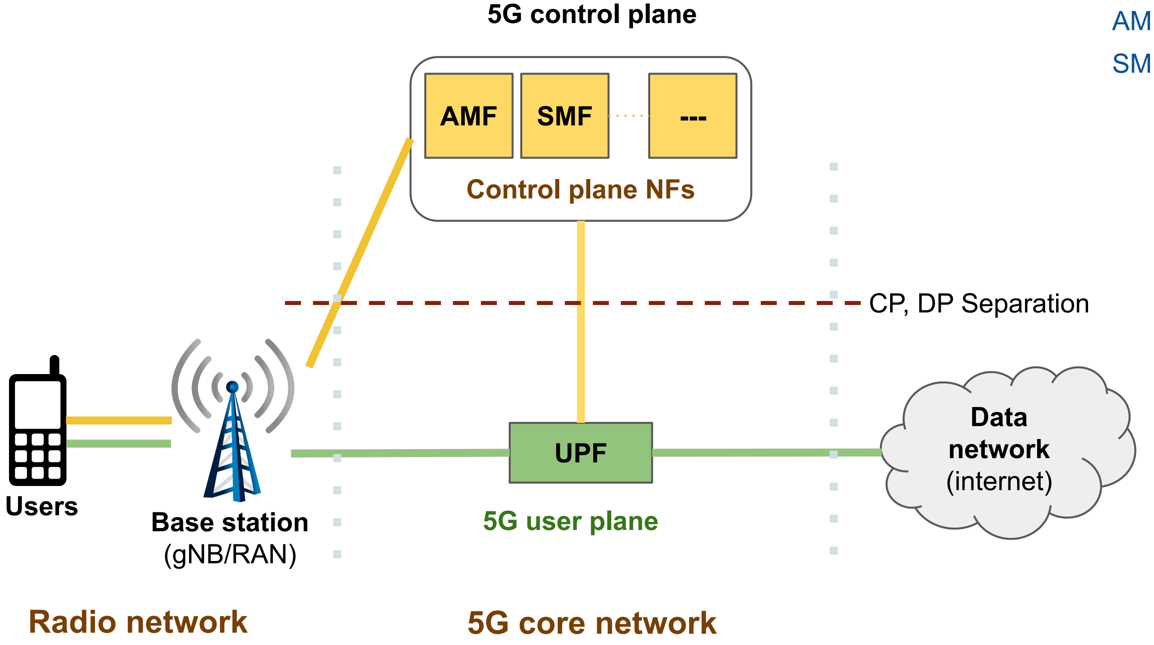
AccelUPF: Accelerating the 5G user plane using programmable hardware

Abhik Bose, Shailendra Kirtikar, Shivaji Chirumamilla, Rinku Shah, Mythili Vutukuru

Department of Computer Science & Engineering
Indian Institute of Technology Bombay

The ACM SIGCOMM Symposium on SDN Research (SOSR) 2022
Virtual event
October 20, 2022

5G architecture



Control plane traffic
User/Data plane traffic

AMF: Access and mobility management function

SMF: Session management function

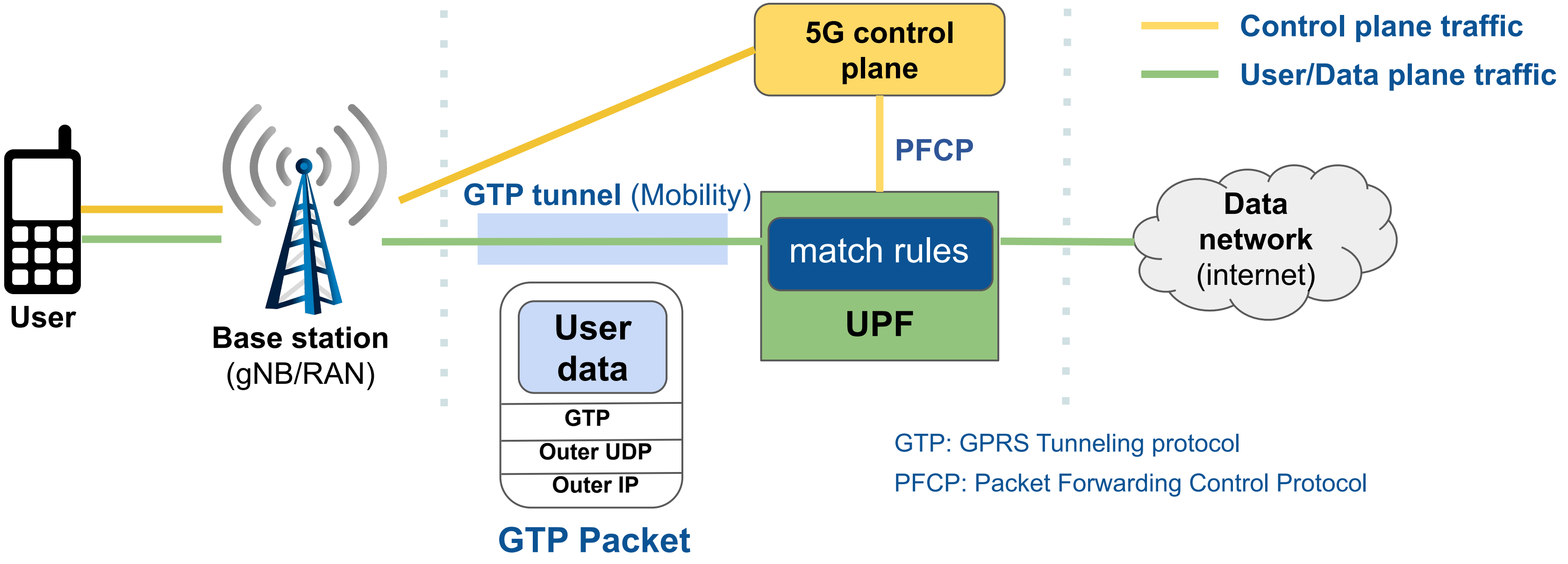
Control plane

- User registration
- Authentication
- Session management

User Plane Function (UPF)

- Forwards user data

5G UPF responsibilities



GTP: GPRS Tunneling protocol

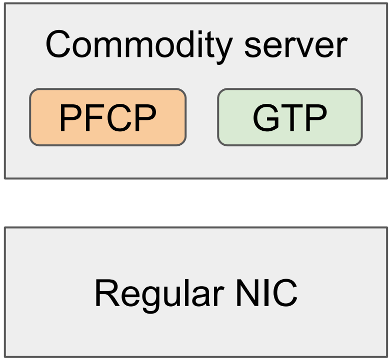
PFCP: Packet Forwarding Control Protocol

Primary roles of 5G UPF

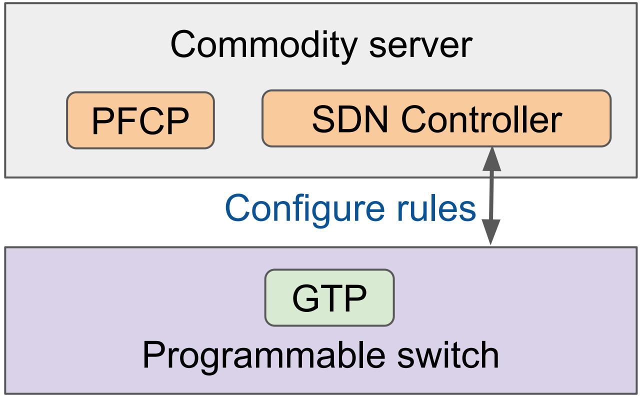
- Get forwarding rules via PFCP protocol
- Forward user data, do GTP en/de capsulation
- Rate enforcement and QoS
- Policy and charging

UPF performance is critical to future 5G success

State-of-the-art UPF design choices



Software UPF



GTPOffload UPF [1,2,3]

GTP performance	Software UPF	GTPOffload UPF
Kpps per USD	17	23
Kpps per Watt	194	373

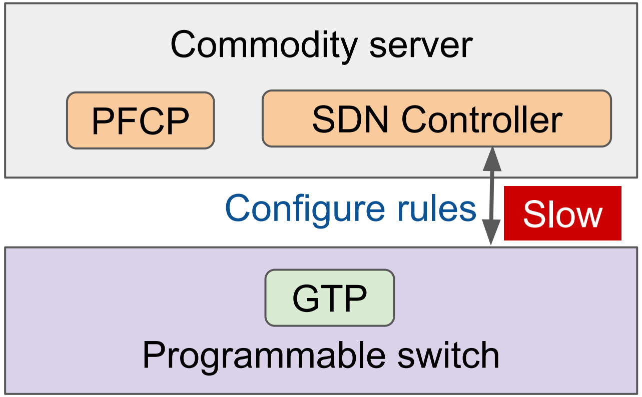
*** Both the designs support line-rate data forwarding**

GTPOffload vs. Software UPF
- **31% cost-efficient**
- **92% power-efficient**

[1] Leveraging Programmable Dataplanes for a High Performance 5G User Plane Function
[2] The Kaloom 5G User Plane Function (UPF). (2019)
[3] Optimizing UPF performance using SmartNIC offload. (2020).

Is GTP offloading always a good idea?

PFCP performance for GTPOffload design



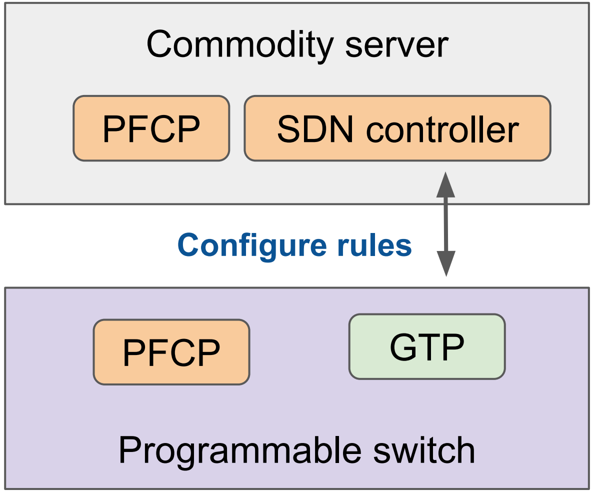
GTPOffload UPF

Controller-switch communication bottleneck

Performance metric	Software UPF	GTPOffload UPF
PFCP Throughput (messages/sec)	8.3K	**499**
PFCP Latency (μS)	40	**447**

PFCP throughput ⬇️
PFCP latency ⬆️

AccelUPF key idea



Move PFCP processing to hardware;
Solves the controller-switch communication bottleneck

Challenge: Parsing complex PFCP messages

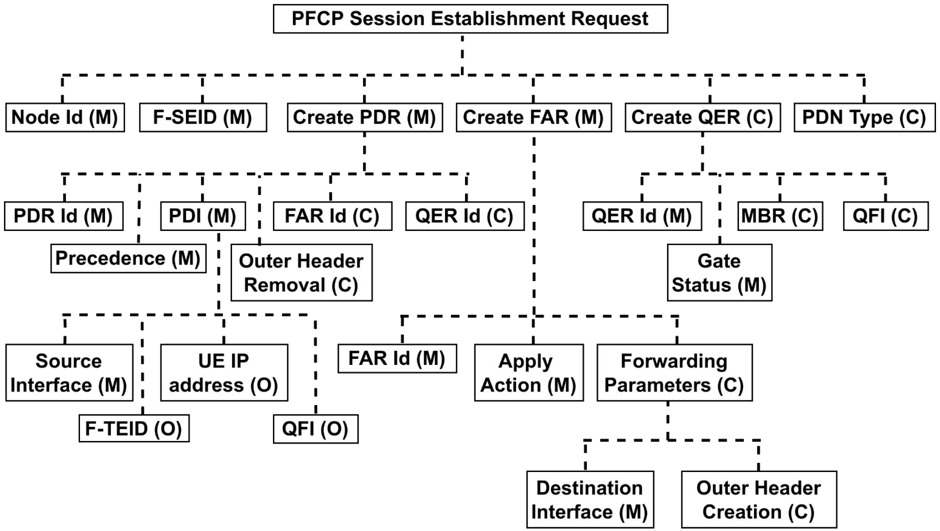
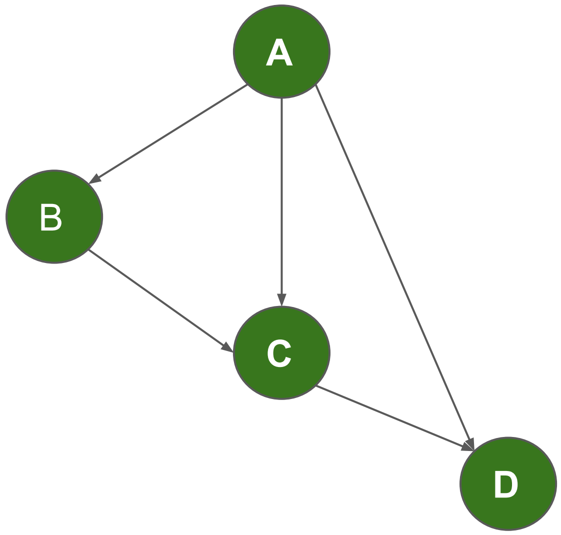


Figure: Minimal PFCP session establishment message

Complex PFCP message

- 321 IEs (Information Elements, like header fields)
- Mandatory & optional IEs
- Recursive (IEs inside bigger IE)
- Any IE order possible

Solution: AccelUPF parser design



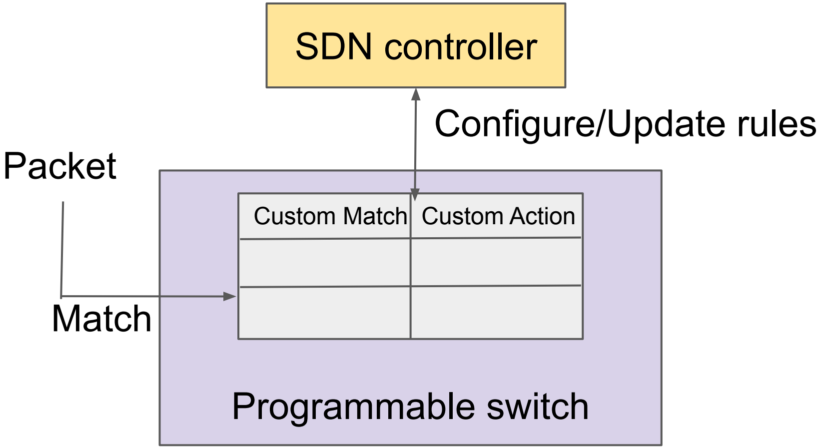
Reasonable simplifications

- 1 match-action rule per packet
- Standard suggested IE order

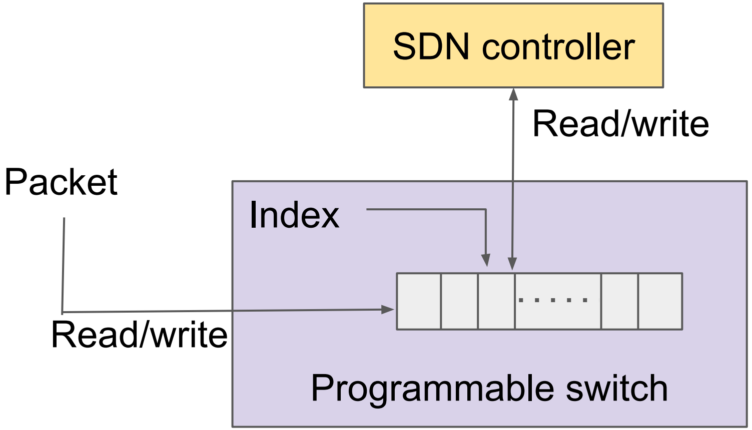
Parser Design

- Identified smallest mandatory unit
- Dynamically choose parse states based on optional IEs

Challenge: Can’t use P4 match-action table



Match-Action table



Register array

Supports Key based matching

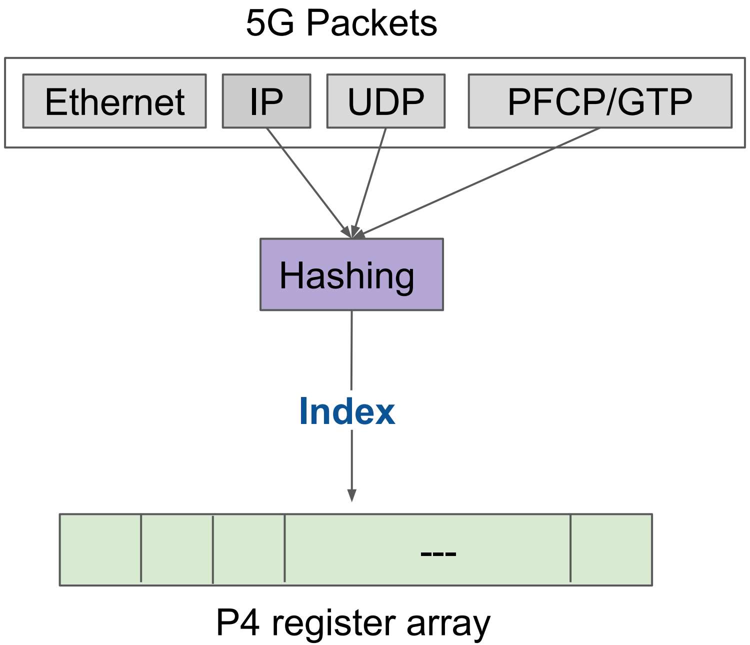
Not modifiable at the data plane

Modifiable at data plane

Support index based access only

Can we use P4 register arrays to store & match session rules?

Challenge: Using P4 register array to store 5G session rules



PFCP session identifier

- Session ID (SEID)

GTP uplink session identifier

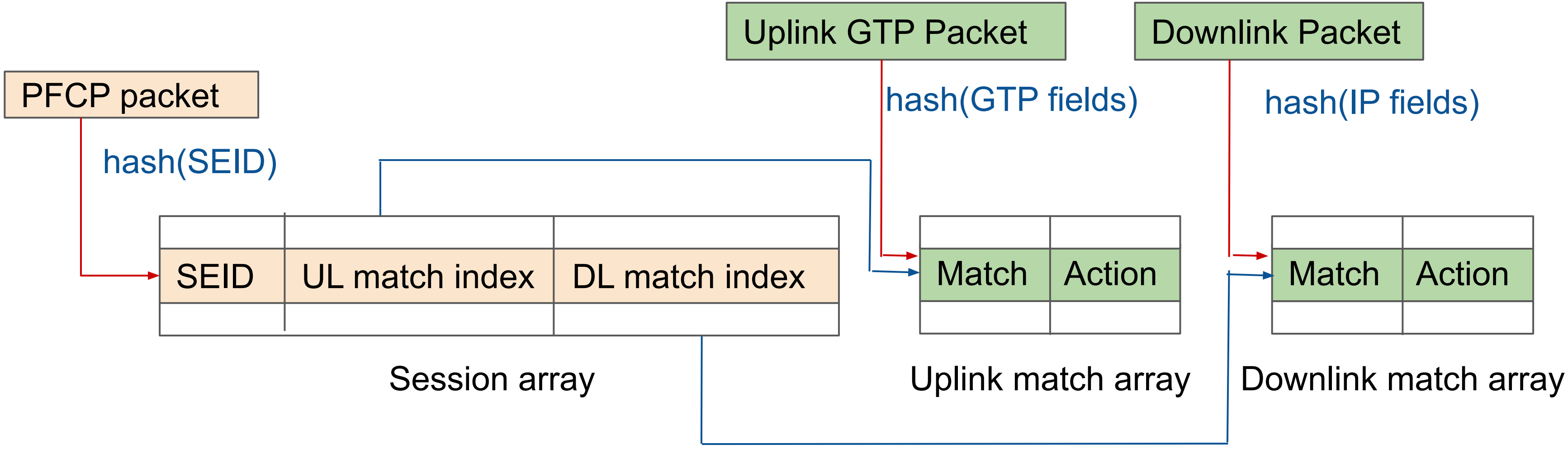
- User IP | Tunnel ID | …

GTP downlink session identifier

- User IP

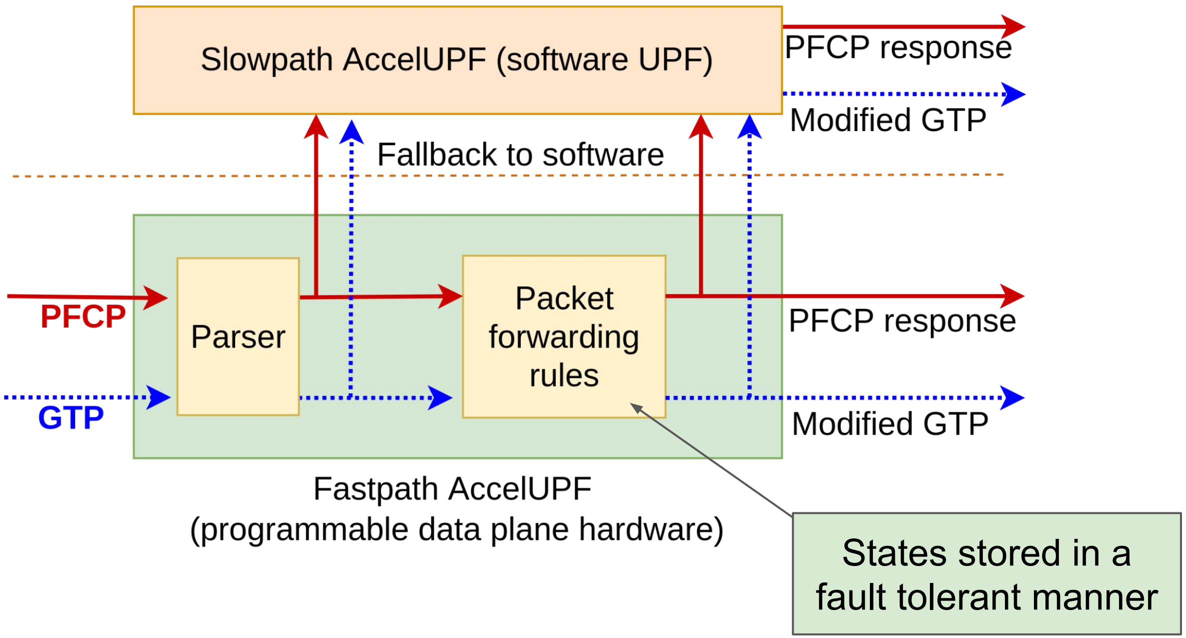
Which fields will be used for indexing?

Solution: in-network modifiable data structure



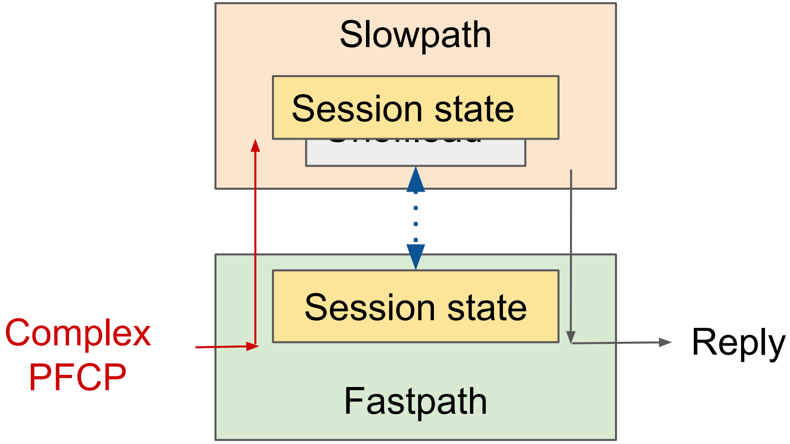
Both PFCP & GTP can index the session rules

AccelUPF design



Split architecture:
Fastpath processes **frequent** PFCP messages
Slowpath processes **complex** PFCP messages

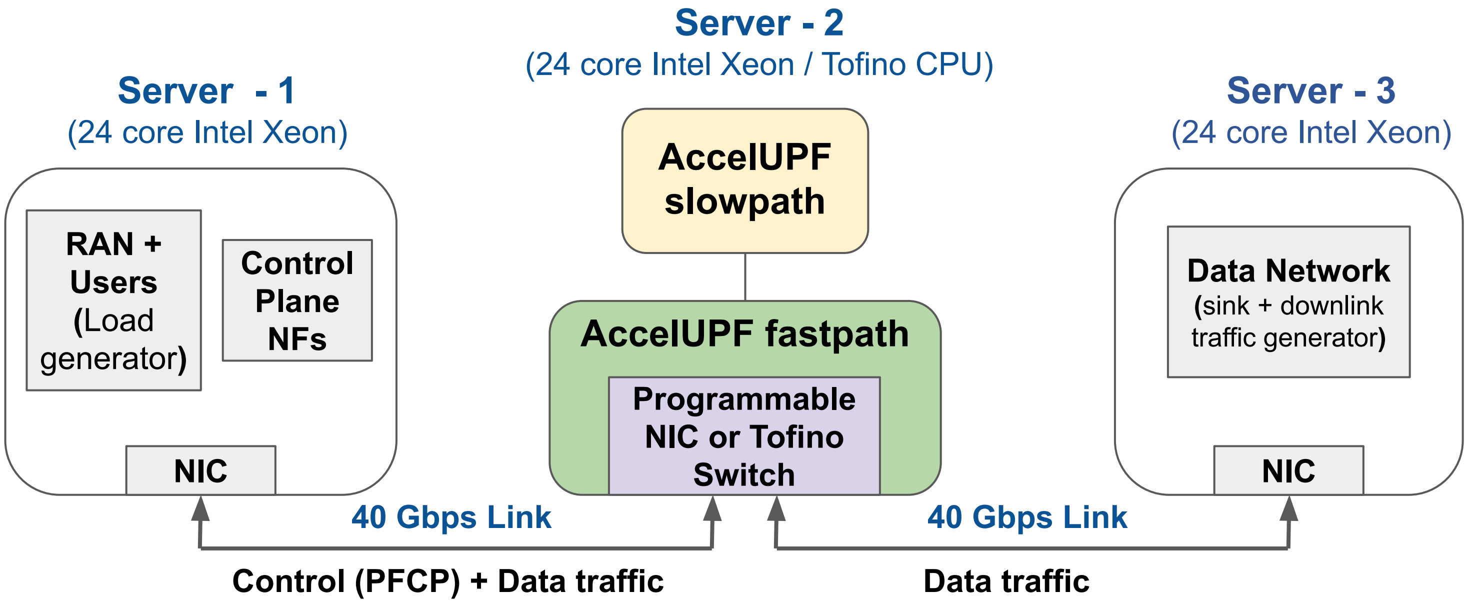
AccelUPF: State ownership



- A session can not be shared between fast and slow path
- Complex PFCP message: Session is migrated to slow path

Not performance critical: Only 1st packet suffers

Experimental setup

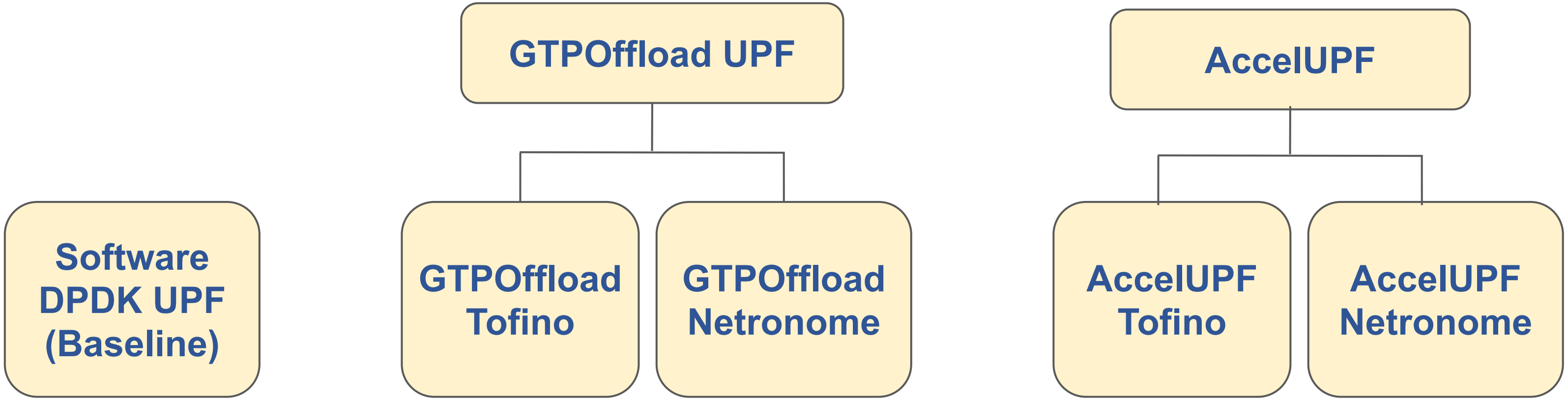


Programmable data plane hardware details

- Netronome Agilio CX 2x40GbE programmable NIC
- Tofino switch Intel Tofino Edgecore Wedge 100BF-32X

Production grade standard compliant 5G testbed

UPF design choices

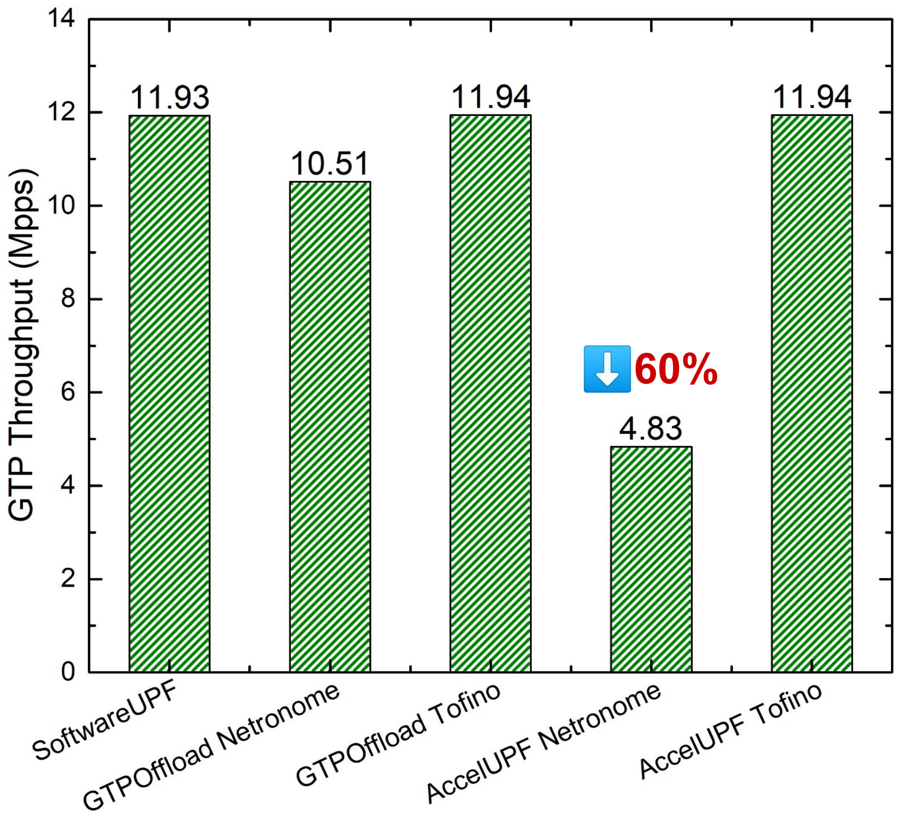


Metrics

- Throughput
- Latency (RTT)
- Cost efficiency
- Power efficiency

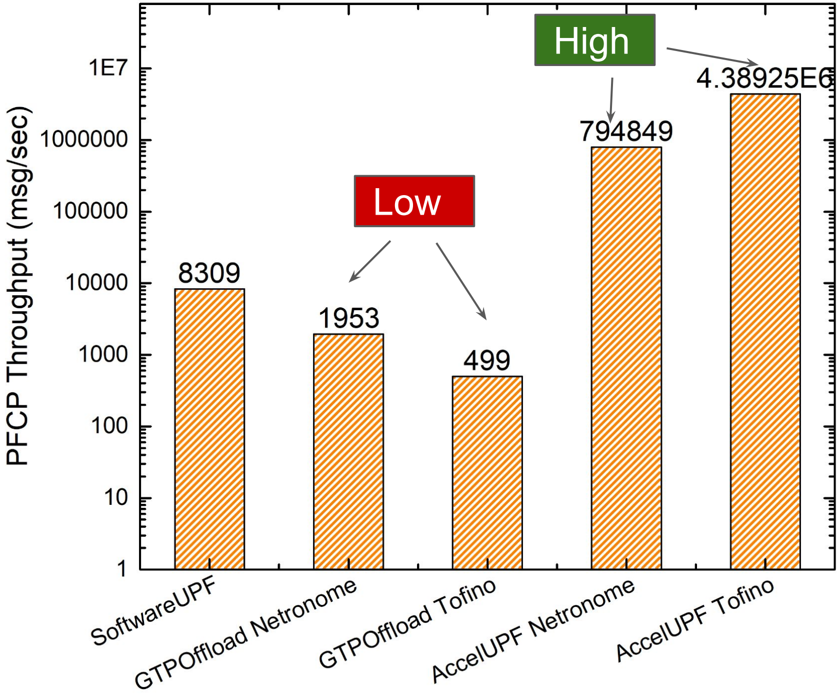
Result: User data forwarding (GTP) throughput

AccelUPF Netronome suffers due to contention for state access



AccelUPF Tofino supports line-rate GTP throughput

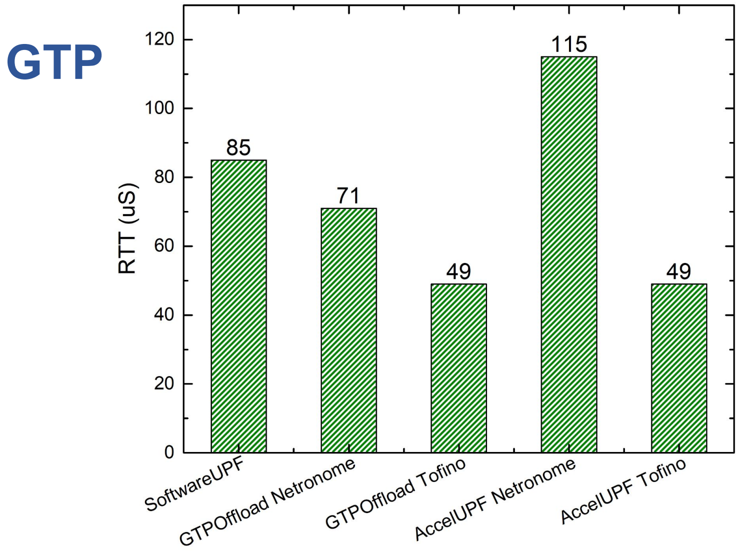
Result: PFCP throughput

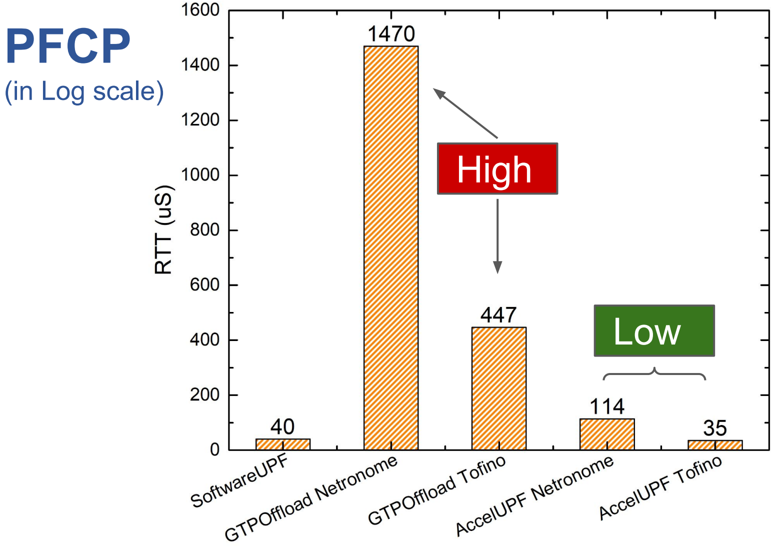


GTPOffload UPF:
Low PFCP throughput

AccelUPF:
High PFCP throughput

Result: GTP and PFCP latency (RTT)





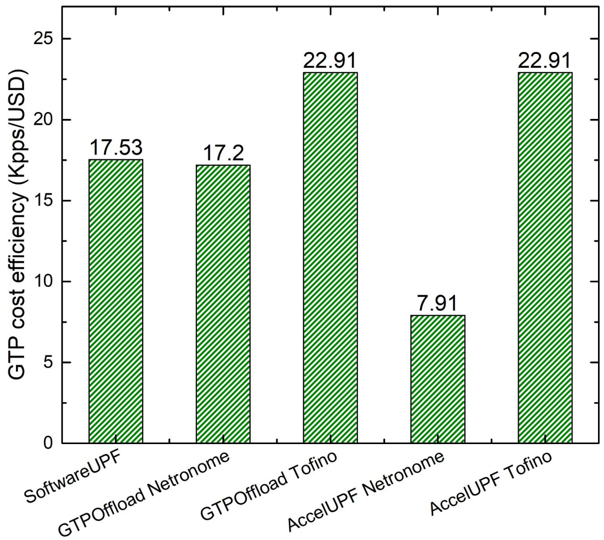
AccelUPF Tofino: GTP latency comparable to prior designs

GTPOffload UPFs: High PFCP latency

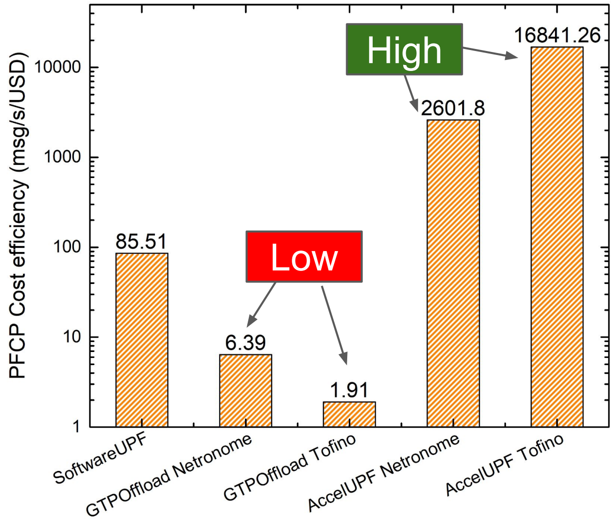
AccelUPFs have lowest GTP and PFCP RTT

Result: GTP and PFCP cost efficiency

GTP



PFCP
(in Log scale)

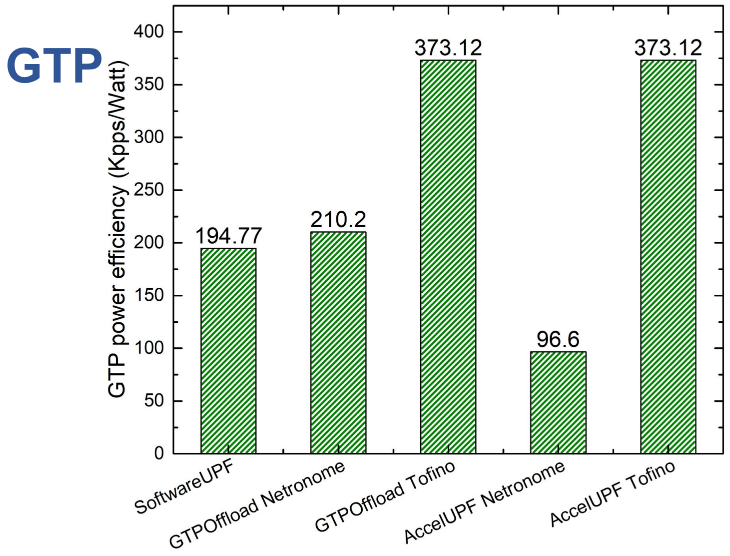


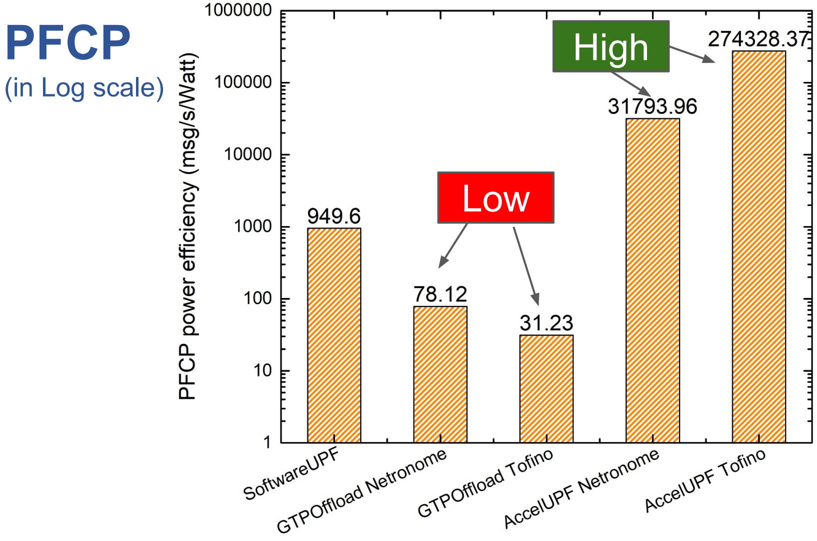
AccelUPF Tofino: GTP Cost efficiency is comparable to prior designs

GTPOffload UPFs PFCP Cost efficiency ⬇️

AccelUPFs PFCP cost efficiency: Much higher than software UPF

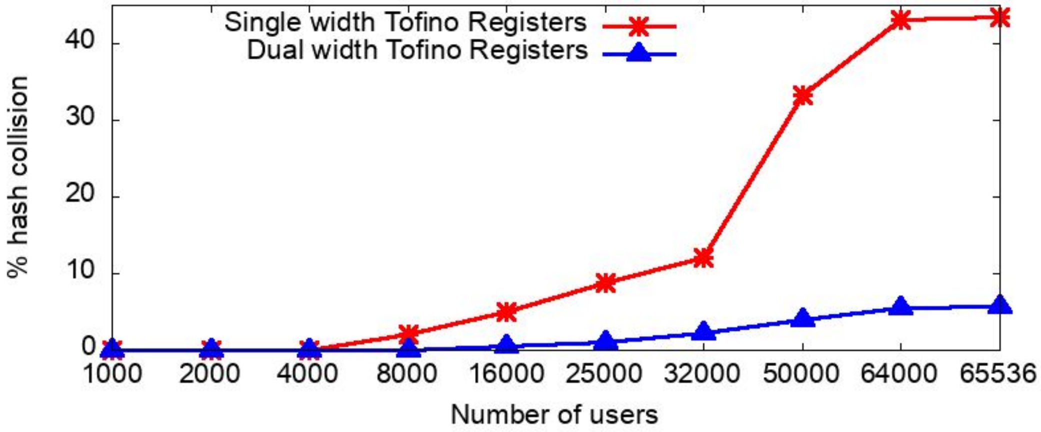
Result: GTP and PFCP power efficiency





AccelUPF power efficiency follows similar trends as cost efficiency

AccelUPF: Maximum number of supported user and hash collision

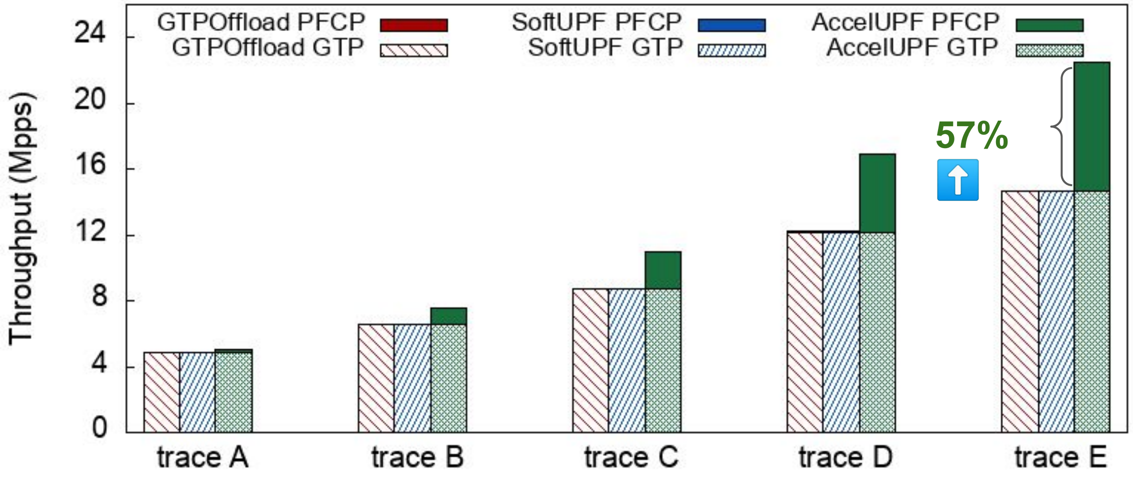


- **AccelUPF - 64K users/pipeline**
- **Tofino dual width registers used to reduce hash collision**

AccelUPF supports 128K users with minimal hash collision

AccelUPF: Evaluation with real world like traffic

Trace	A	B	C	D	E
PFCP %	3.65	12.53	19.63	28.86	35.79



Note: Average packet size decreased hence Mpps increased from trace A to E

- 5 real world ethernet traces[1]
- PFCP messages were inserted
- Session release triggered when users were inactive for 10 sec

AccelUPF:
57% ⬆ throughput

1. *UNSW Sydney. 2021. IOT TRAFFIC TRACES. https://iotanalytics.unsw.edu.au/iottraces.html*

Summary

- Comprehensive programmable data plane offloaded UPF design
- Prior works offloaded GTP only; AccelUPF offloads PFCP too
- Evaluated our prototype on TWO hardware platforms; Netronome and Intel Tofino
- AccelUPF is cost-efficient and power-efficient for high PFCP traffic

Email: abhik@cse.iitb.ac.in
Website: www.cse.iitb.ac.in/~abhik

Thank You!