This presentation covers Gen-Z link flow control. Link flow control uses link-local packets to exchange flow-control credits.
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/
Link flow control is used to prevent buffer overflow. Gen-Z supports two types of link flow control—Implicit and Explicit.
Implicit flow control is applicable to component interfaces that support the P2P-Core, P2P-Coherency, or P2P-Vendor-defined OpClass. Implicit flow control improves wire protocol efficiency by eliminating the exchange of explicit flow-control credit packets.

To support implicit flow control, the Requester tracks each Responder interface’s available receive buffer space. To transmit a request packet, the Requester needs to ensure that it has sufficient receive buffer space to contain the corresponding response packet, and it needs to ensure that the Responder has sufficient buffer space to receive the request packet.
Explicit flow control may be used in any topology. Explicit flow control uses link-local packets to exchange each receiver's present flow-control credit state. Maximum flow-control credits per VC is $2^{14}$ credits:
- A credit represents 16 bytes for a total of 1 MiB/VC
- Reserved bits to enable future growth
- Sufficient to drive multi-TB/s bandwidth
- Sufficient to drive relatively long physical distances

Flow-control credits per VC may vary

Optional support for adaptive flow control:
- Enables credits from underutilized VCs to be used for over-subscribed VCs to alleviate congestion
- Eliminates the need to overprovision credits for worst-case operating conditions—reduces cost and complexity when supporting large number of VCs

Adaptive flow-control enables a transmitter to optimize the aggregate flow-control credits across all VCs to meet current transmission needs. For example, a receiver that supports an aggregate of 4096 flow-control credits equally distributed across 8 VCs, i.e., 512 credits per VC. Dynamic workload conditions could result in some VCs being underutilized and some being oversubscribed, i.e., congested. AFC enables a portion of the flow-control credits from the underutilized VCs (e.g., 256 credits) to be used by the oversubscribed VCs to alleviate congestion. By adaptively adjusting flow-control credit usage, a receiver can avoid overprovisioning resources to meet worst-case workload behavior.
Conceptually, the receive space associated with each VC is a circular buffer containing Max_FC credits. The available receive space is calculated as the difference between the head and tail pointers (modulo Max_FC).

For each supported VC, the implementation needs to provision at least sufficient flow-control credits to support the maximum packet size (sum of the maximum number of protocol bytes for the supported OpClasses plus the maximum number of payload bytes).
Since link-local packets are transmitted as unreliable datagrams, each interface is required to periodically transmit flow-control credit packets for each enabled VC. If an interface fails to receive flow-control credits for each enabled VC for two consecutive timer expirations, then the interface automatically initiates physical layer retraining.
There are two types of link flow-control packets—a single VC-LLR (link-level reliability) combination packet and a dual-VC packet. The single VC / LLR combination can be used to improve protocol efficiency by eliminating the need for a separate LLR acknowledgment packet.
Thank you

This concludes this presentation. Thank you.