Micro Load Balancing in Data Centers with DRILL

Soudeh Ghorbani
Brighten Godfrey
Yashar Ganjali
Amin Firoozshahian

(Presentation by Alexander Canals)

What is the problem?

- Datacenter topologies are designed to deliver high capacity

- But networks experience congestion and packet drops even with low utilisation (25%)

What is the problem?

- Existing load balancing solutions often require global congestion information and they balance at high levels: flowcells, flowlets or flows

- Control loops of those systems require multiple round trips (10s to 100s of microseconds)

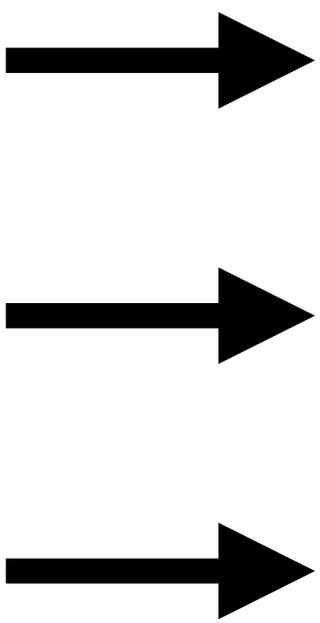
- But majority of congestion incidents are short-lived: Most microbursts that are responsible for 90% of packet-loss last for less than 3 microseconds

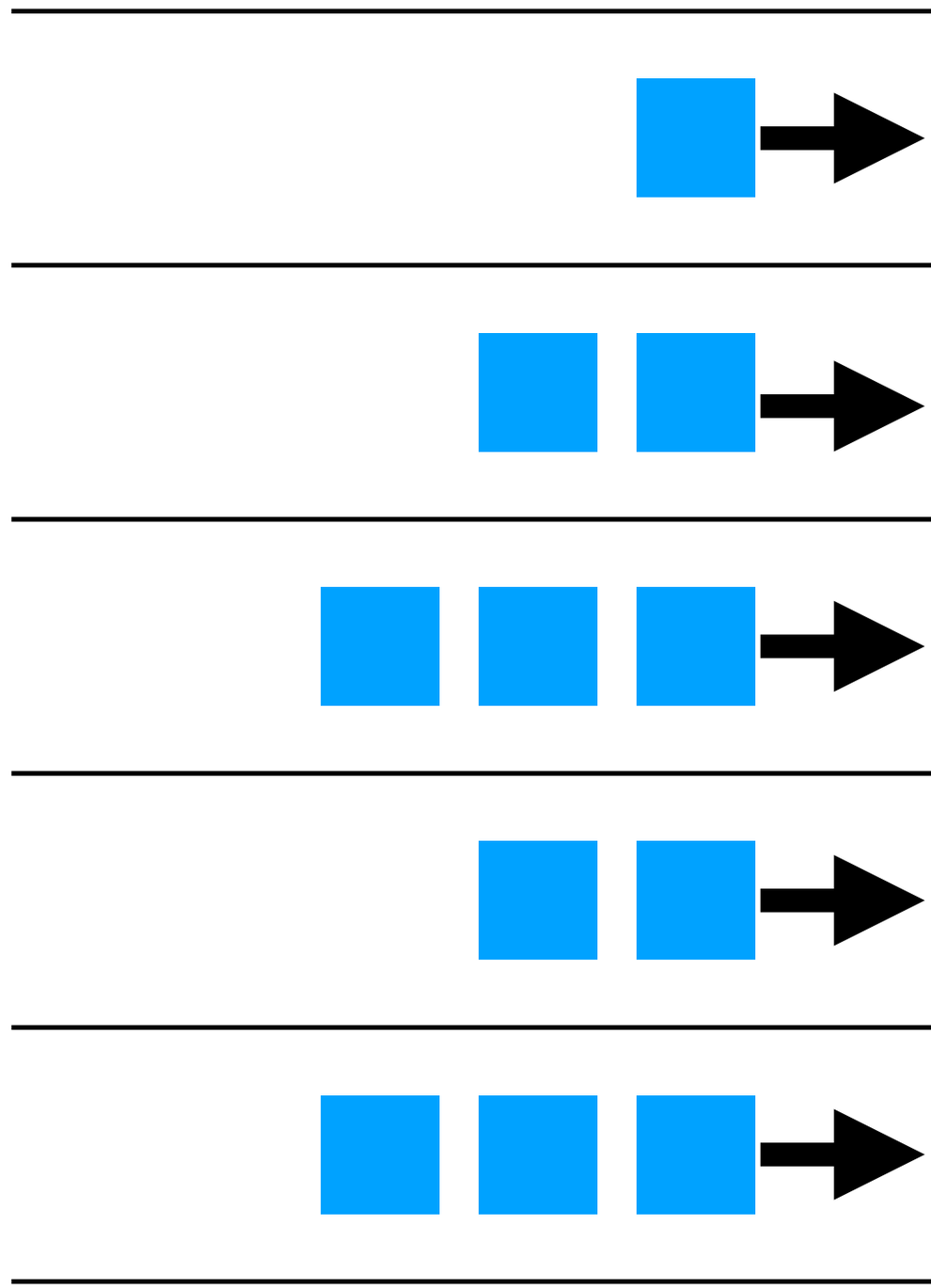
How does DRILL solve this problem?

- Load balancing decisions local to each switch
- Micro load-balancing: microsecond timescales; per-packet decisions
- For every incoming packet, the switch compares the queue lengths of the output ports and picks the one with the shortest queue
- For topological changes (asymmetry) more global path planning might be necessary

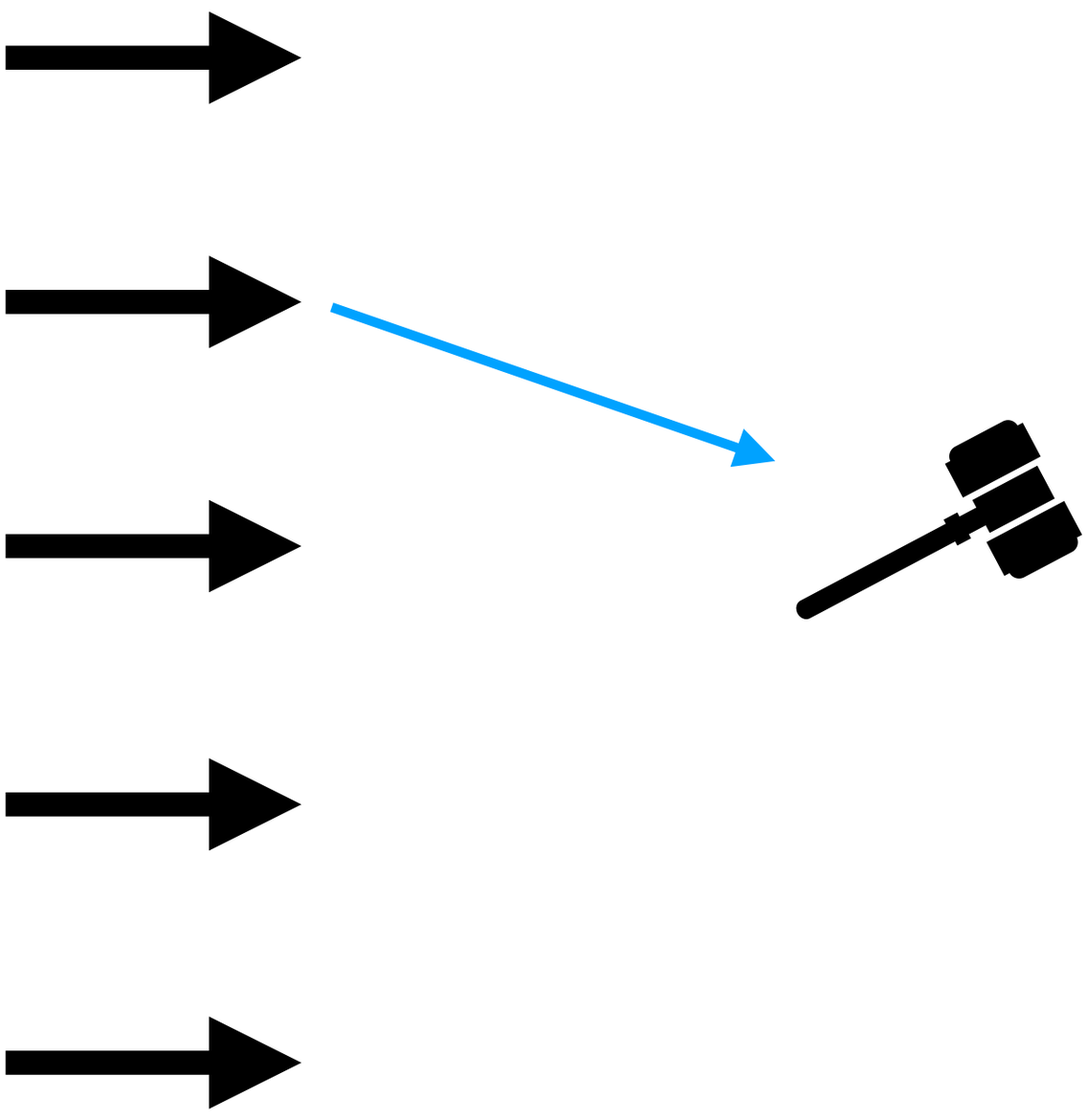
Input ports

Output ports with queues

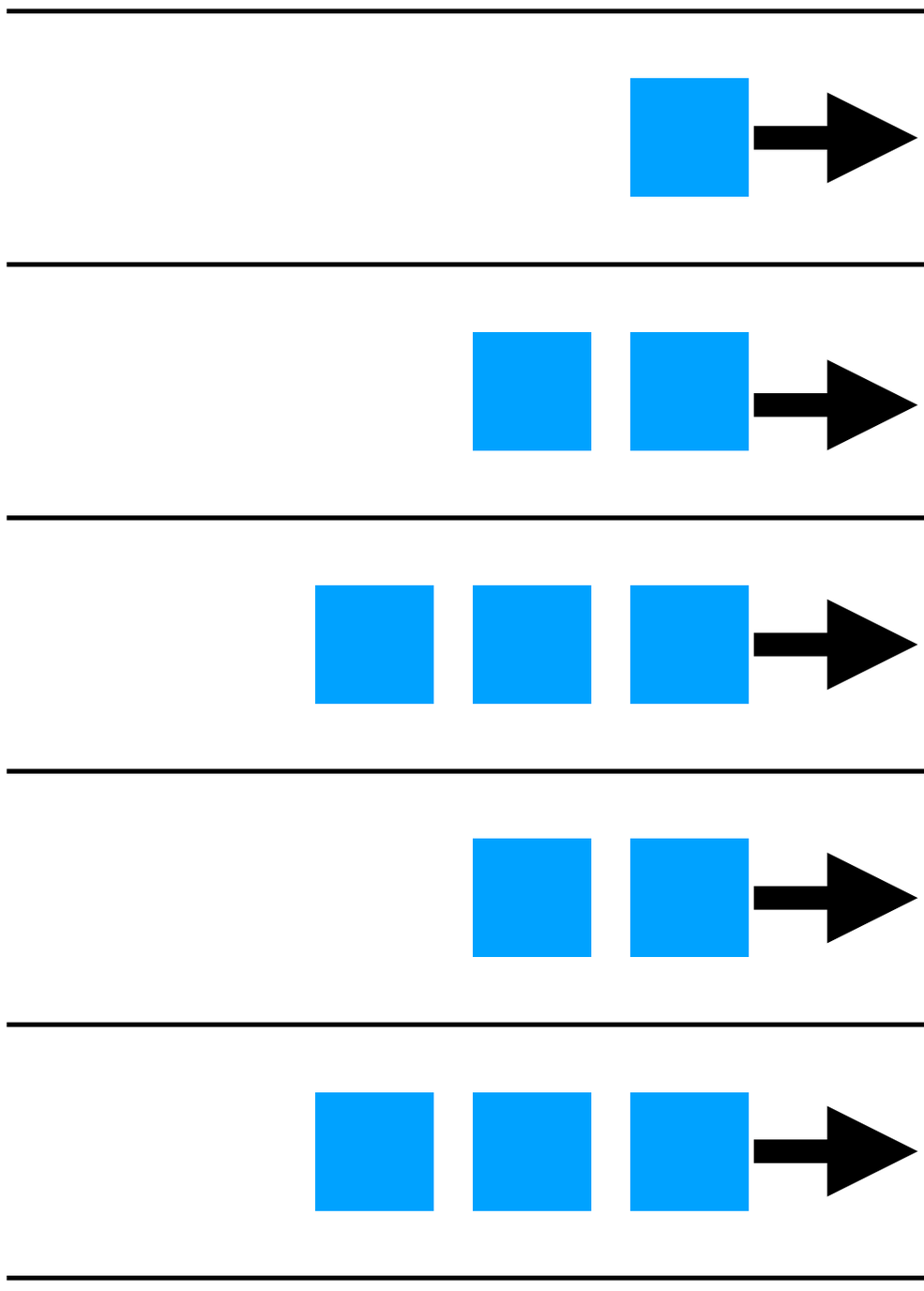




Input ports

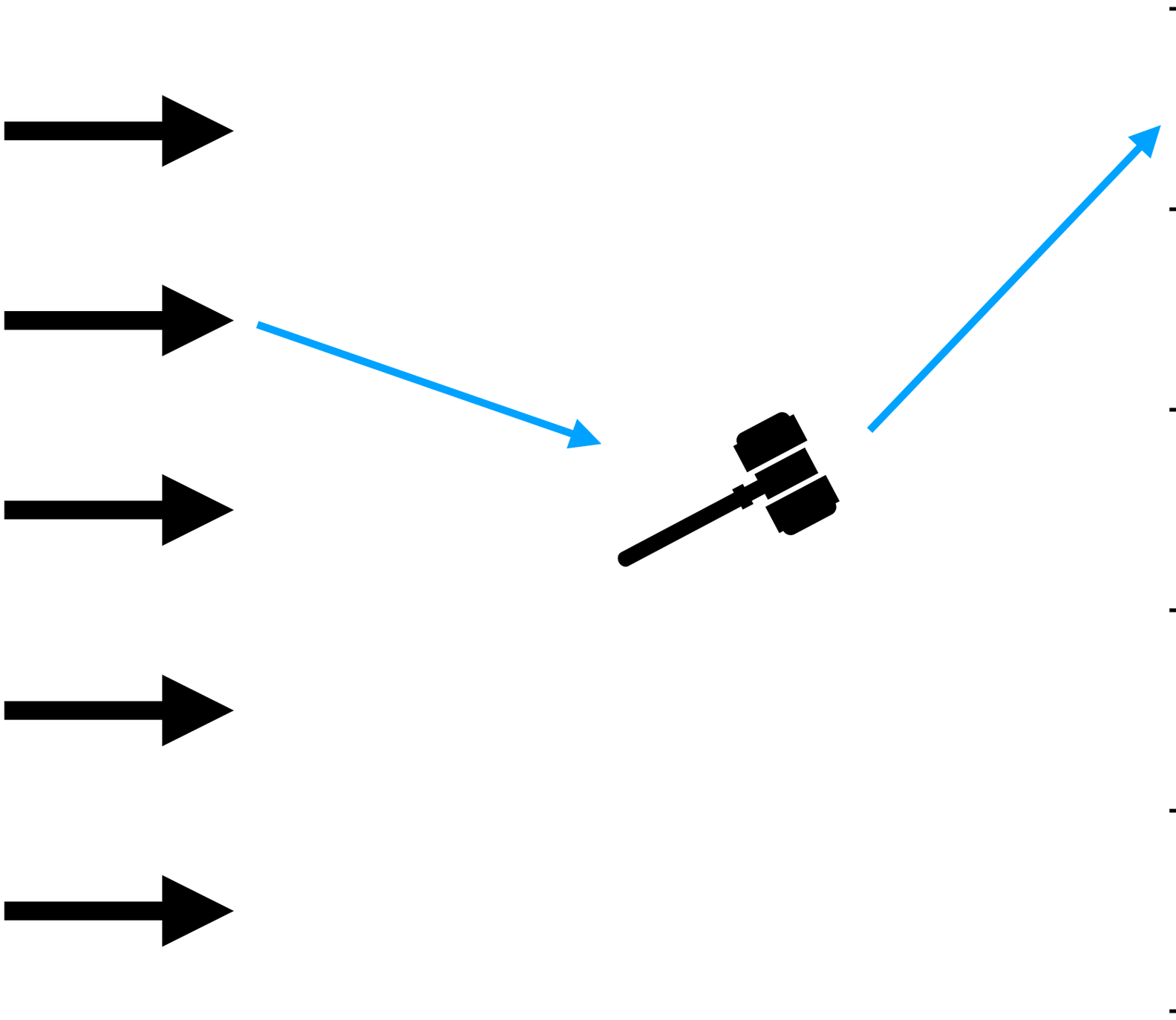


Output ports with queues

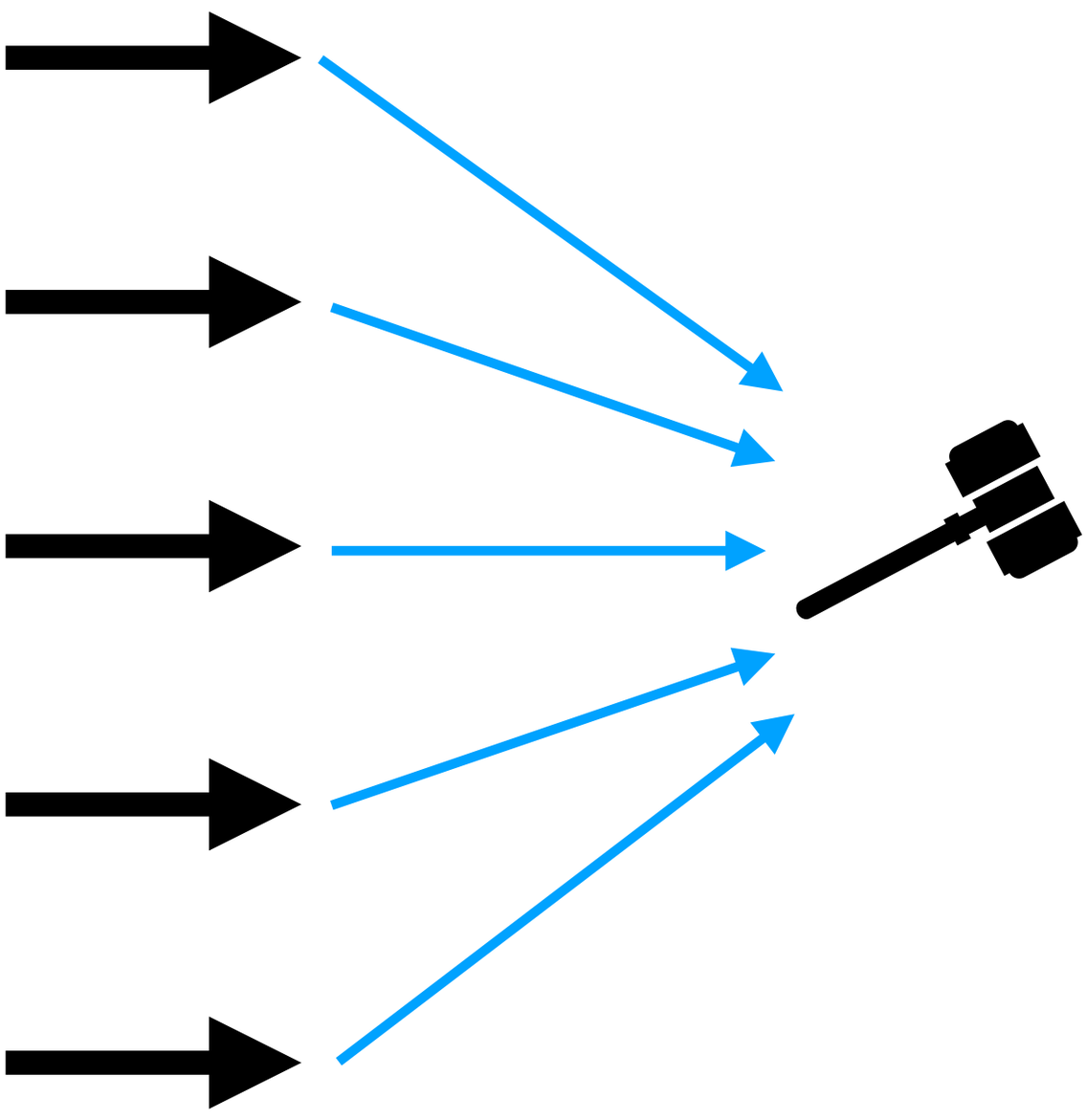


Input ports

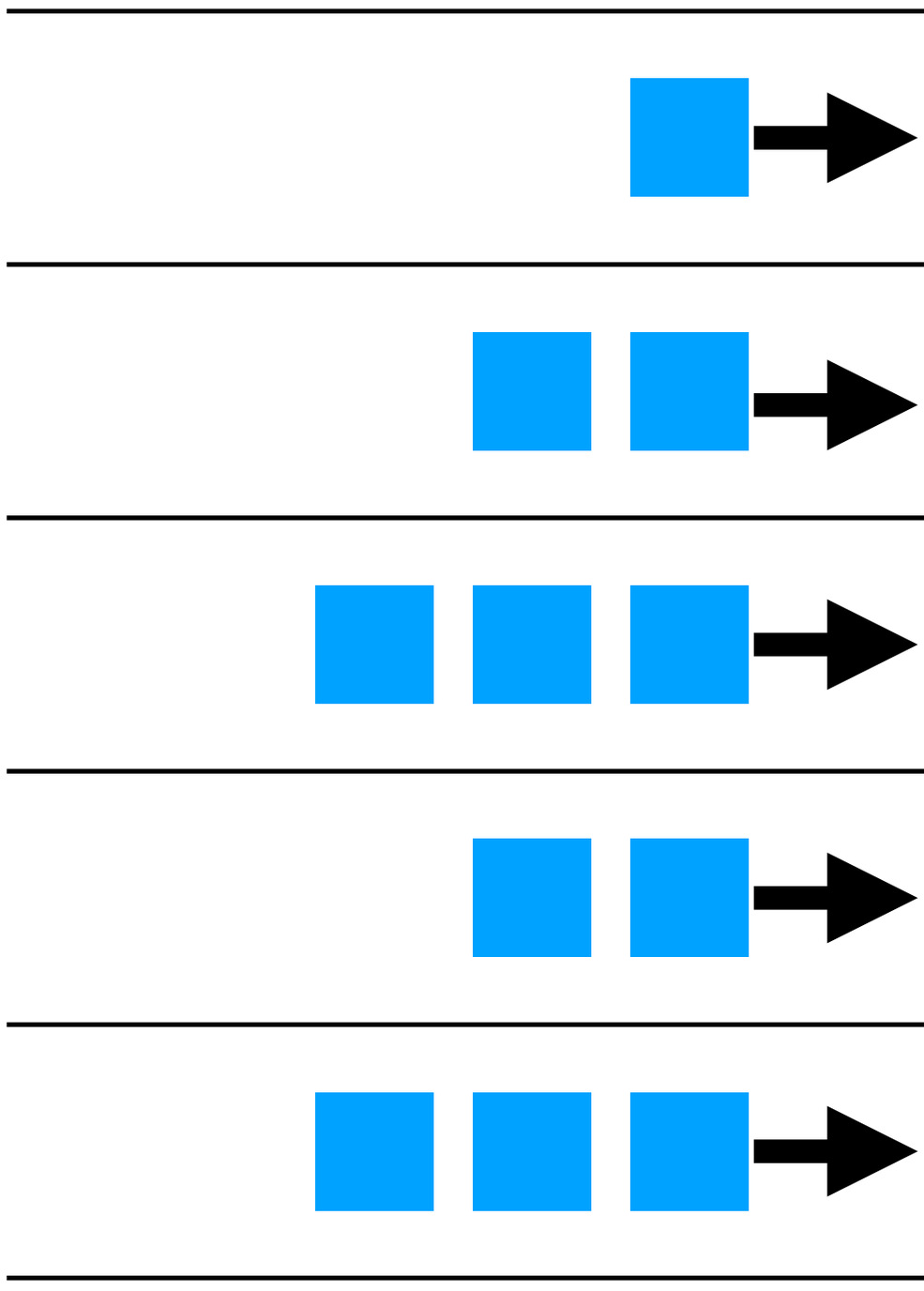
Output ports with queues



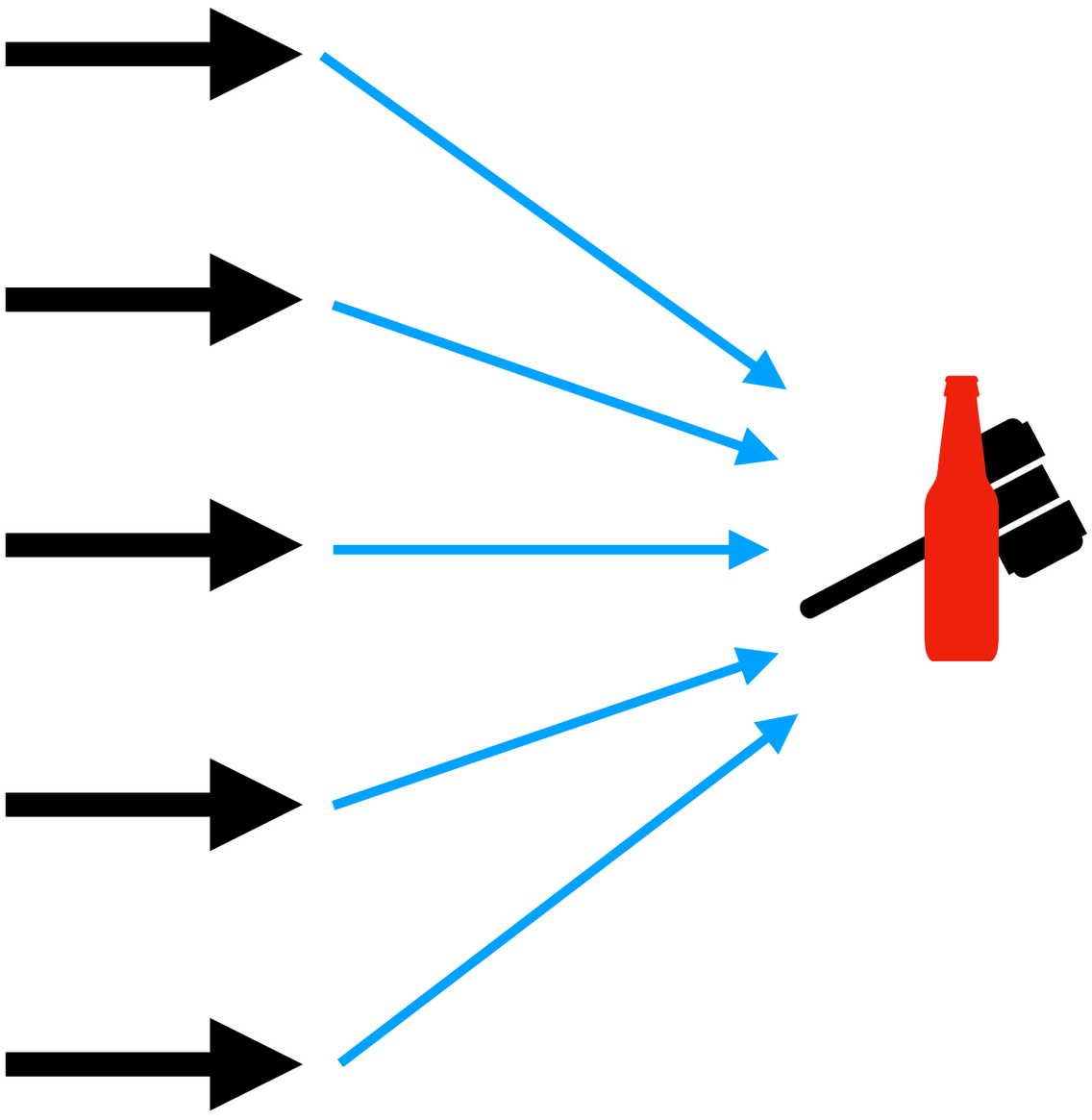
Input ports



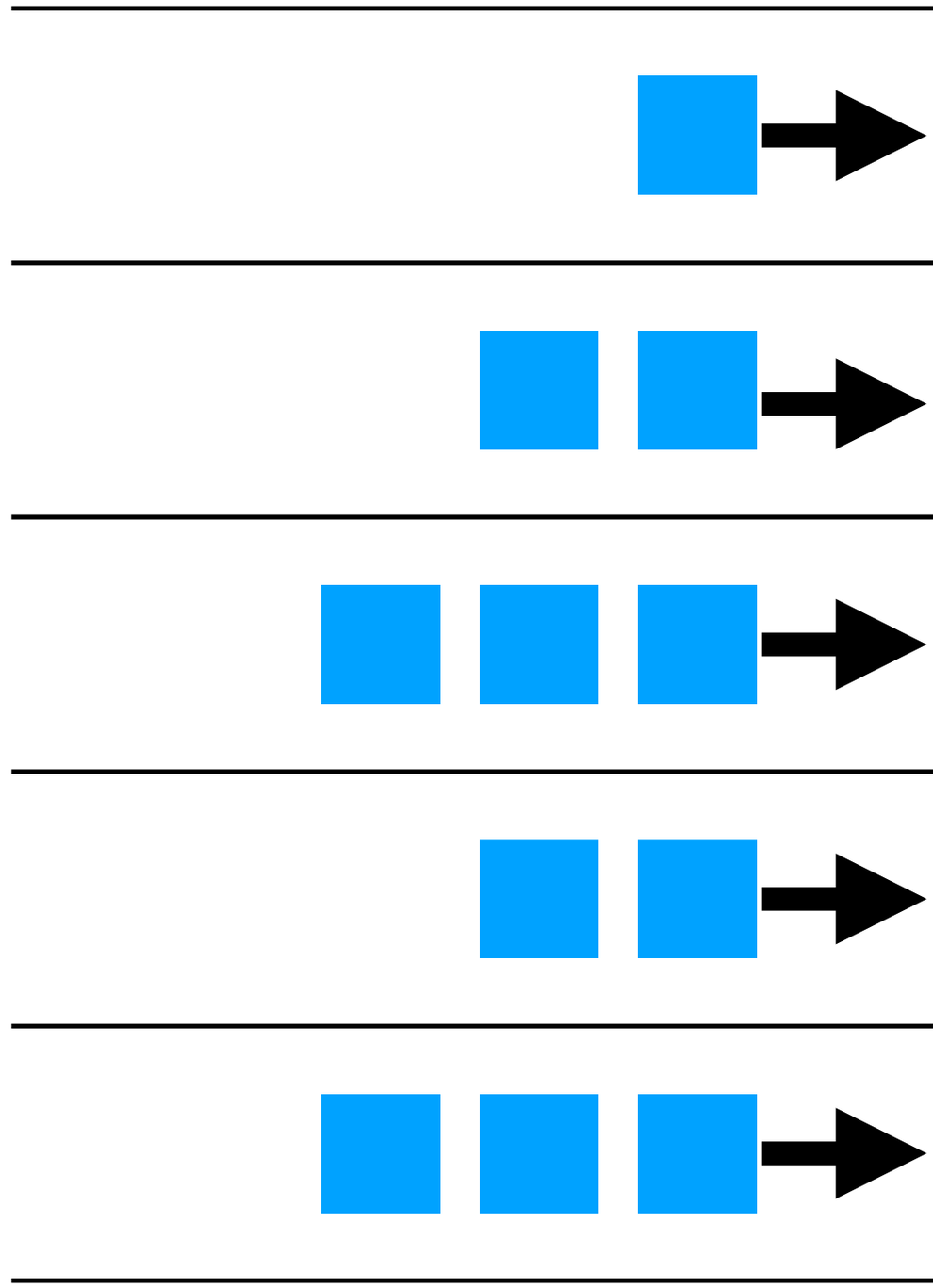
Output ports with queues



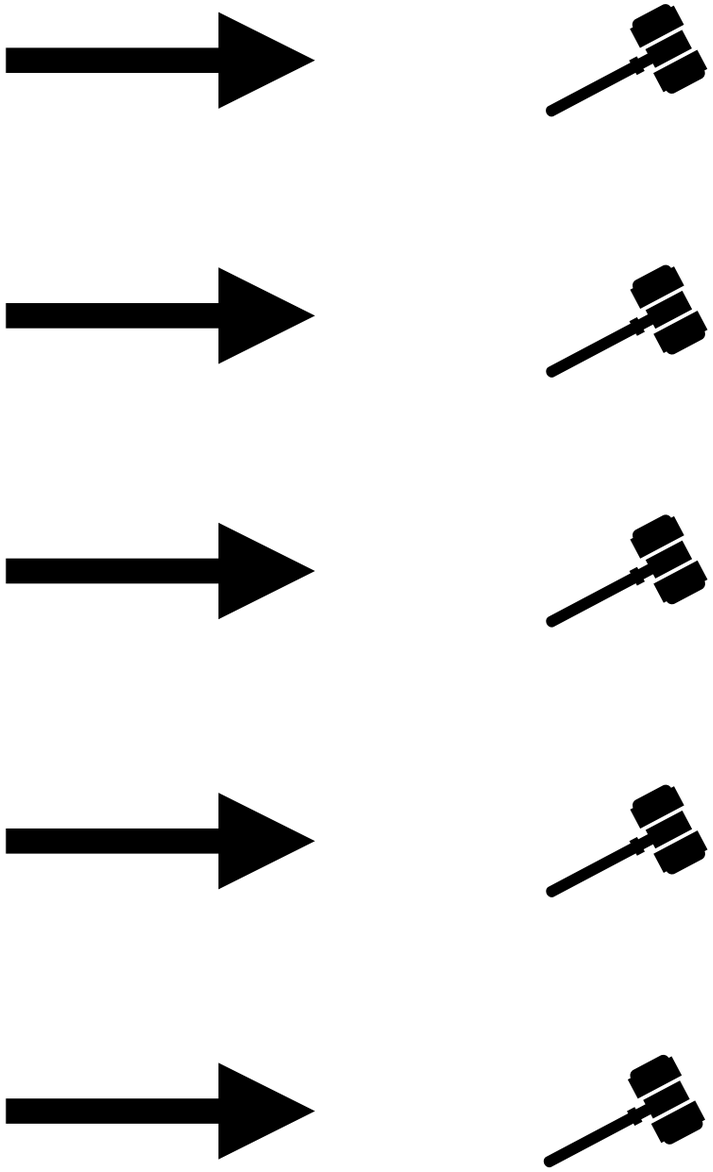
Input ports



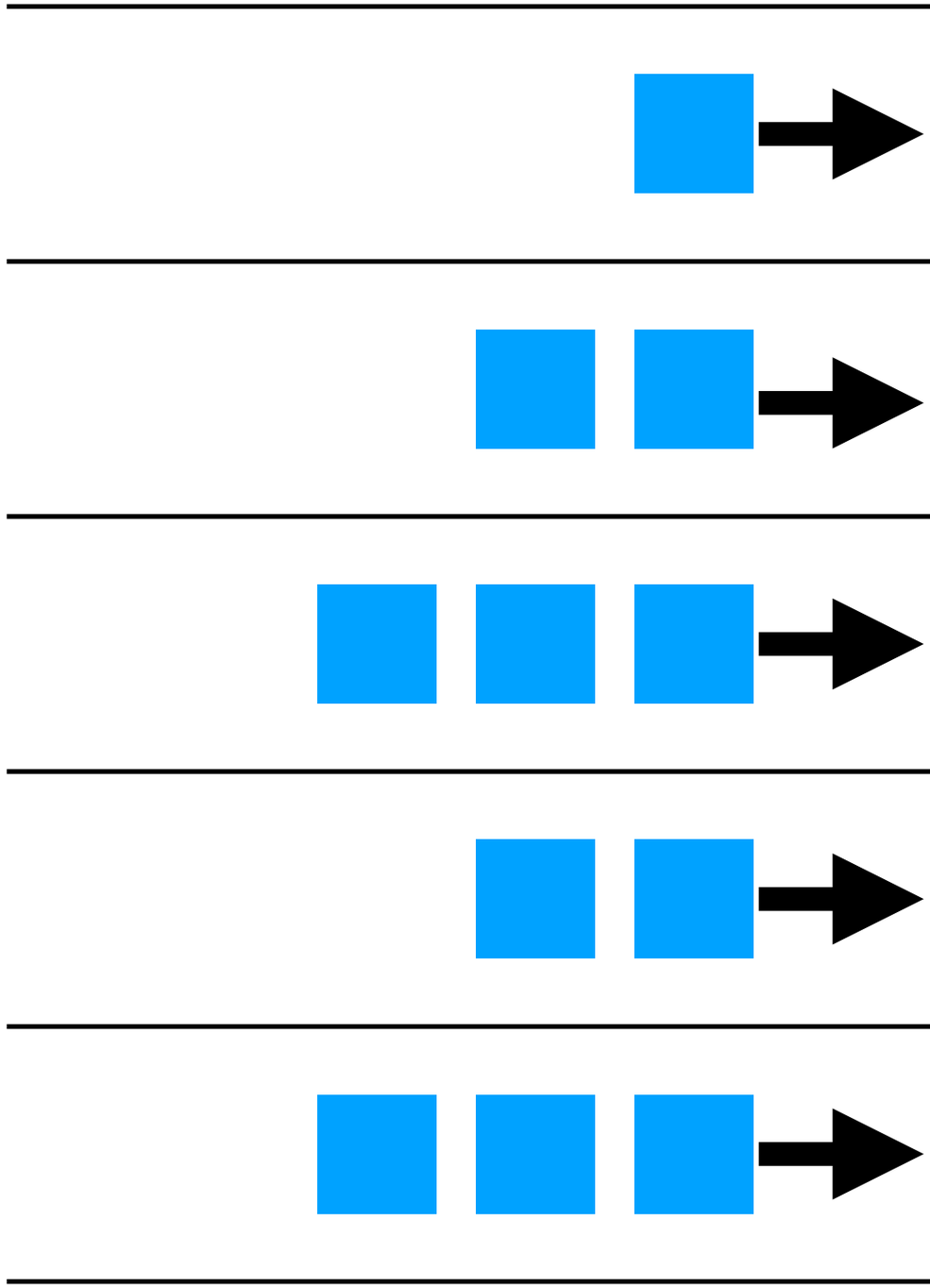
Output ports with queues



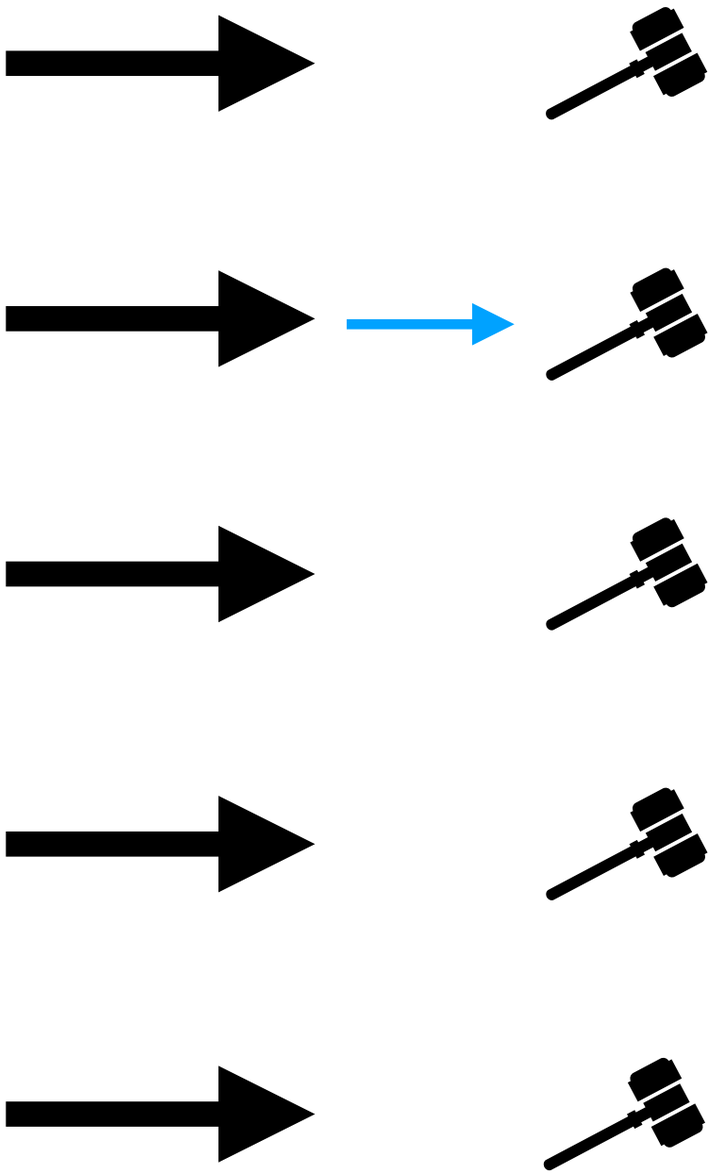
Input ports



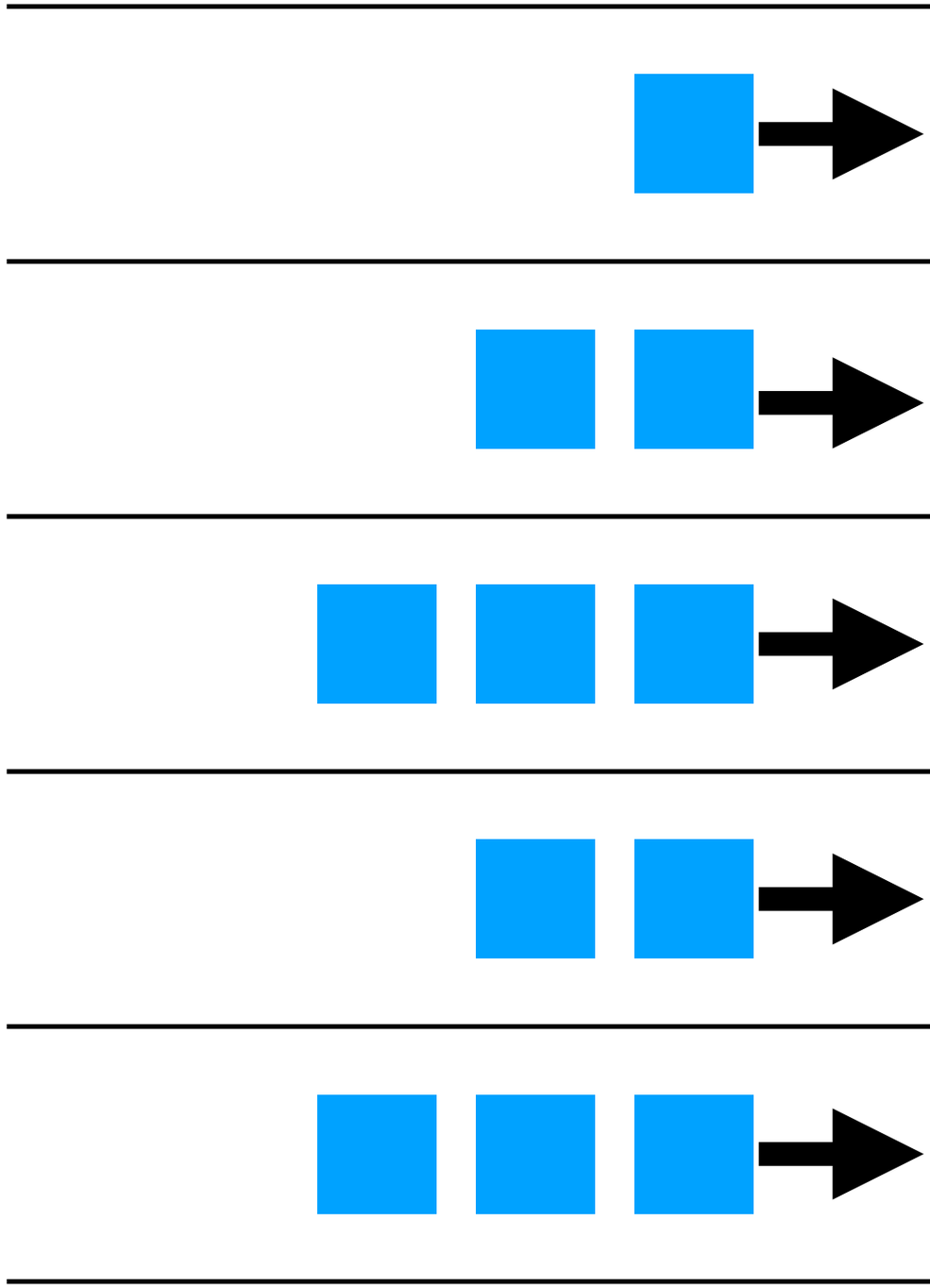
Output ports with queues



Input ports

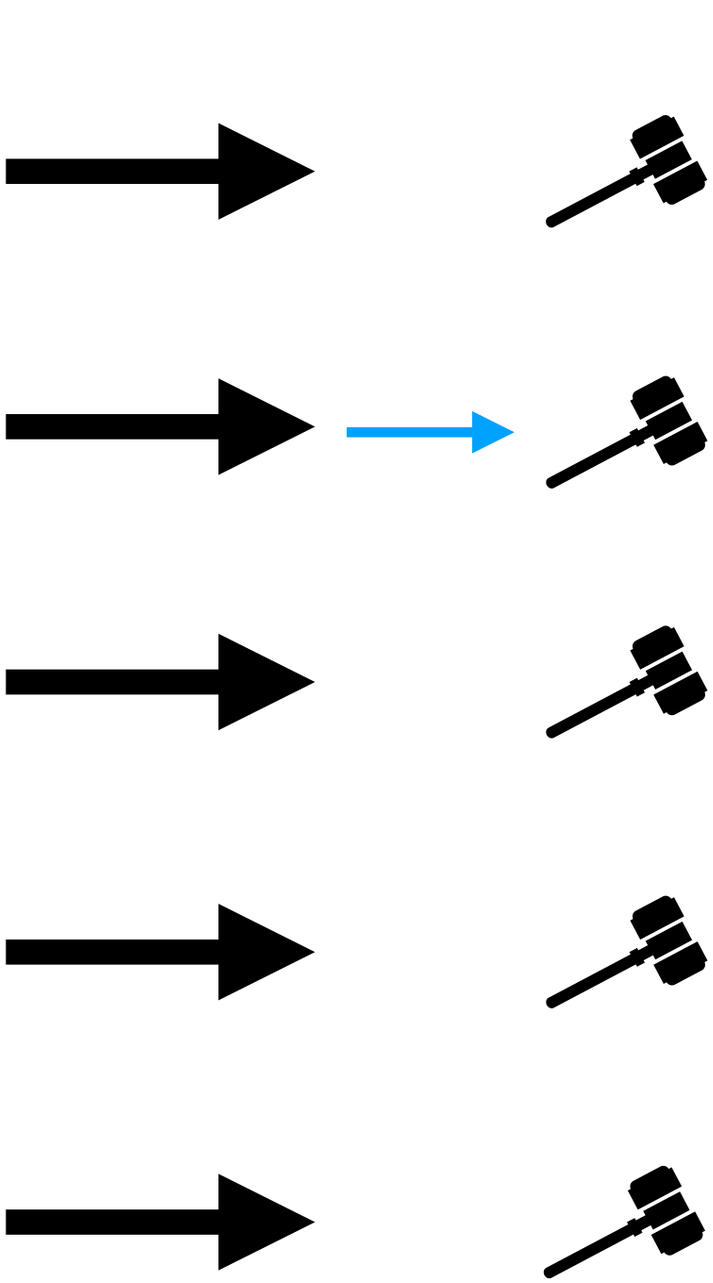


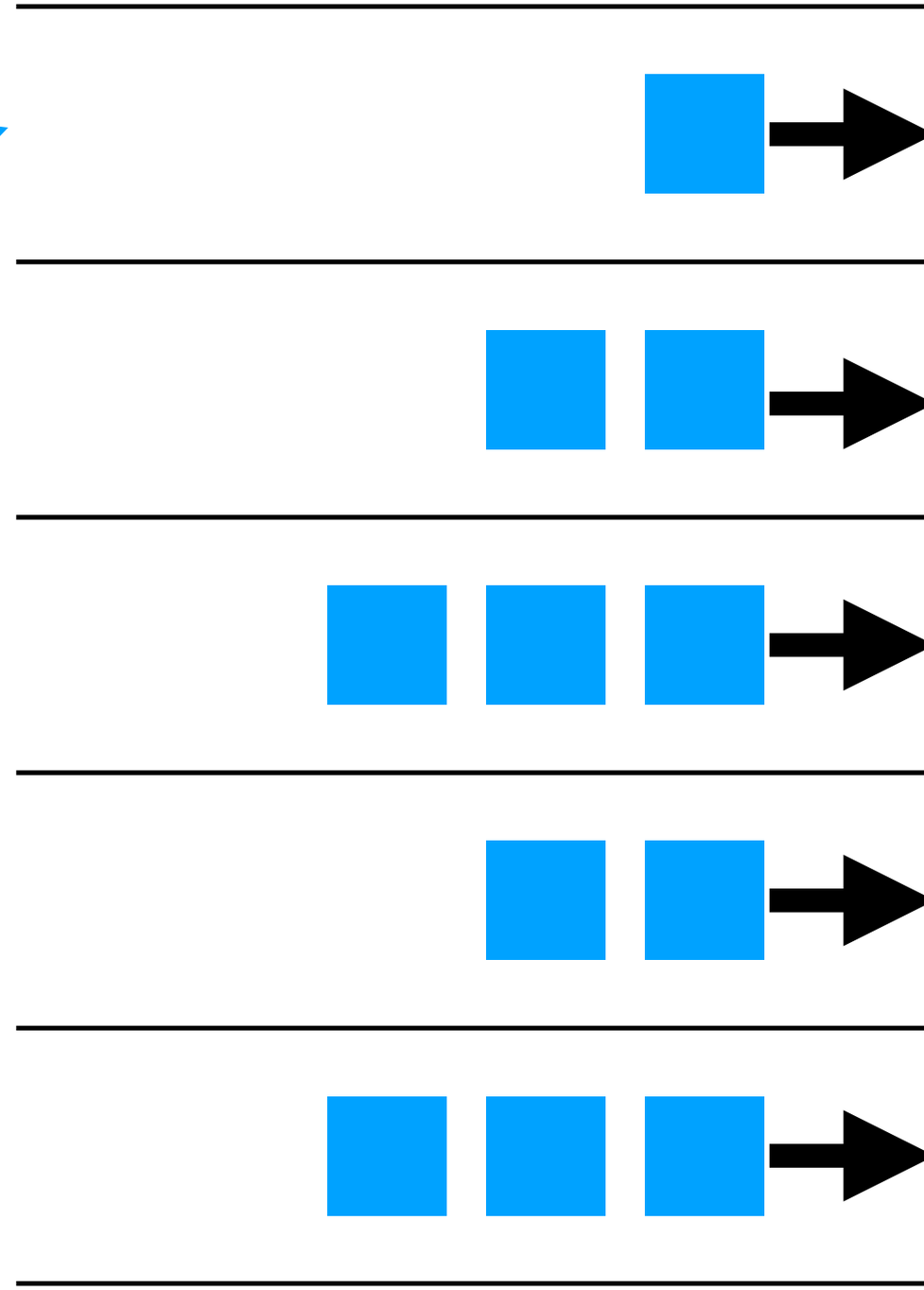
Output ports with queues



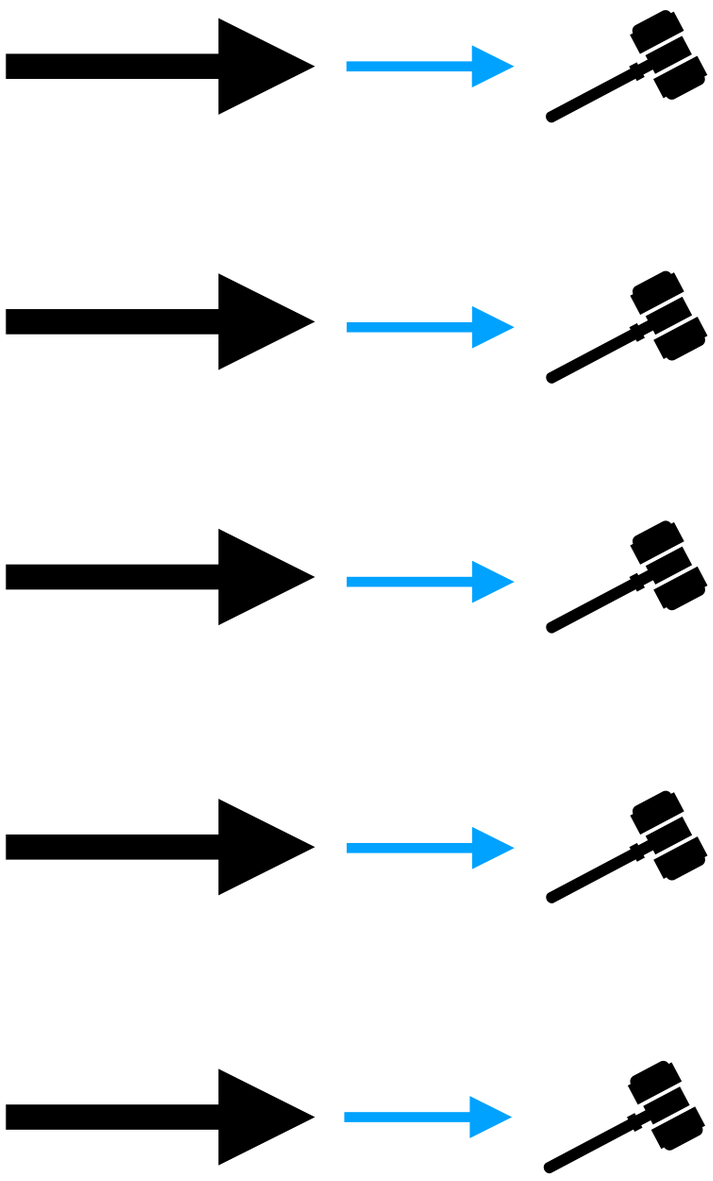
Input ports

Output ports with queues

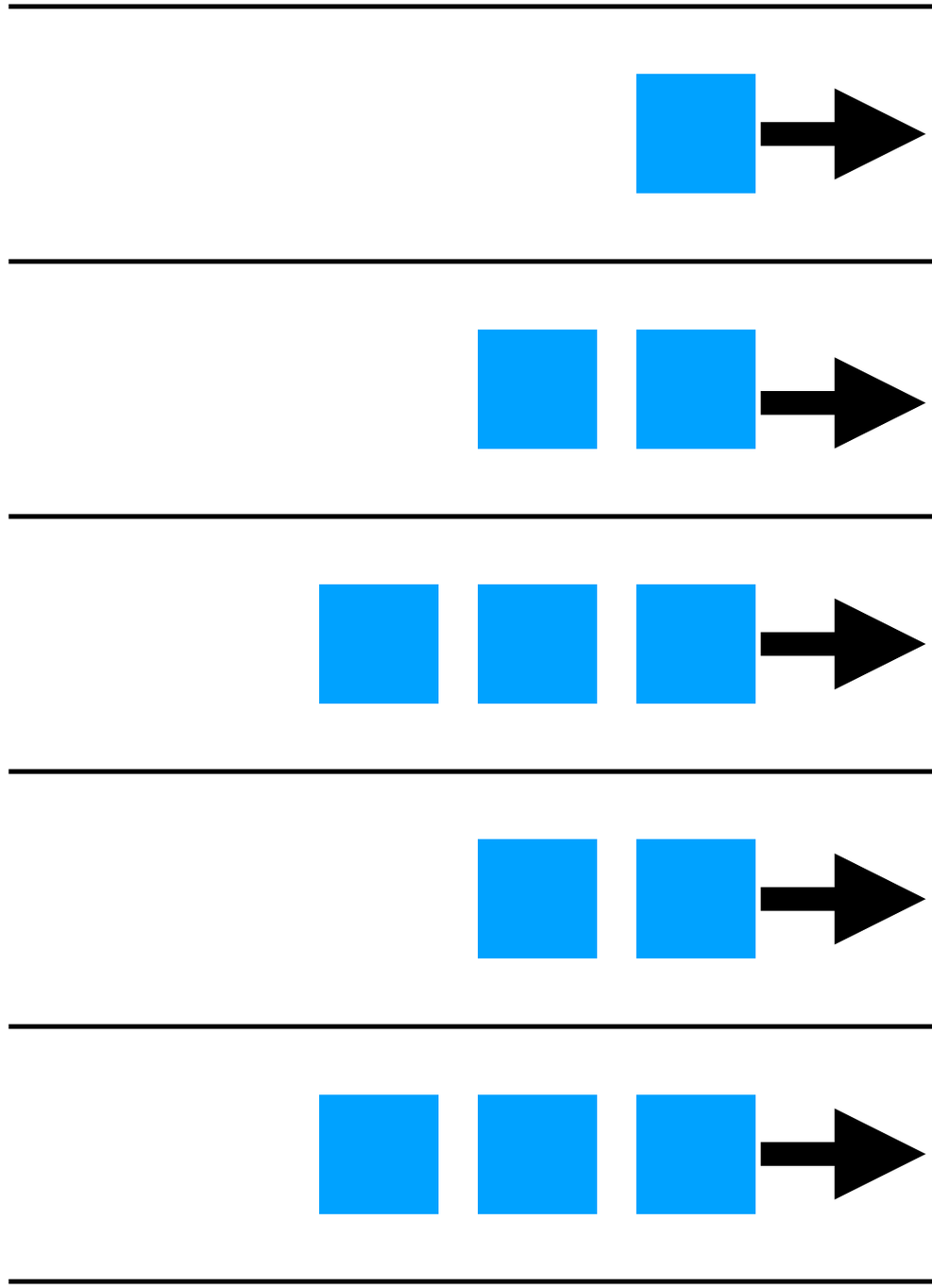




Input ports

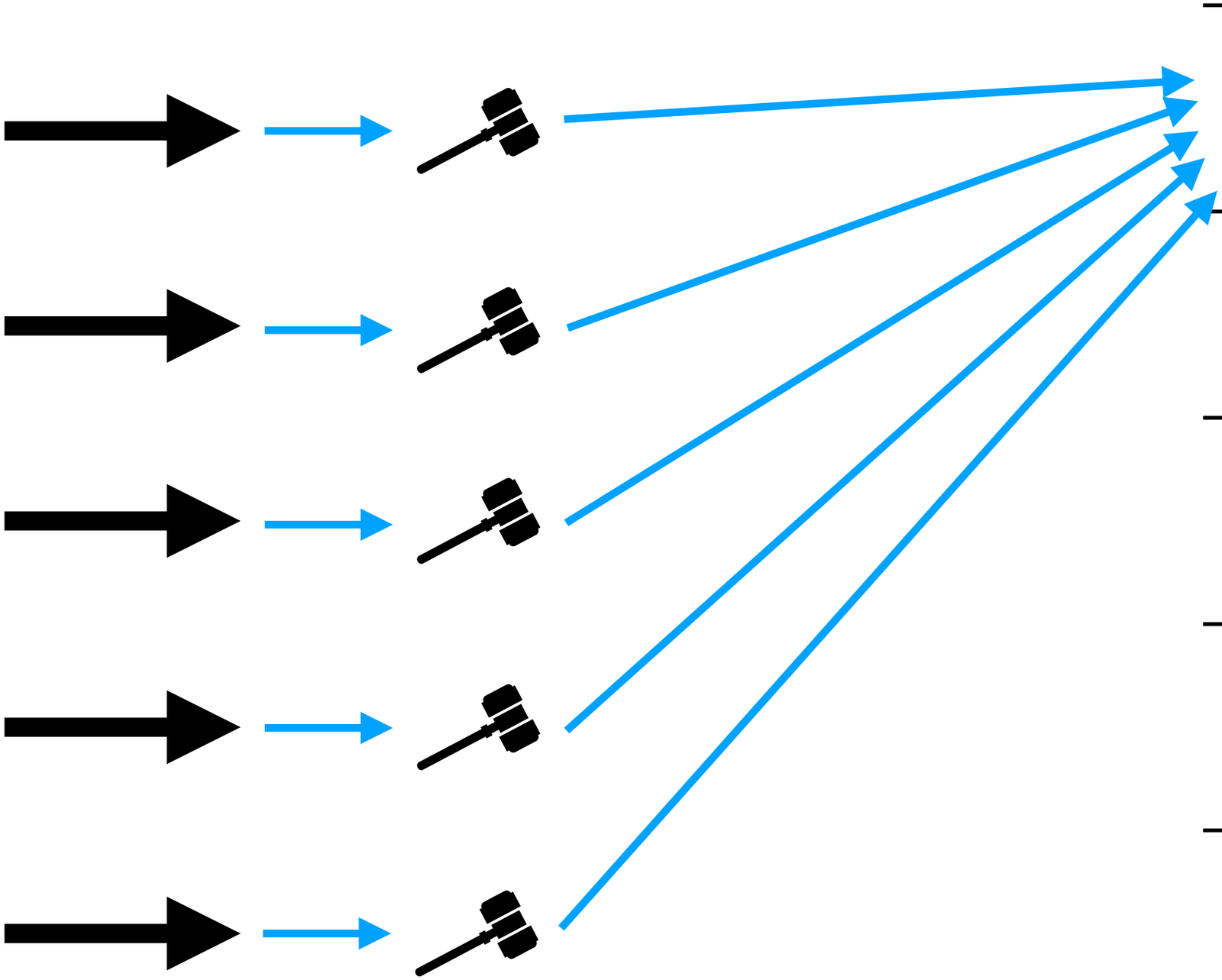


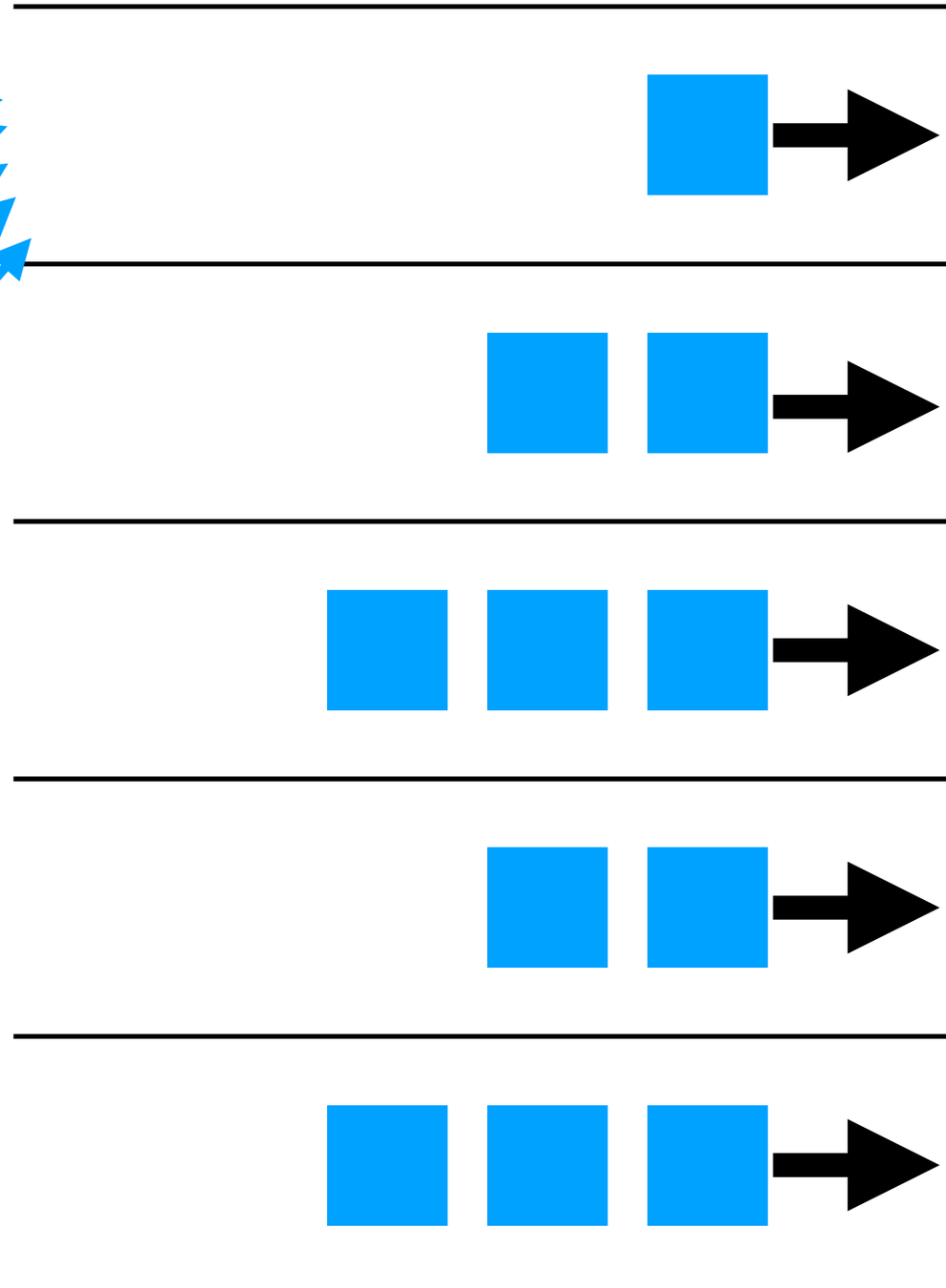
Output ports with queues



Input ports

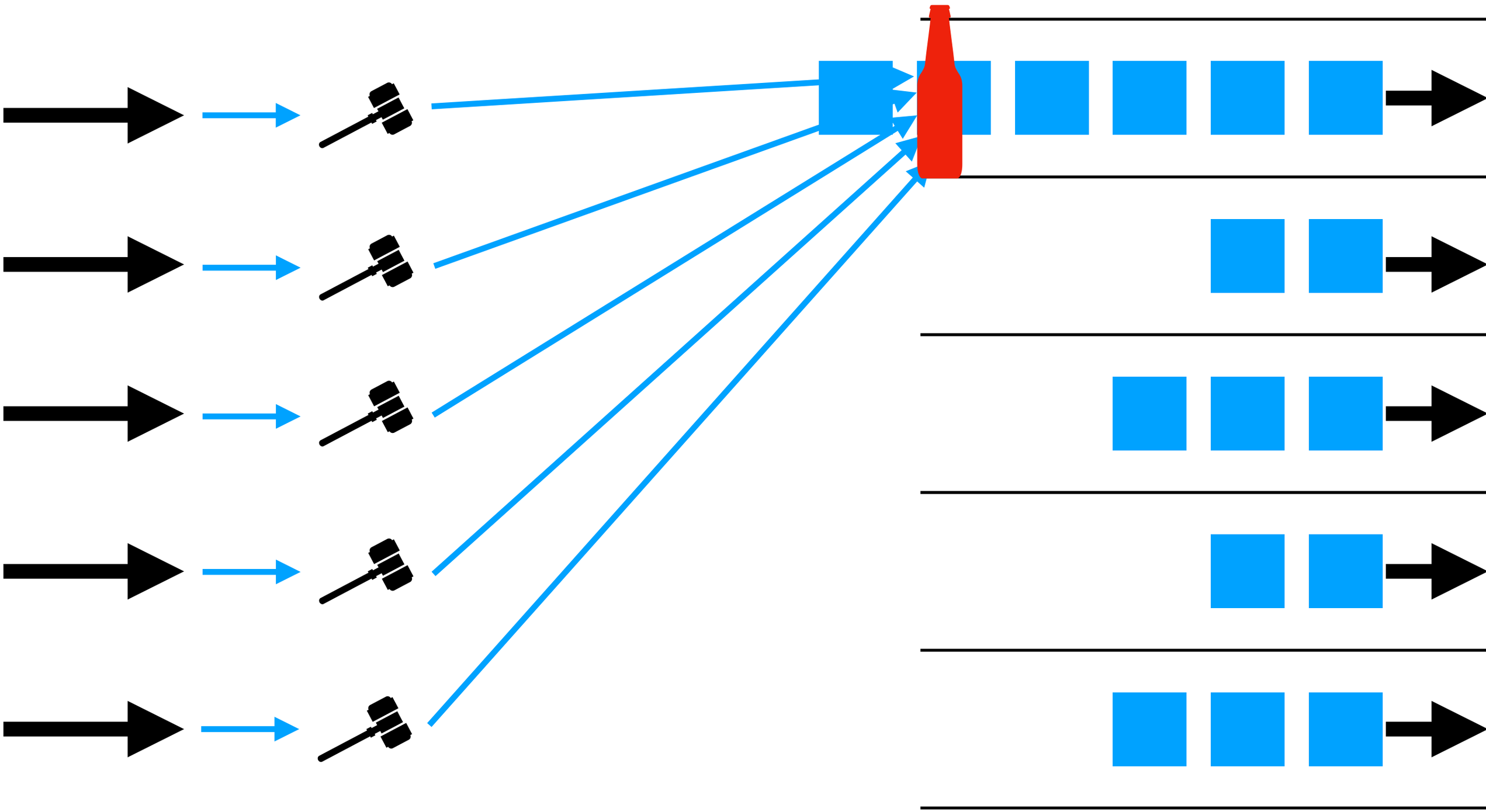
Output ports with queues





Input ports

Output ports with queues

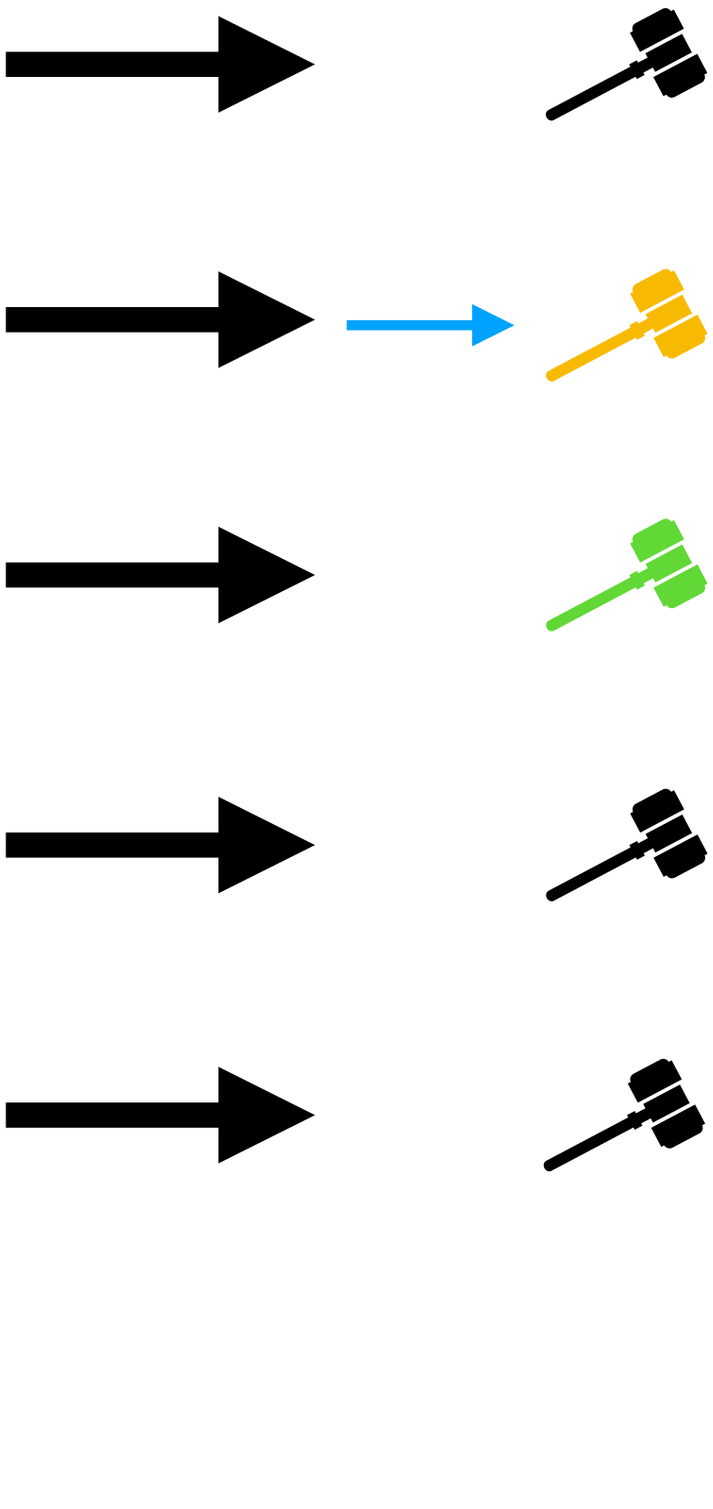


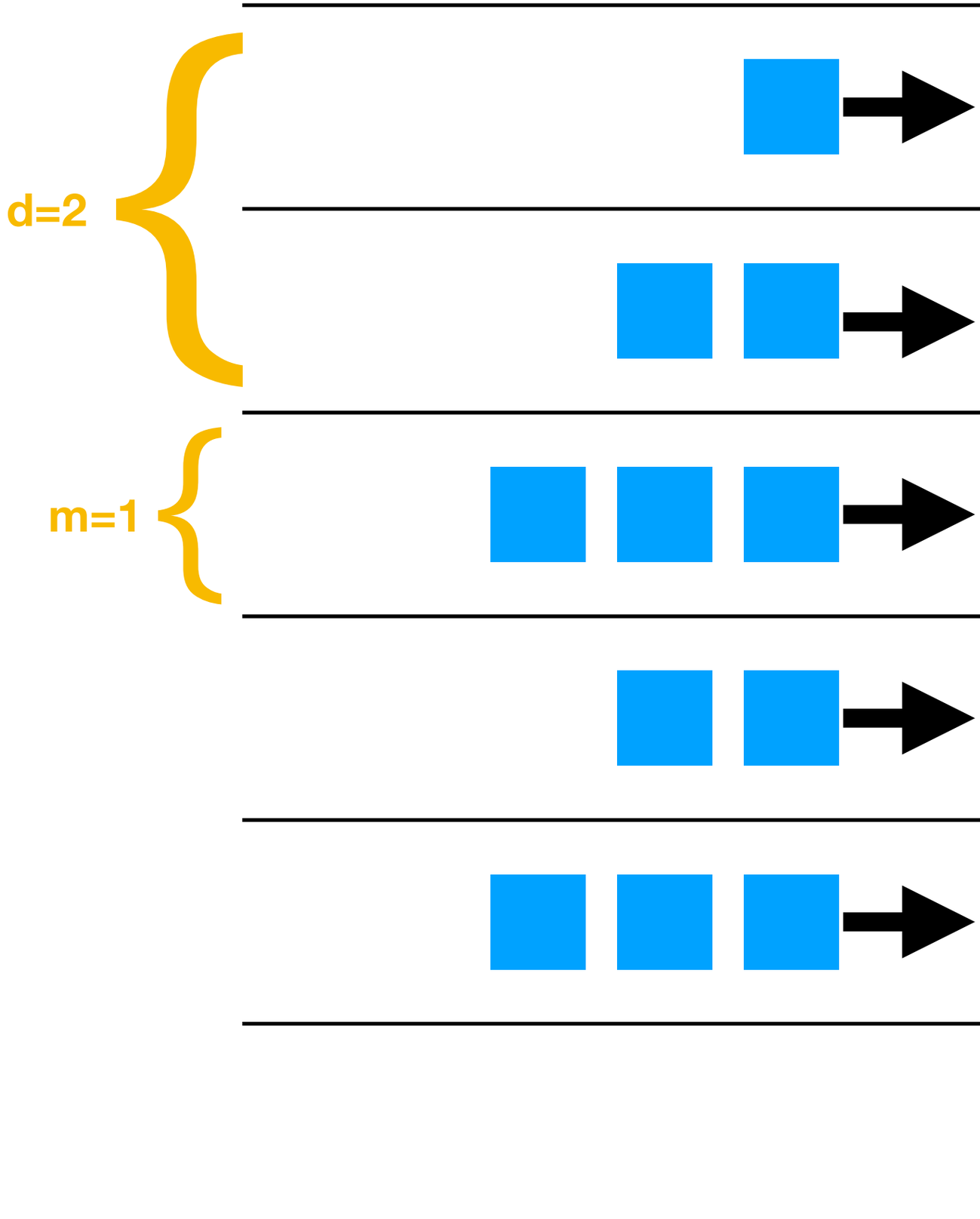
Load-Balancing

- **d** randomly sampled output queues
- **m** of the least-loaded output queues during previous samplings
- Pick minimum
- (d, m) scheduling policy

Input ports

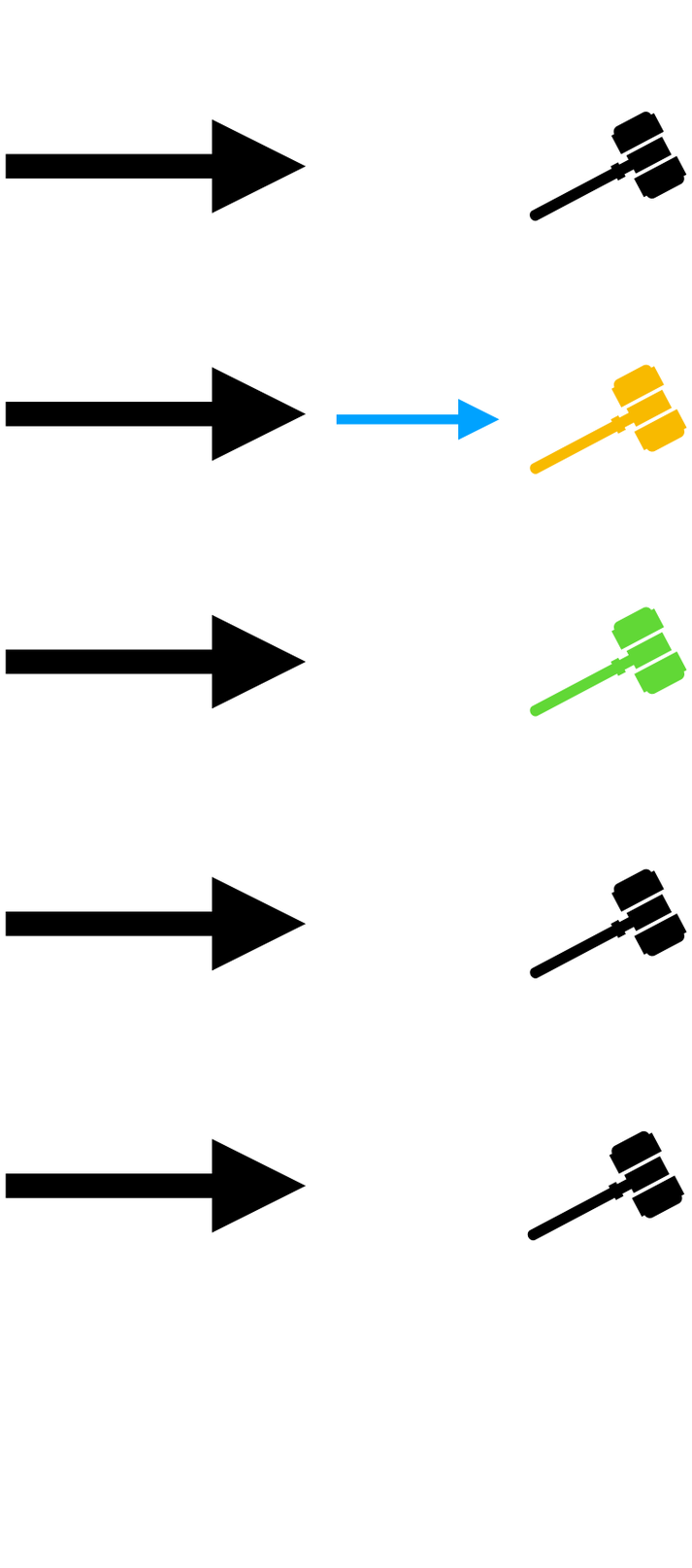
Output ports with queues

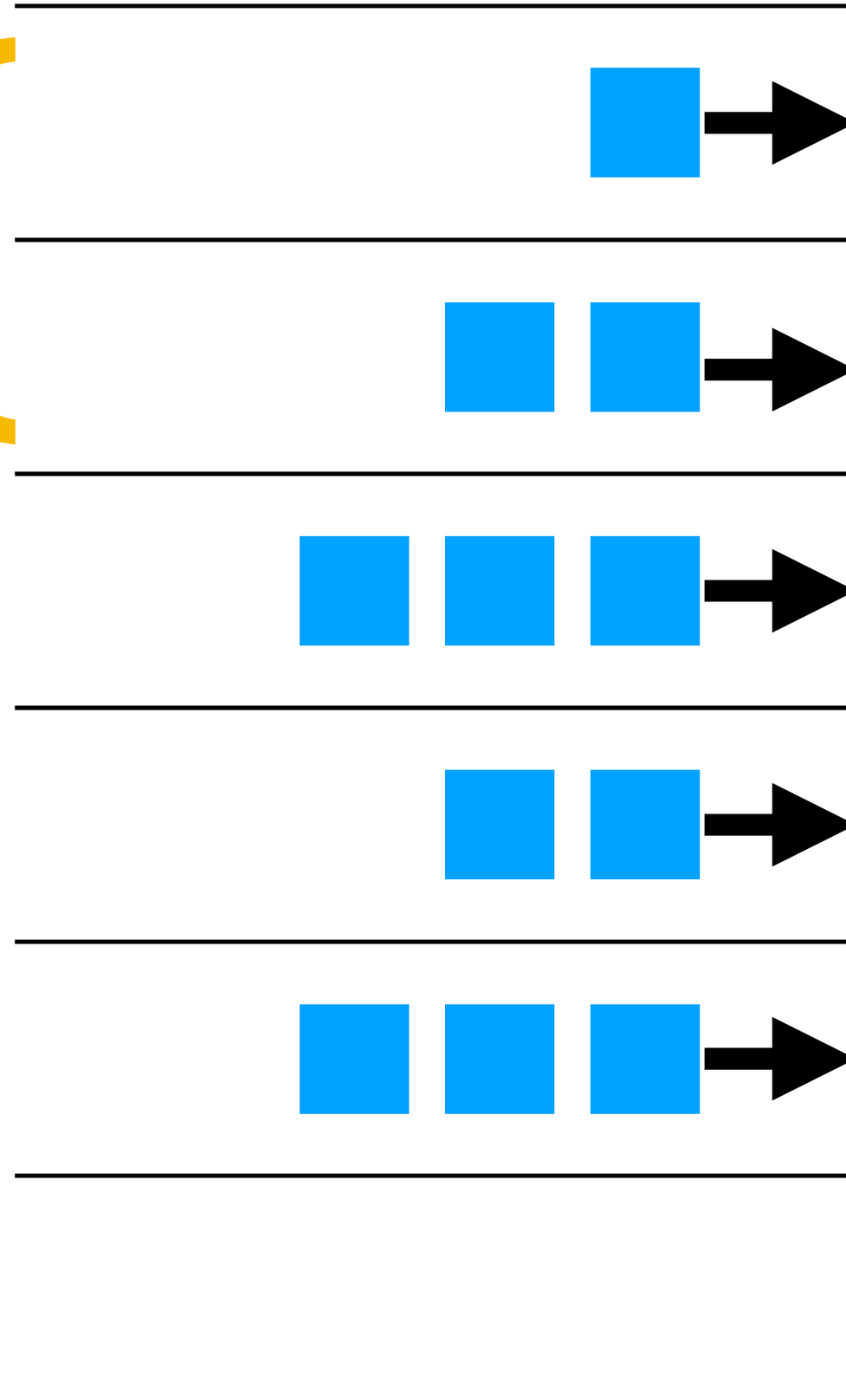




Input ports

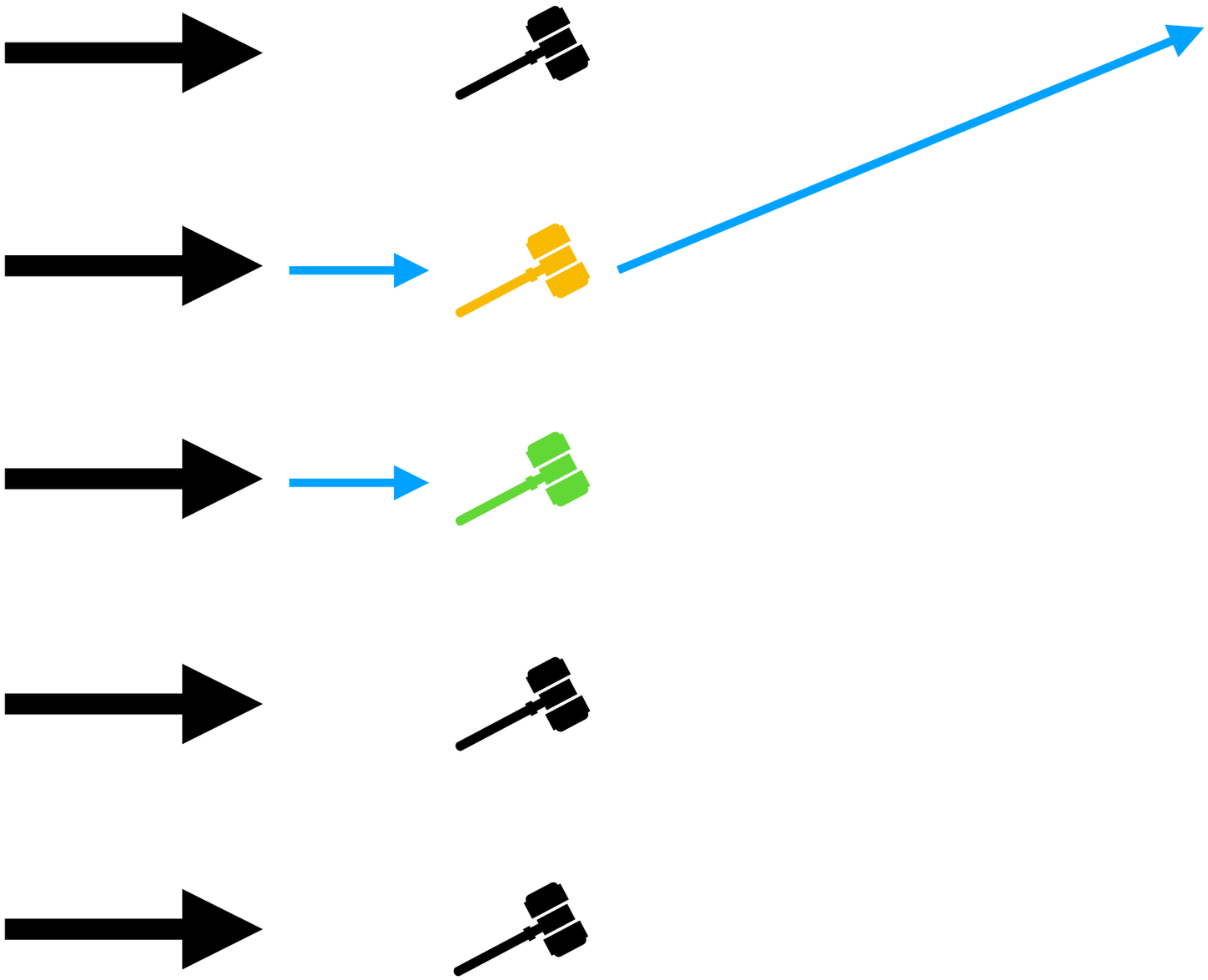
Output ports with queues

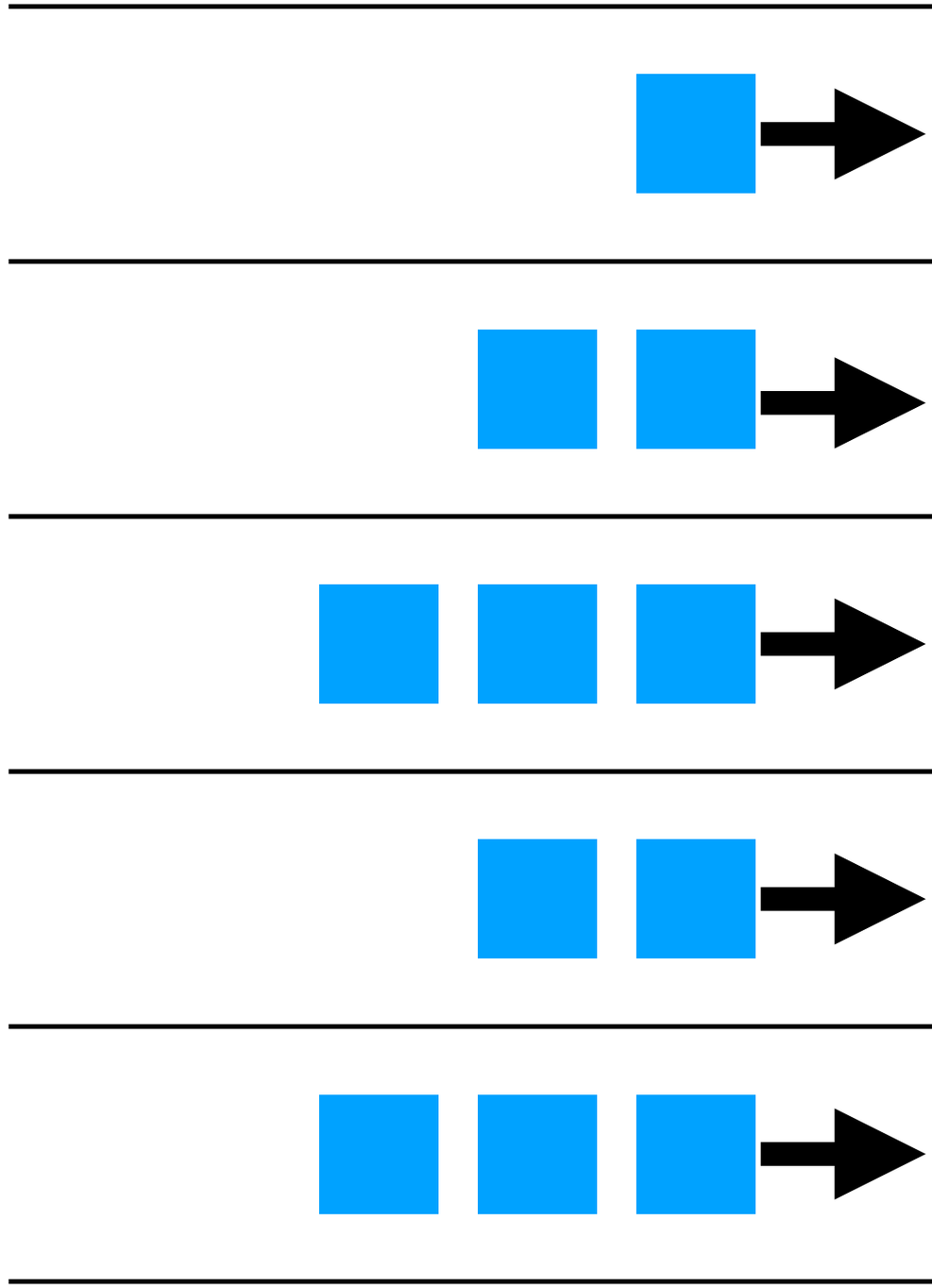




Input ports

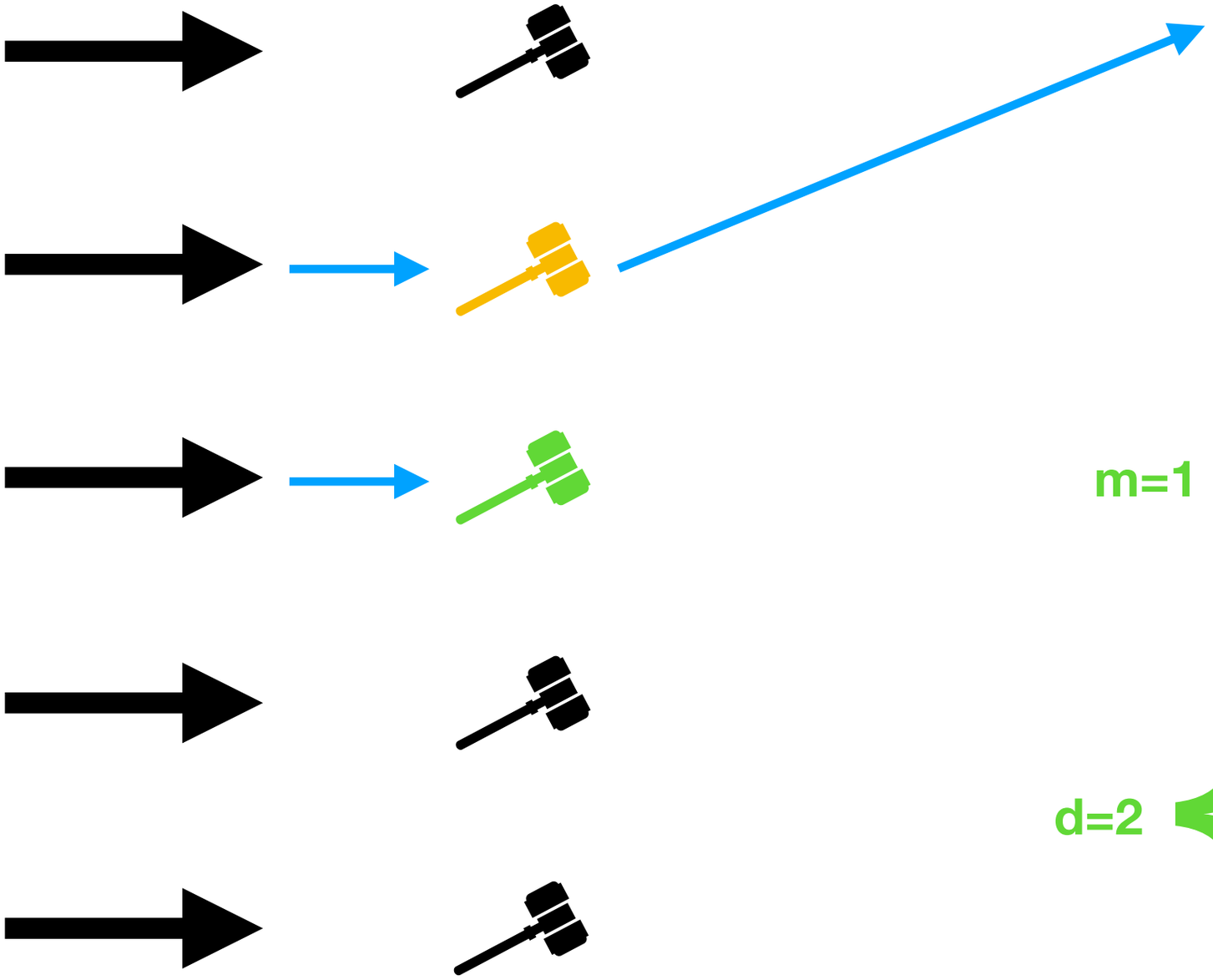
Output ports with queues

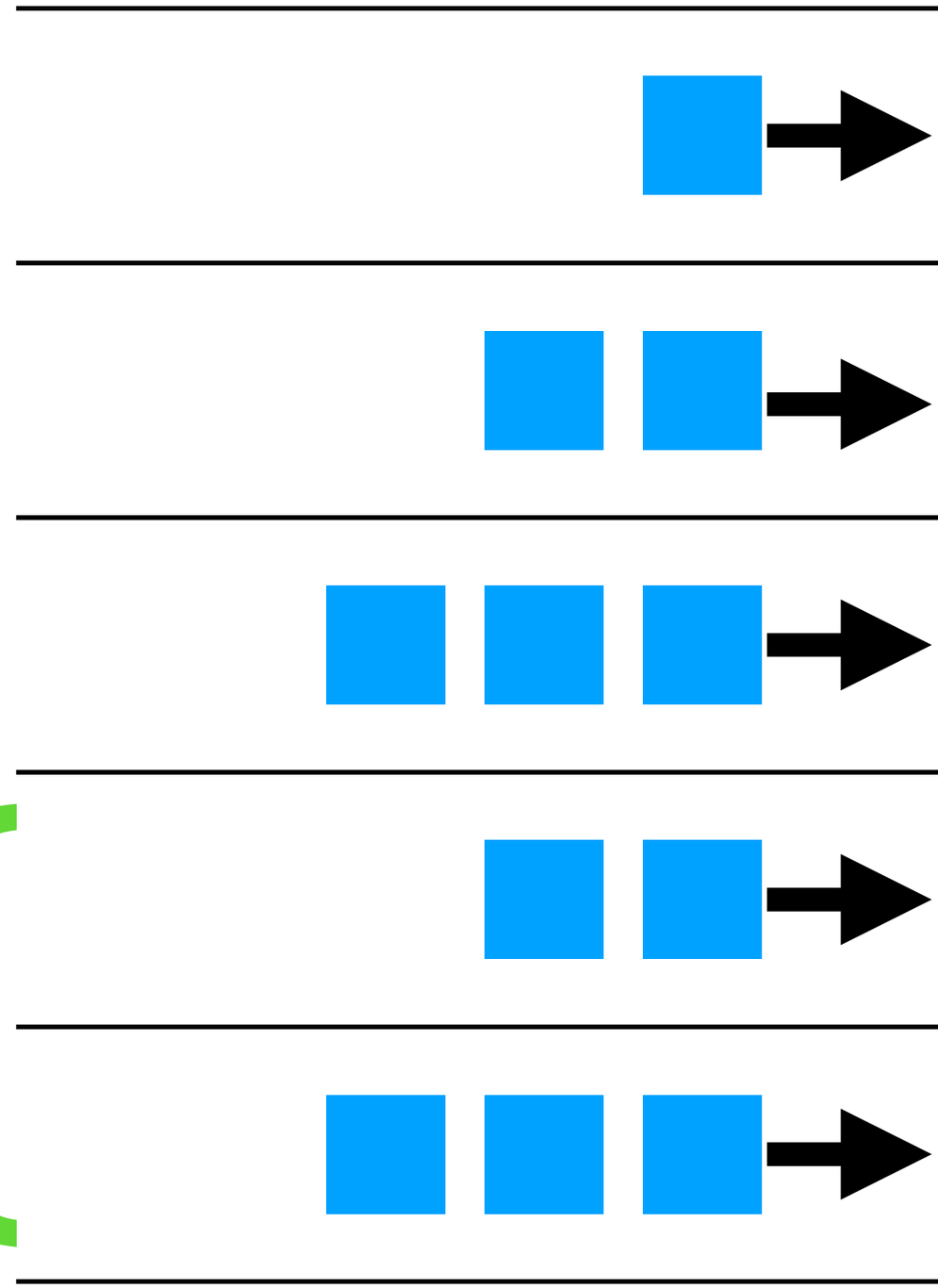




Input ports

Output ports with queues





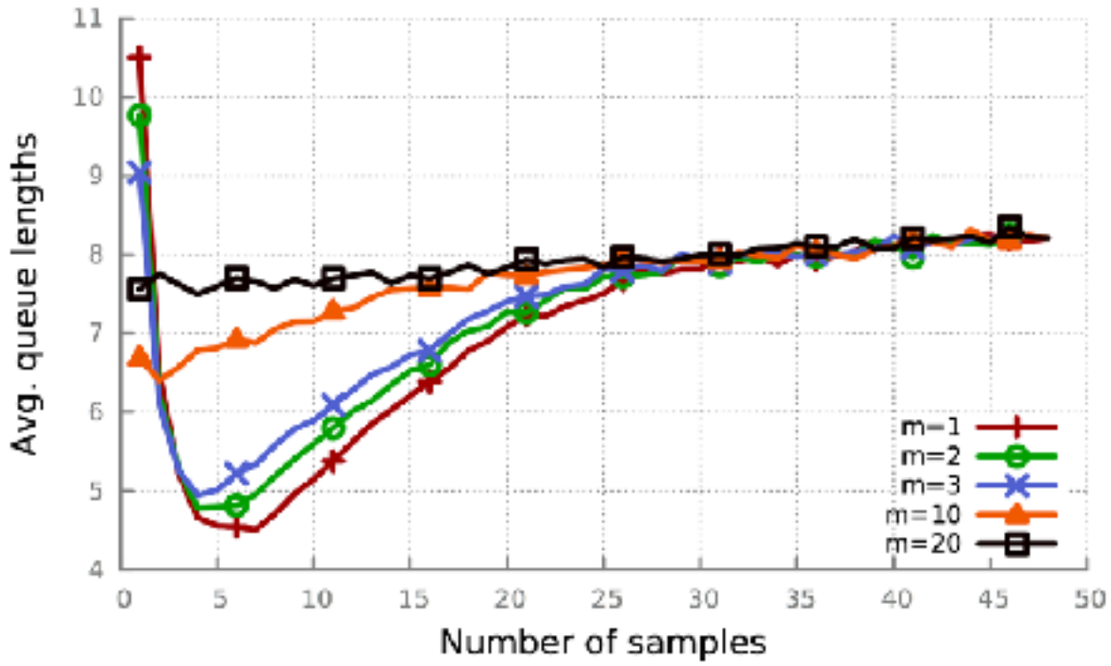
Input ports

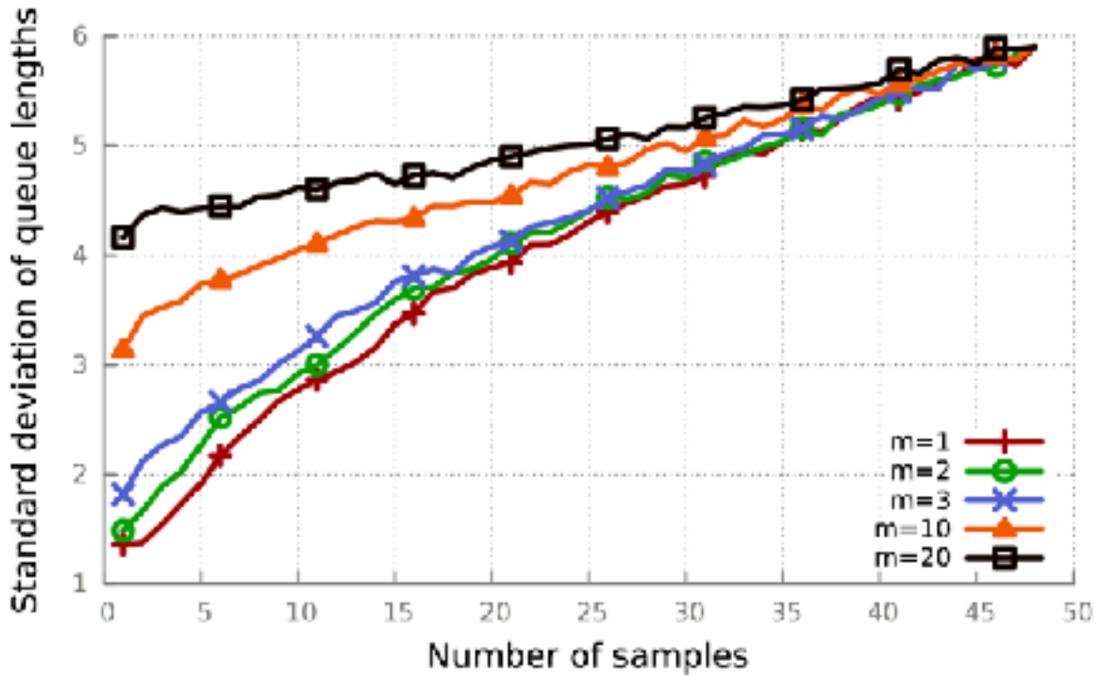
Output ports with queues

m=1

d=2

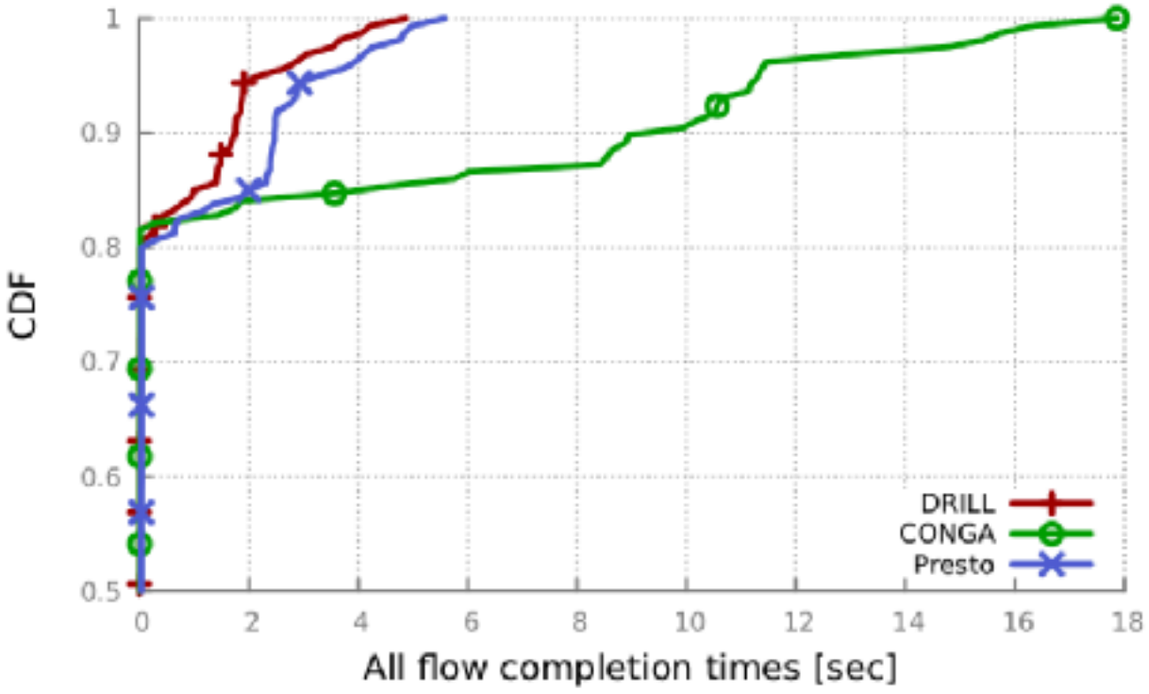
Choosing the right parameters

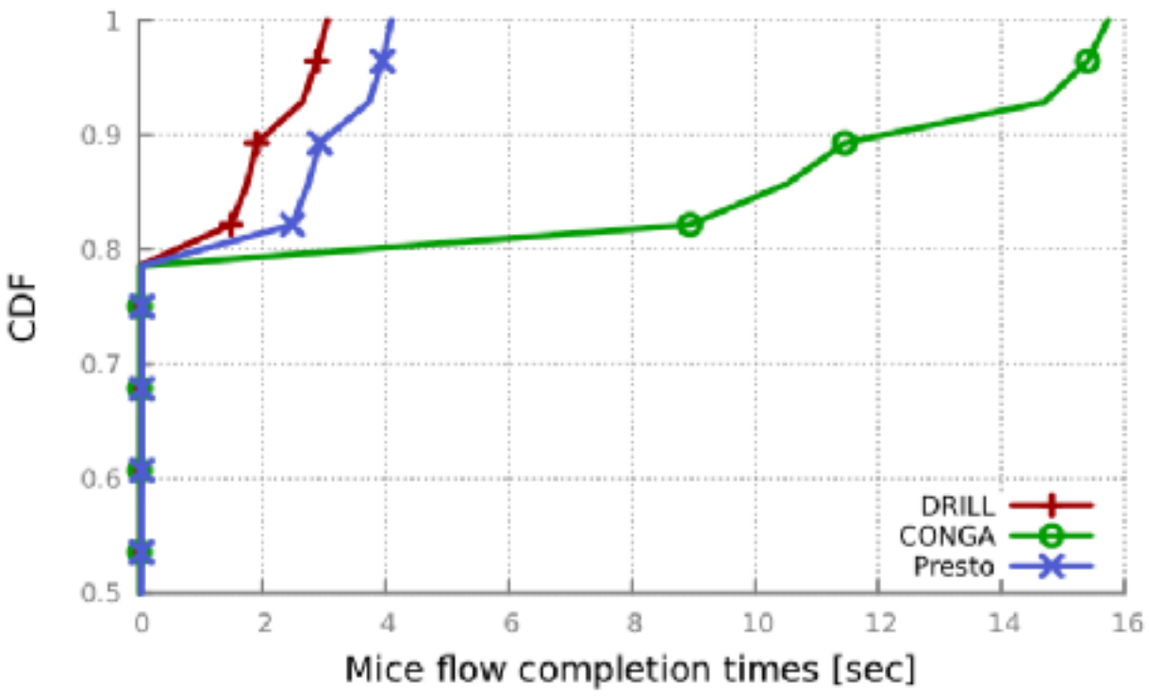




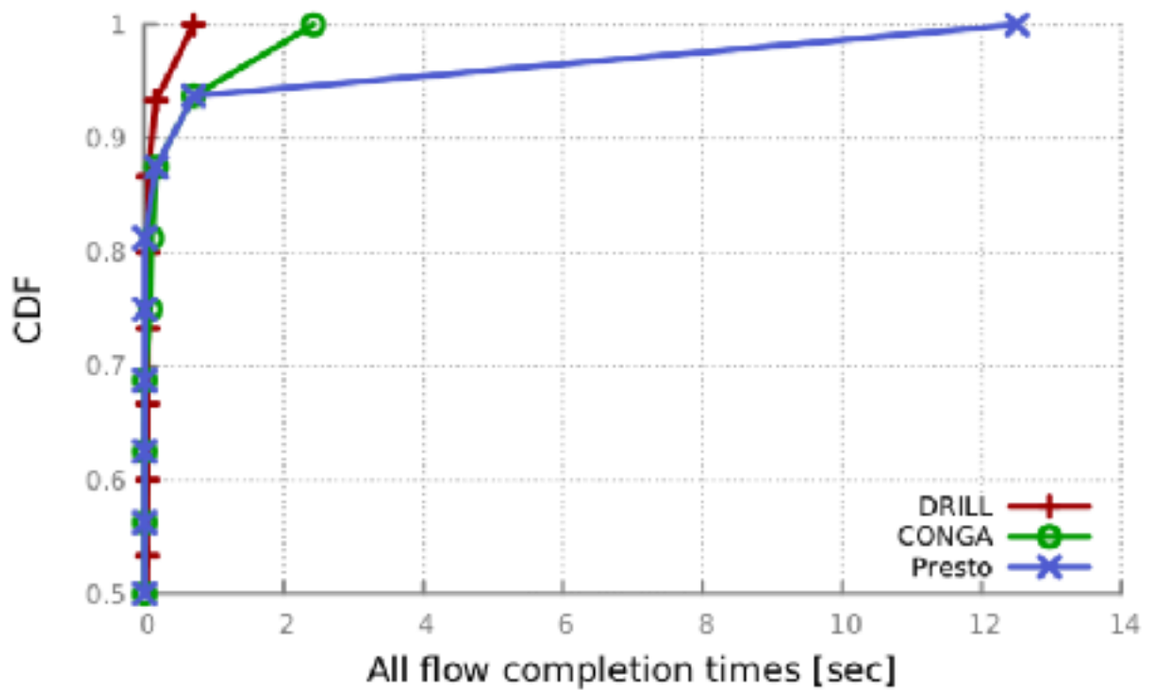
(48-port switch under 100% load)

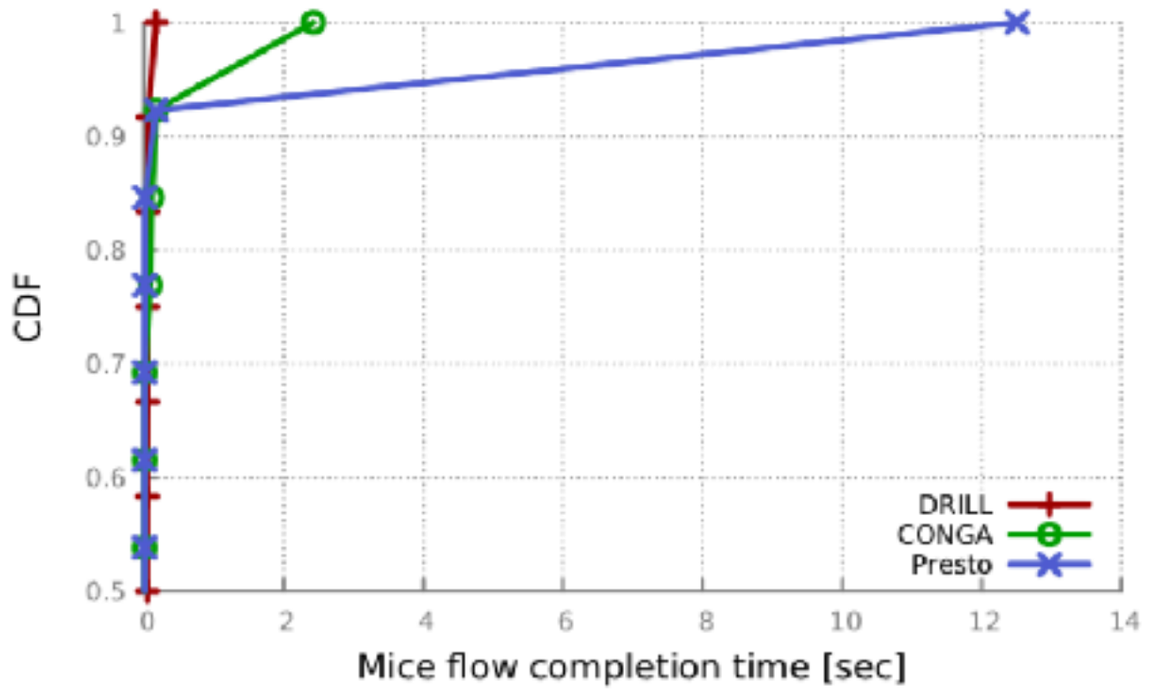
Heavy Load





Incast Scenario



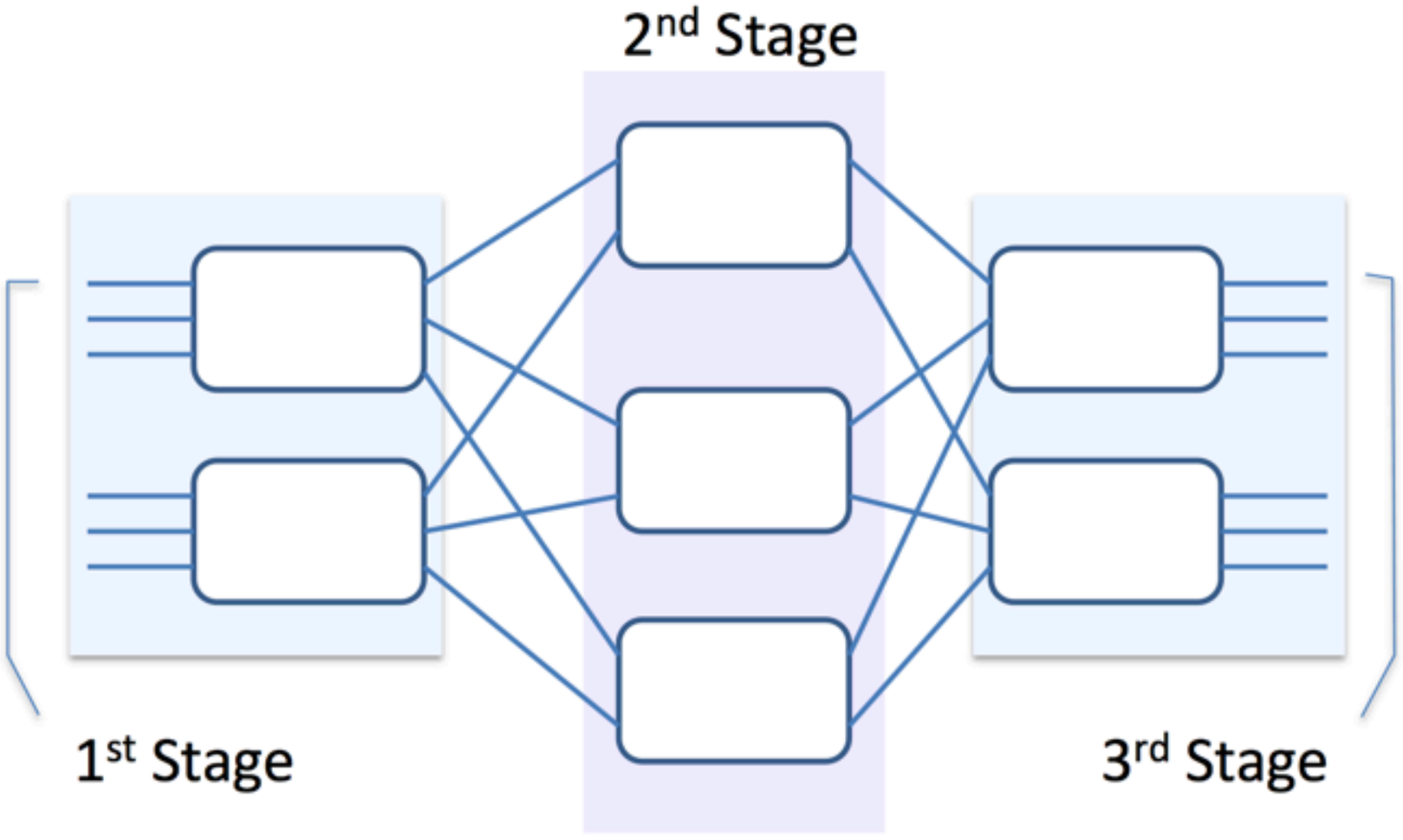


Out of order delivery

- LetFlow taught us that packet-level load balancing causes out of order delivery
- Splitting flows + Latency Variance = Reordering
- DRILL claims that its load balancing is so good that (almost) no reordering happens (no latency difference across different paths); even less than with Presto

Conclusion

- Only simulated evaluation, with synthetic traffic
- Simulated network quite small (4 Leaves, 6 Spines)
- What to do with **m** if network not fully interconnected?
- Asymmetric systems might impose interesting problems: SIGCOMM talk mentions approaches (graph decomposition)


2nd Stage
1st Stage
3rd Stage