

Gen-Z Identifiers

April 2019

This presentation covers Gen-Z Identifiers. Gen-Z supports multiple types of identifiers. These are used to identify components, multicast groups, supported functionality, etc.

Disclaimer

This document is provided 'as is' with no warranties whatsoever, including any warranty of merchantability, noninfringement, fitness for any particular purpose, or any warranty otherwise arising out of any proposal, specification, or sample. Gen-Z Consortium disclaims all liability for infringement of proprietary rights, relating to use of information in this document. No license, express or implied, by estoppel or otherwise, to any intellectual property rights is granted herein.

Gen-Z is a trademark or registered trademark of the Gen-Z Consortium.

All other product names are trademarks, registered trademarks, or servicemarks of their respective owners.

All material is subject to change at any time at the discretion of the Gen-Z Consortium

<http://genzconsortium.org/>

Component Identifiers (CID)

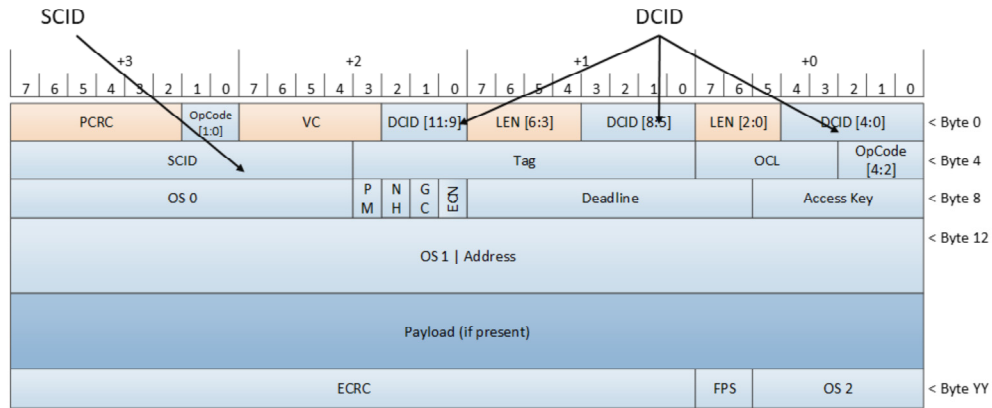
- A Component Identifier (CID) is assigned to uniquely identify a component within a subnet
 - 12-bit, subnet-local identifier
 - Each component is assigned one or more CIDs
 - Within an end-to-end packet,
 - A source CID (SCID) identifies the component that initially transmitted the packet
 - A destination CID (DCID) identifies the destination component
- CIDs are not used in P2P 64 or P2P-Vendor-defined OpClass packets

A Component Identifier (CID) is used to uniquely identify a component within a subnet.

- Each CID is a 12-bit value, enabling each subnet to support up to 4096 components.
- A component can support multiple CIDs. Multiple CIDs should be used only to support additional Tag spaces. Each additional Tag space enables a component to support an additional 4096 outstanding request packets to *each* peer component.
- Each explicit OpClass packets contains a source CID field and a destination CID field. The DCID is used to relay packets within switch components, and the DCID and SCID are used to validate packets.

CIDs are not used in components that support only point-to-point optimized OpClass packets. The SCID and DCID fields are not present in these packet formats.

Example OpClass Packet with SCID / DCID



© Copyright 2016 by Gen-Z. All rights reserved.

GENZ

This slide illustrates an example explicit OpClass packet. The DCID is located in the first 19 bits of the packet to optimize packet relay, validation, and processing, and the SCID is located in first 64 bits to optimize packet validation and processing.

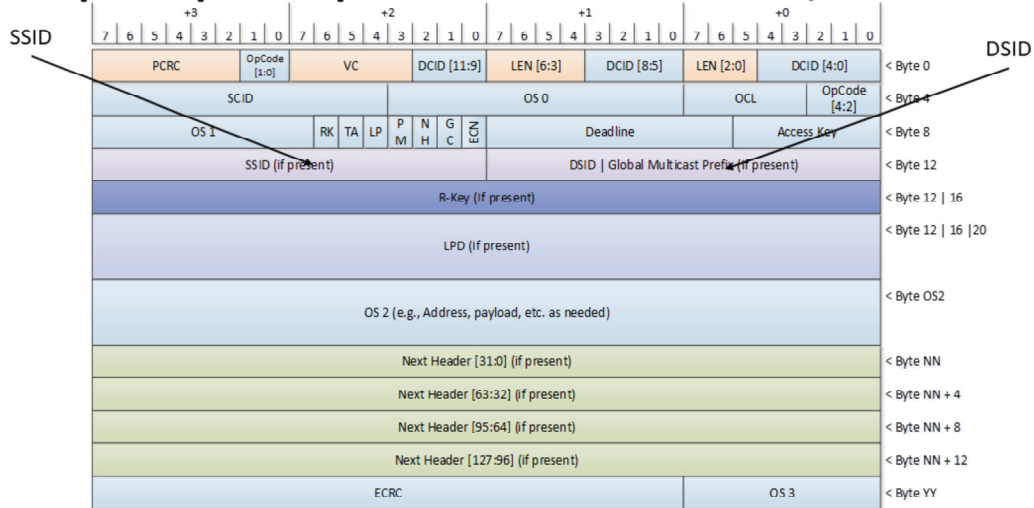
Multi-Subnet Identifiers

- Gen-Z supports 16-bit Subnet Identifiers (SID)
- SIDs enable large-scale switch topologies
 - Only applicable to components that explicitly support this capability
 - Has no impact on subnet-local switch packet relay
- Within an end-to-end packet,
 - A source SID (SSID) identifies the subnet where the packet was initially transmitted
 - A destination SID (DSID) identifies the subnet where the destination component is located
 - [SCID, SSID] identifies the component that initially transmitted the packet
 - [DCIS, DSID] identifies the destination component

Components that explicitly operate across multiple subnets use subnet identifiers (SID) to identify the source and destination subnets.

- A SID is a 16-bit identifier corresponding to a source or destination subnet. This enables Gen-Z to support very large switch topologies.
- Switches that support multi-subnet packet relay use the destination SID (DSID) to determine the egress interface to relay a packet.
 - A SID has no impact on subnet-local packet relay (that is performed using only the packet's DCID).
- A SID is combined with the CID to create a global CID (GCID).
 - Source and destination components are identified by their respective source GCID or destination GCID.

Example Explicit OpClass Packet with SSID / DSID



- SCID and DCID fields are not impacted by the inclusion of the SSID and DSID fields

The GC field indicates whether the SSID and DSID / Global Multicast Prefix fields are present within an explicit OpClass packets. All explicit OpClass packets contain the GC field.

If the OCL == Multicast, then instead of a DSID, these byte locations are treated as a Global Multicast Prefix. A Global Multicast Prefix is combined with a MGID to create a global multicast identifier.

UUID—Universally Unique Identifier

- UUID is a 128-bit identifier as specified in ITU-T X.667
- UUID is the preferred modern mechanism to identify attributes
 - Eliminates the need for company-specific allocation authority
 - Simplifies multi-vendor coordination (easy to allocate and share)
 - Simplifies standardization / de-facto standardization—allocate and publish UUID
- Gen-Z makes extensive use of UUIDs
 - Architecture specifies a single UUID to indicate a component supports Gen-Z
 - Used by management to unambiguously identify a Gen-Z component without maintaining “white lists”
 - Architecture uses UUIDs for multiple purposes, e.g.,:
 - Identify memory media types
 - Identify component types
 - Identify supported services (e.g., used by OS to bind software to hardware)
 - Identify vendor-defined operations and vendor-defined features
 - Identify various managers
 - Identify a FRU
 - Identify services
 - Etc.

© Copyright 2016 by Gen-Z. All rights reserved.

GEN Z

Gen-Z makes extensive use of UUIDs. UUIDs provide multiple advantages including:

- No centralized authority to generate
- Easily and dynamically created using any number of Internet-based services
- UUIDs can be shared to enable agile solution development, interoperability, etc. between hardware and software providers, multiple vendors, etc.
- UUID space is large, and they can be associated with any functionality or technology, e.g., identifying unique media types or media generations
- Etc.

UUID and Vendor-Defined Operations

- OpCode Set structure contains a set of optional UUID
 - Single UUID is associated with vendor-defined P2P Vendor-defined, P2P 64 Vendor-defined, and Multicast
 - Up to 8 UUIDs that are 1:1 associated with Vendor-defined OpClass OpCode Sets [1-8]
 - Management configures a OpClass Label (OCL) that is associated with each UUID
 - OCL enables Gen-Z to transport vendor-defined operations without deep packet inspection
 - Enables any number of vendor-defined OpClasses to simultaneously operate across the fabric
 - In theory, each pair of peer components can support a unique set of vendor-defined operations
 - Configuration indirection enables multiple components to support the same UUID without coordinating the UUID slot
 - Large UUID space and extremely light-weight allocation enables:
 - Vendors to experiment and innovate without fear of running out of identifiers or standardization overheads
 - Increases hardware / software agility
 - Vendors to publish / share UUID among peers or the industry
 - For example, accelerators might use custom Gen-Z operations to improve performance, reduce communication overheads, reduce data movement, etc.

Vendor-defined OpClasses are defined within the OpCode Set structure. To enable interoperability, management dynamically configures an OpClass Label (OCL) value for each supported Vendor-defined OpClass. This eliminates the need to a priori coordinate entries. Software simply configures a given entry with a common OCL.

UUIDs are used to uniquely identify vendor-defined operations. Each Gen-Z can support up to 8 Vendor-defined OpClasses. Though a component can support only 8 Vendor-defined OpClasses at a given time, the UUID is sufficiently large, that there is virtually no practical limit to the actual number of Vendor-defined OpClasses that can be supported within the industry.

Thank you

This concludes this presentation. Thank you.