Modelagem e Avaliação de Desempenho

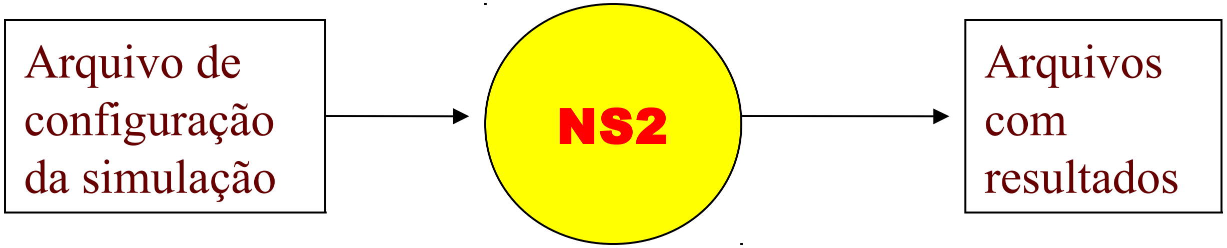
Pós Graduação em Engenharia Elétrica - PPGEE

Prof. Carlos Marcelo Pedroso

2016

Network Simulator Versão 2

- O NS2 é um simulador escrito em C++ com interpretador OTcl como frontend.



Arquivo de configuração da simulação

NS2

Arquivos com resultados

NS2 – Exemplo 1

```
#Create a simulator object
set ns [new Simulator]

#Open the nam trace file
set nf [open out.nam w]
$ns namtrace-all $nf

#Create two nodes
set n0 [$ns node]
set n1 [$ns node]

#Create a duplex link between the nodes
$ns duplex-link $n0 $n1 1Mb 10ms DropTail
```

NS2 – Exemplo1

```
#Create a UDP agent and attach it to node n0
set udp0 [new Agent/UDP]
$ns attach-agent $n0 $udp0

# Create a CBR traffic source and attach it to udp0
set cbr0 [new Application/Traffic/CBR]
$cbr0 set packetSize_ 500
$cbr0 set interval_ 0.005
$cbr0 attach-agent $udp0

#Create a Null agent (a traffic sink) and attach it to node
set null0 [new Agent/Null]
$ns attach-agent $n1 $null0
```

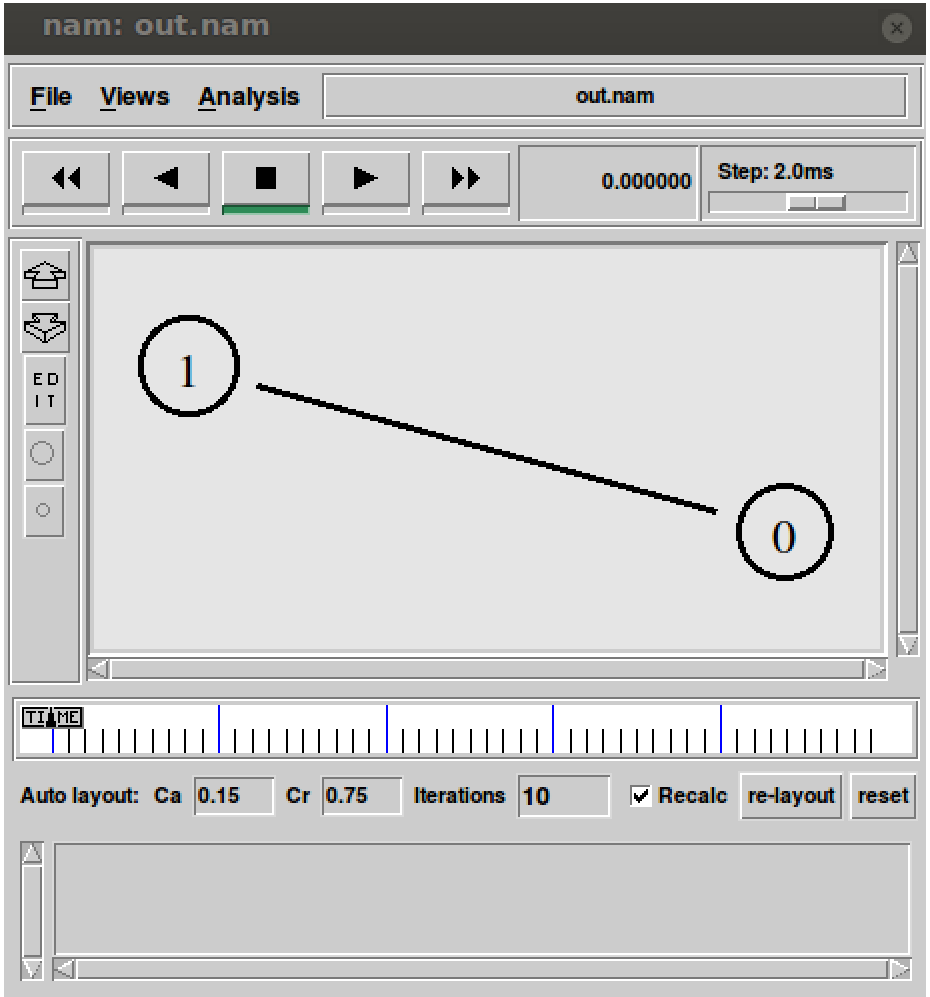
NS2 – Exemplo1

```
#Connect the traffic source with the traffic sink
$ns connect $udp0 $null0

#Schedule events for the CBR agent
$ns at 0.5 "$cbr0 start"
$ns at 4.5 "$cbr0 stop"
#Call the finish procedure after 5 seconds of simulation time
$ns at 5.0 "finish"

#Run the simulation
$ns run
```

NS2 – Exemplo1

```
#Define a 'finish' procedure
proc finish {} {
        global ns nf
        $ns flush-trace
       #Close the trace file
        close $nf
       #Execute nam on the trace file
        exec nam out.nam &
        exit 0
}
```

NAM - Network Animator



0.000000

Step: 2.0ms

Auto layout: Ca 0.15 Cr 0.75 Iterations 10 ☑ Recalc

NS2 – Exemplo2

```
#Create a simulator object
set ns [new Simulator]
#Define different colors for data flows
$ns color 1 Blue
$ns color 2 Red
#Open the nam trace file
set nf [open out.nam w]
$ns namtrace-all $nf
```

NS2 – Exemplo2

```
#Create four nodes
set n0 [$ns node]
set n1 [$ns node]
set n2 [$ns node]
set n3 [$ns node]

#Create links between the nodes
$ns duplex-link $n0 $n2 1Mb 10ms DropTail
$ns duplex-link $n1 $n2 1Mb 10ms DropTail
$ns duplex-link $n3 $n2 1Mb 10ms SFQ
```

NS2 – Exemplo2

```
#Monitor the queue for the link between node 2
and node 3
$ns duplex-link-op $n2 $n3 queuePos 0.5

#Create a UDP agent and attach it to node n0
set udp0 [new Agent/UDP]
$udp0 set class_ 1
$ns attach-agent $n0 $udp0

# Create a CBR traffic source and attach it to
udp0
set cbr0 [new Application/Traffic/CBR]
$cbr0 set packetSize_ 500
$cbr0 set interval_ 0.005
$cbr0 attach-agent $udp0
```

800Kb/s

NS2 – Exemplo2

```
#Create a UDP agent and attach it to node n1
set udp1 [new Agent/UDP]
$ns attach-agent $n1 $udp1

# Create a CBR traffic source and attach it to udp1
set cbr1 [new Application/Traffic/CBR]
$cbr1 set packetSize_ 500
$cbr1 set interval_ 0.005
$cbr1 attach-agent $udp1
```

800Kb/s

NS2 – Exemplo2

```
#Create a Null agent (a traffic sink) and attach it to node n3
set null0 [new Agent/Null]
$ns attach-agent $n3 $null0

#Connect the traffic sources with the traffic sink
$ns connect $udp0 $null0
$ns connect $udp1 $null0
```

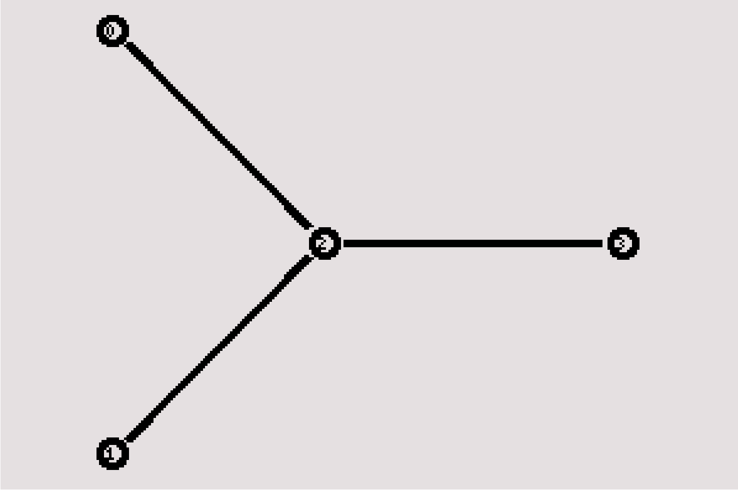
NS2 – Exemplo2

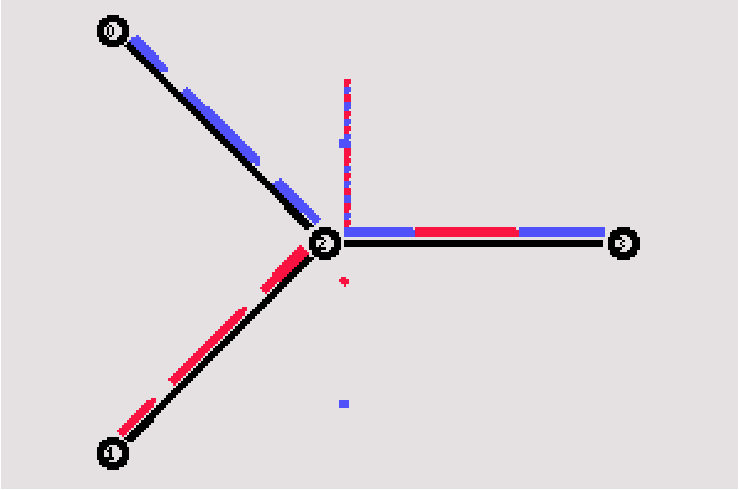
```
#Schedule events for the CBR agents
$ns at 0.5 "$cbr0 start"
$ns at 1.0 "$cbr1 start"
$ns at 4.0 "$cbr1 stop"
$ns at 4.5 "$cbr0 stop"
#Call the finish procedure after 5 seconds of
simulation time
$ns at 5.0 "finish"

#Run the simulation
$ns run
```

NS2 – Exemplo2

```
#Schedule events for the CBR agents
$ns at 0.5 "$cbr0 start"
$ns at 1.0 "$cbr1 start"
$ns at 4.0 "$cbr1 stop"
$ns at 4.5 "$cbr0 stop"

#Call the finish procedure after 5 seconds of simulation time
$ns at 5.0 "finish"

#Run the simulation
$ns run
```

NS2 – Exemplo2





NS2 – Exemplo4

```
#Open the NAM trace file
set nam_file [open out.nam w]
$ns namtrace-all $nam_file
set tf [open out.tr w]
$ns trace-all $tf
```

```
#Simulation time
set SimTime 3.0
#Bottleneck link
Bandwidth
set bw 10Mb
#Bottleneck link delay
set delay 20ms
#Bottleneck link
queuetype
set queuetype DropTail
```

```
#Buffer Size
set BufferSize 50
#TCP packet size
set packetsize 1000
#TCP window size
set windowsize 80
#Initialize a variable
set old_data 0
```

NS2 – Exemplo4

```
#Set Queue size of the bottleneck link (n2-n3) to 20
$ns queue-limit $n2 $n3 $BufferSize
```

```
#Create four nodes
set n0 [$ns node]
set n1 [$ns node]
set n2 [$ns node]
set n3 [$ns node]
#Connect the nodes - Create links between the nodes
$ns duplex-link $n0 $n2 100Mb 2ms DropTail
$ns duplex-link $n1 $n2 100Mb 2ms DropTail
$ns duplex-link $n2 $n3 $bw $delay $queuetype
```

NS2 – Exemplo4

```
#Setup a TCP connection
set agent_tcp [new Agent/TCP]

#Attach TCP Agent to source node n0
$ns attach-agent $n0 $agent_tcp
set agent_sink [new Agent/TCPSink]

#Attach a TCPSink Agent to destination node n3
$ns attach-agent $n3 $agent_sink

#Connect TCP Agent with TCPSink Agent
$ns connect $agent_tcp $agent_sink

#Flow Identity for TCP
$agent_tcp set fid_ 1
```

NS2 – Exemplo4

```
#TCP parameters
$agent_tcp set packet_size_ $packetsize
$agent_tcp set window_ $windowsize
#Setup a FTP traffic over TCP connection
set traf_ftp [new Application/FTP]
$traf_ftp attach-agent $agent_tcp
```

NS2 – Exemplo4

```
#Setup a UDP connection
set agent_udp [new Agent/UDP]
#Attach UDP Agent to source node n1
$ns attach-agent $n1 $agent_udp
set agent_null [new Agent/Null]
#Attach a Null Agent to destination node n3
$ns attach-agent $n3 $agent_null
#Connect UDP Agent with NULL Agent
$ns connect $agent_udp $agent_null
#Flow Identity for UDP
$agent_udp set fid_ 2
#Setup a CBR traffic over UDP connection
set traf_cbr [new Application/Traffic/CBR]
$traf_cbr attach-agent $agent_udp
```

NS2 – Exemplo4

```
#CBR parameters
$traf_cbr set packet_size_ 1000
$traf_cbr set rate_ 4Mb
$ns at 0.0 "$ns trace-queue $n2 $n3 $trace_file"
```

□ Verifique o algoritmo *slow start* do TCP utilizando o NAM

NS2 -Transmitindo sobre o UDP

- 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

Geradores de tráfego sobre o UDP

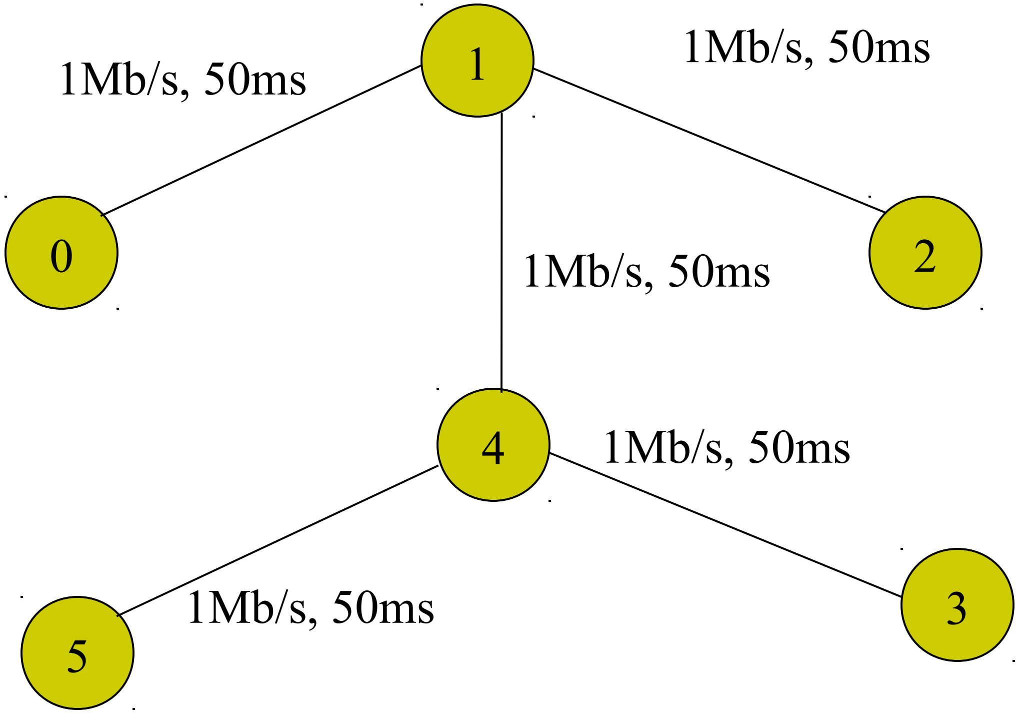
- CBR
  - set src [new Application/Traffic/CBR]
  - $src attach-agent $udp
  - $ns at 3.0 “$src start”
- Exponential
  - set src [new Application/Traffic/Exponential]
- Pareto on/off
  - set src [new Application/Traffic/Pareto]

Criando uma conexão TCP

- 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

Exercícios

- Escreva uma simulação para a topologia abaixo:



Exercício

- Adicione aplicativos CBR transmitindo de 0 para 2, de 3 para 2 e de 5 para 2 sobre o protocolo UDP;
- Aumente progressivamente a taxa de geração de tráfego e determine o ponto de saturação da rede. Compare com o máximo teórico;
- Repita a operação utilizando como gerador de tráfego uma aplicação do tipo FTP e verifique como o algoritmo de gerência de janela ativa reduziu a taxa de transmissão. A divisão de banda é justa?
- Troque o algoritmo de descarte para SFQ e verifique se a justiça melhorou
- Adicione um gerador de tráfego UDP anote o efeito sobre os aplicativos TCP

Network Simulator Versão 3

- O NS3 é um simulador totalmente escrito em C++.
- A topologia de simulação também é configurada em um arquivo fonte C++, e compilada para produzir um executável.
- O NS3 possui capacidades de simulação da camada 1,5 até a camada 4 (com algumas poucas aplicações).

NS3 - Modelos Suportados

Camada de Enlace:

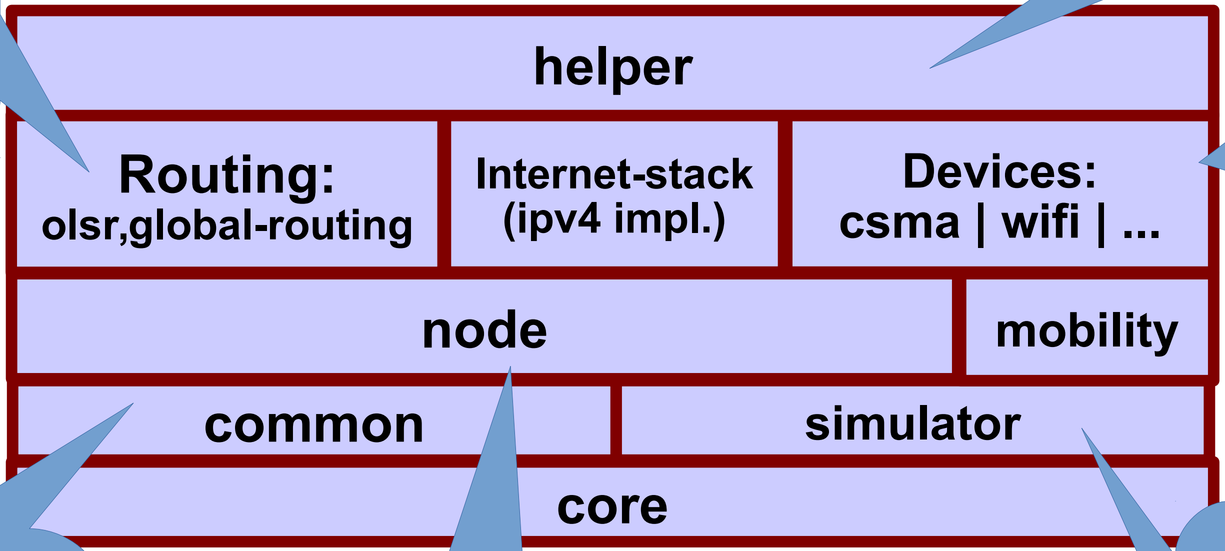
- Point-to-point (PPP links)
- Csma (Ethernet links)
- Bridge: 802.1D Learning Bridge
- Wifi (802.11 links)
    - EDCA QoS support (but not HCCA)
    - Both infrastructure (with beacons), and adhoc modes
- Mesh
    - 802.11s
    - ”Flame”: Forwarding LAyer for MEshing protocol
        - ”Easy Wireless: broadband ad-hoc networking for emergency services”
- LTE, Wimax

NS3 - Modelos Suportados

Aplicações:

- **Onoff**
  - Generates streams, alternating on-and-off periods
  - Highly parameterized
    - Can be configured to generate many types of traffic
      - E.g. OnTime=1 and OffTime=0 means <u>CBR</u>
    - Works with either UDP or TCP
- **Packet sink**: receives packets or TCP connnections
- **Ping6, v4ping**: send ICMP ECHO request
- **Udp-client/server**: sends UDP packet w/ sequence number
- **Udp-echo**: sends UDP packet, no sequence number
- **Radvd**: router advertisement (for IPv6)
- **Socket**: pode ser implementada uma nova aplicação.

NS3: Módulos



helper

Routing:
olsr,global-routing

Internet-stack
(ipv4 impl.)

Devices:
csma | wifi | ...

node

mobility

common

simulator

core

High-level wrappers for
everything else.

No smart pointers used.

Aimed at scripting.

Node class
NetDevice ABC
Address types
(IPv4, MAC, etc.)
Queues
Socket ABC
IPv4 ABCs
Packet Sockets

Mobility Models
(static,
random walk,
etc.)

Smart pointers
Dynamic type system
Attributes
Callbacks, Tracing
Logging
Random Variables

Packets
Packet Tags
Packet Headers
Pcap/ascii file writing

Events
Scheduler
Time arithmetic

NS3

- O código fonte do simulador pode ser encontrado em http://www.nsnam.org/
- A documentação está disponível em: http://www.nsnam.org/documentation/
- Download - Configure - Compile - Run
- Alguns conceitos de orientação a objeto
- Estudo dos exemplos:
  - First.cc
  - Second.cc
  - Third.cc