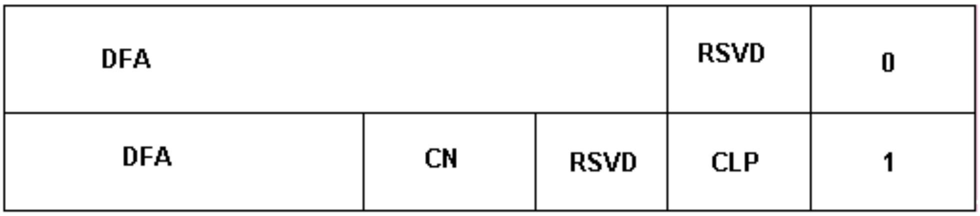
The router's serial interface builds a DXI frame. The complete DXI frame consists of the ATM−DXI header, (optionally) an LLC/SNAP or NLPID header, and the layer−3 protocol data unit.



ATM-DXI Header	LLC/SNAP or NLPID Header	Layer 3 Protocol Data Unit

DXI Header

The router's serial interface creates the DXI frame header, which is two bytes. This header uses this format:



DFA			RSVD	0
DFA	CN	RSVD	CLP	1

The DXI frame address (DFA) field passes the ATM VPI and VCI addressing information to the ADSU. The DFA field typically is ten bits. During transmission out to the ATM network, the ADSU actually removes the DXI header, and maps the VPI/VCI values in the DXI header to the VPI/VCI values in a standard five−byte ATM cell header.

LLC/SNAP, MUX, or NLPID Header

Each ATM−DXI PVC carries one or more layer−3 protocols. RFC 1483 and RFC 1490 define standard ways of encapsulating and transporting multiprotocol traffic over an ATM network. On your serial interface, you must tell the router which method to use with the following command:

```
router(config-if)# dxi pvc vpi vci [snap | nlpid |mux]
```

RFC 1483 defines two transport methods. One method allows multiplexing of multiple protocols over a single PVC. The other method uses different virtual circuits to carry different protocols.

- **mux** The multiplex (MUX) option defines the PVC to carry one protocol only; each protocol must be carried over a different PVC.

  ```
  DXI Header= 0x28A1
  IP Datagram= 0x45000064.....
  ```
- **snap** The SNAP option is LLC/SNAP multiprotocol encapsulation, compatible with RFC1483; SNAP is the current default option. In the following output, the SNAP header has the value 0xAAAA03, which indicates that a SNAP header follows. The Ethertype value of 0x0800 indicates that the DXI frame is carrying an IP packet.

  ```
  DXI Header = 0x28A1
  SNAP Header= 0xAAAA03
  OUI= 0x000000
  Ethertype = 0x0800
  IP Datagram= 0x45000064.....
  ```
- **nlpid** The NLPID option is multiprotocol encapsulation, compatible with RFC 1490; this option is provided for backward compatibility with the default setting in earlier versions in the Cisco IOS® Software.

  ```
  DXI Header= 0x28A1
  Control= 0x03
  NLPID for IP= 0xCC
  IP Datagram= 0x45000064......
  ```

Configuration Steps

Configuring ATM access over a serial interface involves four tasks:

1. Select the serial interface and ensure it is not shutdown. Issue the **no shut** command if necessary.
2. Enable ATM−DXI encapsulation:

   ```
   router(config-if)# encapsulation atm-dxi
   ```

3. Create the ATM−DXI permanent virtual circuit (PVC) by specifying the VPI and VCI. The same PVC values must be configured on the attached device, typically a switch in the provider's ATM network.

   ```
   router(config−if)# dxi pvc vpi vci [snap | nlpid | mux ]
   ```
4. Map the layer−3 protocol addresses to the ATM−DXI PVC's VPI and VCI. The protocol addresses belong to the host at the other end of the link.

   ```
   router(config−if)# dxi map protocol protocol−address vpi vci [broadcast]
   ```

   Repeat this task for each protocol to be carried on the PVC.

Troubleshooting the ATM−DXI Serial Interface

After configuring the serial interface for ATM, you can display the status of the interface, the ATM−DXI PVC, or the ATM−DXI map. To display interface, PVC, or map information, use the following commands in EXEC mode:

- **show interfaces atm** *[slot/port]*
- **show dxi map**
- **show dxi pvc**

```
Router# show dxi map

Serial0 (administratively down): ipx 123.0000.1234.1234
   DFA 69(0x45,0x1050), static, vpi = 4, vci = 5,
   encapsulation: SNAP
Serial0 (administratively down): appletalk 2000.5
   DFA 52(0x34,0xC40), static, vpi = 3, vci = 4,
   encapsulation: NLPID
Serial0 (administratively down): ip 172.21.177.1
   DFA 35(0x23,0x830), static,
   broadcast, vpi = 2, vci = 3,
   encapsulation: VC based MUX,
   Linktype IP
```

Field	**Description**
DFA	DXI Frame Address, similar to a data−link connection identifier (DLCI) for Frame Relay. The DFA is shown in decimal, hexadecimal, and DXI header format. The router computes this address value from the VPI and VCI values.
encapsulation	Encapsulation type selected by the dxi pvc command. Displayed values can be SNAP, NLPID, or VC−based multiplexing device (MUX).
Linktype	Value used only with MUX encapsulation and therefore with only a single network protocol defined for the PVC. Maps configured on a PVC with MUX encapsulation must have the same link type.

```
Router# show dxi pvc
```

```
PVC Statistics for interface Serial0 (ATM DXI)

DFA = 17, VPI = 1, VCI = 1, PVC STATUS = STATIC, INTERFACE = Serial0

  input pkts 0                output pkts 0               in bytes 0
     out bytes 0                 dropped pkts 0

DFA = 34, VPI = 2, VCI = 2, PVC STATUS = STATIC, INTERFACE = Serial0

  input pkts 0                output pkts 0               in bytes 0
     out bytes 0                 dropped pkts 0

DFA = 35, VPI = 2, VCI = 3, PVC STATUS = STATIC, INTERFACE = Serial0

  input pkts 0                output pkts 0               in bytes 0
     out bytes 0                 dropped pkts 0
```

Field	**Description**
DFA	DXI Frame Address, similar to a DLCI for Frame Relay. The DFA is shown in decimal, hexadecimal, and DXI header format. The router computes this address value from the VPI and VCI values.
PVC STATUS = STATIC	Only static maps are supported. Maps are not created dynamically.
input pkts	Number of packets received.
output pkts	Number of packets transmitted.
in bytes	Number of bytes in all packets received.
out bytes	Number of bytes in all packets transmitted.
dropped pkts	Should display a zero (0) value. A nonzero value indicates a configuration problem, specifically that a PVC does not exist.

debug Commands

ATM−DXI encapsulation also supports two **debug** commands. Before issuing **debug** commands, please refer to Important Information on Debug Commands.

- **debug dxi events**
- **debug dxi packet**

Note: The output from the **debug dxi packet** command prints one message per packet. Enabling debugs always should be done very carefully, particularly in a production environment.

Related Information

- **ATM Technology Support**
- **Cisco ATM Port Adapter**
- **Technical Support − Cisco Systems**

Contacts & Feedback | Help | Site Map

Updated: Nov 15, 2007 Document ID: 10418