POLITECNICO DI MILANO

Switching and Routing project 2018/2019

Prof. Guido Maier
Ph.D. student Sebastian Troìa

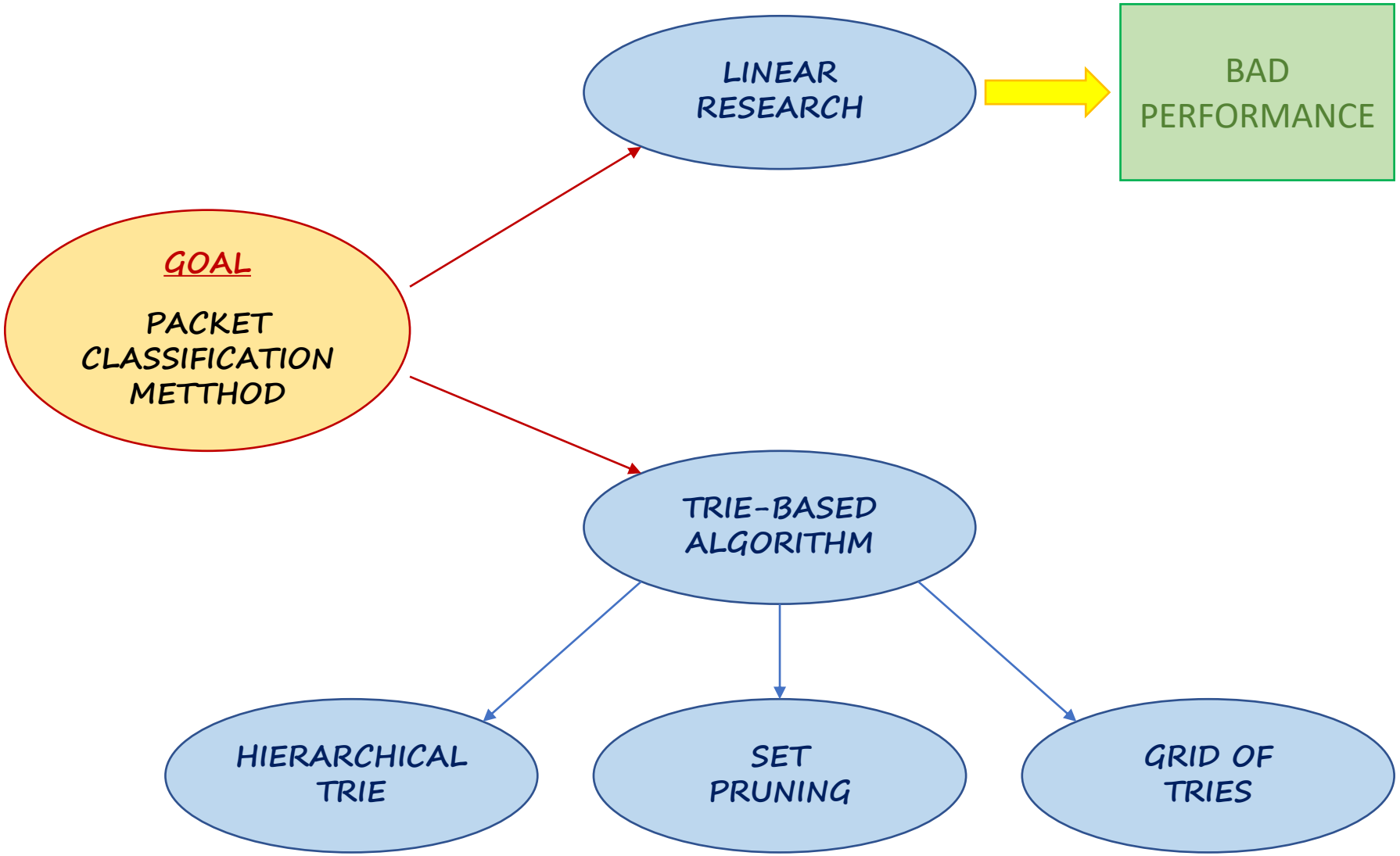
PACKET CLASSIFICATION
TRIE-BASED ALGORITHM

Kasenov Erbol, 916027
Calzi Nicolò, 903430
D'Adda Mattia, 917652

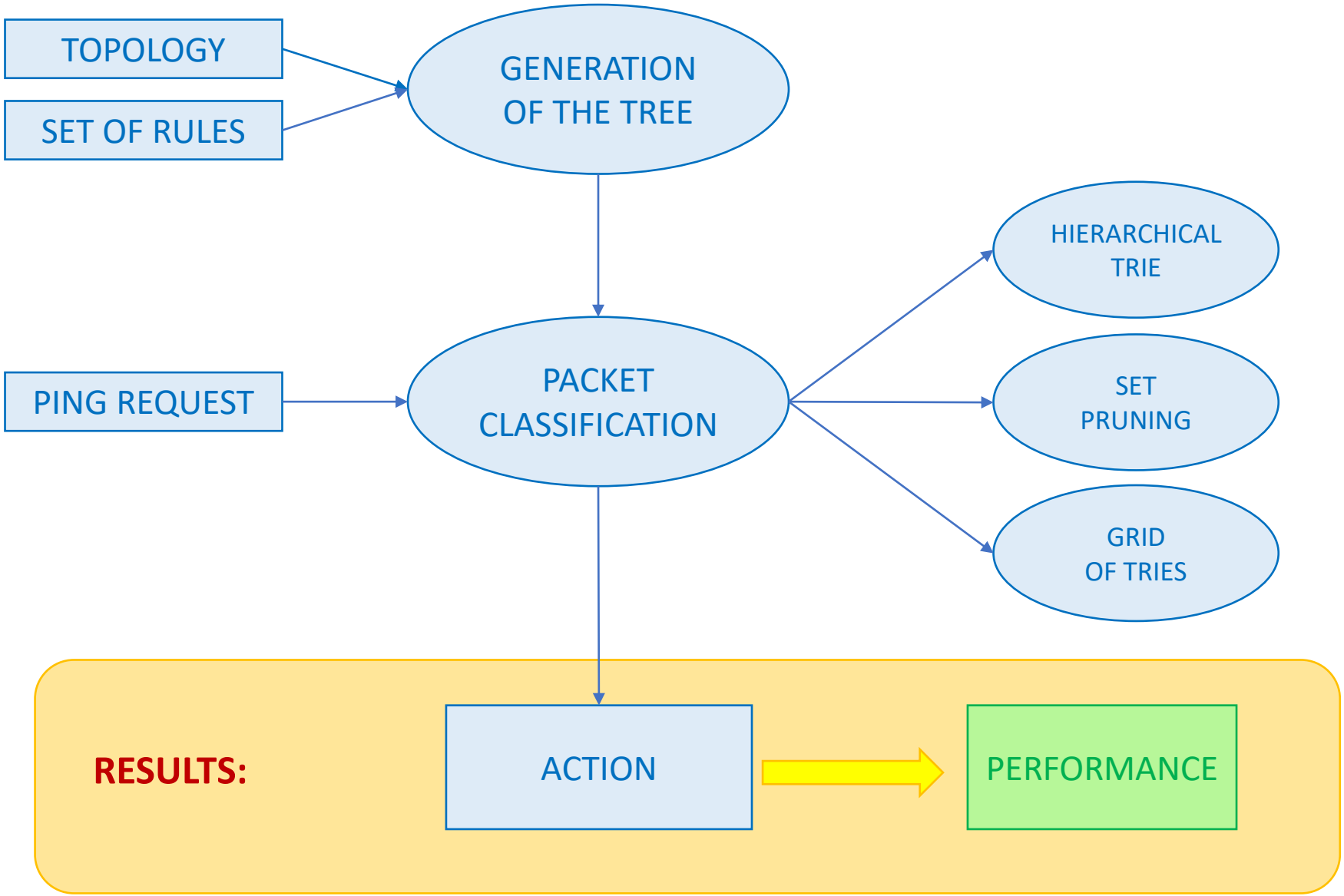
Outline

- Introduction
- Flow diagrams of the code
- Setup of the experiments
- Results
- Performance

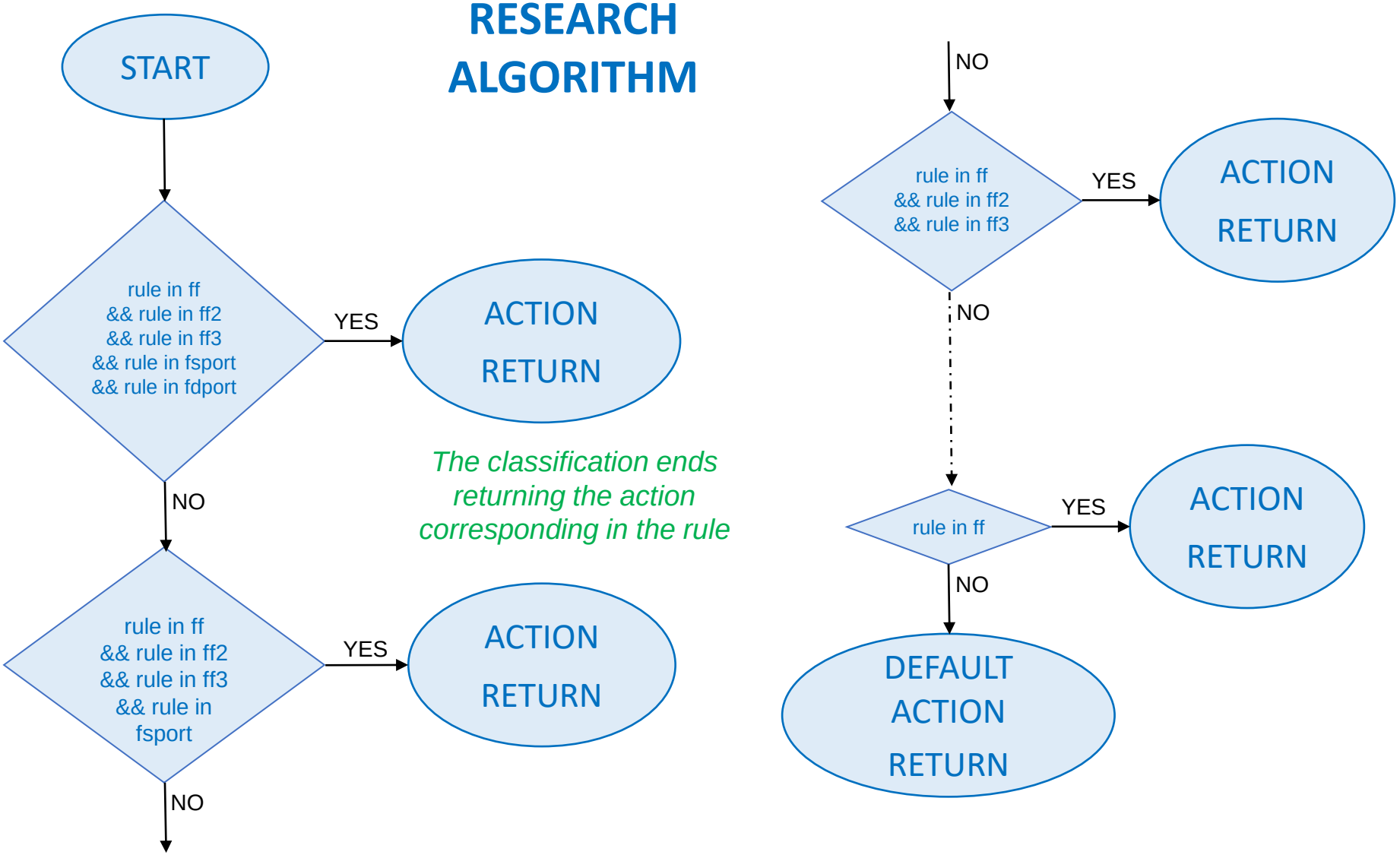
Introduction



Flow diagrams

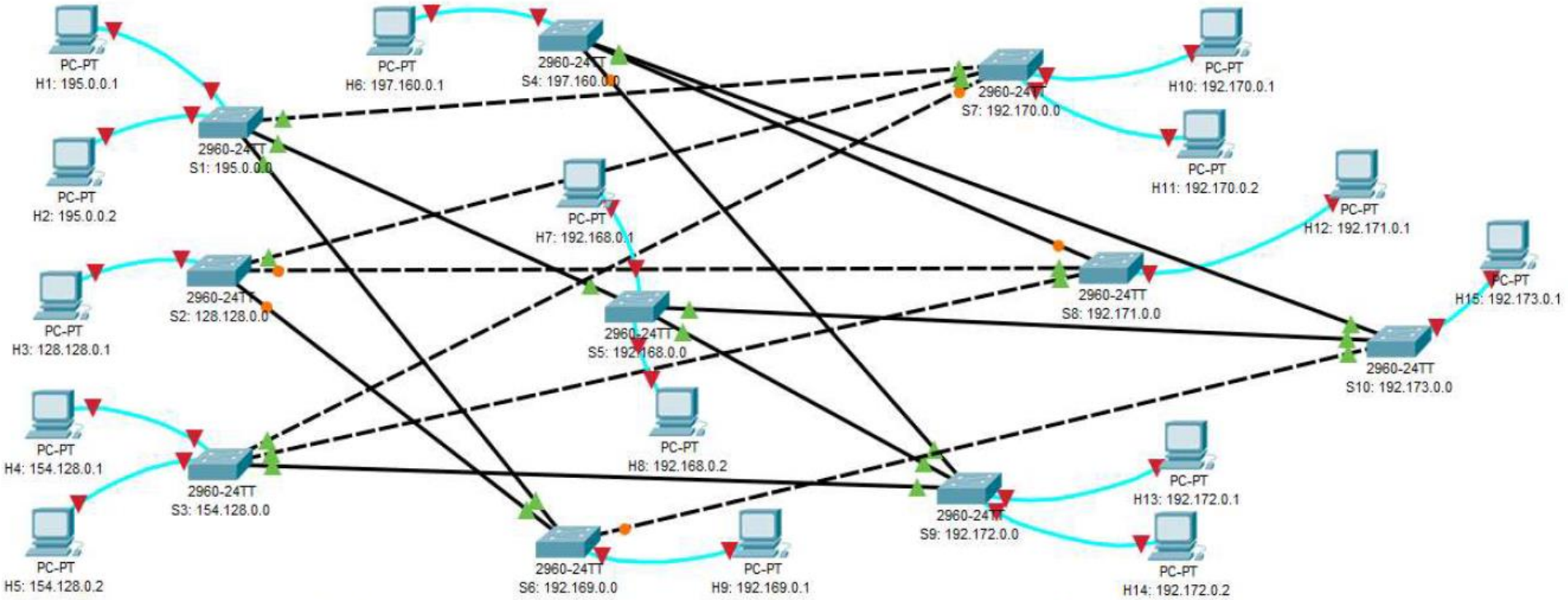


Flow diagrams



Setup of experiments

OUR TOPOLOGY



Switch → Host

S1 {H1,H2}	S6 {H9}
S2 {H3}	S7 {H10,H11}
S3 {H4,H5}	S8 {H12}
S4 {H6}	S9 {H13,H14}
S5 {H7,H8}	S10 {H15}

Switch → Switch

S1 {S5,S6,S7}	S6 {S10,S1,S2}
S2 {S6,S7,S8}	S7 {S1,S2,S3}
S3 {S7,S8,S9}	S8 {S2,S3,S4}
S4 {S8,S9,S10}	S9 {S3,S4,S5}
S5 {S9,S10,S1}	S10 {S4,S5,S6}

Setup of experiments

CONFIGURATION FILE

```
py "-------------------------------------"
py "Configuring network"
py "-------------------------------------"
py "Assign IP address to hosts"
h1 ifconfig h1-eth0 195.0.0.1/8
h2 ifconfig h2-eth0 195.0.0.2/8
h3 ifconfig h3-eth0 128.128.0.1/12
h4 ifconfig h4-eth0 154.128.0.1/16
h5 ifconfig h5-eth0 154.128.0.2/16
h6 ifconfig h6-eth0 197.160.0.1/24
h7 ifconfig h7-eth0 192.168.0.1/24
h8 ifconfig h8-eth0 192.168.0.2/24
h9 ifconfig h9-eth0 192.169.0.1/24
h10 ifconfig h10-eth0 192.170.0.1/24
h11 ifconfig h11-eth0 192.170.0.2/24
h12 ifconfig h12-eth0 192.171.0.1/24
h13 ifconfig h13-eth0 192.172.0.1/24
h14 ifconfig h14-eth0 192.172.0.2/24
h15 ifconfig h15-eth0 192.173.0.1/24
h1 route add default gw 195.0.0.254
h2 route add default gw 195.0.0.254
h3 route add default gw 128.128.0.254
h4 route add default gw 154.128.0.254
h5 route add default gw 154.128.0.254
h6 route add default gw 197.160.0.254
h7 route add default gw 192.168.0.254
h8 route add default gw 192.168.0.254
h9 route add default gw 192.169.0.254
h10 route add default gw 192.170.0.254
h11 route add default gw 192.170.0.254
h12 route add default gw 192.171.0.254
h13 route add default gw 192.172.0.254
h14 route add default gw 192.172.0.254
h15 route add default gw 192.173.0.254
```

CREATION AND NAMING OF HOSTS AND SWITCHES

```
class MyTopo( Topo ):
        "Simple topology example."

        def __init__( self ):
                "Create custom topo."
        # Initialize topology
                Topo.__init__( self )

        # Add hosts and switches
        h1 = self.addHost('h1')
        h2 = self.addHost('h2')
        h3 = self.addHost('h3')
        h4 = self.addHost('h4')
        h5 = self.addHost('h5')
        h6 = self.addHost('h6')
        h7 = self.addHost('h7')
        h8 = self.addHost('h8')
        h9 = self.addHost('h9')
        h10 = self.addHost('h10')
        h11 = self.addHost('h11')
        h12 = self.addHost('h12')
        h13 = self.addHost('h13')
        h14 = self.addHost('h14')
        h15 = self.addHost('h15')
        s1 = self.addSwitch('s1')
        s2 = self.addSwitch('s2')
        s3 = self.addSwitch('s3')
        s4 = self.addSwitch('s4')
        s5 = self.addSwitch('s5')
        s6 = self.addSwitch('s6')
        s7 = self.addSwitch('s7')
        s8 = self.addSwitch('s8')
        s9 = self.addSwitch('s9')
        s10 = self.addSwitch('s10')
```

LINKAGE OF HOSTS

```
# Add links
self.addLink(s1,s5)
self.addLink(s1,s6)
self.addLink(s1,s7)
self.addLink(s2,s6)
self.addLink(s2,s7)
self.addLink(s2,s8)
self.addLink(s3,s7)
self.addLink(s3,s8)
self.addLink(s3,s9)
self.addLink(s4,s8)
self.addLink(s4,s9)
self.addLink(s4,s10)
self.addLink(s5,s9)
self.addLink(s5,s10)
self.addLink(s6,s10)
self.addLink(s1,h1)
self.addLink(s1,h2)
self.addLink(s2,h3)
self.addLink(s3,h4)
self.addLink(s3,h5)
self.addLink(s4,h6)
self.addLink(s5,h7)
self.addLink(s5,h8)
self.addLink(s6,h9)
self.addLink(s7,h10)
self.addLink(s7,h11)
self.addLink(s8,h12)
self.addLink(s9,h13)
self.addLink(s9,h14)
self.addLink(s10,h15)
```

Results

OUTPUT at the end of execution of ping

```
Best prefix match for SRC_IP: ['r43', 'r45', 'r44', 'r47', 'r46', 'r41', 'r40', 'r29', 'r42', 'r49',
 'r48', 'r52', 'r37', 'r32', 'r33', 'r10', 'r11', 'r35', 'r38', 'r39', 'r34', 'r53', 'r50', 'r51', '
r30', 'r31', 'r54', 'r55', 'r8', 'r36']



Best prefix match for DST_IP: ['r14', 'r15', 'r40', 'r39']
Best prefix match for PROTO: ['r10', 'r11', 'r12', 'r15', 'r18', 'r20', 'r22', 'r24', 'r26', 'r28',
'r3', 'r30', 'r32', 'r34', 'r36', 'r38', 'r4', 'r40', 'r42', 'r44', 'r46', 'r48', 'r5', 'r50', 'r52'
, 'r54', 'r6', 'r7', 'r8', 'r9']
Best prefix match S_PORT ['r43', 'r42', 'r45', 'r44', 'r47', 'r46', 'r41', 'r40', 'r29', 'r28', 'r27
', 'r26', 'r25', 'r24', 'r49', 'r22', 'r21', 'r20', 'r37', 'r23', 'r48', 'r34', 'r32', 'r33', 'r17',
 'r14', 'r15', 'r12', 'r13', 'r18', 'r19', 'r35', 'r38', 'r39', 'r52', 'r53', 'r50', 'r51', 'r30', '
r31', 'r54', 'r55', 'r4', 'r5', 'r6', 'r7', 'r36']
Best prefix match D_PORT: ['r43', 'r42', 'r45', 'r44', 'r47', 'r46', 'r41', 'r40', 'r29', 'r28', 'r2
7', 'r26', 'r25', 'r24', 'r49', 'r22', 'r21', 'r20', 'r37', 'r23', 'r48', 'r34', 'r32', 'r33', 'r17'
, 'r14', 'r15', 'r12', 'r13', 'r18', 'r19', 'r35', 'r38', 'r39', 'r52', 'r53', 'r50', 'r51', 'r30',
'r31', 'r54', 'r55', 'r4', 'r6', 'r7', 'r36']
 ========= Packet classification--Set-Prunning-Tree==========
 ---- Packet matched rule-------- r40. Action is allow
Packet in the controller from switch: 1
Packet from the switch: 1, source IP: 195.0.0.2, destination IP: 192.168.0.1, From the port: 2
 --- Packet classification
 --- Packet matched rule r40. Action is allow
```

Rule 40 matches with all of 5 field

Rule 40 is the best prefix

Action correspondig to the rule 40 → **ALLOW**

Performance

HIERARCHICAL TREE

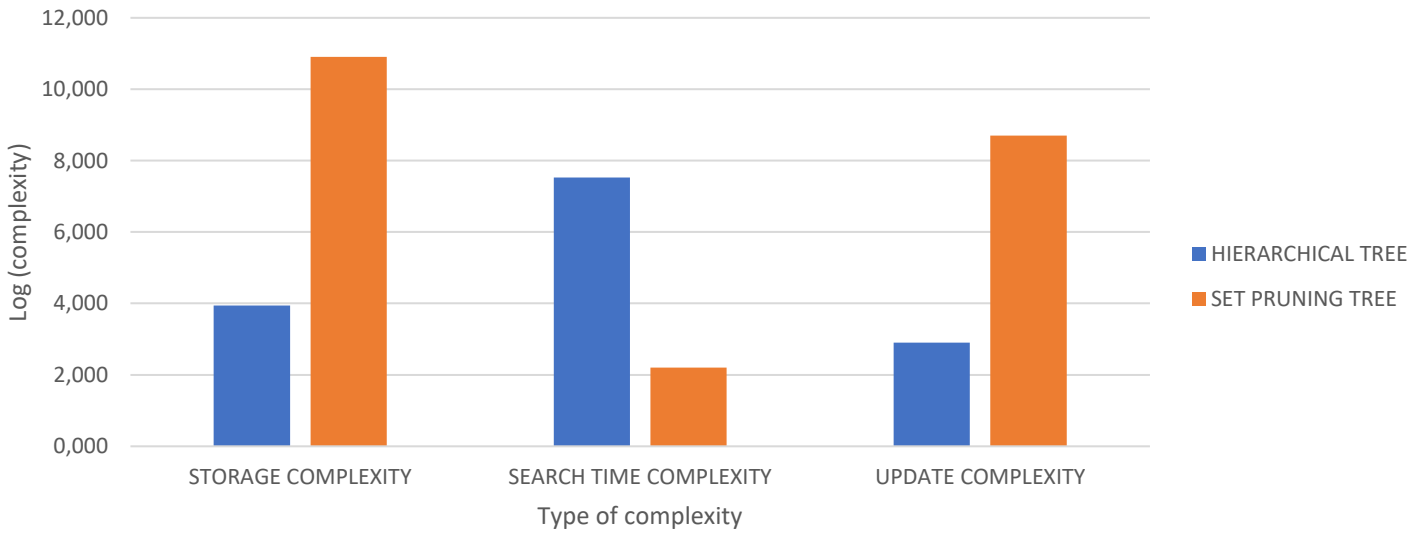
```
 ========= Packet classification--Hierarchical-Tree==========
======storage complexity========:O(8800)
======search time complexity========:O(33554432)
======update complexity========:O(800)
 ---- Packet matched rule-------- r7. Action is allow
```

SET PRUNING TREE

```
 ========= Packet classification--Set-Prunning-Tree==========
======storage complexity========:O(80525500000)
======search time complexity========:O(160)
======update complexity========:O(503284375)
 --- Packet matched rule-------- r16. Action is allow
```

Performance

PERFORMANCE	HIERARCHICAL TREE	SET PRUNING TREE
STORAGE COMPLEXITY	8800	80525500000
SEARCH TIME COMPLEXITY	33554432	160
UPDATE COMPLEXITY	800	503284375



END

Questions?