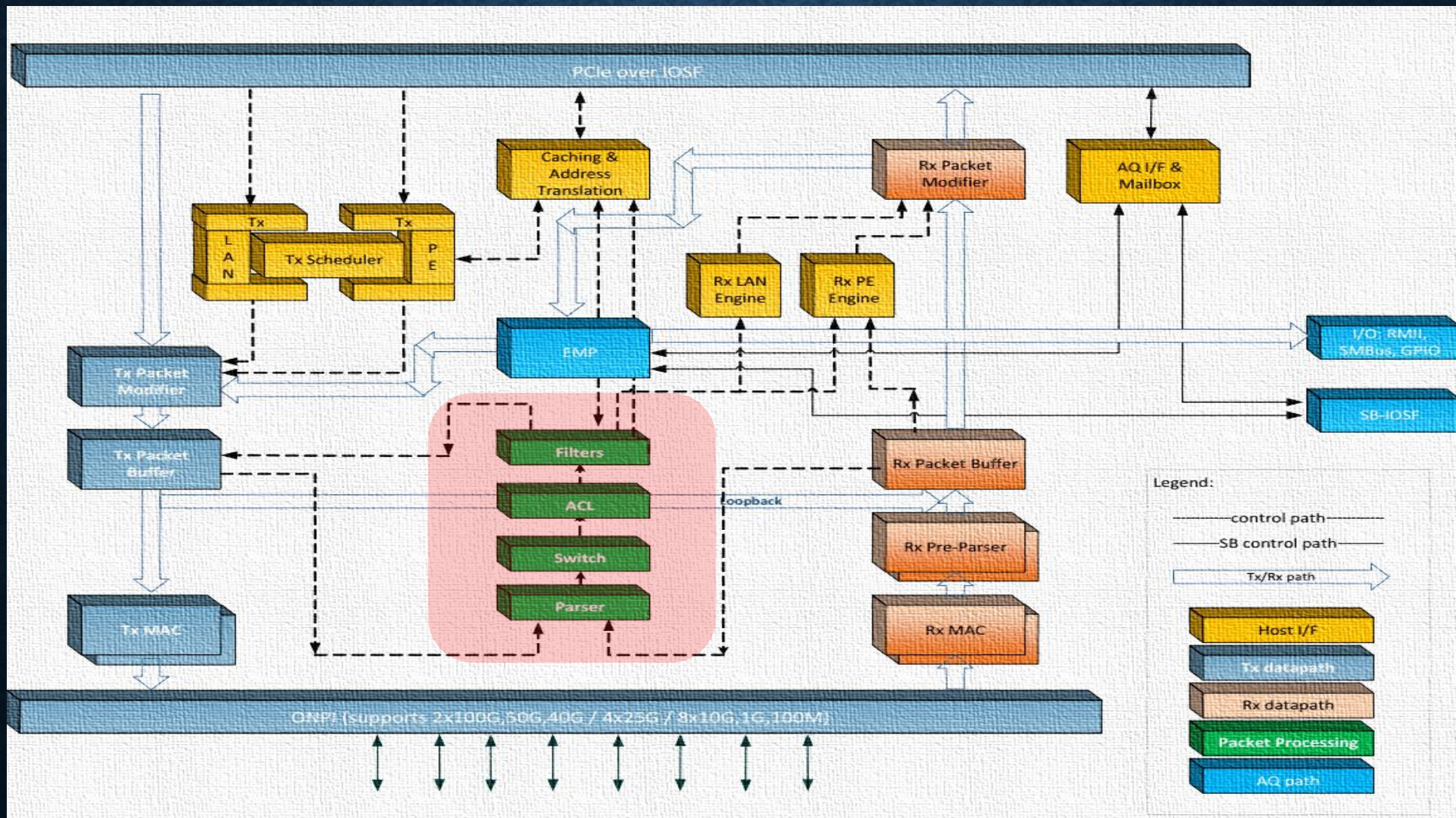


CVL PACKET PROCESSING

Zhang Qi

2020-7

CPK IP BLOCK OVERVIEW

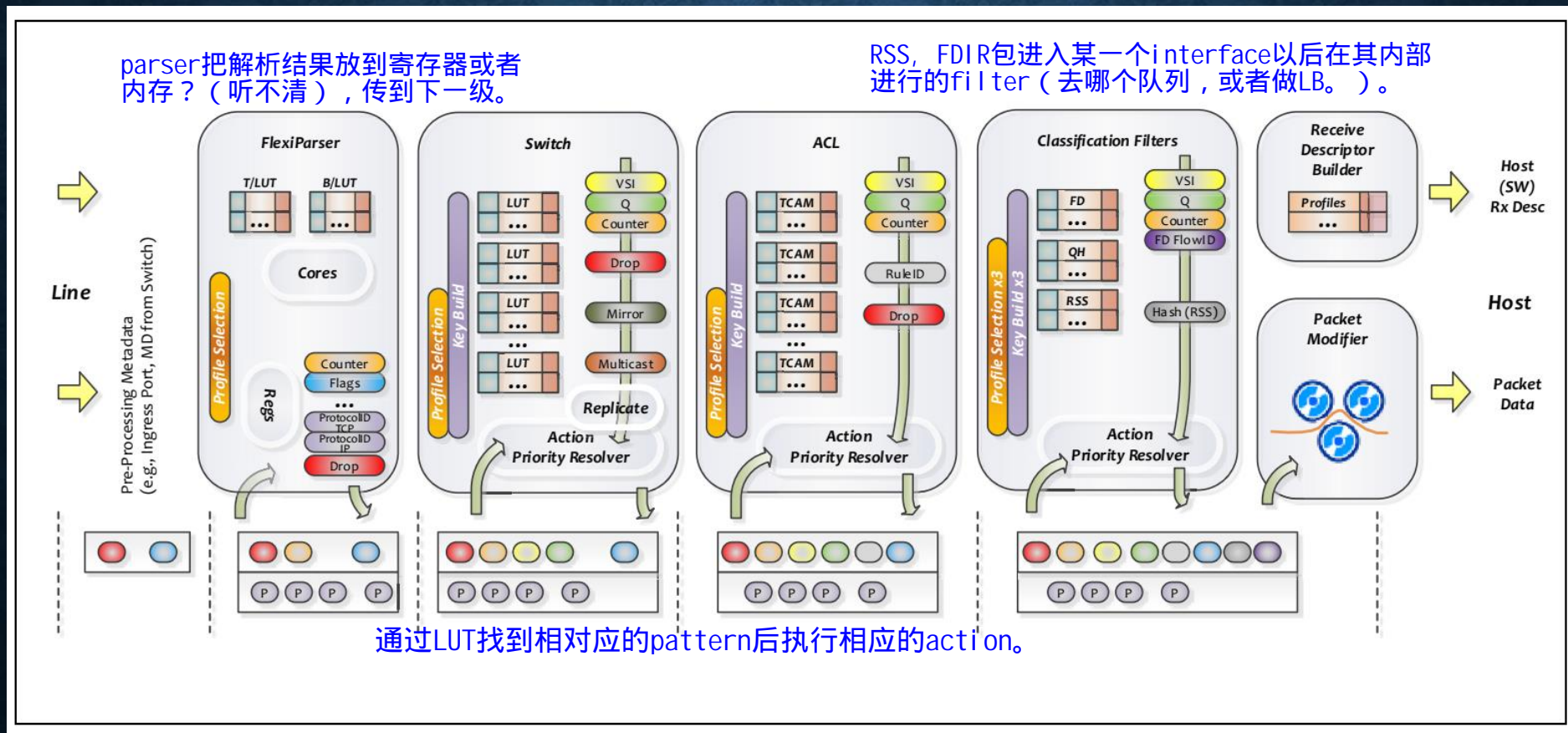


KEY FEATURES

- Steer specific packets to specific workload
- Distribute packets in specific workload for loading balance
- Count statistic for specific workload
- Drop specific packet according to FW policy.
- Mark specific packet.
- Extract packet data into metadata
- Translate packet between network domains

PACKET PROCESSING PIPELINE

- Its all about Key Lookup and Action in a pipeline



PARSER

硬件的parser，用于解析包的。

- Key Components:

- **Parse Graph**

- Represent protocol sequence

- Boost TCAM

- Pattern to match Packet Header to auxiliary Parse Graph to next node.

- Instruction Memory

- ALU instructions.

- Output

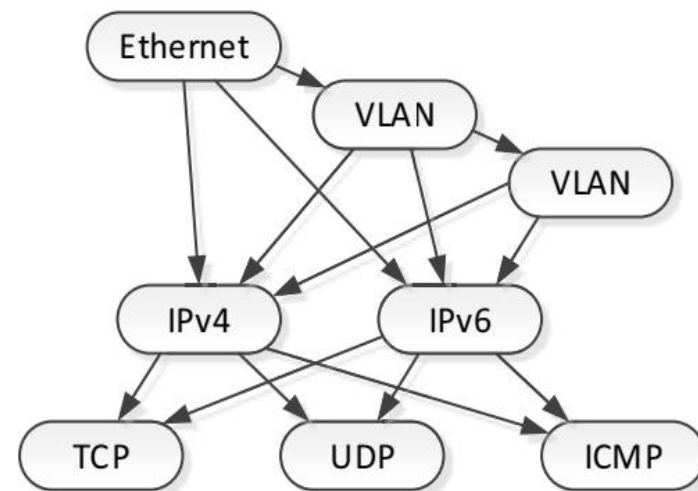
是一个10bit的值。

- **Packet Type Identification (PTYPE)**

- Protocol Start Offset

- Other Metadata

flag, mdi d等

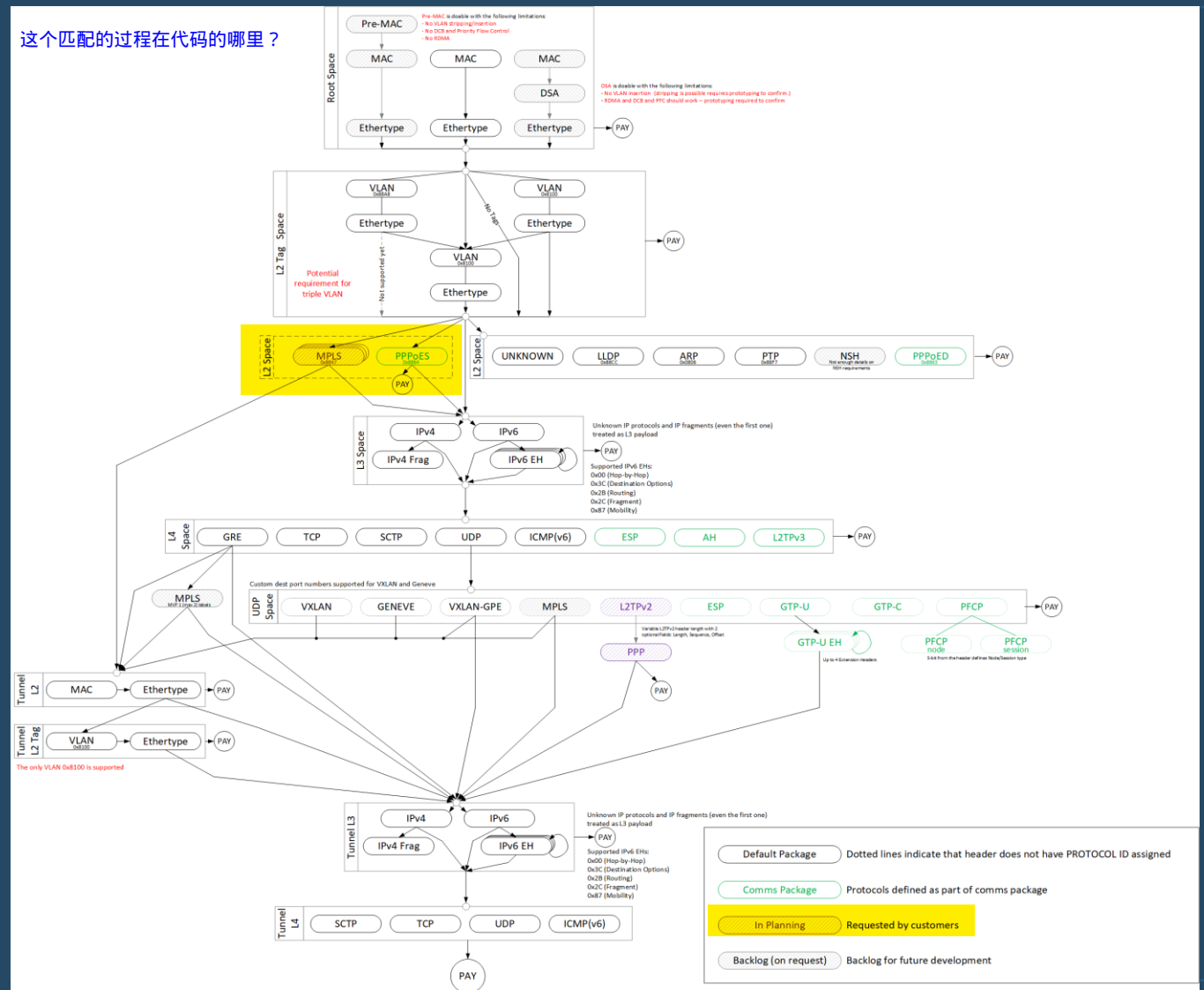


	1	2	3	4
PROTO	ETH	IPv4	UDP	PAYLOAD
START OFFSET	0	12	32	40

CPK PARSE GRAPH

- <https://wiki.ith.intel.com/display/NACSoftware/DDP+Parse+Graph>

这个匹配的过程在代码的哪里？



HOW TO DESCRIBE A RULE?

```
flow create 0 ingress eth / ipv4 src is 1.1.1.1 / udp / end actions queue index 0 end / end
```

Interface

Direction

Packet Type

Input Set Schema

Input Set Value

Action

RULE ID	Lookup KEY	Action
#1	<source, direction, pkt type, input set schema, match value>	action A
#2	<source, direction, pkt type, input set schema, match value>	action B
#3	<source, direction, pkt type, input set schema, match value>	action C
...

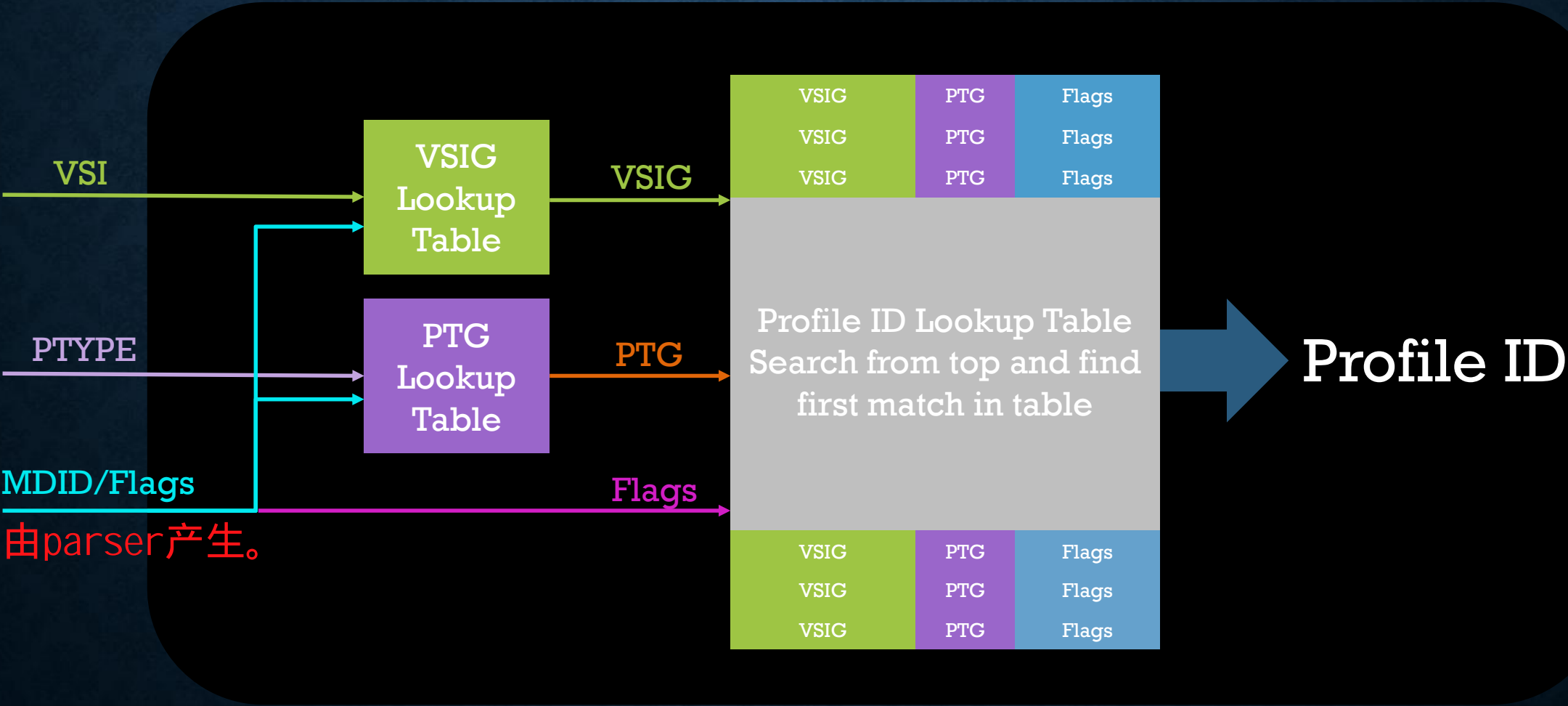
PROBLEM

- Not efficient for resource usage
 - Need to create multiple rules for different packet type / source if we just want to match the same pattern, example: To match all IPv4 packet's source address, we need to create rules for IPv4-UDP, IPv4-TCP, IPv4-other separately.
- Solution:
 - Profile based LUT

参考HAS 7.13.9.

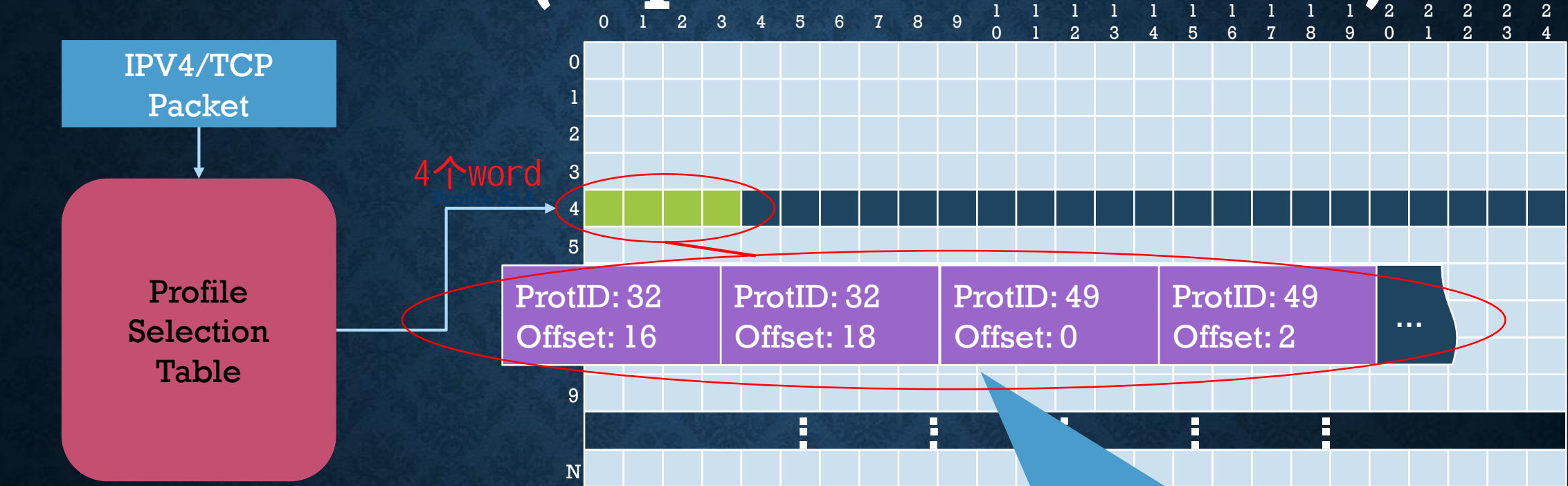
给每个包定义一个profile(由package type, metadata)

PROFILE LOOKUP



profile和input set绑定，一个profile有一个绑定的input set.

Field Vector (Input set Selection)



Protocol ID	Description
32	IPV4
49	TCP

This field vector extracts:
IPV4 dest address (32-bits)
TCP source port (16-bits)
TCP destination port (16-bits)

PROFILE BASED LUT

Lookup Key

<

direction

interface

packet type

input set schema

input set value

>



Lookup Key

<

direction

Profile

input set value

>

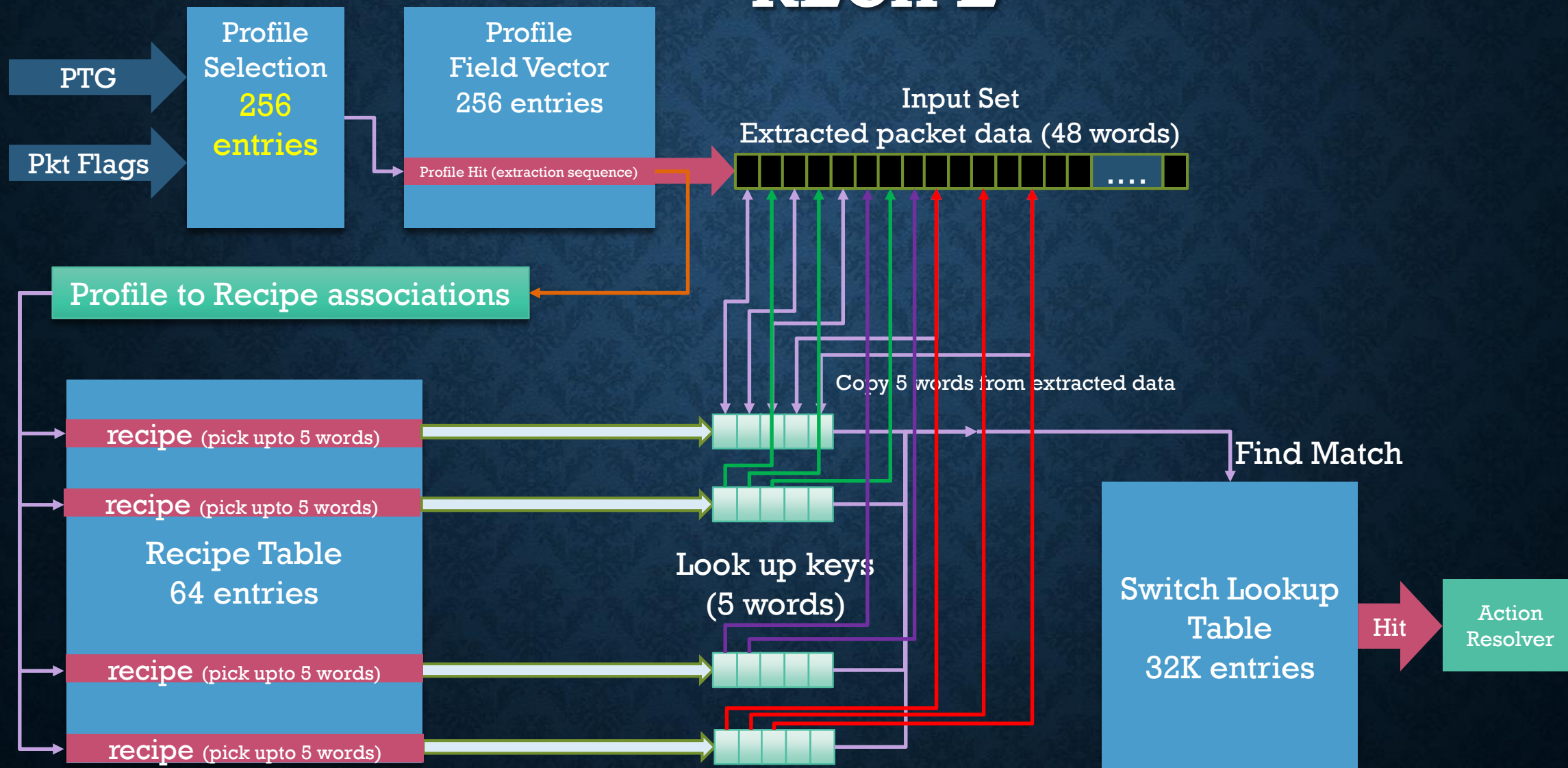
PROBLEM

- How to create **2 rules for same profile** but different input set? E.g.:
 - flow create 0 ingress eth / ipv4 **src is 1.1.1.1** / end actions queue index 0 end / end
 - flow create 0 ingress eth / ipv4 **dst is 2.2.2.2** /end actions queue index 0 end / end
- Each Pipeline stage's answer: 不同的stage使用不同的解决方案。

Switch	You can use Recipe
ACL	I have TCAMs
FDIR	Sorry, it is a limitation
RSS	Not a valid problem

RECIPE

同一个profile通过recipe实现不同的input set。



一个profile可以和一组recipe关联。

每个profile有48个words，recipe进行选择。选择不同的word组成不同的recipe。

RECIPE BASED LUT

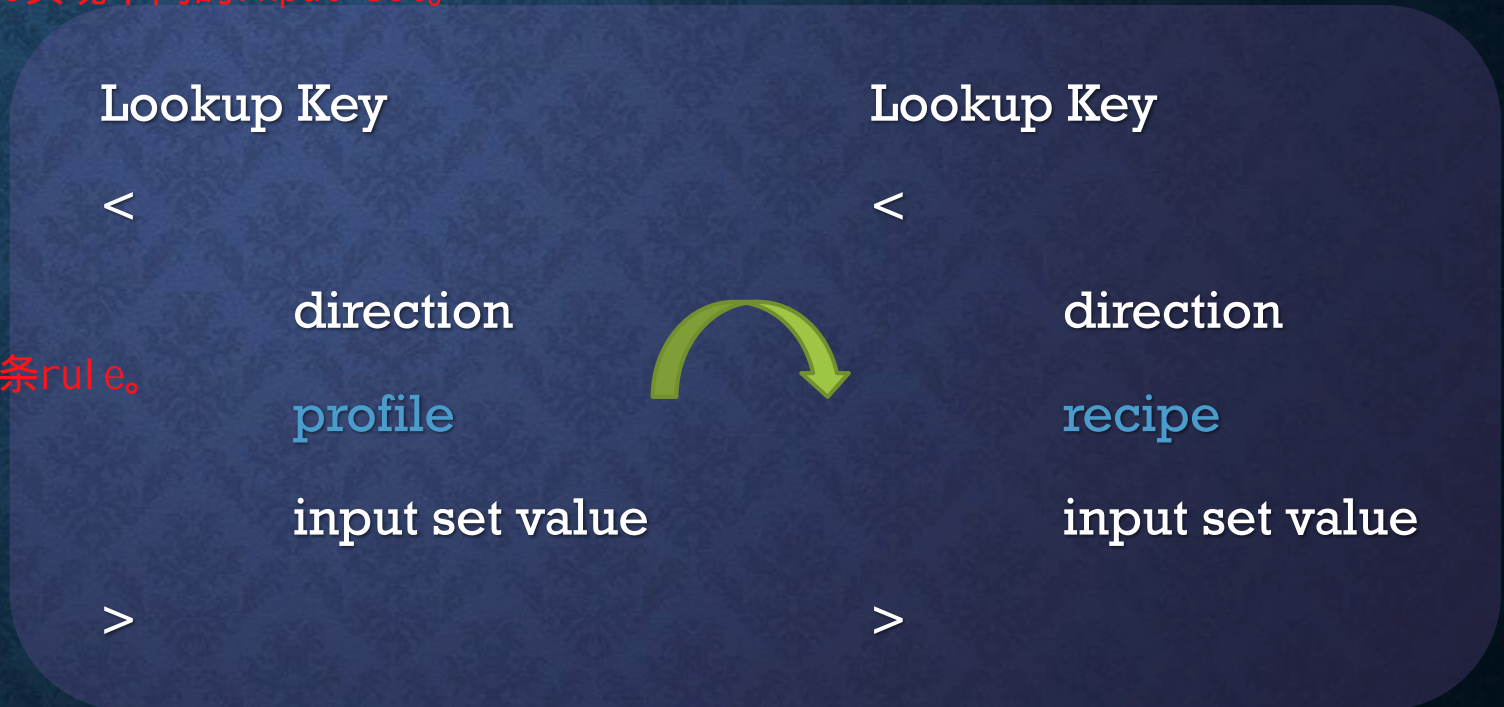
- **Benefit:** 不同的rule, 用同一个profile实现不同的input set。

- Different rules can apply on same profile with different input set.
- Different profile can share the same rule, if the required input set has same field vector index.

- **Limitation:**

- Only match 5 words per recipe.
- Total 64 recipes

不同的profile可以使用一条rule。



CHAINED RECIPE

把recipe串起来组成更大的LU key。

Dependencies indicate chain

Index	Recipe Dependencies	Result Index	Lookups (FV index)				
...							
12	0x00000000000001000	47	0	1	2	3	4
...							
14	0x00000000000004000	46	0	5	6	7	8
15	0x00000000000008000	45	0	9	10	11	12
...							
20	0x0000000000010D000	NA	45	46	47		
...							

Recipes 12, 14, 15 and 20 are associated with Profile 10

recipe : 最多20个word

Root Recipe

Profile 10 - extraction

recipe可以选择parser出来的任何字段，用任何mask实现。

driver会去判断是否已经有匹配的recipe了。对switch来说，不会去建profile。

0 1 2 3 4 5 6 7 8 9 1 1 1 1 1 1 1 1 1 1 2 2 2 2 2 2 2 2 2 2 3 3 3 3 3 3 3 3 3 3 4 4 4 4 4 4 4 4
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7

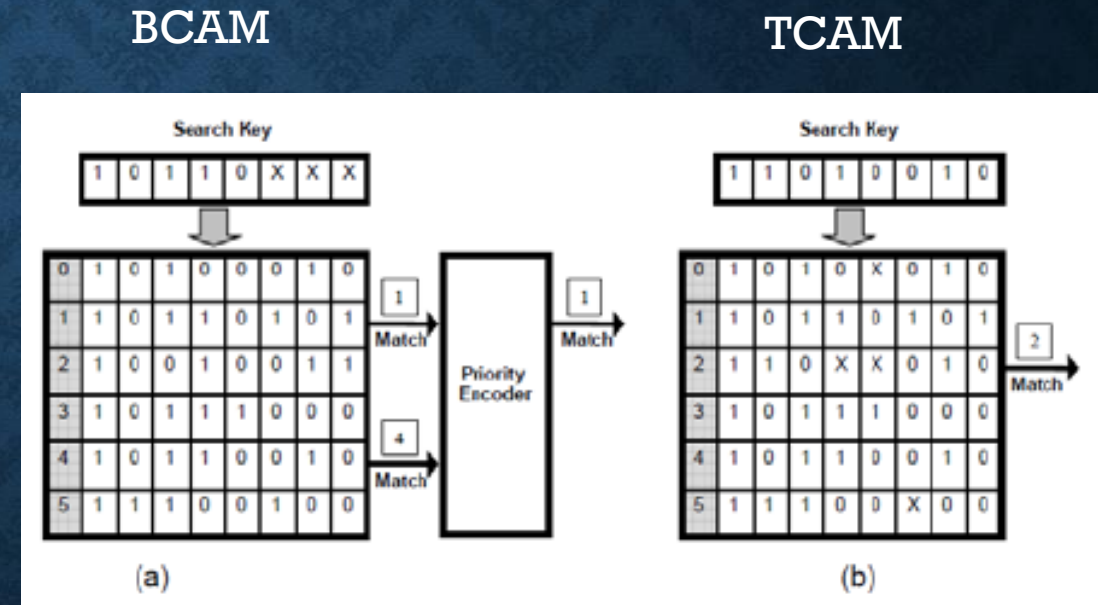


Q&A : 不能下两条一样的rule。如果look up key一样的话会有conflict。recipe有优先级，如果一条rule匹配了两个recipe...

RAM VS. CAM

- RAM (Random-Access Memory)
 - Input Address / Output value
- CAM (Content-Addressable Memory)
 - Input Value / Output Address
- Binary CAM
 - Exact match CAM
- Ternary CAM
 - Wildcard match CAM

ACL使用的就是TCAM

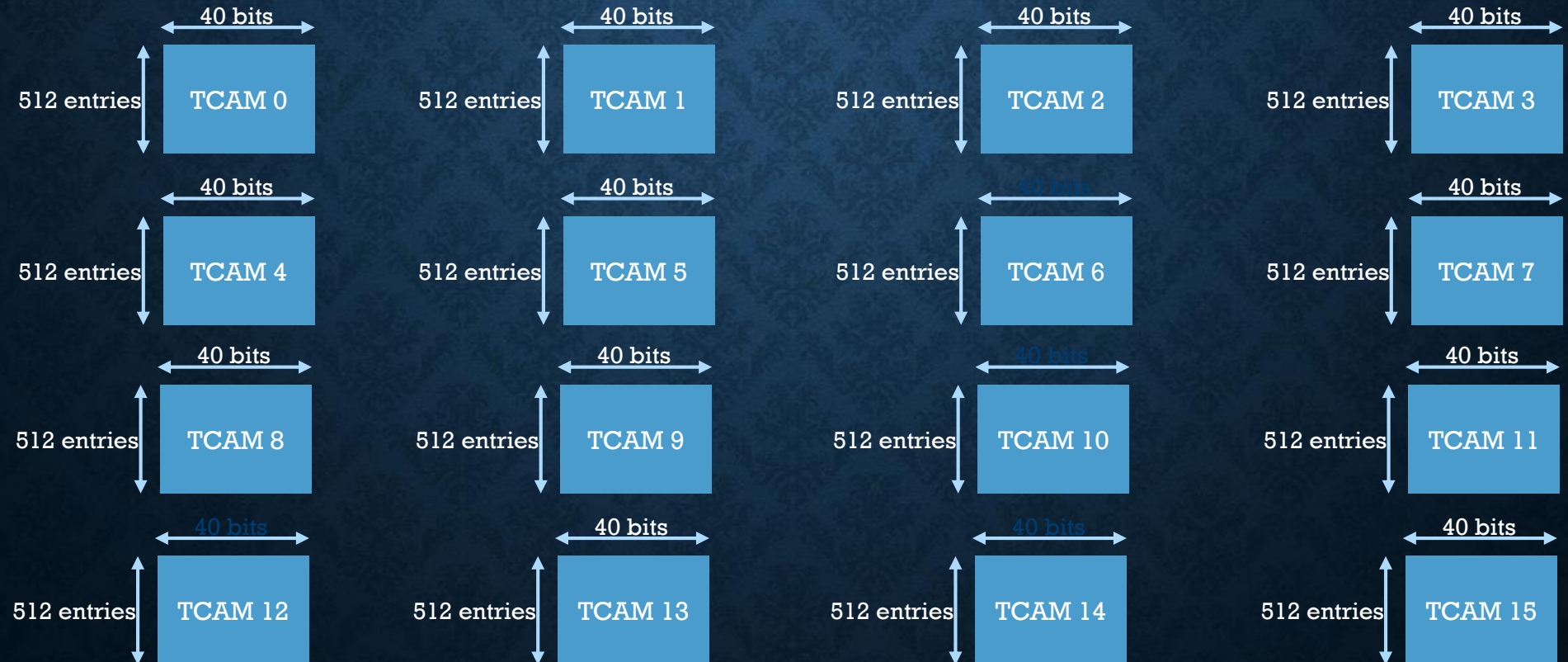


GO BACK TO PREVIOUS QUESTION

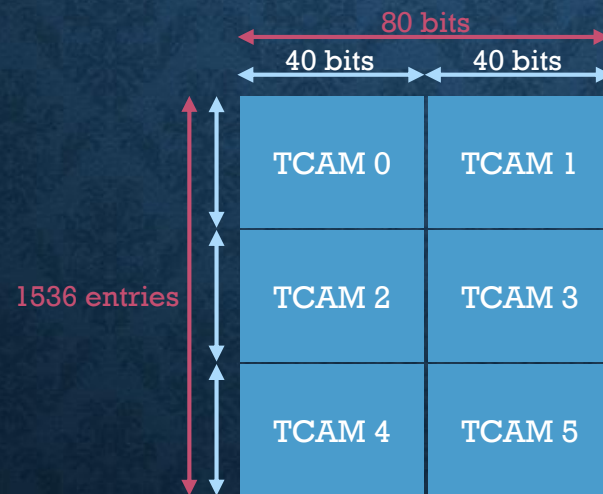
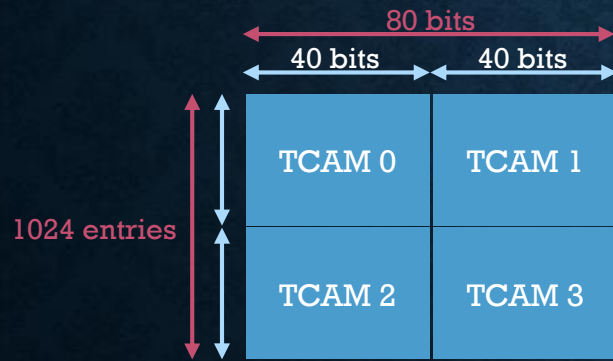
- How to create 2 rules for same profile but different input set? E.g.:
- flow create 0 ingress eth / ipv4 **src is 1.1.1.1** / end actions queue index 0 end / end
- flow create 0 ingress eth / ipv4 **dst is 2.2.2.2** /end actions queue index 0 end / end
- TCAM solution
 - Rule 1: Key <direction , profile, <src is 1.1.1.1, dst is any>
 - Rule 2: Key <direction , profile, <src is any, dst is 2.2.2.2>

MORE ABOUT ACL

- 16 TCAM slice
 - Each is 512 entries x 40 bits wide key



TCAM STACKING AND CASCADING

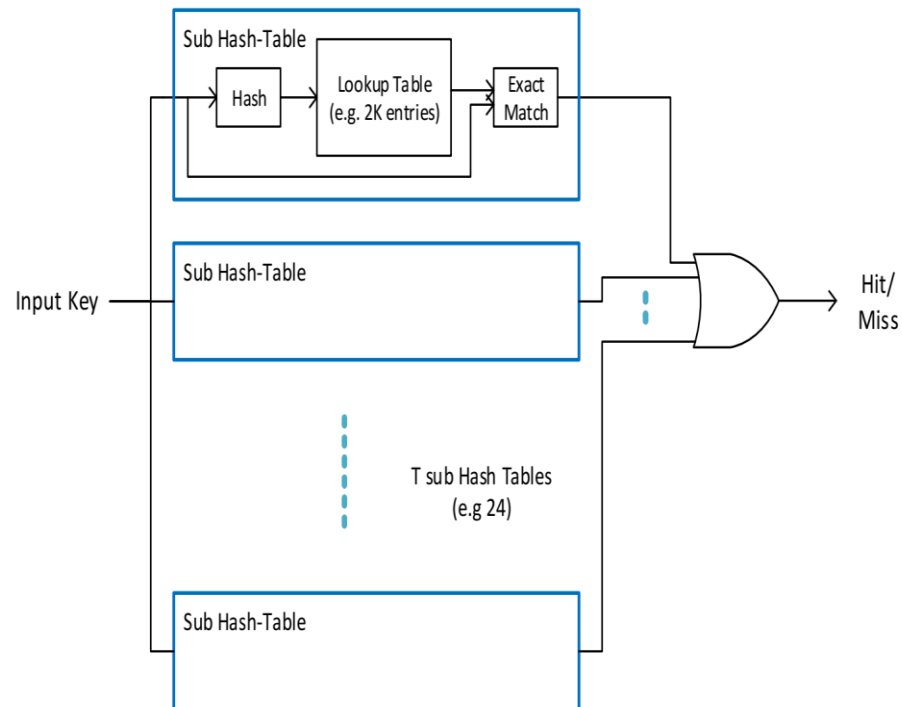


RAM/BCAM/TCAM USAGE

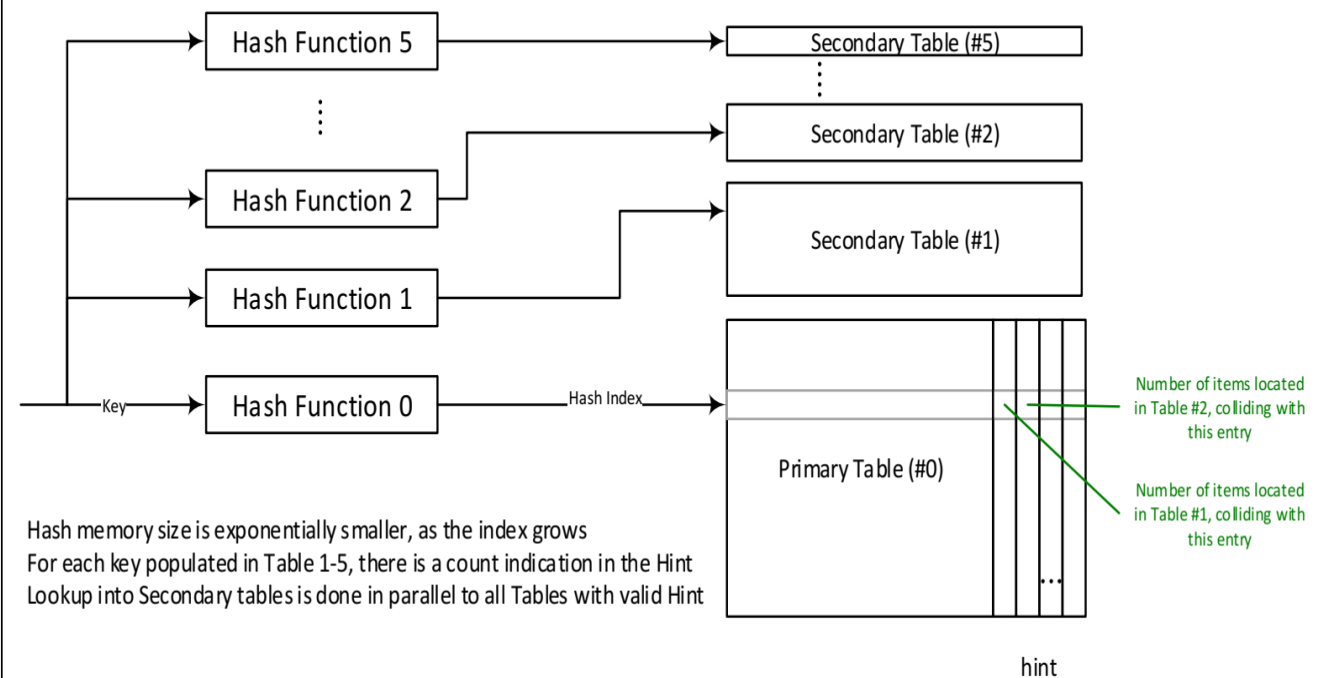
- CAM is expensive and consume a lot of power, not fit for large LUT.
- For all exact match LUT , CVL take RAM + Hash as Pseudo Binary CAM, it does not guarantee consistent insert/add/lookup time when collision happened.

	Parser	Switch	ACL	FDIR / RSS
RAM	Instruction Memory	Recipe Table Field Vector Table Action Table PTYPE Group Table VSI Group Table	Field Vector Table Action Table PTYPE Group Table VSI Group Table	Field Vector Table Action Table PTYPE Group Table VSI Group Table
TCAM	Boost TCAM	Profile LUT	Profile LUT Rule LUT	Profile LUT
Pseudo BCAM	Parse Graph	Rule LUT		Rule LUT

PSEUDO BINARY CAM



Multi Hash Scheme A



Multi Hash Scheme B

WHY IT IS FLEXIBLE ?

- Every thing is configurable

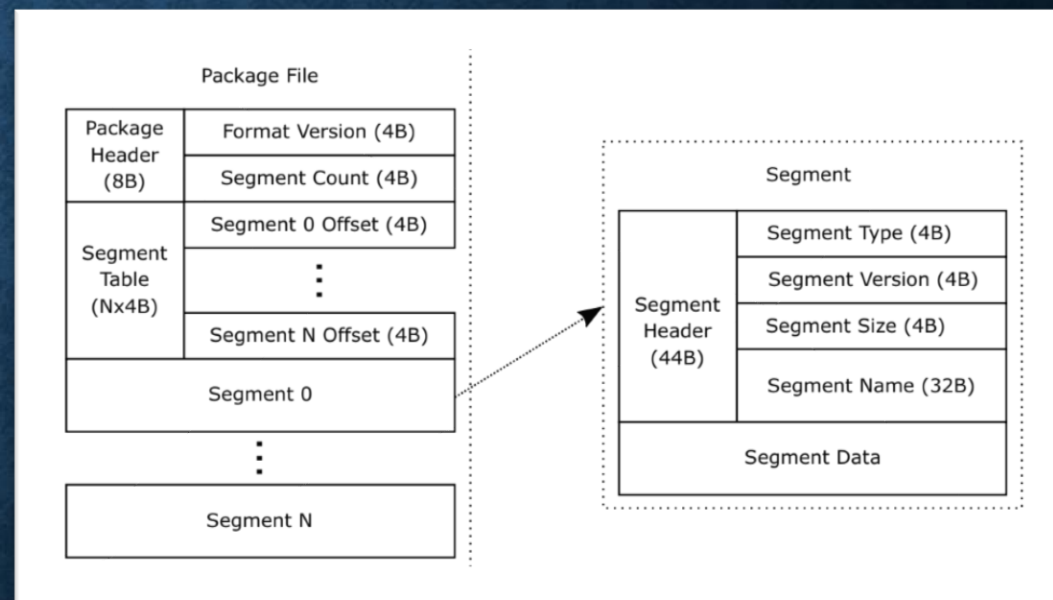
	Parser	Switch	ACL	FDIR / RSS
RAM	Instruction Memory	Recipe Table Field Vector Table Action Table PTYPE Group Table VSI Group Table	Field Vector Table Action Table PTYPE Group Table VSI Group Table	Field Vector Table Action Table PTYPE Group Table VSI Group Table
TCAM	Boost TCAM	Profile LUT	Profile LUT Rule LUT	Profile LUT
Pseudo BCAM	Parse Graph	Rule LUT		Rule LUT

Currently

1. Configured by Driver
2. Configured by DDP Package
3. Partial DDP Partial Driver

WHAT IS DDP

- Package File
 - A binary file that contains the settings for the configurable components
 - Package is downloaded to the FW by the driver
 - Download Package AQ command
 - FW then programs the correct HW registers/tables
- Driver can query and change the configuration at run time
 - Query: Upload Section AQ command
 - Change: Update Package AQ command
- <https://wiki.ith.intel.com/display/NACSoftware/OS+Default+Package>



THANKS