Criação de Cenários e Ambientes Sem Fio

Leonardo Barbosa e Oliveira

Curso de ns
Ciência da Computação UFMG
Julho/Agosto 2003

Sumário I

- LANs
  - Introdução
  - Componentes
  - LL *(Link Layer)*
- Redes Sem Fio
  - Introdução
  - Nodo Móvel
  - Roteamento *ad hoc*

Sumário II

- Redes Sem Fio (Continuação)
  - Modelo de Energia
  - Modelo de Propagação
  - Modelo de Mobilidade
  - Simulação
    - Gerador de Cenários
    - Exemplo TCL
- Prática
- Dúvidas
- Exercícios

LANs *(Local Area Networks)*

- Inerentemente diferente de redes ponto-a-ponto
- Permite o compartilhamento do canal
- Criação de novo tipo de nodo: LanNode
- Exemplo de criação:
  - Ethernet (CSMA/CD)

```
$ns make-lan "$n1 $n2" $bw $delay LL
  Queue/DropTail Mac/Csma/Cd
```

  - Nodos, largura de banda, latência,etc

Componentes de LAN

- LL
- MAC *(Medium Access Control)*
  - Próxima aula
- Camada física
  - Classe *Channel* implementa meio compartilhado

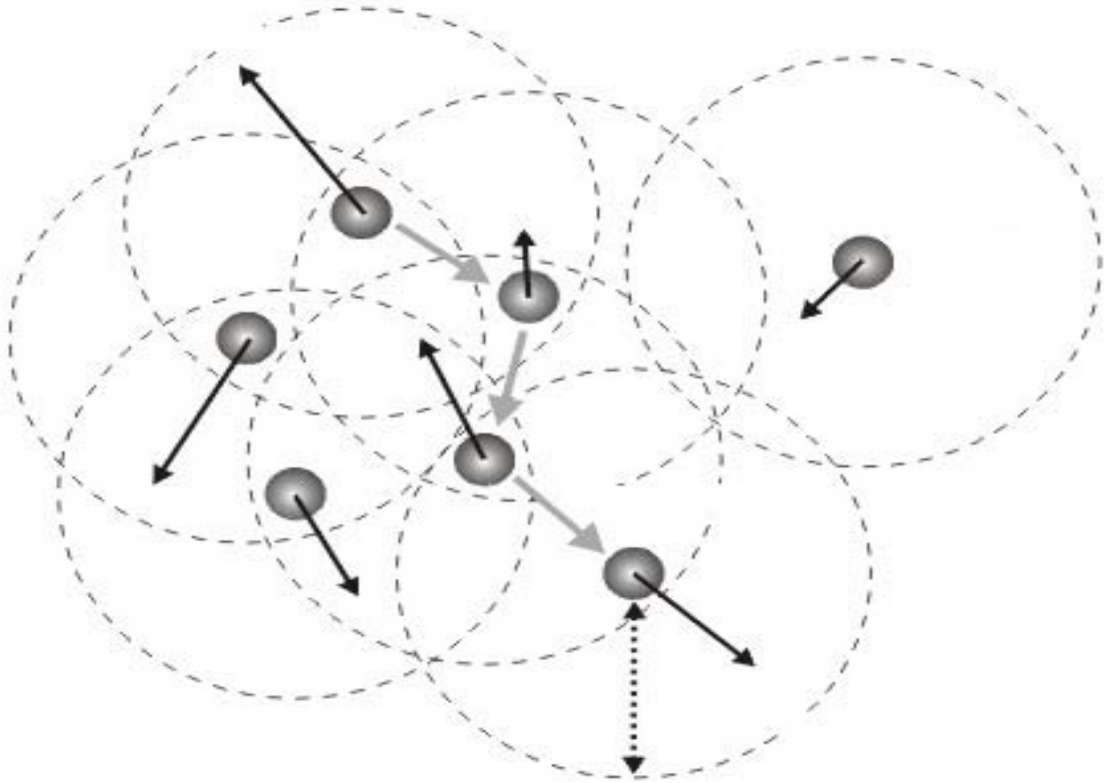
```
set channel_ [new Channel] $channel_
set delay_ 4us # propagation delay
```

LL (*Link Layer*)

- Protocolos da camada *data link* em geral
- Fragmentação e Agregação de dados
- Configuração do endereço destino MAC
  - Encontrar o endereço IP do próximo *hop*
  - "Resolução" de endereço IP para endereço físico (ARP)
  - Mapeamento endereço IP e MAC é um para um (simplicidade)

Computação Ubíqua no NS

MANET



Ambiente Sem Fio: curiosidades

- Contribuições para o NS:
  - Modelo original de mobilidade originado pelo *Monarch group* da CMU
  - Outras contribuições relevantes vieram da UCB, *Sun microsystems,* univ. de cincinnati, etc
  - Existem outros modelos:
    - BlueHoc
    - Mobiwan
    - GSM/GPRS

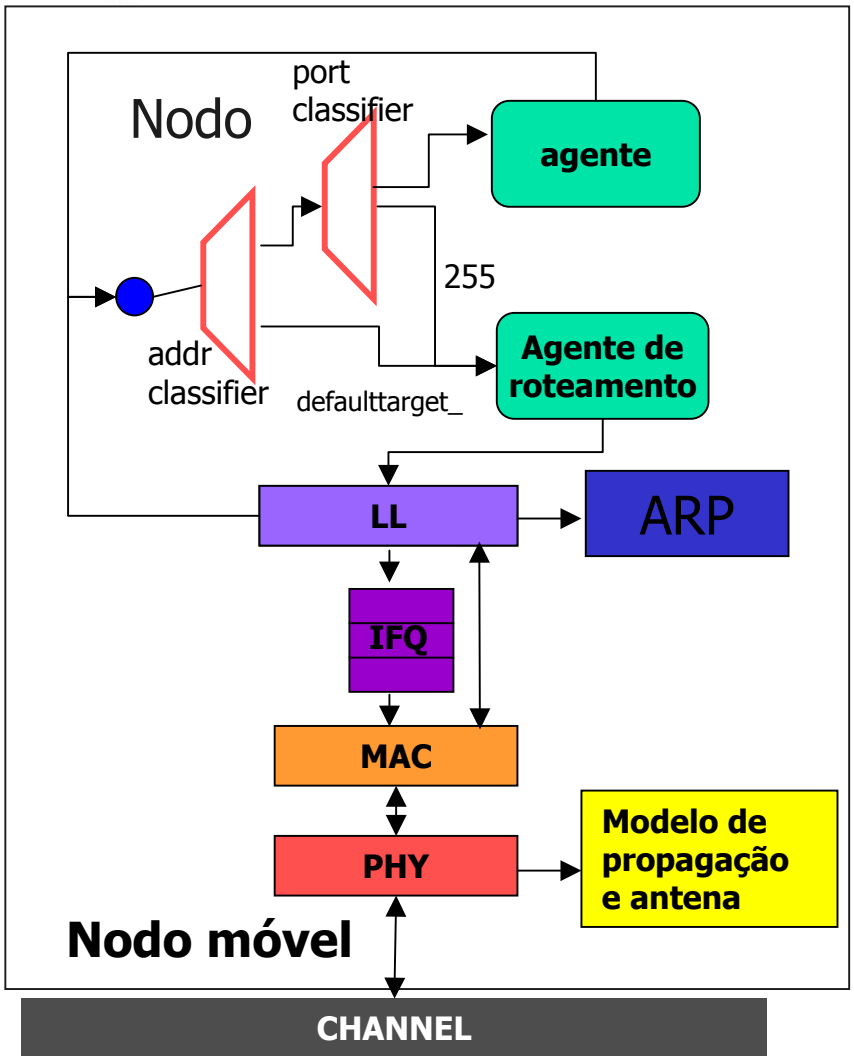
Nodo Móvel

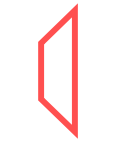
- Nodo móvel: entidade principal do modelo móvel
- Nodo que possui endereço, porto e algoritmo de roteamento
- Diferem de nodos fixos por não compartilharem enlaces pré-definidos com os demais nodos
- São capazes de se moverem dado uma topologia e transmitir e receber sinais através de canais sem fio

Nodo Móvel

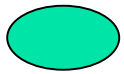
- Possibilitam simulações de MANETs, WSNs, LANs sem fio, etc
- Pilha de componentes da rede consiste de LL, MAC, NetIF, modelo de rádio, etc
- Visualização de movimento, alcance de transmissão, energia

Nodo Móvel: esquema





Classificador: Redirecionador



Agente: Protocolo

Node Entry

LL: Objeto da *link*

IFQ: *Interface queue*

MAC: Objeto Mac

PHY: Net interface

Propagação de rádio e antena

Nodo Móvel: Componentes

- Classificadores
  - *defaulttarget_* aponta para o agente de roteamento
  - 255 é o porto atribuído ao agente de roteamento *rtagent_*
- Agente de Roteamento
  - Pode ser um protocolo de roteamento *ad hoc* ou o de difusão direcionada

Nodo Móvel: Componentes

- Camada *Link*
  - Similar a de LAN, porém possui módulo diferente de ARP
  - Efetua requisições de endereços para ARP
- ARP
  - "Resolve" endereços IPs para endereços físicos (MAC)
  - Se conhece o endereço físico do destinatário, o escreve no cabeçalho pacote MAC
  - Caso contrário, armazena temporariamente o pacote e realiza *broadcast* a procura do endereço
- *Interface queue*
  - Dá prioridade à pacotes de protocolos de roteamento
  - Capacidade de filtrar pacotes – baseado em endereços destinos

Nodo Móvel: Componentes

- MAC
  - 802.11
    - Envia dado através de broadcast
  - SMAC (trabalho em andamento)
    - Utilizado em RSSF
    - Permite padrão de dormência (*sleep pattern)*
    - Reduz consumo de energia durante períodos ociosos (*idle times)*
  - TDMA (GSM, GPRS)

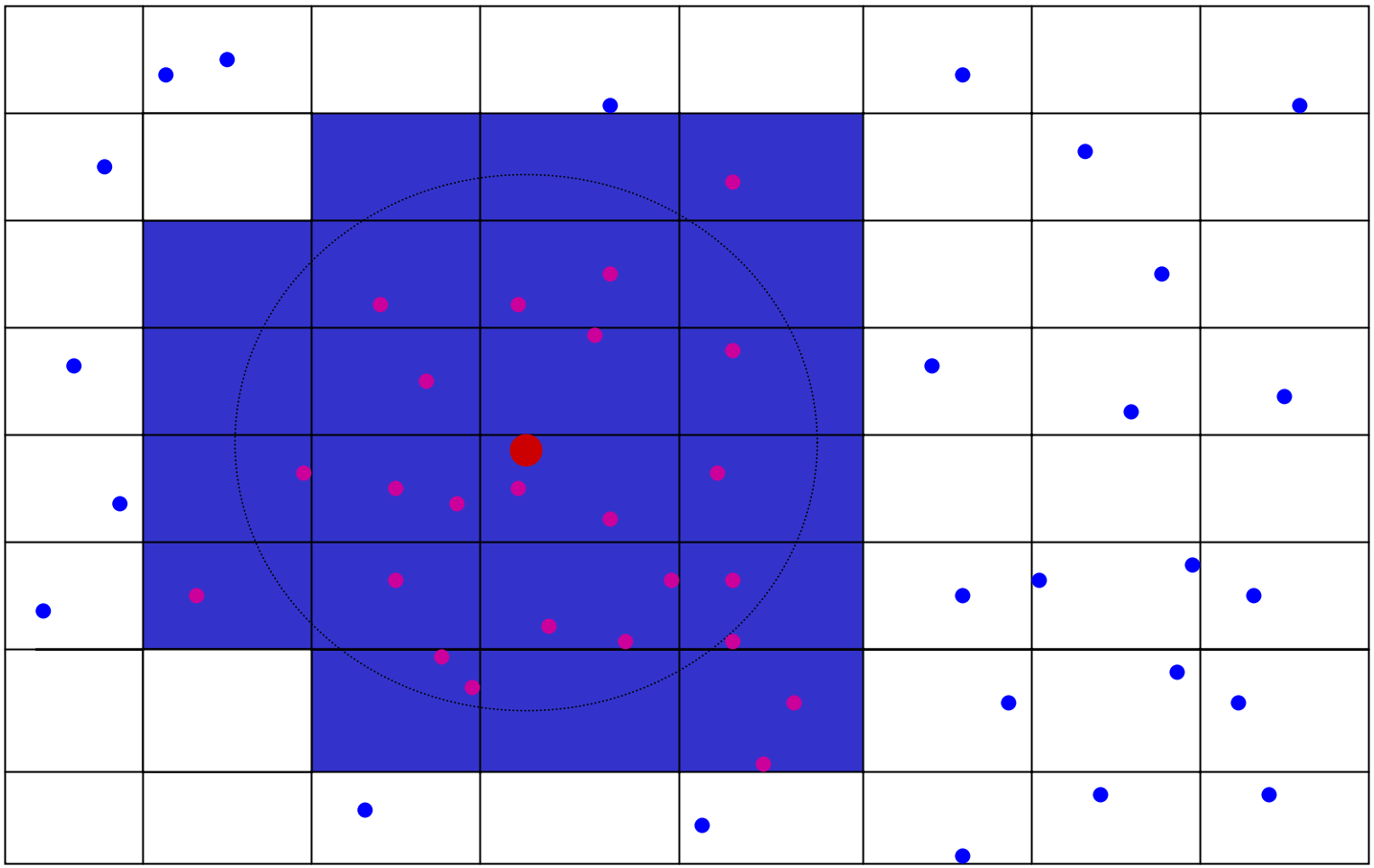
Nodo Móvel: Componentes

- Interface de Rede (*Network interface*)
  - Utilizada pelo nodos móveis para acessar o canal
  - Preenche pacotes encaminhados com meta-dados (comprimento de onda, energia, etc) para tratamento pelo modelo de propagação na chegada
  - Interface com modelo de rádio e antena
- Antena
  - *Omni*-direcional

Canal Sem Fio

- Duplica pacotes para todos nodos móveis do canal (com exceção do remetente)
- Cabe ao receptor decidir se aceita ou não o pacote
  - Colisão é tratada em cada receptor individualmente
  - Verificação se está em seu raio de alcance

Grid



Suporte a *Trace*

- Originalmente existia o estilo desenvolvido pela CMU
- Mais tarde foi desenvolvido um formato de *trace* específico para redes sem fio
- Atualmente existe um esforço para unir ambos os formatos

Trace: formato original

- Exemplo:

  r 160.093884945 _6_ RTR --- 5 tcp 1492 [a2 4 6 800] ---------- [655 36:0 16777984:0 31 16777984] [1 0] 2 0
  - Nodo 6 recebe pacote TCP
  - ID do pacote: 5
  - TTL: 31
  - IP src: 0.1.0
  - IP dst: 1.0.3

Trace: formato sem fio I

- Adicione ao *script* o comando:

```
$ns use-newtrace
```

- Deve ser colocado antes do comando

```
$ns trace-all
```

- Baseado em *tags*
  - ***Ni*** *node id*
  - ***Nx*** *node's x-coordinate*
  - ***Ny*** *node's y-coordinate*
  - ***Nz*** *node's z-coordinate*

Trace: formato sem fio II

- Amostra

  s -t 0.267662078 -Hs 0 -Hd -1 -Ni 0 -Nx 5.00 -Ny 2.00 -Nz 0.00 -Ne -1.000000 -Nl RTR -Nw --- -Ma 0 -Md 0 -Ms 0 -Mt 0 -Is 0.255 -Id -1.255 -It message -Il 32 -If 0 -Ii 0 -Iv 32

- Melhorou, mas ainda não intuitivo
  - -Il ??

Trace: formato sem fio III

- Resposta: tamanho do pacote

Trace: do it yourself

```
sprintf(pt_->buffer() + offset, "%c -t %.9f -Eu %d -
  nexthop %d -Ni %d -Nx %.2f -Ny %.2f -Nz %.2f -
  Ne %f -Nl %3s enviei %d -Nw %s ",
          op,                                  // event type
          Scheduler::instance().clock(),  // time
          src_,                                // this node
          ch->next_hop_,                 // next hop
```

Roteamento Ad hoc

- Suporta diversos algoritmos:
  - DSDV
    - Contribuição da CMU
  - DSR
    - Contribuição da CMU (recentemente atualizado)
  - AODV
    - Recentemente atualizado pela univ. de cincinnati
  - TORA
    - Contribuição da CMU
- ***Exemplos: tcl/test/test-suite-wireless- { lan-newnode.tcl, lan-aodv.tcl, lan-tora.tcl } e cd tcl/ex/; grep "protocolo Ad hoc" *.tcl***

Modelo de Energia I

- Ponto fraco da ferramenta
- É um atributo do nodo no NS que representa o nível de energia no nodo móvel
- Possui um campo de energia inicial *(InitialEnergy_)*
- Campos de consumo de energia *(txPower_, rxPower)*
- Campo da energia corrente *(energy_)*

Modelo de Energia II

Attribute	optional values	default values
-energyModel	"EnergyModel"	none
-rxPower	receiving power in watts (e.g 0.3)	281.8mW
-txPower	transmitting power in watts (e.g 0.4)	281.8mW
-initialEnergy	energy in joules (e.g 0.1)	0.0

Modelos de Propagação I

- Determinam a potência do sinal recebido para cada pacote
- Para a camada física de cada nodo existe um limiar (*threshold)*
  - Caso a potência do sinal de pacote recebido esteja abaixo do limiar o mesmo é marcado como errado e descartado pela camada MAC

Modelos de Propagação II

- Até o presente existem 3 modelos de propagação no NS
  - *Friss-space* – atenuação para pequenas distâncias $1/r^2$
  - *Two-ray ground reflection* – para longas distâncias $1/r^4$
  - *Shadowing* -probabilístico

Modelos de Propagação III

- Modelos *Friss-space e Two-ray ground reflection*
  - Calculam a potência de maneira determinística
  - Representam o raio de comunicação como um círculo ideal
- *Shadowing*
  - Estende a idéia de círculo ideal através de métodos estatísticos (variação do sinal a uma certa distância)
  - Nodos podem comunicar probabilisticamente quando na fronteira do raio de comunicação

Modelo de Mobilidade

- *Random Waypoint*
  - Nodo permanecem parados durante um período de tempo randômico
  - Em seguida seguem para determinado ponto do *grid* em linha reta e velocidade constante
  - Chegando lá, permanecem estáticos novamente e o processo se repete

Exemplo Sem Fio

- Cenário
  - 3 nodos móveis
  - Grid: 670mX670m
  - Utilização do DSDV como protocolo de roteamento ad hoc
  - Modelo de Mobilidade: *Random Waypoint*
- *ns-2/tcl/ex/wireless-demo-csci694.tcl*

Exemplo – Passo 1

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

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

Exemplo – Passo 2

```
# 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
```

GOD (*General Operations Director*)

- Armazena o menor caminho de um nodo a outro
  - Ótimo para ser comparado com o desempenho do algoritmo de roteamento
  - Porém, calcular *on the fly* pode ser muito caro

  $god set-dist <from> <to> <#hops>
- Apenas uma instância do objeto por simulação
- Recebe o número de nodos móveis

  set god [create-god <no of mnodes>]

Exemplo – Passo 3

- Criação do God

  set god [create-god 3]

  $ns at 900.00 “$god setdist 2 3 1”

Exemplo – Passo 4

```
# Define how a mobile node is configured
$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 5

```
# Next create a mobile node, attach it to the
channel
set node(0) [$ns node]
# disable random motion
$node(0) random-motion 0


# Use "for" loop to create 3 nodes:
for {set i < 0} {$i < 3} {incr i} {
    set node($i) [$ns node]
    $node($i) random-motion 0
}
```

Mobilidade

- Posição do nodo é definida através de um modelo *3-D*
- Porém, o eixo Z não é utilizado

```
$node set X_ <x1>
$node set Y_ <y1>
$node set Z_ <z1>
$node at $time setdest <x2> <y2> <speed>
```

- **Movimento pode ser registrado *(logged)***

  $ns_ node-config -movementTrace ON or OFF

Gerador de Cenário: mobilidade

- Gerador

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

  - Fonte :

```
ns-2/indep-utils/cmu-scen-gen/setdest/
```

- Movimentação Randômica

```
$node random-motion
```

Arquivo de Movimentação: posições

- Nodo, coordenadas X, Y e Z

$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

Arquivo de Movimentação: deslocamentos

- Data, nodo, coordenada

$ns_ at 50.000000000000 "$node_(2) setdest 369.463244915743 170.519203111152 3.371785899154"

$ns_ at 51.000000000000 "$node_(1) setdest 221.826585497093 80.855495003839 14.909259208114"

$ns_ at 33.000000000000 "$node_(0) setdest 89.663708107313 283.494644426442 19.153832288917"

Gerador de Cenário: tráfego

- Gerando arquivos de padrões de tráfego
  - Tráfego CBR/TCP: gerador

```
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
```

Gerador de Cenário: Tráfego

- Criação dos agentes (tráfego, transporte e *Null)*
- Configura os atributos do tráfego (tamanho de pacote, razão, etc)
- Fonte: *ns-2/indep-utils/cmu-scen-gen/*
- **Exemplo pode ser encontrado em:**

`tcl/mobility/scene/cbr-50-{10-4-512, 20-4512}`

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"
.....
```

Exemplo – Passo 6

- Carregar os arquivos

```
# Define node movement model
source <movement-scenario-files>

# Define traffic model
source <traffic-scenario-files>
```

Exemplo – Passo 7

```
# Define node initial position in nam
for {set i 0} {$i < 3 } { incr i} {
     $ns initial_node_position $node($i) 20
}

# 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
```

Configuração de Energia

- Energia Inicial
- Gasto com envio
- Gasto com recepção

```
$ns_ node-config \
    -energyModel EnergyModel
    -initialEnergy 100.0
    -txPower             0.6
    -rxPower             0.2
```

nam: visualização I

- Utilize o *nam* para visualizar:
  - Posição do nodo
  - Movimento (direção e velocidade)

nam: visualização II

- Substitua

```
$ns namtrace-all $fd
```

por

```
$ns namtrace-all-wireless $fd
```

ao final da simulação, execute

```
$ns nam-end-wireless [$ns now]
```

Dicas I

- Apontadores
  - int src = Address::instance().get_nodeaddr(ih->saddr());
  - God::instance()->”objeto”

Dicas II

- Crie seu próprio *trace*

```
sprintf(pt_->buffer() + offset," -S sensor -Sh %f -Shm
  %f -Sj %f -Sjm %f -Sl %f  -Slm %f -Spkts %lu",
            hops,
            statistic_sensor->hops_avg,  jitter,
            statistic_sensor->jitter_avg,  latency,
            statistic_sensor->latency_avg,
            statistic_sensor->count_pkts);
      statistic_sensor->old_latency = latency;
```

Dicas III

- Programe, não interprete

```
// Leob && damacedo - Fev 2003
void God::setEnergy(double energy_loss, int i) {

    double energy;
    energy = mb_node[i]->energy_model()->energy();
    mb_node[i]->energy_model()->setenergy(energy -
    energy_loss);
}
```

Dicas IV

```
// Leob && damacedo - Fev 2003
void God::setEnergy(double energy_loss, int i) {

    double energy;
    energy = mb_node[i]->energy_model()->energy();
    mb_node[i]->energy_model()->setenergy(energy -
    energy_loss);
}
```

Dicas V

- Utilize sempre que puder o código C++
- O melhor manual se encontra em *.cc *.h
- A única fonte confiável é o *trace* – enquanto não tiver sido alterado

Dúvidas

leob@dcc.ufmg.br
http://www.dcc.ufmg.br/~leob/#ns

Exercícios

1) Verifique os exemplos de cenários que utilizam LANs (tcl/ex) e tente produzir sua própria rede local

2) Estudar e simular o exemplo sem fio do tutorial do NS (***IX. Running Wireless Simulations in n*s) - http://www.dcc.ufmg.br/~leob/#ns**

3) Compilar e rodar os geradores de cenários (para mobilidade, tráfego e propagação)

4) Adicionar no *script* feito durante a primeira lista de exercícios funcionalidades através dos geradores de cenários, isto é, aumentar o número de nodos, criar arquivo de padrões de mobilidade e tráfego e atribuir diferentes alcances de transmissão para os nodos da rede