MARVELL

Extending ASTRA-sim with HTSim for Ultra Ethernet Simulation

ISCA 2026 Tutorial • Ultra Ethernet Consortium AI Fabric Track

Veerasenareddy Burru
Marvell Technology . 2026

Agenda

- Background: ASTRA-sim and HTSim simulators
- Motivation: why integrate the two
- Integration challenges and solutions
- Integrated simulation flow
- Sample results: increased radix and packet trimming impact
- Summary and key takeaways

ASTRA-sim: Collective Communications Simulator

What is ASTRA-sim?

Open-source simulator for large-scale distributed AI training.

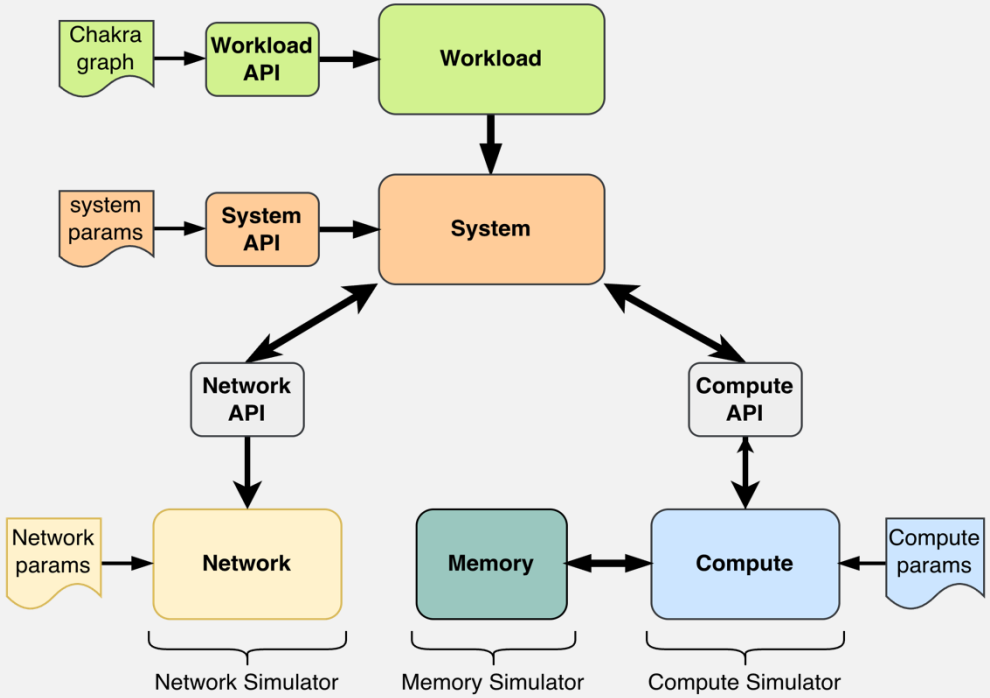
Models the full system stack:
Compute → Memory → **Collective Comms** → **Network**

Key capabilities:

- Chakra trace-driven workload injection
- Collective comms: AllReduce, AllGather, A2A, ReduceScatter
- Pluggable network backends
- Topology-aware scheduling & perf metrics

Widely adopted by AI infrastructure researchers across industry and academia for collective level workload simulation.

System Architecture



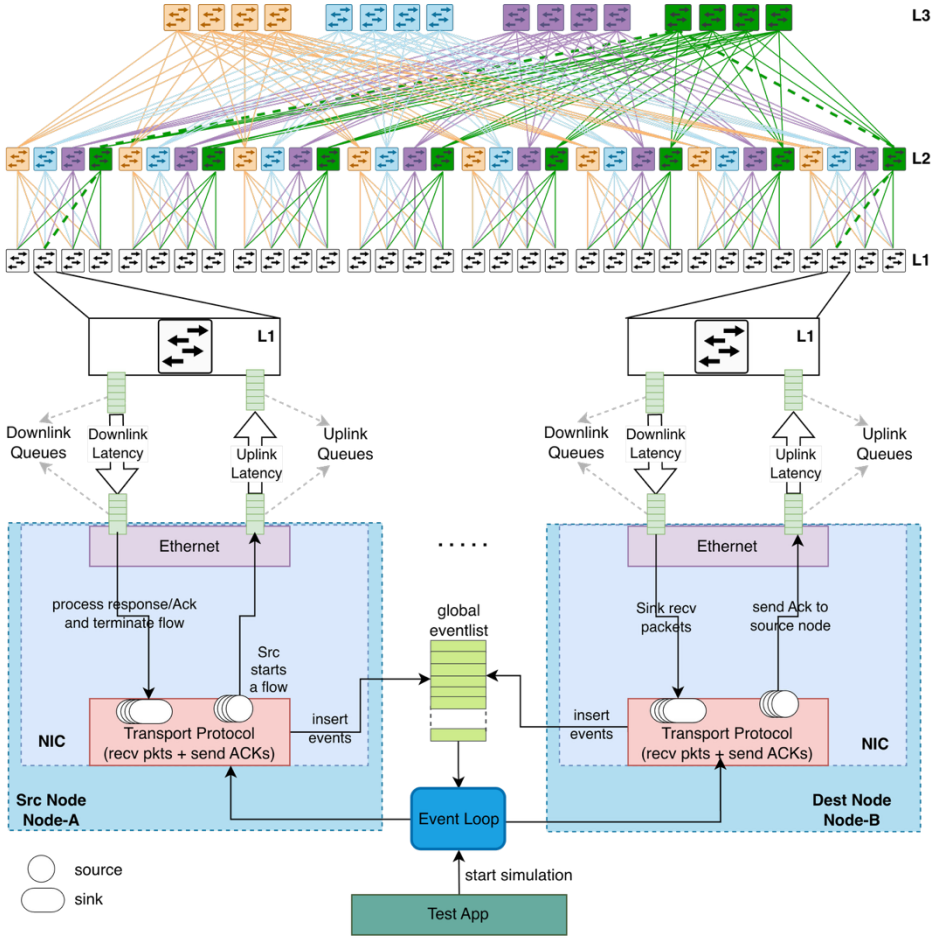
HTSim: UEC's Reference Packet-Level Simulator

HTSim is the UEC WG's reference packet-level simulator — tool for testing UEC mechanisms.

UEC is a next-gen AI fabric transport standard — new semantics beyond RDMA, designed for collective workloads:

HTSim features:

- **Multipath load balancing**
  - Adaptive & Oblivious Packet Spray
- **Congestion control**
  - UEC-specific, tailored to AI collectives
- **New transport semantics**
  - Message-based with per-flow tracking
- **Faster simulation**
- **Discrete Event Engine**
- **Extendable to end point NIC simulation**



Why ASTRA-sim and HTSim integration ?

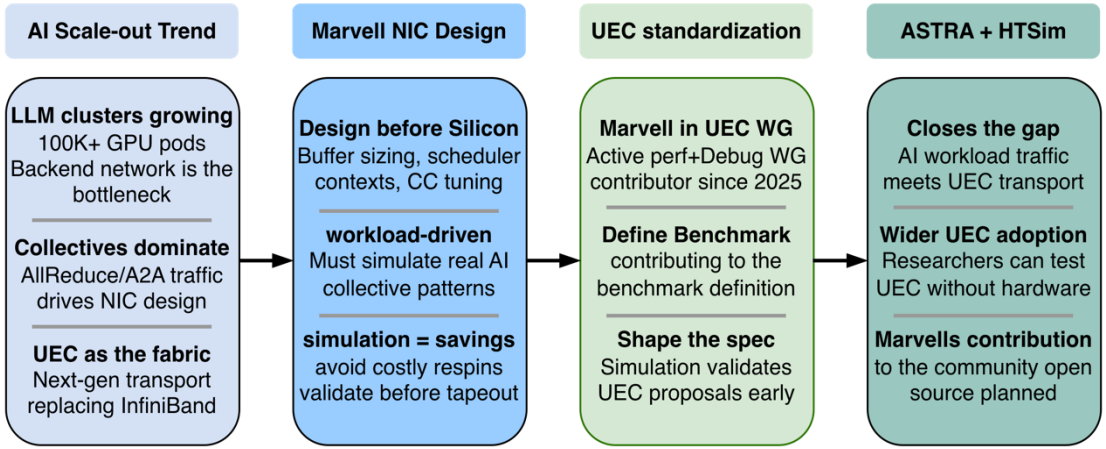
Marvell's AI Network Challenge

Marvell designs AI backend NICs for the world's largest Hyperscale data centers. Getting NIC architecture right requires answering:

1. **How do collective operations stress the NIC?**
   - Buffer sizing, scheduler contexts, CC tuning under a system level workload
2. **Which UEC mechanisms matter most?**
   - Adaptive spray, congestion control, out-of-order delivery
3. **How does topology affect NIC design?**
   - Single-tier vs multi-tier, oversubscription

ASTRA-sim provides the workload-level view. HTSim provides the UEC packet-level view. Together, they answer all three.

Strategic Context & Motivation



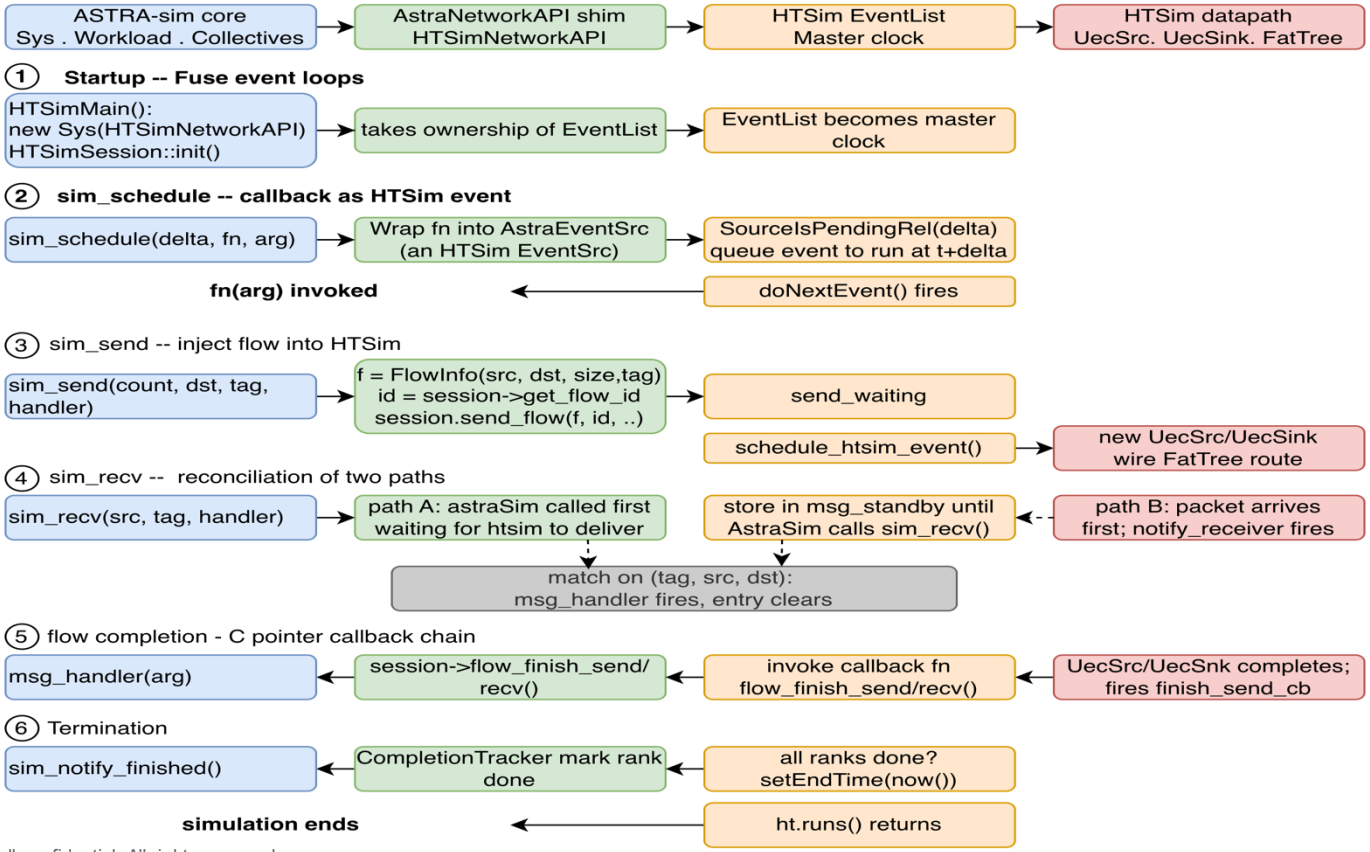
Use ASTRA-sim + HTSim to design better UEC NICs, validate benchmarks and make UEC accessible to the wider research community

Why present at ISCA 2026 ?
Invite the computer architecture community to co-develop, validate, and extend UEC simulation - accelerating ecosystem readiness

Integration Challenges & Marvell's Engineering Solutions

Challenge	Marvell's Solution
Combining event loops	- Made HTSim's EventList the single master loop. - ASTRA-sim's callbacks become HTSim EventSource objects (AstraEventSrc) so compute events and packet events interleave on one unified timeline
Integrating clocks	- Single conversion layer: sim_get_time returns timeAsNs(eventlist.now()) - sim_schedule converts delta with timeFromNs(Δ) before calling sourceIsPendingRel — one place, no drift
Network API plugin	- Built nHTSimNetworkApi + HTSimSession: sim_send wires a fresh UecSrc/Sink through FatTree per flow - sim_schedule wraps callbacks into AstraEventSrc; flow completion returns via C function pointers to msg_handler
HTSim upstream-clean	- Every hook gated behind #ifdef MRVL_HTSIM. - The only cross-boundary contract is raw C function pointers set per flow — no HTSim header ever imports an ASTRA-sim symbol

ASTRA-Sim + HTSim Integrated flow



Sample Integrated simulation – Increased Radix

Protocol: **UET**
Comparison: **T100 (Radix 128, 2-tier, ECMP) vs Generic (Radix 64, 3-tier, ECMP)**
Workload: **Collective All-Reduce Ring Algorithm**
Scenario: **T100 high-radix 2-tier vs Generic 3-tier Clos fabric**
Scale: **8192 Nodes - 2,088,960 Flows**
Oversubscription: **1:1**
Switch Latency: **T100: 421 ns · Generic: 620 ns**
Generic Switch**UET + 3-tier**
T100**UET + 2-tier**

Result:
with increased Radix 128, the flows complete in ~38% less time with ~36% lower tail latency compared to Generic switch with Radix 64

Sample Integrated simulation – Packet Trimming Impact

Protocol: **UET**
Comparison: **T100 (421ns, Trim ON) vs Generic (620ns, Trim OFF)**
Workload: **Collective All-Reduce Ring Algorithm**
Scenario: **4 failure events in 2 waves**
Scale: **1024 Nodes - 130,048 Flows**
Oversubscription: **1:1**
Switch Latency: **T100: 421ns · Generic: 620 ns**

Result:
with packet trimming, the flows complete in ~48% less time with ~42% lower tail latency compared to Generic switch with no packet trimming

Summary & Key Takeaways

Full-stack UEC simulation: AI collectives (ASTRA-sim) over a faithful packet-level network (HTSim)

1. **Integrated simulator.** HTSim — UEC's reference packet-level simulator — now runs inside ASTRA-sim to model AI collectives over Ultra Ethernet in a single simulation.

2. **Co-simulation by event exchange.** ASTRA-sim schedules the workload and collectives while HTSim models packet-level transport; the two exchange events to advance one integrated run.

3. **Hardware design-space evaluation.** Compare switch and routing choices under realistic AI traffic — e.g:
   Teralynx with 128 radix completes the collective flows in **~38% less time and with lower tail latency** than a generic switch with ECMP
   With Packet trimming, collective flows complete in **~48% less time and with lower tail latency** compared to without packet trimming.

***** ASTRA-Sim is an important part of Marvell's Simulation toolkit *****

MARVELL

Thank You

Q&A

Extending ASTRA-sim with HTSim for Ultra Ethernet Simulation • ISCA 2026 Tutorial

Acknowledgments: Ulf Hanebutte, Nikhil Bernard John Stephen, Hemanth Singh

Veerasenareddy Burru

Marvell Technology • 2026