IT 304: Computer Networks

Lab # 6: Unicast Routing in Networks - Part 1*

For the Week of September 16, 2013

Pre-lab preparation:

- Read the relevant portions of the text book on static and dynamic (specifically the distance vector) routing.

1. **Aim:** The purpose of this lab is to compare the performance of a static routing protocol and a class of dynamic routing protocols known as distance vector routing protocols.

2. **Static and Dynamic Routing:** Based on their ability to adapt to changing network conditions, routing protocols can be classified into *static* and *dynamic*. In static routing, the route between any source and destination pair remains fixed. In static routing, alternate paths will have to be specified that can be used in case of any failure. This is important in the case of static routing, since (as the name suggests) they cannot adapt to the changes in network topology.

   Dynamic routing protocols on the other hand can change routing paths as the network traffic load or topology changes. Two important classes of dynamic routing protocols are *distance vector* and *link state*. Distance vector (DV) routing protocols are based on the Bellman-Ford algorithm, while link state routing protocols are based on the Dijkstra's shortest path algorithm. Link state routing protocols would be further discussed in the next lab, in this handout, we shall focus on the DV routing protocols. Popular examples for DV routing protocols are routing information protocol (RIP), interior gateway routing protocol (IGRP), etc. DV routing protocols operate on the principle of *"routing by rumor"*, wherein all the nodes tell their neighbors periodically about the various networks that can be reached through them. Further each node also employs the *"split horizon mechanism"* in which a node does not advertise route to a destination out of the interface that it is using to reach that destination. A variant of split horizon mechanism known as the *"split horizon with poisoned reverse"* used by nodes, advertise a route through the same interface on which it learned that particular route with a metric of infinity.

   To facilitate route computation different metrics like hop count, bandwidth, load, delay, etc. can be used.

3. **Lab Scenarios:** Consider the network scenario in Fig. 1 where we have 7 nodes in a ring topology. The link details namely, rate, delay, nature of links are given in the figure. We assume that the node n0 is a source node and the node n4 to be the sink node. Further node n0 is configured with an FTP source that is in turn connected to the TCP agent. Keep the TCP packet size at their default values. Turn ON the FTP traffic at 0.1 seconds and stop the simulation at 10 seconds.

[*] © Dr. Laxminarayana S Pillutla

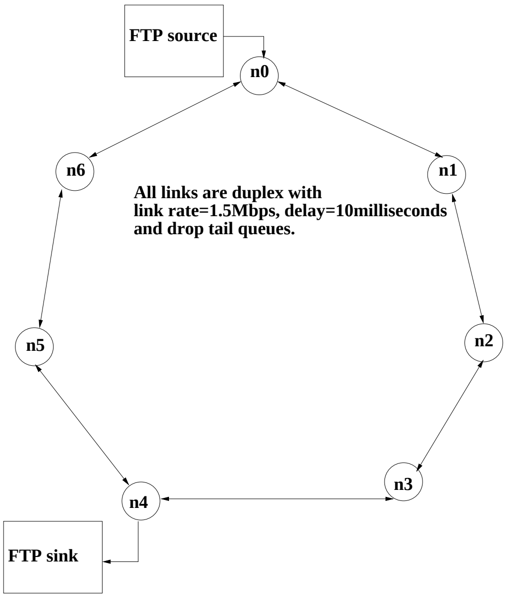
FTP source
n0
n1
n2
n3
n4
n5
n6
All links are duplex with
link rate=1.5Mbps, delay=10milliseconds
and drop tail queues.
FTP sink

Figure 1: The figure above shows the network scenario considered as part of this lab.

(a) Create the network scenario given in Fig. 1 in NS2 by writing the corresponding code in OTcl. Set the link rate, delay and queuing mechanism as given in the Fig. 1. Configure node n0 with a TCP agent and an FTP source and on node n4 configure the corresponding TCP sink.

(b) Next we enable the routing protocol of choice by the following command: `$ns rtproto <args>`, where `args` can be set to `Static or DV or LS` for static, distance vector and link state routing protocols. If `rtproto` command is not specified then NS will run static routing protocol on all the nodes of the topology. Note: One can also specify a list of nodes an which the desired routing protocol has to be run. For more details refer to Chapter 30 of NS2 manual on unicast routing.

(c) Next simulate link failure (at 1.0 second) and restoration (at 4.5 seconds) of the link connecting n4 and n5. This can be accomplished as follows: `$ns rtmodel-at <time> <down (or) up> <from_node_id> <to_node_id>.` Note: one can also simulate node failure and restoration in a similar manner by just specifying the node id in the last argument.

(d) Observe what happens in the above simulation set up for the case of static and distance vector routing protocols. Specifically observe the following in the trace file (ensure that you capture the entire sequence of simulation events in the trace file):

- What happens when there is a link failure or restoration in each case? Does the route change? why or why not?
- In the case of static routing does the TCP transmission recover immediately once the link has been restored? Do you find anything different in the trace file to what you observed in the case of dynamic routing? Explain the reason behind this difference? (to get more intuition behind the exact reason try to plot the TCP window size for the entire simulation for both static and dynamic routing)