# DAFTAR PUSTAKA


**Alwayn**, **Vivek.** *Advanced MPLS Design and Implementation*, Cisco Press, Indianapolis, USA, 2002

**Armitage**, **G.** *Quality of Service in IP Networks*, Morgan Kauffman, RFC 3031, Januari 2001

**Davie, Bruce and Rekhter**, **Yakov.** *MPLS Technology and Applications*, Morgan Kauffman, San Fransisco, CA, 2000

**G, Marc.** *Tutorial for Network Simulator*, http://www.isi.edu/nsnam/ns/ns-documentation

**Jae, C. and G, Mark.** *NS by Example*, Worcester Polytechnic Institute, 2002

**Kevin, F and Kannan, V.** *The NS Manual*, http://www.isi.edu/ns/nsnam/ns/ns-documentation

**M. Thomas II, Thomas.** *Juniper Networks Reference Guide*, Addison-Wesley, Boston, 2002

**Osborne, Eric and Simha, Ajay.** *Traffic Engineering with MPLS*, Cisco Press Indianapolis, USA, 2002

**Purbo, Onno W.** *Desain, Implementasi TCP/IP*, Elex Media Komputindo, 1996

***Source code* Simulasi**

```
set ns [new Simulator]
set f0 [open video.tr w]
set f1 [open audio.tr w]
set f2 [open image.tr w]
exec rm -rf /tmp/rio
exec mkdir /tmp/rio

set nf_nam [ open /tmp/rio/tcpip.nam w]
$ns namtrace-all $nf_nam
set nf [ open /tmp/rio/tcpip.tr w]
$ns trace-all $nf
proc finish {} {
                global ns nf f0 f1 f2
                $ns flush-trace
                close $nf
                close $f0
                close $f1
                close $f2
                #exec nam tcpip.nam &
                    exec xgraph video.tr -geometry
800X600 -x time(s) -y throughput(Mb/s) &
```

```
                    exec xgraph audio.tr -geometry
800X600 -x time(s) -y throughput(Mb/s) &
                    exec xgraph image.tr -geometry
800X600 -x time(s) -y throughput(Mb/s) &
                exec xgraph -m video.tr audio.tr
image.tr -geometry 800X600 -x time(s) -y
throughput(Mb/s) &
                exit 0
                }
    set Node0 [$ns node]
    set Node1 [$ns node]
    set LSR2 [$ns node]
    set LSR3 [$ns node]
    set LSR4 [$ns node]
    set LSR5 [$ns node]
    set LSR6 [$ns node]
    set LSR7 [$ns node]
    set LSR8 [$ns node]
    set Node9 [$ns node]
    set Node10 [$ns node]
    set LSR11 [$ns node]
    set LSR12 [$ns node]
```

```
set Node13 [$ns node]
set Node14 [$ns node]

$ns rtproto LS
$ns duplex-link $Node0 $LSR2 1.5Mb 10ms DropTail
$ns duplex-link $Node1 $LSR3 1.5Mb 10ms DropTail
$ns duplex-link $Node10 $LSR6 1.5Mb 10ms DropTail
$ns duplex-link $LSR2 $LSR6 2Mb 10ms DropTail
$ns duplex-link $LSR6 $LSR4 2Mb 10ms DropTail
$ns duplex-link $LSR6 $LSR7 1.5Mb 10ms DropTail
$ns duplex-link $LSR6 $LSR12 1.5Mb 10ms DropTail
$ns duplex-link $LSR12 $LSR11 1.5Mb 10ms DropTail
$ns duplex-link $LSR12 $LSR8 1.5Mb 10ms DropTail
$ns duplex-link $LSR12 $LSR7 2Mb 10ms DropTail
$ns duplex-link $LSR12 $Node14 1.5Mb 10ms DropTail
$ns duplex-link $LSR11 $LSR8 1.5Mb 10ms DropTail
$ns duplex-link $LSR11 $Node9 1.5Mb 10ms DropTail
$ns duplex-link $LSR8 $LSR7 1.5Mb 10ms DropTail
$ns duplex-link $LSR8 $Node13 1.5Mb 10ms DropTail
$ns duplex-link $LSR8 $LSR5 2Mb 10ms DropTail
$ns duplex-link $LSR5 $LSR4 1.5Mb 10ms DropTail
$ns duplex-link $LSR5 $LSR3 1.5Mb 10ms DropTail
```

```
$ns duplex-link $LSR3 $LSR2 1.5Mb 10ms DropTail
$ns duplex-link $LSR3 $LSR4 1.5Mb 10ms DropTail
$ns duplex-link $LSR2 $LSR4 1.5Mb 10ms DropTail
$ns duplex-link $LSR4 $LSR7 1.5Mb 10ms DropTail
$ns duplex-link-op $Node0 $LSR2 orient right
$ns duplex-link-op $LSR2 $LSR6 orient right
$ns duplex-link-op $LSR1 $LSR3 orient right
$ns duplex-link-op $LSR6 $LSR12 orient right-down
$ns duplex-link-op $LSR6 $LSR7 orient down
$ns duplex-link-op $LSR6 $LSR4 orient left-down
$ns duplex-link-op $LSR12 $Node14 orient right
$ns duplex-link-op $LSR12 $LSR11 orient down
$ns duplex-link-op $LSR12 $LSR8 orient left-down
$ns duplex-link-op $LSR8 $Node13 orient down
$ns duplex-link-op $Node10 $LSR6 orient down
$ns duplex-link-op $LSR2 $LSR4 orient right-down
$ns duplex-link-op $LSR7 $LSR12 orient right
$ns duplex-link-op $LSR7 $LSR4 orient left
$ns duplex-link-op $LSR7 $LSR8 orient down
$ns duplex-link-op $LSR11 $LSR8 orient left
$ns duplex-link-op $LSR11 $Node9 orient right-down
$ns duplex-link-op $LSR4 $LSR3 orient left-down
```

```
$ns duplex-link-op $LSR4 $LSR5 orient down
$ns duplex-link-op $LSR5 $LSR3 orient left
$ns duplex-link-op $LSR5 $LSR8 orient right
$ns duplex-link-op $LSR2 $LSR3 orient down

$ns use-scheduler List
proc attach-expoo-traffic {Node sink size burst
idle rate} {
global ns
set udp [new Agent/UDP]
$ns attach-agent $Node $udp
set traffic [new Application/Traffic/CBR]
$traffic set packetSize_ $size
$traffic set burst_time_ $burst
$traffic set idle_time_ $idle
$traffic set rate_ $rate
$traffic attach-agent $udp
$ns connect $udp $sink
return $traffic
}
set sink0 [new Agent/LossMonitor]
$ns attach-agent $Node9 $sink0
```

```
set sink1 [new Agent/LossMonitor]
$ns attach-agent $Node13 $sink1
set sink2 [new Agent/LossMonitor]
$ns attach-agent $Node14 $sink2
set src0 [attach-expoo-traffic $Node0 $sink0 200 0
0 1.28Mb]
set src1 [attach-expoo-traffic $Node10 $sink1 200
0 0 1.28Mb]
set src2 [attach-expoo-traffic $Node1 $sink2 200 0
0 1.28Mb]
set tcp [new Agent/TCP]
$ns attach-agent $Node1 $tcp
set ftp [new Application/FTP]
$tcp set packetSize_ 1024
set sink [new Agent/TCPSink]
$ns attach-agent $Node14 $sink
$ns connect $tcp $sink
$ftp attach-agent $tcp
```

```
proc monitor {} {
    global tcp
    $tcp instvar ndatapack_
    puts "paket dikirim : $ndatapack_"
    $tcp instvar nackpack_
    puts "paket diterima : $nackpack_"
}
proc record {} {
    global sink0 sink1 sink2 f0 f1 f2 totalpkt
totalpkt1 totalpkt2 rcnt tcp
    set ns [Simulator instance]
set time 0.05
set bw0 [$sink0 set bytes_]
    set bw1 [$sink1 set bytes_]
    set bw2 [$sink2 set bytes_]
    set now [$ns now]
puts $f0 "$now [expr $bw0/$time*8/1000000]"
    puts $f1 "$now [expr $bw1/$time*8/1000000]"
    puts $f2 "$now [expr $bw2/$time*8/1000000]"
    $sink0 set bytes_ 0
    $sink1 set bytes_ 0
    $sink2 set bytes_ 0
```

```
        $ns at [expr $now+$time] "record"
        set bw0 [expr $bw0/200 ]
        set bw1 [expr $bw1/200 ]
        set bw2 [expr $bw2/200 ]
        set totalpkt [$sink0 set npkts_]
        set totalpkt1 [$sink1 set npkts_]
        set totalpkt2 [$sink2 set npkts_]
        incr rcnt 1
        }
    proc recv-pkts {} {
        global totalpkt totalpkt1 totalpkt2 rcnt
        flush stdout
        puts "Total paket yang diterima dari Node0
adalah $totalpkt"
        puts "Total paket yang diterima dari Node10
adalah $totalpkt1"
        puts "Total paket yang diterima dari Node1
adalah $totalpkt2"
        puts "Recording count adalah $rcnt"
        }
```

```
    $ns at 0.0 "record"
    $ns at 0.0 "$src0 start"
    $ns at 0.0 "$src2 start"
    $ns at 0.1 "$src1 start"
    $ns at 1.5 "$src1 stop"
    $ns at 2.0 "$src2 stop"
    $ns at 2.5 "$src0 stop"
    $ns at 2.9 "recv-pkts"
    $ns at 3.0 "finish"
$ns run
```