# IEEE 802.1Qbg
# Proposal to add Channel Type IDs

bg-kamath-channel-typeid-0810-v2

Daya Kamath (BNT)
Jay Kidambi (BNT)
Vijoy Pandey (BNT)
Jeffrey Lynch (IBM)
Rakesh Sharma (IBM)

Aug 2010

# Overview

- Use cases for channel type identifiers
- Limitations of current mechanisms
- Suggested protocol enhancements
- Related concerns and counter arguments
- Modifications to EDCP in 802.1Qbg draft 1.0

# Use cases

- A channel represents an uplink similar to a physical port in many aspects, and should have the capacity to support port related **features** such as QoS, rate limiting, default vid settings etc.

- A bridge may wish to **customize** the properties of each channel on a port differently. For example, support ECP on one channel, and disable it on another.

- A bridge may need to **pre-configure** characteristics of channels, such as enabling flow monitoring features on a channel

- A bridge needs to **identify** and **track** channels consistently in order to support desired feature-sets on the channels

- A bridge may wish to control the rate of **broadcast** and **multicast** traffic on the channel. VDP related QoS parameters apply to specific VSIs, whereas a bridge may wish to apply **shared** properties for all VSIs on a given channel, such as bandwidth limits

# Use Cases

- A channel may  have a VM directly attached, or may be partitioned for a specific workload, needing customized network control policies
- A bridge may need to limit the number of VEBs or VEPAs connected to channels across all the ports, due to resource restrictions. It may want to present different default settings for a subset of connections in order to achieve this
- A channel can be used  by bridges to implement and fine-tune a variety of features like
  - Failover support between channels across different ports
  - Static or dynamic trunk pre-configuration
  - Default vid settings
  - Flow control
  - Netflow/sflow
- Bridges can conserve resources if they have a mechanism to apply aggregate common policies on a channel basis, rather than repeating them for each VSI

# Limitations of current scheme

- A station requests channels by identifier only. No persistent binding guarantees between a channel and its ‘user’ on the station side
- Even if the binding is maintained within the station, the requested ordering of channels in CDCP tlv may change depending upon station internal operational sequence, or parameters, or configuration
  - Bridge has no mechanism available to identify a channel consistently
  - Bridge cannot pre-configure any customized (non-default) parameters on the channel
- Assuming the network admin configures parameters manually after the channel has been set up, these cannot be persistent across bridge or station resets

# Protocol Enhancement

- NOTE - The proposal provides a tool to characterize and customize channels. The actual properties to be applied, enforcement points, and level of enforcement on bridges as well as stations can be vendor-specific customizations
- Mechanism is to add a type identifier and version field to EDCP tlv, similar to the VSI mgr ID, VSI type ID and VSI version ID in the VDP tlv
- NOTE - Channel is link-local, no moves across l2 domain, so a channel ID field is not needed. Station and bridge can use the local channel ID fields exchanged in the CDCP tlv to associate a given channel with its type ID internally.
- Allows bridges and stations to converge on channel type definition using out-of-band mechanisms outside of the scope of the specification (similar to VSI)
- Stations indicate the type of channel being created in the request, and the properties for that channel type can be determined **or** retrieved by bridges
- Typical mode of configuration could be static on bridges, if the number of channels is limited, or the same mechanism as is used for VSI types, or not

# Protocol concerns and counter-arguments

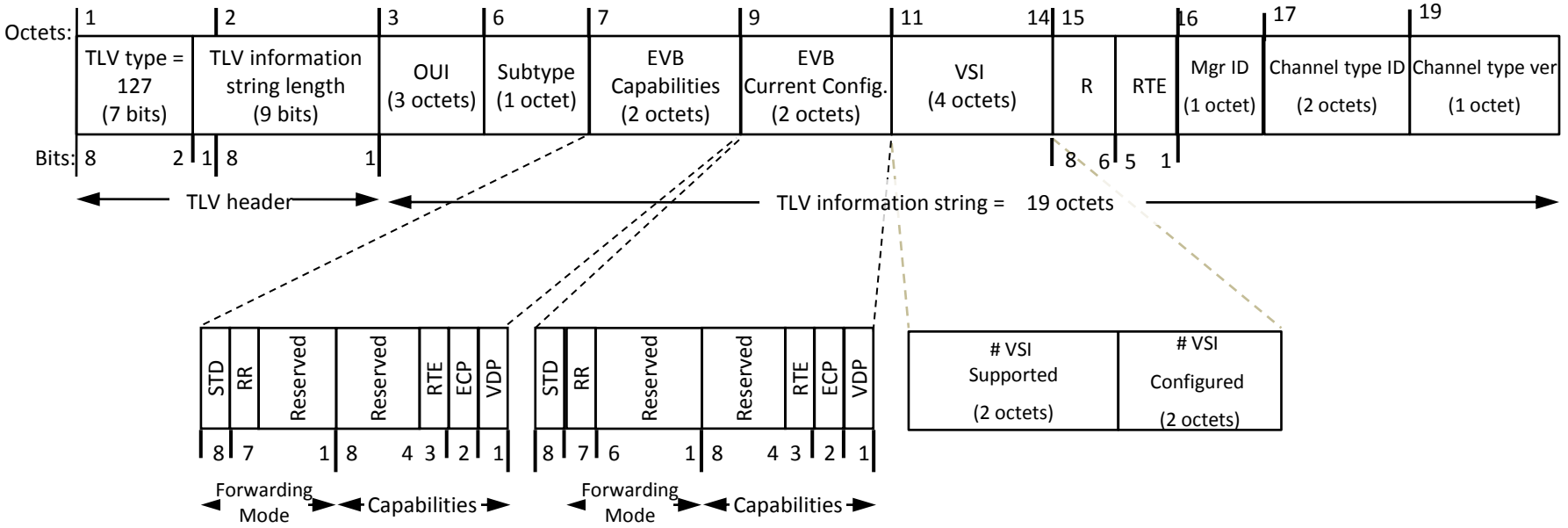
- The bandwidth policies defined for a channel may conflict with the policies defined for the port by another mechanism like ETS/DCBX

This is a shared concern that needs to be handled for VSI policies as well, irrespective of whether channel policies are defined. Any tie break mechanism defined can be applicable for both

- Station and bridge may have conflicts on channel policy request and application

This proposal is meant to aid bridge side application of policies only. A network administrator can choose to maintain a global channel policy database, so it is shared between the station and bridge, or keep it local on the bridge. The station can merely send a persistent type id for the channel each time it is created, to aid the bridge with its implementation. A default value can be used if the station is not able to provide a persistent binding

# Proposed EDCP TLV




- Use reserved values for channel type ID, version ID and manager ID (all zeroes or all ones) to indicate the absence of a specific type classifier
- The Bridge and station can then apply administrative defaults configured locally