LINUX NETWORK PACKET STATISTICS

Contents

Overview

The Network Packet Statistics Display Project is a software application designed to capture and analyze network packet statistics in real-time. It provides users with insights into the traffic on their network by displaying statistics such as packet count and packet sizes categorized by different protocols (TCP, UDP, ICMP). The application offers two display formats: tabular and graphical, allowing users to choose their preferred visualization method.

.

Objectives

- Provide real-time visualization of network packet statistics.
- Display packet counts and sizes for TCP, UDP, and ICMP protocols.
- Display format (tabular or graphical)
- Utilize shared memory for inter-process communication.
- Implement multithreading for concurrent packet capture and UI display.

Project Scope

The project focuses on capturing and displaying network packet statistics within a local environment. It does not involve packet sniffing or network analysis beyond the generated random data. The application is designed to run on Unix-like operating systems due to the utilization of POSIX threads and shared memory.

Code Functionality

- Packet Generation: Simulates packet capture by generating random data for TCP, UDP, and ICMP packets.
- Shared Memory Usage: Utilizes shared memory for communication between the packet capture thread and the UI thread.
- Multithreading: Implements two threads - one for packet capture and analysis, and another for UI display.
- Display Formats: Supports two display formats - tabular and graphical.
- User Options: Allows users to specify display format and choose which protocols to display statistics for.

Mechanisms Used

- POSIX Threads (pthread): Used for implementing multithreading to capture packets and display statistics concurrently.

- Shared Memory (shmget, shmat): Employed for inter-process communication between the packet capture thread and the UI display thread.
- Random Data Generation (rand, srand): Generates random packet data to simulate packet capture
- Command-line Arguments: Allows users to specify display format and protocols to display statistics for.

Implementation

- The project is implemented in C programming language, utilizes POSIX threads, and shared memory mechanisms. Here is a brief overview of the implementation:
- Main Function: Initializes threads for packet capture and UI display, and waits for their completion.
- Packet Capture Thread: Continuously generates random packet data for TCP, UDP, and ICMP protocols and updates shared memory with the statistics.
- UI Display Thread: Periodically reads packet statistics from shared memory and displays them to the user based on the chosen display format.
- Packet Statistics Structure: Defines a structure to hold packet statistics for TCP, UDP, and ICMP protocols, including packet counts and sizes.
- Display Functions: Implements functions to display packet statistics in tabular and graphical formats.
- Argument Parsing: Parses command-line arguments to determine the display format and protocols to display.

Conclusion

This documentation provides a comprehensive overview of the "Network Packet Statistics Display Project," covering its objectives, implementation details, and mechanisms used for functionality. It serves as a guide for developers looking to understand, modify, or extend the project.