Project Report

NetCQ

Network Packet Capture and Query Tool

Athukorala A.S.P. E/11/024

Karunatissa K.P.D.H.M E/11/204

Piyasinghe W.D.Y.N E/11/305

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Contents

[Abstract 1](#_Toc321140622)

[Introduction 2](#_Toc321140623)

[Literature Review 3](#_Toc321140624)

[Requirement Analysis 4](#_Toc321140625)

[System Design and Architecture 5](#_Toc321140626)

[Methodology 6](#_Toc321140627)

[Implementation Details 6](#_Toc321140628)

[Result and Analysis 6](#_Toc321140628)

[Conclusion & Future Work 6](#_Toc321140628)

# Abstract

Network monitoring is essential in modern day because there are heavy downloaders, Hackers and many other people who are affecting the Network. Not only that but also Network should be monitored to get a description of the general network behavior. For that Network packets contain a lot of information about Network activity. Network packet Capture and Query tools become a useful tool for Network Administrators. In this report, an Implementation of a network packet capture and query tool based on JpCap, a java Library for packet capturing, will be described. This can be used to capture packets which are transferring through a chosen network by user like Local Area Network, Wi-Fi Network and etc. Hash Map was used in this tool to efficiency Store and Query data as user needed. After Capturing packets and Saving, User can Query data as he or she wanted and monitor all the computers connected to the network.

# Introduction

In present, we are living in a networked world where we satisfy most of our needs through it like ordering something (food, electronic item) and also Share resources and communicating each other. Therefore computers are no longer an isolated machine. Network packets are units that move data through inside the network from one computer to one another where they carry all the important information from its source to final destination. Beside the actual data carries inside the packet the packet headers contain useful information about the network infrastructure and much more things like indication of the behavior of the network traffic. For example, the header information used to analyze the computers connected to and find who uses the most of the resources in the network and like identifying network attacks.

Therefore Network Packet Capture and Analysis has applications in Network Monitoring, Fault Diagnosis, Network Security, Intrusion & Threat Detection, Network Forensics and many other areas. This Network Packet Capture and Query tool was implemented to monitor the data traffic of the network like which are the computers connected to this network.

This tool based on commonly used free java library called JpCap. It get the interfaces where computer connected to Networks and Capture in other means monitor the network as user needs. Then these captured data is saved to efficiently created hash map set where user can query data quickly even in a Large data usages. It has a Graphical User Interface where users can easily use this tool and query the data as user needs.

This report served as a starting point for users to get to know about features and advantages of this tool. It describe the general information about the tool and the basic understand about the backend of this tool. Not only that but also use this as an help for anyone interested in Advanced Network monitoring by using this technology and querying the relevant data for each scenario network faces like Hacking, freezing, Overflowing and etc.

# Literature Review

There are few software where some of them are Open Source available in the field of packet capturing

**Wireshark** is a free and open-source packet analyzer. It is used for network troubleshooting, analysis, software and communications protocol development, and education. Wireshark is cross-platform, using the GTK+ widget toolkit in current releases, and Qt in the development version, to implement its user interface, and using pcap to capture packets; it runs on GNU/Linux, OS X, BSD, Solaris, some other Unix-like operating systems, and Microsoft Windows.

There is also a terminal-based (non-GUI) version called **TShark.**

**tcpdump** is a common packet analyzer that runs under the command line. It allows the user to display TCP/IP and other packets being transmitted or received over a network to which the computer is attached. Distributed under the BSD license. tcpdump is free software. It works on most Unix-like operating systems: Linux, Solaris, BSD, OS X, HP-UX and AIX among others. In those systems, tcpdump uses the libpcap library to capture packets. The port of tcpdump for Windows is called WinDump; it uses WinPcap, the Windows port of libpcap.

# Requirement Analysis

The tool can performs following tasks.

* Capture all types of IP or Ethernet packets from the Ethernet interface.
* Extract the relevant information from the data packets. The relevant information may include:
  + Source IP and Port
  + Destination IP and Port
  + Layer 4 protocol
  + Packet Length
* User can choose the needed query from the given list or can create a manual query and extract information he needed.

With comparing with Wireshark the tool has an GUI not like in Wireshark but it can query data by selecting given queries or creating manual queries.

# System Design and Architecture

A general description of the tool is shown in below.

It has a GUI and From that user can select following details.

* Select interface and give number of packets and start capturing.
* Those capturing data are shown in a table and they were saved to hashMap set.
* User can select a given query or go to manual mode.
* From manual mode user can create a own query by a filtering data.

# Methodology

User Interface

* User can extract info from given query set or give manual query.

In Backend

* Extract relevant info from the hashMap.

Extract Info.

User Interface

* Shows the capturing packet header information

In Backend

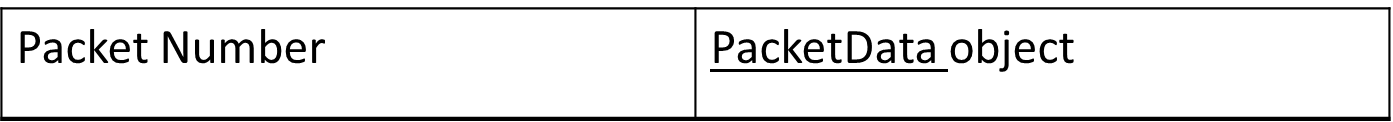
* Save captured packet data info to a Hashmap set.

Capture

User interface

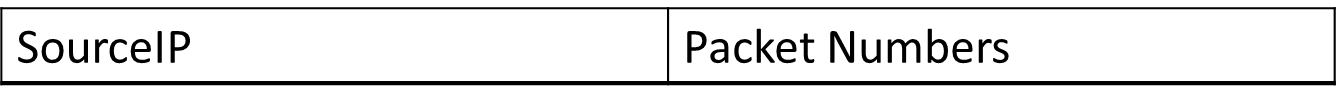
* User can select the Interface he can capture the packet.
* How many packets needs to capture

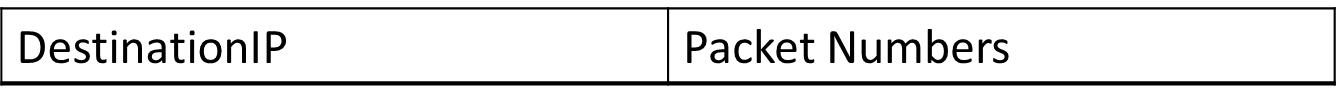
Storage structure is shown below. All packets are numbered as they caputred As packetnumber.

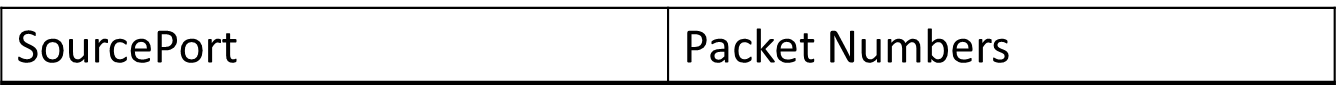
 Hash Map 1(Master one)

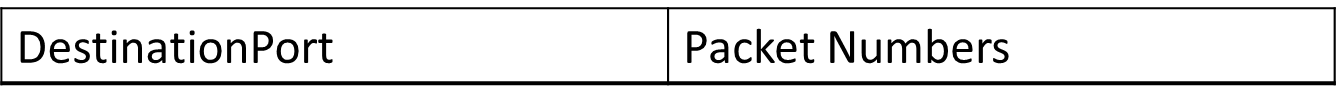
PacketData is an object which has Source IP Address, Destination IP address,

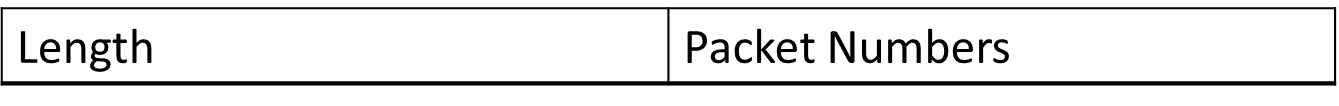
Source Port, Destination Port, packet Length and Layer 4 protocol.

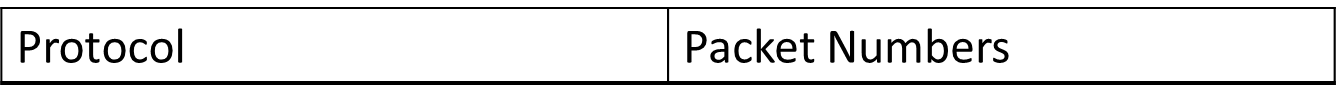
 Hash Map 2

HashMap 3

HashMap 4

HashMap 5

HashMap 6

HashMap 7

When user gives an input as Source IP address, the packet number list is taken from HashMap 2 and Using those packet numbers the info extract from the Master HashMap.

# Implementation Details

Iteration 1- first two weeks

* Capturing part
* Storing those header information
* Some queries for extract data.

Iteration 2 - Next two weeks

* User interface
* Offline storing
* More queries and manual query

# Results and Analysis

# Conclusion & Future Work

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

# References

|  |  |  |  |
| --- | --- | --- | --- |
| <http://en.wikipedia.org/wiki/Packet_analyzer>  <http://searchnetworking.techtarget.com/definition/network-analyzer>  <http://jpcap.sourceforge.net/>  <http://jpcap.sourceforge.net/javadoc/>  <http://en.wikipedia.org/wiki/Wireshark>  http://en.wikipedia.org/wiki/Tcpdump |  |  |  |
|  |  |  |  |
|  |  |  |  |