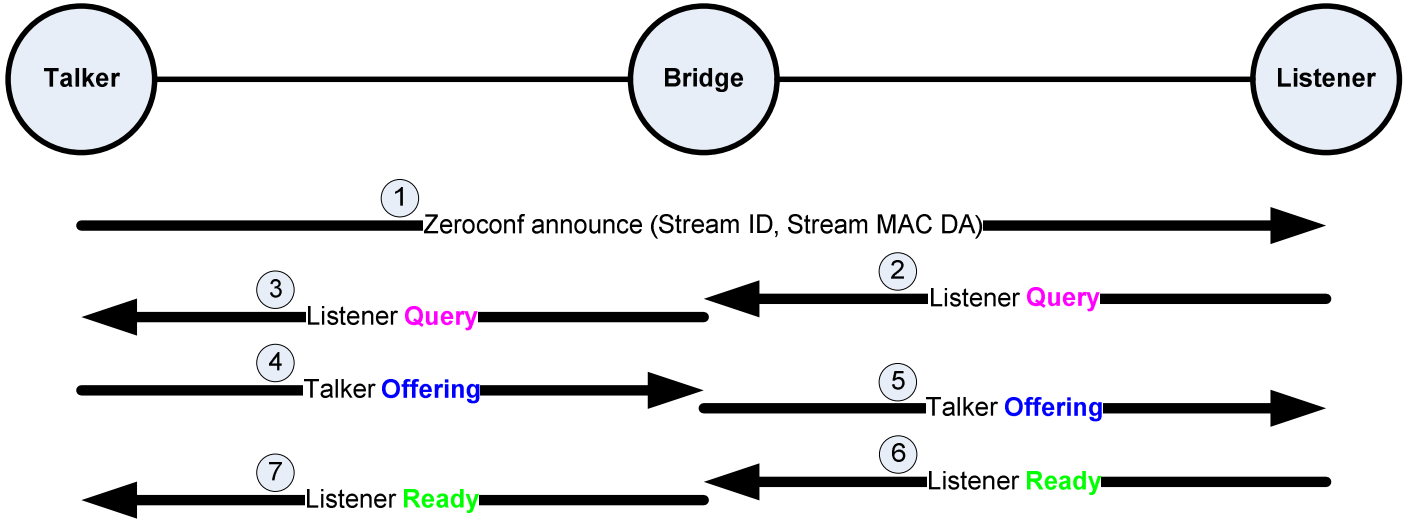
Three Step MSRP

Introducing the Query command

Craig Gunther (cgunther@harman.com)

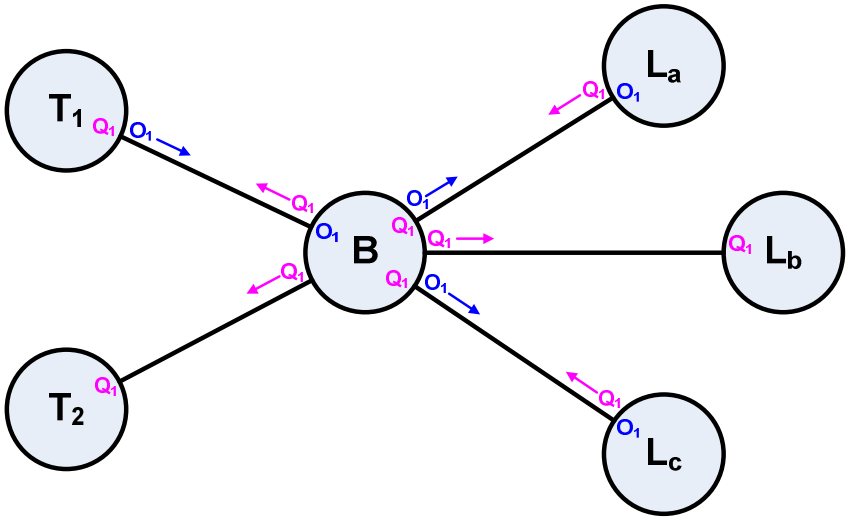
09 April 2008

Successful Stream join



- Talker advertises stream via higher layer protocol (e.g. Zeroconf)
- Listener issues MSRP **Q**uery
- Talker responds with MSRP **O**ffering
- Listener requests Stream with MSRP **R**eady

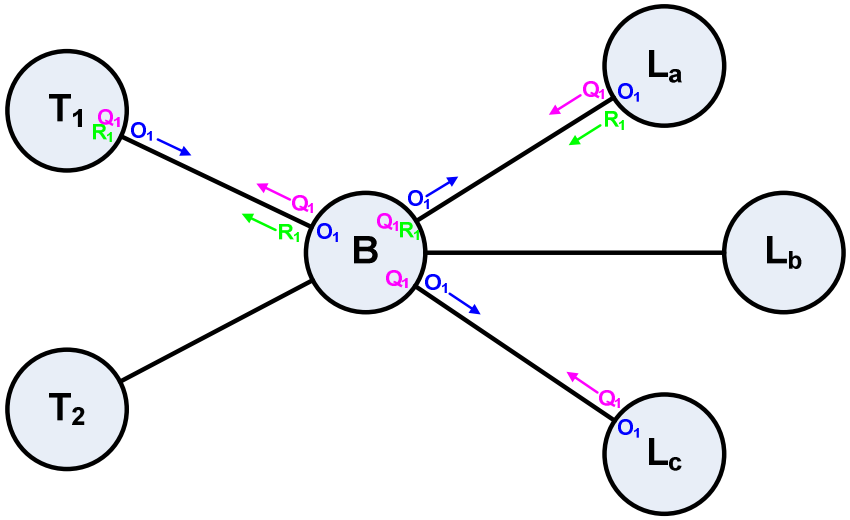
MSRP Query and Offering



Look more closely at the Query and Offering stages with multiple bridge ports:

- Assume Listeners L_a and L_c both issue a Query for the Stream provided by T_1
- Bridge **B** does not know where T_1 is located so it sends a Query on all other ports
- Talker T_1 receives the Query and responds with an Offering
- Talker T_2 ignores the Query because it cannot source that Stream
- Bridge **B** forwards the Offering out all ports that received (registered) the Query

MSRP Query, Offering and Ready



A little housekeeping in the Bridge, and then process the Ready:

- Once **B** has the Offering it can remove the Query from unrelated ports ($\mathbf{T_2}$ and $\mathbf{L_b}$)
- When $\mathbf{L_a}$ is ready to receive the Stream it sends a Ready
- **B** receives the Ready, configures its queues and sends the Ready to $\mathbf{T_1}$
- Talker $\mathbf{T_1}$ receives the Ready and can begin transmitting the Stream
- When $\mathbf{L_c}$ becomes ready to receive the Stream it will send a Ready and **B** will Immediately send the Stream down that port ($\mathbf{T_1}$ will not know there are two Listeners)

MSRP Query format

- What's does a Query look like?
  - **Option 1: Stream ID (64 bits)**
  - Option 2: Stream MAC DA (MMRP compatible 48 bits)
    - Might present some interesting problems when multiple Streams can be sent to a single unicast address.
      - How do Talkers know which Stream the Listener wants?
      - All Talkers would have to send info about all Streams going to the unicast address

MSRP Query format (continued)

- Allow a Query value for “all”?
  - Handy for a peeker/sniffer/analyzer
  - We could define these unique Stream IDs:
    - **`ff:ff:ff:ff:ff:ff.ffff`** means all Streams from all Talkers
    - **`ff:ff:ff:ff:ff:ff.0000`** means default Stream from all Talkers
    - **`xx:xx:xx:xx:xx:xx.ffff`** means all Streams from a specific Talker

New MSRP rules

- Talkers & Bridges must see a **Q**uery before sending an **O**ffering
  - This allows Talkers and Bridges to have a way to re-learn about interested Listeners after a Spanning Tree reconfiguration
  - Listeners can no longer passively discover Streams via **O**ffering gleaning, they must rely on a higher layer protocol
  - “29.1.2.2 Listeners” will no longer allow Listener Declarations before Talker Declarations
- Bridges only forward an **O**ffering to ports that have registered a **Q**uery
  - Reduces traffic between bridges
  - Reduces Bridge RAM requirements (TSpec storage)

New MSRP questions

- Should Talkers hear **O**fferings for a Stream they supply? What if two Talkers supply the same Stream and they want to monitor each other? If we need multi-Talker support someone needs to champion that effort (Annex Z, 2a).
- MRP LeaveAllTimer causes a Declaration/Registration refresh every 10-15 seconds. Will MSRPDUs contain full TSpec every time? There are roughly 64+ bytes per **O**ffering, 24+ bytes per **R**eady.