CSL374

Computer Networks

Assignment 3

Routing Protocols

Prof. Huzur Saran

Ram Agrawal (2011CS10246)
Shashank Jain (2011CS10254)

We have implemented the following Routing Protocols:

1. Learning Switch Protocol
2. Routing Information Protocol
3. Link State Routing Protocol (Extra Credit)
4. Asynchronous Routing Information Protocol (Extra Credit)

Abstraction:

We created different Switch Types, which was inherited from the simple Hub each implementing/overriding a different function for handle_rx().

Learning Switch Protocol

A class called LearningSwitch is implemented in learning_switch.py

In this protocol, we maintain a mapping of destination and port to which we should forward it in case a packet comes for this destination.

When a packet comes from a new source we update this mapping such that this source can be reached through the port through which the packet came.

After the discovery of all links and the routing update transfers, we will have the complete mapping, in the sense that each destination that can be reached will be available in our mapping.

This protocol is not robust and will fail if we have a network loops or failure in some linkage.

Routing Information Protocol

A class called RIPRouter is implemented in rip_router.py. In this protocol, we determine the port to be forwarded to for a destination according to the path with minimum hops.

At each router we maintain triplets of the destinations and the minimum number of hops through which it can be reached and the port to which a packet with this destination should be forwarded to in the case of minimum hops. In case of same number of hops, the minimum hop path is decided by the one for which port number is minimum.

This protocol is robust. We have implemented two different variations for this protocol – synchronous and asynchronous. The synchronous model sends the routing updates periodically after every 5 seconds. The Asynchronous model is more efficient but it is implementing a slightly modified RIP Protocol which is thus not passed by the tests.

Link State Routing Protocol

A class called LinkState is implemented in link_state.py. In this protocol, we store the entire network at each router in terms of the edges present in the network. Each router calculates the best path for each destination using Dijkstras Algorithm and stores the port to be forwarded to for this destination.

Whenever a link is up or down, a routing update stating whether the link is up or down and the two nodes are sent to all the switches. Each switch updates its network graph and updates its best path calculation. This protocol is asynchronous and totally robust.

Scenarios

We created several scenarios and tested each scenario on all the protocols. Apart from LearningSwitch which fails in case of network loops, all the implementations work as intended.

We have submitted some of our personally created scenarios which help in checking the robustness of the implementation.

Scenario/Test1.py

Scenario/Test12py

Problems Faced:

1. Deciding the routing update interval in case of RIPRouter. Finally, set to 5 seconds.

2. Understanding the test cases properly and their requirements. In case of asynchrounous RIPRouter, these test cases cannot be passed.

3. Building own topologies to check the complete robustness of the implemented router protocols.

4. Implemenation of Dijkstra's algorithm in case of LinkStateRouter efficiently.

Time Comparison between RIPRouter and LSRouter:

LSRouter converge more quickly than RIP router because:

1. In LSRouter, as soon as there is a link down, the information is flooded across the network and each router then update its graph, and run Djikstra again to compute the new Forwarding Table; while in case of RIP, the information that the link has gone down is flooded instantaneously, but the new stable routes are calculated only after the next periodic advertisements.

2. Also in case of LSRouter, only the link changes are flooded while in case of RIPRouter, the whole forwarding tables have to be flooded; due to this RIPRouter takes more time. Because of this reason, the performance of LSRouter is better

than our implementation of asynchronous RIPRouter as well.