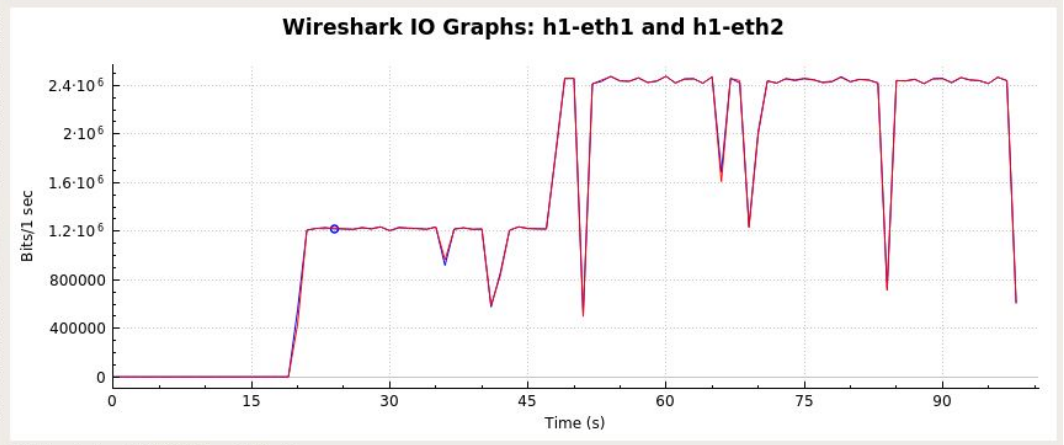
Assignment 1 : MPTCP

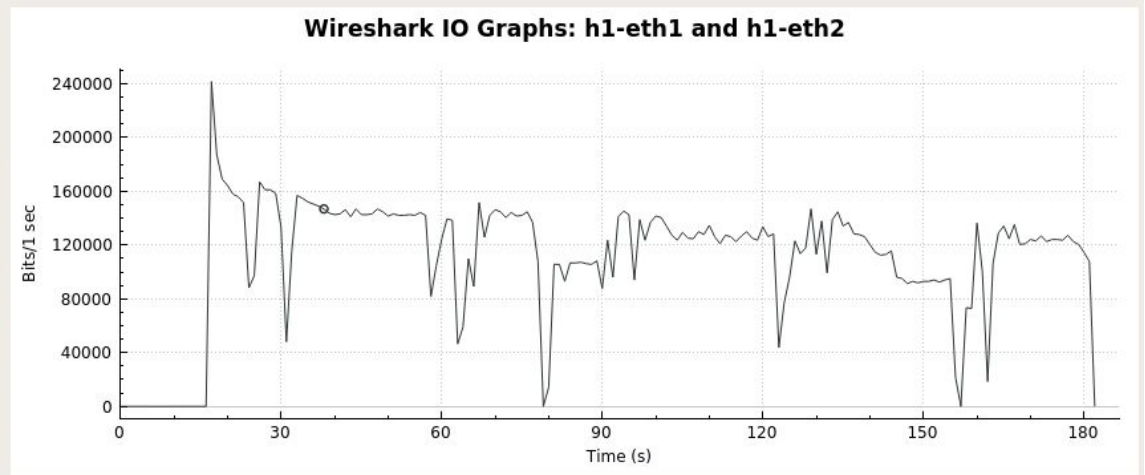
Submitted By: Abhishek Ayachit (862188073)
ACN : Assignment 1 : MPTCP
Date: 05/02/2020
Collaboration with: Sudip Bala(862188812), Lovepreet Singh Dhaliwal(862188799) (only for the Q2. MPTCP enabled related doubts)

Q1. Does MPTCP improve file transfer performance?
Ans:

a) MPTCP enabled:



b) MPTCP disabled:



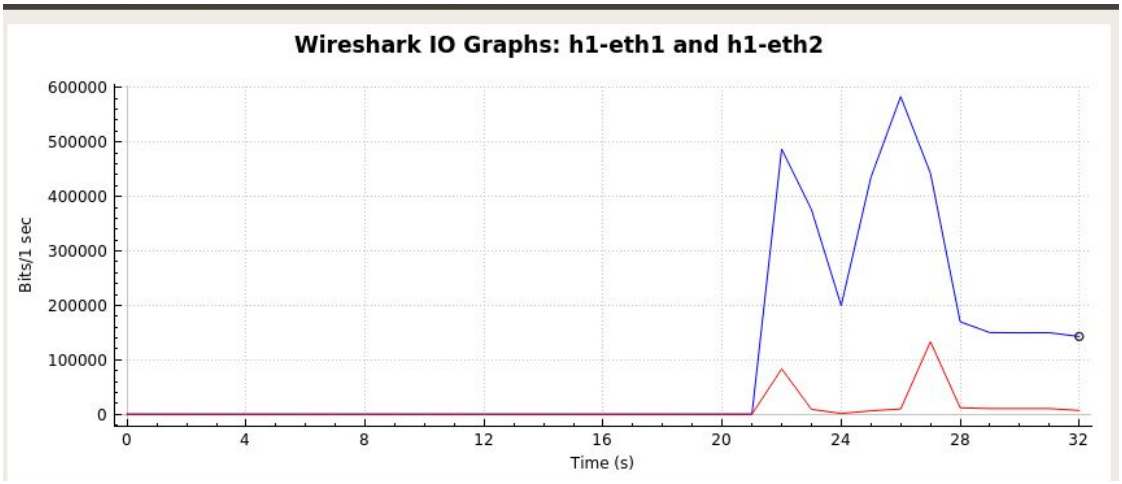
I. Does MPTCP increase the total throughput or decrease the total download time?
Time Taken(MPTCP enabled) : 1min 17.825s
Time Taken(MPTCP disabled) : 2min 45.095s

Yes. MPTCP does increase the total throughput or decrease the total download time.

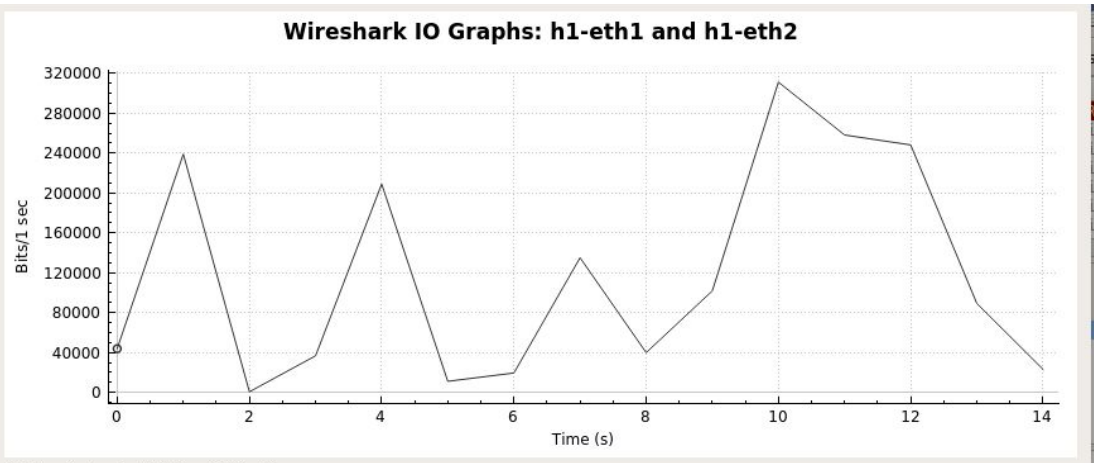
II. Does this make sense, why or why not?
MPTCP does decrease the download time because due to the given bandwidth, the source gets two different paths with the same bandwidth cutting the time in half.

Q2. Does MPTCP improve (approximated) web browsing performance?
Ans:

a) MPTCP enabled:



b) MPTCP disabled:



	Wget Time	Fraction of objects on path 1
MPTCP Enabled	10.631s	~99%
MPTCP Disabled	14.607s	100%

I. Does MPTCP help reduce the page load time?
From the above table, it is clear that MPTCP does help reduce the page load time.

II. What fraction of objects are requested over each path?
Only about 1% of the traffic takes path 2 and about 99% traffic takes path 1.

III. Does this make sense, why or why not?
Only a small fraction of the amount takes path 2 to reach the destination. The reason may be that the file is small and there might not be congestion in path 1.

IV. Is *wget* a reasonable approximation of a web browser, or would you expect a real web browser to have different performance?
The server sees wget and the browser request as the same because the header is almost similar. The main difference would be the browser GUI which will take most of the bandwidth. Therefore, wget does the same in headless mode.

Appendix

```
from mininet.topo import Topo
from mininet.net import Mininet
from mininet.node import Node
from mininet.log import setLogLevel, info
from mininet.cli import CLI


class LinuxRouter( Node ):
        "A Node with IP forwarding enabled."

        def config( self, **params ):
        super( LinuxRouter, self).config( **params )
        # Enable forwarding on the router
        self.cmd( 'sysctl net.ipv4.ip_forward=1' )

        def terminate( self ):
        self.cmd( 'sysctl net.ipv4.ip_forward=0' )
        super( LinuxRouter, self ).terminate()


class NetworkTopo( Topo ):
        "A LinuxRouter connecting three IP subnets"
```

```
        def build( self, **_opts ):


        router0 = self.addNode( 'r0', cls=LinuxRouter, ip='10.0.0.1/24' )
        router1 = self.addNode( 'r1', cls=LinuxRouter, ip='10.0.1.1/24' )

        s1, s2, s3, s4 = [ self.addSwitch( s ) for s in ( 's1', 's2', 's3', 's4' ) ]

        #Routers
        self.addLink( s1, router0, intfName2='r0-eth1',
                params2={ 'ip' : '10.0.0.1/24'} )  # for clarity
        self.addLink( s2, router0, intfName2='r0-eth2',
                params2={ 'ip' : '10.0.2.1/24' } )

        self.addLink( s3, router1, intfName2='r1-eth1',
                params2={ 'ip' : '10.0.1.1/24'} )
         self.addLink( s4, router1, intfName2='r1-eth2',
                params2={ 'ip' : '10.0.3.1/24' } )

        h1 = self.addHost( 'h1')
        h2 = self.addHost( 'h2')

         #Hosts

        self.addLink( s1, h1, intfName2='h1-eth1',
                params2={ 'ip' : '10.0.0.100/24' } )  # for clarity
        self.addLink( s2, h2, intfName2='h2-eth1',
                params2={ 'ip' : '10.0.2.100/24' } )  # for clarity

        self.addLink( s3, h1, intfName2='h1-eth2',
                params2={ 'ip' : '10.0.1.100/24' } )  # for clarity
         self.addLink( s4, h2, intfName2='h2-eth2',
                params2={ 'ip' : '10.0.3.100/24' } )  # for clarity


def run():
        "Test linux router"
        topo = NetworkTopo()
        net = Mininet( topo=topo )
        net.start()
        info( '*** Routing Table on Router:\n' )
        info( net[ 'r0' ].cmd( 'route' ) )
```

```
        info( net[ 'r1' ].cmd( 'route' ) )

        net['h1'].setIP('10.0.0.100/24', intf='h1-eth1')
        net['h1'].setIP('10.0.1.100/24', intf='h1-eth2')
        net['h2'].setIP('10.0.2.100/24', intf='h2-eth1')
        net['h2'].setIP('10.0.3.100/24', intf='h2-eth2')


        net['h1'].cmd('tc qdisc add dev h1-eth1 root tbf rate 50mbit burst 1mbit
latency 1ms')
        net['h1'].cmd('tc qdisc add dev h1-eth2 root tbf rate 50mbit burst 1mbit
latency 1ms')
        net['h2'].cmd('tc qdisc add dev h2-eth1 root tbf rate 50mbit burst 1mbit
latency 1ms')
        net['h2'].cmd('tc qdisc add dev h2-eth2 root tbf rate 50mbit burst 1mbit
latency 1ms')
        net['r0'].cmd('tc qdisc add dev r0-eth1 root tbf rate 50mbit burst 1mbit latency
1ms')
        net['r0'].cmd('tc qdisc add dev r0-eth2 root tbf rate 50mbit burst 1mbit latency
1ms')
        net['r1'].cmd('tc qdisc add dev r1-eth1 root tbf rate 50mbit burst 1mbit latency
1ms')
        net['r1'].cmd('tc qdisc add dev r1-eth2 root tbf rate 50mbit burst 1mbit latency
1ms')

        #h1:
        net['h1'].cmd("ip rule add from 10.0.0.100 table 1")
  net['h1'].cmd("ip rule add from 10.0.1.100 table 2")

  net['h1'].cmd('ip route add 10.0.0.0/24 dev h1-eth1 scope link table 1')
    net['h1'].cmd('ip route add default via 10.0.0.1 dev h1-eth1 table 1')

    net['h1'].cmd('ip route add 10.0.1.0/24 dev h1-eth2 scope link table 2')
    net['h1'].cmd('ip route add default via 10.0.1.1 dev h1-eth2 table 2')

 # default route for the selection process of normal internet-traffic
    net['h1'].cmd('ip route add default scope global nexthop via 10.0.0.1 dev h1-eth1')


        #h2:
        net['h2'].cmd("ip rule add from 10.0.2.100 table 1")
        net['h2'].cmd("ip rule add from 10.0.3.100 table 2")
```

```
        net['h2'].cmd('ip route add 10.0.2.0/24 dev h2-eth1 scope link table 1')
        net['h2'].cmd('ip route add default via 10.0.2.1 dev h2-eth1 table 1')

        net['h2'].cmd('ip route add 10.0.3.0/24 dev h2-eth2 scope link table 2')
        net['h2'].cmd('ip route add default via 10.0.3.1 dev h2-eth2 table 2')

  # default route for the selection process of normal internet-traffic
        net['h1'].cmd('ip route add default scope global nexthop via 10.0.2.1 dev
h2-eth1')

        CLI( net )
        net.stop()

if __name__ == '__main__':
        setLogLevel( 'info' )
        run()
```