

Understanding SSCOP Messages on Router ATM Interfaces

Document ID: 10505

Introduction

Prerequisites

- Requirements

- Components Used

- Conventions

Understanding the QSAAL Protocol Stack

What Is SSCOP?

Understanding the SSCOP Trailer

SSCOP Messages or PDUs

SSCOP Timers

SSCOP Sequence Numbers

Sample debug Output

Related Information

Introduction

A protocol generally is defined as the rules of communication between two devices. A signaling protocol defines the rules of communication between two ATM interfaces that are using signaling messages to create on-demand or switched virtual circuits (SVCs) to carry user data. ATM interfaces actually support a signaling protocol stack that includes "user" signaling messages from the Q.2931 User–Network Interface (UNI) protocol and a special signaling ATM adaptation layer (SAAL). The SAAL is composed of the Service–Specific Connection–Oriented Protocol (SSCOP) and the service–specific coordination function (SSCF).

Clearly, ATM signaling introduces many acronyms, which together can make SSCOP seem complicated when it really performs a simple task transport signaling messages across the UNI.

An understanding of SSCOP can be a key troubleshooting tool when investigating the reason for unexpected LAN Emulation (LANE) client state changes. When such changes occur, the router prints the messages below to the log.

Note: The lines of output below appear on multiple lines due to space limitations.

```
Aug 25 18:32:59.973 MEST: %LANE-5-UPDOWN: ATM0.1 elan default:
LE Client changed state to down
Aug 25 18:32:59.981 MEST: %LANE-5-UPDOWN: ATM0.39 elan admin:
LE Client changed state to down
```

This document provides straightforward theory on SSCOP. It uses simple tables to describe SSCOP protocol data units (PDUs), sequence numbers and state variables. It then presents output from the **debug sscop events** command to illustrate how the PDUs, numbers and variables appear on Cisco routers.

Note: The focus of this document is on Cisco routers acting as the user side of a UNI. This document does not discuss Network–to–Network Interface (NNI) signaling.

Prerequisites

Requirements

There are no specific requirements for this document.

Components Used

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

Conventions

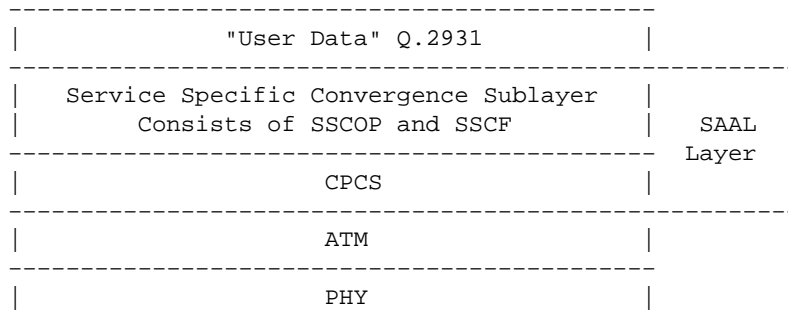
For more information on document conventions, refer to the Cisco Technical Tips Conventions.

Understanding the QSAAL Protocol Stack

ATM is both a protocol and a protocol stack. It is important to consider the illustration below and note how three protocol stacks operate in parallel on an ATM interface supporting signaling and network management. Each protocol stack provides a different function to the successful operation of the interface.

Control Plane		User Plane	Management Plane
Q.2931 UNI Signaling		Voice, video or data	Integrated Local Management Interface (ILMI)
SAAL	SSCF	ATM adaptation layer (AAL)	AAL
	SSCOP		
	Common part convergence sublayer (CPCS)		
ATM Layer			
Physical Layer SONET/Synchronous Digital Hierarchy (SDH), DS3, E3, T1, etc.			

On the user plane, the most common AAL is AAL5, which provides an 8-byte trailer. The SAAL represents a variation of AAL5. What makes it different is a service specific convergence sublayer (SSCS) that consists of SSCOP and SSCF. This diagram illustrates these layers:



ATM interfaces transmit signaling messages "out of band", or outside the bandwidth of the regular data connection. They use a dedicated permanent virtual connection (PVC) configured with a special Q.2931 SAAL (QSAAL) encapsulation type.

Issue the **pvc vpi/vci** command on an ATM router interface to configure the QSAAL PVC.

```
7500-3.4(config)# interface atm 3/0
7500-3.4(config-if)# pvc 0/5 ?
      ilmi   Configure the management PVC for this interface
      qsaal  Configure the signaling PVC for this interface
<cr> 7500-3.4(config-if)# pvc 0/5 qsaal
```

Cisco ATM switches come preconfigured with the QSAAL PVC on each interface. Issue the **show atm vc interface atm** command to confirm this default configuration.

```
ls1010-2# show atm vc interface atm 0/0/2
Interface      VPI   VCI   Type   X-Interface  X-VPI  X-VCI  Encap  Status
ATM0/0/2       0     5     PVC    ATM2/0/0     0      45    QSAAL  UP
ATM0/0/2       0     16    PVC    ATM2/0/0     0      37    ILMI   UP
```

SSCOP is defined in several International Telecommunication Union Telecommunication Standardization Sector (ITU-T) recommendations. The Q.2110 recommendation provides information most relevant to troubleshooting SSCOP-related issues on ATM router interfaces.

- Q.2100 Defines the structure of SAAL.
- Q.2110 Defines SSCOP as a protocol entity.
- Q.2130 Defines the SSCF for UNI interfaces.
- Q.2140 Defines the SSCF for NNI interfaces.
- I.363 Defines the CPCS.

Note: UNI and NNI interfaces use different versions of SSCF. NNI is not discussed in this document.

What Is SSCOP?

SSCOP is a transport protocol that provides guaranteed, in-sequence delivery of messages to the signaling protocols that reside above it in the signaling protocol stack. SSCOP also performs flow control, error reporting to the management plane, and a keepalive function.

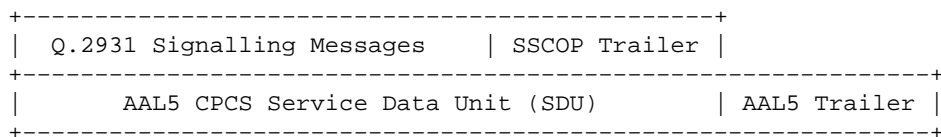
This table describes the many important functions that SSCOP provides to ATM interfaces:

Function	Description
In-sequence and reliable delivery of signaling messages	Signaling messages generated by the UNI Q.2931 protocol constitute the "user data" within the signaling stack. SSCOP preserves the order of these messages through sequence numbers and selective retransmission. Note that SSCOP does not check the contents of the signaling messages themselves.
Flow control	Sets limits on the rate at which the peer ATM interface sends SSCOP messages.
Error reporting	Detects and reports errors in the operation of SSCOP itself.
Keepalive	

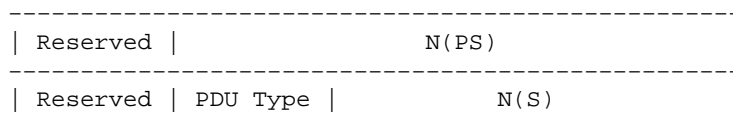
	Exchanges POLL messages at a regular interval to ensure that both ends and the connection itself remain operational and active, particularly during a period when no signaling messages are transmitted.
Local data retrieval	Maintains statistics (viewable using the show sscop command) on signaling messages not yet "released" or acknowledged by the peer ATM interface.
Status reporting	Provides messages that communicate status information, including information to the management plane.

Understanding the SSCOP Trailer

ATM UNI interfaces use Q.2931 as the signaling protocol. SSCOP pads the Q.2931 messages to a multiple of 4 bytes and appends a trailer of SSCOP-specific information that is always a multiple of 4 bytes.



The contents of the SSCOP trailer vary with the type of PDU, which is described in the next section, SSCOP Messages or PDUs. This diagram shows the format of the SSCOP trailer for a POLL PDU:



SSCOP Messages or PDUs

SSCOP uses 15 message types or PDUs to perform its many functions. The **show sscop** command provides statistics on the number of each PDU sent and received. In this sample output, the ATM interface 3/0 has sent and received 11 PDUs, including 8 POLL PDUs and 1 BEGIN PDU:

```

7500# show sscop atm 3/0
SSCOP details for interface ATM3/0
Current State = Active, Uni version = 4.0
[output omitted]
Statistics -
Pdu's Sent = 11, Pdu's Received = 11, Pdu's Ignored = 0
Begin = 1/1, Begin Ack = 1/1, Begin Reject = 0/0
End = 1/0, End Ack = 0/1
Resync = 0/0, Resync Ack = 0/0
Sequenced Data = 0/0, Sequenced Poll Data = 0/0
Poll = 8/8, Stat = 8/8, Unsolicited Stat = 0/0
Unassured Data = 0/0, Mgmt Data = 0/0, Unknown Pdu's = 0
Error Recovery/Ack = 0/0, lack of credit 0

```

This table groups the SSCOP messages based on the function:

Function	Message Abbreviation	Message Name	Description
Connection establishment	BGN	Begin	Starts the SSCOP connection process between two ATM interfaces. Initializes the peer buffers and the transmit and receive counters.
	BGAK	Begin Acknowledgment	Acknowledges the peer connection request.
	BGREJ	Begin Reject	Rejects the peer connection request. The peer retransmits the BGN PDU and continues to initiate a connection.
Connection teardown	END	End	Releases the connection between two peer ATM devices.
	ENDAK	End Acknowledgment	Confirms the release request.
Resynchronization	RS	Resynchronization	Resynchronizes message buffers as well as the transmitter and receiver state variables or counters.
	RS AK	Resynchronization Acknowledgment	Acknowledges the resynchronization request.
Error recovery	ER	Error Recovery	Recovers from errors that occur during an active connection.
	ER AK	Error Recovery Acknowledgment	Acknowledges the error recovery request.
	SD	Sequenced Data	Transfers "user" messages from the UNI Q.2931 signaling protocol

			to the peer.
	POLL	Status Request	Requests status information about the peer.
	STAT	Solicited Status Response	Represents a response to a POLL PDU. Provides information about the successful receipt of SD PDUs, the sequence number of the last POLL PDU. It also contains a credit value that indicates how many more messages the peer can or cannot send before acknowledgment.
	USTAT	Unsolicited Status Response	Communicates lost or missing PDUs that have been detected by analyzing the sequence numbers in other PDUs.
Unassured data transfer	UD	Unnumbered Data	Transmits "user" messages between the peers. Does not include a sequence number and can be lost without notification.
Management data transfer	MD	Management Data	Transmits management information to the management plane. Does not include a sequence number and can be lost without notification.

Note: The ITU-T Q.2110 recommendation defines an invalid PDU as a PDU that has an unknown PDU type code, is not 32-bit aligned, or is not the proper length for a PDU of the stated type.

SSCOP Timers

SSCOP follows a state machine, in which the protocol itself moves through several states before becoming active. A set of five timers controls (in part) when SSCOP transitions to another state. Issue the **sscop** command in interface configuration mode to view these timers.

```
7200(config-if)# sscop ?
cc-timer          timer (in secs) to send BGN/END/RS/ER pdu at the
                  connection control phase
idle-timer        timer (in secs) to send poll pdu at the idle phase
keepalive-timer   timer (in secs) to send poll pdu at the transient
                  phase
noResponse-timer timer (in secs) at lease one STAT PDU needs to be
                  received
poll-timer        timer (in msecs) to send poll pdu at the active
                  phase
```

This table describes the five SSCOP timers:

Timer	Description	Default Value
cc-timer	Connection control (cc) is the set of processes used to establish, release, or resynchronize an SSCOP connection between two ATM interfaces. The cc timer sets the time between retransmissions of BGN, END, or RS PDUs while waiting for an acknowledgment. The max-cc value sets the number of retries.	1 second (sec)
idle-timer	If the connection is stable enough and there are no data messages to transmit and no outstanding acknowledgments, SSCOP switches from timer keepalive to timer idle.	10 sec
keepalive-timer	Controls the maximum time between transmission of a POLL PDU when no SD PDUs are queued for transmission or are outstanding pending acknowledgment.	5 sec

noResponse-timer	Runs in parallel with two other timers poll and keepalive. Sets the maximum time interval during which at least one STAT message must be received in response to a POLL. If this timer expires, the connection is taken down.	45 secs
poll-timer	Sets the maximum time between transmitting a POLL PDU when SD PDUs are queued for transmission or are outstanding pending acknowledgment.	1000 milliseconds (msecs)

Issue the **show sscop atm** command to view the default values of the SSCOP timers.

```
7500# show sscop atm 3/0
SSCOP details for interface ATM3/0
  Current State = Idle,   Uni version = 4.0
  Send Sequence Number: Current = 0,   Maximum = 30
  Send Sequence Number Acked = 0
  Rcv Sequence Number: Lower Edge = 0, Upper Edge = 0, Max = 30
  Poll Sequence Number = 0, Poll Ack Sequence Number = 1
  Vt(Pd) = 0   Vt(Sq) = 0
  Timer_IDLE = 10 - Inactive
  Timer_CC = 1 - Inactive
  Timer_POLL = 1000 - Inactive
  Timer_KEEPA L I V E = 5 - Inactive
  Timer_NO-RESPONSE = 45 - Inactive
  Current Retry Count = 0, Maximum Retry Count = 10
```

!--- Output suppressed.

SSCOP Sequence Numbers

The SSCOP process on an ATM interface tracks two sets of sequence numbers or state variables, and then maps these values into fields in the actual PDUs. Specifically, SD PDUs and POLL PDUs are sequentially and independently numbered. The transmitter and the receiver maintain the sequence numbers as state variables. These variables then map into actual parameters or fields in the SSCOP PDUs. The **show sscop** command displays the current values of the sequence numbers.

```
ATM# show sscop
SSCOP details for interface ATM0
  Current State = Active,   Uni version = 3.1
  Send Sequence Number: Current = 79,   Maximum = 109
  Send Sequence Number Acked = 79
  Rcv Sequence Number: Lower Edge = 93, Upper Edge = 93, Max = 123
  Poll Sequence Number = 32597, Poll Ack Sequence Number = 32597
  Vt(Pd) = 0   Vt(Sq) = 1
  Timer_IDLE = 10 - Active
```

!--- Output suppressed.

The following sections describe the state variables and the actual PDU numbers.

State Variables at the Transmitter

An ATM interface keeps a set of transmit-side state variables that begin with VT.

State Variable	Name	Description
VT(S)	Send	Sequence number that increments with each <i>SD</i> PDU. Does not increment when the same <i>SD</i> PDU is retransmitted.
VT(PS)	Poll Send	Sequence number that increments with each <i>POLL</i> PDU.
VT(A)	Acknowledge	Sequence number of the <i>SD</i> PDU that is expected to be acknowledged next. Increments each time that an <i>SD</i> PDU is acknowledged.
VT(PA)	Poll Acknowledge	Sequence number of the <i>STAT</i> PDU that is expected to receive next as an acknowledgment to the <i>POLL</i> PDU.
VT(MS)	Maximum Send	Highest sequence number of a PDU that the transmitting interface can send (and the receiver will accept) without receipt of one of the following PDUs: <i>USTAT</i> , <i>STAT</i> , <i>BGN</i> , <i>BGAK</i> , <i>RS</i> , <i>RSACK</i> , <i>ER</i> , or <i>ERACK</i> PDU. In other words, <i>VT(MS)</i> defines the transmit window size. <i>VT(S)</i> should not be higher than <i>VT(MS)</i> .
VT(PD)	Poll Data	Number of <i>SD</i> PDUs transmitted between two <i>POLL</i> PDUs. Increments upon transmission of an <i>SD</i> PDU and resets to zero upon transmission of a <i>POLL</i> PDU.
VT(CC)	Connection Control	Number of unacknowledged <i>BGN</i> , <i>END</i> , <i>ER</i> , or <i>RS</i> PDUs. If the ATM interface sends an <i>END</i> PDU in response to a protocol error, <i>SSCOP</i> moves directly to the idle state and does not increment the <i>VT(CC)</i> value.
VT(SQ)	Transmitter Connection Sequence	Identifies retransmitted <i>BGN</i> , <i>ER</i> , and <i>RS</i> PDUs. Is initialized to zero when the <i>SSCOP</i> process starts and then mapped into <i>N(SQ)</i> .

State Variables at the Receiver

An ATM interface keeps a set of receive-side state variables that begin with VR.

State Variable	Name	Description
VR(R)	Receive	Sequence number of the next in-sequence SD PDU that the receiver expects. It is incremented when that message is seen.
VR(H)	Highest Expected	Highest expected sequence number in an SD PDU. Updated from the next SD or POLL message and should roughly be equal to the peer VT(S).
VR(MR)	Maximum Receive	Highest sequence number in an SD PDU that the receiver will accept. In other words, the receiver will allow up to $VR(MR) - 1$, and then it discards any SD PDUs with a higher sequence number. Updating VR(MR) is implementation-dependent.
VR(SQ)	Receiver Connection Sequence	Used to identify retransmitted BGN, ER, and RS PDUs. When an ATM interface receives one of these PDUs, it compares the N(SQ) value with its own VR(SQ) value. If the two values are different, the PDU is processed as a new message. If the two values are equal, the PDU is identified as a retransmission.

State Variables Translated Into PDU Parameters

Receive and transmit state variables are translated or mapped into actual PDU parameters with slightly different names. This table shows the PDU parameters and the state variable from which they are derived:

Parameter	Mapped From	Description
N(SQ)	VR(SQ)	Connection sequence number carried in a BGN, RS, or ER PDU. Used with the VR(SQ) counter at the receiver to identify any retransmissions of these PDUs.
N(S)	VT(S)	Send sequence number carried in each SD or POLL PDU and incremented with each new, non-retransmitted PDU.
N(PS)	VT(PS)	Carried in a POLL PDU and matching STAT PDU to correlate the two messages together.
N(R)	VR(R)	Receive sequence number carried in a

		STAT or USTAT PDU. Sent by the peer device when acknowledging receipt of one or more signaling messages.
N(MR)	VR(MR)	Carried in the following PDUs: STAT, USTAT, RS, RSAK, ER, ERAK, BGN, BGAK. Indicates the number of remaining receive credits and whether the peer can send another message. For example, an N(MR) value of 5 means that the peer can send up to 5 PDUs without waiting for a response.

Sample debug Output

The output below was generated by issuing the **debug sscop event atm 3/0** command on a 7500 series router with a PA-A3. The comments in *blue* are used to interpret the **debug** output.

```
*Mar 21 03:18:43.440: SSCOP(ATM3/0): i Begin pdu, Idle state, length = 8
*Mar 21 03:18:43.440: SSCOP(ATM3/0): Rcv Begin in Idle State
*Mar 21 03:18:43.440: SSCOP(ATM3/0): receive window in Begin Pdu = 30
*Mar 21 03:18:43.440: SSCOP(ATM3/0): o Begin Ack pdu, Idle state, rcv window v(mr) = 30

!--- A BEGIN PDU is received by the router, which responds with a BEGIN ACK PDU.
!--- The window size V(MR) is initialized to 30.

*Mar 21 03:18:43.440: SSCOP(ATM3/0): state changed from Idle to Active
*Mar 21 03:18:47.968: SSCOP(ATM3/0): o Poll pdu, state = Active, n(s) = 0, n(ps) = 1
*Mar 21 03:18:47.968: SSCOP(ATM3/0): i Stat pdu, Active state, length = 12
*Mar 21 03:18:47.968: SSCOP(ATM3/0): Rcv Stat in Active State
*Mar 21 03:18:47.968: SSCOP(ATM3/0): processStatPdu: ps 1, nmr 30, nr 0
*Mar 21 03:18:47.968: SSCOP(ATM3/0): processStatPdu: vtPa 1, vps 1
*Mar 21 03:18:47.968: SSCOP(ATM3/0): processStatPdu: vta 0, vts 0
*Mar 21 03:18:47.968: SSCOP(ATM3/0): processStatPdu: listCount = 0 - normal

!--- This is the first outbound POLL PDU and inbound STAT PDU.

*Mar 21 03:18:48.040: SSCOP(ATM3/0): * Poll pdu, ns = 0, nps = 1
*Mar 21 03:18:48.040: SSCOP(ATM3/0): o Stat pdu, n(r) = 0, n(mr) = 30, n(ps) = 1

!--- The "*" indicates an inbound POLL PDU from the attached ATM switch.
!--- The router responds with an outbound STAT PDU.

*Mar 21 03:18:57.292: SSCOP(ATM3/0): o Poll pdu, state = Active, n(s) = 0, n(ps) = 2
*Mar 21 03:18:57.292: SSCOP(ATM3/0): i Stat pdu, Active state, length = 12
*Mar 21 03:18:57.292: SSCOP(ATM3/0): Rcv Stat in Active State
*Mar 21 03:18:57.292: SSCOP(ATM3/0): processStatPdu: ps 2, nmr 30, nr 0
*Mar 21 03:18:57.292: SSCOP(ATM3/0): processStatPdu: vtPa 1, vps 2
*Mar 21 03:18:57.292: SSCOP(ATM3/0): processStatPdu: vta 0, vts 0
*Mar 21 03:18:57.292: SSCOP(ATM3/0): processStatPdu: listCount = 0 - normal

!--- This is the second outbound POLL PDU and inbound STAT PDU. N(PS) and V(PS)
!--- increment to 2.

*Mar 21 03:18:58.004: SSCOP(ATM3/0): * Poll pdu, ns = 0, nps = 2
*Mar 21 03:18:58.004: SSCOP(ATM3/0): o Stat pdu, n(r) = 0, n(mr) = 30, n(ps) = 2
*Mar 21 03:19:06.812: SSCOP(ATM3/0): o Poll pdu, state = Active, n(s) = 0, n(ps) = 3
*Mar 21 03:19:06.812: SSCOP(ATM3/0): i Stat pdu, Active state, length = 12
*Mar 21 03:19:06.812: SSCOP(ATM3/0): Rcv Stat in Active State
*Mar 21 03:19:06.812: SSCOP(ATM3/0): processStatPdu: ps 3, nmr 30, nr 0
*Mar 21 03:19:06.812: SSCOP(ATM3/0): processStatPdu: vtPa 2, vps 3
*Mar 21 03:19:06.812: SSCOP(ATM3/0): processStatPdu: vta 0, vts 0
```

```
*Mar 21 03:19:06.812: SSCOP(ATM3/0): processStatPdu: listCount = 0 - normal
*Mar 21 03:19:07.228: SSCOP(ATM3/0): * Poll pdu, ns = 0, nps = 3
*Mar 21 03:19:07.228: SSCOP(ATM3/0): o Stat pdu, n(r) = 0, n(mr) = 30, n(ps) = 3
```

```
!--- This is the third outbound POLL PDU and inbound STAT PDU. N(PS) and V(PS)
!--- increment to 3. N(MR) remains at 30. N(S), VT(S), and VT(A) remain at 0 since
!--- no sequenced Q.2931 "user" data is being transmitted.
```

The **debug** output captures SSCOP messages sent during connection establishment and as part of the keepalive mechanism. A simultaneous capture of the **show sscop atm** command while the **debug** commands were running shows incrementing values for Pdu's Sent and Pdu's Received, as well as for Poll and Stat.

```
7500# show sscop atm 3/0
SSCOP details for interface ATM3/0
  Current State = Active,   Uni version = 4.0
  Send Sequence Number: Current = 0,   Maximum = 30
  Send Sequence Number Acked = 0
  Rcv Sequence Number: Lower Edge = 0, Upper Edge = 0, Max = 30
  Poll Sequence Number = 6, Poll Ack Sequence Number = 6
  Vt(Pd) = 0   Vt(Sq) = 1
  Timer_IDLE = 10 - Active
  Timer_CC = 1 - Inactive
  Timer_POLL = 1000 - Inactive
  Timer_KEEPAALIVE = 5 - Inactive
  Timer_NO-RESPONSE = 45 - Inactive
  Current Retry Count = 0, Maximum Retry Count = 10
  AckQ count = 0, RcvQ count = 0, TxQ count = 0
  AckQ HWM = 0, RcvQ HWM = 0, TxQ HWM = 0
  Local connections currently pending = 0
  Max local connections allowed pending = 0
  Statistics -
    Pdu's Sent = 9, Pdu's Received = 9, Pdu's Ignored = 0
    Begin = 1/1, Begin Ack = 1/1, Begin Reject = 0/0
    End = 1/0, End Ack = 0/1
    Resync = 0/0, Resync Ack = 0/0
    Sequenced Data = 0/0, Sequenced Poll Data = 0/0
    Poll = 6/6, Stat = 6/6, Unsolicited Stat = 0/0
    Unassured Data = 0/0, Mgmt Data = 0/0, Unknown Pdu's = 0
    Error Recovery/Ack = 0/0, lack of credit 0
```

```
7500# show sscop atm 3/0
SSCOP details for interface ATM3/0
  Current State = Active,   Uni version = 4.0
  Send Sequence Number: Current = 0,   Maximum = 30
  Send Sequence Number Acked = 0
  Rcv Sequence Number: Lower Edge = 0, Upper Edge = 0, Max = 30
  Poll Sequence Number = 7, Poll Ack Sequence Number = 7
  Vt(Pd) = 0   Vt(Sq) = 1
  Timer_IDLE = 10 - Active
  Timer_CC = 1 - Inactive
  Timer_POLL = 1000 - Inactive
  Timer_KEEPAALIVE = 5 - Inactive
  Timer_NO-RESPONSE = 45 - Inactive
  Current Retry Count = 0, Maximum Retry Count = 10
  AckQ count = 0, RcvQ count = 0, TxQ count = 0
  AckQ HWM = 0, RcvQ HWM = 0, TxQ HWM = 0
  Local connections currently pending = 0
  Max local connections allowed pending = 0
  Statistics -
    Pdu's Sent = 10, Pdu's Received = 10, Pdu's Ignored = 0
    Begin = 1/1, Begin Ack = 1/1, Begin Reject = 0/0
    End = 1/0, End Ack = 0/1
    Resync = 0/0, Resync Ack = 0/0
```

Sequenced Data = 0/0, Sequenced Poll Data = 0/0
Poll = 7/7, Stat = 7/7, Unsolicited Stat = 0/0
Unassured Data = 0/0, Mgmt Data = 0/0, Unknown Pdu's = 0
Error Recovery/Ack = 0/0, lack of credit 0

Related Information

- **ITU-T User-Network Interface (UNI) Specification**
 - **ATM Forum UNI Specifications**
 - **ATM Technology Support Pages**
 - **Technical Support – Cisco Systems**
-

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

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

Updated: Nov 15, 2007

Document ID: 10505
