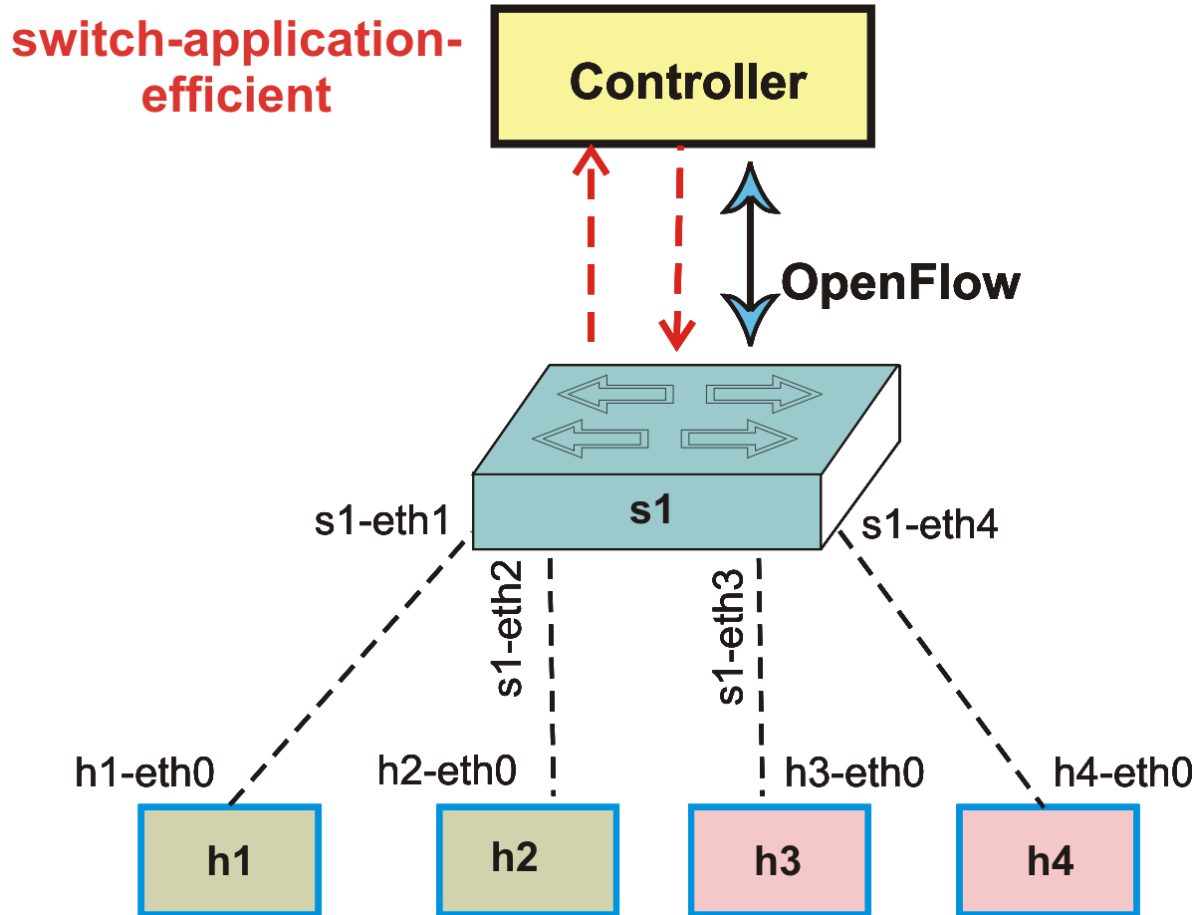


# POX Controller based Switch Application

# Switch Application Logic



Here we are going to create switch application using POX Controller. Switches are intelligent devices. They maintain a dictionary of *mac addresses & ports*. First of all we will discuss the logic of our application. Our switch topology consists of 4 hosts, 1 switch & 1 pox controller as shown.

When we will run our switch application “*switch-application-efficient*” on top of POX controller, 2 tables will be maintained. One table will be maintained at the controller & other table (flow table) will be maintained at the switch. The table maintained at the controller will be “*mac to port table*”. The table maintained at switch will contain flow entries. Initially both tables will be empty.

The different stages involved in creating table at controller & adding the entries in the flow table of switch are shown in next few slides

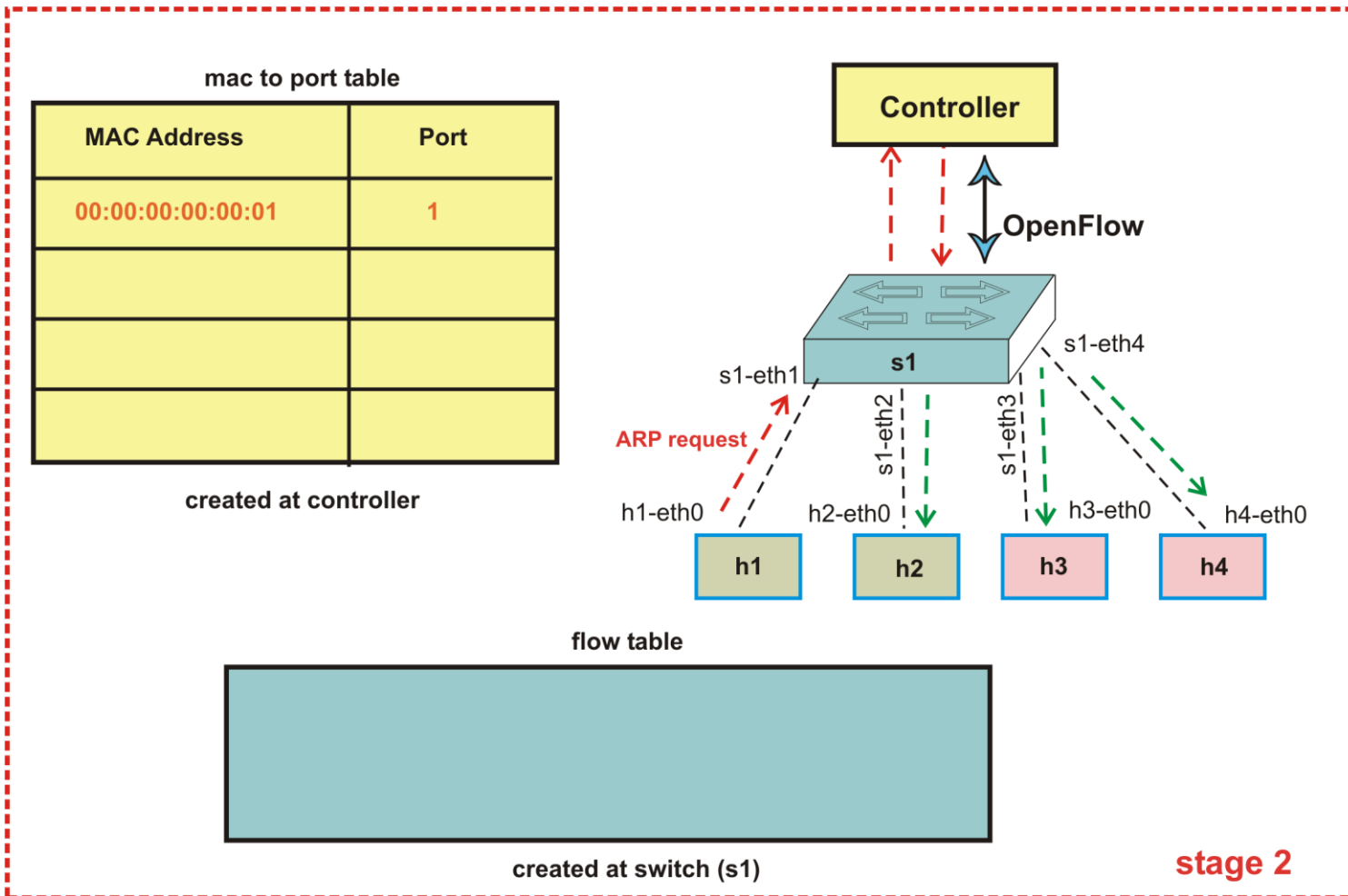
# Stage 1:

mac to port table	
MAC Address	Port

stage 1

Initially the “*mac to port*” at the controller will be empty.

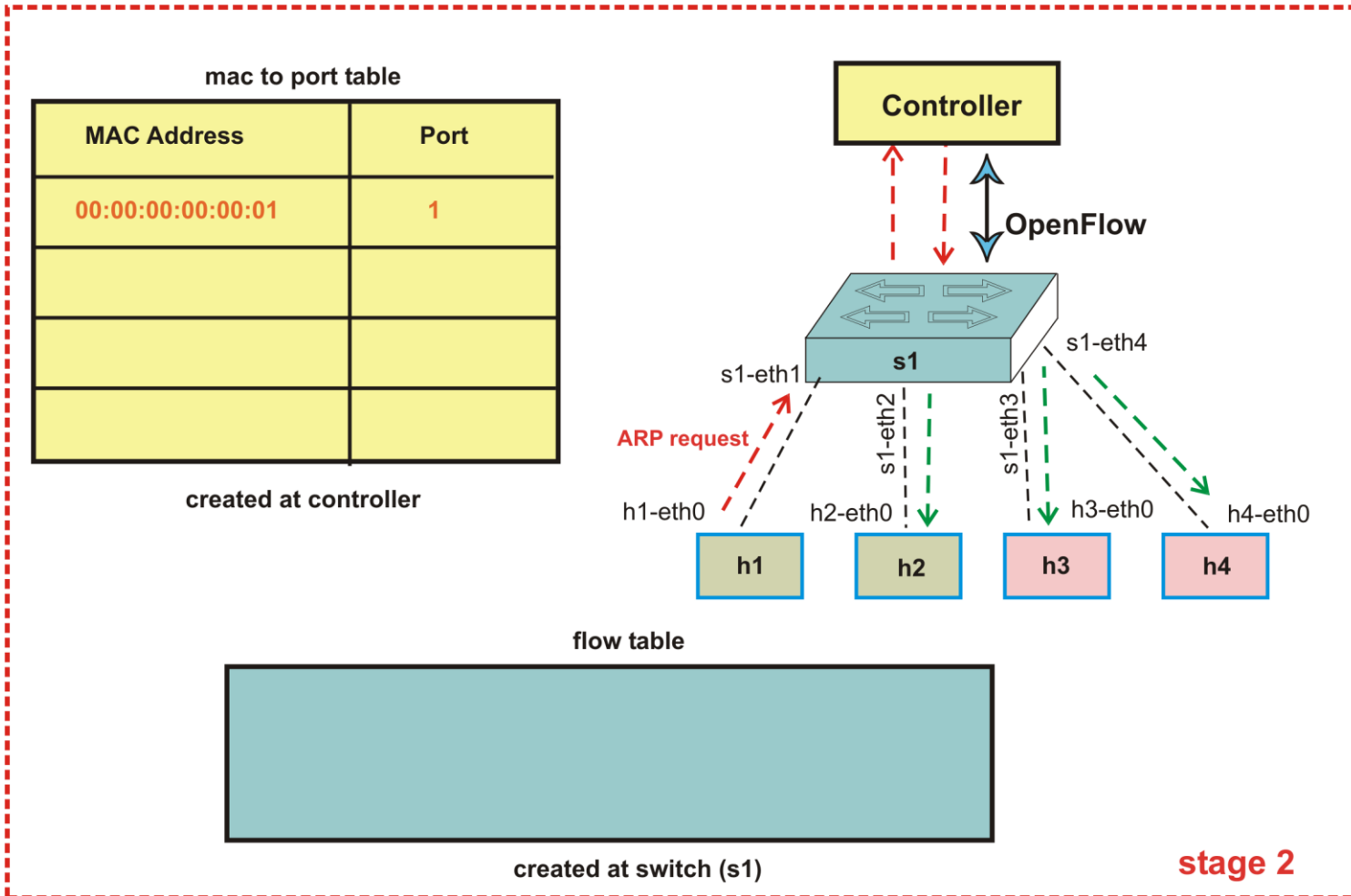
## Stage 2:



Now host h1 wants to ping to host h4. What will happen. First h1 will send ARP request. Since flow table at switch “s1” does not contain any flow entry (ie it does not know how to handle the ARP request), so the packet is sent to the controller.

Controller looks at src mac address & the port on which packet came to the switch & makes 1 entry in the “mac to port” table maintained at the controller. Since in our case, the packet’s src mac address was “00:00:00:00:00:01” & came to the switch through port 1. So the controller made the entry in the controller table as shown.

## Stage 2:

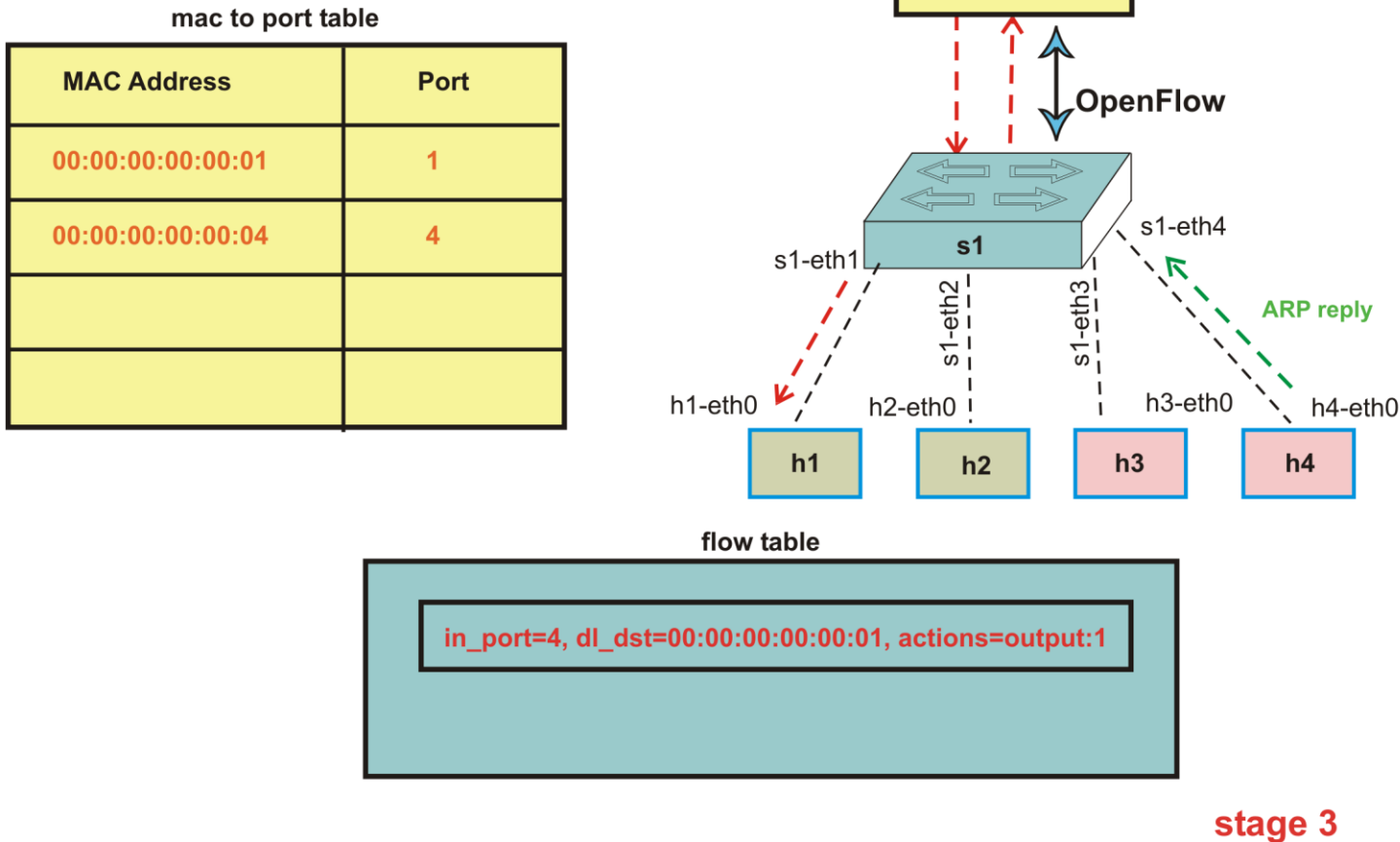


Now controller will look at the dst mac address which is “00:00:00:00:00:04” in our case.

Controller will look for dst mac address in “mac to port” table. If entry is found, the packet will be sent out from the corresponding port, otherwise it will be flooded.

As can be seen, the dst mac address “00:00:00:00:00:04” is not in table, so the packet is getting flooded & no entry is made at switch flow table.

# Stage 3:



Now *h4* will send *ARP reply*. Since flow table at switch “*s1*” does not contain any flow entry (ie it does not know how to handle the *ARP reply*), so the packet is sent to the controller. Controller looks at src mac address & the port on which packet came to the switch & makes 1 entry in the “*mac to port*” table maintained at the controller.

Since in our case, the packet’s src mac address was “00:00:00:00:00:04” & came to the switch through *port 4*. So the controller made the entry in the “*mac to port*” table.

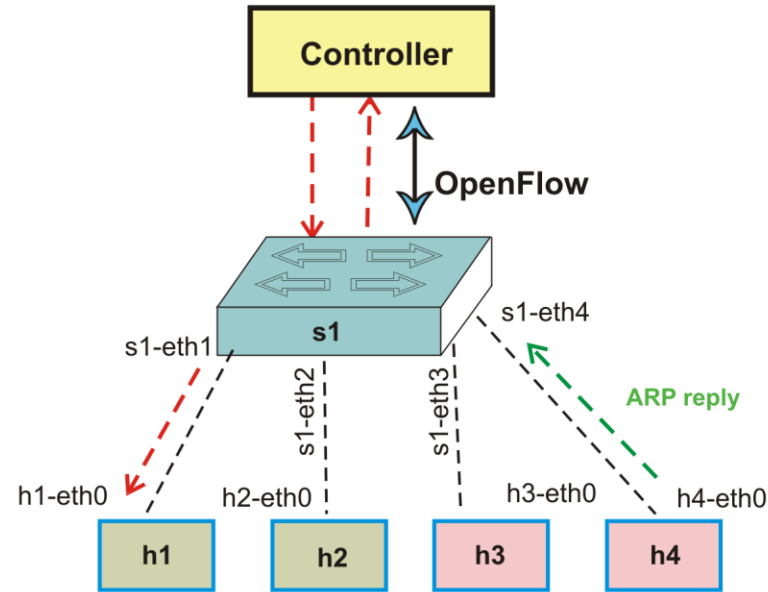
Now controller will look at the dst mac address which is “00:00:00:00:00:01” in our case. Controller will look for dst mac address in “*mac to port*” table. If entry is found, the packet will be sent out from the corresponding port, otherwise it will be flooded.

As can be seen, the dst mac address “00:00:00:00:00:01” is in the table, so the packet will be sent out from port 1 & one flow entry is made at switch flow table.

# Stage 3:

mac to port table

MAC Address	Port
00:00:00:00:00:01	1
00:00:00:00:00:04	4



flow table

in\_port=4, dl\_dst=00:00:00:00:00:01, actions=output:1

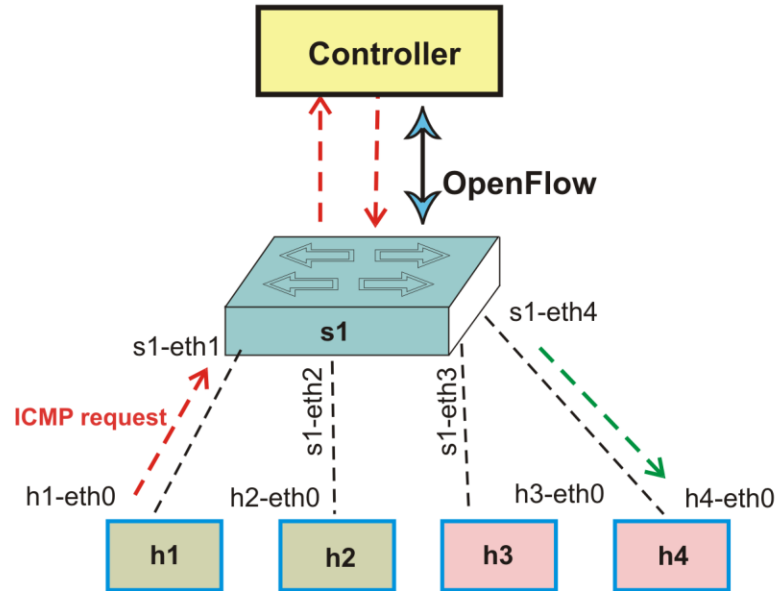
stage 3

The entry added tells the switch if further packets with dst mac address "00:00:00:00:00:01" enters the switch, do not send these packets to the controller. Handle these packets at switch itself & sent the packet out from *port 1*.

# Stage 4:

mac to port table

MAC Address	Port
00:00:00:00:00:01	1
00:00:00:00:00:04	4



flow table

in_port=4, dl_dst=00:00:00:00:00:01, actions=output:1
in_port=1, dl_dst=00:00:00:00:00:04, actions=output:4

stage 4

The next packet will be ICMP request from host *h1* (00:00:00:00:00:01) to host *h4* (00:00:00:00:00:04). The switch has got no entry regarding how to handle traffic going to dst mac "00:00:00:00:00:04", so the packet will be sent to the controller.

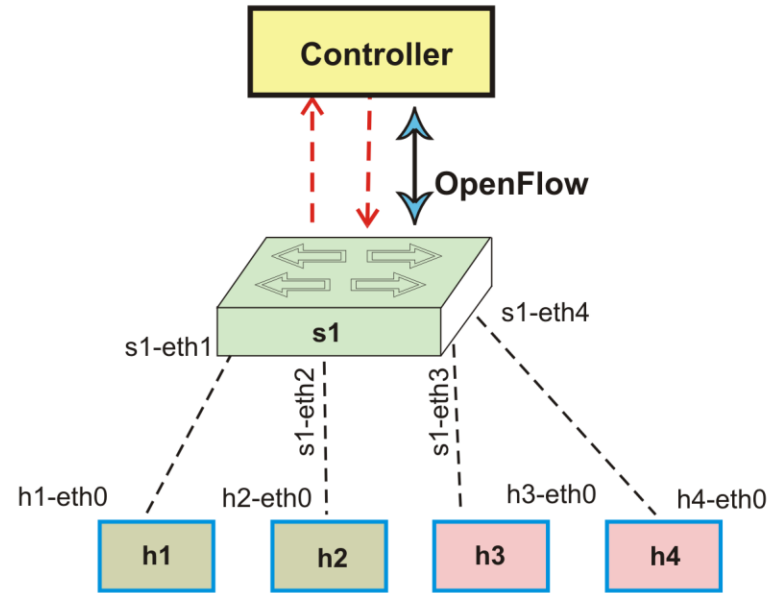
Controller will consult its table. Since now controller has information about dst mac so it will instruct the switch to send the packet out from *port 4*.



# Stage 5:

mac to port table

MAC Address	Port
00:00:00:00:00:01	1
00:00:00:00:00:04	4
00:00:00:00:00:02	2
00:00:00:00:00:03	3



After some time the “*mac to port*” table at controller & flow table at switch will look as shown.

flow table

in_port=4, dl_dst=00:00:00:00:00:01, actions=output:1
in_port=1, dl_dst=00:00:00:00:00:04, actions=output:4
in_port=2, dl_dst=00:00:00:00:00:03, actions=output:3
in_port=3, dl_dst=00:00:00:00:00:02, actions=output:2

stage 5