CPE710: Redes Móveis

Prof. Miguel Elias Mitre Campista

http://www.gta.ufrj.br/~miguel

CPE710: Redes Móveis

INTRODUÇÃO AO NS-3

Preâmbulo

- Este tutorial usa a variável **$(VERSAO)** a ser substituída pela versão mais atual
- Note que novas versões podem surgir e o tutorial talvez precise de atualizações
  - A versão instalada foi a 3.26
- Instruções de instalação do ns-3
  - Disponível em maiores detalhes em:

http://www.nsnam.org/docs/release/$(VERSAO)/tutorial-pt-br/singlehtml/index.html

Download

- Baixar a última versão em:

```
https://www.nsnam.org/release/ns-allinone-$(VERSAO).tar.bz2
```

Instalação

- Escolher um diretório para descompactação (sugestão **/home/$(USUARIO)**), onde a variável **$(USUARIO)** deve ser substituída pelo nome de algum diretório
  - A descompactação é feita da seguinte maneira:

```
tar -jxvf ns-allinone-$(VERSAO).tar.bz2
```

- Entrar no diretório criado e executar o comando de compilação do simulador:

```
cd /home/$(USUARIO)/ns-allinone-$(VERSAO)

./build.py --enable-examples --enable-tests
```

Instalação

- As opções **`--enable-examples --enable-tests`** são incluídas para que os exemplos e os testes sejam também construídos
  - Exemplos e testes não são construídos por padrão

- Após o término da compilação, uma mensagem como a seguir deverá ser exibida na tela:

Instalação

```
Waf: Leaving directory `/home/$(USUARIO)/ns-allinone-$(VERSAO)/ns-
$(VERSAO)/build'
'build' finished successfully (29m13.153s)
Modules built:
antenna                  aodv                     applications
bridge                   buildings                config-store
core                     csma                     csma-layout
dsdv                     dsr                      energy
fd-net-device            flow-monitor             internet
internet-apps            lr-wpan                  lte
mesh                     mobility                 mpi
netanim (no Python)      network                  nix-vector-routing
olsr                     point-to-point           point-to-point-layout
propagation              sixlowpan                spectrum
stats                    tap-bridge               test (no Python)
topology-read            traffic-control          uan
virtual-net-device       visualizer               wave
wifi                     wimax

Modules not built (see ns-3 tutorial for explanation):
brite                    click                       openflow
Leaving directory `./ns-$(VERSAO)'
```

Instalação

- Entrar no diretório de desenvolvimento do ns e em seguida configurar uma versão em modo de depuração (opção debug) para o ns:

```
cd /home/$(USUARIO)/ns-allinone-$(VERSAO)/ns-$(VERSAO)

./waf -d debug --enable-examples --enable-tests configure
```

- Em seguida, realizar a construção do ns:

```
./waf
```

Instalação

- **Neste ponto, o ns está construído e pronto para uso!**
  - Para se certificar que a instalação está correta, o ns disponibiliza uma sequência de testes para avaliar a instalação realizada
    - Alguns testes vão depender dos módulos instalados na máquina

```
./test.py -c core
```

- ***Máquinas mais lentas podem travar durante os testes!***

Execução de Scripts

- Executar o primeiro script (*Hello Simulator*)
  - Ainda no diretório `/home/$(USUARIO)/ns-allinone-$(VERSAO)/ns-$(VERSAO)`, executar:

```
./waf --run hello-simulator
```

- Ao final da execução, a mensagem ***Hello Simulator*** deve ser exibida na tela
  - Caso isso ocorra, a instalação foi bem sucedida!

- Em seguida, todos os programas de teste poderão ser usados da seguinte forma:

```
./waf --run "nome_do_programa"
```

Execução de Scripts

- Todos os programas de teste podem ser encontrados em um diretório padrão do ns-3
  - Localizado em `/home/$(USUARIO)/ns-allinone-$(VERSAO)/ns-$(VERSAO)/examples/tutorial`

- Veja os resultados de cada simulação
  - Resultados impressos na tela!
  - Resultados gerados como arquivo!

Arquivo first.cc

- Simula uma aplicação do tipo **echo client/server**
  - Execute a simulação com o comando:

```
./waf --run first
```

  - Saída

```
Waf: Entering directory `/home/miguel/install/ns3/ns-allinone-3.26/ns-3.26/build'
[ 952/2629] Compiling examples/tutorial/first.cc
[2450/2629] Linking build/examples/tutorial/ns3.26-first-debug
Waf: Leaving directory `/home/miguel/install/ns3/ns-allinone-3.26/ns-3.26/build'
Build commands will be stored in build/compile_commands.json
'build' finished successfully (3.476s)
At time 2s client sent 1024 bytes to 10.1.1.2 port 9
At time 2.00369s server received 1024 bytes from 10.1.1.1 port 49153
At time 2.00369s server sent 1024 bytes to 10.1.1.1 port 49153
At time 2.00737s client received 1024 bytes from 10.1.1.2 port 9
```

Abra o arquivo first.cc
gvim ns-$(VERSAO)/examples/tutorial/first.cc

```
/* -*- Mode:C++; c-file-style:"gnu"; indent-tabs-mode:nil; -*- */
 2 /*
 3  * This program is free software; you can redistribute it and/or modify
 4  * it under the terms of the GNU General Public License version 2 as
 5  * published by the Free Software Foundation;
 6  *
 7  * This program is distributed in the hope that it will be useful,
 8  * but WITHOUT ANY WARRANTY; without even the implied warranty of
 9  * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
10  * GNU General Public License for more details.
11  *
12  * You should have received a copy of the GNU General Public License
13  * along with this program; if not, write to the Free Software
14  * Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA
15  */
16
17 #include "ns3/core-module.h"
18 #include "ns3/network-module.h"
19 #include "ns3/internet-module.h"
20 #include "ns3/point-to-point-module.h"
21 #include "ns3/applications-module.h"
```

Abra o arquivo first.cc
gvim ns-$(VERSAO)/examples/tutorial/first.cc

```
22
23 using namespace ns3;
24
25 NS_LOG_COMPONENT_DEFINE ("FirstScriptExample");
26
27 int
28 main (int argc, char *argv[])
29 {
30  Time::SetResolution (Time::NS);
31  LogComponentEnable ("UdpEchoClientApplication", LOG_LEVEL_INFO);
32  LogComponentEnable ("UdpEchoServerApplication", LOG_LEVEL_INFO);
33
34  NodeContainer nodes;
35  nodes.Create (2);
36
37  PointToPointHelper pointToPoint;
38  pointToPoint.SetDeviceAttribute ("DataRate", StringValue ("5Mbps"));
39  pointToPoint.SetChannelAttribute ("Delay", StringValue ("2ms"));
40
41  NetDeviceContainer devices;
42  devices = pointToPoint.Install (nodes);
43
44  InternetStackHelper stack;
45  stack.Install (nodes);
46
```

Abra o arquivo first.cc
gvim ns-$(VERSAO)/examples/tutorial/first.cc

```
47  Ipv4AddressHelper address;
48  address.SetBase ("10.1.1.0", "255.255.255.0");
49
50  Ipv4InterfaceContainer interfaces = address.Assign (devices);
51
52  UdpEchoServerHelper echoServer (9);
53
54  ApplicationContainer serverApps = echoServer.Install (nodes.Get (1));
55  serverApps.Start (Seconds (1.0));
56  serverApps.Stop (Seconds (10.0));
57
58  UdpEchoClientHelper echoClient (interfaces.GetAddress (1), 9);
59  echoClient.SetAttribute ("MaxPackets", UintegerValue (1));
60  echoClient.SetAttribute ("Interval", TimeValue (Seconds (1.0)));
61  echoClient.SetAttribute ("PacketSize", UintegerValue (1024));
62
63  ApplicationContainer clientApps = echoClient.Install (nodes.Get (0));
64  clientApps.Start (Seconds (2.0));
65  clientApps.Stop (Seconds (10.0));
66
67  Simulator::Run ();
68  Simulator::Destroy ();
69  return 0;
70 }
```

Execução Automatizada das Simulações

- Requer a criação de scripts
  - Script em **tcsh**
    - Usado para automatizar as simulações

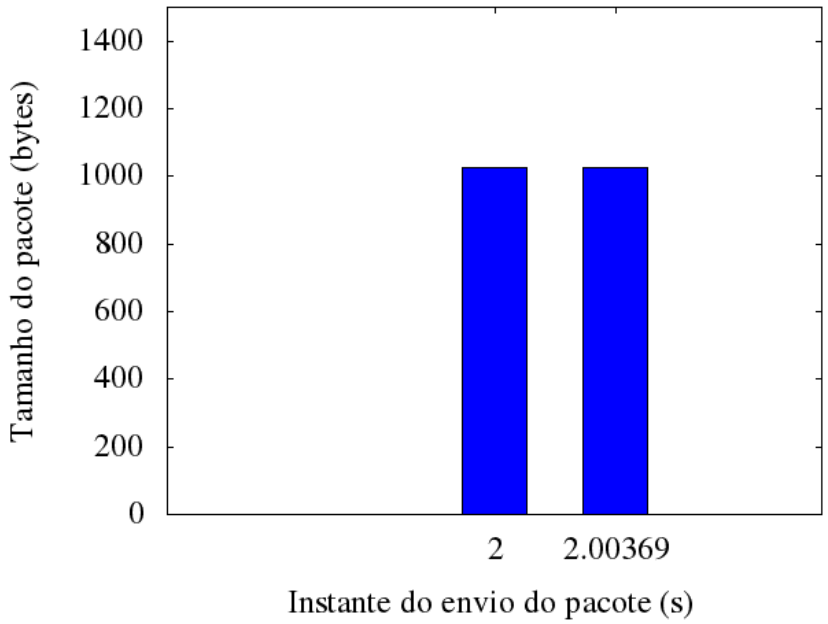
```
#!/usr/bin/tcsh -f

set HOMEDIR=`pwd`
set NSDIR="/home/miguel/install/ns3/ns-allinone-3.26/ns-3.26"

cd $NSDIR

./waf --run first >& $HOMEDIR/saida.dat

cd $HOMEDIR

grep "sent" saida.dat | awk '{print $3, $6;}' | awk -F s '{print $1, $2;}' > log.dat

gnuplot plotapacotes.gnu

rm saida.dat log.dat
```

Execução Automatizada das Simulações

- Requer a criação de scripts
  - Script **gnuplot**
    - Usado para plotar gráficos

```
set terminal png enhanced font 'Times,18'
set output "envio.png"

set style data histogram
set style fill solid border -1
set boxwidth 0.002

set ylabel 'Tamanho do pacote (bytes)'
set xlabel 'Instante do envio do pacote (s)'

set yrange [0:1500]
set xrange [1.99:2.01]
set xtics (2, 2.00369)

plot 'log.dat' using 1:2 with boxes lc rgb 'blue' notitle
```

Execução Automatizada das Simulações



Lembrando a saída da simulação...

```
At time 2s client sent 1024 bytes to 10.1.1.2 port 9
At time 2.00369s server sent 1024 bytes to 10.1.1.1 port 49153
```