An Efficient Design of Intelligent Network Data Plane

Guangmeng Zhou[1], Zhuotao Liu[1,2], Chuanpu Fu[1], Qi Li[1,2], Ke Xu[1,2]

[1]Tsinghua University
[2]Zhongguancun Laboratory

August, 2023

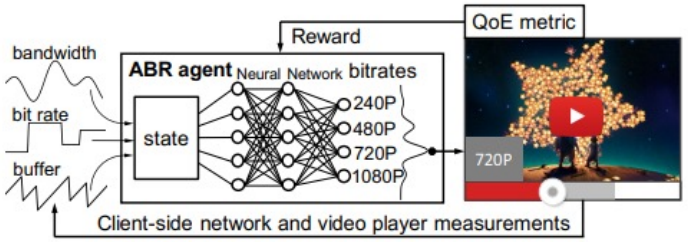
中关村实验室
ZGC Lab

Outline

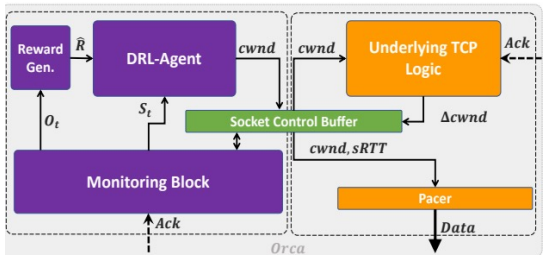
- Background
- Motivation
- Design
- Evaluation
- Conclusion

Background

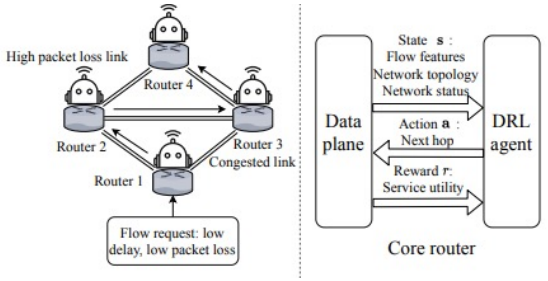
- AI experiences increasingly popularity in network.
  - Model are deployed either on end-hosts or network control plane.



Adaptive Bitrate
End-host



Congestion Control
End-host



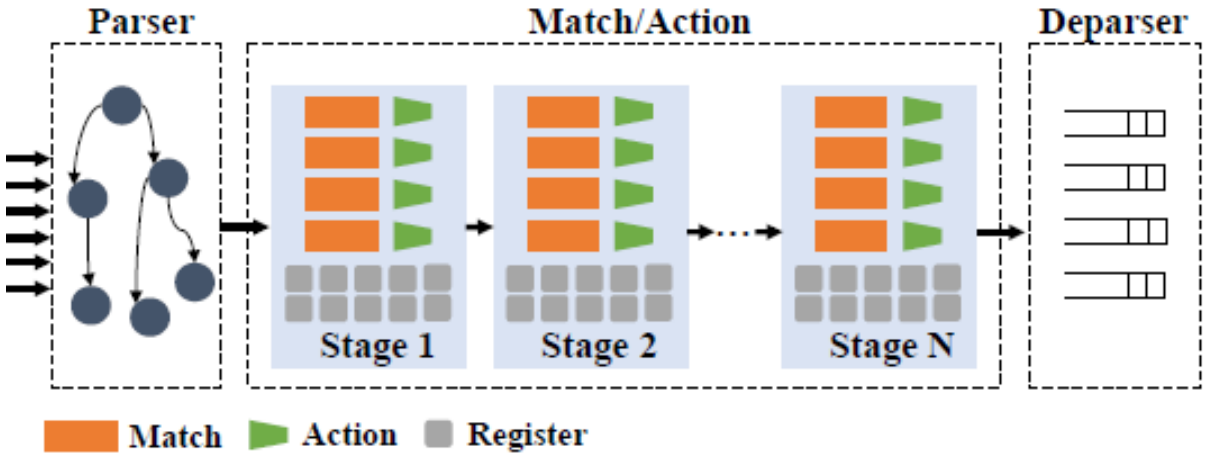
Routing
Control plane

- Intelligent Data Plane (IDP)
  - Malicious Traffic Detection
  - ECN Threshold Adjustment
  - ……

However, IDP is "nearly impossible" using traditional switch chip.

Background

- Programmable switches enable IDP.



Protocol-Independent Switch Architecture (PISA)

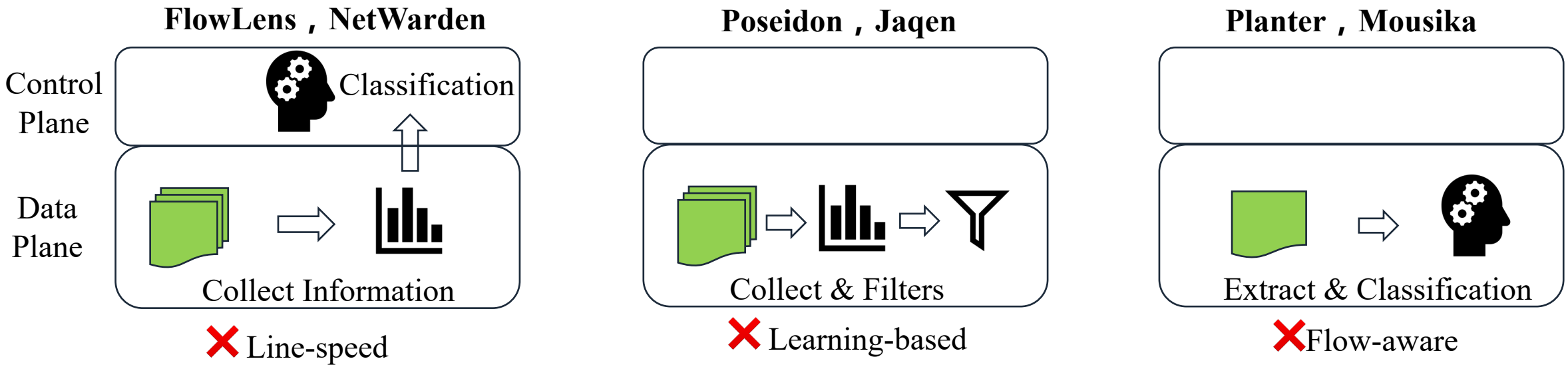
- Customizable Packet Processing
- Stateful and Persistent Storage

- Computation Constraints
  simple operations，limited stages
- Storage Constraints
  once register access，limited storage

Constraints pose many challenges for IDP.

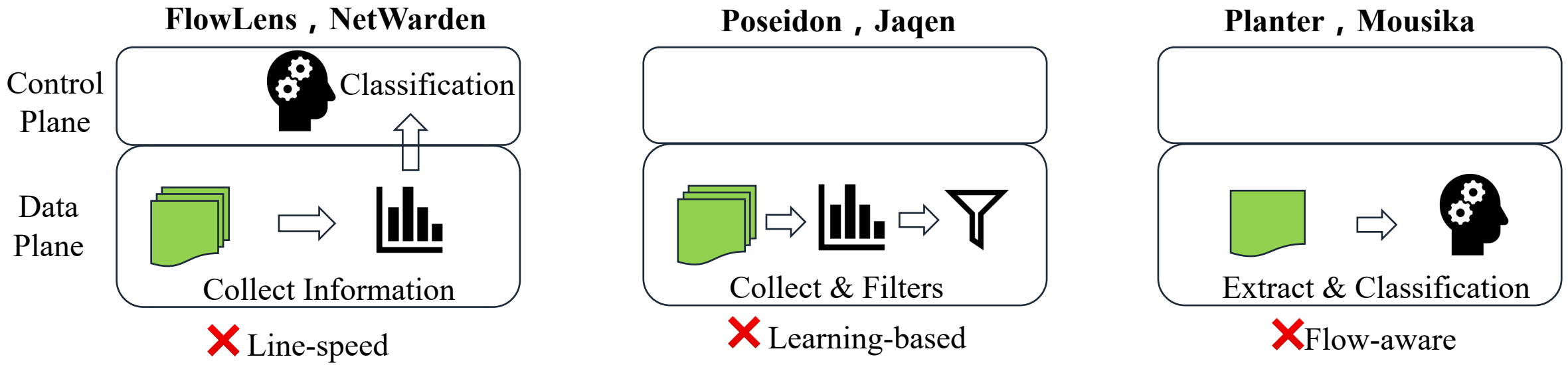
Motivation

- Prior traffic analysis art on programmable switches



Motivation

- Prior art on programmable switches



NetBeacon

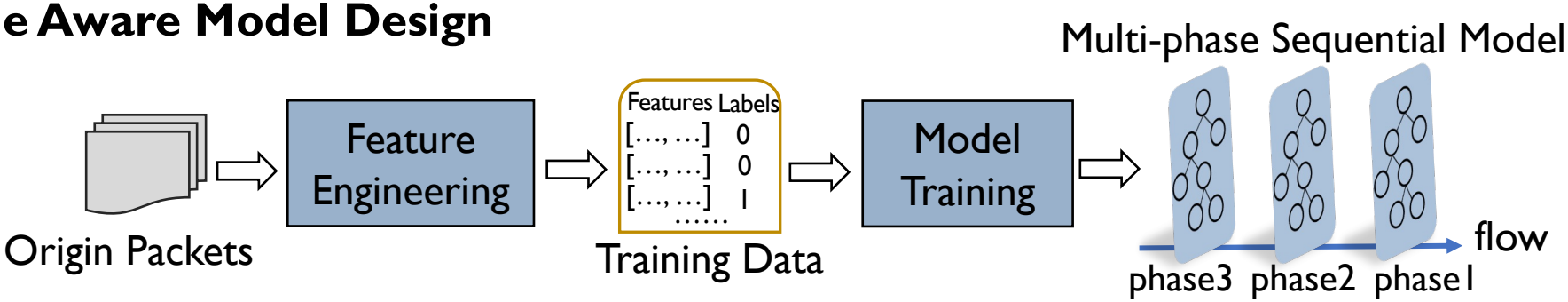
- Line-speed
- Learning-based
- Flow-aware

Design

#1 How to get hardware-friendly learning models?

- Feature Engineering
- Multi-Phase Sequential Models

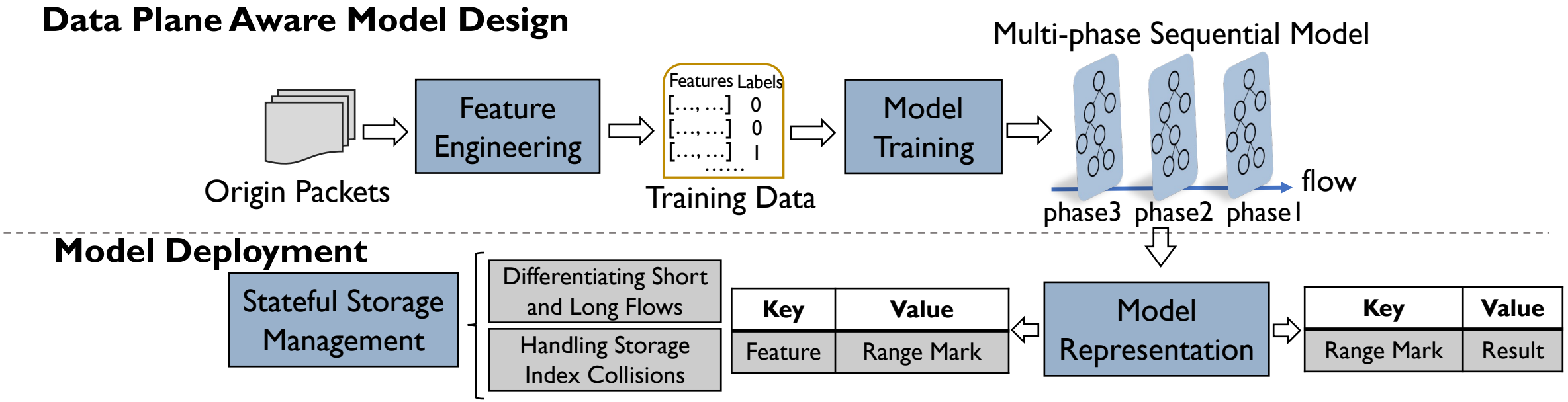
Data Plane Aware Model Design



Design

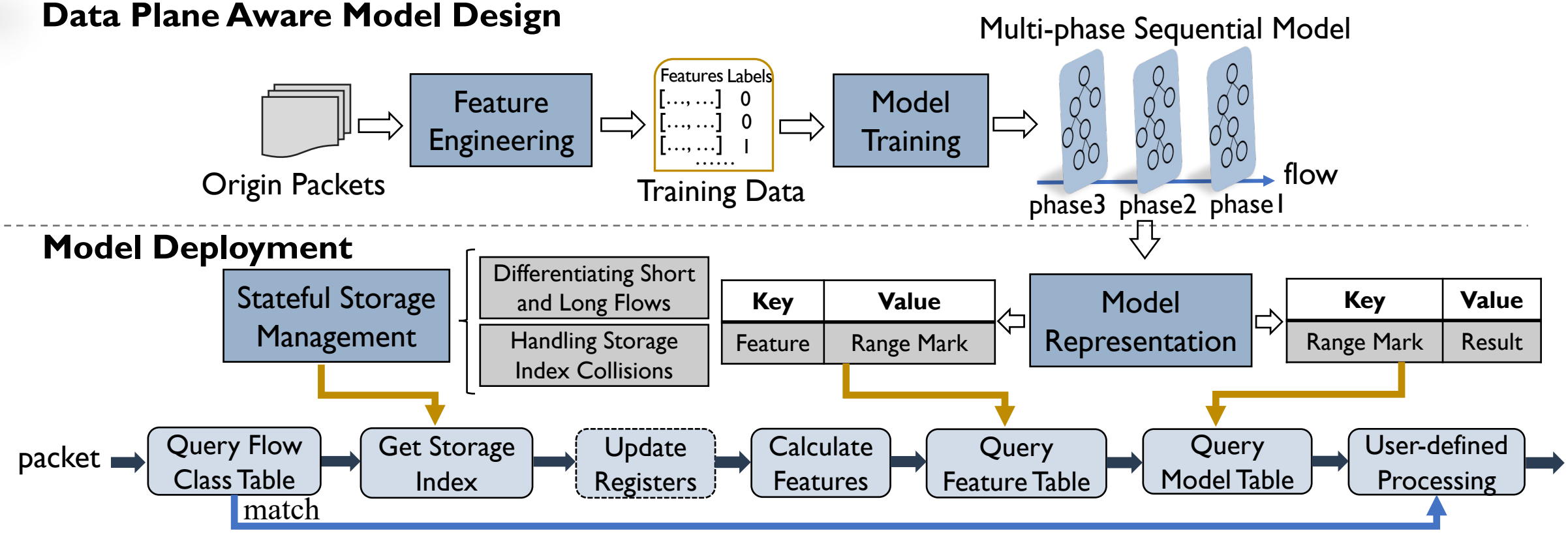
#2 How to deploy learning models efficiently?

- Model Representation
- Stateful Storage Management



Design

Data Plane Aware Model Design



#1 Get hardware-friendly learning models

- Feature Engineering
- Multi-Phase Sequential Models

#2 Deploy learning models efficiently

- Model Representation
- Stateful Storage Management

Design

Feature Engineering

- Features: extractable or computable on the pipeline

Packet-level

ttl, packet size, tcp window, proto, ……

Flow-level

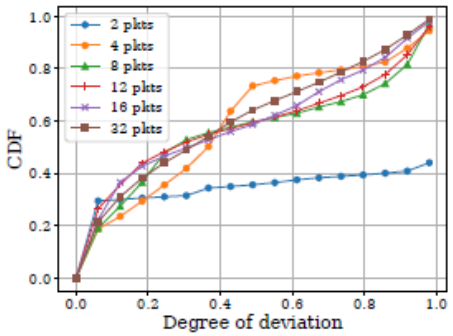
Attributes
Packet size, IPD, …

+

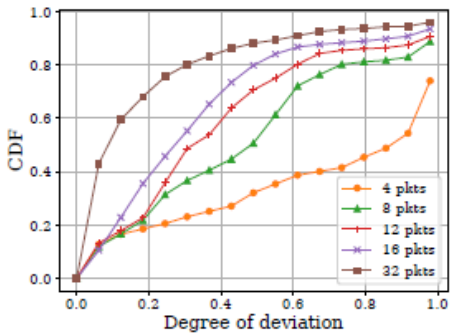
Min, Max, Mean, Var, Distributions,

Multi-Phase Sequential Models

- Model: Decision Tree, e.g., RandomForest, XGBoost, …



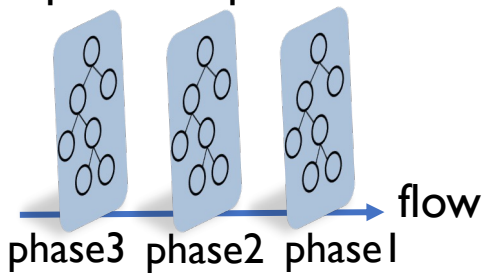
(a) PeerRush, mean of IPD



(b) ISCXVPN, variance of packet size

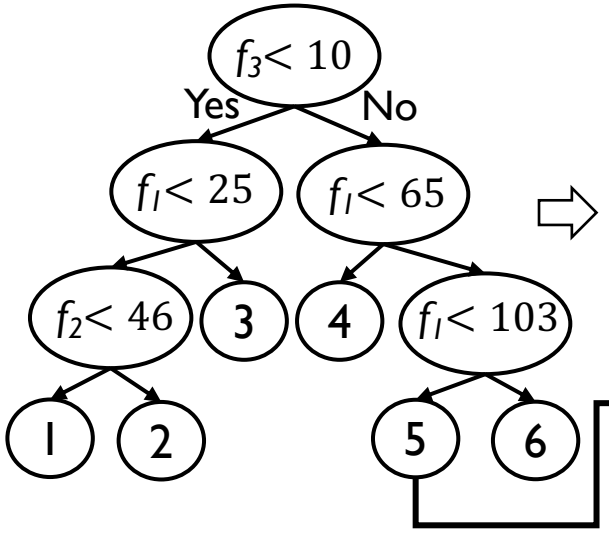
Flow-level features are dynamic as a flow proceeds.

Multi-phase Sequential Model



Design

Model Representation



Key			Value
f_1	f_2	f_3	
[0, 25)	[0, 46)	[0,10)	1
[0, 25)	[46, 256)	[0,10)	2
[25, 256)	[0, 256)	[0,10)	3
[0, 65)	[0, 256)	[10,256)	4
[65, 103)	[0, 256)	[10,256)	5
[103, 256)	[0, 256)	[10,256)	6

(1) Model table under range matching

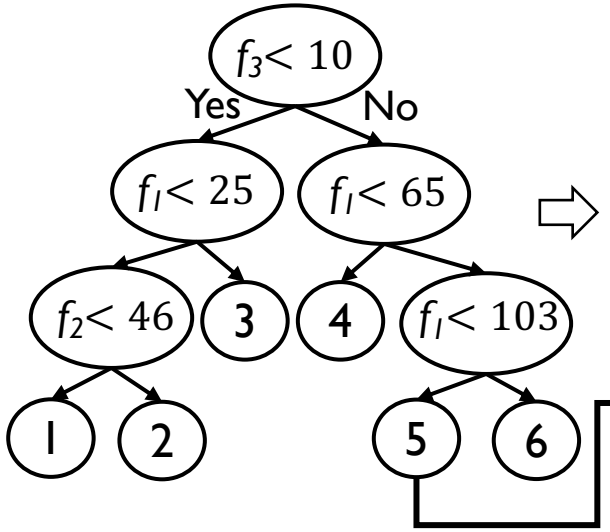
Not supported by the switch

Key			Value
f_1	f_2	f_3	
...	...	...	...
01000001 [65,66)			
0100001* [66,68)		0000101* [10,12)	
010001** [68,72)		000011** [12,16)	
01001*** [72,80)	********	0001**** [16,32)	5
0101**** [80,96)	[0,256)	001***** [32,64)	(8×1×6)
011000** [96,100)		01****** [64,128)	
0110010* [100,102)		1******* [128,256)	
01100110 [102,103)			
...	...	...	...

(2) Model table under ternary matching

Combinatorial explosion of entries

bit<8> f_1, bit<8> f_2, bit<8> f_3



Key f_1	Key f_2	Key f_3	Value
[0, 25)	[0, 46)	[0,10)	1
[0, 25)	[46, 256)	[0,10)	2
[25, 256)	[0, 256)	[0,10)	3
[0, 65)	[0, 256)	[10,256)	4
[65, 103)	[0, 256)	[10,256)	5
[103, 256)	[0, 256)	[10,256)	6

(1) Model table under range matching

Not supported by the switch

Key f_1	Key f_2	Key f_3	Value
...	...	...	...
01000001 [65,66) 0100001* [66,68) 010001** [68,72) 01001*** [72,80) 0101**** [80,96) 011000** [96,100) 0110010* [100,102) 01100110 [102,103)	******** [0,256)	0000101* [10,12) 000011** [12,16) 0001**** [16,32) 001***** [32,64) 01****** [64,128) 1******* [128,256)	5 (8×1×6)
...	...	...	...

(2) Model table under ternary matching

Combinatorial explosion of entries

Our way

Range Marking

f_1: 0 25 65 103 256

Key (f_1)	Value (Range Mark)
[0,25)	111
[25,65)	011
[65, 103)	001
[103,256)	000

(a_1) Feature table (f_1)

f_2: 0 46 256

Key (f_2)	Value
[0,46)	1
[46, 256)	0

(a_2) Feature table (f_2)

f_3: 0 10 256

Key (f_3)	Value
[0,10)	1
[10, 256)	0

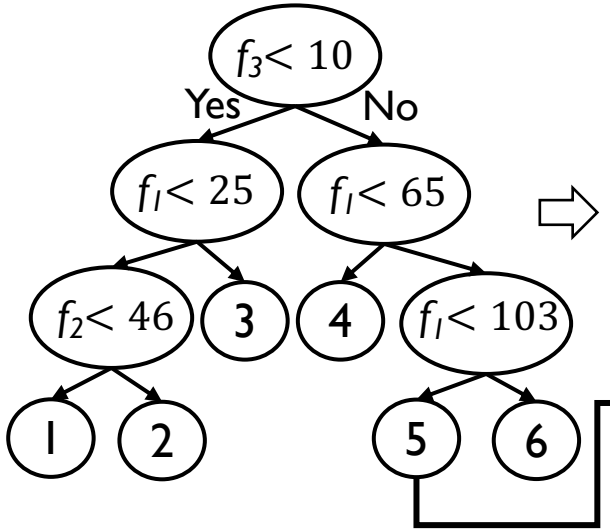
(a_3) Feature table (f_3)

Key (Range Mark) f_1	f_2	f_3	Value
111	1	1	1
111	0	1	2
0**	*	1	3
*11	*	0	4
001	*	0	5
000	*	0	6

(b) Model table

Each leaf node only consumes a single ternary entry in the model table.

bit<8> f_1, bit<8> f_2, bit<8> f_3



Key f_1	Key f_2	Key f_3	Value
[0, 25)	[0, 46)	[0,10)	1
[0, 25)	[46, 256)	[0,10)	2
[25, 256)	[0, 256)	[0,10)	3
[0, 65)	[0, 256)	[10,256)	4
[65, 103)	[0, 256)	[10,256)	5
[103, 256)	[0, 256)	[10,256)	6

(1) Model table under range matching

Not supported by the switch

Key f_1	Key f_2	Key f_3	Value
...	...	...	...
01000001 [65,66)	******** [0,256)	0000101* [10,12)	5 (8×1×6)
0100001* [66,68)		000011** [12,16)	
010001** [68,72)		0001**** [16,32)	
01001*** [72,80)		001***** [32,64)	
0101**** [80,96)		01****** [64,128)	
011000** [96,100)		1******* [128,256)	
0110010* [100,102)			
01100110 [102,103)			
...	...	...	...

(2) Model table under ternary matching

Combinatorial explosion of entries

Our way

f_1 0 25 65 103 256

f_2 0 46 256

f_3 0 10 256

Key (f_1)	Value (Range Mark)
[0,25)	111
[25,65)	011
[65, 103)	001
[103,256)	000

(a_1) Feature table (f_1)

Key (f_2)	Value
[0,46)	1
[46, 256)	0

(a_2) Feature table (f_2)

Key (f_3)	Value
[0,10)	1
[10, 256)	0

(a_3) Feature table (f_3)

0000**** [0,16)	111
00010*** [16,24)	111
00011000 [24,25)	111
00****** [0,64)	011
01000000 [64,65)	011
01100111 [103,104)	000
010***** [64,96)	001
01100*** [96,104)	001

Higher Priority

Entries of a_1 from CRC algorithm

Key (Range Mark) f_1	Key (Range Mark) f_2	Key (Range Mark) f_3	Value
111	1	1	1
111	0	1	2
0**	*	1	3
*11	*	0	4
001	*	0	5
000	*	0	6

(b) Model table

Design

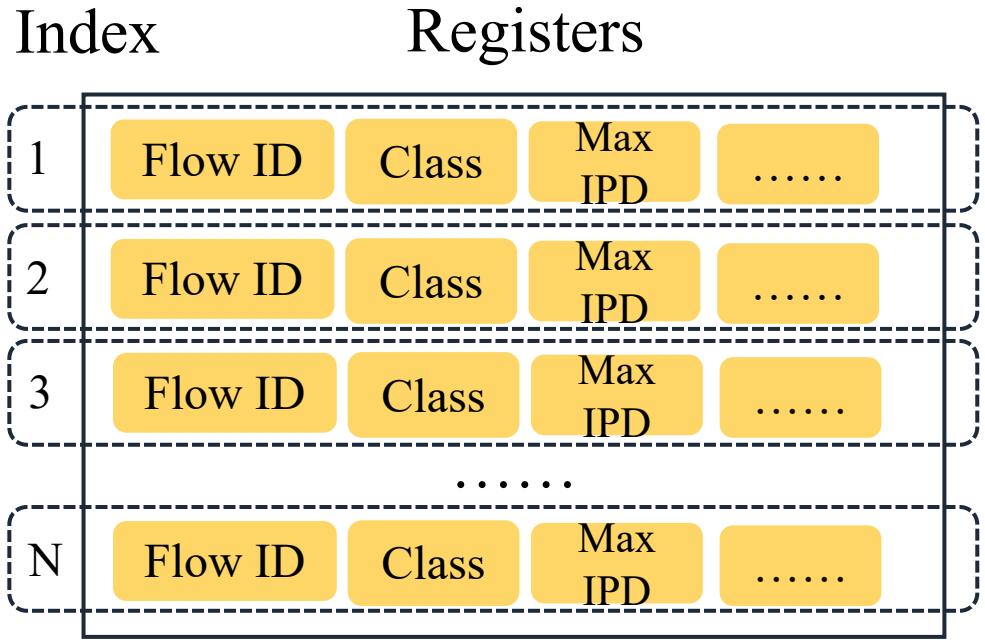
Stateful Storage Management

- Index = Hash (5-tuple)

⬇

Index Collision!

- Solutions



Differentiating Short and Long Flows

- Network traffic is skewed.
  - ✓ Long flows: flow-level features
  - ✓ Short flows: per-packet features

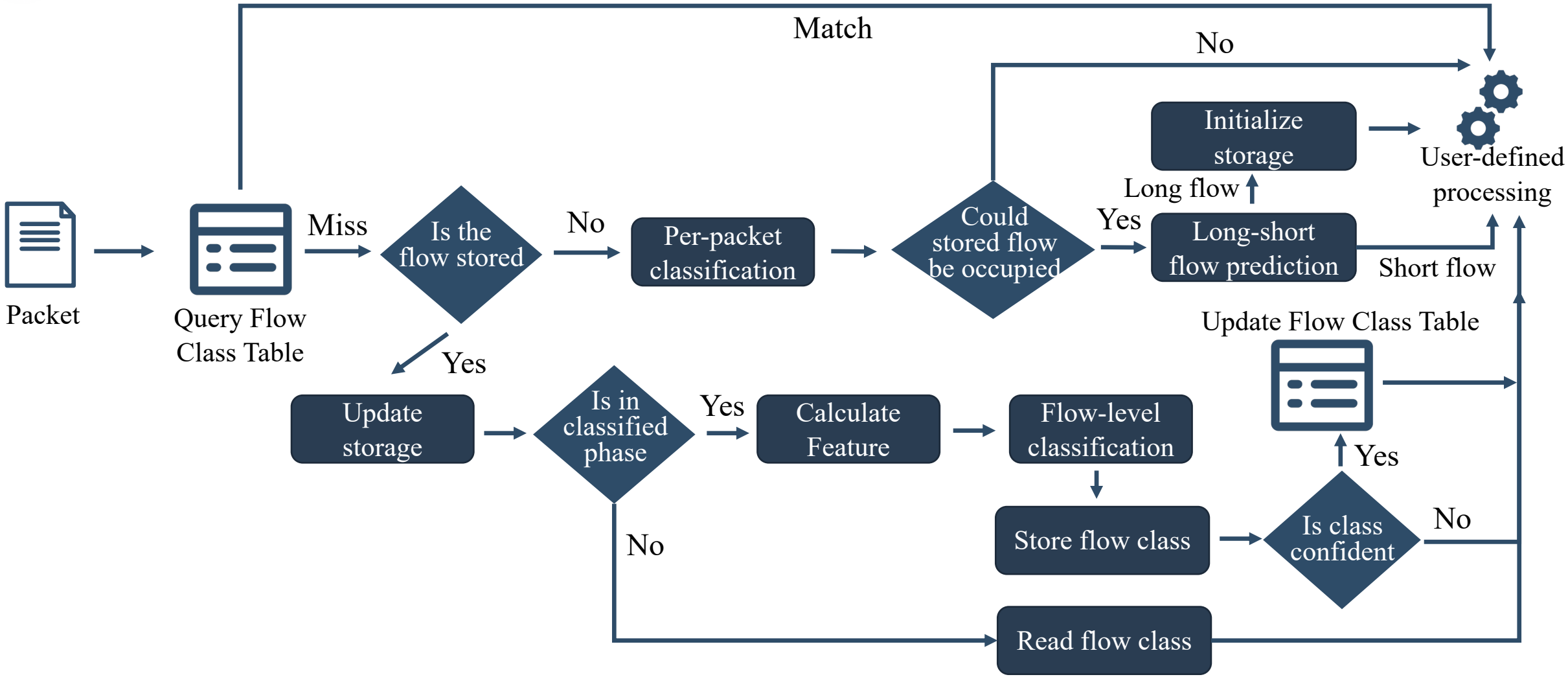
✚ Long-short flow classification model

Handling Storage Index Collisions

- New flow can use the occupied register if
  1. Old flow class is determined confidently.
  2. Old flow is finished.

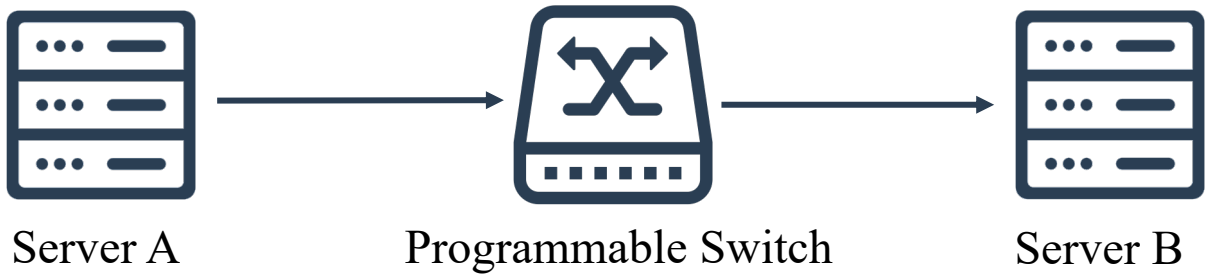
Design

- Integrated Data Plane Processing Logic



Evaluation

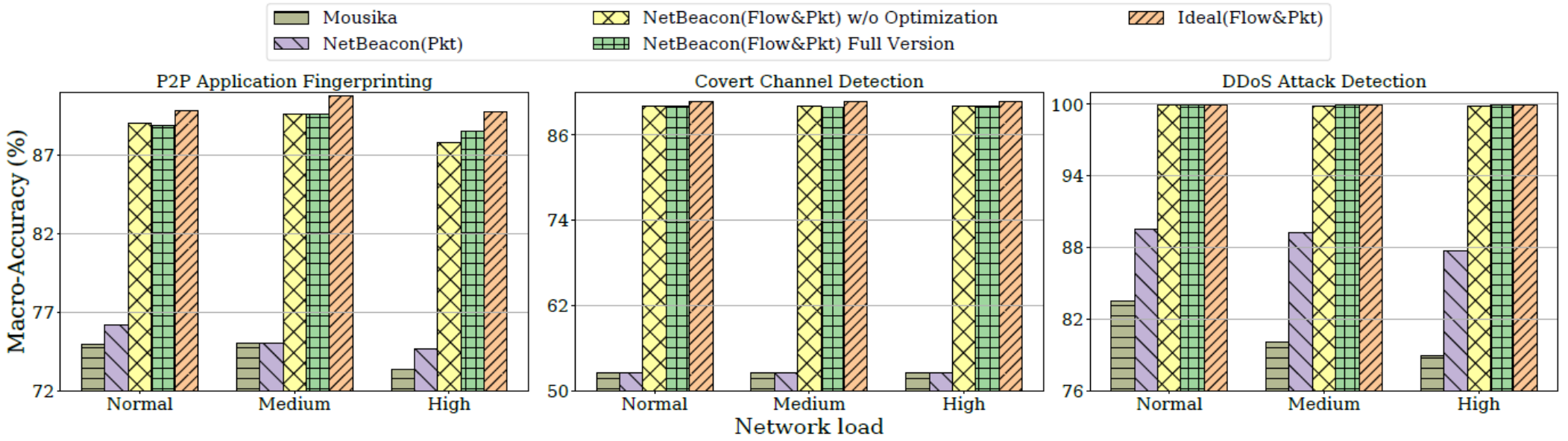
- Setup



- Metric: Packet-level macro-accuracy
- Tasks:
  - P2P Application Fingerprinting
  - Covert Channel Detection
  - DDoS Attack Detection
- Baseline:
  - Mousika

Evaluation

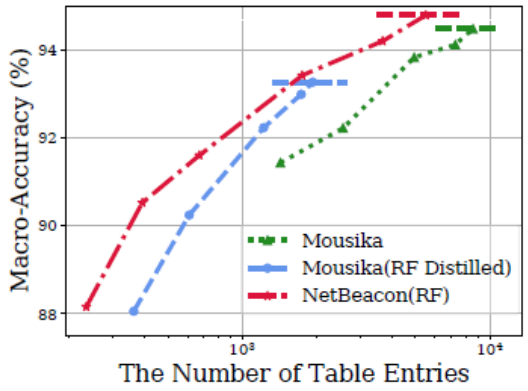
- End-to-end



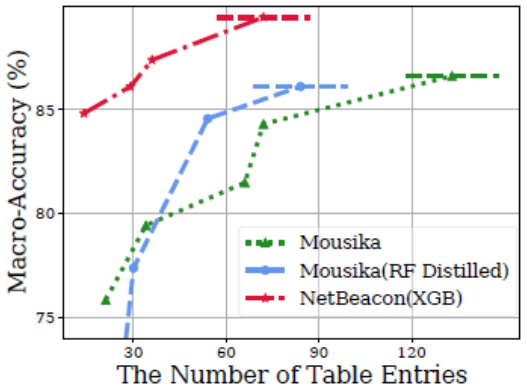
- Compared to Mousika, NetBeacon has a significant accuracy improvement, i.e., 14% in P2P application fingerprinting, 38% in covert channel detection and 20% in DDoS attack detection.

Evaluation

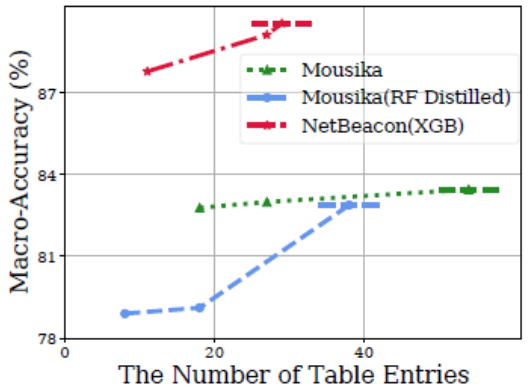
Model Representation



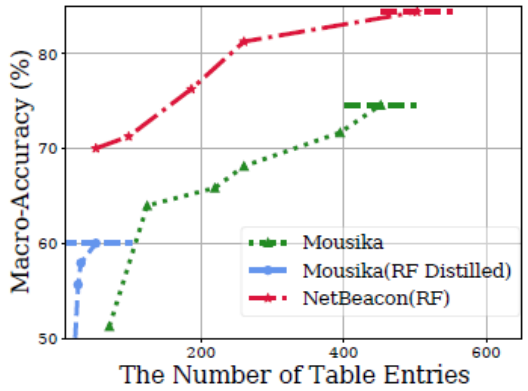
(a) P2P Application Fingerprinting



(b) Covert Channel Detection



(c) DDoS Attack Detection

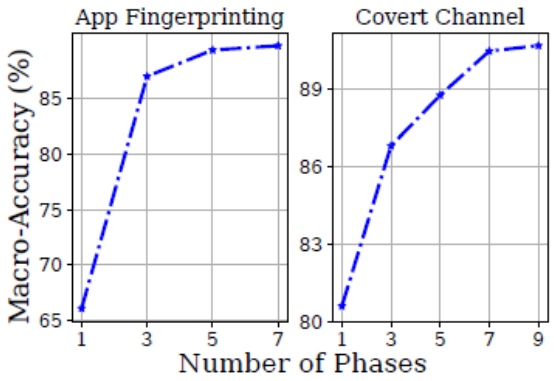


(d) Traffic Type Classification

- NetBeacon achieves higher accuracies when consuming the same number of table entries.
- NetBeacon uses much fewer table entries when representing similarly-performing models.

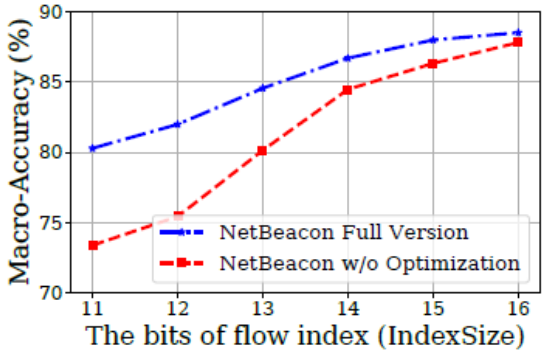
Evaluation

Deep Dive



(a) The impact of number of phases.

- The flow-level information could be premature at earlier phase.
- The accuracy increases as more inference phases are appended.



(c) The impact of the index size.

- The storage size is important for traffic classification.
- The stateful management optimization is useful upon the limited storage.

Conclusion

- An efficient IDP design which outperforms prior art in both model accuracy and representation efficiency.
  - Data plane aware model
  - Efficient model representation
  - Stateful storage management

- Source code: https://github.com/IDP-code/NetBeacon
- Email: zgm19@mails.tsinghua.edu.cn

Thanks! Questions?