



CnuasLink, Datasheet

Accelerator to accelerator fabric interconnect

Document DS-CNU-005 • **Revision** C • **Issued** 23 August 2026 • **Status** Published

Item	Value
Part	cnuasgpu-link-switchd
Type	Soft GPU to GPU fabric interconnect, runs as a host daemon
Version	v0.1.0-3-g812f464
Repo	PacketFive/CnuasLink

1. Overview

CnuasLink is the dedicated GPU-to-GPU fabric of the Cnuas platform, modelling an NVSwitch/NVLink-class interconnect separate from the RoCE/IB NIC fabric. It provides a frame-based protocol for high-bandwidth data movement, doorbells, atomics, and collective operations between CnuasGPU endpoints. Same-host GPU pairs use an ivshmem shared-memory fast path; cross-host traffic is switched by the `cnuasgpu-link-switchd` daemon.

Deployment modes

`cnuasgpu-link-switchd` is **an ordinary host process**, like `CnuasSwitch` and unlike the PCIe device models. Endpoints attach over `SOCK_SEQPACKET` sockets, so a host process can join the fabric directly, without a guest.

Mode	What attaches as an endpoint	Needs a guest	Typical use
Soft fabric (host only)	Any host process speaking the CnuasLink framing	No	Protocol, discovery, forwarding and collective-offload development
Rack fabric	Guests with CnuasGPU, over the QEMU device and ivshmem	Yes, for the endpoints	Full rack deployments and multi-GPU collectives

The host-only mode is what the CnuasLink test suite uses. It spawns the daemon and walks discovery, unicast, broadcast, port status and link-down behaviour with no virtual machine present.

An important scope note. The **fabric** runs on the host, but the ivshmem shared-memory fast path is a property of the QEMU deployment, so host-only use exercises the switched path rather than the peer fast path.

Key features

- Runs entirely as a host daemon; no hypervisor, guest or kernel module.
- Purpose-built GPU fabric, independent of the RoCE/IB NIC fabric.
- Frame-based protocol with data, doorbell, atomic, and collective message types.
- Collective offload types for AllReduce / AllGather (CnuasCCL).
- ivshmem shared-memory fast path for same-host GPU pairs (NVLink-like latency).
- Switched cross-host path via `cnuasgpu-link-switchd`.
- Automatic FDB learning from DISCOVERY frames; discovery + keepalive.



2. Fabric diagram

Diagram available in the online documentation at <https://github.com/PacketFive/cnuas>.

3. Functional specifications

Parameter	Value
Fabric ports (GPU links)	8 default (configurable, max 32)
Inter-switch link	port 8
Console / OTel port	port 9
Frame header size	12 bytes (packed)
Frame magic	0x484C ("HL")
Protocol version	0x01
Max frame	64 KB (12 B header + payload)
GPU endpoint lanes	4 (default, see CnuasGPU)
Concurrency model	Single-threaded epoll loop
Forwarding	FDB learned by source GPU ID; unicast, or fan-out on 0xFFFF

4. Wire protocol

4.1 Frame header (12 bytes)

Field	Size	Notes
Magic	2 B	0x484C
Version	1 B	0x01
Type	1 B	message type (below)
Src GPU ID	2 B	source endpoint
Dst GPU ID	2 B	destination endpoint (0xFFFF = broadcast)
Length	4 B	payload length
Payload	variable	message body

4.2 Frame types

Type	Value	Purpose
CNUASLINK_TYPE_DATA	0x01	Peer-to-peer data copy. Also carries the 40 byte CnuasCCL envelope for v0.2 collective payload, so the switch does not need a collective aware fast path.
CNUASLINK_TYPE_DOORBELL	0x02	Notification
CNUASLINK_TYPE_ATOMIC	0x03	Atomic operation
CNUASLINK_TYPE_COLLECTIVE_REDUCE	0x04	Reserved for a future switch offloaded AllReduce; CnuasCCL v0.2 uses CNUASLINK_TYPE_DATA
CNUASLINK_TYPE_COLLECTIVE_GATHER	0x05	Reserved for a future switch offloaded AllGather; CnuasCCL v0.2 uses CNUASLINK_TYPE_DATA
CNUASLINK_TYPE_DISCOVERY	0x06	GPU discovery / FDB learning
CNUASLINK_TYPE_LINK_KEEPAIVE	0x07	Keep-alive probe



5. Daemon internals

Subsystem	Source	Responsibility
Entry / CLI	main.c	Argument parsing, startup
Core	cnuaslink_switch.c	cnuaslink_switch_init() / _destroy(), port + FDB setup
Event loop	cnuaslink_loop.c	cnuaslink_switch_run(), RX parse, routing, TX
Management	cnuaslink_mgmt.c	cnuaslink_mgmt_handle(), JSON commands
Protocol defs	include/cnuaslink_proto.h	Header + type constants

The forwarding database is driven by `cnuaslink_fdb_learn()`, `cnuaslink_fdb_lookup()`, and `cnuaslink_fdb_dump()`.

Routing runs as follows. The switch receives on `SOCK_SEQPACKET`, parses and validates the 12 B header, and learns `src_gpu` → `port`. It then forwards by `dst_gpu` lookup for unicast, or fans out to every enabled port except the ingress port when `dst_gpu` is `0xFFFF`. Drop counters track invalid headers and frames with no route. Transmission uses `send(..., MSG_EOR)`.

Four `epoll` event tags carry the port identifier and file descriptor packed together: `EV_TAG_PORT_LISTEN`, `EV_TAG_PORT_CLIENT`, `EV_TAG_MGMT_LISTEN`, and `EV_TAG_MGMT_CLIENT`.

6. Management API (JSON)

Command	Purpose
<code>{"cmd": "set_link_state", "port": N, "enabled": true false}</code>	Enable/disable a port

Response: `{"status": "ok"|"error"}`. FDB learning is automatic from DISCOVERY frames and valid source GPU IDs.

7. Interfaces

Interface	Description
Same-host transport	ivshmem (POSIX shared memory + UNIX socket signalling)
Cross-host transport	UNIX domain socket (<code>SOCK_SEQPACKET</code>) to the switch
GPU-side endpoint	MMIO mailboxes in CnuasGPU BAR0
Run directory	<code>/var/run/cnuaslink/</code>

8. Software support

Item	Value
Switch daemon	<code>cnuasgpu-link-switchd</code> (C / GNU C)
Management CLI	<code>cnuaslink-cli</code>
Design reference	CnuasLink Switch Design

9. Ordering / integration information



Item	Value
Repo	PacketFive/CnuasLink
Submodule path	src/cnuaslink (in PacketFive/cnuas)
Version	v0.1.0
Language	C (GNU C)

10. Revision history

Revision	Date	Notes
A	2026-07-05	Initial datasheet
B	2026-07-05	Added fabric diagram, full frame-type table, daemon internals
C	2026-08-12	Recorded that the fabric daemon runs on the host with no guest, as a soft GPU to GPU interconnect

PacketFive, Packet Five Networks Ltd., Dublin, Ireland.

Cnuas Virtual AI/HPC Infrastructure is published at <https://github.com/PacketFive/cnuas> under the Apache License 2.0; read it at <https://github.com/PacketFive/cnuas/blob/main/LICENSE>.

Cnuas Virtual AI/HPC Infrastructure is emulation software. It is not affiliated with, endorsed by, or derived from any hardware vendor, and every device it models is a software artefact. The work is published by its authors in a personal capacity and is not sponsored or endorsed by any employer.

Specifications describe the referenced revision of the software and may change without notice. Contact info@packetfive.com.