

APÊNDICE A - Network Simulator

Neste apêndice constam os detalhes de simulação incluindo código fonte do módulo de monitorização e mapeamento interdomínios e arquivos TCL responsáveis pela sequência das simulações.

A.1 O Simulador

Todo o processo de simulação desta monografia foi realizado com um simulador de redes escrito em linguagem C e projetado pela universidade de Berkeley para trabalhar em ambiente *unix*.

A.2 O Ambiente de Simulação

O ambiente utilizado nas simulações deste trabalho inclui:

A.2.1 Softwares

- a) Simulador Network Simulator versão 2.1b5 [NS-2].
- b) Sistema Operacional FreeBSD versão 3.2 com interface gráfica KDE (<http://www.freebsd.org/>). Este sistema hospedou o simulador NS.
- c) Sistema Operacional Microsoft Windows 95 com Planilha eletrônica Microsoft Excel 97, onde os resultados obtidos nas simulações foram formatados resultando nos gráficos apresentados na dissertação.

A.2.2 Hardware

- a) Microcomputador K6-100Mhz com 40 Mb de memória RAM.
 - b) Disco rígido de 4 Gb particionado da seguinte forma: 1Gb para o FreeBSD e 3 Gb para o Windows 95.
-

A.3 Tutorial

Para quem está iniciando com o NS, recomenda-se o uso do tutorial desenvolvido por Marc Greis disponível em [NS-2].

A.4 Procedimentos Preliminares

- a) Download do NS em sua versão completa (NSallinone) que pode ser feito em <http://www-mash.cs.berkeley.edu/ns>. O tamanho aproximado do arquivo é de 15Mb.
 - b) O arquivo obtido estará compactado com a extensão ".tar". Para descompactá-lo, no unix, use o comando "tar -zxvf <nome do arquivo> ".
Ex.: tar -zxvf ns-allinone-2_1b5a_tar.tar.
 - c) Feito isso, será preciso proceder a instalação. Para fazê-lo, entre no diretório ../nsallinone-2.1b5 e digite "./install"
 - d) Após a instalação ter sido completada, será necessário atualizar o *PATH* para que este reconheça o diretório do NS. Para isso digite "setenv *PATH* /usr/local:\$PATH" (Considerando que o ns foi instalado em usr e não em home).
Ex.: setenv *PATH* /usr/ns-allinone-2.1b5/bin:\$PATH;
setenv *PATH* /usr/ns-allinone-2.1b5/otcl-1.0a4:\$PATH.
 - e) Para testar se a instalação foi bem sucedida execute "./validate". (opcional)
-

Para informações adicionais e links relacionados, acesse o endereço web :
<http://www.mauro.margalho.nom.br/simulador/simulador.htm>

A.5 Lista de Discussão

A lista de discussão do NS pode ser uma grande aliada na troca de experiências e obtenção de códigos fonte ou *patches*. Os procedimentos necessários para assinar a lista estão descritos em [NS-2].

A.6 Patches

Para adquirir os *patches* (códigos fonte) requeridos à reprodução do experimento com o mapeador apresentado nesta dissertação - MATRIX (MApeador de TRáfego Interdomínios eXtensível) - deve-se enviar e-mail no formato mostrado na Figura A.01:

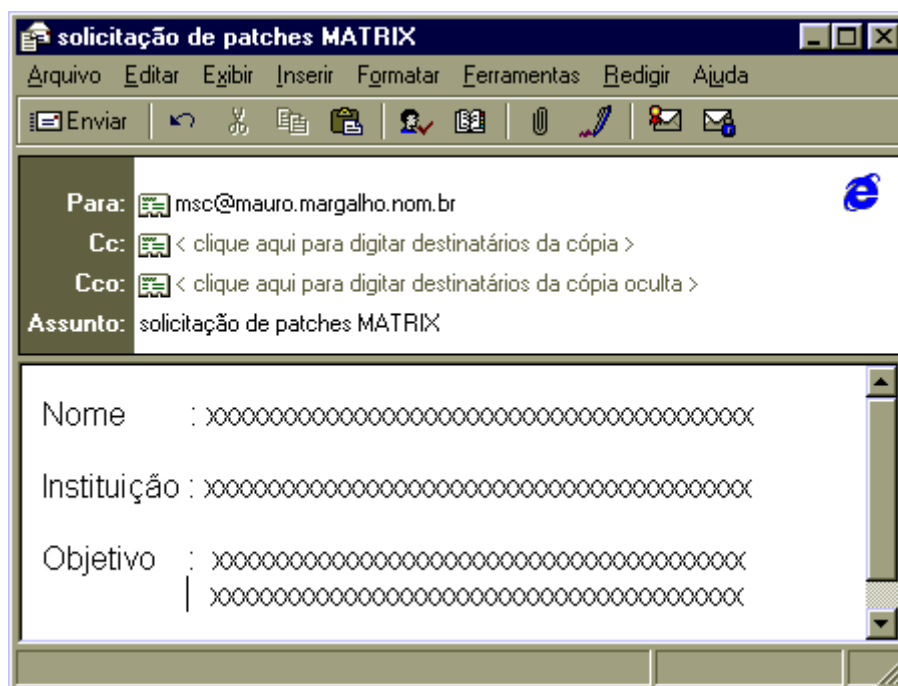


Figura A.01- Formato de E-mail para Requisição de Patches

Trata-se de um procedimento controle que visa interação com outros grupos que utilizam o NS em suas simulações. Pede-se apenas que as referências autorais sejam mantidas.

A seguir referenciamos os diversos módulos da simulação e seus respectivos autores:

MÓDULO	INFORMAÇÕES
Mapeador MATRIX	Autor : Mauro Margalho Coutinho
	E-mail : msc@mauro.margalho.nom.br
	Web Site : http://www.mauromargalho.nom.br
RSVP	Autor : Mark Greis
	E-mail : greis@cs.uni-bonn.de
	Web Site : http://titan.cs.uni-bonn.de/~greis/rsvpn
DiffServ	Autor : Sean Murphey
	E-mail : sean.murphy@teltec.dcu.ie
	Web Site : http://www.teltec.dcu.ie/~murphys/ns-work/diffserv/index.html

MPLS Estático	Autor : Christiane Purificação
	E-mail : csp2@di.ufpe.br
	Web Site : http://www.di.ufpe.br/~csp2
	Autor : Zuleika Nascimento
	E-mail : zuleika@pop-pe.rnp.br
	Web Site : http://www.di.ufpe.br/~ztn

Tabela A.01 - Autores de Módulos NS

MEASURE.CC (Mapeador de Tráfego)

```
/* -*- Mode:C++; c-basic-offset:8; tab-width:8; indent-tabs-mode:t -*- */
/* Medidor-Marcador desenvolvido por Mauro Margalho Coutinho */
/* Mestrando em Ciencia da Computacao na UFPE - Fev. 2000 */
/* e-mail: msc@mauro.margalho.nom.br */

#include "measure.h"
#include "tcp.h"
#include "ip.h"
#include <iostream.h>

static class MeasureClass : public TclClass {
public:
    MeasureClass() : TclClass("Measure") {}
    TclObject* create(int, const char*const*) {
        return (new Measure);
    }
} class_pktstats;

Measure::Measure(): Connector(), src_(0), dst_(0),
    tempo_de_medicao_(0.0),
    inicia_medicao_(0.0), calcula_taxa_(0.0),
    dominio_origem_ (0), dominio_destino_ (0),
    dominio_anterior_ (0)
{
    bind ( "src_", &src_ ) ;
    bind ( "dst_", &dst_ ) ;
    bind ( "tempo_de_medicao_", &tempo_de_medicao_ );
    bind ( "inicia_medicao_", &inicia_medicao_ );
    bind ( "calcula_taxa_", &calcula_taxa_ );
    bind ( "dominio_origem_", &dominio_origem_ );
    bind ( "dominio_destino_", &dominio_destino_ );
}

int Measure::command(int argc, const char*const* argv)
{
    Tcl& tcl = Tcl::instance();
    return Connector::command(argc,argv) ;
}

void Measure::recv (Packet *p, Handler *h) {

    struct hdr_cmh *ch = HDR_CMH(p) ;
    struct hdr_ip *iph = HDR_IP(p) ;

    //int label = ((hdr_cmh*)p->access(off_cmh_))->size();
    //printf ("label: %d \n", ch->label_);

    // Apenas o fluxo informado no TCL e testado.
```

```

if ( (iph->src_ == src_) && ( iph->dst_ == dst_ ) ) {
    int numBytes = ((hdr_cmh*)p->access(off_cmh_))>size();
    int lb = iph->mmcBandW_ ;    // Largura de Banda
    int ht = iph->mmcHTime_ ;    // Hold Time
    int br = iph->mmcBurst_ ;    // Rajada
    int lt = iph->mmcLatSen_ ;   // Sensibilidade a Latencia
    int jt = iph->mmcJitSen_ ;   // Sensibilidade a Jitter
    bool ServAF = false ;
    bool ServEF = false ;
    if (dominio_destino_ != dominio_anterior_) {
        printf ("Convencao: 1-RSVP 2-DiffServ 3-MPLS \n");
        printf ("Mapeando para novo Dominio \n");
        printf ("Fonte: %g Destino: %g \n", src_, dst_ );
        printf ("De: %d \n", dominio_origem_);
        printf ("Para: %d \n", dominio_destino_);
        printf ("Inicio da Medicao: %g Tempo de Medicao: %g s
\n",inicia_medicao_,tempo_de_medicao_ );
        dominio_anterior_ = dominio_destino_;
    }
    // testes para verificar os requisitos ne QoS necessarios ao mapeamento
    //      1->alto 2->medio 3->baixo
    // A marcacao e feita baseada nos requisitos de QoS e na taxa da aplicacao
    // ***** T A B E L A      P R O P O S T A *****
    if (dominio_origem_ == 1 && dominio_destino_ == 2) {
        if ((lb == 1) && (ht == 1) && (br == 3) && (lt == 3) && (jt == 1))
            { ServEF = true; }
        else {
            if ((lb == 3) && (ht == 3) && (br == 3) && (lt == 1) && (jt == 2))
                { ServAF = true; }
        }
    }
    // *****
    double now = Scheduler::instance().clock() ;
    if ((inicia_medicao_ < now) && (( now - inicia_medicao_ ) <=
tempo_de_medicao_ )) {
        conta_pacotes_ = conta_pacotes_ + numBytes; }
    else {
        if ((inicia_medicao_ < now ) && ( tempo_de_medicao_ != 0 )) {
            calcula_taxa_ = conta_pacotes_/tempo_de_medicao_ * 8;
            //printf ("now: %g taxa: %g \n", now, calcula_taxa_);
            // **** I N I C I O      D O      M A P E A M E N T O ****

            if (dominio_origem_ == 1 && dominio_destino_ == 2) {
                // Mapeando de RSVP para DiffServ
                if (calcula_taxa_ < 50000) { iph->DSCP_ = BE;}
                else {
                    if ((calcula_taxa_ >= 50000 && calcula_taxa_ < 100000) &&
ServAF) {

                        iph->DSCP_ = AF12;}
                    else {

```

```
        if ((calcula_taxa_ >= 100000 && calcula_taxa_ <= 250000) &&
ServAF) {

            iph->DSCP_ = AF11;}
        else {
            if (calcula_taxa_ > 250000 && ServEF) {
                iph->DSCP_ = EF;}
            else {iph->DSCP_ = BE;}
        }
    }
}
}
}
}
else {
    if (dominio_origem_ == 2 && dominio_destino_ == 3) {
// Mapeando de DiffServ para MPLS
        if (iph->DSCP_ == EF){ch->label_ = 0;}
        else {
            if (iph->DSCP_ == AF11){ch->label_ = 1;}
            else {
                if (iph->DSCP_ == AF12){ch->label_ = 1;}
                else {
                    if (iph->DSCP_ == BE){ch->label_ = 2;}
                }
            }
        }
    }
}
}
// **** F I M    D O    M A P E A M E N T O ****
}

    }
    target_->recv(p,h) ;
}
```

MEASURE.H

```
/* -*- Mode:C++; c-basic-offset:8; tab-width:8; indent-tabs-mode:t -*- */
```

```
#ifndef _measure_h
#define _measure_h
```

```
#include "connector.h"
#include "packet.h"
#include "tclcl.h"
```

```
class Measure: public Connector {
private:
    nsaddr_t          src_ ;
    nsaddr_t          dst_ ;
```

```
        double        tempo_de_medicao_;
        double        inicia_medicao_;
        double        calcula_taxa_;
        double        conta_pacotes_;
        int           dominio_origem_;
        int           dominio_destino_;
        int           dominio_anterior_;
public:
        Measure() ;
        int          command(int argc, const char*const* argv);
        void         recv(Packet* pkt, Handler* h) ;
} ;
```

#endif

MONITOR.TCL (Executa um dos módulos da Simulação)

```
#####
# Desenvolvido por Mauro Margalho Coutinho
# e-mail: msc@mauro.margalho.nom.br / Fev-2000
#####

set ns [new Simulator]

##### Parametros #####
$ns rtproto Session

set ef_weight 5
set af_weight 3
set be_weight 2

$ns color 0 red
$ns color 1 blue
$ns color 2 green
$ns color 3 yellow
$ns color 46 purple
#####

##### Arquivos de Saida #####
set f0 [open out0.tr w]
set f1 [open out1.tr w]
set f2 [open out2.tr w]
#####

##### Cria os nos #####
#----- Cria os nos do dominio RSVP -----
set bordaRSVP1 [$ns node]
set nucleoRSVP1 [$ns node]
set fonteRSVP1 [$ns node]
set fonteRSVP2 [$ns node]
set fonteRSVP3 [$ns node]
#----- Cria os nos do dominio DS -----
```

```
set bordaDS1 [$ns node]
set nucleoDS1 [$ns node]
set nucleoDS2 [$ns node]
set nucleoDS3 [$ns node]
set nucleoDS4 [$ns node]
set bordaDS2 [$ns node]
#----- Cria os nos do dominio mpls -----
set recife [$ns node]
set brasilia [$ns mplsnode]
set beloehorizonte [$ns node]
set riodejaneiro [$ns node]
set saopaulo [$ns mplsnode]
set curitiba [$ns node]
set portoalegre [$ns mplsnode]
set fortaleza [$ns node]
#----- Cria os nos perifericos -----
set belem [$ns node]
set manaus [$ns node]
set goiania [$ns node]
set campogrande [$ns node]
#=====

#==== Network Animator =====
# Keep nam trace generation off for now
#set nf [open out.nam w]
#$ns namtrace-all $nf
#=====

#==== Cria os Enlaces do RSVP =====
$ns duplex-rsvp-link $bordaRSVP1 $nucleoRSVP1 1Mb 10ms 0.5 200 5000 Param Null
$ns duplex-rsvp-link $nucleoRSVP1 $fonteRSVP1 1Mb 10ms 0.5 200 5000 Param Null
$ns duplex-rsvp-link $nucleoRSVP1 $fonteRSVP2 1Mb 10ms 0.5 200 5000 Param Null
$ns duplex-rsvp-link $nucleoRSVP1 $fonteRSVP3 1Mb 10ms 0.5 200 5000 Param Null

# ===== Cria perfis de trafego =====
set p_af [new DSPeakRateProfile]
$p_af set-scope ONE_TO_ANY
$p_af set src_ -1
$p_af set peak_ 500000
$p_af set deltat_ 0.01
$p_af set-DSCP AF11

set p_ef [new DSPeakRateProfile]
$p_ef set-scope ONE_TO_ANY
$p_ef set src_ -1
$p_ef set peak_ 800000
$p_ef set deltat_ 0.01
$p_ef set-DSCP EF

$ns at 0.0 "$p_af start"
$ns at 0.0 "$p_ef start"
#=====
```

Apêndice A - Network Simulator

```
# ===== Cria condicionadores de trafego =====
set cond_af [new DSConditioner]
$cond_af add-profile $p_af

set cond_ef [new DSConditioner]
$cond_ef add-profile $p_ef
#=====

#==== Cria os Enlaces do DiffServ =====
$ns duplex-link $bordaRSPV1 $bordaDS1 1Mb 10ms DropTail
$ns duplex-link $bordaDS1 $nucleoDS1 1Mb 10ms DropTail
$ns duplex-link $bordaDS1 $nucleoDS2 1Mb 10ms DropTail
$ns duplex-link $bordaDS1 $nucleoDS3 1Mb 10ms DropTail
$ns duplex-link $nucleoDS2 $nucleoDS1 1Mb 10ms DropTail
$ns duplex-link $nucleoDS3 $nucleoDS1 1Mb 10ms DropTail
$ns duplex-link $nucleoDS1 $nucleoDS4 1Mb 10ms DropTail
$ns duplex-link $nucleoDS4 $bordaDS2 1Mb 10ms DropTail
#---- Cria os Enlaces do mpls -----
$ns duplex-link $bordaDS2 $recife 2Mb 10ms DropTail
$ns duplex-link $recife $brasilvia 2Mb 10ms DropTail
$ns duplex-link $brasilvia $saopaulo 2Mb 10ms DropTail
$ns duplex-link $brasilvia $belohorizonte 2Mb 10ms DropTail
$ns duplex-link $belohorizonte $riodejaneiro 2Mb 10ms DropTail
$ns duplex-link $riodejaneiro $saopaulo 2Mb 10ms DropTail
$ns duplex-link $saopaulo $curitiba 2Mb 10ms DropTail
$ns duplex-link $brasilvia $portoalegre 2Mb 10ms DropTail
$ns duplex-link $portoalegre $saopaulo 2Mb 10ms DropTail
$ns duplex-link $riodejaneiro $fortaleza 2Mb 10ms DropTail

#---- Cria os Enlaces de perifericos -----
$ns duplex-link $belem $brasilvia 512Kb 10ms DropTail
$ns duplex-link $goiania $brasilvia 512Kb 10ms DropTail
$ns duplex-link $manaus $brasilvia 512Kb 10ms DropTail
$ns duplex-link $campogrande $brasilvia 512Kb 10ms DropTail
#=====

#==== Alimentando a FIB =====
# no label/nextho
$brasilvia add-label 0 17
$brasilvia add-label 1 17
$brasilvia add-label 2 15

$portoalegre add-label 0 15
$portoalegre add-label 1 15
$portoalegre add-label 2 12
#=====

#=====
# Procedimentos para inserir o medidor mapeador
#=====
Measure set src_ 0
Measure set dst_ 0
```

```
Simulator instproc insert-measure {measure n1 n2} {
    # this method inserts an measure component before node2 on the link
    # from node1 to node2

    $self instvar link_
    set sid [$n1 id]
    set did [$n2 id]
    set templink $link_($sid:$did)
    set linktarget [$templink get-target]
    $templink set-target $measure
    $measure target $linktarget
}

#-----
#Cria o componente para monitorar os pacotes

proc New-medudp {n1 n2 sourceAgent sinkAgent sourceDom destDom iniciomed tempomed} {
    global ns end $sourceAgent $sinkAgent $sourceDom $destDom $iniciomed $tempomed
    set medudp [new Measure]
    $medudp set src_ [$sourceAgent set addr_]
    $medudp set dst_ [$sinkAgent set addr_]
    $ns insert-measure $medudp $n1 $n2
    $medudp set tempo_de_medicao_ $tempomed
    $medudp set inicia_medicao_ $iniciomed
    $medudp set dominio_origem_ $sourceDom
    $medudp set dominio_destino_ $destDom
    return $medudp
}

#-----

#=====
# Procedimentos para inserir componente de contabilizacao de
# atrasos e jitter PktStats
#=====
PktStats set src_ 0
PktStats set dst_ 0

Simulator instproc insert-pktstats {pktstats n1 n2} {
    # this method inserts an pktstats component before node2 on the link
    # from node1 to node2

    $self instvar link_
    set sid [$n1 id]
    set did [$n2 id]
    set templink $link_($sid:$did)
    set linktarget [$templink get-target]
    $templink set-target $pktstats
    $pktstats target $linktarget
}

#Cria o componente para monitorar os pacotes
# Poe no ficheiro send_time, now, pkt_delay, pkt_jitter
```

Apêndice A - Network Simulator

```
proc newPktStats {n1 n2 sourceAgent sinkAgent filename} {
    global ns end $sourceAgent $sinkAgent
    set pktstats [new PktStats]
    set fich [open $filename w]
    $pktstats set-channel $fich
    $pktstats set src_ [$sourceAgent set addr_]
    $pktstats set dst_ [$sinkAgent set addr_]
    $ns insert-pktstats $pktstats $n1 $n2
    $ns at 0.0 "$pktstats record-on"
    $ns at 200 "$pktstats record-off"
    $ns at 200 "close $fich"
    return $pktstats
}
#=====

#===== Insere o escalonador de filas WRR =====
$ns DSSchedulerLink-simplex-link $nucleoDS1 $nucleoDS4 1Mb 10ms
set s_wrr [$ns link $nucleoDS1 $nucleoDS4] queue]
$s_wrr set-ef-queue-length 40
$s_wrr set-af-queue-length 60
$s_wrr set-be-queue-length 150
$s_wrr af-queue-rio-params 0.002 30 60 50 15 30 10
$s_wrr be-queue-red-params 0.002 50 145 20
$s_wrr set ef_queue_weight_ 5
$s_wrr set af_queue_weight_ 4
$s_wrr set be_queue_weight_ 1
$s_wrr set aggregate-bytes-thresh 1000
#=====

#===== Insere os condicionadores na rede =====
$ns insert-conditioner $cond_ef $bordaDS1 $nucleoDS1
$ns insert-conditioner $cond_af $bordaDS1 $nucleoDS1
#=====

#===== R S V P =====
# Enable upcalls on all nodes
Agent/RSVP set noisy_ 255

set rsvp0 [$bordaRSVP1 add-rsvp-agent]
set rsvp1 [$nucleoRSVP1 add-rsvp-agent]
set rsvp2 [$fonteRSVP1 add-rsvp-agent]
set rsvp3 [$fonteRSVP2 add-rsvp-agent]
#=====

#===== F O N T E S      D E      T R A F E G O =====
#----- UDP com CBR (taxa 280k) Perfil de EF -----
set udp0 [new Agent/UDP]
$ns attach-agent $fonteRSVP1 $udp0
$udp0 set-DSCP BE
#Requisitos de QoS do udp0 1->alto 2->Medio 3->Baixo
$udp0 set mmcBandW_ 1
$udp0 set mmCHTime_ 1
$udp0 set mmcBurst_ 3
```

```
$udp0 set mmclatSen_ 3
$udp0 set mmcJitSen_ 1
$udp0 set fid_ 0
$udp0 set-label 2

set video [new Application/Traffic/CBR]
$video attach-agent $udp0
$video set rate_ 280k
$video set random_ 1

set tanque1 [new Agent/LossMonitor]
$ns attach-agent $saopaulo $tanque1

$ns connect $udp0 $tanque1
#----- Cria um agente FTP para a fonteDS2 - AF -----
set udp1 [new Agent/UDP]
$ns attach-agent $fonteRSVP2 $udp1
$udp1 set-DSCP BE
#Requisitos de QoS do udp0 1->alto 2->Medio 3->Baixo
$udp1 set mmcBandW_ 3
$udp1 set mmCHTime_ 3
$udp1 set mmcBurst_ 3
$udp1 set mmclatSen_ 1
$udp1 set mmcJitSen_ 2
$udp1 set fid_ 1
$udp1 set-label 2

set voz [new Application/Traffic/CBR]
$voz attach-agent $udp1
$voz set rate_ 64k
$voz set random_ 1

set tanque2 [new Agent/LossMonitor]
$ns attach-agent $saopaulo $tanque2

$ns connect $udp1 $tanque2
#----- TCP/FTP Perfil de BE -----
set tcp_font_be [new Agent/TCP]
$ns attach-agent $fonteRSVP3 $tcp_font_be
$tcp_font_be set-DSCP BE
#Requisitos de QoS do udp0 1->alto 2->Medio 3->Baixo
$tcp_font_be set fid_ 2
$tcp_font_be set-label 2

set ftp_be [new Application/FTP]
$ftp_be attach-agent $tcp_font_be
$ftp_be set random_ 1

set tanque3 [new Agent/LossMonitor]
$ns attach-agent $bordaDS2 $tanque3
```

```
$ns connect $tcp_font_be $tanque3

#----- TCP/FTP Perfil de BE -----
set tcp_font1_be [new Agent/TCP]
$ns attach-agent $fonteRSVP3 $tcp_font1_be
$tcp_font1_be set-DSCP BE
#Requisitos de QoS do udp0 1->alto 2->Medio 3->Baixo
$tcp_font1_be set fid_ 2
$tcp_font1_be set-label 2

set ftp1_be [new Application/FTP]
$ftp1_be attach-agent $tcp_font1_be
$ftp1_be set random_ 1

set tanque1_ftp [new Agent/LossMonitor]
$ns attach-agent $bordaDS2 $tanque1_ftp

$ns connect $tcp_font1_be $tanque1_ftp
#----- UDP/CBR Perfil de BE -----
set tcp_font2_be [new Agent/TCP]
$ns attach-agent $fonteRSVP3 $tcp_font2_be
$tcp_font2_be set-DSCP BE
#Requisitos de QoS do udp0 1->alto 2->Medio 3->Baixo
$tcp_font2_be set fid_ 2
$tcp_font2_be set-label 2

set ftp2_be [new Application/FTP]
$ftp2_be attach-agent $tcp_font2_be
$ftp2_be set random_ 1

set tanque2_ftp [new Agent/LossMonitor]
$ns attach-agent $bordaDS2 $tanque2_ftp

$ns connect $tcp_font2_be $tanque2_ftp
#----- UDP/CBR Perfil de BE -----
set tcp_font3_be [new Agent/TCP]
$ns attach-agent $fonteRSVP3 $tcp_font3_be
$tcp_font3_be set-DSCP BE
#Requisitos de QoS do udp0 1->alto 2->Medio 3->Baixo
$tcp_font3_be set fid_ 2
$tcp_font3_be set-label 2

set ftp3_be [new Application/FTP]
$ftp3_be attach-agent $tcp_font3_be
$ftp3_be set random_ 1

set tanque3_ftp [new Agent/LossMonitor]
$ns attach-agent $bordaDS2 $tanque3_ftp

$ns connect $tcp_font3_be $tanque3_ftp
#----- Fonte1 BE periferica -----
set udp2 [new Agent/UDP]
```

```
$ns attach-agent $nucleoDS2 $udp2
$udp2 set-DSCP BE
#Requisitos de QoS do udp0 1->alto 2->Medio 3->Baixo
$udp2 set mmcBandW_ 1
$udp2 set mmCHTime_ 1
$udp2 set mmcBurst_ 3
$udp2 set mmcLatSen_ 3
$udp2 set mmcJitSen_ 1
$udp2 set fid_ 8

set cbr_ds1 [new Application/Traffic/CBR]
$cbr_ds1 attach-agent $udp2
$cbr_ds1 set rate_ 500k
$cbr_ds1 set random_ 1

set tanque_udp2 [new Agent/LossMonitor]
$ns attach-agent $bordaDS2 $tanque_udp2

$ns connect $udp2 $tanque_udp2
#----- Fonte2 BE periferica -----
set udp3 [new Agent/UDP]
$ns attach-agent $nucleoDS3 $udp3
$udp3 set-DSCP BE
#Requisitos de QoS do udp0 1->alto 2->Medio 3->Baixo
$udp3 set mmcBandW_ 1
$udp3 set mmCHTime_ 1
$udp3 set mmcBurst_ 3
$udp3 set mmcLatSen_ 3
$udp3 set mmcJitSen_ 1
$udp3 set fid_ 8

set cbr_ds2 [new Application/Traffic/CBR]
$cbr_ds2 attach-agent $udp3
$cbr_ds2 set rate_ 500k
$cbr_ds2 set random_ 1
$cbr_ds2 set fid_ 8

set tanque_udp3 [new Agent/LossMonitor]
$ns attach-agent $bordaDS2 $tanque_udp3

$ns connect $udp3 $tanque_udp3
#-----
#----- Trafego periferico / UDP-CBR (Belem/Sao Paulo) - BE ---
set udp_font_belem [new Agent/UDP]
$ns attach-agent $belem $udp_font_belem
$udp_font_belem set-DSCP BE

set udp_belem [new Application/Traffic/CBR]
$udp_belem attach-agent $udp_font_belem
$udp_belem set rate_ 490k
$udp_belem set random_ 1
$udp_belem set fid_ 3
```

```
set udp_dest_bsaopaulo [new Agent/LossMonitor]
$ns attach-agent $saopaulo $udp_dest_bsaopaulo

$ns connect $udp_font_belem $udp_dest_bsaopaulo
#----- Trafego periferico / UDP-CBR (Manaus/Sao Paulo) - BE --
# Cria um agente TCP/FTP e liga com o noh
set udp_font_manaus [new Agent/UDP]
$ns attach-agent $manaus $udp_font_manaus
$udp_font_manaus set-DSCP BE

set udp_manaus [new Application/Traffic/CBR]
$udp_manaus attach-agent $udp_font_manaus
$udp_manaus set rate_ 490k
$udp_manaus set random_ 1
$udp_manaus set fid_ 3

set udp_dest_msaopaulo [new Agent/LossMonitor]
$ns attach-agent $saopaulo $udp_dest_msaopaulo

$ns connect $udp_font_manaus $udp_dest_msaopaulo
#----- Trafego periferico / UDP-CBR (Goiania/Sao Paulo) - BE -
set udp_font_goiania [new Agent/UDP]
$ns attach-agent $goiania $udp_font_goiania
$udp_font_goiania set-DSCP BE

set udp_goiania [new Application/Traffic/CBR]
$udp_goiania attach-agent $udp_font_goiania
$udp_goiania set rate_ 490k
$udp_goiania set random_ 1
$udp_goiania set fid_ 3

set udp_dest_gsaopaulo [new Agent/LossMonitor]
$ns attach-agent $saopaulo $udp_dest_gsaopaulo

$ns connect $udp_font_goiania $udp_dest_gsaopaulo
#----- Trafego periferico / UDP-CBR (Campo Grande/Sao Paulo)- BE-
set udp_font_campogrande [new Agent/UDP]
$ns attach-agent $campogrande $udp_font_campogrande
$udp_font_campogrande set-DSCP BE

set udp_campogrande [new Application/Traffic/CBR]
$udp_campogrande attach-agent $udp_font_campogrande
$udp_campogrande set rate_ 490k
$udp_campogrande set random_ 1
$udp_campogrande set fid_ 3

set udp_dest_csaopaulo [new Agent/LossMonitor]
$ns attach-agent $saopaulo $udp_dest_csaopaulo

$ns connect $udp_font_campogrande $udp_dest_csaopaulo
#----- Trafego periferico / UDP-CBR (Curitiba/Brasilia) - BE -
set tcp_font_curitiba [new Agent/TCP]
```

```
set tcp_dest_brasilia [new Agent/TCPSink]
$ns attach-agent $curitiba $tcp_font_curitiba
$ns attach-agent $brasilia $tcp_dest_brasilia
$ns connect $tcp_font_curitiba $tcp_dest_brasilia
$tcp_font_curitiba set fid_3
$tcp_font_curitiba set-DSCP BE

set ftp_curitiba [$tcp_font_curitiba attach-source FTP]
#----- Trafego periferico / UDP-CBR (Sao Paulo/Rio) - BE -----
set tcp_font_saopaulo [new Agent/TCP]
set tcp_dest_riodejaneiro [new Agent/TCPSink]
$ns attach-agent $saopaulo $tcp_font_saopaulo
$ns attach-agent $riodejaneiro $tcp_dest_riodejaneiro
$ns connect $tcp_font_saopaulo $tcp_dest_riodejaneiro
$tcp_font_saopaulo set fid_3
$tcp_font_saopaulo set-DSCP BE

set ftp_saopaulo [$tcp_font_saopaulo attach-source FTP]
#-----

#===== R S V P =====
# Create the same session on all sender nodes
$rsvp2 session $bordaRSVP1 0
$rsvp3 session $bordaRSVP1 0

set counter 0
# Start the procedure which logs the rate for the incoming
# flow on node 0
$ns at 0.0 "record"

$ns at 1.0 "$rsvp2 sender 0 +500000 5000 32"
$ns at 1.0 "$rsvp3 sender 0 +500000 5000 32"
# Perform test reservations
$ns at 2.0 "$rsvp0 reserve 0 ff +300000 100000 $fonteRSVP1"
$ns at 2.0 "$rsvp0 reserve 0 ff +200000 100000 $fonteRSVP2"
#=====
#===== Cria os Monitores para delay e Jitter =====
newPktStats $portoalegre $saopaulo $udp0 $tanque1 medidorn7.log
newPktStats $portoalegre $saopaulo $udp1 $tanque2 medidorn8.log
#=====

#===== Cria os Mapeadores/Medidores =====
New-medudp $bordaRSVP1 $bordaDS1 $udp0 $tanque1 1 2 0.1 2.0
New-medudp $bordaRSVP1 $bordaDS1 $udp1 $tanque2 1 2 0.1 2.0
New-medudp $bordaDS2 $recife $udp0 $tanque1 2 3 0.1 2.0
New-medudp $bordaDS2 $recife $udp1 $tanque2 2 3 0.1 2.0
#=====

#===== Sequencia da Simulacao =====
$ns at 0.0 "$video start"
$ns at 70.0 "$video stop"
$ns at 0.0 "$voz start"
$ns at 70.0 "$voz stop"
```

```
$ns at 0.0 "$ftp_be start"
$ns at 70.0 "$ftp_be stop"
$ns at 00.0 "$cbr_ds1 start"
$ns at 70.0 "$cbr_ds1 stop"
$ns at 00.0 "$cbr_ds2 start"
$ns at 70.0 "$cbr_ds2 stop"
$ns at 00.0 "$ftp1_be start"
$ns at 70.0 "$ftp1_be stop"
$ns at 00.0 "$ftp2_be start"
$ns at 70.0 "$ftp2_be stop"
$ns at 00.0 "$ftp3_be start"
$ns at 70.0 "$ftp3_be stop"

#-----perifericos -----
$ns at 0.1 "$udp_belem start"
$ns at 0.1 "$udp_manaus start"
$ns at 0.1 "$udp_goiania start"
$ns at 0.1 "$udp_campogrande start"
$ns at 0.1 "$ftp_curitiba start"
$ns at 0.1 "$ftp_saopaulo start"
$ns at 70.0 "$udp_belem stop"
$ns at 70.0 "$udp_manaus stop"
$ns at 70.0 "$udp_goiania stop"
$ns at 70.0 "$udp_campogrande stop"
$ns at 70.0 "$ftp_curitiba stop"
$ns at 70.0 "$ftp_saopaulo stop"

$ns at 70.0 "finish"
#=====

#===== Grava resultados para gerar os graficos =====
proc record {} {
    global tanque1 tanque2 tanque3 ns f0 f1 f2 counter
    set time 0.5
    set bw0 [$tanque1 set bytes_]
    set bw1 [$tanque2 set bytes_]
    set bw2 [$tanque3 set bytes_]
    set now [$ns now]
    if { $now != 0 && [expr $bw0/$time*8] != 0.0 } {
        puts $f0 "$now [expr $bw0/$time*8]"
    }
    if { $now != 0 && [expr $bw1/$time*8] != 0.0 } {
        puts $f1 "$now [expr $bw1/$time*8]"
    }
    if { $now != 0 && [expr $bw2/$time*8] != 0.0 } {
        puts $f2 "$now [expr $bw2/$time*8]"
    }
    $tanque1 set bytes_ 0
    $tanque2 set bytes_ 0
    $tanque3 set bytes_ 0
    $ns at [expr $now+$time] "record"
    if { $counter == 10 } {
        puts "Time: $now"
        set counter 0
    }
}
```

```
    }
    incr counter
}
=====

#===== Finaliza =====
proc finish {} {
    global f0 f1 f2
    close $f0
    close $f1
    close $f2
    exec xgraph out0.tr out1.tr out2.tr -geometry 800x400 &
    exit 0
    puts "Done."
}
=====
#===== Executa (Ufa!) =====
$ns run
=====
```
