The ATM Forum

Technical Committee

PNNI Routing Congestion Control, Version 1.0

af-cs-0200.000

June 2004

The ATM Forum
Worldwide Headquarters
2570 West El Camino Real, Suite 304
Mountain View, CA 94040-1313
Tel: +1-650-949-6700
Fax: +1-650-949-6705

Preface

During preparation of this addendum, the Control Signaling Working Group was chaired by Gert Oster and Mickey Spiegel. The minutes at related working group meetings were recorded by Dave Paw and Peter Roberts. The editors of this addendum were Gerald Ash and Mostafa Hashem Sherif. The editors would like to thank the following contributors for their help with this addendum as well as all participants of the Control Signaling working group and OSPF/ISIS email list:

Sirak Bahlbi

Gagan Choudhury

Thomas Cornely

Bob Dianda

Jeff Han

Dave Katz

Ghassem Koleyni

Vishwas Manral

Shawn McAllister

Mahmood Noorchashm

Jeff Parker

Carl Rajsic

Peter Roberts

John Rutemiller

Vera Sapozhnikova

Mickey Spiegel

Russ White

Alex Zinin

The editors would also like to thank the many additional contributors who made indirect but significant contributions to this addendum: Margaret Chiosi, Ragu Aswatnarayan, Enrique Cuevas, Tom Helstern, Steve Holmgren, Mike Shapiro, Anurag Maunder, Marty Albright, Sohel Khan, and Greg Wetzel.

This specification uses three levels for indicating the degree of compliance necessary for specific functions, procedures, or coding. They are indicated by the use of key words as follows:

- **Requirement:** "Shall" indicates a required function, procedure, or coding necessary for compliance. The word "shall" used in text indicates a conditional requirement when the operation described is dependent on whether or not an objective or option is chosen.
- **Objective:** "Should" indicates an objective which is not required for compliance, but which is considered desirable.
- **Option:** "May" indicates an optional operation without implying a desirability of one operation over another. That is, it identifies an operation that is allowed while still maintaining compliance.

Table of Contents

1. Introduction

1.1 Scope [Normative]

This document is an optional addendum to PNNI 1.1 [af-pnni-0055.002]. A device supporting PNNI 1.0 [af-pnni-0055.000] may implement the functionality defined in this addendum by treating this addendum as if it were an optional addendum to PNNI 1.0 and PNNI 1.0 Errata and PICS [af-pnni-0081.000].

The document contains the specification for the support of PNNI Routing Congestion Control. Based on overload and failure experience with link-state protocols, this addendum identifies means to enable PNNI Routing protocols to:

- avoid getting into congestion states wherever possible
- respond gracefully to network overloads and failures
- gracefully recover from massive loss of topology database information

The proposed mechanisms in this specification would facilitate:

- detecting and notifying congestion
- reducing the rate of flooding of PTSEs
- maintaining link adjacencies, and
- prioritizing treatment of Hello and PTSE ack packets

Unless otherwise specified by SHALL terminology, the congestion mechanisms described are suggested guidelines that can be adjusted as needed by specific implementations. Since there are many ways to implement a real-time system, specific thresholds and procedures cannot be specified for every implementation. Rather, guidance and default parameters are defined, which are intended to guide implementation.

1.2 Background and Overview [Informative]

Congestion in control protocols can arise in data networks for many different reasons [att, pappalardo1, pappalardo2]. There is evidence based on previous failures that link-state protocols such as PNNI cannot recover from large failures that can result in widespread loss of topology database information. The problem is aggravated when the number of nodes in a peer group is large, which can then lead to an overload in the flooding of topology database information.

Link-state (LS) protocols typically use topology-state update (TSU) mechanisms to build the topology database at each node, typically conveying the topology status through flooding of TSU messages containing link, node, and reachable-address information between nodes. In PNNI, such mechanisms use the PNNI topology state element (PTSE); in OSPF, they use the link state advertisement (LSA); in frame-relay and proprietary-routing networks, they may use other TSU mechanisms to exchange topology status information to build the topology database at each node. Topology information in PNNI is distributed in PTSEs, which are encapsulated in PNNI topology state packets (PTSPs) and periodically flooded to other nodes in the domain through all available links. In some instances of network overload, failure, and/or congestion, redundancy of the flooding mechanism can overwhelm the routing control processors and bring the network down. In this Section we use a generic term – TSU – so as to cover LS protocols in general. In Section 2 and the following Sections, we refer specifically to PTSE and PTSP.

Earlier papers identified issues of congestion control and failure recovery for link-state protocol networks, such as PNNI [choudhury1, choudhury2, choudhury3, pappalardo1, pappalardo2]. The capabilities presented in this addendum allow PNNI to

- avoid getting into congestion states wherever possible, and
- respond gracefully to network overloads and failures.

The need for graceful recovery mechanisms is illustrated below, where we discuss failure experience in which all topology database information is lost in all nodes in a large network, and the link-state protocol is unable to recover.

If a node floods control packets to its peers and has no congestion avoidance and control mechanisms, it can overwhelm its peers and cause congestion. To prevent this we need mechanisms to avoid going into congestion and to recover when congestion occurs. Congestion avoidance prevents the PNNI domain from entering a state of congestive collapse with control traffic. Congestive collapse is a situation where, although the network links are being heavily utilized with an overload of control messages, very little useful work is being done because the nodes are overwhelmed with non-useful work.

Congestion control and avoidance consists of automatic mechanisms to reduce control traffic to avoid congestion, and actions taken if congestion arises. Congestion control mechanisms are defined in this specification, which include:

- detect and notify congestion states between nodes
- reduce the rate of flooding of PTSE
- maintain link adjacencies
- prioritized treatment of Hello and PTSE ack packets

Failure recovery mechanisms are defined in [af-cs-0201.000], which include database backup, PNNI graceful restart for graceful recovery from loss of topology database information, and database resynchronization.

There are examples of serious data network outages in which recovery of the underlying LS protocols was inadequate. For example, in the failure of a large frame relay network [att], an initial procedural error triggered two undetected software bugs, leading to a huge overload of control messages in the network. The result of this control overload was the loss of all topology information, which the LS protocol then attempted to recover using the usual Hello and LS updates. However, the LS protocol was overwhelmed and unable to recover, and manual means had to be used to restart the network after a long outage.

Analysis of the failure has shown that several problems then occurred to prevent the network from recovering properly:

- Very large number of LS updates being sent to every node to process, causing general processor overload
- Route computation based on incomplete topology recovery, causing routes to be generated based on transient, asynchronous topology information and then in need of frequent re-computation
- Inadequate work queue management to allow processes to complete before more work is put into the process queue
- Inability to segment the network (into smaller "areas") to aid in the LS protocol recovery
- Inability to access the node processors with network management commands due to lack of necessary priority of these messages

A more recent occurrence in a large ATM network resulted in an overload of TSUs, and a lengthy network outage [att, pappalardo1, pappalardo2]. Manual procedures were put in place to reduce TSU flooding, which worked to stabilize the network. It is desirable that such TSU flooding reduction be automatic under overload. In general, there have been a number of major outages reported by most major carriers, and routing protocol issues have generally been involved. Other relevant LS-network failures are reported in [cholewka, jander].

Various networks employing LS protocols use various control messages and mechanisms to update the LS database, not necessarily PTSEs, LSAs, or flooding mechanisms. Based on experience, however, the LS protocols including PNNI are found to be vulnerable to loss of topology information, such as occurred in the scenario described above [att], control overload to resynchronize databases, and other failure/overload scenarios which make such networks more vulnerable in the absence of adequate protection mechanisms. Hence we are addressing a generic problem of LS protocols across a variety of implementations, and the basic problem is prevalent in LS protocol networks employing frame-relay, ATM, and IP based technologies. However the solutions given are for PNNI networks alone.

In summary, service providers have suffered a few massive failures of operational networks due to control overloads of LS protocols ('PNNI', ‘OSPF’, etc.). In the instances cited, the LS protocol overwhelmed the network with a control load 'storm' ('PTSE/LSA overload'), which brought the network down, and then prevented its recovery. Fortunately such failures are very rare; however, 'rare' for such events is unacceptable, 'never' is the goal. Such failures are not always the fault of the service provider operation or the vendor/equipment implementation. They are often due to shortcomings in the link-state protocols themselves.

The mechanisms in the specification will help prevent such events from being triggered, and/or provide recovery mechanisms in case such events occur. In Section 2, we describe protocol mechanisms for detecting and notifying congestion, controlling workload once congestion is detected, maintaining link adjacencies, and prioritized processing of Hello and PTSE Acknowledgement packets, and in Section 3 we discuss the procedures for implementing these mechanisms.

1.3 Terminology [Normative]

AvCR_PM	Available Cell Rate Proportional Multiplier
CDV_PM	Cell Delay Variation Proportional Multiplier
CPU	Central Processing Unit
CSI	Congestion State Indication
CTD_PM	Cell Transfer Delay Proportional Multiplier
HWM	High Water Mark
IG	Information Group
IGP	Interior Gateway Protocol
LS	Link State
LSA	Link State Advertisement
LWM	Low Water Mark
OSPF	Open Shortest Path First
PNNI	Private Network-Network Interface
PTSE	PNNI Topology State Element
PTSP	PNNI Topology State Packet
RCI	Resynch Congestion Indication
SPF	Shortest Path First
TSU	Topology State Update

2. Mechanisms for PNNI Routing Congestion Control [Informative]

In the following section, we describe at a general level the following mechanisms

a. detecting congestion and notifying peers,
b. node response when it detects congestion in its neighboring peer node,
c. node response when it detects congestion in its peer group, and
d. maintaining link adjacencies under congestion.

The described mechanisms are based on extensive simulation studies [choudhury2], and are also being investigated for OSPF congestion control [choudhury3]. Procedures for these mechanisms are described in Section 3.

2.1 Detecting Congestion and Notifying Peers

Node congestion can be detected in various ways. For example, local congestion can be inferred at a node by:

- The length of internal work queues
- High processor utilization

Several levels of congestion can be defined, such as high, low, and no congestion, and congestion state changes can be damped by not changing congestion for minimum interval of time, so as to avoid flapping between congestion states.

When a node detects local congestion, it advertises its congestion state to its peers. Since peer nodes perform different actions based on whether they are neighboring peer nodes or non-neighboring peer nodes, two methods of indication are used. The node experiencing local congestion indicates its level of congestion in the PNNI Packet Header of any PTSPs, PTSE Acknowledgement packets, and Hello packets it sends to its neighboring peer nodes and it also indicates its level of congestion in the Nodal Information Group advertised for the logical node.

In addition, neighboring peer congestion can be detected by monitoring the length of the PTSE retransmission list to a neighboring peer node. If the length of the PTSE retransmission list exceeds a threshold, then the neighboring peer node can be considered congested. Multiple levels of congestion and oscillation damping can also apply to neighboring peer congestion.

2.2 Node Response when it Detects Congestion

When a node experiences congestion or detects that its neighboring peer node is experiencing congestion , then it takes the actions to progressively decrease the flooding rate, which can reduce packet drops and PTSE storms. It can do this by increasing the gap between sending successive PTSPs, and also increasing the PTSE retransmission interval, until congestion subsides. Such procedures slow down PTSEs flooded to a congested node and also slow the rate of successive retransmissions of the same PTSE. Slowing PTSE retransmissions is motivated by the observation that during a network congestion event caused by an overload of control messages, a major source for sustaining the congestion is the repeated retransmission of PTSEs. Reducing multiple retransmissions of the same PTSE helps the network get out of the congested state.

Also, a node experiencing local congestion can reduce the rate of SPF computation to reduce CPU overload. Once the congestion subsides, database resynchronization can be initiated using the procedures described in [af-cs-0201.000].

When a node receives an indication that a peer node is experiencing congestion then the node takes actions to reduce the rate of generation of new PTSEs into the peer group by adjusting its significant change variables and increasing its MinPTSEInterval. This action is performed by all peer nodes, not just neighboring peer nodes.

The congestion avoidance measures apply independently to each peer group. That is, if congestion is detected within a peer group, then steps are taken in that peer group, however an LGN does not summarize congestion information from its child peer group. Lower level nodes in child peer groups do not react to the CSI congestion information from a higher level node.

2.3 Maintaining Link Adjacencies under Congestion

Prioritized processing is beneficial for Hello and PTSE Acknowledgement packets, which are critical control messages. Otherwise, delay in processing of Hellos can lead to adjacencies breaking down, and delay in processing acknowledgements can lead to retransmissions, which are conditions that should be avoided if possible. By giving these packets priority treatment in their processing, the chances of adjacencies breaking and retransmissions are lowered.

Link adjacencies with at least one node experiencing congestion should be maintained, even when Hellos are not being originated or processed as frequently as in normal conditions. This is important since exchange of Hellos is significantly less stressful than (re)creating adjacencies. For these reasons, the Inactivity Timer intervals (Hello Interval times Inactivity Factor) used for Hello exchanges should be increased when congestion occurs.

3. Procedures [Normative]

3.1 Changes to PNNI Packet Formats

The level of congestion experienced by a node (high, low, or no congestion) shall be flooded to all peer nodes and be sent to its neighbor nodes. For the purposes of flooding the indication throughout the peer group, a congestion state indication (CSI) is introduced in the nodal information group type of PTSE, as defined in this Section. For the purpose of indicating the congestion to a neighbor node, a CSI is introduced in the PNNI packet header of the PTSP, PTSE Acknowledgement packet, and Hello packet, as defined in this Section.

The following Tables and Sections in the PNNI 1.1 specification SHALL be modified as follows (note that the change marks in this Section are with respect to the PNNI 1.1 Specification:

5.14.9.1.2 The Nodal Information Group

Table 5-36: Nodal Flags

Bit ID:	**bit 8 (MSB)**	**bit 7**	**bit 6**	**bit 5**	**bit 4**	**bits 3,2**	**bit 1**
Bit Name:	“I am Leader” bit	“Restricted Transit” bit	“Nodal Representation” bit	”Restricted Branching” bit	Non-transit for PGL election	Congestion State Indication	Reserved
Description:	value 0: “I am not PGL” value 1: “I am PGL	value 0: “I am a transit node” value 1: “I am a restricted transit node”	value 0: “simple node representation” value 1: “complex node representation”	value 0: “can support additional branch points” value 1: “cannot support additional branch points”	value 0: “normal operation” value 1: “no connectivity through this node for PGL election”	Value 00: “no congestion ” Value 01: “ low congestion ” Value 10: “high congestion ”	

5.14.4 PNNI Packet Header

Table 5-20: The PNNI Packet Header

Offset	**Size (Octets)**	**Name**	**Function/Description**
0	2	Packet Type	See Table 5-21 PNNI Packet Types.
2	2	Packet length	
4	1	Protocol version	Specifies the version according to which this packet was formatted. This specification defines version one of the PNNI routing protocol packet formats.
5	1	Newest version supported	The newest version supported and oldest version supported fields are included in order for nodes to negotiate the most recent protocol version that can be understood by both nodes exchanging a particular type of packet.
6	1	Oldest version supported	See Above.
7	1	*Flags*~~*Reserved*~~	See Table 5-21a PNNI Packet Flags

Table 5-21a: PNNI Packet Flags

Bit ID:	bit 8 (MSB)	bits 7~~..1~~ ,6~~(LSB)~~	bits 5..1 (LSB)
Bit Name:	'Resynch Congestion Indication' (RCI) bit	Congestion State Indication	*Reserved*
Description:	When the packet is type PTSP or PTSE Acknowledgement, set to zero when congestion is detected within the node and one when there is no congestion. Set to zero for other packet types.	When the packet is type PTSP, PTSE Acknowledgement, or Hello set to 00 to indicate "no congestion ", 01 to indicate "low congestion", or 10 to indicate "high congestion".	

The following Sections give procedures for detecting congestion, notifying peers, and node response.

3.2 Detecting and Reacting to Congestion

This section describes routing congestion control procedures, which support the use of the congestion state indications defined in Section 3.1.

3.2.1 Detecting Local Congestion and Notifying Peers

Local congestion SHALL be detected at a node as described in this Section.
Define a 'high water mark (HWMlocal)' and 'low water mark (LWMlocal)' in units of the number of received PTSPs that have not been processed. The following thresholds are defined:

1. high congestion state:
   input work queue > HWMlocal
2. low congestion state:
   HWMlocal >= input work queue >= LWMlocal
3. no congestion state
   input work queue < LWMlocal

A node SHALL notify all peers of its congestion state by advertising a nodal IG with the corresponding bits set. In addition, it shall set the congestion bits in the PNNI Packet Header of any PTSPs, PTSE Acknowledgement packets, and Hello packets sent to its neighbor nodes. A higher congestion state SHALL be held for at least CongestionStateAdvertiseInterval seconds before a lower congestion state can be advertised, even though the state of congestion may have changed before. The congestion state may transition from the high congestion state directly to the no congestion state, if the above threshold is met.

3.2.2 Detecting Neighboring Peer and Aggregate Congestion

A node SHALL implicitly detect congestion at a neighboring peer node based on the number of unacknowledged PTSEs to that node. Neighboring peer congestion SHALL be detected at a node by defining HWMneighbor and LWMneighbor thresholds in units of the number of unacknowledged PTSEs to the neighboring peer. The following thresholds are defined:

1. high congestion state:
   work queue to neighbor > HWMneighbor
2. low congestion state:
   HWMneighbor >= work queue to neighbor >= LWMneighbor
3. no congestion state
   work queue to neighbor < LWMneighbor

The Aggregate Congestion State is defined as the maximum of the congestion level determined by the above thresholds and the level signaled in the CSI bits in the PNNI Packet Header of any PTSPs or PTSE Acknowledgement packets received from that neighboring peer node.

A higher aggregate congestion state SHALL be held for at least CongestionStateAdvertiseInterval seconds before a lower aggregate congestion state is declared, even though the state of aggregate congestion may have changed before the interval expires. The aggregate congestion state may transition from the high congestion state directly to the no congestion state, if the above threshold is met.

3.2.3 Detecting Peer Group Congestion

A node SHALL define the congestion level of the peer group based on the nodal congestion level indicated in the Nodal information groups of nodes in the peer group. The peer group congestion level SHALL be the maximum level indicated in the nodal information groups of all reachable nodes in the peer group. A node in the peer group is considered reachable if there is connectivity to the node as defined for the LoseConnectivityToPGL event in Section 5.10.1.1.3/PNNI 1.1. A higher peer group congestion state SHALL be held for at least CongestionStateAdvertiseInterval seconds before a lower peer group congestion state is declared, even though the state of peer group congestion may have changed before the interval expires. The peer group congestion state may transition from the high congestion state directly to the no congestion state, if the above condition is met.

3.2.4 Congestion Indication and Hierarchy

The CSI congestion avoidance measures apply independently to each peer group, as follows. If congestion is detected within a peer group, then steps SHALL be taken in that peer group. An LGN SHALL NOT summarize CSI congestion information from its child peer group. If there are multiple congested nodes belonging to multiple peer groups on a given switching system, (e.g., a lowest level node in one peer group and an LGN in the parent peer group), then the CSI congestion information SHALL be flooded to each peer group, and the specified congestion avoidance measures taken in each peer group. An LGN will flood the nodal information including CSI for its peer LGNs into its child peer group, based on normal PNNI flooding procedures. However, lower level nodes in child peer groups SHALL NOT react to the CSI congestion information from a higher level node.

3.2.5 Node Response When it Detects Congestion

3.2.5.1 Node Response When it Detects Aggregate Congestion

When a node detects aggregate congestion (see 3.2.2) in its neighboring peer node it SHALL perform the following actions:

1. The rate at which PTSPs are sent to the congested neighboring peer node SHALL be reduced progressively using an exponential backoff mechanism but not below a certain minimum rate. The PTSEs in these PTSPs include those originated at the node, originated by other nodes, and retransmitted PTSEs. At a future time, if the aggregate congestion subsides, then the rate of sending PTSPs should be increased progressively, again using an exponential backoff mechanism. The algorithm is given below:

Let,
G(t) = Gap between sending successive PTSPs at time t on a given adjacency.
T = Minimum time that has to elapse before the existing gap is considered for change.
Gmin = Minimum allowed value of gap.
Gmax = Maximum allowed value of gap.

When low or high aggregate congestion state is detected while the gap timer is disabled, the initial value G(0) SHALL be set to Gmin and the gap timer SHALL be started.

The equation below shows how the gap SHALL be changed after a time T has elapsed since the last change:

$$G(t+T) = \begin{cases} \text{Min}(2G(t), \text{Gmax}) & \text{if node is in high aggregate congestion state at time } t+T \\ G(t) & \text{if node is in low aggregate congestion state at time } t+T \\ G(t)/2 & \text{if node in no aggregate congestion state at time } t+T \end{cases}$$

If G(t)/2 is less than Gmin, the gap timer SHALL be disabled.

2. A node detecting aggregate congestion SHALL use an exponential backoff algorithm for determining the value of the interval at which unacknowledged PTSEs are retransmitted. Let R(i) represent the value used for the i-th retransmission of a PTSE instance since it has been added to the PTSE retransmission list. The node SHALL use the following algorithm to compute R(i)

R(1) = PTSERetransmissionInterval
R(i+1) = Min(2R(i), Rmax) for i >= 1

where Rmax is an architectural variable.

3.2.5.2 Node Response When it Detects Peer Group Congestion

When a node detects peer group congestion (see 3.2.3) it SHALL perform the following action:

To reduce the generation of PTSEs due to resource changes, the significant change variables SHALL be adjusted as follows :

a) for low congestion, set
AvCR_PM = StressAvCR_PM
CDV_PM = StressCDV_PM
maxCTD_PM = StressmaxCTD_PM

MinPTSEInterval = StressMinPTSEIntervalLow, except that the original MinPTSEInterval SHALL apply to deletion of Horizontal Link IGs, deletion of Uplink IGs, deletion of reachable addresses, and changes in Nodal Information.

b) for high congestion, set
AvCR_PM = StressAvCR_PM
CDV_PM = StressCDV_PM
maxCTD_PM = StressmaxCTD_PM
MinPTSEInterval = StressMinPTSEIntervalHigh, except that the original MinPTSEInterval SHALL apply to deletion of Horizontal Link IGs, deletion of Uplink IGs, deletion of reachable addresses, and changes in Nodal Information.

3.2.5.3 Node Response When it Detects Local Congestion

A node detecting local congestion SHALL perform the following actions:

1. SHALL perform all the actions described in Section "3.2.5.1 Node Response When it Detects Aggregate Congestion" for all neighboring peers.
2. If the congested node determines that it needs to recover missing PTSE information and the node supports database resynchronization, the node SHALL initiate database resynchronization with adjacent neighbors, as specified in [af-cs-0201.000]. If the congested node determines that it needs to recover missing PTSE information and the node does not support database resynchronization, the node SHALL initiate database synchronization with adjacent neighbors by generating the event DS mismatch in the neighboring peer FSMs. Note that this causes the node to transition to the Negotiating state. In this case, the node should limit the number of adjacencies synchronizing simultaneously.. 3. (suggested guideline) MAY reduce shortest path computation as follows:
- double shortest path computation interval for low congestion state
- quadruple shortest path computation interval for high congestion state

3.3 Maintaining Link Adjacencies under Congestion

3.3.1 Response to Node Congestion

A node that detects local congestion SHALL increase the Inactivity Timer interval on all associated links, as follows:
- For low congestion, the congested node SHALL set the inactivity factor to StressInactivityFactorLow times InactivityFactor and set the horizontal link inactivity time to StressInactivityFactorLow times HorizontalLinkInactivityTime.
- For high congestion, the congested node SHALL set the inactivity factor to StressInactivityFactorHigh times InactivityFactor and set the horizontal link inactivity time to StressInactivityFactorHigh times HorizontalLinkInactivityTime.

A node that receives congestion indication in the CSI bits in the PNNI packet header of Hello packets SHALL increase the Inactivity Timer interval on that link, as follows:
- For low congestion, the node SHALL set the inactivity factor to StressInactivityFactorLow times InactivityFactor and set the horizontal link inactivity time to StressInactivityFactorLow times HorizontalLinkInactivityTime.
- For high congestion, the node SHALL set the inactivity factor to StressInactivityFactorHigh times InactivityFactor and set the horizontal link inactivity time to StressInactivityFactorHigh times HorizontalLinkInactivityTime.

3.3.2 Control Packet Priority Identification

Hello packets SHALL receive prioritized processing over other PNNI routing packets. PTSE Acknowledgement packets SHALL receive prioritized processing over PNNI routing packets other than Hello packets.

3.4 Feature Compatibility

A node not supporting PNNI routing congestion control will not indicate congestion nor respond to a congestion indication from another node. Having some nodes that do not support PNNI routing congestion control in the peer group may limit the effectiveness of the procedures. It is not expected that such a combination will create additional problems beyond those that appear when there is no support for routing congestion control.

PNNI routing congestion control will operate properly within a PNNI network in which some peer groups entirely support the capability and other peer groups do not support the capability.

3.5 Architectural Variables

These are the architectural variables used in the PNNI specification (to be added to Annex E of PNNI 1.1).

CongestionStateAdvertiseInterval: default value 15 seconds.
: The minimum period of time that a given routing congestion state is advertised before the congestion state can be lowered.

HighWaterMarklocal: default value implementation dependent
: Number of received PTSPs in work queue which have not been processed which indicate high local congestion.

LowWaterMarklocal: default value implementation dependent
: Number of received PTSPs in work queue which have not been processed which indicate low local congestion.

HighWaterMarkneighbor: default value implementation dependent
: Number of PTSEs on the PTSE retransmission list to the neighbor which indicate high neighboring peer congestion.

LowWaterMarkneighbor: default value implementation dependent
: Number of PTSEs on the PTSE retransmission list to the neighbor which indicate low neighboring peer congestion.

Gmin: default value 20 ms.
: Minimum allowed value of gap G(t).

Gmax: default value 1 second
: Maximum allowed value of gap G(t).

Rmax: default value 40 seconds
: Maximum value of PTSE retransmission interval.

StressInactivityFactorLow: default value 2.
: The multiplier to be used to increase the Hello Inactivity time and Horizontal Link Inactivity time during low local congestion.

StressInactivityFactorHigh: default value 4.
: The multiplier to be used to increase the Hello Inactivity time and Horizontal Link Inactivity time during high local congestion and PNNI Graceful Restart.

StressAvCR_PM: default value 75
: The value of AvCR_PM during low or high peer group congestion.

StressCDV_PM: default value 50
: The value of CDV_PM during low or high peer group congestion.

StressMaxCTD_PM: default value 75
: The value of maxCTD_PM during low or high peer group congestion.

StressMinPTSEIntervalLow: default value 15 seconds.
: The minimum interval between updates of any given PTSE during low peer group congestion

StressMinPTSEIntervalHigh: default value 60 seconds.
: The minimum interval between updates of any given PTSE during high peer group congestion

T: default value 1 second
: Minimum time that has to elapse before the existing gap is considered for change.

4. References

4.1 References [Normative]

[af-pnni-0055.000] “Private Network-Network Interface Specification Version 1.0 (PNNI 1.0),” March 1996.

[af-pnni-0055.002] “Private Network-Network Interface Specification Version 1.1 (PNNI 1.1),” April 2002.

[af-pnni-0081.000] “PNNI V1.0 Errata and PICS,” May 1997.

[str-cs-0201.000] “PNNI Routing Resynchronization Control, Version 1,” June 2004.

4.2 References [Informative]

[att] “AT&T announces cause of frame-relay network outage,” AT&T Press Release, April 22, 1998.

[cholewka] Cholewka, K., “MCI Outage Has Domino Effect,” Inter@ctive Week, August 20, 1999.

[choudhury1] Choudhury, G., Maunder, A. S., Sapozhnikova, V., “Faster Link-State Convergence and Improving Network Scalability and Stability,” submitted for presentation at LCN 2001.

[choudhury2] Choudhury, G., Ash, G., Manral, V., Maunder, A., Sapozhnikova, V., "Prioritized Treatment of Specific OSPF Packets and Congestion Avoidance: Algorithms and Simulations," AT&T Technical Report, August 2003.

[choudhury3] Choudhury, G., Ash, G., Manral, V., Maunder, A., Sapozhnikova, V., "Prioritized Treatment of Specific OSPF Packets and Congestion Avoidance," Internet Draft, work in progress.

[jander] Jander, M., “In Qwest Outage, ATM Takes Some Heat,” Light Reading, April 6, 2001.

[pappalardo1] Pappalardo, D., “Can one rogue switch buckle AT&T’s network?,” Network World Fusion, February 23, 2001.

[pappalardo2] Pappalardo, D., “AT&T, customers grapple with ATM net outage,” Network World, February 26, 2001.

Annex A. PNNI Routing Congestion Control MIB [Normative]

```
PNNI-ROUTING-CONGESTION-MIB DEFINITIONS ::= BEGIN

IMPORTS
        MODULE-IDENTITY, OBJECT-TYPE, Integer32, Gauge32,
        NOTIFICATION-TYPE, enterprises
                FROM SNMPv2-SMI
        TEXTUAL-CONVENTION, TimeStamp, TruthValue
                FROM SNMPv2-TC
        MODULE-COMPLIANCE, OBJECT-GROUP, NOTIFICATION-GROUP
                FROM SNMPv2-CONF
        pnniNodeEntry, pnniNbrPeerEntry, pnniMapNodeEntry
                FROM PNNI-MIB;

pnniRtgCongestionMIB MODULE-IDENTITY
    LAST-UPDATED    "200405140000Z"
    ORGANIZATION    "The ATM Forum."
    CONTACT-INFO
        "The ATM Forum
        Presidio of San Francisco
        P.O. Box 29920
        San Francisco, CA 94129-0920 USA
        Phone: +1.415.561-6275
        Fax:   +1.415.561-6120
        info@atmforum.com"
    DESCRIPTION
        "The MIB module for PNNI Routing Congestion Control."
    REVISION        "200405140000Z"
    DESCRIPTION
        "Initial version of the MIB for PNNI Routing Congestion
        Control."
    ::= { atmfPnniRoutingCongestion 1 }

atmForum OBJECT IDENTIFIER ::= { enterprises 353 }
atmForumNetworkManagement OBJECT IDENTIFIER ::= { atmForum 5 }
atmfPnni OBJECT IDENTIFIER ::= { atmForumNetworkManagement 4 }
atmfPnniRoutingCongestion OBJECT IDENTIFIER ::= { atmfPnni 3 }

pnniCgstnMIBObjects OBJECT IDENTIFIER ::= { pnniRtgCongestionMIB 1 }


-- Textual Conventions

PnniRtgCongestionState ::= TEXTUAL-CONVENTION
    STATUS      current
    DESCRIPTION
        "The congestion state of a PNNI routing entity such as a PNNI
         node, an adjacency at a PNNI node, or a peer group as
         perceived by a PNNI node."
    REFERENCE
        "ATM Forum PNNI Routing Congestion Control v1.0
         (af-cs-0200.000)"
    SYNTAX      INTEGER {
                          noCongestion(1),
                          lowCongestion(2),
```

```
                        highCongestion(3)
                        }


-- Node Configuration Group

pCgstnNodeConfigGroup OBJECT IDENTIFIER ::= { pnniCgstnMIBObjects 1 }

pCgstnNodeCfgTable OBJECT-TYPE
    SYNTAX      SEQUENCE OF PCgstnNodeCfgEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "The table containing node configuration variables for PNNI
         routing congestion control."
    ::= { pCgstnNodeConfigGroup 1 }

pCgstnNodeCfgEntry OBJECT-TYPE
    SYNTAX      PCgstnNodeCfgEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "Each entry in this table contains configuration variables for
         PNNI routing congestion control for a given node."
    AUGMENTS    { pnniNodeEntry }
    ::= { pCgstnNodeCfgTable 1 }

PCgstnNodeCfgEntry ::=
    SEQUENCE {
            pCgstnRtgCgstnEnable                TruthValue,
            pCgstnStateAdvertiseInterval        Integer32,
            pCgstnLocalHighWaterMark            Integer32,
            pCgstnLocalLowWaterMark             Integer32,
            pCgstnNbrHighWaterMark              Integer32,
            pCgstnNbrLowWaterMark               Integer32,
            pCgstnGmin                          Integer32,
            pCgstnGmax                          Integer32,
            pCgstnRmax                          Integer32,
            pCgstnStressInactivityFactorLow     Integer32,
            pCgstnStressInactivityFactorHigh    Integer32,
            pCgstnStressAvcrPm                  Integer32,
            pCgstnStressCdvPm                   Integer32,
            pCgstnStressCtdPm                   Integer32,
            pCgstnStressMinPTSEIntervalLow      Integer32,
            pCgstnStressMinPTSEIntervalHigh     Integer32,
            pCgstnT                             Integer32,
            pCgstnLocalCgstnTrapEnable          TruthValue,
            pCgstnNbrPeerImpCgstnTrapEnable     TruthValue
            }

pCgstnRtgCgstnEnable OBJECT-TYPE
    SYNTAX      TruthValue
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION
        "Specifies whether PNNI routing congestion control is enabled
         on this node."
```

```
    DEFVAL { true }
    ::= { pCgstnNodeCfgEntry 1 }

pCgstnStateAdvertiseInterval OBJECT-TYPE
    SYNTAX      Integer32 (1..3600)
    UNITS       "seconds"
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION
        "The minimum period of time that a given routing congestion
         state is advertised before the congestion state can be
         lowered."
    DEFVAL { 15 }
    ::= { pCgstnNodeCfgEntry 2 }

pCgstnLocalHighWaterMark OBJECT-TYPE
    SYNTAX      Integer32 (1..10000)
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION
        "The number of received PTSPs which have not been processed
         that indicate high local congestion."
    ::= { pCgstnNodeCfgEntry 3 }

pCgstnLocalLowWaterMark OBJECT-TYPE
    SYNTAX      Integer32 (1..10000)
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION
        "The number of received PTSPs which have not been processed
         that indicate low local congestion."
    ::= { pCgstnNodeCfgEntry 4 }

pCgstnNbrHighWaterMark OBJECT-TYPE
    SYNTAX      Integer32 (1..10000)
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION
        "The number of PTSEs on the PTSE retransmission list to a
         neighboring peer node that indicate high neighboring peer
         congestion."
    ::= { pCgstnNodeCfgEntry 5 }

pCgstnNbrLowWaterMark OBJECT-TYPE
    SYNTAX      Integer32 (1..10000)
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION
        "The number of PTSEs on the PTSE retransmission list to a
         neighboring peer node that indicate low neighboring peer
         congestion."
    ::= { pCgstnNodeCfgEntry 6 }

pCgstnGmin OBJECT-TYPE
    SYNTAX      Integer32 (1..60000)
    UNITS       "milliseconds"
    MAX-ACCESS  read-create
```

```
    STATUS      current
    DESCRIPTION
        "The minimum initial value of the gap timer that determines the
         gap between sending successive PTSPs on a given adjacency.
         This is the initial value used when the gap timer is started
         after it has been disabled, due to the onset of low or high
         aggregate congestion.  If the rules of the exponential backoff
         mechanism result in an initial value smaller than this value,
         then the backoff timer is disabled."
    DEFVAL { 20 }
    ::= { pCgstnNodeCfgEntry 7 }

pCgstnGmax OBJECT-TYPE
    SYNTAX      Integer32 (1..60000)
    UNITS       "milliseconds"
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION
        "The maximum initial value of the gap timer that determines the
         gap between sending successive PTSPs on a given adjacency."
    DEFVAL { 1000 }
    ::= { pCgstnNodeCfgEntry 8 }

pCgstnRmax OBJECT-TYPE
    SYNTAX      Integer32 (1..3600)
    UNITS       "seconds"
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION
        "The maximum value of the PTSE retransmission interval
         determined using the exponential backoff mechanism when
         aggregate congestion is present."
    DEFVAL { 40 }
    ::= { pCgstnNodeCfgEntry 9 }

pCgstnStressInactivityFactorLow OBJECT-TYPE
    SYNTAX      Integer32 (1..100)
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION
        "The multiplier to be used to increase the local Hello
         inactivity time and the local horizontal link inactivity time
         during low local congestion or when low congestion is
         indicated by the neighbor in Hello packets."
    DEFVAL { 2 }
    ::= { pCgstnNodeCfgEntry 10 }

pCgstnStressInactivityFactorHigh OBJECT-TYPE
    SYNTAX      Integer32 (1..100)
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION
        "The multiplier to be used to increase the local Hello
         inactivity time and the local horizontal link inactivity time
         during high local congestion or when high congestion is
         indicated by the neighbor in Hello packets."
    DEFVAL { 4 }
```

```
    ::= { pCgstnNodeCfgEntry 11 }

pCgstnStressAvcrPm OBJECT-TYPE
    SYNTAX      Integer32 (1..99)
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION
        "The proportional multiplier used in the algorithms that
         determine significant change for AvCR, expressed as a
         percentage, during low or high peer group congestion."
    DEFVAL { 75 }
    ::= { pCgstnNodeCfgEntry 12 }

pCgstnStressCdvPm OBJECT-TYPE
    SYNTAX      Integer32 (1..99)
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION
        "The proportional multiplier used in the algorithms that
         determine significant change for CDV, expressed as a
         percentage, during low or high peer group congestion."
    DEFVAL { 50 }
    ::= { pCgstnNodeCfgEntry 13 }

pCgstnStressCtdPm OBJECT-TYPE
    SYNTAX      Integer32 (1..99)
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION
        "The proportional multiplier used in the algorithms that
         determine significant change for maxCTD, expressed as a
         percentage, during low or high peer group congestion."
    DEFVAL { 75 }
    ::= { pCgstnNodeCfgEntry 14 }

pCgstnStressMinPTSEIntervalLow OBJECT-TYPE
    SYNTAX      Integer32 (1..36000)
    UNITS       "100 milliseconds"
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION
        "The minimum interval between updates of any given PTSE during
         low peer group congestion."
    DEFVAL { 150 }
    ::= { pCgstnNodeCfgEntry 15 }

pCgstnStressMinPTSEIntervalHigh OBJECT-TYPE
    SYNTAX      Integer32 (1..36000)
    UNITS       "100 milliseconds"
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION
        "The minimum interval between updates of any given PTSE during
         high peer group congestion."
    DEFVAL { 600 }
    ::= { pCgstnNodeCfgEntry 16 }
```

```
pCgstnT OBJECT-TYPE
    SYNTAX      Integer32 (1..600)
    UNITS       "100 milliseconds"
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION
        "The minimum time that has to elapse before the existing gap is
         considered for change."
    DEFVAL { 10 }
    ::= { pCgstnNodeCfgEntry 17 }

pCgstnLocalCgstnTrapEnable OBJECT-TYPE
    SYNTAX      TruthValue
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION
        "Specifies whether notifications shall be issued whenever the
         local congestion state changes."
    DEFVAL { false }
    ::= { pCgstnNodeCfgEntry 18 }

pCgstnNbrPeerImpCgstnTrapEnable OBJECT-TYPE
    SYNTAX      TruthValue
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION
        "Specifies whether notifications shall be issued whenever
         there is a change in the implicit congestion state of a
         neighboring peer, as determined by this node based on the
         number of PTSEs sent by this node to the neighboring peer
         node that have not been acknowledged."
    DEFVAL { false }
    ::= { pCgstnNodeCfgEntry 19 }


-- Node Status Group

pCgstnNodeStatusGroup OBJECT IDENTIFIER ::= { pnniCgstnMIBObjects 2 }

pCgstnNodeStatusTable OBJECT-TYPE
    SYNTAX      SEQUENCE OF PCgstnNodeStatusEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "The table containing status information regarding PNNI routing
         congestion control for nodes in this switching system."
    ::= { pCgstnNodeStatusGroup 1 }

pCgstnNodeStatusEntry OBJECT-TYPE
    SYNTAX      PCgstnNodeStatusEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "Each entry in this table contains status information regarding
         PNNI routing congestion control for a given node in this
         switching system."
    AUGMENTS    { pnniNodeEntry }
```

```
    ::= { pCgstnNodeStatusTable 1 }

PCgstnNodeStatusEntry ::=
    SEQUENCE {
            pCgstnNodeStatLclCgstnState     PnniRtgCongestionState,
            pCgstnNodeStatLclLastChange     TimeStamp,
            pCgstnNodeStatUnackedPtsps      Gauge32,
            pCgstnNodeStatPeerGrpCgstnState PnniRtgCongestionState,
            pCgstnNodeStatPeerGrpLastChange TimeStamp
            }

pCgstnNodeStatLclCgstnState OBJECT-TYPE
    SYNTAX      PnniRtgCongestionState
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The local congestion state of the node (subject to the
         congestion state advertise interval).  This value is
         advertised to neighbor nodes in the congestion bits in the
         PNNI Packet Header of PTSPs, PTSE Acknowledgement packets,
         and Hello packets, and is advertised to peers in the
         congestion bits of its nodal information PTSE."
    ::= { pCgstnNodeStatusEntry 1 }

pCgstnNodeStatLclLastChange OBJECT-TYPE
    SYNTAX      TimeStamp
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The time at which the local congestion state of the node last
         changed."
    ::= { pCgstnNodeStatusEntry 2 }

pCgstnNodeStatUnackedPtsps OBJECT-TYPE
    SYNTAX      Gauge32
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The number of PTSPs received by this node that have not been
         processed."
    ::= { pCgstnNodeStatusEntry 3 }

pCgstnNodeStatPeerGrpCgstnState OBJECT-TYPE
    SYNTAX      PnniRtgCongestionState
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The peer group congestion state, as determined by this node
         based on the local congestion level and the congestion level
         indicated in the nodal information groups received from all
         reachable peer nodes."
    ::= { pCgstnNodeStatusEntry 4 }

pCgstnNodeStatPeerGrpLastChange OBJECT-TYPE
    SYNTAX      TimeStamp
    MAX-ACCESS  read-only
    STATUS      current
```

```
    DESCRIPTION
        "The time at which the peer group congestion state last
         changed."
    ::= { pCgstnNodeStatusEntry 5 }


-- Neighboring Peer Status Group

pCgstnNbrPeerGroup OBJECT IDENTIFIER ::= { pnniCgstnMIBObjects 3 }

pCgstnNbrPeerTable OBJECT-TYPE
    SYNTAX      SEQUENCE OF PCgstnNbrPeerEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "The table containing status information regarding PNNI routing
         congestion control for adjacencies between nodes in this
         switching system and neighboring peer nodes."
    ::= { pCgstnNbrPeerGroup 1 }

pCgstnNbrPeerEntry OBJECT-TYPE
    SYNTAX      PCgstnNbrPeerEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "Each entry in this table contains status information regarding
         PNNI routing congestion control for a given adjacency between
         a node in this switching system and a neighboring peer node."
    AUGMENTS    { pnniNbrPeerEntry }
    ::= { pCgstnNbrPeerTable 1 }

PCgstnNbrPeerEntry ::=
    SEQUENCE {
            pCgstnNbrPeerAggCgstnState      PnniRtgCongestionState,
            pCgstnNbrPeerAggLastChange      TimeStamp,
            pCgstnNbrPeerImpCgstnState      PnniRtgCongestionState,
            pCgstnNbrPeerImpLastChange      TimeStamp,
            pCgstnNbrPeerRxmtListSize       Gauge32,
            pCgstnNbrPeerCurGapTime         Integer32
            }

pCgstnNbrPeerAggCgstnState OBJECT-TYPE
    SYNTAX      PnniRtgCongestionState
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The aggregate congestion state of the adjacency with the
         neighboring peer node.  This state is determined from the
         implicit congestion state indicated by
         pCgstnNbrPeerImpCgstnState and the congestion level
         indicated in the PNNI Packet Header of PTSPs and PTSE
         Acknowledgement packets received from the neighboring peer
         node."
    ::= { pCgstnNbrPeerEntry 1 }

pCgstnNbrPeerAggLastChange OBJECT-TYPE
    SYNTAX      TimeStamp
```

```
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The time at which the aggregate congestion state of the
         adjacency with the neighboring peer node last changed."
    ::= { pCgstnNbrPeerEntry 2 }

pCgstnNbrPeerImpCgstnState OBJECT-TYPE
    SYNTAX      PnniRtgCongestionState
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The implicit congestion state of the neighboring peer node, as
         determined by this node based on the number of PTSEs sent by
         this node to the neighboring peer node that have not been
         acknowledged.  A higher implicit congestion state shall be
         indicated for at least pCgstnStateAdvertiseInterval seconds
         before a lower implicit congestion state is declared."
    ::= { pCgstnNbrPeerEntry 3 }

pCgstnNbrPeerImpLastChange OBJECT-TYPE
    SYNTAX      TimeStamp
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The time at which the implicit congestion state of the
         neighboring peer node last changed."
    ::= { pCgstnNbrPeerEntry 4 }

pCgstnNbrPeerRxmtListSize OBJECT-TYPE
    SYNTAX      Gauge32
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The number of PTSEs sent by this node to the neighboring peer
         node that have not been acknowledged."
    ::= { pCgstnNbrPeerEntry 5 }

pCgstnNbrPeerCurGapTime OBJECT-TYPE
    SYNTAX      Integer32
    UNITS       "milliseconds"
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The current value of the time interval between sending
         successive PTSPs to the neighboring peer node.  This value
         varies over time based on an exponential backoff mechanism."
    REFERENCE
        "ATM Forum PNNI Routing Congestion Control v1.0
         (af-cs-0200.000), Section 3.2.5.1"
    ::= { pCgstnNbrPeerEntry 6 }


-- Nodal Topology Map Group

pCgstnMapNodeGroup OBJECT IDENTIFIER ::= { pnniCgstnMIBObjects 4 }
```

```
pCgstnMapNodeTable OBJECT-TYPE
    SYNTAX      SEQUENCE OF PCgstnMapNodeEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "A table containing PNNI routing congestion control information
         learned by the local node from nodal information PTSEs.
         This entire object is read-only, reflecting the fact that
         the map is discovered dynamically during operation of the
         PNNI protocol rather than configured."
    ::= { pCgstnMapNodeGroup 1 }

pCgstnMapNodeEntry OBJECT-TYPE
    SYNTAX      PCgstnMapNodeEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "An entry in the table, containing PNNI routing congestion
         control information about a node in the PNNI routing domain,
         as seen from the perspective of a logical node in this
         switching system."
    AUGMENTS    { pnniMapNodeEntry }
    ::= { pCgstnMapNodeTable 1 }

PCgstnMapNodeEntry ::=
    SEQUENCE {
            pCgstnMapNodeCgstnState         PnniRtgCongestionState
            }

pCgstnMapNodeCgstnState OBJECT-TYPE
    SYNTAX      PnniRtgCongestionState
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "Indicates the congestion state advertised by the originating
         node in its nodal information group."
    ::= { pCgstnMapNodeEntry 1 }


-- Routing Congestion Control Traps

pnniCgstnMIBTrapsPrefix OBJECT IDENTIFIER
    ::= { pnniRtgCongestionMIB 2 }

pnniCgstnMibTraps OBJECT IDENTIFIER ::= { pnniCgstnMIBTrapsPrefix 0 }

pCgstnLclCgstnNotif NOTIFICATION-TYPE
    OBJECTS     {
                  pCgstnNodeStatLclCgstnState
                  }
    STATUS      current
    DESCRIPTION
        "A pCgstnLclCgstnNotif notification is sent when enabled and
         the local congestion state (subject to the congestion state
         advertise interval) of the node changes."
    ::= { pnniCgstnMibTraps 1 }
```

```
pCgstnNbrPeerImpCgstnNotif NOTIFICATION-TYPE
    OBJECTS      {
                  pCgstnNbrPeerImpCgstnState
                  }
    STATUS       current
    DESCRIPTION
        "A pCgstnNbrPeerImpCgstnNotif notification is sent when enabled
         and the implicit congestion state (subject to the congestion
         state advertise interval) of a neighboring peer node changes."
    ::= { pnniCgstnMibTraps 2 }


-- Conformance Information

pnniCgstnMIBConformance OBJECT IDENTIFIER
    ::= { pnniRtgCongestionMIB 3 }

pnniCgstnMIBCompliances OBJECT IDENTIFIER
    ::= { pnniCgstnMIBConformance 1 }

pnniCgstnMIBGroups OBJECT IDENTIFIER
    ::= { pnniCgstnMIBConformance 2 }

-- Compliance Statements

pnniCgstnMibCompliance MODULE-COMPLIANCE
    STATUS       current
    DESCRIPTION
        "The compliance statement for entities which implement PNNI
         Routing Congestion Control Version 1.0."
    MODULE       -- this module
    MANDATORY-GROUPS
        {
          pnniCgstnMIBGeneralGroup,
          pnniCgstnMIBLclCgstnGroup,
          pnniCgstnMIBNbrPeerCgstnGroup,
          pnniCgstnMIBPeerGrpCgstnGroup,
          pnniCgstnMIBHelloGroup,
          pnniCgstnMibNotifMandatoryGroup
          }

    ::= { pnniCgstnMIBCompliances 1 }


-- Units of Conformance

pnniCgstnMIBGeneralGroup OBJECT-GROUP
    OBJECTS {
              pCgstnRtgCgstnEnable
              }
    STATUS       current
    DESCRIPTION
        "A collection of general objects required for PNNI routing
         congestion control. "
    ::= { pnniCgstnMIBGroups 1 }

pnniCgstnMIBLclCgstnGroup OBJECT-GROUP
```

```
    OBJECTS {
              pCgstnStateAdvertiseInterval,
              pCgstnLocalHighWaterMark,
              pCgstnLocalLowWaterMark,
              pCgstnLocalCgstnTrapEnable,
              pCgstnNodeStatLclCgstnState,
              pCgstnNodeStatLclLastChange,
              pCgstnNodeStatUnackedPtsps
              }
    STATUS      current
    DESCRIPTION
        "A collection of objects required for configuration and
         monitoring of local congestion."
    ::= { pnniCgstnMIBGroups 2 }

pnniCgstnMIBNbrPeerCgstnGroup OBJECT-GROUP
    OBJECTS {
              pCgstnNbrHighWaterMark,
              pCgstnNbrLowWaterMark,
              pCgstnGmin,
              pCgstnGmax,
              pCgstnRmax,
              pCgstnT,
              pCgstnNbrPeerAggCgstnState,
              pCgstnNbrPeerAggLastChange,
              pCgstnNbrPeerImpCgstnState,
              pCgstnNbrPeerImpLastChange,
              pCgstnNbrPeerRxmtListSize,
              pCgstnNbrPeerCurGapTime
              }
    STATUS      current
    DESCRIPTION
        "A collection of objects required for configuration and
         monitoring of neighboring peer congestion and the response
         to neighboring peer congestion."
    ::= { pnniCgstnMIBGroups 3 }

pnniCgstnMIBNbrPeerOptionalGroup OBJECT-GROUP
    OBJECTS {
              pCgstnNbrPeerImpCgstnTrapEnable
              }
    STATUS      current
    DESCRIPTION
        "A collection of optional objects for configuration and
         monitoring of neighboring peer congestion and the response
         to neighboring peer congestion."
    ::= { pnniCgstnMIBGroups 4 }

pnniCgstnMIBPeerGrpCgstnGroup OBJECT-GROUP
    OBJECTS {
              pCgstnStressAvcrPm,
              pCgstnStressCdvPm,
              pCgstnStressCtdPm,
              pCgstnStressMinPTSEIntervalLow,
              pCgstnStressMinPTSEIntervalHigh,
              pCgstnNodeStatPeerGrpCgstnState,
              pCgstnNodeStatPeerGrpLastChange
```

```
              }
    STATUS      current
    DESCRIPTION
        "A collection of objects required for configuration and
         monitoring of peer group congestion and the response
         to peer group congestion."
    ::= { pnniCgstnMIBGroups 5 }

pnniCgstnMIBHelloGroup OBJECT-GROUP
    OBJECTS {
              pCgstnStressInactivityFactorLow,
              pCgstnStressInactivityFactorHigh
              }
    STATUS      current
    DESCRIPTION
        "A collection of objects required for configuration and
         monitoring of the response to node congestion in order to
         maintain link adjacencies."
    ::= { pnniCgstnMIBGroups 6 }

pnniCgstnMIBMapNodeGroup OBJECT-GROUP
    OBJECTS {
              pCgstnMapNodeCgstnState
              }
    STATUS      current
    DESCRIPTION
        "A collection of optional objects used to reflect PNNI routing
         congestion status in the map of nodes in the PNNI routing
         domain, as seen by a node in this switching system."
    ::= { pnniCgstnMIBGroups 7 }

pnniCgstnMibNotifMandatoryGroup NOTIFICATION-GROUP
    NOTIFICATIONS {
                    pCgstnLclCgstnNotif
                    }
    STATUS      current
    DESCRIPTION
        "A collection of mandatory notifications used for PNNI routing
         congestion control."
    ::= { pnniCgstnMIBGroups 8 }

pnniCgstnMibNotifOptionalGroup NOTIFICATION-GROUP
    NOTIFICATIONS {
                    pCgstnNbrPeerImpCgstnNotif
                    }
    STATUS      current
    DESCRIPTION
        "A collection of optional notifications used for PNNI routing
         congestion control."
    ::= { pnniCgstnMIBGroups 9 }


END
```

Annex B. Protocol Implementation Conformance Statement (PICS) for PNNI Routing Congestion Control [Normative]

B.1 Introduction

To evaluate conformance of a particular implementation, it is necessary to have a statement of which capabilities and options have been implemented. Such a statement is called a Protocol Implementation Conformance Statement (PICS).

B.1.1 Scope

This document provides the PICS proforma for PNNI Routing Congestion Control, Version, defined in [1], in compliance with the relevant requirements, and in accordance with the relevant guidelines, given in ISO/IEC 9646-7. In most cases, statements contained in notes in the specification, which were intended as information, are not included in the PICS.

B.1.2 Normative References

[1] af-cs-0200.000, PNNI Routing Congestion Control, Version 1, June 2004.

[2] ISO/IEC 9646-1: 1994, Information technology – Open systems interconnection – Conformance testing methodology and framework – Part 1: General Concepts (See also ITU Recommendation X.290 (1995)).

[3] ISO/IEC 9646-7: 1995, Information technology – Open systems interconnection – Conformance testing methodology and framework – Part 7: Implementation Conformance Statements (See also ITU Recommendation X.296 (1995)).

[4] ISO/IEC 9646-3:1998, Information technology – Open systems interconnection – Conformance testing methodology and interconnection – Part 3: The Tree and Tabular Combined Notation (TTCN) (See also ITU telecommunication X.292 (1998)).

[5] af-pnni-0055.002, Private Network-Network Interface Specification Version 1.1 (PNNI 1.1) – April 2002

[6] af-cs-0201.000, PNNI Routing Resynchronization Control, Version 1, June 2004.

B.1.3 Definitions

Terms defined in [1] and [5]

Terms defined in ISO/IEC 9646-1 and in ISO/IEC 9646-7

In particular, the following terms defined in ISO/IEC 9646-1 apply:

Protocol Implementation Conformance Statement (PICS): A statement made by the supplier of an implementation or system, stating which capabilities have been implemented for a given protocol.

PICS proforma: A document, in the form of a questionnaire, designed by the protocol specifier or conformance test suite specifier, which when completed for an implementation or system becomes the PICS.

B.1.4 Acronyms

ASN.1 Abstract Syntax Notation One
ATS Abstract Test Suite
IUT Implementation Under Test
PICS Protocol Implementation Conformance Statement
SUT System Under Test

B.1.5 Conformance

The PICS does not modify any of the requirements detailed in PNNI Routing Congestion Control, Version 1.0. In case of apparent conflict between the statements in the base specification and in the annotations of “M” (mandatory) and “O” (optional) in the PICS, the text of the base specification takes precedence.

The supplier of a protocol implementation, which is claimed to conform to the ATM Forum PNNI Routing Congestion Control, is required to complete a copy of the PICS proforma provided in this document and is required to provide the information necessary to identify both the supplier and the implementation.

B.2 Identification of the Implementation

Identification of the Implementation Under Test (IUT) and system in which it resides (the System Under Test (SUT)) should be filled in so as to provide as much detail as possible regarding version numbers and configuration options.

The product supplier information and client information should both be filled in if they are different.

A person who can answer queries regarding information supplied in the PICS should be named as the contact person.

B.2.1 Date of Statement

____

B.2.2 Implementation Under Test (IUT) Identification

IUT Name: ________________________________

____

____

IUT Version: ________________________________

B.2.3 System Under Test (SUT) Identification

SUT Name: ________________________________

____

Hardware Configuration: ____________________________

____

____

Operating System: ______________________________

B.2.4 Product Supplier

Name: ____________________________________

Address: __________________________________

____

____

Telephone Number: ______________________________

Facsimile Number: ______________________________

Email Address: ______________________________

Additional Information: ______________________________

____

____

B.2.5 Client

Name: ______________________________

Address: ______________________________

____

____

Telephone Number: ______________________________

Facsimile Number: ______________________________

Email Address: ______________________________

Additional Information: ______________________________

____

____

B.2.6 PICS Contact Person

(A person to contact if there are any queries concerning the content of the PICS)

Name: ______________________________

Telephone Number: ______________________________

Facsimile Number: ______________________________

Email Address: ______________________________

Additional Information: ____

____

____

Identification of the Protocol Specification

This PICS proforma applies to the following specification:

af-cs-0200.000, PNNI Routing Congestion Control, Version 1, June 2004.

B.3 PICS Proforma

B.3.1 Global statement of conformance

The implementation described in this PICS meets all of the mandatory requirements of the reference protocol.

[ ] YES

[ ] NO

Note: Answering "No" indicates non-conformance to the specified protocol. Non-supported mandatory capabilities are to be identified in the following tables, with an explanation by the implementor explaining why the implementation is non-conforming.

B.3.2 Instructions for Completing the PICS Proforma

The PICS Proforma is a fixed-format questionnaire. Answers to the questionnaire should be provided in the rightmost columns, either by simply indicating a restricted choice (such as Yes or No), or by entering a value or a set of range of values.

The following notations, defined in ISO/IEC 9647-7, are used for the support column:

Yes supported by the implementation

No not supported by the implementation

The following notations, defined in ISO/IEC 9647-7, are used for the status column:

M mandatory – the capability is required to be supported.

O optional – the capability may be supported or not.

O.i qualified optional – for mutually exclusive or selectable options from a set. "i" is an integer which identifies a unique group of related optional items and the logic of their selection is defined immediately following the table.

A supplier may also provide additional information, categorised as exceptional or supplementary information. These additional information should be provided as items labeled X.<i> for exceptional information, or S.<i> for supplemental information, respectively, for cross reference purposes, where <i> is any unambiguous identification for the item. The exception and supplementary information are not mandatory and the PICS is complete without such information. The presence of optional supplementary or exception information should not affect test execution, and will in no way affect interoperability verification. The column labeled 'Reference' gives a pointer to sections of the protocol specification for which the PICS Proforma is being written.

B.4 PICS for the support of PNNI Routing Congestion Control at the PNNI interface

B.4.1 Major Capability at PNNI (MCP)

Item Number	Item Description	Status	Condition for status	Reference	Support
MCP1	Does the IUT support Routing Congestion Control at the PNNI interface?	M			Yes__ No__
Comments:					

B.4.2 Routing Procedures at the PNNI (RPP)

Item Number	Item Description	Status	Condition for status	Reference	Support
RPP1	Does the IUT detect Local congestion at a node as described in Section 3.2.1?	M		3.2.1	Yes__No__
RPP2	Does the IUT notify all peers of its Local congestion state by advertising a nodal IG with the congestion state indication bits encoded as described in Section 3.2.1?	M		3.2.1	Yes__No__
RPP3	Does the IUT notify all neighbor nodes of its Local congestion state by encoding the congestion state indication bits of the PNNI Packet Header, as described in Section 3.2.1, of any PTSP, PTSE Acknowledgement packets, and Hello packets sent to its neighbor nodes?	M		3.2.1	Yes__No__
RPP4	Does the IUT hold a higher Local congestion state for at least CongestionStateAdvertiseInterval seconds before a lower congestion state can be advertised, even though the state of congestion may have changed before?	M		3.2.1	Yes__No__
RPP5	Does the IUT detect Neighboring Peer congestion at a node as described in Section 3.2.2?	M		3.2.2	Yes__No__
RPP6	Does the IUT set the Aggregate Congestion State of a neighboring peer node to the maximum of the level determined from the level of unacknowledged PTSEs to that node and the level signalled in the CSI bits in the PNNI packet header of any PTSP or PTSE Acknowledgement packets received from that node?	M		3.2.2	Yes__No__

Item Number	Item Description	Status	Condition for status	Reference	Support
RPP7	Does the IUT hold a higher Aggregate Congestion State of a neighboring peer node for at least CongestionStateAdvertiseInterval seconds before a lower congestion state can be declared, even though the state of congestion may have changed before?	M		3.2.2	Yes__No__
RPP8	Does the IUT detect Peer Group congestion at a node as described in Section 3.2.3?	M		3.2.3	Yes__No__
RPP9	Does the IUT hold a higher Peer Group congestion state for at least CongestionStateAdvertiseInterval seconds before a lower congestion state can be declared, even though the state of congestion may have changed before?	M		3.2.3	Yes__No__
RPP10	If congestion is detected within a peer group, does the IUT take steps in that peer group?	M		3.2.4	Yes__No__
RPP11	If the IUT is an LGN, does it NOT summarize CSI congestion information from its child peer group?	M		3.2.4	Yes__No__
RPP12	If there are multiple congested nodes belonging to multiple peer groups on a given switching system, (e.g., a lowest level node in one peer group and an LGN in the parent peer group), then does the IUT flood the CSI congestion information to each peer group, and the specified congestion avoidance measures taken in each peer group?	M		3.2.4	Yes__No__
RPP13	If the IUT is a lower level node in child peer group, does it NOT react to the CSI congestion information from a higher level node?	M		3.2.4	Yes__No__
RPP14	If the IUT detects aggregate congestion, does it reduce progressively the rate at which PTSPs are sent to the congested neighboring peer node using the algorithm described in Section 3.2.5.1?	M		3.2.5.1	Yes__No__
RPP15	If the IUT detects a subsidence of the aggregate congestion, does it increase progressively the rate at which PTSPs are sent to the congested neighboring peer node using the algorithm described in Section 3.2.5.1?	M		3.2.5.1	Yes__No__

Item Number	Item Description	Status	Condition for status	Reference	Support
RPP16	If the IUT detects aggregate congestion, does it use the exponential backoff algorithm described in Section 3.2.5.1 for determining the value of the interval at which unacknowledged PTSEs are retransmitted?	M		3.2.5.1	Yes__No__
RPP17	If the IUT detects peer congestion, does it reduce the generation of PTSEs due to resource changes by adjusting the significant change variables as follows: a) for low congestion, set AvCR_PM = StressAvCR_PM CDV_PM = StressCDV_PM maxCTD_PM = StressmaxCTD_PM MinPTSEInterval = StressMinPTSEIntervalLow, except that the original MinPTSEInterval SHALL apply to deletion of Horizontal Link IGs, deletion of Uplink IGs, deletion of reachable addresses, and changes in Nodal Information?	M		3.2.5.2	Yes__No__
RPP18	If the IUT detects peer congestion, does it reduce the generation of PTSEs due to resource changes by adjusting the significant change variables as follows: b) for high congestion, set AvCR_PM = StressAvCR_PM CDV_PM = StressCDV_PM maxCTD_PM = StressmaxCTD_PM MinPTSEInterval = StressMinPTSEIntervalHigh, except that the original MinPTSEInterval SHALL apply to deletion of Horizontal Link IGs, deletion of Uplink IGs, deletion of reachable addresses, and changes in Nodal Information?	M		3.2.5.2	Yes__No__
RPP19	If the IUT detects local congestion, does it reduce progressively the rate at which PTSPs are sent to neighboring peer nodes using the algorithm described in Section 3.2.5.1?	M		3.2.5.3	Yes__No__
RPP20	If the IUT detects a subsidence of the local congestion, does it increase progressively the rate at which PTSPs are sent to the neighboring peer nodes using the algorithm described in Section 3.2.5.1?	M		3.2.5.3	Yes__No__
RPP21	If the IUT detects local congestion, does it use the exponential backoff algorithm described in Section 3.2.5.1 for determining the value of the interval at which unacknowledged PTSEs are retransmitted?	M		3.2.5.3	Yes__No__

Item Number	Item Description	Status	Condition for status	Reference	Support
RPP22	If the IUT detects local congestion, and determines that it needs to recover missing PTSE information, does the IUT initiate database resynchronization with adjacent neighbors using the procedures described in [6]?	M	MCP1 of [6]	3.2.5.3	Yes__No__
RPP23	If the IUT detects local congestion, does it reduce the shortest path computation as follows: - double shortest path computation interval for low congestion state - quadruple shortest path computation interval for high congestion state?	O		3.2.5.3	Yes__No__
RPP24	If the IUT detects local congestion, does it increase the Inactivity Timer interval on all associated links, as follows: - For low congestion, sets the inactivity factor to StressInactivityFactorLow times InactivityFactor and sets the horizontal link inactivity time to StressInactivityFactorLow times HorizontalLinkInactivityTime. - For high congestion, sets the inactivity factor to StressInactivityFactorHigh times InactivityFactor and set the horizontal link inactivity time to StressInactivityFactorHigh times HorizontalLinkInactivityTime?	M		3.3.1	Yes__No__
RPP25	If the IUT receives congestion indication in the CSI bits in the PNNI packet header of Hello packets does it increase the Inactivity Timer interval on that link, as follows: - For low congestion, sets the inactivity factor to StressInactivityFactorLow times InactivityFactor and sets the horizontal link inactivity time to StressInactivityFactorLow times HorizontalLinkInactivityTime. - For high congestion, sets the inactivity factor to StressInactivityFactorHigh times InactivityFactor and set the horizontal link inactivity time to StressInactivityFactorHigh times HorizontalLinkInactivityTime?	M		3.3.1	Yes__No__
RPP26	In the IUT do Hello packets receive prioritized processing over other PNNI routing packets?	M		3.3.2	Yes__No__

Item Number	Item Description	Status	Condition for status	Reference	Support
RPP27	In the IUT do PTSE Acknowledgement packets receive prioritized processing over PNNI routing packets other than Hello packets?	M		3.3.2	Yes__No__
Comments:					