

## Algoritmos para Projeto de Redes de Telecomunicações: Software de Simulação NS-2

Programa de Pós-Graduação em Informática – PPGIa - PUCPR

Prof. Marcelo E. Pellenz  
<http://www.ppgia.pucpr.br/~marcelo>  
e-mail: [marcelo@ppgia.pucpr.br](mailto:marcelo@ppgia.pucpr.br)

## Agente de Transporte 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
    - \$udp set packetSize\_ <pktsize> (default MSS=1000 bytes)
    - \$udp set dst\_addr\_ <address>
    - \$udp set dst\_port\_ <portnum>
    - \$udp set ttl\_ <time-to-live>

## Geradores de Tráfego sobre o UDP

### ● CBR

- set src [new Application/Traffic/CBR]
- \$src attach-agent \$udp
- \$ns at 3.0 "\$src start"
  - rate\_ the sending rate
  - interval\_ (Optional) interval between packets
  - packetSize\_ the constant size of the packets generated
  - random\_flag indicating whether or not to introduce random "noise" in the scheduled departure times (default is off)
  - maxpkts\_ the maximum number of packets to send

3

## Geradores de Tráfego sobre o UDP

### ● Exponential on/off

- set src [new Application/Traffic/Exponential]
  - packetSize\_ the constant size of the packets generated
  - burst\_time\_ the average "on" time for the generator
  - idle\_time\_ the average "off" time for the generator
  - rate\_ the sending rate during "on" times

### ● Pareto on/off

- set src [new Application/Traffic/Pareto]
  - packetSize\_ the constant size of the packets generated
  - burst\_time\_ the average "on" time for the generator
  - idle\_time\_ the average "off" time for the generator
  - rate\_ the sending rate during "on" times
  - shape\_ the "shape" parameter used by the pareto distribution

4

## Agente de Transporte 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

5

## Agente de Transporte TCP

- |                                    |                                      |
|------------------------------------|--------------------------------------|
| ● Agent/TCP set window_ 20 ;       | # max bound on window size           |
| ● Agent/TCP set windowInit_ 1 ;    | # initial/reset value of cwnd        |
| ● Agent/TCP set packetSize_ 1000 ; | # packet size used by sender (bytes) |
| ● Agent/TCP set dupacks_ 0 ;       | # duplicate ACK counter              |
| ● Agent/TCP set ack_ 0 ;           | # highest ACK received               |
| ● Agent/TCP set cwnd_ 0 ;          | # congestion window (packets)        |
| ● Agent/TCP set ssthresh_ 0 ;      | # slow-stat threshold (packets)      |
| ● Agent/TCP set rtt_ 0 ;           | # rtt sample                         |
| ● Agent/TCP set rttvar_ 0 ;        | # mean deviation of rtt samples      |

6

## 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"
- Web

7

## Controle de Congestionamento do TCP

- Objetivos do protocolo TCP
  - Adaptar a taxa de transmissão de pacotes com a banda disponível
  - Evitar congestionamento na rede
  - Criar uma conexão confiável usando retransmissão dos pacotes perdidos

8

## Controle de Congestionamento do TCP

- **Objetivos das Confirmações (ACKs)**
  - Regular a taxa de transmissão do TCP, garantindo que pacotes podem ser transmitidos apenas quando outro deixou a rede.
  - Manter a conexão confiável, transmitindo para a fonte a informação necessária para retransmissão dos pacotes que ainda não chegaram ao destino.

9

## Controle de Congestionamento do TCP

- **Como sabemos que um pacote foi perdido ?**
- **Qual a informação transportada pelo ACK ?**
  - O ACK informa para a fonte qual o número de sequência do pacote esperado
  - ACKs repetidos indicam perda
  - Método robusto para ACKs perdidos (ACK implícito)
- **Como o TCP detecta um pacote perdido ?**
  - Três ACKs repetidos do mesmo pacote chegam a fonte (fast retransmit)
  - Se o ACK não chegar num período de tempo pré-estabelecido (timeout)
  - Quando um pacote TCP é transmitido, um *timer* é inicializado para contagem do *timeout*

10

## Controle de Congestionamento do TCP

- Principais Parâmetros de Controle de Protocolo TCP:
  - $T_o$  = tempo de *timeout*
  - $T_o = RTT + 4 * D$
  - $RTT$  = Estimativa corrente do *round trip time* médio
  - $D$  = Estimação da variabilidade do  $RTT$

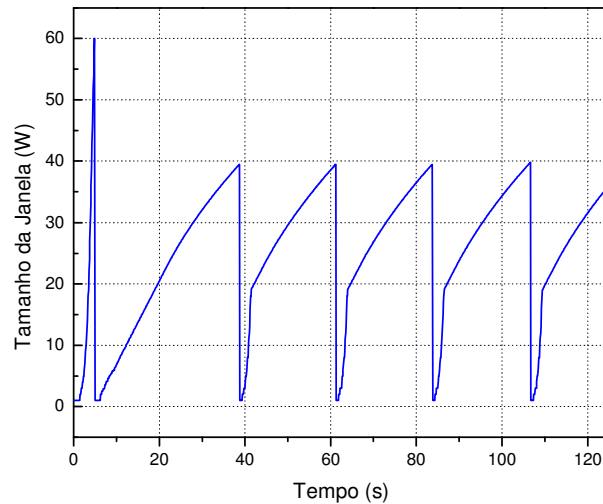
11

## Controle de Congestionamento do TCP

- Executar o **exemplo5.tcl**
- Analisar o comportamento do fluxo TCP usando o **nam**.
- Identificar o que ocorre com o fluxo de dados do TCP quando ocorrer o congestionamento da rede e descarte de pacotes.
- Traçar o gráfico **tamanho da janela** versus **tempo** do arquivo **WinFile**.
- Analisar o throughput (vazão) do fluxo TCP (0-4) utilizando o software TraceGraph.
- Analisar o throughput (vazão) do fluxo UDP (1-5).
- Comparar a vazão média do fluxo TCP (0-4) com a capacidade do enlace (2-3).

12

## Controle de Congestionamento do TCP



13

## Controle de Congestionamento do TCP

### • TCP Tahoe

- A janela de congestionamento é aumentada de um pacote para cada novo ACK recebido durante o período **slow-start** (quando  $cwnd\_ < ssthresh\_$ )
- A janela aumenta de  $1/cwnd\_$  para cada novo ACK durante o período **congestion avoidance** (quando  $cwnd\_ \geq ssthresh\_$ )

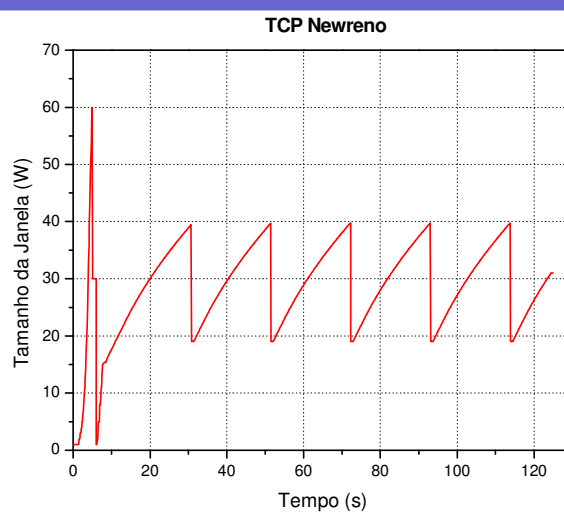
14

## Controle de Congestionamento do TCP

- Alterar a versão do protocolo TCP de **Agent/TCP** para:
  - **set tcp [new Agent/TCP/Newreno]**
  - Traçar o gráfico **tamanho da janela** versus **tempo** do arquivo **WinFile**
  - Analisar o *throughput* (vazão) do fluxo TCP (0-4) utilizando o software TraceGraph
- Para diminuir o número de ACKs no sistema o TCP frequentemente utiliza ACKs atrasados, onde um ACK é transmitido para cada 2 pacotes que chegam.
  - **set sink [new Agent/TCPSink/DelAck]**

15

## Controle de Congestionamento do TCP



16



## Versões do Protocolo TCP

- Agentes TCP de envio (one-way) suportados:
  - Agent/TCP - a "tahoe" TCP sender
  - Agent/TCP/Reno - a "Reno" TCP sender
  - Agent/TCP/Newreno - Reno with a modification
  - Agent/TCP/Sack1 - TCP with selective repeat (segue RFC2018)
  - Agent/TCP/Vegas - TCP Vegas
  - Agent/TCP/Fack - Reno TCP with "forward acknowledgment"
- Agentes TCP de recepção (one-way) suportados:
  - Agent/TCPSink - TCP sink with one ACK per packet
  - Agent/TCPSink/DelAck - TCP sink with configurable delay per ACK
  - Agent/TCPSink/Sack1 - selective ACK sink (segue RFC2018)
  - Agent/TCPSink/Sack1/DelAck - Sack1 with DelAck

17

## Controle de Congestionamento do TCP

- Executar o `exemplo6.tcl` (nam v1.0a10)
- Este exemplo permite monitorar as variáveis do TCP no `nam`
  - `cwnd_`
  - `rtt_`
  - `ssthresh_`

18

## Controle de Congestionamento do TCP

- Executar o **exemplo6.tcl** (nam v1.0a10)

```
1 #Controle de congestionamento no TCP
2 set ns [new Simulator]           ;# Cria o objeto "simulador"
3
4 set nf [open out.nam w]          ;# Abre o arquivo de trace do nam
5 $ns namtrace-all $nf
6
7 proc finish {} {                 ;# Define procedimento a ser executado no final
8     global ns nf
9     $ns flush-trace
10    close $nf
11    exec nam out.nam &
12    exit 0
13 }
14 set n0 [$ns node] ; set n1 [$ns node] ; #Cria 4 nós
15 set n2 [$ns node] ; set n3 [$ns node]
16
17 $ns duplex-link $n0 $n2 1Mb 10ms DropTail ;# Cria links entre os nós
18 $ns duplex-link $n1 $n2 1Mb 10ms DropTail
19 $ns duplex-link $n2 $n3 1Mb 10ms DropTail
20
```

19

## Controle de Congestionamento do TCP

```
21 $ns queue-limit $n2 $n3 10 ;# Limita fila enlace 2-3 (10 pacotes)
22 $ns duplex-link-op $n2 $n3 queuePos 0.5 ;# monitora a fila do enlace 2-3
23
24 # Orientações para o nam
25 $ns duplex-link-op $n1 $n2 orient up
26 $ns duplex-link-op $n0 $n2 orient right
27 $ns duplex-link-op $n2 $n3 orient right
28 Agent/TCP set nam_tracevar_ true ;# Habilita "trace" de var. TCP (nam)
29
30 set tcp0 [new Agent/TCP]         ;# Cria agente TCP
31 $ns attach-agent $n0 $tcp0       ;# Anexa agente tcp0 ao nó 0
32 $tcp0 set packet_size_ 1500     ;# Tamanho máximo de pacote em bytes
33 set sink0 [new Agent/TCPSink]    ;# Cria agente TCP consumidor
34 $ns attach-agent $n3 $sink0      ;# Anexa agente sink0 ao nó 3
35 $ns connect $tcp0 $sink0         ;# Connect TCP source with TCP sink
36 set ftp0 [$tcp0 attach-source FTP] ;# Cria aplic. FTP e anexa ao tcp0
37
```

20

## Controle de Congestionamento do TCP

```
38 set tcp1 [new Agent/TCP]    ;# Cria agente TCP
39 $ns attach-agent $n1 $tcp1  ;# Anexa agente tcp1 ao nó 1
40 $tcp1 set packet_size_ 1500 ;# Tamanho máximo de pacote em bytes
41 set sink1 [new Agent/TCPSink] ;# Cria agente TCP consumidor
42 $ns attach-agent $n3 $sink1  ;# Anexa agente sink1 ao nó 3
43 $ns connect $tcp1 $sink1     ;# Connect TCP source with TCP sink
44 set ftp1 [$tcp1 attach-source FTP] ;# Cria aplic FTP e anexa ao tcp1
45
46 # Define cores para os fluxos de dados tcp0=azul, tcp1=vermelho
47 $ns color 1 Blue ; $ns color 2 Red
48 $tcp0 set fid_ 1 ; $tcp1 set fid_ 2
49
50 # Adiciona o rastreamento de variáveis
51 $ns add-agent-trace $tcp0 tcp0 ;# label "tcp0"
52 $ns add-agent-trace $tcp1 tcp1 ;# label "tcp1"
53 $ns monitor-agent-trace $tcp0 ;# (nam) monitorar variáveis do tcp0
54 $ns monitor-agent-trace $tcp1 ;# (nam) monitorar variáveis do tcp1
55 $tcp0 tracevar cwnd_         ;# rastreia a variável cwnd_ do tcp0
56 $tcp1 tracevar cwnd_         ;# rastreia a variável cwnd_ do tcp1
57
```

21

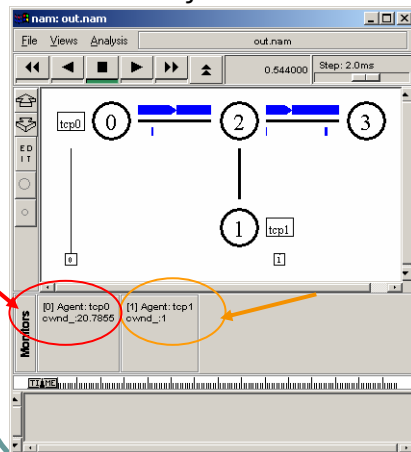
## Controle de Congestionamento do TCP

```
58 # Programa os eventos da simulação
59 $ns at 0.1 "$ftp0 start"
60 $ns at 1.0 "$ftp1 start"
61 $ns at 14.0 "$ftp1 stop"
62 $ns at 14.5 "$ftp0 stop"
63 $ns at 25.0 "finish"
64
65 # Executa a simulação
66 $ns run
```

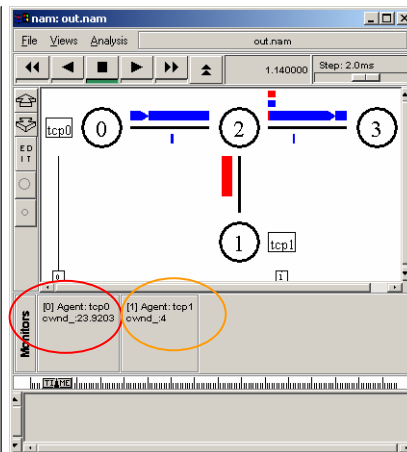
22

## Exemplo - Animação

TCP0 já iniciou



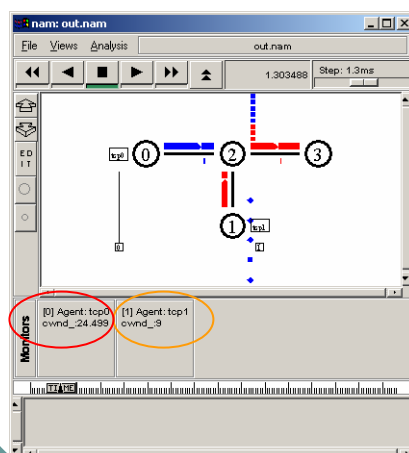
TCP1 iniciou



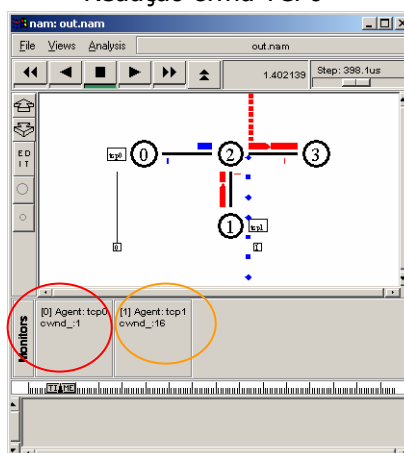
23

## Exemplo - Animação

Descarte fluxo TCP0

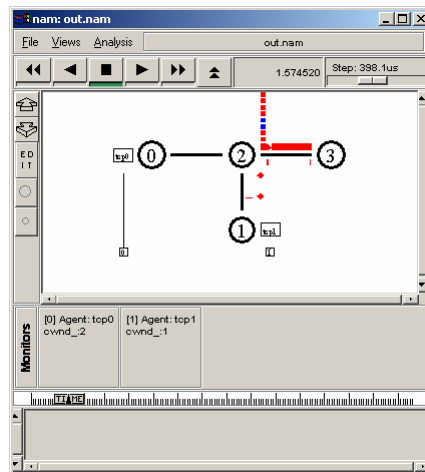


Redução cwnd TCP0



24

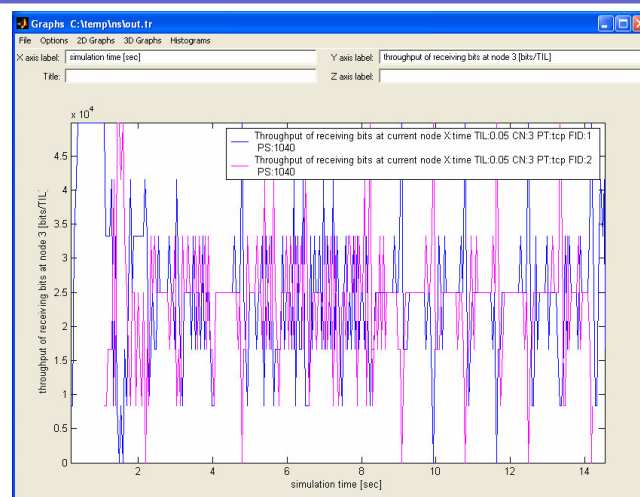
## Exemplo - Animação



Redução cwnd TCP1

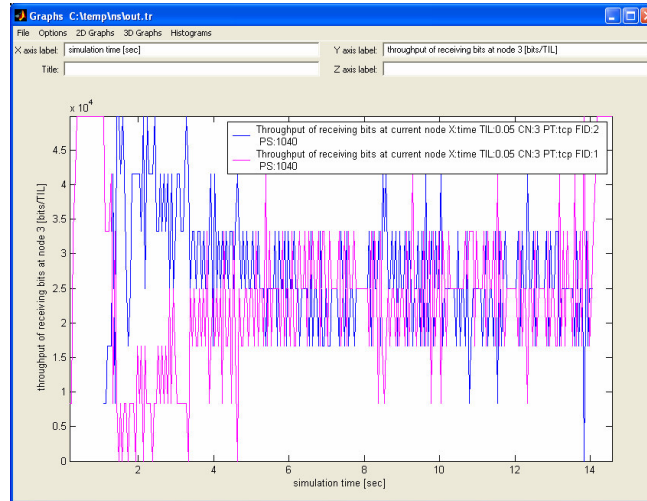
25

## Vazão TCP Tahoe



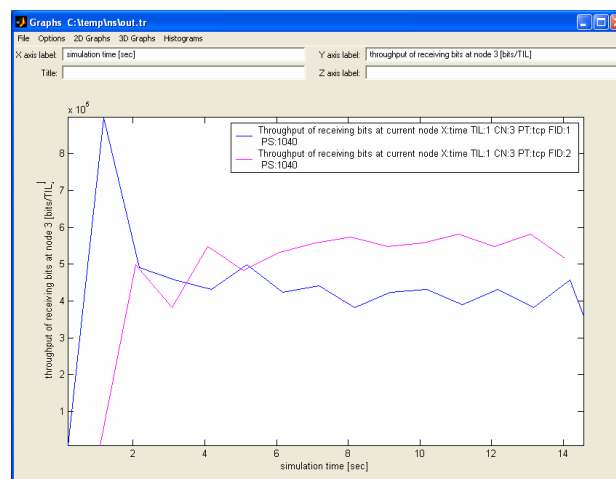
26

## TCP NewReno - vazão



27

## TCP Tahoe vs NewReno



28

## TCP sobre Canais Ruidosos

- Nos exemplos anteriores as perdas foram devido ao congestionamento.
- Em enlaces de rádio (satélite/celular/wlan) as perdas podem ocorrer devido a ruído e/ou interferência.
- Podemos introduzir modelos de erro nos enlaces: (exemplo7.tcl)
  - #Set error model on link n2 to n3.
  - set loss\_module [new ErrorModel]
  - \$loss\_module set rate\_ 0.2 (taxa de erro do enlace = 20%)
  - \$loss\_module ranvar [new RandomVariable/Uniform]
  - \$loss\_module drop-target [new Agent/Null]
  - \$ns lossmodel \$loss\_module \$n2 \$n3

29

## TCP sobre Canais Ruidosos

- Executar exemplo7.tcl
  - Traçar o gráfico da janela de congestionamento e vazão para uma taxa de erro de 20%.
  - Traçar o gráfico da janela de congestionamento e vazão sem erros no enlace.

30

## TCP sobre Canais Ruidosos

- Monitoramento da fila (**exemplo7.tcl**)
  - `$ns monitor-queue $n2 $n3 [open qm.out w] 0.1`
  - `[$ns link $n2 $n3] queue-sample-timeout`
- Estrutura do arquivo **qm.out** (11 colunas)
  - Tempo
  - Nós de entrada e saída que definem a fila
  - Tamanho da fila em bytes (size\_)
  - Tamanho da fila em pacotes (pkt\_)
  - Número de pacotes que chegaram (parrivals\_)
  - Número de pacotes que saíram do enlace (pdepartures\_)
  - Número de pacotes descartados na fila (pdrops\_)
  - Número de bytes que chegaram (barrivals\_)
  - Número de bytes que saíram do enlace (bdepartures\_)
  - Número de bytes descartados (bdrops\_)

31

## Criando Conexões Aleatoriamente

- Para **criar várias conexões** é mais apropriado se definir **vetores** ao invés de especificar individualmente cada nó, enlace, conexão e aplicações.
- O **exemplo8.tcl** cria 5 conexões FTP com início aleatório.
- O tempo de início é uniformemente distribuído entre **0 e 7s**.
- A duração total da simulação é de 10s.
- Os atrasos dos enlaces também são sorteados uniformemente entre **1ms e 5ms**.

32



## Roteamento e Dinâmica da Rede

- Roteamento Estático\Dinâmico
- Roteamento Unicast\Multicast
- O NS-2 pode simular a conexão e desconexão de enlaces:
  - `$ns rtmodel-at 1.0 down $n1 $n2`
  - `$ns rtmodel-at 2.0 up $n1 $n2`
- O roteamento default do NS-2 é o **roteamento unicast estático**, onde a rota mais curta (em termos do número de saltos) é escolhida para conexão

33

## Roteamento e Dinâmica da Rede

- Executar o `exemplo9.tcl`
- É estabelecida uma conexão TCP entre os **nós 0-5**
- O enlace 1-4 será desconectado entre **1s e 4.5s**
- Com roteamento estático (`# $ns rproto DV`), mesmo com a conexão restaurada do enlace em 4.5s, a conexão TCP se restaura apenas em 8s
- A razão é que ocorrem timeout devido a ausência de ACKs para o nó 0, e a duração do timeout dobra com cada novo timeout
- Executar o `exemplo9.tcl` com roteamento dinâmico:
  - `$ns rproto DV`
- Executar o `exemplo10.tcl`

34

## Roteamento Dinâmico

```
1 Roteamento dinâmico
2 set ns [new Simulator]
3 # Indica para o ns que use "dynamic routing"
4 $ns rtproto DV
5
6 # Cria o trace do nam
7 $ns namtrace-all [open validate.nam w]
8
9 # Importa arquivo que contém o procedimento "create_topology"
10 source dumbbell.tcl
11 global num_node n
12
13 create_topology
14
15 set cbr [new Application/Traffic/CBR]
16 set udp [new Agent/UDP]
17 set sink [new Agent/Null]
18 $cbr attach-agent $udp
19 $ns attach-agent $n(8) $udp
20 $ns attach-agent $n(5) $sink
21 $ns connect $udp $sink
22 $cbr set packetSize_ 1000
23 $cbr set interval_ 0.005
24
```

exemplo11.tcl

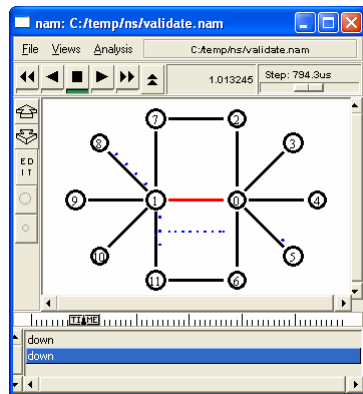
35

## Roteamento Dinâmico

```
25 $ns color 1 Blue ; $udp set fid_ 1
26
27 $ns at 0.0 "$cbr start"
28 $ns rtmodel-at 1 down $n(0) $n(1)
29 $ns rtmodel-at 2 down $n(0) $n(6)
30 $ns rtmodel-at 3 up $n(0) $n(6)
31 $ns rtmodel-at 4 up $n(0) $n(1)
32 $ns at 5 "$cbr stop"
33 $ns at 6 "finish"
34
35 proc finish {} {
36     global ns
37     $ns flush-trace
38     exec nam validate.nam &
39     exit 0
40 }
41 # Inicia a simulação
42 $ns run
```

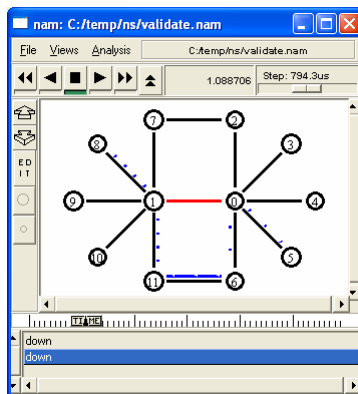
36

## Animação



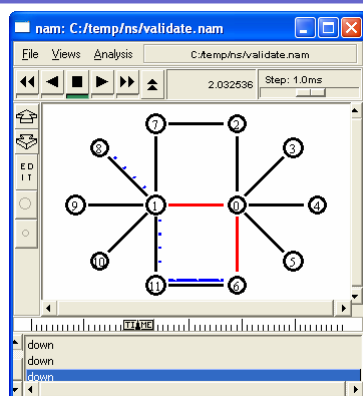
Enlace 0-1 sai fora do ar

Rota alternativa



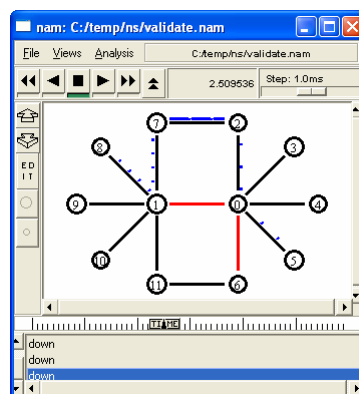
37

## Animação



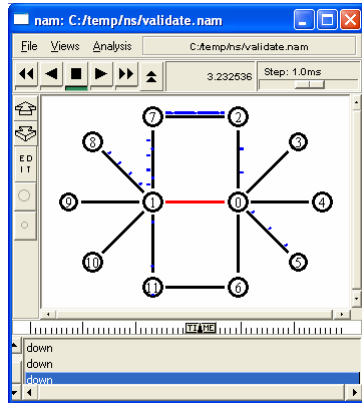
Enlace 0-6 sai fora do ar

Rota alternativa

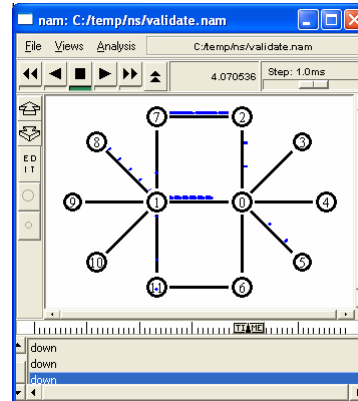


38

## Animação



Enlace 0-1 volta ao ar



39

## Validação de Resultados

- Intervalo de Confiança
  - Compreende um valor numérico que possui uma probabilidade igual a  $(1-\alpha)$  de incluir o verdadeiro valor da variável ou medida de desempenho sob análise, onde  $(1-\alpha)$  é denominado **nível de confiança do intervalo**.
  - Alfa é o erro admitido ao se concluir sobre a presença do verdadeiro valor da variável no intervalo calculado.

40

## Validação de Resultados

- Intervalo de Confiança – Exemplo

Número da Replicação	Tempo de Resposta
1	63,2
2	69,9
3	67,3
4	64,8
5	72

- Média = 67,44
- Variância = 12,95      Desvio Padrão = 3,59

41

## Validação de Resultados

- O semi-intervalo  $h$  é definido como:

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

- onde  $n$  é o número de intervalos,  $\sigma$  é o desvio padrão e  $t$  é a distribuição  $t$ -student.
- Para 99% de confiança  $\alpha=0,01$  e  $t_{4,0,995}=4,6$
- O valor calculado de  $h$  é 7,40
- Os limites para 99% de confiança serão [60,04 – 74.84]

$$\bar{x} \pm t_{n-1, 1-\alpha/2} \cdot \frac{\sigma}{\sqrt{n}}$$

42

## Validação de Resultados

- Distribuição t-student

- Dado  $n$  medidas independentes  $x_i$ , onde  $\mu$  é a média da população,  $\bar{x}$  é a média amostral e  $s$  o desvio padrão estimado

$$t = \frac{\bar{x} - \mu}{s / \sqrt{n}}$$

$$s^2 = \frac{1}{N-1} \sum_{i=1}^N (x_i - \bar{x})^2$$

43