Each Below is a Python script (***sevi\_block\_topology.py***) that sets up the described SD-IoT network topology using Mininet-Wifi.

**Topology Code in Python (for Mininet-Wifi)**

|  |
| --- |
| **from mininet.net import Mininet**  **from mininet.node import Controller, OVSSwitch, RemoteController**  **from mininet.link import TCLink**  **from mininet.topo import Topo**  **from mininet.cli import CLI**  **class SEVITopology(Topo):**  **def build(self):**  **# Create Controllers**  **central\_ctrl = self.addController('c0', controller=RemoteController, ip='127.0.0.1', port=6633)**  **backup\_ctrl = self.addController('c1', controller=RemoteController, ip='127.0.0.1', port=6634)**  **domain\_c1 = self.addController('d1', controller=RemoteController, ip='127.0.0.1', port=6635)**  **domain\_c2 = self.addController('d2', controller=RemoteController, ip='127.0.0.1', port=6636)**  **# Aggregation Switches**  **agg\_sw1 = self.addSwitch('s1', cls=OVSSwitch, protocols='OpenFlow13')**  **agg\_sw2 = self.addSwitch('s2', cls=OVSSwitch, protocols='OpenFlow13')**  **# Edge P4 Switches**  **edge\_sw1 = self.addSwitch('s3', cls=OVSSwitch, protocols='OpenFlow13')**  **edge\_sw2 = self.addSwitch('s4', cls=OVSSwitch, protocols='OpenFlow13')**  **edge\_sw3 = self.addSwitch('s5', cls=OVSSwitch, protocols='OpenFlow13')**  **edge\_sw4 = self.addSwitch('s6', cls=OVSSwitch, protocols='OpenFlow13')**  **# IoT Networks and Hosts**  **for i in range(1, 5):**  **ap = self.addAccessPoint(f'ap{i}', ssid=f'IoT-AP{i}', mode="g", channel="5")**  **for j in range(5):**  **host = self.addHost(f'n{i}{j+1}')**  **self.addLink(host, ap)**  **self.addLink(ap, locals()[f'edge\_sw{i}'])**  **# Linking Edge to Aggregation Switches**  **self.addLink(edge\_sw1, agg\_sw1)**  **self.addLink(edge\_sw2, agg\_sw1)**  **self.addLink(edge\_sw3, agg\_sw2)**  **self.addLink(edge\_sw4, agg\_sw2)**  **# Linking Aggregation to Domain Controllers**  **self.addLink(agg\_sw1, domain\_c1)**  **self.addLink(agg\_sw2, domain\_c2)**  **# Backup and Central Controllers**  **self.addLink(backup\_ctrl, domain\_c1)**  **self.addLink(backup\_ctrl, domain\_c2)**  **self.addLink(central\_ctrl, backup\_ctrl)**  **def run():**  **topo = SEVITopology()**  **net = Mininet(topo=topo, controller=None, link=TCLink)**  **net.start()**  **CLI(net)**  **net.stop()**  **if \_\_name\_\_ == '\_\_main\_\_':**  **run()** |