

Modelagem e Avaliação de Desempenho

Pós Graduação em Engenharia Elétrica - PPGEE

Prof. Carlos Marcelo Pedroso

2011

Simulação de Sistemas

- Simulação é a técnica de solução de um problema pela análise de um modelo que descreve o comportamento do sistema utilizando um computador digital
- Metodologia:
 - Construção de um modelo da situação atual e reproduzir computacionalmente
 - Inclusão de alterações para o estudo de otimizações desejadas

Simulação de Sistemas

- O método de Monte Carlo
 - Deveu-se a revisão de uma técnica matemática utilizada por cientistas do projeto Manhattan, em Los Alamos, década de 1940, publicada em 1949
 - Na aplicação desta técnica, os dados são gerados empregando-se um gerador de número aleatórios e uma distribuição de probabilidade que descreve a variável aleatória de interesse

Simulação de Sistemas

- O método de Monte Carlo
 - 1 Definir o domínio de entradas possíveis
 - 2 Gerar as entradas de acordo com uma distribuição de probabilidade que descreve a entrada
 - 3 Realizar o processamento determinístico das entrada
 - 4 Agregar os resultados e retornar ao passo 2.

Geração de Variáveis Aleatórias

- Método da inversa
 - Toma-se a distribuição acumulada da variável aleatória, da por $P(X \leq x) = F(x)$
 - Atribui-se um valor randômico entre 0 e 1 para $F(x)$.
 - Calcula-se o valor de x
 - Desta forma, para cada valor randômico entre 0 e 1 R_i será obtido um valor de x_i .

- Exemplo: Distribuição Exponencial.

Geração de Variáveis Aleatórias

□ Exercícios:

- Calcule a expressão para obter uma variável aleatória que segue a distribuição uniforme
- Calcule a expressão para obter uma variável aleatória que segue a distribuição triangular
- Determine como utilizar o método para uma distribuição empírica

Geração de Números Randômicos

- Um dos problemas a serem resolvidos é como gerar números randômicos, uniformemente distribuídos entre 0 e 1.
- Gerador Congruente Linear (“LCG”)
 - Definido pela equação linear $x_{n+1} = (ax_n + b) \pmod{m}$
 - Produz uma sequência entre $\{0, 1, \dots, m-1\}$
 - Pode-se chamar $LCG(m, a, b, x_0)$
 - x_0 é a semente (valor inicial)

Geração de Números Randômicos

□ Método Tausworthe

- $$x_n = \theta_1 x_{n-1} \oplus \theta_2 x_{n-2} \oplus \dots \oplus \theta_q x_{n-q}$$

O método é chamado gerador auto regressivo de ordem q (AR(q)). Este método é utilizado em sistemas criptográficos.

Geração de Variáveis aleatórias

- Algumas distribuições podem não possuir expressão analítica para distribuição acumulada (é o caso da distribuição normal).
- Neste caso, é necessário aplicar outros métodos.
- Um dos métodos é o método “acceptance-rejection”
- Para gerar uma VA X com distribuição $F(x)$:
 - = Toma-se uma distribuição $G(y)$ com método

Geração de Variáveis aleatórias

Acceptance-Rejection Algorithm for continuous random variables

1. Generate a rv Y distributed as G .
2. Generate U (independent from Y).
3. If

$$U \leq \frac{f(Y)}{cg(Y)},$$

then set $X = Y$ (“accept”) ; otherwise go back to 1 (“reject”).

Acceptance-Rejection Method

Example 1: Generating a random variable from

$$f_X(x) = 3x^2, \quad 0 \leq x \leq 1.$$

Assume

$$g_X(x) = 1, \quad 0 \leq x \leq 1.$$

Thus

$$\max\left(\frac{f_X(x)}{g_X(x)}\right) = 3 = c.$$

$$\frac{f_X(x)}{cg_X(x)} = x^2.$$

Algorithm:

- 1) Generate two uniform random variables U_1 and U_2 from $U(0, 1)$.
- 2) If $U_2 \leq U_1^2$ accept U_1 as the random variable from $f_X(x)$, otherwise go to step 1).

Distribuição Normal

□ Aproximação:

$$x_i = F^{-1}(R_i) = [R_i^{0,135} - (1-R_i)^{0,135}] / 0,1975$$

– Média 0, desvio padrão 1 [N(0, 1)]

É possível transformar para qualquer outra média μ e desvio padrão σ , fazendo:

- $y_i = \mu + \sigma x_i$

Distribuição Normal

□ *Método acceptance-rejection:*

- 1. Gere duas variáveis randômicas com distr. Uniforme $U(0,1)$, R_1 e R_2*
- 2. Seja $x = -\ln R_1$*
- 3. Se $R_2 > e^{-(1/2)(x-1)^2}$, volte ao passo 1*
- 4. Gere R_3*
- 5. Se $R_3 > 0.5$, retorne $\mu + \sigma x$, caso contrário retorne $\mu - \sigma x$*

Exercício

- 1- Utilize o Método de Monte Carlo para realizar a simulação de uma fila com um servidor, onde o intervalo entre chegadas segue a distribuição exponencial e o tempo de atendimento também segue a distribuição exponencial. Compare o tempo médio na fila com os resultados obtidos com a teoria de filas, modelo M/M/1.
- 2- Utilize o Método de Monte Carlo para realizar uma simulação de forma a determinar o valor do número π através de uma simulação.

Análise de resultados

- A análise de resultados de uma simulação deve ser feita de maneira muito cuidadosa
 - Especialmente, não cometa o erro de generalizar resultados específicos
 - Para fazer qualquer tipo de inferência sobre os resultados, é necessário realizar uma análise estatística

Confiança estatística

- Um intervalo de confiança compreende um intervalo numérico que possui uma probabilidade igual a $(1-\alpha)$ de incluir o verdadeiro valor da medida de desempenho sob análise, com um nível de confiança.
 - $(1-\alpha)$ representa o intervalo de confiança.
 - α representa o erro admitido ao se concluir sobre a presença do verdadeiro valor da variável no intervalo calculado.

Confiança estatística

- Suponha que foi simulado o tempo médio na fila em um sistema.
 - Assumindo que a variável aleatória X representa o tempo médio na fila.
 - A simulação foi realizada 5 vezes, tomando-se o cuidado de iniciar a simulação com valores de sementes diferentes

Confiança estatística

□ Os resultados obtidos foram:

□ O semi-intervalo h é calculado por

$$h = t_{n-1, 1-\alpha/2} \frac{\sigma}{\sqrt{n}}$$

- n é o número de rodadas
- σ é o desvio padrão
- t indica os valores críticos para distr. t student

| Rodada | X |
|--------|------|
| 1 | 63,2 |
| 2 | 69,7 |
| 3 | 67,3 |
| 4 | 64,8 |
| 5 | 72 |

Valores críticos – t student

Valores de t para v graus de liberdade

| v | 0,995 | 0,99 | 0,975 | 0,95 | 0,90 |
|-------|-------|-------|-------|------|------|
| 1 | 63,66 | 31,82 | 12,71 | 6,31 | 3,08 |
| 2 | 9,92 | 6,96 | 4,30 | 2,92 | 1,89 |
| 3 | 5,84 | 4,54 | 3,18 | 2,35 | 1,64 |
| 4 | 4,60 | 3,75 | 2,78 | 2,13 | 1,53 |
| 5 | 4,03 | 3,36 | 2,57 | 2,02 | 1,48 |
| 6 | 3,71 | 3,14 | 2,45 | 1,94 | 1,44 |
| 7 | 3,50 | 3,00 | 2,36 | 1,90 | 1,42 |
| 8 | 3,36 | 2,90 | 2,31 | 1,86 | 1,40 |
| 9 | 3,25 | 2,82 | 2,26 | 1,83 | 1,38 |
| 10 | 3,17 | 2,76 | 2,23 | 1,81 | 1,37 |
| 11 | 3,11 | 2,72 | 2,20 | 1,80 | 1,36 |
| 12 | 3,06 | 2,68 | 2,18 | 1,78 | 1,36 |
| 13 | 3,01 | 2,65 | 2,16 | 1,77 | 1,35 |
| 14 | 2,98 | 2,62 | 2,14 | 1,76 | 1,34 |
| 15 | 2,95 | 2,60 | 2,13 | 1,75 | 1,34 |
| 16 | 2,92 | 2,58 | 2,12 | 1,75 | 1,34 |
| 17 | 2,90 | 2,57 | 2,11 | 1,74 | 1,33 |
| 18 | 2,88 | 2,55 | 2,10 | 1,73 | 1,33 |
| 19 | 2,86 | 2,54 | 2,09 | 1,73 | 1,33 |
| 20 | 2,84 | 2,53 | 2,09 | 1,72 | 1,32 |
| 21 | 2,83 | 2,52 | 2,08 | 1,72 | 1,32 |
| 22 | 2,82 | 2,51 | 2,07 | 1,72 | 1,32 |
| 23 | 2,81 | 2,50 | 2,07 | 1,71 | 1,32 |
| 24 | 2,80 | 2,49 | 2,06 | 1,71 | 1,32 |
| 25 | 2,79 | 2,48 | 2,06 | 1,71 | 1,32 |
| 26 | 2,78 | 2,48 | 2,06 | 1,71 | 1,32 |
| 27 | 2,77 | 2,47 | 2,05 | 1,70 | 1,31 |
| 28 | 2,76 | 2,47 | 2,05 | 1,70 | 1,31 |
| 29 | 2,76 | 2,46 | 2,04 | 1,70 | 1,31 |
| 30 | 2,75 | 2,46 | 2,04 | 1,70 | 1,31 |
| 40 | 2,70 | 2,42 | 2,02 | 1,68 | 1,30 |
| 60 | 2,66 | 2,39 | 2,00 | 1,67 | 1,30 |
| 120 | 2,62 | 2,36 | 1,98 | 1,66 | 1,29 |
| > 120 | 2,58 | 2,33 | 1,96 | 1,65 | 1,28 |

Confiança estatística

- No caso anterior, a média calculada é 67,74 e o desvio padrão σ é igual a 3,57;
- Para 99% de confiança, $\alpha=0,05$ e $t_{4, 0.995}=4,6$
- O valor de h calculado é de 7,34
- Os limites para 99% de confiança serão [60,06 ; 74,74]

Exercícios

- Utilize a simulação de fila realizada anteriormente, para chegadas exponenciais e atendimentos exponenciais.
 - Calcule o semi intervalo h para um nível de confiança de 99%
 - O que fazer para melhorar a resposta? (melhorar a resposta implica em reduzir ao mínimo o valor de h).

Exercícios

- Suponha novamente o sistema com uma fila. No entanto, desta vez, suponha que a chegada é modelada por uma distribuição normal $N(5, 10)$ e o atendimento é modelado também por uma distribuição normal $N(4, 20)$.
 - Determine o tempo médio de fila e tempo médio no sistema.
 - Realize a simulação de forma a obter uma boa resposta para para o nível de confiança de 99%.
 - Interprete os resultados.

Exercícios

- Suponha novamente o sistema com uma fila. No entanto, desta vez, suponha que a chegada é modelada por uma distribuição exponencial com média 4 e o atendimento é modelado também por uma distribuição de Pareto com parâmetros $\alpha=2,5$ e $\beta=2$. A distribuição de Pareto é uma distribuição de cauda pesada.
 - Determine o tempo médio de fila e tempo médio no sistema.
 - Realize a simulação de forma a obter uma boa resposta para para o nível de confiança de 99%.
 - Interprete os resultados.

| Distribuição de Pareto | |
|-------------------------------|---|
| Parâmetros | α, β $\alpha > 0$, parâmetro de forma $\beta > 0$, parâmetro de escala |
| Limites | $b \leq x < +\infty$ |
| Densidade de Probabilidade | $f(x) = \frac{\alpha\beta^\alpha}{x^{\alpha+1}}$ |
| Distribuição Acumulada | $F(x) = 1 - \left(\frac{\beta}{x}\right)^\alpha$ |
| Esperança ($E[X]$) | $\frac{\alpha\beta}{\alpha-1}, \alpha > 1$ |
| Variança ($Var[X]$) | $\frac{\alpha\beta^2}{(\alpha-1)^2(\alpha-2)}, \alpha > 2$ |

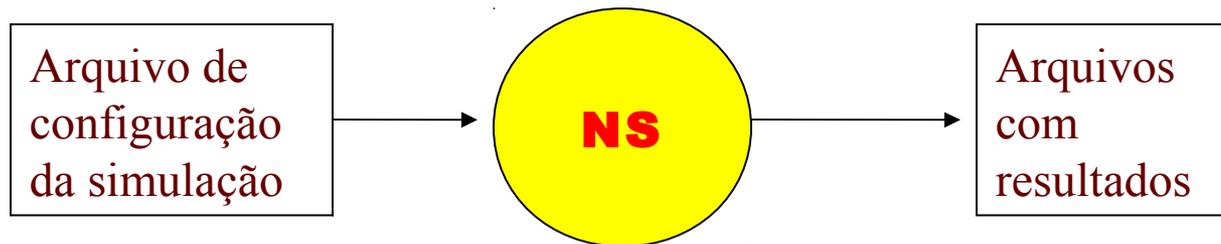


Simuladores

- O desempenho de redes de comunicação pode ser estudado através de simulações;
- O simulador pode apresentar resultados muito próximos do que se obteria na realidade;
- Os resultados devem ser tratados com rigor estatístico.

Network Simulator

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



NS – 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
```

NS – Exemplo 1

```
#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
```

NS – Exemplo 1

```
#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
```

NS – Exemplo 1

```
#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
}
```

NS – Exemplo 1

```
E:\ns>ns exemplo1.tcl
```

```
E:\ns>dir
```

```
Volume in drive E is DADOS
```

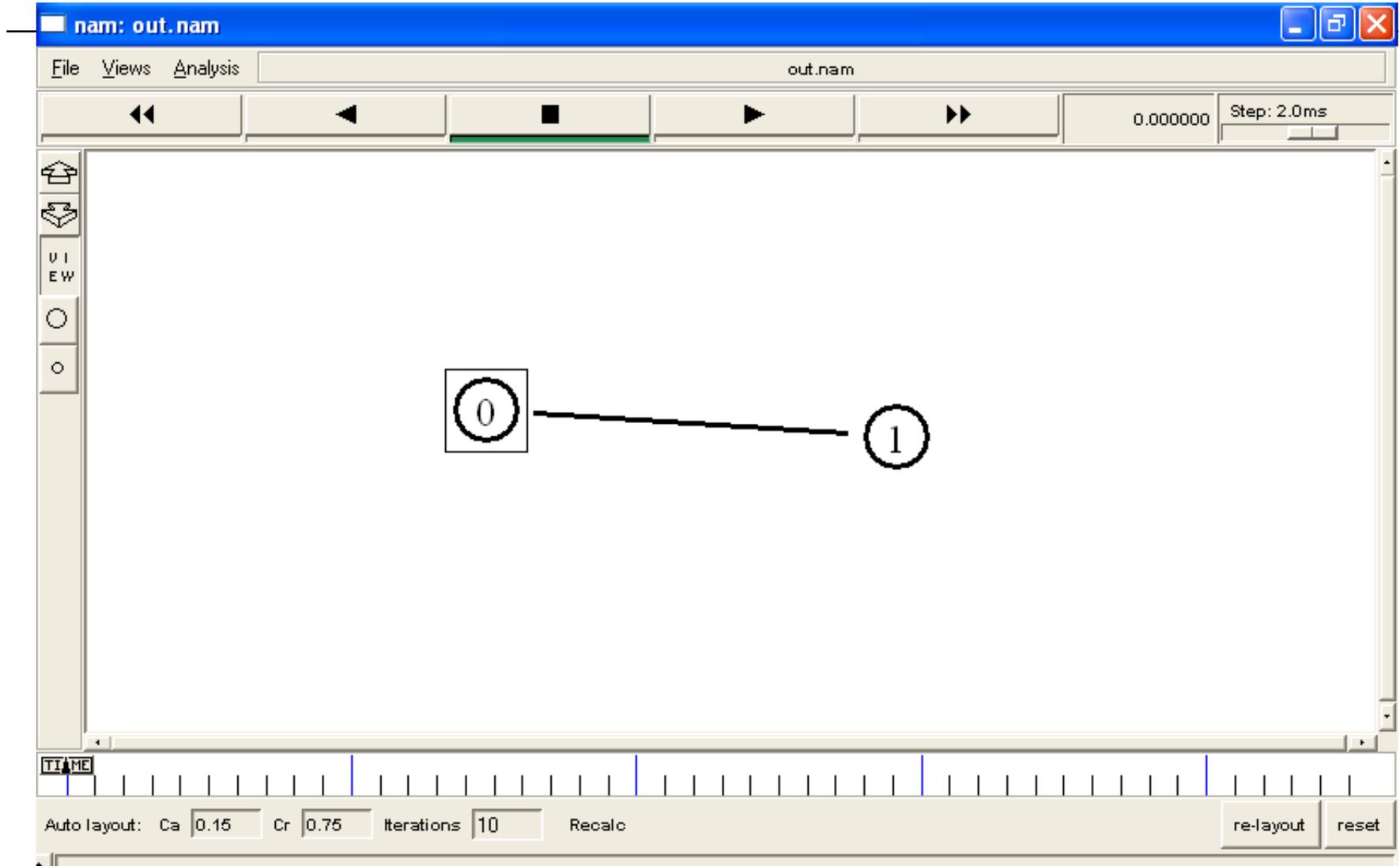
```
Volume Serial Number is 0862-F827
```

```
Directory of E:\ns
```

```
10/06/2003  07:11 PM    <DIR>          .
10/06/2003  07:11 PM    <DIR>          ..
10/06/2003  07:36 PM              1,378 exemplo1.tcl
09/11/2003  01:51 PM          1,531,904 NS.exe
10/06/2003  07:29 PM          1,871,899 NAM.exe
10/06/2003  09:20 PM           292,916 out.nam
10/06/2003  07:36 PM              1,128 exemplo2.tcl
12 File(s)          15,000,772 bytes
                4 Dir(s)    1,362,657,280 bytes free
```

```
E:\ns>nam out.nam
```

NAM - Network Animator



NS – 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
```

NS – Exemplo2

```
#Create four nodes
```

```
set n0 [$ns node]
```

```
set n1 [$ns node]
```

```
set n2 [$ns node]
```

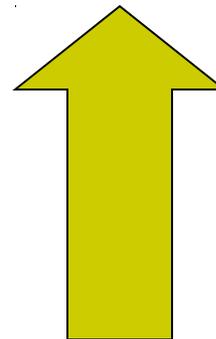
```
set n3 [$ns node]
```

```
#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
```



NS – 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
```

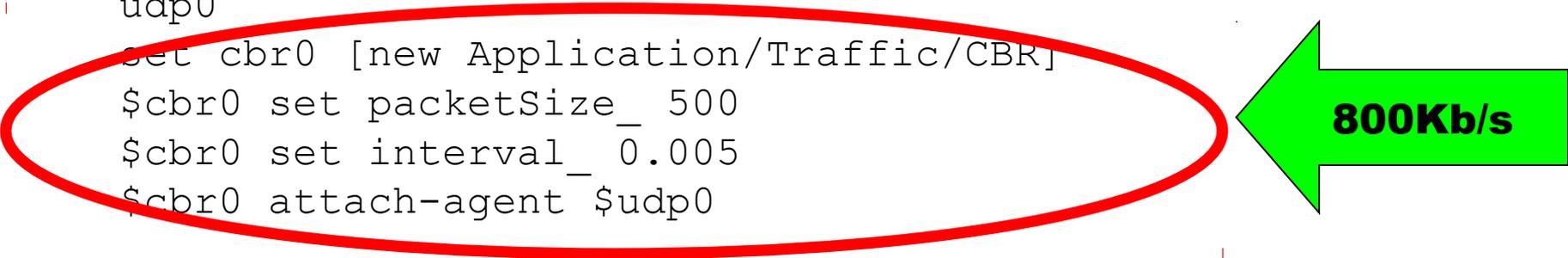
```
# Create a CBR traffic source and attach it to  
udp0
```

```
set cbr0 [new Application/Traffic/CBR]
```

```
$cbr0 set packetSize_ 500
```

```
$cbr0 set interval_ 0.005
```

```
$cbr0 attach-agent $udp0
```

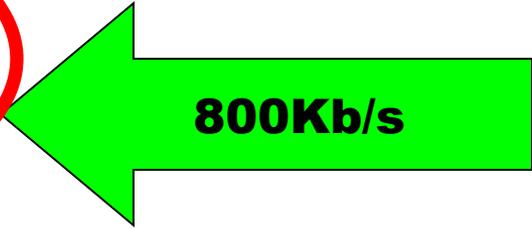


800Kb/s

NS – 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

NS – 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
```

NS – 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
```

NS – 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"
```

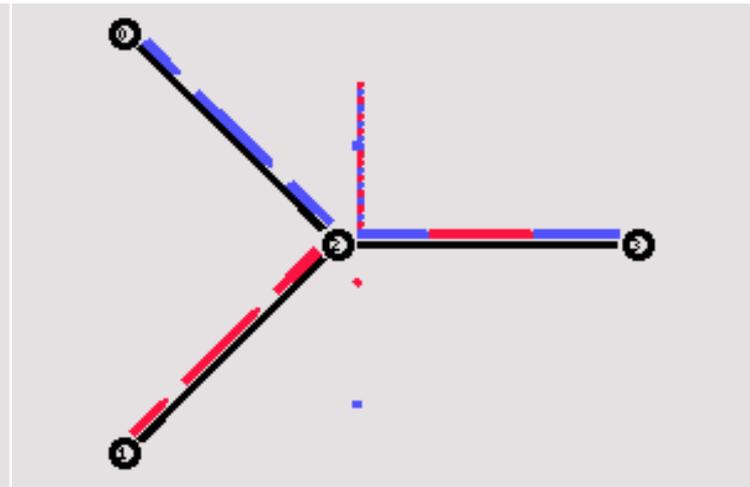
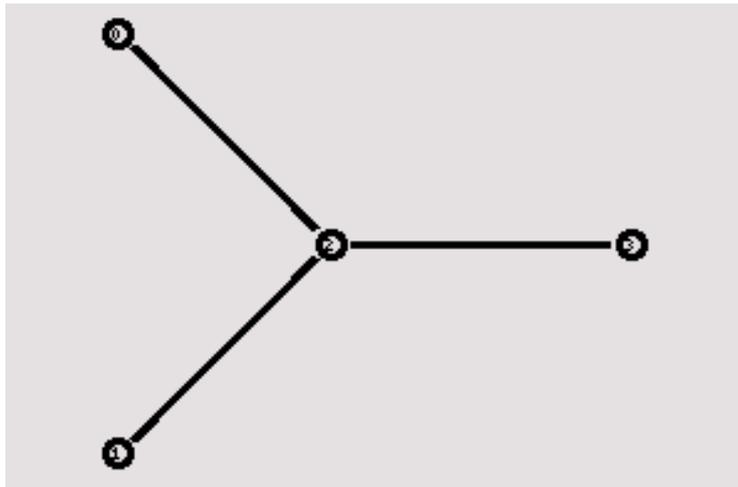
```
#Call the finish procedure after 5 seconds of simulation time
```

```
$ns at 5.0 "finish"
```

```
#Run the simulation
```

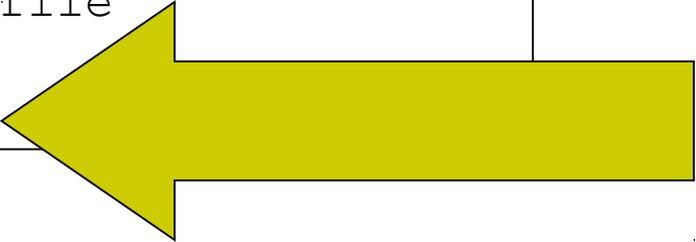
```
$ns run
```

NS – Exemplo2



NS – 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 window size 80
#Initialize a variable
set old_data 0
```

NS – 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
```

NS – 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
```

NS – Exemplo4

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

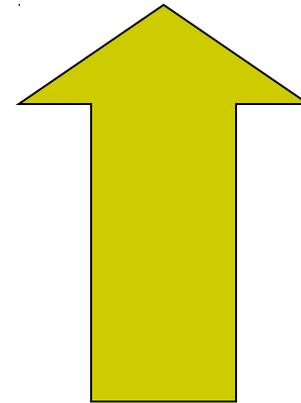
NS – Exemplo4

```
#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
```

NS – 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



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`

Aplicações sobre o TCP

□ FTP

- set ftp [new Application/FTP]
- \$ftp attach-agent \$tcp
- \$ns at 3.0 "\$ftp start"

□ Telnet

- set telnet [new Application/Telnet]
- \$telnet attach-agent \$tcp
- \$ns at 3.0 "\$telnet start"

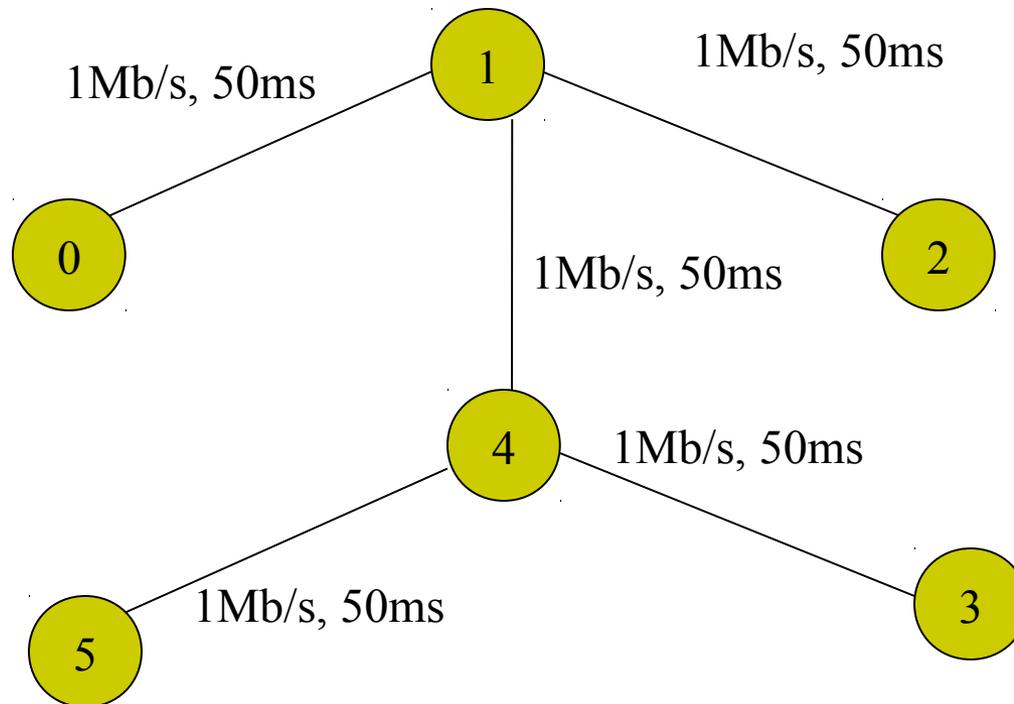
Aplicações sobre o TCP

□ Web

- set session [new httpSession \$ns <numPages>
<clientNode>]
- Exemplo

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