Internet Draft
Document: draft-strahm-ibconn-mib-00.txt

B. Strahm
Sanera Systems Inc.

Expires: April 2002

Oct 2001

Infiniband Connection MIB

1. Status of this Memo

This document is an Internet-Draft and is in full conformance with all provisions of Section 10 of RFC2026.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

The list of current Internet-Drafts can be accessed at
http://www.ietf.org/ietf/1id-abstracts.txt
The list of Internet-Draft Shadow Directories can be accessed at
http://www.ietf.org/shadow.html.

2. Abstract

This memo defines a portion of the Management Information Base (MIB) for use with network management protocols in the Internet community. In particular, it defines objects for managing the InfiniBand Connection Manager (http://www.infinibandta.org)

Table of Contents

3. The SNMP Management Framework

The SNMP Management Framework presently consists of five major components:

- An overall architecture, described in RFC 2571[RFC2571].
- Mechanisms for describing and naming objects and events for the purpose of management. The first version of this Structure of Management Information (SMI) is called SMIv1 and described in STD 16, RFC 1155[RFC1155], STD 16, RFC 1212 [RFC1212] and RFC 1215 [RFC1215]. The second version, called SMIv2, is described in STD 58, RFC 2578[RFC2578], STD 58, RFC 2579[RFC2579], and STD 58, RFC 2580[RFC2580].
- Message protocols for transferring management information. The first version of the SNMP message protocol is called SNMPv1 and described in STD 15, RFC 1157[RFC1157]. A second version of the SNMP message protocol, which is not an Internet standards track protocol, is called SNMPv2c and described in RFC 1901[RFC1901] and RFC 1906[RFC1906]. The third version of the message protocol is called SNMPv3 and described in RFC 1906[RFC1906], RFC 2572[RFC2572] and RFC 2574[RFC2574].
- Protocol operations for accessing management information. The first set of protocol operations and associated PDU formats is described in STD 15, RFC 1157[RFC1157]. A second set of protocol operations and associated PDU formats is described in RFC 1905[RFC1905].
- A set of fundamental applications described in RFC 2573[RFC2573] and the view-based access control mechanism described in RFC 2575[RFC2575].

A more detailed introduction to the current SNMP Management Framework can be found in RFC 2570[RFC2570].

Managed objects are accessed via a virtual information store, termed the Management Information Base or MIB. Objects in the MIB are defined using the mechanisms defined in the SMI.

This memo specifies a MIB module that is compliant to the SMIv2. A MIB conforming to the SMIv1 can be produced through the appropriate translations. The resulting translated MIB must be semantically equivalent, except where objects or events are omitted because no translation is possible (use of Counter64). Some machine readable information in SMIv2 will be converted into textual descriptions in SMIv1 during the translation process. However, this loss of machine readable information is not considered to change the semantics of the MIB.

4. Conventions used in this document

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [RFC2119].

5. Overview

The Infiniband Architecture[IBTAArch] is defined by the Infiniband Trade Association. Infiniband is designed to provide low latency high bandwidth interconnect in a computing environment. This document will define the objects related to the connection manager, managing the transports above layer 3 which are defined in the Infiniband MIB[IBIFMIB]

The Infiniband Architecture defines five transports above Infiniband: Raw, Unreliable Connected, Reliable Connected, Unreliable Datagram, and Reliable Datagram.

Raw packets are not normal usage and therefore will not be covered in this document. Infiniband goes so far as to specify hardware that has the ability to discard all Raw packets in hardware on reception.

The datagram tables can not be combined because the Unreliable Datagram service provides a simple listening service much like UDP, and the Reliable Connected service uses the concept of an End-End context to allow multiple Queue pairs behind the EEC to connect over a single reliability context saving resources on the Channel Adapter.

The connection tables are combined because the only differences

between the data is that a reliability layer is provided in hardware for the Reliable Connected service. This is handled by the single ibConnCReliable object.

5.1. Unreliable Datagram Table

The unreliable datagram service in Infiniband is not required to go through the Connection Manager, and therefore no state is kept. For this reason the MIB will just track what Queue Pair Numbers the channel adapter is listening to.

5.2. Reliable Datagram Table

The reliable datagram service in Infiniband uses the Connection Management service to connect two endpoints through an End to End Context (EEC). This EEC can be used to connect multiple Queue Pairs, from multiple applications on each machine to the other side without having to maintain state on a separate EEC for each Queue pair.

5.3. Connection Table

The connection table is used to display connections between end applications. Each connection includes data indicating if the connection is reliable or unreliable, and what state the connection is in.

6. Next Steps

Currently the following questions need to be dealt with

1. What should be done with RAW IB packets û currently nothing
2. A compliance section needs to be added
3. What interface(s) might be used by an implementation to get this information from the Connection Manager
4. Where do the Unreliable Datagram values come from
5. A ibifMIB value needs to be allocated by the IpoverIB WG.
6. Use textual conventions where possible from the IB IF MIB document

7. Definitions

```
IBCONNECTION-MIB DEFINITIONS ::= BEGIN
IMPORTS  -- Do we use all of these ???
       Counter32,
```

```
       Counter64,
       Integer32,
       Gauge32,
       MODULE-IDENTITY,
       OBJECT-TYPE
             FROM SNMPv2-SMI
       TEXTUAL-CONVENTION,
       TruthValue
             FROM SNMPv2-TC
       OBJECT-GROUP,
       MODULE-COMPLIANCE
             FROM SNMPv2-CONF
       InterfaceIndex
             FROM IF-MIB
       IbIfMIB
             FROM IB-IF-MIB;

ibConnMIB MODULE-IDENTITY
    LAST-UPDATED "200110200000Z"  -- 20 Oct 2001
    ORGANIZATION
      "IETF IP over IB Working Group"
    Email:   "ipoverib@ietf.org"
    CONTACT-INFO
            "Bill Strahm
    Postal: Sanera Systems Inc.
            1925 NW Amberglen Parkway
            Suite 155
            Beaverton, OR   97006
            United States
    Tel:    +1 503 601 0263
    Email:  bill@sanera.net"

    DESCRIPTION
       "The managed objects for an Infiniband Channel Adapter.  This
       MIB provides for management of the Infiniband transports in
       three tables:
       1) Unreliable Datagram
       2) Reliable Datagram
       3) Connected Services"

    REVISION    "200110200000Z" -û 20 Oct 2001
    DESCRIPTION "Initial version of this MIB."

::={ibIFMIB xxx} û- To be determined by the IPoIB WG


Strahm         Internet Draft- Expires April 2002                6
                   Infiniband Connection MIB             Oct 2001


--
```

```
-- Textual Conventions
--

IBCommunicationID ::= TEXTUAL-CONVENTION
    STATUS    current
    DESCRIPTION
       "The Communication ID for a given connection.  Each end of a
       connection is responsible for picking a unique value for this
       field."
    SYNTAX    Integer32

IBQuePairNumber ::= TEXTUAL-CONVENTION
    STATUS    current
    DESCRIPTION
       "The Queue Pair Number of an IB communication endpoint.
       Queue Pairs are 24 bit numbers.  To allow this value to be an
       index, one is added to the value that is sent on the wire"
    SYNTAX    INTEGER (1..16777216)

IBEtoEContext ::= TEXTUAL-CONVENTION
    STATUS    current
    DESCRIPTION
       "The End û End Context Number of an IB communiction endpoint.
       The EECN is a 24 bit number.  To allow this value to be an
       index, one is added to the value that is sent on the wire"
    SYNTAX    INTEGER (1..16777216)

--
-- Objects
--
ibConnMIBObjects OBJECT IDENTIFIER ::= { ibCAMIB 1 }
ibCAConformance  OBJECT IDENTIFIER ::= { ibCAMIB 2 }

--
-- Unreliable Datagram
--
ibConnUD OBJECT IDENTIFIER ::= { ibConnMibObjects 1 }

ibConnUDInDatagrams  OBJECT-TYPE
SYNTAX              Counter64
MAX-ACCESS          read-only
STATUS              current
DESCRIPTION
   "Total number of Unreliable Datagrams delivered."
::= { ibConnUD 1 }
```



```
ibConnUDNoQP         OBJECT-TYPE
```

```
SYNTAX              Counter32
MAX-ACCESS          read-only
STATUS              current
DESCRIPTION
   "Total number of received Unreliable Datagrams discarded because
   the QP wasnÆt allocated."
::= { ibConnUD 2 }

ibConnUDInErrors     OBJECT-TYPE
SYNTAX               Counter32
MAX-ACCESS           read-only
STATUS               current
DESCRIPTION
   "Number of received Unreliable Datagrams that could not be
   delivered for reasons other than the lack of an allocated QP."
::= { ibConnUD 3 }

ibConnUDOutDatagrams OBJECT-TYPE
SYNTAX               Counter64
MAX-ACCESS           read-only
STATUS               current
DESCRIPTION
   "Total number of Unreliable Datagrams sent from this entity."
::= { ibConnUD 4 }

ibConnUDTable OBJECT-TYPE
SYNTAX        SEQUENCE OF IbConnUDEntry
MAX-ACCESS    not-accessible
STATUS        current
DESCRIPTION
   "A table containing information about all of the Unreliable
   Datagram Connections in the device."
::= { ibConnUD 5 }

ibConnUDEntry OBJECT-TYPE
SYNTAX        IbConnUDEntry
MAX-ACCESS    not-accessible
STATUS        current
DESCRIPTION
   "A conceptual row of the containing information about a
   Unreliable connection entry."
INDEX { ibConnUDGID,
        ibConnUDLID,
        ibConnUDLocalQPN}
::= { ibConnUDTable 1 }

IbConnUDEntry ::= SEQUENCE {
        ibConnUDGID       GidTC,
        ibConnUDLID      LidTC,
        ibConnUDLocalQPN  IBQuePairNumber
        }
```

```
ibConnUDGID         OBJECT-TYPE
SYNTAX              GidTC
MAX-ACCESS          read-only
STATUS              current
DESCRIPTION
   "GID of listener."
::= { ibConnUDEntry 1 }


ibConnUDLID         OBJECT-TYPE
SYNTAX              LidTC
MAX-ACCESS          read-only
STATUS              current
DESCRIPTION
   "LID of listener."
::= { ibConnUDEntry 2 }

ibConnUDLocalQPN    OBJECT-TYPE
SYNTAX              IBQuePairNumber
MAX-ACCESS          read-only
STATUS               current
DESCRIPTION
   "Local Que Pair Number of listener."
::= { ibConnUDEntry 3 }

--
-- Reliable Datagram
--

ibConnRD OBJECT IDENTIFIER ::= { ibConnMibObjects 2 }

ibConnRDInDatagrams  OBJECT-TYPE
SYNTAX              Counter64
MAX-ACCESS          read-only
STATUS              current
DESCRIPTION
   "Total number of Reliable Datagrams delivered."
::= { ibConnRD 1 }

ibConnRDNoQP         OBJECT-TYPE
SYNTAX              Counter32
MAX-ACCESS          read-only
STATUS              current
DESCRIPTION
   "Total number of received Reliable Datagrams discarded because
   the QP wasnÆt allocated."
```

```
::= { ibConnRD 2 }
```



```
ibConnRDInErrors     OBJECT-TYPE
SYNTAX               Counter32
MAX-ACCESS           read-only
STATUS               current
DESCRIPTION
   "Number of received Reliable Datagrams that could not be
   delivered for reasons other than the lack of an allocated QP."
::= { ibConnRD 3 }

ibConnRDOutDatagrams OBJECT-TYPE
SYNTAX               Counter64
MAX-ACCESS           read-only
STATUS               current
DESCRIPTION
   "Total number of Reliable Datagrams sent from this entity."
::= { ibConnRD 4 }


ibConnRDTable OBJECT-TYPE
SYNTAX      SEQUENCE OF IbConnRDEntry
MAX-ACCESS  not-accessible
STATUS      current
DESCRIPTION
   "A table containing information about all of the Reliable
   Datagram connections in the device."
::= { ibConnRD 5 }

ibConnRDEntry       OBJECT-TYPE
SYNTAX               IbConnRDEntry
MAX-ACCESS           not-accessible
STATUS               current
DESCRIPTION
   "A conceptual row of the containing information about a Reliable
   Datagram entry."
INDEX { ibConnRDLGID, ibConnRDLLID,
        ibConnRDLEEC, ibConnRDRGID,
        ibConnRDRLID, ibConnRDREEC }
::= { ibConnRDTable 1 }

IbConnRDEntry ::= SEQUENCE {
        ibConnRDLGID      GidTC,
        ibConnRDLLID      LidTC,
       ibConnRDLEEC      IBEtoEContext,
       ibConnRDRGID      GidTC,
```

```
       ibConnRDRLID      LidTC,
        ibConnRDREEC      IBEtoEContext
        }
```



```
ibConnRDLGID OBJECT-TYPE
SYNTAX      GidTC
MAX-ACCESS   read-only
STATUS      current
DESCRIPTION
   "GID of the local end of the connection."
::= { ibConnRDEntry 1 }

ibConnRDLLID OBJECT-TYPE
SYNTAX      LidTC
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION
   "LID of the local end of the connection."
::= { ibConnRDEntry 2 }

ibConnRDLEEC OBJECT-TYPE
SYNTAX      IBEtoEContext
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION
  "End to End Context of the local end of the connection."
::= { ibConnRDEntry 3 }

ibConnRDRGID OBJECT-TYPE
SYNTAX      GidTC
MAX-ACCESS   read-only
STATUS      current
DESCRIPTION
   "GID of the remote end of the connection."
::= { ibConnRDEntry 4 }

ibConnRDRLID OBJECT-TYPE
SYNTAX      LidTC
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION
   "LID of the remote end of the connection."
::= { ibConnRDEntry 5 }

ibConnRDREEC OBJECT-TYPE
SYNTAX      IBEtoEContext
```

```
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION
  "End to End Context of the remote end of the connection."
::= { ibConnRDEntry 6 }
```



```
--
-- Connected
--

ibConnConnected OBJECT IDENTIFIER ::= { ibConnMibObjects 3 }

ibConnCInSegs OBJECT-TYPE
SYNTAX      Counter64
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION
   "The total number of segments received, including those
    received in error.  This count includes segments received on
    currently established connections."
::= { ibConnConnected 1 }

ibConnCOutSegs OBJECT-TYPE
SYNTAX      Counter64
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION
   "The total number of segments sent, including those on
   current connections but excluding those containing only
   retransmitted octets."
::= { ibConnConnected 2 }


ibConnCTable OBJECT-TYPE
SYNTAX      SEQUENCE OF IbConnCEntry
MAX-ACCESS  not-accessible
STATUS      current
DESCRIPTION
   "A table containing information about all of the Connections in
   the device.  This table contains connections that are both
   reliable and unreliable."
::= { ibConnConnected 3 }

ibConnCEntry OBJECT-TYPE
SYNTAX      IbConnCEntry
MAX-ACCESS  not-accessible
STATUS      current
```

```
DESCRIPTION
   "A conceptual row of the containing information about a
   Connection entry."
INDEX { ibConnCLGID, ibConnCLLID, ibConnCLQPN,
        ibConnCRGID, ibConnCRLID, ibConnCRQPN }
::= { ibConnCTable 1 }
```



```
IbConnCEntry ::= SEQUENCE {
        ibConnCLGID       GidTC,
        ibConnCLLID      LidTC,
       ibConnCLQPN       IBQuePairNumber,
       ibConnCRGID       GidTC,
       ibConnCRLID       LidTC,
        ibConnCRQPN       IBQuePairNumber,
       ibConnCState      INTEGER,
        ibConnCReliable   TruthValue
        }

ibConnCLGID OBJECT-TYPE
SYNTAX      GidTC
MAX-ACCESS   read-only
STATUS      current
DESCRIPTION
   "GID of the local end of the connection."
::= { ibConnCEntry 1 }

ibConnCLLID OBJECT-TYPE
SYNTAX      LidTC
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION
   "LID of the local end of the connection."
::= { ibConnCEntry 2 }

ibConnCLQPN OBJECT-TYPE
SYNTAX      IBQuePairNumber
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION
  "Que Pair Number of the local end of the connection."
::= { ibConnCEntry 3 }

ibConnCRGID OBJECT-TYPE
SYNTAX      GidTC
MAX-ACCESS   read-only
STATUS      current
```

```
DESCRIPTION
   "GID of the remote end of the connection."
::= { ibConnCEntry 4 }

ibConnCRLID OBJECT-TYPE
SYNTAX      LidTC
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION
   "LID of the remote end of the connection."
::= { ibConnCEntry 5 }
```



```
ibConnCRQPN OBJECT-TYPE
SYNTAX      IBEtoEContext
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION
  "Que Pair Number of the remote end of the connection."
::= { ibConnCEntry 6 }

ibConnCState OBJECT-TYPE
SYNTAX       INTEGER {
                listen (1),
                reqSent (2),
                reqRcvd (3)
                repRcvd (4),
                Established (5),
                dreqSent (6),
                dreqRcvd (7),
                timeWait (8),
                rtuTimeout (9),
                peerCompare (10),
                timeout (11),
                repWait (12),
                mraREPSent (13),
                drepTimeout (14)
                }

MAX-ACCESS   read-only
STATUS       current
DESCRIPTION
   "State of the connection.  Derived from 12.9.5 and 12.9.6."
::= { ibConnCEntry 7 }

ibConnCReliable     OBJECT-TYPE
SYNTAX              TruthValue
```

```
MAX-ACCESS          read-only
STATUS              current
DESCRIPTION
   "Reliability of the connection."
::= { ibConnCEntry 8 }

--
-- Module Groups
--
ibCAGroups OBJECT IDENTIFIER ::= { ibCAConformance 1}
ibCAUnreliableDatagram OBJECT IDENTIFIER ::= { ibCAGroups 1 }
ibCAReliableDatagram OBJECT IDENTIFIER ::= { ibCAGroups 2 }
ibCAUnreliableConnected OBJECT IDENTIFIER ::= { ibCAGroups 3 }
ibCAReliableConnected OBJECT IDENTIFIER ::= { ibCAGroups 4 }

END
```


8. Security Considerations

There are no management objects defined in this MIB that have a MAX-ACCESS clause of read-write and/or read-create. So, if this MIB is implemented correctly, then there is no risk that an intruder can alter or create any management objects of this MIB via direct SNMP SET operations.

There are a number of managed objects in this MIB that may contain sensitive information. It is thus important to control even GET access to these objects and possibly to even encrypt the values of these objects when sending them over the network via SNMP. Not all versions of SNMP provide features for such a secure environment.

SNMPv1 by itself is not a secure environment. Even if the network itself is secure (for example by using IPSec), even then, there is no control as to who on the secure network is allowed to access and GET/SET (read/change/create/delete) the objects in this MIB. It is recommended that the implementers consider the security features as provided by the SNMPv3 framework. Specifically, the use of the User-based Security Model RFC 2574 [RFC2574] and the View-based Access Control Model RFC 2575 [RFC2575] are recommended.

It is then a customer/user responsibility to ensure that the SNMP entity giving access to an instance of this MIB, is properly configured to give access to the objects only to those principals (users) that have legitimate rights to indeed GET or SET

(change/create/delete) them.

9. Intellectual Property

The IETF takes no position regarding the validity or scope of any intellectual property or other rights that might be claimed to pertain to the implementation or use of the technology described in this document or the extent to which any license under such rights might or might not be available; neither does it represent that it has made any effort to identify any such rights. Information on the IETF's procedures with respect to rights in standards-track and standards-related documentation can be found in BCP-11. Copies of claims of rights made available for publication and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementers or users of this specification can be obtained from the IETF Secretariat.

The IETF invites any interested party to bring to its attention any copyrights, patents or patent applications, or other

proprietary rights which may cover technology that may be required to practice this standard. Please address the information to the IETF Executive Director.

10. References

[RFC2571] Harrington, D., Presuhn, R. and B. Wijnen, "An Architecture for Describing SNMP Management Frameworks", RFC 2571, April 1999.

[RFC1155] Rose, M. and K. McCloghrie, "Structure and Identification of Management Information for TCP/IP-based Internets", STD 16, RFC 1155, May 1990.

[RFC1212] Rose, M. and K. McCloghrie, "Concise MIB Definitions", STD 16, RFC 1212, March 1991.

[RFC1215] Rose, M., "A Convention for Defining Traps for use with the SNMP", RFC 1215, March 1991.

[RFC2578] McCloghrie, K., Perkins, D., Schoenwaelder, J., Case, J., Rose, M. and S. Waldbusser, "Structure of Management Information Version 2 (SMIv2)", STD 58, RFC 2578, April 1999.

[RFC2579] McCloghrie, K., Perkins, D., Schoenwaelder, J., Case, J., Rose, M. and S. Waldbusser, "Textual Conventions for SMIv2", STD 58, RFC 2579, April 1999.

[RFC2580] McCloghrie, K., Perkins, D., Schoenwaelder, J., Case, J., Rose, M. and S. Waldbusser, "Conformance Statements for SMIv2", STD 58, RFC 2580, April 1999.

[RFC1157] Case, J., Fedor, M., Schoffstall, M. and J. Davin, "Simple Network Management Protocol", STD 15, RFC 1157, May 1990.

[RFC1901] Case, J., McCloghrie, K., Rose, M. and S. Waldbusser, "Introduction to Community-based SNMPv2", RFC 1901, January 1996.

[RFC1906] Case, J., McCloghrie, K., Rose, M. and S. Waldbusser, "Transport Mappings for Version 2 of the Simple Network Management Protocol (SNMPv2)", RFC 1906, January 1996.

[RFC2572] Case, J., Harrington D., Presuhn R. and B. Wijnen, "Message Processing and Dispatching for the Simple Network Management Protocol (SNMP)", RFC 2572, April 1999.

[RFC2574] Blumenthal, U. and B. Wijnen, "User-based Security Model (USM) for version 3 of the Simple Network Management Protocol (SNMPv3)", RFC 2574, April 1999.

[RFC1905] Case, J., McCloghrie, K., Rose, M. and S. Waldbusser, "Protocol Operations for Version 2 of the Simple Network Management Protocol (SNMPv2)", RFC 1905, January 1996.

[RFC2573] Levi, D., Meyer, P. and B. Stewart, "SNMPv3 Applications", RFC 2573, April 1999.

[RFC2575] Wijnen, B., Presuhn, R. and K. McCloghrie, "View-based Access Control Model (VACM) for the Simple Network

Management Protocol (SNMP)", RFC 2575, April 1999.

[RFC2570] Case, J., Mundy, R., Partain, D. and B. Stewart, "Introduction to Version 3 of the Internet-standard Network Management Framework", RFC 2570, April 1999.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997

[IBTAArch] Infiniband Trade Association, ôInfiniband(TM) Architecture Specification Vol 1&2 Release 1.0aö, 1999, 2000

[IBIFMIB] Anderson, B., ôDefinitions of Managed Objects Infiniband Interface Typeö, Internet Draft ietf-anderson-ibif-MIB-00.txt, 2001

11. Author's Addresses

Bill Strahm
Sanera Systems Inc
1925 NW AmberGlen Parkway
Suite 155 Phone: 1-503-601-0263
Beaverton, OR 97006 Email: bill@sanera.net
USA