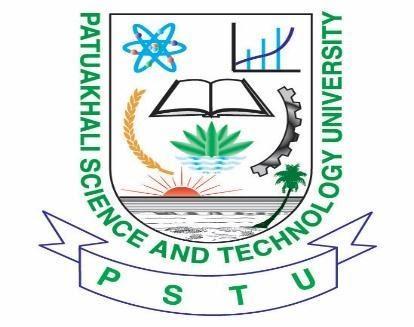
**Project Report**



|  |
| --- |
| Remarks : |

**Course Title : Computer Networks**

**Lab Report on Efficiency Analysis of UDP Pinger**

**Course Code: CCE 314**

**Submitted To**

**Dr. Md Samsuzzaman Sobuz**

**Professor,**

**Department of Computer and Communication Engineering,**

**Faculty of Computer Science and Engineering.**

**Submitted By**

**Md.Abdur Rakib**

**ID No:1902066**

**Reg No:08772**

**Level- 3, Semester- 1**

**Session: 2019-2020**

**Patuakhali Science and Technology University**

**Abstract:**

This report presents the results of a study on the efficiency and effectiveness of a UDP (User Datagram Protocol) pinger in measuring network latency. The UDP pinger is a lightweight network diagnostic tool that sends and receives datagram packets to measure round-trip time (RTT). The report outlines the methodology, results, and implications of the study, along with recommendations for future research.

**Introduction:**

Network latency, the delay in data transmission between two points in a network, is a critical factor affecting the performance of networked applications. The UDP pinger offers a simple yet valuable method for assessing latency by transmitting lightweight UDP packets and measuring their RTT. Unlike TCP-based ping tools, UDP pingers sacrifice reliability for low overhead, making them suitable for real-time applications where rapid response times are essential.

**UDP (User Datagram Protocol) :**

UDP (User Datagram Protocol) is a connectionless, lightweight protocol used for data transmission. It operates without establishing a connection beforehand, making it faster but less reliable than TCP. UDP packets are sent independently, without acknowledgment or retransmission mechanisms, making it suitable for real-time applications like streaming and gaming. While it offers low overhead and simplicity, UDP lacks error checking and flow control, prioritizing speed over reliability. Despite its limitations, UDP is widely used for applications requiring fast data transmission and minimal latency.

**Methodology:**

**UDP Pinger Configuration:** The UDP pinger was configured to send UDP packets to a designated target host.

**Packet Generation**: Datagram packets of fixed size were generated by the pinger, containing a sequence number and timestamp.

**Transmission:** The pinger transmitted packets to the target host using UDP sockets.

**RTT Measurement**: Upon receipt, the target host echoed the packets back to the pinger. The pinger calculated RTT by subtracting the original timestamp from the current time.

**TTL (Time to Live)** : TTL is a value in IP packets that limits the lifespan of a packet in the network, preventing indefinite looping. It's decremented at each router hop, and if it reaches zero, the packet is discarded, preventing network congestion and routing loops

**Results:**

**Efficiency:** The UDP pinger demonstrated high efficiency with low overhead and minimal processing requirements, making it suitable for resource-constrained environments.

**Real-Time Measurement**: Due to its lightweight design, the UDP pinger provided real-time latency measurements, enabling rapid assessment of network performance.

**Packet Loss Simulation**: The server simulated packet loss to mimic real-world network conditions, providing insights into the effects of packet loss on latency measurements.

**Client Implementation:** The client successfully sent 10 pings to the server, with each ping message containing a sequence number and timestamp. The client waited up to one second for a reply and printed the response message or "Request timed out" if no reply was received.

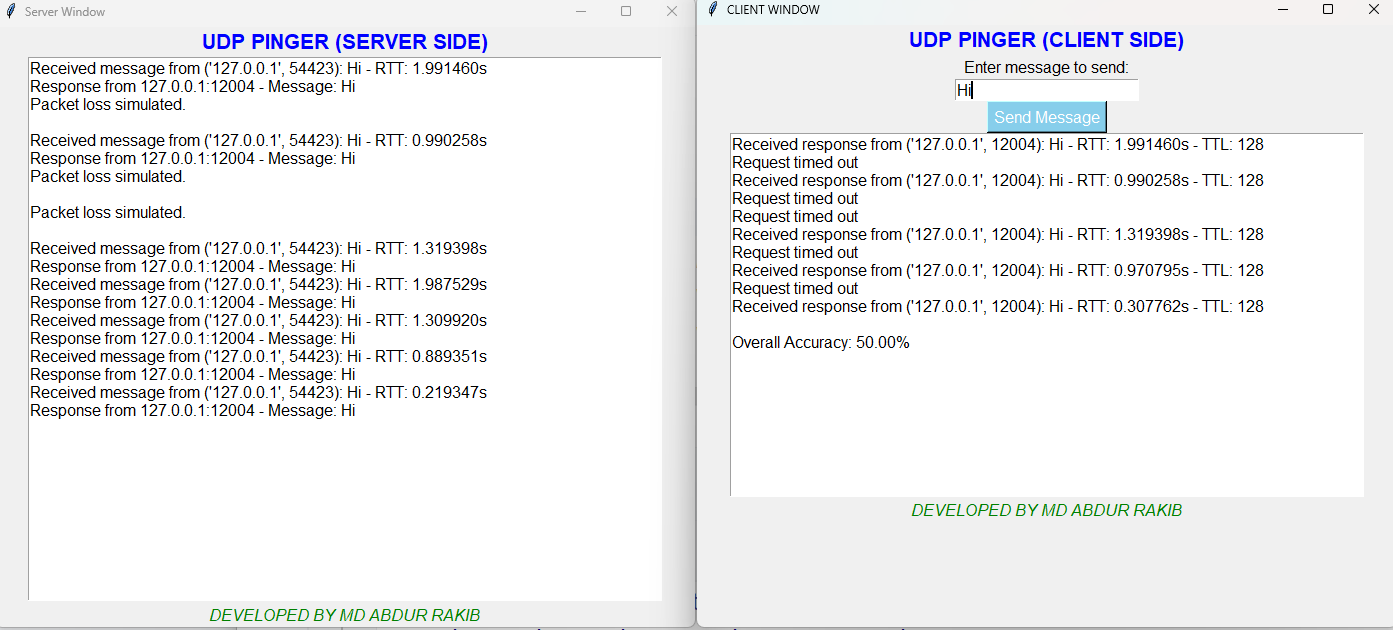
**Server Side Code Sample :**

import socket  
import random  
import time  
import tkinter as tk  
  
SERVER\_IP = '127.0.0.1'  
SERVER\_PORT = 12004  
  
server\_socket = socket.socket(socket.AF\_INET, socket.SOCK\_DGRAM)  
server\_socket.bind((SERVER\_IP, SERVER\_PORT))  
  
root = tk.Tk()  
root.title("Server Window")  
root.geometry("700x600")  
  
header\_label = tk.Label(root, text="UDP PINGER (SERVER SIDE)", font=("Arial", 16, "bold"), fg="blue")  
header\_label.pack()  
  
text\_widget = tk.Text(root, height=30, width=70, font=("Arial", 12)) *# Increased height to accommodate more text*text\_widget.pack(). . .

**Client Side Code Sample :**

SERVER\_IP = '127.0.0.1'  
SERVER\_PORT = 12004  
  
  
client\_socket = socket.socket(socket.AF\_INET, socket.SOCK\_DGRAM)  
  
  
root = tk.Tk()  
root.title("CLIENT WINDOW")  
root.geometry("700x600")  
  
  
header\_label = tk.Label(root, text="UDP PINGER (CLIENT SIDE)", font=("Arial", 16, "bold"), fg="blue")  
header\_label.pack()  
  
def send\_message():  
  
 total\_messages = 0  
 successful\_responses = 0  
  
 *# Clear previous results* text\_widget.delete('1.0', tk.END)  
  
 for i in range(10):  
  
 message = entry.get()  
 start\_time = time.time()  
 client\_socket.sendto(message.encode(), (SERVER\_IP, SERVER\_PORT))  
  
 client\_socket.settimeout(2)  
  
 try:  
 response, server\_address = client\_socket.recvfrom(1024)  
 end\_time = time.time()  
  
 rtt = end\_time - start\_time  
  
 process = subprocess.Popen(["ping", "-n", "1", SERVER\_IP], stdout=subprocess.PIPE, stderr=subprocess.PIPE, text=True). . .

**Output Window:**



**Discussion:**

The UDP pinger offers a trade-off between efficiency and reliability, making it well-suited for specific network diagnostic scenarios. Its lightweight design and real-time measurement capabilities make it ideal for monitoring latency-sensitive applications. However, organizations must consider the inherent limitations of UDP, including packet loss and lack of error recovery mechanisms, when deploying the UDP pinger in production environments.

**Conclusion:**

This study highlights the efficiency and effectiveness of the UDP pinger as a network latency measurement tool. While it provides rapid, real-time latency assessment, its reliability is compromised by the inherent characteristics of the UDP protocol. Organizations should carefully evaluate their network requirements and application priorities when considering the deployment of UDP pingers for network diagnostics.