set ns [new Simulator]

set nf [open out2.nam w]

set tf [open out1.tr w]

$ns trace-all $tf

$ns namtrace-all $nf

set nodes(s) [$ns node]

set nodes(bs1) [$ns node]

set nodes(ms) [$ns node]

set nodes(bs2) [$ns node]

set nodes(d) [$ns node]

proc cell\_topo {} {

global ns nodes

$ns duplex-link $nodes(s) $nodes(bs1) 3Mb 10ms DropTail

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

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

$ns duplex-link $nodes(bs2) $nodes(d) 3Mb 50ms DropTail

puts "Cell Topology"

}

set bwDL(umts) 9600

set bwUL(umts) 9600

set buf(umts) 10

set propDL(umts) .500

set propUL(umts) .500

proc set\_link\_params {t} {

global propUL propDL bwUL bwDL buf ns 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)

}

set opt(type) umts

set opt(ecn) 0

set opt(adaptive) 1

set opt(window) 30

set opt(maxth) 10

set opt(minth) 5

Queue/DropTail set summarystats\_ true

Queue/DropTail set shrink\_drops\_ true

Queue/RED set q\_weight\_ 0.0

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

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

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

Queue/RED set summarystats\_ true

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

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

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]

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

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

$ns at 0.5 "$ftp1 start"

proc stop {} {

global nf tf ns

$ns flush-trace

close $nf

close $tf

exec nam out2.nam &

exit 0

}

$ns at 100 "stop"

$ns run