Date, 1st September, 2017
A01630510
Victor Manuel García Rosales
TC2007 Métodos Cuantitativos y Simulación

Throughput of a TraceFile in Python

Example of Trace File

TCL code

```
#TCL code

#Create a new simulator object
set ns [new Simulator]

#Define different colors for data flows (for NAM)
$ns color 1 Blue
$ns color 2 Red

#Open the NAM trace file
set nf [open out6.nam w]
$ns trace-all $nf

#Enabling tracing of all events of the simulation
set f [open out6.all w]
$ns trace-all $f

#Define a 'finish' procedure
proc finish {} {
     global ns nf f
     $ns flush-trace
     #Close the NAM trace file
     close $nf
     close $f
     #Execute NAM on the trace file
     exec nam out.nam &
     exit 0
}
```

```
#Create four nodes
set n0 [$ns node]
set n1 [$ns node]
set n2 [$ns node]
set n3 [$ns node]

#Create links between the nodes
$ns duplex-link $n0 $n2 2Mb 10ms DropTail
$ns duplex-link $n1 $n2 2Mb 10ms DropTail
$ns duplex-link $n2 $n3 1.7Mb 20ms DropTail

#Set Queue Size of link (n2-n3)
$ns queue-limit $n2 $n3 10

#Give node position (for NAM)
$ns duplex-link-op $n0 $n2 orient right-down
$ns duplex-link-op $n1 $n2 orient right-up
$ns duplex-link-op $n2 $n3 orient right

#Monitor the queue for link (n2-n3). (for NAM)
$ns duplex-link-op $n2 $n3 queuePos 0.5

#Setup a TCP connection
set tcp [new Agent/TCP]
$tcp set class_ 2
$ns attach-agent $n0 $tcp
set sink [new Agent/TCPSink]
$ns attach-agent $n3 $sink
$ns connect $tcp $sink
$tcp set fid_ 1

#Setup a FTP over TCP connection
set ftp [new Application/FTP]
$ftp attach-agent $tcp
$ftp set type_ FTP

#Setup a UDP connection
set udp [new Agent/UDP]
$ns attach-agent $n1 $udp
set null [new Agent/Null]
```

```
$ns attach-agent $n3 $null
$ns connect $udp $null
$udp set fid_ 2

#Schedule events FTP agents
$ns at 1.0 "$ftp start"
$ns at 4.0 "$ftp stop"

#Detach tcp and sink agents (not really necessary)
$ns at 4.5 "$ns detach-agent $n0 $tcp ; $ns detach-agent $n3
$sink"

#Call the finish procedure after 5 seconds of simulation time
$ns at 5.0 "finish"

#Run the simulation
$ns run
```

Example of python file with Throughput

```
from sys import argv, maxsize

#Packet size
Size_packet = 5

#Index of no of packet
Event= 0

#Index of time
Time = 1

def tcput(traceF):
    #Open the traceFile as an F file
    with open(traceF) as F:
        #Start the max time at the max size from system
        St = maxsize
        #Start the end time at 0
        end = 0
        #Start the total at 0
        Total = 0

    for l in F.readlines():
        #read the file for each line
        rows = l.split(" ")
        #get the size of each packet
        size = int(rows[Size_packet])
        #get the time of each packet
        tt = float(rows[Time])
        #get the event for each packet
        e = rows[Event]

        if e == "r": #if the event is equal "r" we get the total size
of the packet and the time elapsed
            Total += size

            end = max(end, tt)

            St = min(St, tt)

    #Get the total time
    TotalTime = end - St

    #Get the result of the throughput
```

```
    Result = Total / TotalTime

    #Finally print the result
    print("bits: " + str(Total)+"\nTime : " + str(TotalTime)+"
seconds\nResult: " + str(Result) + " bits per second")


if __name__ == '__main__':
    traceF = argv[1]
    tcput(traceF)
```

Instructions

Simply create a .py file with the code above and run it.

PS: you should pass a valid trace file name and extension as the example above with 4 nodes.