

Curso de Introdução Prática ao Simulador de Redes NS-2

Instrutor

Eduardo da Silva

Monitores

Elisa Mannes

Fernando H. Gielow

Urlan S. de Barros

Coordenador

Prof. Aldri L. dos Santos

Outubro de 2009



Roteiro

- Discussão dos exercícios anteriores
- Ambientes sem fio
- Ambientes móveis
- Novos exercícios

Exercício 1

Introduzir congestionamento no exemplo anterior, limitando o tamanho máximo da fila entre os roteadores em 15 pacotes

Adicionar ao código

```
$ns queue-limit $r0 $r1 15
```

- Cria um buffer de 15 pacotes entre os roteadores *r0* e *r1*

Exercício 2

Simular uma rede com 10 nós ligados em anel, usando um protocolo de roteamento dinâmico

- Criar duas fontes de tráfego CBR. Uma entre 1 e 4 e outra entre 2 e 5
- Simule um queda do enlace entre os nós 2 e 3 entre os tempos 3.0 e 5.0
- Use o comando “\$ns rtproto DV” para o roteamento dinâmico

Exercício 2

Criação dos nós e enlaces

```
for {set i 0} {$i < 10 } {incr i} {  
    set node_($i) [$ns node]  
}  
  
for {set i 0} {$i < 9 } {incr i} {  
    set p [expr $i+1]  
    $ns duplex-link $node_($i) $node_($p) \  
        10Mb 10ms DropTail  
}  
$ns duplex-link $node_(9) $node_(0) \  
    10Mb 10ms DropTail
```

Exercício 2

Manutenção do roteamento dinâmico

```
$ns rtpproto DV
```

```
$ns rtmodel-at 2.0 down $node_(2) $node_(3)
```

```
$ns rtmodel-at 4.0 up $node_(2) $node_(3)
```

- **Importante:** criar os agentes e as aplicações

Exercício 2

Código completo

```
set ns [new Simulator]
$ns color 1 Blue
$ns color 2 Red
```

```
set nf [open exer2.nam w]
$ns namtrace-all $nf
set f [open exer2.tr w]
$ns trace-all $f
```

```
proc finish {} {
    global ns nf f
    $ns flush-trace
    close $nf
    close $f
    exec nam exer2.nam &
    exit 0
}
```

```
for {set i 0} {$i < 10} {incr i} {
    set node_($i) [$ns node]
}
```

```
for {set i 0} {$i < 9} {incr i} {
    set p [expr $i+1]
    $ns duplex-link $node_($i) $node_($p) 10Mb 10ms DropTail
}
$ns duplex-link $node_(9) $node_(0) 10Mb 10ms DropTail
$ns rtpproto DV
```

```
$ns rtmodel-at 2.0 down $node_(2) $node_(3)
$ns rtmodel-at 4.0 up $node_(2) $node_(3)
```

```
set tcp0 [$ns create-connection \
    TCP $node_(1) TCPSink $node_(2) 2]
tcp0 set packetSize 552
set tcp1 [$ns create-connection \
    TCP $node_(0) TCPSink $node_(3) 1]
tcp0 set packetSize 552
```

```
set ftp0 [tcp0 attach-source FTP]
set ftp1 [tcp1 attach-source FTP]
```

```
$ns at 0.5 "$ftp1_start"
$ns at 1.0 "$ftp0_start"
$ns at 4.0 "$ftp0_stop"
$ns at 4.5 "$ftp1_stop"
$ns at 5.0 "finish"
$ns run
```

Exercícios 3

- ❶ Criar um cenário com as seguintes características:
 - Um servidor de FTP conectado a um roteador R1 a 100Mbps e atraso de 8ms
 - 5 clientes conectados a um roteador R2 a 16Mbps e atraso de 25ms
 - Roteador R1 e R2 interligados a 56Mbps e atraso de 12ms
 - Todos os cliente solicitam dados do servidor entre os tempos 0.5 e 4.5
- ❷ Para melhorar o desempenho do cliente 1, responda:
 - Ele deveria aumentar o seu enlace para 25Mbps ou 50Mbps?
 - Seria melhor investir num enlace *Fast-Path* que reduzisse o atraso para 15ms?
 - Poderia ser usada uma disciplina de enfileiramento para influenciar a decisão do usuário?

Exercício 3

Criação dos agentes e aplicações

```
for {set i 0} {$i < 4 } {incr i} {  
  set tcp_($i) [$ns create-connection \  
    TCP $node_(6) TCPSink $node_($i) $i]  
  $tcp_($i) set packetSize 552  
  set ftp_($i) [$tcp_($i) attach-source FTP]  
  $ns at 0.5 "$ftp_($i)_start"  
  $ns at 4.5 "$ftp_($i)_stop"  
}
```

Exercício 3

Código completo

```
set ns [new Simulator]
$ns color 0 Cyan
$ns color 1 Blue
$ns color 2 Red
$ns color 3 Yellow

set nf [open exe3.nam w]
$ns namtrace-all $nf
set f [open exe3.tr w]
$ns trace-all $f

proc finish {} {
global ns nf f
$ns flush-trace
close $nf
close $f
exec nam exe3.nam &
exit 0
}

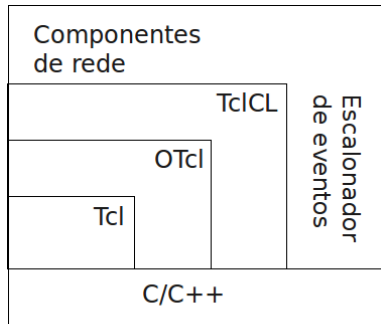
for {set i 0} {$i < 7} {incr i} {
set node_($i) [$ns node]

for {set i 0} {$i < 4} {incr i} {
$ns duplex-link $node_($i) $node_(4) \
16Mb 25ms DropTail
}
$ns duplex-link $node_(4) $node_(5) \
56Mb 12ms DropTail
$ns duplex-link $node_(5) $node_(6) \
100Mb 8ms DropTail

for {set i 0} {$i < 4} {incr i} {
set tcp_($i) [$ns create-connection \
TCP $node_(6) TCPSink $node_($i) $i]
$tcp_($i) set packetSize 552
set ftp_($i) [$tcp_($i) attach-source FTP]
$ns at 0.5 "$ftp_($i)_start"
$ns at 4.5 "$ftp_($i)_stop"
}
$ns at 5.0 "finish"
$ns run
}
```

Arquitetura do simulador

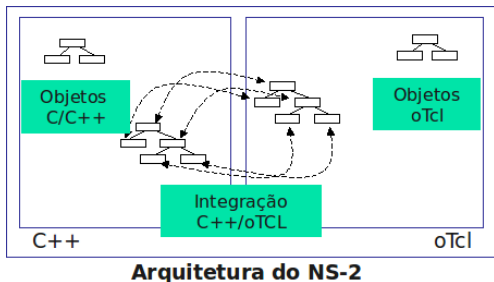
- Tcl e oTcl
 - linguagem de *scripts*
 - configuração
 - oTcl → Tcl orientado a objetos
- C/C++
 - linguagem de sistema
 - programação do *core*
- TclCl (Tcl Class)
 - fornece interface entre o C/C++ e o Tcl



Arquitetura do NS-2

Por que duas linguagens?

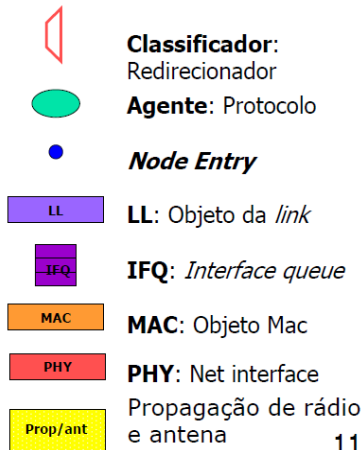
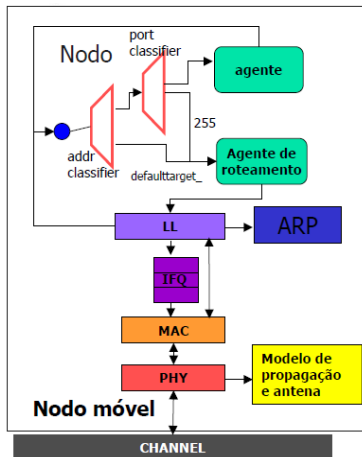
- C++: execução rápida / alteração complexa
 - processamento dos pacotes, estruturas e algoritmos complexos
 - código com poucas alterações
- oTcl: alteração simples / execução lenta
 - configuração, escalonamento de eventos periódicos
 - código com alterações frequentes
- Tclcl → uma forma de unir as duas linguagens



Nó móvel

- Nó móvel:
 - possui endereço, porta e algoritmo de roteamento
- Diferem de nós fixos
 - não compartilham enlaces pré-definidos com os demais nós
- São capazes de se movimentar, e transmitir e receber sinais por canais sem fio
- Possibilitam simulações de MANETs, WSNs, LANs sem fio, etc.
- Componentes da rede relacionados/associados:
 - LL, MAC, NetIF, modelo de rádio, etc

Esquema do Nó móvel



11

Um exemplo de rede sem fio

```

set ns_ [new Simulator]
set fd [open exe4.tr w]
$ns_ trace-all $fd
set nfd [open exe4.nam w]
$ns_ namtrace-all-wireless $nfd 500 500
proc stop {} {
    global ns_ fd
    $ns_ flush-trace
    close $fd
}

```

```

set topo [new Topography]
$topo load_flatgrid 500 500
set chan [new Channel/WirelessChannel]

```

```

$ns_ node-config \
    -adhocRouting DSDV \
    -llType LL \
    -macType Mac/802_11 \
    -ifqType Queue/DropTail/PriQueue \
    -ifqLen 50 \
    -antType Antenna/OmniAntenna \
    -propType Propagation/TwoRayGround \
    -phyType Phy/WirelessPhy \
    -channelType Channel/WirelessChannel \
    -topoInstance $topo \
    -agentTrace ON \
    -routerTrace ON \
    -macTrace OFF \
    -movementTrace OFF

```

```

for {set i 0} {$i < 2} {incr i} {
    set node_($i) [$ns_ node]
    $node_($i) random-motion 0
    $ns_ at 170.0 "$node_($i)_reset";
}

```

```

$node_(0) set X_ 5.0
$node_(0) set Y_ 2.0
$node_(0) set Z_ 0.0

```

```

$node_(1) set X_ 390.0
$node_(1) set Y_ 385.0
$node_(1) set Z_ 0.0

```

```

$ns_ at 10.0 "$node_(0)_setdest_20.0_18.0_1.0"
$ns_ at 50.0 "$node_(1)_setdest_25.0_20.0_15.0"
$ns_ at 100.0 "$node_(1)_setdest_490.0_480.0_15.0"

```

```

set tcp [$ns_ create-connection \
    TCP $node_(0) TCPSink $node_(1) $i]
set ftp [$tcp attach-source FTP]
$ns_ at 20.0 "$ftp_start"

```

```

$ns_ at 170.0 "stop"
$ns_ at 170.01 "puts_ \"NS EXITING...\";_;$ns_ halt"

```

```

$ns_ run

```

Alteração para registro no NAM

Um exemplo de rede sem fio

```

set ns_ [new Simulator]
set fd [open exe4.tr w]
$ns_ trace-all $fd
set nfd [open exe4.nam w]
$ns_ namtrace-all-wireless $nfd 500 500
proc stop {} {
    global ns_ fd
    $ns_ flush-trace
    close $fd
}

```

```

set topo [new Topography]
$topo load_flatgrid 500 500
set chan [new Channel/WirelessChannel]

```

```

$ns_ node-config \
    -adhocRouting DSDV \
    -llType LL \
    -macType Mac/802_11 \
    -ifqType Queue/DropTail/PriQueue \
    -ifqLen 50 \
    -antType Antenna/OmniAntenna \
    -propType Propagation/TwoRayGround \
    -phyType Phy/WirelessPhy \
    -channelType Channel/WirelessChannel \
    -topoInstance $topo \
    -agentTrace ON \
    -routerTrace ON \
    -macTrace OFF \
    -movementTrace OFF

```

```

for {set i 0} {$i < 2} {incr i} {
    set node_($i) [$ns_ node]
    $node_($i) random-motion 0
    $ns_ at 170.0 "$node_($i)_reset";
}

```

```

$node_(0) set X_ 5.0
$node_(0) set Y_ 2.0
$node_(0) set Z_ 0.0

```

```

$node_(1) set X_ 390.0
$node_(1) set Y_ 385.0
$node_(1) set Z_ 0.0

```

```

$ns_ at 10.0 "$node_(0)_setdest_20.0_18.0_1.0"
$ns_ at 50.0 "$node_(1)_setdest_25.0_20.0_15.0"
$ns_ at 100.0 "$node_(1)_setdest_490.0_480.0_15.0"

```

```

set tcp [$ns_ create-connection \
    TCP $node_(0) TCPSink $node_(1) $i]
set ftp [$tcp attach-source FTP]
$ns_ at 20.0 "$ftp_start"

```

```

$ns_ at 170.0 "stop"
$ns_ at 170.01 "puts_ \"NS EXITING...\";_;$ns_ halt"

```

```

$ns_ run

```


Configuração dos nós

Um exemplo de rede sem fio

```

set ns_ [new Simulator]
set fd [open exe4.tr w]
$ns_ trace-all $fd
set nfd [open exe4.nam w]
$ns_ namtrace-all-wireless $nfd 500 500
proc stop {} {
    global ns_ fd
    $ns_ flush-trace
    close $fd
}

```

```

set topo [new Topography]
$topo load_flatgrid 500 500
set chan [new Channel/WirelessChannel]

```

```

$ns_ node-config \
    -adhocRouting DSDV \
    -llType LL \
    -macType Mac/802_11 \
    -ifqType Queue/DropTail/PriQueue \
    -ifqLen 50 \
    -antType Antenna/OmniAntenna \
    -propType Propagation/TwoRayGround \
    -phyType Phy/WirelessPhy \
    -channelType Channel/WirelessChannel \
    -topoInstance $topo \
    -agentTrace ON \
    -routerTrace ON \
    -macTrace OFF \
    -movementTrace OFF

```

```

for {set i 0} {$i < 2} {incr i} {
    set node_($i) [$ns_ node]
    $node_($i) random-motion 0
    $ns_ at 70.0 "$node_($i)_reset";
}

```

```

$node_(0) set X_ 5.0
$node_(0) set Y_ 2.0
$node_(0) set Z_ 0.0

```

```

$node_(1) set X_ 390.0
$node_(1) set Y_ 385.0
$node_(1) set Z_ 0.0

```

```

$ns_ at 10.0 "$node_(0)_setdest_20.0_18.0_1.0"
$ns_ at 50.0 "$node_(1)_setdest_25.0_20.0_15.0"
$ns_ at 100.0 "$node_(1)_setdest_490.0_480.0_15.0"

```

```

set tcp [$ns_ create-connection \
    TCP $node_(0) TCPSink $node_(1) $i]
set ftp [$tcp attach-source FTP]
$ns_ at 20.0 "$ftp_start"

```

```

$ns_ at 170.0 "stop"
$ns_ at 170.01 "puts_ \"NS EXITING...\";_;$ns_ halt"

```

```

$ns_ run

```

Posicionamento inicial dos nós

Um exemplo de rede sem fio

```

set ns_ [new Simulator]
set fd [open exe4.tr w]
$ns_ trace-all $fd
set nfd [open exe4.nam w]
$ns_ namtrace-all-wireless $nfd 500 500
proc stop {} {
    global ns_ fd
    $ns_ flush-trace
    close $fd
}

```

```

set topo [new Topography]
$topo load_flatgrid 500 500
set chan [new Channel/WirelessChannel]

```

```

$ns_ node-config \
    -adhocRouting DSDV \
    -llType LL \
    -macType Mac/802_11 \
    -ifqType Queue/DropTail/PriQueue \
    -ifqLen 50 \
    -antType Antenna/OmniAntenna \
    -propType Propagation/TwoRayGround \
    -phyType Phy/WirelessPhy \
    -channelType Channel/WirelessChannel \
    -topoInstance $topo \
    -agentTrace ON \
    -routerTrace ON \
    -macTrace OFF \
    -movementTrace OFF

```

```

for {set i 0} {i < 2} {incr i} {
    set node_($i) [$ns_ node]
    $node_($i) motion-motion 0
    $ns_ at 170.0 "$node_($i)_reset";
}

```

```

$node_(0) set X_ 5.0
$node_(0) set Y_ 2.0
$node_(0) set Z_ 0.0

```

```

$node_(1) set X_ 390.0
$node_(1) set Y_ 385.0
$node_(1) set Z_ 0.0

```

```

$ns_ at 10.0 "$node_(0)_setdest_20.0_18.0_1.0"
$ns_ at 50.0 "$node_(1)_setdest_25.0_20.0_15.0"
$ns_ at 100.0 "$node_(1)_setdest_490.0_480.0_15.0"

```

```

set tcp [$ns_ create-connection \
    TCP $node_(0) TCPSink $node_(1) $i]
set ftp [$tcp attach-source FTP]
$ns_ at 20.0 "$ftp_start"

```

```

$ns_ at 170.0 "stop"
$ns_ at 170.01 "puts_ \"NS EXITING...\";_;$ns_ halt"

```

```

$ns_ run

```

Movimentação dos nós

Um exemplo de rede sem fio

```

set ns_ [new Simulator]
set fd [open exe4.tr w]
$ns_ trace-all $fd
set nfd [open exe4.nam w]
$ns_ namtrace-all-wireless $nfd 500 500
proc stop {} {
    global ns_ fd
    $ns_ flush-trace
    close $fd
}

```

```

set topo [new Topography]
$topo load_flatgrid 500 500
set chan [new Channel/WirelessChannel]

```

```

$ns_ node-config \
    -adhocRouting DSDV \
    -llType LL \
    -macType Mac/802_11 \
    -ifqType Queue/DropTail/PriQueue \
    -ifqLen 50 \
    -antType Antenna/OmniAntenna \
    -propType Propagation/TwoRayGround \
    -phyType Phy/WirelessPhy \
    -channelType Channel/WirelessChannel \
    -topoInstance $topo \
    -agentTrace ON \
    -routerTrace ON \
    -macTrace OFF \
    -movementTrace OFF

```

```

for {set i 0} {i < 2} {incr i} {
    set node_($i) [$ns_ node]
    $node_($i) random-motion 0
    $ns_ at 170.0 "$node_($i)_reset";
}

```

```

$node_(0) set X_ 5.0
$node_(0) set Y_ 2.0
$node_(0) set Z_ 0.0

```

```

$node_(1) set X_ 390.0
$node_(1) set Y_ 385.0
$node_(1) set Z_ 0.0

```

```

$ns_ at 10.0 "$node_(0)_setdest_20.0_18.0_1.0"
$ns_ at 50.0 "$node_(1)_setdest_25.0_20.0_15.0"
$ns_ at 100.0 "$node_(1)_setdest_490.0_480.0_15.0"

```

```

set tcp [$ns_ create-connection \
    TCP $node_(0) TCPSink $node_(1) $i]
set ftp [$tcp attach-source FTP]
$ns_ at 20.0 "$ftp_start"

```

```

$ns_ at 170.0 "stop"
$ns_ at 170.01 "puts_ \"NS EXITING...\";_;$ns_ halt"

```

```

$ns_ run

```

Exercício

Descrição do cenário

- 1 Elaborar um script e simular uma rede ad hoc com três nós em linha reta
- 2 Criar uma conexão UDP entre os dois nós da ponta
- 3 Gerar um tráfego CBR de um nó ao outro.
- 4 Movimentar os nós de maneira que, até a metade do tempo de simulação, os nós da ponta se comuniquem por intermédio do nó do meio
- 5 Em seguida, movimentar para que eles não precisem mais do nó intermediário

Dica: Por padrão, o alcance do rádio é de 250m, por empregar uma potência de 200mW

Modelo de mobilidade

■ Random Waypoint

- nó permanece parado durante um período de tempo aleatório
- escolhe um ponto aleatório na rede
- escolhe uma velocidade aleatória
- em seguida, segue esse ponto em linha reta e velocidade constante
- chegando lá, permanece parado por um tempo aleatório
- repete todo o processo

■ Existem outros modelos, mas esse é o mais utilizado

O comando setdest

- Gerador de movimento disponível em ns/indep-utils/cmu-scen-gen/setdest
- Gera movimentação no padrão RandomWayPoint
- Sintaxe:

```
./setdest -n num_nodes -p pausetime -s maxspeed -t simtime -x maxX -y maxY
```

- Resultados
 - gerados na saída padrão
 - iniciam com a posição inicial dos nós
- Exemplo:

```
./setdest -n 20 -p 2.0 -s 10.0 -t 200 -x 500 -y 500 > scen-20-test
```

O script cbrgen

- Gerador de tráfego de dados CBR e TCP
- Disponível em ns/indep-utils/cmu-scen-gen
- Sintaxe:

```
ns cbrgen.tcl -type cbr|tcp -nn nodes -seed seed -mc connections -rate rate
```

- Exemplo de tráfego CBR

```
ns cbrgen.tcl -type cbr -nn 20 -seed 1 -mc 8 -rate 4
```

- Exemplo de tráfego TCP

```
ns cbrgen.tcl -type tcp -nn 15 -seed 0 -mc 6
```

Carregando os arquivos gerados

- Usar comando *source* no script oTcl
- Carregando o arquivo de movimentação
 - **source <arquivo-de-movimentação>**
- Carregando o arquivo de tráfego
 - **source <arquivo-de-trafego>**

Exercícios

- Um cenário com 15 nós móveis
- Duas conexões do tipo CBR, com 4 pacotes por segundo de 500 KB
- Realizar simulações usando os protocolos DSDV, AODV e DSR
- Avaliar:
 - ① a taxa de entrega das mensagens
 - ② a sobrecarga do protocolo de roteamento
 - ③ a latência na entrega das mensagens
- Calcular a média de 10 simulações para cada experimento

OBRIGADO!