**PROGRAM 6**

set opt(title) zero;

set opt(stop) 100;

set opt(adaptive) 1;

set opt(ecn) 0;

set opt(secondDelay) 55;

set opt(minth) 30;

set opt(maxth) 0;

set opt(flows) 0;

set opt(wrap) 100;

set opt(quiet) 0;

set opt(type) umts;

set opt(web) 2;

set opt(window) 30;

set opt(srcTrace) is;

set opt(dstTrace) bs2;

set opt(umtsbuf) 10;

set bwDL(umts) 384000

set bwUL(umts) 64000

set propDL(umts) .150

set propUL(umts) .150

set buf(umts) 20

set ns [new Simulator]

set tf [open out.tr w]

$ns trace-all $tf

set nodes(is) [$ns node]

set nodes(ms) [$ns node]

set nodes(bs1) [$ns node]

set nodes(bs2) [$ns node]

set nodes(lp) [$ns node]

proc cell\_topo {} {

global ns nodes

$ns duplex-link $nodes(lp) $nodes(bs1) 3Mbps 10ms DropTail

$ns duplex-link $nodes(bs1) $nodes(ms) 1 1 RED

$ns duplex-link $nodes(ms) $nodes(bs2) 1 1 RED

$ns duplex-link $nodes(bs2) $nodes(is) 3Mbps 50ms DropTail

puts "Cell Topology"

}

proc set\_link\_params {t} {

global ns bwDL bwUL propDL propUL buf nodes

$ns bandwidth $nodes(bs1) $nodes(ms) $bwDL($t) simplex

$ns bandwidth $nodes(ms) $nodes(bs1) $bwUL($t) simplex

$ns bandwidth $nodes(bs2) $nodes(ms) $bwDL($t) simplex

$ns bandwidth $nodes(ms) $nodes(bs2) $bwUL($t) simplex

$ns delay $nodes(bs1) $nodes(ms) $propDL($t) simplex

$ns delay $nodes(ms) $nodes(bs1) $propDL($t) simplex

$ns delay $nodes(bs2) $nodes(ms) $propDL($t) simplex

$ns delay $nodes(ms) $nodes(bs2) $propDL($t) simplex

$ns queue-limit $nodes(bs1) $nodes(ms) $buf($t)

$ns queue-limit $nodes(ms) $nodes(bs1) $buf($t)

$ns queue-limit $nodes(bs2) $nodes(ms) $buf($t)

$ns queue-limit $nodes(ms) $nodes(bs2) $buf($t)

}

Agent/TCP set ecn\_ $opt(ecn)

Agent/TCP set window\_ $opt(window)

Queue/DropTail set summarystats\_ true

Queue/DropTail set shirnk\_drops\_ true

Queue/RED set q\_weight\_ 0.0

Queue/RED set summarystats\_ true

Queue/RED set thresh\_ $opt(minth)

Queue/RED set maxthresh\_ $opt(maxth)

Queue/RED set adaptive\_ $opt(adaptive)

DelayLink set avoidReordering\_ true

switch $opt(type) {

gsm -

gprs -

umts {cell\_topo}

}

set\_link\_params $opt(type)

$ns insert-delayer $nodes(ms) $nodes(bs1) [new Delayer]

$ns insert-delayer $nodes(bs1) $nodes(ms) [new Delayer]

$ns insert-delayer $nodes(ms) $nodes(bs2) [new Delayer]

$ns insert-delayer $nodes(bs2) $nodes(ms) [new Delayer]

if {$opt(flows) == 0} {

set tcp1 [$ns create-connection TCP/Sack1 $nodes(is) TCPSink/Sack1 $nodes(lp) 0]

set ftp1 [[set tcp1] attach-app FTP]

$ns at 0.8 "[set ftp1] start"

}

proc stop {} {

global nodes opt tf

set wrap $opt(wrap)

set sid [$nodes($opt(srcTrace)) id]

set did [$nodes($opt(dstTrace)) id]

if {$opt(srcTrace)=="is"} {

set a "-a out.tr"

} else {

set a "out.tr"

}

set GETRC "/home/sivakamasundari/Documents/ns-allinone-2.35/ns-2.35/bin/getrc"

set RAW2XG "/home/sivakamasundari/Documents/ns-allinone-2.35/ns-2.35/bin/raw2xg"

exec $GETRC -s $sid -d $did -f 0 out.tr | \

$RAW2XG -s 0.01 -m $wrap -r > plot.xgr

exec $GETRC -s $did -d $sid -f 0 out.tr | \

$RAW2XG -a -s 0.01 -m $wrap >> plot.xgr

exec $GETRC -s $sid -d $did -f 1 out.tr | \

$RAW2XG -s 0.01 -m $wrap -r >> plot.xgr

exec $GETRC -s $sid -d $did -f 1 out.tr | \

$RAW2XG -s 0.01 -m $wrap -a >> plot.xgr

exec ./xg2gp.awk plot.xgr

if {!$opt(quiet)} {

exec xgraph -bb -tk -nl -m time -y packets plot.xgr &

}

exit 0

}

$ns at $opt(stop) "stop"

$ns run