

Tp8.

2. ☐ TCP congestion control a/ teknik untuk mencegah congestion atau membantu menyelesaikan congestion yg telah terjadi.
3. ☐ slow start : TCP bekerja dengan mengobservasi laju paket yg dilepaskan ke dlm network dg cara memonitor laju acknowledgment yang dikembalikan. slow start menambahkan sebuah windows pada penerima TCP.
- ☐ Fast Retransmit : TCP bekerja dg menduplikasi ack disebabkan karena tidak diterimanya paket oleh penerima. Penerima mengaggap packet loss atau terhapus. Jika terjadi 3/ lebih duplikasi ack diterima maka packet loss. Pengirim segera melakukan retransmisi tanpa menunggu coarse-grai.
- ☐ Fast Recovery : Pada saat fast retransmission dilakukan, threshold direct setengah dari cwnd, kemudian cwnd dicet ke threshold ditambah tiga paket.
4. ☐ TCP Reno : Penelusuran varian TCP lama yang dilakukan perbaikan diatas dengan mempertahankan semua fitur utama, yaitu slow start, congestion avoidance, fast retransmit dan fast recovery
1. ☐ i) klik link petunjuk di informatic lab
- ☐ 2) akan didapatkan link tinyurl.com /TpModul8
- ☐ 3) setelah di klik akan ada tampilan " ini teh ctf ? "
- ☐ a) ada link tinyurl lagi, lalu klik
- ☐ s) akan ada beberapa clue
- ☐ b) Balik lagi ke tampilan " ini teh ctf ? " lalu buka source dengan cara f12
- ☐ 7) setelah itu didapatkan "jarkom_2019 {tp-mod-8-y3 qy!!!}

Cara buka mininet.

Date:

☒ 1) Buka terminal

☐ 2) Sudo su

☐ 3) Masukkan password

☐ 4) ketik : nautilus

☐ 5) buka komputer

☐ 6) buka home

☐ 7) buka ardra

☐ 8) buka folder mininet

☐ 9) buka folder custom

☐ 10) Copy ~~topo~~ topo-zsw-zhost.py lalu rename.

☐ 11) tulis code sesuai soal tp.

mengexec

1) sudo su

2) ~~python~~ cd mininet /usr

3) python x tp8.py

kelewat 😊 →

```

root@ardhia-VirtualBox:/home/ardhia/mininet/custom_mn --custom tp8.py --topo ja
rkcom --switch ovsk --ovsdpd ovsdpd --ovsdb ovsdb
***
***
Unab
Unab
Sett
***
H1 H1
***
S1 S1
***
(75.00)
100.00
***
H1 H1
***
c8
***
S1 S1
.00KB/s
*** Starting CLI:
mininet> pingall

```

```

mininet> pingall
*** Ping: testing ping reachability
H1
H2
H3
*** Results: 100% dropped (0/5 received)

```

• Mn --custom tp8.py --tp jarkom --switch ovsk --
controller = remotr

Perintah itu y/ membangun topologi lalu pingall mengetest
semua paket blm ke drop karena flow table belum dibuat
maka berikan xterm s1 s2 s3 s4, lalu berikan perintah
spt disamping maka ketika di ping paket akan terdrop.

```

#tp8.py x
from mininet.topo import Topo
from mininet.net import Mininet
from mininet.link import TCLink, Link
class module(Topo):
    def __init__(self):
        Topo.__init__(self)
        H1 = self.addHost('H1', mac = '74:10:82:00:00:01', ip = '30.30.1.20/24')
        #menambahkan device host dengan nama H1, dan mengisi mac dg 5 digit nm terakhir, dan mengisi ip
        H2 = self.addHost('H2', mac = '74:10:82:00:00:02', ip = '30.30.1.21/24')
        #menambahkan device host dengan nama H1, dan mengisi mac dg 5 digit nm terakhir, dan mengisi ip
        H3 = self.addHost('H3', mac = '74:10:82:00:00:03', ip = '30.30.1.22/24')
        #menambahkan device host dengan nama H1, dan mengisi mac dg 5 digit nm terakhir, dan mengisi ip

        S1 = self.addSwitch('S1', mac = '74:10:80:00:00:01')
        #menambahkan switch dg variable s1, dan mac dg 5 digit terakhir
        S2 = self.addSwitch('S2', mac = '74:10:80:00:00:02')
        #menambahkan switch dg variable s1, dan mac dg 5 digit terakhir
        S3 = self.addSwitch('S3', mac = '74:10:80:00:00:03')
        #menambahkan switch dg variable s1, dan mac dg 5 digit terakhir
        S4 = self.addSwitch('S4', mac = '74:10:80:00:00:04')
        #menambahkan switch dg variable s1, dan mac dg 5 digit terakhir

        self.addLink(S1, S2, cls=TCLink, bw=75)
        #menghubungkan device s1 dan s2 dg bandwidth 75 mbps
        self.addLink(S1, S3, cls=TCLink, bw=50)
        #menghubungkan device s1 dan s2 dg bandwidth 50 mbps
        self.addLink(S1, S4, cls=TCLink, bw=100)
        #menghubungkan device s1 dan s2 dg bandwidth 100 mbps
        self.addLink(S2, H1, )
        #menghubungkan device s2 dengan H1
        self.addLink(S3, H2, )
        #menghubungkan device s3 dengan H2
        self.addLink(S4, H3, )
        #menghubungkan device s4 dengan H3
        topos = {'jarkom': (lambda : module())}

```