Department of Computer Engineering and Industrial Automation (DCA)
School of Electrical and Computer Engineering (FEEC)
University of Campinas (UNICAMP)

ERICSSON

Advanced Data Plane Techniques for 5G: Hardware-Accelerated Hybrid User Plane Functions and Intelligent Traffic Identification and Prioritization

PhD Candidate: Suneet Kumar Singh
Advisor: Prof. Christian Esteve Rothenberg

FEEC
Faculdade de Engenharia Elétrica e de Computação
UNICAMP

01 November, 2024

INTRIG
INFORMATION & NETWORKING TECHNOLOGIES RESEARCH & INNOVATION GROUP

Agenda

Introduction and Motivation

5G Use Cases and Requirements

SELF DRIVING

Latency < 1 ms

IoT

Latency < 1 ms

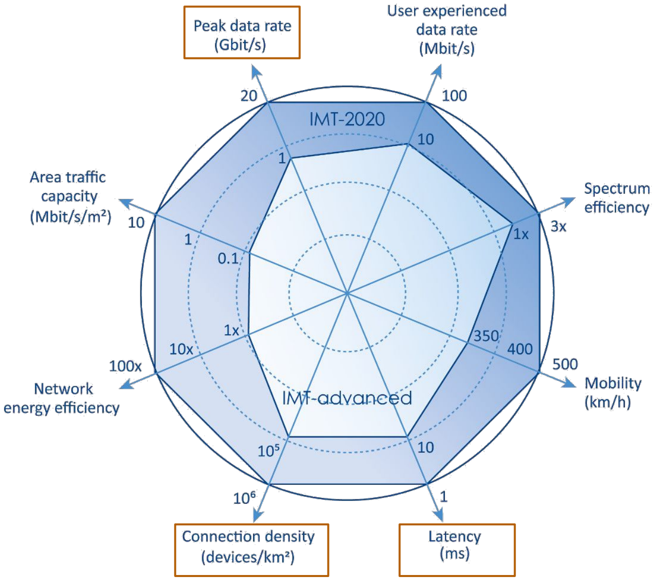
AR/ VR

Latency 5-10 ms

UAV/DRONE

Latency < 20 ms

20 Gbps



1 Million

1 msec

(Source: IoT solution world congress, medium Nayanathara post, forbes)

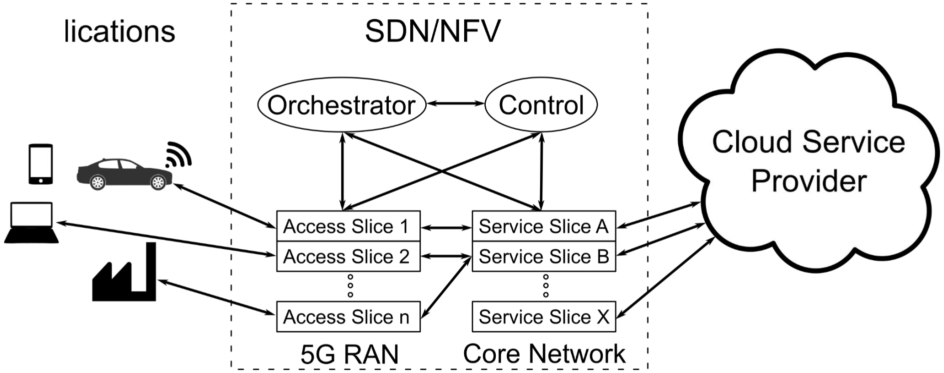
Key Network Requirements for 5G

❖ Flexibility and Scalability

Solution: Softwarization and Virtualization (SDN/NFV)

Advantages

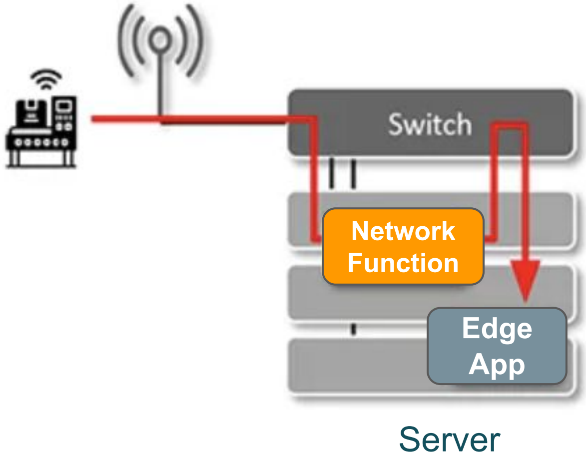
➔ Flexibility and Reconfigurability
➔ Enable network slicing for QoS
➔ High Scalability
➔ Reduced CAPEX and OPEX
➔ Eliminate dedicated hardware



(Source: Per Vices)

Performance bottlenecks in NFV

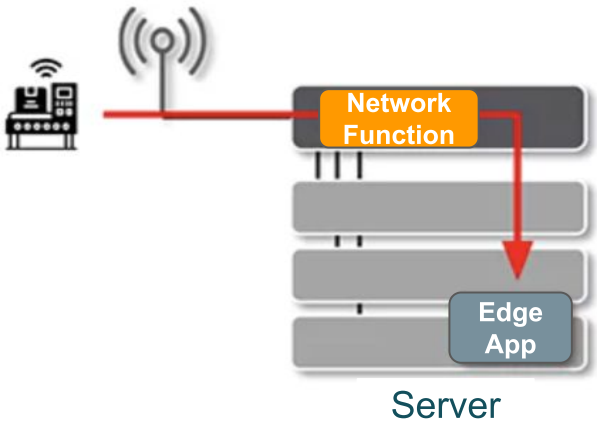
- *High* and *unpredictable* latency of softwarized packet processing and network function execution (more CPU cycles required)
- *Cache misses* increase overhead
- *Interrupt handling* can cause more delay



➔ High latency and jitter
➔ High delay can cause lower QoS specifically for delay critical application flows

Offloading Network Function

- Switch line rate packet processing
- Throughput in the order of *Terabits* per sec with latency below 1.5 *microseconds*
- Jitter of around 4 *nanoseconds*



➔ How can we offload network functions to the switch ?

Data Plane Programmability (P4)

- Avoid *vendor lock-in* & no waiting for silicon upgrades
- Protocol independent
- Add new features in runtime
- Provide *flexibility* to offload network functions to the Programmable Targets (e.g., *x86, FPGA, ASIC*)

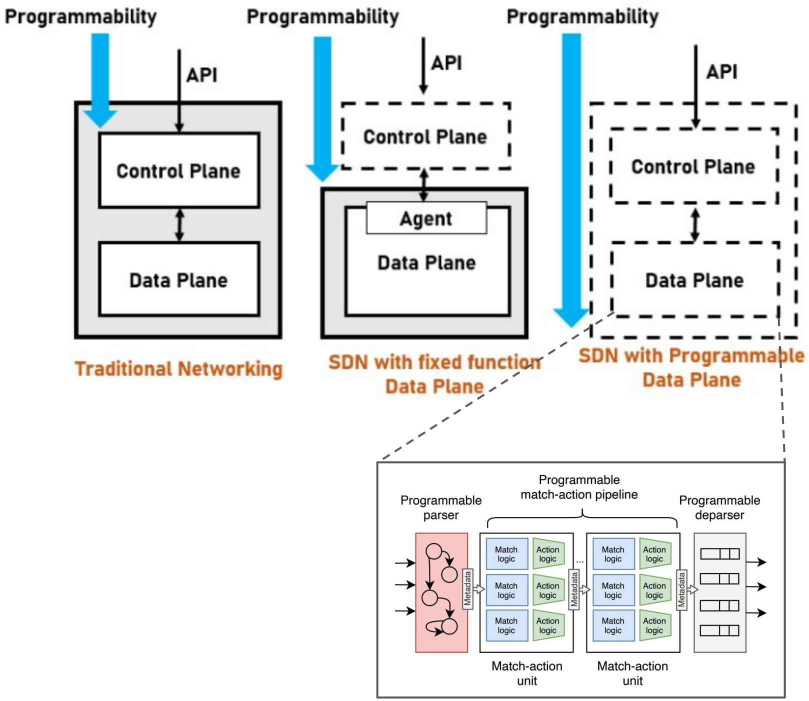
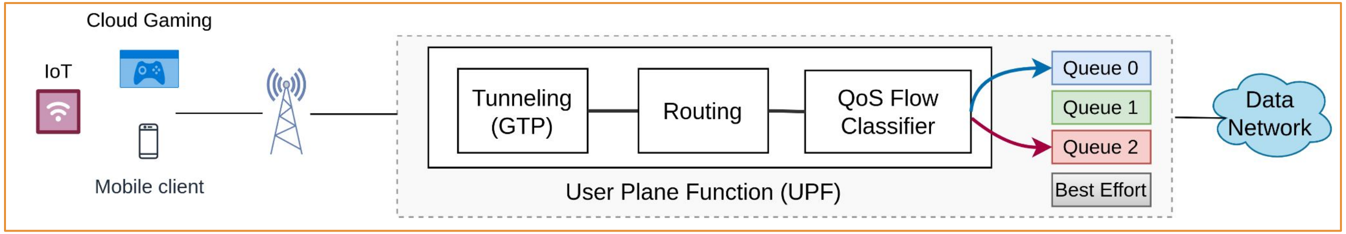


Figure source: Paper "A Survey on P4 Challenges in Software Defined Networks: P4 Programming"
Figure source: Paper "A Survey on Data Plane Programming with P4: Fundamentals, Advances, and Applied Research"

Programmable data plane to improve UPF

- Leverage programmable data plane to design low *latency* and *scalable* user plane function (*critical part of 5G network*)
- Line rate traffic *identification* and *prioritization* to improve QoS



Research Goals and Research Questions

Research Goal 1

❖ Acceleration of user plane function to meet the high performance requirements of 5G mobile networks with maintaining scalability

Research Question #1.1: “How can UPF be designed in programmable switch hardware to satisfy 5G lower latency requirements?”

5G User Plane Function Acceleration (Chapter 2)

➔ Using P4, we implement the core functionalities of 5G UPF in programmable hardware to show performance compared to NFV based implementations.

Research Goal 1

- ❖ Acceleration of user plane function to meet the high performance requirements of 5G mobile networks with maintaining scalability

Research Question #1.2: “How can the UPF be scaled with high performance?”

Hybrid Design for 5G User Plane Function (Chapter 3)

➔ The hybrid pipeline approaches for UPF that make use of target-specific features. We use Heavy Hitter detector for routing packets between programmable devices.

Research Goal 2

- Classify traffic on line rates to meet the desired QoS

Research Question #2: “How to classify high performance application flows in programmable hardware with high accuracy and lower detection time to enable use for QoS scheduling?”

Heavy Hitter Detection in the Data Plane (Chapter 4)

- ➔ We present a novel metric for HH detection in this chapter. New metric makes this method more accurate.

Research Goal 2

- ❖ Classify traffic on line rates to meet the desired QoS

Research Question #2: “How to classify high performance application flows in programmable hardware with high accuracy and lower detection time to enable use for QoS scheduling?”

Ml-based Cloud Gaming Traffic Detection (Chapter 5)

➔ In this chapter, we present a ML model and the computation of important features to train the model for classifying cloud gaming traffic in programmable hardware.

5G User Plane Function Acceleration

User Plane in 5G

1) Software based solution

Benefits:

➔ High flexibility and scalability

Limitations:

➔ Non-deterministic behaviour

➔ High Latency

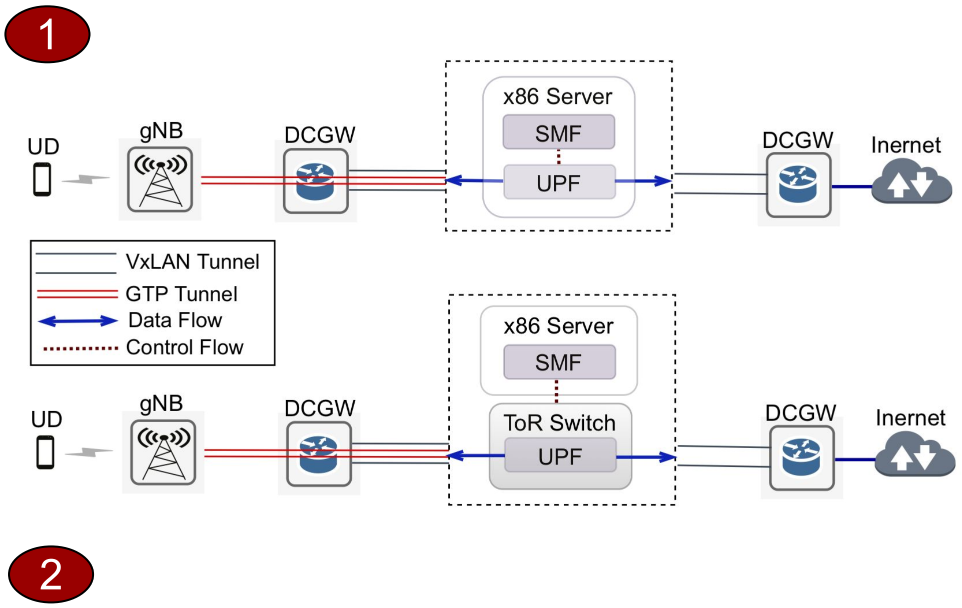
2) Hardware based solution

Benefits:

➔ High throughput and low latency and jitter

Limitations:

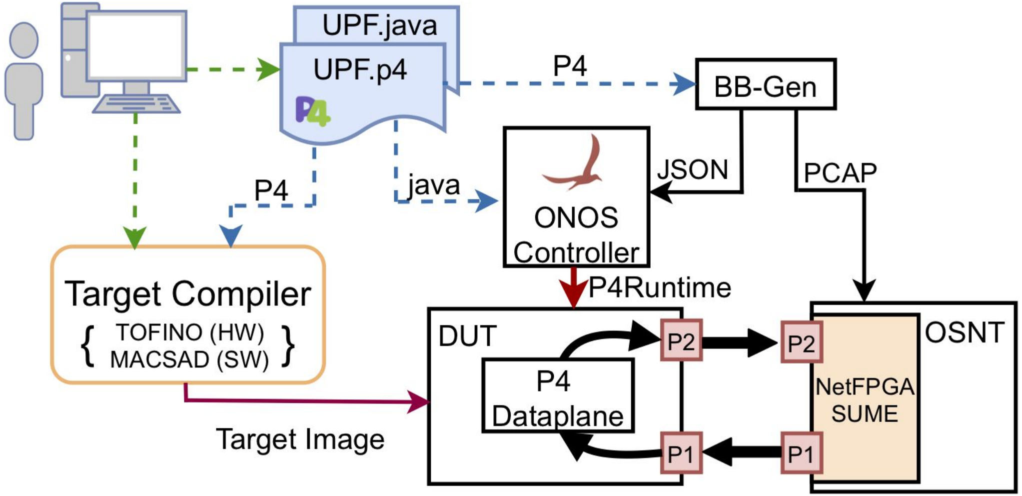
➔ Limited memory resources, limited scalability



Functions Implemented in P4

- GPRS tunnelling protocol (GTP) (de)encapsulation
- VxLAN (de)encapsulation of GTP flow
- Applied metering and set Queue Id
- TEID Management
- Stateless firewall
- IP routing towards internet and gNodeB
- Eth/IP Forwarding to/from datacenter gateways

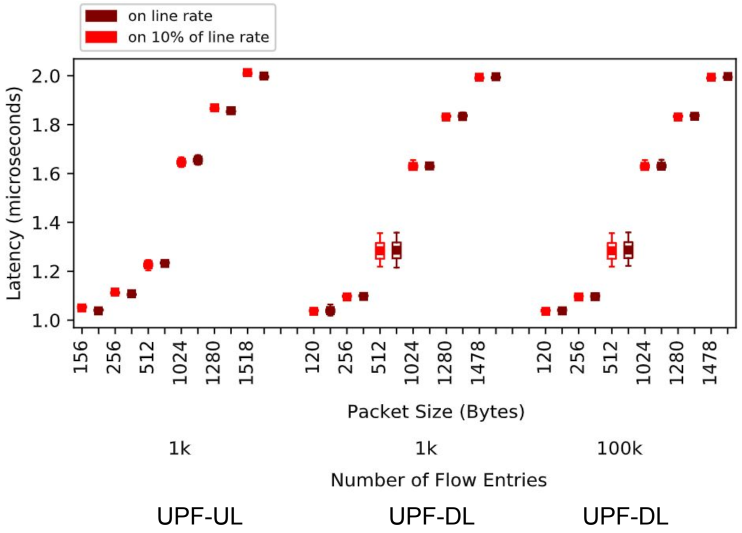
Testbed



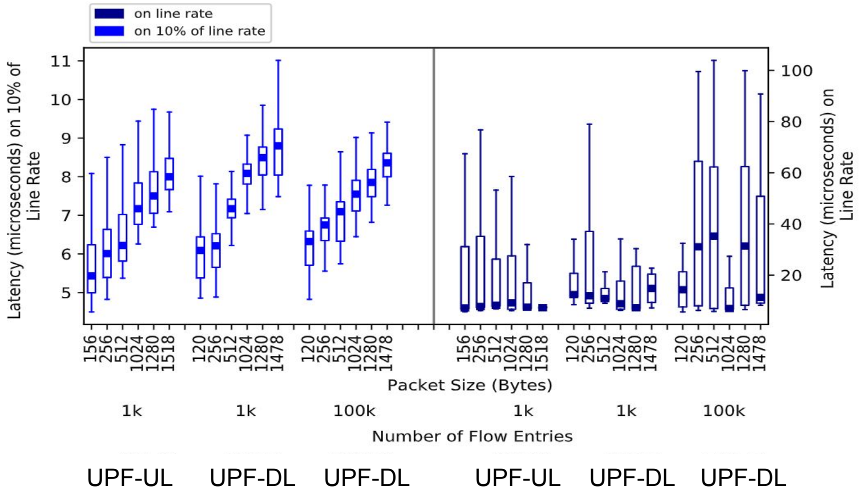
➔ MACSAD is a P4 compiler uses OpenDataPlane (ODP) to achieve portability across targets without compromising performance.

Performance Evaluation

Latency: Tofino UPF : 1-2 microsec, x86 UPF : 20-100 microsec



Tofino ASIC

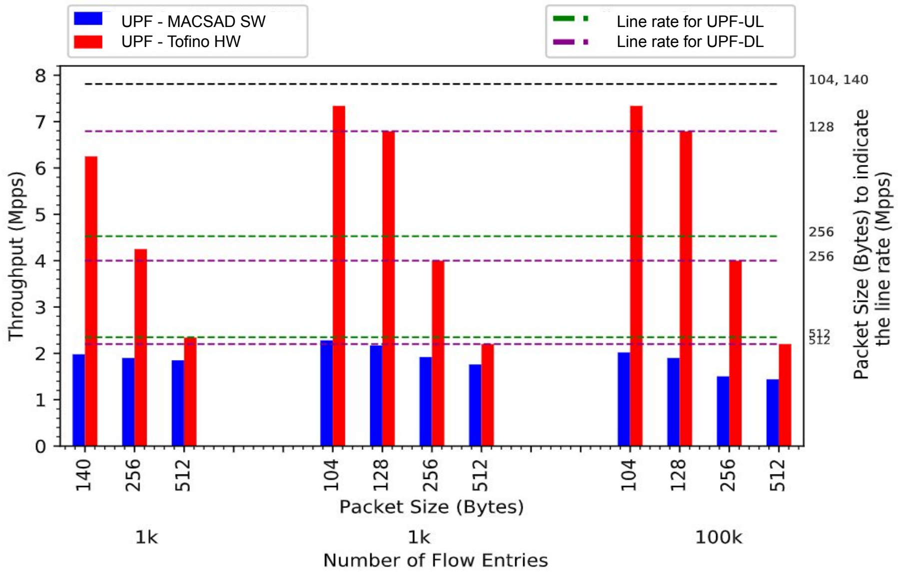


x86

Performance Evaluation

Throughput

- UPF Tofino outperforms 2-3 times compared to UPF x86

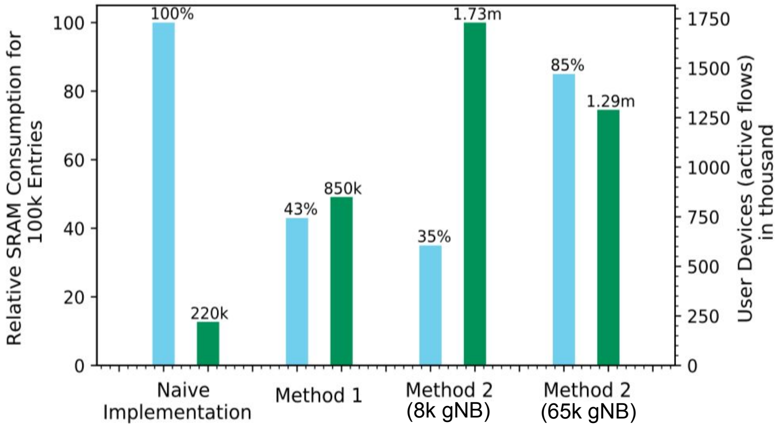


UPF-UL UPF-DL UPF-DL

Scalability and Resource Consumption

Maximum User Devices

❖ UE : 850k, SRAM : 43%



Discussion

- ❖ P4 based Tofino HW allows to implement 5G UPF
- ❖ Packet processing on line rate and lower latency than SW UPF
- ❖ Handle around 1.5 millions active users with only basic UPF functionalities

- ❖ Challenges for improving UPF
  - ➔ Limited scalability in P4 switch HW to handle million of flows
  - ➔ P4 constraints to perform complex operations in dataplane HW

➢ We need a solution to realize UPF on different targets (x86, ASIC, NetFPGA) to improve programmability without impact much on performance ?

Related Work

P4 based solutions

- ❖ P4-BNG: Central Office Network Functions on Programmable Packet Pipelines
- ❖ ONF Project: Offloading VNFs to programmable switches using P4

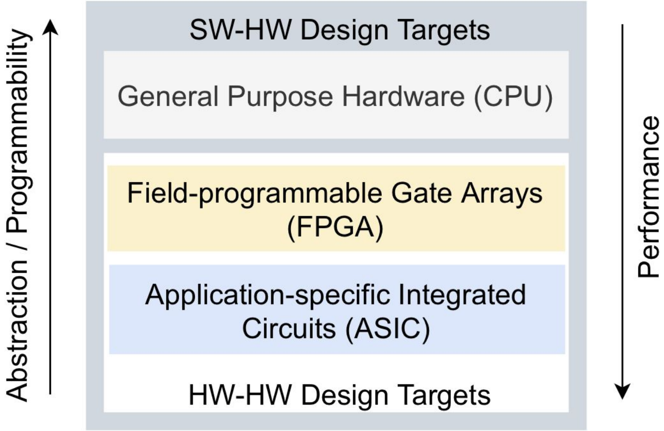
SDN/NFV based solutions

- ❖ CleanG: A Clean-Slate EPC Architecture and Control Plane Protocol for Next Generation Cellular Networks

Hybrid Design for 5G User Plane

Performance and Programmability Trade-offs

- ❖ x86-based packet processing
  - ➔ High programmability
  - ➔ Lower throughput
  - ➔ High latency and jitter
- ❖ ASIC-based packet processing
  - ➔ Limited programmability
  - ➔ High throughput
  - ➔ Low latency and jitter
- ❖ FPGA-based packet processing
  - ➔ More programmability than ASIC
  - ➔ Lower performance than ASIC

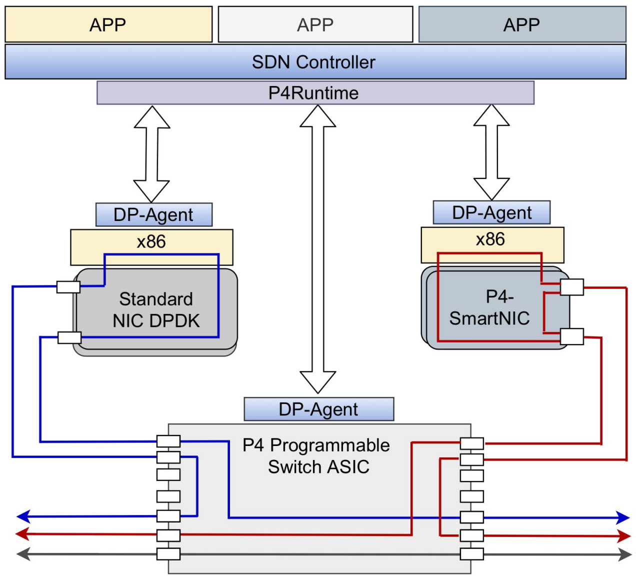


Methods to Offload Flows

- ❖ Offload once resources are full
- ❖ Decision is made based on Match-Action
- ❖ Application based offloading:
    - ➔ Heavy Hitters in switch HW (≈90% of traffic)
    - ➔ Heavy HItters are only 2-5% of total traffic
    - ➔ Critical flows such as Cloud Gaming/AR/VR in switch HW
    - ➔ MAU based decision (CP decides using QoS)

Benefits

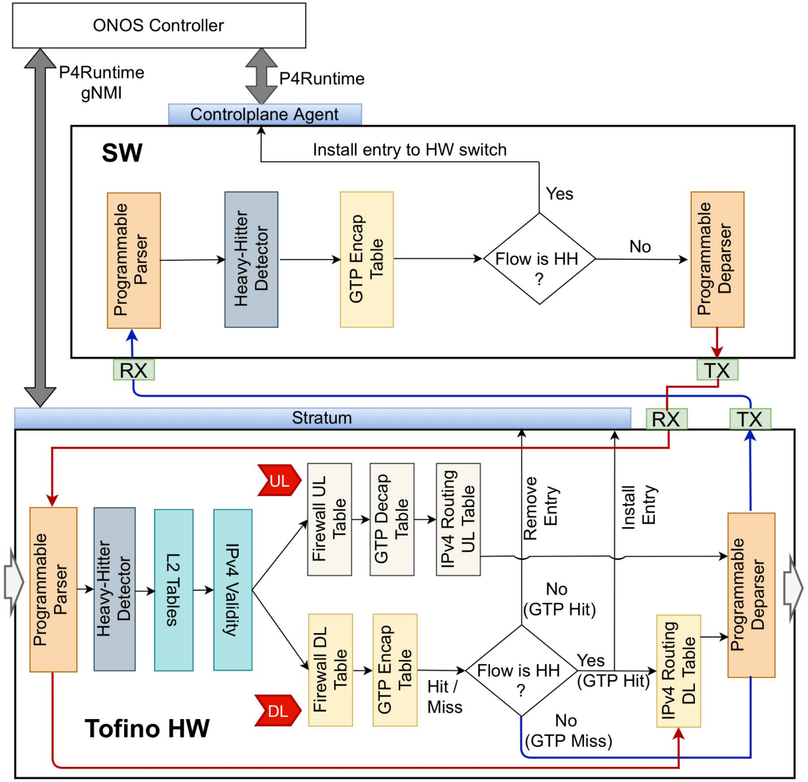
- ➔ Fulfil QoS requirements
- ➔ Lower CPU Utilization (less packets handling)
- ➔ Improve Scalability (no. of UE devices)



General Hybrid Architecture

Hybrid-UPF (ASIC + x86)

- ❖ For scalability, UE IP match-action table can be offloaded to the SW switch
- ❖ How to decide which packet goes to which pipeline ?
  - ➔ Packet routing can be based on heavy hitter detection
  - ➔ Majority flows are inactive (90%-95%) or non-HHs
  - ➔ Only 2-5% of flows are HHs of total traffic
  - ➔ *Novel IPG based HH detection is used for higher accuracy (will be discussed in later slides)*



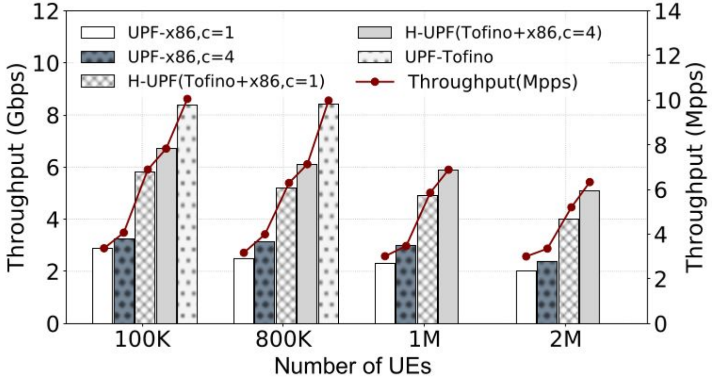
Scalability

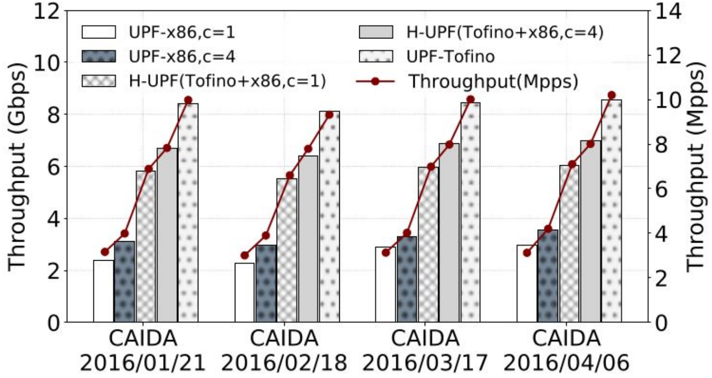
- Maximum number of UE IPv4 match in Tofino
- Carefully optimize to manage table dependencies to utilize maximum SRAM
- UPF-Tofino: 60% SRAM, Max. 850K UEs
- H-UPF Tofino: 34.4% of SRAM, Max. 430K UEs (heavy hitter)
- H-UPF x86: 15 M flows (light flows)

	UPF-Tofino	**H-UPF(Tofino+x86)**
SRAM	60%	34.4%(Tofino)
UEs	850K	430K(Tofino)+15M(x86)

Performance - Throughput

- ❖ UPF-x86 : UPF running entirely on x86
- ❖ UPF-Tofino : UPF offloaded to Tofino
- ❖ H-UPF
  - ➔ HH flows on Tofino switch
  - ➔ Non-HH flows go through Tofino + x86





- ➔ Analyze performance for different number of UEs and traces
- ➔ H-UPF outperforms than UPF-x86

Discussion

- 5G Hybrid UPF by combining features of different targets
- Proposed approach has trade-off to overcome P4 switch challenges
- Evaluation shows that Hybrid user plane is an attractive approach to meet the performance requirements and improve scalability
- Challenges for improving Hybrid-UPF

  - Not only using Heavy Hitter, the other critical flows (e.g., cloud gaming, AR/VR) are required to consider
  - Other programmable targets such as SmartNIC is also required to consider in splitting UPF functionalities

Related Work

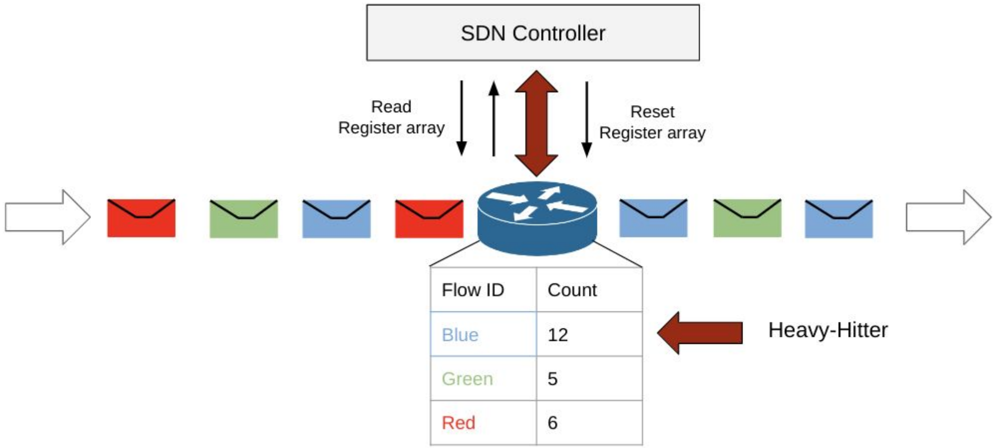
- Hybrid UPF P4 based solutions

  - Leveraging Programmable data planes for a high performance 5G user plane function (DPDK based UPF with smartNIC)
  - Aether Project: Model UPF and Performant-UPF. Executed functionalities such as packet buffering or hierarchical QoS on SmartNIC and x86
  - Openbng: Performed BNG on NetFPGA and Tofino switch ASIC

Heavy Hitter Detection in P4 Hardware

What are Heavy Hitters ?

- Small number of flows that contribute most of the network traffic in a *fixed* time interval

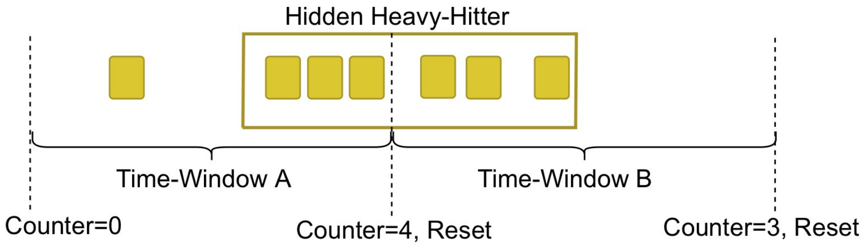


Why is Heavy Hitter Important ?

- Important for many network management applications
  - Hybrid design pipeline
  - Anomaly detection in data centers
  - Improving Quality of Service / Traffic Management

Problem in a Fixed Time Window approach

- *Yellow flow* would *not be considered heavy* because neither time window A nor B has enough occurrences
- This impacts on the *accuracy* and network applications such as *load balancing, hybrid UPF, anomaly detection and network* QoS



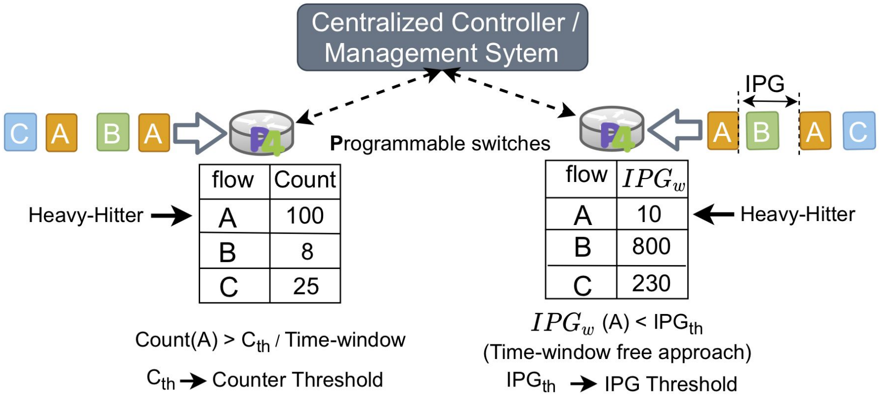
Existing Methods

- ❖ Sliding window based method
- ❖ Elastic Trie

➔ Both approaches are not supported by switch hardware
➔ We need a *non-window based approach* compatible with programmable switch hardware

Proposed Novel approach for HH detection

- ❖ The *smaller* the flow's IPG, the more packet in-flight, *heavier* the flow
- ❖ For counter, the *bigger* the flow counter, the *heavier* the flow

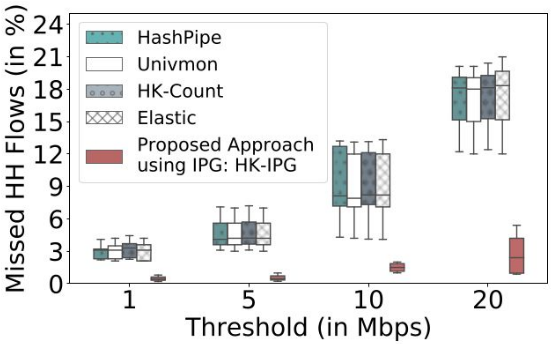


P4 ASIC Implementation Challenges

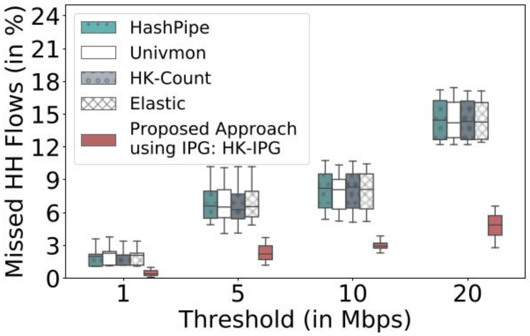
- *Access Limitations to Registers*: The register can be accessed once per packet lifetime in Tofino (e.g., update flow ID)
- *Fixed number of Stages*: Carefully optimize the code to fit in 12 stages
- *Arithmetic and Comparison Operations*: **Rely on an approximate EWMA calculation**

Evaluation (1/2)

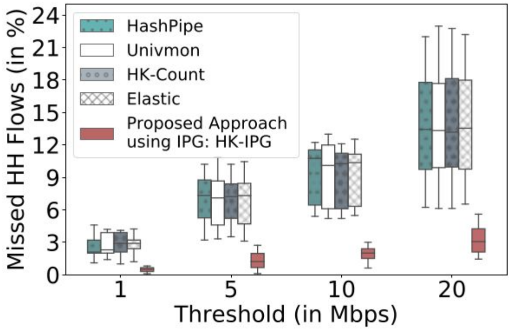
❖ *Accuracy:* Missed HH using disjoint windows



ISP backbone (CAIDA16)



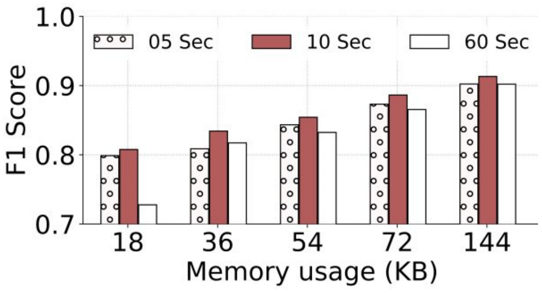
Data Center (IMC10)



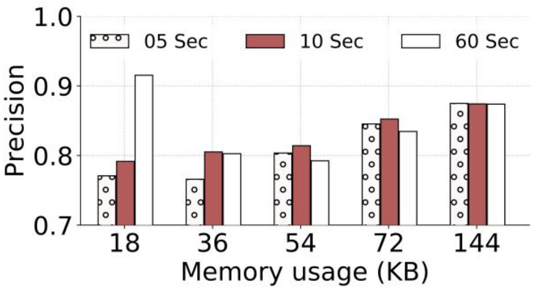
WIDE Backbone (MAWI20)

Evaluation (2/2)

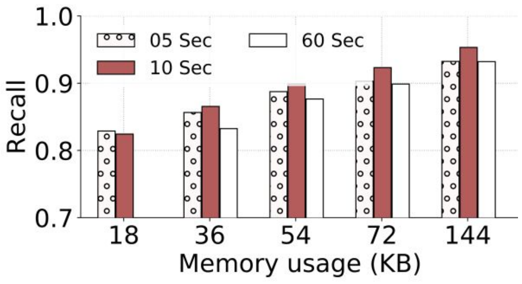
❖ *Accuracy:* On Tofino switch ASIC



F1 Score



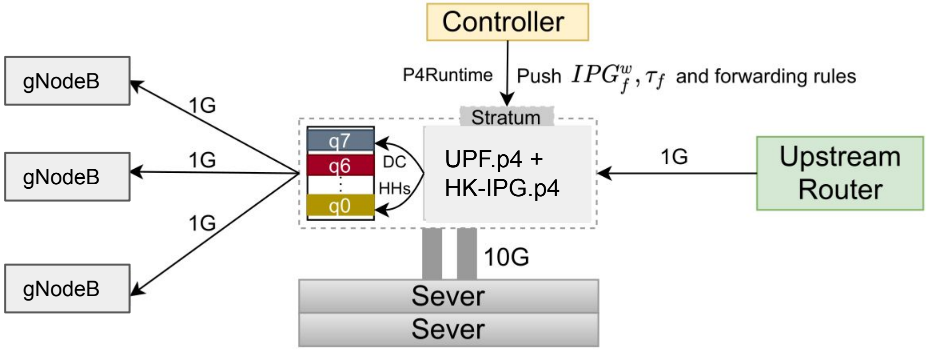
Precision



Recall

Use-Case: Heavy Hitters to lower priority Queue

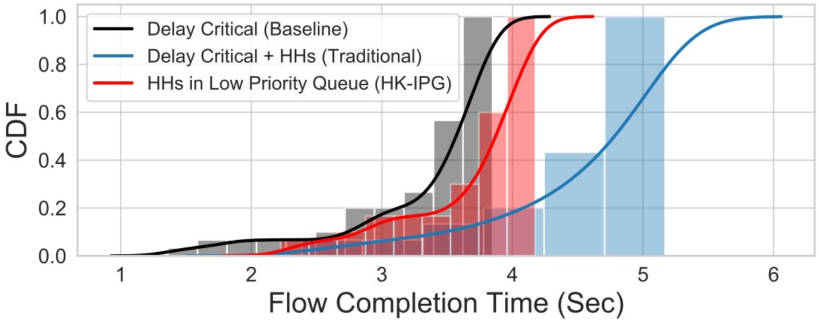
- *Baseline.* Consider Flow Completion Time (FCT) of delay critical flows without background traffic
- *Traditional setup.* Delay critical flows with concurrent HH flows
- *Offloading HH.* Detected HH sent to lower priority queue



Use-Case: Heavy Hitters to lower priority Queue

Setup

- *20 different TCP* flows sharing the available link bandwidth
- *Download 15 Mbytes data* from the Internet
- Evaluate FCT for each delay critical TCP flow
- HH flows occupy 40% of link bandwidth



Discussion

- High accuracy *Heavy Hitter* detection in programmable switch HW
- Used in *Hybrid design* and for QoS improvement
- Evaluation shows higher accuracy compared to existing methods
-
- *Critical flows other than Heavy Hitters* need to be identified on line rate

➔ Other critical flows (e.g., *cloud gaming, AR/VR*) are required to identifies on line rate and prioritize them

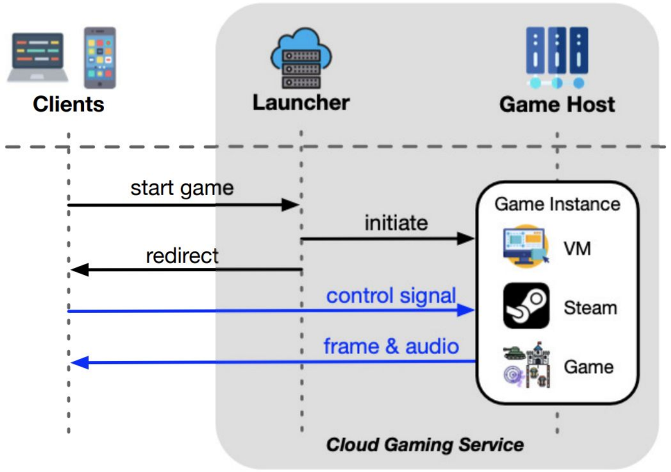
Related Work

- *Univmon:* One Sketch to Rule Them All: Rethinking Network Flow Monitoring with UnivMon
- *HeavyKeeper:* An Accurate Algorithm for Finding Top-k Elephant Flows
- *HashPipe:* Heavy-Hier Detection Entirely in the Data Plane

Machine Learning Based Cloud Gaming Detection

What is Cloud Gaming ?

- *Cloud Gaming* is server-based gaming solution that runs on remote server in data centers
- A player starts the game from PCs or laptops
- The *game launcher* initiates a game instance
- Then player directly sends *control signals* to the game instance
- The *encrypted* video streams are transmitted to the client

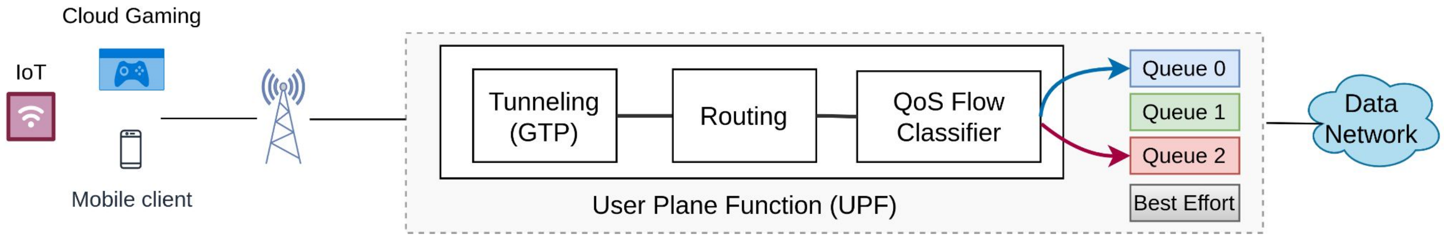


(Source: ACM, CCS, 2022, Ready Raider One: Exploring the Misuse of Cloud Gaming Services)

Latency Requirements and Prioritization

- ❖ Different services have *different requirements*
- ❖ CG QoE *can be affected by network conditions like* ***latency*** *and* ***jitter***
- ❖ Flow *identification* and *prioritization* on line rate to enhance QoE

Service	Recommended Latency
Amazon Luna	< 20 msec
Boosteroid	< 20 msec
GeForce Now	< 80 msec
Shadow PC	< 30 msec

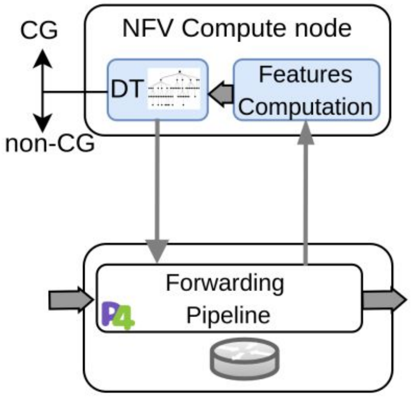


Issues with Existing CG Detection Methods

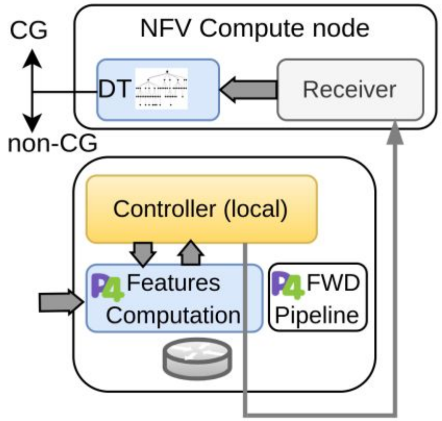
- ❖ Packet header information is not sufficient to detect *encrypted CG flows*
- ❖ Intelligent classifier (*Machine Learning*) is required for traffic identification
- ❖ Detection on NFV nodes can be slow which can impact on QoE

➔ Features computation and ML model are executed on NFV node

➔ Hybrid CG: Features computation in P4 switch and apply ML model in NFV node



CG detection on NFV node

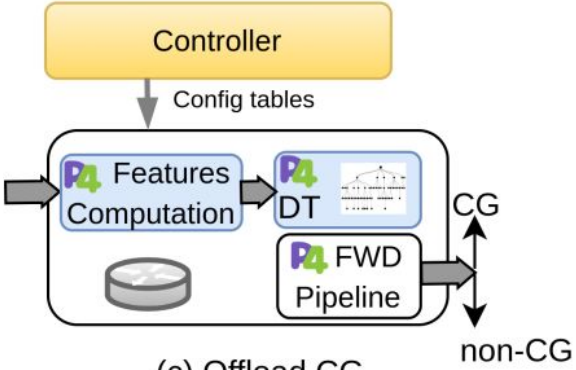


Hybrid CG detection approach

In-Networking CG Detection

- Train the *ML model* in control plane
- *Features* computation and *trained* ML model can be used in programmable hardware for quick identification

➔ CG identification entirely in switch hardware

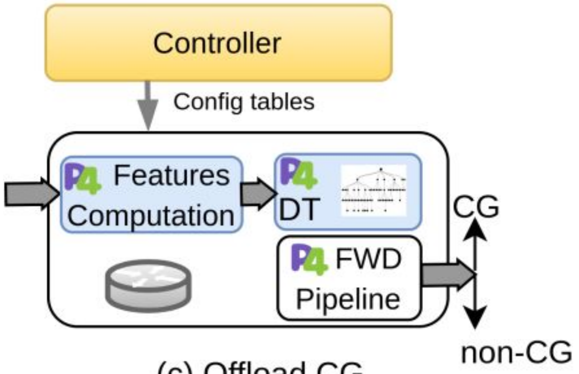


(c) Offload CG detection to switch HW

Challenges

- ❖ Feature computation (e.g., mean, standard deviation)
- ❖ Fit both feature calculation and ML model within the limited memory resources

→ MATADOR address these issues



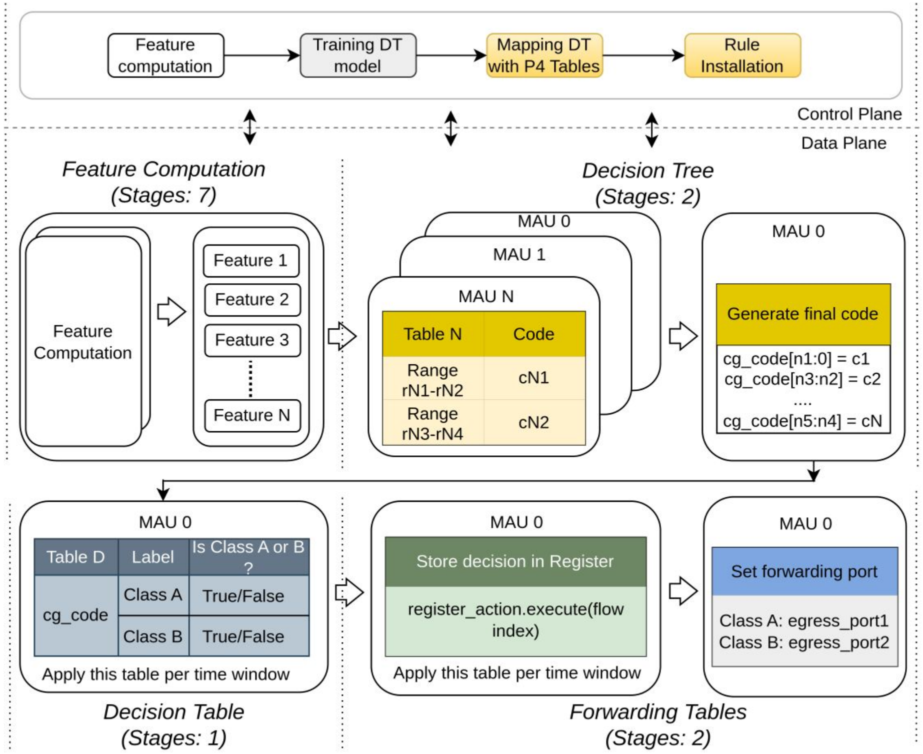
(c) Offload CG detection to switch HW

MATADOR Design (1/4)

- ❖ Line rate feature extraction
- ❖ *P4-TNA DT realization*
- ❖ Control Plane
- ❖ Deployment Limitation with *P4-TNA*

Control Plane
- ➔ *Train DT* model using dataset
- ➔ *Map DT* model with P4 MAU

P4 Data Plane
- ➔ Feature extraction
- ➔ Apply Trained Decision Tree
- ➔ Generate Code
- ➔ Using code to identify class



MATADOR Feature Extraction (2/4)

Line Rate Feature Extraction for CG

- ❖ ***Feature selection is a crucial step*** in *ML-based* classification
- ❖ Both *uplink* and *downlink* directions are considered for feature selection
- ❖ Multimedia flows as downlink and player commands as uplink
- ❖ Six features are taken into consideration

Features used

Uplink:

- ➔ *Exponential weighted moving average* of Inter Packet Gap for Uplink
- ➔ *Exponential weighted moving average* of Packet Size for Uplink
- ➔ *Number of Packets per time window* for Uplink

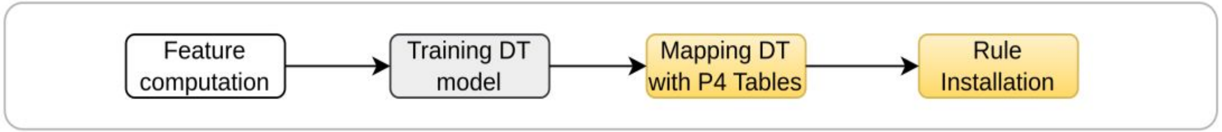
Downlink:

- ➔ *Exponential weighted moving average* of Inter Packet Gap for Downlink
- ➔ *Exponential weighted moving average* of Packet Size for Downlink
- ➔ *Number of Packets per time window* for Downlink

MATADOR Control Plane (3/4)

Control Plane

- ❖ Training the *DT model* using Python script
- ❖ Map the *trained* DT model with the MAU configuration
- ❖ Then, Install rules to the switch using *Barefoot runtime APIs*



MATADOR Limitations (4/4)

Deployment Limitation with P4-TNA

EWMA Computation:

- ❖ EWMA computation is challenging in Tofino switch
- ❖ *Low Pass Filter* (LPF) can be used, but smoothing factor should not be fixed
- ❖ For higher accuracy, we need to fix the smoothing factor (*i.e*, 0.96-0.99)
- ❖ Currently, we rely on average EWMA calculation

Number of Stages:

- ❖ Tofino 1 has fixed 12 *stages*
- ❖ DT algorithm depends on the features, so we cannot parallelize both process
- ❖ Tofino 2 has more flexibility in terms of stages and resources to run ML model

Implementation (1/2)

Dataset

Training Dataset:

- ❖ Dataset used to train the model: *https://cloud-gaming-traces.lhs.loria.fr*
- ❖ The Lorraine laboratory in France
- ❖ The four main CG platforms—*xCloud, PlayStation, GeForce, and Stadia*
- ❖ Non-CG cases: *video streaming* (live streaming, game, remote desktop protocol, youtube), *video conferencing, and Facebook navigation* (browsing)

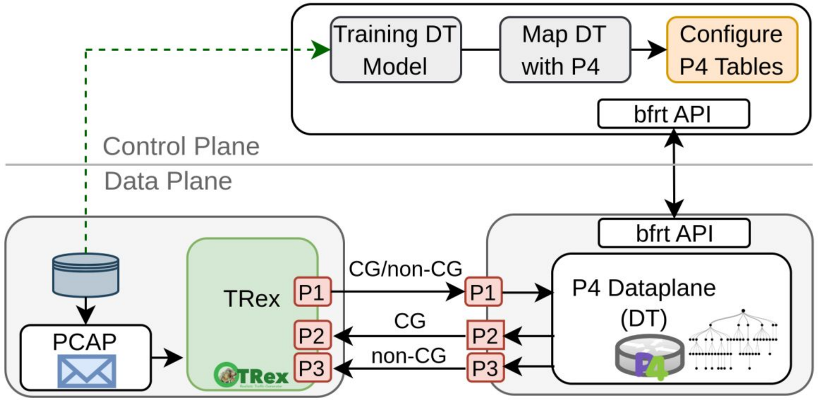
Test Dataset:

- ❖ 80% of above dataset
- ❖ Another dataset, different lab setup (Laris Lab: *https://github.com/dcomp-leris*), displays CG activity from the *Xbox* CG platform (Games: Fortnite, Mortal Kombat, and Forza)
- ❖ Dataset contains two scenarios: 5G wireless and cable internet technology

Implementation (2/2)

Testbed

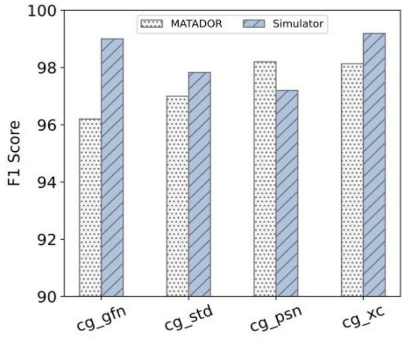
- *x86 server* connected to a P4 Dataplane (Tofino) via 10G SFP+ ports
- Traffic is sent via *TRex*, which is operating on an x86 server
- Python-based controller maps the learned *DT model* with match-action tables in Tofino



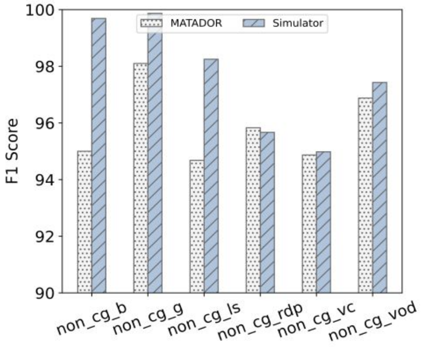
Evaluation (1/3)

F1-score and Accuracy (MATADOR vs Simulator)

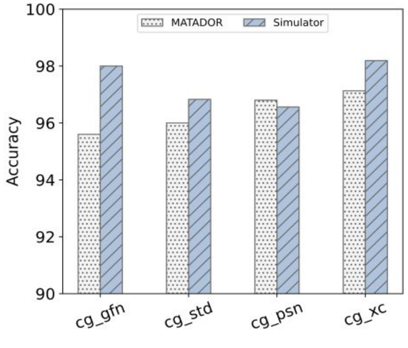
- As expected, because of more exact computations (support floating-point), the *simulator performs better*
- Overall, MATADOR performance is acceptable considering within switch hardware



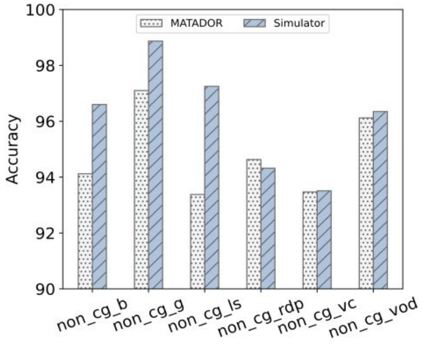
F1 score for CG



F1 score for non-CG



Accuracy for CG



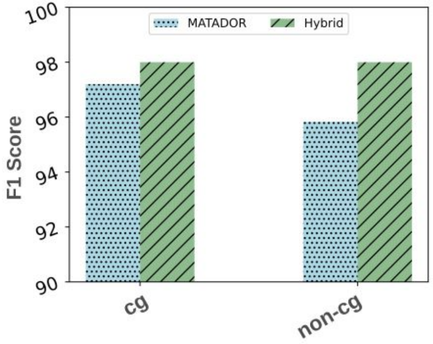
Accuracy for non-CG

Evaluation (2/3)

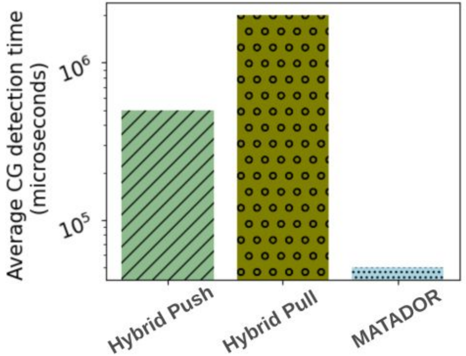
F1-score, Accuracy and Detection time (Compared with SOTA)

- ❖ Results from MATADOR's comparison with the state-of-the-art (CG hybrid) are comparatively close
- ❖ Three distinct time frame sizes—*0.3, 0.5, and 1 seconds*—are used

- ❖ MATADOR can classify each flow in *0.03 second* into CG or non-CG
- ❖ Hybrid push or pull strategy, few seconds for reading registers, apply ML at NFV Node, and push required entries



F1 score for cg and non-cg

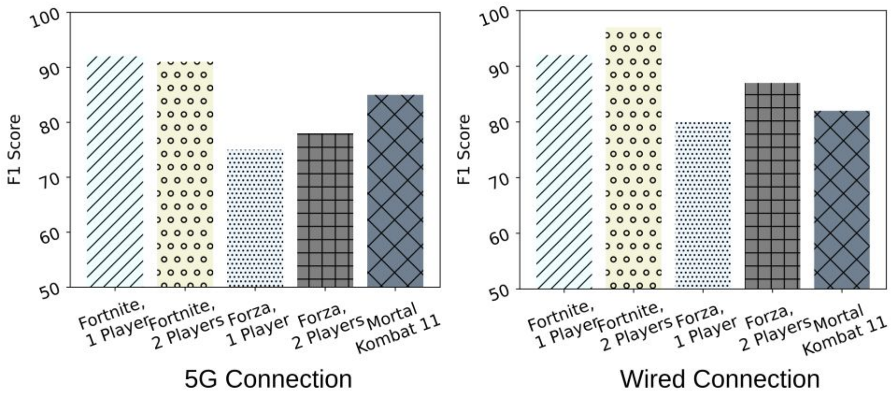


CG detection time (in log scale)

Evaluation (3/3)

Evaluation with new dataset

- ❖ We also evaluate our trained model using a new dataset
- ❖ Findings verify that the model can accurately (80-97%) categorize CG flows (*different dataset*)



Discussion

- Cloud Gaming detection is possible entirely in programmable dataplane
- The proposed approach can be used for *prioritize* CG *to improve* QoE
- Evaluation shows lower detection with higher accuracy
-
- *Other delay critical traffic* needs be identified:

> → Other critical flows (e.g., AR/VR) are required to identifies on line rate and prioritize them
> → The same proposed method can be used for multi-class classification (CG, AR/VR *etc*)

Related Work

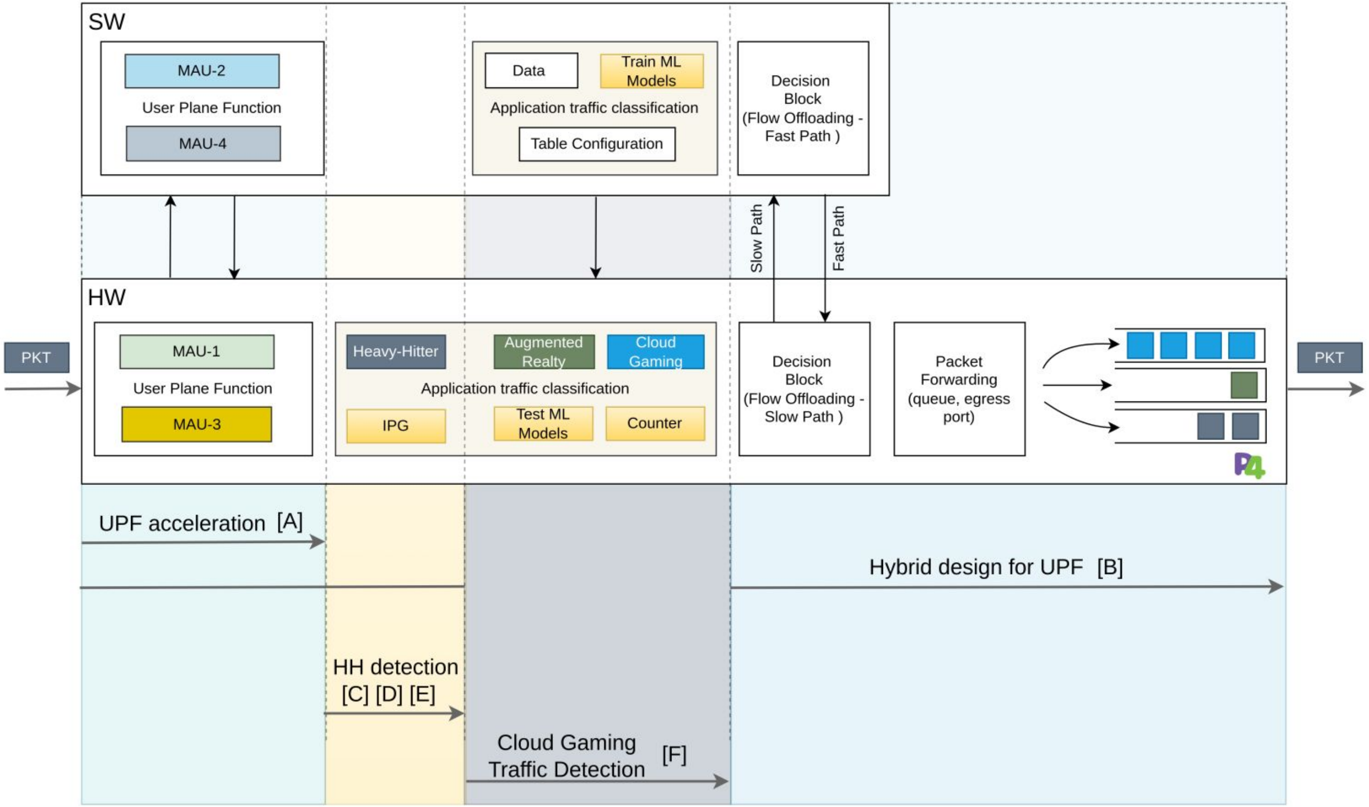
- *CG detection in Software*: Efficient Identification of Cloud Gaming Traffic at the Edge
- *Hybrid approach for CG detection (P4/NFV):* A Hybrid P4/NFV Architecture for Cloud Gaming Traffic Detection with Unsupervised ML

Limitations and Contributions

Limitations

- *On 5G user plane function*
  - Consider other rules (not only HH) for dynamic flows state offloading among programmable targets
  - Explore other programmable targets (e.g., SmartNICs)
  - Missing features (i.e., buffering, proper queue management)
- *On Heavy Hitter detection*
  - Analyze the proposed approach with different datasets
  - Approximate computations in dataplane impacts accuracy
  - Explore more use-cases such as in micro-burst and load balancing
- *On Cloud Gaming Classification*
  - Test with more datasets of CG and non-CG
  - Approximate computations in dataplane impacts accuracy

Contributions



Publications

- *5G user plane function*

  [A] Suneet Kumar Singh, Christian Esteve Rothenberg, Gyanesh Patra, Gergely Pongrácz. "Offloading Virtual Evolved Packet Gateway User Plane Functions to a Programmable ASIC". In 2019 **CoNEXT Workshop on Emerging in-Network Computing Paradigms (ENCP '19)**,, Orlando, FL, USA. December 2019

- *Hybrid 5G user plane function*

  [B] Suneet Kumar Singh, Christian Esteve Rothenberg, Jonathan Langlet, Andreas Kassler, Sandor Laki, Gergely Pongracz. "Hybrid P4 Programmable Pipelines for 5G gNodeB and User Plane Functions". In **IEEE Transactions on mobile computing**. December 2023.

Publications

- *Heavy Hitter detection*

  [C] Suneet Kumar Singh, Christian Esteve Rothenberg, Marcelo Caggiani Luizelli, Gianni Antichi, Gergely Pongrácz. "Revisiting Heavy-Hitters: Don't count packets, compute flow inter-packet metrics in the data plane". In **SIGCOMM Posters.** August, 2020.

  [D] Suneet Kumar Singh, Christian Esteve Rothenberg, Marcelo Caggiani Luizelli, Gianni Antichi, Pedro Henrique Gomes, Gergely Pongrácz. "HH-IPG: Leveraging Inter-Packet Gap Metrics in P4 Hardware for Heavy Hitter Detection". In **IEEE Transactions on Network and Service Management**. September, 2023.

  [E] Suneet Kumar Singh, Christian Esteve Rothenberg, Pedro Henrique GOMES, Gergely Pongrácz. "Heavy hitter flow classification based on inter-packet gap analysis". **PCT Patent Application Serial No**. PCT/IB2021/053738. Filing Date: 02

Publications

❖ *Cloud Gaming Classification*

[F] Suneet Kumar Singh, Christian Esteve Rothenberg, Gergely Pongrácz. "MATADOR: Ml-based Cloud Gaming Traffic Detection entirely in Programmable Hardware". In **IEEE Conference on Network Function Virtualization and Software Defined Networks**. November, 2024.

Further Contributions and Collaborative Activities

- *Cloud Gaming and AR Classification (University Federal de São Carlos)*
  - Alireza Shirmarz, Fábio Luciano Verdi, Suneet Kumar Singh, Christian Esteve Rothenberg. “From Pixels to Packets: Traffic Classification of Augmented Reality and Cloud Gaming”. In 6th IEEE International Conference on Network Softwarization (**NetSoft’24**) - Technical Sessions, MO, USA. Jun, 2024.

  Under Review
  - Alireza Shirmarz, Fábio Luciano Verdi, Suneet Kumar Singh, Christian Esteve Rothenberg. “DCTPQ: Dynamic Cloud Gaming Traffic Prioritization Using Machine Learning and Multi-Queueing for QoE Enhancement”. In **IEEE INFOCOM 2025.**

- *Open Source Project (Internship at Open Networking Foundation and worked at Kaloom)*
  - Contributed in Aether Project. Responsible for Tofino-based QoS solution for UPF

    https://github.com/stratum/fabric-tna, https://github.com/omec-project/up4, https://github.com/stratum/fabric-line-rate-test, https://github.com/opennetworkinglab/onos

Conclusion & Future Work

Summary

What has been done?

- 5G UPF on programmable switch hardware
- Hybrid 5G UPF using different programmable targets
- Heavy Hitter and Cloud Gaming detection on switch hardware

What is going to be done?

- Revisit on the limitations
- IPG metric for capturing more dataplane events
- Expanding ML-based methods for more traffic classes

Acknowledgments

- DCA - FEEC - UNICAMP
  - Christian Rothenberg
- Ericsson Research Hungary
  - Gergely Pongracz
- Ericsson Research Brazil
  - Pedro Henrique Gomes
- Queen Mary University of London (UK)
  - Gianni Antichi

Thank You !

Any Questions ?

https://github.com/intrig-unicamp/cg-classifier

https://github.com/intrig-unicamp/P4-HH

https://github.com/intrig-unicamp/macsad-usecases/tree/master/p4-16/vEPG

FEEC Faculdade de Engenharia Elétrica e de Computação UNICAMP

INTRIG INFORMATION & NETWORKING TECHNOLOGIES RESEARCH & INNOVATION GROUP

https://intrig.dca.fee.unicamp.br/

ERICSSON