Implementation of Spanning Tree Protocol using ns-3

Naishil Shah (nanshah@ucsc.edu)[1] and Sneha Das (sndas@ucsc.edu)[2]

***Abstract*— The goal of this project is to implement the Spanning Tree Protocol (STP) introduced by Radia Perlman. We use a discrete event network simulator NS3 primarily used in research and teaching for the implementation of this project.**

I. INTRODUCTION

A routing loop is a serious network problem which happens when a data packet is continually routed through the same routers over and over. The data packets continue to be routed within the network in an endless circle. A routing loop can have a catastrophic impact on a network, and in some cases, completely disabling the network.A major portion of the precious bandwidth which is available for normal user traffic of the affected routers will be consumed by looping IP datagram packets. Also a major portion of the processing power of the affected routers is used to process the looping IP datagram packets.

Routing Loops can happen in large internetworks when a second topology change emerges before the network is able to converge on the first change. Convergence is the term used to describe the condition when all routers in an internetwork have agreed on a common topology. Link state protocols tend to converge very quickly, while distance vector protocols tend to converge slowly.Different methods are used to avoid routing loops such as using a maximum hop count, split horizon, route poisoning and hold-down timers. Layer 2 switching loops can cause serious problem to network communication. One of the basic functions of a network switch is to eliminate Layer 2 switching loops.

The Spanning Tree Protocol (STP) is a network protocol that builds a logical loop-free topology for Ethernet networks. The basic function of STP is to prevent bridge loops and the broadcast radiation that results from them. Spanning tree also allows a network design to include redundant links to provide automatic backup paths if an active link fails. This is done without the danger of bridge loops, or the need for manual enabling or disabling of these backup links.STP creates a spanning tree within a network of connected layer-2 bridges, and disables those links that are not part of the spanning tree, leaving a single active path between any two network nodes.

Loop prevention in a Layer 2 Ethernet is handled differently than it is in Layer 3. A Layer 3 IP header has a time-to-live (TTL) field that is set by the original host and is decremented at each router, allowing the router to prevent looped datagrams by discarding packets that reach TTL=0. This feature is not available for Layer 2 Ethernet frames.Therefore, after a Layer 2 frame starts to loop in the network, it continues looping forever until one of these events occurs:

- A link is broken.
- One of the bridges is turned off.
- A bridge is rebooted.

Bridging tables that are not configured with STP can be corrupted by the looping created by a broadcast storm.

II. SPANNING TREE PROTOCOL OPERATION

The Spanning tree protocol calculates the best path through a switched network that contains redundant paths. It allows bridges to communicate with each other for the purpose of discovering physical loops in the network. When physical loops are found, the protocol specifies an algorithm that bridges can use to create a loop-free logical topology. Blocking loops prevents broadcast storms in the network. The STP algorithm computes a tree structure of loop-free leaves and branches that spans the entire Layer 2 network.

A. Rules to compute spanning tree

The spanning tree that the bridges compute using the Spanning Tree Protocol can be determined using the following rules.

- Select a root bridge: The root bridge of the spanning tree is the bridge with the smallest (lowest) bridge ID. Each bridge has a configurable priority number and a MAC address; the bridge ID is the concatenation of the bridge priority and the MAC address.The switch with the lowest priority of all the switches will be the root. If there is a tie, then the switch with the lowest priority and lowest MAC address will be the root.
- Determine the least cost paths to the root bridge: The computed spanning tree has the property that messages from any connected device to the root bridge traverse a least cost path. The cost of traversing a path is the sum of the costs of the segments on the path.
- Disable all other root paths: Any active port that is not a root port or a designated port is a blocked port (BP).

In the case of ties, they can be broken by applying one of the following methods:

- By breaking ties for root ports: When multiple paths from a bridge are least-cost paths, the chosen path uses the neighbor bridge with the lower bridge ID.
- When the root bridge has more than one port on a single LAN segment, the bridge ID is effectively tied, as are

all root path costs. The designated port then becomes the port on that LAN segment with the lowest port ID.

III. NS-3 NETWORK SIMULATOR

In communication and computer network research, network simulation is a technique where a program models the behavior of a network either by calculating the interaction between the different network entities (hosts/packets etc.) using mathematical formulas, or actually capturing and playing back observations from a production network. The behavior of the network and the various applications and services it supports can then be observed in a test lab; various attributes of the environment can also be modified in a controlled manner to assess how the network would behave under different conditions.

Among the many network simulators present today, we use ns-3 as the network simulator for this project. It is a discrete-event network simulator, targeted primarily for research and educational use. It helps develop a preferred, open simulation environment for networking research which is aligned with the simulation needs of modern networking research. The ns-3 simulation core supports research on both IP and non-IP based networks. A large majority of its users however focuses on wireless/IP simulations which involve models for Wi-Fi, WiMAX, or LTE for layers 1 and 2 and a variety of static or dynamic routing protocols such as OLSR and AODV for IP-based applications.

IV. METHODOLOGY ADOPTED

A. Approach A

We tried an approach to exchange HELLO messages between all the nodes to implement the spanning tree protocol. The idea was to start by electing the node with the lowest MAC id as the root node. This root node would check its immediate neighbors by exchanging HELLO messages with them. Each node in the topology has an array of its previously unvisited neighbors. For every node, it lists neighbors that are in lower priority. As we traverse through higher MAC ids, this approach makes sure that previously visited neighbors are not added into the nodes array. For each neighbor added to the nodes array, we assign the respective port to be the root port since it will be the least cost path taken from the root node. In this process, we block the remaining unused ports, thereby removing any loops present in the topology.

This distributed mechanism of implementing the protocol was unsuccessful during the course of the project due to two reasons. We could not successfully implement detection of neighboring nodes using HELLO messages. Once this problem was resolved, we could not figure out how to assign root ports and designated ports based on this information. Without this knowledge, we could not proceed with assigning blocked ports and hence removal of loops was a problem. Hence we considered another approach which is discussed in the next section.

B. Approach 2

The in built examples of network simulator 3 were used to obtain the desired result. We used the program of matrix-topology.cc given in the tutorials folder of ns3 and modified it to given us the required result. We understood that this file gave an adjacency matrix as an output to any given network topology. We used Prim's Algorithm on this adjacency matrix to obtain a graphical spanning tree of the network. The final output we obtanined after applying Prim's Algorithm, was again converted to a topology by the mst.cc file. To demonstrate the changes made before and after the application of the algorithm, we depict the topology using NetAnim. NetAnim is an offline animator based on the Qt toolkit to view the output obtained. It animates the simulation using an XML trace file collected during simulation.As shown in the images that follow, we tested it for various topologies and this approcah worked perfectly well for any given topology.

The Prim's Algorithm's Pseudo code is explained below.

Algorithm 1 Pseudo code

$ReachSet = \{0\}$ ▷ You can use any node
$UnReachSet = \{1, 2, ...N - 1\}$
$SpanningTree = \{\}$
while $UnReachSet \neq empty$ **do** ▷ Find edge e= (x,y) such that:
    $x \in ReachSet$
    $y \in UnReachSet$
    $e(smallestcost)$
$SpanningTree = SpanningTree \cup \{e\}$ ▷ Add edge to Spanning tree
$ReachSet = ReachSet \cup \{y\}$ ▷ Add y to visited set
$UnReachSet = UnReachSet\{y\}$ ▷ Remove y from unreached set

Consider the 6-node topology as shown in 1. We use the following helper classes available in ns-3 as given below. The definition of these classes have been taken from the official documentation of ns-3.

- PointToPointHelper :
  It builds a set of PointToPointNetDevice objects.
- NodeContainer :
  It keeps track of a set of node pointers.Typically ns-3 helpers operate on more than one node at a time. For example a device helper may want to install devices on a large number of similar nodes. The helper Install methods usually take a NodeContainer as a parameter. NodeContainers hold the multiple Ptr<Node> which are used to refer to the nodes.
- InternetStackHelper :
  It aggregates IP/TCP/UDP functionality to existing Nodes. This helper enables pcap and ascii tracing of events in the internet stack associated with a node.
- Ipv4AddressHelper :
  A helper class used to make life easier while doing

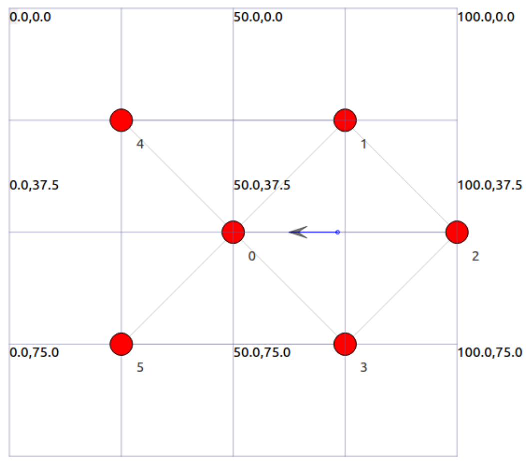

Fig. 1. Input 6-node topology using NetAnim

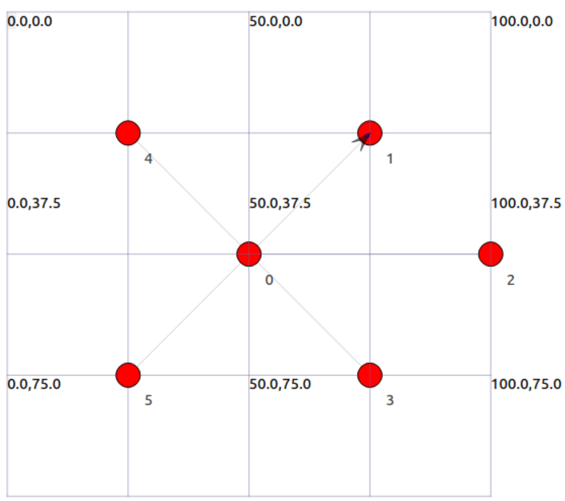

Fig. 2. Minimum Spanning Tree obtained for a 6-node topology using NetAnim

simple IPv4 address assignment in scripts. This class is a very simple IPv4 address generator. You can think of it as a simple local number incrementer.

- MobilityHelper :
  A Helper class used to assign positions and mobility models to nodes. MobilityHelper::Install is the most important method here.
- PacketSinkHelper :
  A helper used to make it easier to instantiate an ns3::PacketSinkApplication on a set of nodes.

The routing tables for the generated nodes are as shown in the images below in 3 and 4:

V. CONCLUSIONS

The algorithm presented here maintains a spanning acyclic subset of a general mesh topology. It requires a very small,bounded amount of memory per node,independent of the total number of nodes. This algorithm can prevent one-way links from causing loops.The formed graph is a tree. It has a unique root and each node other than the root has a unique predecessor closer to the root.Also, given a particular physical topology, behavior of the algorithm is completely predictable.

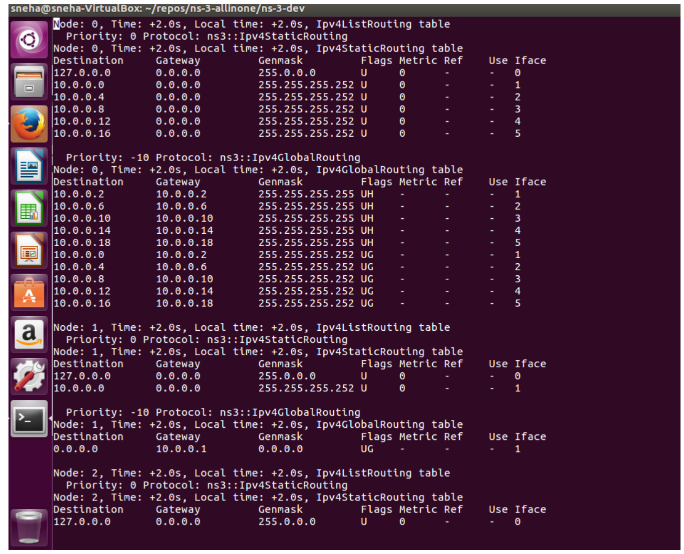

Fig. 3. Routing Table for Minimum Spanning Tree obtained for a 6-node topology- Nodes 0-2

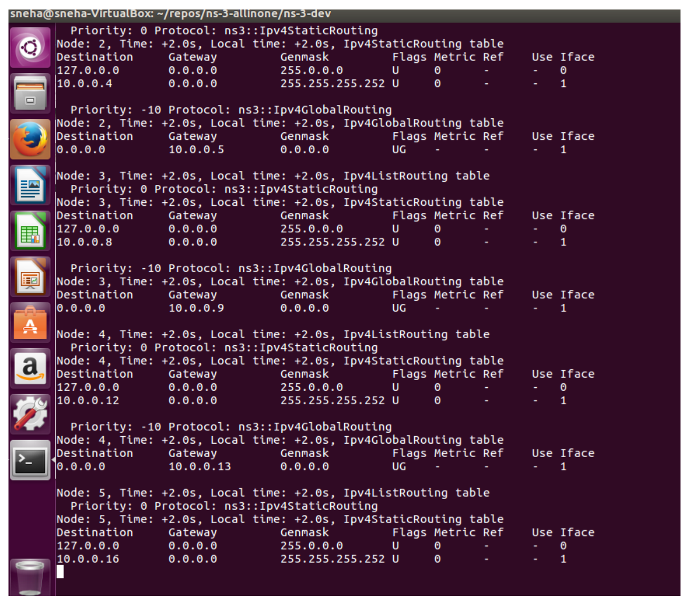

Fig. 4. Routing Table for Minimum Spanning Tree obtained for a 6-node topology- Nodes 2-5

ACKNOWLEDGMENT

We would like to thank Professor J.J. Garcia Luna Aceves one of the most enjoyable classes we have taken that most practically demonstrated how computer networks work, and for all the math. We also acknowledge our Teaching Assistant Ali Dabirmoghaddam and our classmates for all the suggestions and help on working out the minor technical difficulties faced while trying to run the code.

REFERENCES

[1] Perlman, Radia. "An algorithm for distributed computation of a spanningtree in an extended LAN." ACM SIGCOMM Computer Communication Review. Vol. 15. No. 4. ACM, 1985.

[2] Perlman, Radia. "A protocol for distributed computation of a spanning tree in an extended lan." Ninth Data Communications Symposium. 1985.

[3] Marchese, Mario, and Maurizio Mongelli. "Simple protocol enhancements of Rapid Spanning Tree Protocol over ring topologies." Computer Networks 56.4 (2012): 1131-1151.

[4] "TechLibrary." Spanning Tree Protocol Operation - Technical Documentation - Support - Juniper Networks. N.p., n.d. Web. Dec. 2016.

[5] "Ns-3." Ns3 RSS. N.p., n.d. Web.2016.

[6] "Spanning Tree Protocol STP Tutorial." CCNA Training Âż Spanning Tree Protocol STP Tutorial. N.p., n.d. Web. Dec. 2016.

[7] Perlman, Radia (2000). Interconnections, Second Edition. USA: Addison-Wesley. ISBN 0-201-63448-1.

[8] Attacking the Spanning Tree Protocol." Cisco Press. N.p., 4 June 2008. Web. 2 Mar. 2016<http://www.ciscopress.com/articles/article.asp?p=1016582>