RPR and 802.1D Bridging Issues

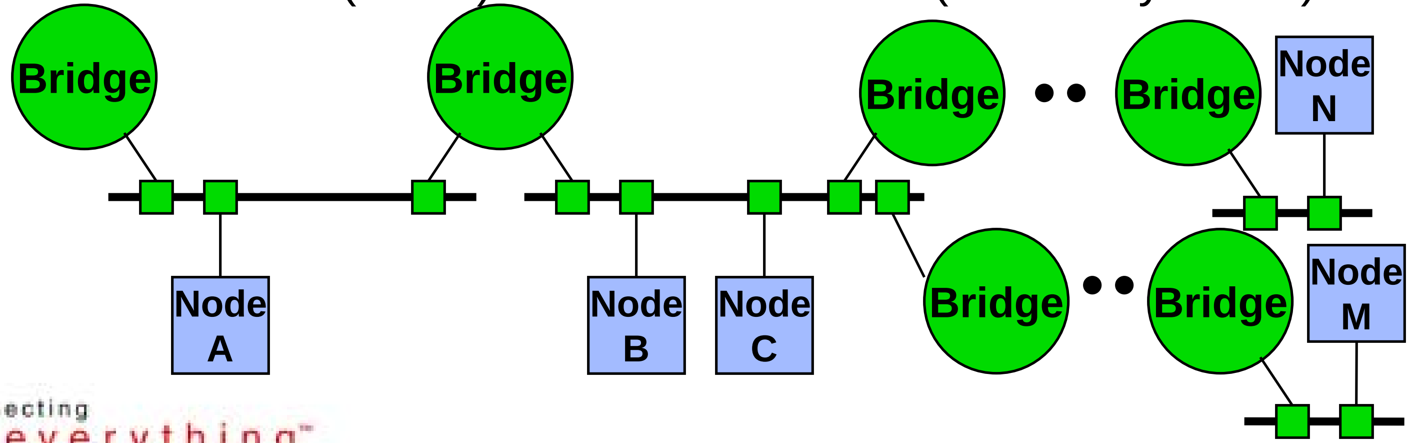
Yong Kim

May 15, 2001

802.1D Transparent Bridging

802.1D Transparent Bridging

- **Spanning Tree Architecture (Plug-&-Play)**
  - No Loops (ST algorithm makes sure of this)
  - Transparent Bridging (Layer 2 format translation) among 802 networks.
    - FYI. Considerable prior work in 802.1, 802.5, and ANSI (FDDI) for encapsulation bridge, with and without Source Routing option.
  - Filter and forward known unicast
    - Initial and Background MAC address learning for unicast forwarding.
  - Broadcast (Flood) unknown [to local bridge] unicast (within any VLAN)
  - Broadcast (Flood) multicast/broadcast (within any VLAN).



802.17 RPR Assumptions

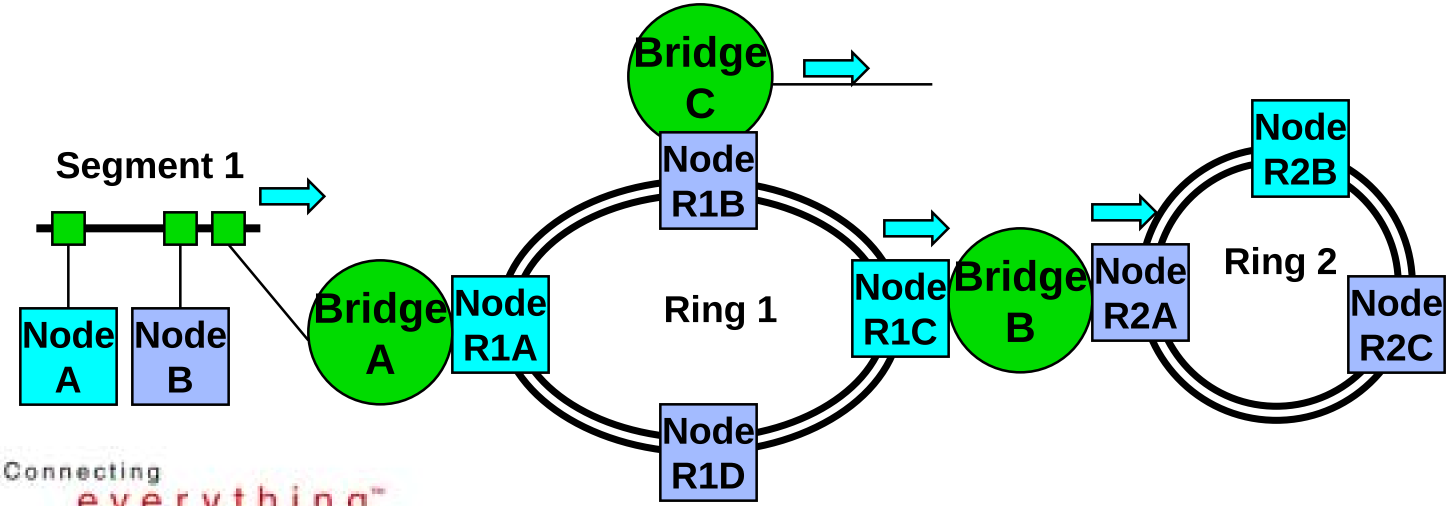
- **Layer 2 Shared Medium Access Control protocol with spatial re-use.**
  - For this presentation, I further assume no frame-loss on the ring.
- **Destination Strip of Unicast**
  - Note that all RPR proposals currently do not support unicast that needs to behave as broadcast – this behavior is assumed in transparent bridging.
- **Source Strip of Multicast/Broadcast**
- **Built-in protection to delete unicasts not claimed by any destination, or multicast where source is no longer available (e.g. TTL<= 1).**

802.17 LAN Segment Transparent Bridging

- **Let's just connect 802.17 to 802.1 Bridging.**
  - Node A (Seg 1) sends a unicast frame to Node R2B (Ring 2).
  - Bridge A broadcasts to all (R2B MAC address is unknown)
  - Node R1A sends the frame w/ destination MAC address unchanged (R2B).
  - Node R1C, either

    a) Ignores the frame because it is not the destination, or

    b) As a MAC attached to a T Bridge, it copies the frame but does not terminate the frame.
  - If a) then the system is broken. If b) then every unknown unicast becomes multicast for every rings in the system (not just the direct path) – no different than the current L2 bridging.



Transparent Bridging, Intermediate Conclusions

- **If b) is acceptable, that is Transparent Bridging is supported via unknown unicast flood (or broadcasting of unicast frames), then there are following hard requirements:**
  - Unknown unicast must be classified as multicast on the RPR LAN.
  - Source strip of unknown unicast
    - Must guarantee unknown unicast to strip after once around the ring.
  - Special Requirements of RPR port on a Bridge
    - The MAC address table size requirements for RPR port of the Bridge is in the order of current L2 Bridge (~8K or more). This really means that Bridge MAC address table and the RPR port MAC address table is integrated, unless you are doing encapsulation (e.g. tunneling) among bridges attached to RPR ring. More discussions on this subject later in this presentation.

- **With the above changes to the RPR requirements, RPR is compatible to 802.1D Transparent Bridging.**
  - No further compatibility issues w/ 802.1D bridging (learning, spanning tree, etc).

Consequences of T. Bridging

- **General Observation**
  - RPR, being shared access media with longer timer spanning tree timeout, does not support Rapid Spanning Tree.
  - Significantly larger L2 MAC address table required, or create nodeID mapping of multiple MAC addresses to nodeID, and then switch on nodeID.
  - Spatial Reuse benefit is somewhat negated due to unicast flood.

Other Potential Solutions

- **No L2 Bridging (may be interesting but out of scope)**
  - Router on every node
  - L2+ Tunneling
    - e.g. MPLS per draft-martini-l2circuit-encap-mpls-01.txt
- **Encapsulation Bridging**
  - Works well for RPR backbone, with Ethernet access.
  - Each Encapsulation Bridge w/ RPR port must decap, bridge, and encap if forwarding.
  - Could be done in 802.17 or in 802.1D, either as standard, or as Normative Annex (which means this is still a part of the standard) as 802.5 source routing got included.

Encapsulation Bridging Discussions

Encapsulation RPR Behavior

- There are no unknown MAC address (or node_ID, if you prefer) on a ring.
- There are only unicast (destination strip) and multicast (source strip) frame delivery mechanism.
- Explicit RPR_Bridge_Multicast MAC address definition (all Bridge node must copy for learning)
- If a node no longer exist, TTL mechanism drops the frame. (further issues w/ Bridge node)

Encap Bridge Node Behavior

- **All frames destined for local ring is sent without encapsulation (remember all MAC address on the ring is known).**

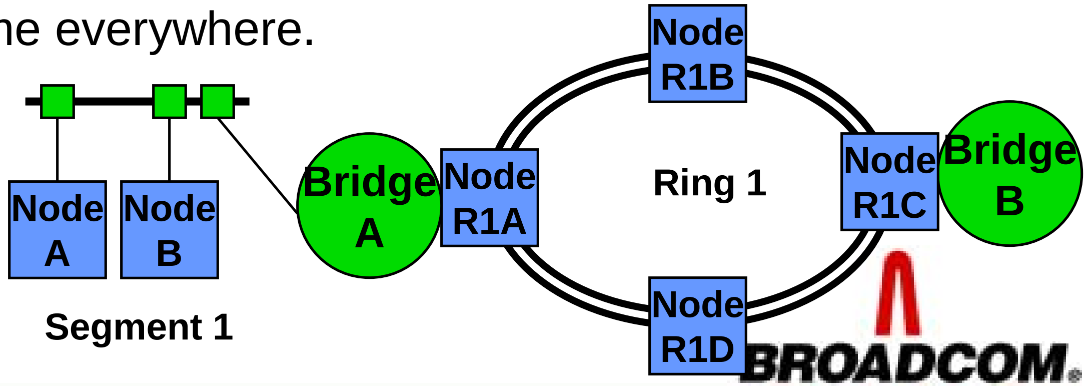
- **Unknown unicast flood (not local) is sent on well known RPR_Bridge_Multicast MAC address.**
  - All RPR Bridge node must copy (versus an option to copy) to perform address learning and forwarding.
  - Optionally BPDU could also use this mechanism.
  - No changes to Bridge behavior (spanning tree states, address aging, etc)

- **All multicast on the ring is stripped by the source.**
  - Protection event may replicate encapsulated RPR_Bridge_Multicast frame reception (For these, TTL may want to be exactly equal to # of nodes on the ring).

RPR Encapsulation Bridging

- **Assumptions of RPR behavior**
  - There are no unknown MAC address (or node_ID, if you prefer) on a ring.
  - If a node no longer exist, TTL mechanism drops the frame.
  - There are only unicast (destination strip) and multicast (source strip) frame delivery mechanism.
  - Explicit RPR_Bridge_Multicast MAC address definition.

- **Benefits**
  - Moves the extended LAN MAC address resolution to Bridge, and allows for RPR MAC to be the same everywhere.
  - RPR MAC L2 address table size remains small, ~ # of nodes on single ring.



Summary & Next Step

- **Transparent Bridging could be made to work.**
  - Requires unknown_unicast in addition to unicast and multicast.
  - Requires large MAC address table to support T. Bridging in RPR node.

- **Encapsulation Bridging could be made to work.**
  - The Burden of large MAC address table, etc,   is put in the Bridging nodes
  - Exact mechanism would have to be specified in .17 or .1.

Decide now, if we are ready.