DPDK
SUMMIT
NORTH AMERICA

RTE_FLOW improvements

ORI KAM
STUFF ENGINEER,
NVIDIA

JULY 12-13, 2021

Who am I

- Stuff engineer at Nvida (Mellanox).
- Over 5 years working in DPDK community.
- RTE flow maintainer.
- Mail: orika@nvidia.com

.

Agenda

DPDK SUMMIT NORTH AMERICA

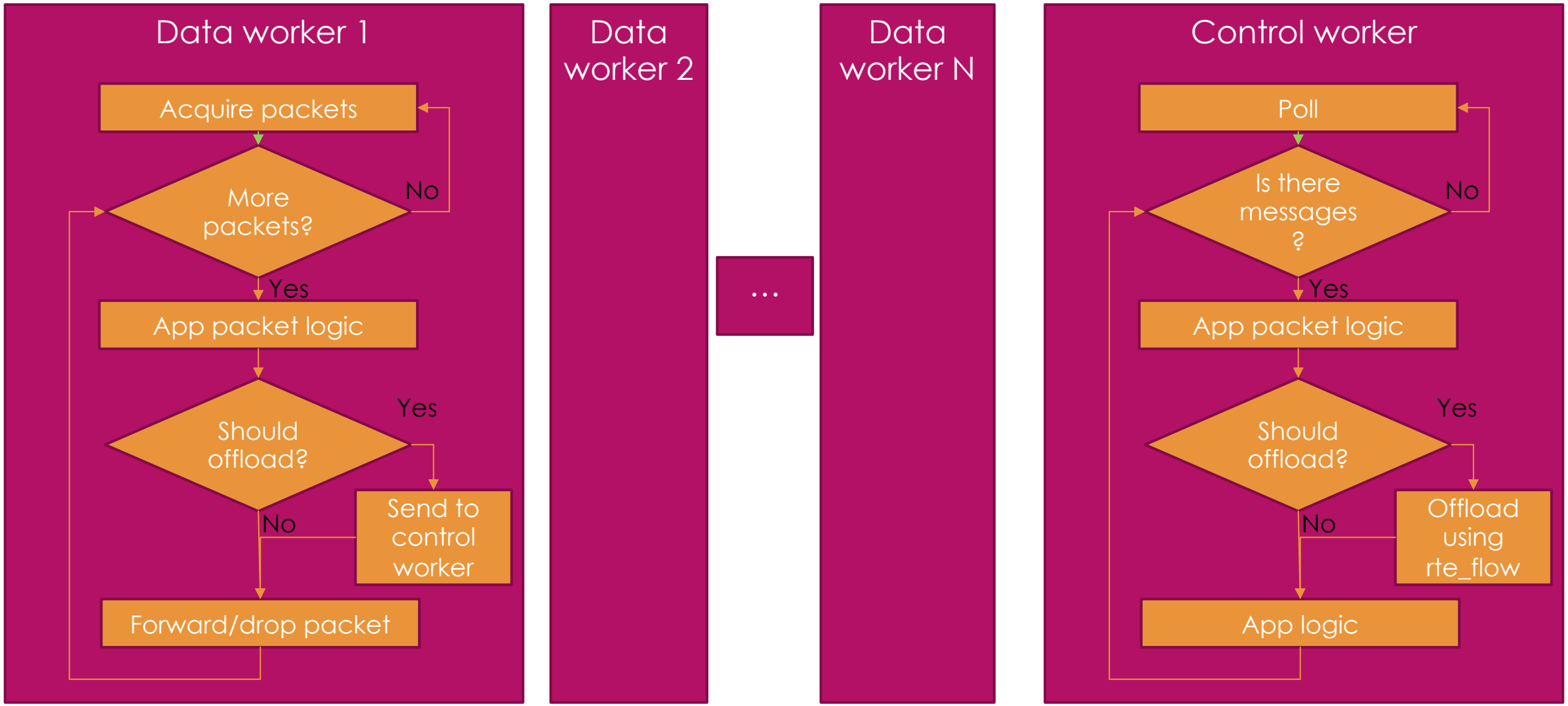
- RTE flow.
- 2 types of applications.
- Current RTE flow limitations.
- The solution.
- Basic components.
- Recap.

.

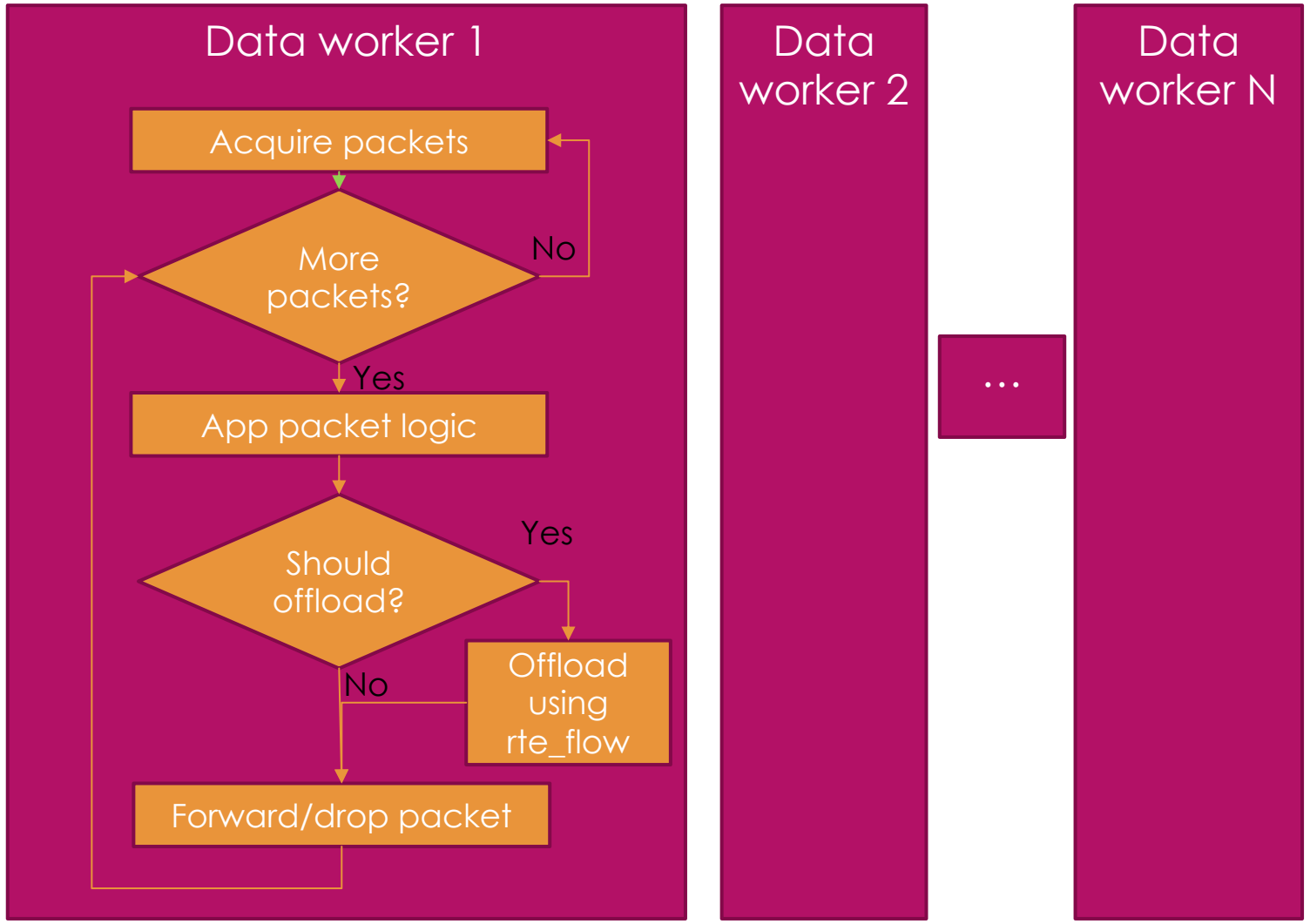
RTE_FLOW

- Was designed around 4 years ago.
- Support around 57 items.
- Support around 70 actions.
- Used by all application that require flow routing.
- Supported by all vendors*
- Sequential.
- Each flow is stand alone.
- Was designed with OVS in mind.

Types of applications - flows as part of the control (OVS)



Types of applications - flows as part of the data path (VNF/security)



Motivation / Target

DPDK SUMMIT NORTH AMERICA

- Support data path insertion application.
- Allow fast insertion for high scale rules
- Support multi-thread
- Support millions flows per second.

Assumptions

DPDK SUMMIT NORTH AMERICA

- Application has knowledge about the expected flows (number of flows/ what each flow includes).
- Application works in multi-thread mode.
- Most flows have common structure.

RTE_FLOW limitations

- Synced.
- Require locks*
- Each flow is stand alone, and all flows are treated the same.
- Is not designed for data path applications.
- Limited insertion rate.

RTE_FLOW advantages

- Widely used.
- Large number of actions and items are already implemented.
- Easy for applications that are using small number of flows.

Solution – Main key points

- Add new API that will enable the following:
  - Queue based (multi thread)
  - async operations.
  - Lockless
  - Optimized for insertion rate.
  - Will be able to connect to existing API.
- Will be split to two types
  - Configuration – cross port, will require locks works under assumption that this is created once for large number of flows. (all possible allocation should be done in this part)
  - Data path – works on queues, optimized as best as possible. (uses the configuration in order to understand how best to optimize)

Async

- Queue based.
- Enqueue creation of flows, destruction of flows, modify and query of actions.
- The result will be exposed to the application using polling.

Lockless

- Queue based.
- Application will configure the requested number of queues for the rte_flow.
- The queues are not thread safe, which mean each queue can only be accessed from one thread, or it is the responsibility of the application to use locks.

Optimized for insertion rate

- Both earlier slides.
- Getting system spec before flows creation.
  - Memory size,
  - Number of counters,
  - Number of different encaps
  - ..
- Using predefined sizes – allows fast allocations, saves locks.
- Move all hard work to initialize stage.
- Get hints from the application on use cases to allow better and faster configuration.
- No PMD assmptions

Key structures and functions

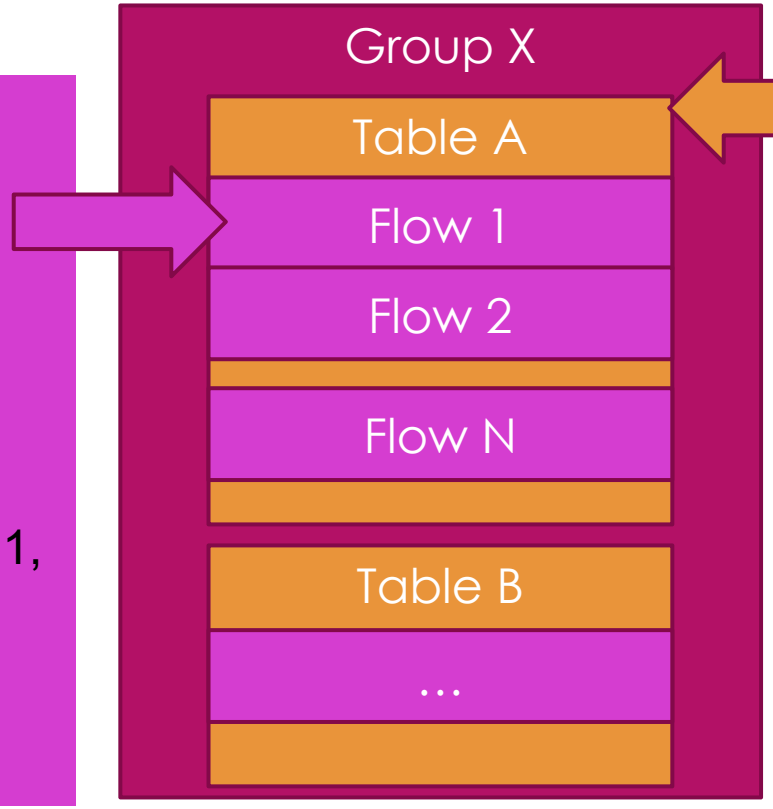
DPDK SUMMIT NORTH AMERICA

- Tables
- Action template.
- configuration
- Queues.

Key structures - Tables

- Groups flows with the same group ID, priority and matching mask.
- Considered as part of the configuration.

- Struct rte_flow_item pattern {
  - Type eth
    - Spec = {s_mac = 0x12:12:12:12:12:12}
    - Mask = NULL
  - Type ipv4
    - Spec = {src = 192.1.1.1, dst = 198.2.2.2}
    - Mask = NULL
- }



- Struct rte_flow_item table {
  - Type eth
    - Spec = NULL
    - Mask = {s_mac = 0xffffffffffff}
  - Type ipv4
    - Spec = NULL
    - Mask = {src = 0xffffffff, dst = 0xffffffff}
- }

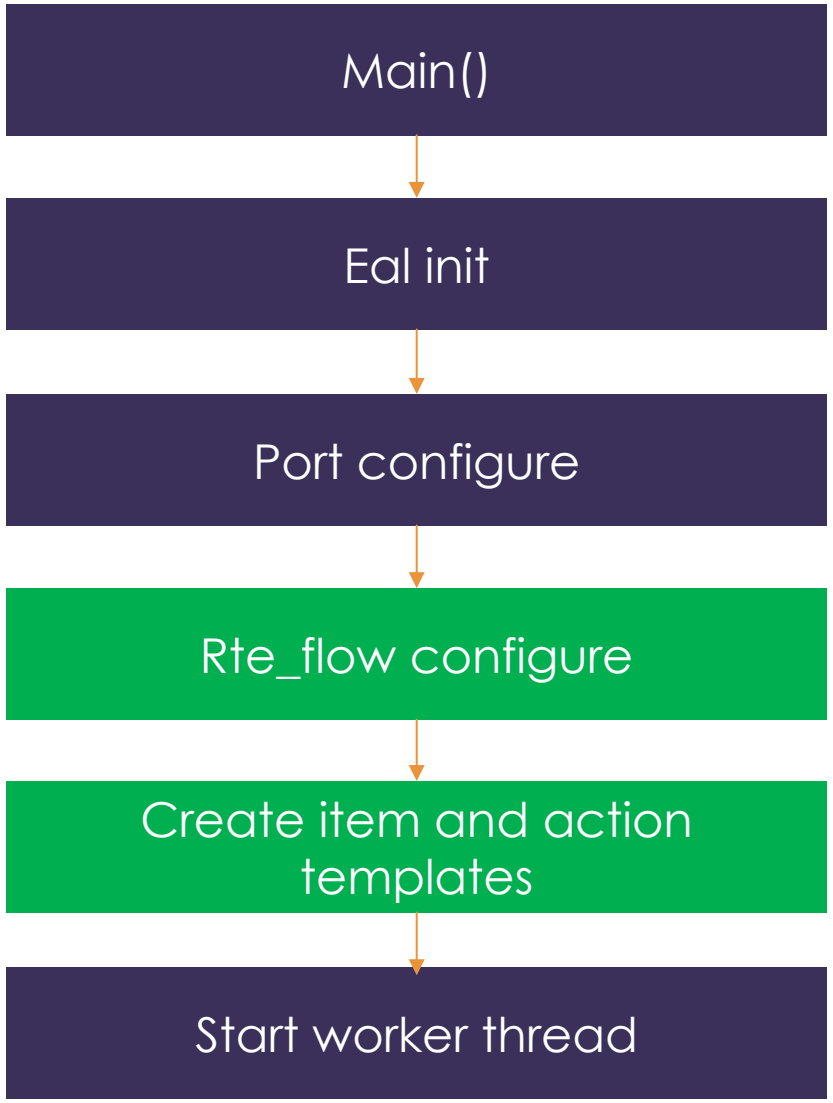
Key structures – action template

- Created as configuration item.
- Holds a list of item types that will be used in flow.
- Example
  - Counter + modify_src_mac + jump
- The values will be set per flow.

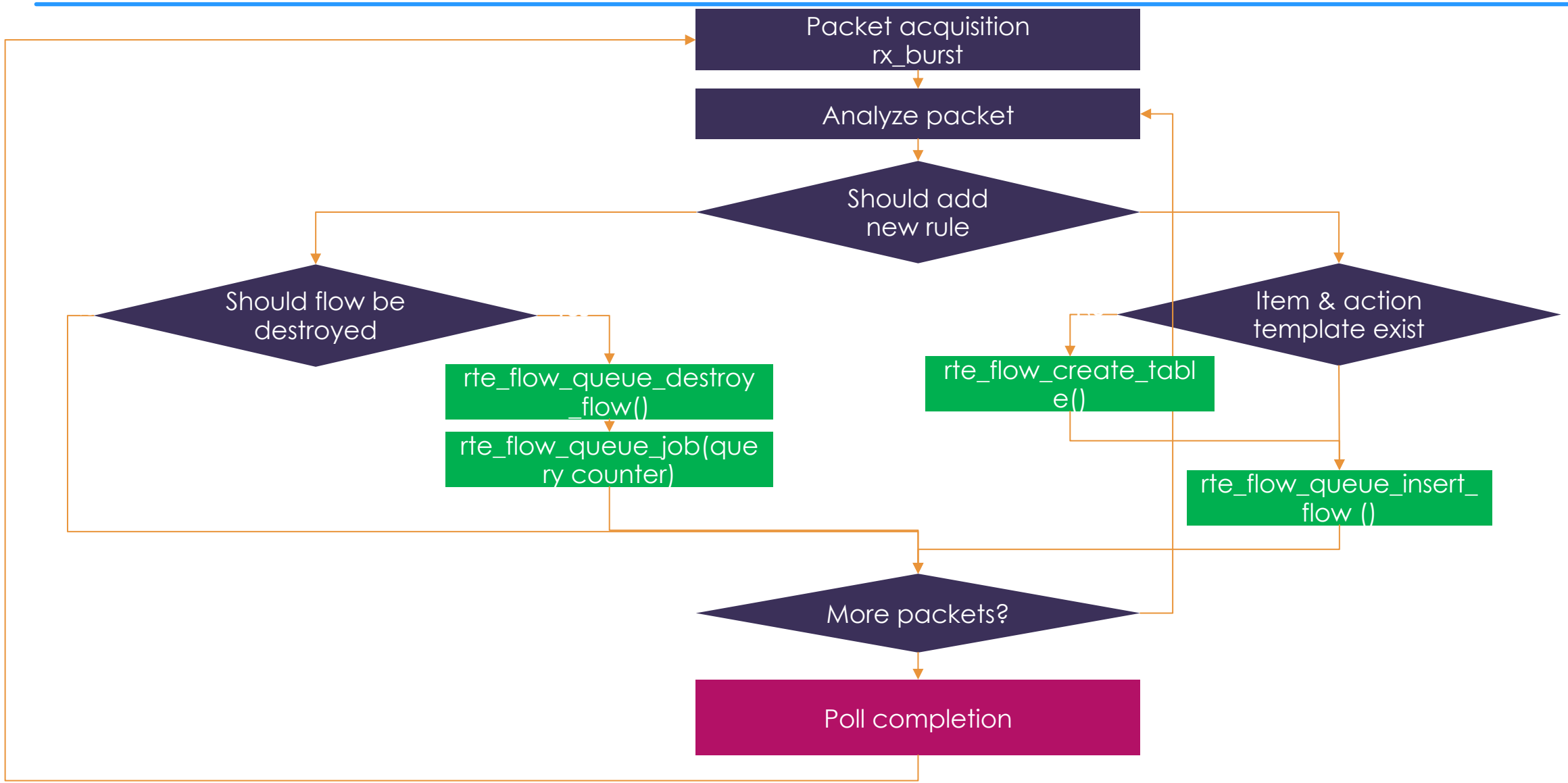
Key structures – rule creation

- Considered data path.
- Works on queues.
- Combine the table + action_template + the item spec + the actions

App flow


Main()
Eal init
Port configure
Rte_flow configure
Create item and action templates
Start worker thread

App flow


Packet acquisition
rx_burst
Analyze packet
Should add
new rule
Should flow be
destroyed
rte_flow_queue_destroy
_flow()
rte_flow_queue_job(que
ry counter)
Item & action
template exist
rte_flow_create_tabl
e()
rte_flow_queue_insert_
flow ()
More packets?
Poll completion

Q/A

DPDK SUMMIT NORTH AMERICA

- Thank you all.

DPDK

SUMMIT

NORTH AMERICA