

Assignment 10

Problem Statement

Use network simulator NS2 to implement

1. Monitoring traffic for given topology
2. Analysis of CSMA and Ethernet protocols
3. Network Routing: Shortest path routing, AODV
4. Analysis of congestion control (TCP and UDP)

Objectives:

1. Write TCL code to simulate traffic monitoring, Ethernet protocols, AODV and congestion control.
2. Capture the NAM output of the same

Learning outcomes: Students will be able to

1. Write code to simulate traffic monitoring, Ethernet protocols, AODV and congestion control
2. Visualize the same using NS2

Requirements:

1. Open source linux based OS
2. NS2 Network simulator

Theory:

- a. Monitoring traffic for given topology

Algorithm

1. Create a simulator object
2. Tell the simulator to use dynamic routing
3. Open the nam trace file
4. Define a 'finish' procedure
5. Create seven nodes

6. Create a link between nodes
7. Create a UDP agent and attach it to node $n(0)$
8. Create a CBR traffic source and sink
9. Schedule events for CBR agent and network dynamics
10. Run the simulator

Simulation workflow

The general process of creating a simulation can be divided into several steps

1. Topology definition

To ease the creation of basic facilities and define their interrelationships, NS-2 has a system of containers and helpers that facilitate this process.

2. Model development

Models are added to simulation, like UDP, IPv4 point-to-point devices and links applications. Most of the time this is done using helpers.

3. Node and link configuration

Models set their default values most of the time. This is done using attribute system.

4. Execution

Simulation facilitates generate events, data requested by user is logged.

5. Performance analysis:

After simulation is finished and data is available as a time-stamped event trace, this data can then be statistically analyzed with tools like R to draw conclusion.

6. Graphical visualization

Raw or processed data collected in a simulation can be graphed using tools like Graplot, Matplotlib or XGraph.

Ad-hoc On-Demand Distance Vector (AODV)

1. AODV routing protocol enables multi-hop routing between participating mobile nodes to establish and maintain an ad-hoc network.
2. AODV is based upon the distance vector algorithm. The difference is that AODV is reactive, as opposed to proactive protocols like DV.
3. Features of the protocol include loop freedom and that link breakage causes immediate notifications to be sent to the affected set of nodes.

Route table management:

Destination IP Address: IP address for destination node. Destination sequence number for this destination.

Hop Count: No of hops to destination

Next Hop: The neighbour, which has been designated to forward packets to destination for this route entry.

Lifetime: The time for which the route is considered valid.

Active Neighbour list: Neighbour nodes that are actively using this route entry.

Request buffer: Makes sure that a request is only processes once.

Conclusion

We successfully made use of network simulator NS-2 and implemented the given tasks and understood the underlying algorithms.