INDIAN INSTITUTE OF INFORMATION TECHNOLOGY, DESIGN AND MANUFACTURING, KANCHEEPURAM

DCN LAB-6

NAME:K.NITHESH
ROLL NO:ESD19I008

AIM: To Create linear,tree and single topologies using mininet in virtualbox

Theory:

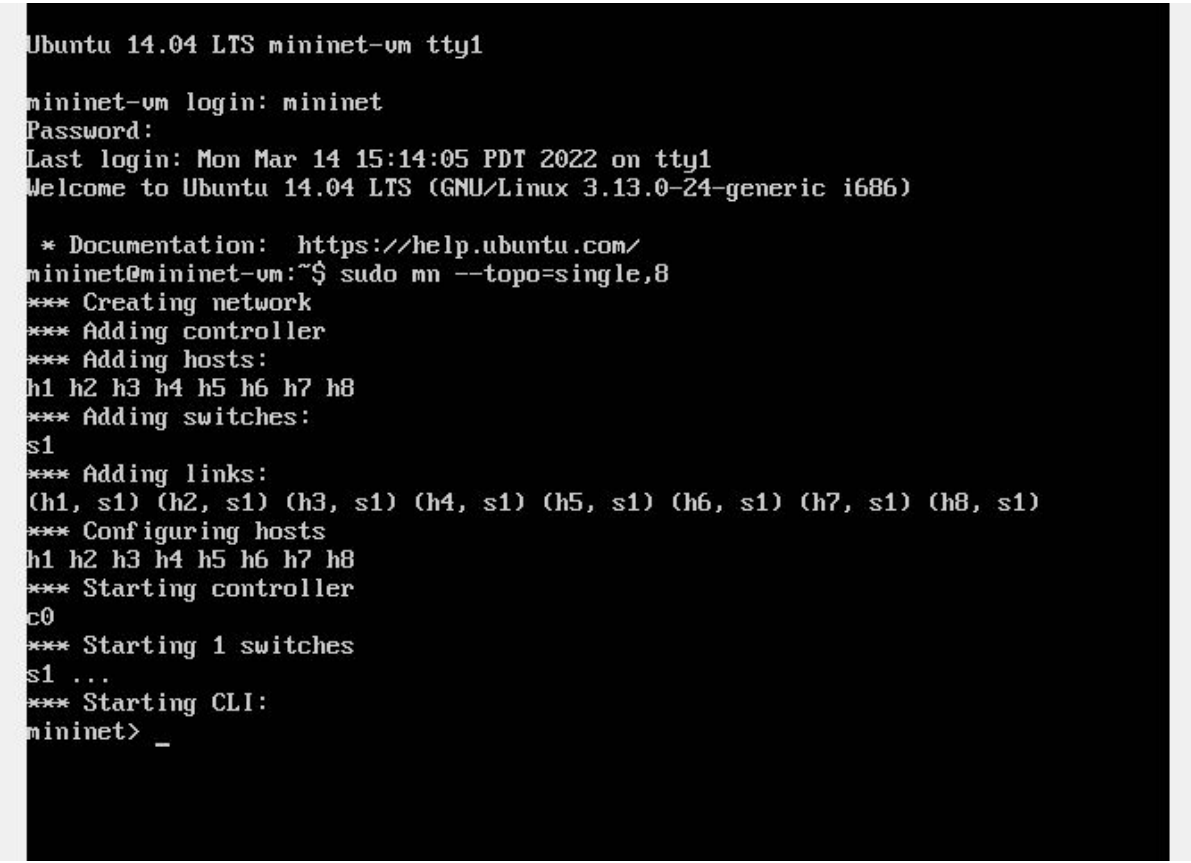
MININET: Mininet is **a software emulator for prototyping a large network on a single machine**. Mininet can be used to quickly create a realistic virtual network running actual kernel, switch and software application code on a personal computer.

SINGLE TOPOLOGY:
A default topology consists of a single node that runs on a host, along with the required application services

LINEAR TOPOLOGY:
A linear topology is **a network topology consisting of a main run of cable with a terminator at each end**.

TREE TOPOLOGY:
A tree topology, or star-bus topology, is **a hybrid network topology in which star networks are interconnected via bus networks**. Tree networks are hierarchical, and each node can have an arbitrary number of child nodes.

CREATING SINGLE TOPOLOGY NETWORK WITH 8 HOSTS

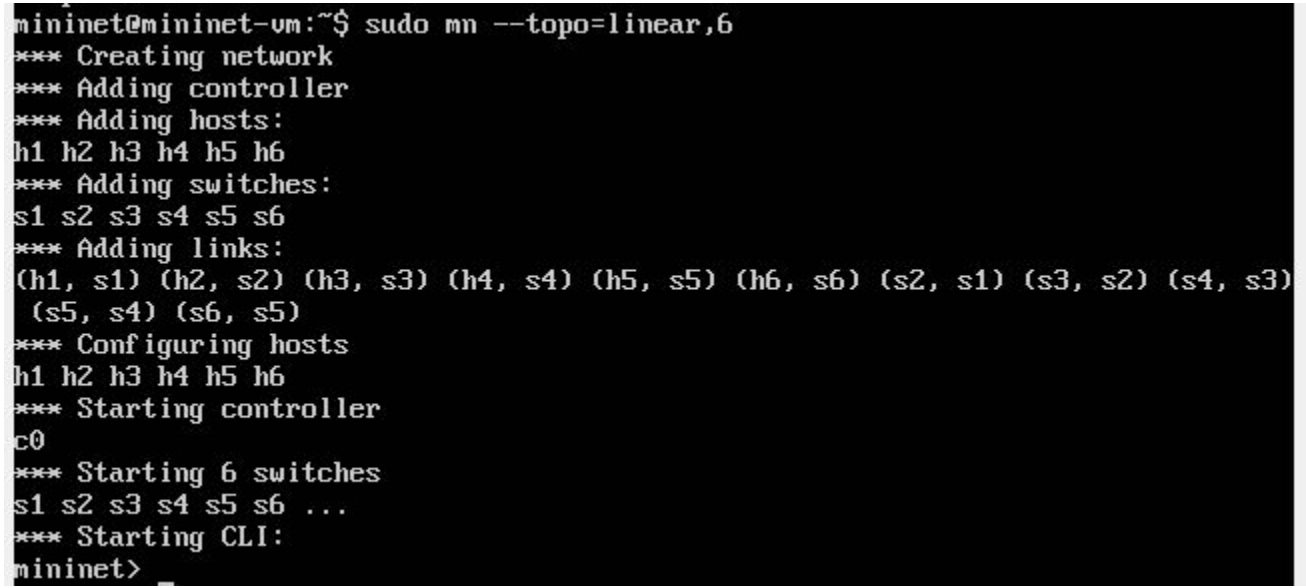


```
*** Starting CLI:
mininet> links
h1-eth0<->s1-eth1 (OK OK)
h2-eth0<->s1-eth2 (OK OK)
h3-eth0<->s1-eth3 (OK OK)
h4-eth0<->s1-eth4 (OK OK)
h5-eth0<->s1-eth5 (OK OK)
h6-eth0<->s1-eth6 (OK OK)
h7-eth0<->s1-eth7 (OK OK)
h8-eth0<->s1-eth8 (OK OK)
mininet> dump
<Host h1: h1-eth0:10.0.0.1 pid=1317>
<Host h2: h2-eth0:10.0.0.2 pid=1320>
<Host h3: h3-eth0:10.0.0.3 pid=1322>
<Host h4: h4-eth0:10.0.0.4 pid=1324>
<Host h5: h5-eth0:10.0.0.5 pid=1326>
<Host h6: h6-eth0:10.0.0.6 pid=1328>
<Host h7: h7-eth0:10.0.0.7 pid=1330>
<Host h8: h8-eth0:10.0.0.8 pid=1332>
<OVSSwitch s1: lo:127.0.0.1,s1-eth1:None,s1-eth2:None,s1-eth3:None,s1-eth4:None,
s1-eth5:None,s1-eth6:None,s1-eth7:None,s1-eth8:None pid=1337>
<Controller c0: 127.0.0.1:6633 pid=1310>
mininet> _
```

```
mininet> h1 ping h2
PING 10.0.0.2 (10.0.0.2) 56(84) bytes of data.
64 bytes from 10.0.0.2: icmp_seq=1 ttl=64 time=1.02 ms
64 bytes from 10.0.0.2: icmp_seq=2 ttl=64 time=0.333 ms
64 bytes from 10.0.0.2: icmp_seq=3 ttl=64 time=0.053 ms
64 bytes from 10.0.0.2: icmp_seq=4 ttl=64 time=0.032 ms
64 bytes from 10.0.0.2: icmp_seq=5 ttl=64 time=0.034 ms
64 bytes from 10.0.0.2: icmp_seq=6 ttl=64 time=0.026 ms
64 bytes from 10.0.0.2: icmp_seq=7 ttl=64 time=0.403 ms
64 bytes from 10.0.0.2: icmp_seq=8 ttl=64 time=0.043 ms
64 bytes from 10.0.0.2: icmp_seq=9 ttl=64 time=0.042 ms
64 bytes from 10.0.0.2: icmp_seq=10 ttl=64 time=0.054 ms
^C
--- 10.0.0.2 ping statistics ---
10 packets transmitted, 10 received, 0% packet loss, time 9001ms
rtt min/avg/max/mdev = 0.026/0.204/1.025/0.303 ms
```

```
mininet> exit
*** Stopping 1 controllers
c0
*** Stopping 8 links
........
*** Stopping 1 switches
s1
*** Stopping 8 hosts
h1 h2 h3 h4 h5 h6 h7 h8
*** Done
completed in 273.134 seconds
```

CREATING LINEAR TOPOLOGY NETWORK WITH 6 HOSTS



```
mininet> links
h1-eth0<->s1-eth1 (OK OK)
h2-eth0<->s2-eth1 (OK OK)
h3-eth0<->s3-eth1 (OK OK)
h4-eth0<->s4-eth1 (OK OK)
h5-eth0<->s5-eth1 (OK OK)
h6-eth0<->s6-eth1 (OK OK)
s2-eth2<->s1-eth2 (OK OK)
s3-eth2<->s2-eth3 (OK OK)
s4-eth2<->s3-eth3 (OK OK)
s5-eth2<->s4-eth3 (OK OK)
s6-eth2<->s5-eth3 (OK OK)
mininet> dump
<Host h1: h1-eth0:10.0.0.1 pid=1730>
<Host h2: h2-eth0:10.0.0.2 pid=1733>
<Host h3: h3-eth0:10.0.0.3 pid=1735>
<Host h4: h4-eth0:10.0.0.4 pid=1737>
<Host h5: h5-eth0:10.0.0.5 pid=1739>
<Host h6: h6-eth0:10.0.0.6 pid=1741>
<OVSSwitch s1: lo:127.0.0.1,s1-eth1:None,s1-eth2:None pid=1746>
<OVSSwitch s2: lo:127.0.0.1,s2-eth1:None,s2-eth2:None,s2-eth3:None pid=1749>
<OVSSwitch s3: lo:127.0.0.1,s3-eth1:None,s3-eth2:None,s3-eth3:None pid=1752>
<OVSSwitch s4: lo:127.0.0.1,s4-eth1:None,s4-eth2:None,s4-eth3:None pid=1755>
<OVSSwitch s5: lo:127.0.0.1,s5-eth1:None,s5-eth2:None,s5-eth3:None pid=1758>
<OVSSwitch s6: lo:127.0.0.1,s6-eth1:None,s6-eth2:None pid=1761>
<Controller c0: 127.0.0.1:6633 pid=1723>
mininet>
```

```
mininet> h1 ping h4
PING 10.0.0.4 (10.0.0.4) 56(84) bytes of data.
64 bytes from 10.0.0.4: icmp_seq=1 ttl=64 time=6.57 ms
64 bytes from 10.0.0.4: icmp_seq=2 ttl=64 time=2.01 ms
64 bytes from 10.0.0.4: icmp_seq=3 ttl=64 time=0.041 ms
64 bytes from 10.0.0.4: icmp_seq=4 ttl=64 time=0.054 ms
64 bytes from 10.0.0.4: icmp_seq=5 ttl=64 time=0.056 ms
64 bytes from 10.0.0.4: icmp_seq=6 ttl=64 time=0.056 ms
64 bytes from 10.0.0.4: icmp_seq=7 ttl=64 time=0.613 ms
64 bytes from 10.0.0.4: icmp_seq=8 ttl=64 time=0.055 ms
64 bytes from 10.0.0.4: icmp_seq=9 ttl=64 time=0.056 ms
64 bytes from 10.0.0.4: icmp_seq=10 ttl=64 time=0.056 ms
^C
--- 10.0.0.4 ping statistics ---
10 packets transmitted, 10 received, 0% packet loss, time 9004ms
rtt min/avg/max/mdev = 0.041/0.957/6.575/1.962 ms

mininet> exit
*** Stopping 1 controllers
c0
*** Stopping 11 links
...........
*** Stopping 6 switches
s1 s2 s3 s4 s5 s6
*** Stopping 6 hosts
h1 h2 h3 h4 h5 h6
*** Done
completed in 165.432 seconds
mininet@mininet-vm:~$
```

CREATING TREE TOPOLOGY NETWORK WITH DEPTH 3 AND FANOUT 3

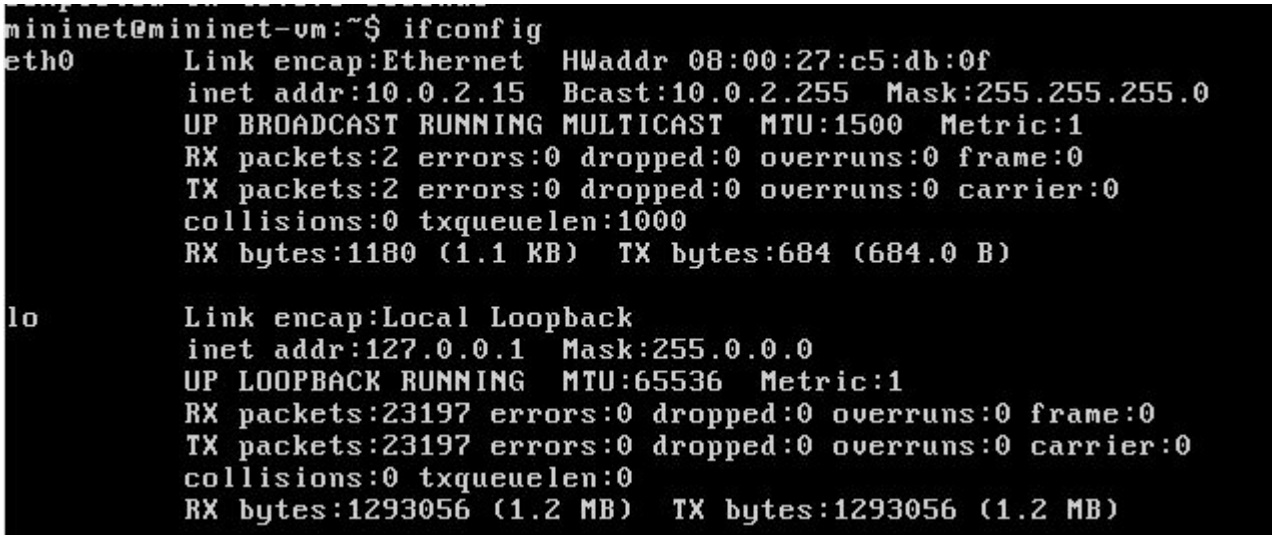
```
 * Documentation:  https://help.ubuntu.com/
mininet@mininet-vm:~$ sudo mn --topo=tree,3,3
*** Creating network
*** Adding controller
*** Adding hosts:
h1 h2 h3 h4 h5 h6 h7 h8 h9 h10 h11 h12 h13 h14 h15 h16 h17 h18 h19 h20 h21 h22 h
23 h24 h25 h26 h27
*** Adding switches:
s1 s2 s3 s4 s5 s6 s7 s8 s9 s10 s11 s12 s13
*** Adding links:
(s1, s2) (s1, s6) (s1, s10) (s2, s3) (s2, s4) (s2, s5) (s3, h1) (s3, h2) (s3, h3
) (s4, h4) (s4, h5) (s4, h6) (s5, h7) (s5, h8) (s5, h9) (s6, s7) (s6, s8) (s6, s
9) (s7, h10) (s7, h11) (s7, h12) (s8, h13) (s8, h14) (s8, h15) (s9, h16) (s9, h1
7) (s9, h18) (s10, s11) (s10, s12) (s10, s13) (s11, h19) (s11, h20) (s11, h21) (
s12, h22) (s12, h23) (s12, h24) (s13, h25) (s13, h26) (s13, h27)
*** Configuring hosts
h1 h2 h3 h4 h5 h6 h7 h8 h9 h10 h11 h12 h13 h14 h15 h16 h17 h18 h19 h20 h21 h22 h
23 h24 h25 h26 h27
*** Starting controller
c0
*** Starting 13 switches
s1 s2 s3 s4 s5 s6 s7 s8 s9 s10 s11 s12 s13 ...
*** Starting CLI:
```

```
mininet> exit
*** Stopping 1 controllers
c0
*** Stopping 39 links
.......................................
*** Stopping 13 switches
s1 s2 s3 s4 s5 s6 s7 s8 s9 s10 s11 s12 s13
*** Stopping 27 hosts
h1 h2 h3 h4 h5 h6 h7 h8 h9 h10 h11 h12 h13 h14 h15 h16 h17 h18 h19 h20 h21 h22 h
23 h24 h25 h26 h27
*** Done
completed in 200.335 seconds
```

CREATING TREE TOPOLOGY NETWORK WITH DEPTH 4 AND FANOUT 4

```
*** Stopping 85 switches
s1 s2 s3 s4 s5 s6 s7 s8 s9 s10 s11 s12 s13 s14 s15 s16 s17 s18 s19 s20 s21 s22 s
23 s24 s25 s26 s27 s28 s29 s30 s31 s32 s33 s34 s35 s36 s37 s38 s39 s40 s41 s42 s
43 s44 s45 s46 s47 s48 s49 s50 s51 s52 s53 s54 s55 s56 s57 s58 s59 s60 s61 s62 s
63 s64 s65 s66 s67 s68 s69 s70 s71 s72 s73 s74 s75 s76 s77 s78 s79 s80 s81 s82 s
83 s84 s85
*** Stopping 256 hosts
h1 h2 h3 h4 h5 h6 h7 h8 h9 h10 h11 h12 h13 h14 h15 h16 h17 h18 h19 h20 h21 h22 h
23 h24 h25 h26 h27 h28 h29 h30 h31 h32 h33 h34 h35 h36 h37 h38 h39 h40 h41 h42 h
43 h44 h45 h46 h47 h48 h49 h50 h51 h52 h53 h54 h55 h56 h57 h58 h59 h60 h61 h62 h
63 h64 h65 h66 h67 h68 h69 h70 h71 h72 h73 h74 h75 h76 h77 h78 h79 h80 h81 h82 h
83 h84 h85 h86 h87 h88 h89 h90 h91 h92 h93 h94 h95 h96 h97 h98 h99 h100 h101 h10
2 h103 h104 h105 h106 h107 h108 h109 h110 h111 h112 h113 h114 h115 h116 h117 h11
8 h119 h120 h121 h122 h123 h124 h125 h126 h127 h128 h129 h130 h131 h132 h133 h13
4 h135 h136 h137 h138 h139 h140 h141 h142 h143 h144 h145 h146 h147 h148 h149 h15
0 h151 h152 h153 h154 h155 h156 h157 h158 h159 h160 h161 h162 h163 h164 h165 h16
6 h167 h168 h169 h170 h171 h172 h173 h174 h175 h176 h177 h178 h179 h180 h181 h18
2 h183 h184 h185 h186 h187 h188 h189 h190 h191 h192 h193 h194 h195 h196 h197 h19
8 h199 h200 h201 h202 h203 h204 h205 h206 h207 h208 h209 h210 h211 h212 h213 h21
4 h215 h216 h217 h218 h219 h220 h221 h222 h223 h224 h225 h226 h227 h228 h229 h23
0 h231 h232 h233 h234 h235 h236 h237 h238 h239 h240 h241 h242 h243 h244 h245 h24
6 h247 h248 h249 h250 h251 h252 h253 h254 h255 h256
*** Done
completed in 106.338 seconds
mininet@mininet-vm:~$
```

MININET COMMANDS :



```
*** Removing excess controllers/ofprotocols/ofdatapaths/pings/noxes
killall controller ofprotocol ofdatapath ping nox_core lt-nox_core ovs-openflowd
 ovs-controller udpbwtest mnexec ivs 2> /dev/null
killall -9 controller ofprotocol ofdatapath ping nox_core lt-nox_core ovs-openfl
owd ovs-controller udpbwtest mnexec ivs 2> /dev/null
pkill -9 -f "sudo mnexec"
*** Removing junk from /tmp
rm -f /tmp/vconn* /tmp/vlogs* /tmp/*.out /tmp/*.log
*** Removing old X11 tunnels
*** Removing excess kernel datapaths
ps ax | egrep -o 'dp[0-9]+' | sed 's/dp/nl:/'
***  Removing OVS datapaths
ovs-vsctl --timeout=1 list-br
ovs-vsctl --timeout=1 list-br
*** Removing all links of the pattern foo-ethX
ip link show | egrep -o '([-_.[:alnum:]]+-eth[[:digit:]]+)'
ip link show
*** Killing stale mininet node processes
pkill -9 -f mininet:
*** Shutting down stale tunnels
pkill -9 -f Tunnel=Ethernet
pkill -9 -f .ssh/mn
rm -f ~/.ssh/mn/*
*** Cleanup complete.
mininet@mininet-vm:~$
```

RESULT:

CREATED SINGLE,LINEAR AND TREE TOPOLOGY NETWORKS IN MINIVET USING VIRTUALBOX.

GOT FAMILIARIZED WITH CREATING NETWORKS IN MININET AND MININET COMMANDS