Project Plan
v1.0~

November 14, 2016

Network Simulator
(nsLite)

Karthika Krishnamurthy
Naveed Mohammed Vaheed
Tahir Irshad
Nguyen Van Tan

Submitted in partial fulfillment
Of the requirements of
ELEC-A7150 C++ Programming

1.0. Introduction

1.1. Purpose

The purpose of this document is to present a detailed description of the network simulator application (nsLite). It will explain the purpose, features, constraints and future scope of nsLite.

1.2. Overview

Network simulator models a group of network nodes, and links that connect the nodes. The nodes communicate with each other by sending and receiving packets. Each node can have multiple interfaces and each interface will have its own IP address and MAC address. There are roughly two kinds of nodes: routers, that forward packets intended to other nodes; and end-hosts that run an application that either sends or receives packets (or does both). Sometimes a node can act in both roles. Each link can be described by two basic characteristics: transmission speed, that determines the interval at which new packets can be transmitted to the link; and propagation delay, which is the time it takes for packet to travel across the link. Network simulator measures the network performance with the given network say delay measurement, throughput measurement etc.

1.3. Scope of Project

nsLite is a simple network simulator that allows defining different network setups (i.e., collections of nodes and links between them), and "applications" that generate and receive packets in different network scenarios for a specified amount of time. The different network setups will be retrieved from the respective xml configuration file. Each network setup can be differentiated by the number of nodes, number of links, link costs, link transmission speed and queue size. nsLite will have a graphical UI, showing the network topology and packets traversing through the network as time progresses. The graphical UI will display statistics of how long does it take for packets to reach their destination i.e. one-way delay and how utilized different links are, for each simulation. nsLite will also produce a log at a given link, that shows each packet that has been transmitted or received, together with a timestamp.

1.4. Architecture

nsLite will have the following elements and its properties, that form the network:

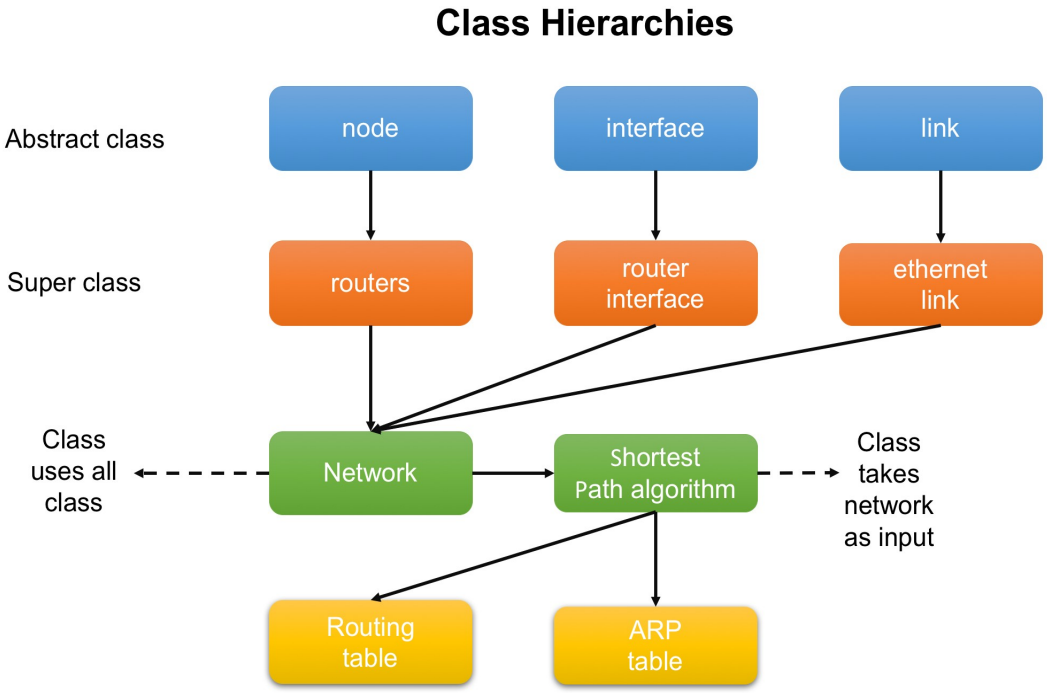
1. ***Node:***
    - Router ID
    - List of Interfaces

2. ***Interface:***
    - Interface ID
    - MAC address
    - IP address with mask
    - Queue size
    - Interface speed

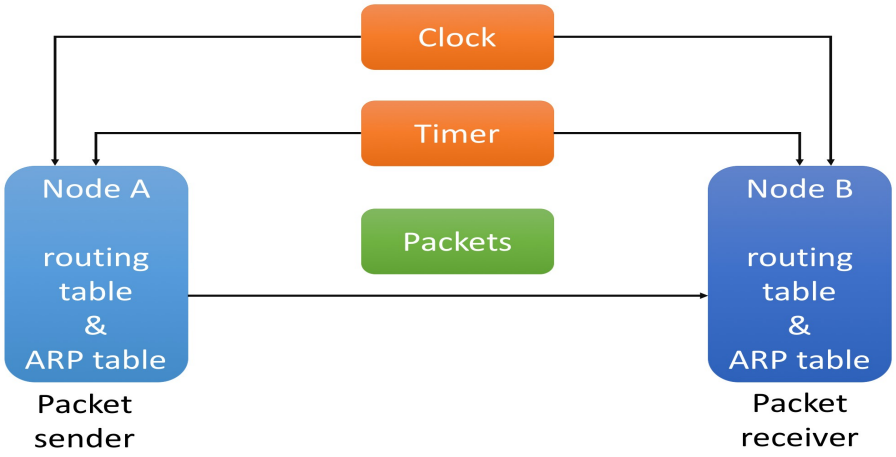
3. ***Links:***
    - Link cost
    - Transmission Speed
    - Source node
    - Source interface
    - Destination node
    - Destination interface

4. ***Packet:***
    - Ethernet II frame
    - MAC header
    - IPv4 header
    - 64Bytes size

5. ***Network*:**
    - List of nodes
    - List of links
    - Shortest path algorithm
    - Routing table
    - ARP table



Routing table and ARP table id formed from the network class which is in turn uses router, router_interface and router_link classes.



The important elements of measurements are

Clock:

Common reference clock is needed to perform delay tests between nodes. Since network simulator runs on a same physical machine, time sync between nodes is not necessary but still reference clock is needed to time-stamp the packets for delay measurement. Clock and related functionalities are implemented as separate class.

Poller:

A poller is needed to scan all ports of the routers to receive packets. This poller uses timer to poll interfaces at regular intervals. Ideally poller speed must be greater than combined speed of the interfaces. Poller and its functions are implemented as separate class.

Packet sender and Packet receiver or Packet forwarder(routing):

Active measurement test packets are sent in reference to timer. Timer must be adjusted according to the interface speed to measure throughout. Raw Ethernet packets are sent out for measurements. Packet receiver works in reference to poller. Packet receiver/Packet sender/Packet forwarder are implemented as separate classes.

1.5. Development Tools

This section explains tools used for

Git

Git repository is used for network simulator software development and version control. A single master branch is maintained in the repository.

Google Unit Test

Google Unit Test is used unit testing of each and every module of the nsLite software.

Gcov

Gov tool is used for measuring code coverage and profiling.

Valgrind
Valgrind is used for analyzing memory leaks.

1.6. Development Ethics

Google calendars:
Google calendars are used to schedule meetings to track progress of the project.

Communication channels:
Messenger and Aalto mail are used for communication between group members.

Meeting frequency:
Since the time frame to complete the project, it is important to have meeting once in a week.
It is decided to have a meeting with all members to have quick weekly update of tasks.

1.7. Tentative Plan

Phase	**Task**	**Completion Date**	**Responsible person**
Phase 1	Development of static network elements.	18/11/2016	-
Phase 2	Development of tests for developed network topology.	30/11/2016	-
Phase 3	Integration and bug fixing.	10/12/2016	-
Phase 4	Documentation	12/12/2016	-