Networks and System Security 2014

Lab #3: TCP Performance Monitoring

ns-3 Tutorial
Lab date: Nov. 13 & 24 2014

1 Installation

ns-3 is the discrete event simulator used for networking research. It supports simulation of TCP, routing, and multicast protocols over wired and wireless networks.

Install from source:

- Homepage: http://www.nsnam.org/
- Version: ns-3.17
- Source code: https://www.nsnam.org/release/ns-allinone-3.17.tar.bz2
- Use *build.py* to build.

To install the ns-3 in the linux lab machines, you should download and unzip the ns-allinone-3.17.zip file from the blackboard. If you don't have enough space in your home directory, then use the scratch directory. However don't forget to save your work and delete it from the scratch directory, before to leave.

2 Related documents

- Installation guide: http://www.nsnam.org/wiki/index.php/Installation
- ns-3 manual: http://www.nsnam.org/docs/release/3.17/manual/html/index.html
- ns-3 tutorial: http://www.nsnam.org/docs/release/3.17/tutorial/ns-3-tutorial.pdf

3 NS-3 crash course

3.1 Run simulation script

A simple way to run a simulation script (e.g. example.cc) is to move the script to $NS3_HOME/scratch folder and execute the following command:

```
./waf --run scratch/example
```

Be careful, all the scripts in the scratch directory should be able to be compiled (they should not have any errors), in order to run a simulation script.

3.2 Examples

Study the following examples:

1. Open $NS3_HOME/examples/tutorial/first.cc to study how to create nodes, setup network topology, assign IP addresses and install UDP applications.
2. Read carefully the following documentation page, for a conceptual overview of the system and for a detailed explanation of first.cc.
   http://www.nsnam.org/docs/release/3.17/tutorial/html/conceptual-overview.html
3. Open $NS3_HOME/examples/tcp/star.cc to study how to use the *OnOffHelper* class to create a TCP application client, and the *PacketSinkHelper* class to create a TCP application server.
4. The $NS3_HOME/examples/tutorial/fifth.cc demonstrates how to create a TCP socket and how to monitor the congestion window.

3.3 Simulation snippets

3.3.1 Congestion Window Monitoring

Monitor congestion window (CWND) changes: see $NS3_HOME/examples/tutorial/fifth.cc

Listing 1: Monitor the TCP Congestion Window at the client node

```
/**
 * Callback function to handle CWND changes
 */
void CWndTracer (uint32_t oldCwnd, uint32_t newCwnd)
{
        std::cout << Simulator::Now ().GetSeconds () << "\t" << newCwnd  << "\n"
            );
}

// tcpSocket is the Ptr<Socket> object, pointing to the Client node socket
Ptr<Socket> tcpSocket = Socket::CreateSocket (clietnNode, TcpSocketFactory::
    GetTypeId ());
tcpSocket->TraceConnectWithoutContext ("CongestionWindow", MakeCallback (&
    CWndTracer)); //add this line at the appropriate place in your code
```

3.3.2 Set the data sending rate for an OnOffHelper application

```
onOffHelper.SetConstantRate(DataRate("2Mbps"));
```

3.3.3 TCP Tuning

1. Change Receiving Window (RWIN) (default is 65535):

```
Config::SetDefault ("ns3::TcpSocketBase::MaxWindowSize", UintegerValue(MAX_RWIN));
```

2. Change receiving buffer size (DropTailQueue) of the TCP stack (default is 100 packets):

```
Config::SetDefault ("ns3::DropTailQueue::MaxPackets", UintegerValue (MAX_PACKET));
```

3. Set socket types:
   - Tahoe: ns3::TcpTahoe
   - Reno: ns3::TcpReno
   - NewReno: ns3::TcpNewReno

```
Config::SetDefault ("ns3::TcpL4Protocol::SocketType", StringValue (tcpModel));
```

4. Disable TCP Delayed ACK:

```
Config::SetDefault ("ns3::TcpSocket::DelAckCount", UintegerValue (1));
```

3.3.4 Export the logs in a file

If you want to run a script and export the data log to a file, then use the following syntax:

```
./waf --run scratch/script > out.txt
```

The logs of the script.cc will be stored in the out.txt file under the directory ns-3.17.