Interface Management API Implementation Agreement (ATM Interfaces)

Revision 3.0

Editor: Tim Shanley, TranSwitch, tim.shanley@transwitch.com

The words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", “MAY”, and "OPTIONAL" in the remainder of this document are to be interpreted as described in the NPF Software API Conventions Implementation Agreement revision 2.0.

For additional information contact:
The Network Processing Forum, 39355 California Street,
Suite 307, Fremont, CA 94538
+1 510 608-5990 phone info@npforum.org

Table of Contents

Table of Figures

Table of Tables

1 Introduction

This document defines the data structures and functions calls for the ATM Interface Manageent API.

ATM Forum specification AF-PHY-0086.001 describes Inverse Multiplexing for ATM (IMA). IMA allows aggregation of multiple low-speed links into one larger virtual trunk or IMA group. An inverse multiplexer appears to the ATM switch router as one logical pipe. This IMA group provides modular bandwidth for user access to ATM networks or for connections between ATM network elements at rates between the traditional order multiplex levels, such as between T1 or E1 and T3 or E3.

IMA involves inverse multiplexing and demultiplexing of ATM cells in a cyclical fashion among links grouped to form a higher bandwidth logical group with a rate approximately the sum of the link rates. This group of links is called an IMA group.

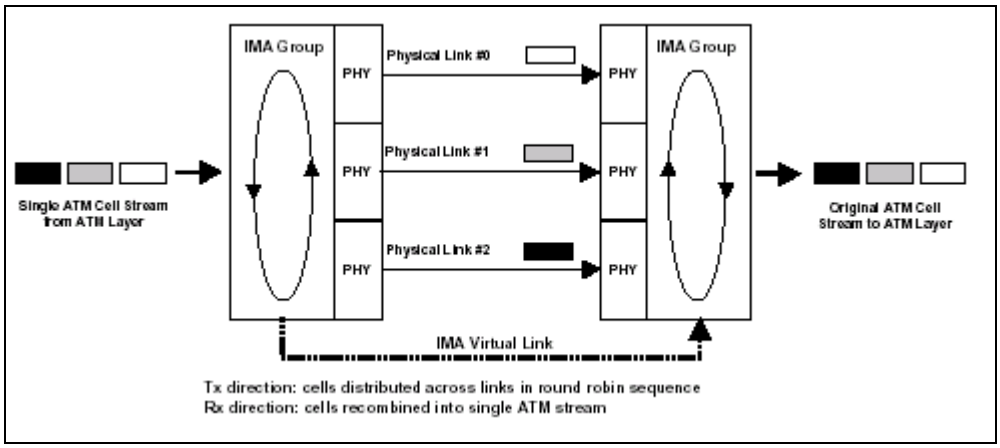

Figure 1-1 Inverse Multiplexing

An ATM IMA interface is typically in a parent-child relationship with the “IMA group” of interfaces. This “IMA group” consists typically of multiple PDH interfaces, so several PDH interfaces are parents of one ATM IMA interface.

Inverse multiplexing in the transmit direction controls the distribution of cells onto the group of physical links available to the IMA group interfaces. It also aligns the transmission frames and deals with links that are added or dropped, or fail and are later restored. In the receive direction, the IMA interface performs differential delay compensation and recombines the cells into the original ATM cell stream while allowing minimal cell delay variation (CDV). The IMA process of splitting and recombining the ATM cell stream is as transparent to the layer above as a traditional single-link physical layer interface.

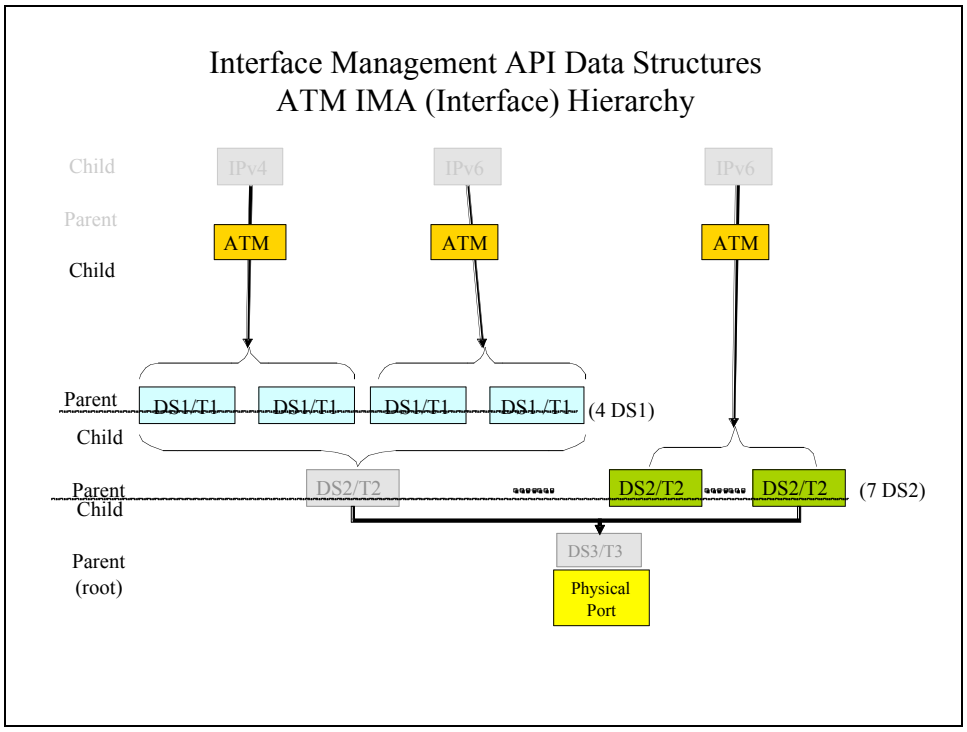

Figure 1-2 Interface Management API Data Structures ATM IMA Hierarchy

2 Data Definitions

The following new data definitions and types are to be added to the Interface Management API description:

2.1 ATM Interface Data structures

2.1.1 ATM Interface Type

This interface type introduces an abstraction of the mapping of ATM cells into SONET, SDH or PDH frames. An ATM interface is an interface that is the child of a physical layer interface, or some interface that abstracts the functions of a layer that carries ATM cells. Typically, the parent of an ATM interface is a SONET, a SDH, or a PDH interface.

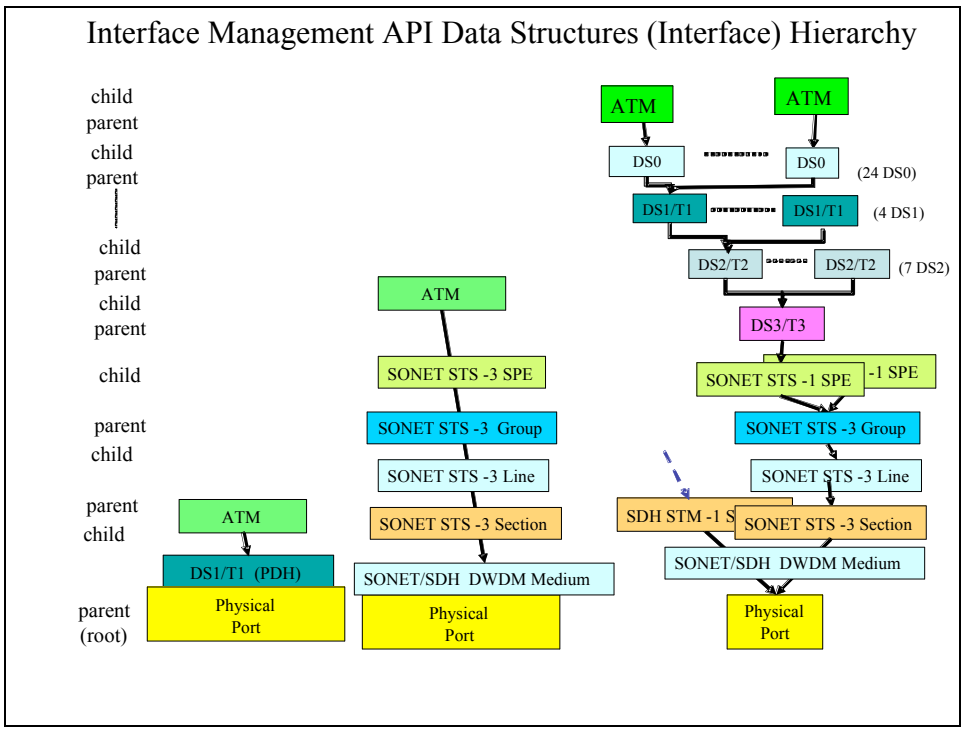

Figure 2-1 ATM Interface relationship with lower later interfaces

The ATM interface type can be used for abstracting the functions of both "ATM UNI" and "ATM NNI".

For supporting the ATM interface, the ATM interface type must be exist in the Core Interface Management API interface structure.

```
#define IF_TYPE_ATM   3   /* ATM interface type */
```

2.1.2 ATM Interface Attributes Data Structure: NPF_IfATM_t

A forward reference to the ATM NPF Interface Attributes data structure must exist in the **`NPF_IfGeneric_t`** structure in **`npf_if_core.h`** if ATM interfaces are supported. So, before the declaration of **`NPF_IfGeneric_t`**, the following must appear:

```
typedef struct NPF_IfATM NPF_IfATM_t;
```

The following must also appear inside the union within the **`NPF_IfGeneric_t`** structure:

```
NPF_IfATM_t *ATM_Attr; /* ATM interface attributes */
```

The following data structure contains configuration parameters for the ATM interface type. It is used when setting, or querying ATM interface attributes.

```
/*
```

```
 *    ATM Interface Attributes
 */
struct IfATM {

    NPF_IfATM_IMA_LineFill_t  fill_pattern;
            /* Cell filling excess bandwidth [TC, and IMA Group] */
    NPF_boolean_t       HEC_error_discard;
            /* When set to TRUE, discard single bit errored cells */
            /* When set to False, correct single bit error */
    NPF_uint8_t         nLines;
            /* # of lines in the IMA group*/
    NPF_uint8_t         maxLinesInGroup;
            /* Max # lines in the IMA group */
    NPF_uint8_t         minRx;
            /* Min # lines for receive IMA Group */
    NPF_uint8_t         minTx;
            /* Min # lines for xmit IMA Group */
    NPF_uint8_t         defTxRefLink;
            /* Default Transmit Reference Link in the IMA Group */
    NPF_uint32_t        dcbCells;
            /* Delay Compensation Buffer size in cells [IMA Group] */
    NPF_uint32_t        lddTol;
            /* Link Differential Delay Tolerance in msec [IMA Group] */

    NPF_IfATM_IMA_Symmetry_t  symmetry;
            /* IMA Group Symmetry Mode */
    NPF_IfATM_IMA_FrameLength  frameLen;
            /* IMA Frame length */
    NPF_IfATM_IMA_Tclock_t    tClock;
            /* Transmit Clock Configuration [IMA Group} */
    NPF_IfATM_IMA_Ver_t       imaVer;
            /* IMA Version code */

};
```

In the ATM Interface Attributes Data Structure the field “nLines” indicates whether the structure is used as a single line ATM, or as a multi-line ATM interface using Inverse Multiplexing (IMA).

If an ATM interface uses a serial line port or SONET port directly, the **`nLines`** variable is set to zero to indicate that an IMA group is not used. If an ATM interface is built upon an IMA group, the **`nLines`** is set to a non-zero value, to indicate the number of lines which are part of the IMA group. The members of the IMA group are specified as parents of the ATM interface, using unique interface Identifiers in the parent interface identifier array **parentIDs* of the generic interface data structure *NPF_IfGeneric_t*.

The maximum lines in the IMA group, *maxLinesInGroup*, the minimum receives lines, *minRx*, and the minimum transmit lines, *minTx* must satisfy the following:

`maxLinesInGoup` $\geq$ ***`nLines`***,
`minRx` and ***`minTx`*** *nonzero,* and $\leq$ ***`maxLinesInGroup`***

The ***defTxRefLink*** variable (see AF-PHY-0086.001, section 8.1.1) must be less than ***nLines***; it is the index of one of the interface IDs in the ****parentIDs*** array in the *NPF_IfGeneric_t*, indicating the timing reference interface for the IMA group.

dcbCells is the size of the receive delay compensation buffer, in cells (see AF-PHY-0086.001 section 9.2 and Appendix B).

lddTol is the Link Differential Delay Tolerance: the number of cells above which a Loss of Delay Synchronization is declared. This is the total window of time allowed between the slowest and fastest link (see AF-PHY-0086.001, section 9.2).

The other member structures of the ATM interface attributes data structure are defined as follows :

2.1.2.1 ATM IMA Line Filling

Rate Decoupling Filler Cell (see AF-PHY-0086.001 section 5.2.2.2.1):

```
typedef struct   {
    NPF_uint8_t                   oamlabel;   /* OAM label for IMA */
                                              /* filler cell       */
    NPF_IfATM_IMA_Stuffing_t      atmfiller;  /* Traditional filler*/
                                              /* cell idle/unassign*/
} NPF_IfATM_IMA_LineFill_t;

typedef enum   {
  NPF_IF_IMA_LineStuff_IdleCells         = 1,
  NPF_IF_IMA_LineStuff_UnassignedCells   = 2
} NPF_IfATM_IMA_Stuffing_t;
```

2.1.2.2 ATM IMA Group Symmetry Mode

Group Symmetry Mode (see AF-PHY-0086.001 section 5.2.2.7):

```
typedef enum    {
  NPF_IF_IMA_SYMMETRICAL_CONFIG_OP             = 1,
  NPF_IF_IMA_SYMMETRICAL_CONFIG_ASYMM_OP       = 2,
  NPF_IF_IMA_ASYMMETRICAL_CONFIG_OP            = 3
} NPF_IfATM_IMA_Symmetry_t;
```

2.1.2.3 ATM IMA Frame Length

IMA Frame Length (see AF-PHY-0086.001, section 5.2.2.4.2):

```
typedef enum    {
  NPF_IF_IMA_FL_32     = 1,       /* 32 Cells */
  NPF_IF_IMA_FL_64     = 2,       /* 64 Cells */
  NPF_IF_IMA_FL_128    = 3,       /* 128 Cells */
  NPF_IF_IMA_FL_256    = 4        /* 256 Cells */
```

```
} NPF_IfATM_IMA_FrameLength;
```

2.1.2.4 ATM IMA Transmit Clock Configuration

IMA Transmit Clock Configuration (AF-PHY-0086.001, sections 4.3 and 7):

```
typedef enum    {
  NPF_IF_IMA_TCLOCK_CTC     = 1,        /* Common Transmit Clock */
  NPF_IF_IMA_TCLOCK_ITC     = 2         /* Indep. Transmit Clock */
} NPF_IfATM_IMA_Tclock_t;
```

2.1.2.5 ATM IMA protocol version

The IMA protocol version in use (see AF-PHY-0086.001, section 5.2.2.3). Auto-negotiation means the near end will attempt to use version 1.1 but fall back to 1.0 if the far end indicates it is using the older version of IMA protocol.

```
typedef enum    {
  NPF_IF_IMA_VER_AUTO  = 1,             /* Auto-negotiate */
  NPF_IF_IMA_VER_1_1   = 2,             /* Force V1.1 */
  NPF_IF_IMA_VER_1_0   = 3              /* Use deprecated V1.0 */
} NPF_IfATM_IMA_Ver_t;
```

2.2 ATM Interface Fault Management Data Structures

2.2.1 ATM Interface Fault Management Status

2.2.1.1 ATM Status Bit Definitions

The ATM TC Layer status is a 32 bit statistic where each bit represents a different piece of information. The bits are described as follows:

```
typedef struct {
      NPF_IfATM_Status_Bits_t    ATMStatus;
}NPF_IfATM_Status_t;


typedef enum {

NPF_IF_ATM_HEC_Error  = 1,
      /*  HEC error detected  */
} NPF_IfATM_Status_Bits_t;
```

2.2.1.2 NPF_IfATM_IMA_AlarmStatus_t

The structure points to an array of alarm types; each element present identifies an alarm that is currently active.

```
struct NPF_IfATM_IMA_AlarmStatus {
      NPF_uint32_t            nAlarms;    /* # of alarms in array */
      NPF_IfATM_IMA_Alarms_t  *alarms;    /* 1 for each active alarm */
};
```

2.2.2 ATM Interface IMA Alarm Indication Structures

These structures contain arrays of alarm codes representing alarms that are active. There is one for each of the four groups of alarms.

```
typedef enum {

/*
** Link alarms (following Table 20 in AF-PHY-0086.001)
*/


      NPF_IF_IMA_Alarm_LIF = 1,      /* Persistence of LIF at the NE*/
                                           /*(R-138) */
      NPF_IF_IMA_Alarm_LODS = 2,     /* Persistence of LODS at the NE*/
                                           /*(R-139) */
      NPF_IF_IMA_Alarm_RFI_IMA = 3,  /* Persistence of RFI-IMA at NE*/
                                           /*(R-140) */
      NPF_IF_IMA_Alarm_TX_MisCon = 4,
                                 /* Tx Link detected as misconnected */
                                 /*(R-141) */
      NPF_IF_IMA_Alarm_RX_MisCon = 5,
                                 /* Rx Link detected as misconnected */
                                 /*(R-142) */
      NPF_IF_IMA_Alarm_TX_Fault = 6,
                                 /* Tx Fault declared at NE */
                                 /*(O-28) */
      NPF_IF_IMA_Alarm_RX_Fault = 7,
                                 /* Rx Fault declared at NE */
                                 /*(O-29) *
      NPF_IF_IMA_Alarm_TX_Unusable_FE  = 8,
                                 /* Tx Unusable declared at FE */
                                 /*(R-143) */
      NPF_IF_IMA_Alarm_RX_Unusable_FE = 9,
                                 /* Rx Unusable declared at FE */
                                 /*(R-144) *
/*
** Group Alarms (following Table 20 in AF-PHY-0086.001)
*/
      NPF_IF_IMA_Alarm_STARTUP_FE    = 10, /*FE is in Start-up */
                                           /* State (r-145) */
```

```
      NPF_IF_IMA_Alarm_CONFIG_ABORT  = 11, /*NE Configuration Aborted */
                                          /*State (R-146) */
      NPF_IF_IMA_Alarm_CONFIG_ABORT_FE = 12,/*FE Configure Aborted*/
                                           /*State (R-147) */
      NPF_IF_IMA_Alarm_INSUF_LINKS   = 13, /*NE Insufficient Links */
                                          /*State  (R-148)*/
      NPF_IF_IMA_Alarm_INSUF_LINKS_FE = 14,/*FE Insufficient Links */
                                          /*State* (R-149) */
      NPF_IF_IMA_Alarm_BLOCKED_FE    = 15, /*FE Blocked State (R-150)*/

      NPF_IF_IMA_Alarm_TIMING_MIS    = 16  /*FE & NE timing MISMATCH  */
                                       /*(CTC/ITC) mismatch (r-151)*/


} NPF_IfATM_IMA_Alarms_t;
```

2.3 ATM Interface Performance Monitoring Data Structures

2.3.1 NPF_IfATM_GlobalStatistics_t

This structure contains ATM global statistics. Counters are 32 bits, except for transmitted cell counts, which are 64-bit counters. Counters are unsigned, and wrap when they reach the maximum value.

```
typedef struct NPF_IfATM_GlobalStatistics_t {
  /* Ingress counters */

  NPF_uint32_t IHECErrCrcted;  /*Total number of ingress cells */
                               /*with HEC errors corrected     */
  NPF_uint32_t IHECErrDrop;    /*Total number of ingress cells */
                               /*discarded with HEC errors     */
  NPF_uint32_t IRdfoverflow;   /*Total number of ingress cells */
                               /*discarded - rate decoupler */
                               /*FIFO overflow at PHY layer    */
 NPF_uint64_t IRcvCells;       /*Total ingress cells received, */
 NPF_uint64_t EXmtCells;       /*Total egress cells transmitted, */

  NPF_IfATM_Status_t    ATM_Status; /*The ATM Layer Status */
};
```

2.3.2 NPF_IfATM_IMA_Statistics_t

This structure contains IMA link and group statistics. Counters are unsigned, and wrap from 0xFFFFFFFF to zero, using twos-complement arithmetic.

```
/*
** ATM IMA Statistics: Follow Table 19 in AF-PHY-0086.001
*/


struct NPF_IfATM_IMA_Statistics{
```

```
/*
** Link Statistics following Table 19 in AF-PHY-0086.001
*/

  NPF_uint32_t IV_IMA;  /*ICP violations */
                              /* (R-125) */
  NPF_uint32_t OIF_IMA; /*OIF anomalies; out of IMA frame */
                              /* (R-125) */
  NPF_uint32_t SES_IMA; /*Count of Severly Erorred Second
                         /* intervals at the NE */
                         /* (R-126) */

  NPF_uint32_t SES_IMA_FE; /*Count of */
                          /* Severly Erorred Second intervals at FE */
                          /* (R-127) */

  NPF_uint32_t UAS_IMA;       /*Unavailable seconds at NE */
                              /* (R-128) */

  NPF_uint32_t UAS_IMA_FE;    /*Unavailable seconds at FE */
                              /* (R-129) */

  NPF_uint32_t TX_UUS_IMA; /*TX UNUSABLE seconds */
                        /* count of unusable seconds at the TX NE LSM */
                        /* (R-130) */

  NPF_uint32_t RX_UUS_IMA; /*RX UNUSABLE seconds */
                        /* count of unusable seconds at the RX NE LSM */
                        /* (R-131) */

  NPF_uint32_t TX_UUS_IMA_FE; /*TX UNUSABLE seconds */
                        /* count of unusable seconds at the TX FE LSM */
                        /* (R-132) */

  NPF_uint32_t RX_UUS_IMA_FE; /*RX UNUSABLE seconds */
                        /* count of unusable seconds at the RX FE LSM */
                        /* (R-133) */

  NPF_uint32_t TX_FC;   /*Count of NE TX group failure conditions */
                         /* Possible NE group failure alarms:*/
                         /* NPF_IF_IMA_Alarm_TX_MisCon, and */
                        /* NPF_IF_IMA_Alaram_TX-Fault */
                          /* (R-134) */

  NPF_uint32_t RX_FC;   /*Count of NE RX group failure conditions */
                         /* Possible NE group failure alarms:*/
                         /* NPF_IF_IMA_Alarm_LIF, */
                         /* NPF_IF_IMA_Alarm_LODS, */
                         /* NPF_IF_IMA_Alarm_Rx_Mis-Connected, and */
                         /* NPF_IF_IMA_Alarm_RX_Fault */
                          /* (R-135) */

  NPF_uint32_t TX_FC_FE; /*Count of FE TX group failure conditions */
                          /* Possible FE group failure alarms:*/
                          /* NPF_IF_IMA_Alarm_Tx_Unusable_FE*/
                          /* (O-21) */
```

```
  NPF_uint32_t RX_FC_FE; /*Count of FE RX group failure conditions */
                         /* Possible NE group failure alarms:*/
                         /* NPF_IF_IMA_Alarm_RFI_IMA, and */
                        /* NPF_IF_IMA_Alarm_TX_Unusable_FE */
                         /* (O-22) */

  NPF_uint32_t TX_Stuff_IMA;
            /* Count of stuff events inserted in TX direction */
              /* (O-23) */

  NPF_uint32_t RX_Stuff_IMA;
            /* Count of stuff events inserted in RX direction */
               /* (O-24*/

/*
** Group statistics following Table 19 in AF-PHY-0086.001
*/

  NPF_uint32_t GR_UAS_IMA;     /*Count of one second intervals where*/
                               /*the GTSM is Down (R-136) */

  NPF_uint32_t GR_FC;          /*Count of NE group failure condition*/
                               /*entrances. The possible NE group*/
                               /*failure alarm conditions are:*/
                               /*Config-Aborted and Insufficient-Links*/
                               /* (R-137) */

  NPF_uint32_t Group_FC_FE;    /*Count of FE group failure condition*/
                               /*entrances. The possible FE group*/
                               /*failure alarm conditions are:*/
                               /*Start-up-FE, Config-Aborted-FE,*/
                               /*Insufficient-Links-FE, and Blocked-FE*/
                               /* (O-25) */
};
```

2.4 ATM Completion Callback Type Codes

The following Call Back Types are used by ATM interfaces in the **`NPF_IfCallbackType_t`** variable in asynchronous callbacks; this value indicates what function is generating the callback.

```
/*
** ATM Completion Callback Type Codes
**/

/*
** Fault Management
*/

#define NPF_IF_ATM_IMA_ALARMS_GET ((NPF_IF_TYPE_ATM_<<16)+1)
#define NPF_IF_ATM_IMA_ALARMS_SET ((NPF_IF_TYPE_ATM_<<16)+2)

/*
** Performance Monitoring
```

```
#define NPF_IF_ATM_GLOBAL_STATISTICS_GET ((NPF_IF_TYPE_ATM_<<16)+10)
#define NPF_IF_ATM_GLOBAL_STATISTICS_SET ((NPF_IF_TYPE_ATM_<<16)+11)
#define NPF_IF_ATM_IMA_STATISTICS_GET ((NPF_IF_TYPE_ATM_<<16)+12)
#define NPF_IF_ATM_IMA_STATISTICS_SET ((NPF_IF_TYPE_ATM_<<16)+13)
```

2.4.1 Asynchronous Response Array Element: NPF_IfAsyncResponse_t

The **NPF_IfAsyncResponse_t** type is defined in the Core Interface Management IA. This structure contains a union. In this union are pointers to various structures returned by Interface Management API functions. If the ATM interface type is supported, the following must be included in the union within the **NPF_IfAsyncResponse_t** structure:

```
/*
** Definitions to be inserted in 'npf_if_core.h'
*/

typedef struct NPF_IfATM_GlobalStatistics
       NPF_IfATM_GlobalStatistics_t;
typedef struct NPF_IfATM_IMA_AlarmStatus
       NPF_IfATM_IMA_AlarmStatus_t;
typedef struct NPF_IfATM_IMA_Statistics
      NPF_IfATM_IMA_Statistics_t;

/*
** Asynchronous Response types for ATM interfaces
*/


/*
** Fault Managament
*/
      NPF_IfATM_IMA_AlarmStatus_t    *imaAlarms;
/*
**  ATM Performance Monitoring
*/
      NPF_IfATM_GlobalStatistics_t   *atmGlobalStatistics;
      NPF_IfATM_IMA_Statistics_t *imaStatistics;
```

Function Name	Type Code	Structure Returned
NPF_IfATM_IMA_AlarmsGet	NPF_IF_ATM_IMA_GROUP_ALARMS_GET	NPF_IfATM_IMA_AlarmsStatus_t
NPF_IfATM_IMA_AlarmsSet	NPF_IF_ATM_IMA_GROUP_ALARMS_GET	unused
NPF_IfATM_GlobalStatisticsGet	NPF_IF_ATM_GLOBAL_STATISTICS_GET	NPF_IfATM_GlobalStatistics_t
NPF_IfATM_GlobalStatisticsSet	NPF_IF_ATM_GLOBAL_STATISTICS_SET	unused
NPF_IfATM_IMA_StatisticsGet	NPF_IF_ATM_IMA_GROUP_STATISTICS_GET	NPF_IfATM_IMA_Statistics_t
NPF_IfATM_IMA_StatisticsSet	NPF_IF_ATM_IMA_GROUP_STATISTICS_SET	unused

Table 2-1 Function, Callback Type Code, Callback Return Structure

2.5 ATM Interface Management Events

The following is the list of event types used by ATM

```
/*
** ATM Interface Management IMA Events
*/


#define NPF_IF_ATM_EV_IMA_Start_up_FE      ((NPF_IF_TYPE_ATM<<16)+1)
#define NPF_IF_ATM_EV_IMA_Config-Aborted   ((NPF_IF_TYPE_ATM<<16)+2)
#define NPF_IF_ATM_EV_IMA_Config-Aborted-FE      ((NPF_IF_TYPE_ATM<<16)+3)
#define NPF_IF_ATM_EV_IMA_Insufficient-Links     ((NPF_IF_TYPE_ATM<<16)+4)
#define NPF_IF_ATM_EV_IMA_Insufficient-Links-FE ((NPF_IF_TYPE_ATM<<16)+5)
#define NPF_IF_ATM_EV_IMA_Blocked-FE             ((NPF_IF_TYPE_ATM<<16)+6)
#define NPF_IF_ATM_EV_IMA_GR-Timing-Mismatch     ((NPF_IF_TYPE_ATM<<16)+7)
#define NPF_IF_ATM_EV_IMA_LossHEC-Delineation((NPF_IF_TYPE_ATM<<16)+8)
```

This list must be added to the definitions of events in the “core” document.

The following must be added to the union defined in the **`NPF_IfEventData_t`** structure defined in the IM Core API document:

```
NPF_IfATM_IMA_AlarmStatus_t    *imaAlarms; /* ATM-IMA alarms */
```

3 Function Definitions

The following existing functions can be used with PDH and ATM/IMA interfaces:

- **`NPF_IfCreate()`**
- **`NPF_IfDelete()`**
- **`NPF_IfCreateAndSet()`**
- **`NPF_IfGenericStatsGet()`**
- **`NPF_IfAttrSet()`**
- **`NPF_IfEnable()`**
- **`NPF_IfDisable()`**
- **`NPF_IfOperStatusGet()`**

Setting the attributes of an ATM interface with correct, nonzero parameters in the **`NPF_IfIMA_Grp_t`** structure makes it an ATM IMA interface.

In addition, the following new functions are defined:

3.1 Fault Management Functions

3.1.1 NPF_IfATM_IMA_AlarmsGet

```
NPF_error_t NPF_IfATM_IMA_AlarmsGet(
      NPF_IN NPF_callbackHandle_t         cbHandle,
      NPF_IN NPF_correlator_t             correlator,
      NPF_IN NPF_errorReporting_t         errorReporting,
      NPF_IN NPF_IfHandle_t               handle
);
```

Description
This function call is used to retrieve ATM IMA group alarm status from the device. The function returns an array of codes, indicating all the IMA group alarms currently active. These codes are enumerated at "ATM **Error! Reference source not found.**."
Arguments

- `NPF_callbackHandle_t cbHandle`: Callback Handle that was returned when the callback function was registered.
- `NPF_correlator_t correlator:` Correlator value of the application's choosing. It will be returned with the callback.
- `NPF_errorReporting errorReporting:` specifies whether a callback is desired always, never, or only if an error occurs.
- `NPF_IfHandle_t handle`: the handle of an ATM IMA interface.

Callback Response
This function returns a NPF_IfATM_IMA_AlarmStatus_t structure in the callback response.

3.1.2 NPF_IfATM_IMA_AlarmsSet

```
NPF_error_t NPF_IfATM_IMA_AlarmsSet(
      NPF_IN NPF_callbackHandle_t         cbHandle,
      NPF_IN NPF_correlator_t             correlator,
      NPF_IN NPF_errorReporting_t         errorReporting,
      NPF_IN NPF_IfHandle_t               handle,
      NPF_IN NPF_IfATM_IMA_Alarms_t       *pIMA_AlarmType,
      NPF_IN NPF_uint32_t                 nSize,
      NPF_boolean_t                       enable
);
```

Description
This function call is used to configure ATM IMA group alarm reporting from the device. The function can accept an array of alarm identifiers that the user would like to enable or disable. The parameter `nSize` denotes the size of the array being passed to the function.
Arguments

- `NPF_callbackHandle_t cbHandle`: Callback Handle that was returned when the callback function was registered.
- `NPF_correlator_t correlator:` Correlator value of the application's choosing. It will be returned with the callback.
- `NPF_errorReporting errorReporting:` specifies whether a callback is desired always, never, or only if an error occurs.
- `NPF_IfHandle_t handle`: the handle of an ATM IMA interface.
- `NPF_IfATM_IMA_Alarms_t *pIMA_AlarmType`: Pointer to an array that contains the list of identifiers for ATM IMA group alarm that need to be enabled or disabled. Alarm type codes are listed in "ATM **Error! Reference source not found.**."
- `NPF_uint16_t nSize`: The number of elements in the `pIMA_AlarmType` array.
- `NPF_boolean_t enable`: Set to TRUE to enable and FALSE to disable the array of alarms passed to the function.

Callback Response
This function returns only a success/failure code in the callback response.

3.2 Performance Monitoring Functions

3.2.1 NPF_IfATM_GlobalStatisticsGet

```
NPF_error_t NPF_IfATM_GlobalStatisticsGet  (
      NPF_IN NPF_callbackHandle_t          cbHandle,
      NPF_IN NPF_correlator_t              correlator,
      NPF_IN NPF_errorReporting_t          errorReporting,
      NPF_IN NPF_IfHandle_t                handle
);
```

Description
This function retrieves Global statistics for an ATM interface and returns them via a callback.

Arguments

- `NPF_IN NPF_callbackHandle_t cbHandle`: Callback Handle that was returned when the callback function was registered.
- `NPF_IN NPF_correlator_t correlator:` Correlator value of the application's choosing. It will be returned with the callback.

- `NPF_errorReporting errorReporting:` specifies whether a callback is desired always, never, or only if an error occurs. This argument is ignored, because the callback must always be made to return the statistics.
- `NPF_IfHandle_t handle`: The Interface Handle of the ATM interface for which statistics are requested.

Callback Response
The function returns an NPF_IfATM_GlobalStatistics_t structure.

3.2.2 NPF_IfATM_GlobalStatisticsSet

```
NPF_error_t NPF_IfATM_GlobalStatisticsSet (
      NPF_IN NPF_callbackHandle_t          cbHandle,
      NPF_IN NPF_correlator_t              correlator,
      NPF_IN NPF_errorReporting_t          errorReporting,
      NPF_IN NPF_IfHandle_t                handle,
      NPF_IN NPF_IfATM_GlobalStatistics_t  *pAtmStatsIdentifierArray,
      NPF_IN NPF_uint32_t                  nArrayLength,
      NPF_IN NPF_boolean_t                 isEnabled
);
```

Description
This function call is used to enable or disable ATM global statistics on an ATM interface. You pass it an array containing the identifier codes of the statistics you want to enable or disable. These codes are defined in NPF_IfATM_GlobalStatistics_t.

Arguments
- `NPF_IN NPF_callbackHandle_t cbHandle`: Callback Handle that was returned when the callback function was registered.
- `NPF_IN NPF_correlator_t correlator:` Correlator value of the application's choosing. It will be returned with the callback.
- `NPF_IN NPF_errorReporting errorReporting:` specifies whether a callback is desired always, never, or only if an error occurs. This argument is ignored, because the callback must always be made to return the statistics.
- `NPF_IN NPF_IfHandle_t handle`: The Interface Handle of the ATM interface for which statistics are to be enabled or disabled.
- `NPF_IN NPF_IfATM_GlobalStats_t *pAtmStatsIdentifierArray`: This points to an array of ATM global statistics identifiers (numbers) to enable or disable.
- `NPF_IN NPF_uint32_t nArrayLength`: The number of elements in the array pointed to by `pAtmStatsIdentifierArray`.
- `NPF_IN NPF_boolean_t isEnabled`: This enables (TRUE) or disables (FALSE) the statistics that are listed in the array pointed to by `pAtmStatsIdentifierArray`.

Callback Response
This function returns only a success/failure code in the callback response.

3.2.3 NPF_IfATM_IMA_StatisticsGet

```
NPF_error_t NPF_IfATM_IMA_StatisticsGet    (
      NPF_IN NPF_callbackHandle_t          cbHandle,
      NPF_IN NPF_correlator_t              correlator,
      NPF_IN NPF_errorReporting_t          errorReporting,
      NPF_IN NPF_IfHandle_t                handle
);
```

Description

This function retrieves IMA Group statistics and returns them via a callback.

Arguments

- `NPF_IN NPF_callbackHandle_t cbHandle`: Callback Handle that was returned when the callback function was registered.
- `NPF_IN NPF_correlator_t correlator:` Correlator value of the application's choosing. It will be returned with the callback.
- `NPF_errorReporting errorReporting:` specifies whether a callback is desired always, never, or only if an error occurs. This argument is ignored, because the callback must always be made to return the statistics.
- `NPF_IfHandle_t handle`: The Interface Handle associated with the ATM IMA interface for which statistics are requested.

Callback Response

The function returns an NPF_IfATM_IMA_Statistics_t structure.

3.2.4 NPF_IfATM_IMA_StatisticsSet

```
NPF_error_t NPF_IfATM_IMA_StatisticsSet    (
      NPF_IN NPF_callbackHandle_t          cbHandle,
      NPF_IN NPF_correlator_t              correlator,
      NPF_IN NPF_errorReporting_t          errorReporting,
      NPF_IN NPF_IfHandle_t                handle,
      NPF_IN NPF_IfATM_IMA_Statistics_t
                                           *pAtmStatsIdentifierArray,
      NPF_IN NPF_uint32_t                  nArrayLength,
      NPF_IN NPF_boolean_t                 isEnabled
);
```

Description

This function call is used to enable or disable IMA Group statistics on an ATM interface. You pass it an array containing the identifier codes of the statistics you want to enable or disable. These codes are defined in NPF_IfATM_IMA_Statistics_t.

Arguments

- `NPF_IN NPF_callbackHandle_t cbHandle`: Callback Handle that was returned when the callback function was registered.
- `NPF_IN NPF_correlator_t correlator:` Correlator value of the application's choosing. It will be returned with the callback.
- `NPF_IN NPF_errorReporting errorReporting:` specifies whether a callback is desired always, never, or only if an error occurs. This argument is ignored, because the callback must always be made to return the statistics.
- `NPF_IN NPF_IfHandle_t handle`: The Interface Handle of the ATM interface for which statistics are to be enabled or disabled.
- `NPF_IN NPF_IfATM_IMA_GroupStaistics_t *pAtmStatsIdentifierArray`: This points to an array of IMA Group statistics identifiers (numbers) to enable or disable.
- `NPF_IN NPF_uint32_t nArrayLength`: The number of elements in the array pointed to by `pAtmStatsIdentifierArray`.
- `NPF_IN NPF_boolean_t isEnabled`: This enables (TRUE) or disables (FALSE) the statistics that are listed in the array pointed to by `pAtmStatsIdentifierArray`.

Callback Response

This function returns only a success/failure code in the callback response.

4 Summary

4.1 Summary of API Functions, and Input Data Structures

The following is a summary table of the ATM, requirements, and ATM IMA Function Calls:

Function Name	Required?	Data Structure
NPF_IfATM_IMA_AlarmsGet	Optional	None
NPF_IfATM_IMA_AlarmsSet	Optional	NPF_IfATM_IMA_AlarmStatus_t
NPF_IfATM_GlobalStatisticsGet	Optional	None
NPF_IfATM_GlobalStatisticsSet	Optional	NPF_IfATM_GlobalStatistics_t
NPF_IfATM_IMA_StatisticsGet	Optional	None
NPF_IfATM_IMA_StatisticsSet	Optional	NPF_IfATM_IMA_Statistics_t

Table 4-1 Summary of Function Calls, Reuirements, and Input Data Structures

5 References

- AF-PHY-0086.001 – ATM Forum
- NPF Core Interface Management API
- NPF SONET/SDH Interface Management API

- NPF PDF Interface Management API

6 Revision History

V00	04/30/2004	Separate ATM IMA text from NPF2004.080.
V01	05/13/2004	Add ATM IMA interface type and IMA Interface Data Strcutures Hierarchy text and figure
V02	07/08/2004	Add ATM interface, and change to new IM API format
V03	07/14/2004	Merge ATM and IMA into one interface type.
V04	07/14/2004	Cleanup text. Add Figures, Table, Appendix A, and Appendix B
V05	07/21/2004	Editorial updates to Configuration data structure, Global Statistics.
V06	08/31/2004	Make corrections to Appendix A so that the *.h file compiles /w errors
V07	09/02/2004	Make some additional corrections to Appendix A so that the *.h file compiles /w errors
V08	10/10/2004	Straw Ballot Comment Resolution
V09	10/13/2004	Finalize Straw Ballot Comment Resolution
V10	11/03/2004	Finalize Straw Ballot Comment Resolution, and IA headers

APPENDIX A NPF_IF_ATM.H

```
/* NPF_IF_ATM.h */

/*
**  This header file defines typedefs, constants, and functions
**  that apply to the NPF PDH Interface Management API.
**  It is defined based on the Interface Management API structures.
**  It contains some of the structures from "npf_if.h" file, which were
**  extended with SONET/SDH API Data Structures definitions.
**
**
*/


#ifndef __NPF_IF_ATM_H_
#define __NPF_IF_ATM_H_

#ifdef __cplusplus
extern "C"  {
#endif

/*
**  *** ATM Definitions
*/

/*
** +++ Interface Type definitions
*/

#define     NPF_IF_TYPE_ATM   3           /* ATM interface */



typedef enum   {
    NPF_IF_IMA_LineStuff_IdleCells        = 1,
    NPF_IF_IMA_LineStuff_UnassignedCells  = 2
  } NPF_IfATM_IMA_Stuffing_t;


typedef struct    {
      NPF_uint8_t                   oamlabel;   /* OAM label for IMA */
                                                /* filler cell       */
      NPF_IfATM_IMA_Stuffing_t      atmfiller;  /* Traditional filler*/
                                                /* cell idle/unassign*/
  } NPF_IfATM_IMA_LineFill_t;

typedef enum      {
    NPF_IF_IMA_SYMMETRICAL_CONFIG_OP            = 1,
    NPF_IF_IMA_SYMMETRICAL_CONFIG_ASYMM_OP      = 2,
    NPF_IF_IMA_ASYMMETRICAL_CONFIG_OP           = 3
  } NPF_IfATM_IMA_Symmetry_t;

typedef enum      {
```

```
    NPF_IF_IMA_FL_32      = 1,        /* 32 Cells */
    NPF_IF_IMA_FL_64      = 2,        /* 64 Cells */
    NPF_IF_IMA_FL_128     = 3,        /* 128 Cells */
    NPF_IF_IMA_FL_256     = 4         /* 256 Cells */
  } NPF_IfATM_IMA_FrameLength;


  typedef enum    {
    NPF_IF_IMA_TCLOCK_CTC      = 1,        /* Common Transmit Clock */
    NPF_IF_IMA_TCLOCK_ITC      = 2         /* Indep. Transmit Clock */
  } NPF_IfATM_IMA_Tclock_t;

  typedef enum    {
    NPF_IF_IMA_VER_AUTO  = 1,              /* Auto-negotiate */
    NPF_IF_IMA_VER_1_1   = 2,              /* Force V1.1 */
    NPF_IF_IMA_VER_1_0   = 3               /* Use deprecated V1.0 */
  } NPF_IfATM_IMA_Ver_t;

/*
** ATM IMA Alarm bit definitions
*/

typedef enum {

/*
** Link alarms (following Table 20 in AF-PHY-0086.001)
*/


      NPF_IF_IMA_Alarm_LIF = 1,      /* Persistence of LIF at the NE*/
                                           /*(R-138) */
      NPF_IF_IMA_Alarm_LODS = 2,     /* Persistence of LODS at the NE*/
                                           /*(R-139) */
      NPF_IF_IMA_Alarm_RFI_IMA = 3,  /* Persistence of RFI-IMA at NE*/
                                           /*(R-140) */
      NPF_IF_IMA_Alarm_TX_MisCon = 4,
                                /* Tx Link detected as misconnected */
                                /*(R-141) */
      NPF_IF_IMA_Alarm_RX_MisCon = 5,
                                /* Rx Link detected as misconnected */
                                /*(R-142) */
      NPF_IF_IMA_Alarm_TX_Fault = 6,
                                /* Tx Fault declared at NE */
                                /*(O-28) */
      NPF_IF_IMA_Alarm_RX_Fault = 7,
                                /* Rx Fault declared at NE */
                                /*(O-29) *
      NPF_IF_IMA_Alarm_TX_Unusable_FE  = 8,
                                /* Tx Unusable declared at FE */
                                /*(R-143) */
      NPF_IF_IMA_Alarm_RX_Unusable_FE = 9,
                                /* Rx Unusable declared at FE */
                                /*(R-144) *
/*
** Group Alarms (following Table 20 in AF-PHY-0086.001)
*/
      NPF_IF_IMA_Alarm_STARTUP_FE    = 10, /*FE is in Start-up */
```

```
                                        /* State (r-145) */
      NPF_IF_IMA_Alarm_CONFIG_ABORT  = 11, /*NE Configuration Aborted */
                                        /*State (R-146) */
      NPF_IF_IMA_Alarm_CONFIG_ABORT_FE = 12,/*FE Configure Aborted*/
                                         /*State (R-147) */
      NPF_IF_IMA_Alarm_INSUF_LINKS   = 13, /*NE Insufficient Links */
                                        /*State  (R-148)*/
      NPF_IF_IMA_Alarm_INSUF_LINKS_FE = 14,/*FE Insufficient Links */
                                        /*State* (R-149) */
      NPF_IF_IMA_Alarm_BLOCKED_FE    = 15, /*FE Blocked State (R-150)*/

      NPF_IF_IMA_Alarm_TIMING_MIS    = 16  /*FE & NE timing MISMATCH  */
                                      /*(CTC/ITC) mismatch (r-151)*/


} NPF_IfATM_IMA_Alarms_t;

/*
** ATM Alarm Status
*/

typedef struct    {
      NPF_uint32_t            nAlarms;    /* # of alarms in array */
      NPF_IfATM_IMA_Alarms_t  *alarms;    /* 1 for each active alarm */
} NPF_IfATM_IMA_AlarmStatus_t;

/*
** ATM Status bits
*/

typedef enum {

NPF_IF_ATM_HEC_Error  = 1,
      /*  HEC error detected  */
} NPF_IfATM_Status_Bits_t;

/*
** ATM Status
*/


typedef struct {
      NPF_IfATM_Status_Bits_t    ATMStatus;
}NPF_IfATM_Status_t;


/*
 *    ATM Interface Attributes
 */
struct IfATM {

    NPF_IfATM_IMA_LineFill_t  fill_pattern;
            /* Cell filling excess bandwidth [TC, and IMA Group] */
    NPF_boolean_t       HEC_error_discard;
            /* When set to TRUE, discard single bit errored cells */
            /* When set to False, correct single bit error */
    NPF_uint8_t         nLines;
```

```
            /* # of lines in the IMA group*/
    NPF_uint8_t         maxLinesInGroup;
            /* Max # lines in the IMA group */
    NPF_uint8_t         minRx;
            /* Min # lines for receive IMA Group */
    NPF_uint8_t         minTx;
            /* Min # lines for xmit IMA Group */
    NPF_uint8_t         defTxRefLink;
            /* Default Transmit Reference Link in the IMA Group */
    NPF_uint32_t        dcbCells;
            /* Delay Compensation Buffer size in cells [IMA Group] */
    NPF_uint32_t        lddTol;
            /* Link Differential Delay Tolerance in msec [IMA Group] */

    NPF_IfATM_IMA_Symmetry_t  symmetry;
            /* IMA Group Symmetry Mode */
    NPF_IfATM_IMA_FrameLength frameLen;
            /* IMA Frame length */
    NPF_IfATM_IMA_Tclock_t    tClock;
            /* Transmit Clock Configuration [IMA Group} */
    NPF_IfATM_IMA_Ver_t       imaVer;
            /* IMA Version code */

};

/*
** ATM Global Statistics
*/

typedef struct NPF_IfATM_GlobalStatistics_t {
  /* Ingress counters */

  NPF_uint32_t IHECErrCrcted;  /*Total number of ingress cells */
                               /*with HEC errors corrected     */
  NPF_uint32_t IHECErrDrop;    /*Total number of ingress cells */
                               /*discarded with HEC errors     */
  NPF_uint32_t IRdfoverflow;   /*Total number of ingress cells */
                               /*discarded - rate decoupler */
                               /*FIFO overflow at PHY layer    */
 NPF_uint64_t IRcvCells;       /*Total ingress cells received, */
 NPF_uint64_t EXmtCells;       /*Total egress cells transmitted, */

  NPF_IfATM_Status_t    ATM_Status; /*The ATM Layer Status */
};


/*
** ATM IMA Statistics: Follow Table 19 in AF-PHY-0086.001
*/


struct NPF_IfATM_IMA_Statistics{

/*
** Link Statistics following Table 19 in AF-PHY-0086.001
*/
```

```
NPF_uint32_t IV_IMA;   /*ICP violations */
                              /* (R-125) */
NPF_uint32_t OIF_IMA; /*OIF anomalies; out of IMA frame */
                              /* (R-125) */
NPF_uint32_t SES_IMA; /*Count of Severly Erorred Second
                      /* intervals at the NE */
                      /* (R-126) */

NPF_uint32_t SES_IMA_FE; /*Count of */
                      /* Severly Erorred Second intervals at FE */
                      /* (R-127) */

NPF_uint32_t UAS_IMA;        /*Unavailable seconds at NE */
                              /* (R-128) */

NPF_uint32_t UAS_IMA_FE;     /*Unavailable seconds at FE */
                              /* (R-129) */

NPF_uint32_t TX_UUS_IMA; /*TX UNUSABLE seconds */
                     /* count of unusable seconds at the TX NE LSM */
                     /* (R-130) */

NPF_uint32_t RX_UUS_IMA; /*RX UNUSABLE seconds */
                     /* count of unusable seconds at the RX NE LSM */
                     /* (R-131) */

NPF_uint32_t TX_UUS_IMA_FE; /*TX UNUSABLE seconds */
                     /* count of unusable seconds at the TX FE LSM */
                     /* (R-132) */

NPF_uint32_t RX_UUS_IMA_FE; /*RX UNUSABLE seconds */
                     /* count of unusable seconds at the RX FE LSM */
                     /* (R-133) */

NPF_uint32_t TX_FC;    /*Count of NE TX group failure conditions */
                       /* Possible NE group failure alarms:*/
                       /* NPF_IF_IMA_Alarm_TX_MisCon, and */
                       /* NPF_IF_IMA_Alaram_TX-Fault */
                        /* (R-134) */

NPF_uint32_t RX_FC;    /*Count of NE RX group failure conditions */
                       /* Possible NE group failure alarms:*/
                       /* NPF_IF_IMA_Alarm_LIF, */
                       /* NPF_IF_IMA_Alarm_LODS, */
                       /* NPF_IF_IMA_Alarm_Rx_Mis-Connected, and */
                       /* NPF_IF_IMA_Alarm_RX_Fault */
                        /* (R-135) */

NPF_uint32_t TX_FC_FE; /*Count of FE TX group failure conditions */
                        /* Possible FE group failure alarms:*/
                        /* NPF_IF_IMA_Alarm_Tx_Unusable_FE*/
                        /* (O-21) */

NPF_uint32_t RX_FC_FE; /*Count of FE RX group failure conditions */
                       /* Possible NE group failure alarms:*/
                       /* NPF_IF_IMA_Alarm_RFI_IMA, and */
                       /* NPF_IF_IMA_Alarm_TX_Unusable_FE */
```

```
                          /* (O-22) */

  NPF_uint32_t TX_Stuff_IMA;
            /* Count of stuff events inserted in TX direction */
              /* (O-23) */

  NPF_uint32_t RX_Stuff_IMA;
            /* Count of stuff events inserted in RX direction */
               /* (O-24*/

/*
** Group statistics following Table 19 in AF-PHY-0086.001
*/

  NPF_uint32_t GR_UAS_IMA;     /*Count of one second intervals where*/
                               /*the GTSM is Down (R-136) */

  NPF_uint32_t GR_FC;          /*Count of NE group failure condition*/
                               /*entrances. The possible NE group*/
                               /*failure alarm conditions are:*/
                               /*Config-Aborted and Insufficient-Links*/
                               /* (R-137) */

  NPF_uint32_t Group_FC_FE;    /*Count of FE group failure condition*/
                               /*entrances. The possible FE group*/
                               /*failure alarm conditions are:*/
                               /*Start-up-FE, Config-Aborted-FE,*/
                               /*Insufficient-Links-FE, and Blocked-FE*/
                               /* (O-25) */
};



/*
** ATM Completion Callback Type Codes
*/

/*
** Fault Management
*/

#define NPF_IF_ATM_IMA_ALARMS_GET ((NPF_IF_TYPE_ATM_<<16)+1)
#define NPF_IF_ATM_IMA_ALARMS_SET ((NPF_IF_TYPE_ATM_<<16)+2)

/*
** Performance Monitoring
*/

#define NPF_IF_ATM_GLOBAL_STATISTICS_GET ((NPF_IF_TYPE_ATM_<<16)+10)
#define NPF_IF_ATM_GLOBAL_STATISTICS_SET ((NPF_IF_TYPE_ATM_<<16)+11)
#define NPF_IF_ATM_IMA_STATISTICS_GET ((NPF_IF_TYPE_ATM_<<16)+12)
#define NPF_IF_ATM_IMA_STATISTICS_SET ((NPF_IF_TYPE_ATM_<<16)+13)



/*
** Asynchronous Response types for ATM interfaces
```

```
*/

/*
** The following definitions are to be included in 'npf_if_core.h'
*/
typedef struct NPF_IfATM_GlobalStatistics
      NPF_IfATM_GlobalStatistics_t;
typedef struct NPF_IfATM_IMA_AlarmStatus
      NPF_IfATM_IMA_AlarmStatus_t;
      typedef struct NPF_IfATM_IMA_Statistics
      NPF_IfATM_IMA_Statistics_t;



/*
** Fault Managament
*/
         NPF_IfATM_IMA_AlarmStatus_t       *imaGroupAlarms;
/*
** Performance Monitoring
*/
         NPF_IfATM_GlobalStatistics_t      *atmGlobalStatistics;
         NPF_IfATM_IMA_Statistics_t *imaGroupStatistics;

/*
** End of definitions to be included in 'npf_if_core.h'
*/



/*
** ATM Interface Management IMA Events
*/

#define NPF_IF_ATM_EV_IMA_Start_up_FE      ((NPF_IF_TYPE_ATM<<16)+1)
#define NPF_IF_ATM_EV_MA_Config_Aborted    ((NPF_IF_TYPE_ATM<<16)+2)
#define NPF_IF_ATM_EV_IMA_Config_Aborted_FE
      ((NPF_IF_TYPE_ATM<<16)+3)
#define NPF_IF_ATM_EV_IMA_Insufficient_Links
      ((NPF_IF_TYPE_ATM<<16)+4)
#define NPF_IF_ATM_EV_IMA_Insufficient_Links_FE
((NPF_IF_TYPE_ATM<<16)+5)
#define NPF_IF_ATM_EV_IMA_Blocked_FE
      ((NPF_IF_TYPE_ATM<<16)+6)
#define NPF_IF_ATM_EV_IMA_GR_Timing_Mismatch
      ((NPF_IF_TYPE_ATM<<16)+7)
#define NPF_IF_ATM_EV_IMA_LossHEC_Delineation ((NPF_IF_TYPE_ATM<<16)+8)


#ifdef no_core_definitions
/*
** To the Core NPF_IfEventData_t structure adds the following:
*/
union {
      NPF_uint32_t                        unused;

         NPF_IfATM_IMA_AlarmStatus_t             imaAlarms;
```

```
} u;

#endif /* no_core_definitions */

/*
** Function Definitions
*/

NPF_error_t NPF_IfATM_IMA_AlarmsGet (
      NPF_IN NPF_callbackHandle_t         cbHandle,
      NPF_IN NPF_correlator_t             correlator,
      NPF_IN NPF_errorReporting_t         errorReporting,
      NPF_IN NPF_IfHandle_t               handle
);


NPF_error_t NPF_IfATM_IMA_AlarmsSet(
      NPF_IN NPF_callbackHandle_t         cbHandle,
      NPF_IN NPF_correlator_t             correlator,
      NPF_IN NPF_errorReporting_t         errorReporting,
      NPF_IN NPF_IfHandle_t               handle,
      NPF_IN NPF_IfATM_IMA_Alarms_t       *pIMA_AlarmType,
      NPF_IN NPF_uint32_t                       nSize,
      NPF_IN NPF_boolean_t                enable );


NPF_error_t NPF_IfATM_GlobalStatisticsGet  (
      NPF_IN NPF_callbackHandle_t         cbHandle,
      NPF_IN NPF_correlator_t             correlator,
      NPF_IN NPF_errorReporting_t         errorReporting,
      NPF_IN NPF_IfHandle_t               handle );


NPF_error_t NPF_IfATM_GlobalStatisticsSet  (
      NPF_IN NPF_callbackHandle_t         cbHandle,
      NPF_IN NPF_correlator_t             correlator,
      NPF_IN NPF_errorReporting_t         errorReporting,
      NPF_IN NPF_IfHandle_t               handle,
    NPF_IN NPF_IfATM_GlobalStatistics_t   *pAtmStatsIdentifierArray,
    NPF_IN NPF_uint32_t                   nArrayLength,
    NPF_IN NPF_boolean_t                  isEnabled );


NPF_error_t NPF_IfATM_IMA_StatisticsGet (
      NPF_IN NPF_callbackHandle_t         cbHandle,
      NPF_IN NPF_correlator_t             correlator,
      NPF_IN NPF_errorReporting_t         errorReporting,
      NPF_IN NPF_IfHandle_t               handle );



NPF_error_t NPF_IfATM_IMA_StatisticsSet    (
      NPF_IN NPF_callbackHandle_t         cbHandle,
      NPF_IN NPF_correlator_t             correlator,
      NPF_IN NPF_errorReporting_t         errorReporting,
      NPF_IN NPF_IfHandle_t               handle,
```

```
    NPF_IN NPF_IfATM_IMA_Statistics_t
      *pAtmStatsIdentifierArray,
    NPF_IN NPF_uint32_t                   nArrayLength,
    NPF_IN NPF_boolean_t                  isEnabled );

/*
** *** End of ATM Data Structure Definitions
*/

#ifdef __cplusplus
}
#endif

#endif

 /* __NPF_IF_ATM_H_ */
```

APPENDIX B LIST OF COMPANIES BELONGING TO NPF DURING APPROVAL PROCESS

Agere Systems
Alcatel
Altera
AMCC
Analog Devices
Avici Systems
Azanda Network Devices
Cypress Semiconductor
Ericsson
Erlang Technologies
EZ Chip
Flextronics
Fujitsu Ltd.
FutureSoft
HCL Technologies
Hi/fn
IBM
IDT
Intel
IP Infusion
Kawasaki LSI
LSI Logic
Modelware
Mosaid
Motorola
NEC
NetLogic
Nokia
Paion Co., Ltd.
PMC Sierra
RadiSys
Samsung Electronics
Sandburst Corporation
Silicon & Software Systems
Silicon Access
Sony Electronics
STMicroelectronics
Sun Microsystems
Teja Technologies
TranSwitch
U4EA Group
Xelerated
Xilinx
Zettacom
ZTE

APPENDIX C ACKNOWLEDGEMENTS

Working Group Chair:

Alex Conta, Transwitch, aconta@txc.com

Task Group Chair:

Alex Conta, Transwitch, aconta@txc.com

Task Group Editor:

John Renwick, Agere Systems, jrenwick@agere.com

The following individuals are acknowledged for their participation to IM API TG teleconferences, plenary meetings, mailing list, and/or for their NPF contributions used for the development of this Implementation Agreement. This list may not be all-inclusive since only names supplied by member companies for inclusion here will be listed. The NPF wishes to thank all active participants to this Implementation Agreement, whether listed here or not.

The list is in alphabetical order of last names:

Francesco Caggioni (Transwitch)
Alex Conta (Transwitch)
Vinoj Kumar (Agere Systems)
Karen Nielsen (Ericsson)
Erik Pedersen (Ericsson)
John Renwick (Agere Systems)