**Part (a): Why the pings fail initially**

**Ping Fails: Explanation**

In the default **Mininet environment**, switches are **OpenFlow switches** (by default OVSSwitch) that **do not forward packets unless explicitly programmed**.

That means:

* There is **no learning switch behavior** by default.
* **No rules** are installed in the switches to forward traffic between ports.
* Even though IPs are assigned and hosts can send ICMP packets, the switches don’t know what to do with them → they get **dropped**.

**Conceptual Breakdown**

* The topology has **redundant links** → this causes **Layer 2 loops**.
* In **traditional Layer 2 networks**, loops must be **avoided** or **managed** using protocols like **Spanning Tree Protocol (STP)**.
* In **Mininet/SDN setups**, loops are okay if you have **a controller** handling forwarding logic (like ONOS, POX, Ryu, etc.).

So the initial failure is due to:

1. No controller is active
2. Switches are not learning or forwarding
3. Packets sent from hosts never make it to the destination

b). Since we **don't want to modify the Mininet topology** and we're looking for a **protocol-based solution** to prevent loops in **Layer 2 networks**, the **ideal answer is to simulate or enable a loop prevention mechanism like STP**.

Mininet **does not natively support STP**, but **Open vSwitch (OVS)** — which Mininet uses by default for switches — **does** support Spanning Tree Protocol.

**How to Enable STP in Mininet**

After you start your Mininet topology (e.g., sudo python3 topo.py), open a terminal and run the following commands:

sudo ovs-vsctl set Bridge s1 stp\_enable=true

sudo ovs-vsctl set Bridge s2 stp\_enable=true

sudo ovs-vsctl set Bridge s3 stp\_enable=true

sudo ovs-vsctl set Bridge s4 stp\_enable=true

* Later on when some protocol is enabled to avoid the loop then all hosts are reachable to one another.