

#6. Implement and study the performance of CDMA on NS2/NS3 (Using stack called Call net) or equivalent environment.

```
set stop 100;
set type cdma;
set minth 0;
set maxth 30;
set adaptive 1;
set flows 0;
set window 30;
set web 2;
set opt(wrap) 100;
set opt(srcTrace) is;
set opt(dstTrace) bs2;
set bwDL(cdma) 384000
set bwUL(cdma) 64000
set propDL(cdma) .150
set propUL(cdma) .150
```

```
set ns [new Simulator]
set tf [open out.tr w]
$ns trace-all $tf
set nf [open out.nam w]
$ns namtrace-all $nf
```

```
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]
```

```
$nodes(is) label "Sender"
$nodes(ms) label "Mobile Station"
$nodes(bs1) label "Base Station 1"
$nodes(bs2) label "Base Station 2"
$nodes(lp) label "Destination"
$nodes(is) color "magenta"
```

```
proc cell_topo { } {
global ns nodes
$ns duplex-link $nodes(lp) $nodes(bs1) 3Mbps 10nodes(ms) 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 50nodes(ms) DropTail
puts "CDMA Cell Topology"
}
```

```
proc set_link_para {t} {
global ns nodes bwUL bwDL propUL propDL buf
```

```

$ns bandwidth $nodes(bs1) $nodes(ms) $bwDL($t) duplex
$ns bandwidth $nodes(bs2) $nodes(ms) $bwDL($t) duplex
$ns delay $nodes(bs1) $nodes(ms) $propDL($t) duplex
$ns delay $nodes(bs2) $nodes(ms) $propDL($t) duplex
$ns queue-limit $nodes(bs1) $nodes(ms) 20
$ns queue-limit $nodes(bs2) $nodes(ms) 20
}

Queue/RED set adaptive_ $adaptive
Queue/RED set thresh_ $minth
Queue/RED set maxthresh_ $maxth
Agent/TCP set window_ $window

source web.tcl

switch $type {
cdma {cell_topo}
}

set_link_para $type
$ns insert-delayer $nodes(ms) $nodes(bs1) [new Delayer]
$ns insert-delayer $nodes(ms) $nodes(bs2) [new Delayer]

if {$flows == 0} {
set tcp1 [$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"
}

if {$flows > 0} {
set tcp1 [$ns create-connection TCP/Sack1 $nodes(is) TCPSink/Sack1 $nodes(lp) 0]
set ftp1 [[set tcp1] attach-app FTP]
$tcp1 set window_ 100
$ns at 0.0 "[set ftp1] start"
$ns at 3.5 "[set ftp1] stop"
set tcp2 [$ns create-connection TCP/Sack1 $nodes(is) TCPSink/Sack1 $nodes(lp) 0]
set ftp2 [[set tcp2] attach-app FTP]
$tcp2 set window_ 3
$ns at 1.0 "[set ftp2] start"
$ns at 8.0 "[set ftp2] stop"
}

proc stop {} {
global nodes opt
set wrap $opt(wrap)
set sid [$nodes($opt(srcTrace)) id]
set did [$nodes($opt(dstTrace)) id]
set a "out.tr"
set GETRC "/home/cs/ns-allinone-2.35/ns-2.35/bin/getrc"
set RAW2XG "/home/cs/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 xgraph -x time -y packets plot.xgr &  
exec nam out.nam &  
exit 0  
}
```

```
$ns at $stop "stop"  
$ns run
```