HELIX: Co-designing the Hardware, Software and Network Protocol for Reliable High-Bandwidth Communication in Constrained Systems

Leon Schuermann and Amit Levy
Princeton University

Frank Duerr
University of Stuttgart

PRINCETON UNIVERSITY

University of Stuttgart

Project Overview

We present HELIX (*Hardware-Enabled Lossless Internet Information eXchange*), an ongoing research project enabling constrained computer systems to reliably exchange high-bandwidth data streams over commodity IP networks.

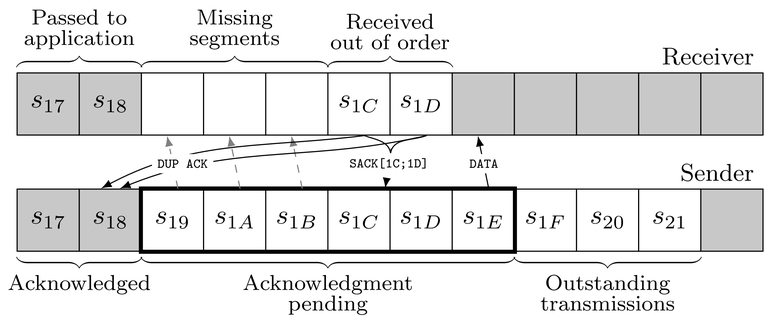
➡ HELIX introduces a new memory model which is deeply integrated into the network protocol: it transmits addresses referencing receiver memory alongside data frames and acknowledgments. This simplifies the logic on the transmitter for processing acknowledgments or retransmissions.

➡ HELIX further embraces hardware-software co-design to offload complex management tasks and congestion control to an integrated soft-core CPU, reducing the implementation's logic complexity.

By co-designing the network protocol along with its hardware and software implementation, our system demonstrates the advantages of such a holistic approach.

Limitations of TCP

TCP's *receive sliding window* indicates available buffer space to the sender. It *slides* forward when acknowledgments (ACKs) are received. Lost segments are indicated by *Duplicate ACKs* or *Selective ACKs*:



Problem: the window cannot move past non-acknowledged segments (*head-of-line blocking*). TCP's byte-addressed segments require linear buffer data structures (e.g., ring buffers). Together, this prevents reclaiming memory from selective acknowledgments.

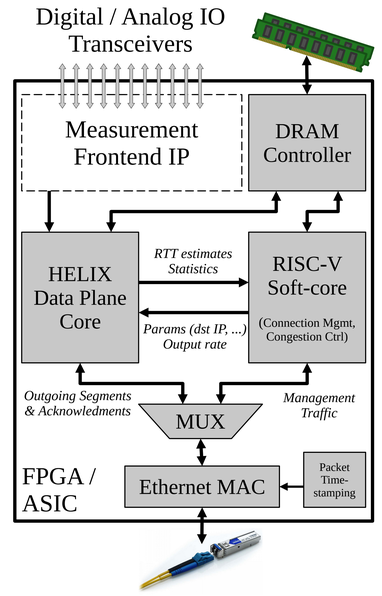
HW/SW Co-design

HELIX circumvents hardware complexities for connection management & benefits from existing research in congestion control (CC) by integrating with a "soft-core" CPU.

➡ RTT-aware CC algorithms with hardware timestamping.

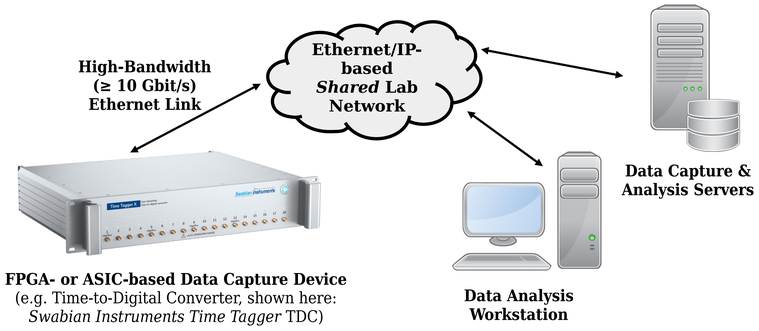
➡ Implement connection state machine & all management traffic in SW to keep HW complexity low.

➡ Soft-core CPU usable for general hardware design control as well.



Application Areas

HELIX is designed to work with constrained hardware systems and integrate into existing infrastructures:



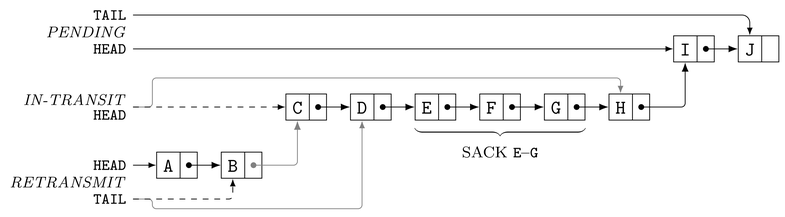
➡ Limited on-device memory and compute resources.

➡ Inherent and environmental constraints (noise level, thermal stability, and economical factors) make integration of general-purpose CPUs infeasible.

➡ FPGAs & ASICs are well-suited for high-speed Ethernet through wide datapaths & highly parallel data processing.

Network-Integrated Memory Model

HELIX proposes a new, linked-list based memory model:



Through use of fixed-size segments and a segmented linked list, we design a *hardware-walkable* data structure allowing significant optimizations:

➡ Non-contiguous nature allows re-use of memory occupied by acknowledged segments and interleaving of multiple buffers.

➡ List-transformations don't change memory locations of existing segments, allowing HELIX to **transmit memory addresses alongside segments**. This eliminates memory management complexity at the sender: for example, ACKs include pointers into transmitter memory, enabling $O(1)$ indexing.

Evaluation Plan

We plan to evaluate and optimize HELIX in the following dimensions:

➡ Memory utilization and overheads at sender and receiver.

➡ Suitability of its memory model for software-based systems, both as a receiver and as a transmitter (overheads incurred through linked list structure, data stream reassembly, etc).

➡ Compatibility with existing congestion control algorithms and traffic fairness.

➡ Assumptions on the underlying network behavior and end-hosts.

We implemented a proof-of-concept hardware design of the proposed network-integrated memory model in approx. 3kLOC of the Migen hardware description language. It has been verified to deliver a goodput of 23.2 Gbit/s in a sustained read and write (worst-case) workload using the DRAM of a NetFPGA-SUME development board.