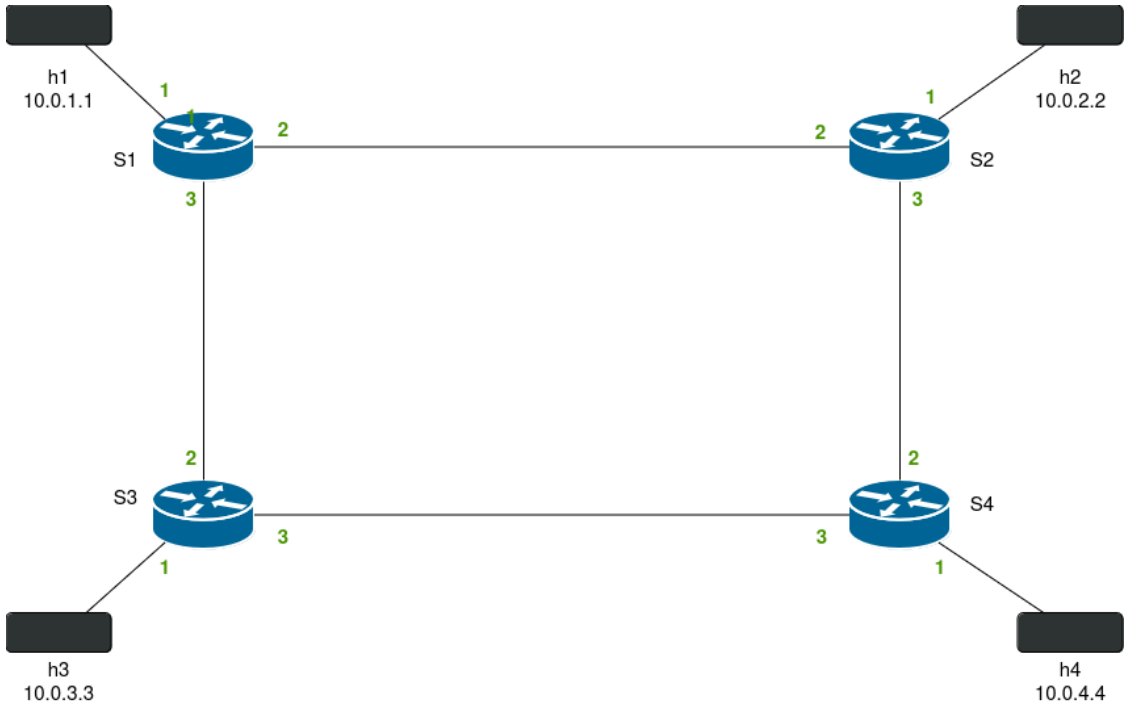
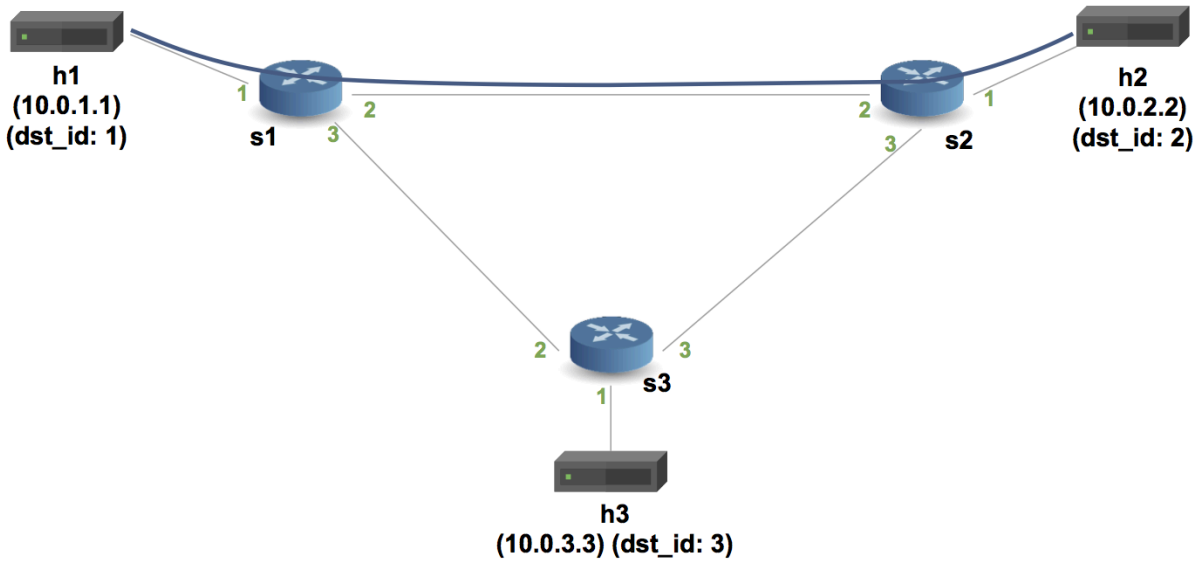
Assignment 2: Basic Tunneling in a Square Topology

Aim: Implement a square topology with hosts H1, H2, H3, H4 and switches S1, S2, S3, S4 as follows:



The original topology was as follows:



Changes made:

1. Topology definition: This states the physical connections between the nodes.
2. Table entries: Runtime table entries populated by the control plane need to also be updated, to actually send the packets at runtime

1. **Topology Definition**

   In the `topology.json` we need to

Insertions	Deletions
`s3-port3 <---> s4-port3`	`s3-port3 <---> s2-port3`
`s2-port3 <---> s4-port2`	
`s4-port1 <---> h4-eth0`	

   Changes in `topology.json`

```
{
      "hosts": {
            "h4": {
                  "ip": "10.0.4.4/24",
                  "mac": "08:00:00:00:04:44",
                  "commands": [
                        "route add default gw 10.0.4.40 dev eth0",
                        "arp -i eth0 -s 10.0.4.40 08:00:00:00:04:00"
                        ]
                 }
            },
```

```
"switches": {
       "s4": {
           "runtime_json": "s4-runtime.json"
       }
},
"links": [
       [
           "h4",
           "s4-p1"
       ],
       [
           "s3-p3",
           "s4-p3"
```

```
        ],
        [
            "s4-p2",
            "s2-p3"
        ]
    ]
}
```

2. **Table entries**

    1. **S1**

New - ipv4_lpm		
dst_IP	**dst_mac**	**port**
10.0.4.4	08:00:00:00:03:00	3

New - myTunnel_exact	
dst_id	**port**
4	3

    2. **S2**

New - ipv4_lpm		
dst_IP	**dst_mac**	**port**
10.0.4.4	08:00:00:00:04:00	3

New - myTunnel_exact	
dst_id	**port**
4	3

3. S3

New - ipv4_lpm		
dst_IP	**dst_mac**	**port**
10.0.4.4	08:00:00:00:04:00	3

New - myTunnel_exact	
dst_id	**port**
4	3

4. S4

New - ipv4_lpm		
dst_IP	**dst_mac**	**port**
10.0.1.1	08:00:00:00:03:00	3
10.0.2.2	08:00:00:00:02:00	2
10.0.3.3	08:00:00:00:03:00	3
10.0.4.4	08:00:00:00:04:44	1

New - myTunnel_exact	
dst_id	**port**
1	3
2	2
3	3
4	1

File changes

1. S1

```
  {
     "table": "MyIngress.ipv4_lpm",
     "match": {
       "hdr.ipv4.dstAddr": [
         "10.0.4.4",
         32
       ]
     },
     "action_name": "MyIngress.ipv4_forward",
     "action_params": {
       "dstAddr": "08:00:00:00:03:00",
       "port": 3
     }
   },
   {
     "table": "MyIngress.myTunnel_exact",
     "match": {
       "hdr.myTunnel.dst_id": [
         4
       ]
     },
     "action_name": "MyIngress.myTunnel_forward",
     "action_params": {
       "port": 3
     }
   }
```

2. S2

```
{
     "table": "MyIngress.ipv4_lpm",
     "match": {
       "hdr.ipv4.dstAddr": [
         "10.0.4.4",
         32
       ]
     },
```

```
    "action_name": "MyIngress.ipv4_forward",
    "action_params": {
      "dstAddr": "08:00:00:00:04:00",
      "port": 3
    }
  },
  {
    "table": "MyIngress.myTunnel_exact",
    "match": {
      "hdr.myTunnel.dst_id": [
        4
      ]
    },
    "action_name": "MyIngress.myTunnel_forward",
    "action_params": {
      "port": 3
    }
  }
```

3. S3

```
  {
    "table": "MyIngress.ipv4_lpm",
    "match": {
      "hdr.ipv4.dstAddr": [
        "10.0.4.4",
        32
      ]
    },
    "action_name": "MyIngress.ipv4_forward",
    "action_params": {
      "dstAddr": "08:00:00:00:04:00",
      "port": 3
    }
  },
  {
    "table": "MyIngress.myTunnel_exact",
    "match": {
      "hdr.myTunnel.dst_id": [
```

```
        4
      ]
    },
    "action_name": "MyIngress.myTunnel_forward",
    "action_params": {
      "port": 3
    }
  }
```

4. S4

```
{
 "target": "bmv2",
 "p4info": "build/basic_tunnel.p4.p4info.txt",
 "bmv2_json": "build/basic_tunnel.json",
 "table_entries": [
   {
     "table": "MyIngress.ipv4_lpm",
     "match": {
       "hdr.ipv4.dstAddr": [
         "10.0.1.1",
         32
       ]
     },
     "action_name": "MyIngress.ipv4_forward",
     "action_params": {
       "dstAddr": "08:00:00:00:03:00",
       "port": 3
     }
   },
   {
     "table": "MyIngress.ipv4_lpm",
     "match": {
       "hdr.ipv4.dstAddr": [
         "10.0.2.2",
         32
       ]
     },
     "action_name": "MyIngress.ipv4_forward",
     "action_params": {
```

```
      "dstAddr": "08:00:00:00:02:00",
      "port": 2
    }
  },
  {
    "table": "MyIngress.ipv4_lpm",
    "match": {
      "hdr.ipv4.dstAddr": [
        "10.0.3.3",
        32
      ]
    },
    "action_name": "MyIngress.ipv4_forward",
    "action_params": {
      "dstAddr": "08:00:00:00:03:00",
      "port": 3
    }
  },
  {
    "table": "MyIngress.myTunnel_exact",
    "match": {
      "hdr.myTunnel.dst_id": [
        1
      ]
    },
    "action_name": "MyIngress.myTunnel_forward",
    "action_params": {
      "port": 3
    }
  },
  {
    "table": "MyIngress.myTunnel_exact",
    "match": {
      "hdr.myTunnel.dst_id": [
        2
      ]
    },
    "action_name": "MyIngress.myTunnel_forward",
    "action_params": {
```

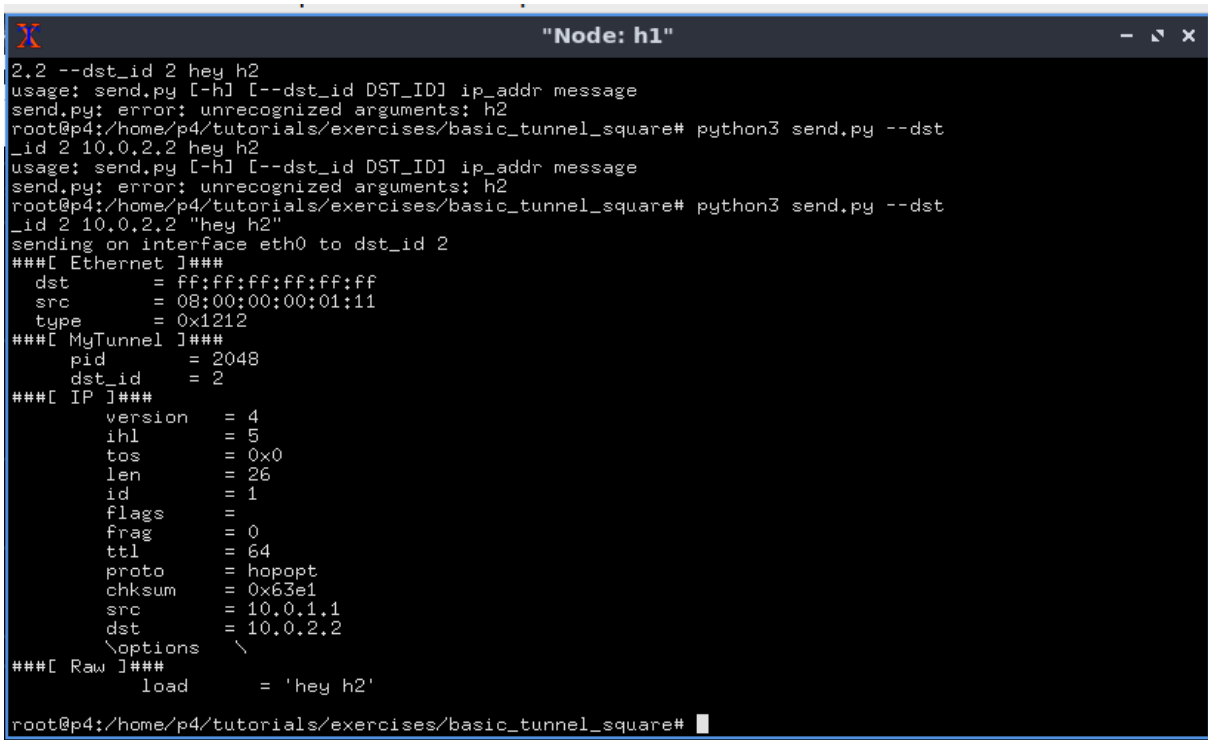
```
      "port": 2
    }
  },
  {
    "table": "MyIngress.myTunnel_exact",
    "match": {
      "hdr.myTunnel.dst_id": [
        3
      ]
    },
    "action_name": "MyIngress.myTunnel_forward",
    "action_params": {
      "port": 3
    }
  },


  {
    "table": "MyIngress.ipv4_lpm",
    "match": {
      "hdr.ipv4.dstAddr": [
        "10.0.4.4",
        32
      ]
    },
    "action_name": "MyIngress.ipv4_forward",
    "action_params": {
      "dstAddr": "08:00:00:00:04:44",
      "port": 1
    }
  },
  {
    "table": "MyIngress.myTunnel_exact",
    "match": {
      "hdr.myTunnel.dst_id": [
        4
      ]
    },
    "action_name": "MyIngress.myTunnel_forward",
```

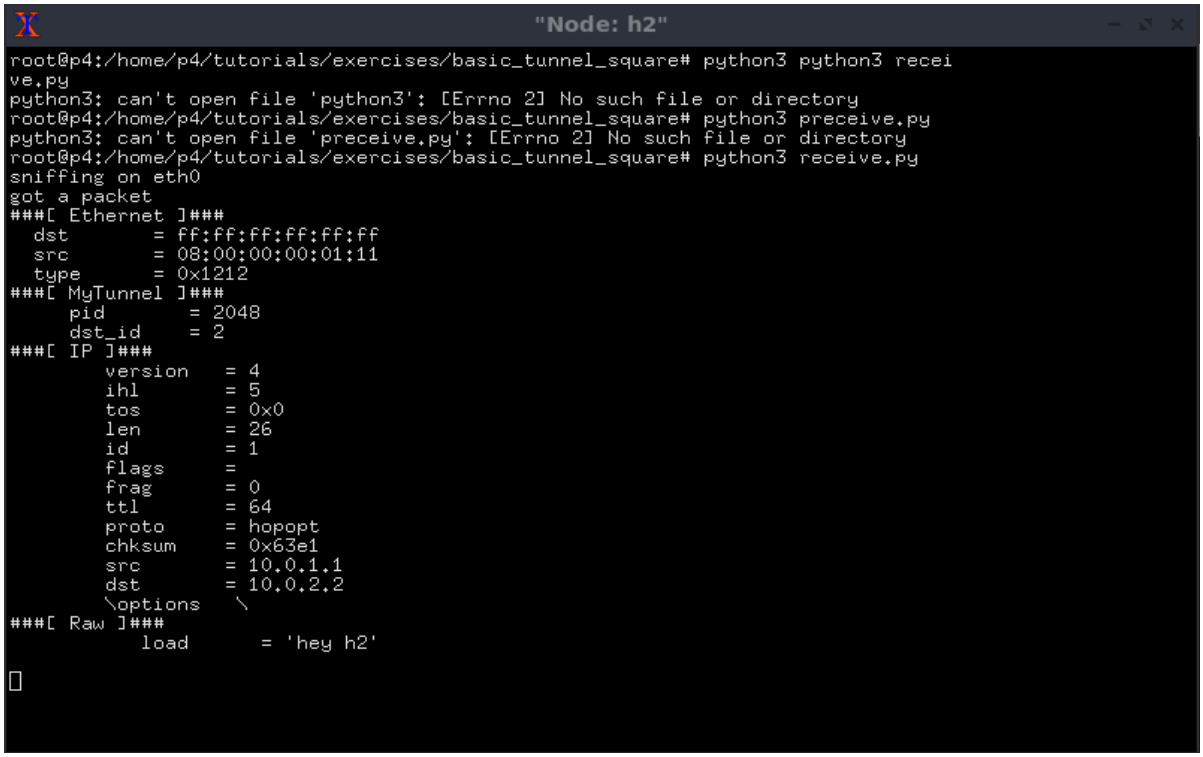
```
      "action_params": {
        "port": 1
      }
    }
  ]
}
```

Running the simulation

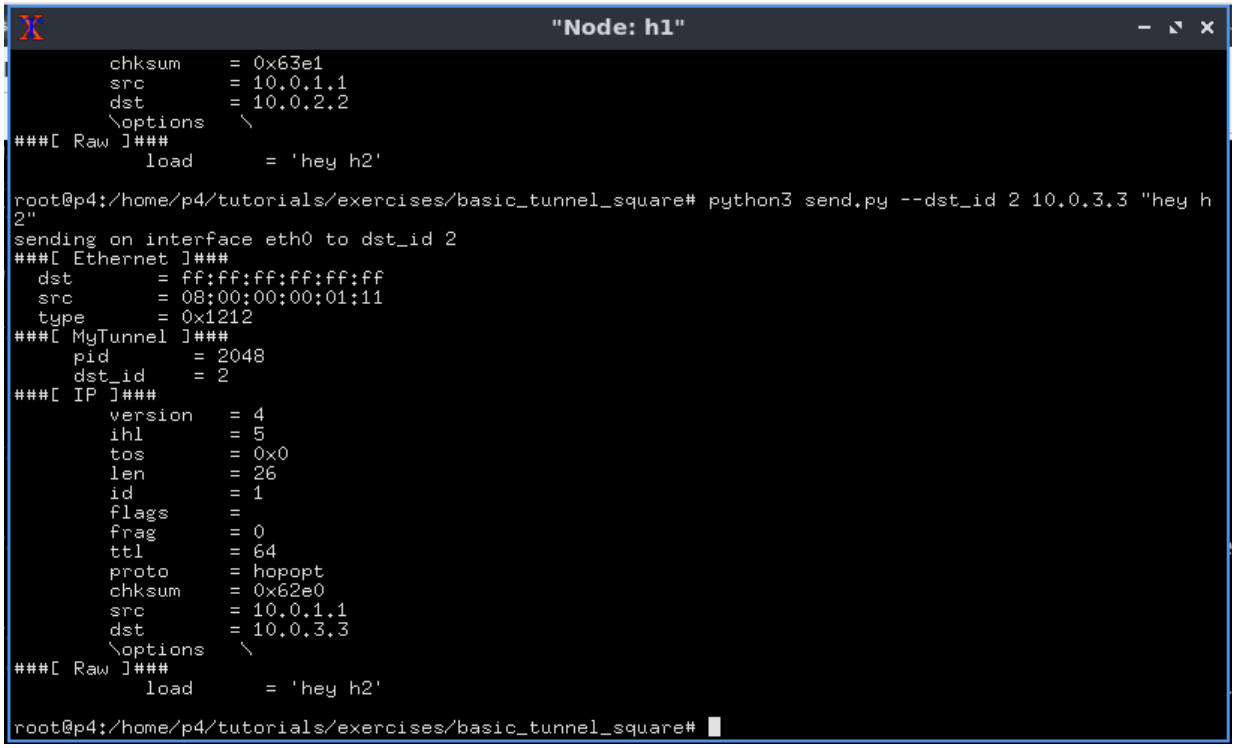
```
mininet> links
eth0<->s1-eth1 (OK OK)
eth0<->s2-eth1 (OK OK)
eth0<->s3-eth1 (OK OK)
eth0<->s4-eth1 (OK OK)
s1-eth2<->s2-eth2 (OK OK)
s1-eth3<->s3-eth2 (OK OK)
s2-eth3<->s4-eth2 (OK OK)
s3-eth3<->s4-eth3 (OK OK)
mininet> pingall
*** Ping: testing ping reachability
h1 -> h2 h3 h4
h2 -> h1 h3 h4
h3 -> h1 h2 h4
h4 -> h1 h2 h3
*** Results: 0% dropped (12/12 received)
mininet> xterm h1 h2 h3 h4
mininet>
```

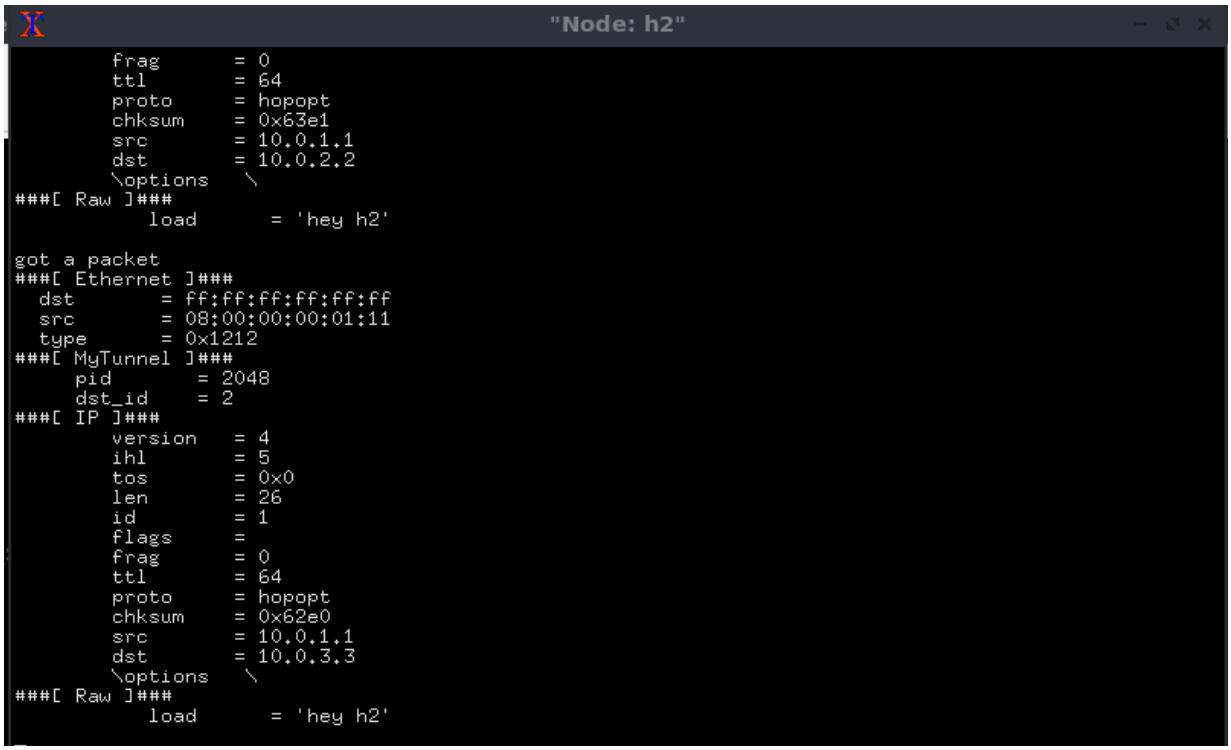
Updated links and pingall





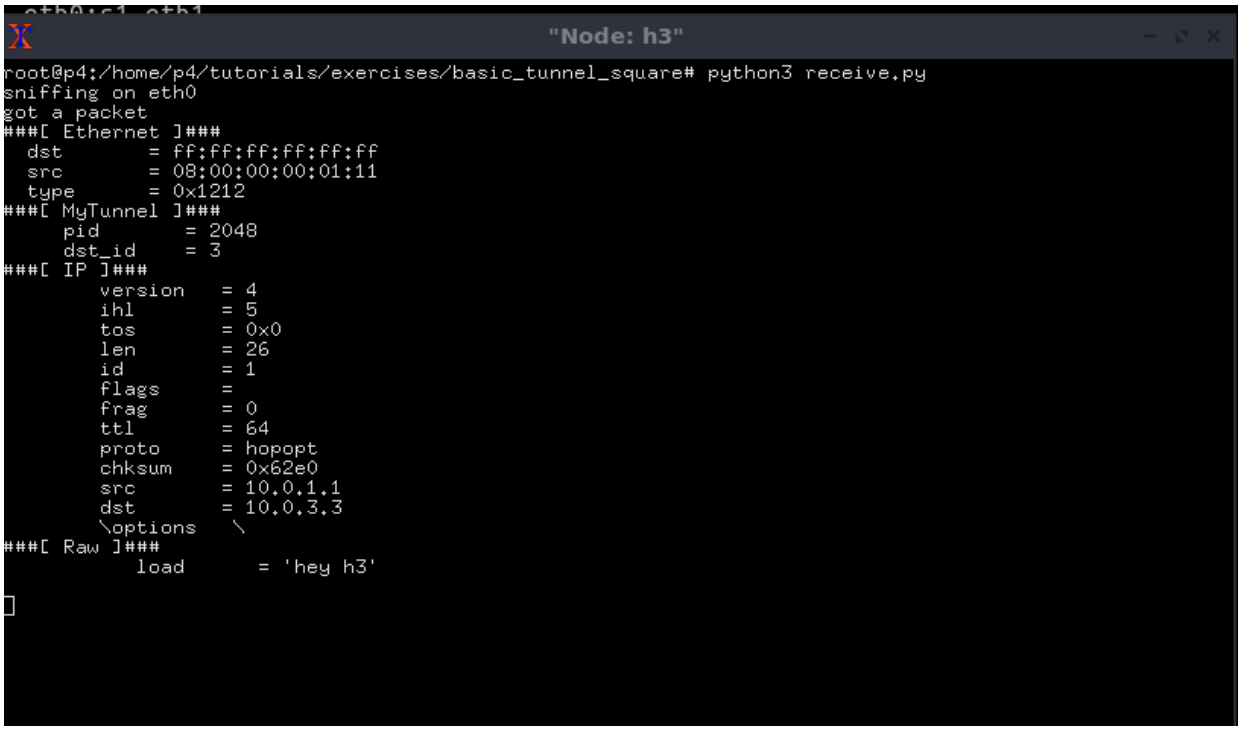
Sending packet from H1 to H2

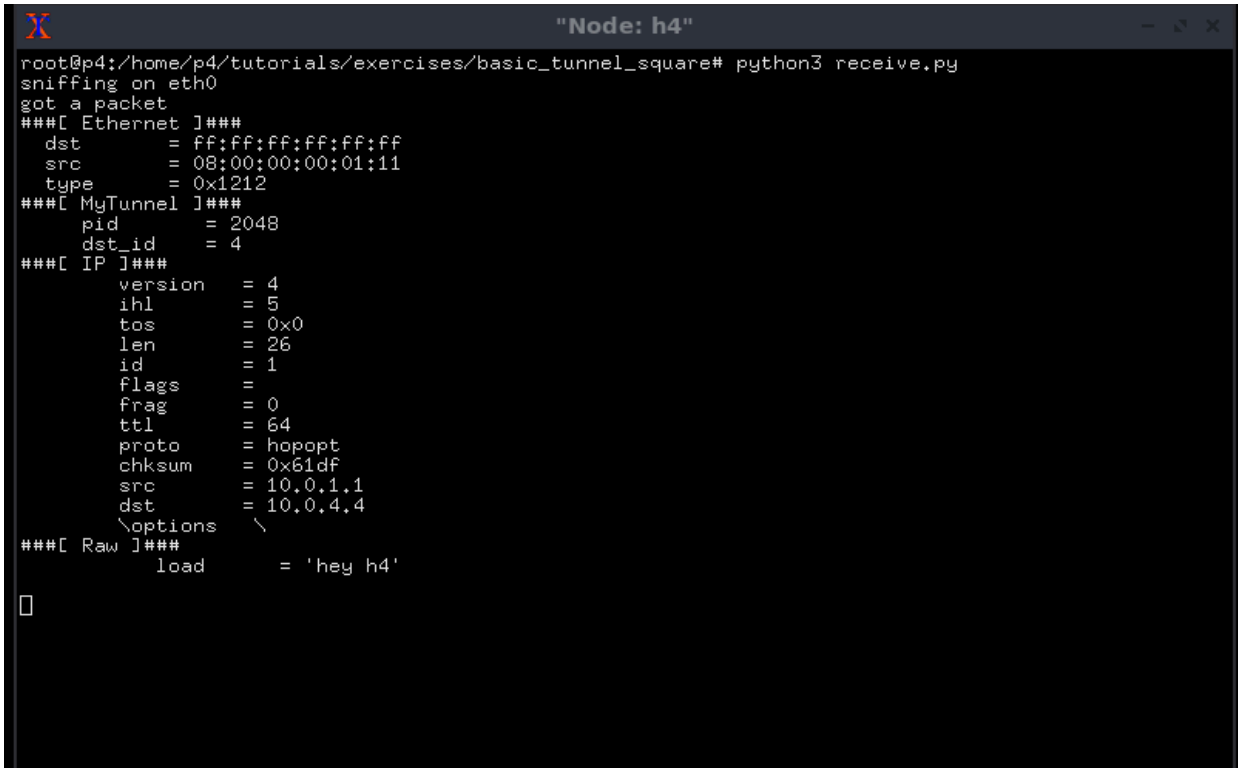




Sending packet from H1 to H2, using **`dst_id=2`** *and* **`dst_ip=10.0.3.3`**. *Meaning network is using the new myTunnel header for forwarding and not IP*


"Node: h1"
chksum = 0x62e0
src = 10.0.1.1
dst = 10.0.3.3
\options \
###[Raw]###
load = 'hey h3'
root@p4:/home/p4/tutorials/exercises/basic_tunnel_square# python3 send.py --dst_id 4 10.0.4.4 "hey h 4"
sending on interface eth0 to dst_id 4
###[Ethernet]###
dst = ff:ff:ff:ff:ff:ff
src = 08:00:00:00:01:11
type = 0x1212
###[MyTunnel]###
pid = 2048
dst_id = 4
###[IP]###
version = 4
ihl = 5
tos = 0x0
len = 26
id = 1
flags =
frag = 0
ttl = 64
proto = hopopt
chksum = 0x61df
src = 10.0.1.1
dst = 10.0.4.4
\options \
###[Raw]###
load = 'hey h4'
root@p4:/home/p4/tutorials/exercises/basic_tunnel_square#


"Node: h3"
root@p4:/home/p4/tutorials/exercises/basic_tunnel_square# python3 receive.py
sniffing on eth0
got a packet
###[Ethernet]###
dst = ff:ff:ff:ff:ff:ff
src = 08:00:00:00:01:11
type = 0x1212
###[MyTunnel]###
pid = 2048
dst_id = 3
###[IP]###
version = 4
ihl = 5
tos = 0x0
len = 26
id = 1
flags =
frag = 0
ttl = 64
proto = hopopt
chksum = 0x62e0
src = 10.0.1.1
dst = 10.0.3.3
\options \
###[Raw]###
load = 'hey h3'



Sending packets from H1 to H3, H4