

Universidade Federal Fluminense
Departamento de Engenharia de Telecomunicações
Mestrado em Engenharia de Telecomunicações

Aula de introdução ao NS-2

Disciplina: Fundamentos de Sistemas Multimídia
Professora: Débora Christina Muchaluat Saade
Monitor: Diogo Lino P. Machado

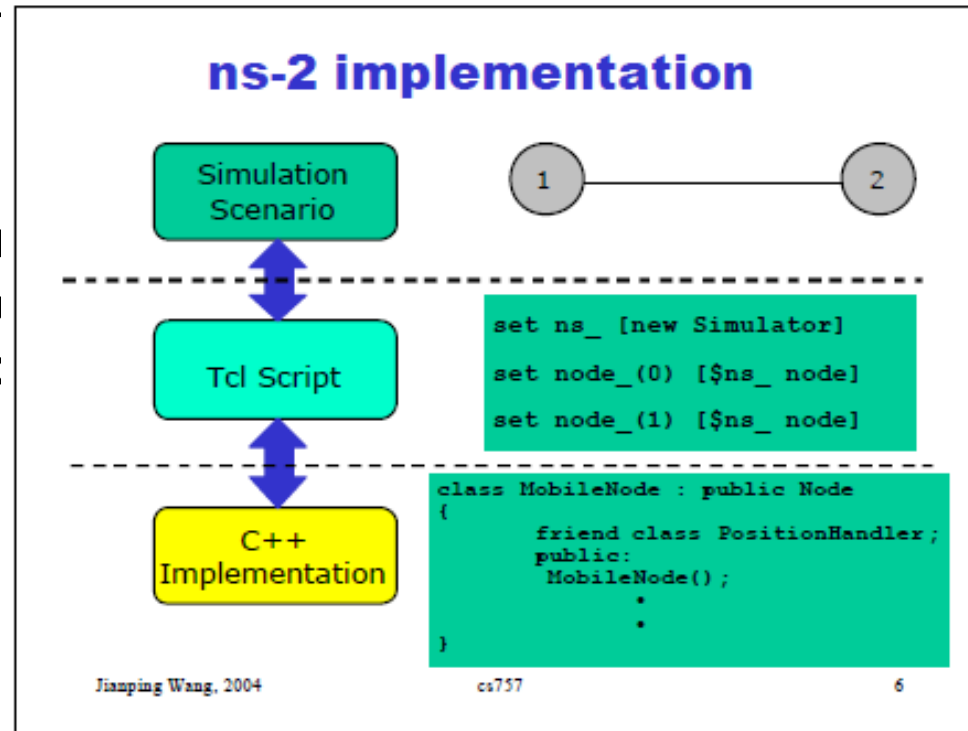
Abril/2011

NS2 – Tópicos gerais

- Simulador de eventos para simulações de rede em geral;
 - Gratuito e possui código aberto;
 - Possibilidade de simulação de distintos cenários:
 - Diferentes topologias (quantidade de nós, links, delay, BW, etc.);
 - Variação do protocolo de transporte (TCP/UDP);
 - Baseado em pacotes;
 - Suporte a redes WLAN 802.11 com diversos protocolos de roteamento (AODV,DSR, TORA, OLSR, etc.);
 - Suporte ao modelo QoS DiffServ com diferentes modelos de admissão e políticas de enfileiramento (FIFO, PRI, WRR,etc.) e descarte (DropTail, RED, WRED, etc.);
 - Agentes são associados aos nós e responsáveis pela geração de diversos tráfegos;
-

NS2 – Tópicos gerais

- Utiliza 2 linguagens: C++ para estrutura básica (protocolos, agentes, etc.) e OTCL como *front-end* para as simulações;
- C++: linguagem para pacotes, cabeçalhos, interpretada a rede simulada



ulação de bytes,
CL: linguagem
nfigurações da

Criando Simulações

- Criação do objeto simulador (escalonador de eventos);
 - Abertura de arquivos para análise (trace);
 - Criação da topologia da rede (nós e enlaces) e seus parâmetros (delay, BW, disposição física, etc.);
 - Criação dos agentes da camada de transporte e associação aos nós;
 - Criação dos geradores de tráfego e conexão com os agentes da camada de transporte;
 - Programação dos eventos (início/fim da transmissão);
 - Fechamento da simulação, animação (NAM) e análise do formato *trace*;
-

Criação do objeto simulador

- Define a variável que inicia a simulação:
`set ns [new Simulator]`
 - Instancia o objeto simulador e o associa à variável ns;
 - Inicializa o formato dos pacotes;
 - Cria um escalonador de eventos;
 - Seleciona o formato padrão de endereçamento;
-

Criando Simulações

- Criação do objeto simulador (escalonador de eventos);
 - Abertura de arquivos para análise (trace);
 - Criação da topologia da rede (nós e enlaces) e seus parâmetros (delay, BW, disposição física, etc.);
 - Criação dos agentes da camada de transporte e associação aos nós;
 - Criação dos geradores de tráfego e conexão com os agentes da camada de transporte;
 - Programação dos eventos (início/fim da transmissão);
 - Fechamento da simulação, animação (NAM) e análise do formato *trace*;
-

Abertura de arquivos de trace

- Pode-se abrir um arquivo que servirá de base para a construção de uma animação gráfica (.nam):

```
set nf [open out.nam w] # cria um arquivo que será lido pelo NAM  
$ns namtrace-all $nf # diz ao simulador para gravar os passos no  
arquivo out.nam
```

- Pode-se abrir um arquivo para gravar os traces e possibilitar posterior análise dos dados (.tr):

```
set tf [open out.tr w] # cria o arquivo de trace em formato geral  
$ns trace-all $tf # grava as informações no arquivo out.tr
```

Criando Simulações

- Criação do objeto simulador (escalonador de eventos);
 - Abertura de arquivos para análise (trace);
 - Criação da topologia da rede (nós e enlaces) e seus parâmetros (delay, BW, disposição física, etc.);
 - Criação dos agentes da camada de transporte e associação aos nós;
 - Criação dos geradores de tráfego e conexão com os agentes da camada de transporte;
 - Programação dos eventos (início/fim da transmissão);
 - Fechamento da simulação, animação (NAM) e análise do formato *trace*;
-

Criação da topologia da rede

- Definição dos nós da topologia:

`set n0 [$ns node] # define a criação do nó n0 para o objeto simulador ns`

`set n1 [$ns node]`

- para grande quantidade de nós:

`set num_nodes 50`

`for {set j 1} {$j <= $num_nodes} {incr j} {
 set n($j) [$ns node] }`

- Criação da topologia:

`$ns <simplex/duplex-link> <node0> <node1> <BW> <delay> <tipo de fila>`

- tipo de fila pode ser DropTail ou dsRED quando se está dentro do domínio DiffServ;

`$ns <simplex/duplex-link-op> <node0> <node1> <orientação>`

Criando Simulações

- Criação do objeto simulador (escalonador de eventos);
 - Abertura de arquivos para análise (trace);
 - Criação da topologia da rede (nós e enlaces) e seus parâmetros (delay, BW, disposição física, etc.);
 - Criação dos agentes da camada de transporte e associação aos nós;
 - Criação dos geradores de tráfego e conexão com os agentes da camada de transporte;
 - Programação dos eventos (início/fim da transmissão);
 - Fechamento da simulação, animação (NAM) e análise do formato *trace*;
-

Criação dos agentes da camada de transporte

- TCP e UDP são os mais utilizados, embora o NS suporte outros agentes como o TCP;
 - Exemplo de agente UDP emissor:
`set udp0 [new Agent/UDP] # cria o agente udp0`
`$ns attach-agent $n0 $udp0 # associa o agente udp0 ao nó n0`
 - Exemplo de agente UDP receptor:
`set null0 [new Agent/Null] # cria o agente null0`
`$ns attach-agent $n1 $null0 # associa o agente null0 ao nó n1`
 - Estabelecimento do canal de comunicação:
`$ns connect $udp0 $null0 # conecta os agentes udp0 e null0`
“A geração de tráfego no NS é baseada nas classes *Agent* e *Application*. Cada nó que necessite de enviar ou receber dados necessita de um agente associado. No topo do agente roda uma aplicação e esta determina o tráfego a ser gerado”.
-

Criando Simulações

- Criação do objeto simulador (escalonador de eventos);
 - Abertura de arquivos para análise (trace);
 - Criação da topologia da rede (nós e enlaces) e seus parâmetros (delay, BW, disposição física, etc.);
 - Criação dos agentes da camada de transporte e associação aos nós;
 - Criação dos geradores de tráfego e conexão com os agentes da camada de transporte;
 - Programação dos eventos (início/fim da transmissão);
 - Fechamento da simulação, animação (NAM) e análise do formato *trace*;
-

Criação dos geradores de tráfego e conexão aos agentes

- Determina a aplicação que gerará o tráfego no canal de comunicação;
- Determina parâmetros como tamanho do pacote e intervalo de transmissão:

`$ns color 3 orange`

`set tcp1 [new Agent/TCP/Newreno] # tipo do protocolo`

`$tcp1 set class_ 3 # numero associado a uma cor visualizada no NAM`

`$tcp1 wet fid_ 1 # fluxo id`

`$tcp 1 set windows_ 2000 # tamanho da janela tcp`

`set ftp1 [$tcp1 attach-source FTP] # associa o fluxo ftp1 ao agente tcp1`

Criando Simulações

- Criação do objeto simulador (escalonador de eventos);
 - Abertura de arquivos para análise (trace);
 - Criação da topologia da rede (nós e enlaces) e seus parâmetros (delay, BW, disposição física, etc.);
 - Criação dos agentes da camada de transporte e associação aos nós;
 - Criação dos geradores de tráfego e conexão com os agentes da camada de transporte;
 - Programação dos eventos (início/fim da transmissão);
 - Fechamento da simulação, animação (NAM) e análise do formato *trace*;
-

Programação dos eventos

- A associação dos diversos eventos aos instantes de tempo determinam o comportamento da simulação;

\$ns at 0.5 “\$cbr0 start”

\$ns at 4.5 “\$cbr0 stop”

\$ns at 5.0 “finish”

Criando Simulações

- Criação do objeto simulador (escalonador de eventos);
 - Abertura de arquivos para análise (trace);
 - Criação da topologia da rede (nós e enlaces) e seus parâmetros (delay, BW, disposição física, etc.);
 - Criação dos agentes da camada de transporte e associação aos nós;
 - Criação dos geradores de tráfego e conexão com os agentes da camada de transporte;
 - Programação dos eventos (início/fim da transmissão);
 - Fechamento da simulação, animação (NAM) e análise do formato *trace*;
-

Fechamento da simulação

- As simulações e arquivos de trace devem ser encerrados:

```
proc finish {} {  
  global ns nf  
  $ns flush-trace  
  close $nf  
  exec nam out.nam &  
  exit 0  
}  
$ns run #executa a simulação e inicia o escalonador de eventos
```