


Beehive: A Modular Flexible Network Stack for Direct Attached Accelerators

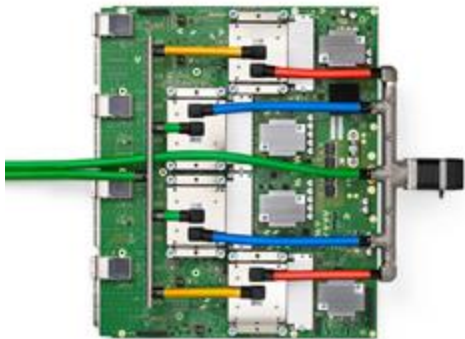
Katie Lim, Pratyush Patel, Jacob Nelson, Irene Zhang,
Tom Anderson



Accelerators in the datacenter



Ranganathan, Parthasarathy et al., "Warehouse-scale video acceleration: co-design and deployment in the wild", ASPLOS '21



Cloud TPU. <https://cloud.google.com/tpu>

- Datacenters increasingly moving computation into dedicated hardware leading to better energy efficiency
- Applications:
 - Video encoding: Google
 - ML: Google, Facebook, Microsoft
- Infrastructure
 - Network virtualization: AWS, Microsoft
 - Storage: AWS

Accelerator Efficiency

- Various research has shown accelerators on FPGA to have energy efficiency benefits across a range of applications
- Efficiency doesn't account for surrounding infrastructure required to integrate these accelerators into a system

Application	Efficiency	Speedup
CNN Inference [103]	4x–34x	1x–7.7x
RNN Inference [77]	4x–9x	0.7x–4.1x
Web Search [87]	1x–1.95x	1x–1.23x
Image Processing [89]	1.2x–22.3x	N/A
Intrusion Detection [137]	5x–52x	3x–10x
Document Filtering [17]	5.25x	0.85x
Video Compression [16]	5.2x	8.4x
Decision Trees [2]	23x–72x	2x–30x
Bzip2 Compression [90]	N/A	1.6x–2.3x
Key-value Stores [62]	2.29x	1.02x
Databases [52]	1.42x	1x–2x

Network-attached accelerators

- Some accelerators may be directly attached to a network, so they can communicate without CPU intervention
 - **Ex:** Microsoft, IBM both have deployments of FPGAs attached to their general purpose datacenter networks
- Energy efficiency benefits both for application and for infrastructure
- What should a hardware network stack look like?

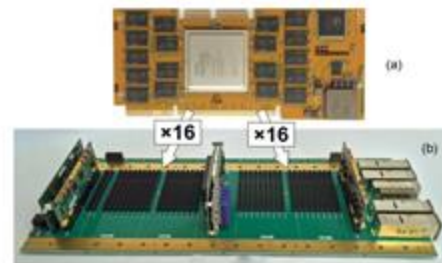
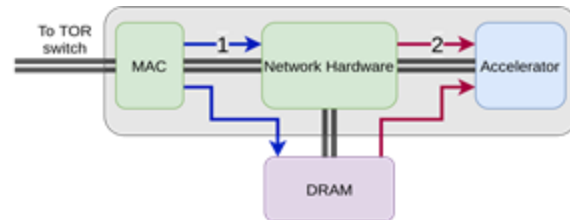
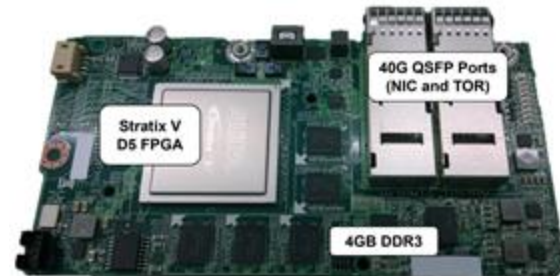


Figure 1: (a) The disaggregated FPGA and (b) the carrier board.

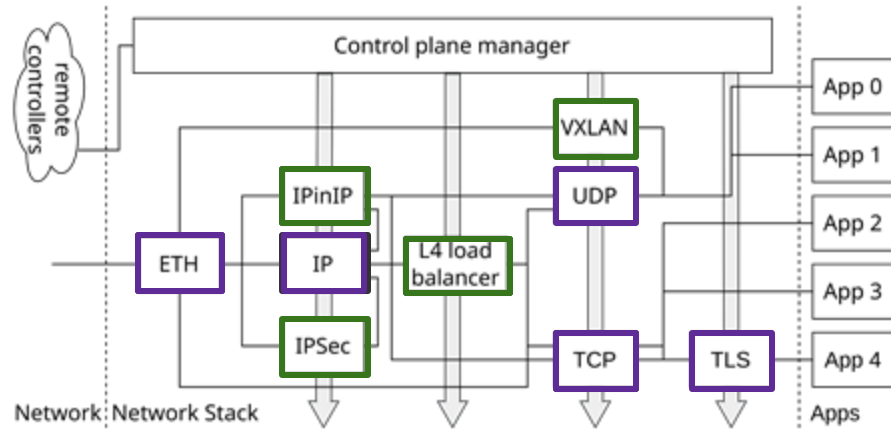
Abel, Francois. et al., "An FPGA Platform for Hyperscalers", HOTI '17



Caulfield, Adrian M. et al., "A cloud-scale acceleration architecture", MICRO'16

Software network stacks

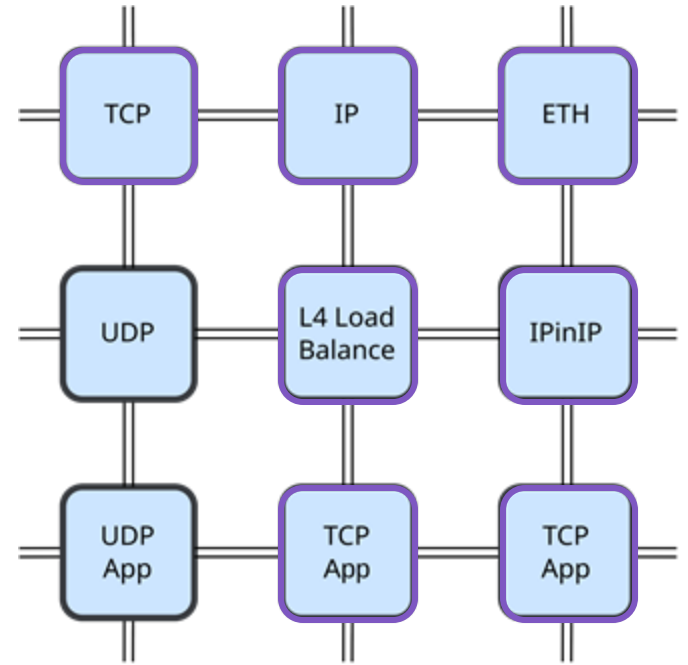
- Recent work in network stacks (e.g. Google Snap, eBPF) prioritizes modularity, customizability
- Variety of protocols that can be changed
 - e.g. Snap integrates a new transport protocol
- Custom network functions
 - E.g. load balancing, network virtualization
- Complex interconnections in the stack
- Potentially all layers need control plane access



Example software network stack overview

Beehive

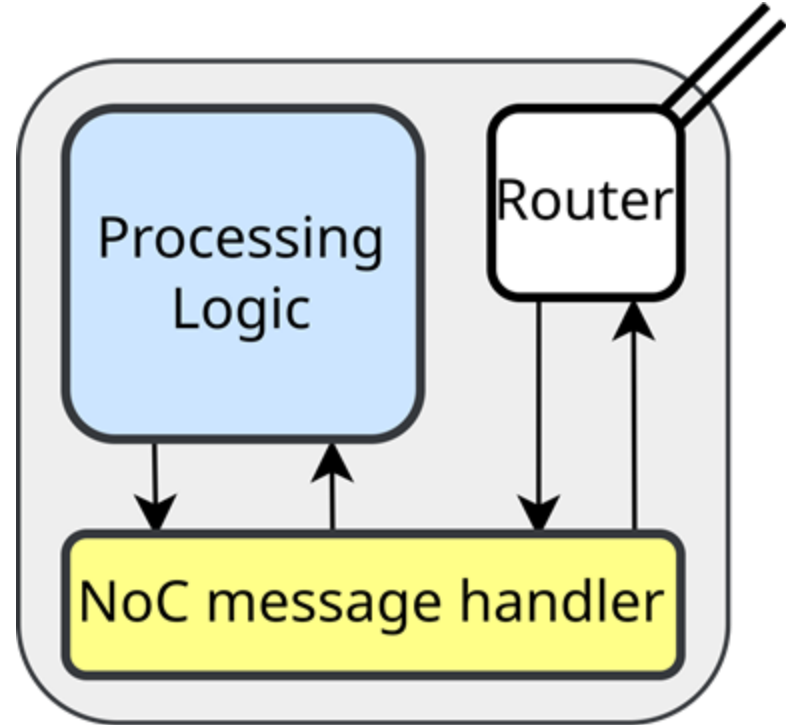
- Our proposal: Beehive, a network-on-chip (NoC) based network stack
- Each protocol or network functions is a tile. Tiles communicate via message passing and can be composed
- Scale up processing capacity by duplicating tiles within the architecture
- Focus on providing support for both flexible packet operations and reliable protocols
 - Previous work focuses on one or the other



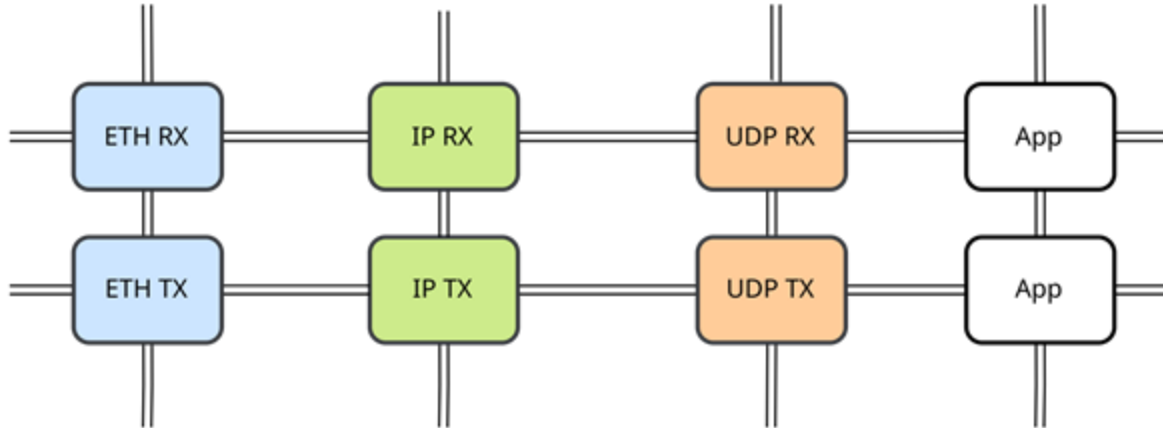
Proposed design with a mesh topology

Beehive Tile

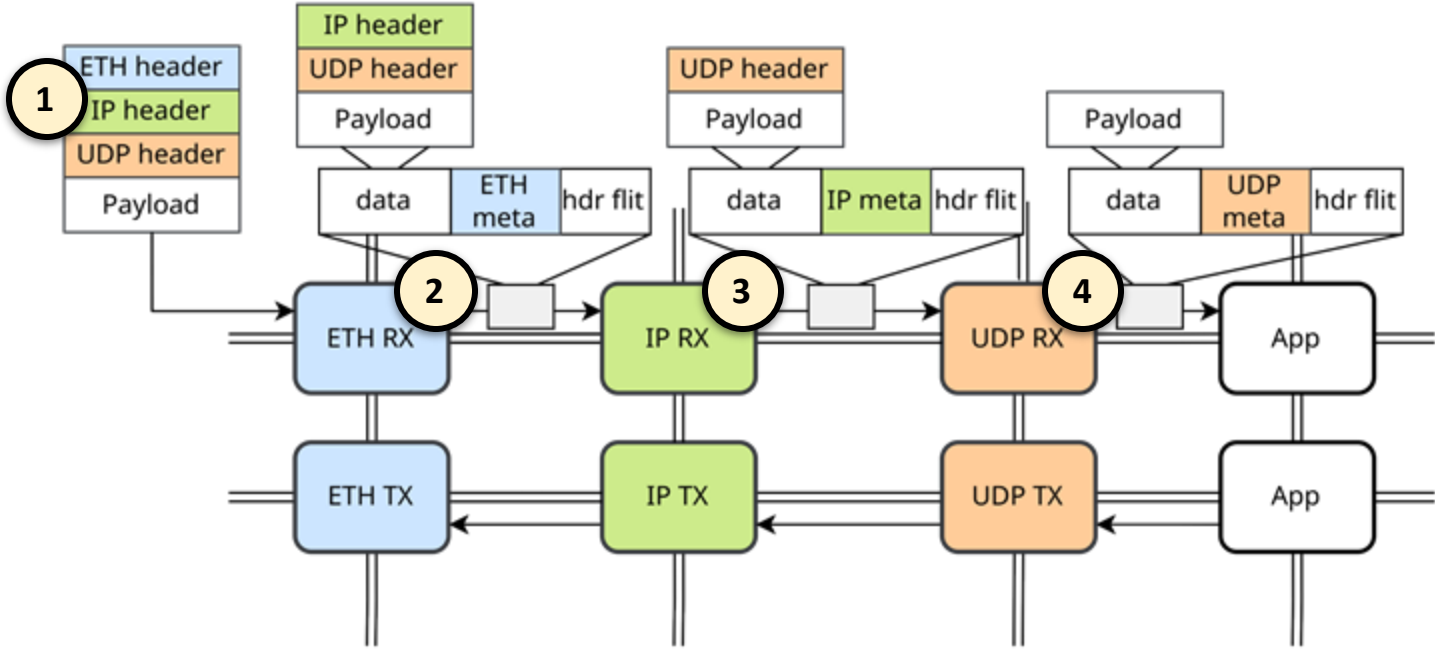
- Processing logic modules are wrapped in a tile
- Processing logic can be anything: protocol, network function, application logic
- NoC message handling includes message construction/deconstruction and network packet level routing
- Router handles NoC message level routing



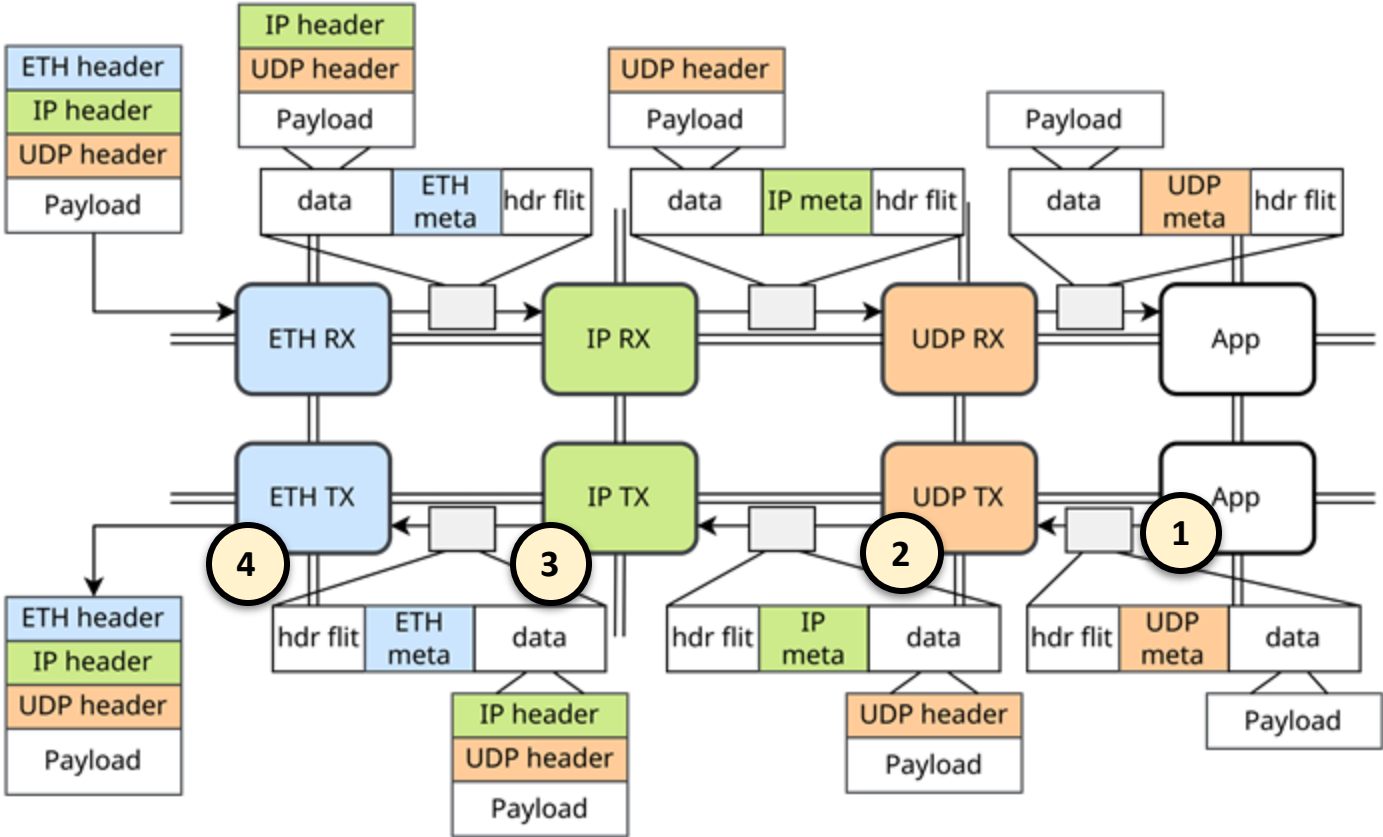
How do we process a packet?



How do we process a packet?



How do we process a packet?

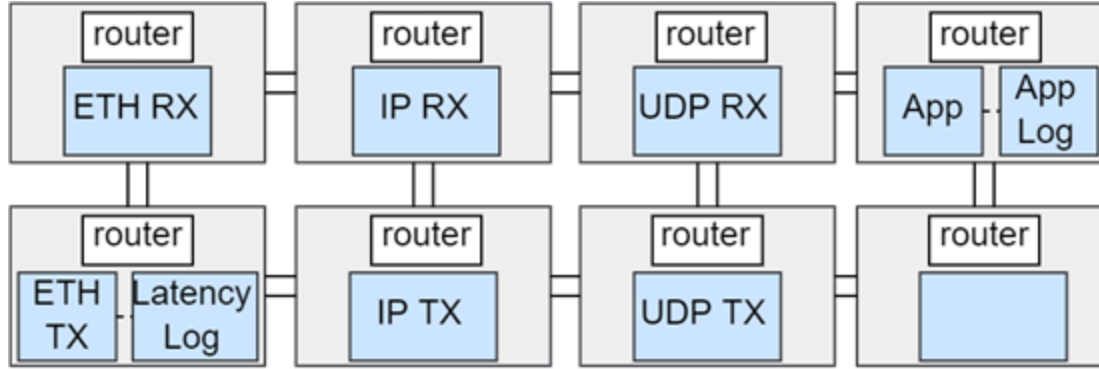


Prototype & Evaluation

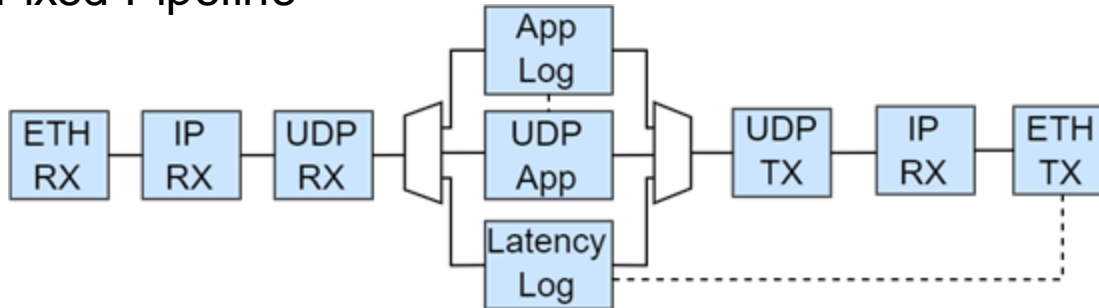
- Prototyped on Xilinx Alveo U200 running at 250 MHz
 - Mesh topology, 512 bit NoC width
 - Protocols: Ethernet, IP, TCP, UDP
 - Network functions: NAT or IP encapsulation
- Testbed
 - Switch: Edgecore Wedge 100BF-32X 100G, jumbo frames enabled
 - 3 CPU clients: 2 have Intel Xeon Gold 6226R CPUs, one has Intel Xeon Gold 5218 CPU. All have Mellanox ConnectX-5 NICs
- FPGA and CPU clients all connected to the same switch

Overhead from message passing/routing

Beehive



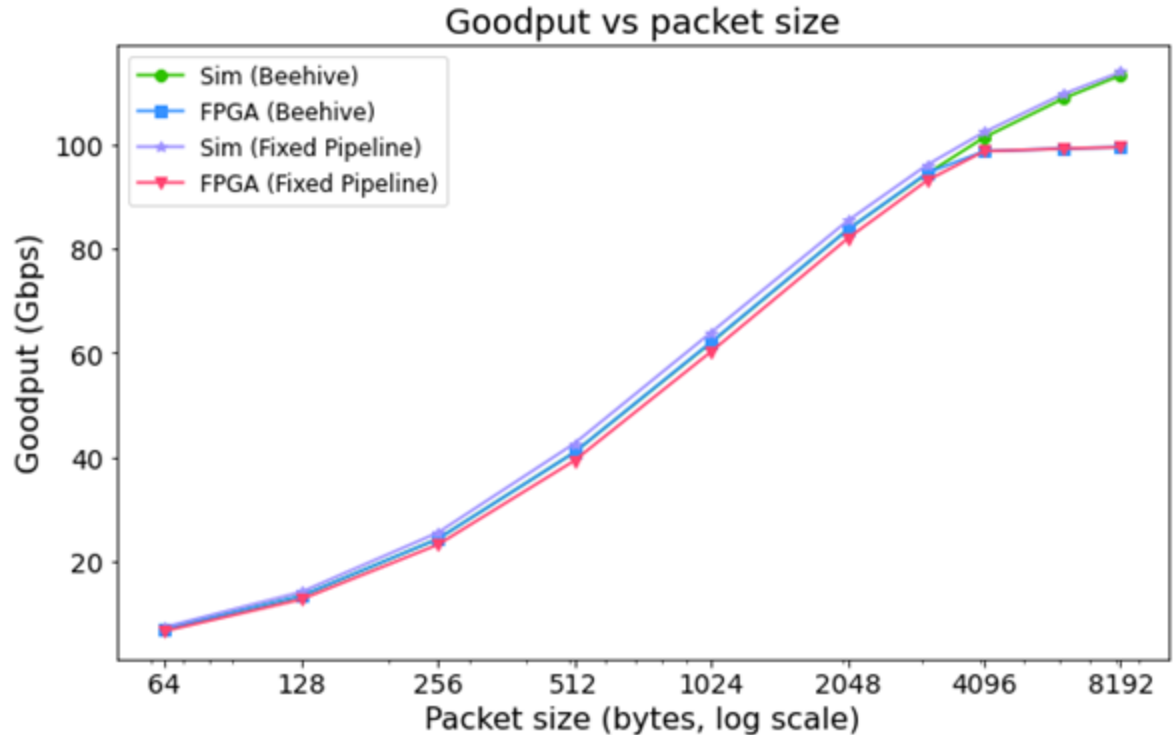
Fixed Pipeline



- Compare Beehive versus a fixed pipeline design
- Fixed pipeline uses same processing components, but no NoC infrastructure
- Integrated logs used for measuring statistics

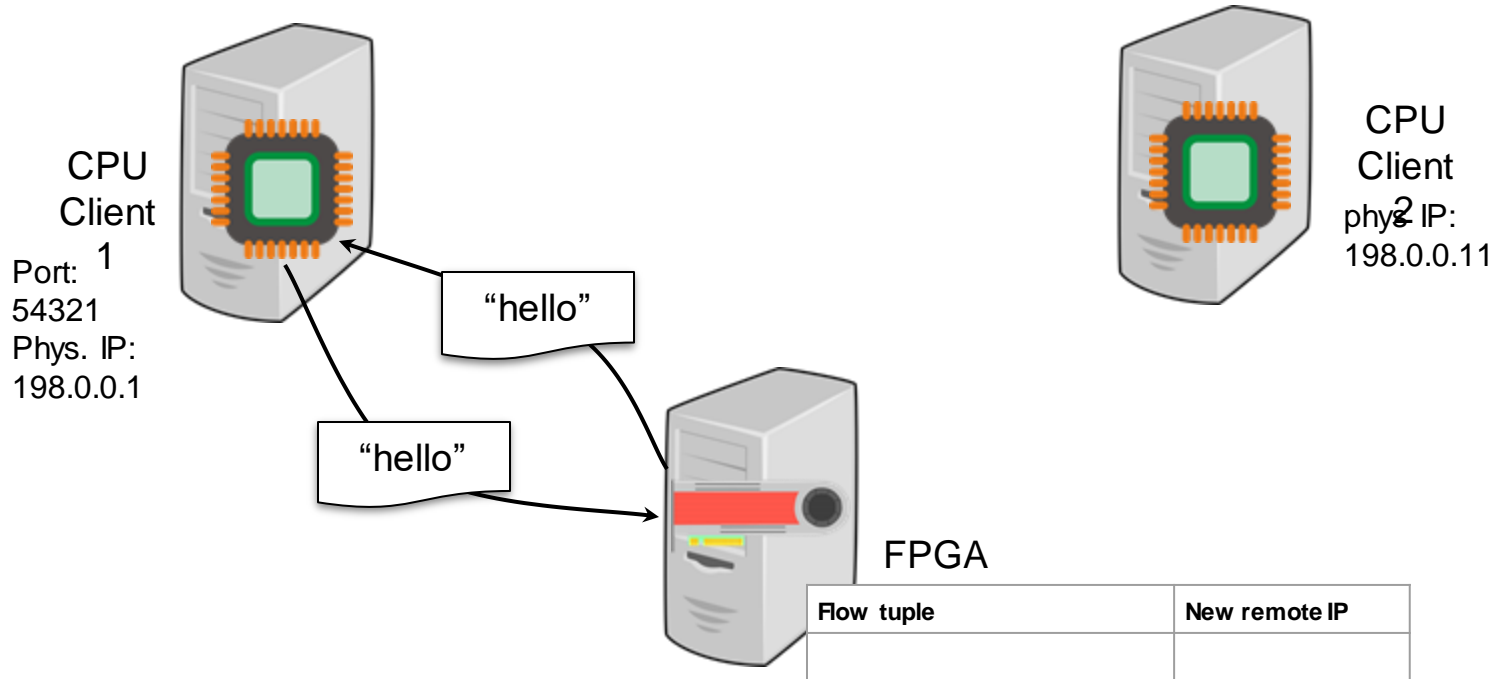
Overhead from message passing/routing

- Fixed pipeline better in simulation
- NoC has small overhead
- Beehive slightly better than fixed pipeline on FPGA due to jitter and increased buffering



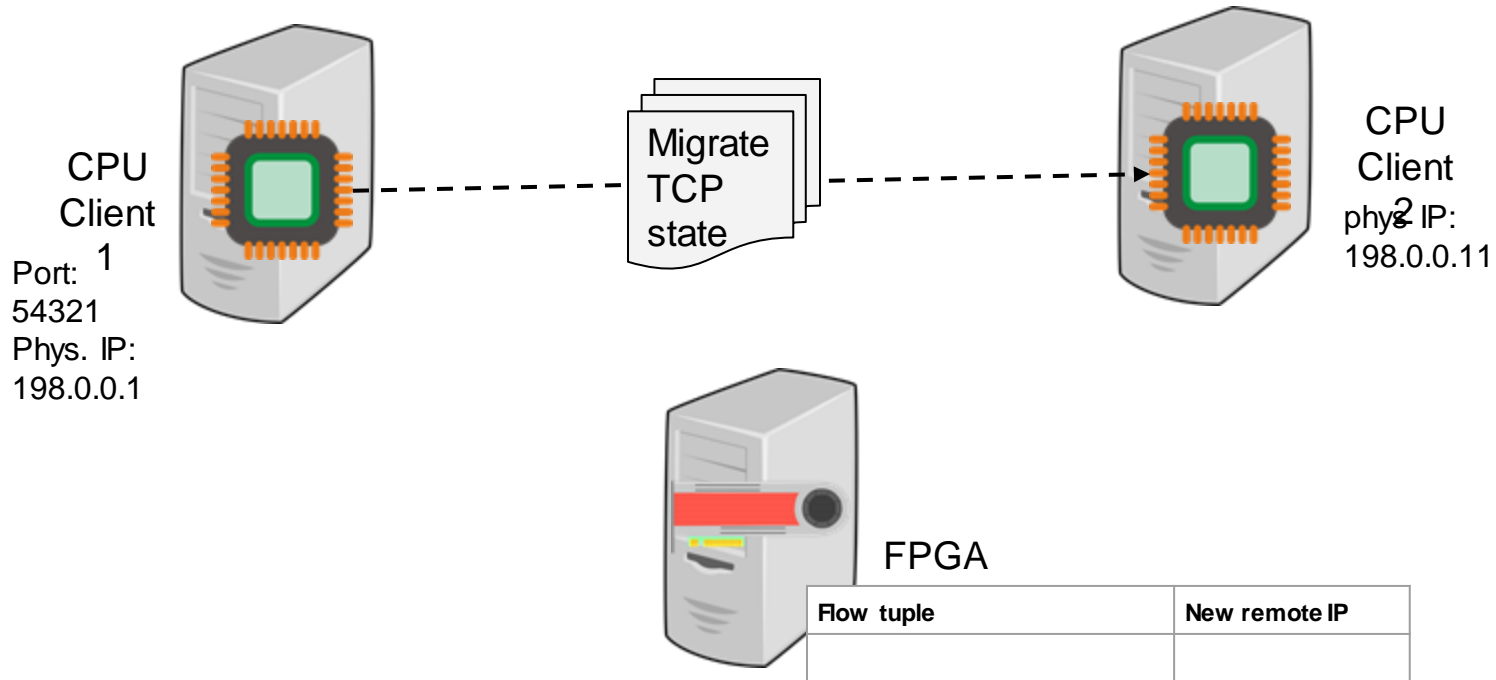
TCP migration experiment

- Migrate established TCP connection between two CPU clients using the Demikernel TCP stack without restarting the connection.



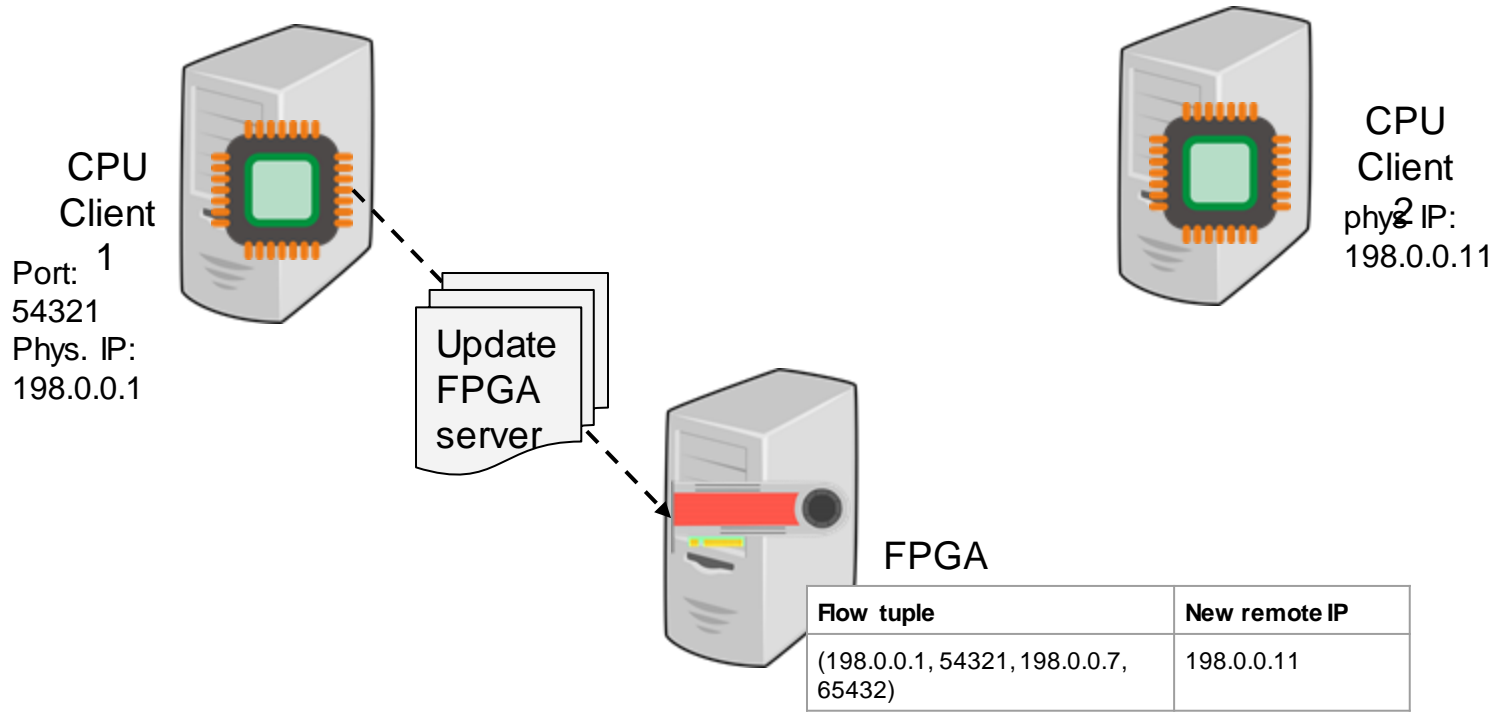
TCP migration experiment

- Migrate established TCP connection between two CPU clients running the Demikernel TCP stack without restarting the connection



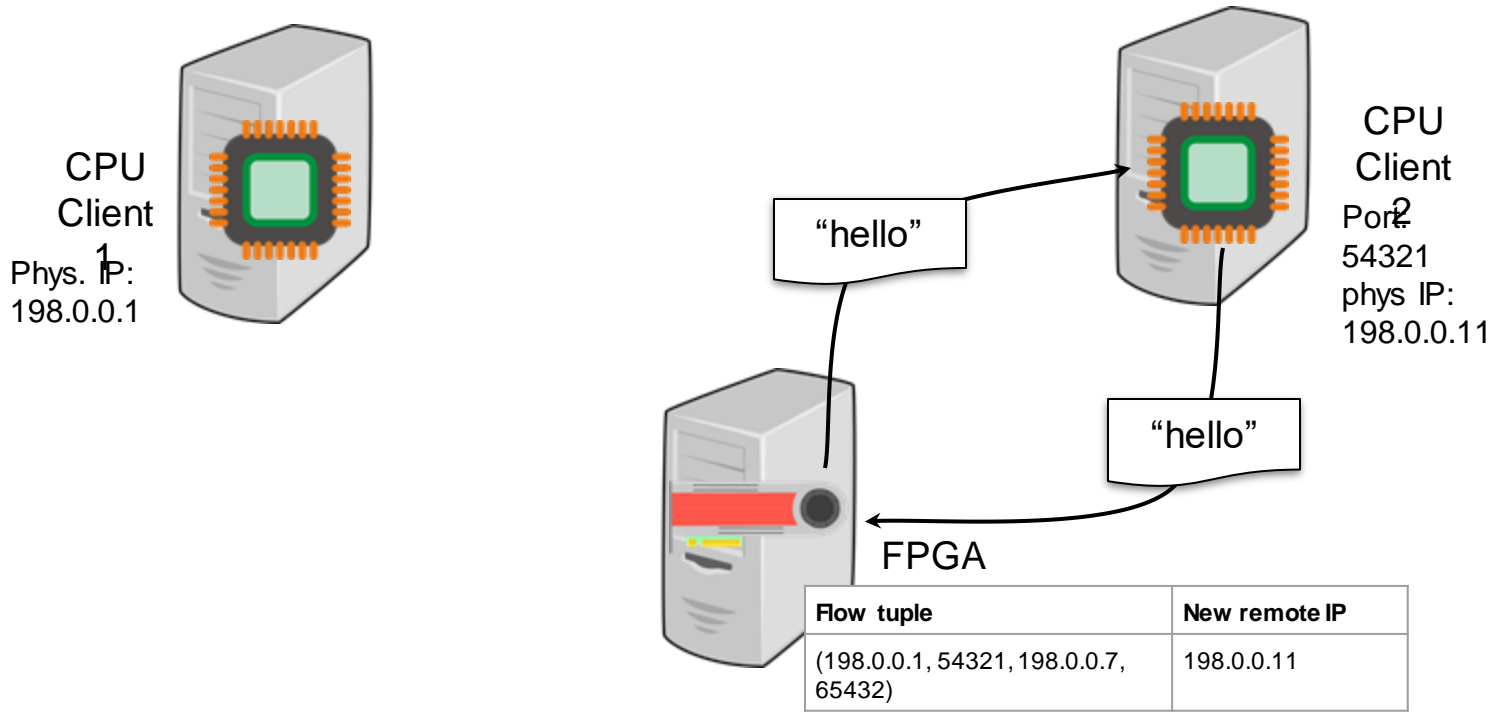
TCP migration experiment

- Migrate established TCP connection between two CPU clients running the Demikernel TCP stack without restarting the connection

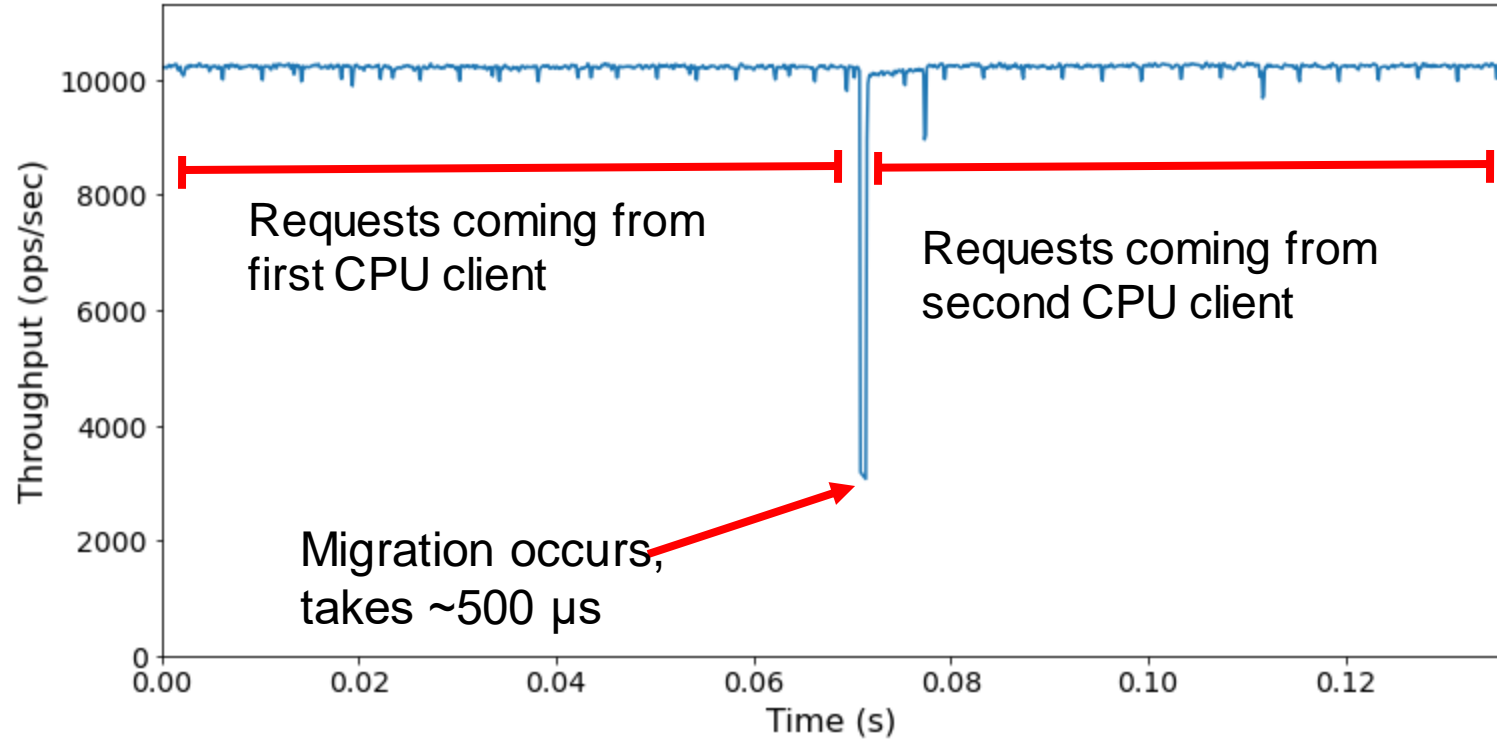


TCP migration experiment

- Migrate established TCP connection between two CPU clients running the Demikernel TCP stack without restarting the connection

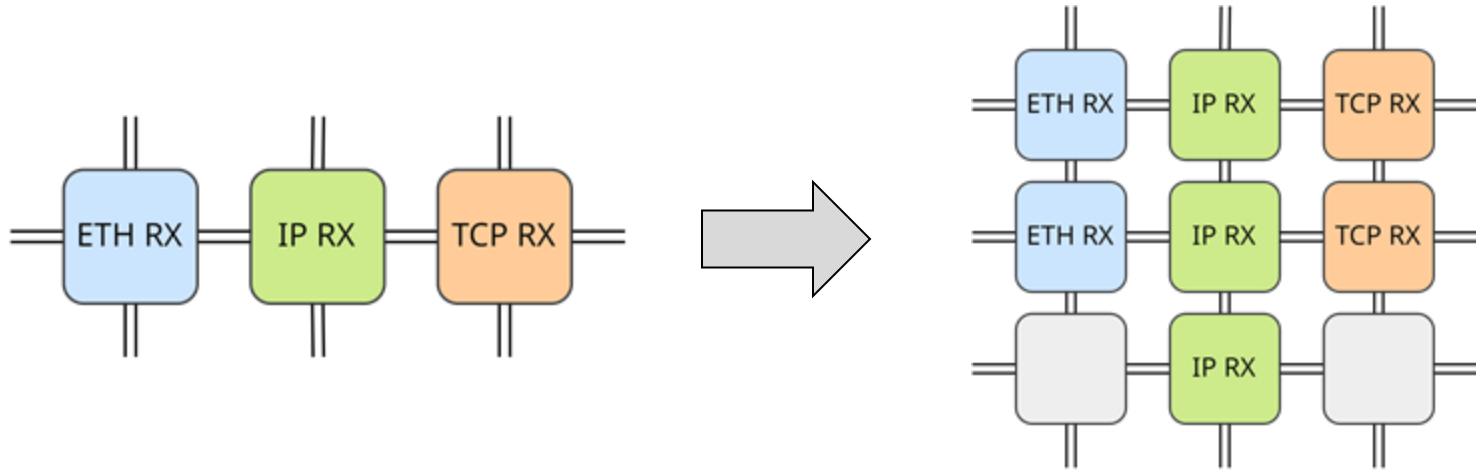


TCP migration experiment



Ongoing Work

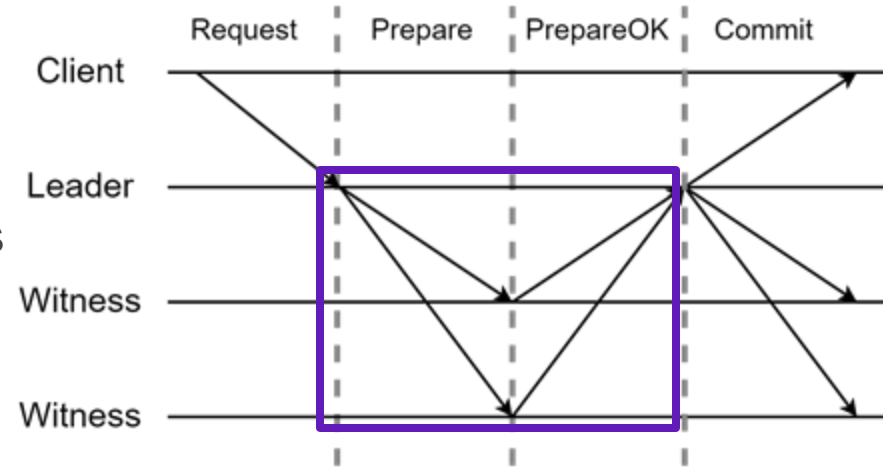
- Internal load balancing across duplicated components to support multiple instances of tiles



- Allows scaling up of processing capacity
- Requires flow-based steering to keep packets in order

Ex: Viewstamped Replication Witness

- Consensus algorithms allow agreement on an order of operations and are important for building replicated, distributed systems
- Each consensus round requires leaders collect responses from a majority of witnesses
- Can we accelerate the witnesses in hardware?



Normal operation of the VR protocol

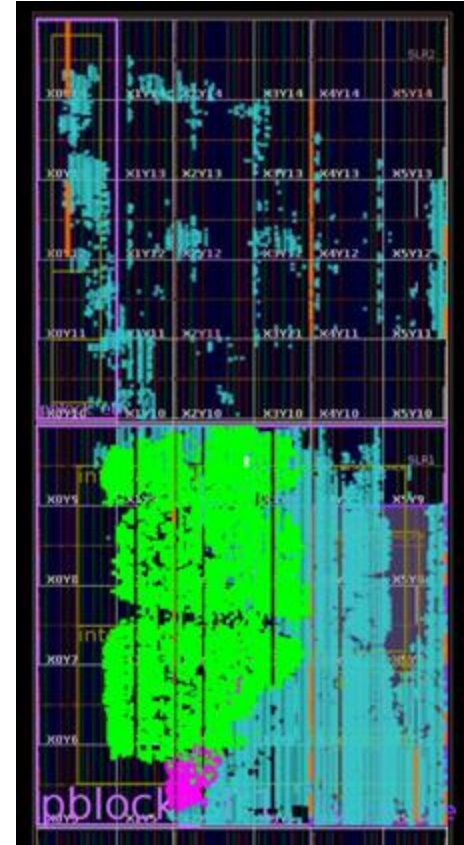
Why witnesses?

- In a typical, only CPU case, one node can be either leader or witness
- What are witnesses responsible for?
 - During typical processing: verifying that proposals carry the correct view number, a valid operation number
 - During failures: can initiate recovery, but also can just respond appropriately to view change messages
- Advantageous for hardware:
 - They do not need to execute application logic
 - Low latency good for achieving quick quorums
 - Messages are typically small packets

Preliminary Results

- Latency: 64ns per consensus round
- Bandwidth: 15.6 Mrounds/sec, ~2Gbps
- Implemented on Xilinx Alveo U200 with Vivado 2021.2 at 250 MHz
- Utilization promising that we can replicate tile to scale up processing bandwidth

Utilization of 1 VR Witness Tile	
LUTs	1918
URAMs	4



Conclusion

- Built Beehive, a NoC-based network stack designed to be modular and support complex network functionality
- Demonstrated that Beehive has a small overhead on bandwidth (~5%) versus a fixed pipeline design while enabling complex functionality like TCP connection migration
- Working on leveraging tile-based design to scale up processing with VR witness example application