PNNI: What It Is, What It Does, & How To Configure It

1. Overview

PNNI is an ATM Forum standard that supports routing and signalling to establish new connections in ATM networks. The hierarchical nature of PNNI provides smooth scalability when growing from small to large or very large networks.

IBM has integrated PNNI into its Control Point thus adding the most advanced routing standard to its private ATM switches. PNNI, which stands for Private Network Node Interface (or Private Network-to-Network Interface), includes protocols for distributing network topology information among its switches. This information is used for selecting optimal paths through the network. PNNI also includes signalling, to establish point-to-point and point-to-multipoint connections across the network.

The document begins by discussing (section 2) routing system alternatives prior to introducing PNNI. Basic PNNI concepts are then introduced in section 3. The focus moves on to a two level hierarchical network example (sections 4 and 5). Single peer group networks (one level PNNI networks) are then discussed (section 6) before resuming the descriptions of hierarchical PNNI networks in greater detail. The document then focuses the configuration of individual PNNI parameters. Finally the interworking between switches that support and do not support the PNNI hierarchy is described.

2. From Flat to Hierarchical Networks

2.1 Flat Networks

In a flat network all switches are in a single cluster, and therefore the topology seen by each switch includes every switch and link in the network. Since the routing function is able to see all details of the network, it is able to compute the best possible path to each destination in the network. Furthermore if the network is controlled by a single organization, it is very simple to administer.

As the network grows, however, so does the overhead due to flooding topology information in the network, and the amount of memory consumed and processing in each switch eventually becomes excessive. At this point the network must be clustered to remain stable and allow for further growth.

2.2 IISP

IISP provides a powerful mechanism for clustering switches into independent groups. Using this method, the different clusters are completely independent, except that calls can still be established across cluster boundaries. The only additional requirement is that prefixes describing addresses that can reached across an IISP link must be configured. This is not a significant problem in environments which are stable and if addresses are assigned such that only a small number of prefixes need to be configured at the IISP links. Aside from the addressing consideration, the different clusters can be independently administered. Because it is static**, a network clustered using IISP is very stable**, even more so than a large flat network.

An inherent drawback to any clustering scheme is that information is hidden. Therefore it is not possible to compute optimal routes beyond the border of a cluster. The IBM implementation provides extensions to the PNNI crankback mechanism (crankback is described later in this paper) for IISP links that significantly reduce this problem such that the likelihood of finding a good path to any destination is increased, despite the lack of routing information.

By hiding information IISP can be used to solve the scalability problem in terms of the number of nodes and links in the network. However if a network becomes too large or dynamic the effort involved in properly configuring reachability information may become a limiting factor.

2.3 PNNI

PNNI provides a hierarchical mechanism for clustering switches that is more dynamic than IISP. Like IISP it is scalable in terms of the number of nodes and links in the network, and furthermore reachability information does not have to be configured to cross cluster boundaries. More information is available to the routing function regarding the state of the network beyond the boundaries of a single cluster of switches and therefore better paths can be found using PNNI compared to IISP, but since information is still hidden they are not as good as can be obtained in a flat network.

The scalability and flexibility of PNNI requires that a routing hierarchy be created. The IBM implementation has the ability to automatically configure the hierarchy. To accomplish this efficiently, as in all large networks the addressing plan must ensure that a switch's ATM address reflects its position in the complete hierarchy. When a group of switches are clustered they must have a common address prefix unique to that cluster. When clusters are further grouped the resulting cluster must also have a (shorter) common and unique address prefix.

2.4 Migration

The safest migration path in terms of network stability is to first partition a single peer group network using IISP, and then later convert IISP links to PNNI links. Regardless of whether IISP or the PNNI hierarchy is used to cluster a network, the addressing plan must ensure that groups of clusters have a unique common prefix. This will reduce the amount of configuration for IISP links, and allow a PNNI hierarchy to be created with very little effort using the IBM implementation.

3. PNNI Hierarchy Introduction

PNNI supports a hierarchy mechanism that allows groups of switches to be clustered, and then clusters of switches to be further clustered, and so on. In PNNI terminology these clusters are called **peer group**s. At the bottom level of the PNNI hierarchy, a peer group is a cluster of real switches. At the next higher level of hierarchy, each lower-level peer group is represented by a **logical group node**. Thus a peer group at this higher level is a cluster of logical group nodes. Each peer group elects a leader from its switches or logical group nodes to be the **peer group leader**. The peer group leader creates the logical group node that represents the peer group.

The job of the logical group node is to provide the switches outside of a peer group with some information regarding that peer group. For example instead of advertising the addresses of all switches that can be reached in a peer group the logical group node will advertise a common prefix of those addresses (this is assuming that the addresses have been assigned such that this is possible). Furthermore switches outside of a peer group do not see the individual switches of the peer group, but rather just the logical group node. In this way the amount of topology information is significantly reduced by the hierarchy.

Going up the hierarchy, the peer groups are called the **parent peer group**s of the next lower level. Similarly, going down the hierarchy the peer groups are called the **child peer group**s of the next higher level.

The logical group node function at a given layer is performed inside the switch in which the peer group leader at the next lower layer is located. Thus in a single physical switch, several logical PNNI nodes may be active. There is a node performing the functions of that switch at the bottom level of the hierarchy called **node 0**. If this node is elected peer group leader, then there will be a node representing the bottom-level peer group in this switch called **node 1**. Node 1 is part of the next higher level peer group, and if it is elected peer group leader there will also be a **node 2** in this switch representing the peer group in which node 1 is located, and so on.

For a node in a switch to be active, it must be configured. Please note that the configuration is normally automatic, but parameters may also be manually configured by the administrator if desired. Node 0 in a switch must always be configured, since a switch is always active at the bottom level of the hierarchy. In a given switch node 1 will only become active if that switch's node 0 is elected peer group leader in the bottom level peer group. A node can only be elected peer group leader if it has a non-zero leadership priority. The node with the highest leadership priority will be elected peer group leader. Therefore if node 0 is configured with a non-zero leadership priority node 1 must be configured, since there is a chance that node 0 will indeed be elected. In this case the IBM implementation automatically configures node 1 with a default configuration based on the parameters of node 0. It is possible to manually override the configuration of any node in a switch.

Each node is configured with a **peer group identifier** (or **peer group ID**). The peer group ID is always encoded as 14 bytes, where the first byte, called the **level**, tells how many bits of the remaining 13 bytes are significant. Those significant bits, which are padded with zeros to fill up the peer group ID, identify the peer group and therefore some mechanism is required to ensure that they are unique. As stated in the ATM Forum's PNNI 1.0 specification: "Peer group identifiers must be prefixes of ATM End System Addresses such that the organization that administers the peer group has assignment authority over that prefix. For example, if an organization is given an n-bit prefix it may assign peer group identifiers with length n or greater, but not less than n." Since to guarantee uniqueness, the peer group ID must be created using the prefix of an ATM address, by default, the IBM implementation chooses the prefix of the ATM address of the switch in which it is located. This configuration is simple and automatic as explained later in this document.

PNNI automatically groups nodes with the same peer group ID into the same peer group. Since the level is part of the peer group ID, it must also be true that nodes in the same peer group are also at the same level. The value of the level decreases as we go up the hierarchy. Thus the level of node 0 is greater than the level of node 1, and the level of node 1 is greater than the level of node 2, and so on.

Adjacent nodes (i.e., nodes with a link between them) exchange their peer group ids in PNNI Hello messages. If the peer group ids match, the 2 nodes are in the same peer group, and they exchange detailed topology information for that peer group. If the peer group ids do not match, the nodes are in different peer groups. In this case these nodes are called **border node**s. Border nodes do not exchange any topology information, since the purpose of clustering is to hide the detailed topology information inside of the peer group boundaries. However border nodes exchange information on their respective active hierarchies. This allows the two adjacent peer groups to determine if they have a common peer group at some higher level, and if so allows them to become neighbors at that level (an SVC is established between the appropriate peer group leaders) and to exchange topology information at that level.

In this document the expressions 'node i' and 'node i subsystem' are used interchangeably. We also use the same label (e.g. a.1.1 or a.2.1) to refer to a switch or to the node 0 subsystem within that switch.

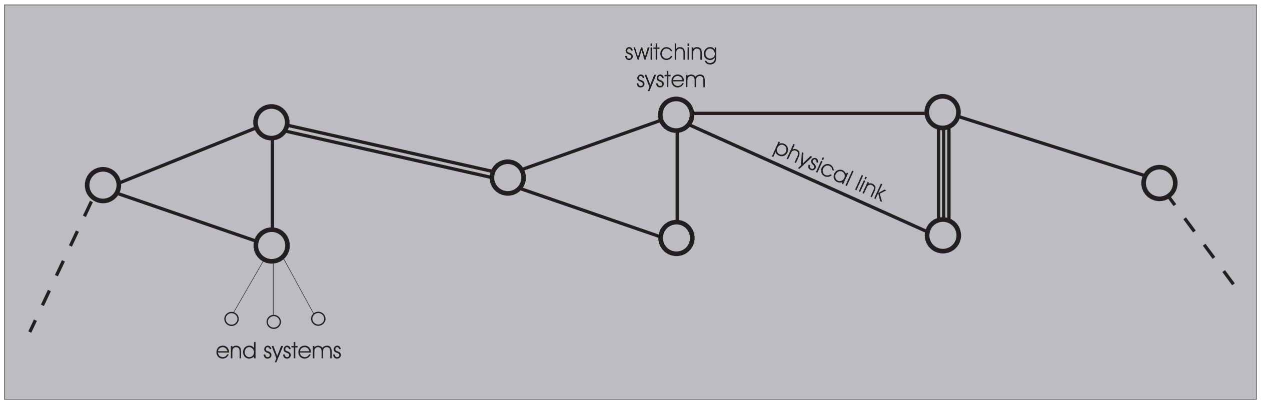

Figure 1: Private ATM Network of Switches and Bidirectional Links

4. Example of a Two-Level Hierarchical PNNI Network

Figure 1 illustrates a physical network of 9 switches and figure 2 illustrates one way of configuring this network into a two level hierarchical PNNI network. The bottom of the hierarchy consists of three peer groups A.1, A.2 and A.3. Peer group A.1 consists of a cluster of three switches represented by node 0 subsystems a.1.1, a.1.2 and a.1.3. Similarly peer group A.2 consists of node 0 subsystems a.2.1, a.2.2, a.2.3 and so on.

Each bottom level peer group has one node 0 that is peer group leader. It's the job of the PNNI system at the switch, supporting a node 0 peer group leader, to create a node 1 subsystem that represents node 0's peer group (peer group containing node 0) in the parent peer group. For example figure 2 shows the node 0 subsystem a.1.3 to be leader of peer group A.1, consequently switch a.1.3 supports a node 1 subsystem a.1 to represent peer group A.1 in the parent peer group A. Similarly the node 1 subsystem a.2 in peer group A represents the child peer group A.2 and node a.3 represents peer group A.3.

The level ids or peer group ID lengths **must** decrease as one goes up the hierarchy. For example the level ID of peer group A must be smaller than the level ID of peer group A.1, the level ID of peer group A.2 and the level ID of peer group A.3.

In the same way that higher level nodes represent clusters of lower level nodes so can higher level links represent groups of lower level links. For example link (a.1, a.2) in peer group A

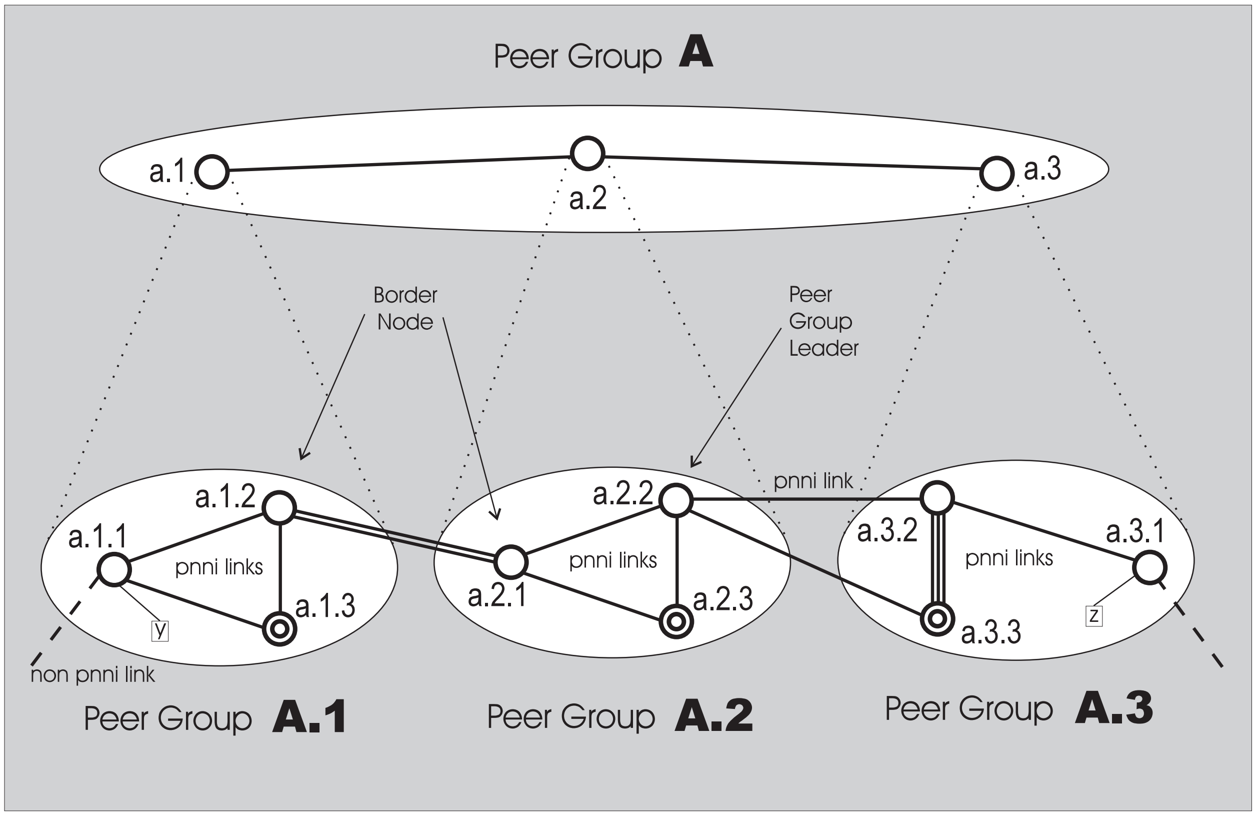

Figure 2: Example of a Two-Layer Hierarchical PNNI Network

represents the two parallel links joining peer groups A.1 with A.2. In a similar way link (a.2, a.3) represents the two links (a.2.2, a.3.2) and (a.2.2., a.3.3) joining peer groups A.2 with A.3.

5. Configuring a Two-Level Hierarchical PNNI Network

We will now go through the steps of configuring the physical network of figure 1 into the two level hierarchical PNNI network represented in figure 2. We assume all switches to be powered up for the first time, i.e. uninitialized. The case where you are reconfiguring an already existing PNNI network is discussed later.

5.1 Default Versus Nondefault Configuration

There are two different alternatives to configuring a given PNNI parameter:

- **Default configuration** where you let PNNI select the value of the parameter (this corresponds to a **plug and play approach**)
- **Nondefault configuration** where you specify the value of the parameter explicitly

As there are multiple parameters to configure at a PNNI switch you have several configuration approaches. You can let PNNI default configure as many parameters as possible or you can nondefault configure every parameter yourself or you can nondefault configure some parameters and let the others be default configured. One parameter that you always must nondefault configure is the switch's private ATM address (see section 5.2).

The advantage of letting PNNI default configure as much as possible is that it minimizes network resources required such as routing related information storage and route computation with a minimum of effort or actions on your part. On the other hand individual network requirements may necessitate the nondefault configuration of certain parameters. This does not necessarily imply loss of PNNI's resource optimization.

The following subsections cover the case where you let PNNI default configure as much as possible. Nondefault configuration is discussed in the sections covering individual parameters.

5.2 Configuring the ATM Address: Basics

When a PNNI switch is powered up for the very first time it automatically generates a bottom level peer group network consisting of only one node 0 subsystem. This node 0 runs with the same private ATM address for every switch, and therefore it is necessary to reconfigure the address to a unique value by issuing, at the switch's console (or via TELNET), the following SET command:

```
8265ATM> set pnni node:0 atm_addr: 13BytePrefix . 6ByteESI.1ByteSel             (01)
```

PNNI effectively executes the SET command after a COMMIT is issued. The COMMIT command and its advantages are described in section 9.

5.3 Configuring Bottom Level or Node 0 Peer Groups

Returning to figure 2, we take you through the steps of configuring peer group A.1.Configure the ATM address of switch a.1.1 by entering at the switch's console:

```
8265ATM> set pnni node: 0 atm_addr: 39.00079999999999990C0AA101.000000000000.00  (02)
```

The 3 dots in the string 39.00079999999999990C0AA101.000000000000.00 are there to clarify only and are not entered (as shown) at the switch console, i.e. they are not part of the entered string. This approach of including dots in address or address prefix strings is applied throughout this document.

PNNI actually executes the SET command only after a COMMIT is issued. The COMMIT command is explained section 9. From now on when we mention a SET command, a subsequent COMMIT is always assumed without explicit mention.

Returning to (02), the first byte is the Authority & Formats Identifier. The value 39 represents the Data Country Code Format which cause the next two bytes 0007 to be interpreted as Country Code. The 13 bytes 39.00079999999999990C0AA101 uniquely identify the switch and 000000000000 identifies the switch's own end system, i.e. its Control Point.

If that is the only parameter you configure then PNNI will default reconfigure for you the peer group ID of the switch's node 0 to:

```
39.00079999999999990C0AA1                                                (03)
```

It is the 96 bit (12 byte) prefix of the switch's ATM address. The level ID of this peer group is the peer group id's length in bits, i.e. 96.

To ensure that nodes a.1.2 and a.1.3 belong to the same peer group, we choose their switch's address to have the same 96 bit prefix. For example enter:

```
8265ATM> set pnni node: 0 atm_addr:39.00079999999999990C0AA102.000000000000.00   (04)
```

at the console of switch a.1.2 and enter:

```
8265ATM> set pnni node: 0 atm_addr:39.00079999999999990C0AA103.000000000000.00   (05)
```

at the console of switch a.1.3. The three nodes a.1.1, a.1.2, a.1.3 still form three separate peer groups, only by joining them with PNNI links is the peer group A.1 of figure 2 formed. You set up a PNNI link as follows. If, for example, port 1 of module 2 at switch a.1.1 is fiber connected to port 3 of module 4 at switch a.1.3 then you enter:

```
8265ATM> set module 2 connected                                          (06)
8265ATM> set port 2.1 enable pnni                                        (07)
```

at the console of switch a.1.1, and you enter:

```
8265ATM> set module 4 connected                                          (08)
8265ATM> set port 4.3 enable pnni                                        (09)
```

at the console of switch a.1.3.

You may have noticed that the peer group id's last byte (hex A1) was chosen identical to the peer groups name A.1. This is just a help in associating peer group names to peer group ids, nothing more.

Configuring the addresses of the next group of switches a.2.1, a.2.2, and a.2.3 to have a 96 bit prefix different from the peer group ID in expression (03) is the next step in creating a next bottom level peer group A.2 different from A.1, see figure 2. For example configure the ATM address of the switches a.2.1, a.2.2 and a.2.3 respectively to:

```
39.00079999999999990C0AA201.000000000000.00                              (10)
39.00079999999999990C0AA202.000000000000.00                              (11)
39.00079999999999990C0AA203.000000000000.00                              (12)
```

default configures the next bottom layer A.2 peer group ID to:

```
39.00079999999999990C0AA2                                                (13)
```

Here again, only by joining switches a.2.1, a.2.2 and a.2.3 with PNNI links do you obtain the peer group A.2.This results in two separate PNNI networks one consisting of peer group A.1 and the other of peer group A.2. To merge them into a single network you need to set up at least one PNNI link between them. In the example of figure 2 there are two such PNNI links (a.1.2, a.2.1) in parallel.

To obtain the third bottom layer peer group, we configure the ATM address of switches a.3.1, a.3.2 and a.3.3 respectively to:

```
39.00079999999999990C0AA301.000000000000.00                                  (14)
39.00079999999999990C0AA302.000000000000.00                                  (15)
39.00079999999999990C0AA303.000000000000.00                                  (16)
```

resulting in the last bottom layer A.3 peer group ID:

```
39.00079999999999990C0AA3                                                    (17)
```

Here also, only by joining the switches a.3.1, a.3.2 and a.3.3 with PNNI links, do you obtain the peer group A.3.

5.4 Configuring the Parent Peer Group

When the ATM address is the only parameter you configure then PNNI will default configure the leadership priority of node 0 to the smallest positive integer value, namely 1. This automatically configures a corresponding node 1 subsystem and causes node 0 to begin its peer group leader election algorithm the moment it detects the presence of at least one neighbor within its peer group.

It follows that every node 0 subsystem in figure 2 (a.1.1, a.1.2, a.1.3, a.2.1, a.2.2, a.2.3, a.3.1, a.3.2, a.3.3) activates its peer group leader election algorithm. Since they all have the same default configured leadership priority of 1 the nodes with the largest node ID, in each peer group, become peer group leaders. As node ID is a PNNI internal parameter based on the ATM address it follows that nodes with the largest ATM address become peer group leaders. Thus node 0 subsystem a.1.3 wins peer group leadership in A.1, node 0 subsystem a.2.3 wins peer group leadership in A.2 and node 0 subsystem a.3.3 wins peer group leadership in A.3.

When a node 0 is configured with a positive leadership priority then IBM’s PNNI automatically default configures a node 1 (at the same switch) with leadership priority 0. Then if a node 0 wins peer group leadership it activates the node 1 subsystem in the same switch to represent node 0’s peer group in the parent peer group. Consequently node 0 subsystem a.1.3 activates node 1 subsystem a.1 at switch a.1.3 to represent peer group A.1, node 0 subsystem a.2.3 activates node 1 subsystem a.2 to represent A.2, and node 0 subsystem activates node 1 subsystem a.3 to represent A.3.

A default configured node 1 has its peer group ID set to the node 0 (at the same switch) peer group ID truncated from the right by 8 bits. Consequently the peer group ID of node 1 subsystem a.1 is derived from (03) to:

```
39.00079999999999990C0A                                                      (18)
```

with a length or level ID of 96 - 8 = 88 (11 bytes).

Deriving the peer group ids of node 0 subsystem a.2 and node 0 subsystem a.3 yield the same expression (18). This is no accident. When setting up the addresses of a.1.1, a.1.2 etc. (section 5.3) we wanted nodes a.1, a.2 and a.3 to lie in a same peer group. We therefore intentionally chose all switch addresses to have the same 88 bit prefix (18).

Nodes a.1, a.2 and a.3 become part of the same peer group only when joined by SVCs that correspond to (though not identical with) the links shown in peer group A, see figure 2. These Switched Virtual Circuits are automatically created by PNNI if PNNI links exist between the border nodes of the respective peer groups. They are routed across the corresponding physical network, see figure 1. For example the SVC joining node a.1 to node a.2 might get routed via switches a.1.3, a.1.2, a.2.1, a.2.2 or a.1.3, a.1.2, a.2.1, a.2.3, a.2.2. They are used for carrying PNNI network related topology and address information needed for maintaining the PNNI network.

Node 0 subsystems that are linked to node 0 subsystems belonging to other bottom level peer groups are called border nodes. For example nodes a.1.2 and a.2.1 are border nodes.

All nodes in peer group A have leadership priorities default configured to 0. There is therefore no apparent peer group leader election activity in peer group A and consequently the PNNI hierarchy ends at A.

6. Single Peer Group Networks

If all switches of a network are configured with private ATM addresses having the same prefix and that prefix is selected as the peer group ID at each node 0 then all node 0s have the same peer group ID. If in addition the links joining switches to each other are configured to be PNNI links then all node 0 subsystems form one peer group and hence a single peer group network. A single peer group network belongs to the category of flat networks discussed in section 2.1.

Single peer group networks are the simplest PNNI network structures. They are intended for small networks where geographically and/or organizationally defined subnetworks are not necessary. Single peer group networks are simple to configure; all you have to do is configure switch addresses and PNNI links between switches. Interconnected single peer group networks further require the configuration of IISP links so that connectivity between the single networks is established. The following 3 subsections describe in more detail these configuration issues.

6.1 Creating a Single Peer Group Network

We now configure the three switches a.1.1, a.1.2 and a.1.3 of figure 2 to form a single peer group network. We assume all switches to be powered up for the first time, i.e. uninitialized. The first step is to configure the ATM address of all switches to say:

```
39.00079999999999990C0AA101.000000000000.00                             (19)
39.00079999999999990C0AA102.000000000000.00                             (20)
39.00079999999999990C0AA103.000000000000.00                             (21)
```

These addresses (identical to the ones used in the figure 2 network) are chosen so that all node 0 subsystems have one and the same default peer group ID of:

```
39.00079999999999990C0AA1                                               (22)
```

Consequently when the physical links (a.1.1, a.1.2), (a.1.1, a.1.3) and (a.1.2, a.1.3) are configured to PNNI links (see expressions (06) to (09)), all node 0 subsystems form a single peer group.

This peer group represents a single peer group network because no PNNI links connect it to other peer groups, i.e. the links (a.1.2, a.2.1), see figure 2, are not configured. This network is shown as single peer group network A.1 in figure 3.

For the case where the switch addresses are insufficiently homogeneous to achieve a common peer group ID we can explicitly configure the same peer group ID at every node 0 to obtain the single peer group and hence the single peer group network.

6.2 Creating Multiple Single Peer Group Networks

We can also configure the switches a.2.1, a.2.2 and a.2.3 of figure 2 to form a single peer group network A.2 by configuring the switch addresses to the values given in expressions (10), (11), and (12) respectively, see figure 3. We also configure the switches a.3.1, a.3.2 and a.3.3 to form a single peer group network A.3 by configuring the switch addresses to the values given in expressions (14), (15), and (16) respectively.

In a next step we connect the three networks by configuring the links joining them to be IISP links, see section 2.2. For these IISP links to effectively interconnect the single peer group networks, reachability information (see sections 7.2 and 7.3) needs to be associated with them.

For example entering:

```
8265ATM> set reachable_address:  u.v  39.00079999999999990C0AA1                  (23)
```

at switch a.2.1, informs the single peer group network A.2 that addresses with prefix 39.00079999999999990C0AA1 must be routed over the IISP links (a.2.1, a.1.2) to the single peer group network A.1 (‘u.v’ represents the blade, port tuple to which the IISP link is connected). Similarly entering:

```
8265ATM> set reachable_address:  u.v  39.00079999999999990C0AA2                  (24)
```

at switch a.1.2, informs the single peer group network A.1 that addresses with prefix 39.00079999999999990C0AA2 must be routed over the IISP links (a.1.2, a.2.1) to the single peer group network A.2.

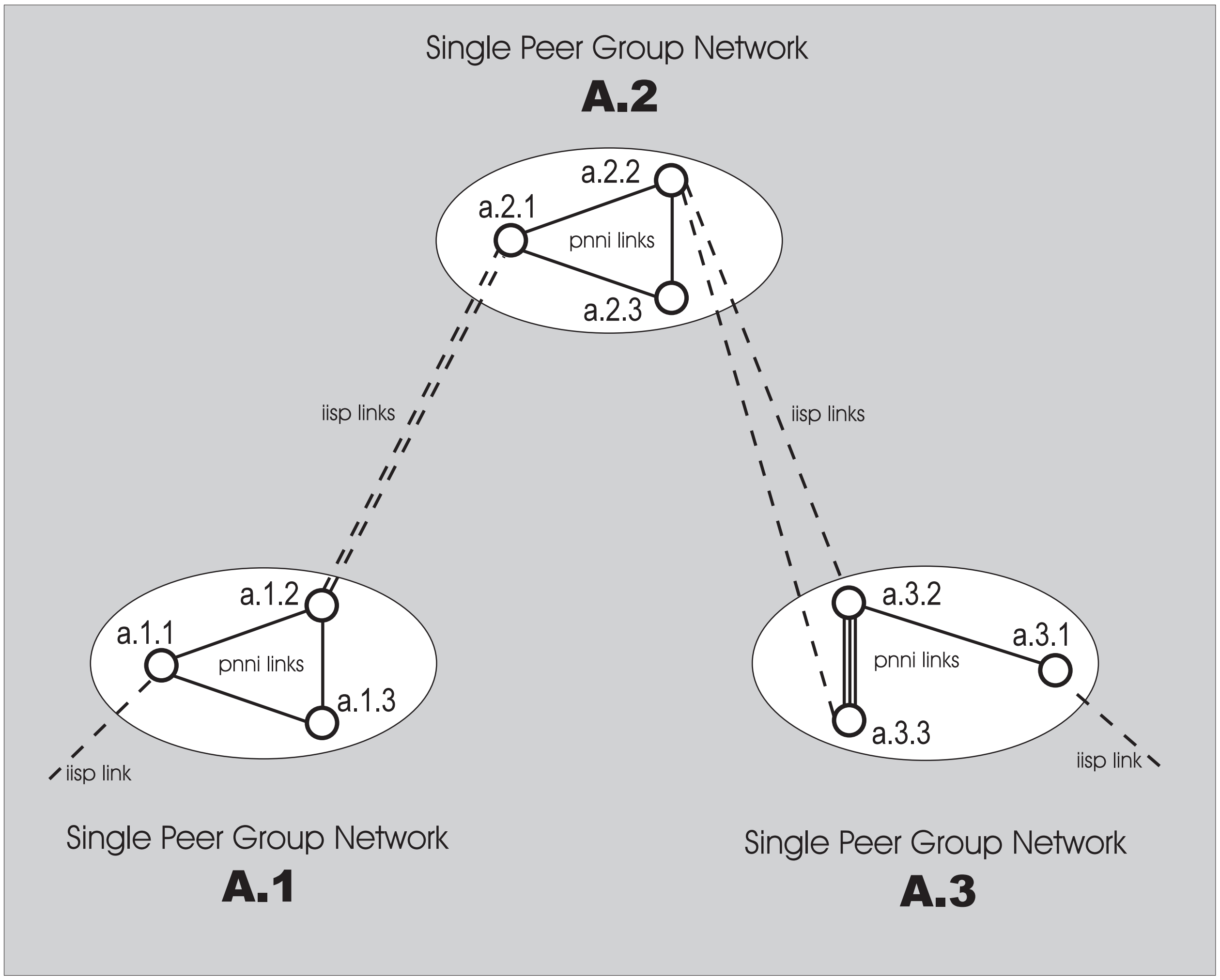

Figure 3: Three Clusters of Switches Configured As Three Separate Single Peer Group Networks

Since network A.2 acts as a relay between networks A.1 and A.3, one also needs to enter:

```
8265ATM> set reachable_address:  u.v  39.00079999999999990C0AA3                  (25)
```

at switch a.1.2, so that addresses with prefix 39.00079999999999990C0AA3 get routed from network A.1 over the IISP links (a.1.2, a.2.1) to network A.2 and on to network A.3 when the IISP links at node a.2.3 have been configured with:

```
8265ATM> set reachable_address:  u.v  39.00079999999999990C0AA3                  (26)
```

The above ‘SET REACHABLE_ADDRESS’ examples, though incomplete, make clear how reachability needs to be associated with IISP links connecting single peer group networks to enhance them into an interconnected network of multiple single peer group networks as discussed in section 2.2.

6.3 Converting Single Peer Group Networks Into One Hierarchical Network

6.3.1 Case Where All Switches Support the Hierarchy

Consider the three separate ‘single peer group’ networks, in figure 3, interconnected by non PNNI links (IISP links).

We assume in the single peer group network A.1 that switches a.1.1, a.1.2, a.1.3 are configured with the addresses specified in expressions (19), (20) and (21) respectively. We assume in the single peer group network A.2 that switches a.2.1, a.2.2, a.2.3 are configured with the addresses specified in expressions (10), (11) and (12) respectively. And we assume in the single peer group network A.3 that switches a.3.1, a.3.2, a.3.3 are configured with the addresses specified in expressions (14), (15) and (16) respectively.

We furthermore assume that all switches support the PNNI hierarchy. Then simply reconfiguring IISP links between networks A.1, A.2 and networks A.2, A.3 into PNNI links, causes the 3 single peer group networks of figure 3 to transform themselves into the two layer hierarchical network of figure 2.

Note that all the statically configured reachabilities associated with the IISP links need to be deconfigured prior to transforming the IISP links into PNNI links. This is because PNNI automatically generates its own set of default reachabilities, see sections 7.2, 7.3 and 7.4.

6.3.2 Case Where Not All Switches Support the Hierarchy

The case where not all switches support the PNNI hierarchy is similar to the case discussed in the previous section 6.3.1 in which the IISP links are converted to PNNI links. But in addition care must be taken to ensure that all border nodes be switches that support the PNNI hierarchy. These switches can also take on peer group leadership functions. Thus in the example of figure 3, switches a.1.2, a.2.1, a.2.2, a.3.2 and a.3.3 must support the PNNI hierarchy.

Let us assume that these five switches are the only ones that support the PNNI hierarchy. Consequently each peer group has at least one switch that does not support the hierarchy. We consequently need to maintain the reachabilities configured with the original IISP links (see section 6.2), that have now become PNNI links, to ensure full connectivity of the switches that do not support the hierarchy.

The reason that switches with no hierarchy support require reachabilites to be associated with certain PNNI links is that routing for these switches is limited to the peer group in which they are located. Associating a reachability (that represents end systems in an adjacent peer group) to the PNNI link, makes the adjacent peer group reachable to the peer group containing the non-hierarchy supporting switch in question.

This subject, i.e. the interworking between switches that support the PNNI hierarchy and that do not, is treated in greater detail in section 14.

7. PNNI Hierarchy Concepts

The PNNI hierarchy is a system that is superimposed on a flat physical network of ATM switches. It consists of successive levels of network abstractions derived from topology aggregations and address summarizations. Its intent is to reduce the amount of stored routing information needed at each switch, to simplify path computation and to increase network stability as well as security.

The successive levels of network abstraction permit for successive reductions in the amount of routing information needed by the path selection component to compute routes.

7.1 Node and Link Aggregation

In the example of figure 2, node a.2 is the **aggregate** representation of nodes a.2.1, a.2.2 and a.2.3. Nodes outside peer group A.2 need only know about a.2 for purposes of routing, i.e. the detailed information about the sub-network a.2.1 and a.2.2 and a.2.3 is not needed.

In a similar way links in the higher level peer group A are aggregate representations of links, at the bottom level, that join peer groups among each other. For example link (a.2, a.3) is the aggregate representation of links (a.2.2, a.3.2) and (a.2.2, a.3.3) joining peer groups A.2 and A.3.

7.2 Address Summarization

In the following discussion we limit ourselves to end systems attached to their respective switches via the UNI interface, that have their private ATM address specified with the help of the ILMI protocol. ILMI ensures that an end system’s address consists of that end system’s individual End System Identifier (MAC address) plus Selector Byte preceded by the 13 byte prefix of the address of the switch to which the end system is attached.

Summarization within the context of PNNI is the notion of reducing addressing information. Since the considered end systems have their 13 byte address prefix in common with that of their switch, it follows that addresses of such end systems can be summarized by their switch’s 13 byte address prefix.

PNNI associates this 13 byte address prefix with every node 0 subsystem. This prefix is the node 0 default internal **summary address**. It represents all end systems whose addresses begin with this prefix. It is used in routing within a peer group to find the switch to which is attached the ‘looked for’ end system. An example of such a summary address is:

```
39.00079999999999990C0AA302                                                        (27)
```

which is the 13 byte address prefix of switch a.3.1 (see expression 14). It represents all end systems that are UNI attached to switch a.3.1 and that have their addresses defined with the help of ILMI.

In a similar way PNNI associates with every node 1 subsystem, a 12 byte prefix of the switch address. This prefix is the node 1 default internal summary address. It summarizes the addresses of all end systems that are attached to switches contained in the peer group that node 1 represents. It is used for routing to nodes in the parent peer group of the peer group that node 1 represents. An example of such a summary address is:

```
39.00079999999999990C0AA3                                                          (28)
```

It represents all end systems that are attached to switches a.3.1, a.3.2 and a.3.3.

Not all ATM end system addresses begin with the 13 byte prefix of the address of the switch to which the end system is attached. Furthermore new summary addresses can be configured via the switch console. These issues are discussed in section 11.8.

7.3 Reachability Information and Its Distribution

Reachability information consists of addresses and address prefixes which describe the destinations to which calls or connections may be routed to. The summary addresses presented in section 7.2 are address prefixes created for the purpose of routing and hence fall into the category of reachability information. Here also we limit ourselves to the class of end systems described in section 7.2. Consequently we limit ourselves to default internal summary addresses.

As we saw in section 7.2 the default summary address associated with a node 0 represents all attached end system addresses whose 13 byte prefix matches that of the switch. This reachability information must be made available to other nodes to be useful for routing. It is therefore distributed or flooded to all other node 0 subsystems in the peer group.

Similarly the default summary address associated with a node 1 represents all end system addresses attached to all the switches in the peer group that node 1 represents with a matching prefix. This reachability information must be flooded to other nodes to be useful for routing. This flooding executes in several steps. First the reachability information is distributed to all other members of the peer group containing node 1. Secondly each node subsystem that receives the information passes it down to all members of the peer group it represents. This down flow continues till it reaches members of bottom level peer groups.

7.4 Reducing Routing Related Storage

Applying the reachability distribution criterion’s of section 7.3 to node a.1.1 of figure 2 results in the Path Selection at switch a.1.1 storing the following reachability information in its route tables:

- Table 1 pertaining to members of peer group A.1:

```
a.1.2’s reachability info: 39.00079999999999990C0AA102                       (29)
a.1.3’s reachability info: 39.00079999999999990C0AA103                       (30)
```

- Table 2 pertaining to members of peer group A:

```
a.2’s reachability info:    39.00079999999999990C0AA2                        (31)
a.3’s reachability info:    39.00079999999999990C0AA3                        (32)
```

These 4 reachabilities is what Path Selection at switch a.1.1 needs to know to identify a destination switch within the network of figure 2. This holds true independently of the number of end systems that are attached to the switches of the network.

The savings in routing related storage increases if we increase the number of switches at each bottom layer peer group from say 3 to 50. Table 1 (expressions 29, 30) then increases from 2 to 49 entries whereas Table 2 (expressions 31, 32) remains unchanged. This results in a total of 51 table entries for the PNNI network as opposed to (49 + 2x50) = 149 entries that would be needed for the corresponding flat or unstructured network of figure 1.

In preparation for the discussion of the next section, we apply the reachability distribution criteria of section 7.3 to node a.3.2 resulting in the Path Selection at switch a.3.2 storing the following reachability information:

- Table 1 pertaining to member of peer group A.3:

```
a.3.1’s reachability info: 39.00079999999999990C0AA302                       (33)
a.3.3’s reachability info: 39.00079999999999990C0AA303                       (34)
```

- Table 2 pertaining to members of peer group A:

```
a.1’s reachability info:    39.00079999999999990C0AA1                        (35)
a.2’s reachability info:    39.00079999999999990C0AA2                        (36)
```

7.5 Reducing Route Computation Time

An accurate treatment of path selection is beyond the scope of this document so that the following discussion is based on important simplifications (no uplinks, no DTLs).

Routing takes advantage of PNNI's topology aggregation, address summarization and reachability distribution to reduce route computation time. We use an example from figure 2 to explain: consider the end system y attached to switch a.1.1 that wants to set-up a connection to end system z attached to switch a.3.1. End system z is assumed attached to its switch via the UNI interface, and its address is assumed specified with the help of the ILMI protocol. Furthermore z's address is say:

```
39.0007999999999990C0AA302.000000000111.00                                  (37)
```

In a first step switch a.1.1's Path Selection does a longest matching (prefix) search between the destination's address (expression 37) and its routing tables (29) to (32). This yields reachability (expression 32) which identifies node a.3. So a.1.1's Path Selection knows that it must route to a.3.

In a second step a.1.1's Path Selection sees in peer group A' that to reach a.3 it must first go to a.2. Since a.2 represents peer group A.2, the Path Selector knows that it must route to its neighbor peer group A.2. Consequently all it has to do is compute a route across its own peer group A.1 to a node bordering on A.2. Node a.1.2 is such a border node, see figure 2.

The connection set-up that reaches node a.1.2 contains the intermediate destination a.2. Since a.2 represents peer group A.2, node a.1.2 forwards the connection to its neighbor in A.2, i.e. to node a.2.1.

The connection set-up that reaches node a.2.1 contains the destination a.3. Since a.3 represents A.3, switch a.2.1's Path Selection knows it must route to its neighbor peer group A.3. Consequently all it has to do is compute a route across its own peer group A.2 to a node bordering on A.3. Nodes a.2.2 is such a border node, see figure 2.

The connection set-up that reaches node a.2.2 contains the destination a.3. Since a.3 represents peer group A.3, node a.2.2 forwards the connection to one of its neighbor nodes in A.3, i.e. to nodes a.3.2 or a.3.3.

The connection set-up that reaches say a.3.2 contains the destination a.3. As a.3 represents peer group A.3, switch a.3.2's Path Selection knows that the destination is in its own peer group. Since it does not know the destination switch it executes a longest matching (prefix) search between the destination's address (expression 37) and its routing tables (expression 33) to (expression 36). This yields expression (33) which identifies node a.3.1. Consequently all a.3.2's Path Selection has to do is compute a route within its own peer group A.3 to switch a.3.1 (see figure 2).

The connection set-up that reaches switch a.3.1 contains the original destination address (expression 37) and hence its End System Identifier 000000000111 which switch a.3.1 uses to locate the end system on which the destination address resides.

This above example shows that route computation repeats at each bottom level peer group that contains a section of the resulting connection, i.e. it is distributed among these peer groups. Each route computation step routes through a single peer group to a neighboring border node or a node representing the destination end system. Route computation time is to a first order approximation proportional to the square of the switch population involved therefore each route computation step is to a first order approximation proportional to the square of the switches in the respective peer group. Consequently (to a first approximation) the gain ratio, or speed up, of one PNNI route computation step as opposed to routing without PNNI hierarchy is $9^2/3^2 = 9$.

7.6 Increasing Network Stability and Security

Peer groups have a partitioning effect that isolates individual peer groups from the rest of the network. This enhances security in that network details apparent within a peer group remain masked outside the peer group.

7.6.1 Topology Changes

The perturbing effects of any topology reconfiguration action, such as adding/removing links and/or switches that occurs within a bottom level peer group is contained in that peer group. It does not propagate to nodes lying outside the affected peer group, i.e. nodes in other peer groups are not impacted. For example, in figure 2, removing link (a.2.1, a.2.3) only affects the peer group A.2. Peer groups A.1, A.3 and A are not affected.

A topology change that affects the border between two peer groups, such as a link change, affects the involved bottom level peer groups as well as all higher level peer groups up to and including the peer group that contains representations of both affected bottom layer peer groups. For example, in figure 2 removing say link (a.2.2, a.3.3) affects peer groups A.2, A.3 and A. It does not affect peer group A.1.

7.6.2 Resource Availability Information Group Changes

PNNI associates Resource Availability Information Groups (RAIGs) to PNNI links and reachability information (reachable addresses or address prefixes). RAIGs include a collection of traffic related parameters or traffic metrics such as maximum cell rate and cell loss ration. Each class of service potentially has its own RAIGs and each RAIG is updated to reflect the state of the represented class of service across a link or of a reachability entity.

RAIG updates that occur within a bottom level peer group are contained in that peer group. They do not propagate to nodes lying outside the affected peer group. For example, in figure 2, a RAIG update for the service class ABR (Available Bit Rate) of the link (a.2.1, a.2.3) only affects the subnetwork identified by peer group A.2. Peer groups A.1, A.3 and A are not affected.

RAIG updates that affects the border between two peer groups, affect the involved bottom level peer groups as well as all higher level peer groups up to and including the peer group that contains representations of both affected bottom layer peer groups. For example, in figure 2, a RAIG update for the service class VBR (Variable Bit Rate) of the link (a.2.2, a.3.3) affects peer groups A.2, A.3 and A. It does not affect peer group A.1.

8. Path Selection Strategies

8.1 Introduction

The role of path selection function is not only to compute a feasible path from a source switch to a destination switch. The computed path must also take into account incoming connection requirements like bandwidth, real time constraints etc. … In the long term, it is also desirable to balance the load on the overall network. This section starts with the definition of some path selection strategies. The strategies used to route connections belonging to the various classes of service defined by PNNI are then discussed.

8.2 Path Selection Mode

Basically there are two modes of performing path selection: *pre-computed* mode and *on-demand* mode.

In *pre-computed* mode the optimal paths from a source switch to *all* the other switches in the network are computed at the same time and stored in *routing tables*. When a route to a given destination is requested, the path is extracted from the routing tables. Of course, routes are automatically recomputed when “significant” changes occur in the network e.g. topology changes or link metric changes. The advantage of pre-computed mode is to be able to provide paths very quickly just by retrieving them from routing tables. Routes are computed using one or more general criteria. This mode cannot be used if the path computation must take into account criteria specific to each entering connection.

In such conditions, it is necessary to use the *on-demand* mode. In this case a path is computed on individual each route request. Of course, this slows down the call processing time, but it computes a path best fitting the connection requirements.

8.3 Path Selection Criteria

Finding a “best path” from a source switch to one or more destination switches is actually a minimization problem. This can be accomplished according to one or more criteria. For the sake of simplicity, let us consider here only two ways of treating this problem: either one looks for the *widest path* or for the *shortest path*.

Computing a *widest path* from a source switch to a destination switch, means finding the path that maximizes a given link parameter. Suppose that this parameter is the available bandwidth on each link. Figure 4 shows a sample network where the bandwidth available on each link is shown. In this example, the widest path from switch 0 to switch 4 is the succession of switches {0, 1, 3, 4}. Since on this path the “narrowest” link is the one between switches 3 and 4, the “size” of this widest path is 20 Mbps. That is, if a connection requires more then 20 Mbps then it cannot be routed. The fact that the widest path has been selected guarantees that there are no other paths in the network capable of routing such a connection.

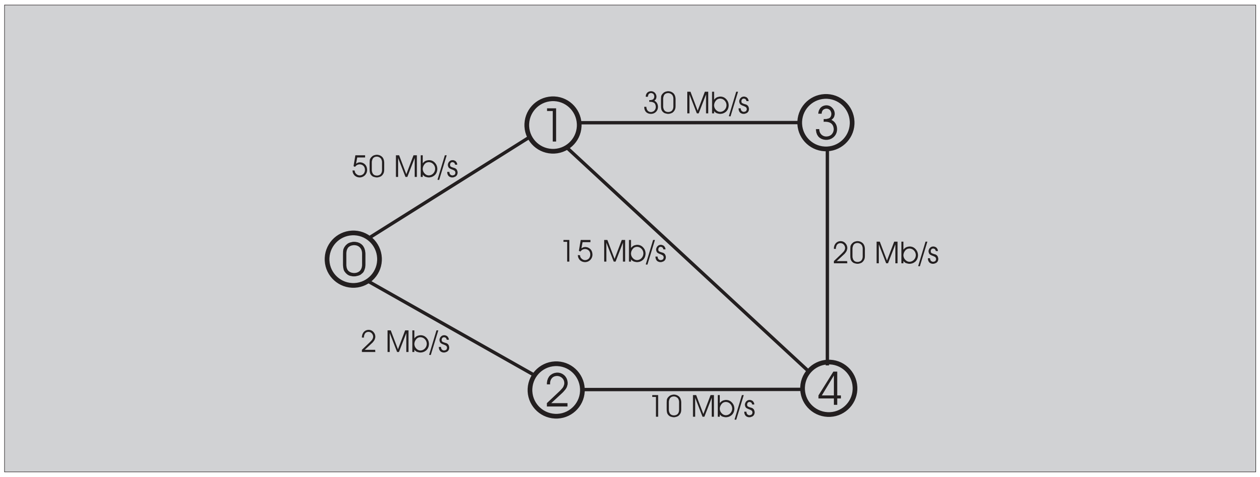

Figure 4: Simple Network Showing Available Bandwidth On Each Link

The simplest way to compute a *shortest path* is to compute this path according to the number of hops. Therefore the best path between two switches, will be the path with the smallest hop count. Let us consider again the example given on Figure 4. Now, the shortest path between switch 0 and switch 4 is the succession of switches {0, 1, 4}. Actually one can notice that there is a further possible shortest path, namely switches {0, 2, 4}. To select the best path it is possible to take into account the available bandwidth. In this case the optimal path would be {0, 1, 4}.

More generally, the shortest path can also be computed according to a weight associated with each links. In this case the shortest path would be the one with the minimal sum of weights. Let us now consider the network of Figure 5 showing the weight of each link. In this case the shortest path from switch 0 to switch 4 is the succession {0, 1, 3, 4}. The total weight is 4. Note that if the weight of all links in the network are equal, computing the shortest path on weights is equivalent to computing the shortest path on the number of hops. In PNNI this weight can defined for each link and is termed *Administrative Weight*.

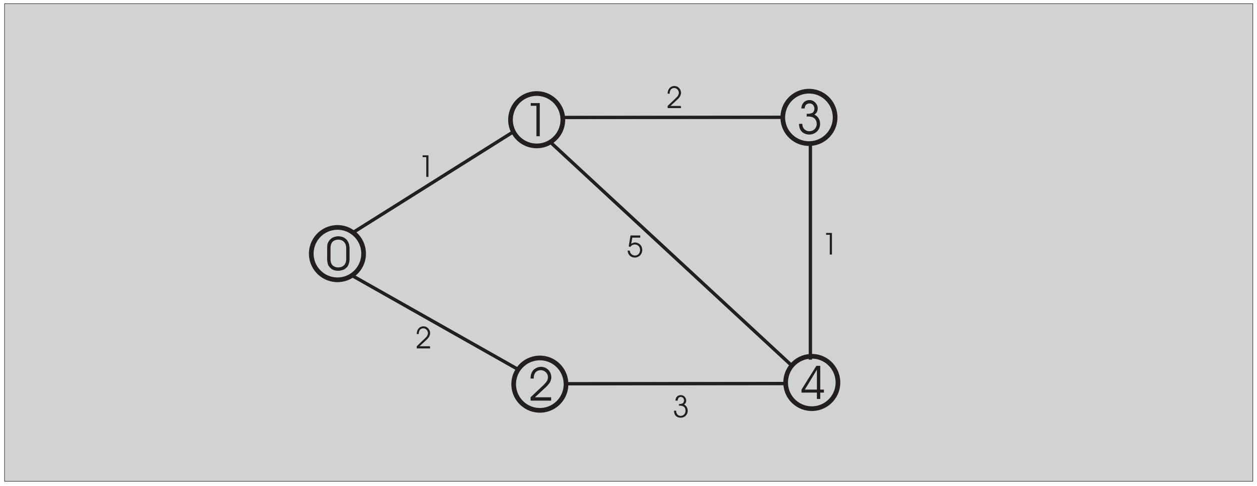

Figure 5: Simple Network Showing Weight of Each Link

8.4 Routing of UBR and ABR Connections

UBR connections are routed by default according to the shortest path based on *Administrative Weight* (AW): the shortest path between two switches is the one with the lowest sum of AW. The reason for this choice is that shortest path allows controlling the length of the selected path and therefore the amount of involved network resources (switching labels…).

UBR shortest path can be used in a pre-computed mode since no bandwidth requirements are given by the entering connection.

ABR connections are a bit more complicated. Actually, such connections may be defined in two ways:

- If the *Minimum Cell Rate* (MCR) – which is the bandwidth that must be guaranteed to the connection – is null then the connection behaves exactly as a UBR connection. The same pre-computed mode is used.
- If the MCR is not null, then this minimum value should be guaranteed to the connection. The computed path must support the requested MCR and therefore the computation should be done on-demand to ensure that a path will be found if one exists.

In specific network environments *load balancing* may be required. To achieve this it is possible to enforce widest path computation for both UBR and ABR connections.

For both UBR and ABR this type of computation can be done in pre-computed mode. In the special case of ABR with MCR not null, the algorithm has to check that the selected path supports the required bandwidth. Since the computed path is the widest one, if this path does not support the requested bandwidth it is obvious that there are no other paths in the network that can support this bandwidth; the connection can therefore be rejected.

The drawback of using widest path is that the path length cannot be controlled.

8.5 Routing of rt-VBR, nrt-VBR and CBR Connections

These types of connections are routed using shortest path on AW. Actually, this decision is motivated by two reasons:

- It is important for connections having real time constraints (rt-VBR, CBR) to reduce as far as possible transmission time and jitter. Therefore the usage of short routes potentially reduces those two parameters.
- VBR and CBR connections are treated as “reserved bandwidth” connections. That is, requested bandwidth is actually reserved in the switching devices and on the link. Reducing the route length contributes to globally reduce the resources allocated in the network.

Since the bandwidth requirements are specific to each entering connection, the path selection algorithm should be run on on-demand mode. The algorithm proceeds as following:

- All the links of the network not supporting the bandwidth requested by the incoming connection are pruned.
- The shortest path according to the AW is computed.
- If there are several shortest paths the one with the minimal number of hops is selected.
- If there are several shortest paths with minimal number of hops, then the one with the bigger available bandwidth is selected.

8.6 Routing of Multicast Connections

Multicast connections must be treated in a special way in order to take advantage of ATM properties. When a multicast connection from a source to several destination switches is established, it is worth to share as much as possible the common links in the paths.

Let us consider the example given in Figure 6. Station A is setting a first multicast connection to station B. Suppose that the route goes through switches {0, 1, 3, 4}. Then a party is added to station C for this same connection. The route from A to C should share as many links as possible with the route from A to B. Actually, on the common links between the two connections cells are not duplicated. That is, if a cell has to be transmitted from A to B and C there will one cell transiting on the common links and this cell will be duplicate at the switch were the paths become disjoined. In this example, cells are duplicated at switch 3.

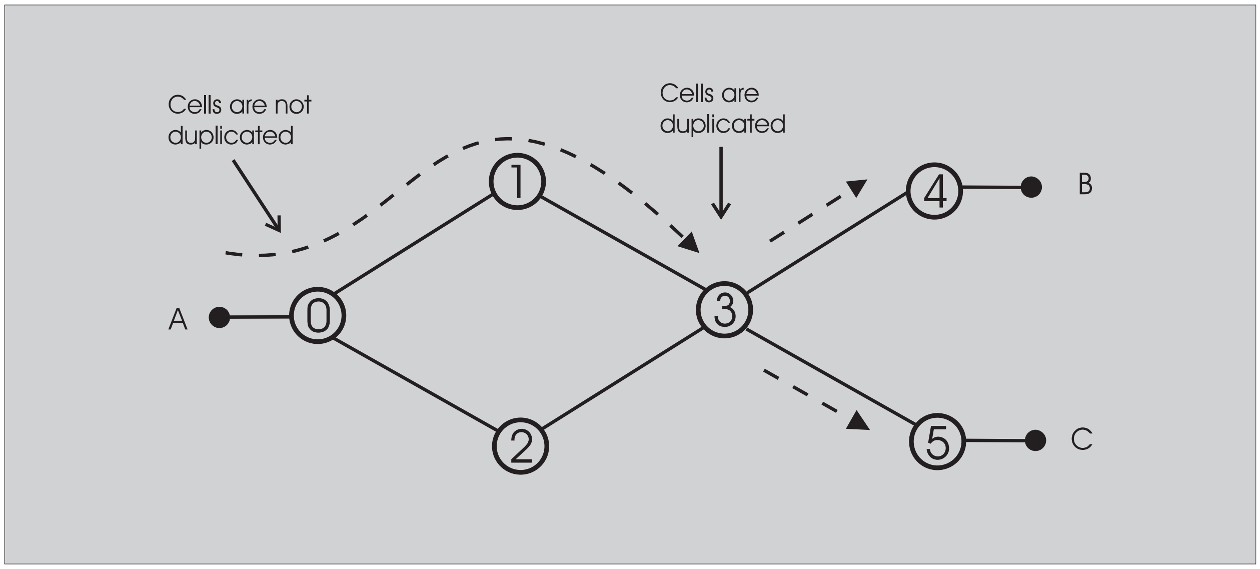

Figure 6: Routing of Multicast Connections

The principle used for achieving this kind of path computation is the following:

- At the first connection the shortest path according to the administrative is computed. The links used for this connection are stored in a dedicated data structure called *multicast tree*.
- When a party is added to the connection, first all the links on the multicast tree are set to be "cost-less" for path computation. Then the shortest path on the AW to the destination party is computed. The additional links used for this new connection are added to the multicast tree.
- When a party is removed, all the links used for the connection to this party are removed from the multicast tree.

Of course, if the whole multicast connection is dropped the multicast tree associated to the multicast connection is deleted.

Setting links as cost-less enforces path selection algorithm to examine first those links when searching the optimal path. This enables the algorithm to find paths having as much common links as possible.

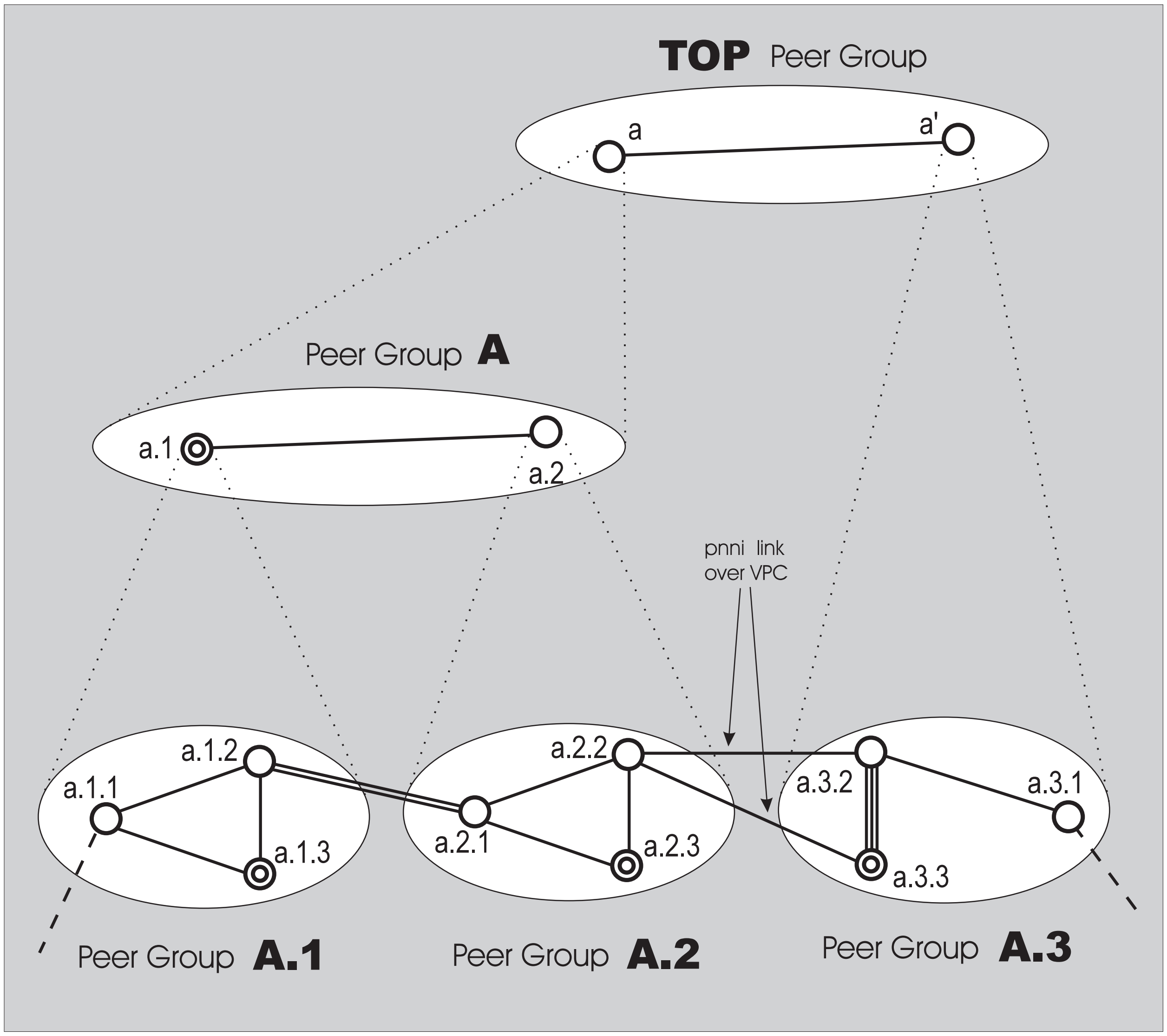

Figure 7: Modifying the Two-Layer Network of Figure 2 Into a Three-Layer Hierarchical Network Without Switch-Address Change

9. Modifying and Extending a PNNI Hierarchical Network

This section is a follow up to section 5. It explains how to reconfigure the network example of figure 2 into a three level hierarchical PNNI network. The focus is on reconfiguration and increasing the hierarchy.

Assume that the peer group A.3 needs to migrate to another location that is joined by two VPCs to the original campus that now only contains peer groups A.1 and A.2. Two alternative approaches are discussed, the first one assumes that addresses of the relocated switches are not changed the second approach assumes switch address changes. Please note that this change in the PNNI hierarchy is not required due to changes in the physical network. This is just an example.

9.1 Approach Without Address Change

Assume you move peer group A.3 to another location, see figure 7, for example to a three story building with a switch on each floor. You want for administrative reasons to maintain same addresses. You also want to join up the ‘moved’ peer group A.3 to the rest of the network at a new higher level of the hierarchy.

You first need to configure a node 2 subsystem a to represent the original campus at the new hierarchy level (third level). Assume you want a switch in peer group A.1 to support this new node 2. Since node a.1 already represents peer group A.1 at the second hierarchy level (peer group A) you naturally configure node 2 at the switch supporting node a.1, i.e. at switch a.1.3. You do this by increasing the leadership priority of node 1 subsystem a.1 by entering at switch a.1.3:

```
8265ATM> set pnni node: 1 leader_priority: 1                                  (38)
```

We could have chosen any other positive value, smaller than or equal to the upper limit 205, for leadership priority since the leadership priority of the only other peer group A member a.2 is 0 (section 5).

Setting the leadership priority of node 1 subsystem a.1 positive causes PNNI to default configure a new node 2 subsystem in switch a.1.3. It also initiates the peer group leader election process at a.1 which activates this node 2 subsystem (denoted by a in figure 7) when the node 1 subsystem a.1 wins peer group leadership.

The default configured node 2 subsystem a has its peer group ID set to the corresponding node 1 subsystem a.1 peer group ID truncated from the right by 8 bits. Consequently a’s peer group is the 80 bit prefix of expression 18:

```
39.00079999999999990C                                                         (39)
```

Node 0 subsystem a.3.3 is peer group leader of A.3, consequently node 1 subsystem a’ at switch a.3.3 is active and should represent A.3 in the new peer group (TOP peer group in figure 7). To achieve this the peer group configured at node a’ must have the same value as that configured at node a, i.e. the value in expression 39. This is not the case when node a’ is default configured since it takes on the peer group ID specified in expression 18. Now expression 39 turns out to be the 80 bit prefix of expression 18 so that you only need to enforce a peer group length reduction to 80 at node a’, i e. a level ID reconfiguration by entering at the switch a.3.3:

```
8265ATM> set pnni node: 1 level_identifier: 80                                (40)
```

When the SET action (expression 40) is committed (section 10.1) PNNI creates nodes a and a’ into the TOP peer group resulting in the new PNNI network of figure 7.

In the example the additional configuration was only performed at switches a.1.3 and a.3.3. This is not sufficient since if one of these switches were not active the hierarchy would not function. Therefore the necessary reconfiguration is normally performed at several or all switches, see section 9.3.4.

9.2 Approach With Address Change

Assume we move peer group A.3 to another country with country code 0008. We call the moved peer group, peer group B.1, see figure 8. Furthermore assume that the first address byte value (Authority and Formats Identifier) remains 39 (Data Country Code Format).

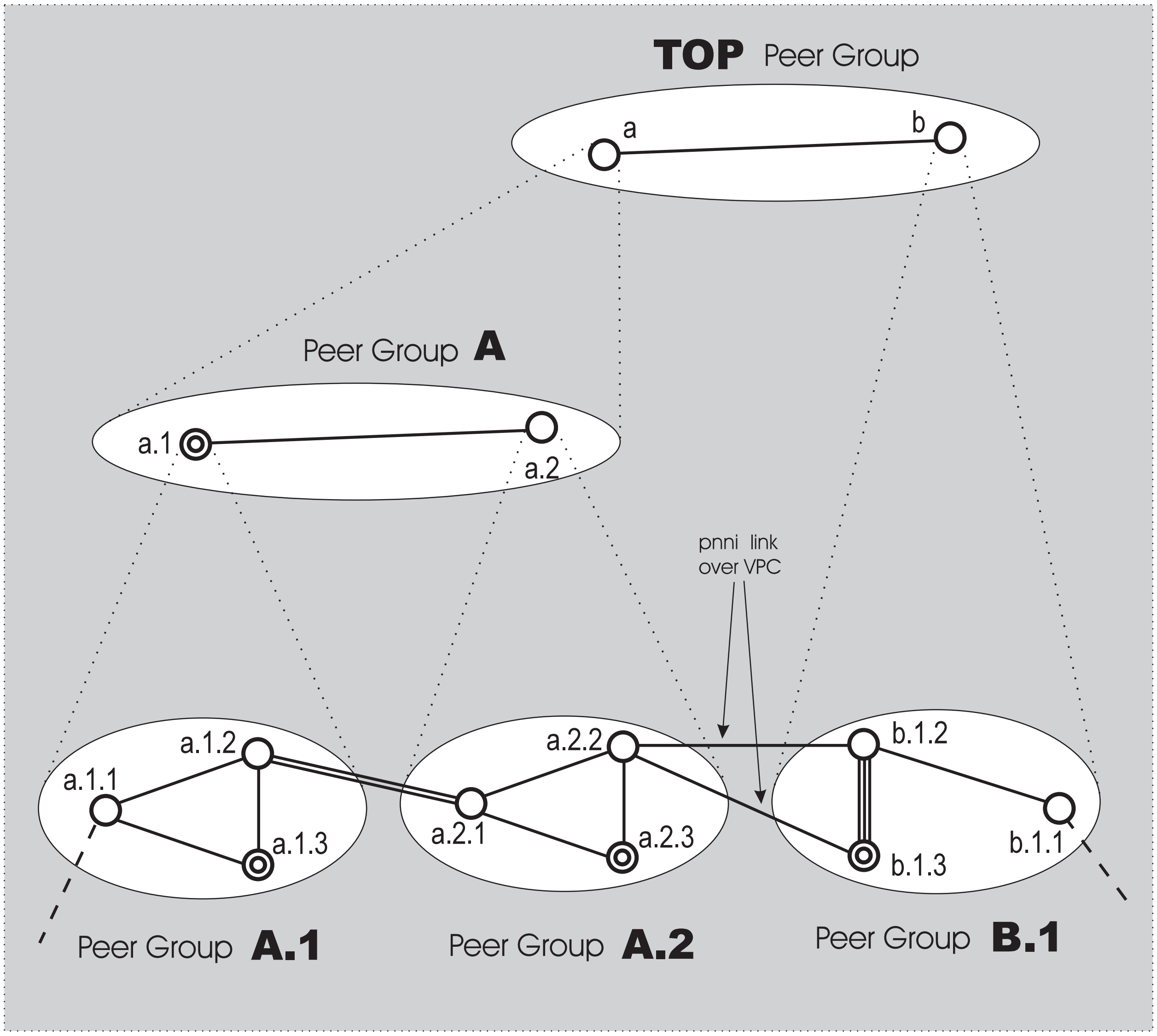

Figure 8: Modifying the Two-Layer Network of Figure 2 Into a Three-Layer Hierarchical Network With Switch-Address Change

Then the private ATM address of switches b.1.1, b.1.2 and b.1.3 in B might be changed, for example, to:

```
39.00089999999999990C0BB101.000000000000.00                                    (41)
39.00089999999999990C0BB102.000000000000.00                                    (42)
39.00089999999999990C0BB103.000000000000.00                                    (43)
```

respectively. As before, the node 0 subsystem b.1.3 wins peer group leadership (by virtue of switch b.1.3’s largest address) causing a node 1 to be default configured and activated at switch b.1.3.

This node 1 has a default peer group ID:

```
39.00089999999999990C0BB1                                                 (44)
```

which is the 88 bit prefix of address (43). Expression (44) can never be truncated to equal expression (39), the TOP peer group ID. We therefore need to enforce the peer group ID expression (39) onto the node 1 subsystem at switch b.1.3 by non-default configuring it with the SET command:

```
8265ATM> set pnni node: 1 peer_group_id: 80 39.00.07.99.99.99.99.99.0C       (45)
```

Expression (45) refers to setting the peer group ID, consisting of the level ID 80 (length in bits of the peer group ID) plus peer group ID itself. This peer group reconfiguration action places the node 1 subsystem, now identified by b, into the TOP peer group thereby creating the new PNNI network of figure 8.

9.3 Configuring More Complex PNNI Networks

The above PNNI network examples are fairly simple. Significantly more complex structures with more hierarchy levels, up to 10, will be possible. As an example lets look at extending the network of figure 8 to that of figure 9. The resulting configuration is still relatively simple from a PNNI point of view, it also does not go beyond three hierarchy levels.

9.3.1 Extending the Network of Figure 8 to That of Figure 9

Peer groups A.1 and B.1 remain unchanged except that the three PNNI links from b.1.2 to b.1.3 are reduced to two. Link (a.2.1, a.2.3) is removed in peer group A.2 which has otherwise grown to include two new nodes a.2.4 and a.2.5 with say addresses:

```
39.00079999999999990C0AA204.000000000000.00                               (46)
39.00079999999999990C0AA205.000000000000.00                               (47)
```

The peer group A.3, see figure 2, is reintroduced hence its switches a.3.1, a.3.2 and a.3.3 have the addresses specified in expressions (14), (15) and (16).

A new peer group A.4 is added whose switches a.4.1, a.4.2 and a.4.3 have say the addresses:

```
39.00079999999999990C0AA401.000000000000.00                               (48)
39.00079999999999990C0AA40d.000000000000.00                               (49)
39.00079999999999990C0AA40f.000000000000.00                               (50)
```

9.3.2 Moving Peer Group Leadership to Another Node

In the new peer group A.4 (figure 9), switch a.4.3 has the highest address; so according to the default configuration set-up (section 5), node 0 at switch a.4.3 should be peer group leader. This is not so, for after reconfiguring the ATM address at switch a.4.1, to the value of expression 48, we reconfigured (at the same switch a.4.1) the leadership priority of node 0 to 52, i.e. we entered:

```
8265ATM> set pnni node: 0 leader_priority:52                              (51)
```

52 is the smallest value that ensures node 0 leadership at a.4.1. The reason is as follows: when a node gains peer group leadership it increments its operational leadership priority by an additive factor of 50 to ensure a certain leadership stability. If both, the address of switches a.4.2 and a.4.3 are reconfigured and the link (a.4.2, a.4.3) is PNNI configured before the address of switch a.4.1 is changed then it is likely that the node 0 subsystem a.4.3 will gain peer group leadership (by virtue if its larger address) before node a.4.1 gets the new leadership priority. In that case node a.4.3 will win peer group leadership with a resulting operational leadership priority of 51 (remember, a node 0 default configuration sets leadership priority to 1). Thus if we select for node a.4.1 a leadership priority of less than 52 then it may never gain peer group leadership. Even a leadership priority of 51 would not help since the node with the larger address wins in a leadership contention with identical leadership priorities.

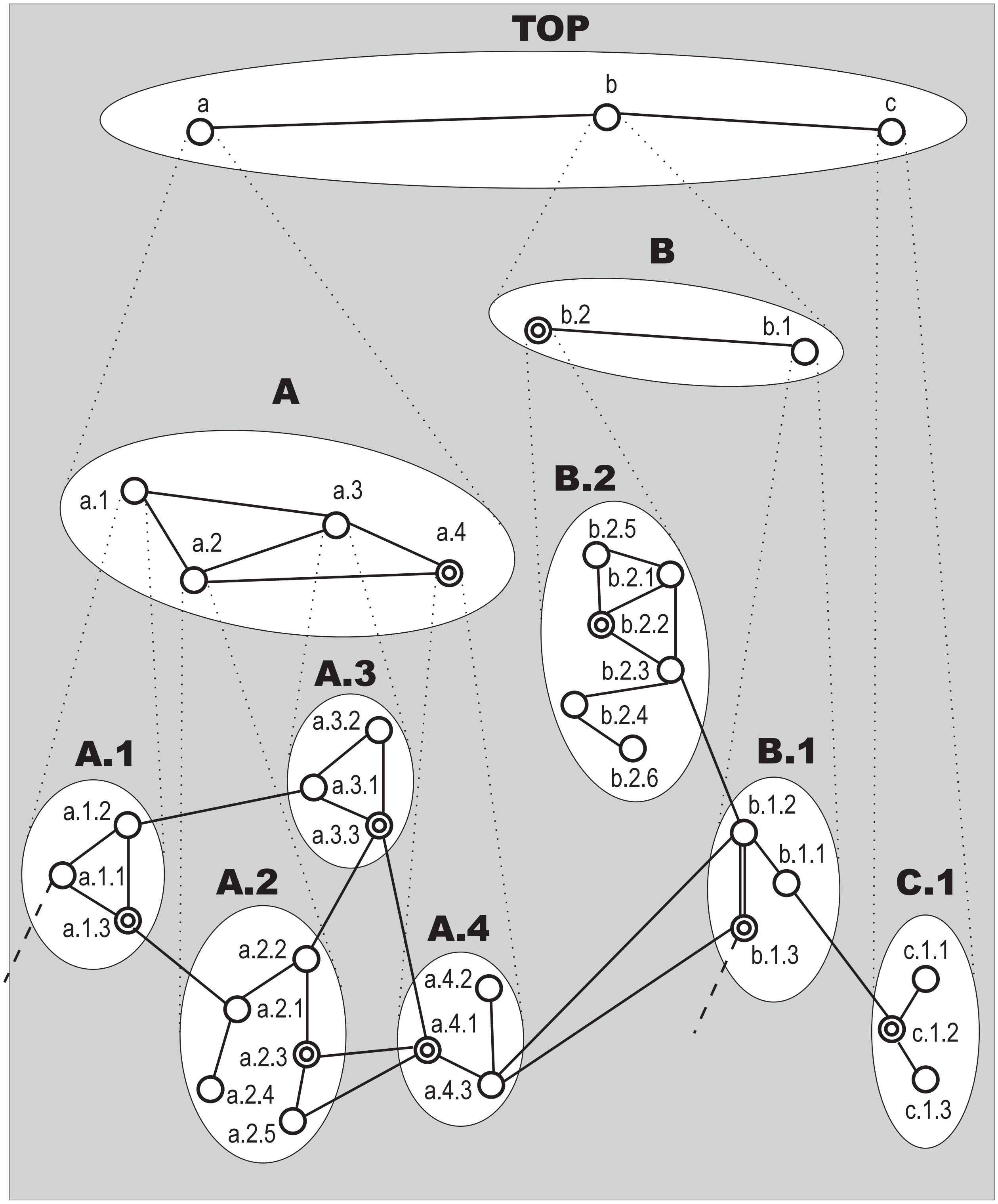

Figure 9: Evolving the Network of Figure 4 Into a More Complex Network

When by virtue of expression (51), node a.4.1 gains peer group leadership its operational leadership criteria becomes 52 + 50 = 102. If now we were to insist on node a.4.3 regaining peer group leadership then we would have to configure its leadership priority to at least 102 (not 103 since switch a.4.2 has higher address value than a.4.1). There is an upper limit to the settable leadership priority, it is 255 - 50 = 205.

Note how the parent peer group A (figure 9) reflects the interconnection topology of its child peer groups A.1, A.2, A.3 and A.4. Peer group A's leader was the node 1 subsystem a.1 (figure 8); it now is the node 1 subsystem a.4 (figure 9). The previously described approach could be used for executing this leadership move. An alternative way is to set node a.1's leadership priority to 0 by entering at switch a.1.3:

```
8265ATM> set pnni node: 1 leader_priority:0                                (52)
```

This action ensures that node a.1 cannot become peer group leader in peer group A. So when we reconfigure the leadership priority of the node 1 subsystem a.4 to greater than or equal to 1, node a.4 becomes peer group leader thereby activating the default configured node 2 subsystem a at switch a.4.1. This new node a is member of the TOP peer group (figure 9). The implicit assumption is that the leadership priorities of nodes a.2 and a.3 were never set, i.e. that they have default value 0.

An alternative way of configuring this node 2 subsystem is to enter at switch a.4.1:

```
8265ATM> set pnni node: 2 configured                                       (53)
```

This default configures node 2 and sets node a.4's leadership priority to 1, see section 11.6. Then when a.4 becomes peer group leader, the node 2 subsystem a is activated.

9.3.3 Continuing the Figure 9 Network Configuration

Another peer group B.2 is added whose switches b.2.1 to b.2.6 have the addresses:

```
39.00089999999999990C0BB201.aaaaaaaaaaaa.00                                (54)
39.00089999999999990C0BB202.000000000101.00                                (55)
39.00089999999999990C0BB203.000000000000.00                                (56)
39.00089999999999990C0BB204.c12346da1010.00                                (57)
39.00089999999999990C0BB205.000000000000.00                                (58)
39.00089999999999990C0BB206.ff00000f0000.00                                (59)
```

Previously all End System Identifiers (last 6 bytes) where chosen with the zero value. The above address list shows that this is not a requirement, i.e. that we can choose whatever switch End System Identifiers are convenient for our network requirements.

Since all node 0 subsystems in peer group B.2 are assumed default configured the level ID of B.2 has the default value of 96 (12 bytes). The default configuration works since the 12 byte prefix of addresses 54 to 59 are identical. Since the 13 byte prefix of a switch address uniquely identifies that switch, it follows that the thirteenth byte (first hexadecimal pair to the left of the second '.') of a switch in B.2 uniquely identifies it in the peer group B.2 (for more information on this see section 11.4). This thirteenth byte can be chosen to whatever value is convenient as long as it is unique for each switch within the peer group.

We choose for the node 0 subsystem b.2.2 to become peer group leader, for example by setting its leadership priority to 100. This causes switch b.2.2 to provide the node 1 subsystem b.2 with a default level ID of 88. It is not necessary for peer group B to have this default level ID 88. For example by entering at switch b.2.2:

```
8265ATM> set pnni node: 1 level_identifier: 83                             (60)
```

we shorten the peer group ID of B by 5 bits, i.e. generate a peer group ID which is the 83 bit prefix of the previous value. The value 83 does not lie on a byte boundary (though it could) and is chosen greater than 80 so that we can maintain unchanged the default level ID 80 for the TOP peer group. If we had chosen say 72 as the level ID of peer group B then the level ID of the TOP peer group would have had to be reduced to at least 71, since level ids (peer group ID lengths in bits) at successive higher hierarchical levels are required to decrease in value.

We also need to enter expression (52) at switch b.1.3 to ensure that the node 1 subsystem b.1 lies in the same peer group as b.2.

The next step is to default configure the node 2 subsystem b, for example by entering expression (45) at switch b.2.2.

Since the node 2 subsystem a has the default peer group ID of expression (39), we need to enforce the node 2 subsystem b to take on expression (39) as peer group ID, i.e. we need to enter at switch b.2.2:

```
8265ATM> set pnni node: 2 peer_group_id: 80 39.00.07.99.99.99.99.99.99.0C        (61)
```

This causes node b to join the TOP peer group which thus contains two node 2 subsystems a and b.

All that remains to do is add the bottom layer peer group C.1 for example by configuring the switch c.1.1, c.1.2 and c.1.3 addresses to:

```
39.00087712ACD43298110B3241.000000000000.00                                     (62)
47.01ABCC12836450090C0CD202.000000000000.00                                     (63)
45.5007891204274629BBBAA810.000000000000.00                                     (64)
```

These 3 private ATM addresses with valid Authority and Formats Identifier (first 2 hexadecimal characters) have little in common so that their respective default configured peer group ids would lead to 3 distinct peer groups.

We can enforce all 3 switches to be members of a single peer group by (nondefault) configuring their respective node 1 subsystems with the same peer group ID, for example by entering at each switch:

```
8265ATM> set pnni node: 0 peer_group_id: 86 39.00.07.99.99.99.99.99.99.0C.0C     (65)
```

The entry of 86 (decimal) causes this bottom level peer group to take on the nondefault level ID value 86. Since 86 equals a byte boundary less 2 (86 = 8x10 - 2), the right most 2 bits of the right most hexadecimal digit C are ignored therefore instead of entering the right most C (bin 1100) we could just as well have entered D (bin 1101), or E (bin 1110) or F (bin 1111).

By configuring the node 0 subsystem c.1.2 with a leadership priority of say 100 we ensure that c.1.2 becomes peer group leader when it is active. Then by configuring the level ID of the resulting node 1 subsystem c to 80 we ensure that node c joins the TOP peer group since the 80 bit prefix of the peer group ID of expression (65) precisely corresponds to the peer group ID of the already existing TOP peer group.

9.3.4 Anticipating Possible Node Failures When Configuring Leadership Priorities

Powering a switch up for the first time generates a default PNNI configuration where node 0 has a leadership priority of 1. Consequently if in a bottom layer peer group, such as B.2 (figure 9), the peer group leader b.2.2 fails then another node 0 can win leadership to ensure peer group representation in the parent peer group.

The situation is different at the next hierarchical level up. Powering a switch up for the first time generates a default PNNI configuration where the node 1 has a leadership priority 0. Consequently in a next hierarchical level up, for example in peer groups B, the node 1 subsystem b.1 has its leadership priority configured to 0 so if node b.2 fails (i.e., switch b.2.2 in the example) b.1 cannot take over peer group leadership. As a result peer group B would no longer be represented in the TOP peer group and PNNI routing would be incomplete.

This vulnerability should be avoided by nondefault configuring at least one backup node 1 subsystem with positive leadership priority. One way of doing this for peer group B is to have the node 1 subsystems at switches b.1.3 and b.1.2 configured with positive leadership priority. That way the PNNI network remains operational even when node b.2 and node b.1.3 fail. For when b.2 fails, switch b.1.3 provides the node 2 subsystem that takes over the b function and when b.1.3 fails, switch b.1.2 provides the node 1 subsystem that takes over the b.1 function and the node 2 subsystem that takes over the b function.

Returning to the bottom level peer group B.2. You may not want all peer group members to have peer group leadership capability. For example you may decide that having b.2.3 as backup suffices. You therefore configure node 0 subsystems at switches b.2.1, b.2.4, b.2.5, b.2.6 with leadership priority 0. You also configure the node 1 subsystem at switch b.2.3 to have positive leadership priority so that switch b.2.3 is in a position to support a node 2 subsystem b in case switch b.2.2 fails.

9.3.5 Peer Group Partitioning Due To Failures

Switch and/or PNNI link failures can lead to peer group partitioning resulting in two different situations. Either the partitioned off cluster is completely isolated from the rest of the PNNI network or it can be reached via alternate routes.

An isolated cluster is generated when for example link (b.2.3, b.2.4) fails. The failure partitions peer group B.2 into cluster 1 (b.2.4, b,2.6) and cluster 2 (b.2.1, b.2.2, b.2.3, b.2.5). Cluster 2 is connected to the rest of the network and continues to function normally as peer group B.2.

Cluster 1 is completely isolated so that communication between it and the remaining network is not possible. Cluster 1 forms a separate peer group with the same peer group ID as B.2. If all members of cluster 1 have their node 0 leadership priority set to 0 then cluster 1 forms a single peer group network. If one or more node 0 subsystems in cluster 1 have positive leadership priority then a parent peer group consisting of only one node 1 subsystem is created. Whichever way the only approach to re-establishing full connectivity is to repair the failed link.

Two non-isolated clusters are created if for example switch a.2.2 fails. The failure partitions peer group A.2 into cluster x (a.2.1, a,2.4) and cluster y (a.2.3, a.2.5) whereby each cluster maintains connectivity with the rest of the network. Since cluster y contains the peer group's original leader a.2.3, it continues to function normally as peer group A.2

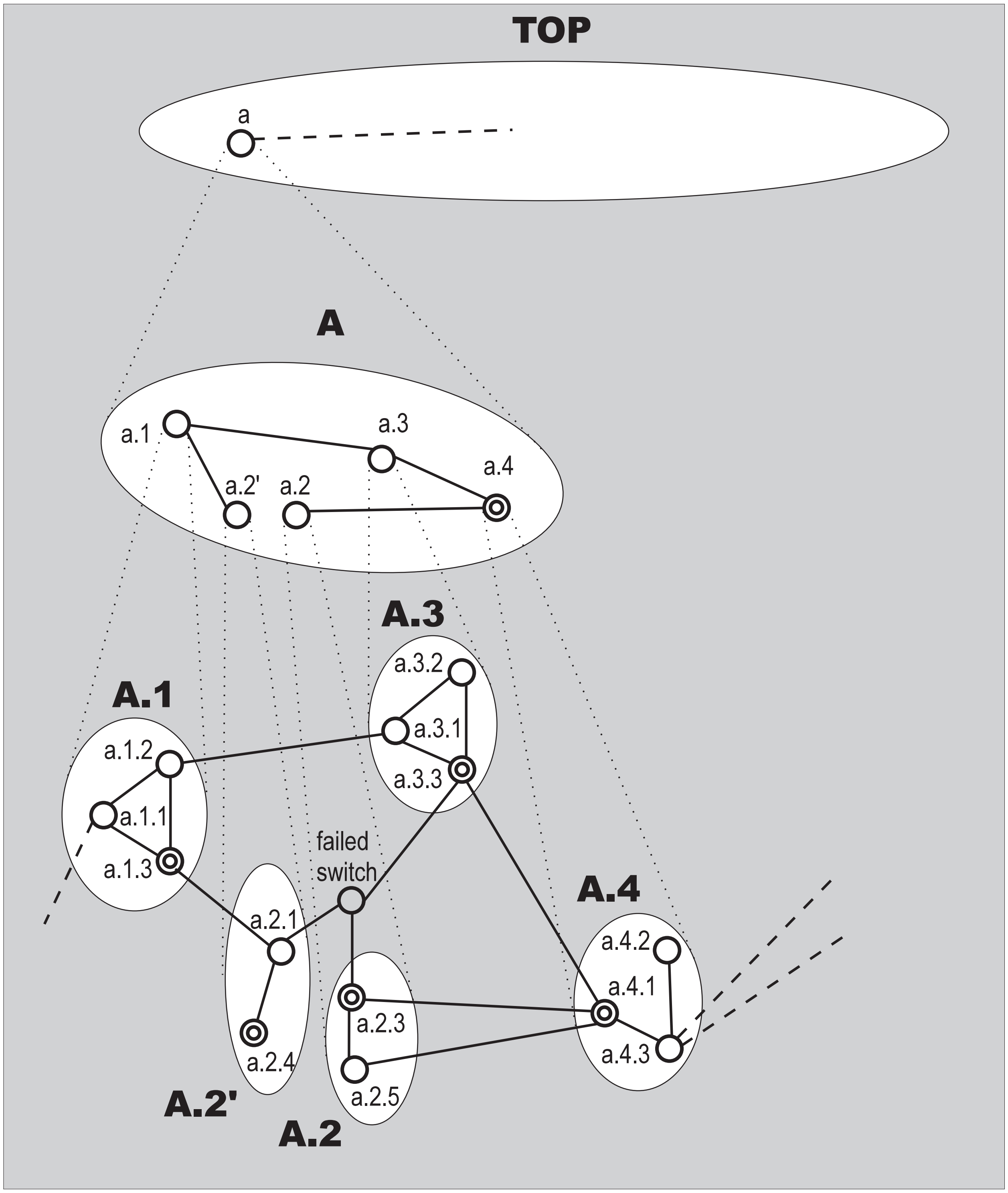

Figure 10: Example of a Peer Group Partition

Cluster x forms a separate peer group A.2’ (see figure 10) with peer group ID identical to that of A.2. The node 0 subsystem a.2.4 becomes peer group leader by virtue of its larger address expression (46). This in turn generates a node 1 subsystem a.2’ at switch a.2.4. Since a.2’ has the same peer group value as a.2 it becomes member of the parent peer group A. Consequently peer group A has two child peer groups A.2 and A.2’ each representing a different part of the originally partitioned peer group A.2. The topology of peer group A clearly reflects the consequences of the partition, namely that the link (a.3, a.2), present in figure 9, has disappeared and that communication between a.1 (peer group A.1) and a.4 (peer group A.4) can only go via a.3.

The resulting PNNI network, shown in figure 10, works well when the **crankback** (see section 13) capability is set to active throughout the network. For example if a set-up is routed from a.1.3 to a.2.1 that is intended for say a.2.3 then crankback will back track the set-up to peer group A.1 with a subsequent re-route via A.3 and A.4 to A.2 and finally to a.2.3.

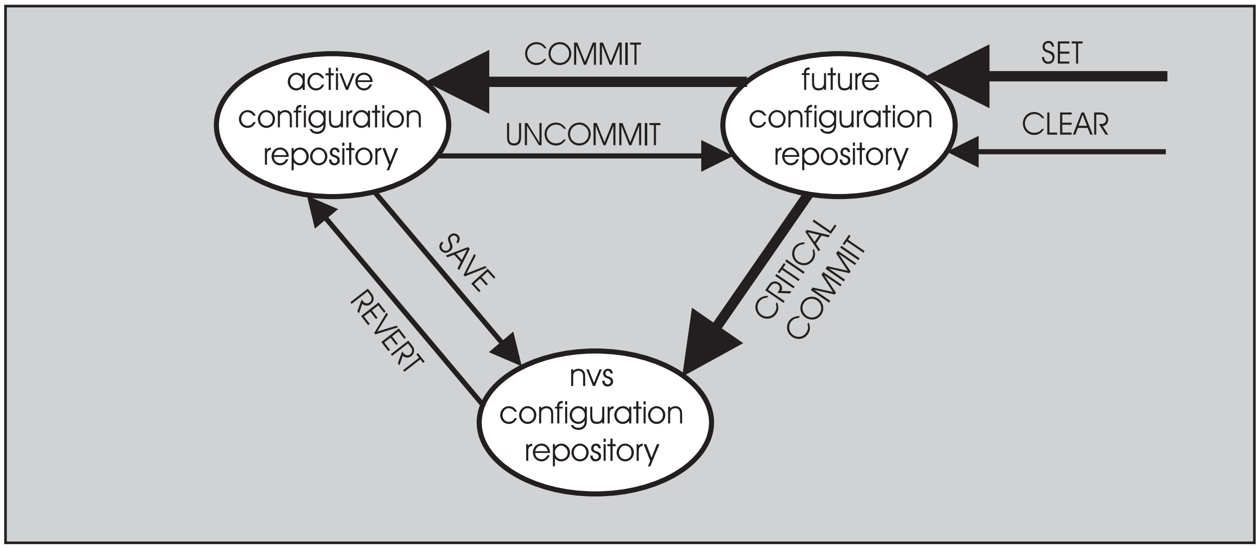
active configuration repository
future configuration repository
nvs configuration repository
COMMIT
UNCOMMIT
SET
CLEAR
SAVE
REVERT
CRITICAL COMMIT

Figure 11: PNNI Configuration Commands and How They Relate To the Three Configuration Repositories

10. Commands for Configuring PNNI

10.1 The SET, COMMIT and UNCOMMIT Commands

When you power up a PNNI supporting ATM switch for the very first time it configures its PNNI system to a default configuration. Parameter values constituting this default configuration are placed in the active configuration repository (figure 11). This is a data repository containing all configurable PNNI parameter values used by the running or active PNNI system. You can reconfigured or change these values with the SET command. For example expression (01) illustrates how this SET command is used to reconfigure a switch's address. Details on the SET command are to be found in subsequent sections.

Reconfigured values are always placed in a future configuration (figure 11) repository. The future configuration repository is a copy of the active repository in which newly reconfigured parameter values replace the old values. When you have set all parameters you want to reconfigure you issue the command:

```
8265ATM> commit pnni                                                    (66)
```

this causes the future configuration repository to replace the current active configuration repository. COMMIT therefore forces PNNI to run with the newly reconfigured parameter values, i.e. all parameter changes entered via the SET command since the last COMMIT execution.

If you do not know or remember whether any reconfiguration actions were previously entered or not and so whether a COMMIT should be issued (is pending) or not, you can query PNNI by issuing a SHOW PNNI CONFIGURATION_STATE which informs you whether a COMMIT is pending or not.

Reconfiguring parameters such as address, peer group level ID and leadership priority can drastically affect your network. Consequently reconfiguring one PNNI parameter at a time will unnecessarily perturb you operational network. You minimize disturbances to your running network by reconfiguring all PNNI parameters you want changed at a given switch before issuing COMMIT.

A second advantage of COMMIT is that when you issue the command, PNNI responds with a short text informing you how critical its execution is, i.e. what the implication of replacing the active with the future repository is. PNNI then queries you as to whether you want to go through with the COMMIT execution or not. If you decide not to go through with it you issue:

```
8260ATM> uncommit pnni                                                  (67)
```

which clears the Future Configuration repository and so removes all SET values issued since the last issued COMMIT.

PNNI accepts a parameter reconfiguration only if the entered value lies within its correct range and is consistent with all the other, already configured, parameter values. Thus PNNI assures that when COMMIT is issued, the new configuration is consistent. All the same you may want to inspect the future configuration repository with one of the SHOW FUTURE_PNNI commands, to make sure the parameter values you entered are the intended ones. If you detect one or more mistakes you simply repeat the corresponding SET command with the correct value before issuing COMMIT.

10.2 The SAVE and REVERT Commands

PNNI supports a third configuration repository, the non-volatile storage or NVS configuration repository (figure 11). Whenever you power up a switch, PNNI loads the configuration image stored in its NVS repository to the active repository.

A special case is when the NVS repository is uninitialized, i.e. the switch is powered up for the first time. PNNI then does a onetime load of the default configuration image to the NVS repository before loading it back from NVS to the active repository.

Whenever you have successfully reconfigured PNNI at a given switch and you are satisfied with the modification(s), you should issue:

```
8260ATM> save pnni                                                   (68)
```

This replaces the PNNI configuration image in the NVS repository with the one in the active repository. It ensures that the current active configuration is automatically reinstalled after a reset or power up action. If you do not know or remember the status of a hub's PNNI configuration you can query its status by issuing a SHOW PNNI CONFIGURATION_STATE which informs you whether the active configuration is saved yes or no.

Alternatively if after successfully reconfiguring PNNI at a given switch you decide that the configuration image prior to the reconfiguration is the better one you enter:

```
8260ATM> revert pnni                                                 (65)
```

which replaces the configuration image in the active repository with the one stored in the NVS repository.

10.3 Critical COMMITs

The consequence of a COMMIT is either uncritical to an operational PNNI network or it is critical in that it may initiate significant temporary and/or lasting changes to the network. The following subsections discuss the critical cases.

When you issue a COMMIT you are always informed when the COMMIT is critical. If it is and you do not want to go through with it, you issue UNCOMMIT (section 10.1).

10.3.1 COMMITs That Generate Reset

A COMMIT generating reset is a COMMIT that is issued when at least one of the pending reconfigurations requires all node subsystems at the given switch be restarted. A COMMIT generating reset first saves the new configurations (future configuration repository) into NVS (non volatile storage configuration repository) before issuing reset. The reset action then reinitializes PNNI by copying the NVS configuration image to the active configuration repository and thereby activating the newly configured parameters.

A COMMIT generating reset temporarily removes the affected switch together with all its links and end systems from the network. This can significantly affect an operational PNNI network, when one or more of the nodes you temporarily deactivate are peer group leaders. In fact such nodes may not regain peer group leadership after the reset due to the additive leadership enhancement factor that favors a peer group leader to maintaining its won leadership (section 11.5).

Issuing a COMMIT generates a reset when PNNI detects that the new configuration image includes at least:

- An address reconfiguration
- A node 0 peer group ID reconfiguration
- A node 0 level ID (peer group length) reconfiguration
- Crankback reconfiguration.

10.3.2 COMMITs That Temporarily Take Down a Section of the PNNI Hierarchy

A ‘temporary hierarchy removal’ COMMIT is a COMMIT that is issued when one or more entered reconfigurations require all nodes above node i to be restarted if they are active. A ‘temporary hierarchy removal’ COMMIT first sets the leadership priority of nodes i, i+1 etc. to zero then executes the parameter updates and finally reconfigures the leadership priority of nodes i, i+1 etc. to their original (or if changed to their reconfigured) values.

If any node j, for j greater/equal i, is peer group leader then this can significantly affect the PNNI network. In fact such a node j may not regain peer group leadership after the COMMIT because during the time that node j’s leadership priority is zero some other node may replace node j as peer group leader. Then when node j regains its original leadership priority, it does not regain peer group leadership by virtue of the additive leadership enhancement factor that favors a current peer group leader maintain its won leadership (section 11.5).

Issuing a COMMIT generates a temporary hierarchy removal down to node i when PNNI detects that the new configuration image includes:

- A node i+1 peer group ID reconfiguration and/or
- A node i+1 level ID (peer group length) reconfiguration

whereby i+1 is the smallest integer for which this is true.

10.3.3 Other Critical COMMITs

Other critical COMMITs are COMMITs that increase the PNNI hierarchy, that decrease the PNNI hierarchy, that cause loss of peer group leaderships or that cause gain of peer group leaderships. The critical parameter for all these COMMITs is the node leadership priority.

A noteworthy case is for example when nodes 0, 1, 2, 3, 4. and 5 are configured at a given switch and say the leadership priority of node 3 is reconfigured to zero. Then when COMMIT is executed, not only does node 3’s leadership priority become zero but nodes 4 and 5 are deconfigured since they can never appear in any PNNI network configuration as long as the leadership priority constellation does not change.

11. The SET PNNI NODE Command

This section focuses on configuring the PNNI node i subsystem. The generic configuration command is:

```
8265ATM> set pnni node: i   <expression>                                      (70)
```

where i identifies the subsystem in which the configuration action is to occur and where <expression> contains more precise information about the specific reconfiguration action. IBM's PNNI implementation will support up to 10 hierarchical levels, so i can take on the values 0, 1, 2, 3, 4, 5, 6, 7, 8 and 9.

When a PNNI supporting switch is powered on for the very first time, only two nodes, node 0 and node 1 subsystems are automatically, i.e. default configured. Sections 11.5 and 11.6 explain how higher level nodes, for example node 2, 3 and 4, get configured.

The following subsections focus on the different PNNI parameters that are configured via expression (66) at a switch's terminal or via TELNET.

11.1 Private ATM Address Configuration

Configuring a switch's ATM address was already discussed in section 5.2. This section just includes additional details.

Each node in a switch's configured PNNI hierarchy has its own address. This address consists of the 19 byte prefix of the switch's address (entered via expression 01) followed by a unique (one byte) Selector value. The node 0 Selector value is 0, the node 1 Selector value is 64 (decimal), the node 2 Selector value is 65 and so on.

IBM's PNNI checks for correct private ATM address format before accepting a new switch address.

11.2 Peer Groups and Their Configuration

A peer group is a cluster of nodes. When one node in the cluster, say node i (for i = 0, 1,..,9) at switch z, wins peer group leadership then node i+1 is activated at switch z to represent the node cluster or peer group in the next level up or parent peer group.

A peer group is defined by an address prefix that we call peer group ID. You make node i member of a peer group by configuring its peer group ID which is a tuple consisting of the peer group ID length in bits, called the level ID, and the peer group ID itself, i.e. the alphanumeric prefix string.

A peer group ID can either be autonomously configured by the PNNI system, this is called a default configured peer group or it can be configured by using the SET command (expression 70), this is called a nondefault configured peer group. The properties of default and nondefault configured peer groups are different.

11.2.1 Default Configured Peer Groups

When a peer group at node 0 of switch z is default configured, i.e. configured by the PNNI system at z, then it always is a prefix to switch z's address whereby the prefix length must be equal to or smaller than 100 bits. This prefix length (level ID) can, in turn, either be default configured by PNNI or nondefault configured by a network administrator. In either case the peer group is considered default configured. If the prefix length is default configured then it takes on the value 96 (12 bytes).

Thus a node 0, where neither peer group nor level ID are configured by expression 70, the peer group ID is the 96 bit prefix of the address at node 0's switch. Alternatively if only the level ID is configure by expression 70, to say 89, then the peer group ID is default configured to the 89 bit prefix of the address of node 0's switch.

When a peer group at node i (for i = 1, 2,.., 9) of switch z is default configured then it always is a prefix to the peer group ID configured at node i-1 of the same switch z. Here also the prefix length (level ID) can either be default configured by PNNI or nondefault configured by a network administrator. In either case the peer group is considered default configured. The default prefix length is detailed in section 11.3, but in general it is 8 bits shorter than the length of node i-1's peer group ID.

The following example shows the effect of letting PNNI (default) configure peer group ids and their lengths (level ids) in successive node i subsystems. Assume that only nodes 0, 1, 2, 3, 4 and 5 are configured at switch z whose address is:

```
41.dddddddddddddddddddd3344.000000000000.00                             (71)
```

Then the peer group ids at successive nodes are:

```
at node 0:  45.dd.dd.dd.dd.dd.dd.dd.dd.dd.dd.33                         (72)
at node 1:  45.dd.dd.dd.dd.dd.dd.dd.dd.dd.dd                            (73)
at node 2:  45.dd.dd.dd.dd.dd.dd.dd.dd.dd                               (74)
at node 3:  45.dd.dd.dd.dd.dd.dd.dd.dd                                  (75)
at node 4:  45.dd.dd.dd.dd.dd.dd.dd                                     (76)
at node 5:  45.dd.dd.dd.dd.dd.dd                                        (77)
```

11.2.2 Nondefault Configured Peer Groups

When a peer group at node i (for i = 0, 1, 2,.., 9) is nondefault configured, i.e. configured via expression (70), it takes on the value of the entered string. For example, to nondefault configure the peer group ID at node 2 of switch z, you enter at the switch's console or via TELNET:

```
8265ATM> set pnni node: 2 peer_group_id: 76 39.00.07.66.55.44.33.22.11.a        (78)
```

where 76 is the bit length of the chosen peer group ID 39.00.07.66.55.44.33.22.11.a. Issuing COMMIT will modify the successive peer group ids in expressions (72) to (77) to:

```
at node 0:  45.dd.dd.dd.dd.dd.dd.dd.dd.dd.dd.33                         (79)
at node 1:  45.dd.dd.dd.dd.dd.dd.dd.dd.dd.dd                            (80)
at node 2:  39.00.07.66.55.44.33.22.11.a                                (81)
at node 3:  39.00.07.66.55.44.33.22.11                                  (82)
at node 4:  39.00.07.66.55.44.33.22                                     (83)
at node 5:  39.00.07.66.55.44.33                                        (84)
```

We see that not only is the peer group ID at node 2 modified but that all subsequent peer groups values at higher levels are modified to prefixes of the new configured value. This is because peer groups at nodes 3, 4 and 5 are default configured. Consequently the peer group ids at node 3 becomes prefix to the new peer group ID at node 2 and so on. Note that the level ids at nodes 3, 4 and 5 have not changed though the level ID at node 2 has been reconfigured from 80 to 76.

If we now reconfigure the peer group of node 4 by issuing:

```
8265ATM> set pnni node: 4 peer_group_id: 64 47.aa.aa.aa.aa.aa.aa.aa             (85)
```

Then issuing COMMIT will modify the successive peer group ids in expressions (79) to (84) to:

```
at node 0:  45.dd.dd.dd.dd.dd.dd.dd.dd.dd.dd.33                         (86)
at node 1:  45.dd.dd.dd.dd.dd.dd.dd.dd.dd.dd                            (87)
at node 2:  39.00.07.66.55.44.33.22.11.a                                (88)
at node 3:  39.00.07.66.55.44.33.22.11                                  (89)
at node 4:  47.aa.aa.aa.aa.aa.aa.aa                                     (90)
at node 5:  47.aa.aa.aa.aa.aa.aa                                        (91)
```

If we now nondefault reconfigure the peer group at node 2 to a new value then issuing COMMIT will only affect the peer group ids of nodes 2 and 3. This is because node 4's peer group is nondefault configured and hence maintains its last configured value independently of the peer group ID configured at the node below it, i.e. at node 3.

A last point to consider is that you revert from a nondefault to a default configured peer group by reconfiguring the level ID at the respective node. This issue is clarified in the next section 11.3.

11.3 Level ID Configuration

The level ID at node i (for i = 0, 1,..,9) is the length in bits of the address prefix that constitutes the peer group ID configured at node i.

A level ID can either be autonomously configured by the PNNI system, this is called a default configured level ID or it can be configured with the help of the SET command of expression 66, this is called a nondefault configured level ID. The properties of default and nondefault configured level ids are different in one respect.

When at a given node i the peer group is nondefault configured then if you nondefault reconfigure the level ID at node i you change the peer group at node i to from a nondefault to a default configured value. An example helps clarify; if after configuring node 2’s peer group (expression 78) you nondefault configure the level ID at the same node 2 by entering:

```
8265ATM> set pnni node: 2 level_identifier: 80                                    (92)
```

where 80 is the new bit length of the peer group ID at node 2. Then issuing COMMIT will modify the successive peer group ids from expressions (79) to (84) to those of expressions (72) to (77). What happened is that the peer group ID at node 2 becomes default configured and therefore becomes the 80 bit prefix of node 1’s peer group ID. But since node 3’s peer group is default configured it also changes and so on.

The reason for having a length change, trigger a content change, is that it is a convenient way of reverting from the nondefault configured (peer group) category to that of the default configured category (section 11.2.2) without introducing additional commands. It does imply though that if you only want to change the length of a nondefault configured peer group you must enter:

```
8265ATM> set pnni node: 4 peer_group_id: k  abc                                   (92)
```

where k is the new length and where abc is a repeat of the old the peer group ID truncated to k bits.

IBM’s PNNI implementation currently limits the range of nondefault configurable level ID from 100 to 3.

The level ID at a node i subsystem must be chosen larger than the level ID at the node i-1 subsystem of the same switch. This imposes bounds on the choice of level ids at successive levels of the hierarchy within a switch. In extreme cases, it can even inhibit further growth of a PNNI hierarchy. For example if you nondefault configure the level ID at node 0 to 5 then node 1’s level ID must be smaller than 5, say 4, and node 2’s level ID must be smaller than 4, so its 3. Consequently node 3 cannot be configured for its level ID would have to be smaller than 3 which is beyond the current lower limit of 3. The choice of such small level ids makes little sense as it unnecessarily limits peer group membership. For more on this see the next section 11.4

The default configured level ID at node 0 is 96 this implies a 96/8 = 12 byte long node 0 peer group ID. The default configured level ID at node i for i>0 is the level ID at node i+1 minus 8. For example, the default configured level ID at node 1 is always 88 (11 bytes) if the level ID at node 0 is 96. But if the level ID at node 0 were to be explicitly (nondefault) configured to say 55 then when PNNI configured node 1 it would (default) configured the level ID at node 1 to 55 - 8 = 47.

In extreme cases where level ids are too smaller to be decremented by 8, the ’decrement by 8’ strategy is replaced by a ‘decrement by 1’ strategy. For example if you nondefault configure the level ID of node 0 to 7 then when PNNI configures node 1 it will (default) configure the level ID at node 1 to 7 - 1 = 6. This strategy change is there only to avoid, as much as possible, that PNNI refuses to configure new nodes just because level ids are too small.

In any case IBM's implementation ensures that nondefault configured level ids are only accepted by PNNI if they are consistent with already configured level ids be they default or nondefault configured.

11.4 Upper Limits On the Number of Peer Group Members

This section is only concerned with default configured peer group values

There is a close relationship between switch address and key PNNI parameters such as peer group ids and default configured summary addresses (section 11.8.3). We now investigate these relationships with the help of figure 12.

The default configured peer group ID at a node 0 has a prefix length which is the node 0 level ID. The address subfield that lies between node 0's default peer group ID and the End System Identifier, figure 9, is [104 - level ID of node 0] bits long and uniquely identifies the switch within its node 0 peer group. Therefore the length of this subfield determines the maximum number of switches that can be contained in node 0's peer group. When you power up a switch for the very first time the default node 0 level ID is 96 hence this subfield is [104 - 96] = 8 bits long which equates to a maximum population of 256 switches in the bottom peer group. If you reconfigure node 0's level ID to say 100 then the subfield shrinks to [104 - 100] = 4 bits which then equates to a maximum population of 16 switches in node 0's peer group.

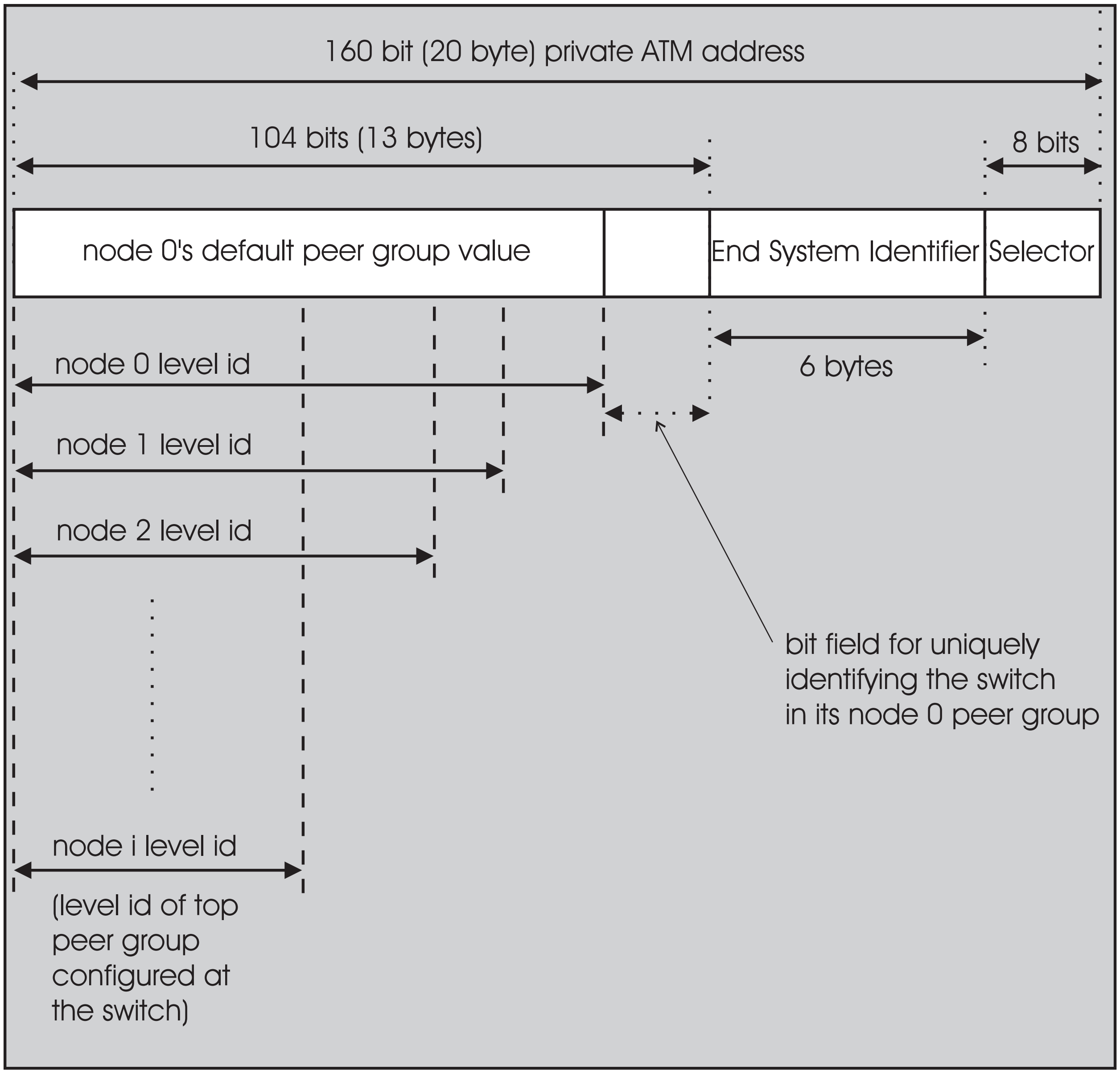
160 bit (20 byte) private ATM address
104 bits (13 bytes)
8 bits
node 0's default peer group value
End System Identifier
Selector
6 bytes
node 0 level id
node 1 level id
node 2 level id
node i level id
(level id of top peer group configured at the switch)
bit field for uniquely identifying the switch in its node 0 peer group

Figure 12: A Switch's Private ATM Address and Its Relation To the Level Ids of the Default Configured Peer Group Values At That Switch

If the considered node 0 is peer group leader then a corresponding node 1 is activated and the address subfield that lies between the node 0 and node 1 peer group ids is [level ID of node 0 - level ID of node 1] bits long and uniquely identifies (in node 1's peer group) the child peer group that node 1 represents. Therefore the length of this subfield determines the maximum number of bottom level child peer groups that can be represented in the parent peer group. Consider for example the case where you reconfigure the level ID of node 1 to 91 (whereby node 0 level ID is still 96) then this subfield shrinks to 5 bits which implies that if node 0 wins peer group leadership then the parent peer group (containing that switch's node 1) can maximally support 32 child peer groups, i.e. 31 other child peer groups.

This relationship between level ids of successive default configured peer groups and the maximum number of child peer groups that can be represented in the corresponding parent peer group, is maintained as we progress up the PNNI hierarchy till the level ID of the highest configured node is reached, i.e. node i in figure 12.

11.5 Configuring Leadership Priorities

The leadership priority can only be set at a node i that is already configured. It is set by entering:

```
8265ATM> set pnni node: i leader_priority <k>                              (93)
```

where k becomes the new (nondefault) leadership priority at node i after issuing COMMIT. Setting the leadership priority at node i has two effects: first it influences node i's chance of winning or loosing peer group leadership in node i's peer group, secondly it influences the presence or absence of the node i+1 subsystem at node i's switch. These two aspects are detailed in the next two sections.

11.5.1 The Effect Of Leadership Priority On Peer Group Leader Election

In a peer group, the node with the highest leadership priority wins peer group leadership. When two or more nodes have identical peer group leadership then the node with the highest address wins peer group leadership. If a node wins peer group leadership it increments its operational (as opposed to its configured) leadership priority by the additive factor of 50 so as to favor its current hold on the leadership in the interest of network stability. The maximum configurable leadership priority is 255 - 50 = 205 and the minimum configurable leadership priority of 0 excludes the node from the peer group leader election process, i.e. it cannot become peer group leader.

Consider the case where node i at switch z has a configured leadership priority of 31 which causes it to win peer group leadership resulting in an operational leadership priority of 31 + 50 = 81. Then a second node i at switch y appears, in the same peer group as node i at z, with a leadership priority of 81. Nothing happens if the address of switch y is smaller than that of z. But if switch y's address is greater than z's then node i at y wins peer group leadership and increments its operational leadership priority to 81 + 50 = 131 and node i's behavior at switch z is henceforth dictated by it configured value of 31.

Returning to the case where nothing happens, i.e. where switch y's address is smaller than z's. If now switch z is momentarily powered down for example by a reset causing action (section 10.3.1) then node i at y will win and remain peer group leader even after node i at z regains operational status.

11.5.2 The Effect Of Leadership Priority On The PNNI Hierarchy

When a node i subsystem at switch z is configured with a positive leadership priority then it can in principle always become peer group leader. If it does then the PNNI system at z creates a node i+1 subsystem to represent node i's peer group in the parent peer group. Now that's only possible if PNNI knows or has been told what the parameter values are that constitute node i+1. Therefore when node i (at switch z) is configured (via expression 93) with a positive leadership priority, then PNNI (default) configures the node i+1 subsystem at z, i.e. default configures all the parameters that constitute node i+1.

Consequently you configure a new node i+1 by changing node i's leadership priority from zero to a positive value. The resulting node i+1 default configuration is:

a) Leadership priority = 0

b) Level ID = (level ID of node i) - 8   [in extreme cases (level ID of node i) - 1]

c) Peer group ID = prefix to node i peer group ID with node i's level ID as prefix length

d) Default summ addr = node i's peer group ID

Node i+1's leadership priority is default set to zero (a) to ensure that configuring node i's leadership priority does not cause a chain reaction with node i+2 getting configured and so on.

When COMMIT is issued, node i may win peer group leadership resulting in node i+1 becoming active. If you therefore want node i+1 to have parameter value(s) different from the default configured values, then you must reconfigure them with the help of expression (70) before you issue COMMIT.

This approach to configuring a new node has the advantage that the resulting configurations are always consistent; for the original default configuration is consistent and any additional nondefault configurations, entered via expression (70), are only accepted if they are consistent with the already configured values be they default or nondefault configured.

To deconfigure a node i+1 you set the corresponding node i's leadership priority to 0. If you deconfigure a node i+1 that was active then issuing the subsequent COMMIT causes that node to disappear from the PNNI network. If in addition nodes i+2, i+3 and i+4 are also configured then setting node i+1's leadership priority to 0 also deconfigures nodes i+2, i+3 and i+4.

Before terminating this section criteria (b) needs clarification. Normally when a new node i+1 is configured its default level ID is the level ID of the corresponding node i minus 8. But for cases where the level ID of node i is too small to permit a 'decrement by 8' it is replaced by a 'decrement by 1'. For example if node i's level ID is smaller than 8, say 6 then the 'decrement by 1' is selected resulting in a node i+1 level ID of 5. For the singular case of a node i level ID of 3 (smallest permitted value) the 'decrement by 1' is no more applicable and node i+1 cannot be configured. In that case node i's level ID must be increased to at least 4 before attempting to configure node i+1 again.

11.6 Configuring Nodes

If node i is configured you can default configure nodes i+1, i+2,.., i+n by entering at the switch's console:

```
8265ATM> set pnni node: i+n configured                                       (94)
```

This causes the hierarchy to grow by n levels. A case of special interest is where n equals 1, i.e.:

```
8265ATM> set pnni node: i+1 configured                                       (95)
```

It is equivalent to the following set leadership priority:

```
8265ATM> set pnni node: i leader_priority  1                                 (96)
```

Thus entering expression (95) is absolutely equivalent to entering expression (96). You are encouraged to use expressions (95), or (94) with n=1, when building up your hierarchy configuration. In other words its good practice to configure one node at a time.

Expression (94) can of course be entered with n > 1. For example if i = 1 and n = 3 then nodes 2, 3 and 4 are default configured when COMMIT is executed. This consists in configuring consecutive nodes according to criteria (b), (c) and (d) (see section 11.5.2) whereby criteria (a) is replaced by the new criteria:

a) leadership priorities at nodes i, i+1,..,i+n-1 are default configured to 1 and the leadership priority at node i+n is default configured to 0

Then when COMMIT is issue, node i or nodes i & i+1 or nodes i & i+1 & i+2 may win peer group leadership resulting in the corresponding nodes i & i+1 or nodes i & i+1 & i+2 or nodes i & i+1 & i+2 & i+3 becoming active. If you therefore want these nodes to have parameter value(s) different from the default configured values, then you must reconfigure them via expression (70) before you issue COMMIT.

11.7 Deconfiguring Nodes

If nodes i, i+1,.., i+n are configured you can deconfigure them by entering at the switch's console:

```
8265ATM> set pnni node: i not_configured                                     (97)
```

This will shrink the hierarchy to only include nodes 0, 1,.., i-1 with node i-1’s leadership priority set to 0. If nodes i, i+1,.., i+n are active then executing COMMIT after issuing expression (97) causes all of them to disappear from the PNNI network.

11.8 Summary Addresses and Their Configurations

In PNNI reachability is the advertising of end system addresses within the network for the purpose of setting up connections between end systems. Reachability in PNNI routing is simplified by the capability of having groups of addresses with a common prefix be represented by that prefix. Such a prefix is called a summary address. PNNI generates default summary addresses to provide reachability to all end systems attached to the switch whose addresses share the switch's 13 byte ATM address prefix, i.e. whose addresses are generated by the ILMI address notification protocol (section 7.2). Additional non-default summary addresses can be configured to provide user defined reachability summarizations for end system addresses that do not share their switch’s 13 byte address prefix.

11.8.1 An Example As It Applies To the Bottom Level

An example will help clarify the concept of nondefault summary addresses. Assume you configure three ‘non ILMI’ endsystem addresses, that form part of the considered PNNI network, at a switch of that network. You do this by first configuring the switch ports, to which the end systems are attached, to be of type UNI (see 8265 documentation). You do this by entering:

```
8265ATM> set port 3.2 enabled UNI                                       (98)
8265ATM> set port 2.3 enabled UNI                                       (99)
8265ATM> set port 4.1 enabled UNI                                      (100)
```

where 3.2, 2.3 and 4.1 specify the slot number, port number tuples to which the considered end systems are attached. Expressions (98) to (100) cover other parameter settings but these are beyond the scope of this document.

You then configure the three end systems addresses by entering:

```
8265ATM> set reachable_addr 3.2 152 39.0007444444444444444444a.111155555555   (101)
8265ATM> set reachable_addr 2.3 152 39.0007444444444444444444b.111155555555   (102)
8265ATM> set reachable_addr 4.1 152 39.0007444444444444444444b.111166666666   (103)
```

where for example 39.0007444444444444444444a.111155555555 is the 152 bit (19 byte) address of the end system attached to the slot port tuple 3.2. Note that in expressions (101) to (103), the commands as well as formats are slightly compressed in the interest of clarity so as to obtain ‘one liners’ per command. Here also expressions (101) to (103) cover other parameter settings that are not shown.

Inspection of expressions (101) to (103) show that all three addresses have a common 100 bit (12 & 1/2 byte) prefix:

```
3900074444444444444444444                                              (104)
```

and so offer an opportunity for address summarization within the PNNI network. You exploit this commonality by configuring one nondefault summary address to represent the three considered addresses. You do this by entering:

```
8265ATM> set pnni node 0 summ_addr internal 100 3900074444444444444444444      (105)
```

If you were to enter expression (105) followed by a COMMIT before entering expressions (101) to (103) then the switch’s node 0 subsystem stores the summary address without using it. Then when you enter expressions (101) and (102), PNNI compares the previously entered summary address with the currently entered addresses, detects a match and activates the summary address to represent the two new end system addresses. If you then enter the third address (expression 103) PNNI will detect that, the now active, summary address also matches the third address, it will therefore extend its end system address coverage to include the new address.

If you deactivate the addresses specified in expressions (101) to (103), then the summary address does not represent any reachabilities and so is saved up for future use. Of course the summary address can always be removed by deconfiguring it , see section 11.8.7.

You can remove a summary address any time you want to. If its is not being used then removal has no impact on PNNI's operation. Alternately if you remove a summary address that represents end system addresses then the affected end systems continue to be represented in PNNI, i.e. reachability is in no way impaired, but the storage gain (storage reduction) that summarization represents is lost.

Another property of summary addresses is that PNNI maps reachabilities to summary addresses according to the longest matching prefix. For example if you configure a second summary address by entering:

```
8265ATM> set pnni node 0 summ_addr internal 120 390007444444444444444444b1111  (106)
```

then since this second summary address constitutes a longer matching prefix for addresses (102), and (103) but not for (101), PNNI will change the 'summary address to reachability mapping' so that the second summary address represents reachabilities (102), (103) whereas the first summary address now only represents reachability (101).

There are different kinds of summary addresses hence the qualifying parameter 'internal' in expression (105) and (106). 'Internal' specifies that the configured summary address is to apply to the addresses of end systems that belong to the considered PNNI network.

11.8.2 Extending the Example To Higher Levels of the Hierarchy

Assume that the three end systems with their respective addresses, considered in section 11.8.1, are attached to switch a.1.3 of the network shown in figure 2 (see also figure 1). Furthermore assume that another group of end systems attached to switch a.1.1 are represented at the node 0 subsystem a.1.1 by the 112 bit (14 byte) summary address:

```
3900074444444444444477777733                                                   (107)
```

As discussed in section 7.3, this summary address constitutes a reachability that is flooded to all switches in peer group A.1 (figure 2). It therefore reaches switch a.1.3 that updates its routing tables with it to:

- Table 1 pertaining to members of peer group A.1:

```
a.1.1's nondefault reachability info: 3900074444444444444477777733              (108)
a.1.1's default reachability info:       390007999999999990C0AA102              (109)
a.1.2's default reachability info:       390007999999999990C0AA102              (110)
```

- Table 2 pertaining to members of peer group A:

```
a.2's default  reachability info:        390007999999999990C0AA2                (111)
a.3's default reachability info:         390007999999999990C0AA3                (112)
```

These routing tables are similar to those at switch a.1.1 that were introduced in section 7.4. Path Selection at a.1.3 can now use entry (108) to compute routes to end systems attached to switch a.1.1 that are represented in the PNNI network by summary address (107).

Next, let us configure another summary address but this time associated with the node 1 subsystem a.1 at switch a.1.3 by entering:

```
8265ATM> set pnni node 1 summ_addr internal 80 39000744444444444444            (113)
```

this 10 byte summary address represents at the next level up (i.e. at peer group A), all reachabilities contained in peer group A.1 for which expression (113) is a longest matching prefix. It therefore represents summary addresses (105) & (106) at a.1.3 and by virtue of the principle successive abstraction the reachabilities that these summary addresses represent. It also represents summary addresses 107 and 108 at a.1.1 and the reachabilities that (107) represents.

Since summary address (113) represents all reachabilities in A.1, for which it is a longest matching prefix, it follows that (113) is not limited to only representing summary addresses. It also represents individual end system addresses in A.1 that are not represented by summary addresses and for which (113) is a longest matching prefix.

For example if an end system is configured at switch a.1.1 by entering:

```
8265ATM> set port 1.3 enabled UNI                                              (114)
8265ATM> set reachable_addr 1.3 152 39.00074444444444444404444a.111155555555    (115)
```

then its address cannot be represented by the previously nondefault configured summary address at a.1.1, expression (107), for (107) is no prefix to address (115). The reachability that address (115) represents is therefore flooded in peer group A.1, reaches switch a.1.3 and updates a.1.3's route tables (expressions 108, 109, & 110 values) to:

- Table 1 pertaining to members of peer group A.1:

```
reachability at a.1.1: 39000744444444444444404444a111155555555                 (116)
reachability at a.1.1: 3900074444444444444477777733                             (117)
reachability at a.1.1: 390007999999999999 0C0AA102                              (118)
reachability at a.1.2: 390007999999999999 0C0AA102                              (119)
```

Now since summary address (113) (represents all reachabilities in A.1 for which it is a longest matching prefix) is longest matching prefix to entry (116), it also represents (116) and hence the end system address (115) at switch a.1.1.

Summary address (113) was configured at node a.1, it therefore represents a subset of reachabilities of A.1 in the parent peer group of A.1, namely in peer group A. Reachability (113) is therefore flooded (section 7.3) to every member of peer group A then down and on to every switch represented by every member of peer group A. It therefore reaches, say switch a.3.2, that updates its routing tables (expressions 33 to 36 in section 7.4) with it to:

- Table 1 pertaining to member of peer group A.3:

```
    a.3.1's default reachability info:        39.0007999999999999 0C0AA302      (120)
    a.3.3's default reachability info:        39.0007999999999999 0C0AA303      (121)
```

- Table 2 pertaining to members of peer group A:

```
    a.1's nondefault reachability info:       39000744444444444444              (122)
    a.1's default reachability info:          39.0007999999999999 0C0AA1        (123)
    a.2's default reachability info:          39.0007999999999999 0C0AA2        (124)
```

Path Selection at a.3.2 can now use entry (122) to compute routes to any end system in peer group A.1 for which expression (122), i.e. expression (113), is a longest matching prefix.

This process continues up the PNNI hierarchy. For example in the network of figure 4 (whose peer groups A.1 and A are sufficiently similar to those of the previously considered figure 2 network to permit a seamless continuation) lets configure another summary address but this time associated with the node 2 subsystem at switch a.1.3 by entering:

```
8265ATM> set pnni node 2 summ_addr internal 32 39000744                         (125)
```

This 4 byte summary address is associated with node a and therefore represents, at the TOP peer group, reachability in the child peer group A, and by virtue of the principle successive abstraction, reachabilities in peer groups A.1 and A.2. Now summary address 125 turns out to be 'a longest matching prefix' to all reachabilities in peer group A (assuming no additional reachabilities configured in A.2). It therefore represents all reachabilities in A and hence in A's child peer groups A.1 and A.2.

Summary address (125) was configured at node a, it is therefore flooded (section 7.3) to all members of the TOP peer group then down and on to eventually reach every switch represented by each member of the TOP peer group. It therefore reaches, say switch b.1.2, that updates its routing tables with expression

(125), section 7.4. That way Path Selection at b.1.2 can use reachability (124) to compute routes to any end system represented by node a in the TOP peer group.

11.8.3 Default Configured Summary Addresses

PNNI automatically create an internal summary address, the so-called default configured summary addresses, at every configured node i for i = 0, 1, 2,.., 9.

The default internal summary address associated with the node 0 subsystem at a switch is the 13 byte address prefix of that switch. It represents all end systems attached to the switch via the UNI interface whose addresses are specified with the help of the ILMI protocols. It can never be removed (section 11.8.7) because it would cause loss of connectivity for all end systems that share a switch's 13 byte prefix

The default internal summary address associated with a node i, for i = 1, 2,.., 9, at a switch, is the peer group ID configured at node i-1 of that same switch. It represents the end systems attached to all switches (via the UNI interface whose addresses are specified with the help of the ILMI protocols) that are represented in the peer group configured at the considered node i-1.

No other types of summary addresses are default configured.

11.8.4 Configuring Nondefault Summary Addresses

It is important to note that PNNI's reachability capability is in no way impaired by the absence of nondefault configured summary addresses. The presence of nondefault configured summary addresses simply reduces routing related storage and speeds up route computation.

There are four types of summary addresses: internal summary addresses, exterior summary addresses, internal suppressed summary addresses and exterior suppressed summary addresses. They can be nondefault configured by entering:

```
8265ATM> set pnni node i summary_addr internal <prefix_len> <prefix>            (126)
8265ATM> set pnni node i summary_addr exterior <prefix_len> <prefix>            (127)
8265ATM> set pnni node i summary_addr suppressed_internal <prefix_len> <prefix> (128)
8265ATM> set pnni node i summary_addr suppressed_exterior <prefix_len> <prefix> (129)
```

respectively, where i is the number of the node subsystem at which the summary address in question is configured. The entered prefix in expressions (126) and (128) must be prefix to a private ATM address since these expressions refer to PNNI network internal addresses.

Internal and exterior identify different classes of addresses whereby internal refers to PNNI and exterior refers to non PNNI. The rules that govern internal and exterior summary addresses are very similar. We now describe the different summary addresses configurable via expressions (126) to (129).

A nondefault configured internal summary address at the node 0 subsystem of a switch (expression 126 with i = 0) is an internal reachability that represents the addresses of all PNNI end systems attached to that switch for which the summary address is a longest address matching prefix. An end system is a PNNI end system if its attachment is configured by the command SET PORT *u.v* ENABLED UNI and its address is configured by the command SET REACHABLE ADDRESS *u.v addr_len addr* where *u, v* is the slot, port tuple to which the considered end system is attached, *addr_len* its address length in bits and *addr* the alphanumeric string representing the address.

A nondefault configured internal summary address at node i for i>0 of a switch (expression 126 with i > 0) represents all PNNI internal reachabilities, associated with the peer group that node i represents, for which the summary address is a longest matching prefix. Internal reachabilities refer to reachabilities that represent other lower level internal reachabilities and/or reachabilities of PNNI end systems (see section 11.8.2).

A configured exterior summary address at the node 0 subsystem of a switch (expression 127 with i = 0) is an exterior reachability that represents the addresses of all non PNNI end systems attached to that switch for which the summary address is a longest address matching prefix. An end system is a non PNNI end system if its attachment is configured by the command SET PORT *u.v* ENABLED IISP and its address is configured by the command SET REACHABLE ADDRESS *u.v addr_len addr*, where *u, v* is the slot, port tuple to which the considered end system is attached, *addr_len* its address length in bits and *addr* the alphanumeric string representing the address.

A configured exterior summary address at node i for i>0 of a switch (expression 127 with i > 0) represents all exterior (non PNNI) reachabilities, associated with the peer group that node i represents, for which the summary address is a longest matching prefix. Exterior reachabilities refer to reachabilities that represent other lower level exterior reachabilities and/or reachabilities of non PNNI end systems.

Suppressed summary addresses are like ordinary summary addresses except that the reachabilities that a suppressed summary address represents are not advertised (suppressed) beyond the PNNI subnetwork covered by the suppressed summary address. The coverage of a summary address is determined by the number of the node subsystem at which it is configured. For example if you configure a suppressed summary address at node i (expressions 128 and 129) then all reachabilities represented by the suppressed summary address will not be advertised beyond the peer group that node i represents, i.e. they are not flooded across node i’s peer group and higher level peer groups.

This is also true for i = 0. Thus if we configure say an internal suppressed summary address at a node 0 (expression 128 with i = 0) then all internal end system addresses that are represented by the suppressed summary address are not advertised in the bottom level peer group to which the node 0 belongs. Consequently such end systems can only communicate with other end systems, directly attached to the same switch.

11.8.5 The Impact Of Summary Addresses On Suppressed Summary Addresses

A configured (non suppressed) summary address identifies a range of address prefixes such that if any one of them is configured as a suppressed summary addresses then the affect of that suppressed summary address may be influenced by the non suppressed summary address. For example consider the case where SET REACHABLE_ADDRESS is used to configure the following three internal end system addresses:

```
39.1188000000000000000000000.440000000000                                        (130)
39.1188000000000000000000000.550000000000                                        (131)
39.1199000000000000000000000.660000000000                                        (132)
```

which are represented at a node 0 subsystem by a previously configured internal summary address:

```
39.11                                                                            (133)
```

If you then issue:

```
8265ATM> set pnni node 0 summary_address suppressed_internal:  24  39.11.88     (134)
```

you would expect the addresses (130) and (131) to become ‘non advertised’. But address (132) is still reachable via summary address 39.11 and addresses (130) and (131) share that same prefix. Therefore by virtue of matching prefixes, addresses (130) and (131) remain advertised!

One can deduce from this example that if a suppressed summary address contains a previously configured summary address as prefix then that suppressed summary address has no effect on addresses that it is representing. This holds true for both internal and exterior suppressed summary addresses.

You can avoid this configuration limitation by removing the summary address that is inhibiting the suppressed summary addresses. In the above example, if you remove the internal summary address 39.11 then configuring the internal suppressed summary address to 39.11.88 will effectively make addresses (130) and (131) ‘non advertised’. Note that address (132) remains advertised since it is now individually represented in PNNI.

11.8.6 Number Of Configurable Summary Addresses

The total number of configurable summary addresses per switch is 50 plus 10 that are reserved for default configured addresses since you can maximally configure 10 levels of hierarchy. You can allocate the 50 summary addresses any way you want. For example if you are running a 5 level hierarchy you could configure 10 summary addresses at each level or you could allocate all 50 summary addresses to the bottom level only.

If all 50 summary addresses have been configured and you need to configure one more summary address, you must first free an already configured summary address, preferably one which is unused, see section 11.8.7.

You enter ‘SHOW PNNI SUMMARY ADDRESS’ to find out how many summary addresses are still available for configuration in the active configuration image. You enter ‘SHOW FUTURE_PNNI SUMMARY ADDRESS’ to find out how many summary addresses are still available for configuration in the future configuration image, i.e. during a reconfiguration process.

11.8.7 Clearing Summary Addresses

You remove summary addresses in two steps. First you enter ‘SHOW PNNI SUMMARY ADDRESS’. This lists all the configured summary addresses whereby an index (entry index) is displayed with each summary address element listed. You then note the index value k of the summary address you want to delete. Finally you issue

```
8265ATM> clear pnni summary_address  k                                     (135)
```

When you issue ‘SHOW PNNI SUMMARY ADDRESS’, each listed summary address includes information as to whether the summary address is used or not by PNNI. It is advisable to remove unused summary addresses first.

Expression (135) removes summary addresses from the active configuration image. To remove summary addresses from the future repository, i.e. when a COMMIT is pending, you first enter ‘SHOW FUTURE_PNNI SUMMARY ADDRESS’ to obtain the index k of the summary address to be removed. Then you issue:

```
8265ATM> clear future_pnni summary_address  k                              (136)
```

11.8.8 Address and Summary Address Scope

When you configure an end system address by entering 'SET REACHABLE ADDRESS' you can also specify an organizational scope parameter. PNNI then, inhibits advertisement of the end system address at all hierarchical levels above the PNNI level corresponding to the specified organizational scope. The mapping of organizational scope to PNNI level (PNNI scope) is specified by default in PNNI standard. An organizational scope of 1 to 3 is associated to PNNI level 96, thus restricts the advertisement of the reachability to the lowest level of the hierarchy if the lowest level is 96 (default level). An organizational level of 15 permits the advertisement at any layer of the hierarchy. By default a statically defined reachable address (SET REACHABLE ADDRESS) gets a default organizational scope of 1 while an ILMI registered address gets a default organizational scope of 15.

Warning: Changing the bottom layer PNNI level from 96 to a lower value may require to change the organization scope of the defined reachable addresses to avoid unintended suppression of reachability.

PNNI associates a PNNI scope (PNNI level) to every non-suppressed summary address that is active, that is, that represents lower level summary addresses and/or end system addresses. The PNNI scope of such a summary address is the smallest numerical PNNI scope value of the summary addresses and/or end system addresses it represents.

PNNI inhibits advertisement of a non-suppressed active summary address at all hierarchical levels for which the respective PNNI level is smaller than the summary address's PNNI scope value.

One consequence of this is that an end system address is advertised beyond its scope if it is represented in the PNNI network by a summary address that also represents another end system address with a smaller numerical PNNI scope.

12. Configuring Path Selection Related Parameters

12.1 Configuring On-Demand Versus Precomputed Path Selection

On demand verses precomputed path selection IBM’s PNNI Path Selection supports ABR (Available Bit Rate) in one of two ways. Either the paths are precomputed and a specific route is obtained via table lookup resulting in fast connection set-up. Or the paths are computed on demand resulting in slower connection set-ups but with more optimization for the individual routs. The default configured setting is ‘on demand’. You can change it to ‘precomputed’ by entering:

```
8265ATM> set  pnni  path_selection  abr:    precomputed_path                   (136)
```

and change it back again to ‘on demand’ by entering:

```
8265ATM> set  pnni  path_selection  abr:    on_demand_path                     (137)
```

12.2 Configuring Shortest Path Versus Widest Path

IBM’s PNNI Path Selection also supports UBR (Unspecified Bit Rate) in one of two ways. Either the "widest path" or "shortest path" routing approach can be selected. The widest path approach finds the least loaded path in terms of bandwidth regardless of the number of hops required to reach the destination. This is an elegant approach to load balance the paths through a network in the absence of critical constraints within that network.

The shortest path approach follows a two step algorithm. In step one, paths with minimal hop count to the destination are selected. In a second step, the widest path approach is applied to the previously select group of shortest paths to select the final route. This approach is favored when the network contains critical constraints such as links (VCIs, VPIs) and/or switches that tend to become traffic bottlenecks. The drawback of the shortest path, when compared to the widest path approach, is its reduced load balancing capability.

The default configured setting is the shortest path approach. You can change it to widest path by entering:

```
8265ATM> set  pnni  path_selection  ubr:    widest_path                        (138)
```

and change it back again to shortest path by entering:

```
8265ATM> set  pnni  path_selection  ubr:    shortest_path                      (139)
```

13. Crankback Support

Note: This part does not contain the description of the crankback principles as defined in the ATM Forum PNNI 1.0 specification. Please refer to this document for more details on the standard crankback procedures.

Crankback and alternate routing are mechanisms used to cope with transient or permanent situations in order to avoid clearing the call back to the call setup source. When the call cannot be processed according to the DTL, it is cranked back to the initiator of that DTL, with an indication of the cause of the problem. This node may choose an alternate path over which to progress the call or may further crankback the call. An alternate path must obey all received higher-level DTLs and must avoid the blocked node(s) or link(s).

13.1 Crankback Process

The crankback process may be divided into three parts:

1. Crankback generation at the point of blockage,
2. Process of the crankback information during the clearing process, and
3. Upon the availability of an alternate path, reroute the call setup

The IBM PNNI Control Point support two different kinds of alternate path.

13.1.1 Alternate Route

Alternate route is the case where we must recompute a new, or a part of the, stack of DTLs.

Therefore, an alternate route is always tried at a node where a DTL has been generated. It may be either the DTL originator (where the full stack of DTLs has been generated) or an entry border node (where only a part of the DTL stack). It's relying on the crankback information which is carried back from the point of blockage to generate a new route excluding the points of failure. The alternate route is performed when all links are PNNI links, but also when an IISP link is blocked.

The number of alternate route retries is implementation dependent.

13.1.2 Alternate Link

Alternate link is the case where a new path can be found locally without violating, and therefore without changing, the stack of DTLs.

Aggregated links are defined as being links between two switching systems which have the same reachability information. The interfaces defined on these links may be either IISPs or PNNIs. In the case where the PNNI aggregation token is used, there is the additional constraint that links are aggregated if they have the same aggregation token.

An alternate link is used for parallel links between two switching nodes. In that particular case, a process allows to try all possible parallel links, locally, without cranking back the call toward the source node. This dramatically improves the latency and the network control overhead.

When all possible links have been tried or if the maximum number of retries has been reached, then the crankback is propagated toward the source. The alternate link is performed for parallel PNNI inside links, outside links and IISP links as well.

The number of alternate link retries is implementation dependent.

13.2 Crankback Control

13.2.1 Crankback and RAIG Aggregation

RAIG aggregation is a way to specify the way parallel (or aggregated) links are seen by the routing entity from the metrics point of view. Here is an example where two nodes are connected together with two parallel PNNI links. Let's assume that the first link advertises an available cell rate of 50 Mbps and a delay of 50 ms while the second link advertises an available cell rate of 20 Mbps and a delay of 20 ms.

There are 2 different views of their aggregation:

- **Optimistic View**: The result gives an available cell rate of 50 Mb/sec (aggregation shows the maximum value) with a delay of 20 ms
- **Pessimistic View**: The result gives an available cell rate of 20 Mb/sec (aggregation shows the minimum value) with a delay of 50 ms

It is clear that the optimistic RAIG aggregation may lead to have more call rejections than with the pessimistic view. For example, a call requesting 40 Mb/sec with a delay lower than 40 ms will be routed toward these links although it will not be satisfied by the local Call Admission Control.

Crankback is the solution to bypass this problem. For that reason, the RAIG aggregation selection and the crankback setting are coupled as follows:

- If **crankback** flag is **OFF**, the RAIG aggregation is **pessimistic**
- If **crankback** flag is **ON**, the RAIG aggregation is **optimistic**

13.2.2 Crankback Parameters

A set of parameters are configurable to allow the control of the crankback and of the alternate path procedures:

- Crankback flag: This allows to enable/disable the full crankback process (including crankback generation, crankback process during the clearing of the call and alternate path procedures). Indeed, sometime it's good not to run the network with the crankback enabled. For example, crankback slightly affects the call setup performance. As we've seen with the RAIG aggregation, it has also some impacts on the network behavior.
- Try Alternate Route (TAR) flag [crankback flag is ON]. This allows to enable/disable the alternate route procedures.
- Try Alternate Link (TAL) flag [crankback flag is ON]. This allows to enable/disable the alternate link procedures.
- Number of alternate route retries [try alternate route flag is ON]. This allows to specify the maximum number of retries performed during the alternate route procedures. Today, this number has been fixed to 1.

- Number of alternate link retries [try alternate link flag is ON]. This allows to specify the maximum number of retries performed during the alternate link procedures. Today, this number can be chosen between 1 and 3.

13.2.3 PNNI Port ID Handling

The process of alternate link is different depending on the presence of the PNNI port ID in the DTL.

- No port ID in the DTL: if there no port ID specified in the DTL, then we may try all possible links available. There is no aggregation token constraint in this case.
- Port ID is present in the DTL: if there is a port ID specified in the DTL, then we have to distinguish the bottom layer and the upper layers:
  - ◊ If a physical level port ID is specified in the DTL, then we must not perform alternate link.
  - ◊ If a upper level (which may correspond to an uplink) port ID is specified in the DTL, then we may perform alternate link. In that case, the selection of the physical level port ID is based on the aggregation token which must be equal for all aggregated links.

13.3 IBM Private Crankback Extensions

13.3.1 Extension For IISP Links

When a call is cleared through an IISP link, there is NO crankback information element in the clearing message (simply because a crankback information can only cross a PNNI interface). This is also called as a regular or normal call clearing message.

So, only a regular clearing message can be received in a DTL terminator switch for an IISP link. In that case, only a cause information element is present in the clearing message. Normally, the call should be cleared toward the source without crankback. What the IBM PNNI Control Point does is the following: considering that the network has been designed with at least a redundant link somewhere else in the network, a crankback will be generated from the received clearing message. For that, it filters the clearing cause and decides whether or not it can initiate crankback. If yes, then it cranks back the call toward the source, specifying that the IISP link is blocked.

From that point, the regular crankback procedures apply so that both alternate link and alternate route procedures may follow.

13.3.2 Extension For PNNI Outside Links

Let's assume that a node A2 (parent A) is connected to two different nodes B1 and B2 (parent B) with one PNNI outside link for each. Therefore, A2 and B1/B2 belong to two different peer groups. Now, let's assume that node B1 is blocked. When receiving the initial call setup from the link from A1, B1 must crankback the call toward the source. As it is an entry border node of peer group B, it cranks back to the next higher level and sends a clearing message with the blocked transit type set to node (B). Normally, node A1 can't do anything because node B is the next node to reach in the initial DTL and it is blocked.

A special feature detects that, B being blocked, A2 is connected to B through two different switches. Knowing that the link is an outside link, it is worth trying to perform an alternate link retry to reach B but eliminating the links to B1. In that particular case, it solves the problem and the call setup is successfully rerouted.

14. Interworking With PNNI Nonhierarchical Nodes

14.1 Introduction

One or more nonhierarchical nodes may be part of a peer group which is itself part of a bigger network with several levels of hierarchy. Unfortunately, for such a nonhierarchical node, it is currently impossible to reach a destination which is out of its local peer group because it has no knowledge of the links going outside its peer group. Even though, such a node can't generate a hierarchical route and the route must be generated from the source node. So, it is not possible to go out to another peer group from a nonhierarchical node.

Of course, there is a need for a mechanism which extend the current PNNI procedures so that it becomes possible to solve this problem without violating any standard procedure.

IBM PNNI Control Point implements such a procedure. The general idea is to use the exit border node as a default gateway for the calls originated from the nonhierarchical node toward the rest of the network, outside the peer group.

14.2 Detailed Description

On figure 5, let's assume that node a.3.2 is a nonhierarchical node and that it must reach a destination located in peer group A.1.

14.2.1 Exit Border Node Configuration

To define the exit border node as a gateway, we must define an additional reachability information on the outside link so that the nonhierarchical node is aware of this reachability. A problem is that a reachability already exists for the hierarchical nodes (normal reachability obtained through the PNNI hierarchy). Of course, these two reachabilities must not interfere. The normal reachability is the length of the prefix at the peer group level. So, the solution to differentiate the priority of the additional reachability, is to define it with a length shorter than the default prefix.

This approach will work in all cases because the routing is primarily based on the longest match of the addresses. From now on, this additional reachability information is spread inside the local peer group to all nodes, including the nonhierarchical ones.

In our example, on node a.3.1, we have to define a reachability on the outside link toward peer group A.1 with a length strictly lower than the PNNI level of peer group A.1 (say 95 if PNNI level is 96).

14.2.2 Nonhierarchical Node Generates A Call Setup

When the nonhierarchical node issues a call setup to reach a destination which is outside of the local peer group, the call setup is routed to the exit border node defined as the default gateway (see before).

The route computed in this source node only comprises the identification of the route inside the local peer group. Of course, it can't be a hierarchical route because of the nonhierarchical nature of the source node. Here, the destination node in the DTL is the exit border node itself, not the node of final destination.

14.2.3 The Nonhierarchical Call Setup Is Received In The Exit Border Node

When the exit border node defined as the gateway receives the call setup request, it analyzes the route and discovers that it is the last node in the route. The standard procedures specify that the call setup must be forwarded toward an interface on which the reachability is locally defined on the switch.

In our case, it is not sufficient to simply forward the call setup without doing anything because this call setup would not contain any route information. The problem is that the target interface is a PNNI interface (outside link) and it is mandatory to have the route information in the call setup on such an interface.

In order to keep the full compatibility with the PNNI standard, the call setup processing must be modified so that the call has to be routed to the outside link. For that, a path computation is requested to find the hierarchical route from the exit border node to the actual call setup final destination.

14.3 Restrictions

Of course, this method provides a good solution for the inter working but it's as not good as a network full of hierarchical nodes.

In particular, it is clear that the crankback function cannot be performed as efficiently. This is because the DTL originator is not the originator of the hierarchical DTL. It is therefore impossible to perform a hierarchical crankback from this node.

So, in the first place, the crankback functions are not performed at all, but only for those calls which are originating from a nonhierarchical node. It means that the regular crankback procedures still apply for all other calls. Note that regular crankback procedures still apply within the local peer group.

14.4 Conclusion

Only one of the nodes of the peer group must implement these features in order to provide the inter working function. It is the node defined as the exit border node. All the other nodes (nonhierarchical and hierarchical nodes) in the peer group are not modified.

Note that this implementation works with IBM products implementing the IBM PNNI Control Point. Some other interoperability issues may appear when trying to interwork with products from other vendors.