V. Sahay
B. Narayanan
R. Ramaswami
O. Aboul-Magd
B. Jamoussi
Nortel Networks

R. Jain
S. Dharanikota
Nayna Networks

R. Hartani
Caspian Networks

Internet Draft
Document: draft-sahay-ccamp-ntip-01.txt

July 2001

**Network Transport Interface Protocol (NTIP) for Photonic Cross Connects (PXC)**

Status of this Memo



## 1. Abstract

This draft describes the transport network interface protocol (NTIP) for photonic cross connects (PXC). NTIP is implemented between a PXC and transport network element (TNEs), also known as line systems. NTIP is a protocol that uses TCP/IP for the transport of information related to defect notification, trace monitoring, adjacency discovery, and diagnostic messages between directly attached PXC and TNE. The use of TCP as the transport protocol ensures reliable and in-sequence delivery of NTIP messages.

## 2. 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 [2].

## 3. Introduction

Fast restoration of failed light paths is critical for PXCs and it requires support for fast and accurate failure detection. Most PXCs do not look into the optical signals that flow through them. Some can detect presence or loss of light on a port. However presence of light does not necessarily mean that the light path is fine. For example when a link fails, generators between the point of failure and the PXC may inject an alarm indication signal (AIS) on a light signal. Unless the PXC decodes the framing of the light signal (possibly using some electrical circuitry), AIS will appear as presence of light. Also, some of the DWDM equipment (also called Transport Network Elements in this document) can control the switching ON and OFF of the AIS by configuration (or on demand). This feature can be leveraged to the advantage of the faster fault detection and hence the recovery times.

Unlike the PXC, the transport network elements (TNE) attached to it are aware of their equipment failure as well as the quality and framing of the light paths passing through them. Failure information can be dynamically conveyed from the TNE to the PXC using out of band IP messaging.

The transport network interface protocol (NTIP) defines the set of messages and the transport mechanism used for the exchange of failure conditions. NTIP is implemented between the PXC and the TNE. NTIP is an IP message handshake transported over an IP network. The implementation of NTIP between TNE and PXC achieves the following goals:

- Defect Notification: NTIP is used by the TNE to relay failure conditions and precise link, fiber, and equipment status notification to PXC. Defect reporting can be both event-driven (e.g., link failures), polling-driven (e.g., regular link sanity checks) or time-driven (e.g., periodic).

- Trace Monitoring: A PXC can use NTIP to request a TNE to monitor a certain pattern in the overhead of a light path. This capability could be used by PXC for validating light path identity.

- Diagnostics: NTIP could be used by OXC to request a TNE to perform diagnostics on any port or channel.

- Adjacency Discovery: NTIP could be used by both PXC and TNE to send and receive in band signals for automatic discovery of their optical connectivity.





This draft describes NTIP procedures and messages to achieve its design goals.

## 4. Reference Model

The current generation of PXCs will interoperate with opaque line systems such as SONET regenerators, wavelength translators, etc. Eventually when enough transparent line systems are deployed, PXCs will interwork with them leading to an all optical network (AON). The need for NTIP exists in both cases. The following reference model focuses on interoperation of PXCs with opaque line systems.

A reference model is shown in Figure 1. TNE are used to connect PXC ports to the transport system. TNE could be a set of OEO devices followed by a DWDM mux as shown in Figure 1. A TNE regenerates the signal coming from the PXC and might be capable of translating the wavelength. The key features of a TNE in the reference model are its ability to detect defects on its ports and to communicate defect information back to the PXC using NTIP.

```
                Drop    Line
                Side    side
                 |      |
                 |      |
      +---------+|  TNE | |\                /|   TNE   +---------+
      |     +---|V +---+V | \              / |  +---+  |---+     |
      |     |Por|->|   |->|  \            /  |<-|   |<-|Por|     |
      |     |  t|  |---|  |   \          /   |  |---|  |  t|     |
      |     |   |<-|   |<-|    \        /    |->|   |->|   |     |
      |     +---|  +---+  |     \=====>/     |  +---+  |---+     |
      |         |         |     /<=====\     |         |     PXC |
      | PXC +---|  +---+  |    /        \    |  +---+  |---+     |
      |     |Por|->|   |->|   /          \   |<-|   |<-|Por|     |
      |     |  t|  |---|  |  /            \  |  |---|  |  t|     |
      |     |   |<-|   |<-| /  DWDM MUX    \ |->|   |->|   |     |
      |     +---|  +---+  |/                \|  +---+  |---+     |
      |=========|                                      |=========|
```

```
      |controlle|  +===+                          +===+  |controlle|
      | r       |<>|   |                          |   |<>| r       |
      +---------+^ +===+                          +===+ ^+---------+
                 |  TNE                            TNE  |
                 | controller               controller  |
               NTIP                                    NTIP
```

PXC = Photonic CrossConnect        TNE = Transport Network Element
TX = Transmit                      RX = Receive

FIGURE 1: PXC to TNE Reference Configuration

Each PXC port consists of two unidirectional fibers, one for input (RX) and one for output (TX). Four TNE ports are associated with each PXC port (RX and TX). The TNE port that is connected to PXC is referred to as drop-side port, and the one that is connected to the





line is called the line-side port. Figure 1 shows that a PXC will communicate with several TNEs. On the other hand, a TNE only communicates with a single PXC.

A TNE can detect failures on signals it receives. Additionally some TNEs may be able to detect certain failures on their output ports such as laser failure.

## 5. NTIP Overview

### 5.1 Configuration

Whenever automatic discovery is not available, a PXC is provisioned with information on how its ports are connected to TNE. It is also provisioned with the primary and the secondary IP addresses of each TNE connected to it.

NTIP defines a set of configurable parameters such as protocol timers, retry counts, etc. Those parameters could be assigned default values or allowed to be set as needed or negotiated.

### 5.2 Registration

As a TNE starts up, it registers with the PXC whose address is configured in it. A TNE registers by opening a TCP session with the PXC, and sending a registration request message. TNE registration allows PXC to create its internal context structure for this particular TNE.

### 5.3 Keep Alive

NTIP utilizes keep alive messages. Keep alive messages are exchanged periodically between TNE and PXC to verify the sanity of the connection. The keep alive message interval is a configurable parameter.

### 5.4 Automatic Discovery

A PXC might have hundreds or even thousands of ports. Manual entries of port adjacency between PXC and TNEs connected to it is tedious and error prone. Furthermore manual entry has to be repeated every time there is change in adjacency due to, for instance, re-cabling.

One application of the NTIP interface is for use in automatic adjacency discovery between PXC and TNEs. For that purpose it is assumed that the TNE is aware of its ports mapping, i.e. the mapping between its drop side TX and RX ports, and its line side ports. This mapping is mad available using NTIP to the auto discovery process (ADP).

The TNE drop side port is supposed to operate in one of two modes, Pass-through and Insert. In the Insert mode the drop side TX inserts an identity pattern into the signal overhead, e.g. J0. The PXC is





made aware of the TNE port identity patterns over the NTIP interface using a MappingTable message. In the Pass-through mode, the TNE drop side TX passes the signal without modification.

The auto discovery process starts with the PXC port establishing a cross connect to a well-known Test RX as shown in Figure 2. The PXC then sends a ModeSwitch message to the TNE to switch the TNE drop side TX mode to the Insert mode. The TNE starts inserting its identity pattern (could be a system wide unique id or a trail trace id).

```
+-------------+     +----+----+
|             |---->| RX | TX |--->
|             |     +----+----+
|    PXC      |
|             |     +----+----+
|             |<----| TX | RX |<---
|             |     +----+----+
+-------------+
  |       ^    <--  NTIP-->
  |       |
  V       |
+---+   +---+
```

```
                 |   |   |   |
                 +---+   +---+

                  Test    Test
                   RX      TX
```

Figure 2: Auto Discovery Configuration

The Test RX is requested to report the Identity message it sees. The ADP maps the identity message to the drop side TX port id and the PXC port id. To verify mapping, the Test TX is requested to send the same identity pattern to the TNE drop side RX over NTIP using the discovered PXC port. The TNE drop side RX verify the received identity pattern and reports match pr mismatch.

The Test TX/RX could be internal or external to the PXC. In the case of external Test TX/RX, communication would be done over NTIP interface.

The procedure for auto-discovery is also applicable for link verification.

### 5.5 Defect Monitoring

Defect monitoring is initiated by the PXC by telling the TNE which ports to monitor for defects (signal degradation, loss of light, etc.). Defect monitoring will be initiated by the PXC after setting a light path through a TNE port. PXC terminates defect monitoring on a port before it disconnects the light path on that port.





### 5.6 Trace Monitoring

Trace monitoring ensures that a light path is connected to the correct client. A trace (light path Id) is injected by the client in the light path. After setting up a light path, PXC will request TNE to match the supplied trace id with that in the light path. The TNE informs the PXC in case of a mismatch.

Trace Id is a pattern that can be injected and monitored in a light path without affecting its payload. A few different alternative ways of implementing it are possible. Trace Id may be injected in wrapper or as a pilot tone. NTIP supports all of them.

The PXC discontinue the trace monitoring process for a particular light path before deleting it.

## 5.7 Status Request/Response

Status request is initiated by the PXC and is triggered by an NTIP user application, e.g. NMS. Status response contains the status of the requested to TNE ports. It is sent by the TNE in response to status request.

## 5.8 Batching of Messages

To reduce message traffic, TNE can pack several defect notification messages into a single message. Latency could be experienced as a result of batching since TNE has to hold off sending defects for an amount of time necessary to collect enough defects.

## 5.9 Resynchronization

Resynchronization is needed every time an NTIP session restarts. An NTIP session could restart due to TCP connection failure, PXC restart due to internal reasons, e.g. control plan restart or PXC restart, TNE restart, or as a result of a deletion of the NTIP session due to time out.

Each TNE keeps track of its NTIP session. If the session is deleted, it attempts to create another session over TCP/IP periodically. The time it waits before initiating a second attempt is a configurable parameter.

Once TNE succeeds in creating a TCP connection with PXC, it repeats the registration procedure as mentioned in section 5.2. Upon receiving the registration request, the PXC goes into a resynchronization phase requesting an update on the port status of all TNE ports that it is interested in. The PXC confirms the end of the synchronization phase to the TNE when it receives the status of all the requested ports.

Resynchronization helps the reporting of failure events that would have been missed by the PXC due to the NTIP session being down. It





also allows the TNE not to remember the defects monitoring commands given before resynchronization.

## 5.10 TNE to PXC Transport

A single high availability router/switch is recommended for connecting TNE to PXC. This transport arrangement is referred to as NTIP transport network. TNE defect messages are required to reach the PXC with little delays. It is recommended that the NTIP network supports the priority handling of packets, e.g. differentiated services. Messages related to defect reporting are transported with high priority. All other messages, that are not time critical, are transported using a lower priority.

NTIP messages are transported reliably using TCP as the transport protocol. The use of TCP also ensures in-sequence delivery of NTIP messages, hence relieving the protocol developer from creating an additional layer for reliable delivery.

Figure 3 shows the logical connectivity between PXC and TNE.

```
                  +-----+                             +-----+
                  |     |   +---------------+         +----+   |
                  |     |   |             |-------+----+ |--+
                  | PXC |---|  IP Network |-------|    |-+
                  |     |   |             |       +----+
                  |     |   +-------------+         TNEs
                  +-----+
                     |<---------------------------->|
                            TCP Session
```

FIGURE 3: PXC-TNE Logical Connectivity

Figure 3 emphasizes the fact that a PXC communicates with several TNEs while a TNE only communicates with a single PXC.

## 5.11 Defect Types

A TNE will report the following defects to PXC.

- Signal Degrade (SD): TNE reports this type of failure when the light signal on one of its ports degrades below a configured threshold.

- Signal Fails (SF): TNE reports SF to PXC when the incoming signal fails on one of its ports.

- AIS: TNE reports AIS to PXC when it detects AIS-LINE on one of its ports.

- Trace Mismatch: TNE reports Trace Mismatch when it mismatch is detected by one of its ports between an injected pattern and the expected pattern.





- Equipment Failure: TNE will notify PXC of equipment failure (e.g.

OEO card or laser failure on the transport side) and the port number that suffered the failure.

## 6.0 NTIP and OLI Requirements

The requirements for the interface between PXC and the line system, denoted as OLI (Optical Link Interface), have been discusses in [3]. Most of those requirements have been taken from the earlier version of this draft that was presented back in March during the IETF 50 meeting. In this section we discuss how far NTIP satisfies those requirements.

## 6.1 General OLI Characteristics

General OLI requirements are reliabile, secure, and simple. NTIP meets the reliability and security requirements by employing TCP as its transport mechanism. TCP provide reliable and orderly transmission of NTIP messages. Furthermore it provides a flow control mechanism that allows for bulk transmission of NTIP messages. The TCP window mechanism paces messages for cases where the traffic volume increase for instance due to, for instance, re-synchronization.

Line systems (TNEs) do not usually have much memory, and can only keep a limited amount of state. NTIP achieves simplicity by establishing a master-slave, as opposed to peer, relationship between the PXC and TNEs. This minimizes the amount of states kept at the TNE and makes efficient used of the limited memory available.

## 6.2 OLI Functionality

OLI basic functionality as defined in [3] are: neighbor discovery, control channel maintenance, re-synchronization, connectivity discovery, fault management, and link property information.

Neighbor discovery and control channel maintenance are achieved in NTIP by means of a simple registration and keep alive procedures.

Connectivity discovery has been discussed in section 5.4. It allows for the automatic discovery and mapping between PXC ports and TNE ports.

Fault management and re-synchronization have always been the main focus of NTIP as discussed in a previous revision of this draft. Fault management includes fault notification and trace monitoring, both have originally been introduced in a previous revision of this draft. NTIP introduced the notion of priority handing of fault notification messages to expedite light path recovery and restoration in the order of few tens ms.

Trace monitoring has ingeniously been introduced by NTIP and has been adopted as one of the main requirements of OLI.

Most of the link property information, e.g. SRLG and span length can be configured. If necessary they can be added to NTIP with minor modifications.

## 7.0 NTIP Messages and Procedures

All NTIP messages start with the following two fields:

NTIP Vers:
  2-byte field that contains the NTIP protocol version number.

Type:
  2-byte field that contains the message type

The version number provides the features supported by the other side of the NTIP communication. It is only verified at the establishment of NTIP session, and is used during registration and resynchronization states.

## 7.1 Registration Message

The format of the Registration Message is:

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     NTIP Vers                 |       Type                    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
+                                                               +
|                                                               |
+                 TNE Model Number                              +
|                                                               |
+                                                               +
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Type:
  Type = REG-REQ

TNE Model Number:
  16-byte filed that specifies the TNE model number

Procedure:
Registration Message is sent by the TNE to the PXC at start up. If a PXC receives a Registration Message with a different NTIP Vers than expected it may take one of the following actions:

- Reject the registration request





- Reject the registration request if the received NTIP Vers differs by more than an allowed difference. If the difference is acceptable, then the message set common to the two versions will be supported.

## 7.2 Registration Complete Message

The format of the Registration Complete Message is:

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     NTIP Vers               |       Type                      |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Type:
Type = REG-COMPLETE

Procedure:
The Registration Complete Message is sent by PXC to TNE indicating a successful completion of the registration. The message is sent only during the initial synchronization phase.

## 7.3 KeepAlive Message

The format of the KeepAlive Message is:

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     NTIP Vers               |       Type                      |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Type:
Type = KEEP-ALIVE-REQ

Procedure:
The KeepAlive Message is sent by the TNE to PXC every T-keep-alive-

interval seconds (the default is 60 seconds). If PXC does not receive a KeepAlive Message every T-keep-alive-timeout (t-keep-alive-timeout = 3*T-keep-alive-interval), it takes down the NTIP session.

## 7.4 KeepAlive Response Message

The format of the KeepAlive Response Message is:

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     NTIP Vers                 |       Type                    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Type:





Type = KEEP-ALIVE_RES

Procedure:
The KeepAlive Response Message is sent by PXC to TNE in response to a KeepAlive Message.

## 7.5 Monitor Request Message

The format of the Monitor Request Message is:

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     NTIP Vers                 |       Type                    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|            Length             |       Reserved                |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|        No. of Ports           |       Reserved                |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                         Port Address                          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|AR |DM | TType |MT | Tr Len    |        Reserved               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
~                             Trace ID                          ~
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
~                                                               ~
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

```
   |                         Port Address                          |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |AR |DM | TType |MT | Tr Len    |       Reserved                |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                                                               |
   ~                           Trace ID                            ~
   |                                                               |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Type:
  Type = MON-REQ

No. of Ports:
  2-byte field that contains the number of ports reported in this message.

Port Address:
  4-byte field that is formatted as:

```
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```





```
   |    shelf      |      slot     |    Sub Slot   |    port       |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Alarm Reporting, AR:
  2-bit field that indicates start, stop, or no change to alarm reporting.

Defect Monitoring, DM:
  2-bit field that indicates start, stop, or no change to failure monitoring.

Trace Type, TType:
  4-bit used to indicate the trace type. Possible types are, pilot tone, wrapper, J0 bytes.

Monitor of Trace, MT:
  2-bit field that indicates start, stop, or no change to trace monitoring

Trace Length, Tr Len:
  6-bit field that indicates the length of the monitored trace.

Trace ID:
  Up to 63 bytes. It defines the trace to be monitored. User could treat the Trace ID as ASCII or binary

Procedure:
  The Monitor Request Message is sent by PXC to TNE. After setting a light path through TNE, PXC will send Monitor Request Message indicating the start of defect monitoring and the list of monitored ports. Before disconnecting a light path, PXC sends Monitor Request Message with DM set to stop indicating the end of the defect monitoring process.

  The Monitor Request Message is also used for the start and the stop of alarm reporting and trace monitoring. Upon the start of trace monitoring, the PXC supplies the Trace ID to be compared to that in the light path (say, in SONET J0 bytes, or in wrapper, or pilot tone). Trace ID is not included in the Monitor Request Message when MT is set to stop or no change.

### 7.6 Defect Notification

The format for the Defect Notification Message is:

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     NTIP Vers                 |      Type                     |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|            Length             |      Reserved                 |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```





```
|        No. of Ports           |      Reserved                 |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                          Port Address                         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  FS   |       FT      |       Reserved                        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
~                                                               ~
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                          Port Address                         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  FS   |       FT      |       Reserved                        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Type:
  Type = DEFECT-NOTIFICATION

Failure Status, FS:
  2-bit field that indicates fail or clear.

Failure Type, FT:
  1-byte filed that indicates the failure (defect) type as discussed in section 5.10.

Procedure:
  The Defect Notification Message is sent by the TNE in response to a PXC Monitor Request Message. The Defect Notification Message is sent with the highest possible priority to reach the destination PXC in a timely fashion.

### 7.7 Status Request Message

The format of the Status Request Message is:

```
    0                   1                   2                   3
    0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |     NTIP Vers                 |       Type                    |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |            Length             |       Reserved                |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |        No. of Ports           |       Reserved                |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                           Port Address                        |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   | Tag   |                     Reserved                          |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                                                               |
   ~                                                               ~
   |                                                               |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```





```
   |                           Port Address                        |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   | Tag   |                     Reserved                          |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Type:
  Type = STATUS-REQ

Tag:
  A 4-bit field that is used for message identification

Procedure:
  The Status Request Message is sent by PXC to TNE to solicit the status of some or all of the TNE ports. Status Request Message could also be sent to query the status of a previously issued Monitor Request Message.

### 7.8 Status Response Message

The format of the Status Response Message is:

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     NTIP Vers                 |      Type                     |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|            Length             |      Reserved                 |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|        No. of Ports           |      Reserved                 |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                          Port Address                         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Tag   | CStat |    Dyn Stat   |   Reserved                    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
~                                                               ~
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                          Port Address                         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Tag   | CStat |    Dyn Stat   |   Reserved                    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Type:
  Type = STATUS-RESP

Configuration Status, CStat:
  4-bit field that indicates the provisioned state of a TNE port. It indicates whether the port is enabled or disabled.

Dynamic State, Dyn Stat:

1-byte field that indicates the failure type experience by a TNE port.

Procedure:
The Status Response Message is sent from TNE to PXC. A TNE sends a Status Response Message in response to a Status Request Message. For each port requested, the Status Response Message includes a configuration status (enabled or disabled), and a dynamic port defect status that specify the failure type as discussed in section 5.10.

### 7.9 Configuration Update

The format of the Configuration Update Message is:

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     NTIP Vers                 |      Type                     |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|            Length             |      Reserved                 |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|        No. of Ports           |      Reserved                 |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                         Port Address                          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|   CStat       |    Dyn Stat   |   Reserved                    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
~                                                               ~
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                         Port Address                          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|   CStat       |    Dyn Stat   |   Reserved                    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Type:
Type = CONFIG-UPDATE

Procedure:
The Configuration Update Message is sent unsolicited by TNE to PXC. It is used for dynamically modifying the configuration while in operation.

## 8. Security Considerations

This draft does not introduce any new security issues.

## 9. References

1 Bradner, S., "The Internet Standards Process -- Revision 3", BCP 9, RFC 2026, October 1996.

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

3 A. Fredette, Editor, ô Optical Link Interface Requirementsö, draft-many-oli-reqts-00.txt, work in progress, June 2001

## 10. Author's Addresses

Vasant Sahay
Nortel Networks
2305 Mission College Blvd
Santa Clara, CA 95054, USA
Phone: 408-565-4601
E.mail: vasants@nortelnetworks.com

Rajiv Ramaswami
Nortel Networks
2305 Mission College Blvd
Santa Clara, CA 95054, USA
Phone: 408-565-4621
Email: rajiv@nortelnetworks.com

Babu Narayanan
Nortel Networks
2305 Mission College Blvd
Santa Clara, CA 95054, USA
Phone: 408-565-4537
Email: bon@nortelnetworks.com

Osama S. Aboul-Magd
Nortel Networks
P.O. Box 3511, Station C
Ottawa, Ontario, Canada
K1Y 4H7

Phone: 613-763-5827
Email: osama@nortelnetworks.com

Bilel Jamoussi
Nortel Networks
600 Technology Park Drive
Billerica, MA 01821, USA
Phone: 978-288-4506
Email: jamoussi@nortelnetworks.com





Raj Jain
Nayna Networks, Inc.
157 Topaz St
Milpitas, CA 95035, USA
Phone: 408-956-8000 Ext 309
Email: raj@nayna.com

Sudheer Dharanikota
Nayna Networks, Inc.
157 Topaz St
Milpitas, CA 95035, USA
Phone: 408-956-8000 Ext 357
Email: Sudheer@nayna.com

Riad Hartani
Caspian Networks
170 Bytech Drive
San Jose, CA 95134, USA
Phone: 408-382-5216
Email: riad@caspiannetworks.com

Full Copyright Statement