Gen-Z Unsolicited Events and Acknowledgments

July 2017

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/

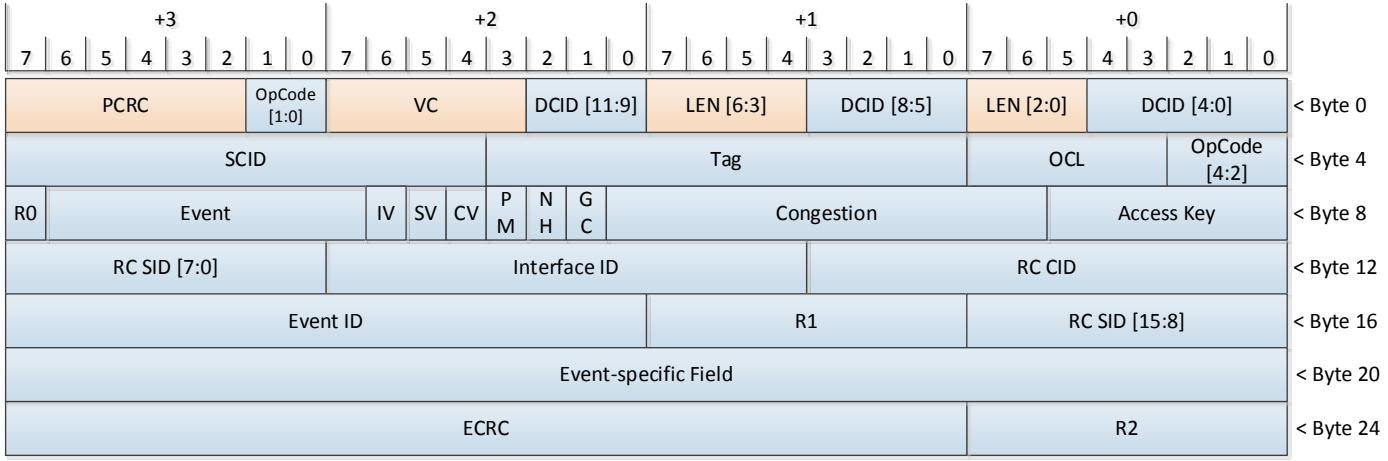
Unsolicited Events (UE)

- UE packets used to notify designated management entity detected events
  - Requires in-band management support
- Events are generated for various reasons:
  - Component failure detection
  - New component discovery
  - Access Violations
  - Excessive Responder-Not-Ready (RNR) events
  - Thermal Shutdown
  - Power events—power state change, faults, emergency shutdown, etc.
  - Attention Button Pressed
  - Dynamic Component Insertion / Removal
  - Component Containment
  - Fatal Media Error Containment
  - Etc.

Event Precedence

- Events are organized into precedence levels with multiple events per level:
  - Critical—these events include failure conditions, containment events, etc.
  - Serious—these events include unrecoverable errors, access permission violations, excessive retry events, malicious behavior detected, etc.
  - Recoverable—these events include recoverable errors, etc.
  - Resource—these events include new resource (e.g., a new component) discovery, resource service events, low-power entry, low-power exit, etc.
- Architecture supports up to 256 unique event types
  - 0x0-0xEF are specified by the architecture
  - 0xF0-0xFF are vendor-defined

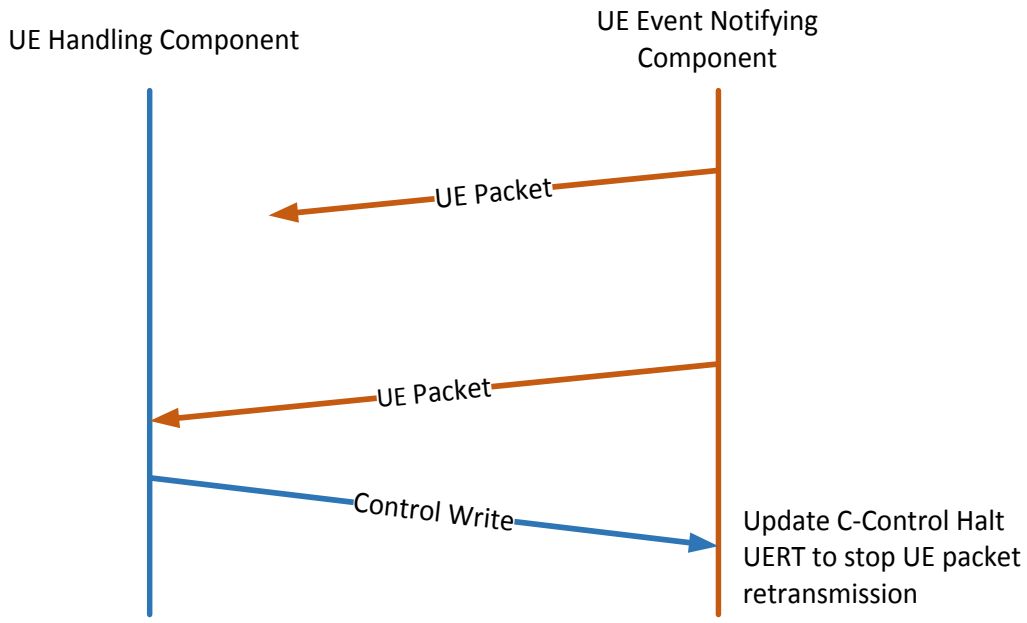
UE Packet



- UE-specific protocol fields
  - Event—indicates the event number
  - Event ID—unique identifier associated with this UE packet. Value is monotonically incremented modulo 2^{16}
  - Interface ID—if an event is associated with a specific interface, then the identifier assigned to the interface is copied to this field
  - RC CID / RC SID—CID and SID (if configured) of the source component that caused the event
    - For example, if a protocol event was detected, these contain the packet's SCID / SSID
  - Event-specific field—provides additional information about the event
- UE packets are not acknowledged at the protocol level
  - Component periodically transmits the UE packet until management clears the event notification
  - At most, one outstanding UE packet
  - Components required to provision resources to track:
    - At least one event per precedence level
    - At least one event per component interface

UE Packets

- UE packets are not acknowledged at the protocol level
- Upon scheduling a UE packet, the component:
  - Starts a UERT timer—minimum time between retransmission
  - Upon the Halt UERT being updated, the component halts the UERT
    - A Control Write request is used to modify the Halt UERT field
  - If the UERT timeout expires, the component retransmits the UE packet
    - Component should retransmit on alternative paths (if available)
    - Maximum of 32 retransmissions before targeting an alternative management component
    - If no management component ever responds, then the component stops UE packet retransmission and clears the event tracking logic.



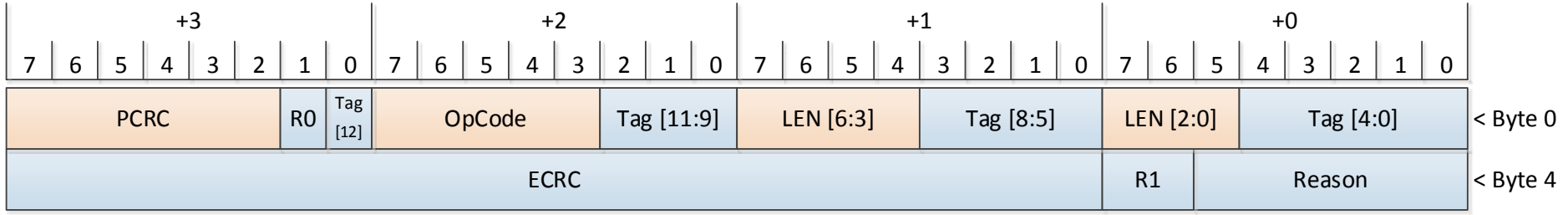
UE Configuration

- Component Error and Signal Event structure
  - If supported and configured, the Error CID and Error SID fields are configured with the CID and SID of the component responsible for error handling
    - This may be dedicated error and event handling manager
    - This may be the Primary Manager or the Primary / Secondary Fabric Manager
  - A subset of errors may be configured to cause UE packets to be transmitted
- Component Mechanical Structure
  - If supported and configured, then the Mechanical CID and Mechanical SID are configured with the CID and SID of the component responsible for mechanical management
    - This may be dedicated mechanical manager
    - This may be the Primary Manager or the Primary / Secondary Fabric Manager
- Events not handled by either of these managers are handled by the Primary Manager or the Primary / Secondary Fabric Manager

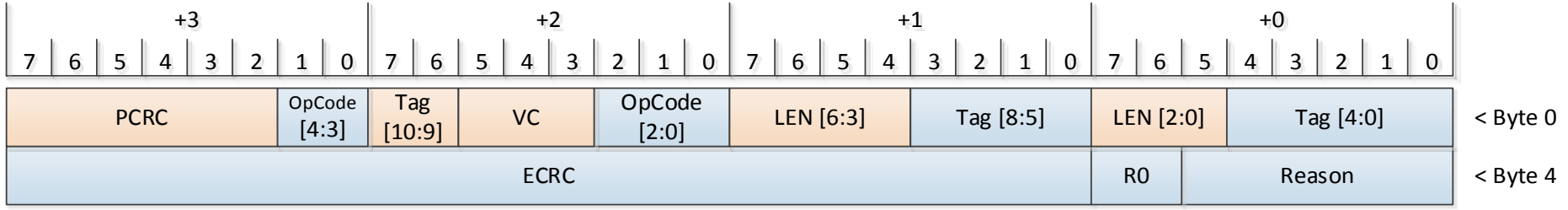
Standalone Acknowledgments

- Standalone Acknowledgment packets are used to positively or negatively acknowledge a request packet
  - A positive acknowledgment indicates the request was successfully validated and executed
  - A negative acknowledgment indicates that an issue was detected (validation or execution)
- All components are required to support Standalone Acknowledgment packets
- All packet formats contain:
  - A Tag field used to correlate the response with the request
  - A Reason field used to indicate if any issues were detected with the request
    - A Reason field and subset of the encodings are used in other response packet formats
    - Reasons are classified as:
      - No Error
      - Transient Error (e.g., SOD out-of-order packet receipt)
      - Non-transient Error (e.g., protocol error)
      - Transient Condition (e.g., RNR NAK)
      - Non-transient Condition (e.g., poison data, unsupported service, etc.)

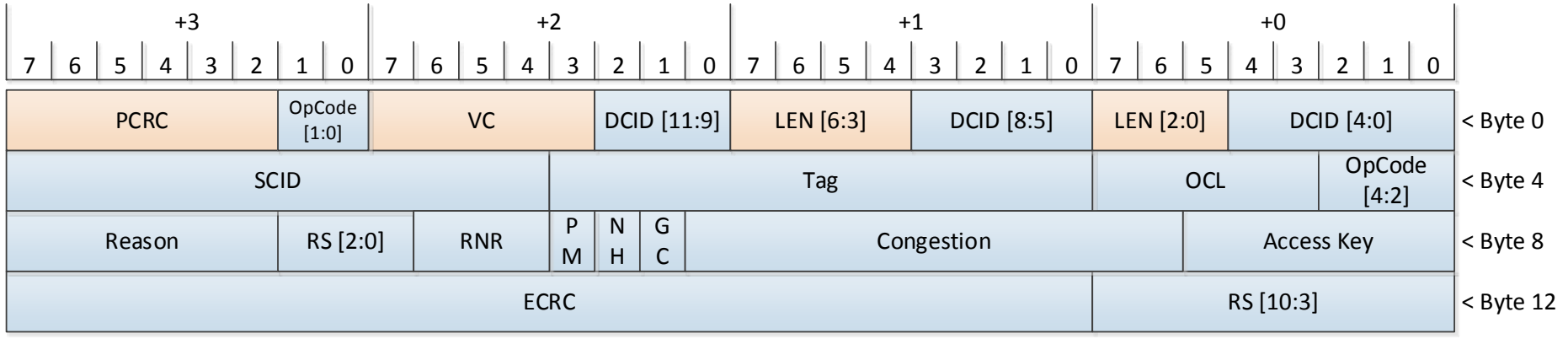
Standalone Acknowledgment Packet Formats



P2P-Core Acknowledgment



P2P-Coherency Acknowledgment



Core 64 Acknowledgment

RNR NAK

- Responder-Not-Ready Negative Acknowledgment
  - May be returned for any supported request packet
- RNR NAK represents a transient operating condition, e.g., a resource shortage
- Upon receipt of a RNR NAK, if the Requester intends to retransmit the request, it waits the indicated RNR time interval
  - Time interval is represented as an encoded value from 0 to 100000 ns
- Forward Progress Screens (FPS) are used to ensure forward progress
  - Two RNR NAK encodings—one associated with Epoch 0 and one with Epoch 1
  - Responders service requests associated with the current epoch and return RNR NAKs for all others
  - Epochs progress over time as requests are drained from the current epoch

Thank you