Proposal for an ISDN MIB

Günter Röck

Goals and Non-Goals

- Goals
  - enable remote configuration an administration
  - support all kinds of ISDN interface and switch types
  - supply statistics information
  - supply call history information
  - supply traps
- Non-Goals
  - no unusable or irrelevant information
  - no information already provided by other MIBs
  - no multilink related information
  - no static neighbor-port relationship

ISDN MIB: What do we need ?

- Port configuration group
  - Basic configuration for each B and D channel
- Call/Neighbor group
  - Information related to each neighbor
  - This is configuration and neigbor related statistics information
- Statistics group
  - B and D channel statistics
- Accounting/Call history group
- Traps

ISDN MIB:
What do we not want ?

- hardware related information (e.g. I/O addresses)
  - ➪ will never be configured by SNMP
- multilink related data
  - ➪ should be handled by upper layer (e.g. PPP)
- ISDN L1/L2/L3 specific data (e.g. timers, TEI information)
  - ➪ t.b.d.: would such information be of any use ?
- call forwarding information
  - ➪ t.b.d.: is this of any use in a router ?
- suppport of calling/called line information restriction options
  - ➪ t.b.d.: this would be a potential security hole; I always want a calling line to identify itself
- Any information already supported by ifTable

Port configuration group

- Number of D and B channels
- Additional information (e.g. firmware version, controller type)
- D channel table
  - interface type (basicRate, primaryRate --> tbd: ifTable to be used here ?)
  - switch type (e.g. TPH, VN2, VN3, DSS1, 1TR6, INS64, NET3, 5ESS, DMS100, Leased Line, ...)
  - X.25 related information (tbd: what do we need here ?)
  - charging enabled/disabled flag (tbd: should always be enabled ?!?)
  - tbd: ifSpeed, ifStructure as in XXXX MIB)
- B channel table
  - associated D channel ifIndex
  - my_address (note: must not be neighbor based due to network restrictions)
  - my_subaddress / my_eaz
  - call forward information (tbd: do we want this type of information ?)
  - security enable/disable flag (enable/disable calling address check for incoming calls)

Call/neighbor group

- Configuration table
  - ➪ to describe each potential neighbor
- Statistics table
  - ➪ to supply neighbor related statistica data

Neighbor group: Configuration table

- Physical interface
  - pointer to D or B channel; required for SPV, leased lines and CUG; otherwise optional
- permissions (incoming/outgoing/both directions)
- called (remote) number and subaddress
- Closed User Group for this call (DSS1, 1TR6)
- connection type (dialup, SPV, Leased Line, ...)
- connection information
  - e.g. on-demand, active, backup, passive, ...
  - timeout information (e.g. minimum/default idle time until disconnect, usage of network charging information)
- callback, reverse charging, Send Own EAZ (1TR6) flags
- Maximum number of retries (or maximum retry time) when calling
- tbd: Link Layer/Encapsulation Layer protocols (can this be done using the ifStack table ?)
- tbd: ISDN Layer 2 / Layer 3 protocols, OperStatus, AdminStatus

Neighbor group: Statistics table

- Last call (including current call ?) Duration
- Clear Reason (ASCII)
- Clear Code (encoded)
- Number of succeeded calls
- Number of failed calls
- Number of accepted calls
- Number of refused calls
- Time of last call attempt
- Current status (admin, oper)
- calling/called flag (calling, called or callback)
- number of charged units for current/last call
- calling number/subaddress for current/last call as delivered by network
- tbd: ifIndex for current/last call, B channel number, ...

Statistics group

- tbd: D channel statistics
  - L1 statistics
  - L2 activities, warnings, errors
  - L3 activities, aborts, warnings, errors, timeouts
- B channel statistics
  - Number of calls on this interface
  - Current / Last calling and called number / subaddress
  - calling/called flag
  - number of charged units
  - ifOperStatus
  - tbd: ifAdminStatus (other, disconnect)

Accounting/History group

- Configuration and status information
  - status (enabled/disabled/delete)
  - accounting table size
  - tbd: time to hold entries
  - tbd: trap configuration
- Accounting/History table
  - connect and disconnect times (tbd: sysUptime or time since 01/01/70 to be used ?)
  - associated logical and physical interfaces
  - called number
  - calling/called/callback flag
  - number of charged units
  - call termination reason (encoded; tbd: network or predefined codes ?)
  - tbd: furhter information (e.g. destination network layer address or host name)
- tbd: Accounting event table
  - event type (e.g. system-on, accounting-on, accounting-off, overflow, internal-error, delete-all), event time

Traps

- accounting/history table overflow trap
- connected/disconnected trap
- ...

Problems to be solved

- how to connect incoming calls to upper layer ?
  - calling number can be substantially different to called number, e.g. if a PBX system is involved or if caller supports multiple D channels
  - may need list of acceptable numbers for eachneighbor
- how do we get the called (remote) number (neighbor table entry) from the upper layer(s) ?
  - do we need a table for mapping Network/Link Layer addresses to neighbor table entries ?
- multilink handling (several neighbor entries to same destination)
- do we always expect the customer to enter the calling/called number information ?
- restrictive call permissions can be problematic with low disconnect timeouts
  - server might send response after disconnect timeout
- do we need P-t-P/P-t-MP configuration parameters?
- relationship (D and B channel interfaces) to ifEntries

Problems with callback and calling number verification

- incoming number can be entirely different from number to call back
  - ➪ leading digits can differ
    - e.g. incoming 721-xxx, outgoing 0721-xxx or incoming 0721-xxx and outgoing 1234-0721-xxx
  - ➪ trailing digits can differ
    - e.g. incoming xxx-264 or xxx-0, outgoing xxx-265
  - ➪ numbers can differ entirely
    - might happen with MSN in DSS-1 or if system supports more than one D channel