PNNI – Private Network to Network Interface

Principles

Topology concepts

Routing Protocols

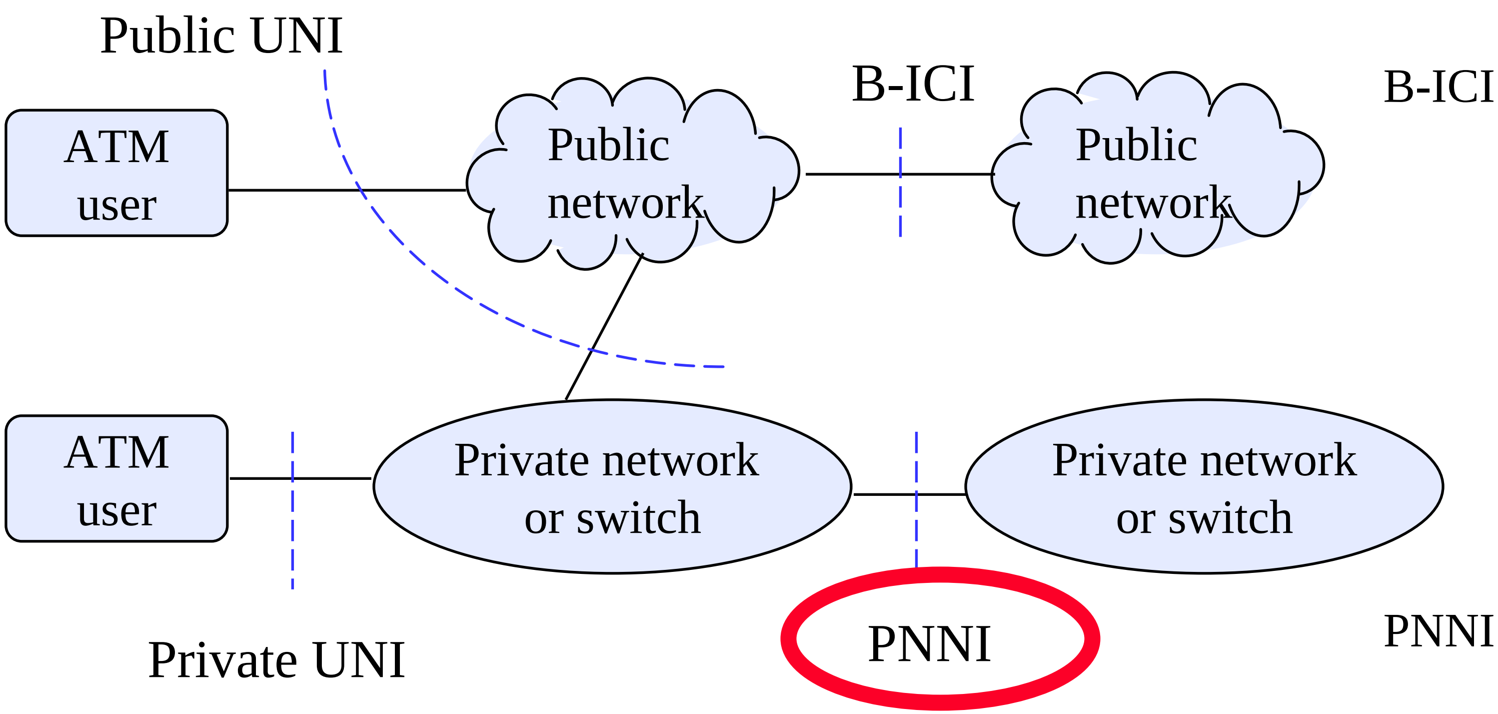
Topology aggregation

Call setup and routing algorithm

ATM background

- ATM = Asynchronous Transfer Mode
- Connection oriented
  - VCI (Virtual channel identifier)
  - VPI (Virtual path identifier)
- Information is sent in fixed-size packets, in *cells*
  - 5 bytes header + 48 bytes data → cell length 53 bytes
- Two types of interfaces
  - UNI (User-network interface)
    - Connects end user with switch
  - NNI (Network-network interface)
    - Between two switches
- Both UNI and NNI can be divided into a private and a public version

In the ATM Forum model
PNNI interconnects private networks



B-ICI = B-ISDN Inter Carrier Interface

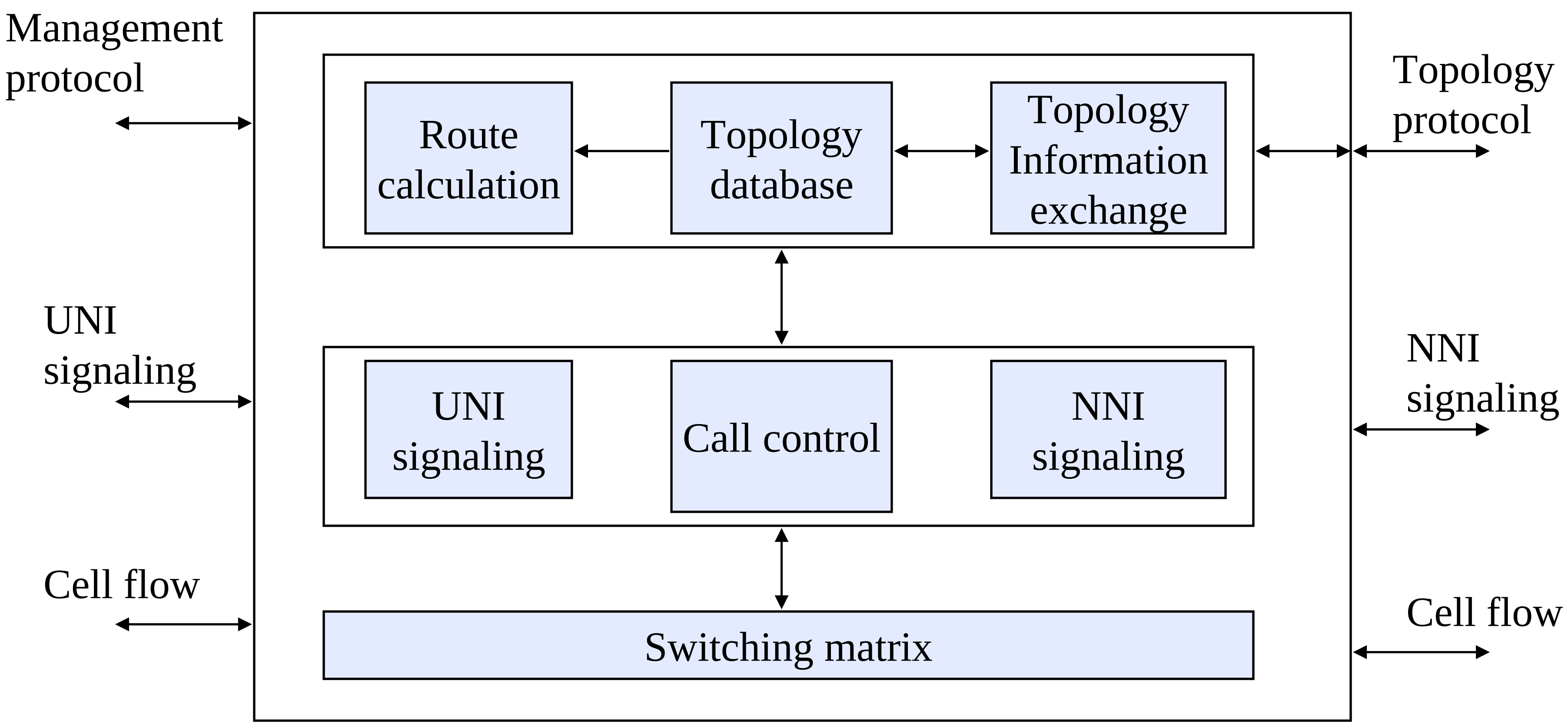
PNNI = Private Network-to-Network Interface

PNNI 1.0 specification is af-pnni-0055.000, dated March 1996, over 365 pages

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

- PNNI includes both a routing and a signaling 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 signaling is inherited from the ATM-Forum UNI signaling. Additions are source routing and crankback.

The reference model of a PNNI node



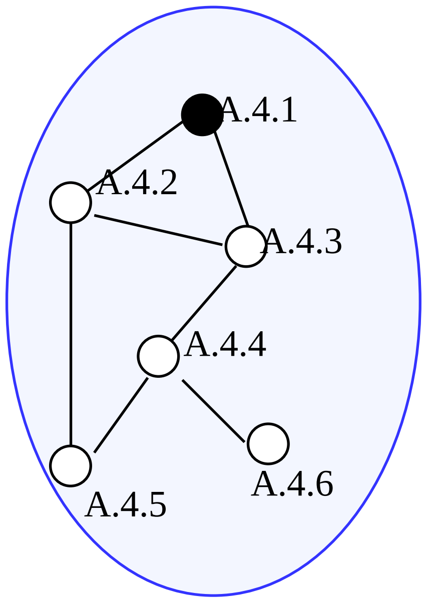
PNNI routing functions include

- 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* (*PTSE*s) 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)

PNNI topology concepts and topology protocols

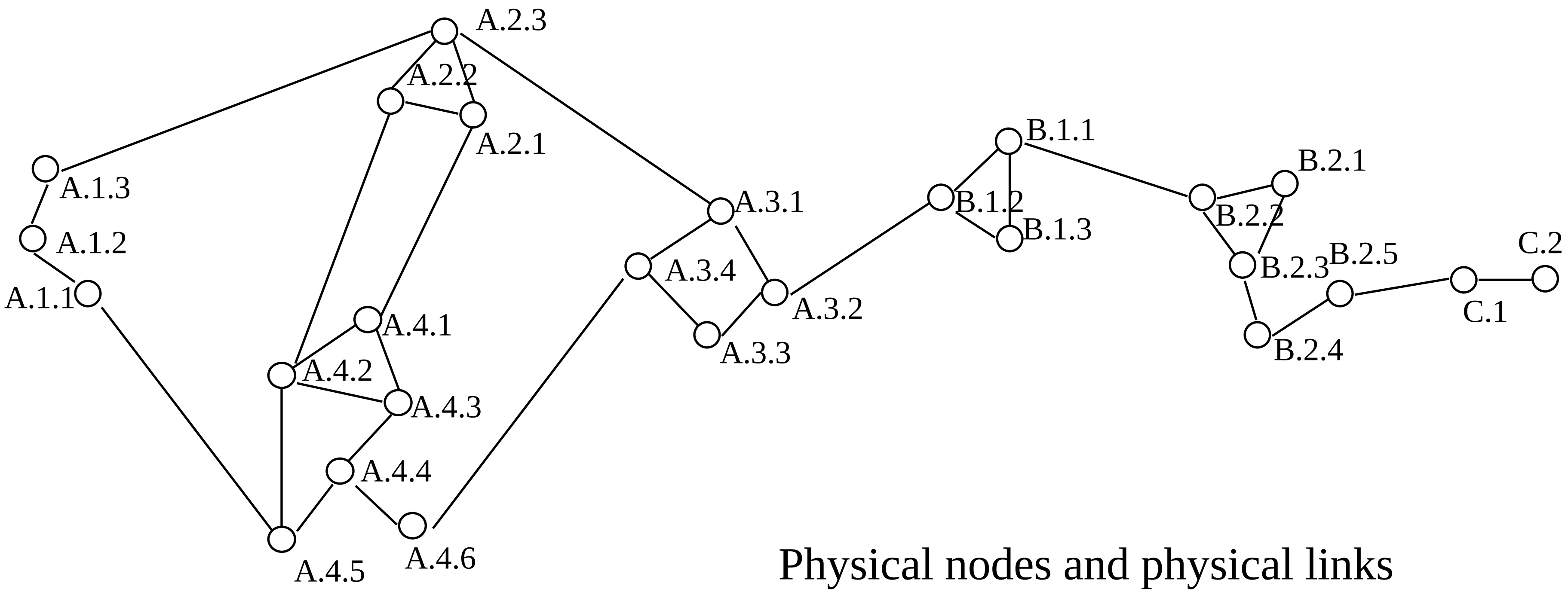
Peer group is the key concept in PNNI routing

- A *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. 20 .... 50).



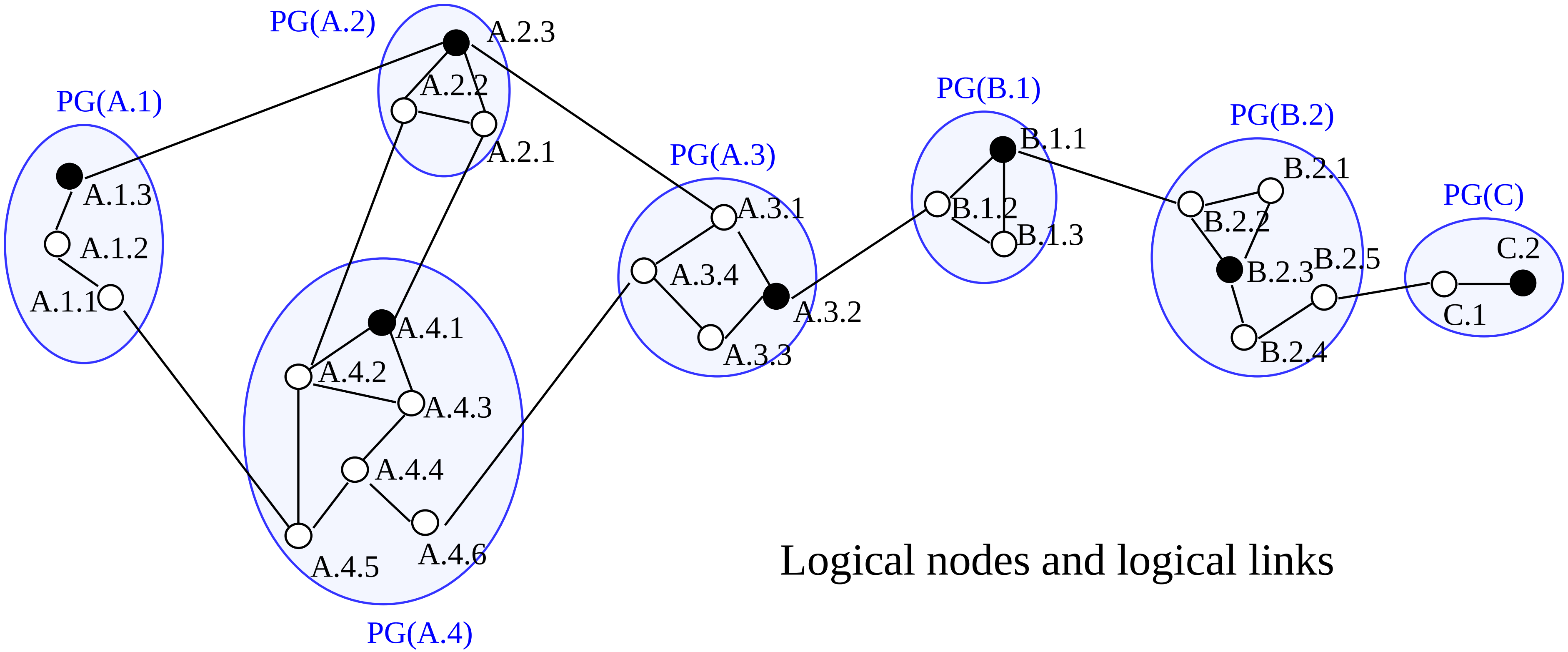
PG(A.4)

Example topology (1)



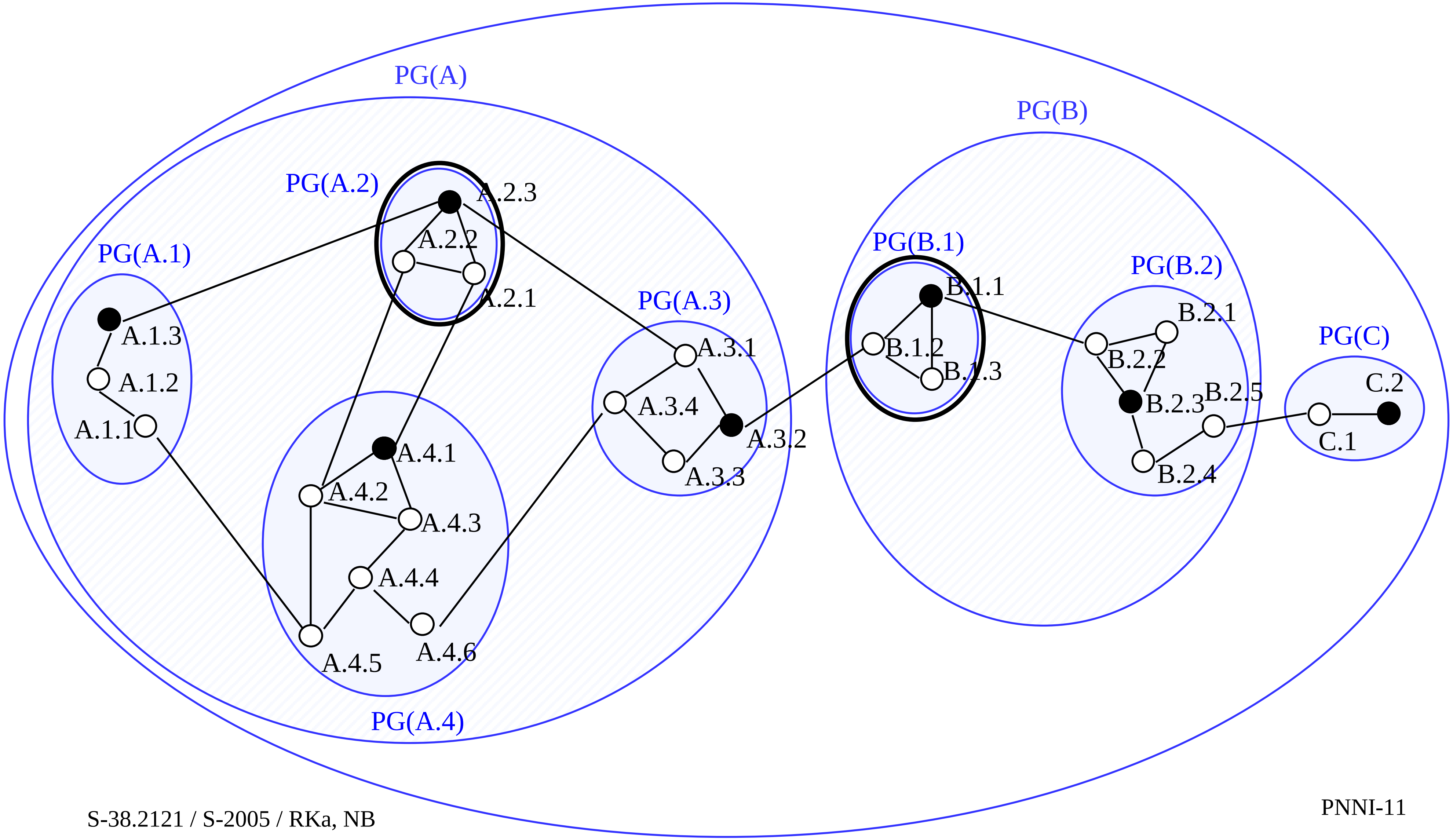
Physical nodes and physical links

Example topology (2)

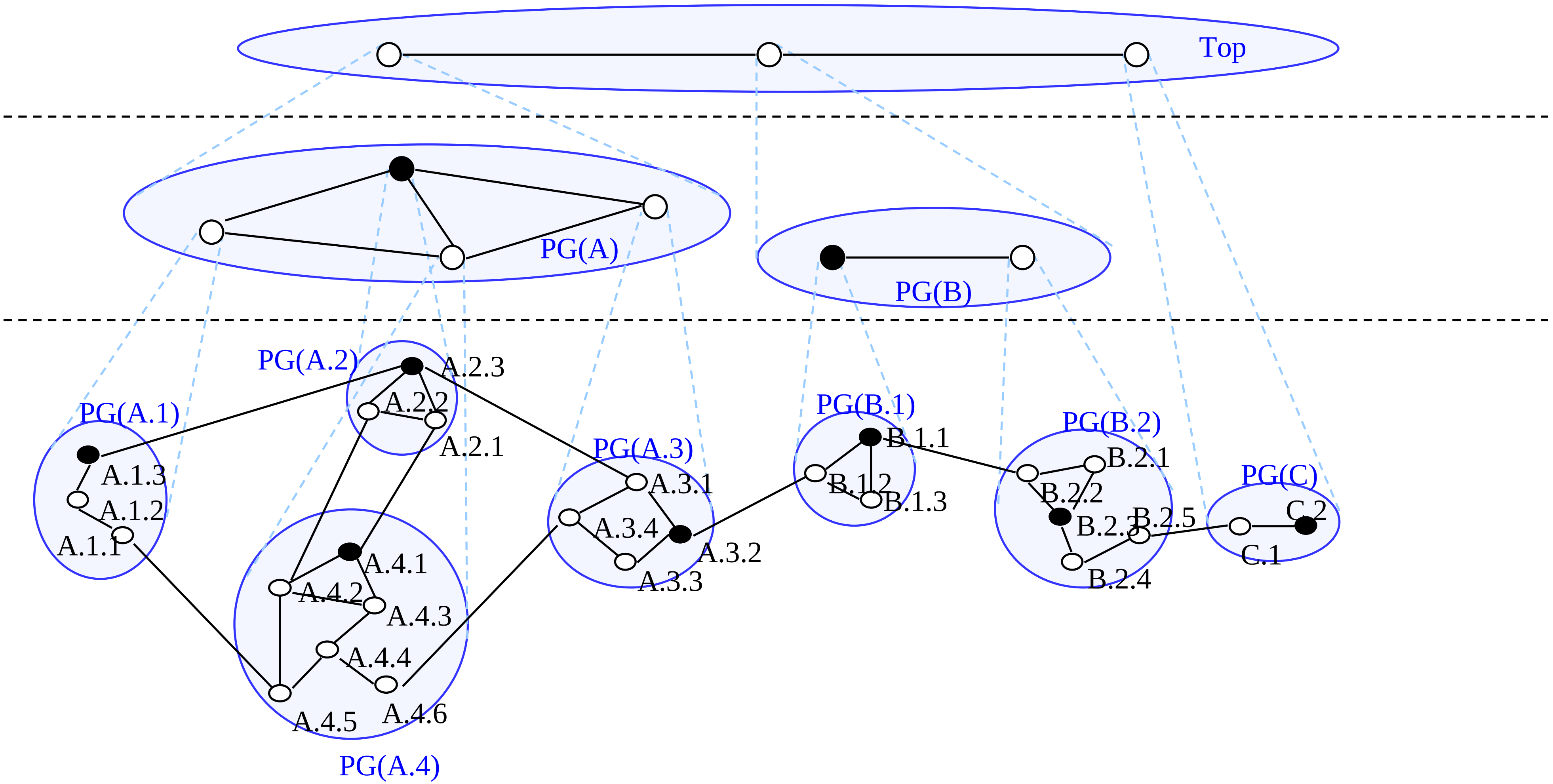


Logical nodes and logical links

Example topology (3)

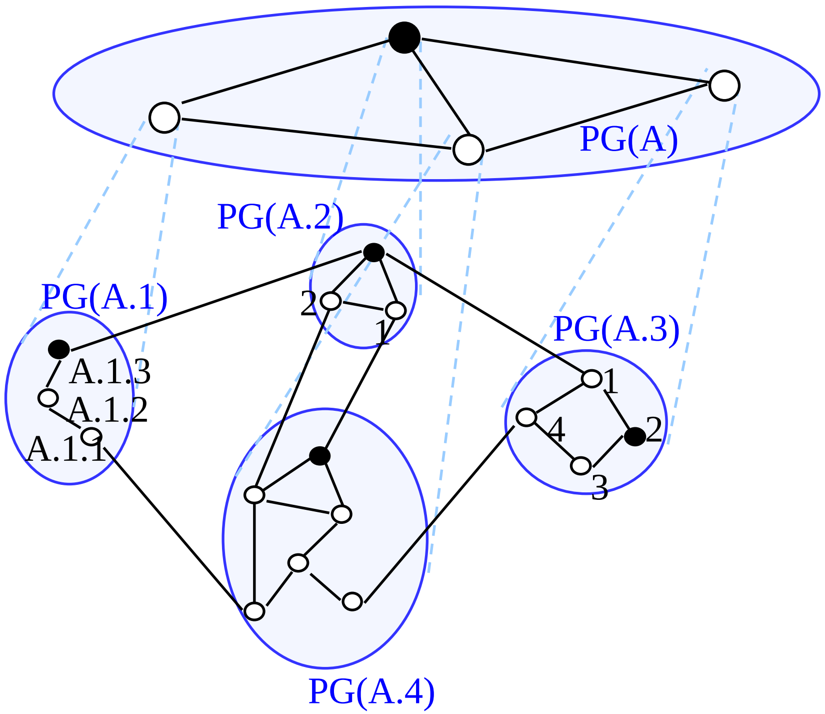


Example hierarchical topology



Peer groups form a hierarchy

- Address resolution decreases higher in the hierarchy, i.e. the prefix becomes shorter. The length of the prefix tells the level in the hierarchy. The numbering of levels starts from the top.
- The *peer group leader* (*PGL*) 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.



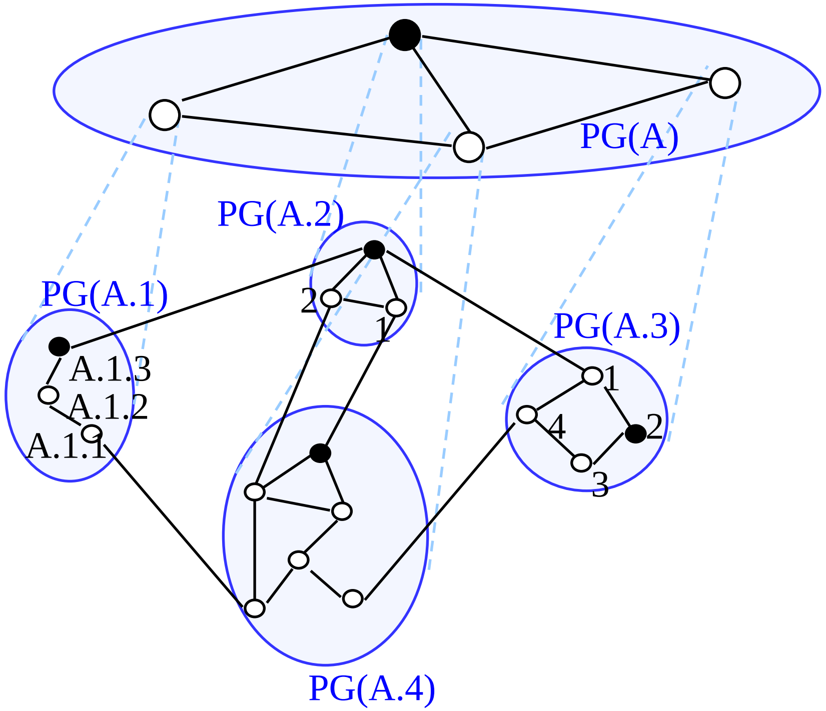
The topology consists of logical nodes and logical links

On upper levels:

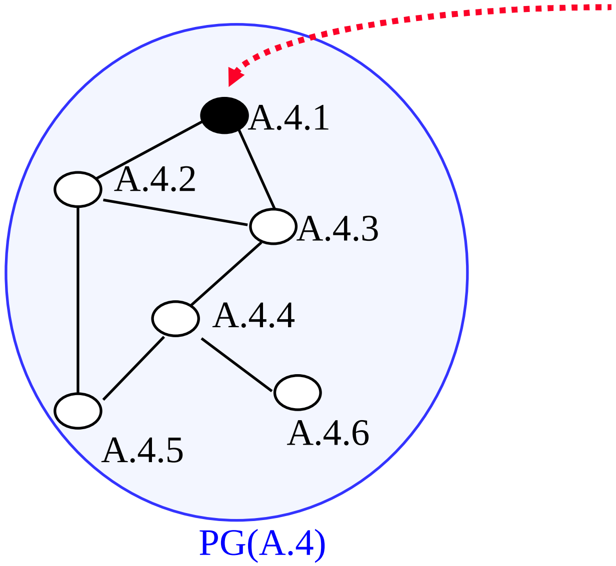
- A logical node represents the child peer group.
  - In practice the functions of the logical node are taken care of by the PGL of the child group.
- Logical link = direct link connecting child peer groups

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

Election of PGL

- Election of the PGL is based on collected topology information.
- 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
- Not all nodes need to be eligible.
- PGL can be re-elected automatically without operator interference.

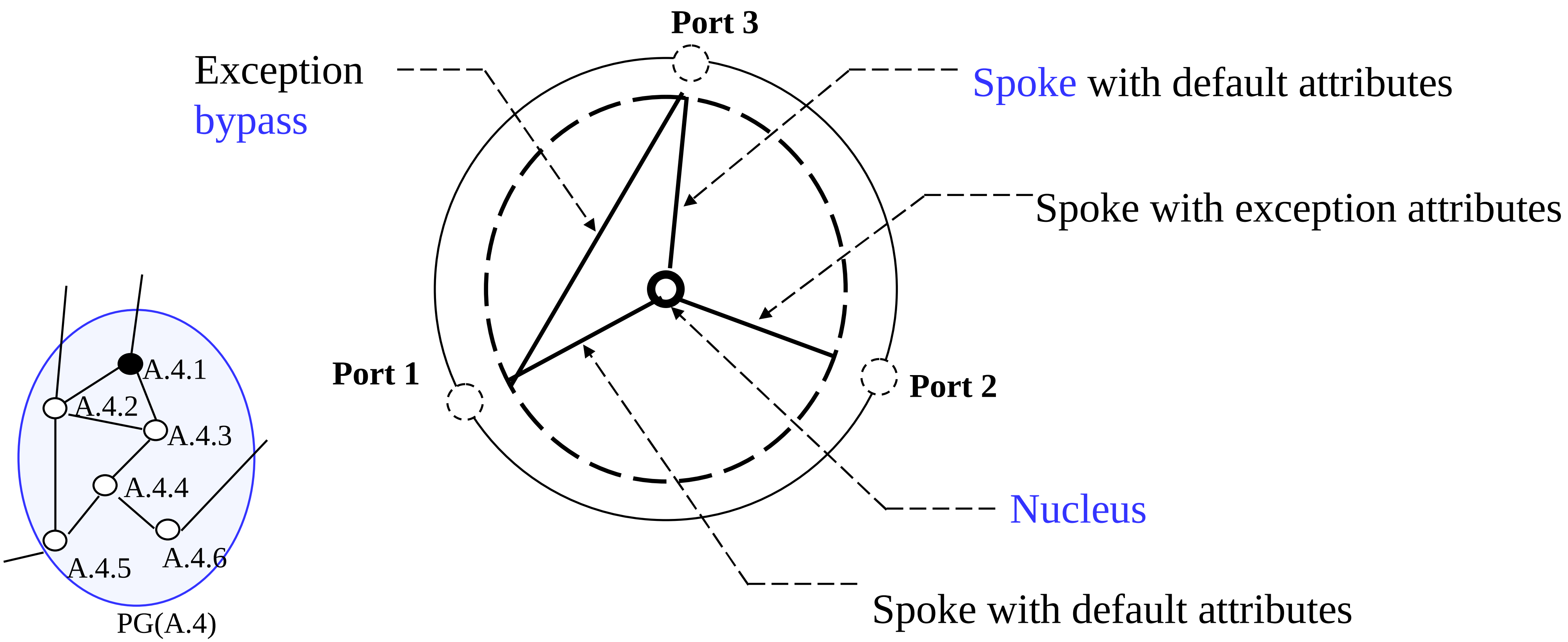
PNNI Topology State Elements describe the topology

PNNI Topology State Elements (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*

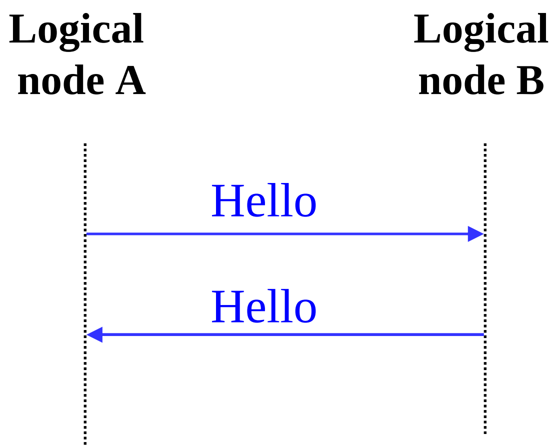
The peer group topology is aggregated by abstracting its real structure into a logical node

Complex Node Representation (CNR) of logical node A.4:

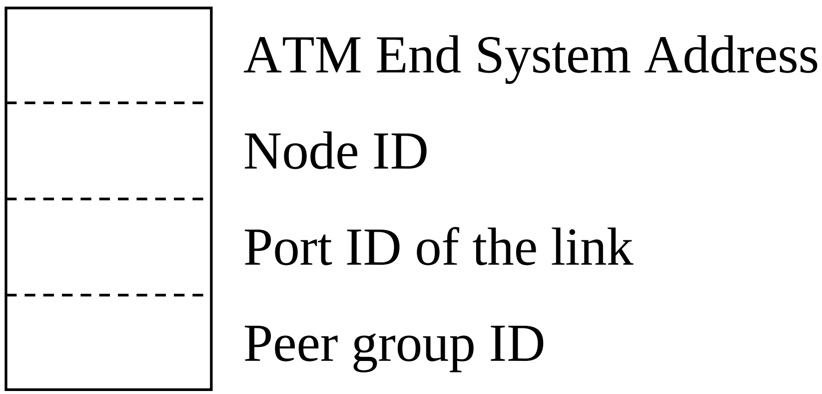


Hello protocol works on a well defined VCC between neighbors

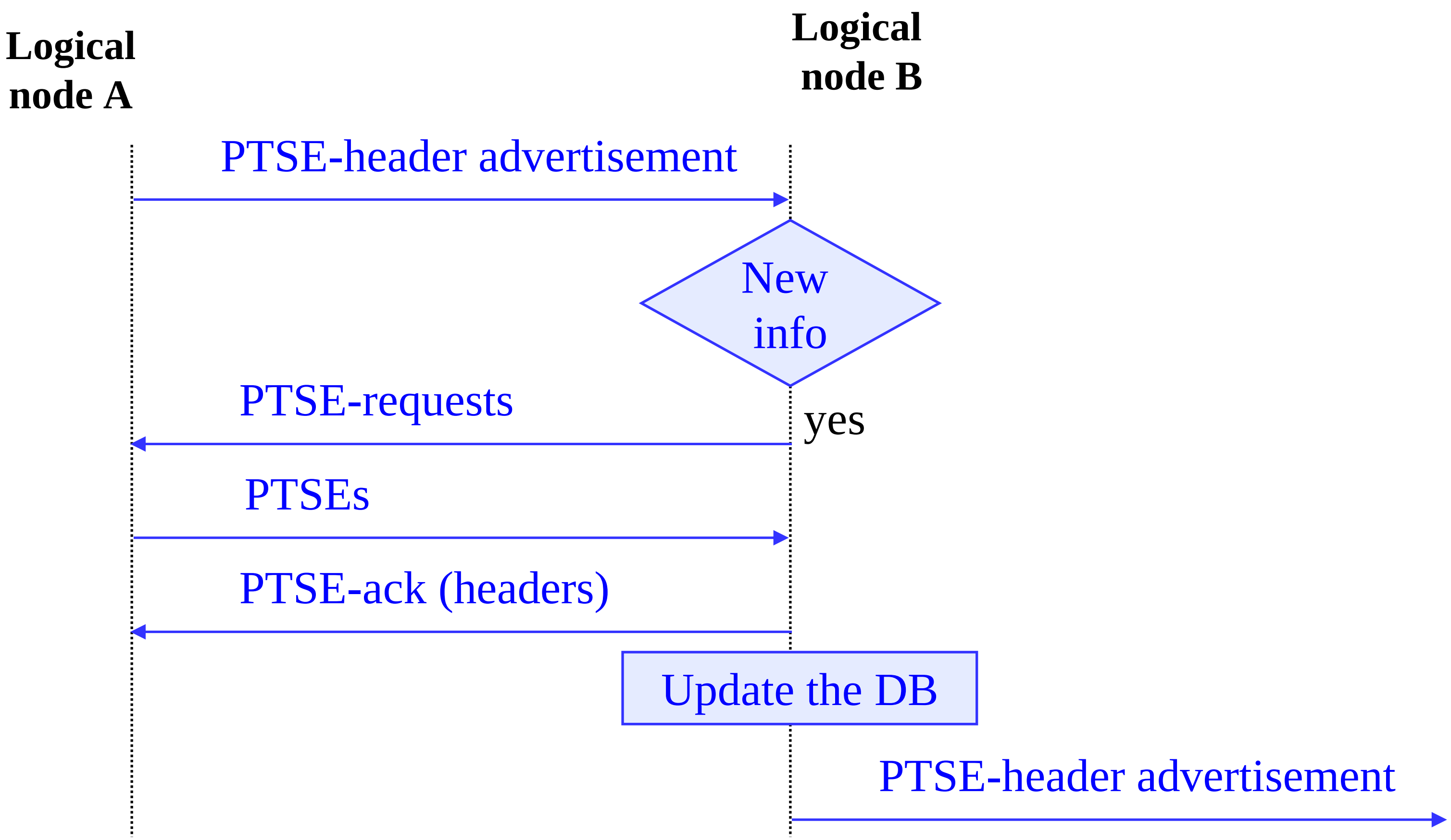
- The *Hello protocol* works continuously and reveals link failures.
- Hello protocol data is used to form the initial version of the topology database.



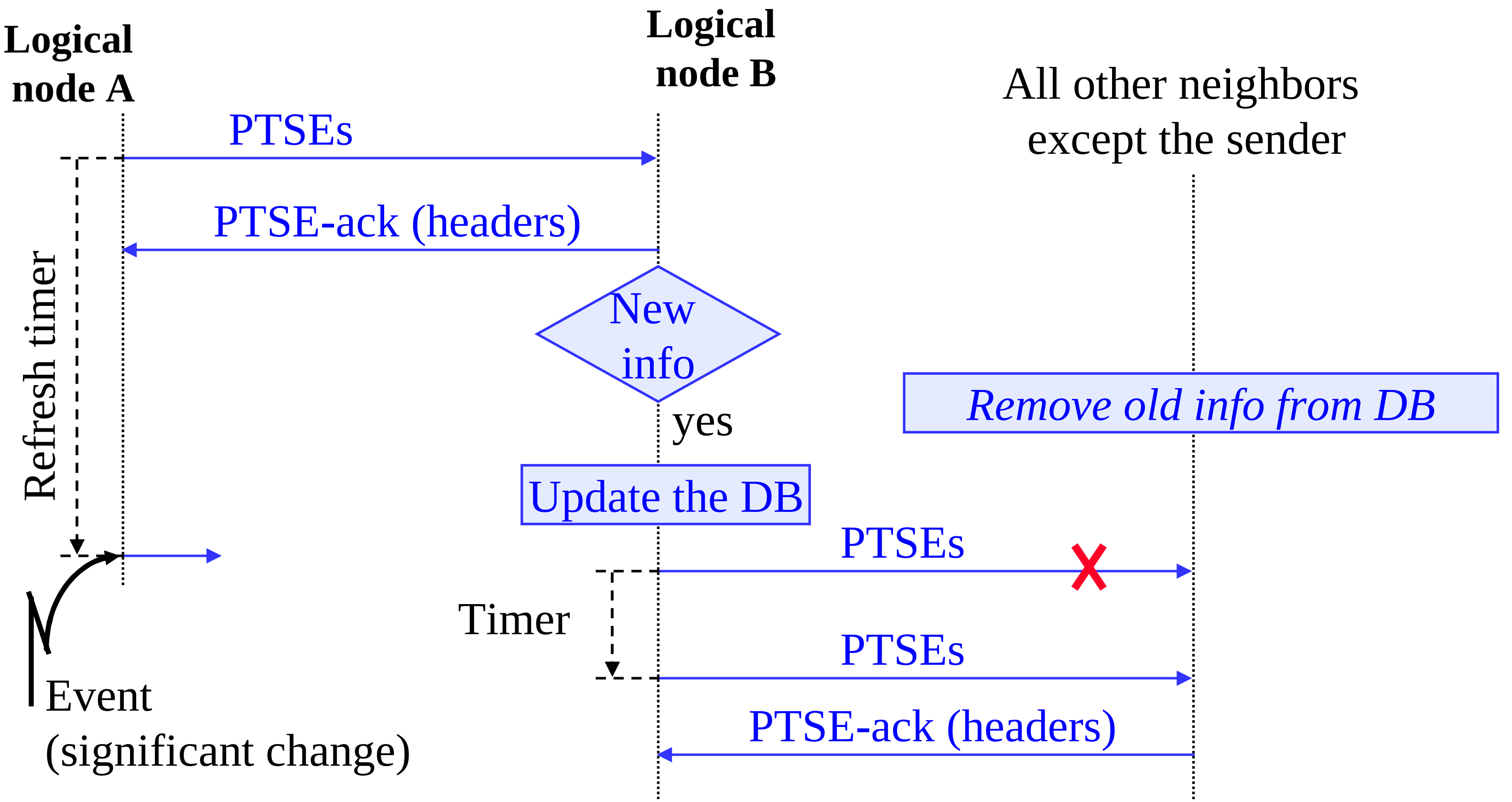
The Hello packet contain



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



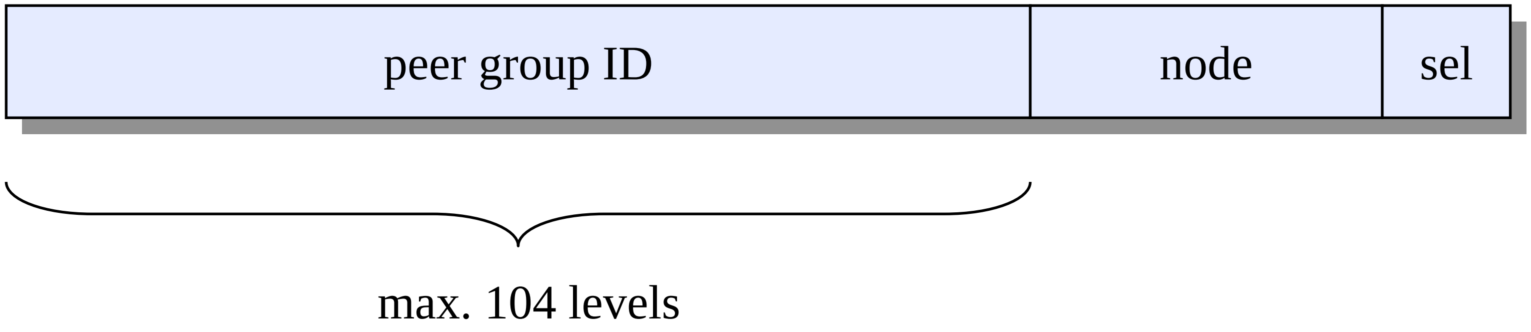
PNNI flooding protocol is similar to OSPF-flooding



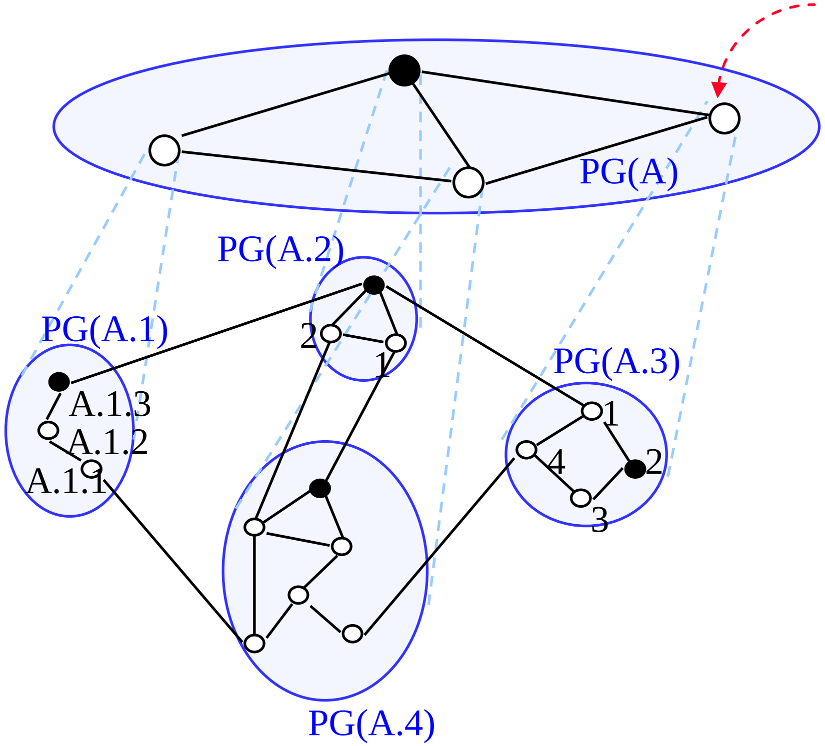
- *Send frequency of PTSEs is a compromise between probability of misrouting and the need to minimize the amount of PTSE-information.*
- *What a significant change is, is configurable.*

ATM Addresses

- 19 octet address + 1 octet selector
- Peer group ID at most 13 octets
  - 8 * 13 = 104 levels
- 10 levels should be enough in international networks



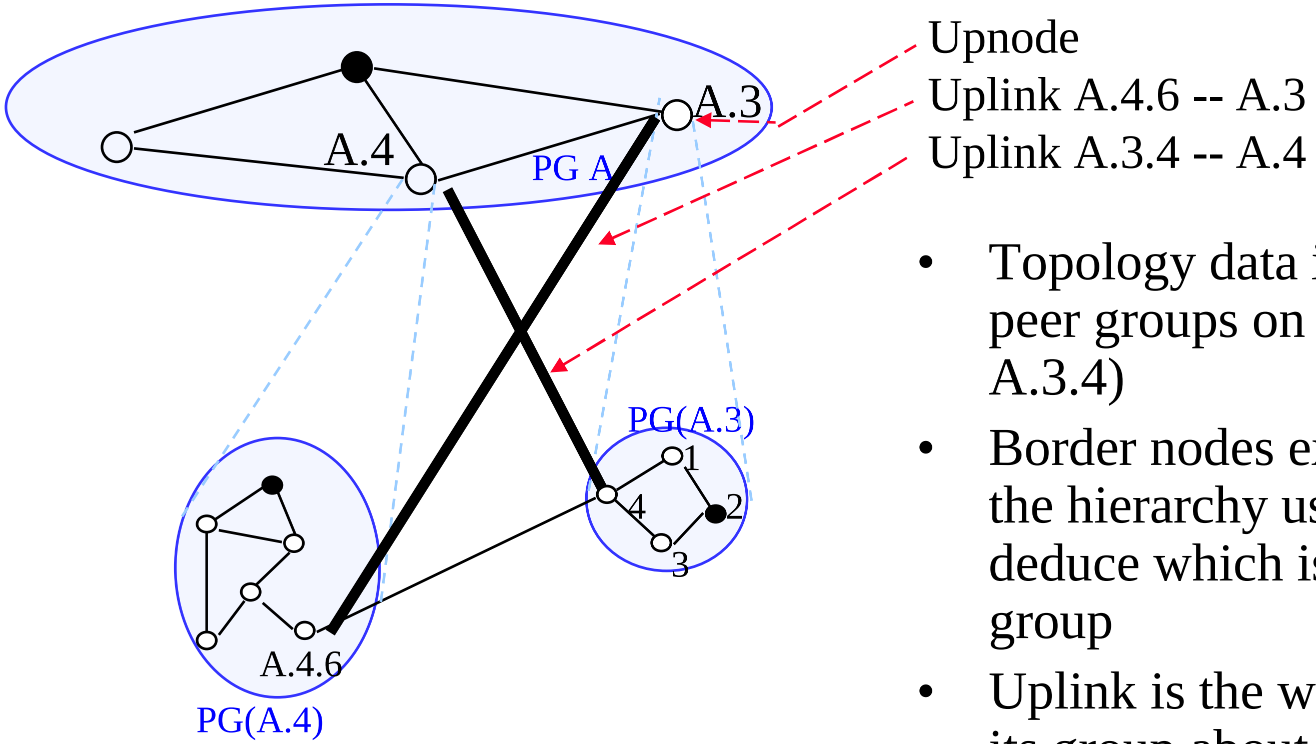
Parent peer groups are similar to lowest level peer groups



Logical group node (logical node) has

- ATM End System Address (a different SEL than PGL)
- Virtual Channel Connections (VCCs) are set up between logical group nodes for communication between them
- 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

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)
  - The 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 for CLP=0 cells (CLR_0)
- Cell loss probability for CLP=0+1 cells (CLR_{0+1})
- Maximum Cell Rate (maxCR)
- Available Cell Rate (AvCR)
- Cell Rate Margin (CRM)
- Variance factor (VF)
- Restricted Branching Flag
  - No branching points for point-to-multipoint calls
- Restricted Transit Flag
  - No transit traffic

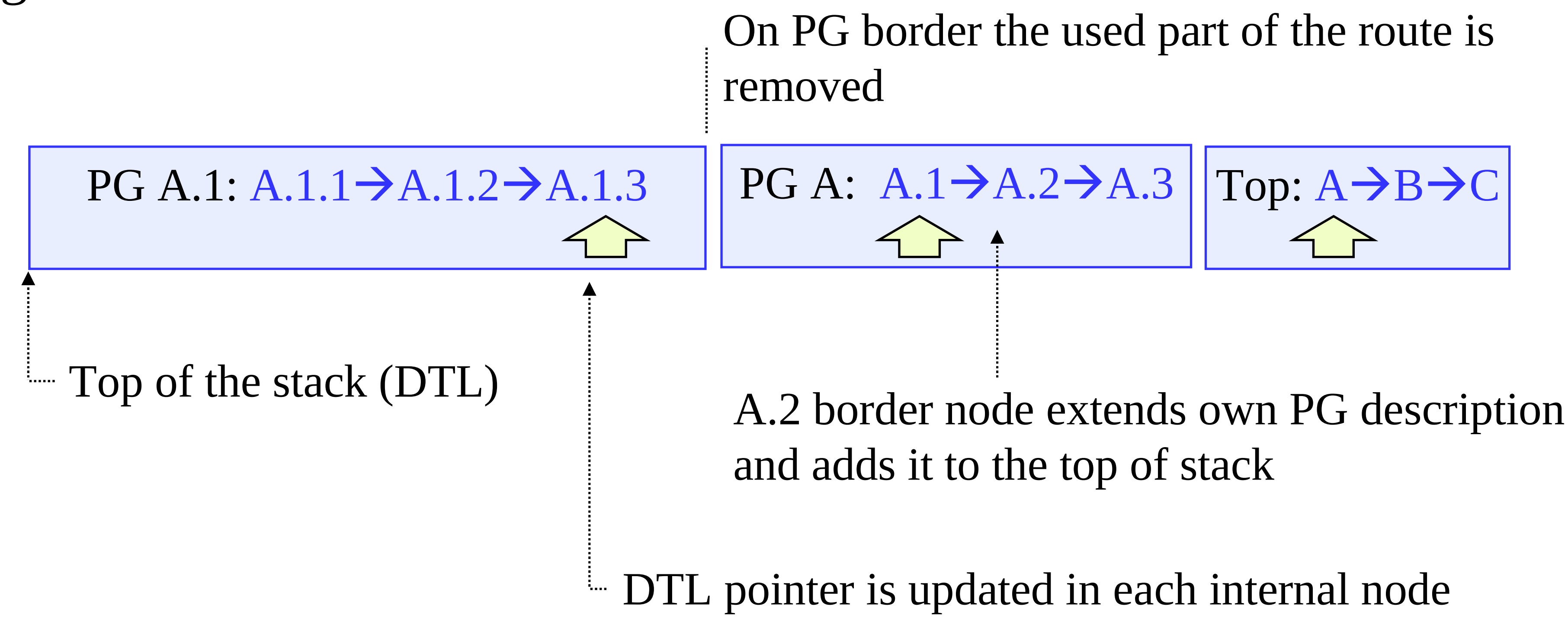
(The first six parameters above:) Resource Availability Information Group (RAIG) information

⇨ PNNI uses dynamic resource availability info! Is a dynamic routing system!

PNNI signaling
and
routing algorithm

The *Designated Transit List* (DTL) is a stack representation of the route

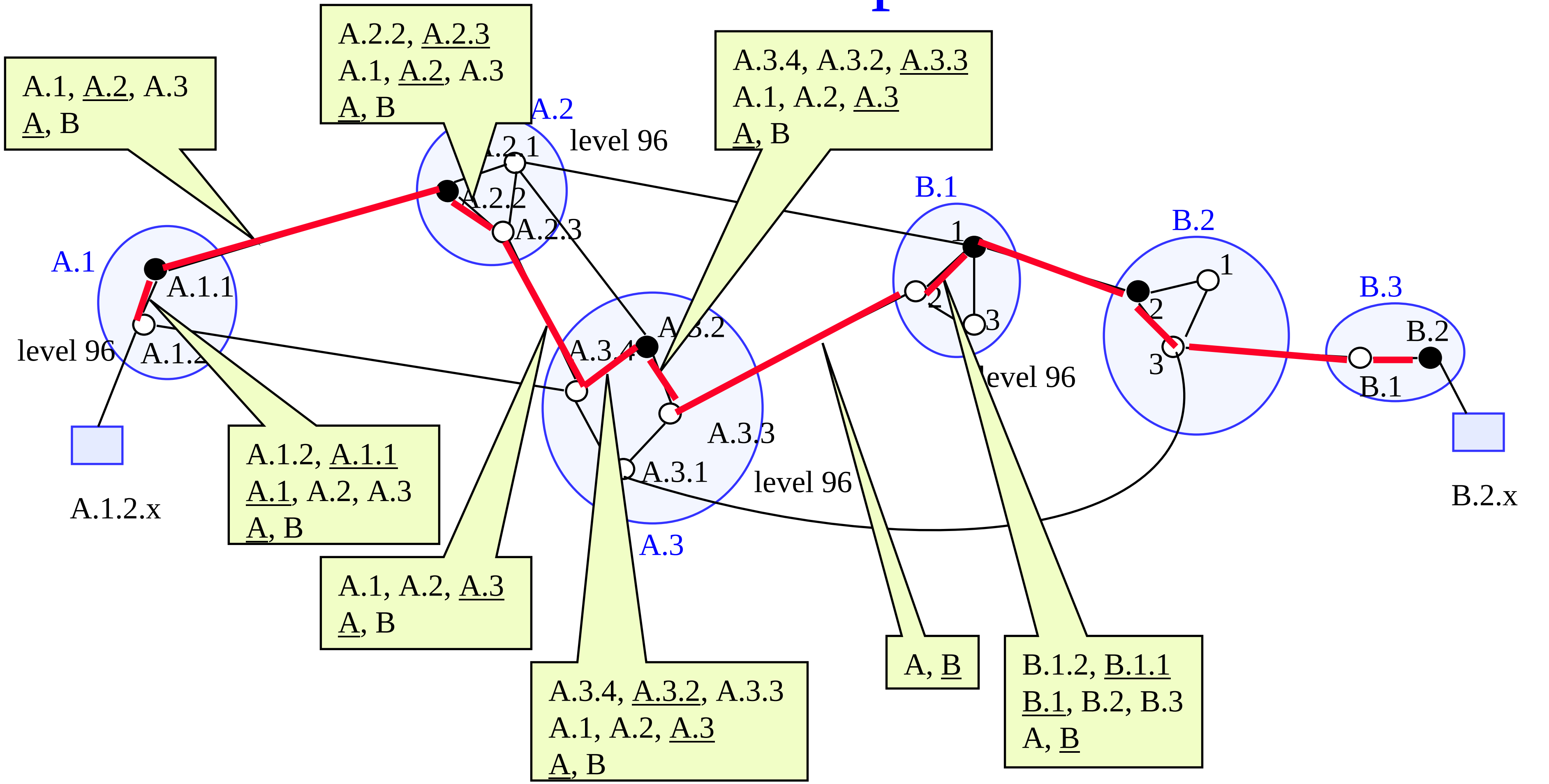
e.g. from A.1.1 to C.2:



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
   1. Source node determines lowest common peer group and forms the DTL. It initiates connection setup using info at the top of DTL.
   2. Internal nodes update the DTL pointer.
   3. At PG border lowest level peer group route info has been used and is removed. Connection setup request is sent over the PG border.
   4. 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 through lowest level PG towards a node with a suitable external link and inserts the partial route at the top of the DTL.
   5. Continue at 2.2.

Example

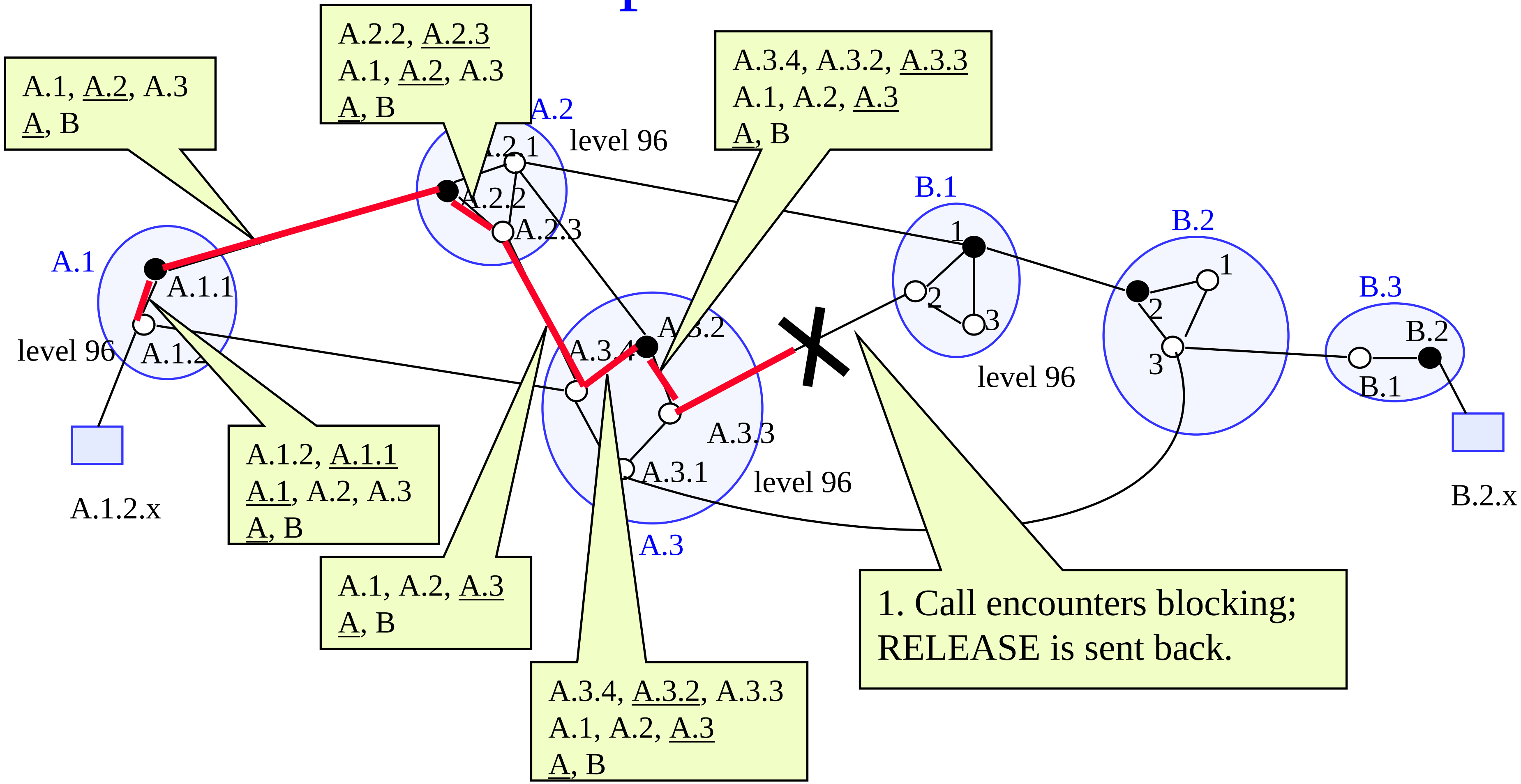


A.1, A.2, A.3
A, B
A.2.2, A.2.3
A.1, A.2, A.3
A, B
A.3.4, A.3.2, A.3.3
A.1, A.2, A.3
A, B
A.2
level 96
A.2.1
A.2.2
A.2.3
B.1
B.2
B.3
A.1
A.1.1
A.1.2
level 96
A.3.4
A.3.2
A.3.3
A.3.1
level 96
level 96
A.3
B.2
B.1
A.1.2.x
B.2.x
A.1.2, A.1.1
A.1, A.2, A.3
A, B
A.1, A.2, A.3
A, B
A.3.4, A.3.2, A.3.3
A.1, A.2, A.3
A, B
A, B
B.1.2, B.1.1
B.1, B.2, B.3
A, B

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

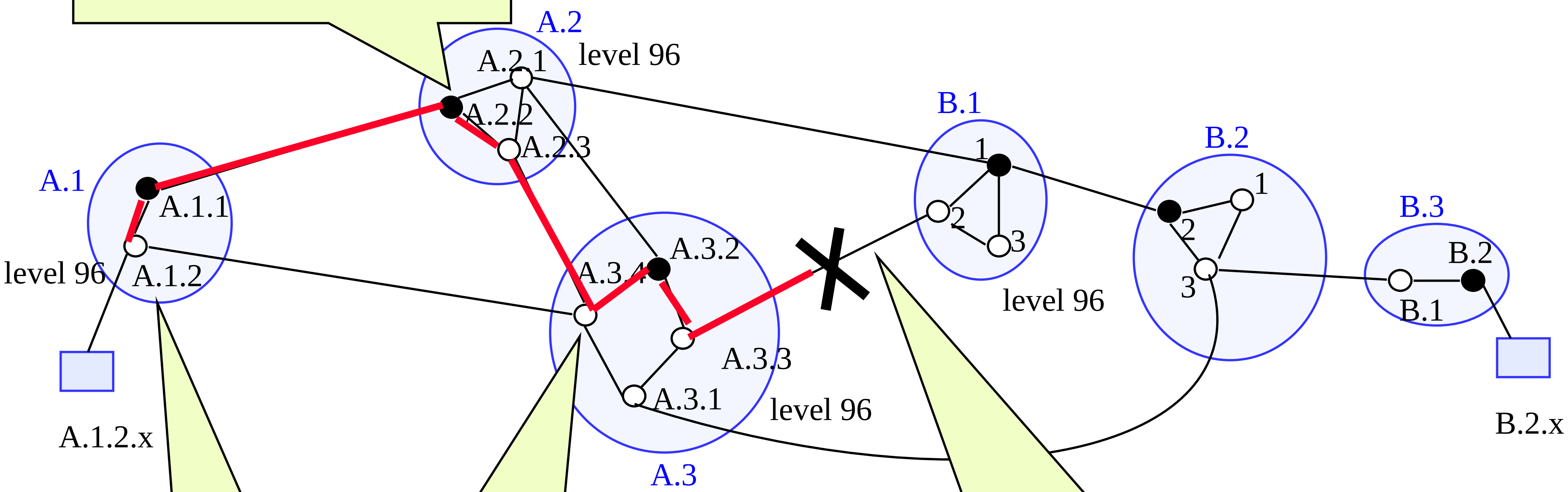
- Crankback may become necessary if newest topology information has not been advertised to the node that calculated a portion of the route.
- Because of crankback any node on the path may need to make a routing decision.
- Crankback returns the call in the order determined by DTL.
- Normally crankback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

Example with crankback



Example with crankback

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



4. Source node calculates a new route: A.1 → A.2 → B.1

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

1. Call encounters blocking; RELEASE is sent back.

Summary of PNNI routing principles

- Route calculation is done *peer group by peer group*
- The selected route is described in the designated transit list (DTL), the 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* as long that a suitable border node or the source node itself can select a new route.
- PNNI always seeks to satisfy the *QoS parameters* accepted by the source node.

Why is PNNI based on source routing?

- Algorithm can be different in different systems
  - Inconsistency in routing decisions when switches use different routing algorithms
- Circuit switching → loops and inefficient routes serious
  - Inconsistency in routing databases among the switches (typically due to changes in topology information that have not fully propagated yet)
- Replicates the cost of the path selection at each system
  - QoS calculations may be heavy