

Bus Number Maintenance

David Wooten

Compaq Fellow

Compaq Computer Corporation

Some Terminology

Prime Portal – this is some arbitrary portal on a network of multiple portals/buses. This GUID of this portal is used to help setup an acyclic tree of connections of buses.

Alpha Portal – this is the portal on a bus through which traffic must flow in order to reach the Prime Portal.

Victim Bus – When two networks are joined together, there is a probability that one of them will have to change some mapping. This is the *victim bus*.

Survivor Bus – After two or more buses is joined, the *survivor bus* is the one that doesn't need to change any addressing.

General Approach

In constructing the protocols that are presented here, the overriding concern has been reliability in messaging. In no place are broadcast packets used as there is a danger of irregular results (some portals get the message while some do not). Instead, directed writes are used so that we always know which portals have received the message and which have not.

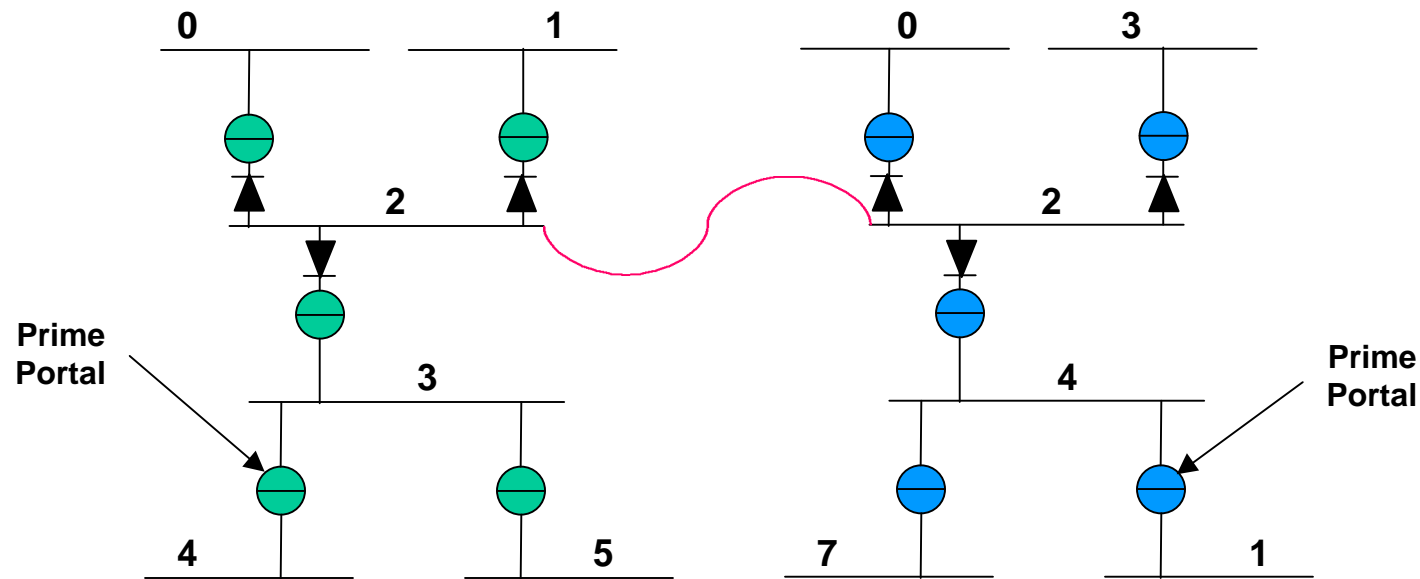
We should discuss the merits of this approach and decide if multiple 'broadcasts' can serve the same purpose.

Joining Buses

There are two approaches to trying to join buses. One tries to preserve any numbers that don't conflict. The other simply 'resets' the victim bus.

Joining Buses, the First Steps

First Steps (1a)



After bus reset, when new portal detected, all traffic that crosses the joined bus must stop.

Portals do this by not allowing a packet from another bus to another bus to be placed on the joined bus.

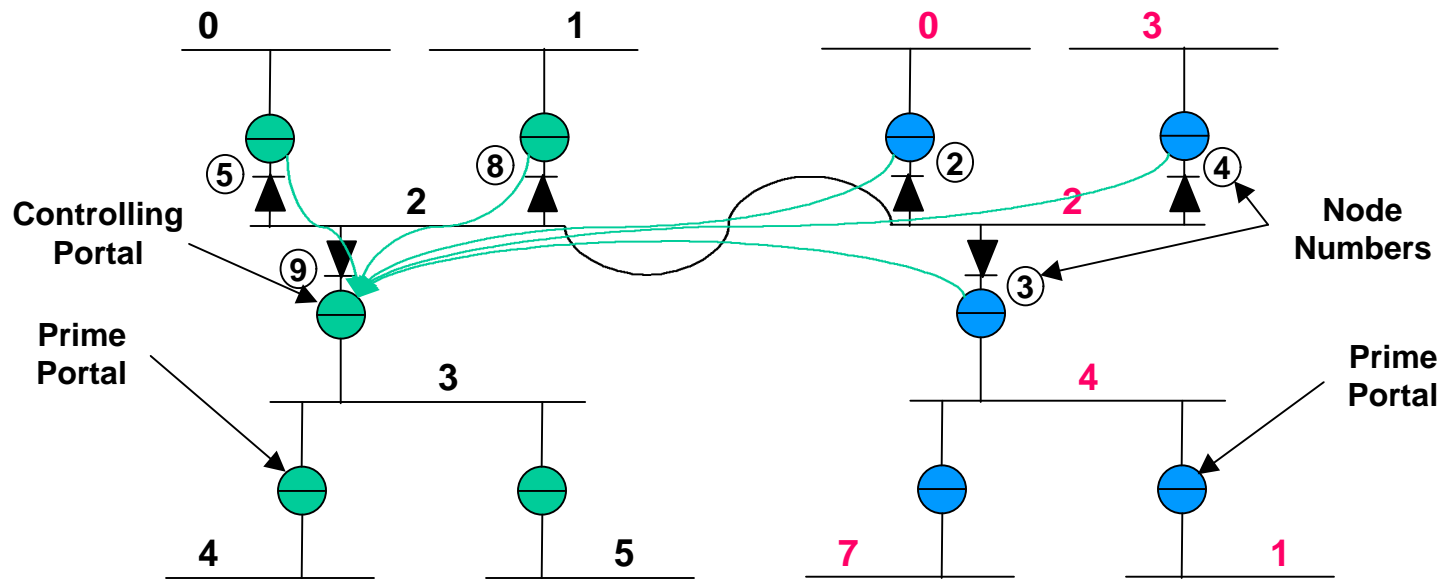
Portals need not ignore outbound traffic*.

Outbound Traffic

All portals stop sending packets that are not addressed to the local bus. They block this cross bus traffic from the time of a bus reset until they determine that it is OK to continue. If no new portals were added to the bus, the traffic can resume immediately. Otherwise, they must wait until they are notified by the 'controlling portal'.

Nodes (things other than portals) follow the same rules. If a node is not capable of detecting new portals it must wait until it gets notification of whether it is on victim or survivor bus before resuming outbound traffic. Could have bridges enforce these rules on nodes (not portals).

First Steps (1b)



Portal with highest node number ('controlling portal') reads 'relevant data'* from all other portals and picks a surviving and victim bus.

Relevant Data

When the controlling portal reads from a portal on the 'other' bus, it reads the following information:

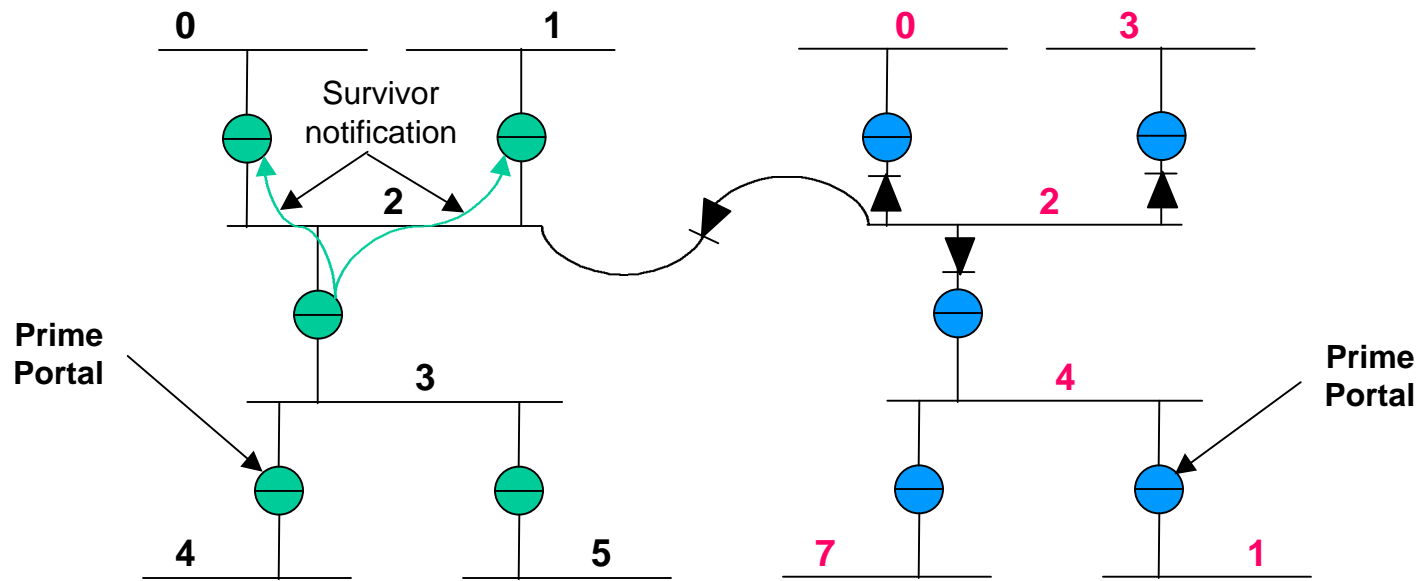
- 1) the node number of the Alpha portal of the segment
- 2) the GUID of the Prime portal for the segment
- 3) the node number of the Prime portal for the segment
- 4) the size of the attached network
- 5) the bitmap of outbound mappings

Picking the Winners

The surviving network should be the one with the most connectivity (most other buses attached.) In the case of ties, the GUID of the Prime Portal for the network will decide.

The accumulated bitmaps will tell us how big the network is. However, the bit maps should only be accumulated for the survivor bus. So, the size of the attached network is read along with the GUID. If this matches the 'chosen' Prime, then the bitmap information is added to the accumulated value.

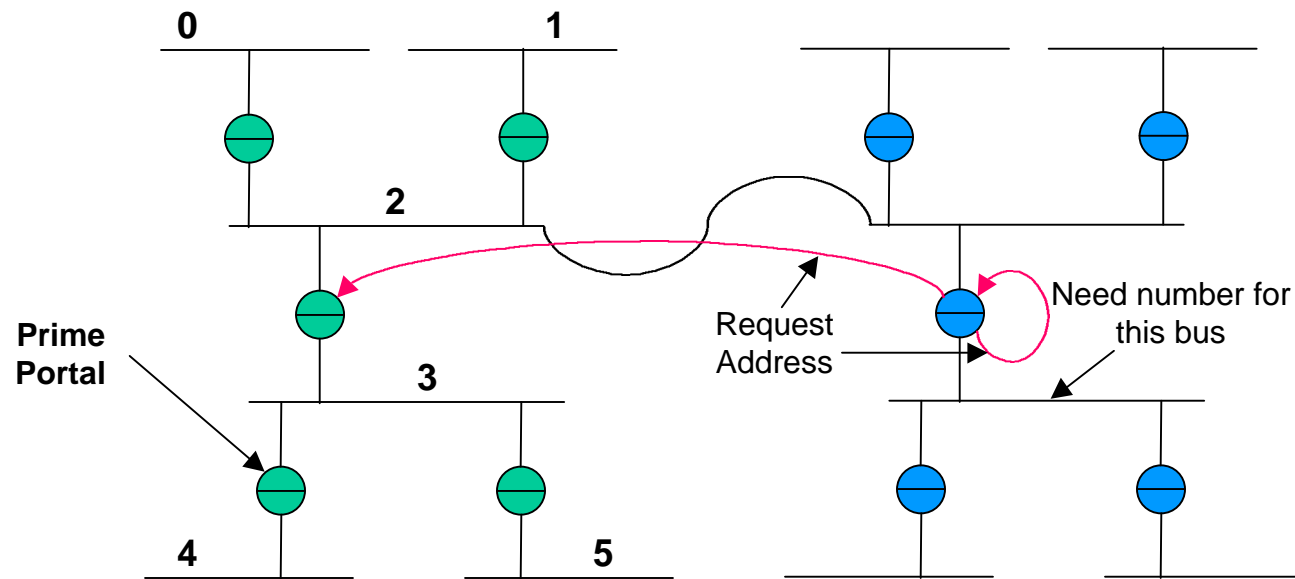
First Steps (1d)



Portals determine if they are part of the victim set or survivor set by comparing their Alpha portal number to the one given. If the same, then the portal is a survivor. When survivor nodes are informed, they can resume cross bus traffic.

Assigning Addresses

Assigning Addresses (1)

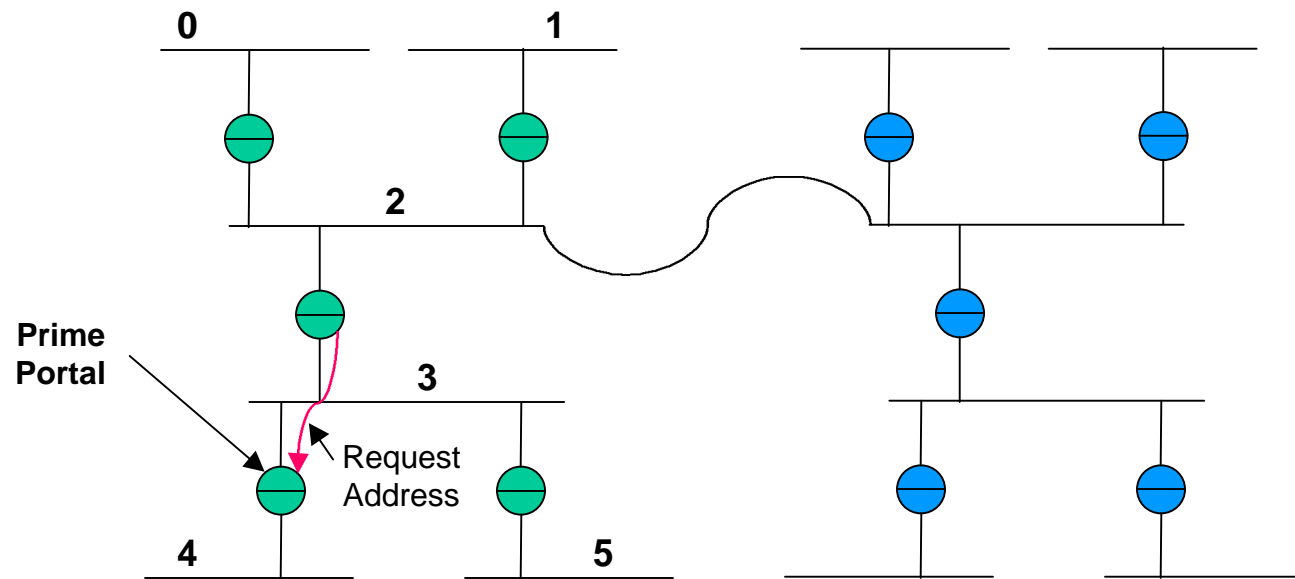


Portal needing an address makes a request to its co-portal.

Co-portal forwards request to Alpha portal on its bus.

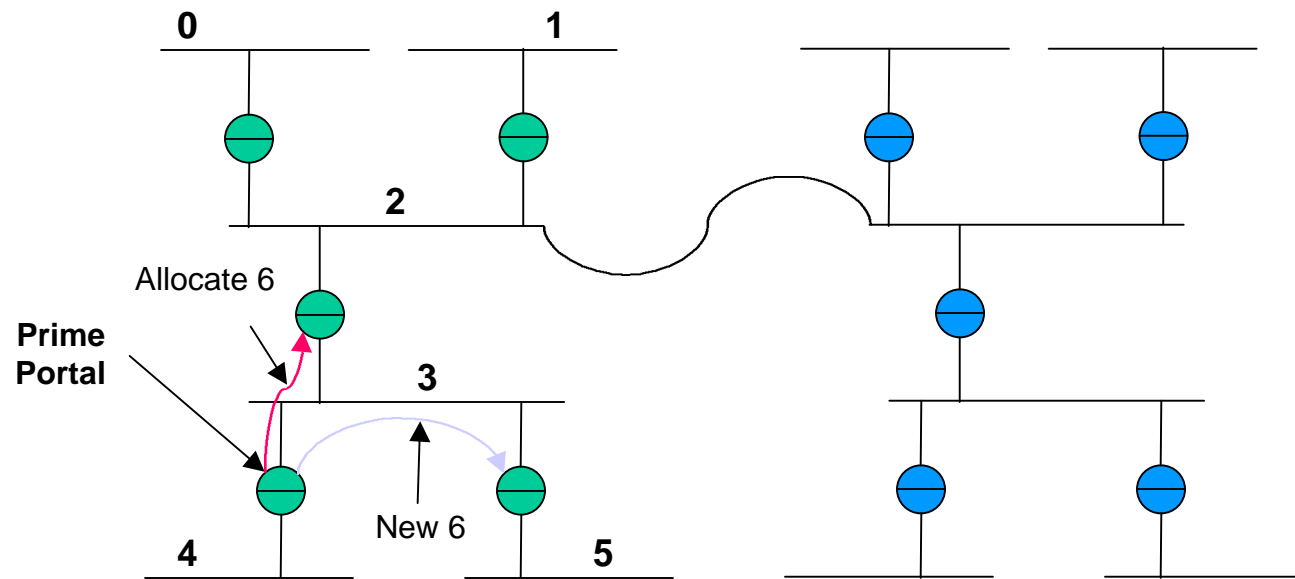
Note: request use 'local' portal-to-portal requests.

Assigning Addresses (2)



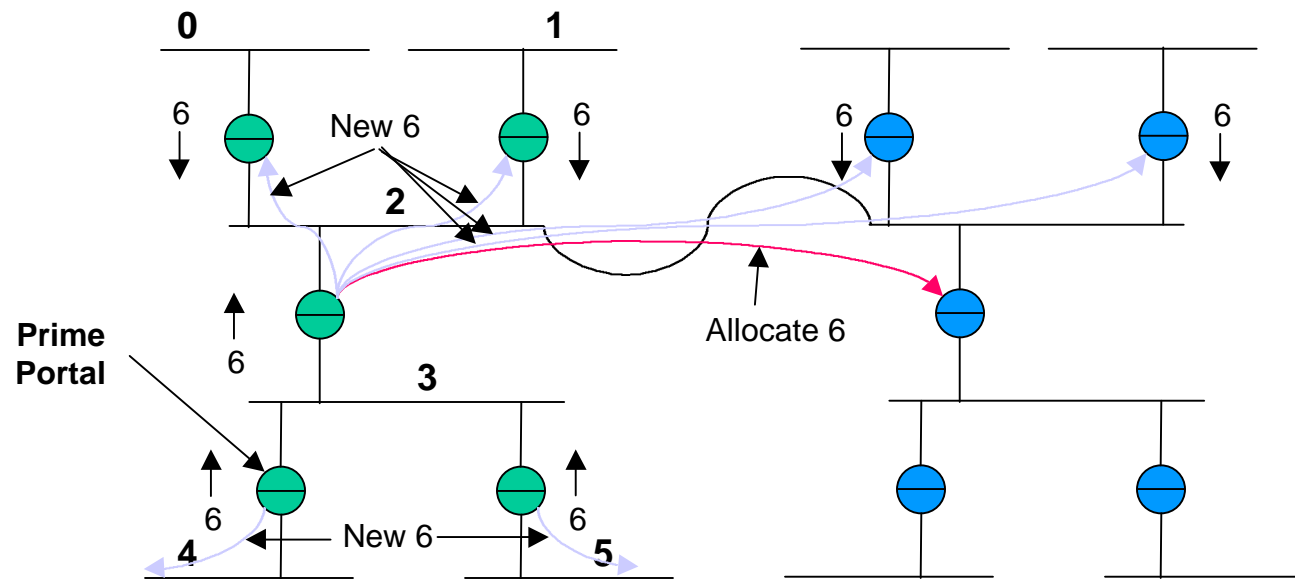
Request is forwarded through co-portals and Alpha Portals until it reaches the Prime Portal.

Assigning Addresses (3)



Prime Portal allocates address and sends two messages. One message to requestor indicating address is allocated. Other message to other portals on its bus to indicate that a new mapping is to be established.

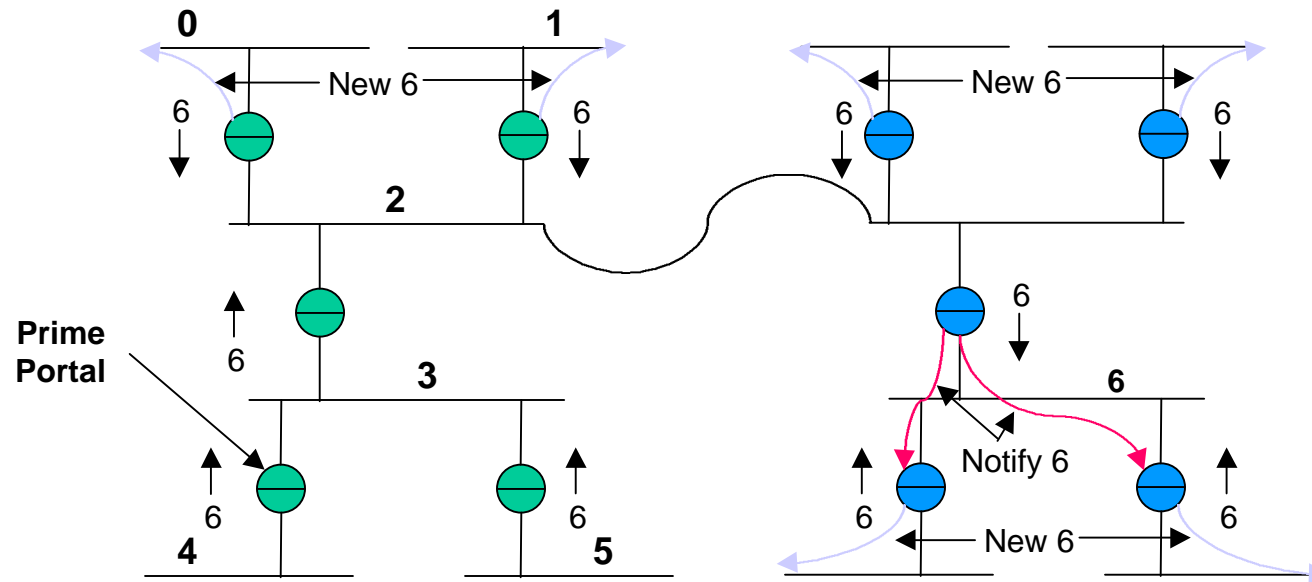
Assigning Addresses (4)



Allocate message propagated back to requestor. When a portal receives an *Allocate* it sets up an out bound mapping on the receiving portal

New messages generated to other portals along path. When a portal receives a *New* message, it sets up an outbound mapping on its co-portal.

Assigning Addresses (5)



When requesting portal receives the allocation, it sends an *Notify* message to all other portals on the bus. This lets those portals know the address that was just assigned to the bus.

Portals receiving a *Notify* message will send a *New* message on their co-portals.

Assigning Addresses - More

Another way of assigning addresses is to simply have the bus number returned to the requestor. The requesting portal can then send the bus number out on all attached portals. This causes the address to radiate to the other buses. If we are careful about this, we can allow each of the buses to be the center of its own acyclic tree which should result in the lowest hop count for any packets going to/from the bus.

This is not presented here because I haven't got enough of the details worked out.

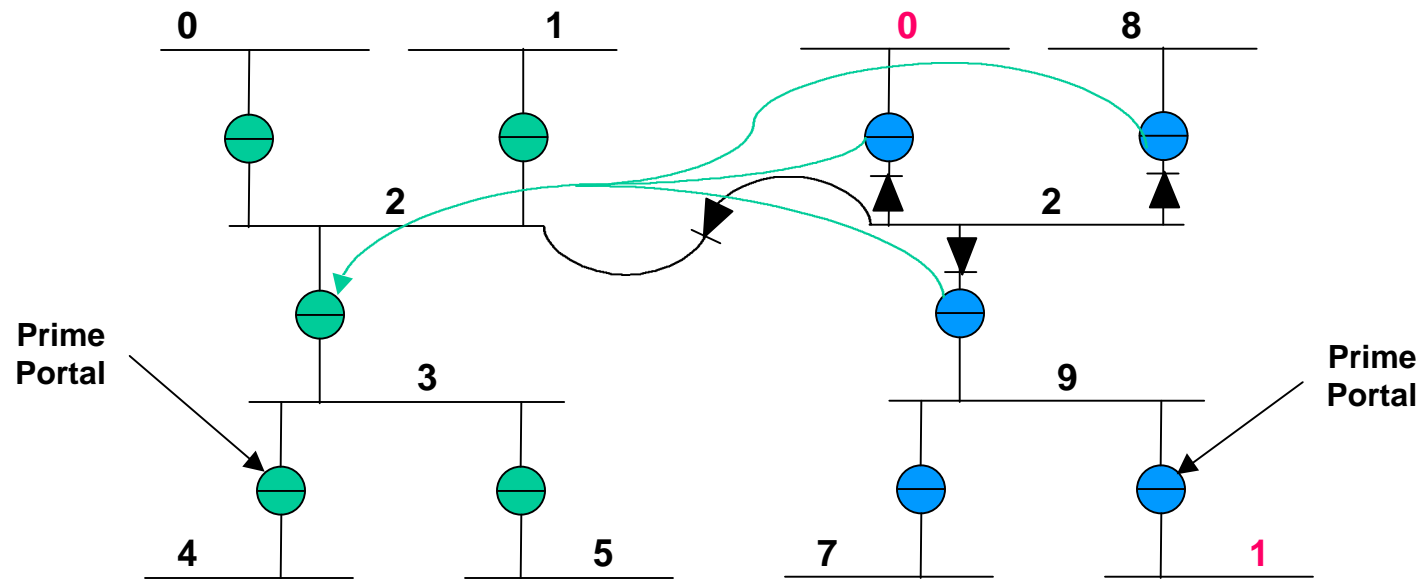
Preserving Numbers

WARNING: this is not well thought through.

Preserving Numbers

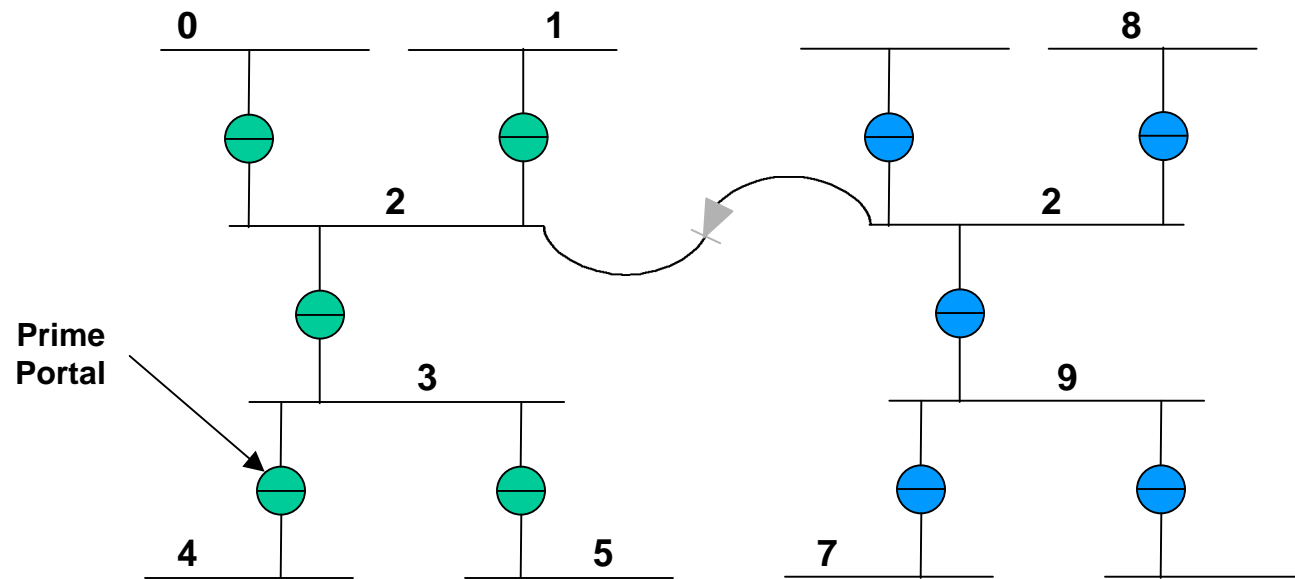
Concept here is that when two networks are joined, many of the addresses are likely to be unique and there is no reason to have to get a new bus number when the addresses don't overlap.

Preserving Numbers (3)



The Alpha portal on the surviving side reads the current bitmaps of the victim nodes. Victim nodes do not provide response to this request until they have completed notification of deletions.

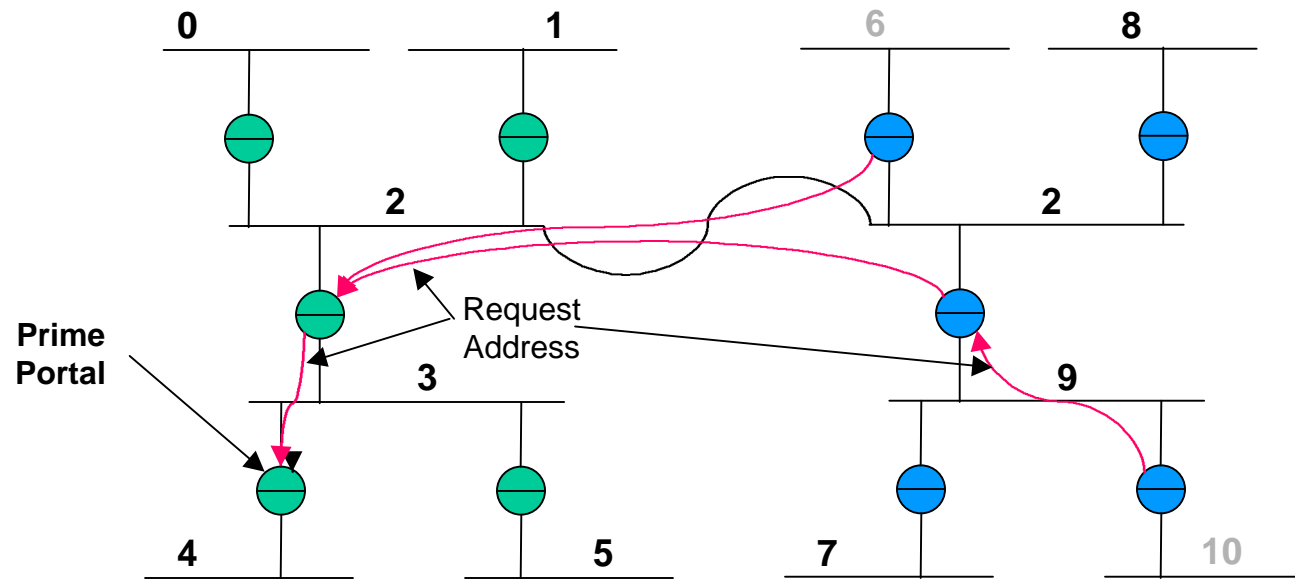
Preserving Numbers (6)



Victim buses 0, and 1 can no longer be reached.

Must stay in this state until all requests on victim bus have timed out (can eliminate timeout if all nodes are informed about changes.)

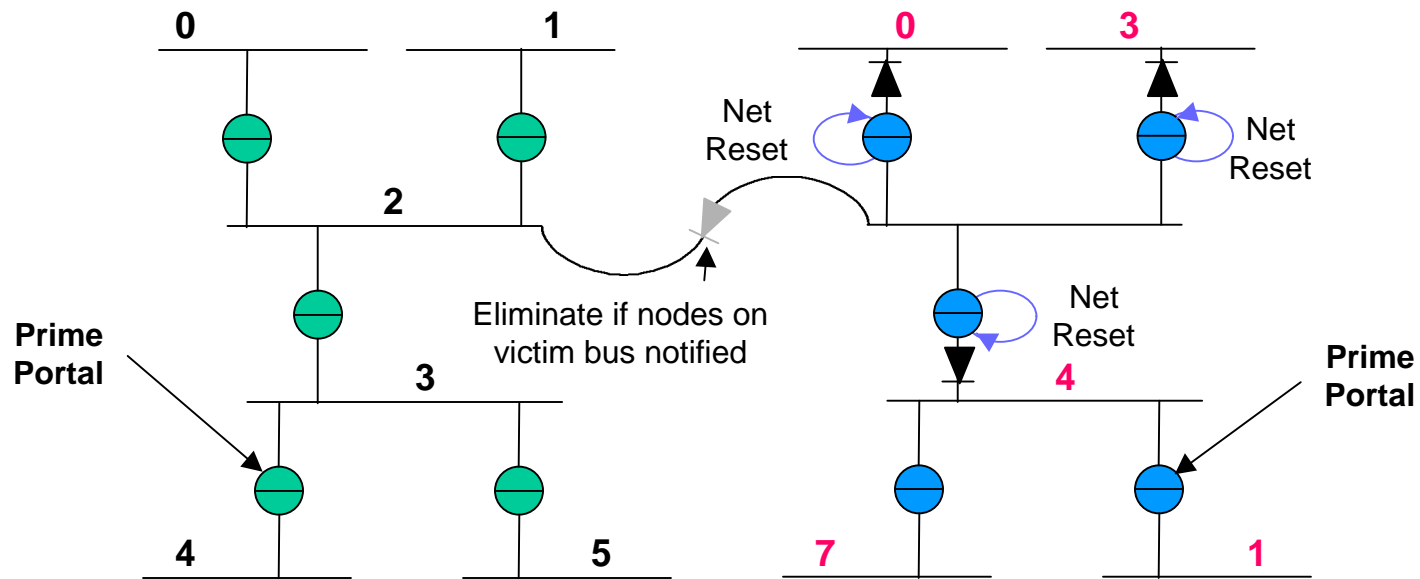
Preserving Numbers (7)



After address refresh timeout (seconds? minutes?) each bus that needs a number can now access Prime Portal to get an unused address.

Net Reset

Net Reset (2)

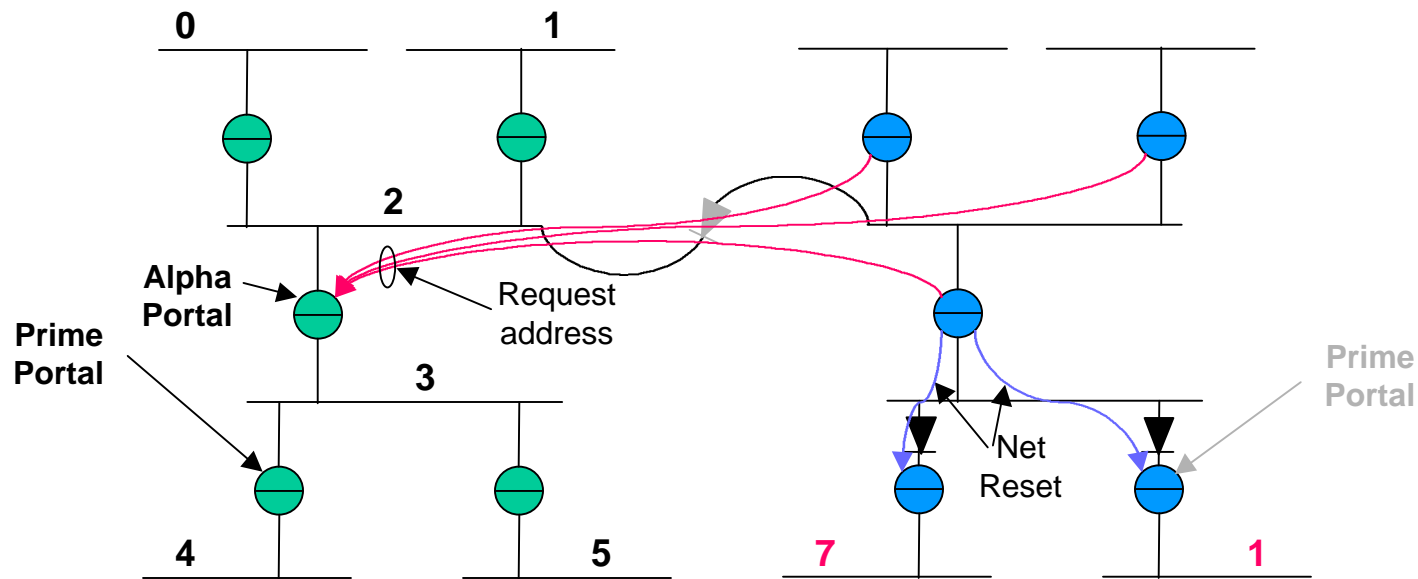


Propagate victim notification to co-portal. This is treated as a net reset. Notification contains bitmap of all bus numbers used by survivor net.

As reset propagates, requests are 'turned around' (send back response to indicate change in progress).

Responses are pushed ahead of the reset.

Net Reset (3)

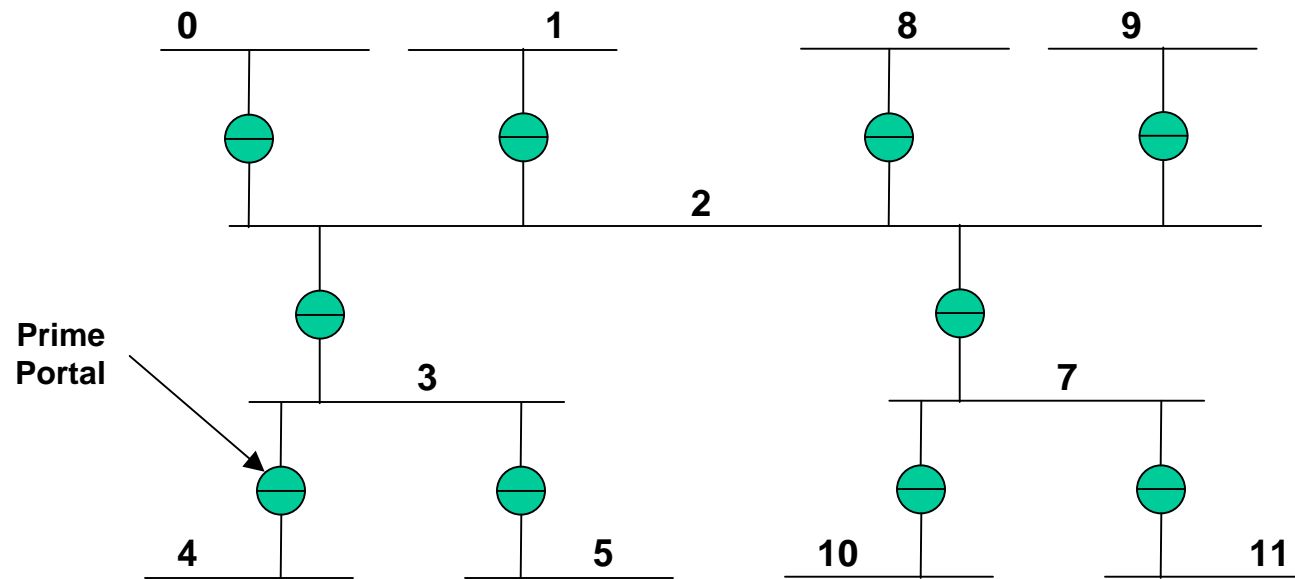


Continue to propagate bridge reset (directed writes to each bridge and node).

When a Prime Portal receives reset, it is no longer a Prime Portal.

Can request address assignment as soon as net reset passes portal.

Net Reset (4)



If bridge aware nodes are notified of net reset, then operation is completed. Otherwise, must wait from time of reset until address refresh interval is over before putting the new addresses in effect.

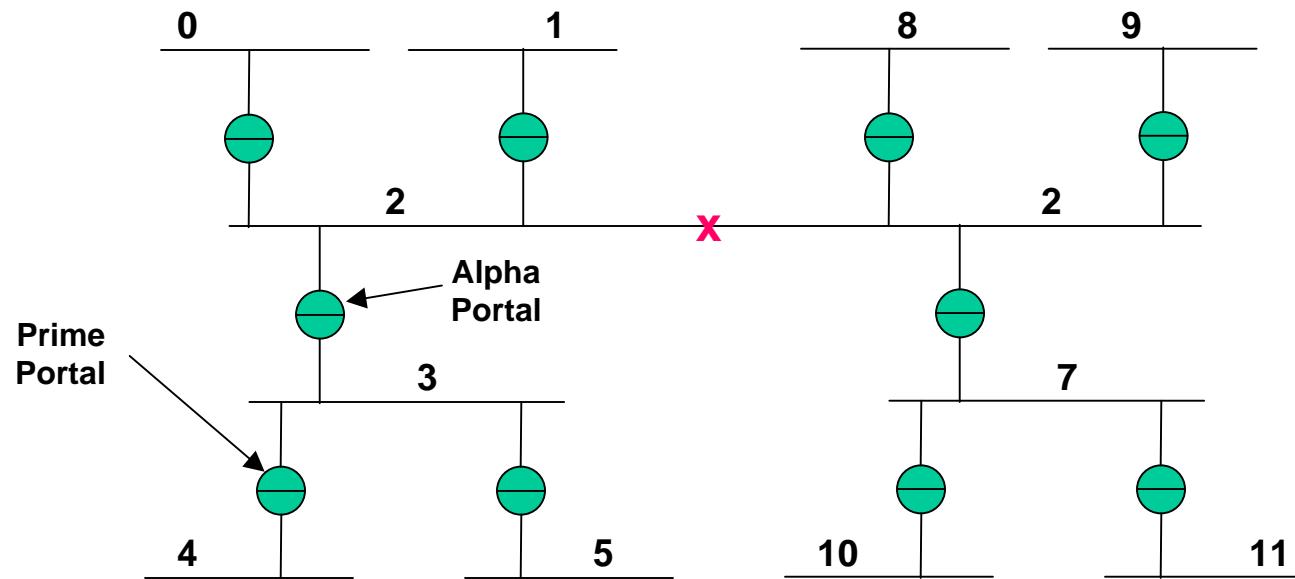
Recommendation

Use the Net Reset approach to joining buses. Don't try to preserve any node numbers in the old network.

Have Net Reset sent to nodes to avoid timeouts and (address refresh and remote transaction.)

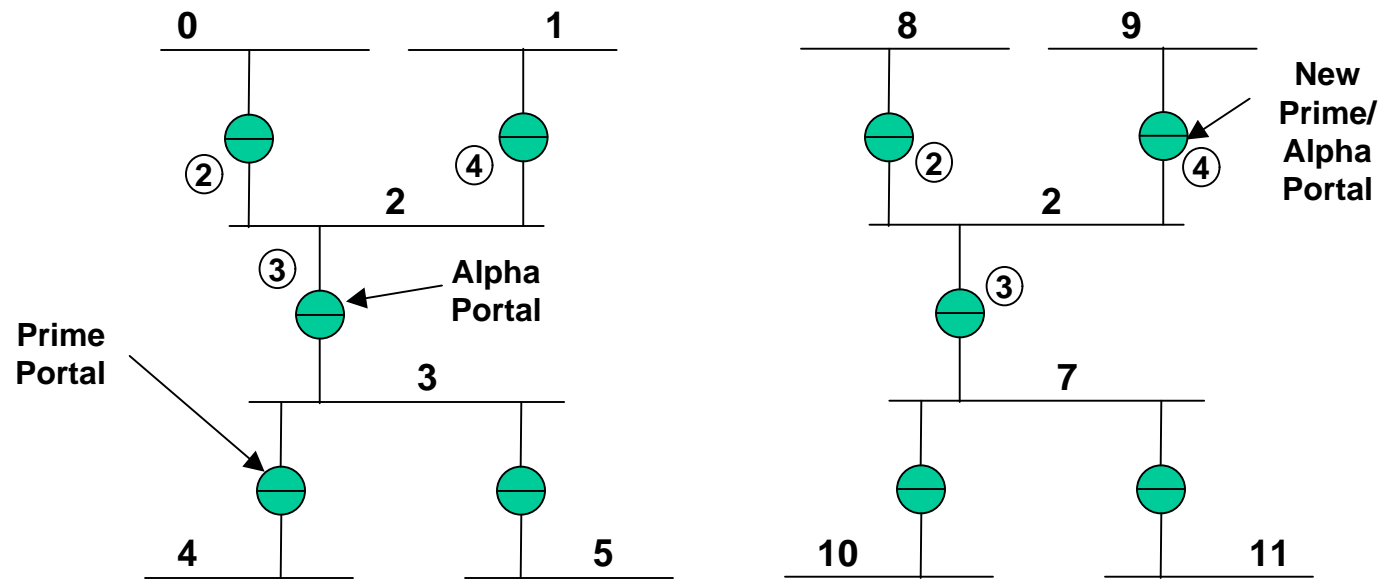
Breaking the Net

Breaking the Net (1)



When a bus connection between portals is removed, the portals end up in two groups. One group is still connected to the Alpha Portal and one is not.

Breaking the Net (2)



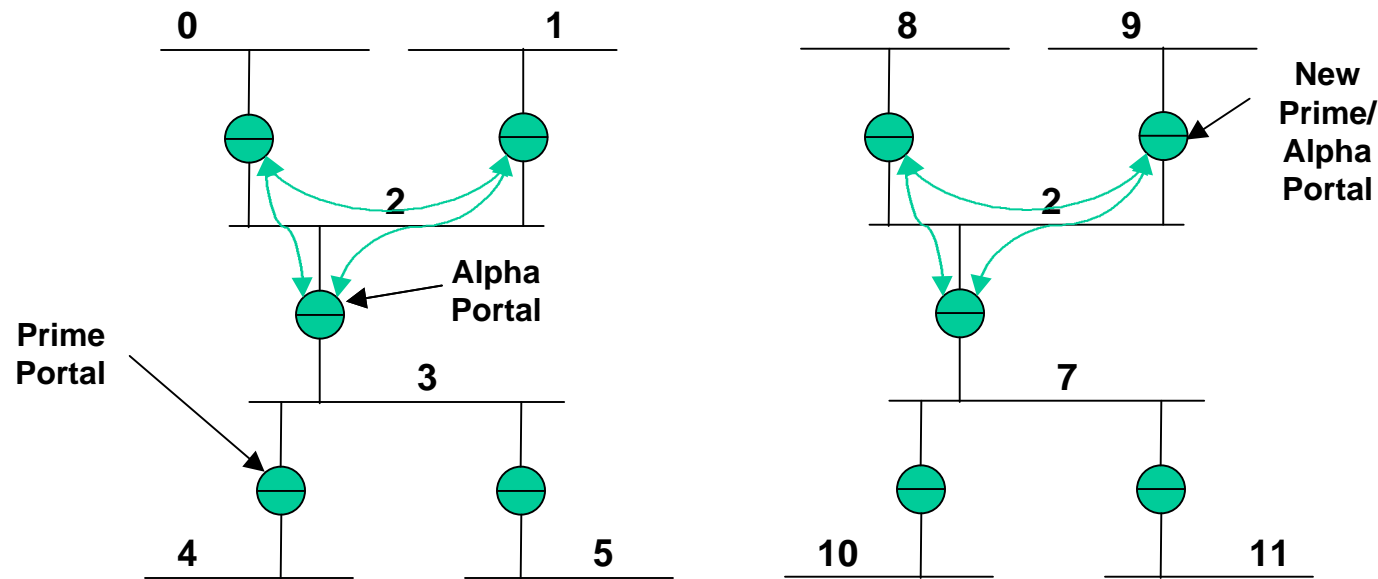
Cross bus traffic continues*.

On the segment not connected to the Alpha Portal, the highest numbered portal becomes the Prime Portal for the new net and the Alpha Portal for the new segment.

Cross Bus Traffic

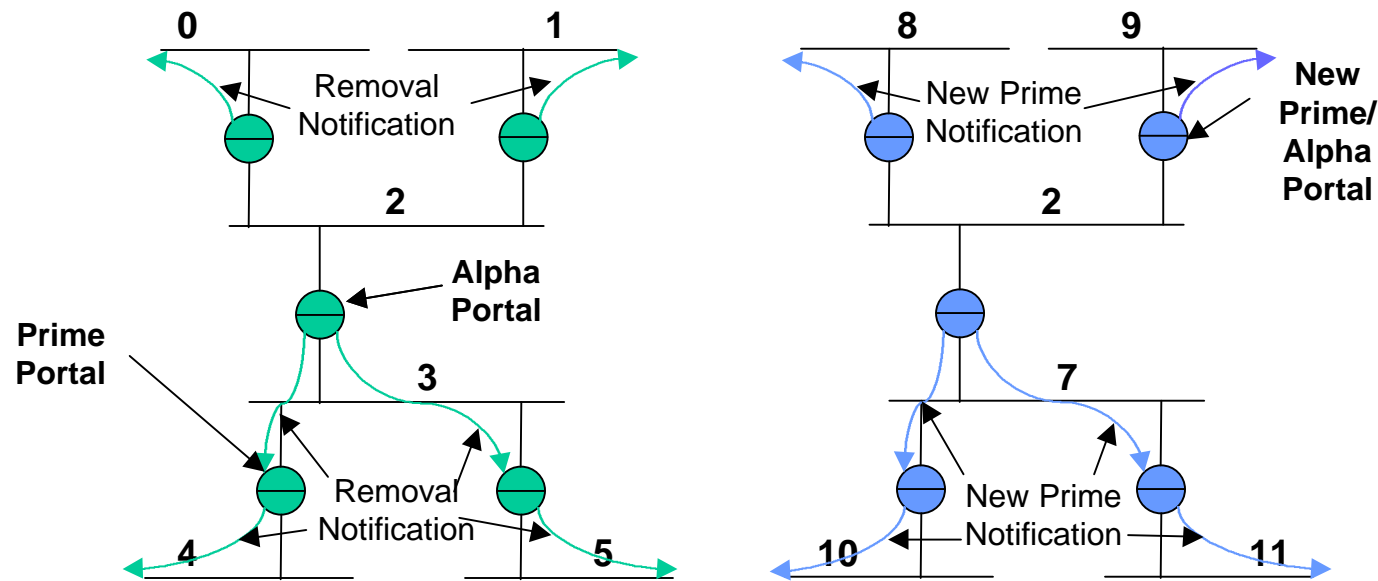
When a portal puts a packet on the bus that is intended for another bus, and no ACK is received, the portal should not give up. It should do an inquiry on the bus number to see if it is still mapped (put a specially formatted packet on the bus and see if it gets ack'ed.) If the inquiry packet does not get an ACK, then the assumption is that the addresses bus is no longer reachable and the request is turned around with the appropriate error indication.

Breaking the Net (3)



Each Alpha Portal reads outbound mapping for each of the other portals on its segment. This represents the map that the co-portal should have as an inbound map. The difference in the number of mapped buses is computed to give the new net size.

Breaking the Net (4)



Each portal communicates new map to co-portal.

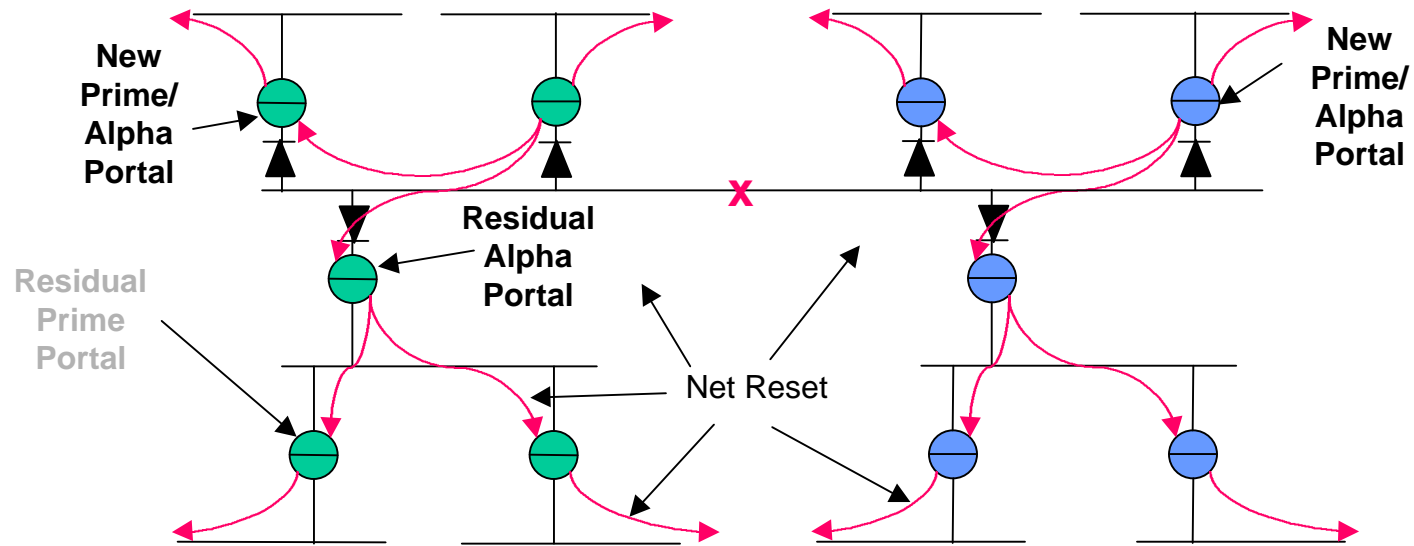
The co-portals take the difference between the inbound maps and transmit a Removal notification to all other portals on its bus.

On net with new Prime, send GUID of new Prime Portal (not a net reset notification) along with Removal map.

Why Not Just Reset?

One alternative to all the previous machinations would be to send a *Net Reset* whenever there is a change in topology. This can be made to work. Primary issues that needs to be resolved before we take such a step is what efforts should we expend to try to minimize the network disruption when there is a change in network topology. This is analogous to the question that we have to resolve w/r/t node ID's and persistent node numbers. It is not identical to persistent node numbers because bridging does not have to deal with things like suspend/resume and other things that can cause a fairly high reset rate.

Breaking the Net - Simplified

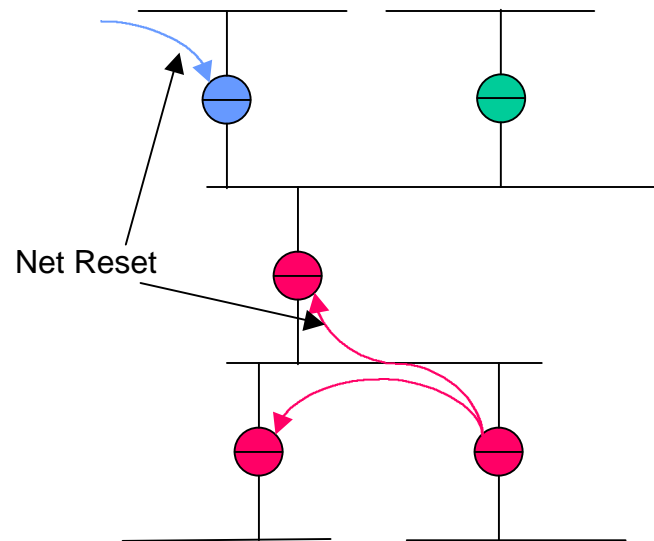


Cross bus traffic stops.

Choose Prime/Alpha portals for each net segment.

Send Net Reset and start all over.

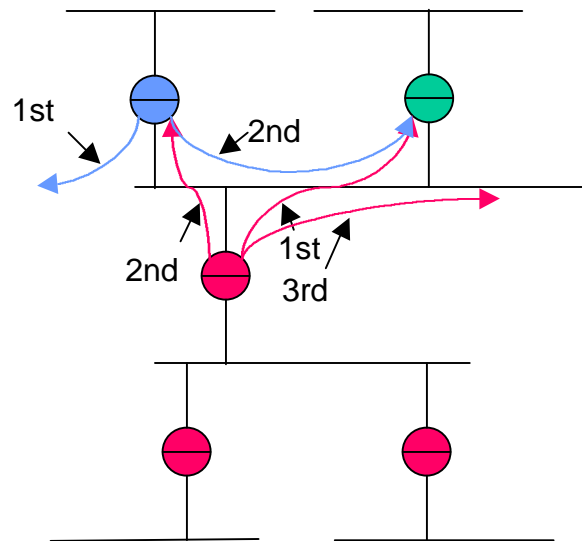
Net Reset Collision (1)



Normally, when a portal receives a Net Reset notification it simply propagates it to all other portals attached to the co-portal.

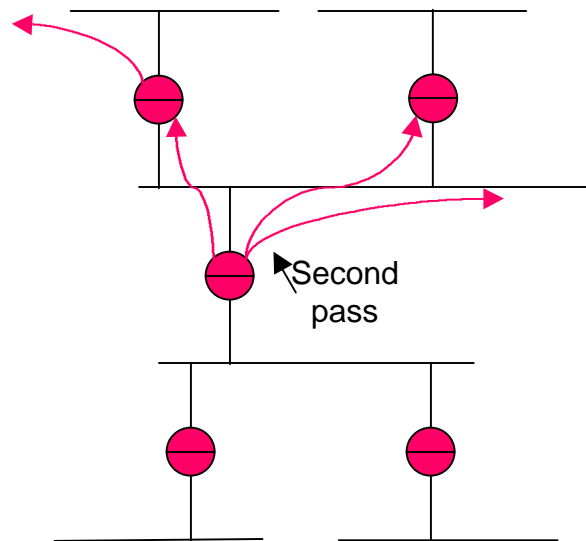
When Net Resets arrive at a bus from different directions, the portals need to adopt a modified behavior to make sure that we end up with consistent results

Net Reset Collision (2)



If a portal is propagating a reset and receives a reset it determines which reset should survive by looking at the associated net size and GUID of Prime Portal associated with the reset. If the other reset is the survivor, it quits sending its reset notifications.

Net Reset Collision (3)



Winner must do a second pass.

Loser must propagate winning reset back.

Nodes receiving the same reset values twice, ignore the second reset value.

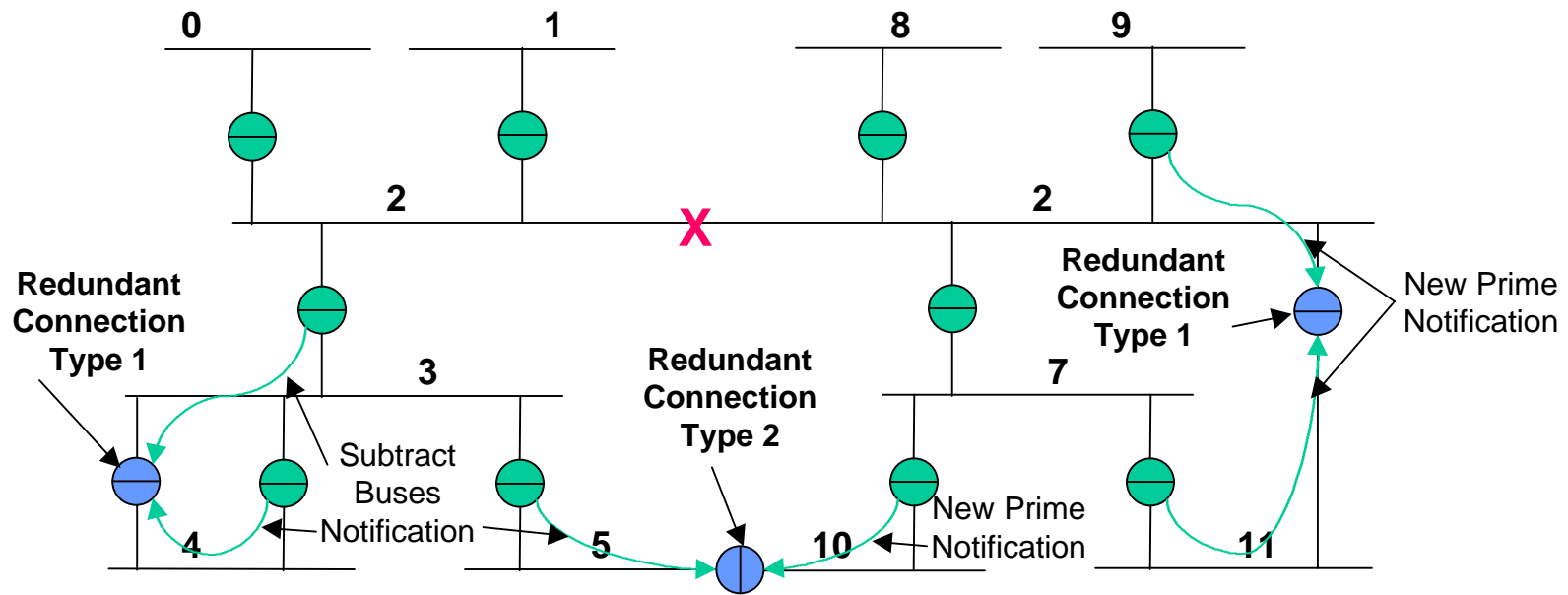
Notification to Nodes

It is highly beneficial to send the address change and *Net Reset* notifications to all bridge aware nodes on the reset net. If reset is not sent, then address changes to the net cannot be enabled until an address refresh interval has expired.

Sending these notifications also lets nodes know that any responses that have not been received, will not be received. This avoids having lost responses during reset and allows us to not have to push responses ahead of the reset

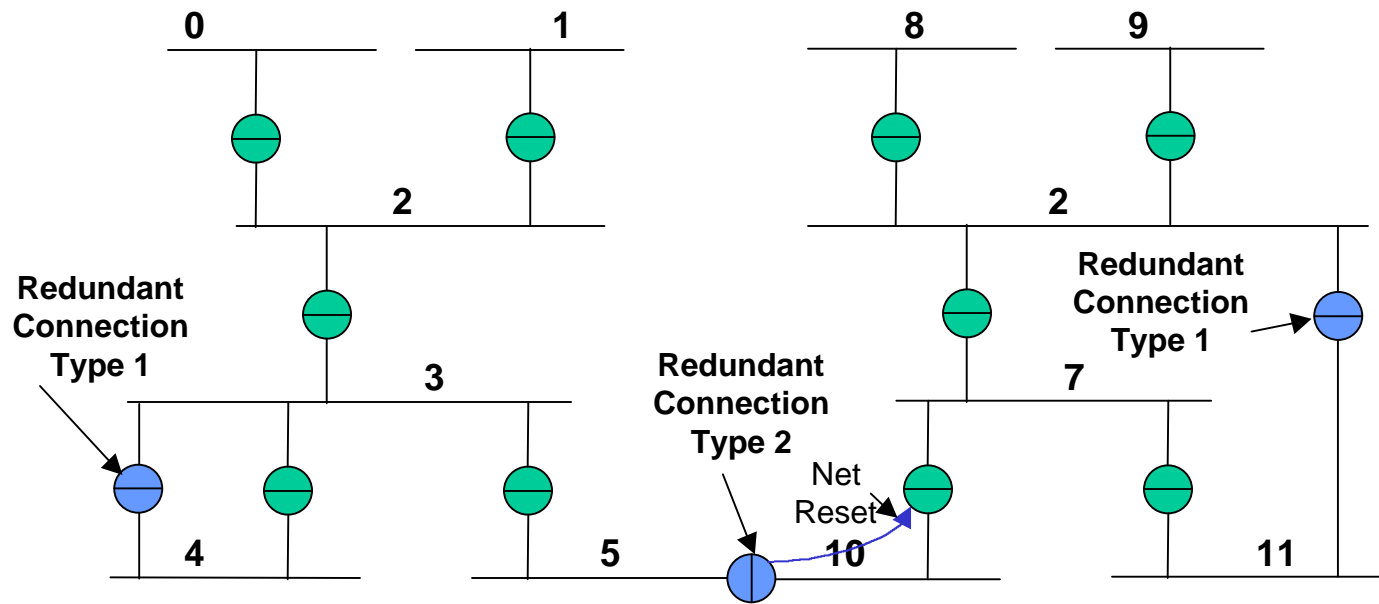
Problem is making sure that all bridge aware nodes are capable of receiving the reset notification (what if ack_busy or suspended connection?)

Redundant Connections (3a)



After break, two types of notifications are sent.
 Bridge waits until it has received notification from on both portals
 If notification type is the same, then the connection is Type 1. If the notification is different, it is Type 2

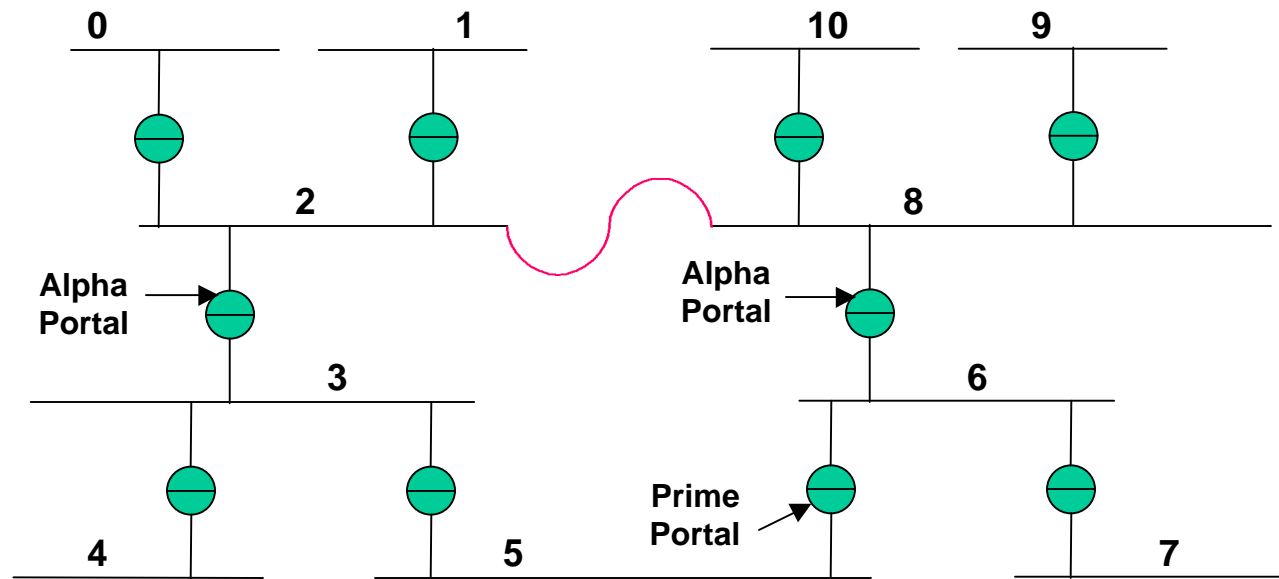
Redundant Connections (3b)



Type 1 bridges remain 'inert'.

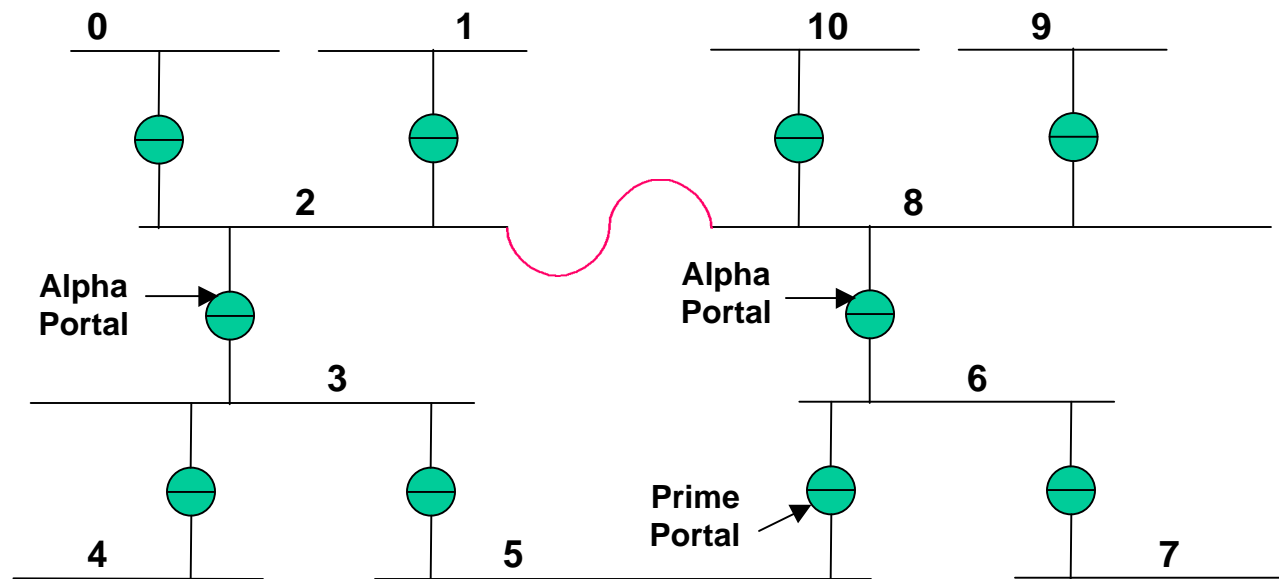
Type 2 bridges become a controlling portal as if a connection was made. It compares the sizes of the two buses and, if necessary, the GUIDs of the Prime Portals and resets the smaller.

Redundant Connections (4a)



Here is a harder problem. Have a network and someone makes a connection that creates a loop. During the initial processing, we find two Alpha Portals associated with the same Prime Portal.

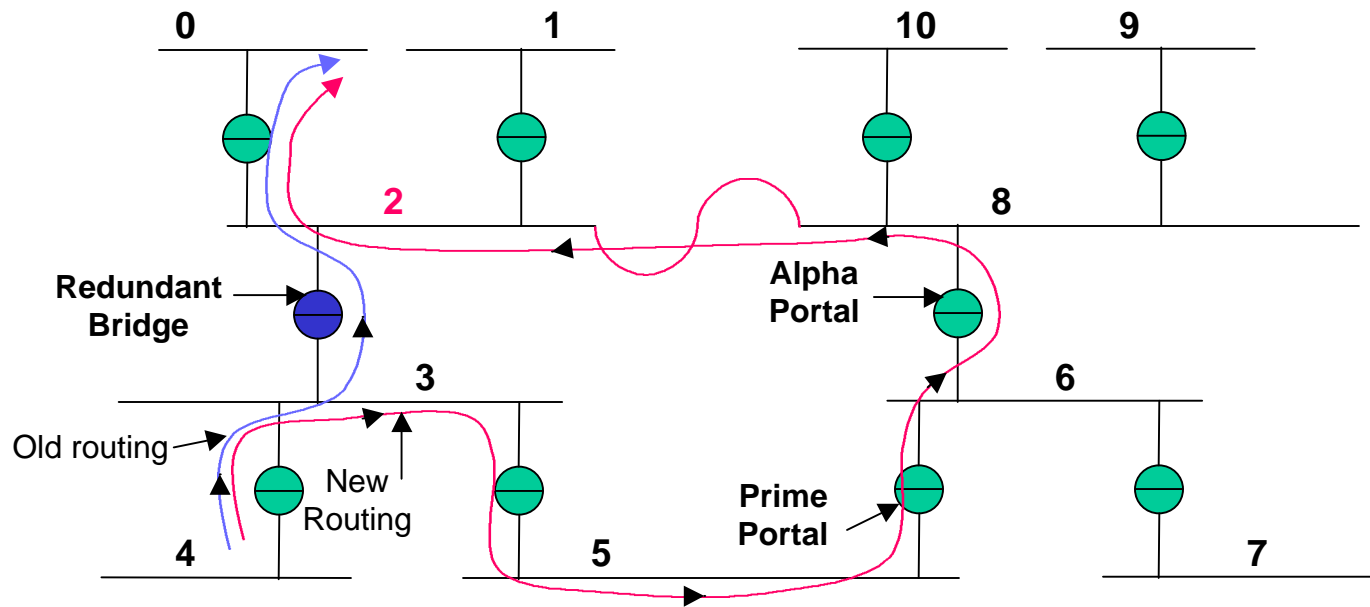
Redundant Connections (4b)



Have two choices at this point. Can either pick a portal on the new bus to be Prime and do a Net Reset or try to 'clean up'.

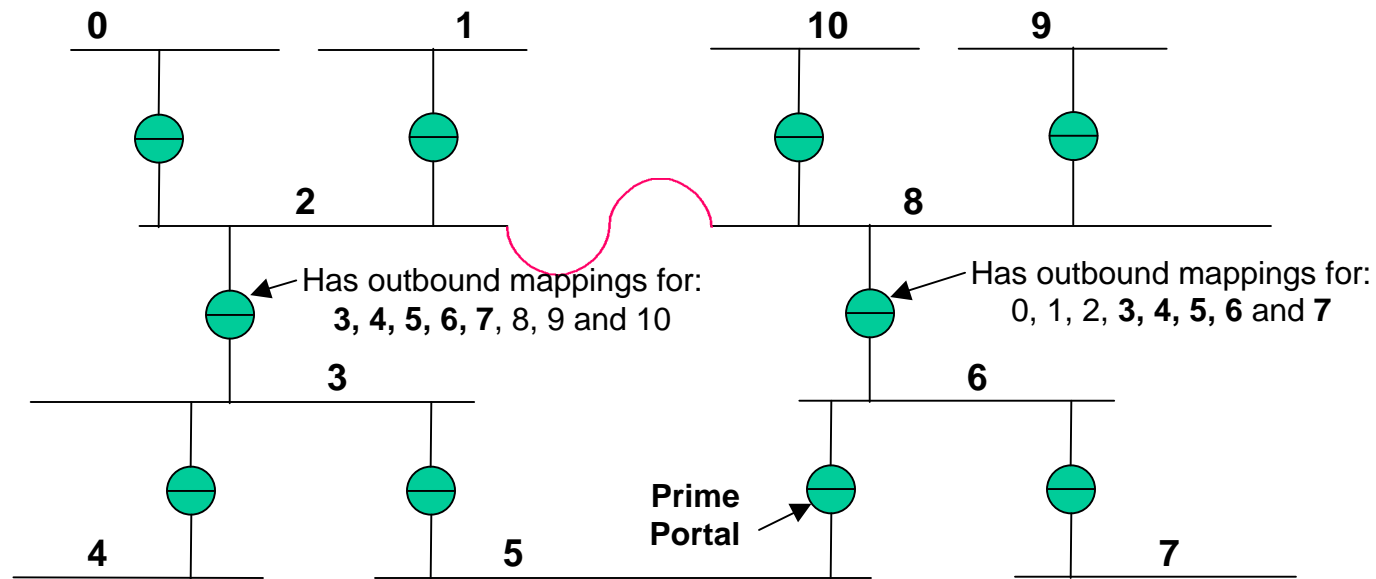
'Clean-up' means that we have to eliminate a bus number, disable one of the Alpha Portals and reroute traffic.

Redundant Connections (4c)



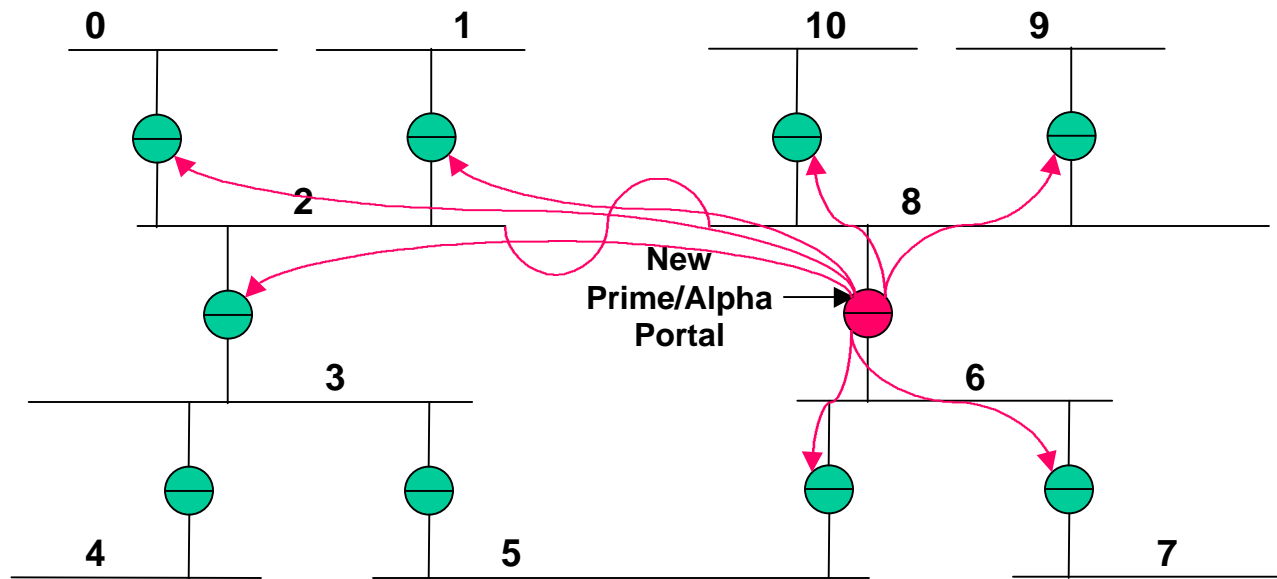
An example of the rerouting. Packets from bus 4 would get to bus 0 via the now redundant bridge between buses 3 and 2. They would have to be rerouted through 3, 5, 6 and 2/8.

Redundant Connections (4d)



Reason we have to set up new routing is that on the bus with the redundant connection, there are two outbound mappings for any address.

Redundant Connections (4e)



I think we can/should, simply have one of the portals generate a net reset for the whole network (I'm not yet sure that we don't have a problem with nodes ignoring the reset because of the size of the net or the value of the GUID).

Mixed Messages

Need to make sure that portals know how to handle the confluence of multiple messages on the same bus. These can occur because different things can happen on the net at the same time.

The Messages

The messages defined so far are:

- 1) Net Reset – portals receiving this erase all bus numbering information
- 2) Change Addressing (Subtract or New) – portals receiving this retain old Prime but get new mapping information.
- 3) New Prime – same as above but with new Prime
- 4) Request Address – request to Alpha portal to get address from Prime Portal
- 5) Allocate - Notification of new address being allocated
- 6) Notify - Notification of other portals on same bus as to new bus number.

Combinations

On the same bus can have:

- multiple types of messages e.g., Net reset and Notify.
- multiple instances of the same type of message e.g., Subtract Buses

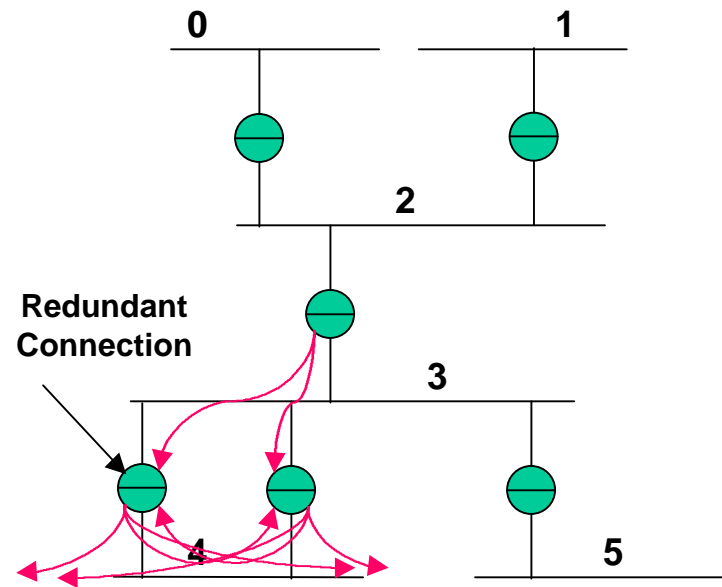
Analysis done but not written up yet. Believe that it all works.

Multiple Resets

A bridge can receive resets on both portals. When the reset values are different, the bridge decides which will survive and will send that reset.

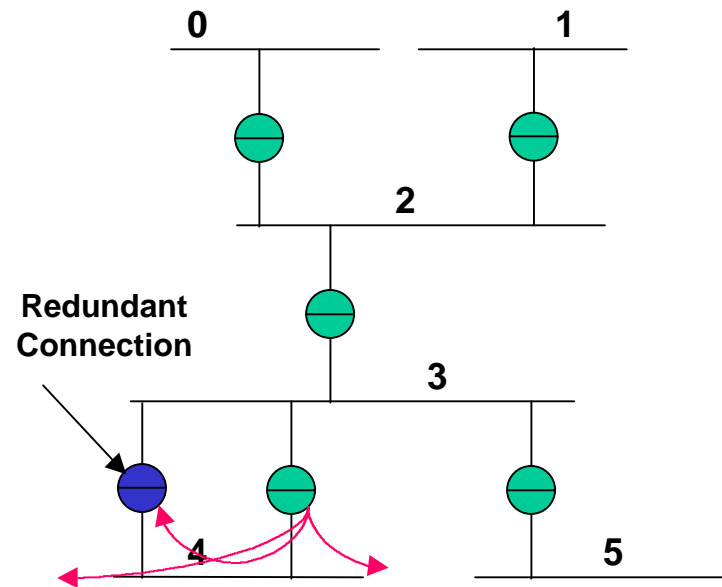
When the reset values are the same, it is an indication of a loop.

Duplicate Reset (1)



When there is a redundant connection, two portals may be trying to reset the same bus. When a portal doing a reset receives a *Reset* with the GUID that it is trying to send, it stops and becomes 'inert' (doesn't forward).

Duplicate Reset (2)



Portals need to send *Reset* twice to each other portal to make sure that Alpha portal is properly established (send to every portal and then send again to every portal.)

Reset Notification to Node (1)

When Reset is sent to a node, the node may not be able to accept it (`ack_busy`). Furthermore, the node may not be able to accept the Reset for quite some time and we can't have the bridges waiting for indeterminate periods of time before they can enable cross bus traffic. Propose that we maintain a counter on each bridge to indicate the number of Net Resets that it has processed. When a node has been unable to accept packets for some period of time, it must check the counter in the Alpha portal. If the counter has changed, a Net Reset is indicated.

Reset Notification to Node (2)

Recommend that new node S_i have a status bit that is set when the node receives a Net Reset Notification (write to a WKA.)

Conclusions

Management of bus numbers can be done by portals.
Very important that we find way to notify bridge aware nodes about changes to bus numbers (Net Reset).
Simplest way to handle additions and deletions to topology is Net Reset. Might also be the fastest in many applications.
Lots to write up/down.