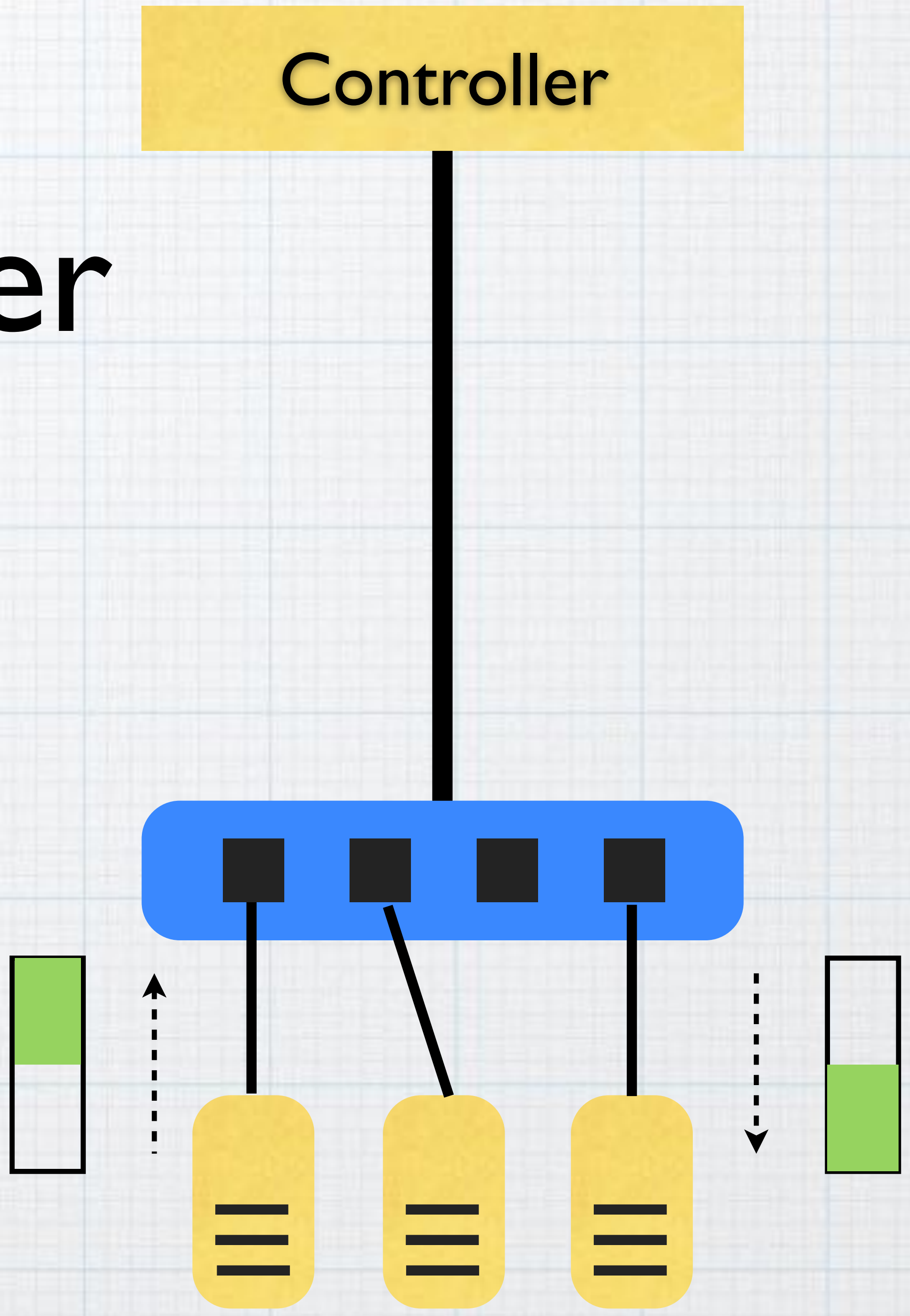
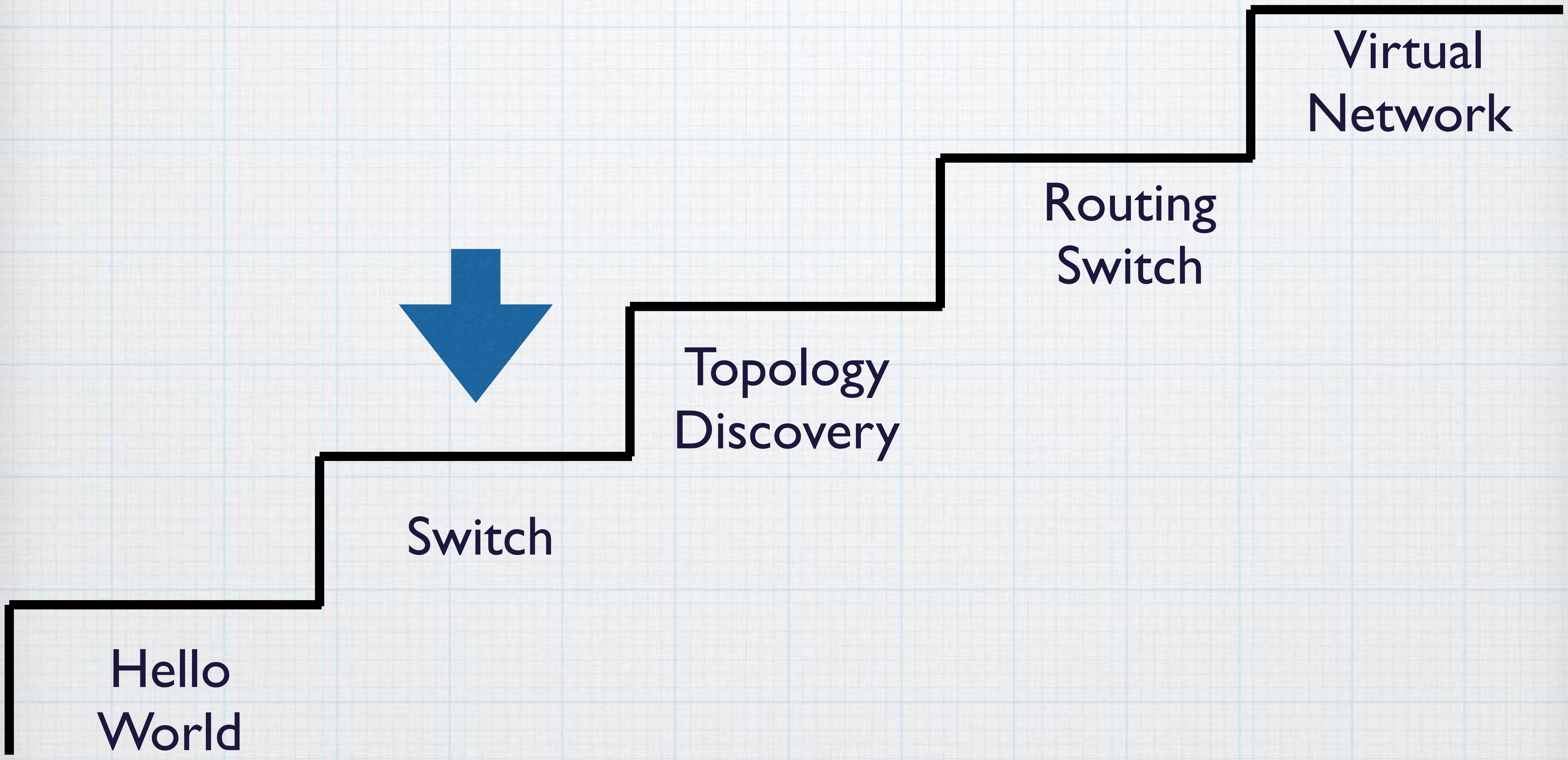
Developing a Learning Switch and a Controller

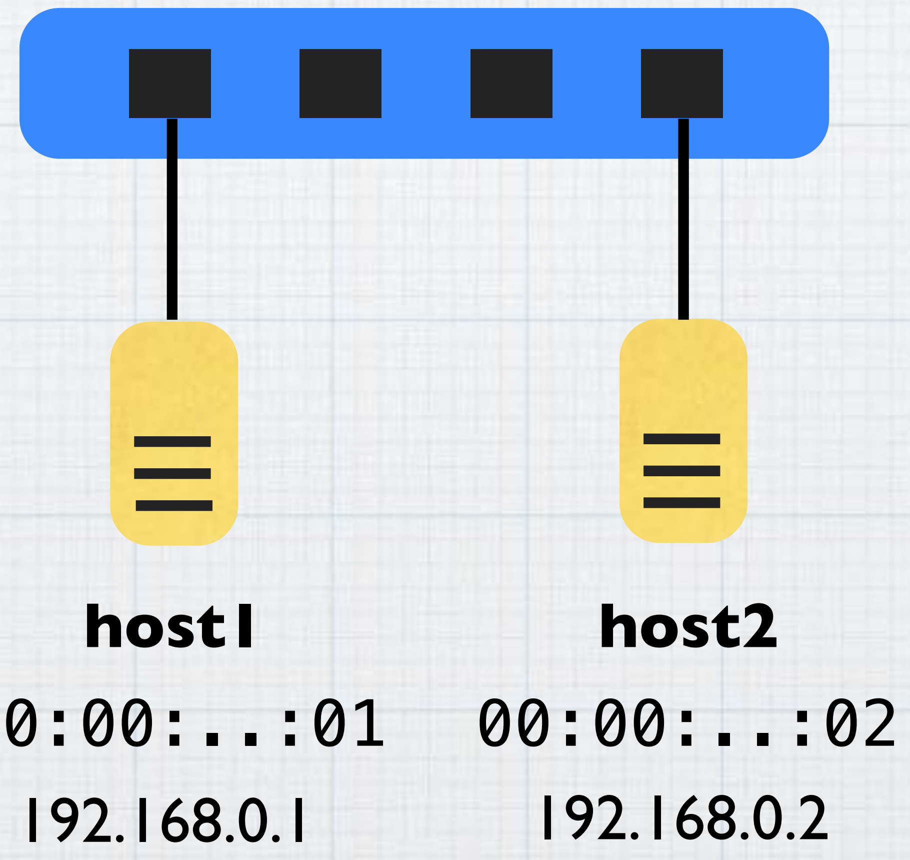



Hello World
Switch
Topology Discovery
Routing Switch
Virtual Network

A Mechanism of a General Purpose Switch

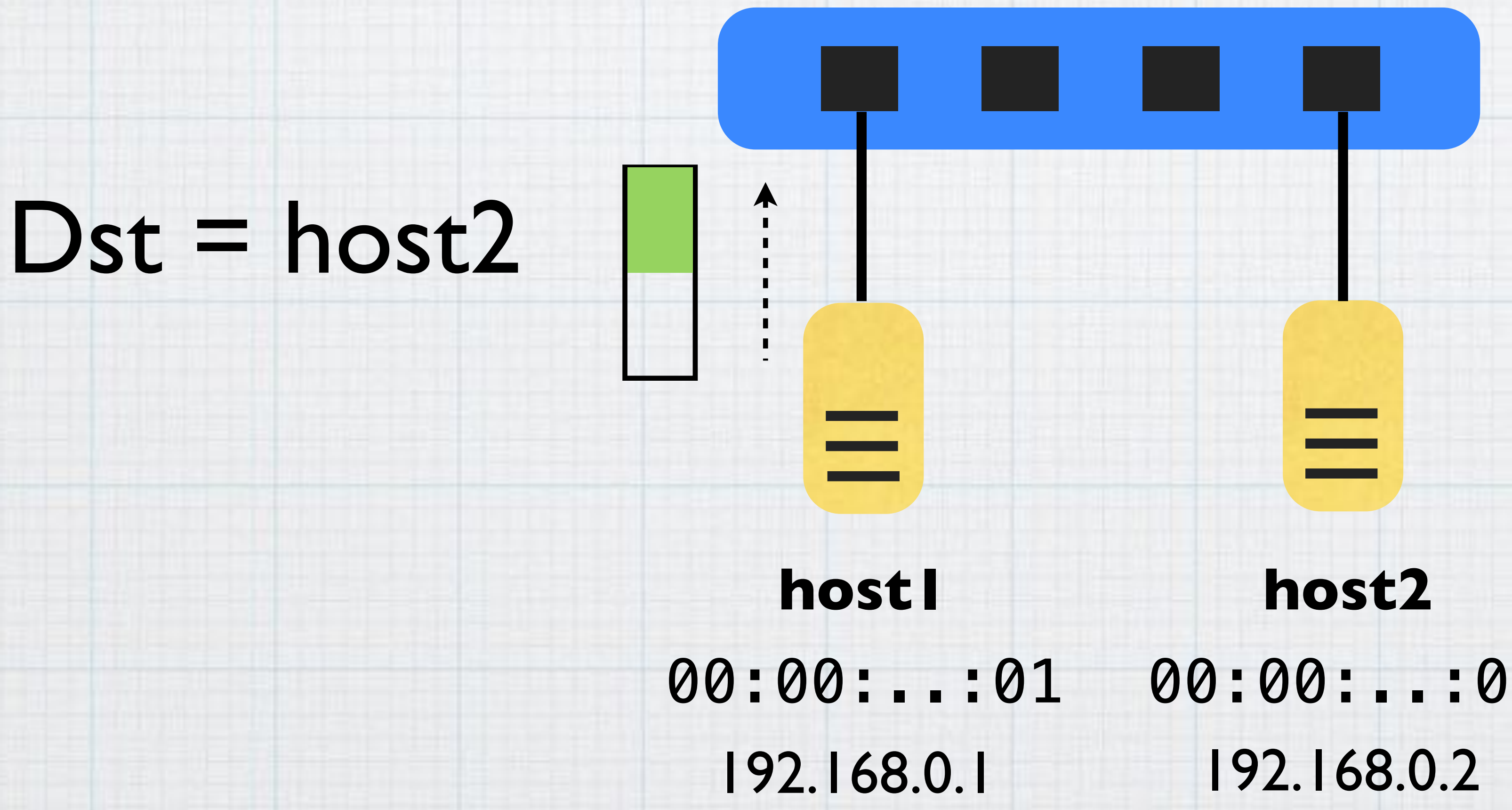
Forwarding DB

00:00:00:00:00:01	→	1
00:00:00:00:00:02	→	4



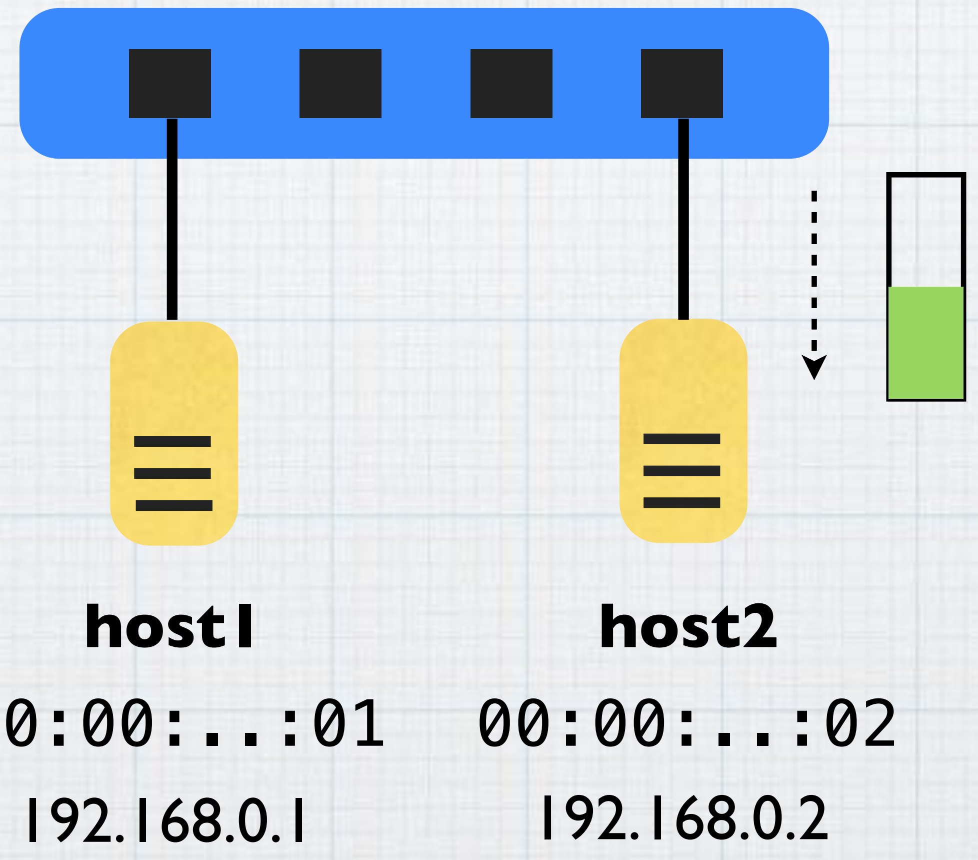
Forwarding DB

00:00:00:00:00:01 → 1
00:00:00:00:00:02 → 4

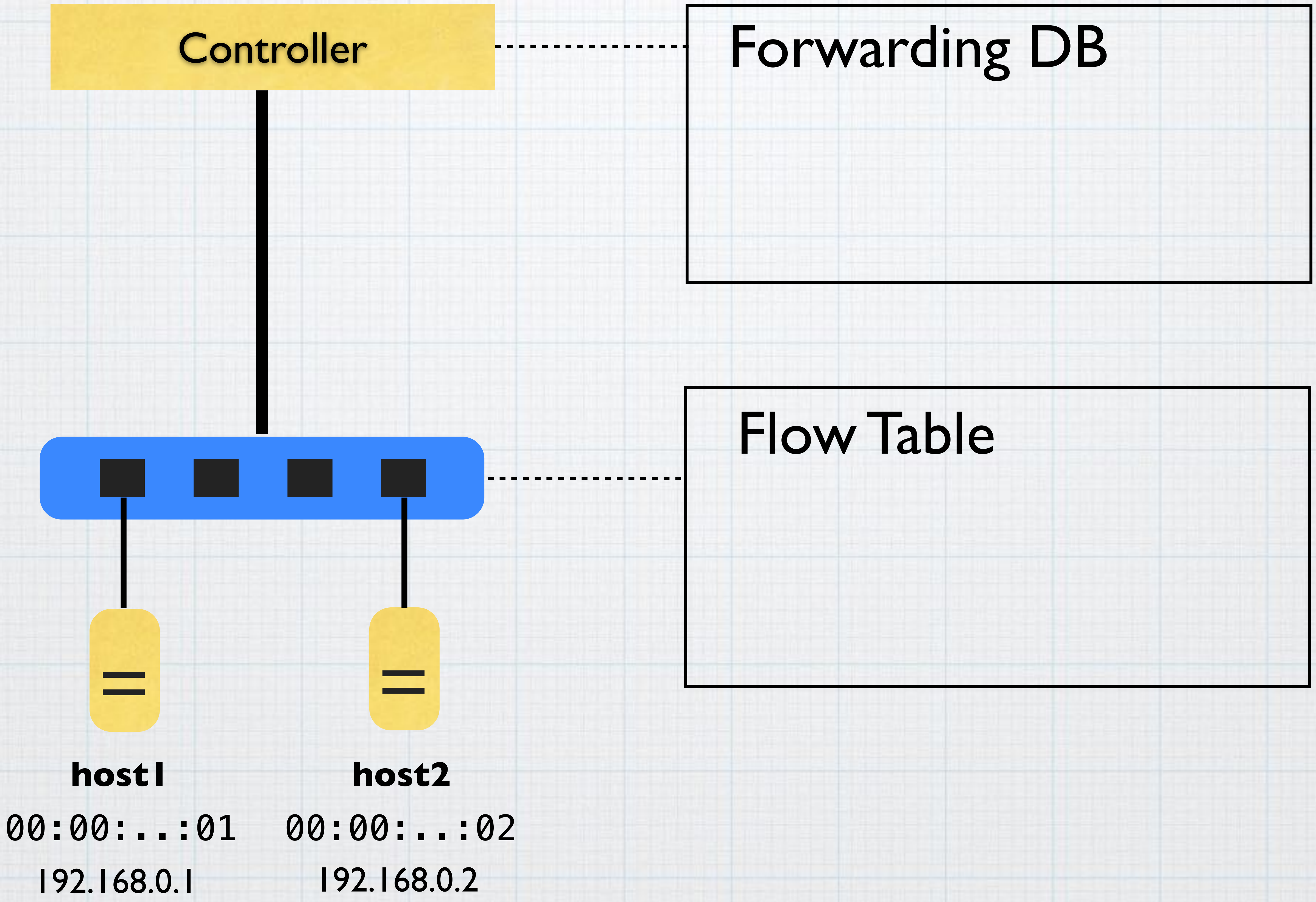


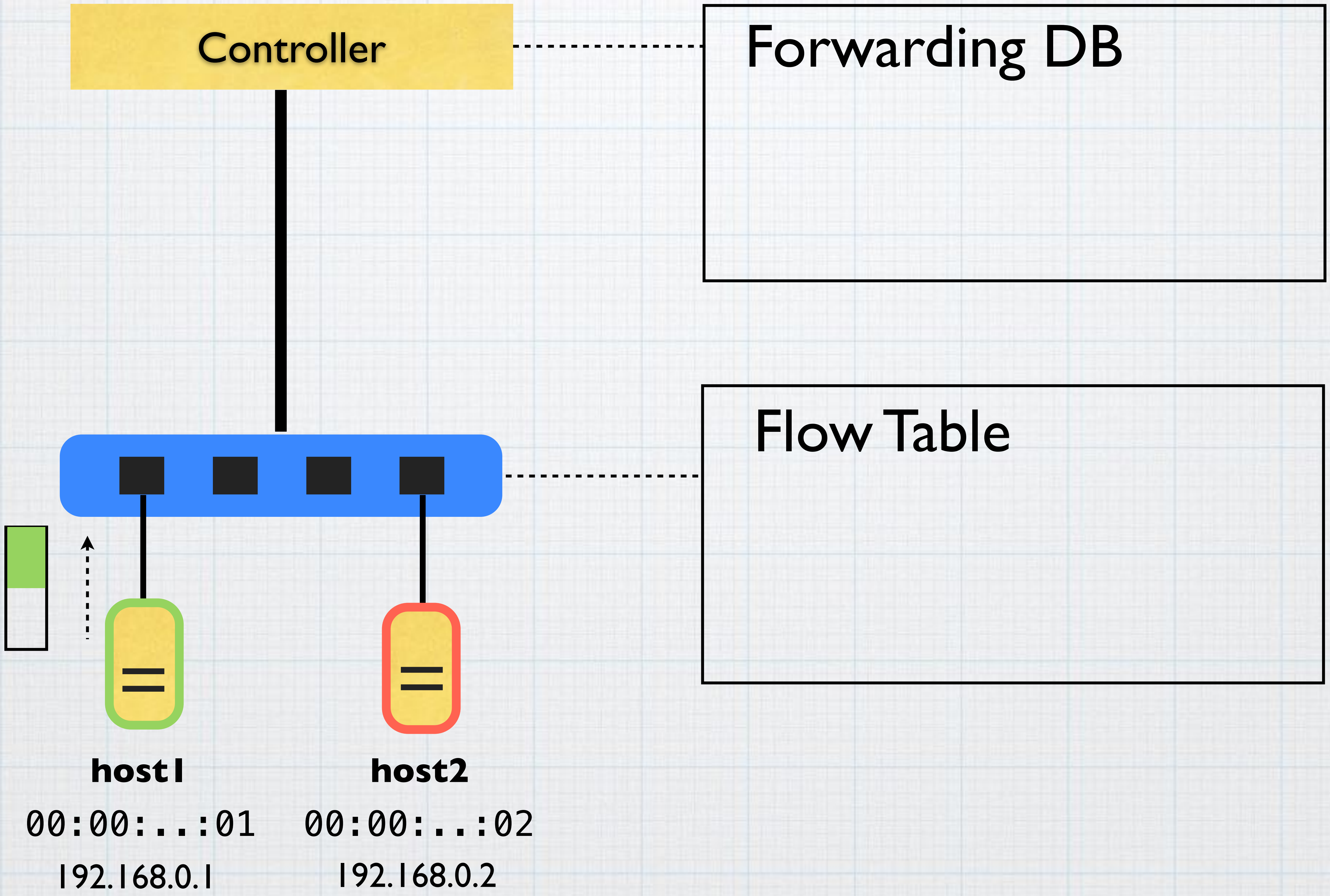
Forwarding DB

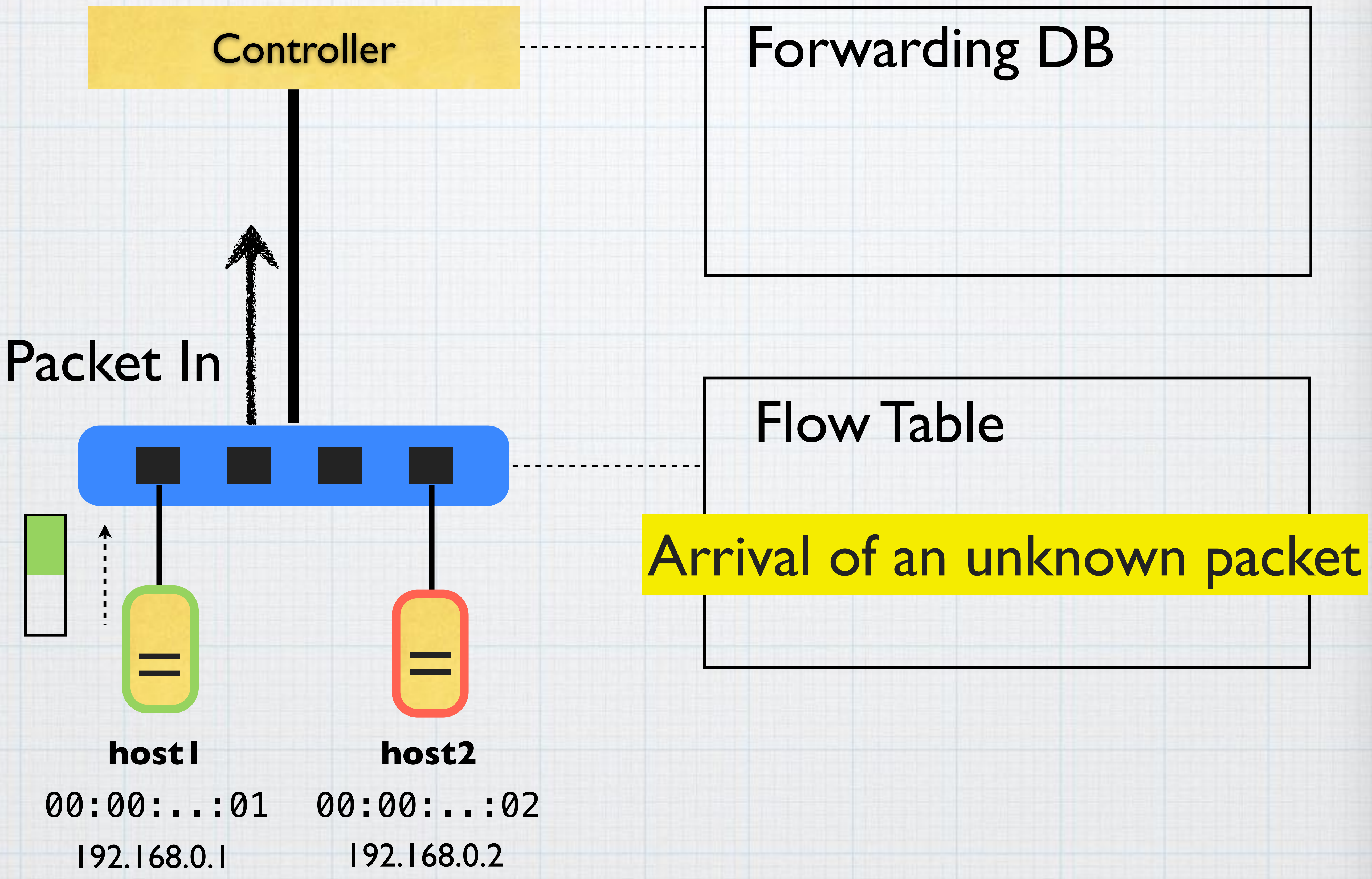
00:00:00:00:00:01 → 1
00:00:00:00:00:02 → 4

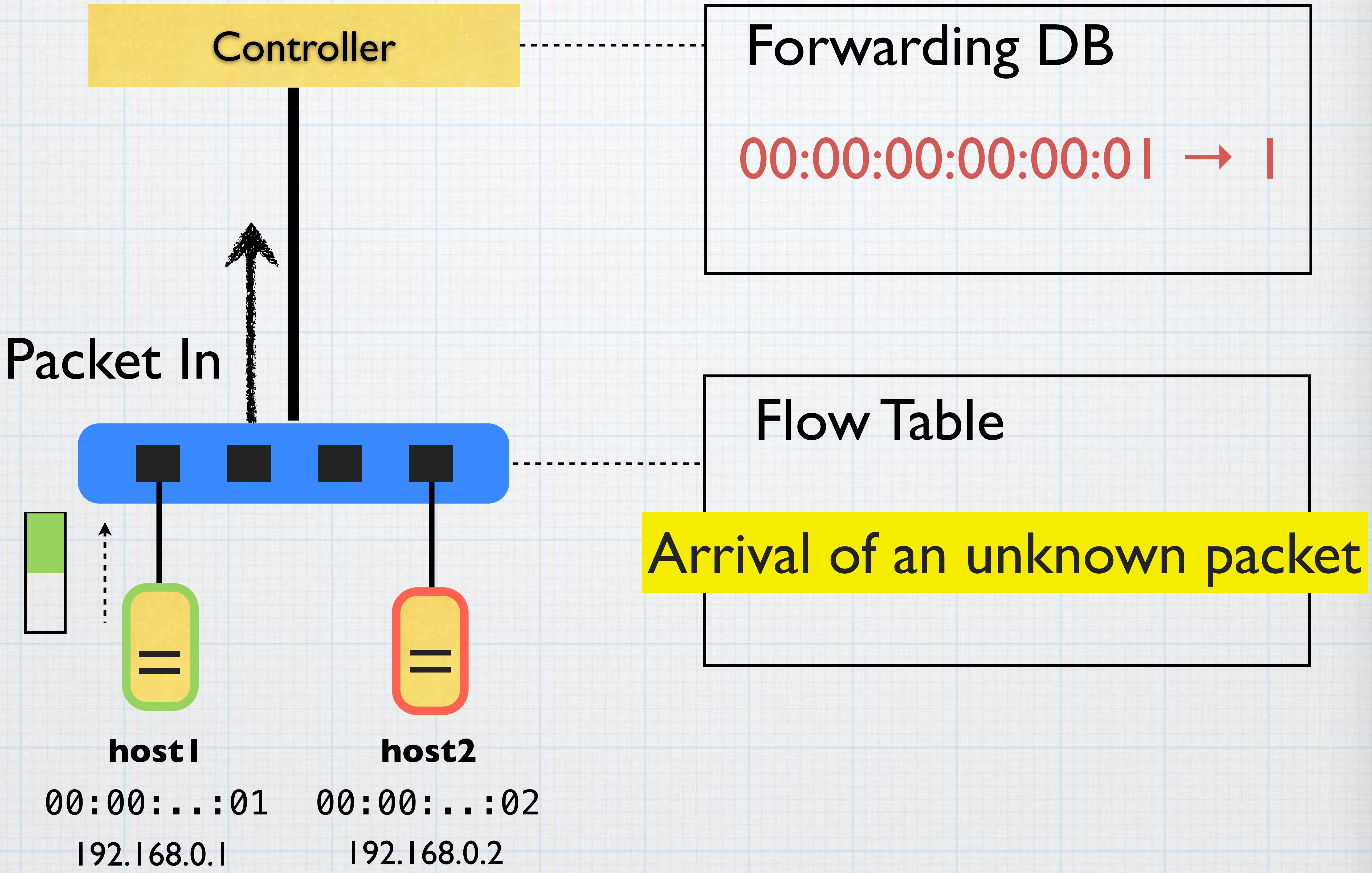


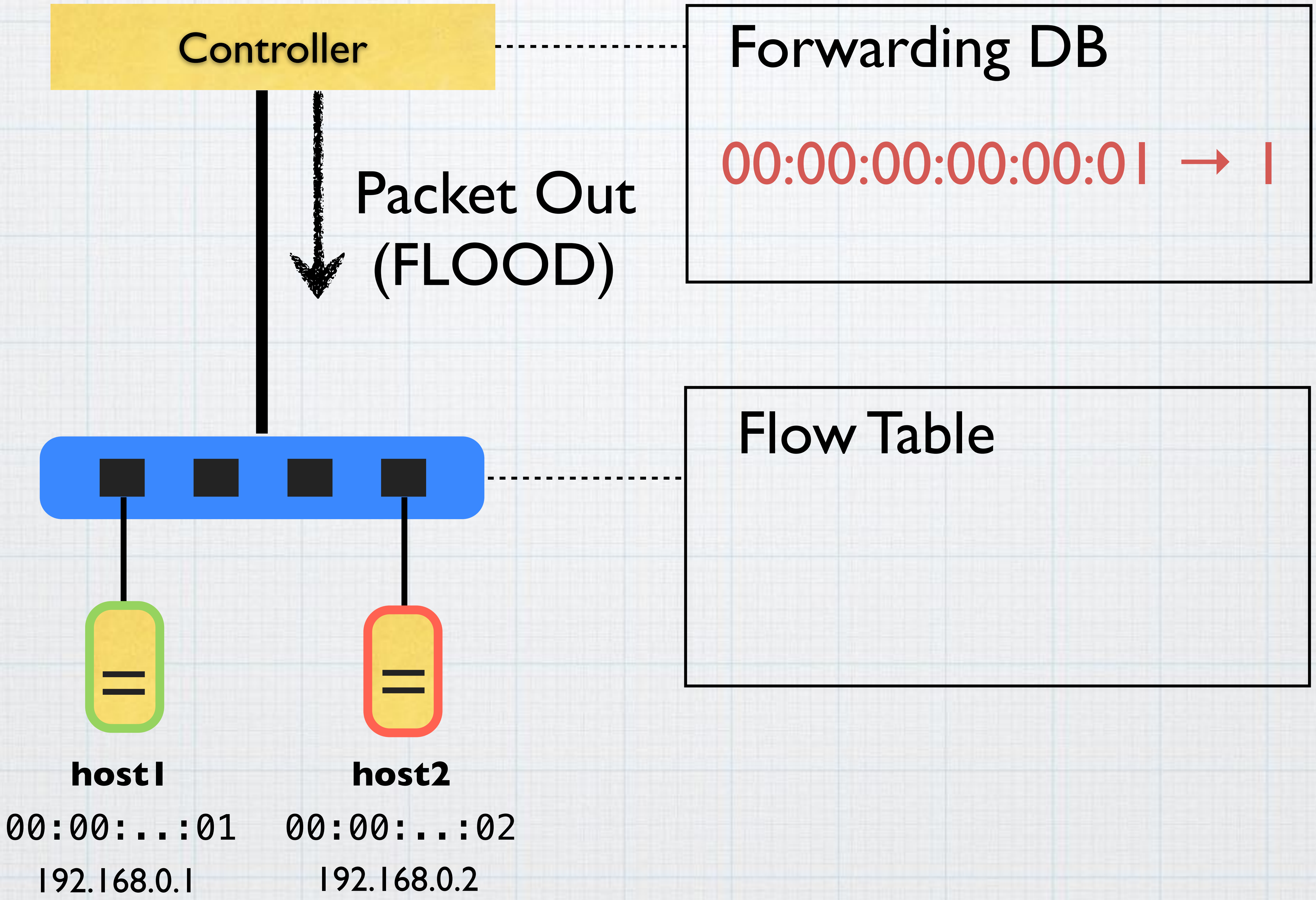
A Mechanism of a Switch based on OpenFlow

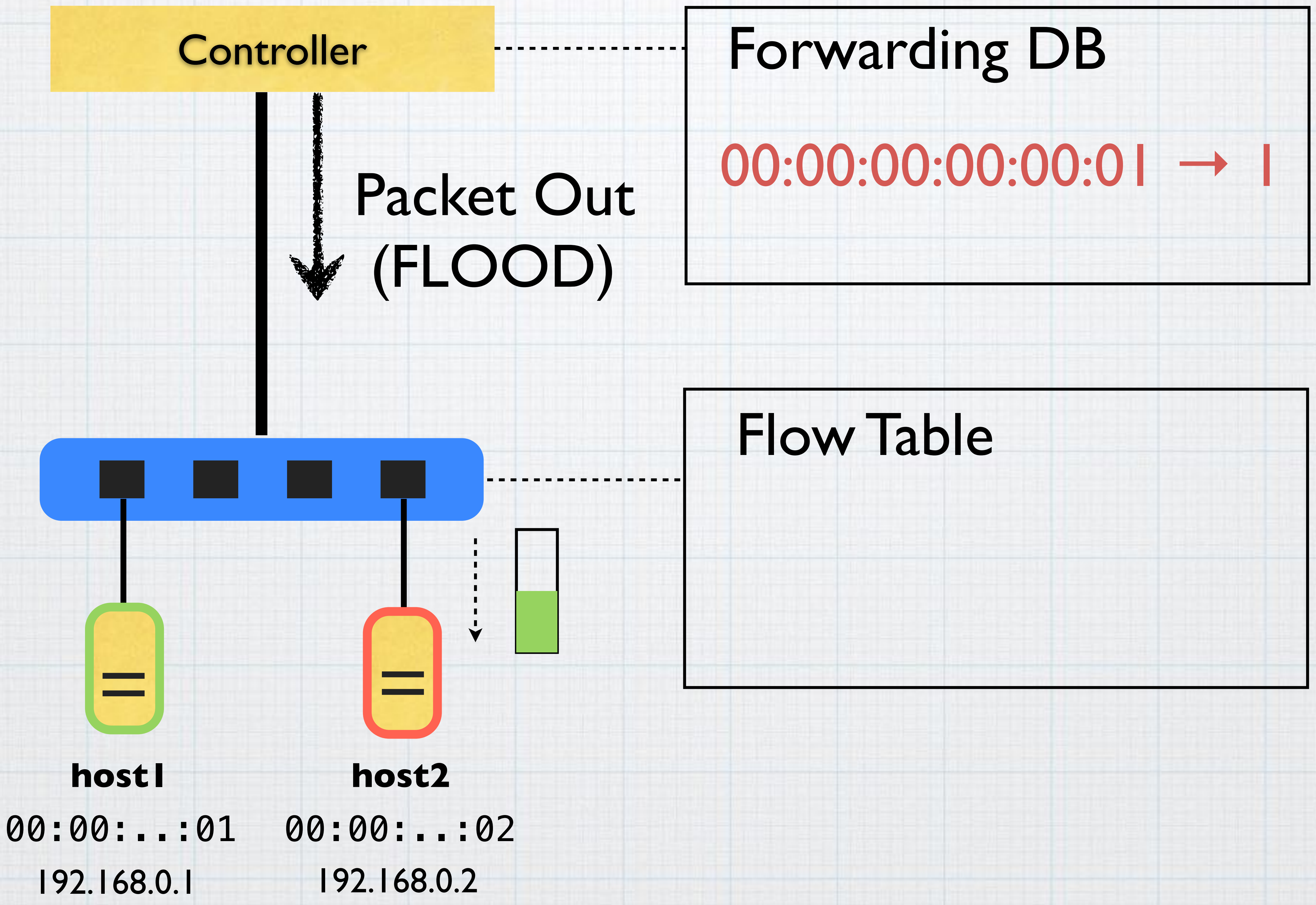

Controller
Forwarding DB
Flow Table
host1
00:00:...:01
192.168.0.1
host2
00:00:...:02
192.168.0.2

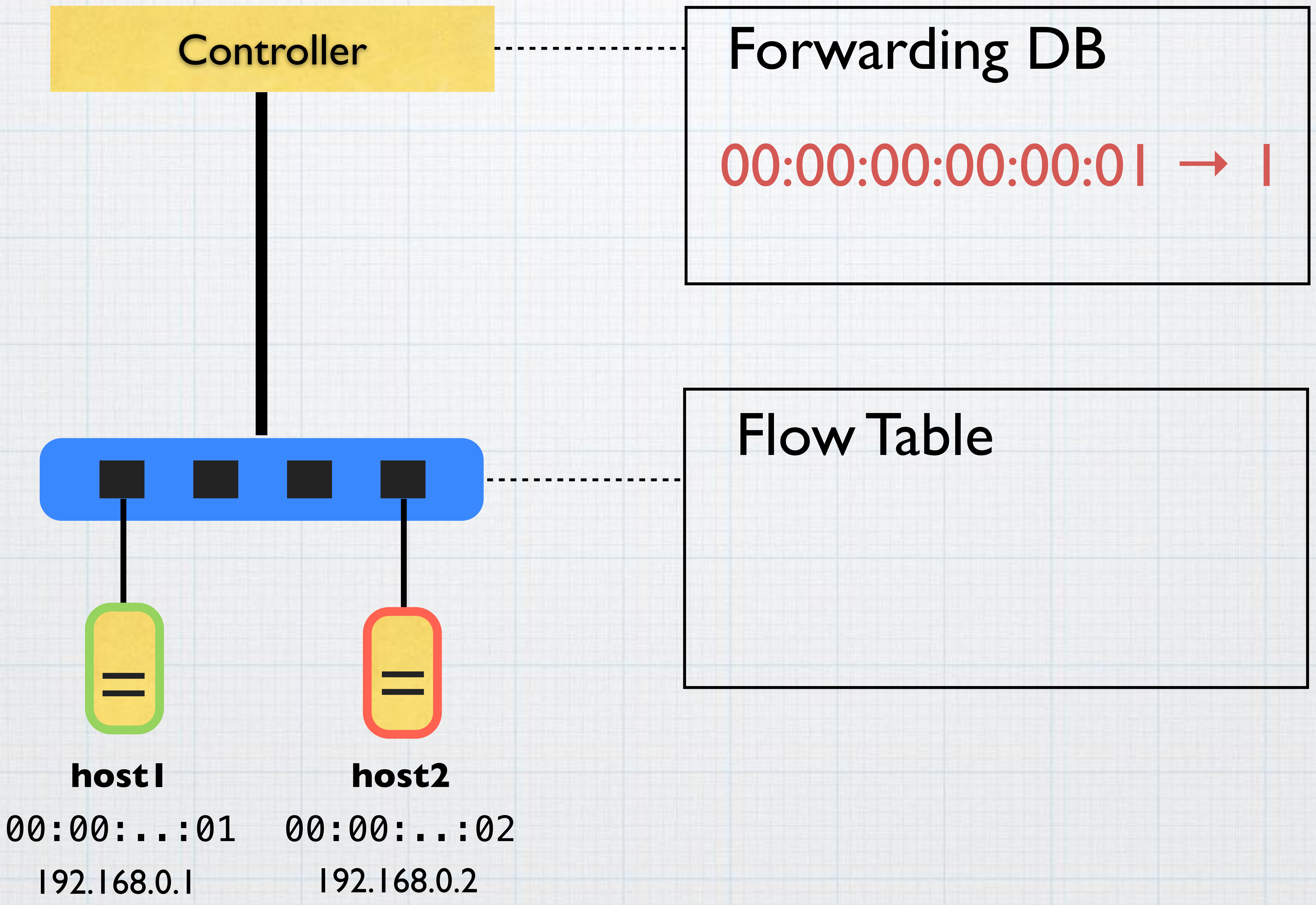

Controller
Forwarding DB
Flow Table
host1
00:00:...:01
192.168.0.1
host2
00:00:...:02
192.168.0.2

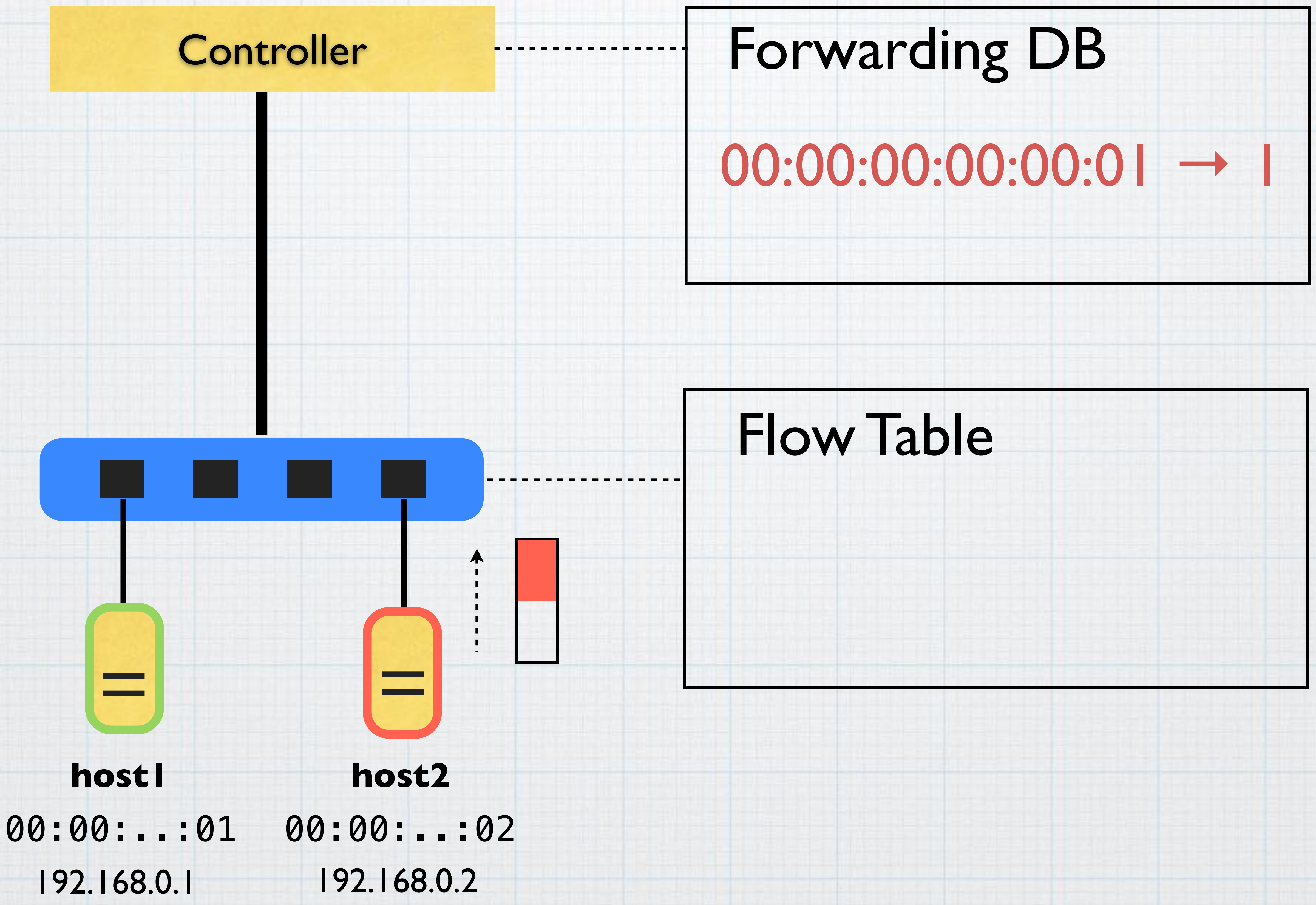

Controller
Forwarding DB
Packet In
Flow Table
Arrival of an unknown packet
host1
00:00:...:01
192.168.0.1
host2
00:00:...:02
192.168.0.2

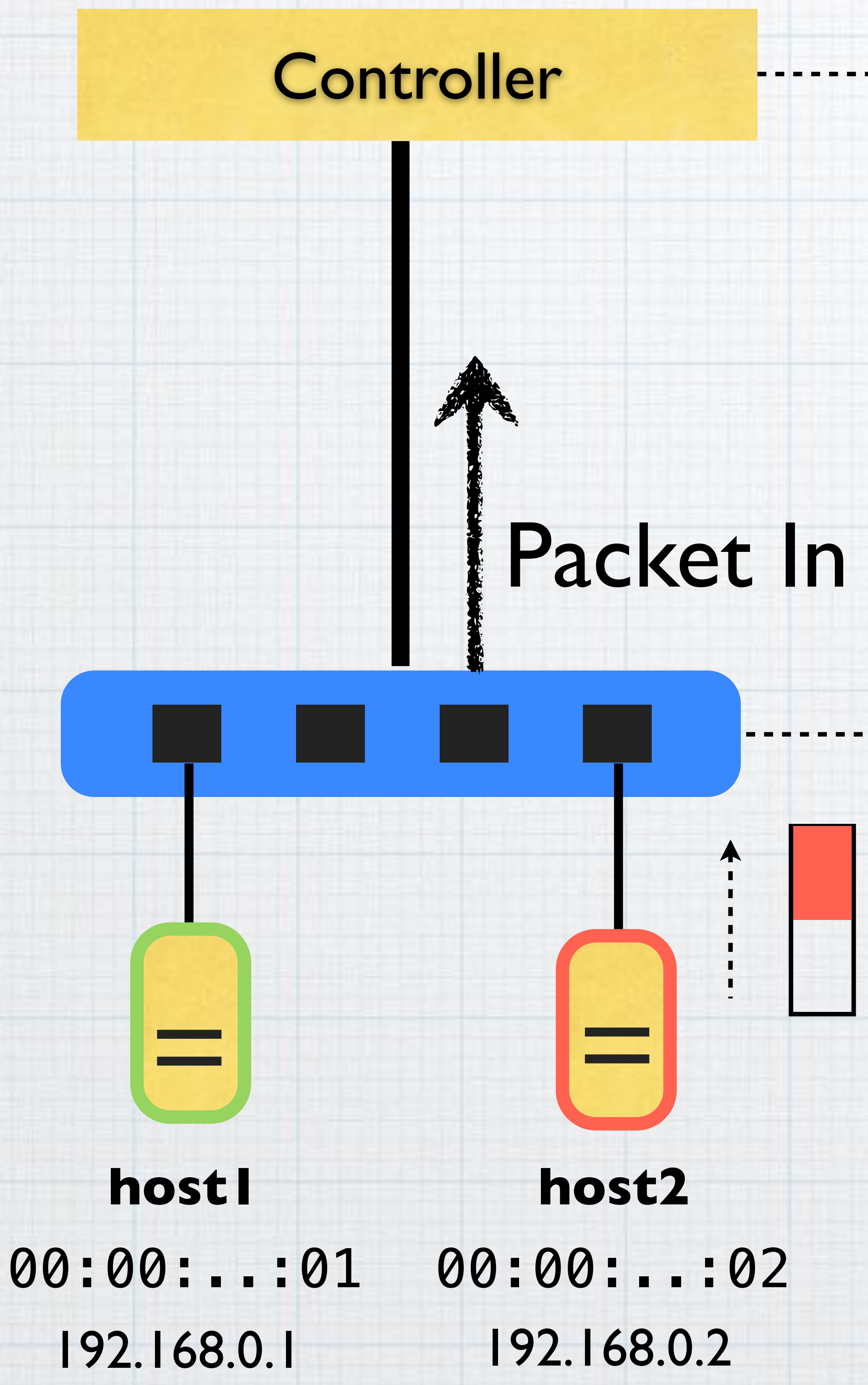

Controller
Forwarding DB
00:00:00:00:00:01 → 1
Packet In
Flow Table
Arrival of an unknown packet
host1
00:00:...:01
192.168.0.1
host2
00:00:...:02
192.168.0.2


Controller
Packet Out
(FLOOD)
Forwarding DB
00:00:00:00:00:01 → 1
Flow Table
host1
00:00:...:01
192.168.0.1
host2
00:00:...:02
192.168.0.2


Controller
Packet Out
(FLOOD)
Forwarding DB
00:00:00:00:00:01 → 1
Flow Table
host1
00:00:...:01
192.168.0.1
host2
00:00:...:02
192.168.0.2


Controller
Forwarding DB
00:00:00:00:00:01 → 1
Flow Table
host1
00:00:...:01
192.168.0.1
host2
00:00:...:02
192.168.0.2


Controller
Forwarding DB
00:00:00:00:00:01 → 1
Flow Table
host1
00:00:...:01
192.168.0.1
host2
00:00:...:02
192.168.0.2

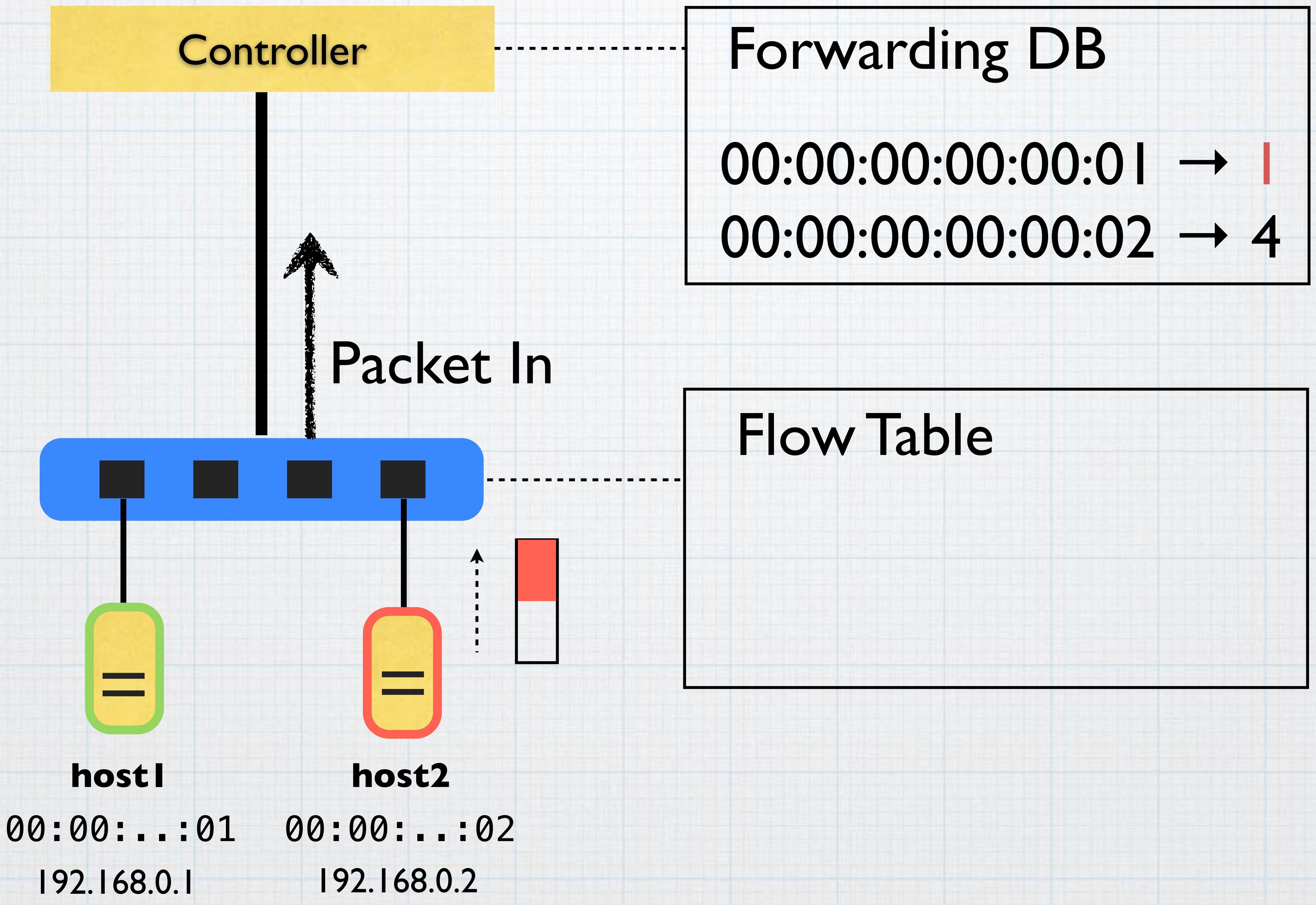


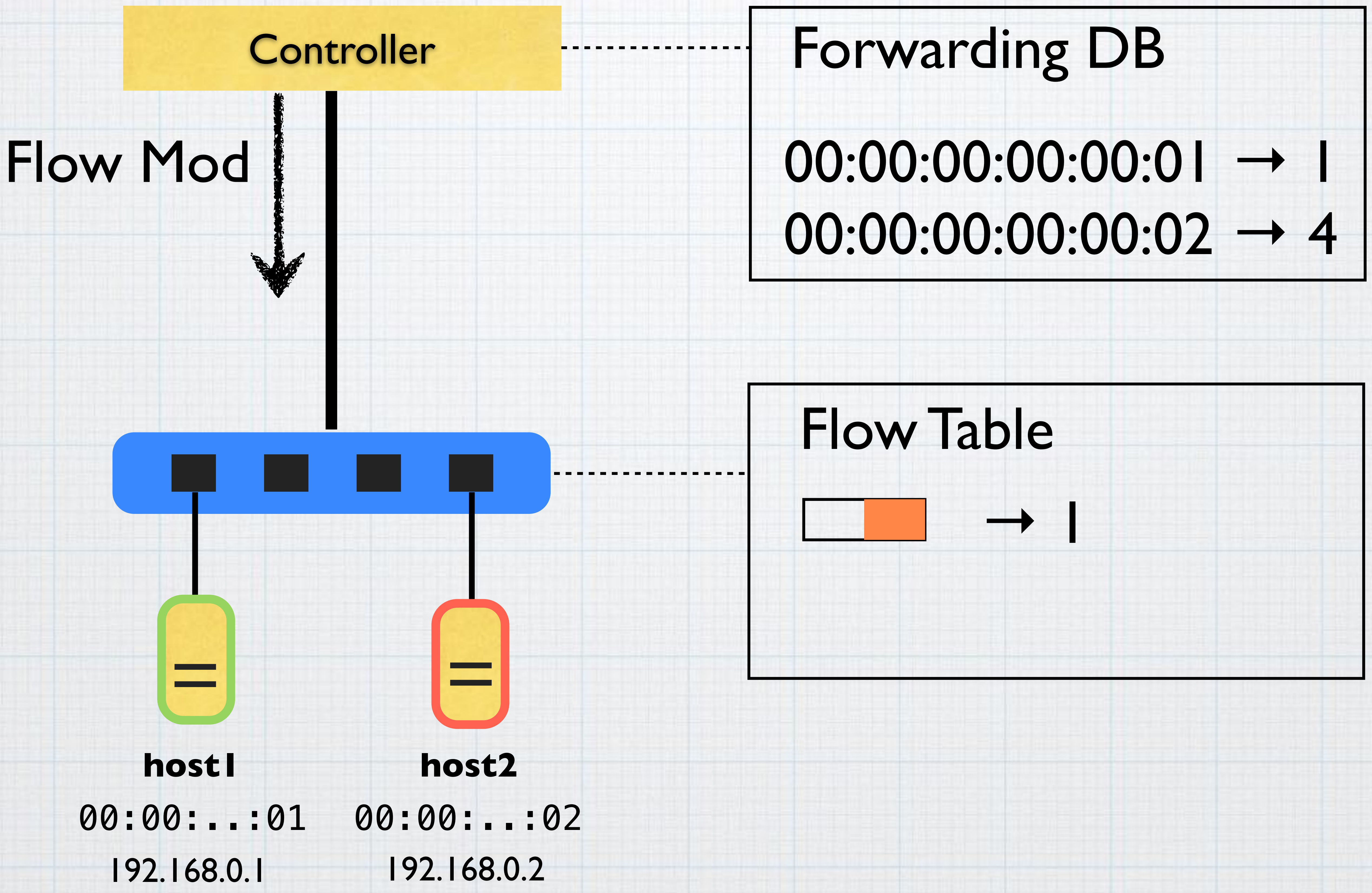
Forwarding DB

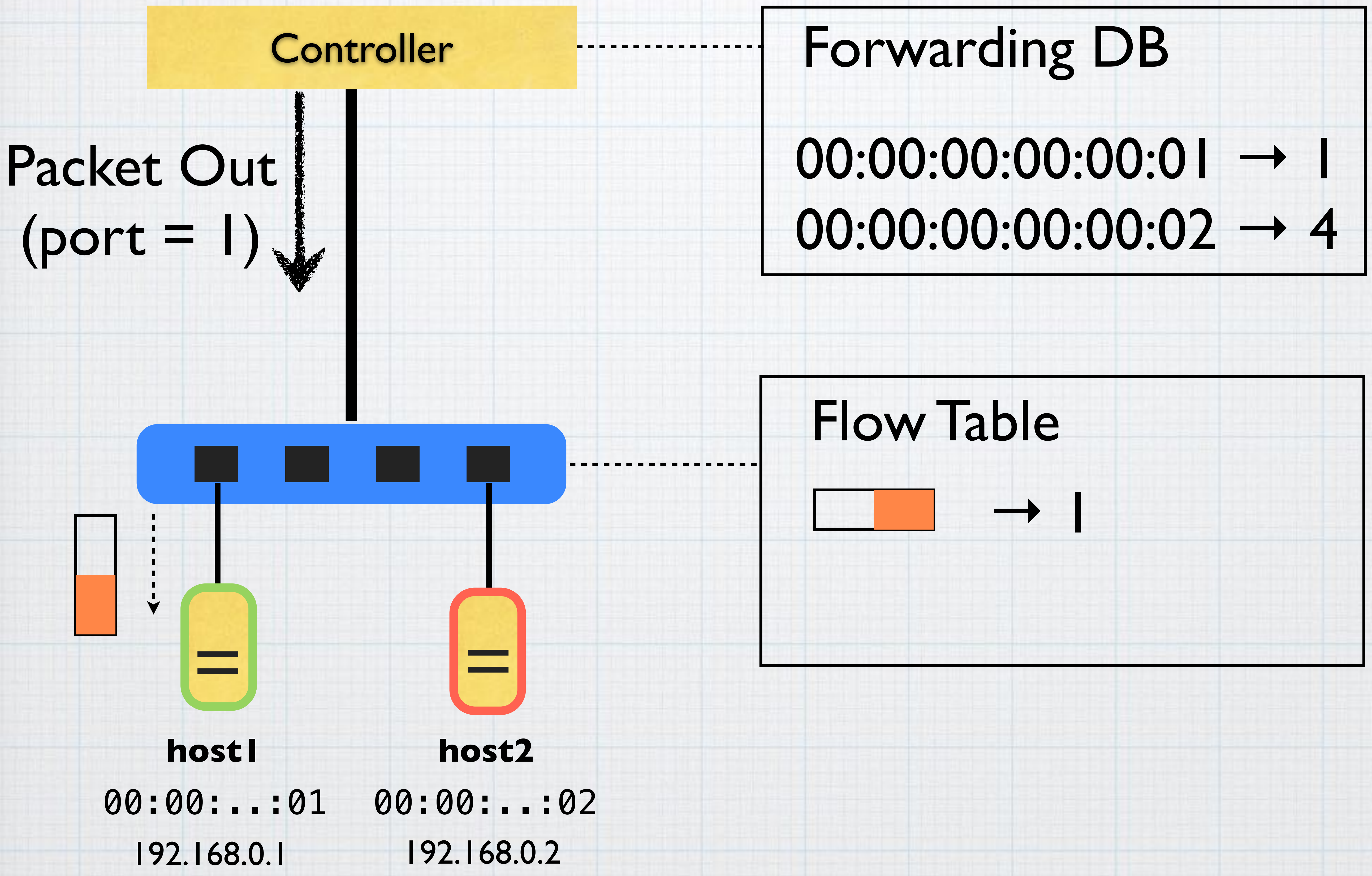
00:00:00:00:00:01 →	1
00:00:00:00:00:02 →	4

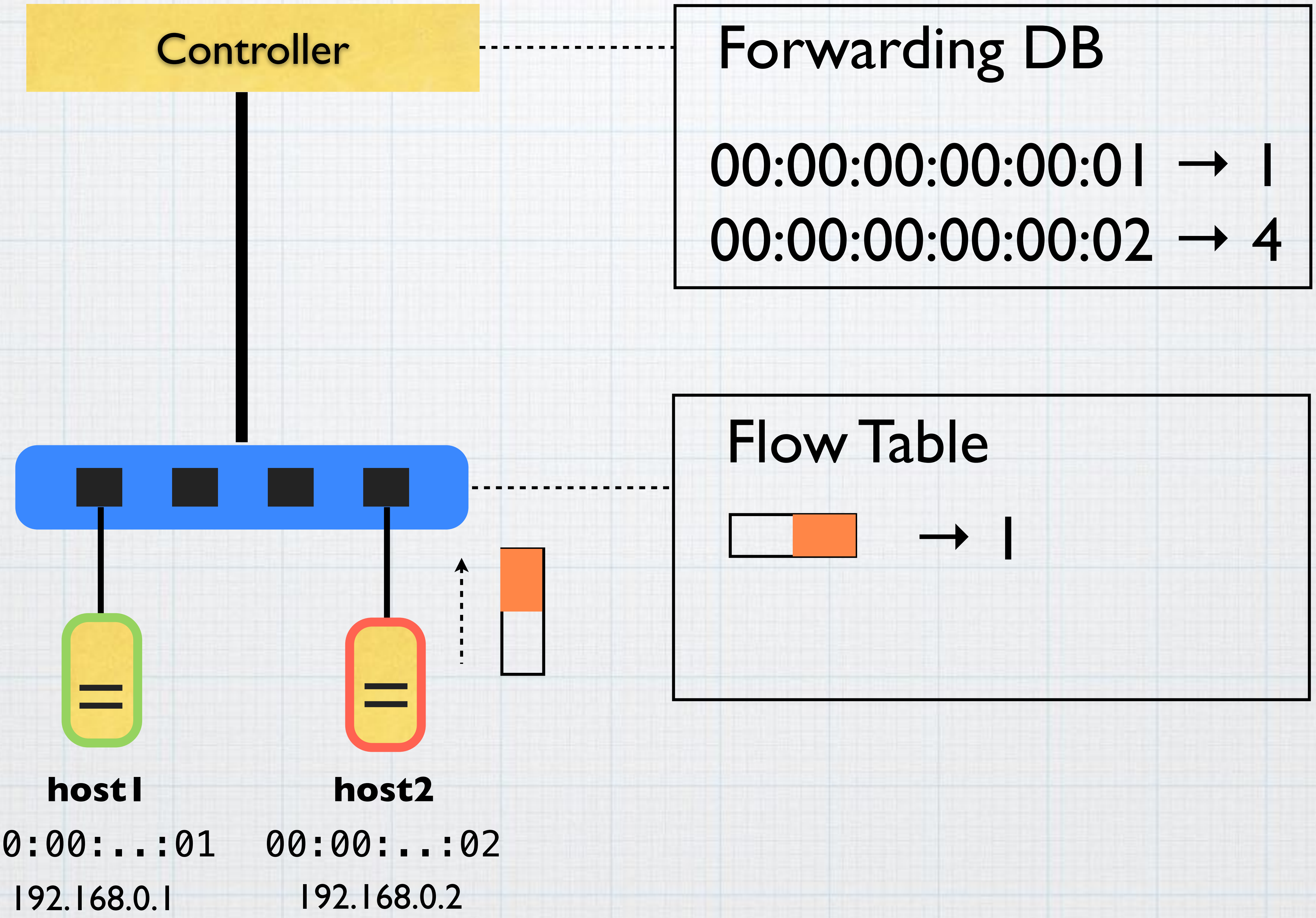
Flow Table

Arrival of an unknown packet

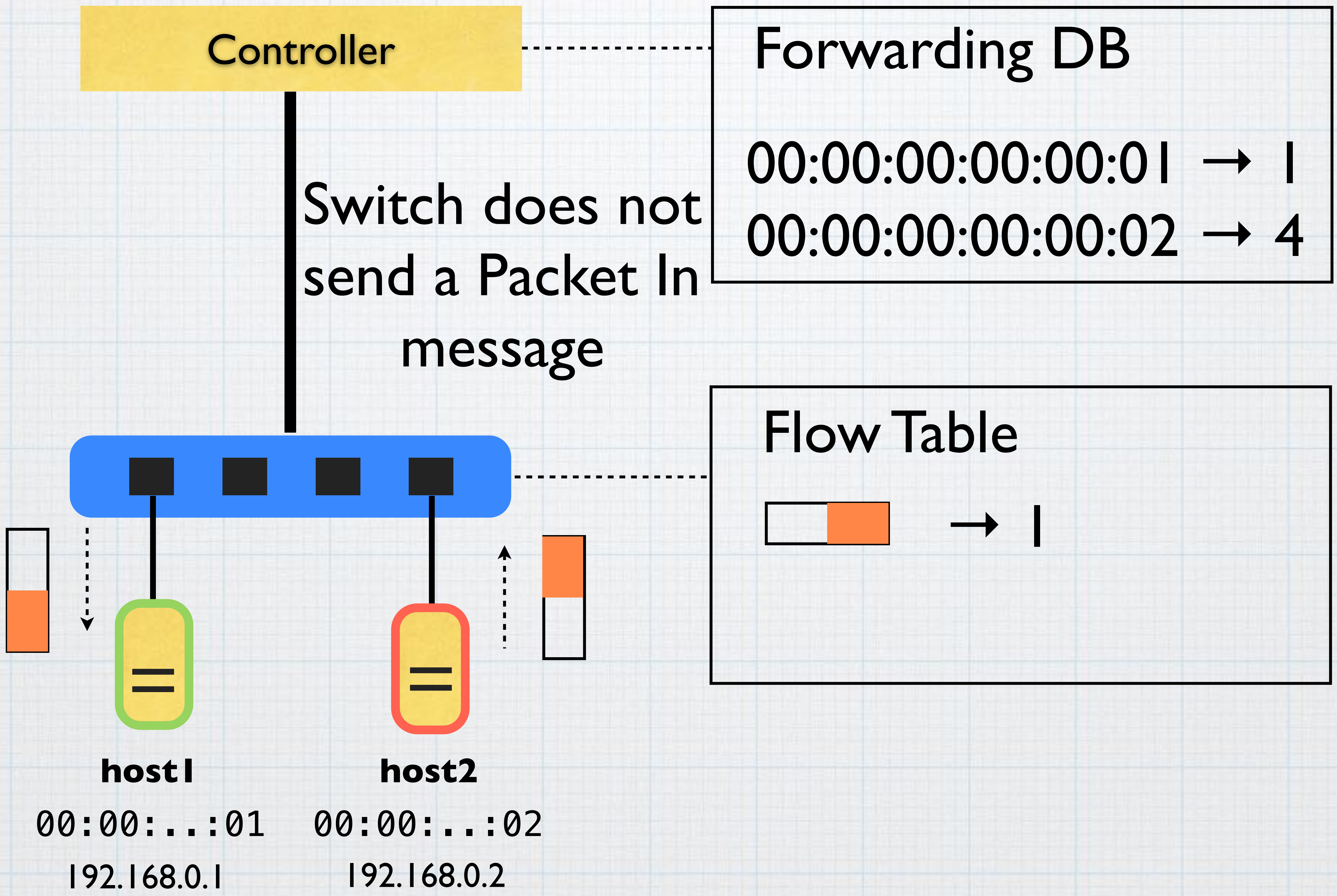

Controller
Forwarding DB
00:00:00:00:00:01 → 1
00:00:00:00:00:02 → 4
Packet In
Flow Table
host1
00:00:...:01
192.168.0.1
host2
00:00:...:02
192.168.0.2

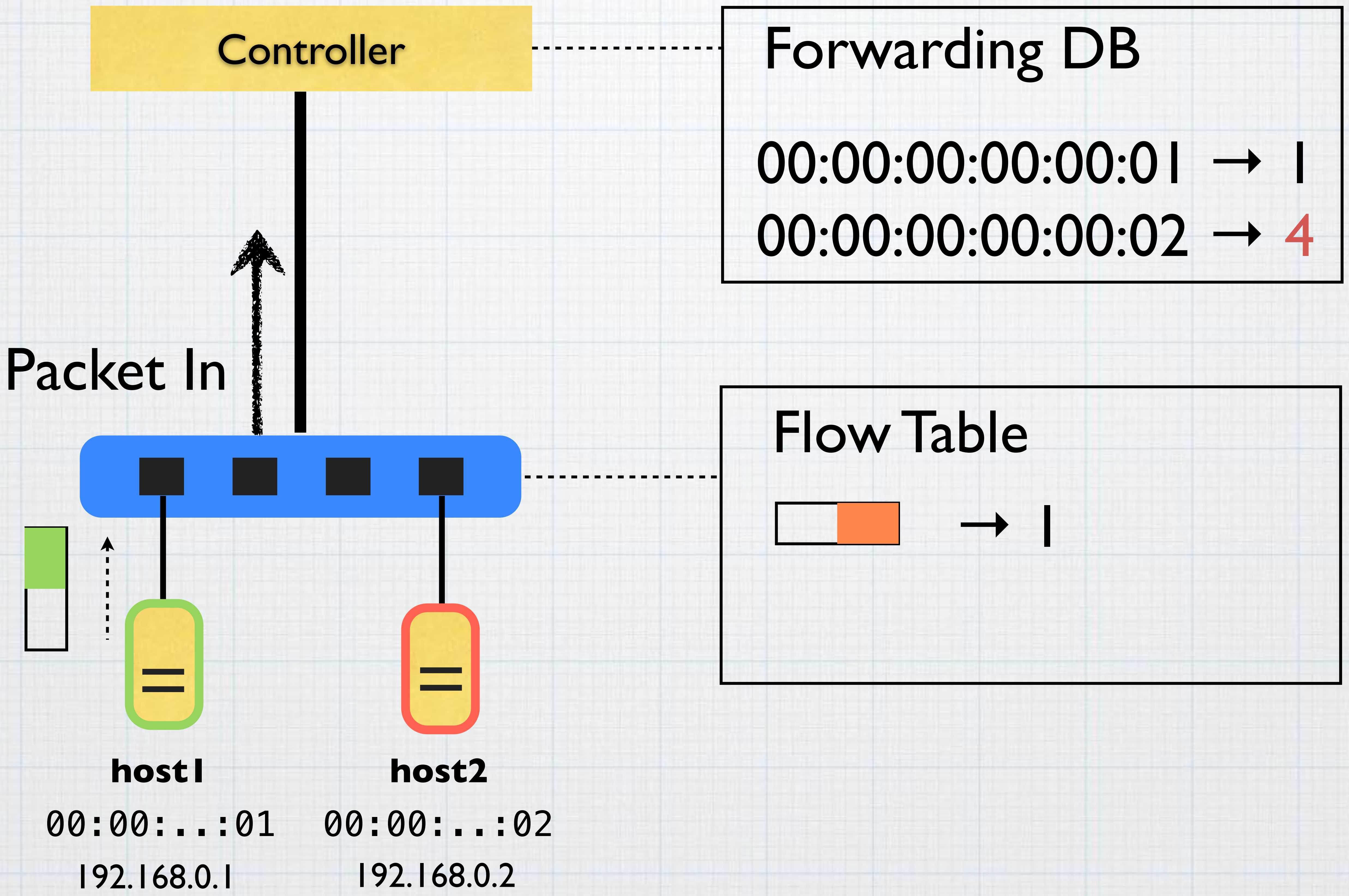

Controller
Flow Mod
Forwarding DB
00:00:00:00:00:01 → 1
00:00:00:00:00:02 → 4
Flow Table
→ 1
host1
00:00:...:01
192.168.0.1
host2
00:00:...:02
192.168.0.2

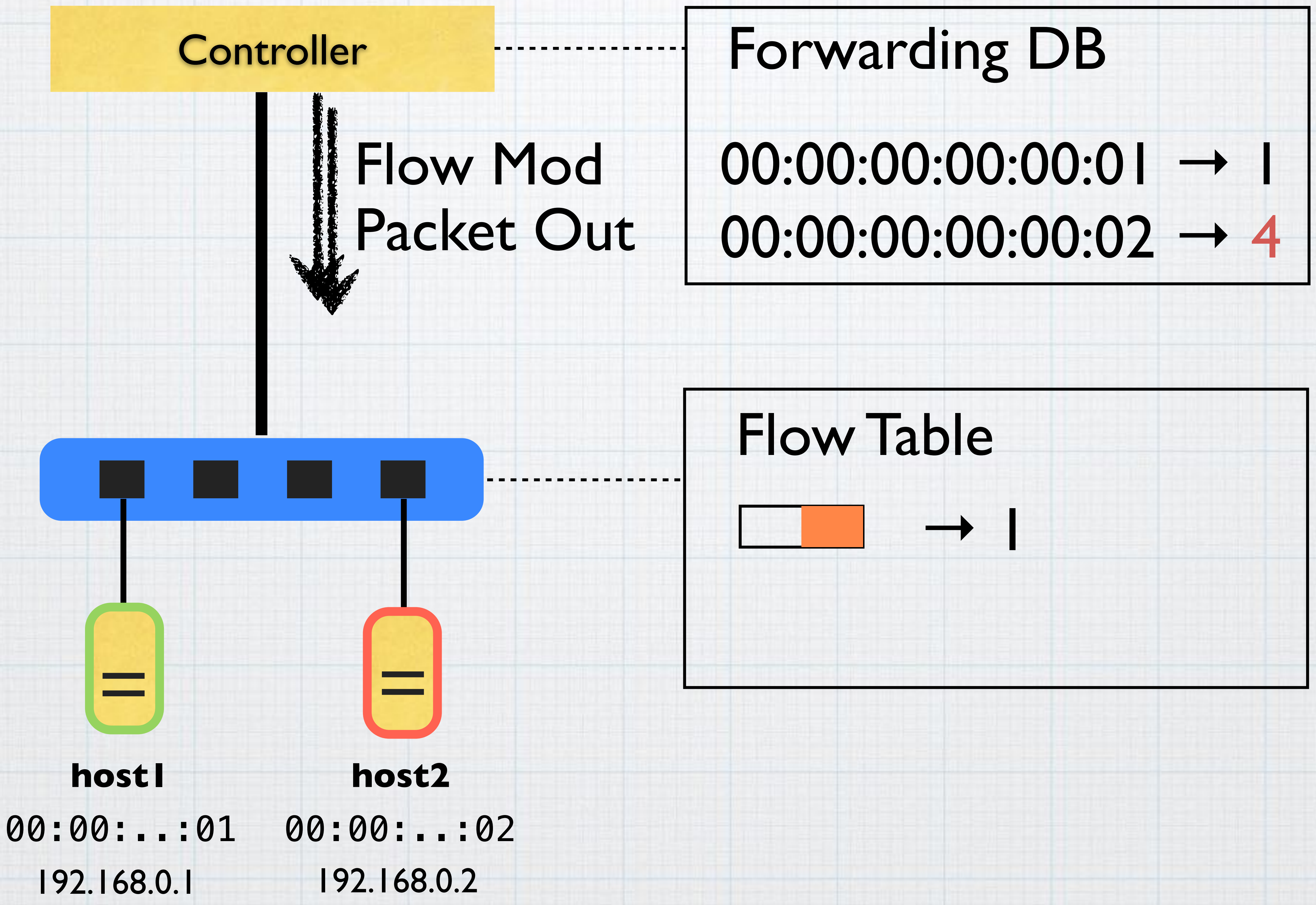

Controller
Packet Out
(port = 1)
Forwarding DB
00:00:00:00:00:01 → 1
00:00:00:00:00:02 → 4
Flow Table
→ 1
host1
00:00:...:01
192.168.0.1
host2
00:00:...:02
192.168.0.2

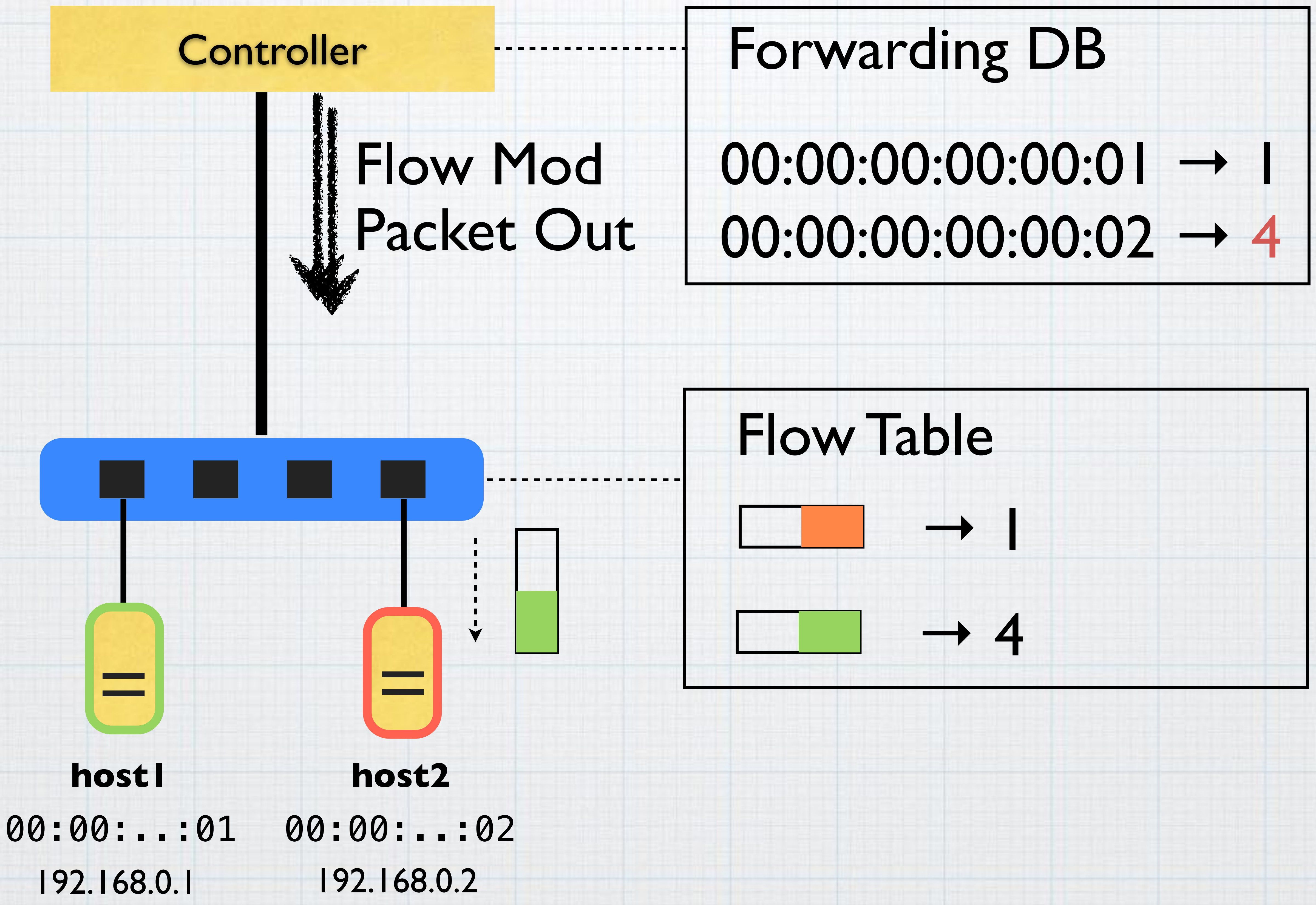


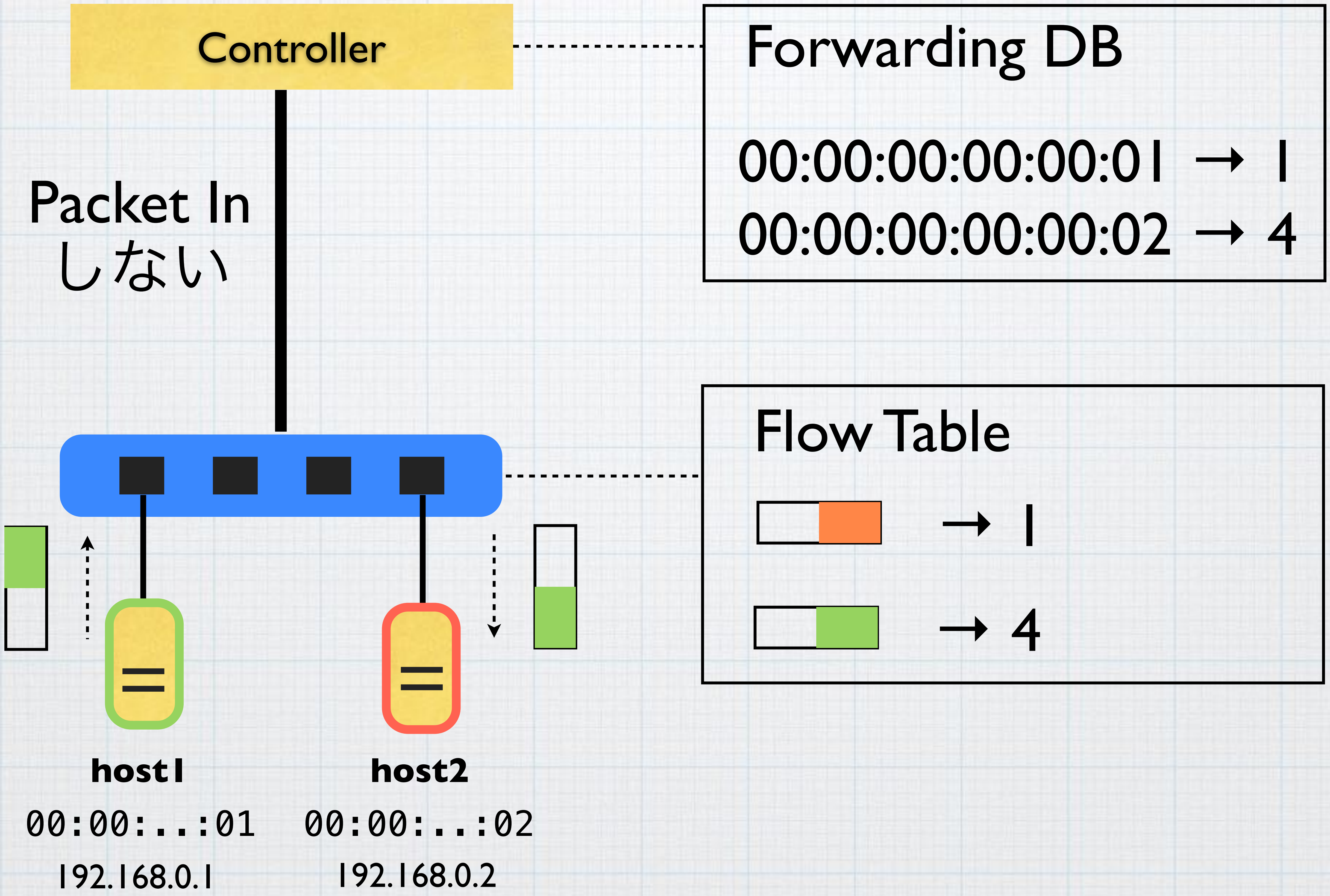
Controller
Forwarding DB
00:00:00:00:00:01 → 1
00:00:00:00:00:02 → 4
Flow Table
→ 1
host1
00:00:...:01
192.168.0.1
host2
00:00:...:02
192.168.0.2

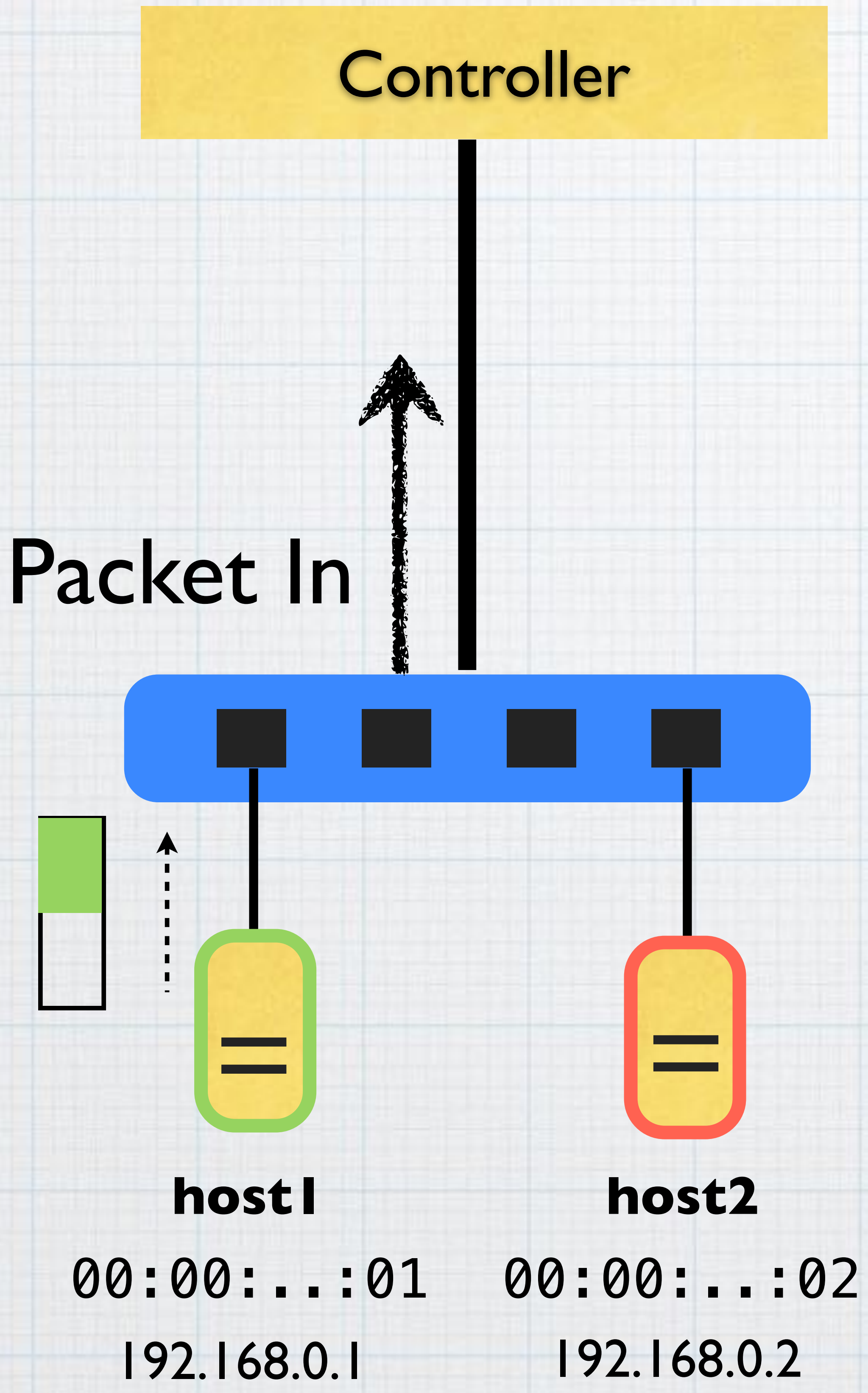

Controller
Switch does not send a Packet In message
Forwarding DB
00:00:00:00:00:01 → 1
00:00:00:00:00:02 → 4
Flow Table
→ 1
host1
00:00:...:01
192.168.0.1
host2
00:00:...:02
192.168.0.2


Controller
Forwarding DB
00:00:00:00:00:01 → 1
00:00:00:00:00:02 → 4
Packet In
Flow Table
→ 1
host1
00:00:...:01
192.168.0.1
host2
00:00:...:02
192.168.0.2


Controller
Flow Mod
Packet Out
Forwarding DB
00:00:00:00:00:01 → 1
00:00:00:00:00:02 → 4
Flow Table
→ 1
host1
00:00:...:01
192.168.0.1
host2
00:00:...:02
192.168.0.2


Controller
Flow Mod
Packet Out
Forwarding DB
00:00:00:00:00:01 → 1
00:00:00:00:00:02 → 4
Flow Table
→ 1
→ 4
host1
00:00:...:01
192.168.0.1
host2
00:00:...:02
192.168.0.2


Controller
Packet In
しない
Forwarding DB
00:00:00:00:00:01 → 1
00:00:00:00:00:02 → 4
Flow Table
→ 1
→ 4
host1
00:00:...:01
192.168.0.1
host2
00:00:...:02
192.168.0.2



Assignment

Generate Packet In events

```
$ trema send_packets \
  --source host1 --dest host2
```

- Send a test packet from host1 to host2

```
$ trema show_stats host2
```

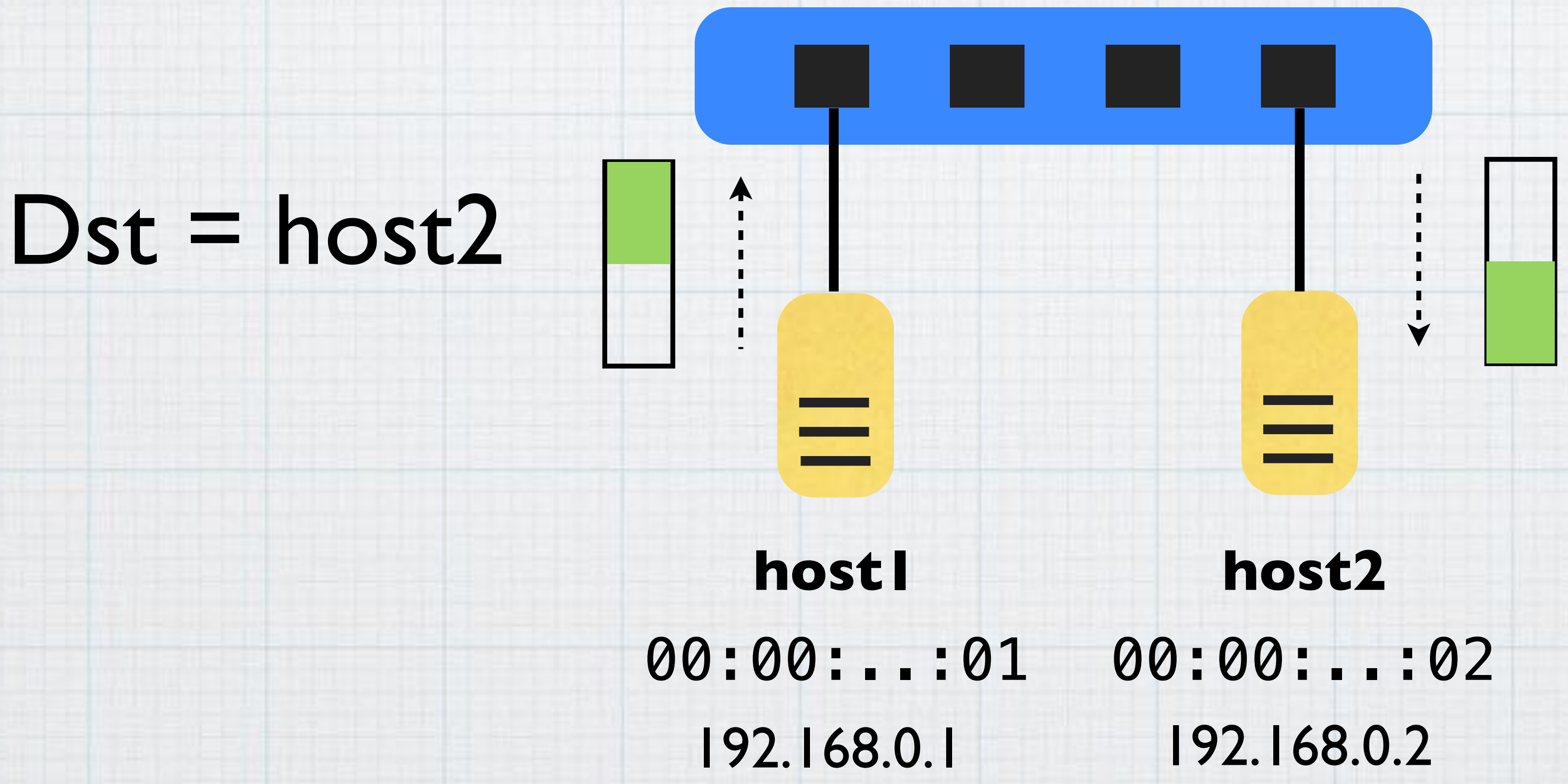
- Show statistics of RX on host2
- You can check whether packets properly arrive at host2.

How to send packets to only destination nodes

Hash table

Forwarding DB

00:00:00:00:00:01 → 1
00:00:00:00:00:02 → 4



```
# Learn a map from MAC addresses to
ports

@fdb[message.macsa] = message.in_port


# Look up a destination port with a
MAC address in the hash table

port_no = @fdb[message.macda]
```

- Learning: hash[key] = value
- Lookup: hash[key]

Pseudo Code

```
def packet_in(dpid, message)
  Learn a map from a MAC address to a port ID
  Look up a port ID with message.macda in the FDB
  if the port ID is in the FDB
    Send message out the port
  else
    Flood message
  end
end
```

```
$ trema dump_flows 0xabc
```

- Dump flows on switch 0xabc

Conclusion

- A mechanisms of an OpenFlow switch
  Packet In/Packet Out/Flow Mod
- How to send test packets
- How to dump a flow table