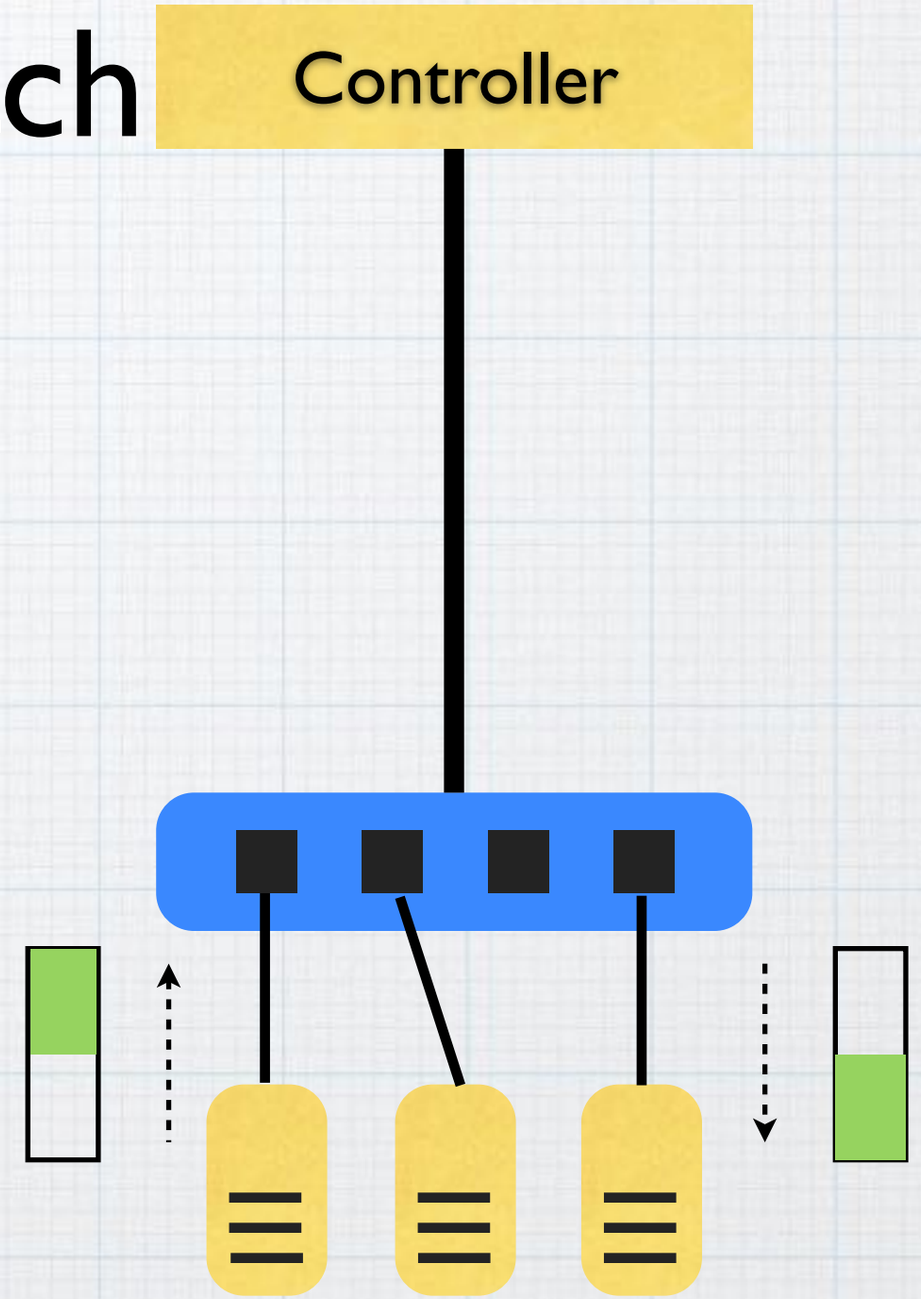
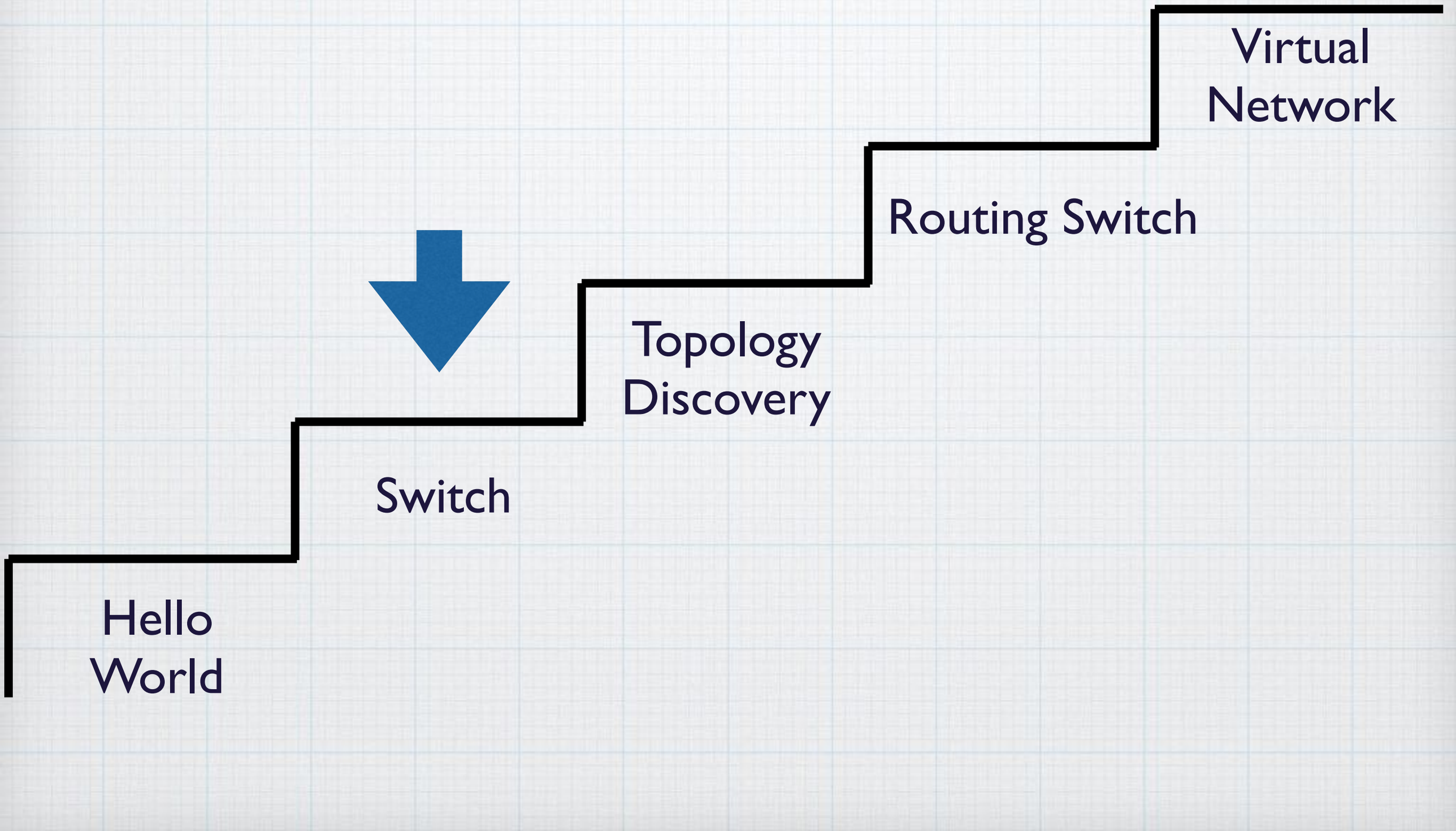
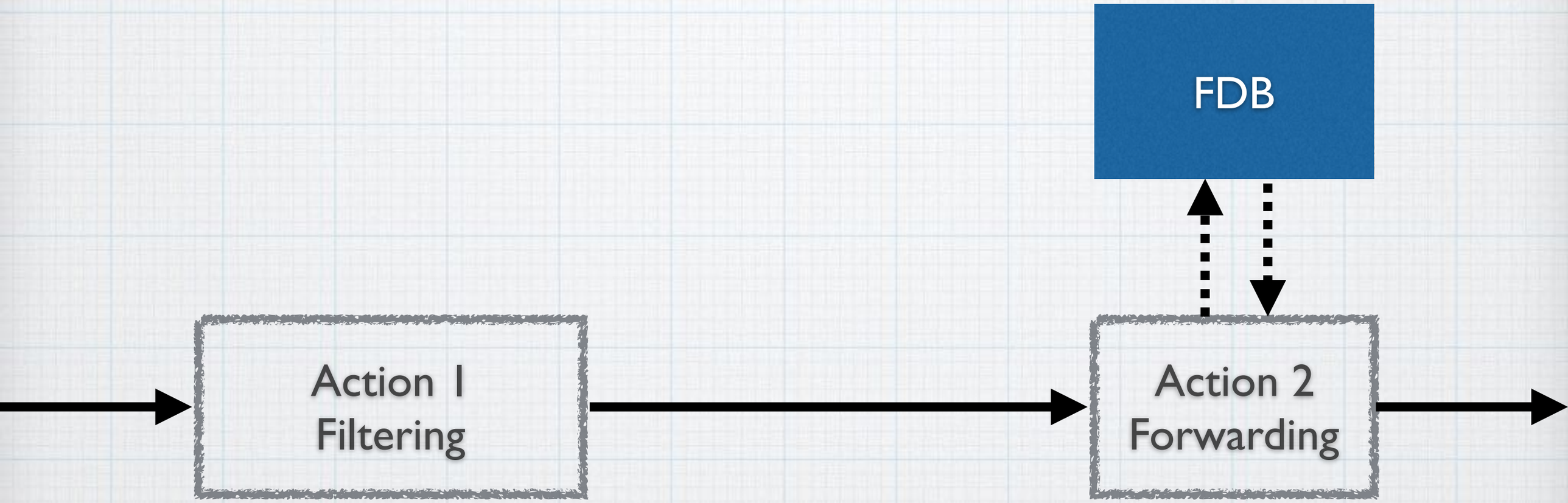
Design a Learning Switch and Controller with OpenFlow1.3




Virtual
Network
Routing Switch
Topology
Discovery
Switch
Hello
World

Packet processing with OpenFlow 1.0

Learning Switch



Drop packets that matches

Dst MAC =

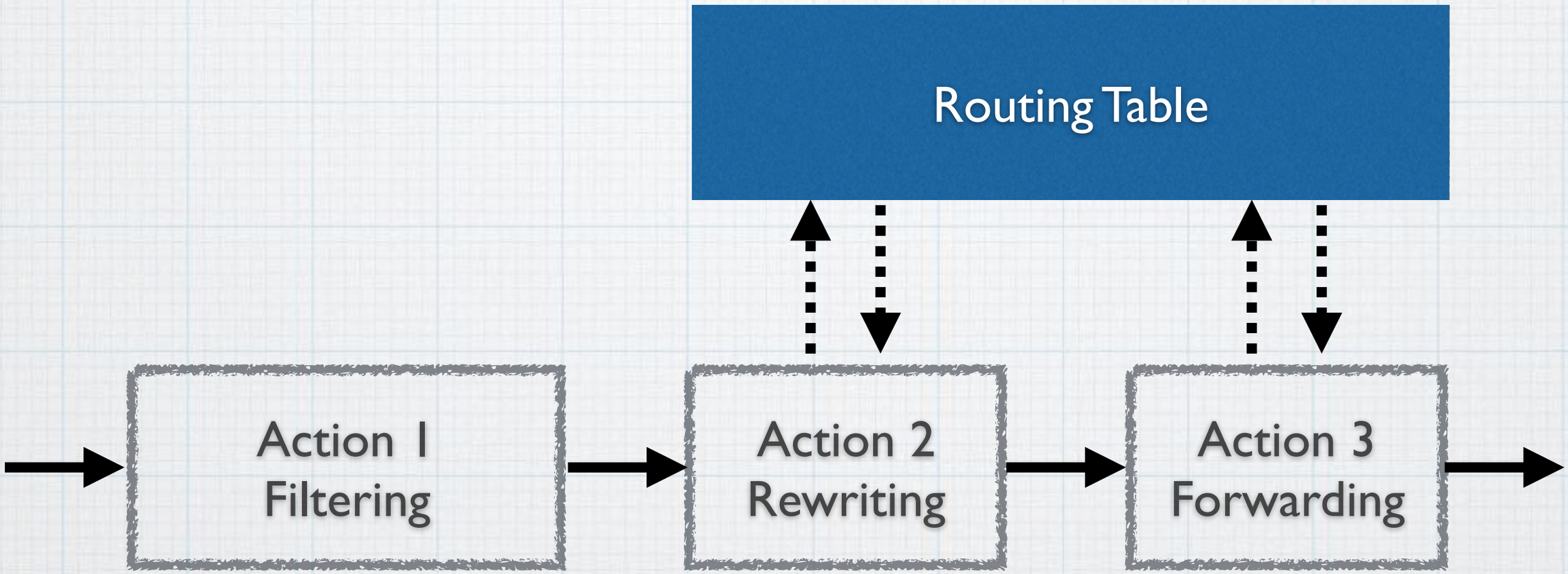
- 802.1D/802.1Q reserved MAC
- Multicast packet

Send packets out port #n

or

Flood them

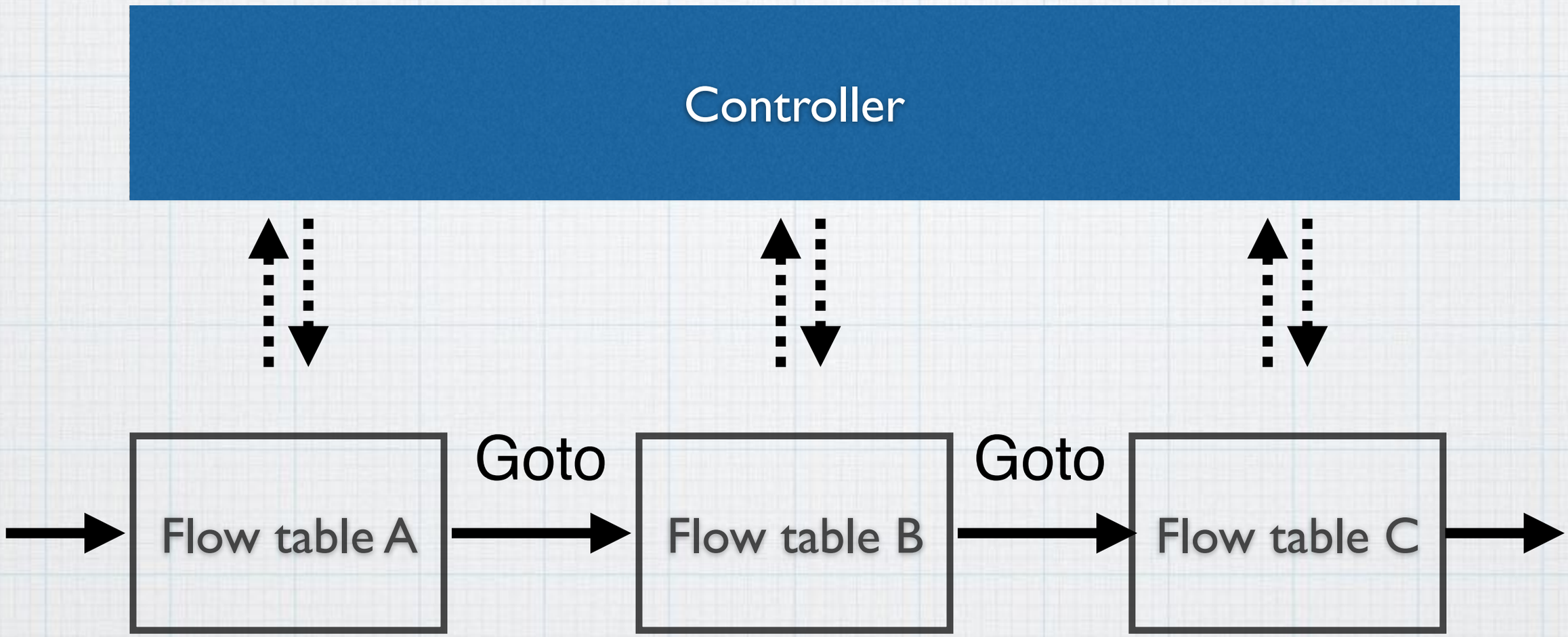
Example: Router



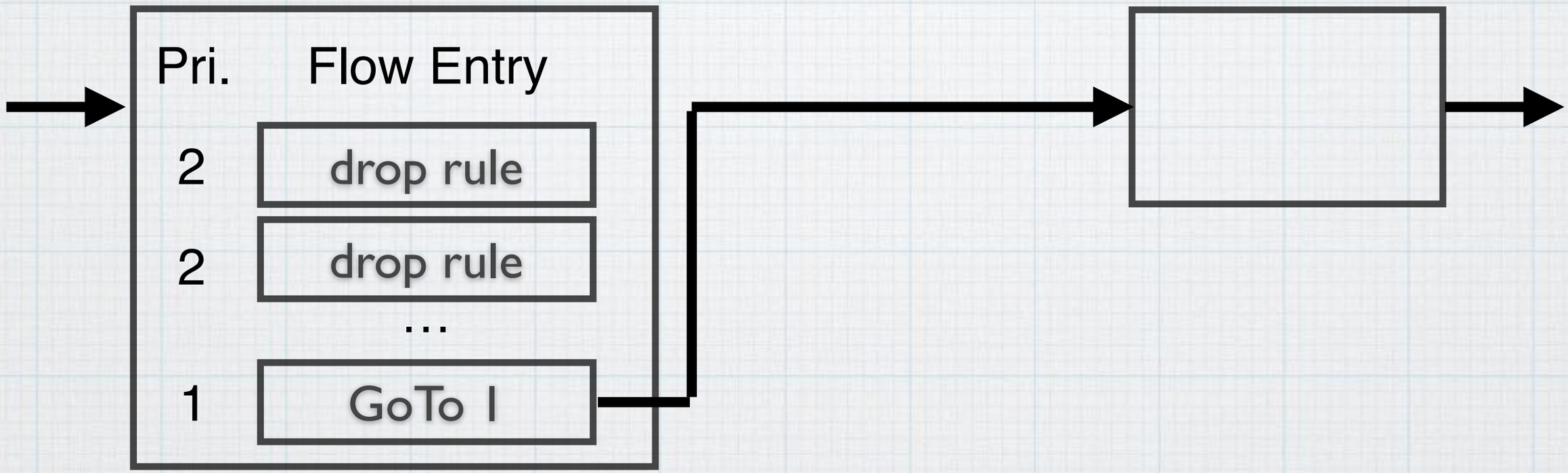
Src/Dst
MAC address

Packet processing with OpenFlow 1.3

Multiple Tables



Learning Switch



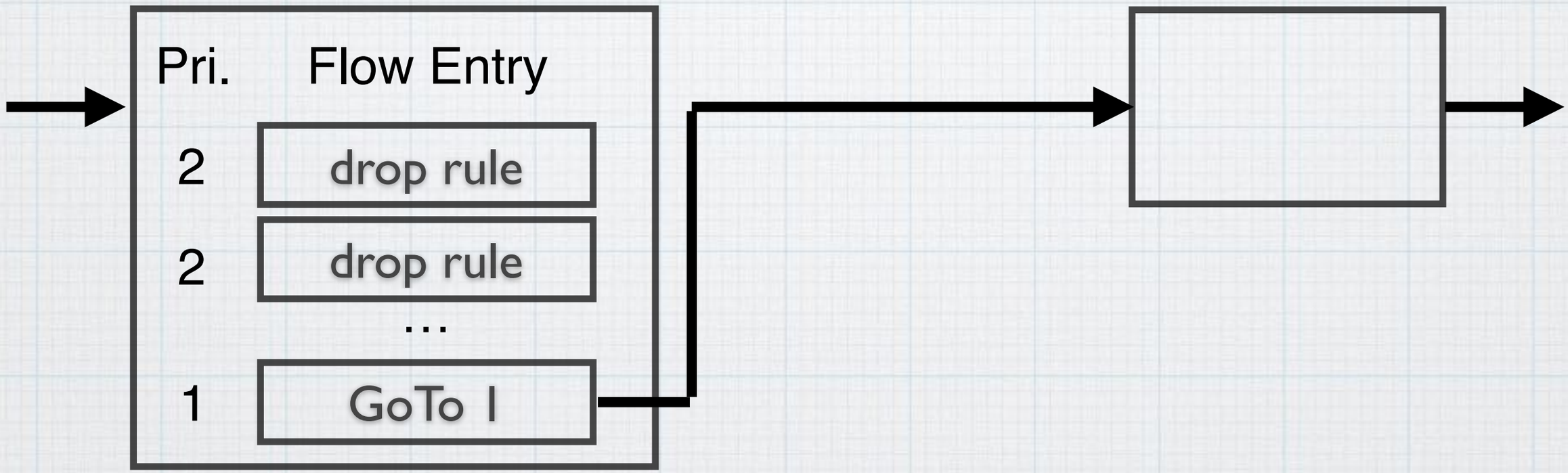
```
send_flow_mod_add(
      datapath_id,
      table_id: 0,
      idle_timeout: 0,
      priority: 2,
      match: Match.new(ether_destination_address: '01:00:5e:00:00:00',
                       ether_destination_address_mask: 'ff:ff:ff:00:00:00')
)
```

- Discard multicast packets
- Process packets by using the first table (ID=0)

```
send_flow_mod_add(
      datapath_id,
      table_id: 0,
      idle_timeout: 0,
      priority: 1,
      match: Match.new,
      instructions: GotoTable.new(1)
    )
```

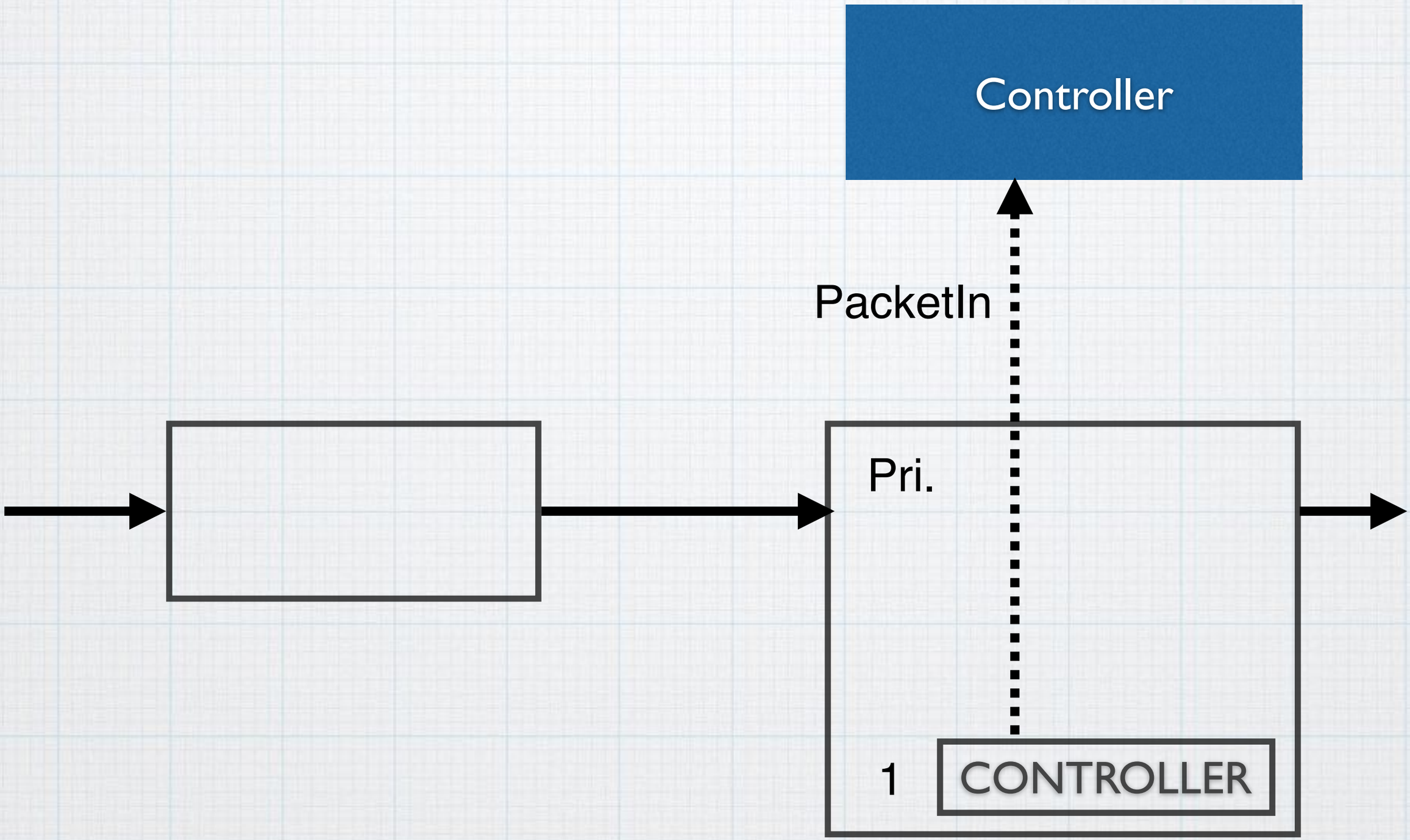
- If a packet does not match any rules in Table 0, then go to Table 1.
- First packets are matched with Filtering rules whose priority is high (pri=2), then they are matched with this rule (pri=1).

Learning Switch



Filtering table (ID=0)

Forwarding table (ID=1)



Filtering table (ID=0)

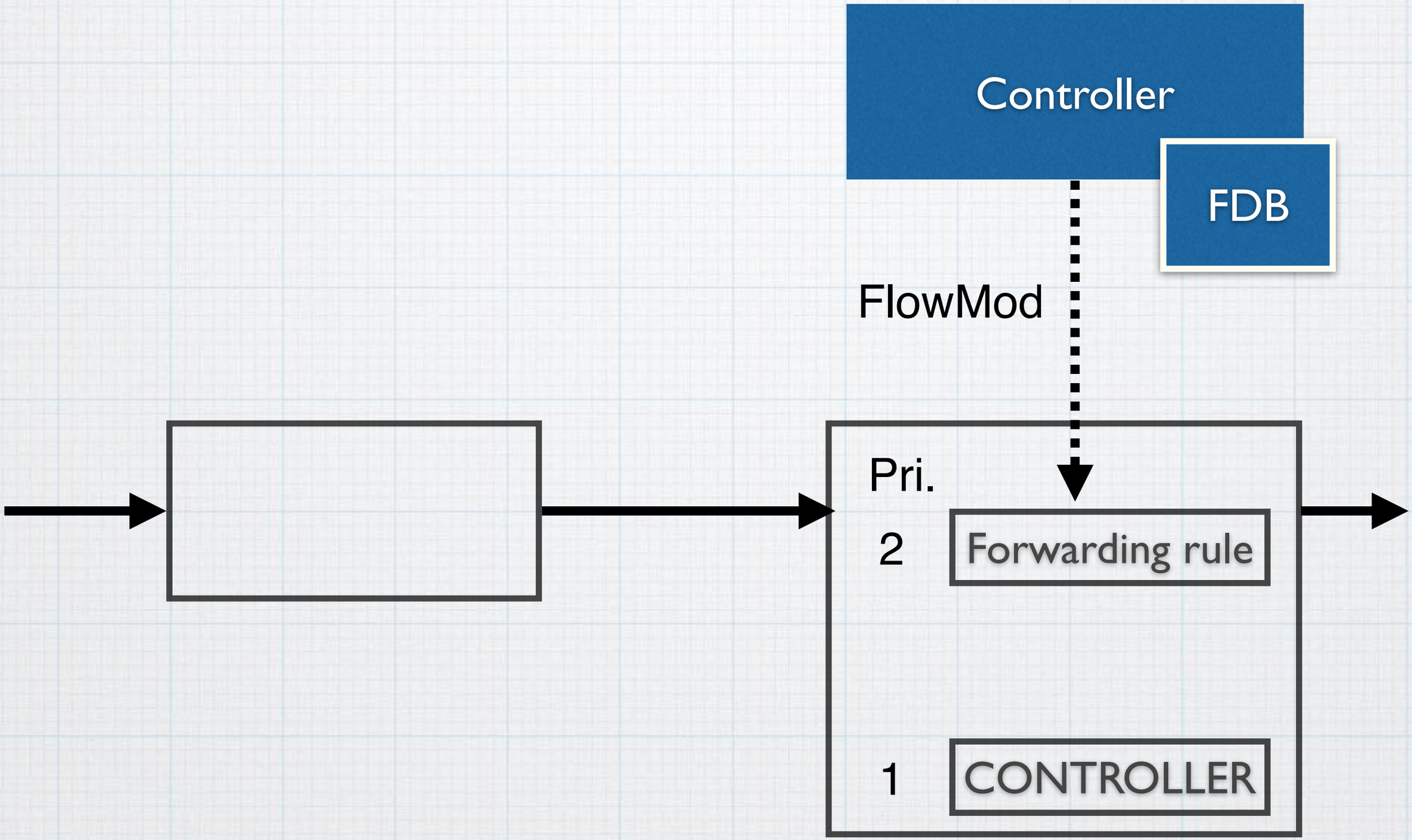
Forwarding table (ID=1)

```
send_flow_mod_add(
  datapath_id,
  table_id: 1,
  idle_timeout: 0,
  priority: 1,
  match: Match.new,
  instructions: Apply.new(SendOutPort.new(:controller))
)
```

- Invoke a PacketIn event explicitly
- Note that the default action of OpenFlow 1.3 switch/controller against unknown packets is to discard them.

Default Drop (OpenFlow 1.3)

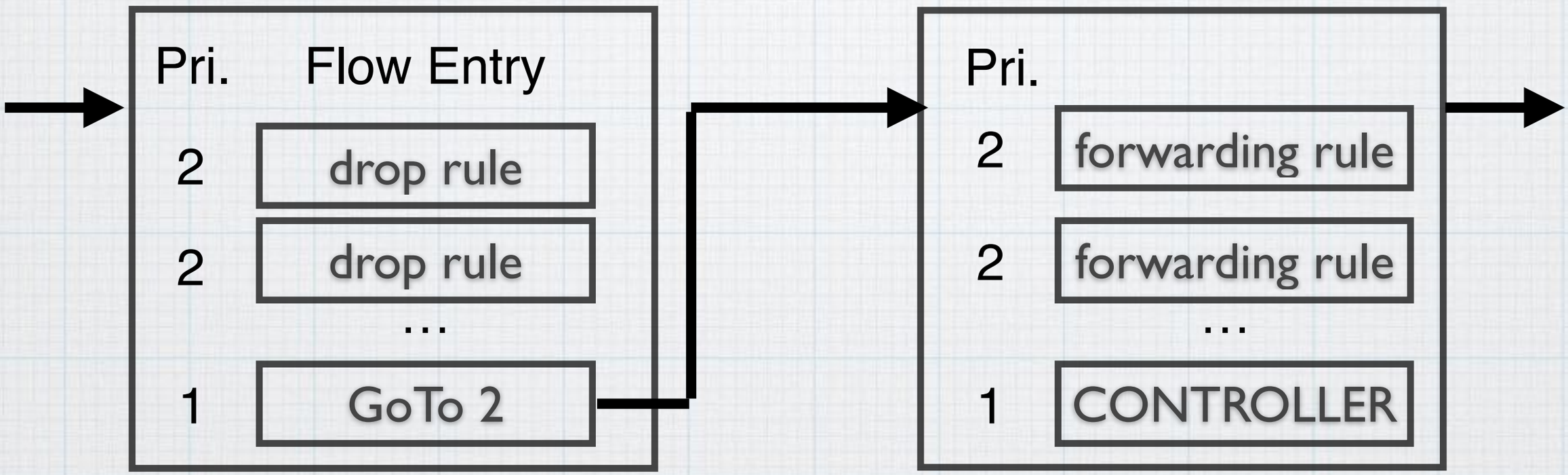
- In OpenFlow1.0, the default action for unknown packets is PacketIn
  - If a huge number of packets arrive befor creating a flow table entry for the packets, a controller might overflow.
- To prevent such a situation, the default action is changed to discarding packets, i.e., implicit PacketIn events do not occur.


Controller
FDB
FlowMod
Pri.
2
Forwarding rule
1
CONTROLLER
Filtering table (ID=0)
Forwarding table (ID=1)

```
send_flow_mod_add(
      datapath_id,
      table_id: FORWARDING_TABLE_ID,
      idle_timeout: AGING_TIME,
      priority: 2,
      match: Match.new(in_port: packet_in.in_port,
                       ether_destination_address: packet_in.destination_mac,
                       ether_source_address: packet_in.source_mac),
    instructions: Apply.new(SendOutPort.new(port_no))
  )
```

- Add a flow entry with SRC/DST MAC addresses and an incoming port recorded in the packet_in object
- Set the higher priority than PacketIn rule

Controller



Filtering table (ID=0)

Forwarding table (ID=1)

Conclusion

Developing a Learning Switch/Controller, using OpenFlow 1.3

- Using multiple tables
- Transit between tables with GotoTable instruction
- The default action is now packet drop to prevent unintentional PacketIn events