Implementation Notes and Experience for
The Internet Ethernet MIB

Status of this Memo

   This memo provides information for the Internet community. It does
   not specify an Internet standard. Distribution of this memo is
   unlimited.

Table of Contents

1. Introduction

   The Ethernet MIB Working group has been tasked with the following two
   work items:

   1) Develop a document explaining the rationale for assigning
       MANDATORY status to MIB variables which are optional in
       the relevant IEEE 802.3 specification (the technical
       basis for the Internet Ethernet MIB). This shall not be a
       standards-track document.

   (2) Develop an implementation report on the Ethernet MIB.
       This report shall cover MIB variables which are
       implemented in both Ethernet interface chips, and in
       software (i.e., drivers), and discuss the issues
       pertaining to both. This report shall also summarize
       field experience with the MIB variables, especially
       concentrating on those variables which are in dispute.
       This document shall not be a standards-track document.
       While the Ethernet MIB is progressing through the
       standardization process, this document shall be
       periodically updated to reflect the latest implementation
       and operational experience.

This document reflects the currently known status of 11 different
implementations of the MIB by 7 different vendors on 7 different
Ethernet interface chips.

2. Observations

There are some interesting points to be noted from this information:

1) Only 4 variables are actually implemented in all
    implementations: AlignmentErrors, FCSErrors,
    ExcessiveCollisions and InternalMacTransmitErrors.

2) There were another five variables implemented in all but
    one of the reported implementations,
    SingleCollisionFrames, MultipleCollisionFrames,
    LateCollisions, FrameTooLongs, and CarrierSenseErrors.

    Three of these variables exist in implementations that
    use the same chip as the implementation that does not
    contain the variable. Specifically:

  A)   SingleCollisionFrames is not implemented in
       implementation number 3, which uses the AMD LANCE.
       However, other AMD LANCE implementations (7, 8, and 10)
       do implement the variable, implying that it is
       available on the LANCE.

  B)   MultipleCollisionFrames is not implemented in
       implementation number 3, which uses the AMD LANCE.
       However, other AMD LANCE implementations (7, 8, and 10)
       do implement the variable, implying that it is
       available on the LANCE.

  C)   LateCollisions is not implemented in implementation
       number 1, which uses the Intel 82586. However, another
       Intel 82586 based implementation (11) does implement
       the variable, implying that it is available on the
       Intel 82586.

  D)   CarrierSenseErrors is not implemented on implementation
       number 2, which is based on the Fujitsu 86950 chip.
       However, there is only one implementation based on this
       chip and I have not been able to locate a data sheet on
       this part so no conclusion can be drawn at this time.

  E)   FrameTooLongs is not implemented on implementation
       number 5, which is based on the National NIC 8390 chip.
       However, there is only one implementation based on this

          chip and I have not been able to locate a data sheet on
          this part. It should also be noted that this variable
          is easily maintained by software as a "driver-level"
          function.

     (3) Of the 22 variables in the MIB, 11, or 1/2 of the
          variables, were implemented in about 1/2 or less of the
          implementations.

     4)   The number of variables implemented per implementation
          ranges from a low of 11 to a high of 16. The average
          number of variables truly implemented is 12.8.

     5)   The IEEE 802.3 encapsulation-specific variables
          (InRangeLengthErrors, and OutOfRangeLengthFields) are in
          2 and 0 implementations respectively.

3. Conclusions

     From this, the author concludes that:

     The control variables (IntializeMAC, etc.) are not widely
     implemented, but this may be due to an aversion to implementing
     writable variables until security is in place.

     One vendor has stated that the reason that these variables were not
     implemented was that the vendor did not believe the variables to be
     useful, and that they were hard to implement. Furthermore, this
     vendor has recommended dropping the variables entirely.

     The two IEEE 802.3 encapsulation variables (InRangeLengthErrors and
     OutOfRangeLengthFields) are barely implemented. In Santa Fe, the
     Working group discussed moving them to an optional, 802.3 specific,
     group. The author believes that this is justified by this
     implementation data.

     The collision histogram variables are also barely implemented. They
     should be in their own optional group -- and they are.

     Of the remaining 13 statistical variables, 9 of them are in 10 or 11
     implementations. This is good.

     Two of them (SQETestErrors and ExcessiveDeferrals) are in 3 and 1
     implementations, respectively. This is bad.

     The remaining variables (DeferredTransmissions and
     InternalMacReceiveErrors) are in 8 or 9 implementations.

It should be noted that one of the two systems that do not implement
DeferredTransmissions is based on the AMD LANCE, and other AMD LANCE
based systems do implement this counter, leading to the conclusion
that DeferredTransmissions could easily be on all but one of the
implementations.

The other such variable, InternalMacReceiveErrors, is a general
catchall for all other errors. If no other errors are detected by the
hardware or software then returning 0 for the counter is perfectly
acceptable.

This all seems to imply either:

1)   Splitting the statistics group into two groups, one of
     which is optional and contains SQETestErrors and
     ExcessiveDeferrals, or

2)   Eliminating SQETestErrors and ExcessiveDeferrals) from
     the MIB.

The variables with 8 or 9 implementations are a bit more problematic.
They are implemented in more than 2/3s of the implementations, but it
may not be appropriate to call this widespread implementation.
However, it seems to be safe to conclude that the non-implementations
of these variables is due to local implementation considerations
rather than a fundamental lack of support for the variable.

4. Final Action

   After consideration at the San Diego IETF Meeting on 17 March 1992,
   the Ethernet MIB Working Group made the following recommendations:

   1)   The dot3TestTdrValue object will be deprecated from the
        standard mib. There are effectively no implementations
        of this object, and some chips were reported to return an
        incorrect value for the TDR count.

   2)   The dot3StatsInRangeLengthErrors object and the
        dot3StatsOutOfRangeLengthFields object will be deprecated
        from the MIB. These objects were not widely implemented
        and their utility in diagnosing network problems was
        strongly questioned.

   3)   The dot3InitializeMac object, the dot3MacSubLayerStatus
        object, the dot3MulticastReceiveStatus object, and the
        dot3TxEnabled object will be deprecated from the MIB.
        These objects were not widely implemented and their
        utility in diagnosing network problems was strongly

questioned.

4) The dot3StatsExcessiveDeferrals object will be deprecated
    from the MIB. Only one system implemented this object.
    Furthermore, its exact definition was called into question.

5) The dot3StatsSQETestErrors object received few
    implementations. However, the working group strongly
    supported its retention in the MIB on the basis that
    certain forms of transceiver and cable errors that are
    not uncommon can only be detected with this counter.

6) The collision histogram table (dot3CollTable) will be
    kept as an optional group, even though the objects are
    not widely implemented nor is there hardware support on
    all reported chips.

5. Implementation Data

The following raw data has been provided by vendors, each developing
an implementation of the Ethernet MIB. Each reported implementation
has a separate column in the following table. For each
implementation/MIB Variable, a single character code has been entered
indicating the rough implementation status of the variable. These
codes are:

Y   Fully implemented, reports a truthful count, or
     indication of state. All values may be written to the
     variable with the expected action occurring.

N   Not implemented at all. Would return a noSuchName error
     if accessed.

C   Implemented but returns a constant value for gets and
     returns a badValue error for any set attempt to set the
     variable to a value other than this constant (writable
     variables only).

MIB				Implementation								
Variable	1	2	3	4	5	6	7	8	9	10	11	Yesses
InitializeMac	C	C	Y	Y	Y	Y	Y	C7	C7	N	Y	6
MacSubLayerStatus	C	C	Y	Y	Y	Y	Y	C7	C7	N	C	5
MulticastReceiveStatus	C	C	Y	C3	Y	C	C	C7	C7	N	C	2
TxEnabled	C	C	Y	Y	Y	Y	Y	C7	C7	N	C	5
TestTdrValue	C	1	C	C4	C	C	C	C4	C4	N	C	1
AlignmentErrors	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	11
FCSErrors	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	11
SingleCollisionFrames	Y	Y	Y	N	Y	Y	Y	Y	Y	Y	Y	10
MultipleCollisionFrames	Y	Y	Y	N	Y	Y	Y	Y	Y	Y	Y	10
SQETestErrors	Y	C	C	C	Y	C	C	C	C	Y	C	3
DeferredTransmissions	Y	C	Y	N	Y	Y	Y	Y	Y	Y	Y	9
LateCollisions	C	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	10
ExcessiveCollisions	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	11
InternalMacTransmitErrors	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	11
CarrierSenseErrors	Y	C	Y	Y	Y	Y	Y	Y	Y	Y	Y	10
ExcessiveDeferrals	C	C	Y	C	C	C	C	C	C	N	C	1
FrameTooLongs	Y	Y2	Y	Y	C	Y	Y	Y	Y	Y	Y	10
InRangeLengthErrors	C	C	C	N5	C	Y	Y	C	C	N	C	2
OutOfRangeLengthFields	C	C	C	C6	C	C	C	C	C	N	C	0
InternalMacReceiveErrors	Y	Y	Y	Y	Y	C	C	Y	Y	Y	C	8
CollCount	Y	Y	C	N	N	N	N	C	C	N	Y	3
CollFrequencies	Y	Y	C	N	N	N	N	C	C	N	Y	3
Yesses	13	11	16	11	15	14	14	11	11	12	13	

Notes:

1   does not implement TDR test, but reports TDR from last
     collision!

2   Not supported by the chip, detected solely in software.

3   But set to disabled(2) -> badValue

4   Underlying TDR function not implemented on this chip.

5   Only counts frames too short though.

6   Due to Ethernet encapsulation

7   Implementation does not support set operations but
     reports the correct value for these.

The implementations are:

```
        Implementation   Vendor   Chip
                  1      1        Intel 82586
                  2      1        Fujitsu 86950
                  3      2        Sonic
                  4      3        AMD Lance
                  5      4        National NIC 8390C
                  6      4        Intel 82596
                  7      4        AMD Lance
                  8      5        AMD Lance
                  9      5        AMD ILACC
                 10      6        AMD Lance
                 11      7        Intel 82586
```

6. Security Considerations

   Security issues are not discussed in this memo.

7. Author's Address

   Frank J. Kastenholz
   FTP Software
   2 High Street
   North Andover Mass 01845

   Phone: 508-685-4000
   EMail: kasten@ftp.com