Jean Paul Accarie
Email: accarie@crf.canon.fr
Laurent Frouin
Email: frouin@crf.canon.fr

Simple Routing Method proposal

SRM Basic Principles

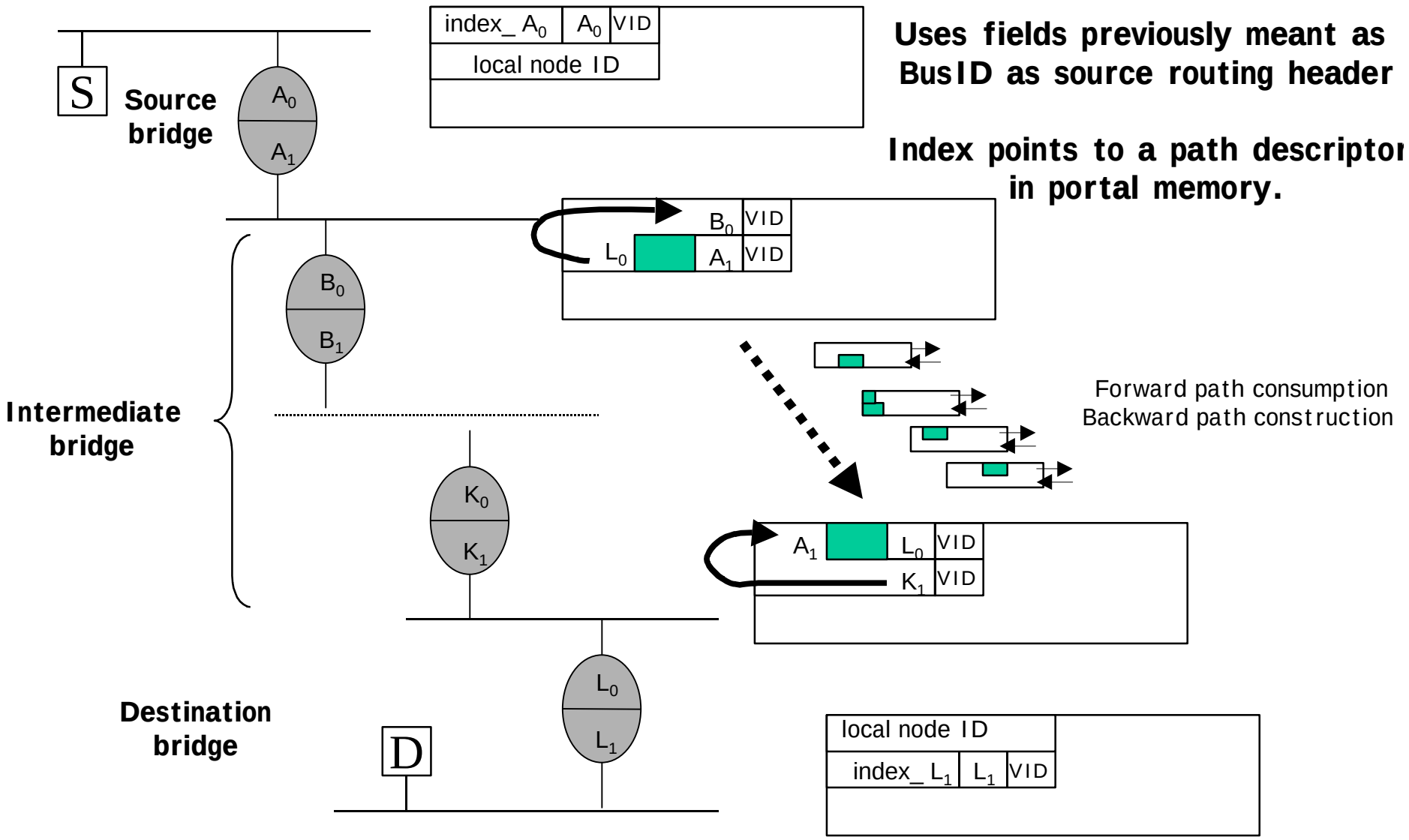
Adding / Removing portal

Path Descriptor Table management

P1394.1 requirements

Conclusions

Basic Principles - Overview -



Basic Principles - Main steps -

Portal numbering

- based on alpha-portal mechanism
- routing label mapping is added for portal only
- persistent across bus reset

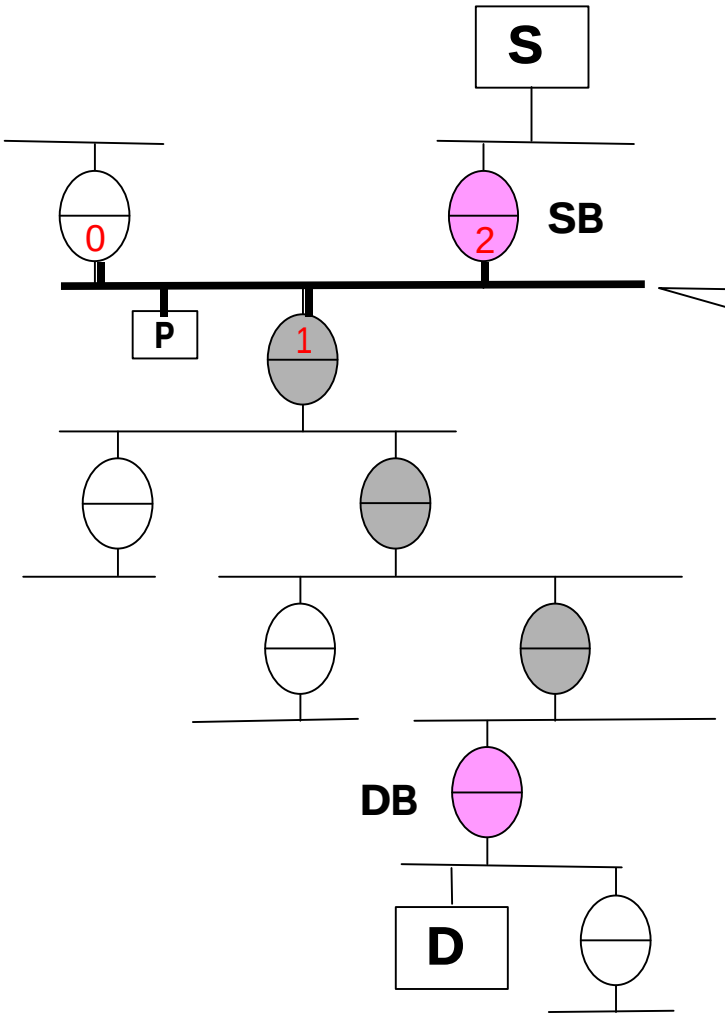
Path descriptor discovery

- uses GASP packets (default broadcast channel)
- bridge (that knows about targeted peripheral) responds with a block write to “source bridge” (then to requester)

Asynchronous transactions

- 1394-1995 compliant on source & destination bus
- SRMized between source & destination bridges

Basic Principles - Portal numbering -

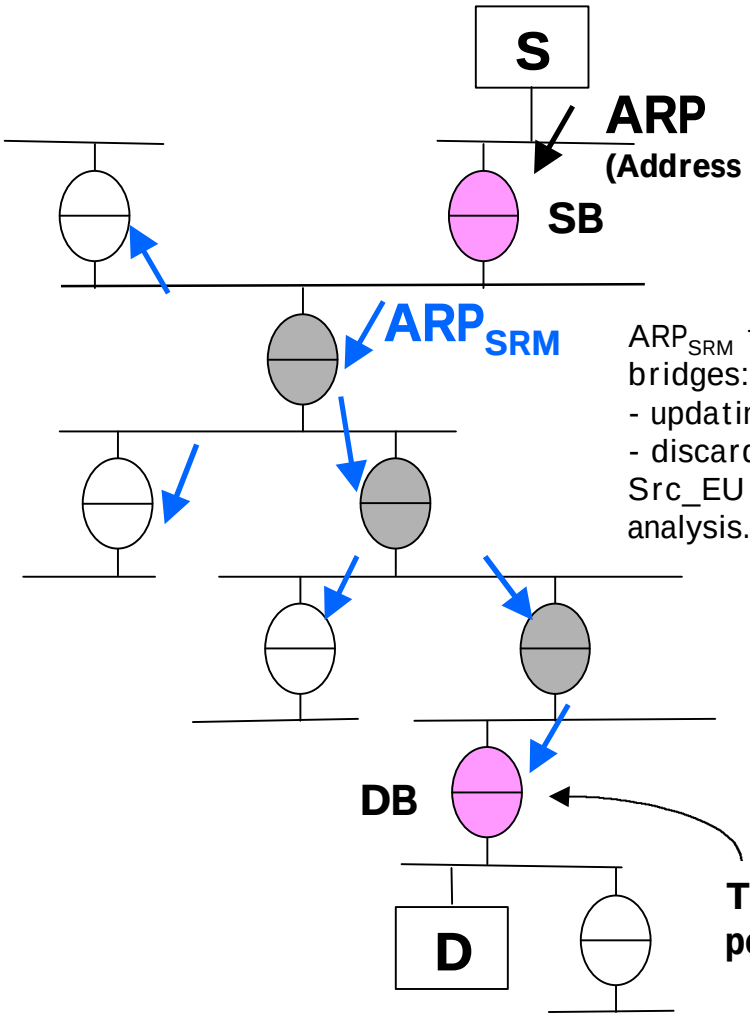


On each bus, a Routing Label is assigned to each bridge portal.
The value 0 is assigned to the Alpha portal.

Physical ID	Virtual ID	EUI-64	Routing label
4_p	0	@α	0
7_p	4	@β	2
2	1	@φ	-
0_p	3	@ε	1
1	2	@μ	-
.....			

S: Source peripheral
D: Destination peripheral
SB: Source bridge
DB: Destination bridge
P: Any other peripheral

Basic Principles - Path Discovery - 1



ARP$_{SRM}$ flooding, performed by bridges:
- updating path_descriptor field,
- discarding loop packet based on Src_EUI and sequence_nb fields analysis.

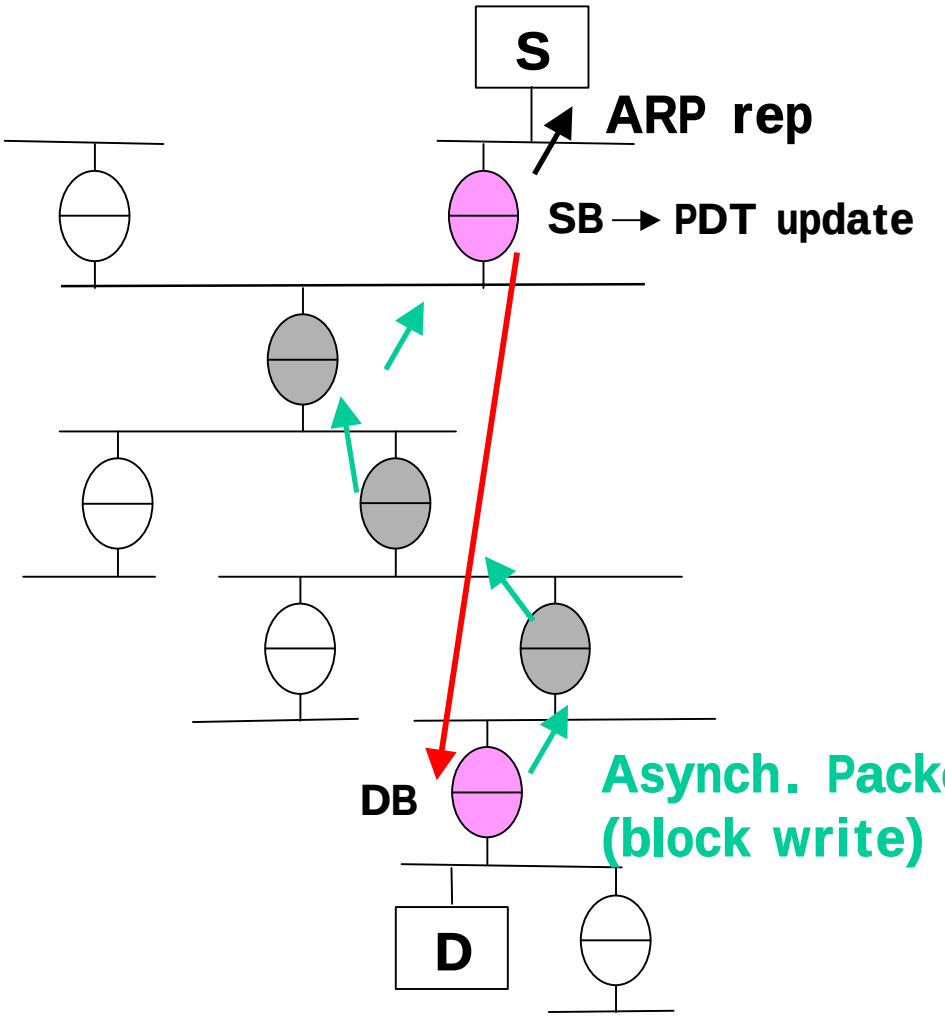
ARP$_{SRM}$ is based on GASP packet:

data_length	tag	channel	A_{16}	sy
header_CRC				
source_ID	specifier_ID_hi			
Specifier_ID_lo	version			
path_descriptor	sequence_number			
Src_EUI_64_hi				
Src_EUI_64_lo				
Dev_EUI_64_hi				
Dev_EUI_64_lo				
response packet type specific information (destination_offset)	reserved			
data_CRC				

32

The destination bridge, that knows about the targeted peripheral, sends a response (and stops ARP$_{SRM}$ flooding)

Basic Principles - Path Discovery - 2

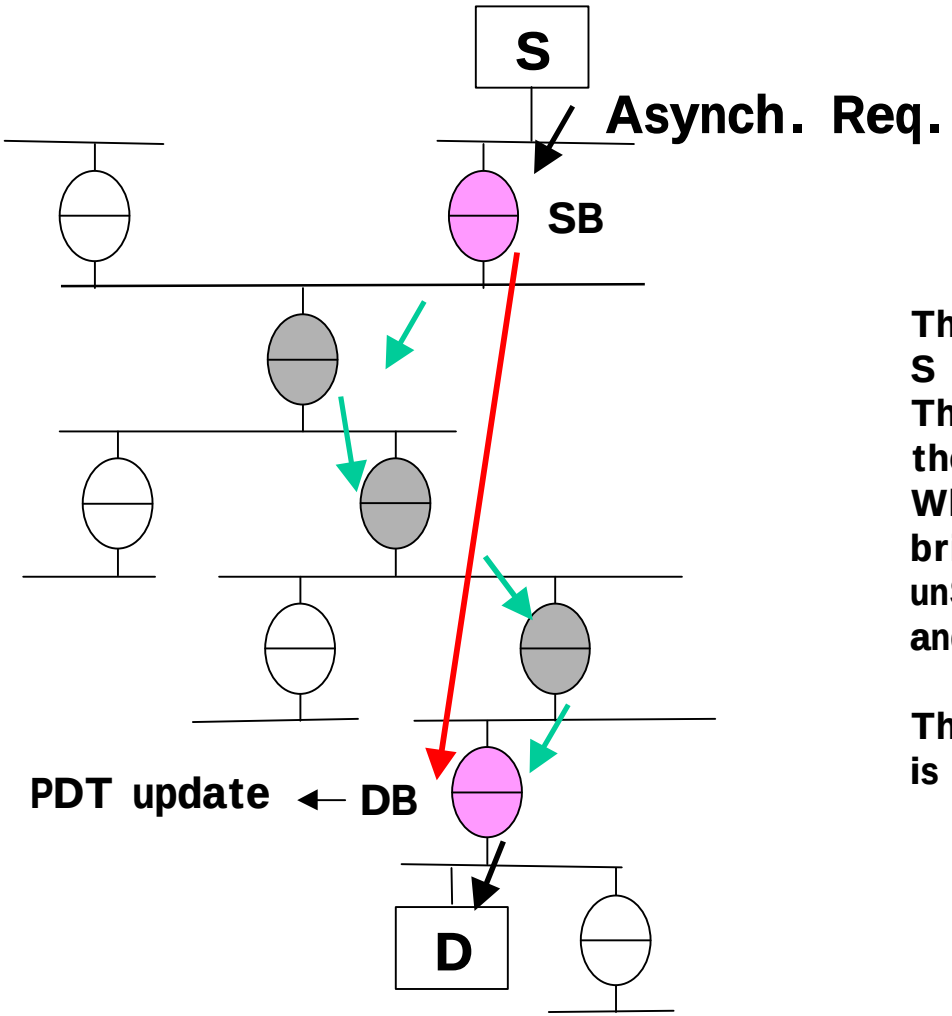


The asynchronous packet is routed from DB up to the source bridge using SRM principle. The source bridge (SB) updates its PDT, then builds the response to the ARP request and send it to the source peripheral (S).

S is ready to communicate with D.

Write request
Write response

Basic Principles - Asynchronous transactions -



The asynchronous request packet issued from S is catched by SB.
The packet is SRMized and is routed till the destination bridge using SRM principle.
When receiving the packet the destination bridge (DB) updates its PDT,
unSRMizes the packet
and sends it to the source peripheral (D).

The response packet issued from D is similarly routed back to S.

Transaction request
Transaction response

SRM Basic Principles

Adding / Removing portal

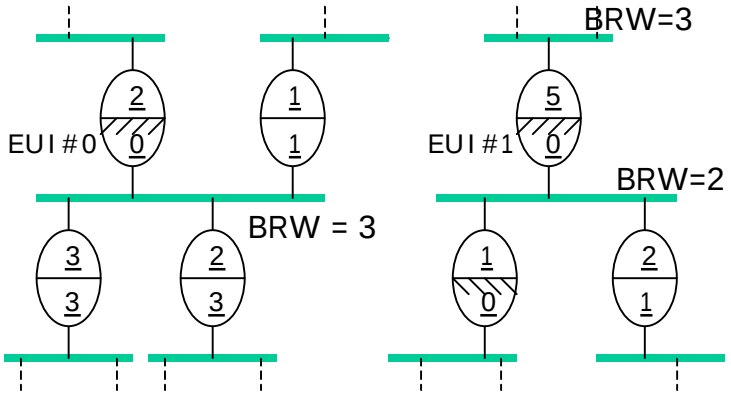
Path Descriptor Table management

P1394.1 requirements

Conclusions

Adding portal - Detect new portal -

Independent buses

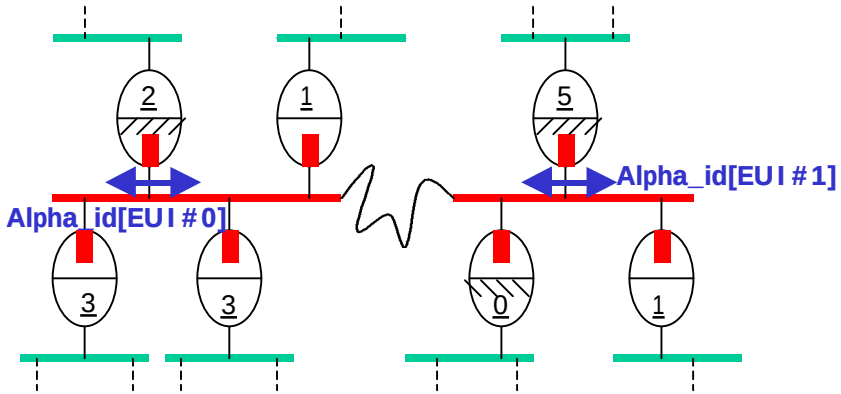


2 independent buses are represented:

- one includes a bus with EUI#0 alpha-portal
- other includes a bus with EUI#1 alpha-portal
- each bus is provided with a Bus Routing Width (BRW) that defines routing label range BRW = 2 RL = 0 -> 2
  BRW = 3 RL = 0 -> 6

Those 2 buses are properly functioning, whatever if they belong to the same network or not.

Joining buses



A wire is added somewhere, then modifying the physical topology From alpha portal viewpoint, it is like adding at least one portal on their own bus:

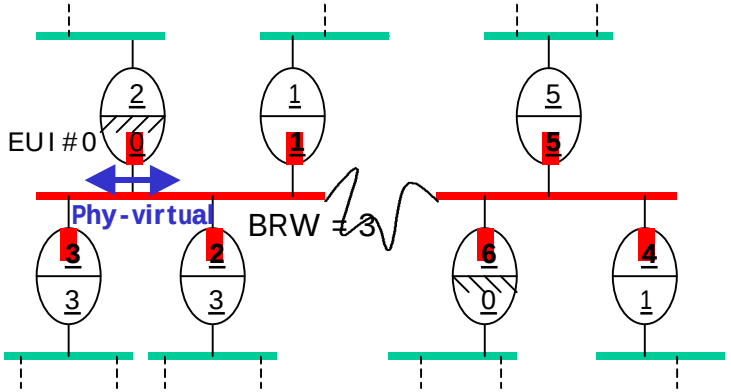
Every time a new portal is detected on the bus, alpha sends an alpha_id command.

The alpha_id command could be a local GASP (broadcast write?) that includes:
- alpha_EUI,
- alpha_phy_id,
- number of portals previously known,
- number of devices previously known.

Asynchronous traffic going-through joined buses is stopped.

Adding portal - Select surviving part -

Surviving alpha selection

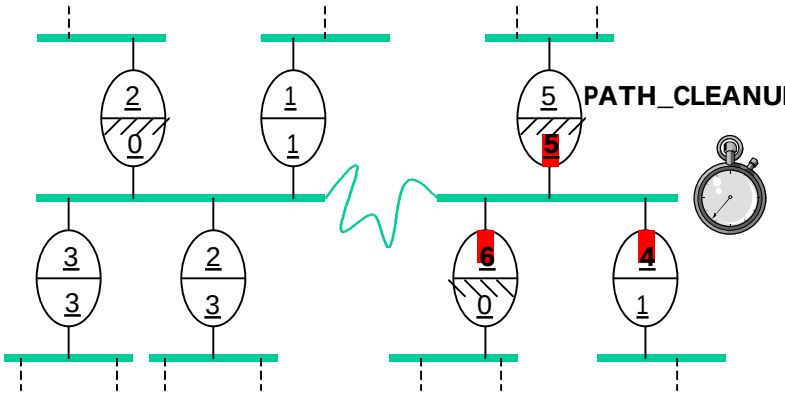


Every portal is selecting the same alpha. The winner is:
- the bigger BRW, and if equals
- with the larger number of portal, and if equals
- with larger number of devices, and if equals
- with the bigger EUI-64.

Then new physical <-> virtual ID mapping is forced by the alpha through local GASP command (to minimise virtual ID re-assignment). Then new Routing Label can be assigned according to new virtual ID, based on new BRW needs.

Asynchronous traffic aimed to re-assigned virtual ID is discarded. Asynchronous traffic going-through victim bus is also discarded.

Resuming "surviving" traffic

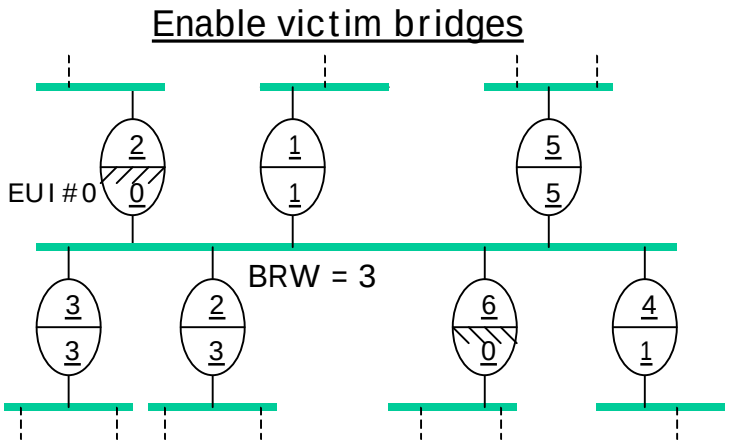


Traffic going-through surviving bus is resumed if the BRW remains the same. Else traffic going through the surviving bus is also discarded.

During the pre-defined time period PATH_CLEANUP_TIME all the victim traffic is discarded. It includes:
- asynchronous traffic aimed to re-assigned virtual ID,
- asynchronous traffic going-through victim bus,
- asynchronous traffic going-through surviving bus, only if the BRW of the surviving bus has been modified.

Local traffic is allowed.

Adding portal - Resume all -



When PATH_CLEANUP_TIME period elapsed, everything should be properly functioning within the new single network.

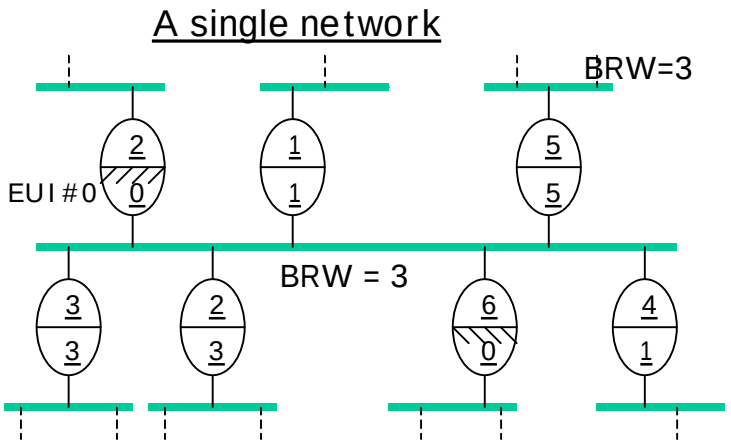
PATH_CLEANUP_TIME period duration has a main role when modifying either:
- Routing Label,
- Bus Routing Width.

It has to be defined regarding Path Descriptor Table management.

It should allow to manage Virtual ID re-assignment on a victim bus.

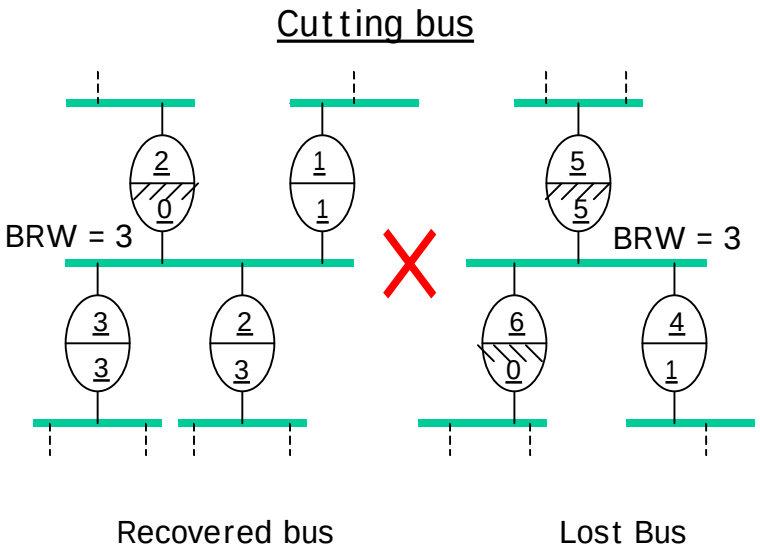
Any disrupted traffic could be resumed by initiators after new ARP request.

Removing portal - Insulating portal -



Let's assume a single network where one bus owns an alpha-portal with EUI#0.

From this stable state, a link will be removed within the bus ...



The recovered bus owns the alpha portal from previous state. It still is properly functioning regarding both local and remote traffic while it is discarding :
- remote traffic aimed to invalid routing label,
- remote traffic aimed to invalid virtual ID of the recovered bus.

The lost bus needs first to select an alpha-portal (the portal with the biggest physical ID) but still is proceeding with traffic going through this new bus while it is maintaining routing label previously assigned. Of course remote traffic aimed to invalid routing label or to any invalid virtual ID has to be discarded.

Asynchronous packet going through the bus have to be discarded if the Bus Routing Width is decreased and then wait for PATH_CLEANUP_TIME period before enabling attached portals.

SRM Basic Principles

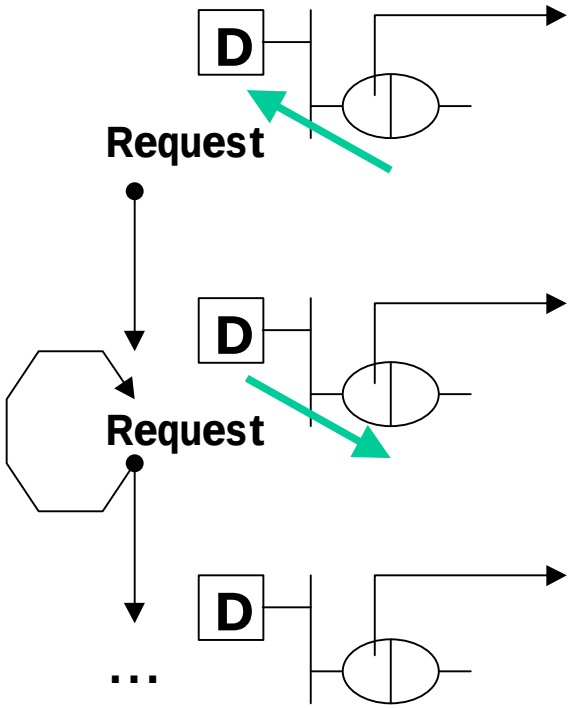
Adding / Removing portal

Path Descriptor Table management

P1394.1 requirements

Conclusions

Path Descriptor Table - Target -



PDT initialisation

Portal receive any kind of transaction requests, then acts as a destination node for transaction processing:
- get an index in the PDT,
- store the backward path built from transaction request routing,
- forward a local asynchronous packet with a source_bus_ID field that includes the index value and portal's routing label.

This new entry in the PDT then becomes available for a transaction request.

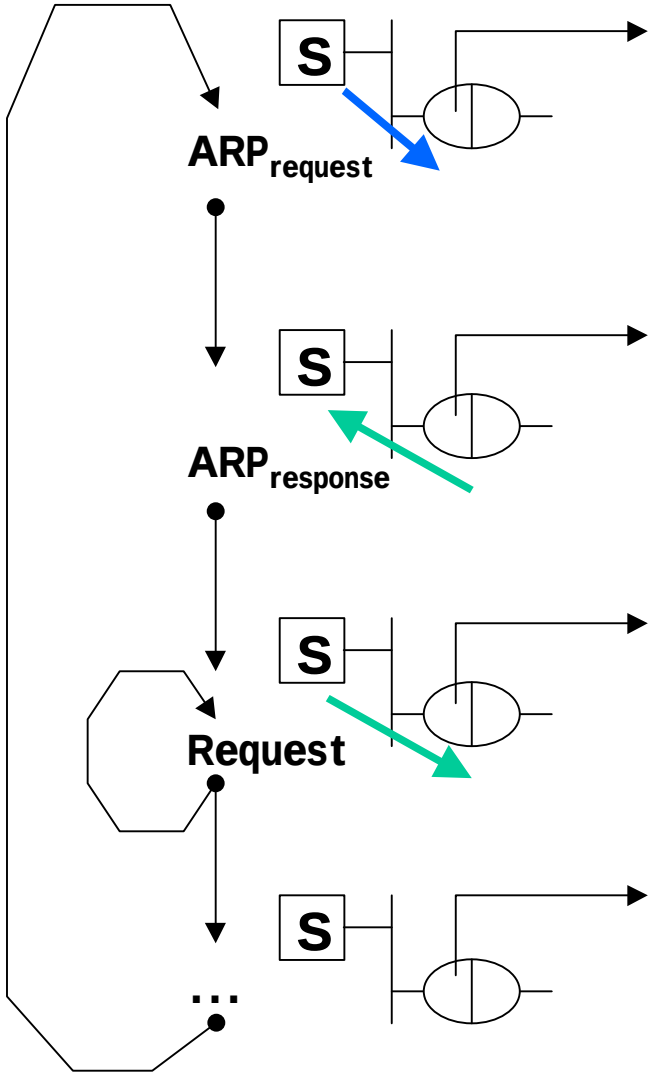
PDT maintenance (refresh)

Each time a transaction request is issued from local bus, portal refreshes the corresponding entry (if valid !) in its PDT, for instance, setting an activity bit.

PDT maintenance (deletion)

Every PATH_TIMEofLIFE period activity bit could be checked. If not set, the path may become useless and should be removed from PDT. Portal will prevent previous node next access to this path if it is a transaction request.

Path Descriptor Table - Initiator -



Portal detects ARP request issued from local bus.
It has to wait for response processing.

PDT initialisation

Portal receive ARP response (transaction request), then acts as a destination node for transaction processing:
- get an index in the PDT,
- store the backward path built from transaction request routing,
- forward a local asynchronous packet with a source_bus_ID field that includes the index value and portal's routing label.

Portal may have to deal with multiple response. When discarding invalid ARP response, it acknowledges the transaction without setting the PDT.

PDT maintenance (refresh)

Each time a transaction request is issued from local bus, portal refreshes the corresponding entry (if valid !) in its PDT, for instance, setting an activity bit.

PDT maintenance (deletion)

Every PATH_TIMEofLIFE period activity bit could be checked. If not set, the path may become useless and should be removed from PDT. Portal will prevent previous node next access to this path, if it is a transaction request.

Path Descriptor Table - Contents -

path_descriptor	phy_id	act
25	6	1

32 x 32

Here is an example of 1024 bits PDT provided with 32 entries, with index from 0 up to 31.

Each entry includes:

- path_descriptor field (25 bits) that should store the source route.

- phy_id field (6 bits) that should store the physical_id of the source node that is allowed to request this path. All transactions going from a source node to a destination node will use the same path for both request and response (part of 1394.1 requirements).

- act field (1 bit) that is set every time source node is referencing corresponding index. This bit is reset every PATH_TIMEofLIFE period. When it remains null, portal assumes path inactivity then invalids the index (set phy_id to 0x3F for instance).

This PDT content is illustrative only and is implementation free as far as PATH_TIMEofLIFE activity period is managed.

PATH_TIMEofLIFE duration, attached to PDT management,
has to be shorter than
PATH_CLEANUP_TIME duration, attached to routing label update (add/remove portal).

SRM Basic Principles

Adding / Removing portal

Path Descriptor Table management

P1394.1 requirements

Conclusions

P1394.1 requirements - Clock synchro -

Clock synchronisation needs a spanning tree which root is the Net Cycle Master.

Very first ARP request could be the opportunity to establish a spanning tree.

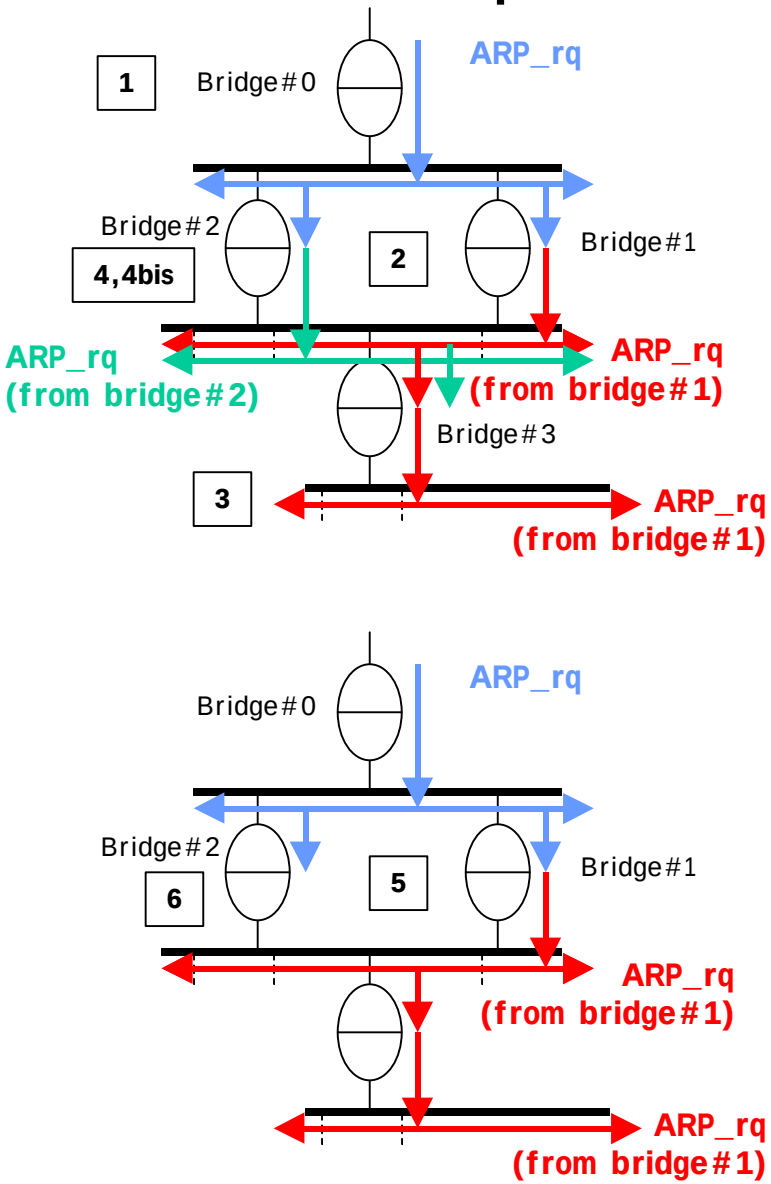
A specific NCM bit within ARP request could be use to identify a NCM candidate. This specific ARP request is a NCM selection command.

When concurrent NCM selection commands are flooding simultaneously, the bigger EUI-64 could be selected as the root.

Joining bus operation suite may trigger a new NCM selection command from surviving NCM. This NCM update should be a specific command to define the spanning tree within the victim network. NCM are informed about joining bus event when receiving a join_notification command from alpha-portal involved in joining operations.

The NCM selection command allows any node to become the Net Cycle Master, then it becomes possible to select among those who are candidates.

P1394.1 requirements - Breaking loop -



1/ Very first ARP request is propagated through bridge#0, then reaches bridge#1 and bridge#2 that form a local loop.

2/ Both bridge#2 and bridge#1 propagate the ARP request after updating the path descriptor field. Each of them receives its own ARP request and probably neighbour bridge ARP request.

3/ First ARP request that is forwarded will be propagated by bridge#3, the other being discarded based on ARP request contents analysis (breaking loop).

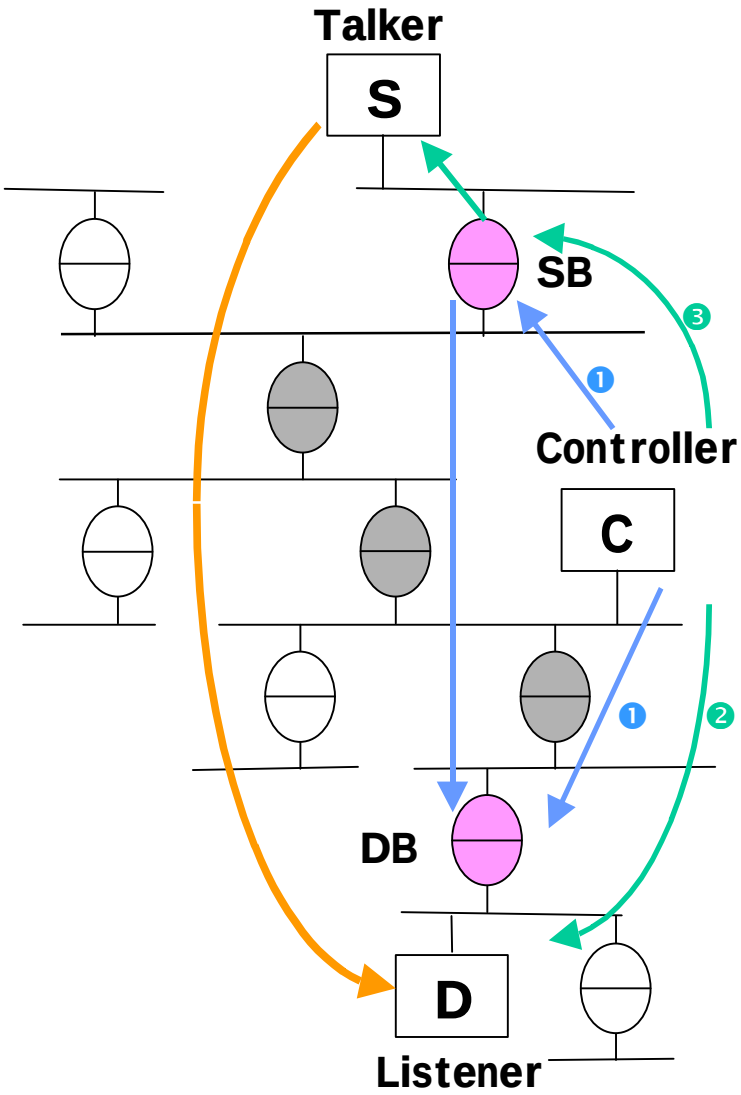
4/ Bridge#2 receives ARP request from bridge#1 before its own transmission, then detects local loop. It has to disable itself, then will not propagate next ARP request.

4bis/ Bridge#2 receives ARP request propagated from bridge#1 before forwarding the same ARP request. It can disable itself from this stage.

5/ Bridge#2 will not propagate anymore ARP request from bridge#0, then this path can't be used by any transaction as ARP request is made prior to any remote asynchronous traffic.

6/ This situation remains true while bridge#2 is observing the local loop.

P1394.1 requirements - Matchmaker -



Aim is to make S (or Talker) send data to D (or Listener)

❶ The controller C has to know both S and D EUI-64 so that C can use ARP_{SRM} mechanism to find paths descriptors to talk to S and D

❷ C sends a command to D to indicate it is going to receive data

❸ C sends a command to SB to indicate that S has to send data to D:

first, SB uses ARP_{SRM} mechanism to find path descriptor from SB to D
second, SB sends the command to S

Now S is able send data to D

SRM Basic Principles

Adding / Removing portal

Path Descriptor Table management

P1394.1 requirements

Conclusions

Conclusions - Much more simple ! -

Command name	Type	Comments
alpha_id “To select surviving alpha” issued by alpha portal	Local GASP (Broadcast write ?)	Required anyway by virtual Id mechanism.
phy_virtual “To communicate phy<->virtual mapping table” issued by alpha portal	Local GASP	Required anyway by virtual Id mechanism.
ARP request “To get path description to a node” issued by source device, bridge aware	Global GASP	Similar command is required by every routing mechanism to find-out a destination.
ARP response “To trace a possible path to a node” issued by a destination bridge	write transaction	Multiple answer could be possible that is a way to deal with congestion !
NCM selection/update “To set a spanning tree” issued by NCM candidate	Global GASP	Multiple command may flow simultaneously, bigger EUI wins.
join_notification “To inform NCM about joined bus” issued by alpha-portal	write transaction	If NCM receives two of them it is from the same network else NCM update command has to be transmitted.

Conclusions - Key advantages -

Less protocol overhead (reduced command set).

Easy to implement (trivial algorithm).

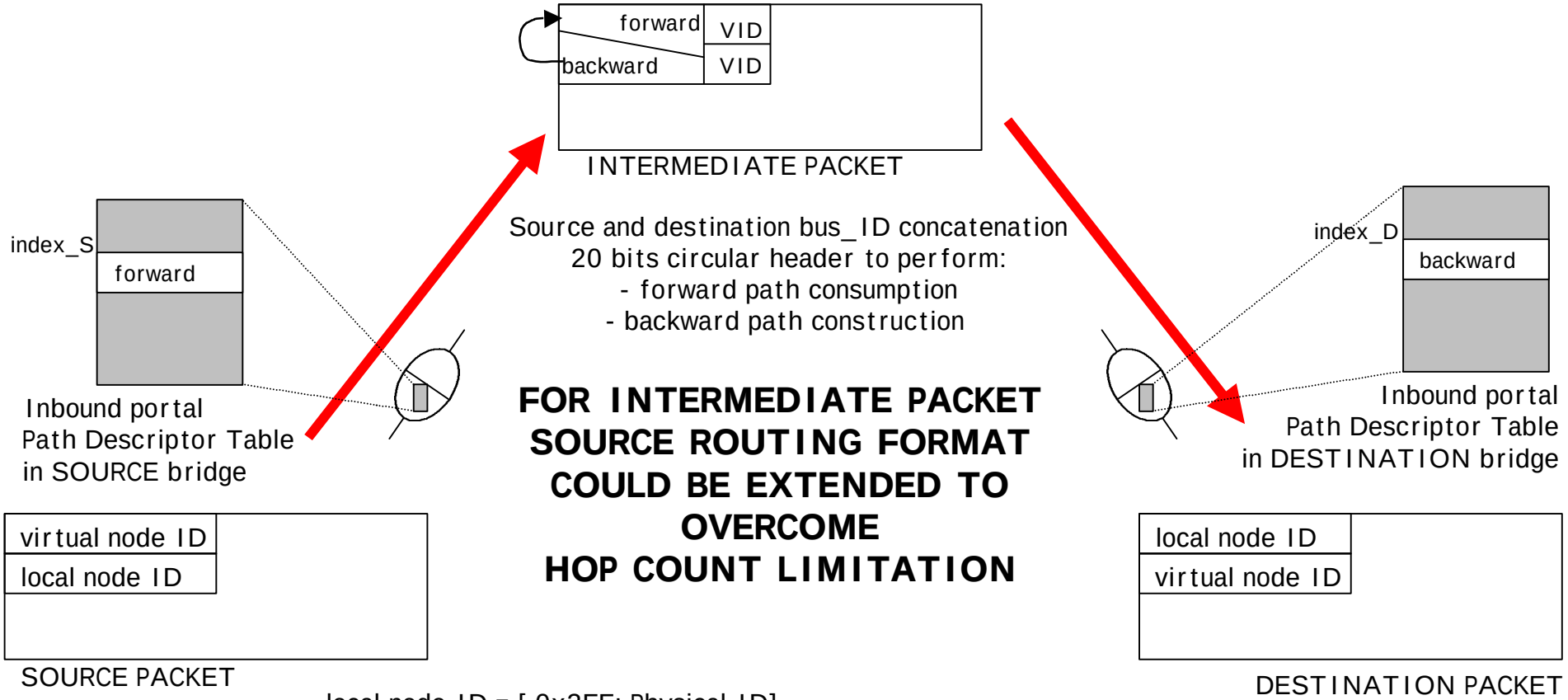
Rather low cost (similar to bit mapped routing).

NCM selection is made possible.

Scaleable and reliable hot-plug at bridge level.

Fulfil P1394.1 requirements.

Conclusions - Extend hop number -



local node ID = [0x3FF; Physical ID]

virtual node ID = [Index ; Routing Info ; Virtual ID]

- Give inbound portal Routing Label (RL)
  "N" bits, size of the routing info, number of portals per bus
- Index to retrieve the source route in the Path Descriptor Table (PDT)
  "10-N" bits, maximum number of entries in the PDT