Miriam Avila 11/27/2023

Simple Simluation

Group project

This report presents a comparative analysis of CSMA/CD (Carrier Sense Multiple Access with Collision Detection) and (Carrier Sense Multiple Access with Collision Avoidance) 802.11 protocols in wired and wireless network environments. I used *simple-w.tcl* for a wired network with five nodes, and used wireless.tcl for a wireless network with twenty nodes to simulate both protocols using the Network Simulator 2 (NS2). The aim of the study was to study and evaluate the effectiveness and efficiency of various protocols in varying network loads and circumstances. Below shows the simulation setup.

**Simulation Setup**:

Tools Used: NS2 Simulator, Oracle VirtualBox

**Wired Network Simulation** (CSMA/CD):

Script: *simple-w.tcl*

Number of Nodes: 5

Traffic Type: Constant Bit Rate (CBR) over UDP

**Wireless Network Simulation** :

Script: *wireless.tcl*

Number of Nodes: 20

Traffic Type: CBR over UDP

**Parameters**:

Frame Size: 1,000 Bytes

Simulation Duration: 100 seconds

The *wireless.tcl* script was used to simulate an protocol-based environment for the wireless network. Twenty nodes that were set up to simulate a wireless ad hoc network were used. Here we can examine that the primary node sends data to node 19. In the files, I observed that my code created the nam file. After multiple changes of my code, this is the final product that worked for me. I used Excel because gnu plot did not work. The line graph below demonstrates a visual representation of how throughput varies over time in your simulation. ​

A graph of a line graph

Description automatically generated

​

**Network Model**

**Number of Nodes**: 20 mobile nodes

**Node Configuration**:

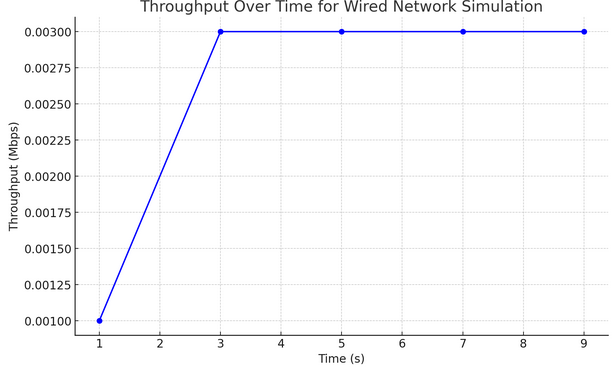
Color Coding and Mobility: The distinct colors of each node made tracking them in NAM file simple, and their positions and movements within the simulation region offers information about how well the protocol performed in a dynamic wireless environment.

A computer screen shot of a computer

Description automatically generated

The script used twenty nodes, each placed in an area measuring 1000 by 1000 meters, to create a distributed wireless network. A combination of TCP and UDP protocols was used to create network traffic, representing a range of actual network usage scenarios. To be more precise, Node 19 was set up as a TCPsink, which acted as the traffic's destination, and Node 0 was configured as a TCP agent, acting as the source of TCP traffic. The FTP application was connected to the TCP agent to simulate network file transfers. Including, many nodes were set up with UDP agents, resulting in Constant Bit Rate traffic from each node.

Next, I used the *simple-w.tcl* script to simulate a network of five nodes using the CSMA/CD protocol in the wired network section. Evaluating the network's capacity to handle packets and transfer data under controlled load was the goal. The graph below shows that across the observed period, throughput in the wired network simulation remained persistently low at 0.003 Mbps, with just an initial improvement from 0.001 Mbps.



**Network Model:**

**Number of Nodes**: 5

Link Type: Duplex Links with 2 Mb bandwidth and 10 ms delay, using DropTail queue.

**Node Configuration**:

Nodes were assigned individual colors for visualization in the NAM (Network Animator) tool: Pink, Red, Black, Purple, and Blue. A computer screen shot of a network

Description automatically generated

The traffic in the network was created by the agents. Connecting Constant Bit Rate traffic sources to every one of these UDP agents to mimic a constant flow of traffic. The CBR sources were configured to transmit 1000-byte packets, which enable substantial data transfer without exceeding the network's capacity. Staggered traffic was used in the simulation design, with CBR agents starting the simulation at 1.0, 1.5, and 2.5 seconds in advance. Rather than having every node shown, this method was designed to mimic an environment where network traffic varies over time. I was able to watch how the CSMA/CD protocol handles possible collisions and concurrent transmissions because of the traffic demand.

In conclusion, Important insights into the CSMA/CD and the protocols were gained from the simulations conducted in NS2 using *simple-w.tcl* and *wireless.tcl*. The wireless configuration showcased the adaptability of the protocol in a dynamic context, while the wired network effectively showed the CSMA/CD protocol's capacity to handle data traffic. This analysis provides insights for practical applications and shows the significance of various protocols in network design. The report shows how useful NS2 is as a tool for deciphering and checking network protocols.