PNNI - Private Network to Network Interface

- Principles
- Topology concepts
- Routing Protocols
- Topology aggregation
- Call setup and routing algorithm

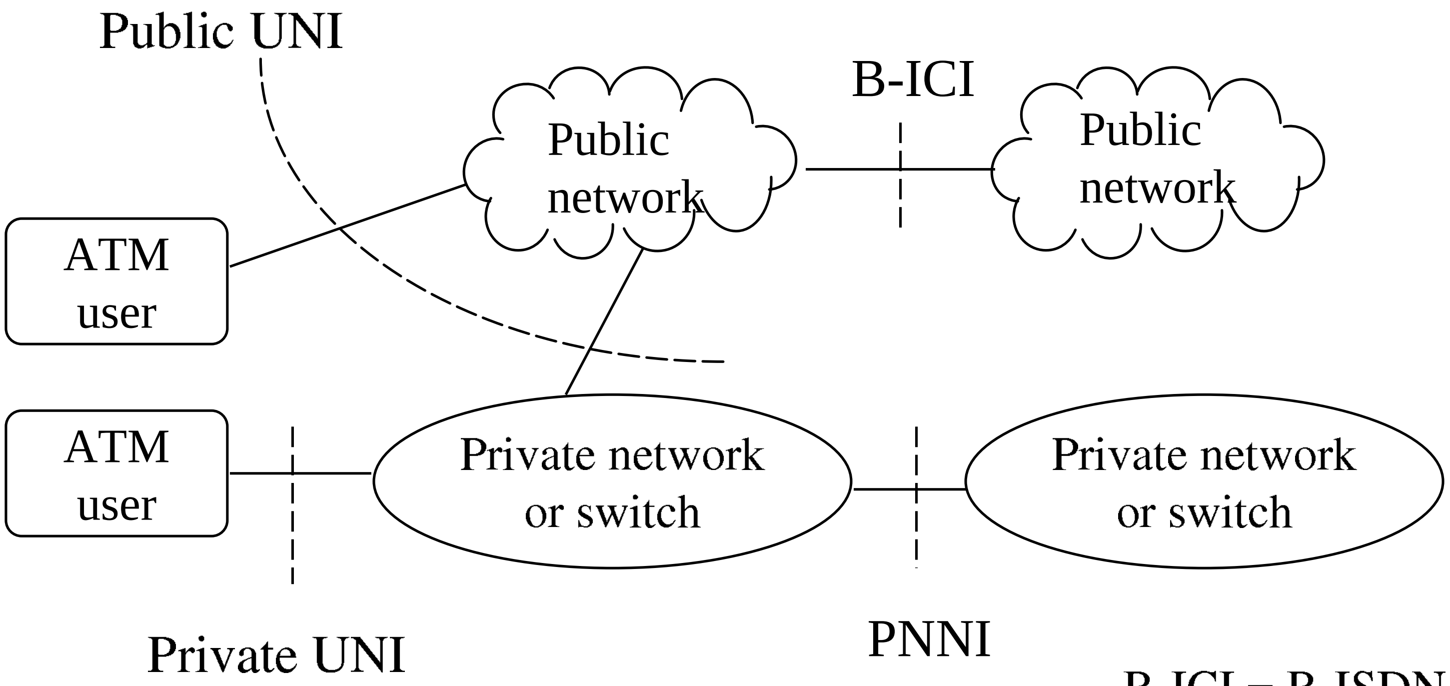
Private-Network-to-Network Interface (PNNI) is ment for interconnection of private network ATM switches

- PNNI includes both a routing and a signalling protocol.
- Requirements include scalability, efficiency, QoS support, fault tolerance in case of link and node failures and Interoperability with other protocols.
- PNNI-routing, like OSPF routing, is based on network topology information which may be aggregated. PNNI supports hierarchy.
- PNNI signalling is inherited from the ATM-Forum UNI signalling. Additions are source routing and crankback.

Why?

Why is PNNI based on source routing?

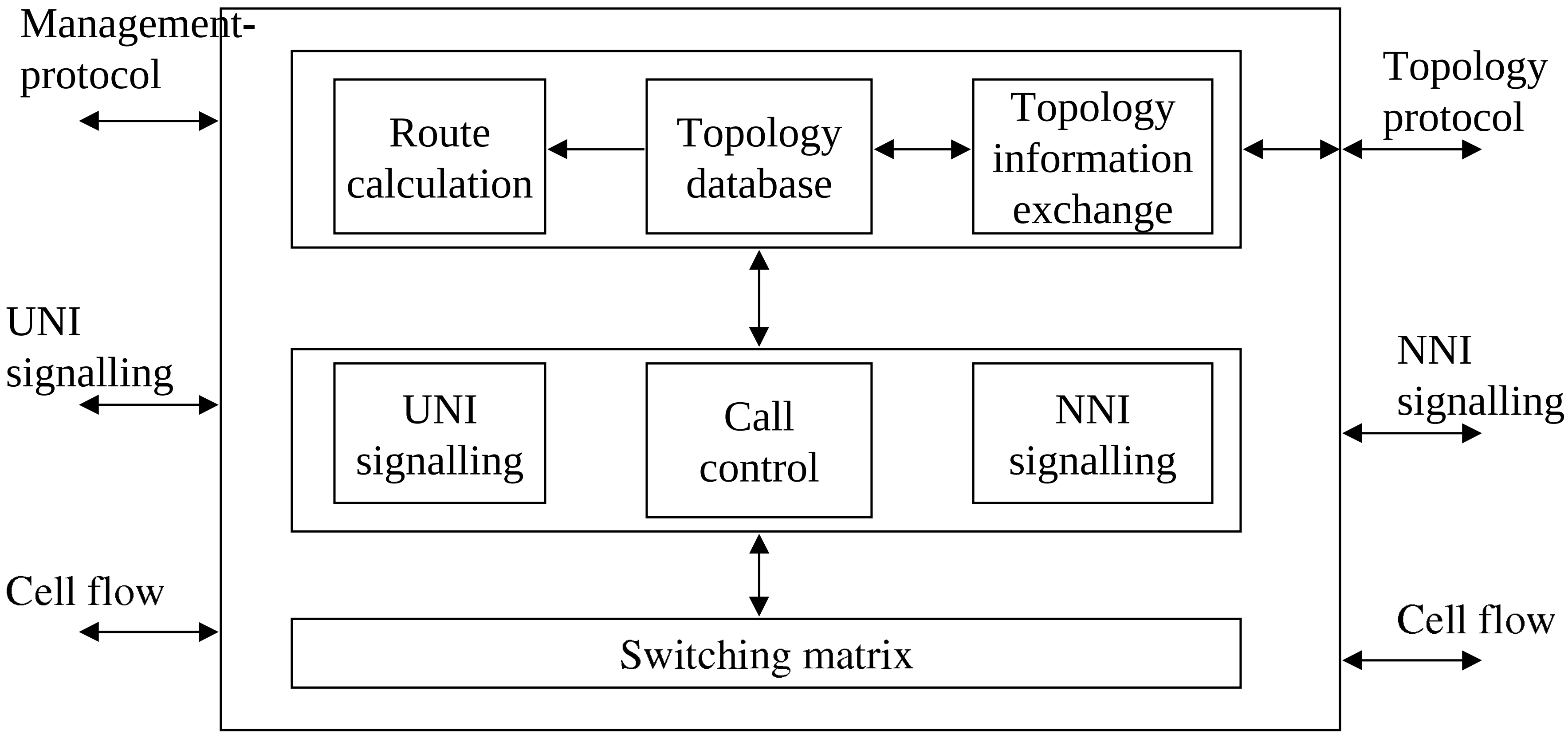
In the ATM Forum Model PNNI interconnects private networks



B-ICI = B-ISDN Inter
Carrier Interface

PNNI 1.0 specs is af-pnni-0055.000, updated in march 1996, more than 365 pgs.

PNNI node reference model

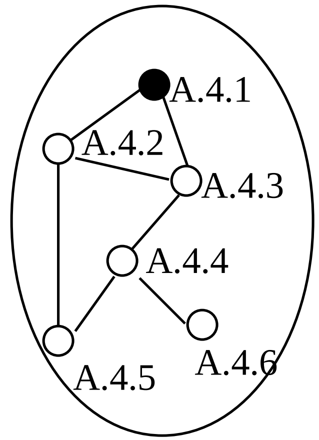


PNNI routing functions include

- Finding neighbors, links and link states using the Hello -protocol. *Establishment of Peer Groups*.
- Synchronization of the Topology databases by exchanging PNNI Topology State Elements (PTSEs) horizontally inside a peer group.
- Election of Peer Group Leaders (PGL) based on PTSEs.
- Aggregation of Topology information (task of PGL).
- Building up the routing hierarchy (PGL passes to the parent group an aggregated description of his peer group)

Let us view the PNNI topology concepts and topology protocols

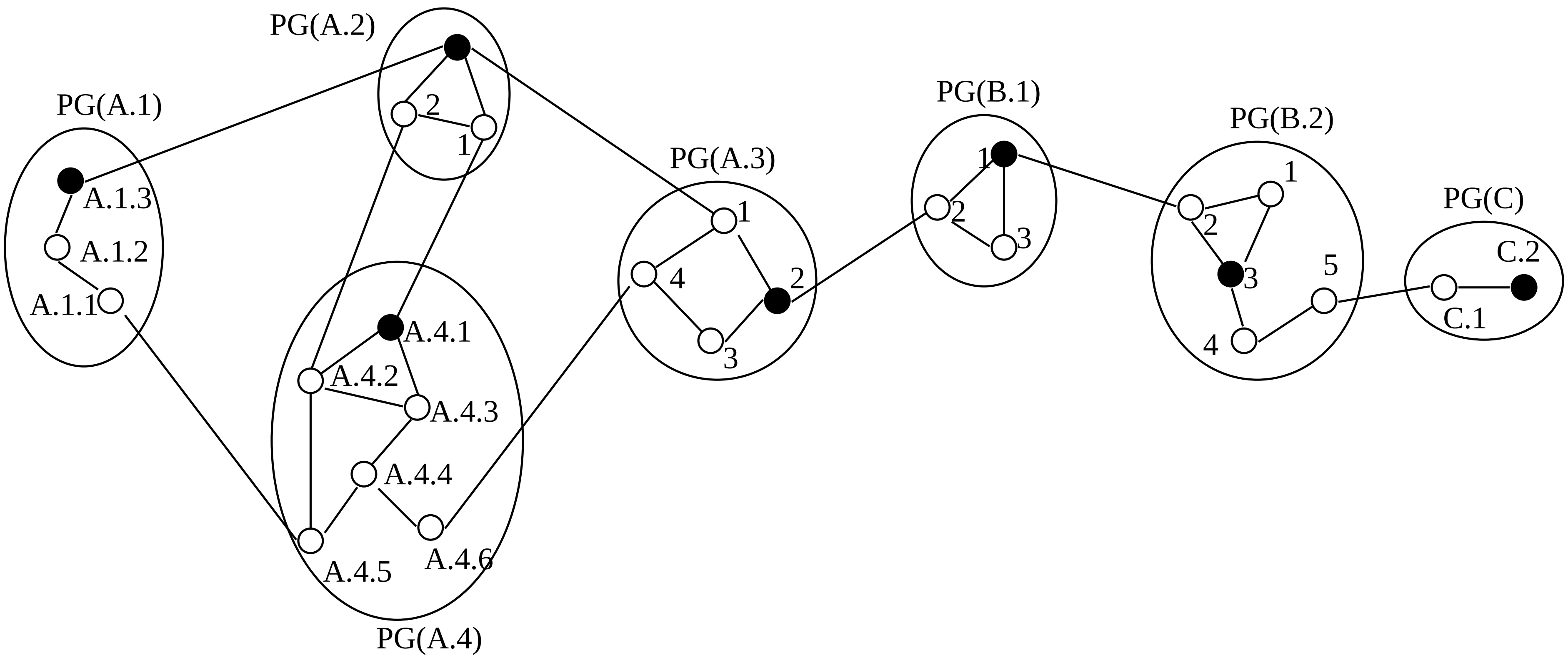
Peer Group is the key concept in PNNI routing



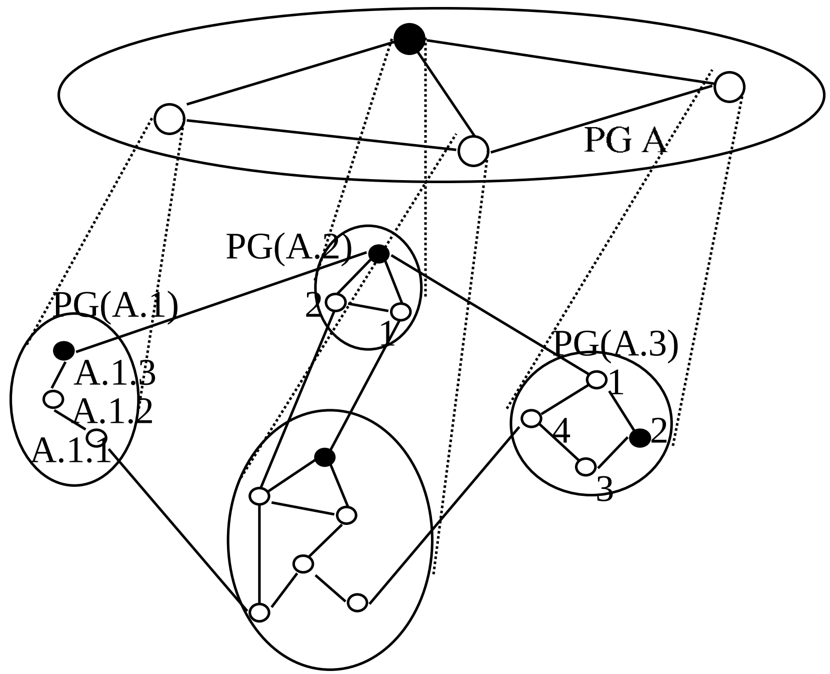
PG(A.4)

- Peer Group is a *set of logical nodes*, such that they have the same topology information. This includes both the information about the group itself as well as the description of the rest of the network.
- Nodes have a common address prefix (e.g. A.4) for the sake of efficient coding. The prefix is a configuration parameter set by the operator.
- A reasonable size of a Peer Group is max. tens of nodes (e.g. 30 .... 50).

An example topology

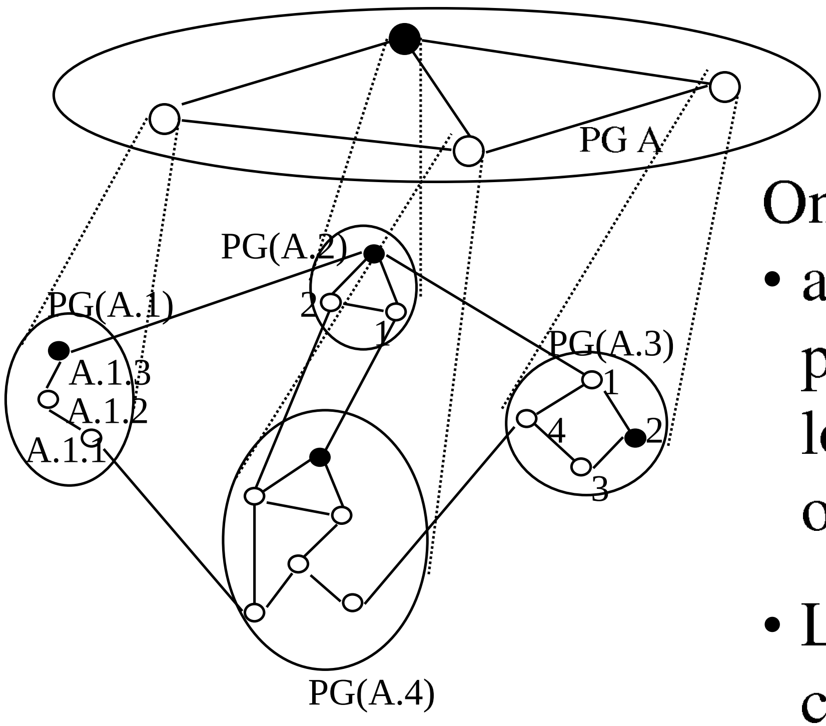


Peer Groups form a hierarchy



- PGL -peer group leader (cmp. designated router in OSPF) aggregates the description of the group and passes it up in the hierarchy to the next higher level peer group.
- PGL also receives external topology info and distributes it in its group.
- Peer groups form a hierarchy.
  Address resolution decreases up i.e. prefix becomes shorter.
  Length of Prefix tells the level in the hierarchy, numbering of levels starts from the top.

Topology consists of logical nodes and logical links



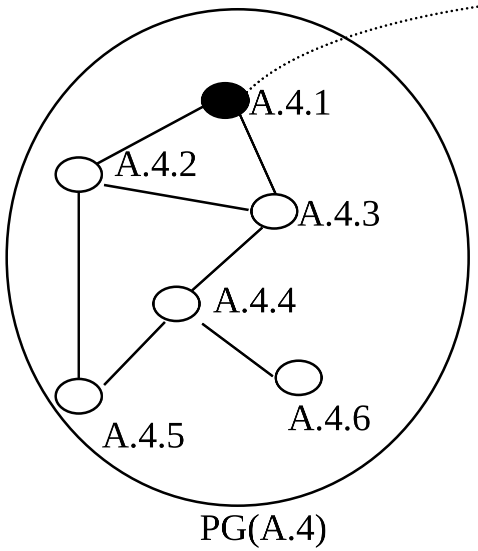
On upper levels:

- a logical node represents the child peer group. In practice the functions of the logical node are taken care of by the PGL of the child group.
- Logical link = direct link connecting child peer groups

In the lowest level Peer Group

- logical node = physical node.
- logical link = physical link

Election of Peer Group Leader is largely automatic and does not interfere setting up connections



Tasks of the PGL are
- to aggregate the group topology description
- pass it upwards in the group hierarchy
- receive topology information sent by the parent group and distribute it in its group.

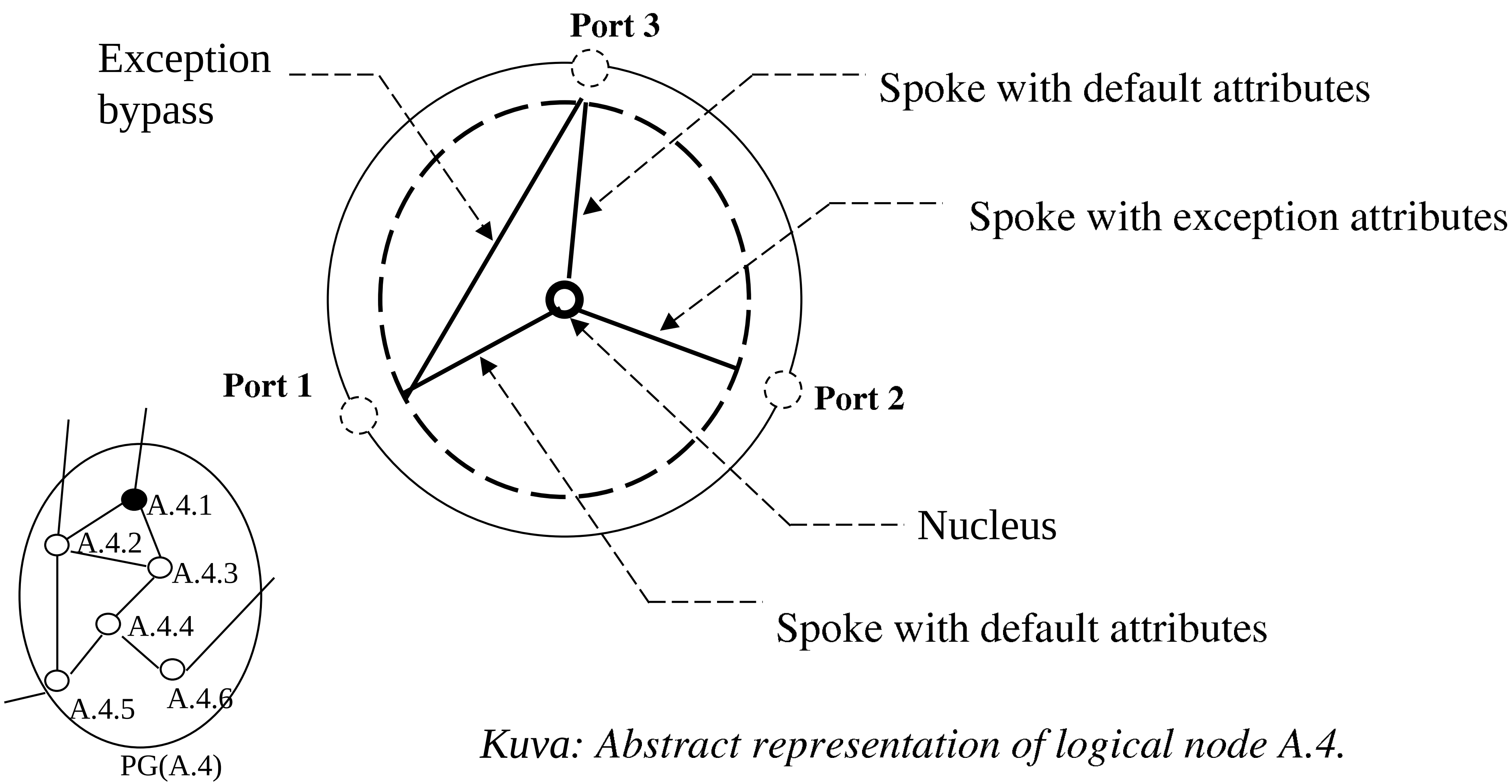
- *PGL can be re-elected automatically without operator interference.*
- *Election of the PGL is based on collected topology info.*
- *Not all nodes need to be eligible.*
- *To be elected a node needs to have a high enough priority and it must know the identity of the parent group*
- *The priority of the elected PGL is increased for stability*

PNNI Topology State Elements (PTSEs) describe the topology

PTSEs are built of data sent by the Hello protocol and distributed into the Peer Groups.

Header	*PTSE identity and order* *PTSE aging*
Sender information	*Sender identity* *Sender routing capability, eligibility and PGL priority*
Topology information	*Link (horizontal/vertical) and node parameters:* *divided into attributes and metrics*
Reachability information	*Internal and External (also non-PNNI) addresses, to which the node will route traffic*

Peer Group topology is aggregated by abstracting its real structure into a logical node

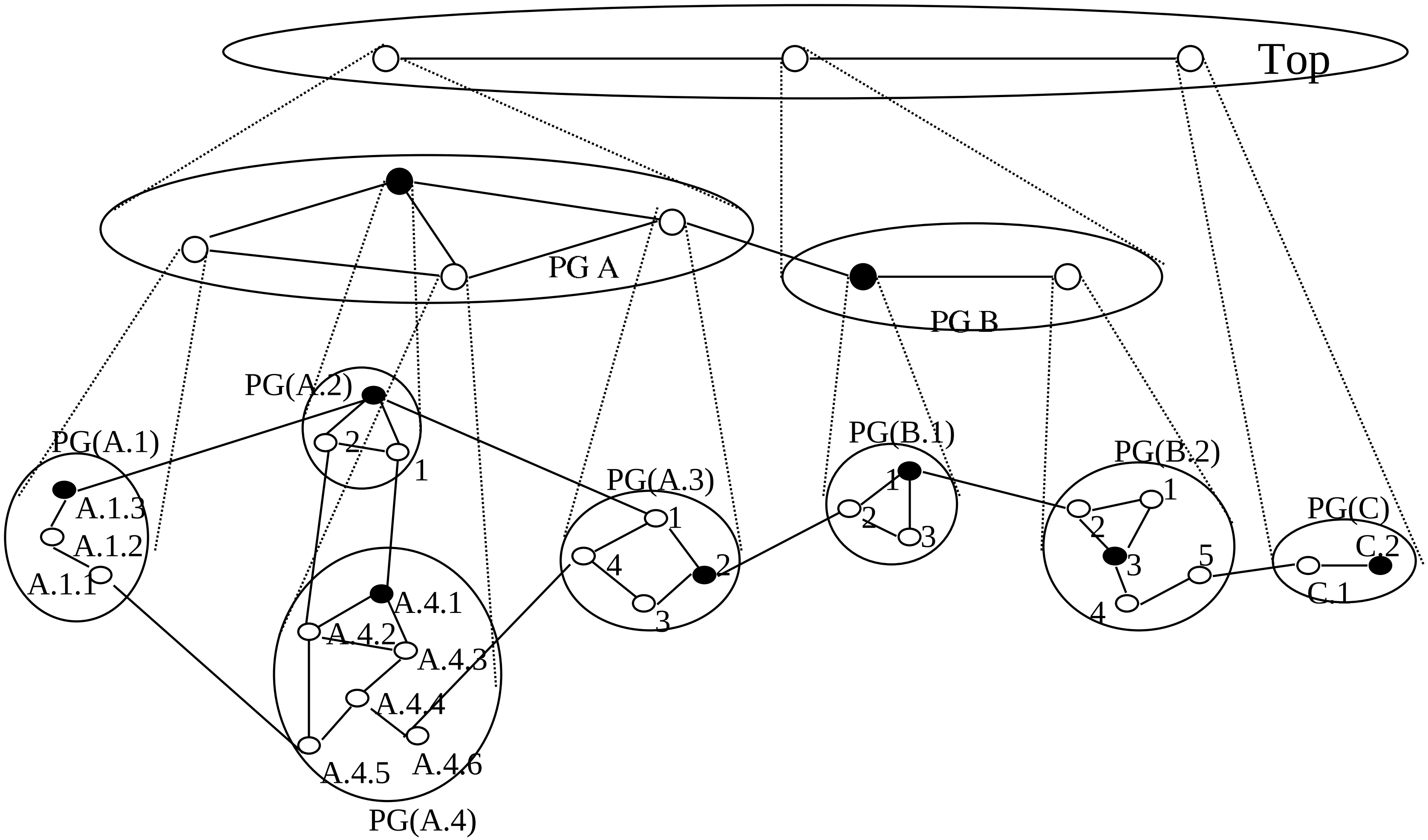


Kuva: Abstract representation of logical node A.4.

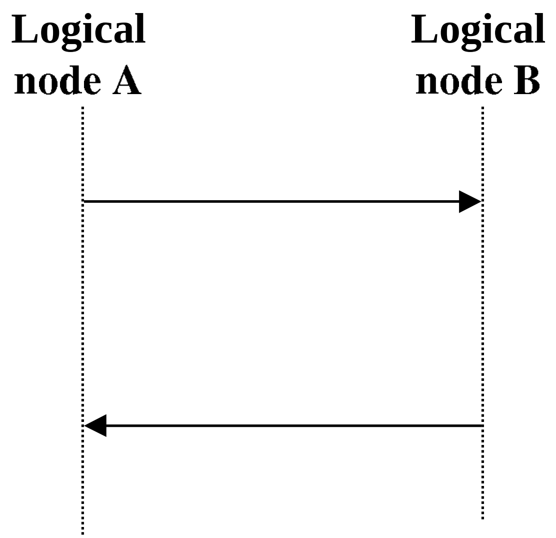
Peer Group Leaders build and maintain group hierarchy

- Bottom level PGLs build their parent peer groups (NOTE: children create their parents!!!)
- Parent peer group has a consistent topology database
- Topology of the parent group is distributed in the child groups
- A PGL is elected in the parent group
- The PGL of the parent group represents the group in the next upper level parent peer group
- Key criteria of group membership is longest common address prefix

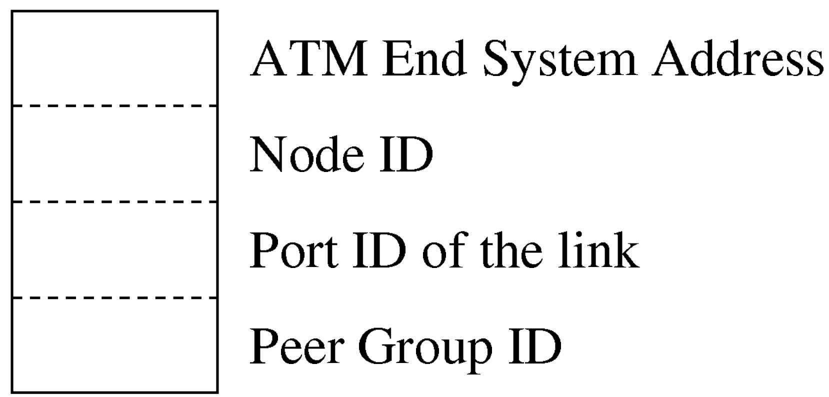
An example hierarchical topology



Hello protocol works on a well defined VCC between neighbors

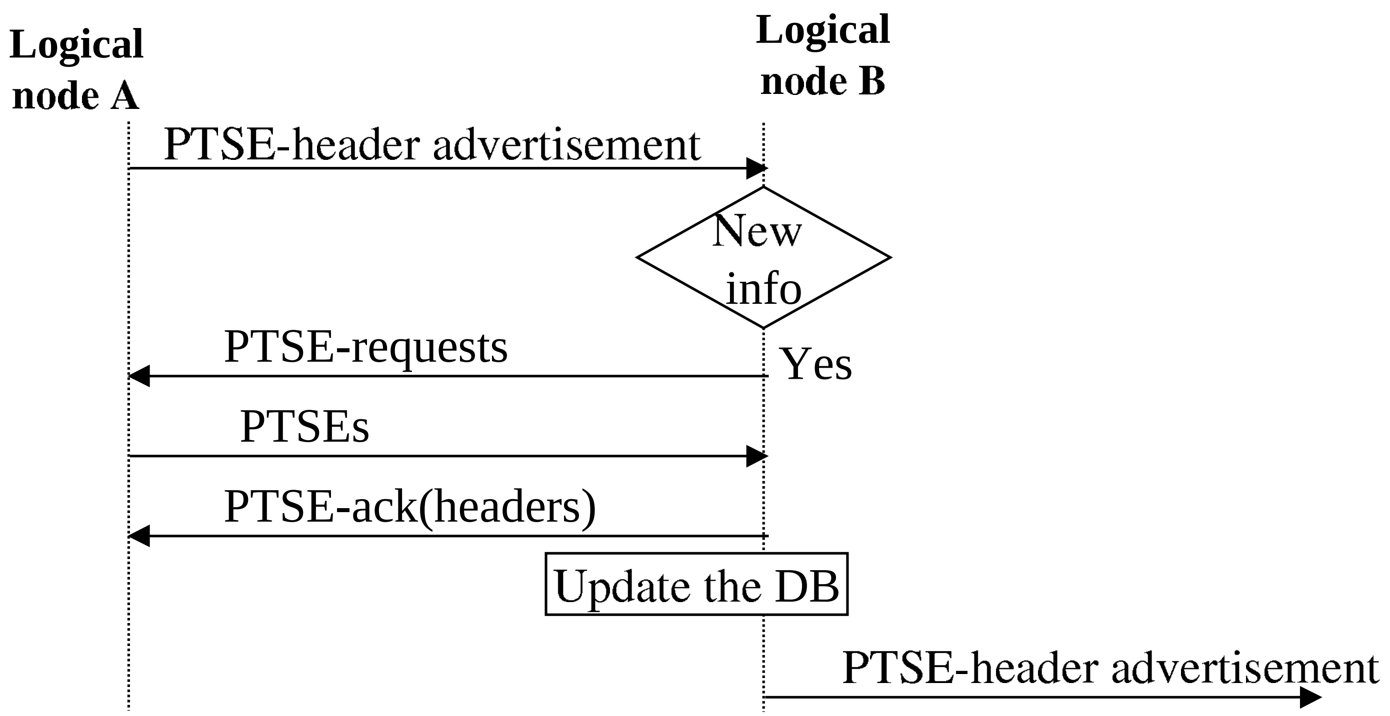


Hello packet contains

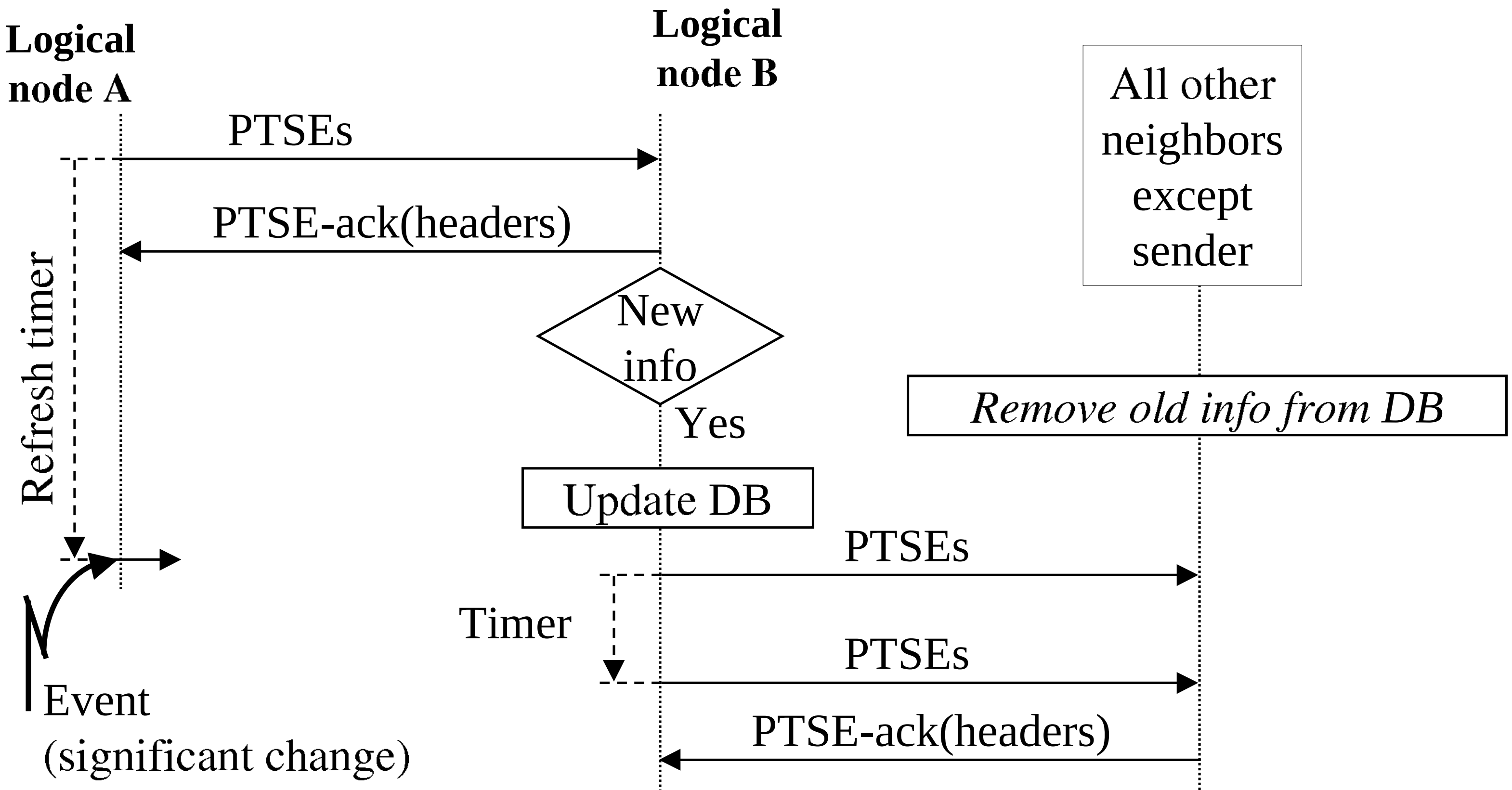


- *Hello protocol works continuously and reveals link failures.*
- *Hello protocol data is used to form the initial version of the topology database.*

When neighbors have been identified by the Hello protocol, topology databases are synchronized

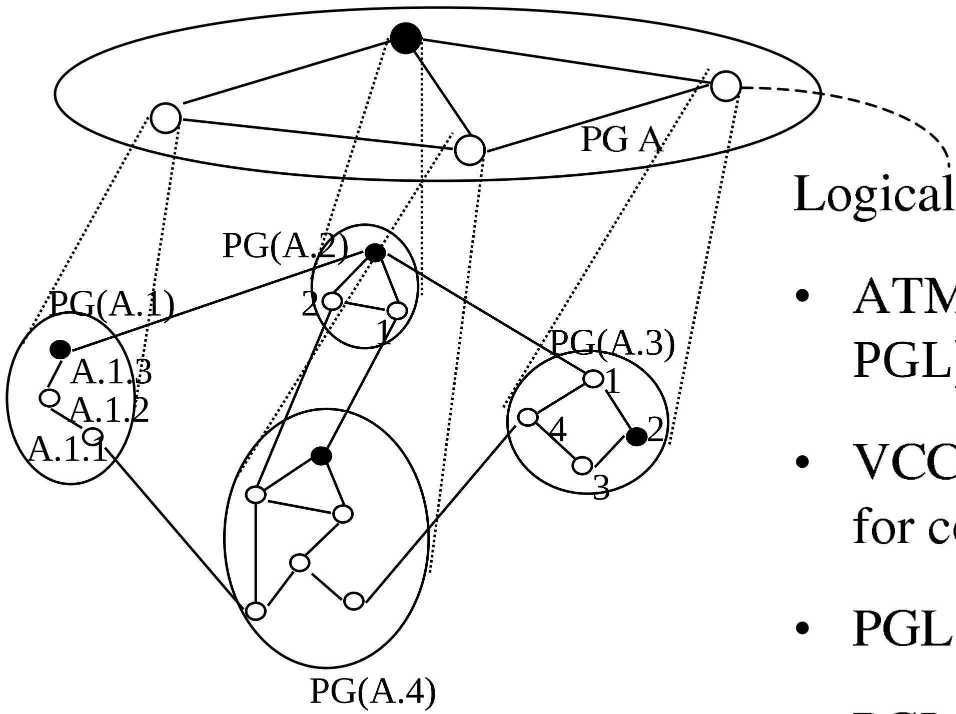


PNNI flooding protocol is similar to OSPF-flooding



- *Send frequency of PTSEs is a compromise between probability of misrouting and the need to minimize the amount of PTSE-information.*
- *What is a significant change should be configurable.*

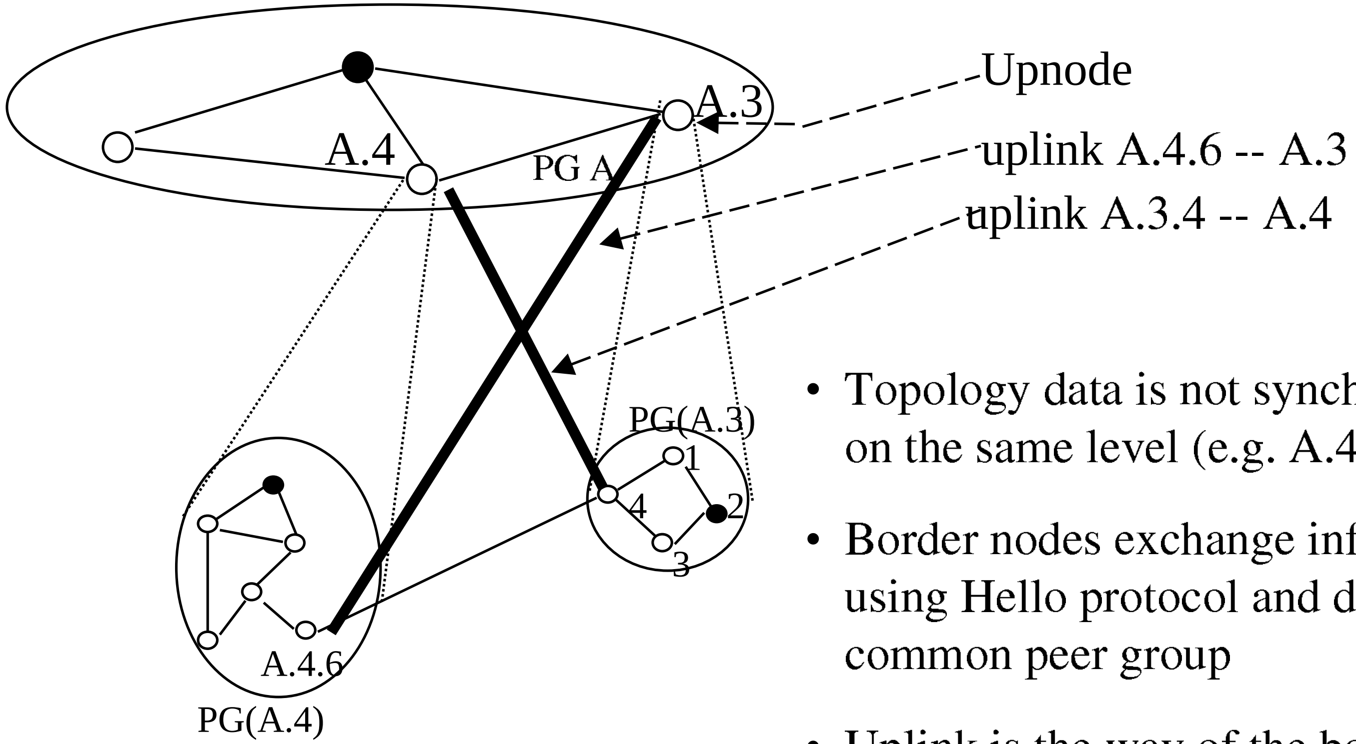
Parent peer groups are similar to lowest level peer groups



Logical group node - LGN (logical node) has

- ATM End System Address (a different SEL than PGL)
- VCCs are set up between logical group nodes for communication acc to PNNI
- PGL is elected in the parent group as well
- PGL is not needed on the topmost level.

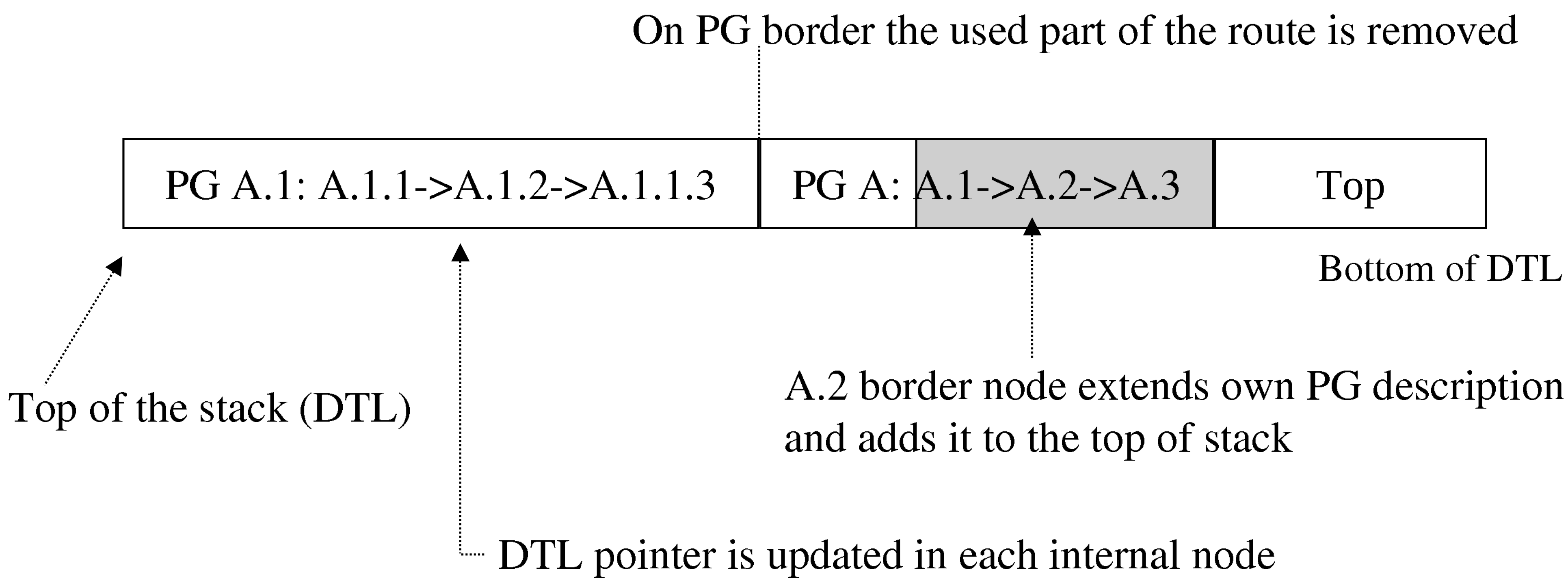
Border nodes describe connections to neighboring groups as uplinks



- Topology data is not synchronized between peer groups on the same level (e.g. A.4.6 -- A.3.4)
- Border nodes exchange information about the hierarchy using Hello protocol and deduce which is the lowest common peer group
- Uplink is the way of the border node to tell its group about a connection to the parent group
- Using uplink info (PGLs)/LGNs can set up VCCs between nodes

PNNI signaling
and
routing algorithm

Designated Transit List is a stack representation of the route e.g. from A.1.1 to C.2



DTL = designated transit list

Metrics are additive in route calculations

PNNI supports QoS routing/route optimization using metrics:

- Cell delay variation (CDV)
- Maximum Cell Transfer Delay (maxCTD)
- Administrative weight (AW)
  - administrator can define the interpretation of AW

Optimization is done using one metric at a time.

Topology attributes are considered one at a time in route calculations

Performance/resource related parameters

• Cell loss probability CLP=0 for cells CLP_0 • Cell loss probability CLP=0+1 for cells CLP_{0+1} • Maximum Cell Rate (maxCR) • Available Cell Rate (AvCR) • Cell Rate Margin (CRM) • Variance factor (VF)	RAIG information

• Restricted Branching Flag

Restricted transit flag is considered a policy parameter.
RAIG - Resource Availability Information Group

⇨ *PNNI uses dynamic resource availability Info!! Is a dynamic routing system!!*

Connections are set up using hierarchical source routing

1. If destination address is in the same peer group, source node calculates the route

2. If destination address is in a different peer group

   2.1. Source node determines lowest common peer group and forms the DTL

   2.2. Source node initiates connection setup using info at the top of DTL.
   Internal nodes update the DTL pointer.
   At PG border lowest level peer group route info has been used and is removed.
   Connection setup request is sent over the PG border.

   2.3 Receiving *border node* looks for the destination in its peer group, if found, it will calculate the route to destination. If not found, it calculates the route throug lowest level PG towards a node with a suitable external link and inserts the partial route at the top of the DTL. Continue at 2.2.

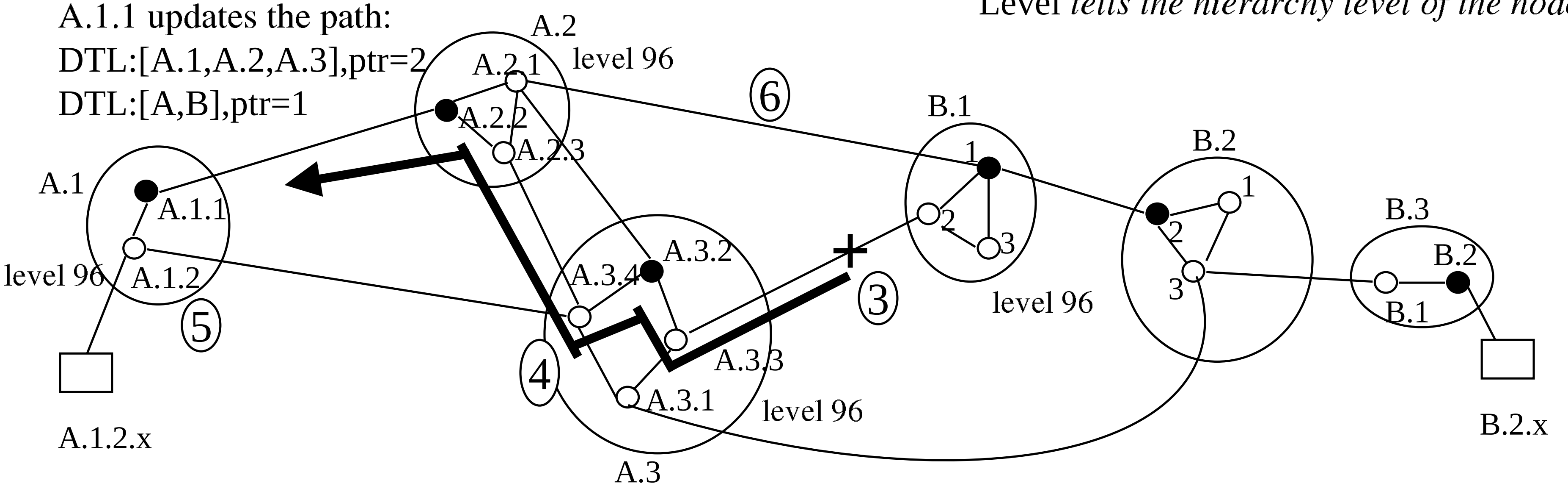
If a PNNI connection setup encounters blocking, crankback is used to try again

- Crankback may become necessary if newest topology information has not been advertised to the node that calculated a portion of the route.
- Because of crankback any node on the path may need to make a routing decision.
- Crankback returns the call in the order determined by DTL.
- Normally crankback continues to a border node, such that the original routing policy can be preserved: First to the closest border node, then until the border node of parent peer group or the source node whichever is closer, etc.

A crankback until a border or the source node

② A.1.1 updates the path:
DTL:[A.1,A.2,A.3],ptr=2
DTL:[A,B],ptr=1

Level *tells the hierarchy level of the node.*



① A.1.2 selects the path:
DTL:[A1.2,A.1.1],ptr=2
DTL:[A.1,A.2,A.3],ptr=1
DTL:[A,B],ptr=1

③ DTL:[A,B],ptr=2; call encounters blocking; RELEASE is sent back.

④ A.3.4 last changed the DTL and therefore will try an alternative route. We assume not enough resources. Crankback level is increased.

⑤ Crankback bypasses A.2.2 based on level, returns to source node.

⑥ Source node calculates a new route: A.1->A.2->B.1

Summary of PNNI routing principles

- Route calculation is done peer group by peer group
- Route is described in designated transit list - DTL, original DTL is built by the source node.
- In each PG the *Entry Border Node* updates the DTL by calculating the route though its own PG and inserting it at the top of DTL.
- Internal nodes of a PG read and execute the DTL-instruction and update the ***DTL-pointer***.
- If blocking is encountered, connection request is returned back so long that a suitable border node or the source node itself can select a new route.
- Always PNNI seeks to satisfy the ***QoS parameters*** accepted by the source node.