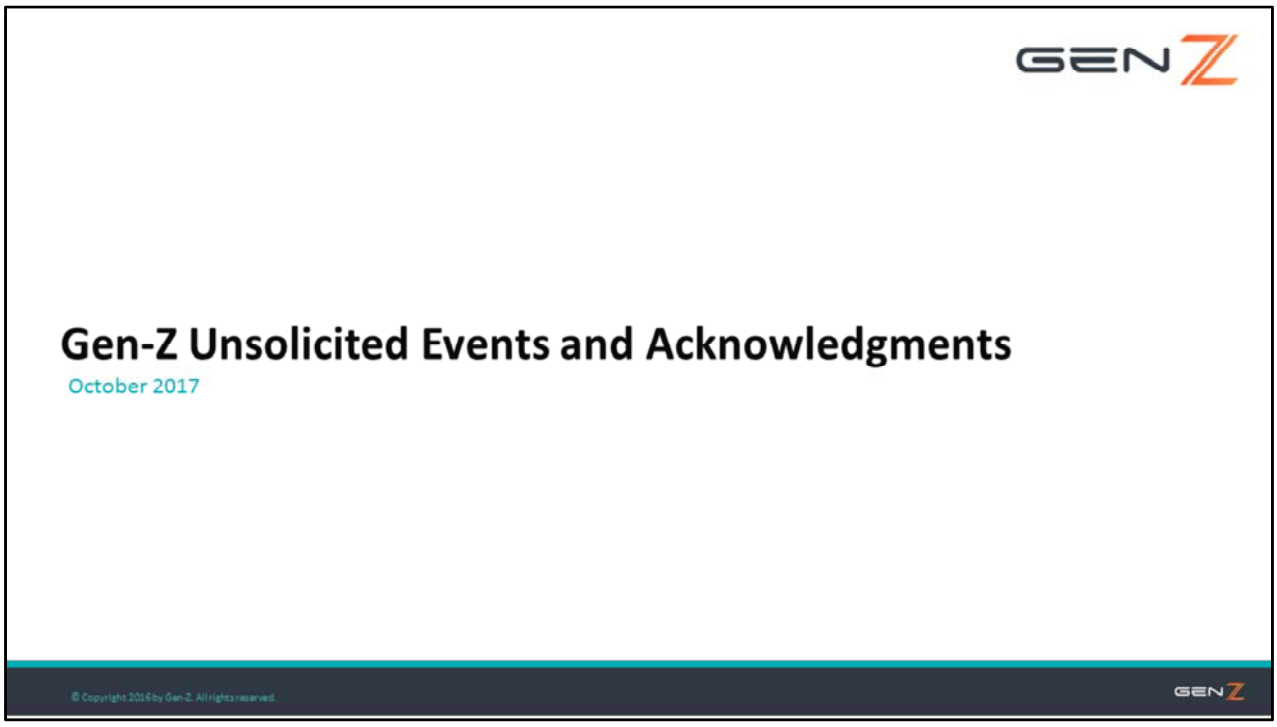


This presentation covers Unsolicited Events notification and handling, and standalone acknowledgments used to confirm reliable delivery or communicate errors.

Disclaimer

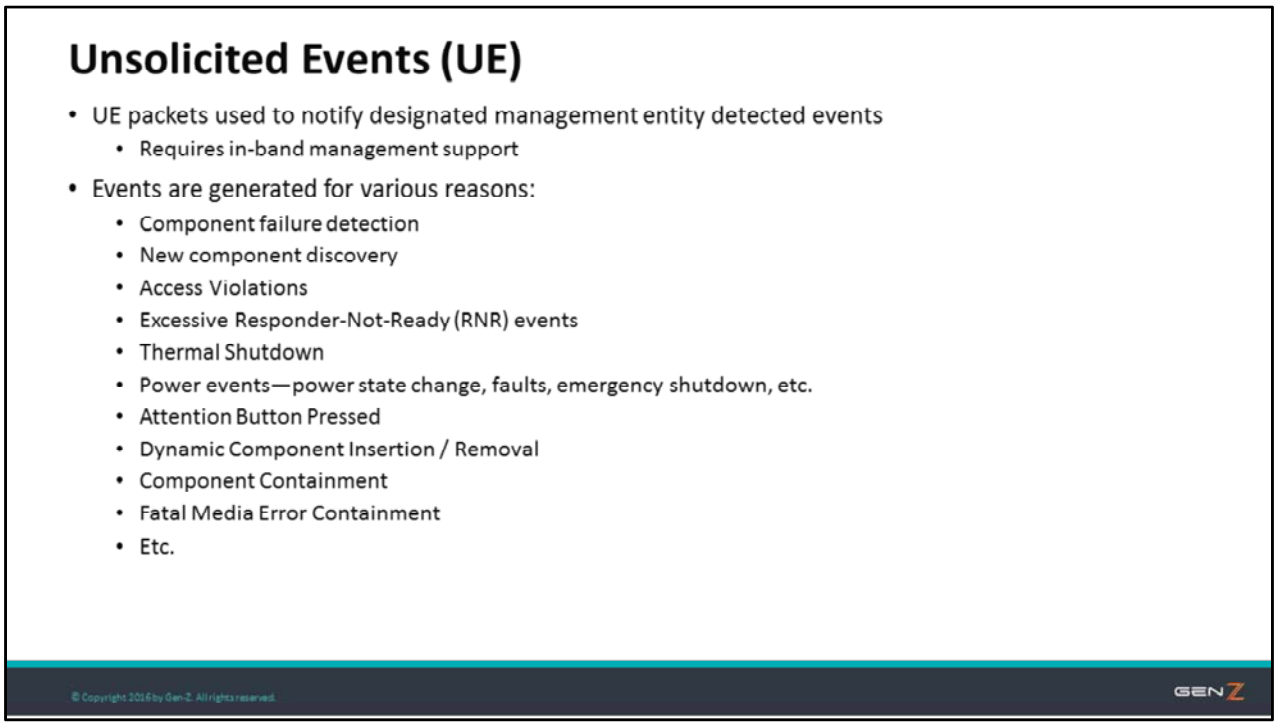
This document is provided 'as is' with no warranties whatsoever, including any warranty of merchantability, noninfringement, fitness for any particular purpose, or any warranty otherwise arising out of any proposal, specification, or sample. Gen-Z Consortium disclaims all liability for infringement of 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/

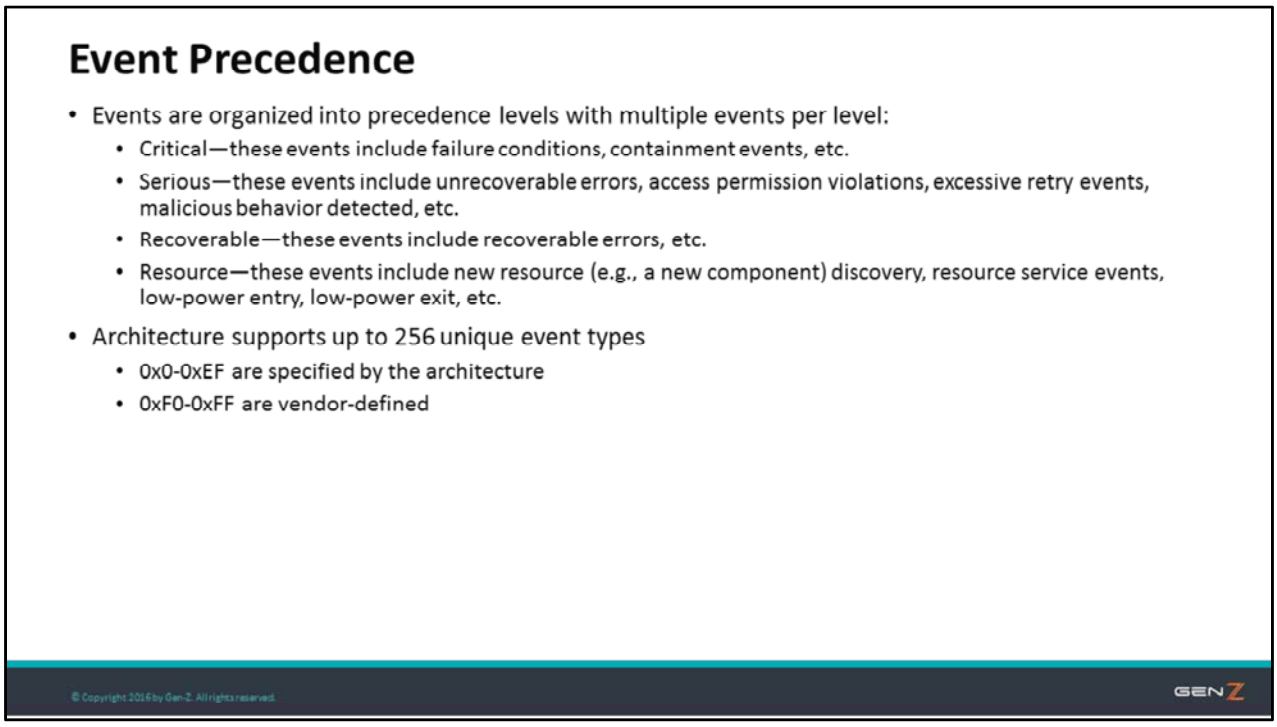


UE packets are used to asynchronously notify management of events that impact component operation.

UE packets require Control OpClass support, i.e., in-band management.

The architecture supports UE notification for a wide-range of events. These include failure events, malicious or rogue component detection events, power and thermal events, error containment events, performance events, and many more.

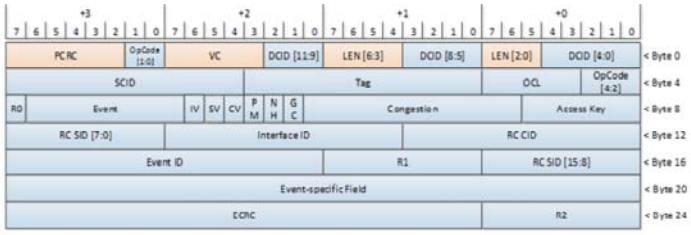
Management selectively configures which events are to be communicated to management.



Events are organized into four basic groups. Group precedence is established to ensure that the highest-precedence issue or error is communicated by the UE packet.

The architecture supports up to 256 unique event types (only a fraction have been specified). A subset of events are allocated for vendor-defined use to enable customization to meet component-specific needs.

Control OpClass 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

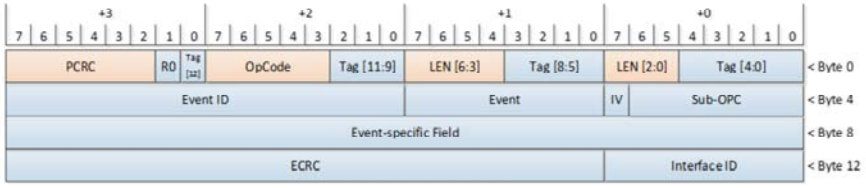
GEN Z

The UE packet is a Control OpClass packet, and as such, it contains all of the common explicit OpClass packet fields. The packet also contains the UE-specific fields described above.

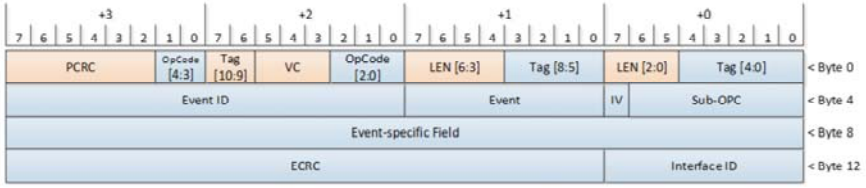
UE packets are not acknowledged, i.e., they are treated as unreliable datagram packets. A component periodically transmits the UE packet until management clears event notification (a Control Write operation to the Core structure). Not only does this approach ensure reliable delivery due to fabric events or transient errors, but it also handles management component over-subscription (slow response or resource exhaustion) and management failure (component can target a second manager should it be unable to communicate with the primary manager).

To ensure that events are not lost, a component is required to provision resources to track multiple events—one per precedence level and one per component interface. To track an event, the component needs to provision resources to record the Event, RC CID, RC SID, Interface ID, and the Event-specific field.

P2P-Core and P2P-Coherency UE Packet Formats



P2P-Core UE Packet

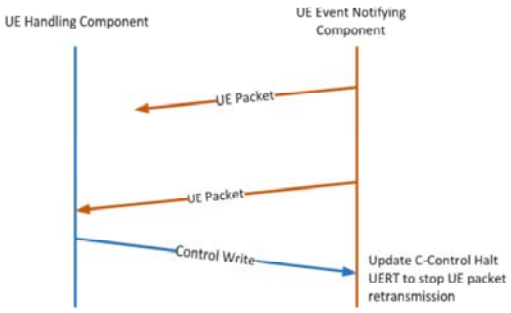


P2P-Coherency UE Packet

- P2P-Core and P2P-Coherency UE packets contain a subset of the Control OpClass UE packet fields
  - Operates only in point-to-point topologies

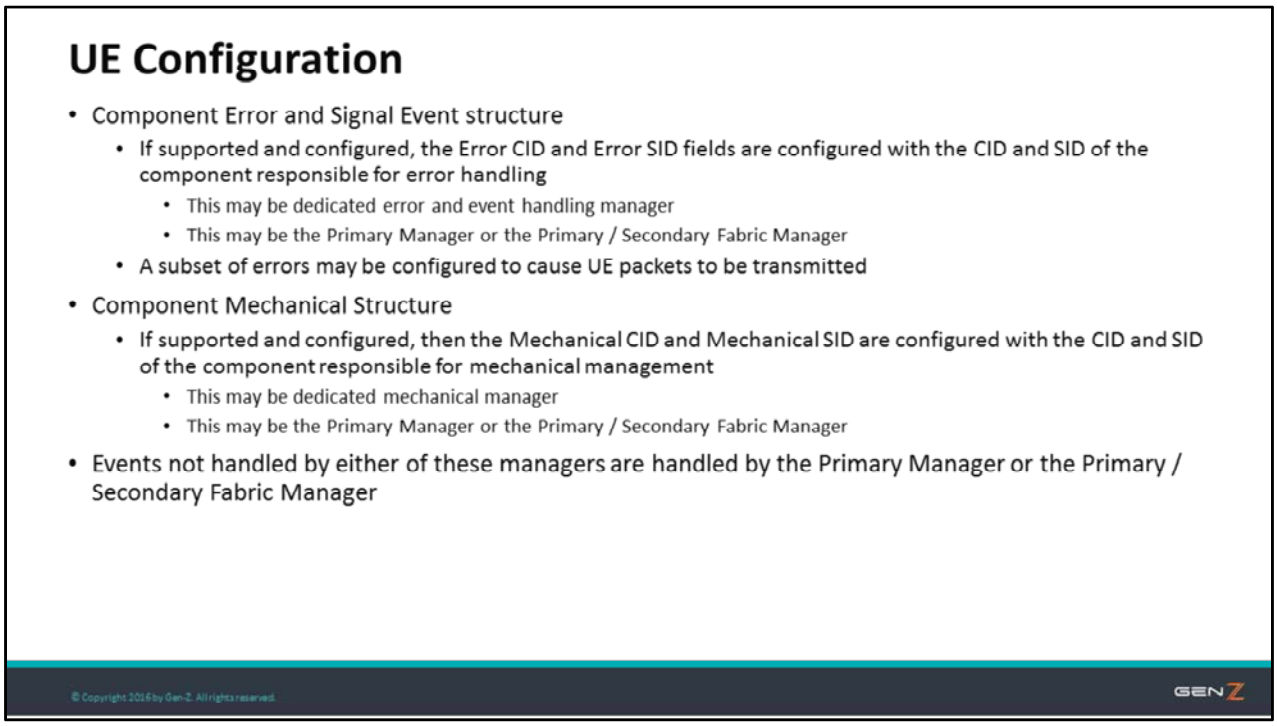
UE Packets

- 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.



GEN Z

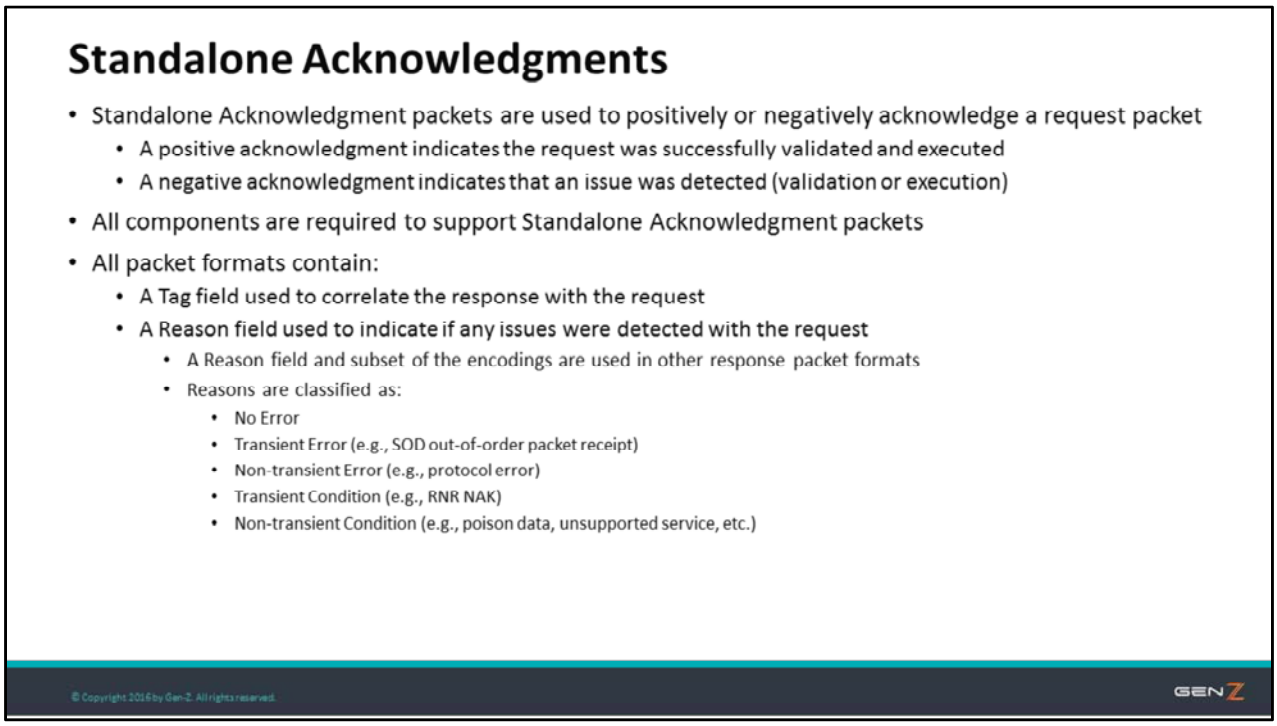
This figure illustrates how UE packets are transmitted and retransmitted upon expiration of a UERT timer. If retransmission is triggered, then the component should transmit the UE packet on a different interface capable of reaching the management component. If it repeatedly fails to reach the primary management component, then it should try to reach the secondary (if available). If a component cannot reach any component, then it should stop UE packet retransmission and clear the event tracking logic.



If the component supports the Component Error and Signal Event structure, then it may be configured to target a specific error and event handling management component. This structure can also be used to control which error and events require a UE packet to be transmitted, e.g., management may determine that simply protocol errors do not require UE notification and component power events do.

Similarly, if the component supports the Component Mechanical structure, then it can target a specific mechanical management component for mechanical-specific events, e.g., the detection of a new component, surprise removal of a component, etc.

If either of these structures is not supported, or a specific event is not configured through these structures, then the component targets the Primary manager or either of the fabric managers.

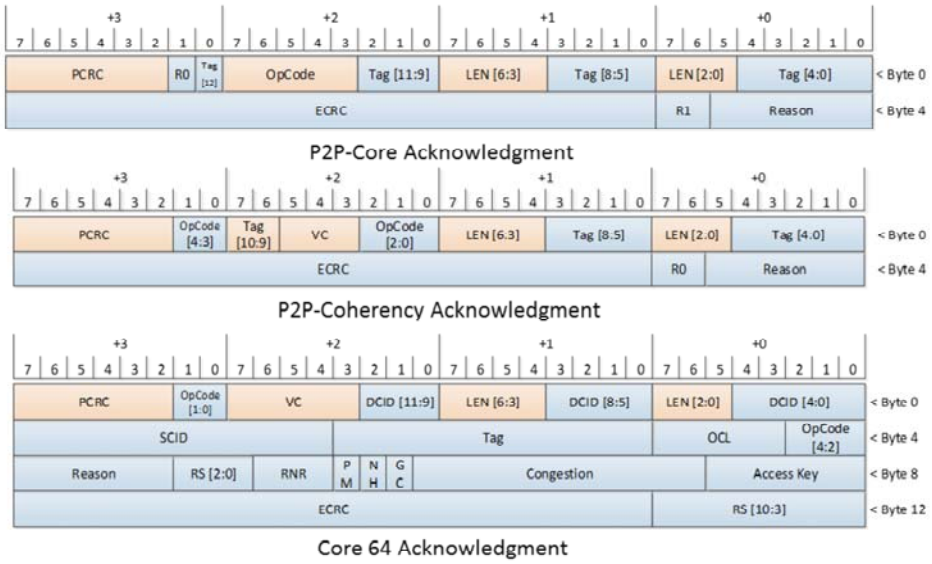


Standalone acknowledgments are used to communicate the success or failure of a request packet. As such, all components are required to support Standalone Acknowledgments for any supported OpClass.

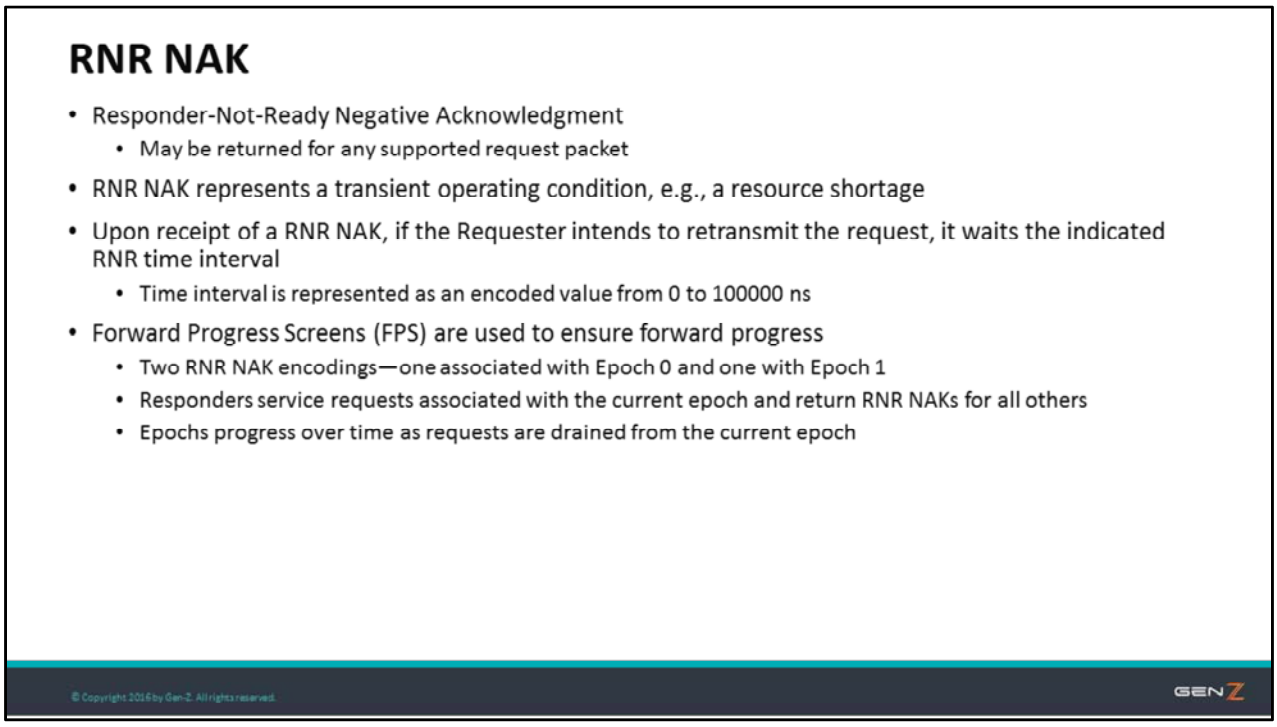
A Standalone Acknowledgment packet contains sufficient information to correlate it with the request packet. It also contains a Reason field that communicates the success or specific failure / error / operating condition. Reasons are classified to indicate whether request packet retransmission is required. Request packet retransmission is performed if a transient condition is detected; it is not performed of a non-transient condition is detected.

Some response packets contain a Reason field that uses a subset of the encodings. In general, this subset is focused on underlying media state, e.g., if a correctable or uncorrectable error was detected in the underlying media. The Reason field in these response packets is not used for any reason not explicitly stated by the specification.

Standalone Acknowledgment Packet Formats

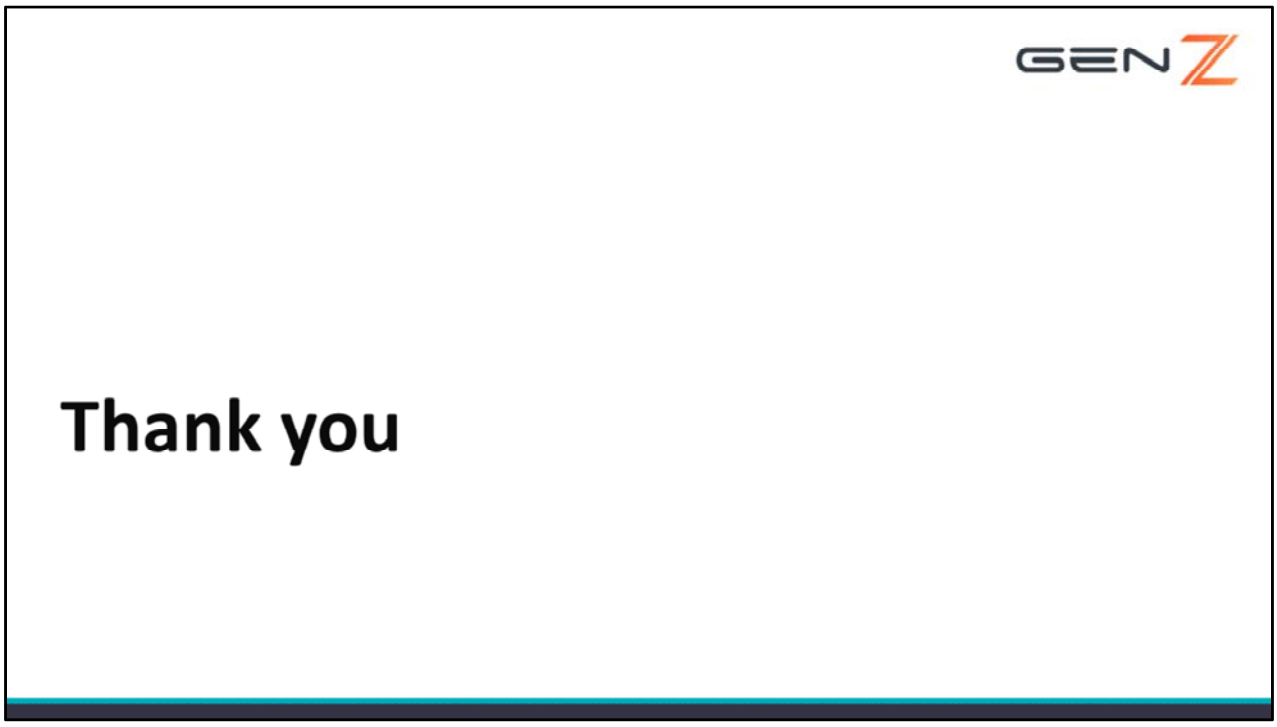


P2P-Core and P2P-Coherency standalone acknowledgment packets are used only for their respective OpClasses. Core 64 standalone acknowledgment is used by all explicit OpClasses.



If a Responder is unable to execute a request packet, it should return an RNR NAK Reason code and an RNR time value to the Requester.

- The RNR NAK encoding indicates the forward progress epoch to communicate in the retransmitted packet. This is used by the Responder to ensure that all Requesters make forward progress in the face of heavy load.
- The RNR time value is the minimum amount of time that the Requester is required to wait prior to retransmitting the request packet.



This concludes this presentation. Thank you.