# Functional Description: Streaming Protocol

A bridge can connect a master or slave block to the NoC and perform the required operations to support the master and slave communication as per the NSIP protocol standards.

## Bridging from Host to NoC

A host may have multiple Streaming ports through which it can transmit and receive data to/from the NoC. A bridge component converts a host port’s messaging protocol into a packetized protocol for NoC. Bridges are automatically instantiated by NocStudio based on the protocol specified for host port in NocStudio. There is one bridge per host port; the bridge connects the host port to routers at the grid points. Multiple routers may exist at a grid point of the mesh, one router for each NoC layer, in which case the bridge will connect the host port to each router.

Bridge parameters and properties are assigned by NocStudio based on the high-level specification of traffic and hosts. Some bridge properties are made visible to the user; these properties can be directly modified with “host prop” and “bridge\_prop” commands in the NocStudio. Please refer to these commands for the list of user-modifiable bridge parameters. Bridges are designed and optimized for low-latency and high-frequency operation. Address lookup, route information encoding, QoS, protocol-related conversions and processing, etc. are all tuned and configured through NocStudio based on optimizations or user specification.

### NetSpeed Streaming Bridge

The Streaming Bridge maps the transaction messages issued from host port interfaces to NoC layers and VCs. The bridge also performs the route computation, enforces QoS (fixed priority and weights), and performs the flow control and arbitration.

Streaming Bridge



Figure 12: NetSpeed Streaming Bridge functions

The NetSpeed streaming protocol bridge provides a relatively simple interface to the host ports and performs simple packetization of host port data into NoC packets.

Streaming hosts using the streaming bridges to communicate with each other can use up to four interfaces allowed by the bridge. The interfaces are named a, b, c and d. Whether an interface is bidirectional or not, depends on the traffic specification, i.e. if a traffic transaction message only leaves (arrives at) an interface, then the interface becomes an output-only (input-only), otherwise it remains bidirectional. The width of each direction of an interface is configurable in terms of power-of-two multiples of cell\_size, and can be viewed and modified using the **bridge\_prop** command.

A streaming host’s interface may communicate with any other streaming host’s interface. Each injected transaction message must have a destination bridge id (globally unique identifier) and interface id. Note that two interfaces with different width can communicate with each other. For streaming host ports, each transaction message hop must be explicitly specified using **add\_traffic**. The dependencies between different transaction hop messages are inferred from **add\_traffic** and they can also be specified explicitly with the **add\_dep** command. An interface may transmit or receive traffic of multiple priorities and weights, properties indicated in the transaction using the 4-bit QoS signal.

The signals at the host port side of a NetSpeed streaming bridge are listed below; the direction is with respect to the host port.

Table 1 Host and NetSpeed Streaming Bridge Interface

|  |  |  |  |
| --- | --- | --- | --- |
| **Signal Name** | **Dir** |  | **Description** |
| **Ingress interface to NoC from Host** | | | |
| destination bridge id | OUT | 8-bit | Destination bridge id (its globally unique identifier) for this message |
| interface id | OUT | 2-bit | Destination host port’s interface id for this message |
| data beat valid | OUT | 1-bit | data valid |
| start of transaction message | OUT | 1-bit | start of message |
| end of transaction message | OUT | 1-bit | end of message |
| quality of service attribute of the message | OUT | 4-bit | QoS of the message |
| data beat | OUT | EGRESS\_WIDTH\_\*-bit | Data |
| credit increment return | IN | 1-bit | credit increment feedback |
| **Egress interface from NoC to Host** | | | |
| data beat valid | IN | 1-bit | data valid |
| start of transaction message | IN | 1-bit | start of message |
| end of transaction message | IN | 1-bit | end of message |
| data beat | IN | INGRESS\_WIDTH\_\*-bit | Data |
| credit increment return | OUT | 1-bit | credit increment feedback |

Here \* can be replaced with a, b, c or d for the four host port interfaces

Each transaction injected at an interface forms a single NoC packet.