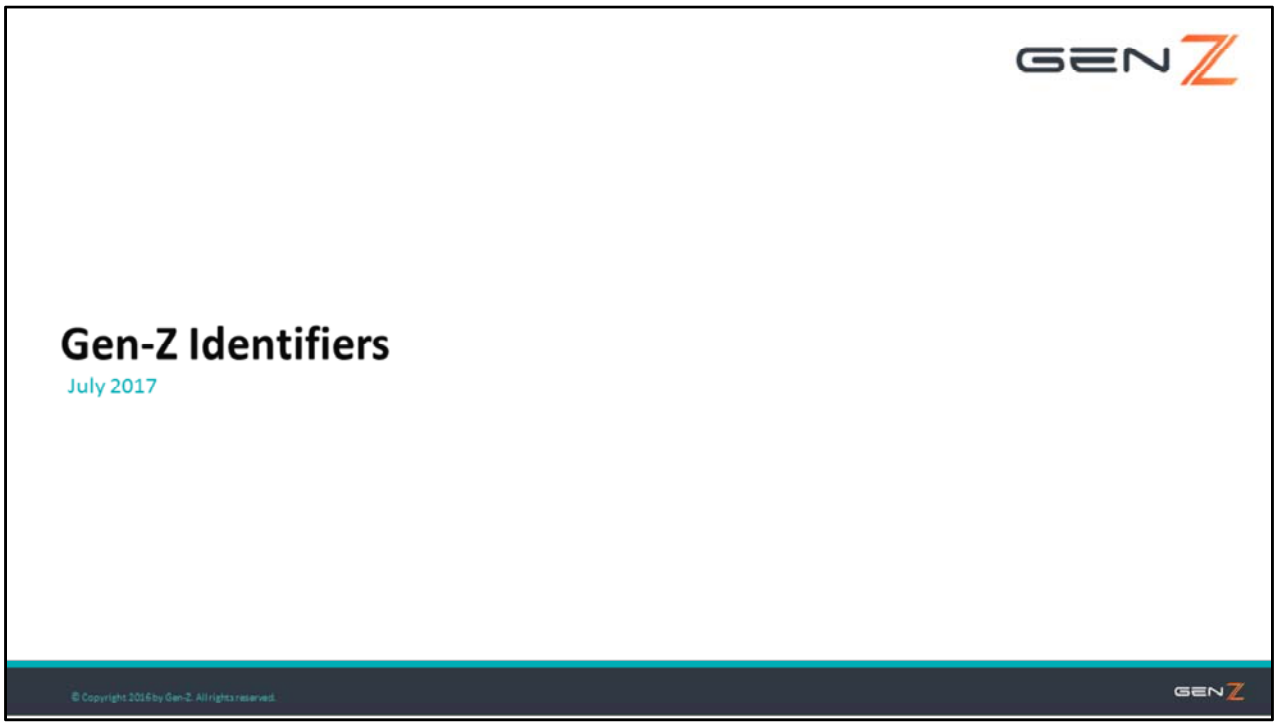


This presentation covers Gen-Z Identifiers. Gen-Z supports multiple types of identifiers. These are used to identify components, multicast groups, supported functionality, etc.

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.

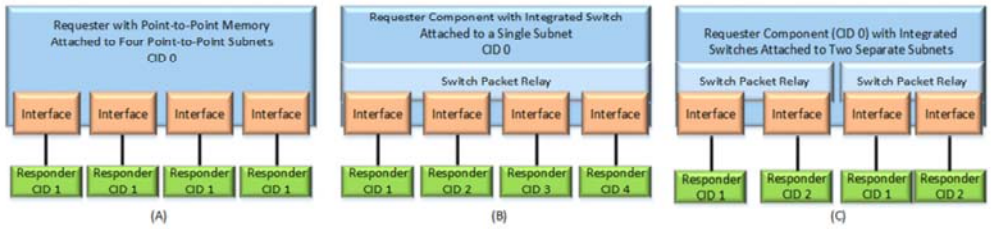
Gen-Z is a trademark or registered trademark of the Gen-Z Consortium.

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-Core, P2P-Coherency, or P2P-Vendor-defined OpClass packets

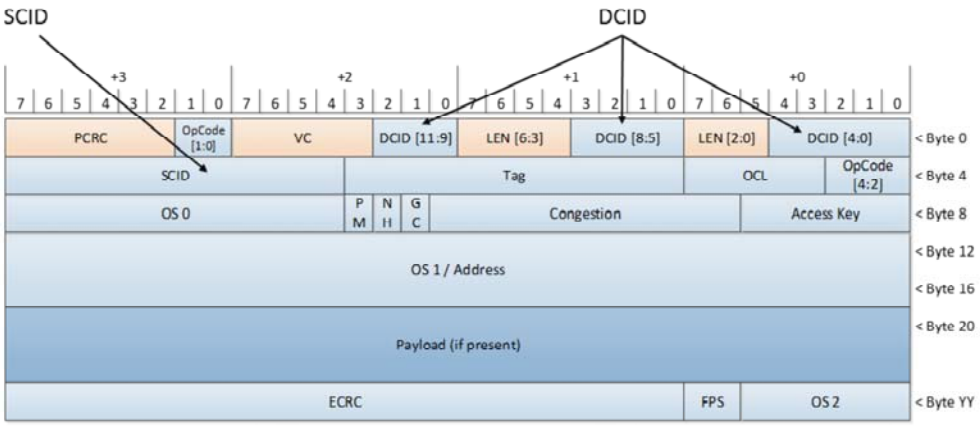
GEN Z

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

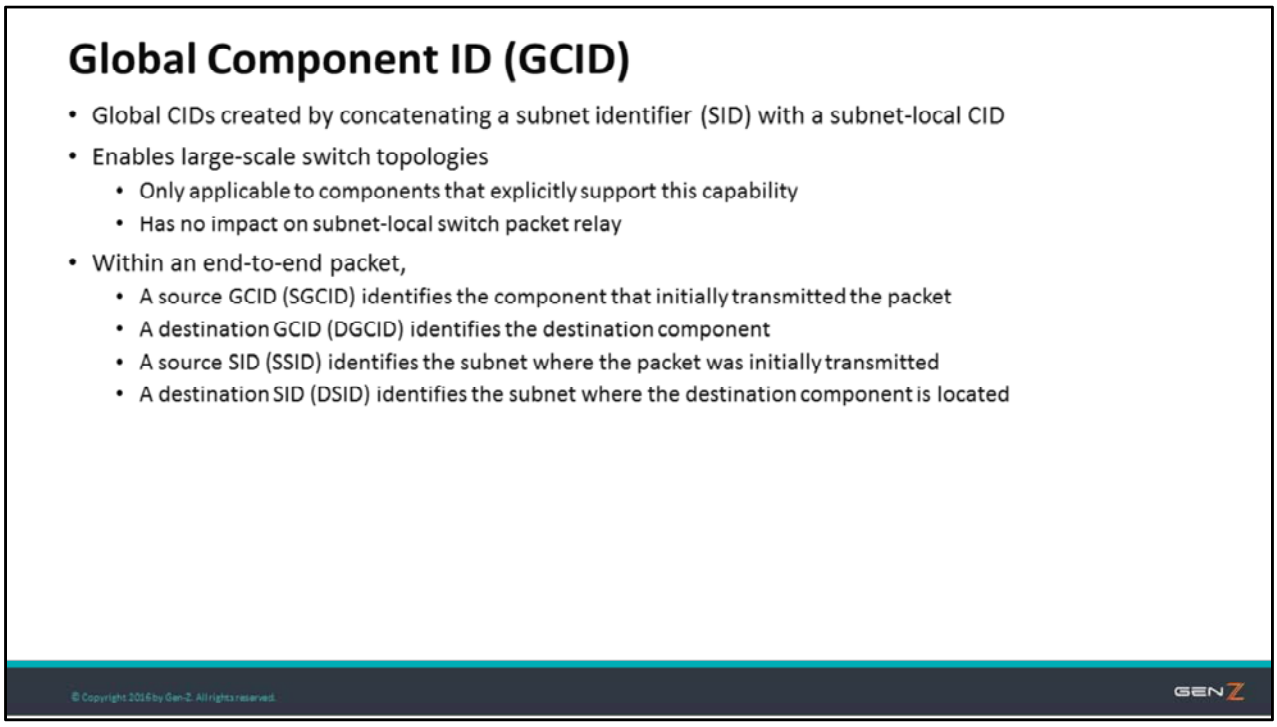
- Each CID is a 12-bit value, enabling each subnet to support up to 4096 components.
- A component can support multiple CIDs. Multiple CIDs should be used only to support additional Tag spaces. Each additional Tag space enables a component to support an additional 4096 outstanding request packets to ***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



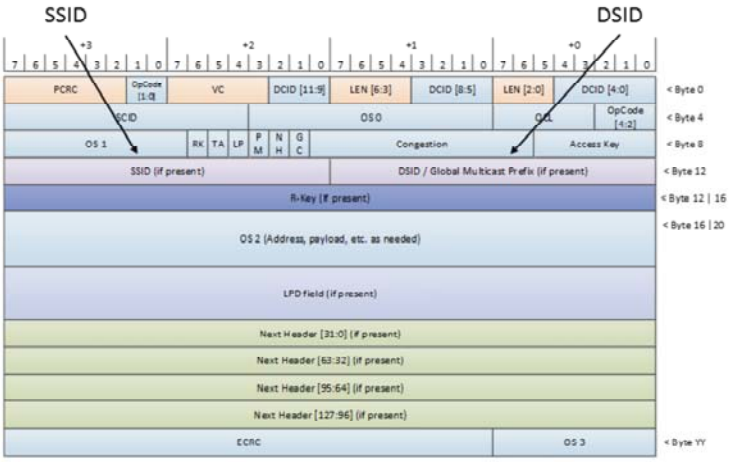
This slide illustrates an example explicit OpClass packet. The DCID is located in the first 19 bits of the packet to optimize packet relay, validation, and processing, and the SCID is located in first 64 bits to optimize packet validation and processing.



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

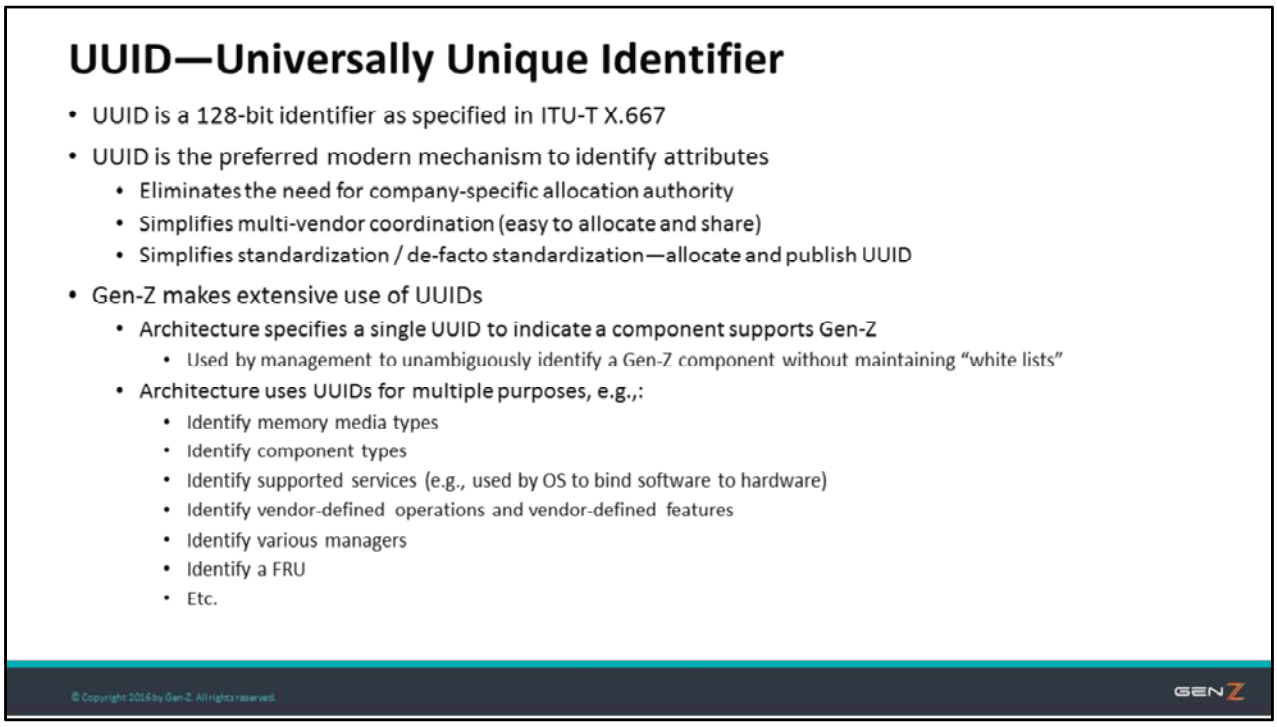


- SCID and DCID fields are not impacted by the inclusion of the SSID and DSID fields

GEN Z

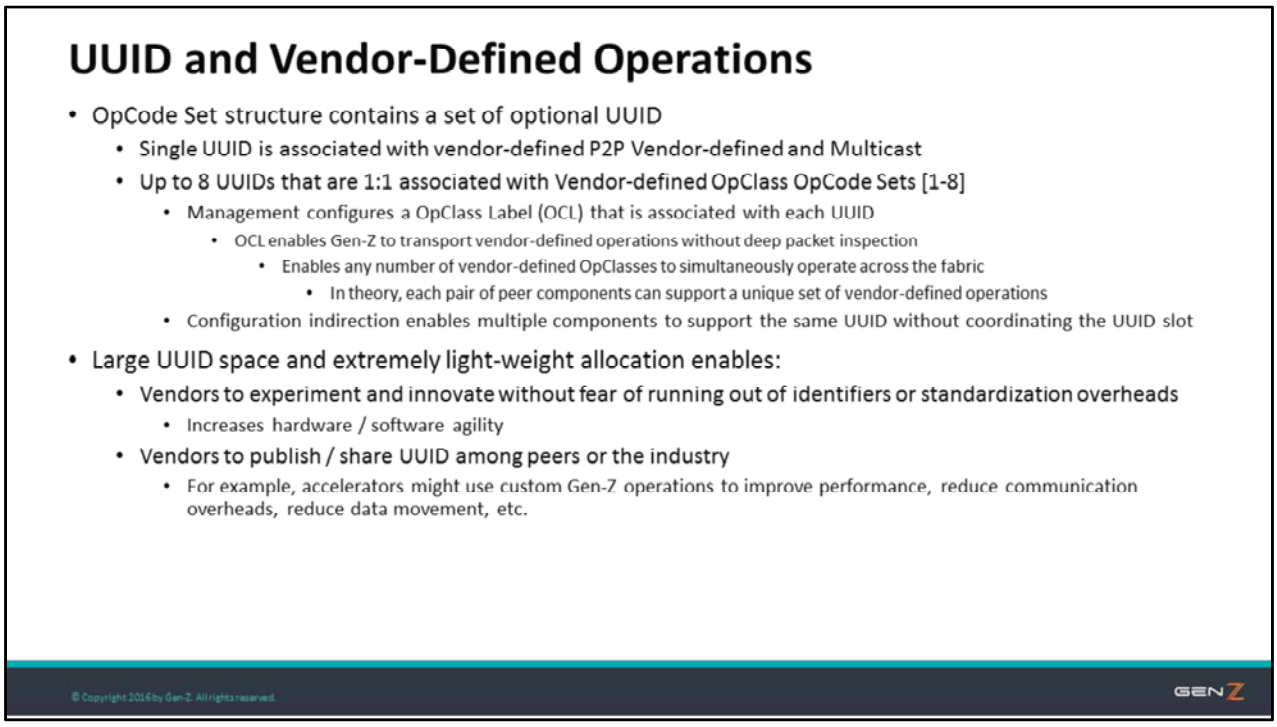
The GC field indicates whether the SSID and DSID / Global Multicast Prefix fields are present within an explicit OpClass packets. All explicit OpClass packets contain the GC field.

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.



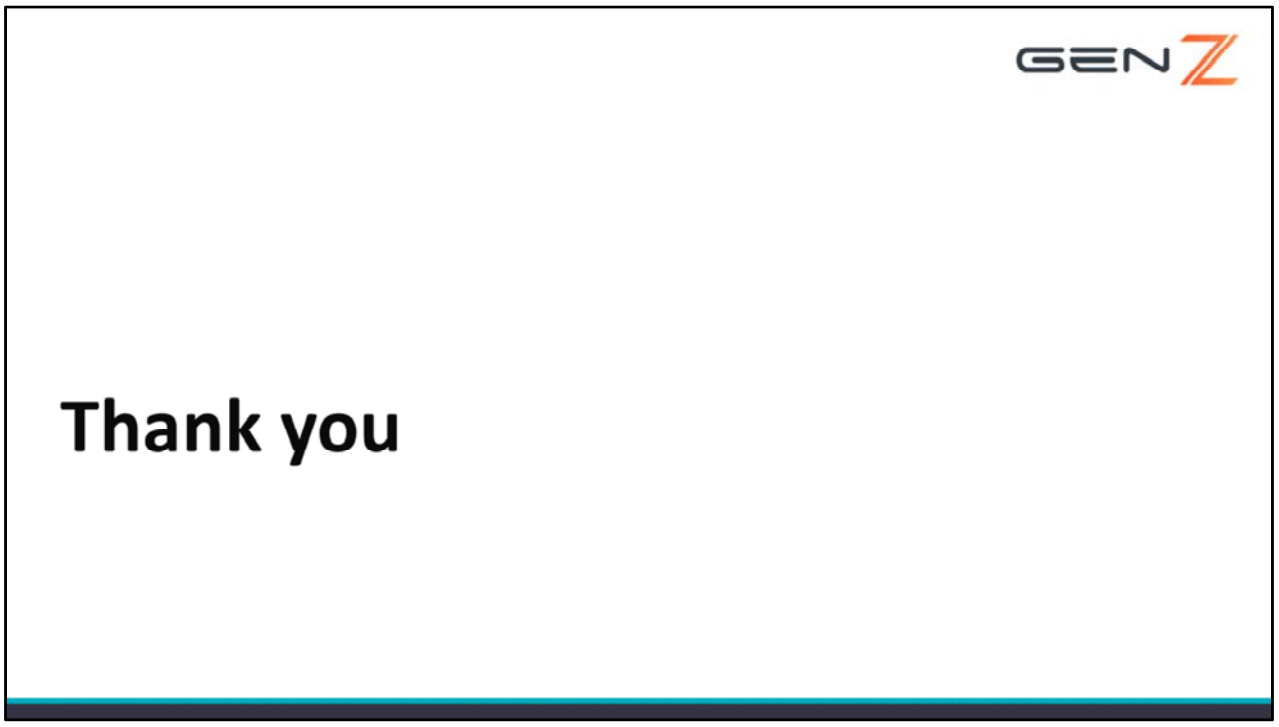
Gen-Z makes extensive use of UUIDs. UUIDs provide multiple advantages including:

- 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 associated with any functionality or technology, e.g., identifying unique media types or media generations
- 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.



This concludes this presentation. Thank you.