Code source

March 22, 2011

Abstract

Le projet est disponible sur le répository mercurial suivant:
`https://bitbucket.org/kakwa/simres_2/overview`

1 Simulateur

```
set ns [new Simulator]
#set tf [open out.tr w]
#$ns trace-all $tf
###################### PARAM #################

set bite_rate 100Mb
set delay 0.000001ms
set tf1 [open trace1 w]
set tf2 [open trace2 w]
set intervalle_trace 1
set intervalle_relance 1
##################### PARAM ##################
proc finish {} {

global tf1 tf2 ns
$ns flush-trace
close $tf1
close $tf2
exit 0
}
################# LINKS ####################


set n1 [$ns node]
set n2 [$ns node]
set n3 [$ns node]

set nred [$ns node]

set ndest [$ns node]

$ns simplex-link $n1 $nred 9999Mb 0ms DropTail
```

```
$ns queue-limit $n1 $nred 999999999
$ns simplex-link $n2 $nred 9999Mb 0ms DropTail
$ns queue-limit $n2 $nred 999999999
$ns simplex-link $n3 $nred 9999Mb 0ms DropTail
$ns queue-limit $n3 $nred 999999999

#$ns simplex-link $nred $ndest $bite_rate $delay DropTail
$ns simplex-link $nred $ndest $bite_rate $delay dsRED/edge
$ns queue-limit $nred $ndest 999999999

set q0n1 [[$ns link $nred $ndest] queue]

$q0n1 set numQueues_ 3
$q0n1 setNumPrec 3
$q0n1 meanPktSize 632.5

$q0n1 addPolicyEntry [$n1 id] [$ndest id] Null 10
$q0n1 addPolicyEntry [$n2 id] [$ndest id] Null 11
$q0n1 addPolicyEntry [$n3 id] [$ndest id] Null 12
$q0n1 addPolicerEntry Null 10
$q0n1 addPolicerEntry Null 11
$q0n1 addPolicerEntry Null 12


$q0n1 addPHBEntry 10 0 0
$q0n1 addPHBEntry 11 1 0
$q0n1 addPHBEntry 12 2 0

$q0n1 setMREDMode DROP 0
$q0n1 setMREDMode DROP 1
$q0n1 setMREDMode DROP 2

$q0n1 configQ 0 0 999999990 999999999 1
$q0n1 configQ 1 0 999999990 999999999 1
$q0n1 configQ 2 0 999999990 999999999 1

$q0n1 setSchedularMode WRR
$q0n1 addQueueWeights 0 50
$q0n1 addQueueWeights 1 100
$q0n1 addQueueWeights 2 750

########### DATA APPLICATIONN ############

set donnees1 [new Agent/UDP]
$donnees1 set fid_ 1
set donnees2 [new Agent/UDP]
$donnees2 set fid_ 1
set donnees3 [new Agent/UDP]
$donnees3 set fid_ 1
```

```
$donnees1 set packetSize_ 50
$donnees2 set packetSize_ 500
$donnees3 set packetSize_ 1500

set video [new Agent/UDP]
$video set fid_ 2

$video set packetSize_ 1000

set voix [new Agent/UDP]

$voix set packetSize_ 100
$voix set fid_ 3

set reception1 [new Agent/Null]
set reception2 [new Agent/Null]
set reception3 [new Agent/Null]
set reception4 [new Agent/Null]
set reception5 [new Agent/Null]


$ns attach-agent $n1 $donnees1
$ns attach-agent $n1 $donnees2
$ns attach-agent $n1 $donnees3
$ns attach-agent $n2 $video
$ns attach-agent $n3 $voix

$ns attach-agent $ndest $reception1
$ns attach-agent $ndest $reception2
$ns attach-agent $ndest $reception3
$ns attach-agent $ndest $reception4
$ns attach-agent $ndest $reception5

$ns connect $donnees1 $reception1
$ns connect $donnees2 $reception2
$ns connect $donnees3 $reception3
$ns connect $video $reception4
$ns connect $voix $reception5

set cbr1 [new Application/Traffic/CBR]
$cbr1 set packet_size 100
$cbr1 set rate_ 20Mb

$cbr1 attach-agent $voix
#$ns connect $cbr1 $reception

set poisson1 [new Application/Traffic/Exponential]
$poisson1 set packetSize_ 50
$poisson1 set burst_time_ 0
$poisson1 set idle_time_ 0.4134ms
```

```
$poisson1 set rate_ 9999Mb

set poisson2 [new Application/Traffic/Exponential]
$poisson2 set packetSize_ 500
$poisson2 set burst_time_ 0
$poisson2 set idle_time_ 0.5510ms
$poisson2 set rate_ 9999Mb

set poisson3 [new Application/Traffic/Exponential]
$poisson3 set packetSize_ 1500
$poisson3 set burst_time_ 0
$poisson3 set idle_time_ 0.5510ms
$poisson3 set rate_ 9999Mb

$poisson1 attach-agent $donnees1
$poisson2 attach-agent $donnees2
$poisson3 attach-agent $donnees3

#$ns connect $poisson1 $reception
#$ns connect $poisson2 $reception
#$ns connect $poisson3 $reception

set exp [new Application/Traffic/Exponential]
$exp set packetSize_ 1000
$exp set burst_time_ 1ms
$exp set idle_time_ 7ms
$exp set rate_ 240Mb

$exp attach-agent $video
#$ns connect $video $reception

############## Monitor ###############


set monitor [$ns makeflowmon Fid]
$ns attach-fmon [$ns link $nred $ndest] $monitor
set samples_object [new Samples]
$monitor set-delay-samples $samples_object

set mon_video [new QueueMonitor/ED/Flow]
set sample_video [new Samples]
$mon_video  set-delay-samples $sample_video
set classif_video [$monitor classifier]
set slot_video [$classif_video installNext $mon_video]
$classif_video set-hash auto $video $reception4 2 $slot_video

set mon_donnee [new QueueMonitor/ED/Flow]
set sample_donnee [new Samples]
$mon_donnee  set-delay-samples $sample_donnee
set classif_donnee [$monitor classifier]
set slot_donnee [$classif_donnee installNext $mon_donnee]
$classif_donnee set-hash auto $donnees1 $reception1 1 $slot_donnee
$classif_donnee set-hash auto $donnees2 $reception2 1 $slot_donnee
$classif_donnee set-hash auto $donnees3 $reception3 1 $slot_donnee

set mon_voix [new QueueMonitor/ED/Flow]
set sample_voix [new Samples]
$mon_voix  set-delay-samples $sample_voix
set classif_voix [$monitor classifier]
set slot_voix [$classif_voix installNext $mon_voix]
$classif_voix set-hash auto $voix $reception5 3 $slot_voix

proc affiche-delais {} {

global ns tf1 tf2 intervalle_trace  monitor mon_voix mon_video mon_donnee
sample_voix sample_video sample_donnee samples_object;

puts $tf1 "[$ns now] [$monitor set pdrops_] [$monitor set parrivals_]
 [$mon_donnee set pdrops_] [$mon_donnee set parrivals_]
[$mon_video set pdrops_] [$mon_video set parrivals_]
 [$mon_voix set pdrops_] [$mon_voix set parrivals_]";

puts $tf2 "[$ns now] [$samples_object mean] [$sample_video mean]
 [$sample_voix mean] [$sample_donnee mean]";

$ns at [expr [$ns now]+$intervalle_trace] "affiche-delais"
}

proc relance-cbr {} {
global cbr1 ns intervalle_relance voix
$cbr1 stop
delete $cbr1
set cbr1 [new Application/Traffic/CBR]
$cbr1 set packet_size 800
$cbr1 set rate_ 20Mb
$cbr1 set random 0

$cbr1 attach-agent $voix
$cbr1 start
$ns at [expr [$ns now]+$intervalle_relance] "relance-cbr"
}

#set udp1 [new Agent/UDP]
#$udp1 set packetSize_ 125

#set udp2 [new Agent/Null]

#$ns attach-agent $n0 $udp1
#$ns attach-agent $n1 $udp2
```

```
#ns connect $udp1 $udp2

#set cbr1 [new Application/Traffic/CBR]
#$cbr1 set packet_size 125
#$cbr1 set rate_ $bite_rate
#$cbr1 set random 1

#$cbr1 attach-agent $tcp1

#$ns at 0.0 "$tcp1 send 1500000"
#$ns at 0.0 "$cbr1 start"
#$ns at 100000.0 "$cbr1 stop"

#$cbr1 start

$ns at 0 "relance-cbr"
$poisson1 start
$poisson2 start
$poisson3 start
$exp start
$ns at $intervalle_trace "affiche-delais"

$ns at 30000.0 finish
$ns run
```

2 Calcul de délais

```
#!/bin/awk -f
BEGIN{
        s1=0;
        s2=0;
        c=0;
        s1_vid=0;
        s2_vid=0;
        s1_vx=0;
        s2_vx=0;
        s1_d=0;
        s2_d=0;
}
{
s1=$2+s1; s2=$2^2+s2; c++;
s1_vid=$3+s1_vid; s2_vid=$3^2+s2_vid;
s1_vx=$4+s1_vx; s2_vx=$4^2+s2_vx;
s1_d=$5+s1_d; s2_d=$5^2+s2_d;
}

END{
        sigma=sqrt((s2+(s1^2)/c)/(c-1));
        average=s1/c;
        epsi=sigma*4.5*sqrt(1/c);
        conf=epsi/average

        sigma_vid=sqrt((s2_vid+(s1_vid^2)/c)/(c-1));
        average_vid=s1_vid/c;
        epsi_vid=sigma_vid*4.5*sqrt(1/c);
        conf_vid=epsi_vid/average_vid

        sigma_vx=sqrt((s2_vx+(s1_vx^2)/c)/(c-1));
        average_vx=s1_vx/c;
        epsi_vx=sigma_vx*4.5*sqrt(1/c);
        conf_vx=epsi_vx/average_vx

        sigma_d=sqrt((s2_d+(s1_d^2)/c)/(c-1));
        average_d=s1_d/c;
        epsi_d=sigma_d*4.5*sqrt(1/c);
        conf_d=epsi_d/average_d


        print "trafic  epsilonne  moyenne  confiance";
        print "globale:"  epsi, average, conf;
        print "video:"  epsi_vid, average_vid, conf_vid;
        print "voix:"  epsi_vx, average_vx, conf_vx;
        print "donnees:"  epsi_d, average_d, conf_d;

        }
```

3 Calcul de taux de perte

```
#!/bin/awk -f
BEGIN{  s1=0;
        s2=0;
        s1p=0
        s2p=0
        s11=0
        s21=0
        s12=0
        s22=0
        s13=0
        s23=0

        c=0;
        print "tata:" s1, s2, c;
        mem=0
        memp=0
        mem1=0
        memp1=0
        mem2=0
        memp2=0
        mem3=0
        memp3=0
```

```
}

{
if  (($3-mem)>0 && ($2-memp>=0)){

            s1=($2-memp)/($3-mem+$2-memp)+s1;
         s2=(($2-memp)/($3-mem+$2-memp))^2+s2;  c++;

            s11=($4-memp1)/($5-mem1+$4-memp1)+s11;
s21=(($4-memp1)/($5-mem1+$4-memp1))^2+s21

            s12=($6-memp2+$8-memp3)/($7-mem2+$6-memp2+$9-mem1+$8-memp3)+s11;
s22=(($6-memp2+$8-memp3)/($7-mem2+$6-memp2+$9-mem1+$8-memp3))^2+s22;
            s13=($8-memp3)/($9-mem1+$8-memp3)+s13;
s23=(($8-memp3)/($9-mem3+$6-memp3))^2+s23;
            mem=$3;
            memp=$2;
            mem1=$3;
            memp1=$2;
            mem2=$3;
            memp2=$2;
            mem3=$3;
            memp3=$2;
            }
}

END{print "toto:", s1, s2, c;
            sigma=sqrt((s2+(s1^2)/c)/(c-1));
            epsi=sigma*4.5*sqrt(1/c);
            average=s1/c
            conf=epsi/average
            sigma1=sqrt((s21+(s11^2)/c)/(c-1));
            epsi1=sigma1*4.5*sqrt(1/c);
            average1=s11/c
            conf1=epsi1/average1
            sigma2=sqrt((s22+(s12^2)/c)/(c-1));
            epsi2=sigma2*4.5*sqrt(1/c);
            average2=s12/c
            conf2=epsi2/average2
            sigma3=sqrt((s23+(s13^2)/c)/(c-1));
            epsi3=sigma3*4.5*sqrt(1/c);
            average3=s13/c
            conf3=epsi3/average3

            #sigmap=sqrt((s2p+(s1p^2)/c)/(c-1));
            #epsip=sigmap*4.5*sqrt(1/c);
            print "globale:" epsi, average, conf;
            print "donnees" epsi1, average1, conf1;
            print "video&voix:" epsi2, average2, conf2;
```

```
}
```