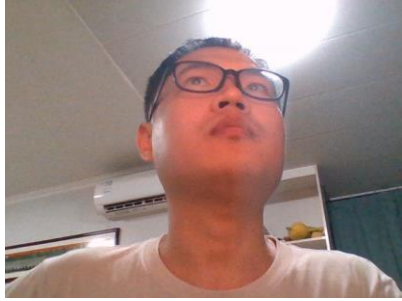
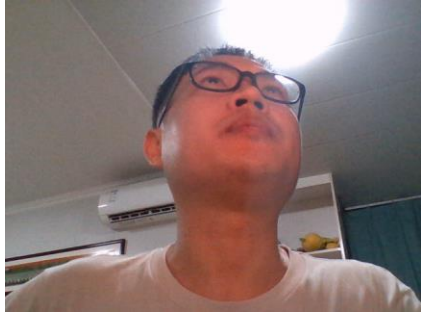


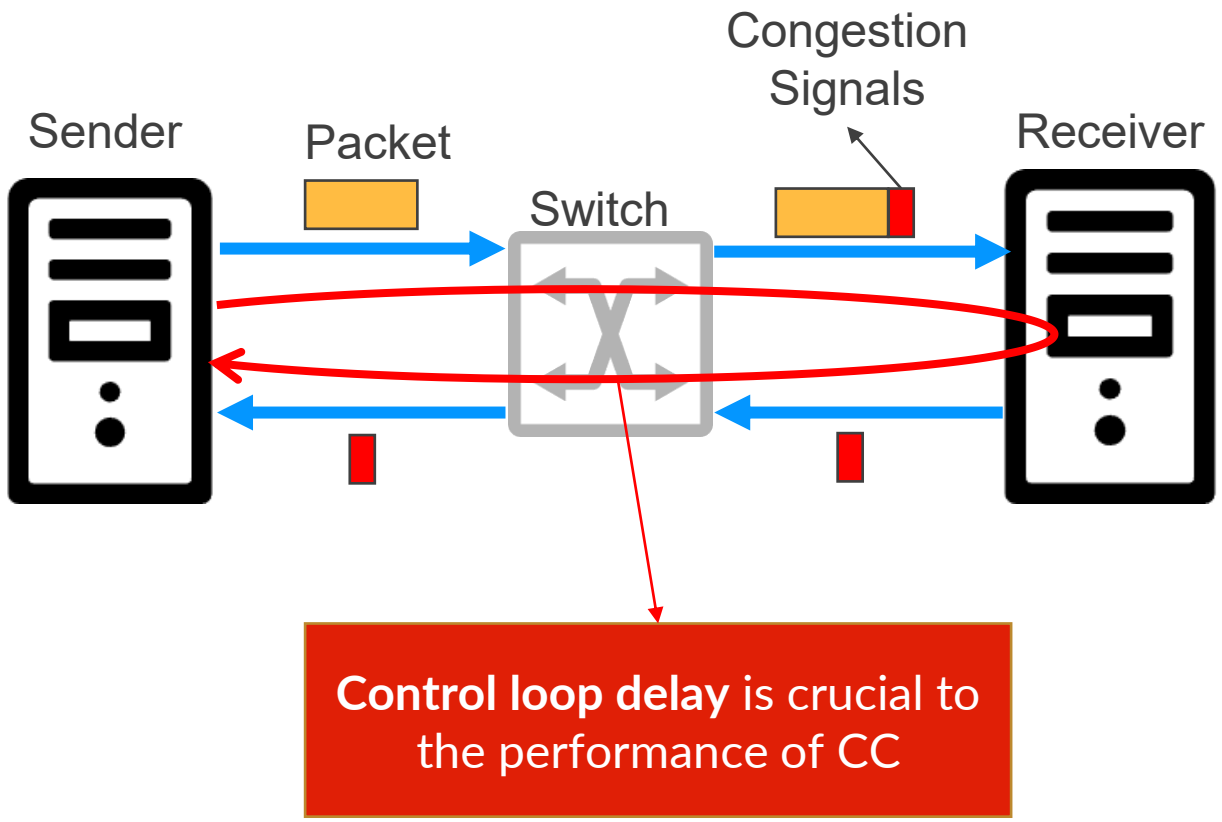
SwCC: Software-Programmable and Per-Packet Congestion Control in RDMA Engine

Hongjing Huang, *Jie Zhang, Xuzheng Chen, Ziyu Song, Jiajun Qin, Zeke Wang*

ZHEJIANG UNIVERSITY

Congestion Control





Continous Innovation in Congestion Control

Congestion Control for Large-Scale RDMA Deployments

TIMELY: RTT-based Congestion Control for the Datacenter

dhika Mittal*(UC Berkeley), Vinh The Lam, Nandita Dukkipati, Emily Blem, Hassan Wassel,

pHost: Distributed Near-Optimal Datacenter Transport Over Commodity Network Fabric

Peter X. Gao
petergao@berkeley.edu

Akshay Narayan
akshay@berkeley.edu

Swift: Delay is Simple and Effective for Congestion Control the Datacenter

on Jang (MPI-SWS)*, Hassan M. G. Wassel, Xian Wu, Behnam Montazeri
Christopher Alfeld, Michael Ryan, David Wetherall, and Amin Vahdat

Bolt: Sub-RTT Congestion Control for Ultra-Low Latency

Serhat Arslan*

Yuliang Li

ACC: Automatic ECN Tuning for High-Speed Datacenter Networks

Host Congestion Control

Saksham Agarwal
Cornell University

Arvind Krishnamurthy
Google & University of Washington

POWERTCP: Pushing the Performance Limits of Datacenter Networks*

Oliver Michel
Princeton University
University of Vienna

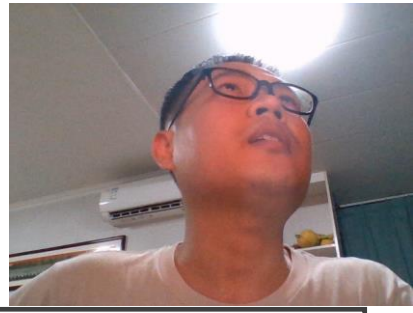
Stefan Schmid
TU Berlin
University of Vienna

Poseidon: Efficient, Robust, and Practical Datacenter CC via Deployable INT

Weitao Wang*†, Masoud Moshref*, Yuliang Li*, Gautam Kumar*,
T. S. Eugene Ng†, Neal Cardwell*, and Nandita Dukkipati*

*Google LLC, †Rice University

Motivation

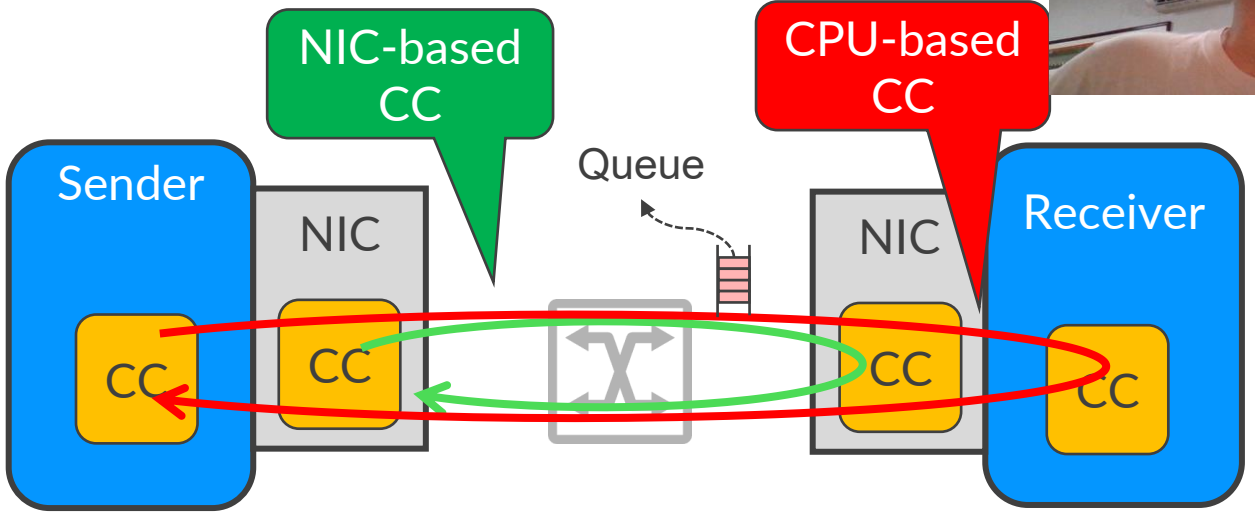


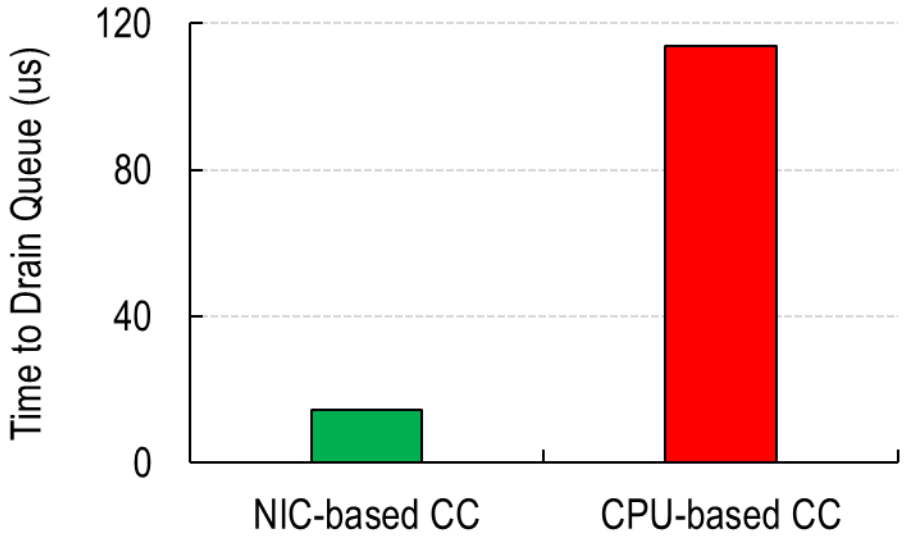
	CPU-based CC	ASIC NIC-based CC	FPGA SmartNIC-based CC	SoC SmartNIC-based CC
Control Loop Delay				
Flexibility				
Programmability				

Motivation

CPU-based CC

Control Loop Delay	**High** ☹
Flexibility	**High** ☺
Programmability	**High** ☺

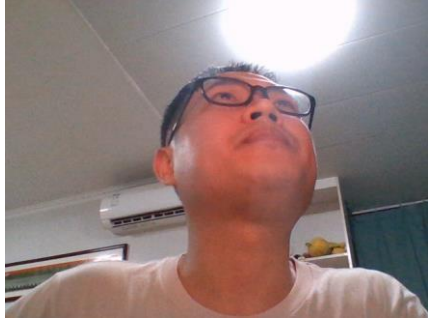




Motivation

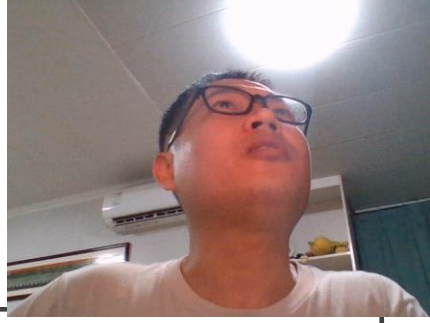
	CPU-based CC	ASIC NIC-based CC
Control Loop Delay	High ☹	Low ☺
Flexibility	High ☺	Low ☹
Programmability	High ☺	Low ☹

Motivation



	CPU-based CC	ASIC NIC-based CC	FPGA SmartNIC-based CC
Control Loop Delay	High ☹	Low ☺	Low ☺
Flexibility	Hi	More engineering effort	High ☺
Programmability	Hi	Long Development Cycles	Low 😐

Motivation



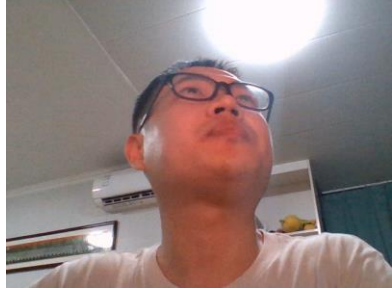
	CPU-based CC	ASIC NIC-based CC	FPGA SmartNIC-based CC	SoC SmartNIC-based CC
Control Loop Delay			Poor cache/memory performance for DPA → PCC can't support per-packet CCAs	Low
Flexibility			❑ 10/300 ns L1 cache/memory latency, which is 10.5×/3× that of the host's latency. ❑ PCC only supports rate-based CCAs	Low
Programmability			❑ at a 100Gbps line rate, a 1 KB packet arrives every 89 ns	High

Goal

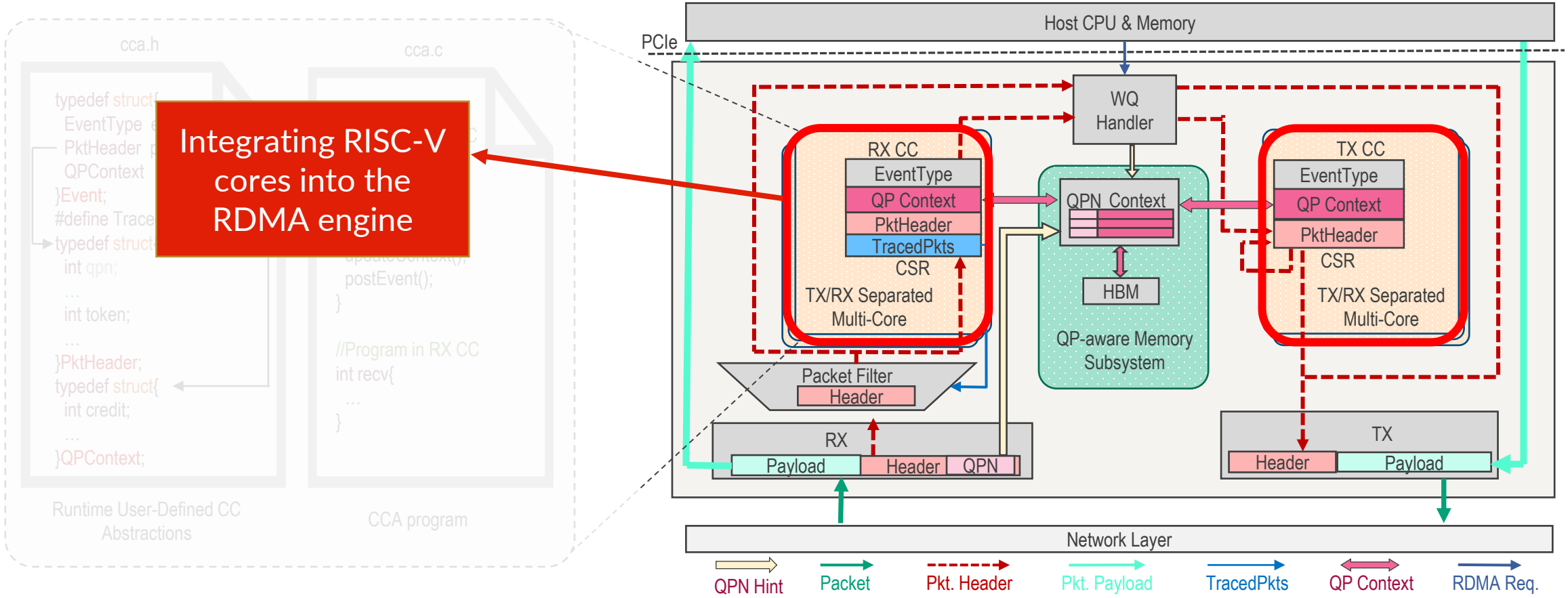
SwCC: a software-programmable and per-packet congestion control in RDMA Engine

- Low control loop delay
- Short CC controller triggering interval
- High flexibility
- High programmability

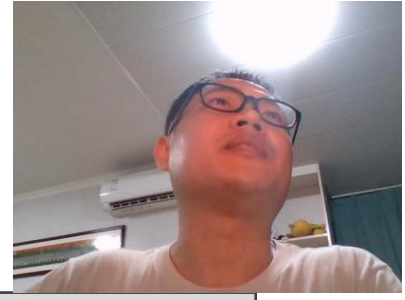
How to Keep Low Control Loop Delay



Integrating RISC-V cores into the RDMA engine

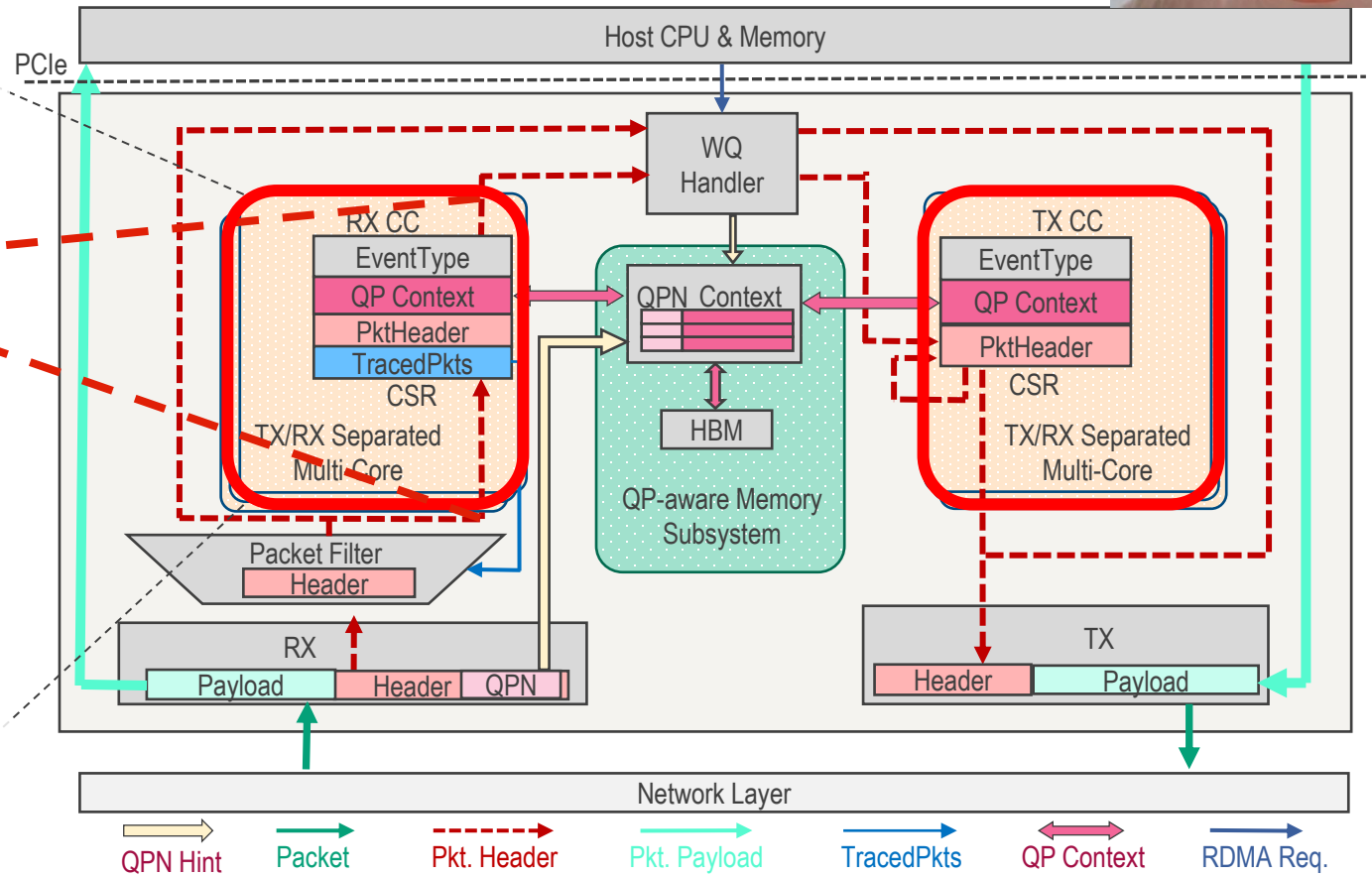


How to Keep a Short CC Controller Triggering Interval



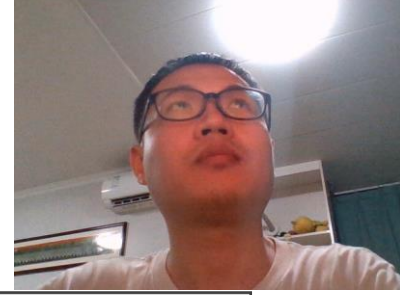
A single core only deal with a subset of CC logic

Increase the CC controller hardware frequency as much as possible



- TX/RX Separated Multi-Core

How to Keep a Short CC Controller Triggering Interval

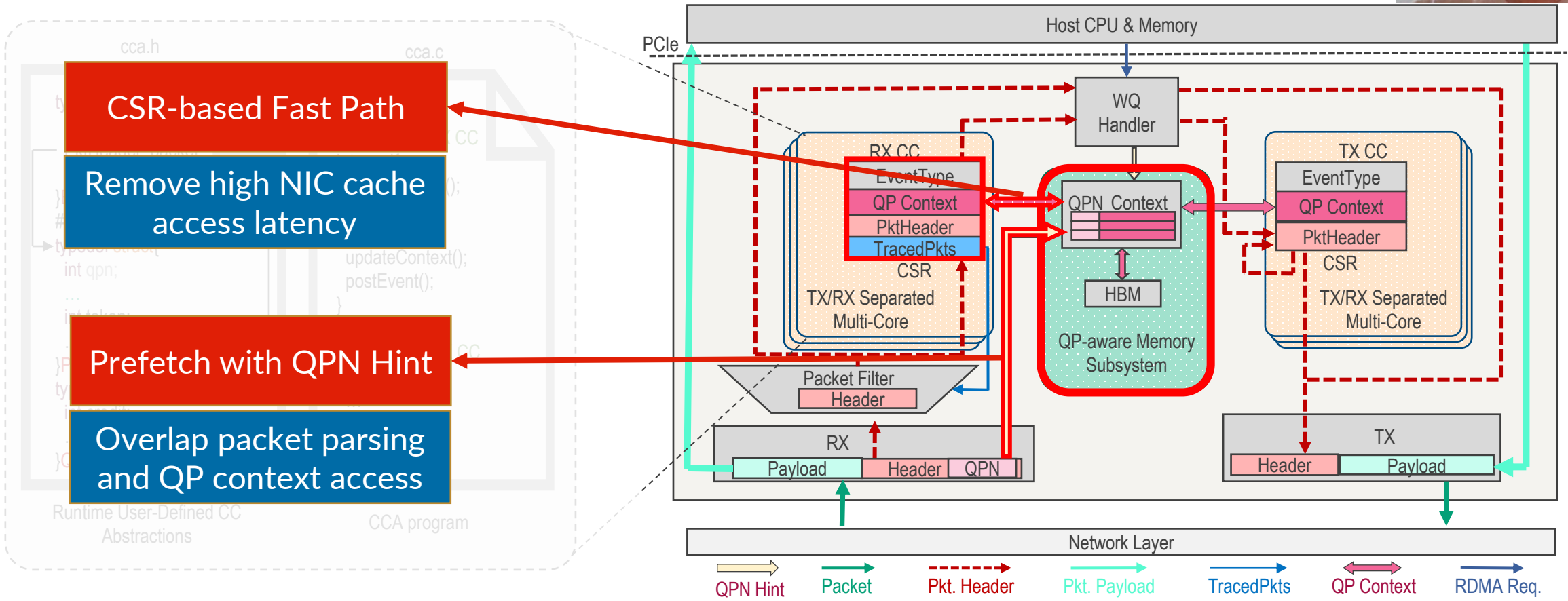


CSR-based Fast Path

Remove high NIC cache access latency

Prefetch with QPN Hint

Overlap packet parsing and QP context access



- TX/RX Separated Multi-Core
- QP-aware Memory Subsystem

How to Keep High Flexibility

cca.h

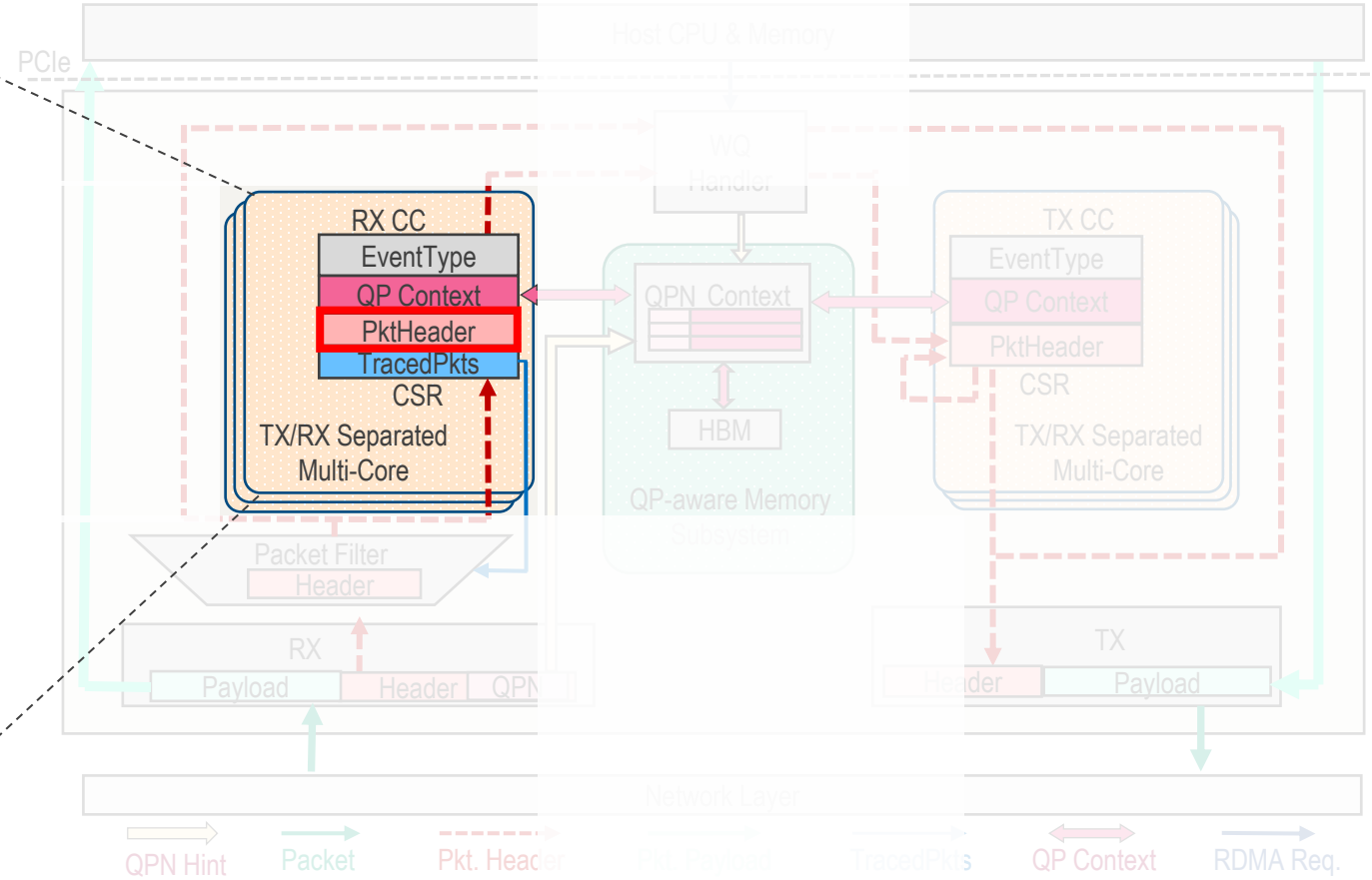
```
typedef struct{
  EventType  event;
  PktHeader  packet;
  QPContext   context;
}Event;
#define TracedPkts ACK
typedef struct{
  int qpn;
  …
  int token;
  …
}PktHeader;
typedef struct{
  int credit;
  …
}QPContext;
```

Runtime User-Defined CC Abstractions

cca.c

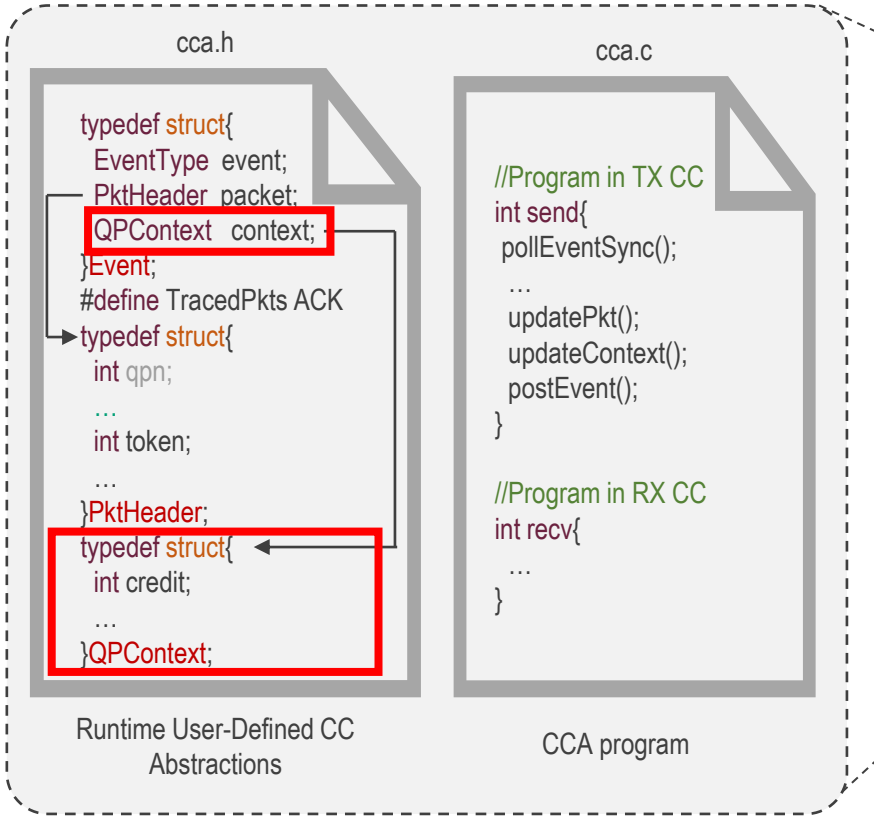
```
//Program in TX CC
int send{
 pollEventSync();
  …
  updatePkt();
  updateContext();
  postEvent();
}

//Program in RX CC
int recv{
  …
}
```

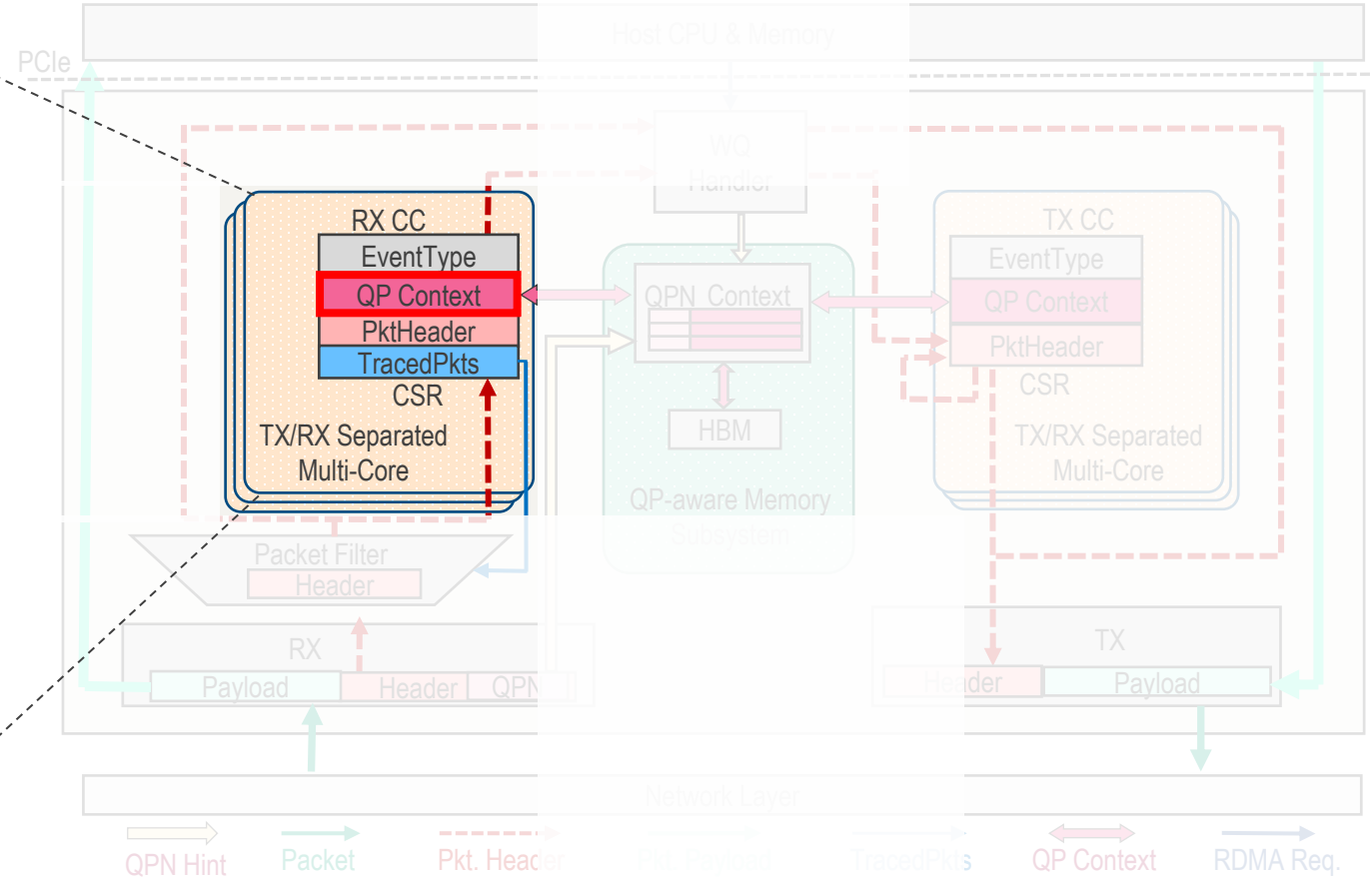
CCA program



- Extensible CC Header

How to Keep High Flexibility





- Extensible CC Header
- User-defined QP Context

How to Keep High Flexibility

cca.h

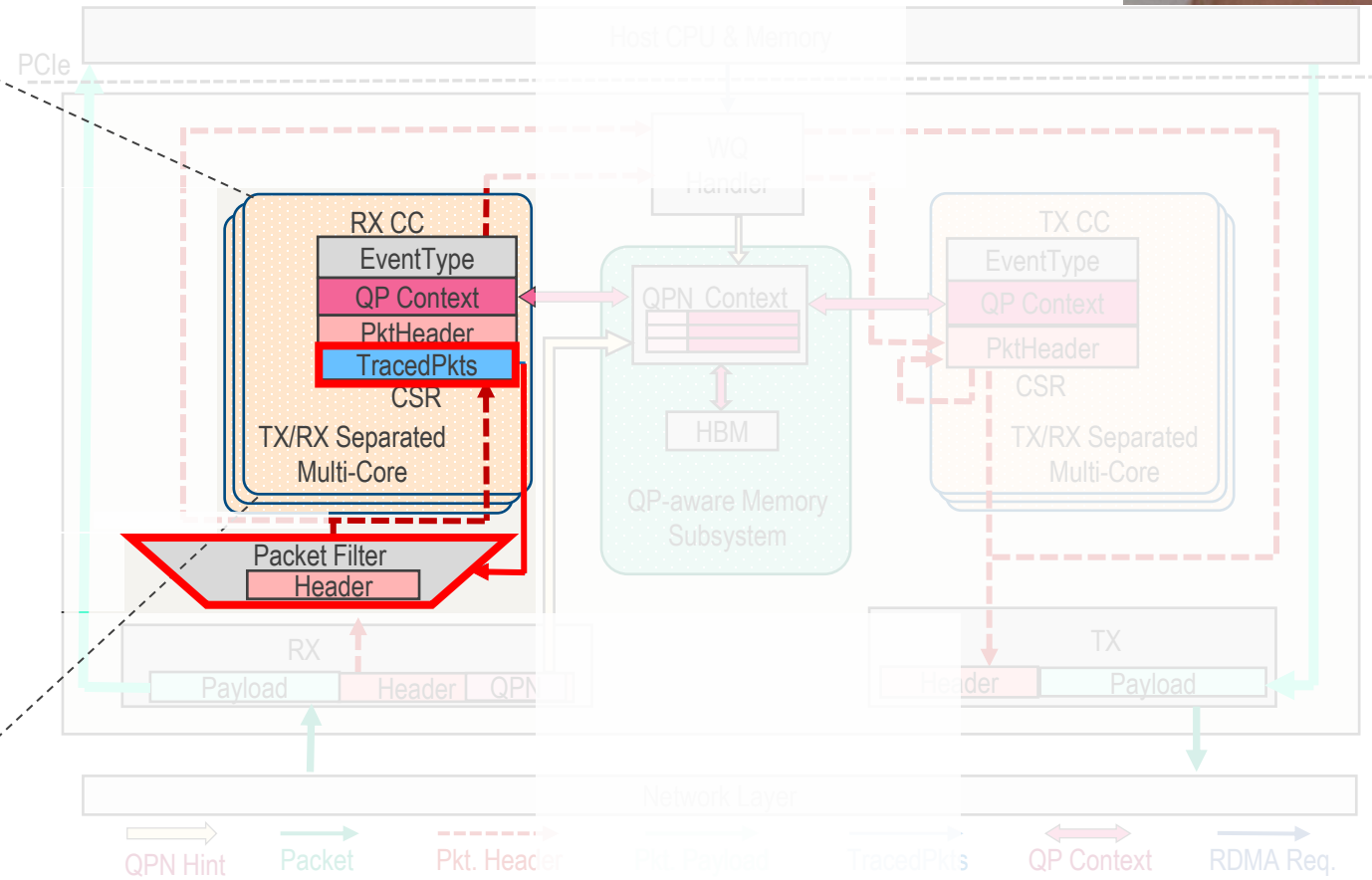
```
typedef struct{
  EventType  event;
  PktHeader  packet;
  QPContext   context;
}Event;
#define TracedPkts ACK
typedef struct{
  int qpn;
  …
  int token;
  …
}PktHeader;
typedef struct{
  int credit;
  …
}QPContext;
```

Runtime User-Defined CC Abstractions

cca.c

```
//Program in TX CC
int send{
 pollEventSync();
  …
  updatePkt();
  updateContext();
  postEvent();
}

//Program in RX CC
int recv{
  …
}
```

CCA program



- Extensible CC Header
- User-defined QP Context
- Selective Triggering

How to Keep High Programmability

cca.h

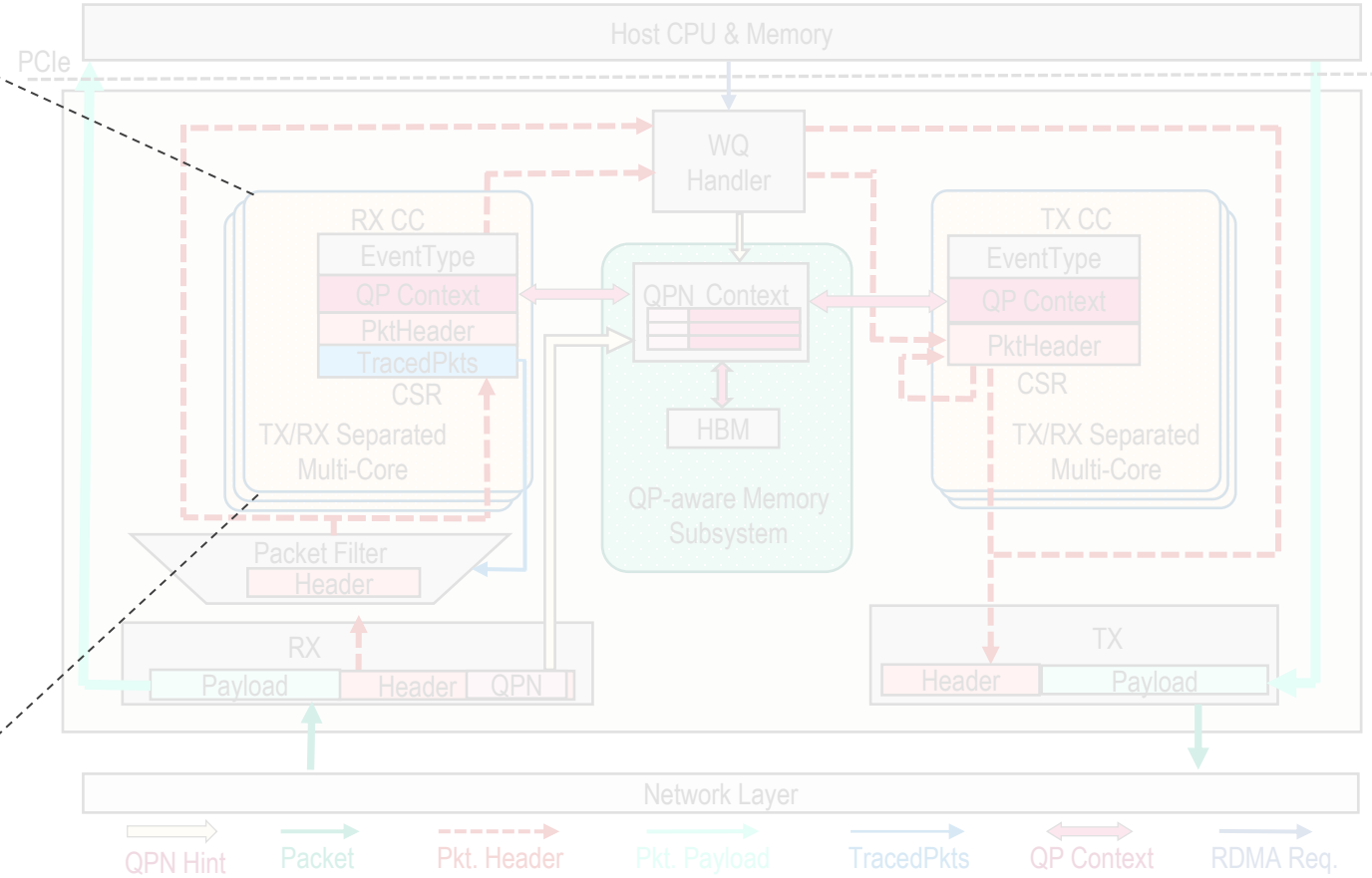
```
typedef struct{
  EventType  event;
  PktHeader  packet;
  QPContext   context;
}Event;
#define TracedPkts ACK
typedef struct{
  int qpn;
  …
  int token;
  …
}PktHeader;
typedef struct{
  int credit;
  …
}QPContext;
```

Runtime User-Defined CC Abstractions

cca.c

```
//Program in TX CC
int send{
 pollEventSync();
  …
  updatePkt();
  updateContext();
  postEvent();
}

//Program in RX CC
int recv{
  …
}
```

CCA program

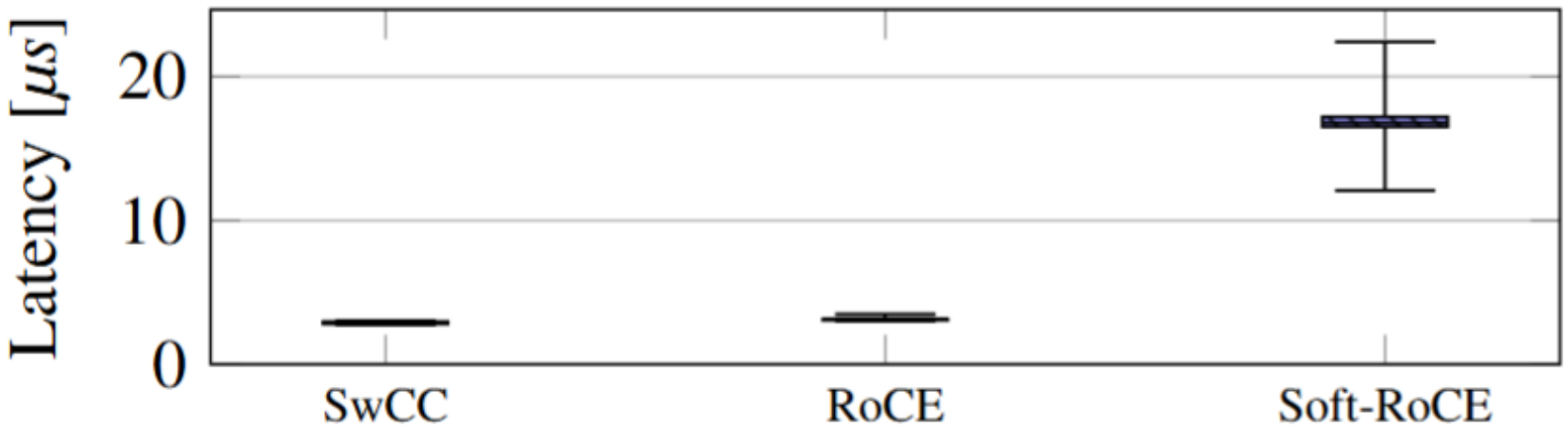


- easy-to-use programming APIs

Experimental Setup

- **SwCC**
  - Embed RISC-V in RDMA engine
- **RoCE**
  - AISC-based NIC CC solution
- **Soft-RoCE**
  - CPU-based CC solution
- **BF3**
  - SoC-based SmartNIC CC solution

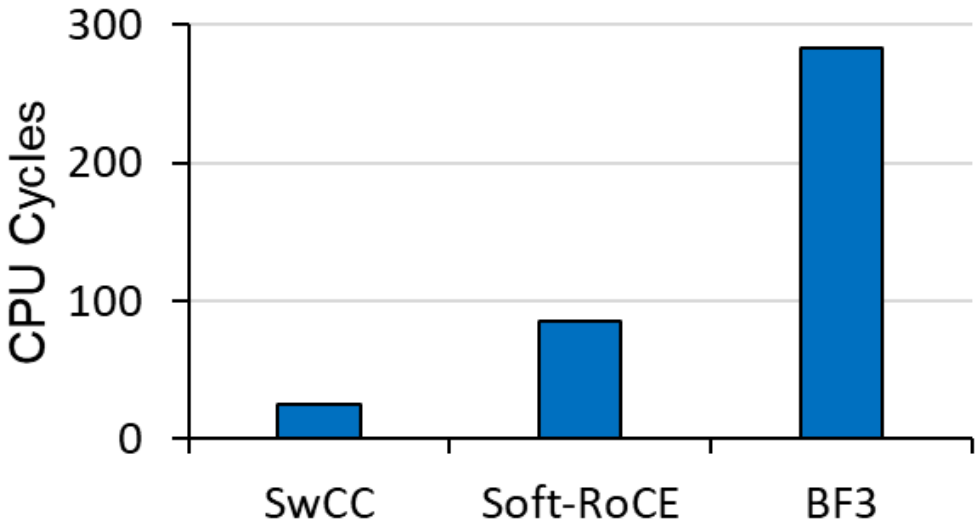
Comparison of Control Loop Delay



The RTT control loop delay

- SwCC and RoCE achieve a similar control loop delay of 3.1 μs
- The control loop delay of Soft-RoCE is 6x that of SwCC.

Comparison of CC Controller Triggering Interval



The cycles of CC controller

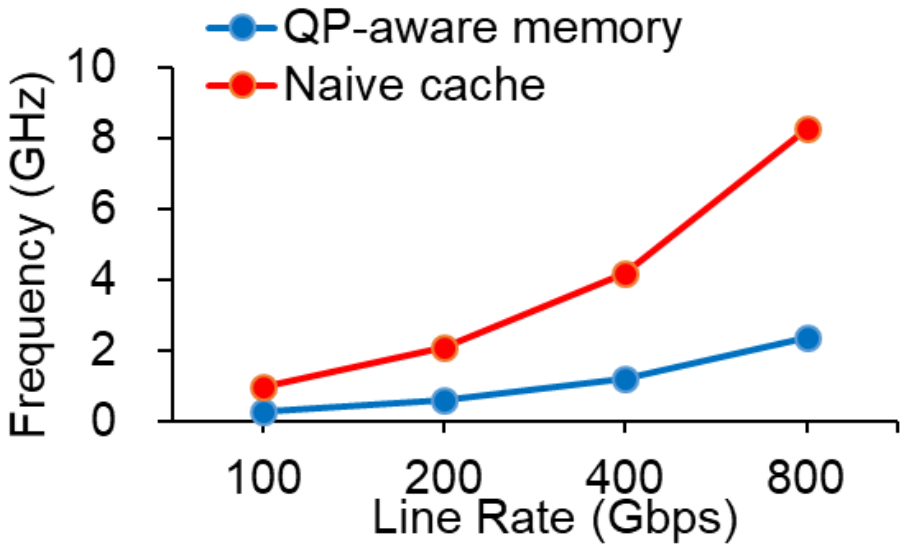
- SwCC reduces CPU cycles by 91% compared to BF3.

Flexibility and Programmability of SwCC

CCA	CC Signal	Triggering Event	Adjustment Strategy	C code (lines)
DCQCN	ECN	CNP, DATA, Timer	rate	140
TIMELY	timestamp	ACK	rate	102
HPCC	INT	ACK	wnd	148
Swift	timestamp	ACK	wnd	164
Homa	token	DATA, GRANT, RESEND, BUSY	credit	95

- SwCC supports a broad range of CCAs.
- SwCC implements CCAs with less than 200 lines of code

Potential ASIC Design.



The frequency required to achieve various line rates

- QP-aware memory needs only 2.4 GHz to reach an 800 Gbps line rate

Summary

SwCC:

1. Embedding RISC-V cores in the NIC hardware to achieve low control loop delay.
2. TX/RX separated multi-core and QP-aware memory subsystem to achieve short CC controller triggering interval.
3. Extensible CC header and Selective Triggering to achieve high flexibility.
4. A set of programming APIs to achieve high programmability.

Thank you!

SwCC: Software-Programmable and Per-Packet Congestion Control in RDMA Engine

Hongjing Huang, *Jie Zhang, Xuzheng Chen, Ziyu Song, Jiajun Qin, Zeke Wang*

Email us at: huang_hj@zju.edu.cn
Open-sourced at: https://github.com/RC4ML/SwCC

ZHEJIANG UNIVERSITY