AVB HopStatus

Hop information for AVB Talkers and Listeners

Craig Gunther

(cgunther@harman.com)

October, 2007

Purpose

- Communicate hop status information between Talkers and Listeners
- Talkers learn of all attached Listeners and latency requirements
- Discover network topology
- Detect network changes (e.g. Spanning Tree reconfigurations)
- Identify point of reservation failure for higher-layer applications
- Useful for higher layer application stream path analysis (like poorly performing bridges, etc)

Overview

- Each Listener sends a periodic AVB HopStatus packet towards Talkers
- Each Talker sends a periodic AVB HopStatus packet towards Listeners
- Bridges update HopStatus summary information, append HopStatus detail, and then forward out appropriate port(s)
- Minimal impact on bridge resources

A potential approach*

*If you know of another existing solution, please share it!!!

HopStatus packet format

- Layer 2 addressing
- StaticInformation
  - Subtype
  - StartingHop
  - StreamID
- HopStatusSummary
  - PathHopCount
  - PathLatency
  - OffsetToNextHSD
- HopStatusDetail PDUs
  - TLV based variable length PDUs
  - Multiple HopStatusDetail PDUs per packet
  - Optional VendorSpecificInfo
  - ReservationStatusAndTechnology
  - EgressMACAddress
  - ResourceUtilization
  - Latency

StaticInformation

- Subtype
  - Allows for future expansion
- StartingHop
  - One-based hop count of first HopStatusDetail PDU in this packet
  - Packets created by the Talker or Listener will set this field to 1
  - If a bridge along the path needs to create a new packet the StartingHop will be set to the hop count of that bridge
- StreamID
  - Currently defined as the Talkers MAC Address plus a 16-bit unique Talker generated identifier

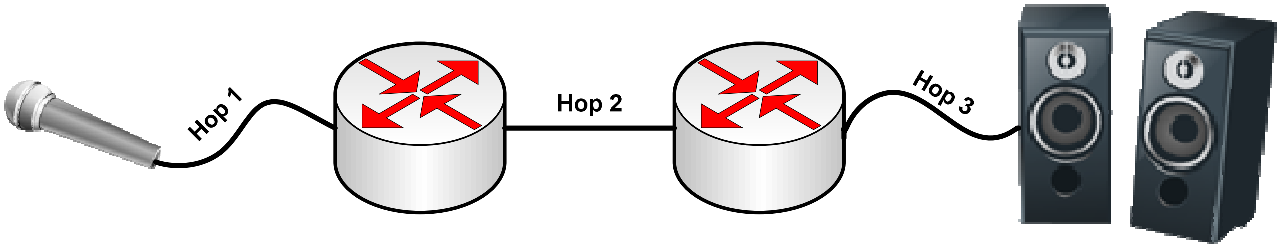
HopStatusSummary

- PathHopCount
  - Total number of hops between Talker and Listener; incremented by each hop
- PathLatency
  - Total latency (nanoseconds) between Talker and Listener; updated by each hop
- OffsetToNextHSD
  - Offset from end of HopStatusSummary to beginning of next available HopStatusDetail (initialized to 0).
  - Used by each hop as a pointer to available space in packet to append current hops HopStatusDetail
  - Updated by each hop to point to available space for next hops HopStatusDetail
  - Highest bit is a “PacketFull” flag.
    - If this flag is set the bridge will simply update HopStatusSummary information and forward the packet
    - Otherwise, if there is available space, the HopStatusDetail from this hop is appended to the packet and the OffsetToNextHSD is updated
    - If there is no available space for the HopStatusDetail the bridge will set the PacketFull flag and forward the packet. The packet will be copied into a new packet and the OffsetToNextHSD will be reset (which implies all HopStatusDetail information in the new packet is cleared). StartingHop will be set to PathHopCount. The HopStatusDetail and OffsetToNextHSD will be updated and the new packet will also be forwarded.

HopStatusDetail

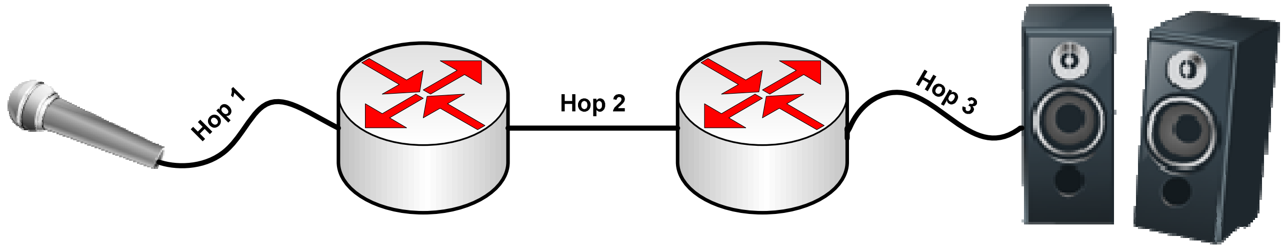
- TLV based for easy expansion
  - Allows for vendor specific info, etc.
- ReservationStatusAndTechnology
  - Succeeded, Failed, Upgrade Denied, Unknown
  - 100baseT, 1000baseT, MOCA
- EgressMACAddress
  - 48-bit MAC Address of stream egress port
- ResourceUtilization
  - % of remaining bandwidth available
- Latency
  - Maximum nanosecond latency through hop

Scenarios



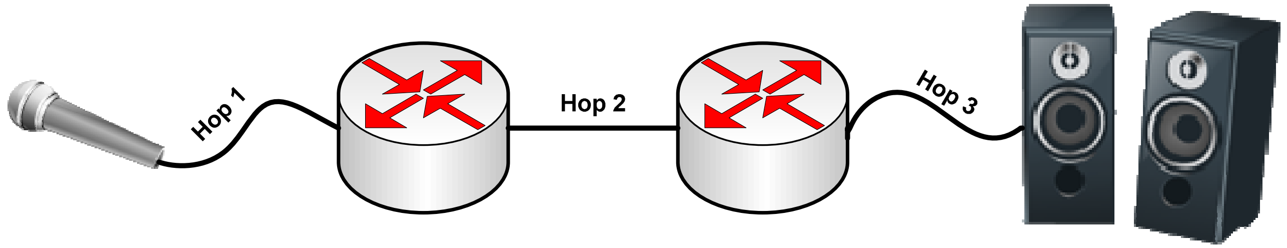
- Hop 1 is a 100BaseT connection
  - Maximum packet latency is 250usec
- Hop 2 is a 1000BaseT connection
  - Maximum packet latency is 25usec
- Hop 3 is a 100BaseT connection
  - Maximum packet latency is 250usec

Scenario: Simple 3 Hop



- StaticInformation
  - StartingHop = 1
- HopStatusSummary
  - PathHopCount = 3
  - PathLatency = 525us
  - PacketFull = 0
- HopStatusDetail (hop #1)
  - TLV:ReservationStatusAndTechnology = SUCCEEDED, 100BaseT
  - TLV:Latency = 250us
  - TLV:End
- HopStatusDetail (hop #2)
  - TLV:ReservationStatusAndTechnology = SUCCEEDED, 1000BaseT
  - TLV:Latency = 25us
  - TLV:End
- HopStatusDetail (hop #3)
  - TLV:ReservationStatusAndTechnology = SUCCEEDED, 100BaseT
  - TLV:Latency = 250us
  - TLV:End

Scenario: Full 3 Hop



For this example assume the HopStatus information requires two packets. When the second bridge attempts to append its information it finds there is not enough available space in Packet #1 so it flags that packet as full and generates Packet #2. Note that the HopStatusSummary HopCount and Latency are the same for each packet.

Packet #1

- StaticInformation
  - StartingHop = 1
- HopStatusSummary
  - PathHopCount = 3
  - PathLatency = 525us
  - PacketFull = 1
- HopStatusDetail (hop #1)
  - TLV:ReservationStatusAndTechnology = SUCCEEDED, 100BaseT
  - TLV:Latency = 250us
  - TLV:End
- HopStatusDetail (hop #2)
  - TLV:ReservationStatusAndTechnology = SUCCEEDED, 1000BaseT
  - TLV:Latency = 25us
  - TLV:End

Packet #2

- StaticInformation
  - StartingHop = 3
- HopStatusSummary
  - PathHopCount = 3
  - PathLatency = 525us
  - PacketFull = 0
- HopStatusDetail (hop #3)
  - TLV:ReservationStatusAndTechnology = SUCCEEDED, 100BaseT
  - TLV:Latency = 250us
  - TLV:End