ATM PNNI Routing

S-38.130 Postgraduate Course in Telecommunications

27.11.1996

Risto.Mononen@iki.fi

Abstract

ATM Forum has defined Private Network-to-Network Interface (PNNI) for routing between ATM switches. There are actually two protocols in the PNNI specification: a routing protocol distributing consistent network topology information between switches and signalling protocol setting up a connection. The requirements for both protocols are efficiency and scalability to large networks, support for QoS, fault tolerance in the case of a link or node failure and interoperability with external non-PNNI routing domains.

The PNNI routing protocol distributes the topology information. A logical hierarchy of routing domains and topology information aggregation are used to satisfy the efficiency and scalability requirements. For best performance the address structure should reflect the network topology in the way Classless Interdomain Routing (CIDR) prefixes in the Internet do. Auto-configuration mechanism makes the domains easily manageable.

The signalling protocol is based on the User to Network Interface (UNI) specification of the ATM Forum. Additional features handle source routing and crankback to find alternate route in the case of a link or node unavailability.

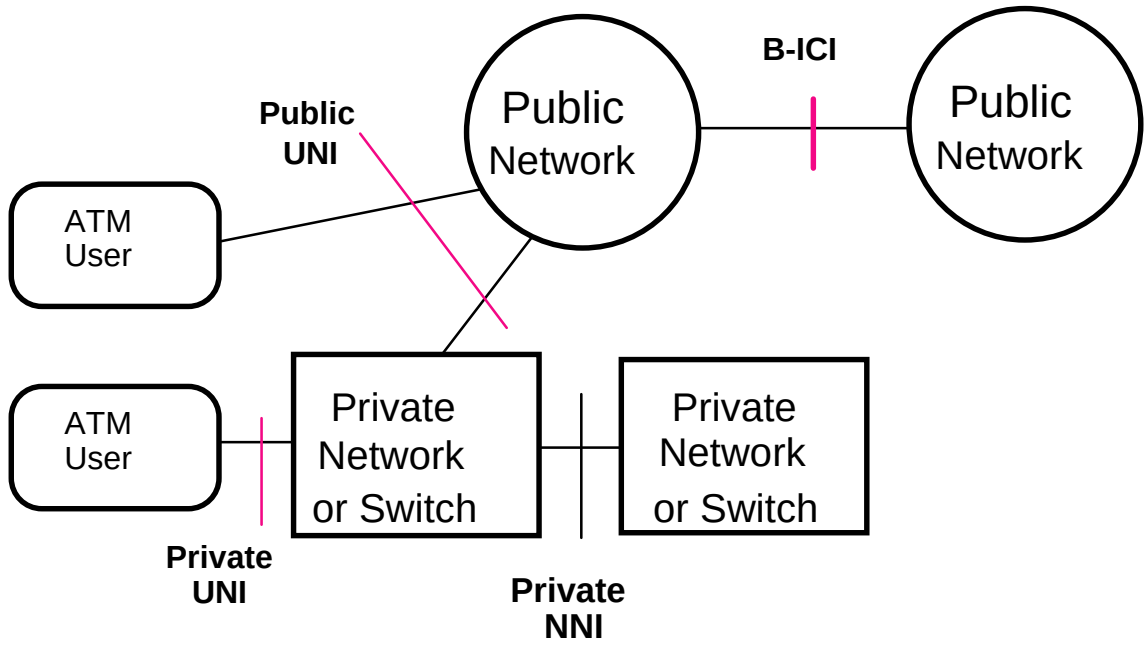
PNNI's original purpose was routing in the private ATM networks but it may prove to be useful in larger domains too. A public backbone network could be implemented internally with PNNI; or as the ATM Forum document modestly puts it: "Use of this specification by public netwoks that wish to do so is not precluded".

Contents

INTRODUCTION

In the future ATM networks there will be a wide variety in the sizes of private networks. Large corporations will have world-wide networks while a couple of switches will suffice to small organizations [1]. The former appreciate efficiency and scalability and the latter ease of use. Various policy constraints have to be applied in routing.

The use of the network will also be more heterogenous than in current nets. To ensure cost effective network usage the user QoS [2], specific requirements of a call, have to be taken into account in connection establishment and network resource reservation. A rich enough signalling scheme must be available to carry the information needed for optimal connection setup. PNNI provides solutions for both routing and signalling.



Where:

Public UNI: the user-network interface between an ATM user and a public ATM network
Private UNI: the user-network interface between an ATM user and a private ATM network
B-ICI: the network-network interface between two public networks or switching systems. B-ICI stands for B-ISDN Inter Carrier Interface.
Private-NNI: the network-network interface between two private networks or switching systems.

Figure 1: ATM network interfaces [3]

Figure 1 shows the interfaces of private ATM networks. The private networks communicate via private NNI with each other, private UNI [4] with the end systems (end users, terminals or whatever) and public UNI with public networks. PNNI nodes inside the private networks may contain gateways to exterior routing domains. PNNI may be used even inside the public networks.

This paper describes the basic concepts of PNNI 1.0 [3] concentrating mainly in the routing protocol. There is also a short description of the signalling protocol at the connection establishment. The topics are quite closely related and they both use the concept of a *Peer group* heavily. In addition signalling uses *Designated Transit Lists* (DTLs) to inform intermediate nodes about the preferred path to the destination.

PNNI routing is *map based* like traditional link state protocols OSPF and IS-IS [1]. In signalling PNNI uses *source routing* which enables the originating switch select the QoS metrics and constraints it wishes.

ABBREVIATIONS

ABR Available Bit Rate

ATM Asyncronous Transfer Mode

CBR Constant Bit Rate

CIDR Classless Interdomain Routing

DTL Designated Transit List

PNNI Private Network-to-Network Interface

PTSE PNNI Topology State Element

PG Peer Group

PGL Peer Group Leader

QoS Quality of Service

UBR Unspecified Bit Rate

UNI User to Network Interface

VBR Variable Bit Rate

VCC Virtual Channel Connection

CONCEPTS

Peer group

The concept of a peer group is the basis for aggregating the topology information. A peer group is a set of nodes that share a common view of the network. The shared view means both the group's internal topology and the external world. For efficient aggregation the nodes in a peer group shall have a common address prefix like CIDR prefixes in the Internet. The prefix is a configuration parameter.

Each peer group has a leader who aggregates the topology information and distributes it outside. The leader also collects and floods outside information to its own group. Aggregation helps keep the routing databases small enough for efficient handling.

Several peer groups may be collected to higher level groups. Grouping these nodes to even higher level groups leads to a tree like hierarchy. At each level the "address resolution" decreases; the upper levels handle just a prefix of a child group's address. Group leaders represent the whole group at the upper levels. The address prefix length identifies peer group level in the hierarchy; hence the smaller the level number is the higher the group is located in the tree. Parent - child - ancestor - descendant terminology is also often used.

The group size is expected to be about 30 to 50 switches in practice [1]. Tens of switches in a group should be handled easily by PNNI, hundreds will work very badly if at all.

Designated Transit List

A path through a peer group is called a Designated Transit List. Connection setup uses a stack of DTLs for source routing and selecting an alternate route if failures occur (crankback). The setup messages carry the DTLs and a pointer to the current position. The nodes move the pointer as the message advances or takes backward steps on crankback.

ROUTING

The functions of the PNNI routing protocol include [3]:

- Discovery of neighbors and link status. This is accomplished by exchanging Hello packets on point-to-point links.
- Synchronization of topology databases. Each peer group floods the topology information horizontally inside the group in PNNI Topology State Elements (PTSEs) to keep the databases consistent.
- Election of PGLs. PTSEs contain information to select the peer group leader without network operator interaction.
- Summarization of topology state information is the group leaders task.
- Construction of the routing hierarchy. Peer group leaders deliver summarized topology information to parent groups.

PNNI routing uses the 19 most significant octets of ATM End System Addresses. The actual use of the octets is a configuration parameter. Address prefixes reflect the logical hierarchy of the network. The end systems may use the remaining one octet.

The following discussion is based on the assumption that the addressing follows strictly the logical peer group hierarchy. This is not always the case but simplifies the representation. Actually PNNI allows *foreign addresses* reside in a group if eg. a legacy addressing cannot be changed. Perhaps even restricted mobility inside a lower level group could be implemented with the foreign address mechanism.

Hello protocol

Neighbor nodes find each other by exchanging Hello packets on a well known VCC. The packets contain the node's ATM End System Address, node ID and port ID of the link. The packets also contain the peer group IDs so that the nodes can determine if they belong to the same or different groups. Group membership knowledge will be used later when forming a hierarchy of peer groups.

The nodes create their initial topology information database from the data they gather in Hello packet exchange. The Hello protocol keeps running while the node is operational thus providing means for link failure detection.

Lowest level peer groups

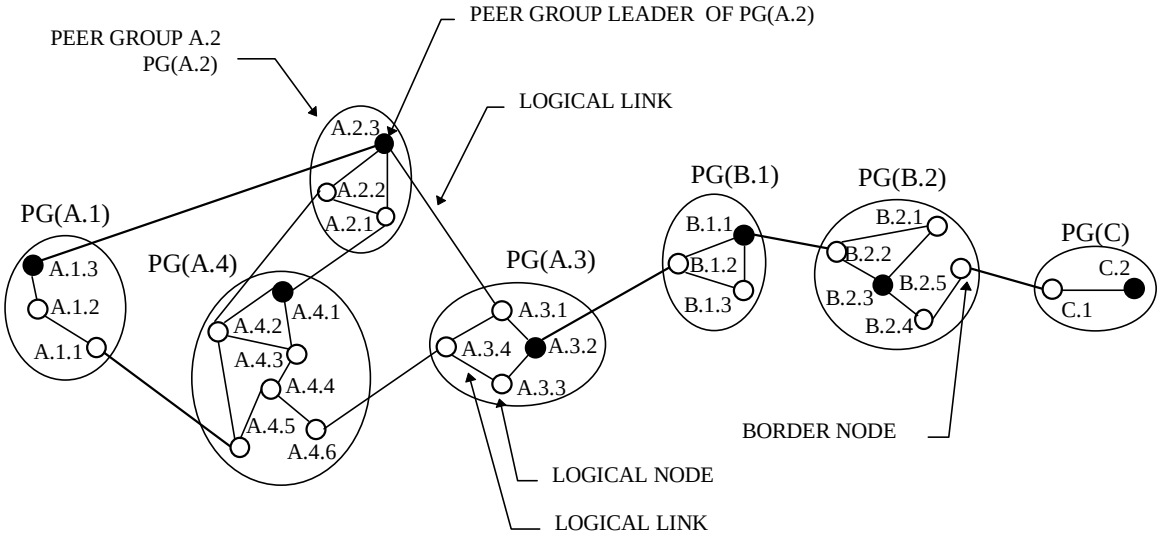


Figure 2: Partially configured PNNI hierarchy showing lowest level nodes [3].

Figure 2 shows one possible partition of a network into lowest level peer groups.

Logical nodes (links) at the lowest level peer group are exactly the physical nodes (links). In higher level peer groups a logical node means a group leader representing its own lower level group, a child group. Logical link in a higher level just means that there is a direct connection between the two peer groups.

Peer group leader aggregates the topology information of its own group and delivers it upwards in the network hierarchy. The leader receives aggregated information from the upper level peer group and floods it in the child group. This vertical information distribution is the leader's sole purpose. It has not any special role in the group's internal function. The actual connection setup (see Signalling) and data flow may take any route which the group leader may or may not be part of.

Groups and group leadership is designed to be as self-configuring as possible to minimize operator's management actions. The leader may change without operator interaction which makes error recovery fast and robust. Leader selection is based on the data collected in the topology data exchange (see PTSEs in Horizontal topology data distribution). All nodes need not be capable of becoming a leader. The selected one must have high enough leadership priority and it must know the parent peer group ID. The operator configures these parameters. A leader shutdown and re-selection does not disturb the group's internal functionality.

Border nodes connect the lowest level peer groups together. A border node receiving a call setup request from neighbor group computes the route through its own group. This border node's task is much like source routing at the originating node and is explained in more detail in the chapter Connection setup.

Horizontal topology data distribution

The nodes create ``PNNI Topology State Elements" (PTSEs) based on the result of Hello packet exchange and flood them through their own peer group. The PTSEs contain data about the originating node, its view of the topology and reachability information. The PTSE header contains data about PTSE identification, ordering and aging among other things. There is no flooding on outside logical links between the groups eg. A.2.3 - A.3.1 in Figure 2.

In the originating node part of PTSE the sender identifies himself and describes certain capabilities for routing and managing the PNNI hierarchy. These capabilities affect QoS based routing decisions and leader election process.

The topology information part contains link and node parameters. Parameters are classified as attributes and metrics. Attributes are considered individually when making the routing decisions. A security attribute on a node may prevent using certain paths. Metrics cumulate along a path, eg. delay characteristics.

Reachability information contains addresses and other peer group address prefixes to which the node provides routing. The addresses are classified as interior or exterior. The latter may contain non-PNNI domains whose addresses must not be advertised to other exterior domains to prevent loops.

An advertised link may be a *horizontal link* inside the peer group or an *uplink* to a higher level in the logical network hierarchy. The group membership information in the Hello packets lets nodes decide if a link is horizontal or uplink. The horizontal link data comes quite straight from the received Hello packets. Uplink information shall be somewhat less detailed. The border node receiving a Hello packet from neighbor group finds out the longest common address prefix to included in the uplink PTSE. Uplink PTSE thus hides some details of the neighboring group which helps keep the database size reasonable and adds some security by hiding the neighbor border node's exact address so that it can't be attacked directly.

QoS support

The topology parameters are needed when considering the quality of service the user requests[1]. Table 1 shows the currently specified parameters. The parameter set structure allows extensibility in the future

Table 1: Topology States Parameters

Topology Metrics	**Topology Attributes**	
	Performance/Resource Related	**Policy Related**
Cell Delay Variation (CDV)	Cell Loss Ratio for CLP=0 (CLR_0)	Restricted Transit flag
Maximum Cell Transfer Delay (maxCTD)	Cell Loss Ratio for CLP=0+1 (CLR_{0+1})	
Administrative Weight (AW)	Maximum Cell Rate (maxCR)	
	Available Cell Rate (AvCR)	
	Cell Rate Margin (CRM)	
	Variance Factor (VF)	
	Restricted Branching Flag	

as more experience on the needs is gained.

The requested service category (ABR, CBR, VBR, UBR) defines whether certain parameter is required or applicable at all. Eg. CDV may only be used with CBR and real time VBR.

The delay metrics in the table are quite self-explanatory if one is familiar with the ATM concepts. The Administrative Weight (AW) metric is a "generic" metric in the sense that the network operator may set it to indicate desirability of using a node or link. The reason for setting AW to certain value may be whatever the operator has decided but it should not reflect properties expressable in other parameters.

Resource Availability Information Group (RAIG) parameters reflect the properties of a node, link or reachable address. All the parameters in the table except Restricted Branching and Restricted Transit Flags are RAIG parameters and may be used on any of the topology elements.

The flags are nodal information. Restricted Transit Flag indicates a non-transit node; complex exception conditions may allow transit for certain traffic. Restricted Branching means the node doesn't support branching points for point-to-multipoint calls.

[1] Jung [2] presents various views of QoS: subjective measure of channel quality, network's performance bounds or its availability have all been discussed as quality of service. He sees QoS as the end user's view of a service as opposed to network provider's view.

Forming the hierarchy

The PNNI logical node hierarchy provides means for aggregating the topology information to enable efficient and scalable routing and signalling. Establishing and maintaining this hierarchy is where the group leaders role as the group representative becomes visible. The lowest level peer group leaders create a parent peer group[2] in much the same way the bottom group was formed. The logical nodes at the new level, bottom level leaders, share summarized reachability and topology information from their own child peer groups. Basic summarization means finding the longest common address prefixes in the child group and the uplinks advertised at the lover level ie. the links between the groups. Group internals are lower level's secrets. Further filtering may be applied to suppress information delivery up to certain maximum level in the peer group hierarchy. Finally one of the nodes is selected to be the peer group leader of the parent peer group and the group's representative on the higher level(s).

Forming higher level peer groups continues recursively until the whole logical tree has been constructed (Figure 3). Each level's group leaders represent the child group at the higher level and always one of them becomes the new group's leader. On the top there must be single node which leads a group at all levels from top to bottom. However, the hierarchy may be, and in practice it will be, asymmetric so that the top nodes may skip some level to avoid too many "leader positions". PG(C) position in Figure 3 shows the asymmetry of the tree. Furthermore, the specification assumes less than ten hierarchical levels is enough even in international networks so that no node should get overloaded from too many leader jobs.

[2] I would like to call it a "second level group" but PNNI [3] counts the levels from the top. Here were creating the group just above the bottom.

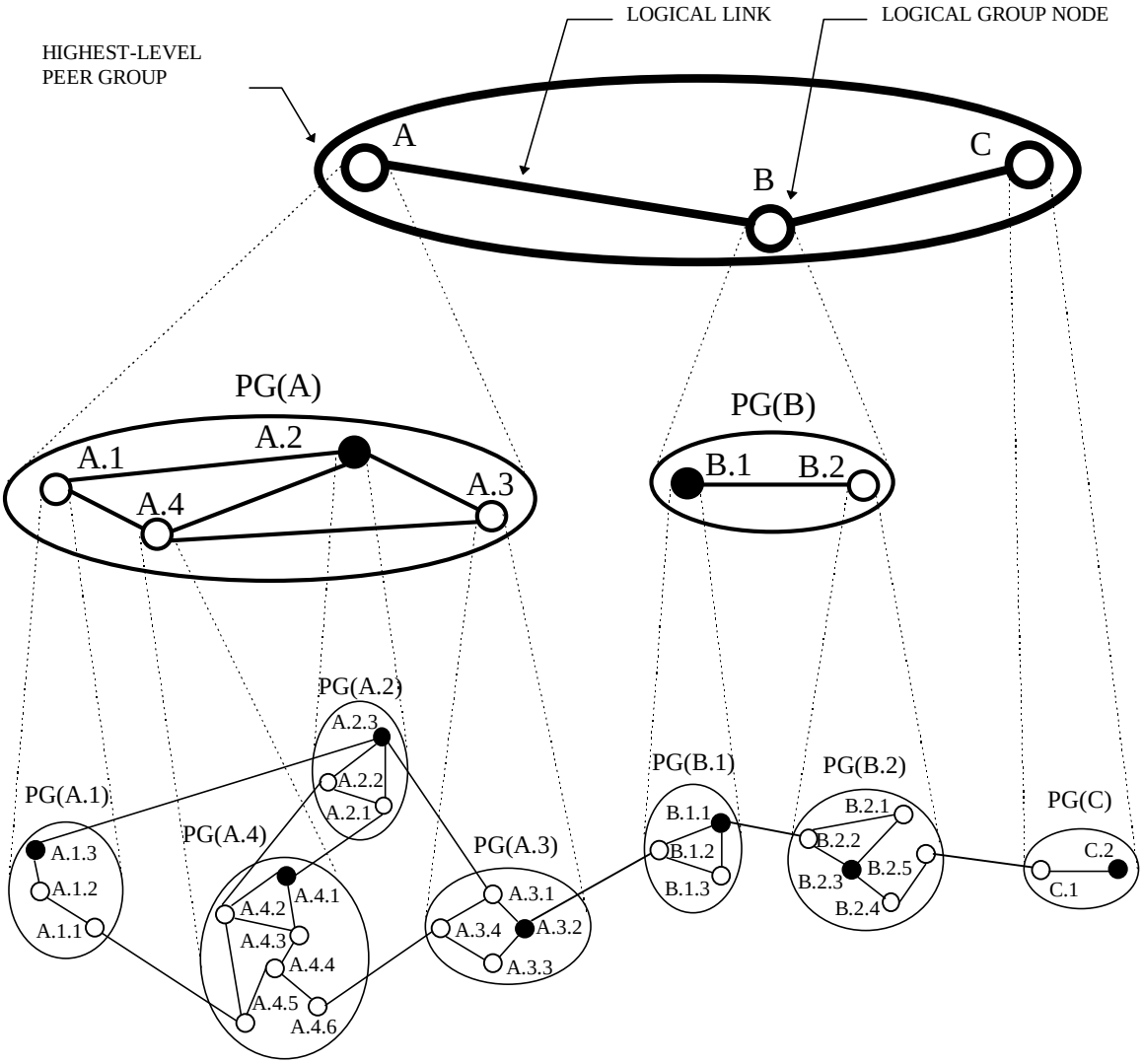


Figure 3: Example of a complete PNNI hierarchically configured network.

The Hello protocol runs on the higher level peer groups in addition to the lowest level. Its purpose is to monitor the status of PNNI routing control channels.

Logical hierarchy asymmetry may cause a lowest level border node to be a peer to a Logical group node (LGN). Such peer relationship doesn't affect the Hello protocol. It does however complicate the uplink topology. In Figure 3 node C.1 finds and advertises uplink to logical node B which is in the lowest common peer group with B.2.5. B.2.5. does the same within its group and *induces* an uplink from B.2 to C too. The induced uplink aggregates in PG(B) all the links between PG(B.2) and PG(C).

Complex node representation

Topology aggregation reduces both nodal and link information to be addressed. Link aggregation simply means advertising several uplinks in the child group as a single logical link in the parent. Logical nodes above the bottom level require a more complex representation. The reason is that though a simple weighted graph, like PG(A), describes the possible higher level PG paths between the logical nodes, it gives no information on the cost of traversing through a logical node. In other words the intergroup paths may be prioritized based on the higher level picture but occasionally the decisive factor lies inside child PGs.

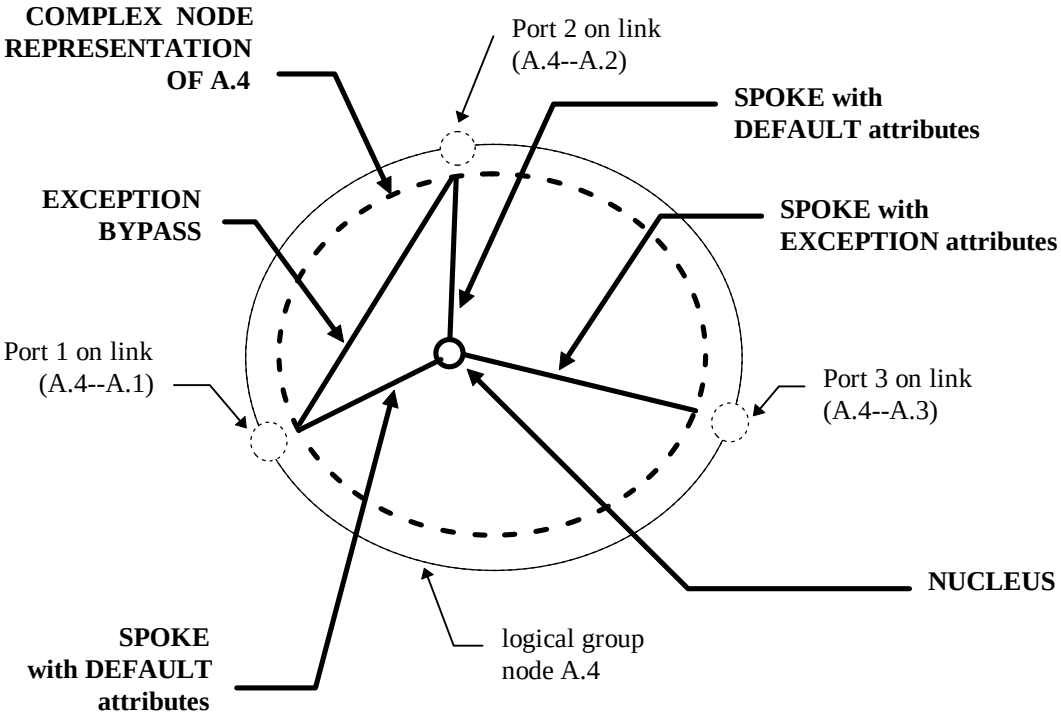


Figure 4: Complex node representation of LGN A.4 [3]

PNNI encodes the logical node topology information into a *complex node representation* as shown in Figure 4. The default is to show the logical node as a *nucleus* (star point) with *spokes* to ports for each logical link. The spokes may be given *exception attributes* to indicate specially good or bad choises. *Exception bypass* means a shortcut from one port to another.

Example: A slow link (A.4.6--A.4.4) in Figure 3 would cause setting the exception attributes between the nucleus and port 3 in Figure 4 telling PG(A) that routing traffic from A.3 is possible but probably not what is wanted. On the other hand a very fast link (A.4.2--A.4.5) would be expressed with the bypass making A.4 a good route for (A.1--A.2) traffic.

The complex node representation in nodal information aggregation is really quite complex. Perhaps a simpler scheme could do the same. The essential thing is to hide the most inefficient links from being used for transit purposes. Probably just dropping out the link in question from the PTSEs delivered to the parent group would do. Advertising the inefficient ones could be done only if the parent group becomes disconnected otherwise; then the hidden ones pop-up connecting the group again. Or maybe the Restricted Transit Flag should be allowed on the links too.

Vertical topology distribution

A logical node in a parent peer group shows the outside logical links of its child group to the other parent group's logical nodes. All the parent group nodes update their own database according to received PTSEs and flood the information to the other members of their own child groups. Thus the lowest level nodes shall have a common view of both the bottom group's internal topology and that of ancestor groups' in the tree.

Example. In Figure 3 node A.1.3 as PG(A.1) leader and logical node A.1 lets the other PG(A) logical nodes know that there is a connection from PG(A.1) to PG(A.2) and PG(A.4). Information aggregation means it does not tell exactly *how* the links reside inside the child group. Just the precense of connections and possibility to route between PG(A.2) and PG(A.4) is advertised.

SIGNALLING

Signalling protocols setup and clear connections between end nodes. PNNI signalling is based on a subset of UNI 4.0 [4] signalling. In addition to UNI 4.0 features PNNI signalling gathers complete source routes to DTLs and uses crankback to cope with obsolete topology information.

Designated transit listst (DTLs)

Designated transit lists are the core of the PNNI signalling protocol. They specify a complete path across a peer group and a pointer to the next node. The source node creates the original DTL and the entry border nodes update it each time a peer group border is crossed.

The path information in DTLs reflects the logical peer group hierarchy. The data consists of paths ordered as a stack. The topmost path represents a path through a lowest level peer group. The deeper a path is in the stack the higher it is in the logical hierarchy. The bottom of the stack contains a path through the lowest peer group common to the source and destination.

Connection setup

The calling party first tries to find the destination from the own peer group. If both the source and the destination are in the same group their routers can immediately compute the most appropriate route to the destination since the group has synchronized its database. In case of differing lowest level peer groups:

1. The source node finds the lowest common peer group and creates the original DTL stack. A path through the common group is pushed into the empty stack. Then a path through each PG level down to the bottom level is pushed initializing the stack.

2. The node sends connection setup request via the path on the top of DTL. Intermediate nodes forward the request according to the topmost DTL path and update the DTL pointer. The path is exhausted and removed at the border of the lowest level peer group.

3. A border node in the neighbor group receives the request. It examines the topmost DTL path and forwards the request. Forwarding is trivial if the destination is in the same group. Otherwise compute a path through the own lowest level peer group to a border node with a link to the right direction. Computation uses the current topmost path and local peer group topology data. Push the new path to the DTL stack and repeat step 2.

Crankback

The source router and the peer group border nodes are the only ones who normally generate new components to the DTL. However, in case of obsolete topology data, eg. a decreased link or node availability which has not yet been advertised, it is possible that a *crankback* procedure is needed to find an alternate route. Any node on the path may have to make new routing decision in such a situation. The crankback may take several steps back which is why the DTL needs to have a memory in the form of stack of paths. The crankbacking node could compute a new path from the scratch based on the destination address only but in that case the original policy in the routing decisions would be lost.

CONCLUSIONS

PNNI seems to be adequate solution to routing between large private ATM subnetworks. Probably it will be used in the public networks too. Co-working with external routing domains will be crucial to its fast adoption. Especially connections to old Internet hosts should be fluent. The specification is new and real field experience on its use is still rare. Recent IETF work on the Internet protocols has obviously given a lot to PNNI specifiers too [1]. Nimrod was mentioned in particular and such a background should guarantee that the behavior of the new protocol is what was expected.

REFERENCES

[1] Joel M. Halpern. The Architecture and Status of PNNI. http://www.vivid.newbridge.com/documents/Joel.html, Newbridge Networks Inc., 1996.

[2] Jae-Il Jung,.Quality of Service in Telecommunications. IEEE Communications magazine, August 1996, Vol. 34, No. 8, pp. 108-118.

[3] The ATM Forum Technical Committee. Private Network-Network Interface Specification Version 1.0 (PNNI 1.0). af-pnni-0055.000, March 1996. 379 p.

[4] The ATM Forum Technical Committee. ATM User-Network Interface (UNI) Signalling Specification Version 4.0. af-sig-0061.000, July, 1996. 129 p.

[5] Marko Luoma. Classless interdomain routing - CIDR. http://keskus.hut.fi/opetus/s38130/s96/cidr.pdf, 15.10.1996, 9 p.