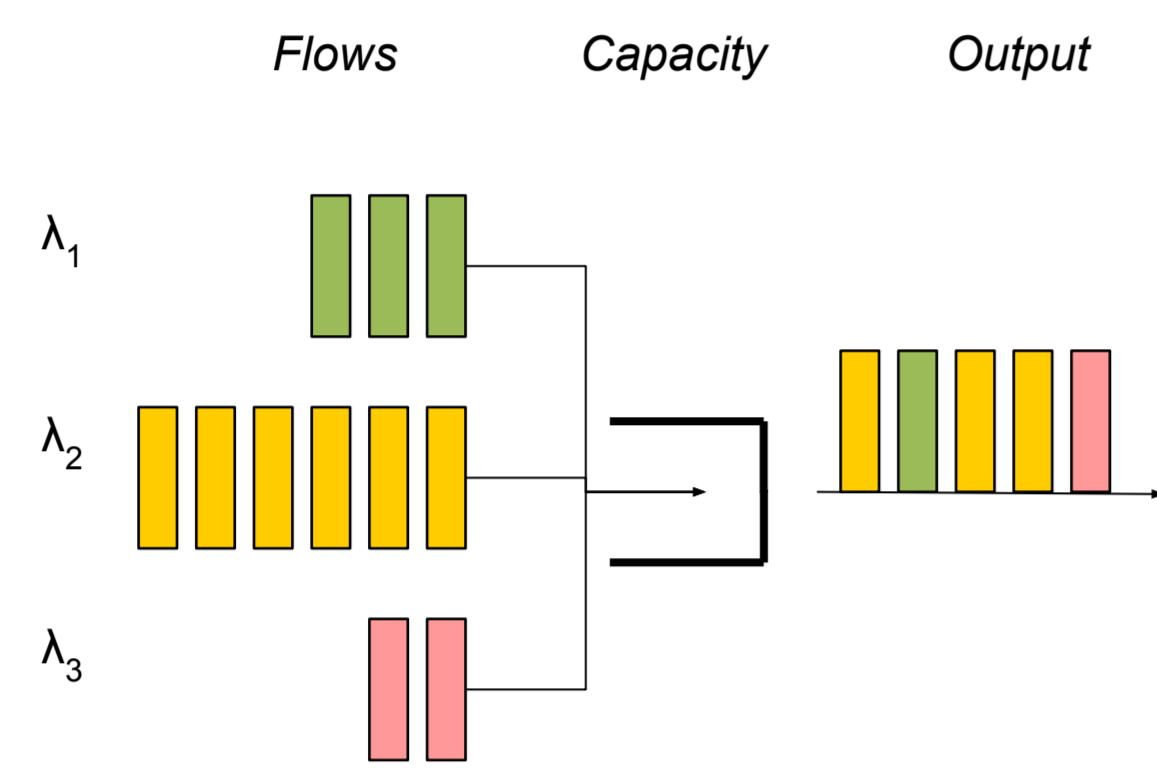# DEMO: Controlling software router resource sharing by fair packet dropping
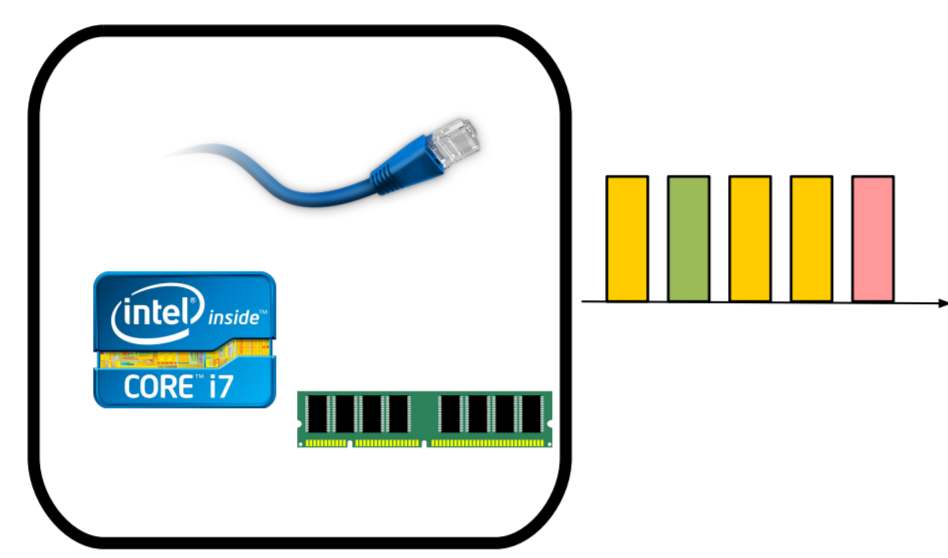
Vamsi Addanki, Leonardo Linguaglossa, Jim Roberts, Dario Rossi
Telecom ParisTech, Paris

Classical bandwidth sharing

Multi resource sharing

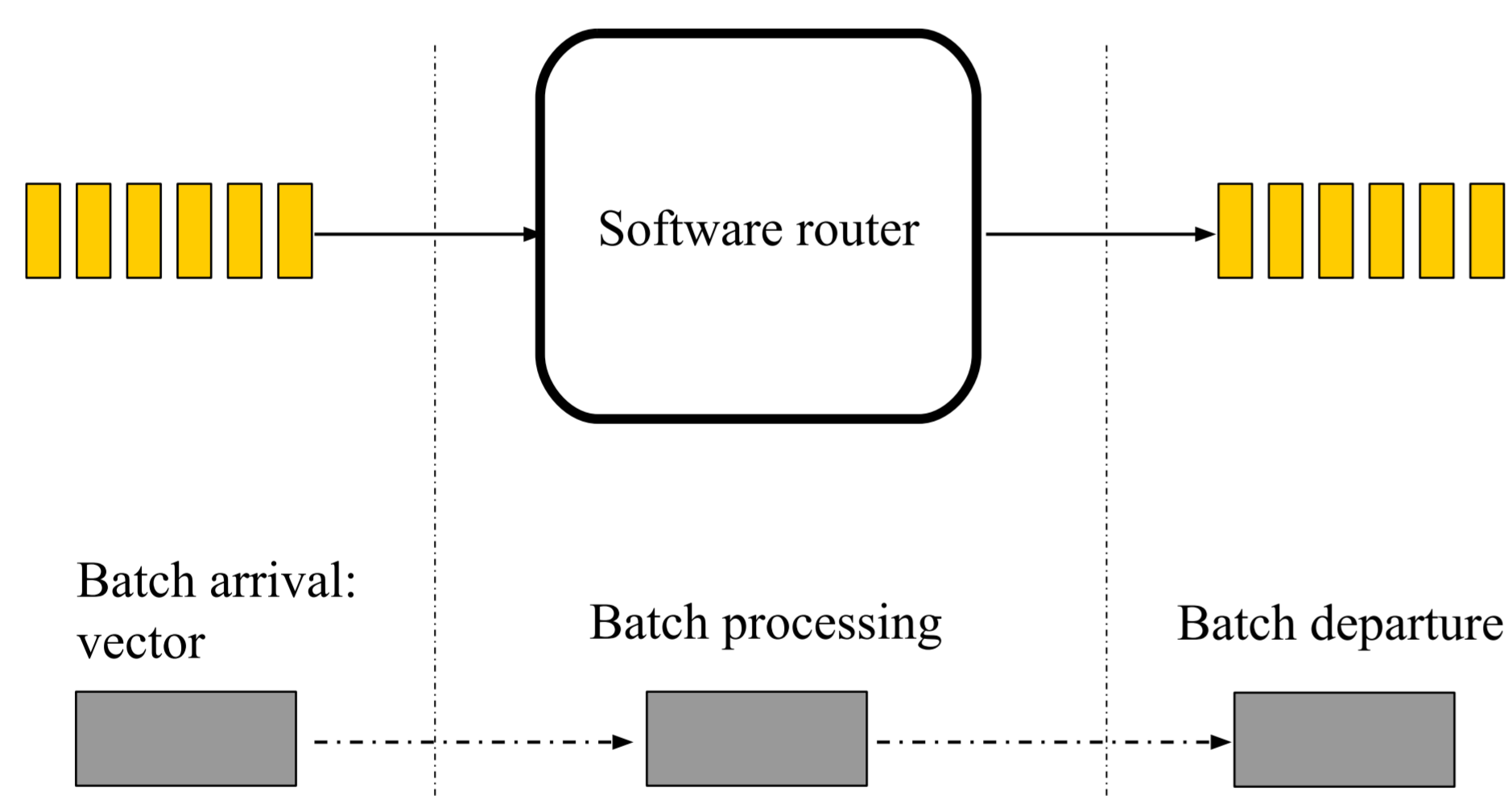- Implementation of existing mechanisms in a software router is not flexible for multi resource sharing.
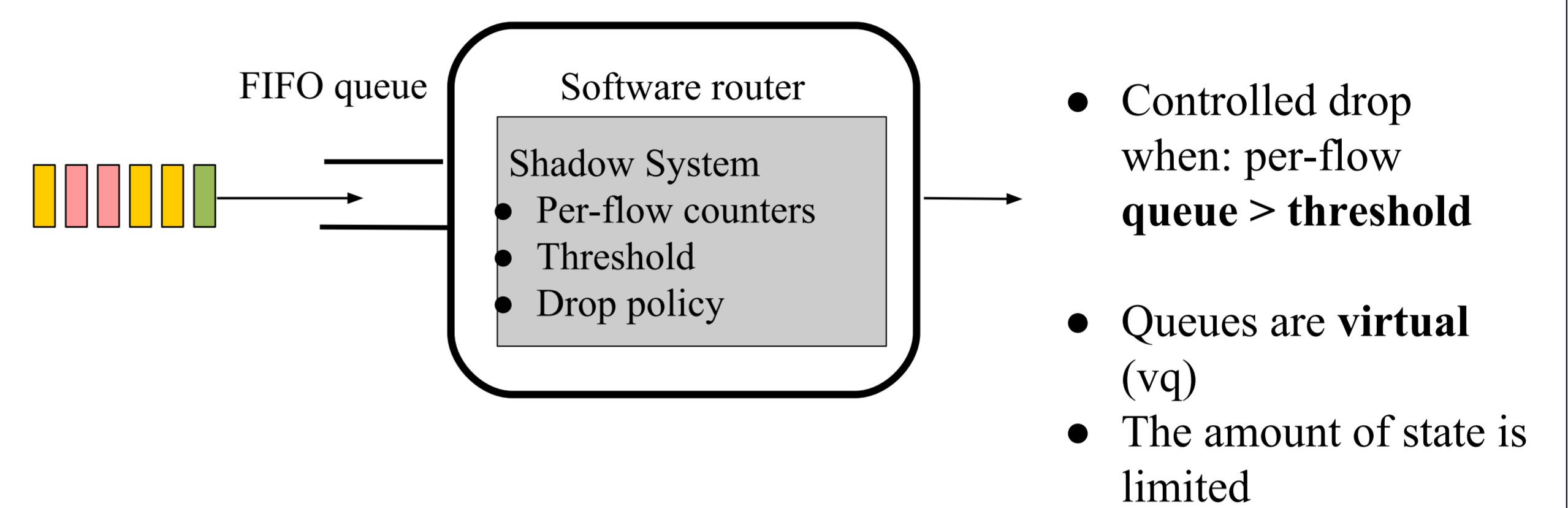
Existing Mechanisms

- Fair queuing : DRR, STFQ
  - Highly complex to implement in a software router
- Fair dropping: FRED, CHOke, AFD
  - Lack precision

- ❏ Common problem is implementation in software routers
- ❏ *Fair dropping is essential*, scheduling is not necessary

Software router architecture vs Scheduling (Motivation)

Software Router	Scheduling
Batch Arrivals, batch processing and batch departure	Requires individual packet arrival and departure

Fair Drop Algorithm

- Controlled drop when: per-flow **queue > threshold**
- Queues are **virtual** (vq)
- The amount of state is limited

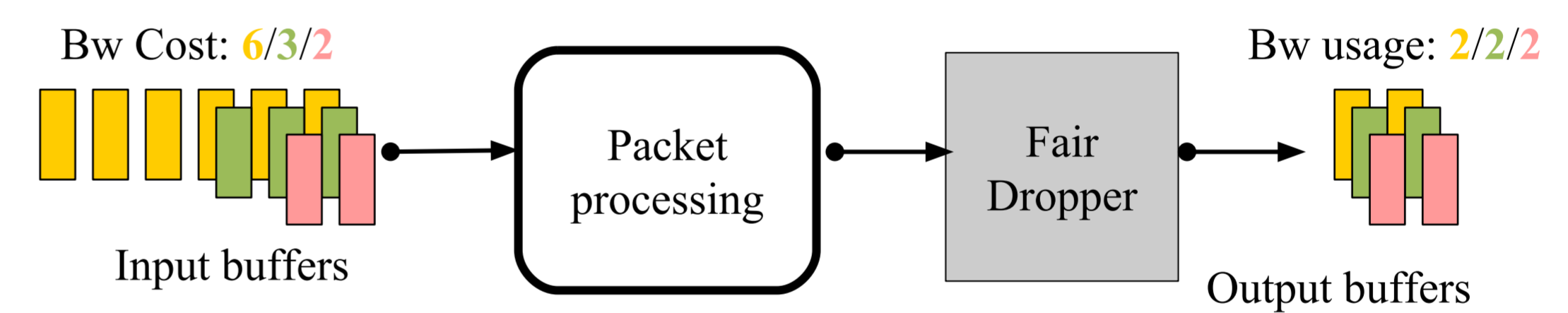**On batch (b_n) arrival at time T_n**
- Decrement backlogged flows by fair share of service capacity
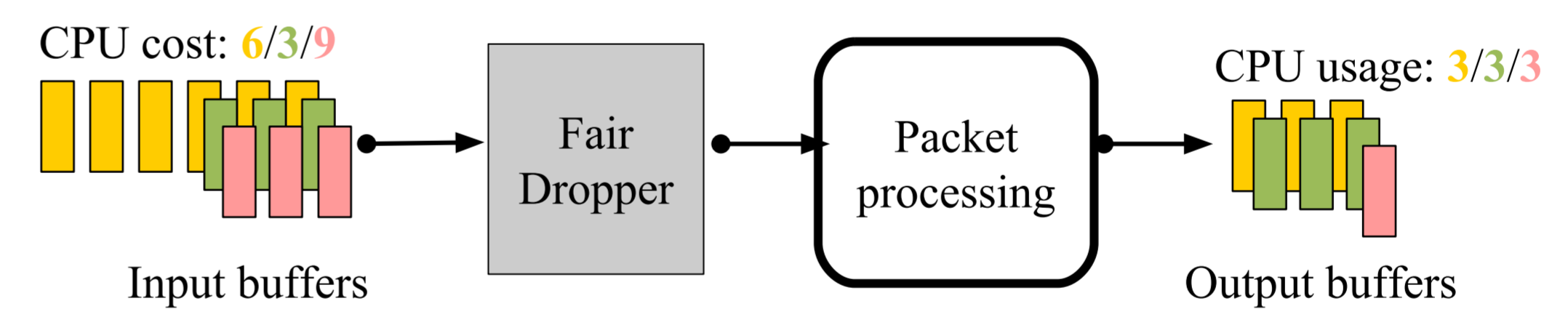 $(T_n - T_{n-1})*C$

On packet processing
- Store current flow id in the activelist (if not present)
- Drop packet if flow virtual queue (vq) > threshold
- If not dropped, increment vq with packet length (packet size in case of bandwidth sharing)

Reference Paper: V. Addanki, L. Linguaglossa, J. Roberts and D. Rossi. Controlling software router resource sharing by fair packet dropping. IFIP Networking 2018,Zurich.
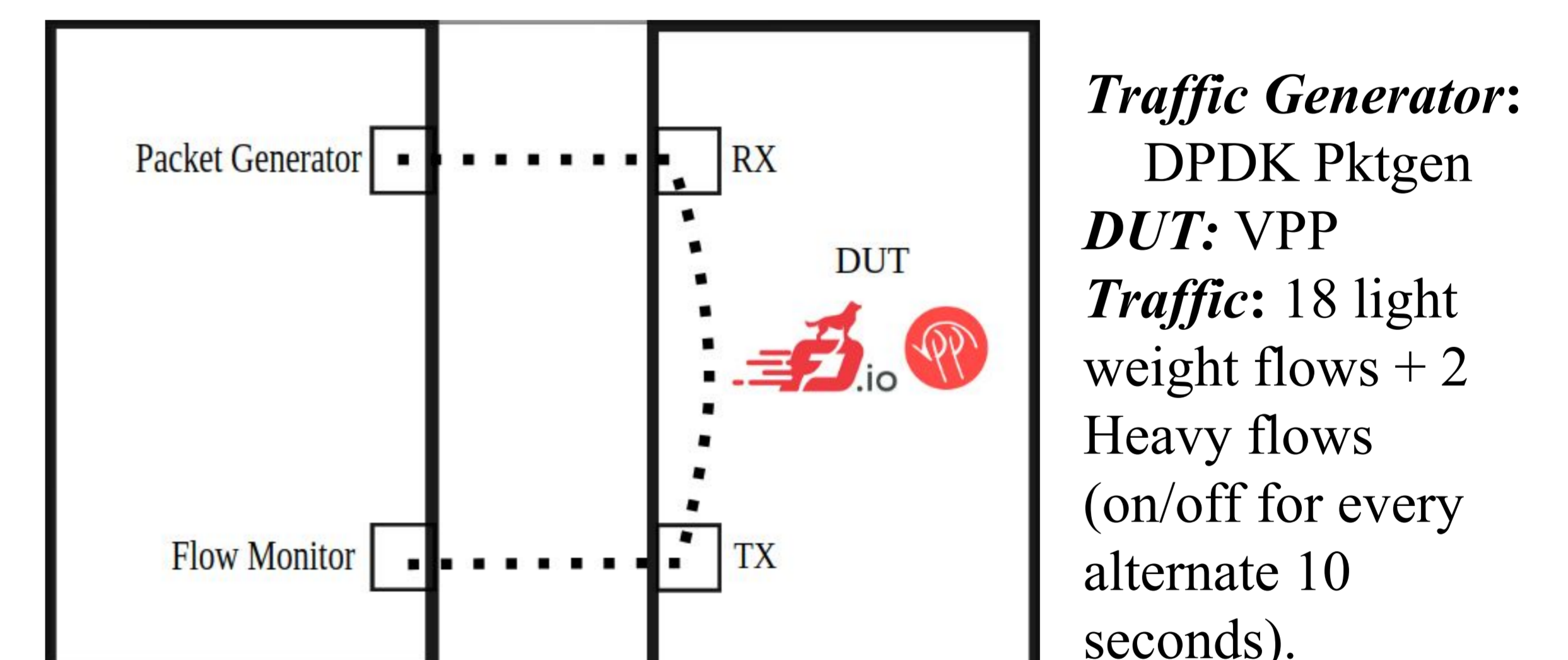
Dropper Placement

Bandwidth Sharing: Dropper before the output

CPU Sharing: Dropper before the input

Test Bed

Traffic Generator: DPDK Pktgen
DUT: VPP
Traffic: 18 light weight flows + 2 Heavy flows (on/off for every alternate 10 seconds).

Experimental results