Nebula GPU Interconnect System: Technical Architecture and Implementation

Pranav Chandra, Pramit Pal, Meghadri Ghosh
Team Bob

September 2025

Outline

Outline

Project Overview

- **Output:** A scalable NOC design for multi-GPU system
- **Topology:** 2D mesh, 2x2 to 8x8 grid (up to 64 GPUs)
- **Protocols:** ARM AMBA AXI4 (non-coherent), CHI (coherent)
- **Languages Used:** SystemVerilog (RTL), Python (Analysis)
- **Key Features:**
  - Five-stage router pipeline
  - Adaptive and deterministic routing
  - Virtual channels for deadlock avoidance
  - Credit based flow and arbitration

Outline

The Packet

- Packets are sent as FLITs. Each packet consists of:
  - **Header:** 48 bits
  - **Payload:** 512 bits, data being transferred
  - **Tail:** 32 bits, error checking and end of packet marker
- Packets may be sent as single or multi-flit.
- Packets are usable in up to 16x16 meshes.
- QoS allows for packet prioritization.

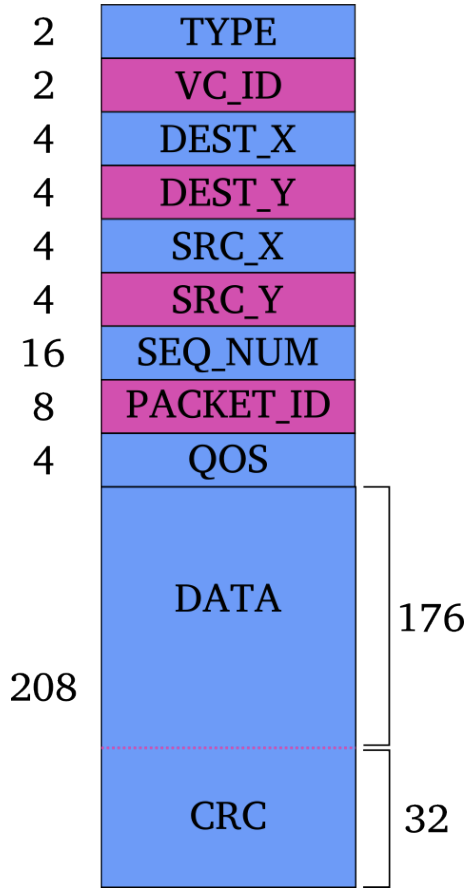

Figure: Packet Structure

Packet Assembly and Disassembly

- Handled by `nebula_packet_assembler.sv` and `nebula_packet_disassembler.sv`.
- Converts coordinates and metadata into predefined packets.
- Payload segmentation splits large payloads across multiple flits to optimize transmission.
- Ensures proper packet formation for both AXI4 and CHI protocols.

Outline

Router Microarchitecture

- Routers use a 5-stage pipeline for processing packets:
  1. Buffer Write (BW)
  2. Route Computation (RC)
  3. Virtual Channel Allocation (VA)
  4. Switch Allocation (SA)
  5. Switch Traversal (ST)
- Routers have 5 ports each.
- Each port has 4 virtual channels (VCs).
- Each VC has a buffer depth of 8 flits.
- Four stage FSM to determine control.

Buffer Write (BW)

- Incoming flits are written to the appropriate VC buffer.
- Flow control is managed using a credit-based system.
- Ensures that buffers do not overflow.

Route Computation (RC)

- Determines the next hop for the packet based on destination address.
- Implements XY routing algorithm for deadlock-free routing.
- Considers network congestion and adaptively selects paths.

Virtual Channel Allocation (VA)

- Allocates a virtual channel for the packet to use on the next hop.
- Uses a round-robin arbitration scheme to ensure fairness.
- Prevents head-of-line blocking by allowing multiple packets to share physical channels.

Switch Allocation (SA)

- Allocates the switch fabric to the selected virtual channel.
- Uses a priority-based arbitration scheme to resolve conflicts.
- Ensures that high-priority packets are transmitted first.

Switch Traversal (ST)

- The packet is transmitted through the switch to the next router or destination.
- Updates credit counters for flow control.
- Ensures that packets are sent in the correct order.

Virtual Channel FSM

- Each virtual channel maintains independent state, enabling concurrent packet processing across flows.
- The FSM follows a four-state cycle:
  1. **IDLE** → awaits packet arrival
  2. **ROUTING** → performs route computation
  3. **WAITING VC** → waits for VC allocation
  4. **ACTIVE** → transmits flits, returns to IDLE after TAIL/SINGLE
- Transitions are driven by pipeline events: head/single flit arrival, route computation results, VC allocation success, and tail/single flit detection.
- Coordination relies on `rc_valid`, `va_valid`, switch arbitration signals, and generates `vc_read_en` for FIFO timing.

Routing Algorithms

1. Deterministic XY Routing
2. Adaptive Routing

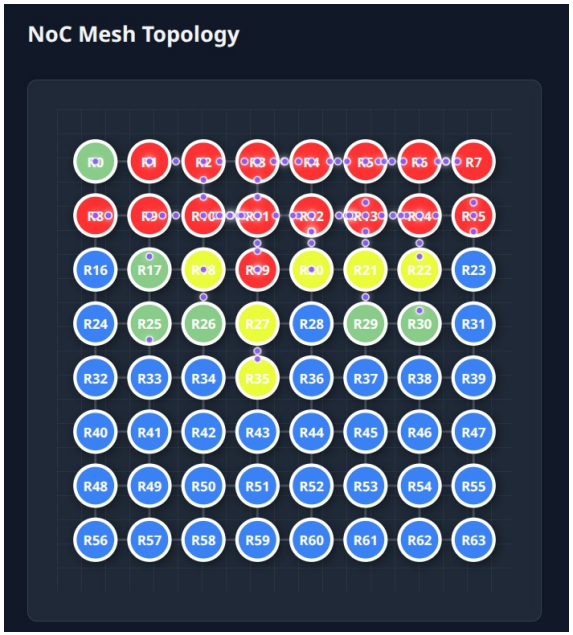

Figure: XY Routing Example

Figure: Adaptive Routing Example

Outline

Credit-Based Flow Control

- **Credit Controller (`nebula_credit_flow_ctrl.sv`):**
  - Per-VC credit counters, max 16
  - Increment on flit acceptance, decrement on allocation
  - Prevents buffer overflow, deadlock
  - `credits_available` signals for arbitration
- **Flow Control Protocol:**
  - Sender stalls if credits $= 0$
  - Lossless operation guaranteed

Arbitration Mechanism

Outline

Questions

Questions?