

## Simulador de redes ns

José Rezende  
GTA/COPPE/UFRJ

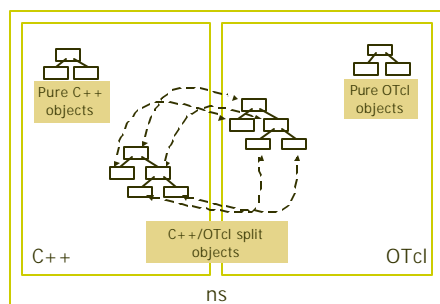


## Arquitetura do ns

- Orientado a objeto (C++, OTcl): fina granulosidade
  - Facilita a reutilização e a manutenção
- C++ para o tratamentos dos dados
  - Ação por pacote
  - Desempenho e uso de memória
- OTcl para a parte de controle
  - Tcl orientado a objeto
  - Ações periódicas ou disparadas por um evento
- TclCL
  - "fusão" de C++ e Otcl

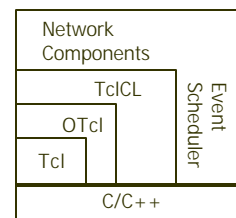


## OTcl e C++



## Componentes do ns-2

- Escalonador de eventos discretos
- Componentes de rede
  - Camada de enlace e superiores
  - Suporte a emulação



ns-2



## Elementos do ns-2

- Criar o escalonador de eventos
- Habilitar trace
- Criar rede
- Estabelecer o roteamento
- Criar conexões de Transporte
- Criar geradores de tráfego



## Criação de uma Simulação

- Criar escalonador de eventos
  - set ns [new Simulator]
- Escalonar eventos
  - \$ns at <time> <event>
  - <event>: qualquer comando ns/tcl válido
- Iniciar escalonador
  - \$ns run



## Geração de *traces*

- Trace de pacotes em todos os enlaces
  - \$ns trace-all [open test.out w]

```
<event> <time> <from> <to> <pkt> <size> - <fid> <src> <dst> <seq> <attr>
+ 1 0.2 cbr 210 ----- 0 0.0 3.1 0 0
- 1 0.2 cbr 210 ----- 0 0.0 3.1 0 0
r 1.00234 0.2 cbr 210 ----- 0 0.0 3.1 0 0
```
- Trace de pacotes em todos os enlaces no formato nam
  - \$ns namtrace-all [open test.nam w]
- Trace em enlaces específicos
  - \$ns trace-queue \$n0 \$n1
  - \$ns namtrace-queue \$n0 \$n1



## Criação da rede

- Nós
  - set n0 [\$ns node]
  - set n1 [\$ns node]
- Enlaces e filas
  - \$ns duplex-link \$n0 \$n1 <bandwidth> <delay> <queue\_type>
  - <queue\_type>: DropTail, RED, CBQ, FQ, SFQ, DRR
- Redes Locais (LANs)
  - \$ns make-lan <node\_list> <b w> <delay> <ll\_type>
  - <ll\_type>: LL
  - <ll\_type>: Queue/DropTail,
  - <mac\_type>: MAC/802\_11
  - <channel\_type>: Channel



## Dinâmica da rede

- Falhas dos enlaces
  - mudanças no roteamento

- Quatro modelos

```
$ns rtmodel Trace <config_file> $n0 $n1
$ns rtmodel Exponential {<params>} $n0 $n1
$ns rtmodel Deterministic {<params>} $n0 $n1
$ns rtmodel-at <time> up|down $n0 $n1
```



## Estabelecer o roteamento

- Unicast
  - \$ns rtrproto <type>
  - <type>: Static, Session, DV, cost, multi-path
- Multicast
  - \$ns multicast (right after [new Simulator])
  - \$ns mrtproto <type>
  - <type>: CtrMcast, DM, ST, BST



## Criação de conexões

- UDP
  - set udp [new Agent/UDP]
  - set null [new Agent/Null]
  - \$ns attach-agent \$n0 \$udp
  - \$ns attach-agent \$n1 \$null
  - \$ns connect \$udp \$null
- TCP
  - set tcp [new Agent/TCP]
  - set tcpsink [new Agent/TCPSink]
  - \$ns attach-agent \$n0 \$tcp
  - \$ns attach-agent \$n1 \$tcpsink
  - \$ns connect \$tcp \$tcpsink



## Criação de geradores de tráfego

- em cima de UDP
  - CBR
    - set src [new Application/Traffic/CBR]
  - Exponential or Pareto on-off
    - set src [new Application/Traffic/Exponential]
    - set src [new Application/Traffic/Pareto]
- em cima de TCP
  - FTP
    - set ftp [new Application/FTP]
    - \$ftp attach-agent \$tcp
  - Telnet
    - set telnet [new Application/Telnet]
    - \$telnet attach-agent \$tcp



## Criação de tráfego: a partir de um *trace*

- **Trace driven**
  - set tfile [new Tracefile]
  - \$tfile filename <file>
  - set src [new Application/Traffic/Trace]
  - \$src attach-tracefile \$tfile
- **<file>:**
  - intervalo entre os pacotes (msec) e tamanho do pacote(bytes)



## Estrutura Genérica de um script

```
set ns [new Simulator]
# [habilita tracing]
# Cria topologia
# Estabelece modelos de erros, dinâmica dos enlaces
# Cria agentes de roteamento
# Cria:
#   - grupos multicast
#   - agentes de protocolos
#   - fontes de tráfego
#   - procedures de processamento pós-simulação
# Inicia simulação
```



## Redes sem-fio no ns

- Roteamento Ad hoc
- Mobile IP
- Redes de Sensores
- Redes de Satélite
- Redes Celulares: GPRS, TDMA

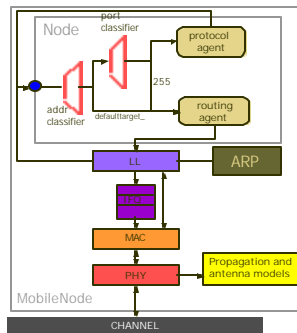


## Redes sem-fio no ns

- **Nó móvel: Abstração**
  - Localização
    - Coordenadas (x,y,z)
  - Movimento
    - Velocidade, direção, localização inicial e fina
- **Canal sem fio**
  - entrega pacotes a todos os nós móveis ligados ao canal exceto o emissor
  - receptor decide se deve aceitar ou não o pacote
    - Colisões gerenciadas individualmente no receptor
    - Erros de transmissão



## Estrutura de um nó móvel



## Nó móvel: Componentes

- Subcamada LLC
  - módulo ARP separado
- Fila da interface
  - Prioridade aos pacotes de roteamento
- Camada MAC
  - IEEE 802.11
- Camada Física ou itf de rede (PHY)
  - Parâmetros baseados no DSSS
  - Inclui modelos de antenas e de propagação do sinal
  - Contabilização do consumo de energia
- Modelo de propagação de rádio
  - Atenuação Friss-space ( $1/r^2$ ) em pequenas distâncias
  - Two-ray Ground ( $1/r^4$ ) distâncias longas
- Antena
  - Omni-directional



## Roteamento Ad Hoc: Cenário

- 3 nós móveis
- se movendo dentro de uma topologia plana de 670mX670m
- usando o protocolo de roteamento ad hoc DSDV
- modelo de mobilidade Random Waypoint
- Tráfego TCP e CBR



## Exemplo – Passo 1

```
# Define Global Variables
# create simulator
set ns [new Simulator]

# create a topology in a 670m x 670m area
set topo [new Topography]
$topo load_flatgrid 670 670

# Define standard ns/nam trace
# ns trace
set tracefd [open demo.tr w]
$ns trace-all $tracefd

# nam trace
set namtrace [open demo.nam w]
$ns namtrace-all-wireless $namtrace 670 670
```



## Exemplo – Passo 2

```
# Create God
set god [create-god 3]
```

- Armazena um array do menor número de saltos para um nó alcançar um outro nó
- Automaticamente gerado pelo arquivo de cenário
- Permite comparações de um protocolo de roteamento com o caso ótimo



## Exemplo – Passo 3

```
# Define how a mobile node should be created
$ns node-config \
  -adhocRouting DSDV \
  -llType LL \
  -macType Mac/802_11 \
  -ifqLen 50 \
  -ifqType Queue/DropTail/PriQueue \
  -antType Antenna/OmniAntenna \
  -propType Propagation/TwoRayGround \
  -phyType Phy/WirelessPhy \
  -channelType Channel/WirelessChannel \
  -topoInstance $topo
-agentTrace ON \
-routerTrace OFF \
-macTrace OFF
```



## Exemplo – Passo 4

```
# Create 3 mobile nodes
for {set i 0} {$i < 3} {incr i} {
  set node($i) [$ns node]
  # disable random motion
  $node(0) random-motion 0
}

# Define node movement model
source movement-scenario-files

# Define traffic model
source traffic-scenario-files

# Tell ns/nam the simulation stop time
$ns at 200.0 "$ns nam-end-wireless 200.0"
$ns at 200.0 "$ns halt"

# Start your simulation
$ns run
```



## Geradores de Cenário: Movimento e Tráfego

### ● Gerador de movimento

```
setdest -n <num_of_nodes> -p pausetime -s
<maxspeed> -t <simtime> -x <maxx> -y <maxy>
```

### ● Gerador de padrões de tráfego

```
- CBR traffic
ns cbrgen.tcl [-type cbr/tcp] [-nn nodes]
[-seed seed] [-mc connections] [-rate rate]

- TCP traffic
ns tcpgen.tcl [-nn nodes] [-seed seed]
```



## Um cenário de movimento

```
$node_(2) set Z_ 0.000000000000
$node_(2) set Y_ 199.373306816804
$node_(2) set X_ 591.256560093833
$node_(1) set Z_ 0.000000000000
$node_(1) set Y_ 345.357731779204
$node_(1) set X_ 257.046298323157
$node_(0) set Z_ 0.000000000000
$node_(0) set Y_ 239.438009831261
$node_(0) set X_ 83.364418416244
$ns_ at 50.000000000000 "$node_(2) setdest 369.463244915743
170.519203111152 3.371785699154"
$ns_ at 51.000000000000 "$node_(1) setdest 221.826585497093
80.855495003839 14.909259208114"
$ns_ at 33.000000000000 "$node_(0) setdest 89.663708107313
283.494644426442 19.153832288917"
```



## Um cenário de tráfego

```
set udp_(0) [new Agent/UDP]
$ns_ attach-agent $node_(0) $udp_(0)
set null_(0) [new Agent/Null]
$ns_ attach-agent $node_(2) $null_(0)
set cbr_(0) [new Application/Traffic/CBR]
$cbr_(0) set packetSize_ 512
$cbr_(0) set interval_ 4.0
$cbr_(0) set random_ 1
$cbr_(0) set maxpkts_ 10000
$cbr_(0) attach-agent $udp_(0)
$ns_ connect $udp_(0) $null_(0)
$ns_ at 127.93667922166023 "$cbr_(0) start"
.....
```



## Ferramentas de Visualização

- **nam (Network AniMator)**
  - Animação no nível de pacotes
  - Topologia
  - Controle da animação
- **xgraph**
  - Conversão do trace do ns para o formato xgraph
- **gnuplot**
  - Geração de curvas



## Interface ns→nam

- Cores
- Manipulação de nós e enlaces
- *Layout* da topologia
- Estado do Protocolo
- Miscelâneos



## Interface nam : Cores

- Mapeamento das cores

```
$ns color 40 red
$ns color 41 blue
$ns color 42 chocolate
```

- Associação Cor « flow id

```
$tcp0 set fid_ 40      ;# red packets
$tcp1 set fid_ 41      ;# blue packets
```



## Interface nam: Nós

- Cor

```
$node color red
```

- Forma (não pode ser mudada depois do início da simulação)

```
$node shape box ;# circle, box, hexagon
```

- Marcações (formas concêntricas)

```
$ns at 1.0 "$n0 add-mark m0 blue box"
$ns at 2.0 "$n0 delete-mark m0"
```

- Label

```
$ns at 1.1 "$n0 label \"web cache 0\""
```



## Interface nam : Enlaces

- Cor

```
$ns duplex-link-op $n0 $n1 color "green"
```

- Label

```
$ns duplex-link-op $n0 $n1 label "abcd"
```

- Dinâmica (controlada automaticamente)

```
$ns rtmodel Deterministic {2.0 0.9 0.1} $n0 $n1
```



## Interface nam : Layout da Topologia

- Manual

```
$ns duplex-link-op $n(0) $n(1) orient right
$ns duplex-link-op $n(1) $n(2) orient right
$ns duplex-link-op $n(2) $n(3) orient right
$ns duplex-link-op $n(3) $n(4) orient 60deg
```





## Interface nam: Misc

- **Anotações**

- Adicionar explicações da simulação
- `$ns at 3.5 "$ns trace-annotate \"packet drop\""`

- **Ajustar taxa de animação**

- `$ns at 0.0 "$ns set-animation-rate 0.1ms"`

