Software Requirements Specification

for

Packet Sniffer

Version 1.0 approved

Sunil Baliganahalli NarayanMurthy
Nehal Kamat
Apoorva Bapat

University of Colorado, Boulder

Feb 17, 2016

Table of Contents

Revision History

Name	Date	Reason For Changes	Version
Sunil Baliganahalli Narayana Murthy	2/17/2016	Initial draft	1.0
Sunil Baliganahalli Narayana Murthy	2/21/2016	Incorporated review comments from teammates	1.1
Sunil Baliganahalli Narayana Murthy	3/4/2016	Incorporated review comments from teammates	1.2
Sunil Baliganahalli Narayana Murthy	3/7/2016	Included Activity & Sequence diagrams	1.3

1. Introduction

1.1 Purpose

Packet sniffing is defined as a technique that is used to monitor every packet that crosses the network. A packet sniffer is a piece of hardware or software that monitors all network traffic. Using the information captured by the packet sniffers an administrator can identify erroneous packets and use the data to pinpoint bottlenecks and help to maintain efficient network data transmission. For most organizations packet sniffer is largely an internal threat.

Packet sniffers can be operated in both switched and non-switched environment. Determination of packet sniffing in a non-switched environment is technologies that can be understand by everyone. In this technology all hosts are connected to a hub. There are a large number of commercial and non-commercial tools are available that makes possible eavesdropping of network traffic. Now a problem comes that how this network traffic can be eavesdrop; this problem can be solved by setting network card into a special “promiscuous mode”. Now businesses are updating their network infrastructure, replacing aging hubs with new switches. The replacement of hub with new switches that makes switched environment is widely used because “it increases security”. However, the thinking behind is somewhat flawed. It cannot be said that packet sniffing is not possible in switched environment. It is also possible in switched environment.

1.2 Intended Audience and Reading Suggestions

This document is intended for User, Developer and tester.

1.3 Product Scope

<Provide a short description of the software being specified and its purpose, including relevant benefits, objectives, and goals. Relate the software to corporate goals or business strategies. If a separate vision and scope document is available, refer to it rather than duplicating its contents here.>

2. System Features

Business Requirements - [Not Applicable]

User Requirements				
ID	**Requirements**	**Topic Area**	**User**	**Priority**
UR-001	Users should have the option of choosing the client machine to monitor packets from	Freedom	Any	High
UR-002	Users should be able to deploy the application on any operating system/work environment	Deployment	Any	High
UR-003	Users should have the option to run the application either using a graphical interface or via the command	Interaction	Any	Medium
UR-004	Users should be able to extract required information and save it	Logging	Any	High

Functional Requirements				
ID	**Requirements**	**Topic Area**	**User**	**Priority**
FR-001	The user shall we be able to select the client for which he wants to monitor the network traffic.		User	High
FR-002	The user shall be able to capture live packet data from a selected network interface.		User	High
FR-003	The user shall be able to save the captured packets or discard.		User	Low
FR-004	The user shall be able to filter the packets like filter all TCP, ICMP etc.		User	Medium
FR-005	The user shall be able to open the saved packets for analysis.		User	Medium
FR-006	The user shall be import/export the saved packets.		User	Medium

FR-007	The user shall be able to look at the header data or packet data of the captured packet.		User	High
FR-008	The user shall be able to stop the capturing of the packets.		User	Medium
FR-009	The user shall be able to see the basic stats about the monitored client like # of TCP packets captured, # of UDP packets captured, etc.		User	Low
FR-010	The user shall be able to search for packets on many criteria		User	Low
FR-011	Colorize packet display based on filters.		User	Low
FR-012				

Non-Functional Requirements

ID	**Requirements**	**Topic Area**	**User**	**Priority**
NF001	Sufficient network bandwidth			High
NF002	The application should be reliable			High
NF003	Application should be robust and handle at-least 5 clients			High
NF004	Application should be responsive			High
NF005	Application should have a reasonable performance (1sec)			Medium
NF006				

Use case documents:

Use Case ID:	UC-001
Use Case Name:	Open Graphical User Interface
Description:	Select application icon on desktop/ in the start menu to open a graphical interface for running the application

Actors:	Any		
Pre-conditions	User should choose to use graphical interface to application in place of command line access to application		
Post conditions	User should understand the layout of the interface and should understand how the information is being displayed		
Frequency of Use:	User might use the GUI as primary interaction with application		
Flow of Events:		Actor Action	System Response
	1	Double-click application shortcut on desktop	Application GUI opens
	2	Click application entry in all programs menu	Application GUI opens

Use Case ID:	UC-002
Use Case Name:	Open Command Line Interface
Description:	Display the network statistics on the command line instead of a graphical interface

Actors:	Advanced Users		
Pre conditions	User should choose to use the command line interface to application in place of a graphical interface		
Post conditions	Users should know basic command prompt commands to understand how to navigate and run the application from the command line		
Frequency of Use:	Not as frequent as GUI, but equally important		
Flow of Events:		Actor Action	System Response
	1	Open command prompt	Command prompt displayed
	2	Type in application name and press enter	Text version of application is displayed on prompt
	3	Type in commands to access different functionality of the application	Appropriate command is executed and corresponding information is shown

Use Case ID:	UC-003
Use Case Name:	Monitor Packets
Description:	Allows the user to be displayed the packets being transmitted in real time

<table>
<tr><td><b>Actors:</b></td><td colspan="3">All users</td></tr>
<tr><td><b>Pre conditions</b></td><td colspan="3">Users should have opened either the graphical interface or the command line interface</td></tr>
<tr><td><b>Post conditions</b></td><td colspan="3">Users should have basic knowledge of packet formats and should be able to read them</td></tr>
<tr><td><b>Frequency of Use:</b></td><td colspan="3">Frequently</td></tr>
<tr><td rowspan="3"><b>Flow of Events:</b></td><td></td><td>Actor Action</td><td>System Response</td></tr>
<tr><td>1</td><td>Open application</td><td>Application user interface is displayed</td></tr>
<tr><td>2</td><td>Click ‘monitor’</td><td>Transmitted packet details are displayed on the UI</td></tr>
</table>

Use Case ID:	UC-004
Use Case Name:	Save Packet Information
Description:	Enables the user to store packet information for offline analysis

<table>
<tr><td><b>Actors:</b></td><td colspan="3">All users</td></tr>
<tr><td><b>Pre conditions</b></td><td colspan="3">Application should be running and packets being monitored</td></tr>
<tr><td><b>Post conditions</b></td><td colspan="3">A log file should have been created with the required information saved in it</td></tr>
<tr><td><b>Frequency of Use:</b></td><td colspan="3">Very frequent</td></tr>
<tr><td rowspan="4"><b>Flow of Events:</b></td><td></td><td>Actor Action</td><td>System Response</td></tr>
<tr><td>1</td><td>Start application</td><td>Application interface displayed to user</td></tr>
<tr><td>2</td><td>Click monitor</td><td>Packets start being monitored and their information displayed on the interface</td></tr>
<tr><td>3</td><td>Select packet information to be saved by clicking check boxes against the packet names</td><td>Packet information is saved in a log file created in a pre-specified local directory</td></tr>
</table>

Use Case ID:	UC-005
Use Case Name:	Filter Packets
Description:	Enables users to view information of packets of their preference

Actors:	All users		
Pre conditions	Users should start the application and select the type of packets to filter		
Post conditions	Users should be displayed only those type of packets that have been filtered out by the user		
Frequency of Use:	Very frequent		
Flow of Events:		Actor Action	System Response
	1	Start the application	User interface displayed
	2	Select packet types to view and start monitoring	System displays only filtered packet information

Use Case ID:	UC-006
Use Case Name:	Display Packet Header
Description:	Enables users to view expanded information of selected packet(s)

Actors:	All users		
Pre conditions	Users should start the application, start monitoring packets and select the packet whose header is to be expanded		
Post conditions	Users should be displayed the entire packet information in its correct form		
Frequency of Use:	Less frequent		
Flow of Events:		Actor Action	System Response
	1	Start application and click monitor	User interface opens up and transmitted packet information is displayed
	2	Double click on packet to view full header	New application window displays full header of selected packet

Use Case ID:	UC-006
Use Case Name:	Display Network Statistics
Description:	Enables user to view real time statistics of the information being transmitted along the network

Actors:	All users		
Pre conditions	Users should start the applications and start monitoring packets		
Post conditions	Users should be displayed real-time statistics of all transmitted packets such as number of a particular type of packet, origin and destination		
Frequency of Use:	Very frequent		
Flow of Events:		Actor Action	System Response
	1	Start application, start monitoring packets	User interface displayed and packet information displayed on interface
	2	Select Show Network Statistics	A new window application windows displays the relevant statistics of the transmitted packets

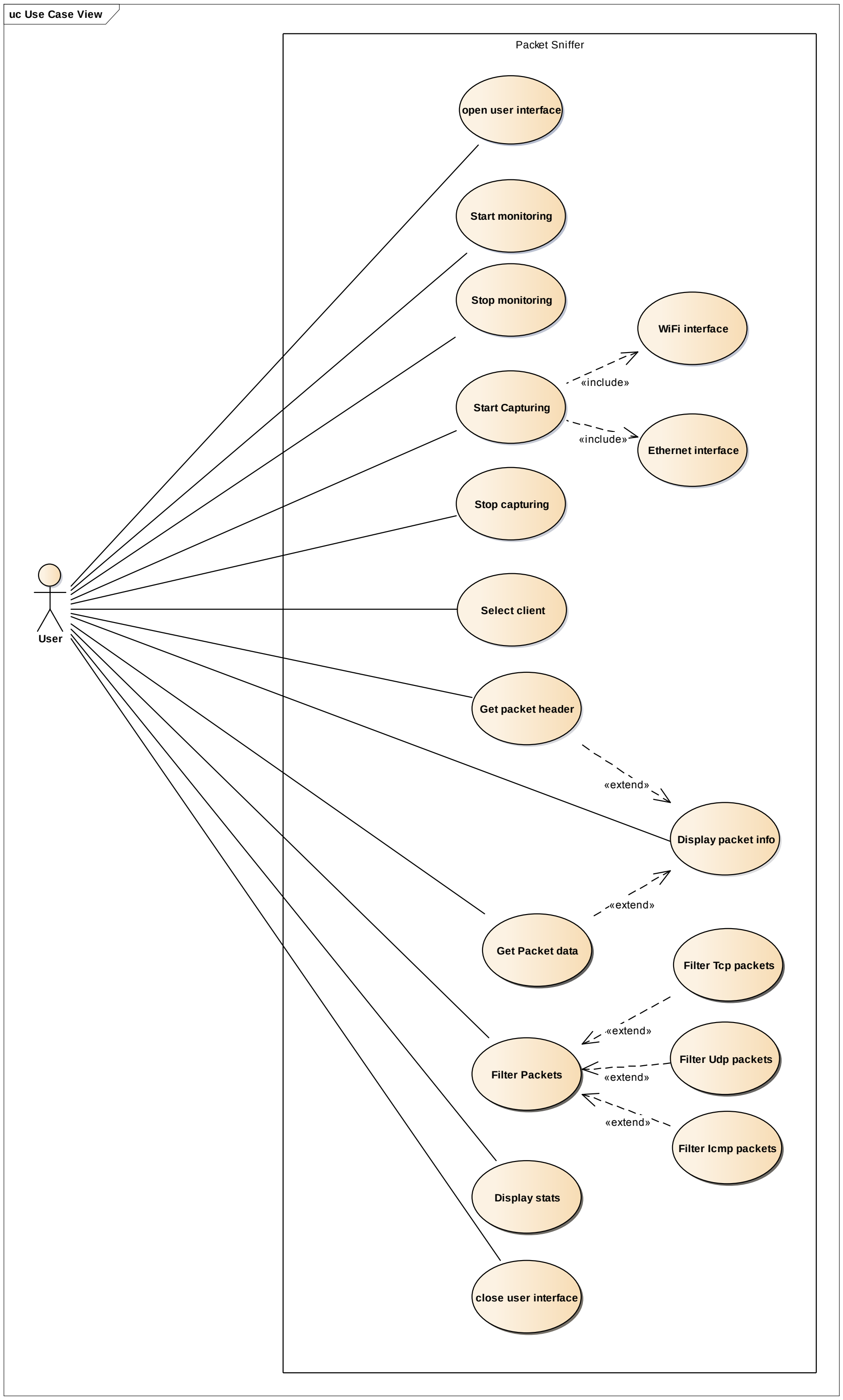
Use Case ID:	UC-007
Use Case Name:	
Description:	

Actors:			
Pre conditions			
Post conditions			
Frequency of Use:			
Flow of Events:		Actor Action	System Response
	1		
	2		
	3		

	4		
Variations:			
Notes and Issues:			
Developer Notes:			

6. Functional View

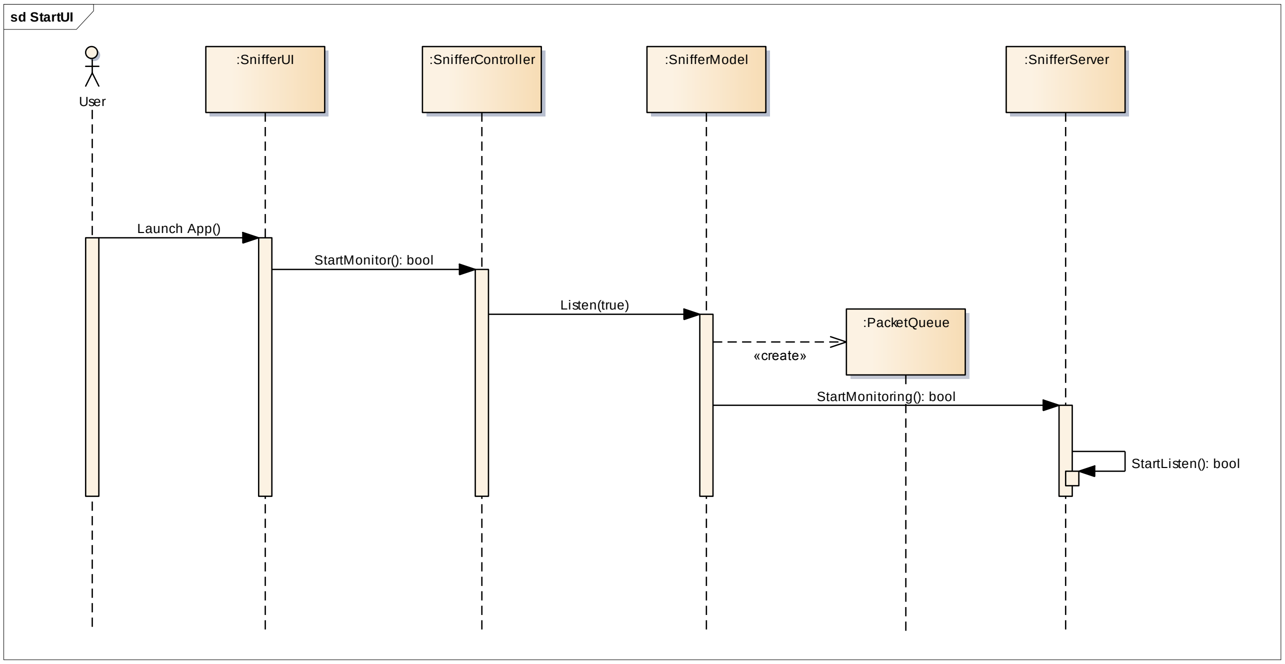
6.1 Use case view



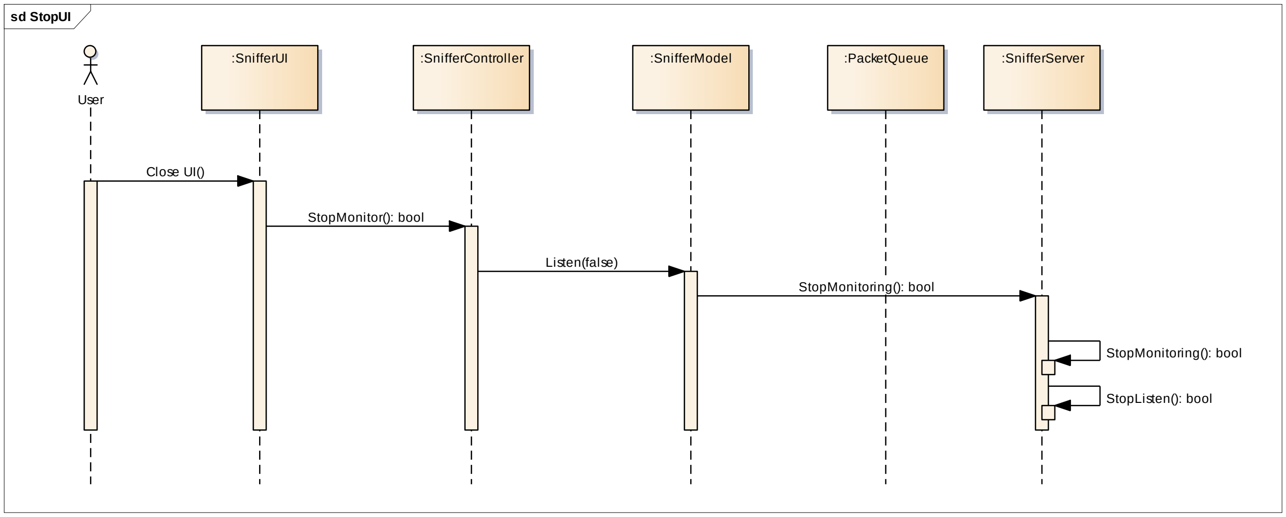
6.2 Logical View

6.2.1 Sequence diagrams

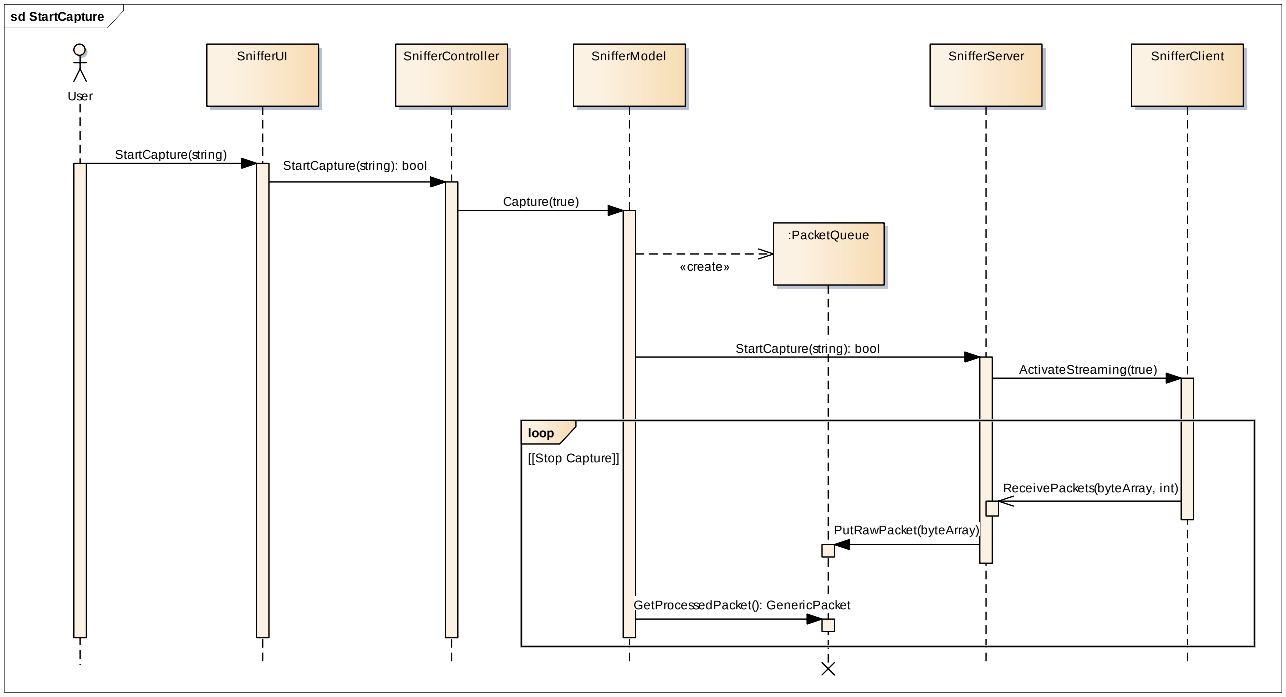
Application launch sequence



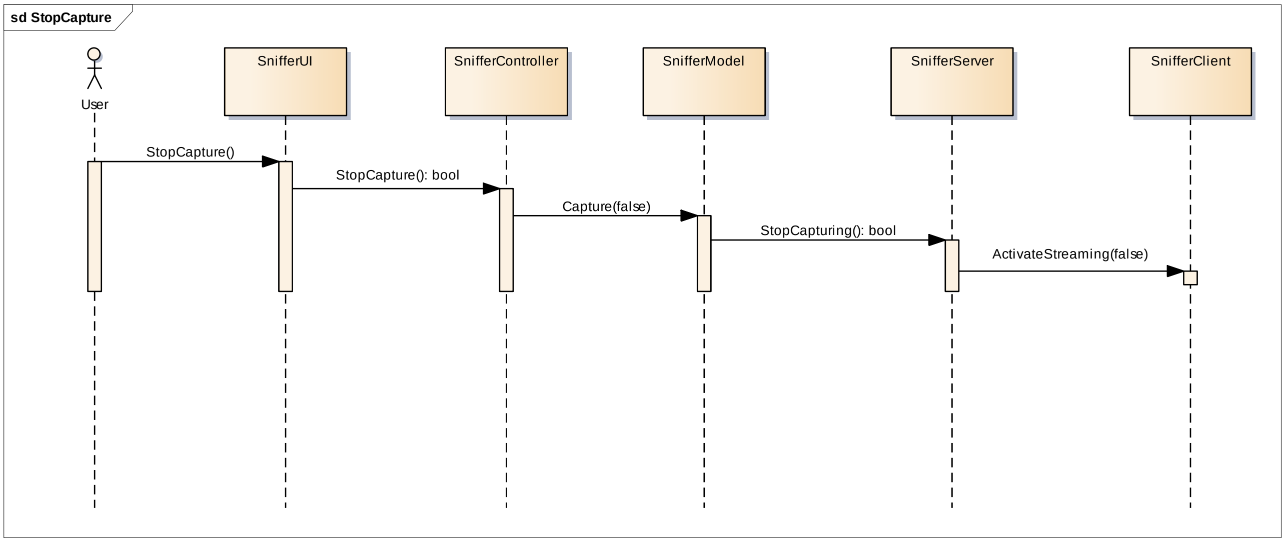
Application stop sequence



Start packet capture

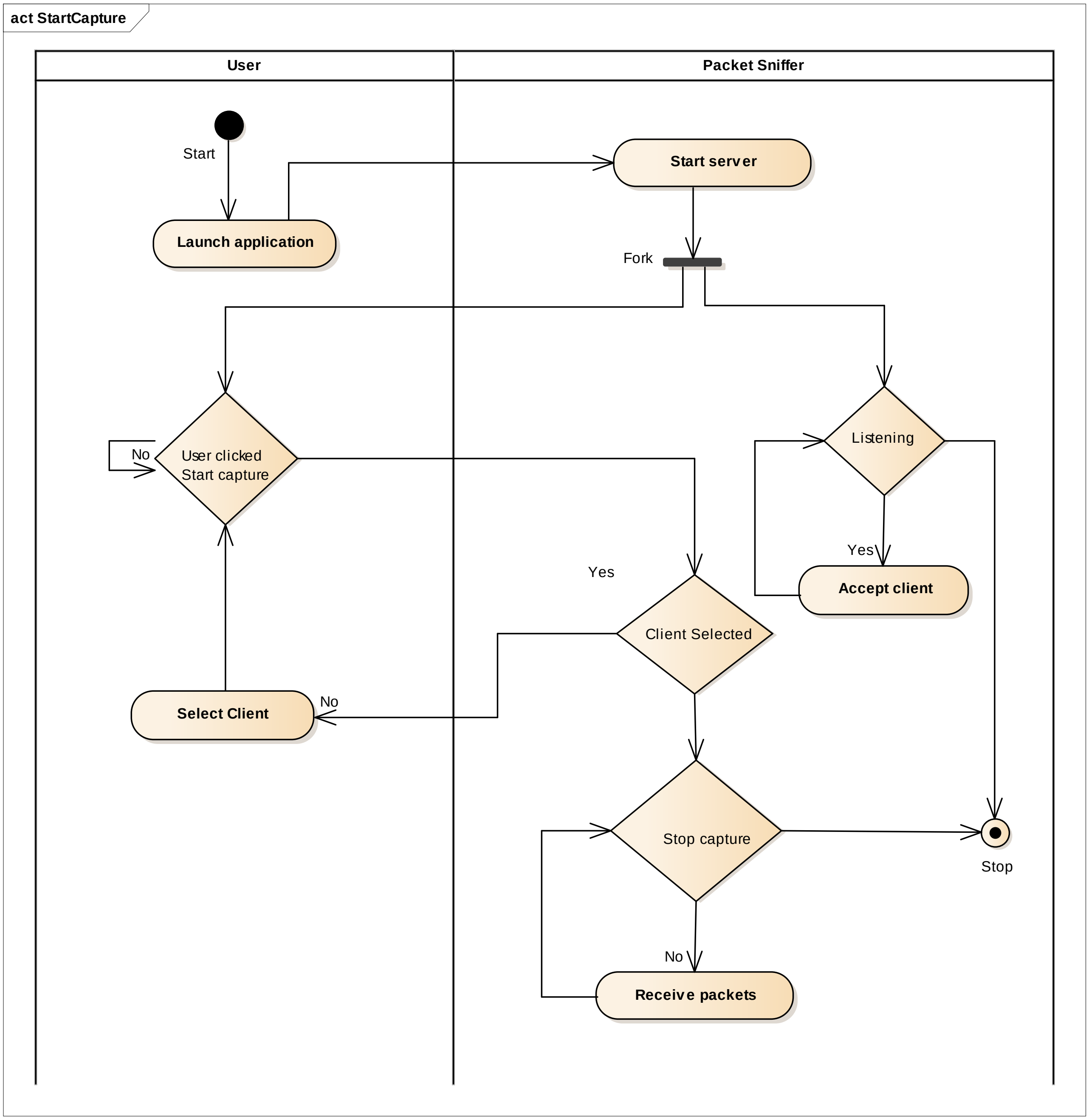


Stop packet capture

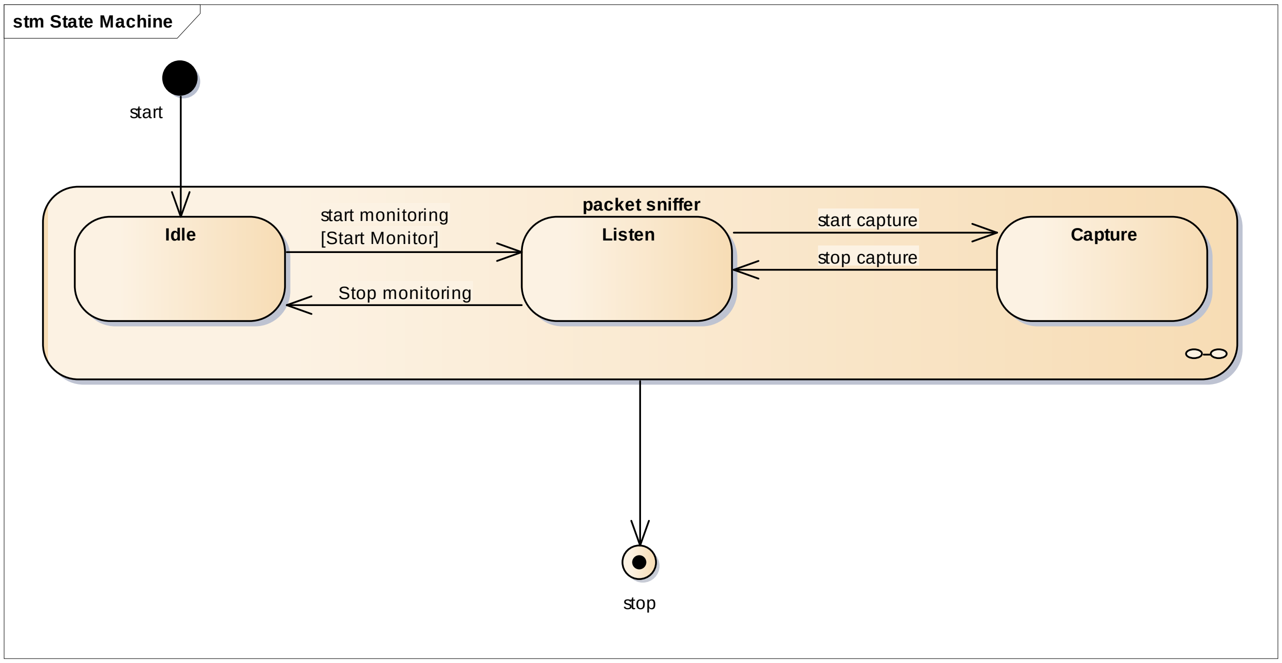


6.2.2 Activity diagrams

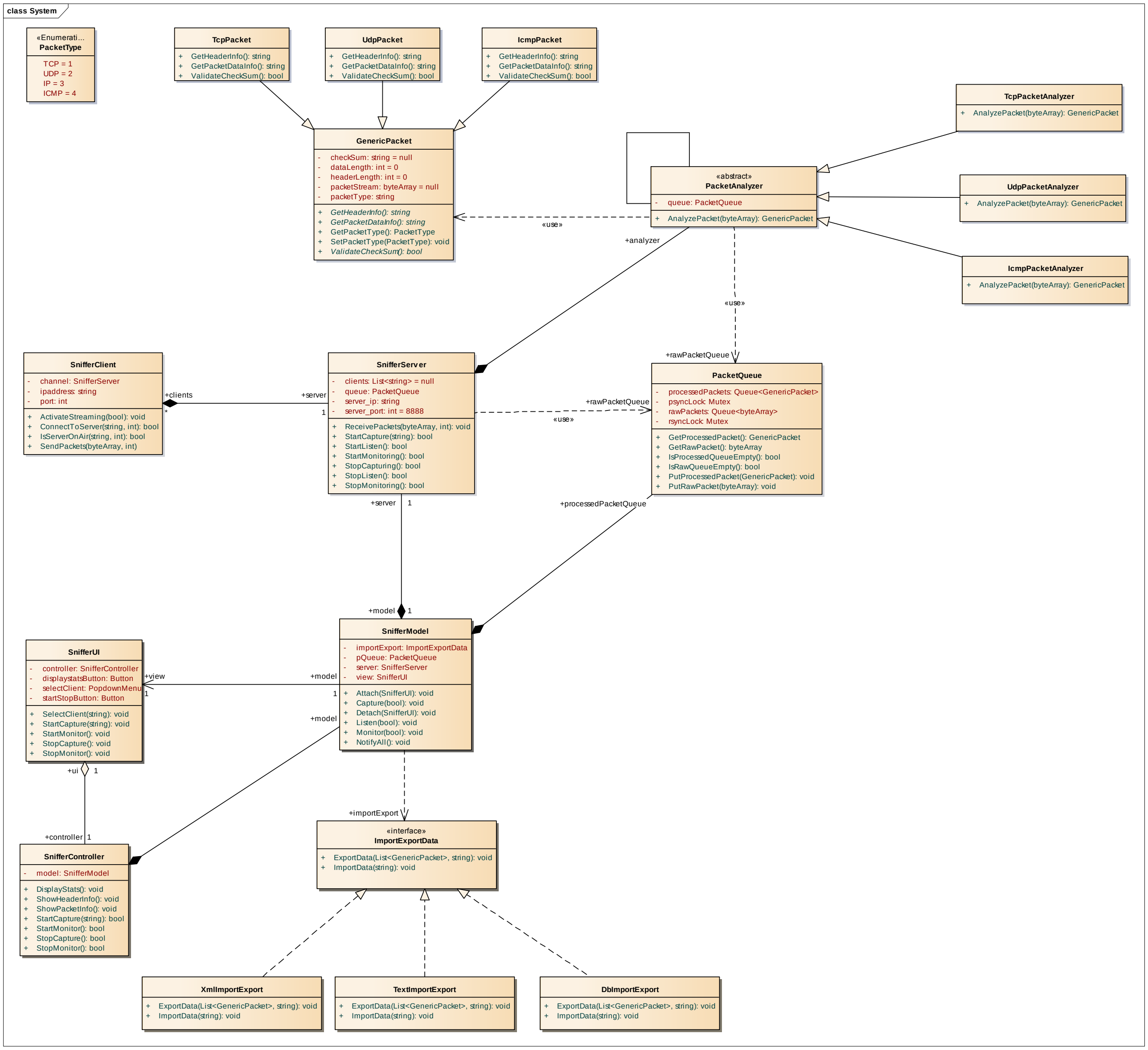
Start capture



6.2.3 State chart diagrams

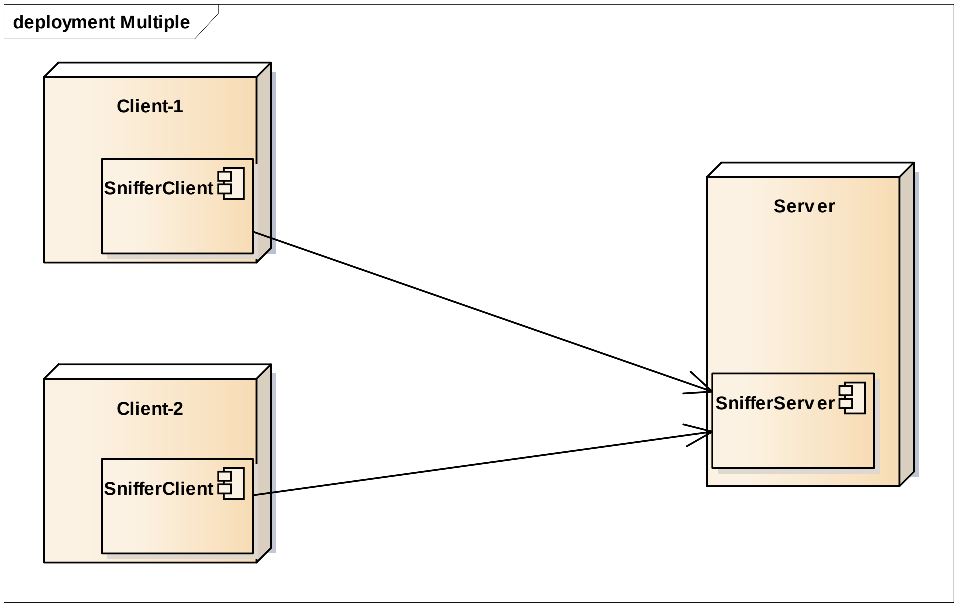


6.2.4 Class diagrams

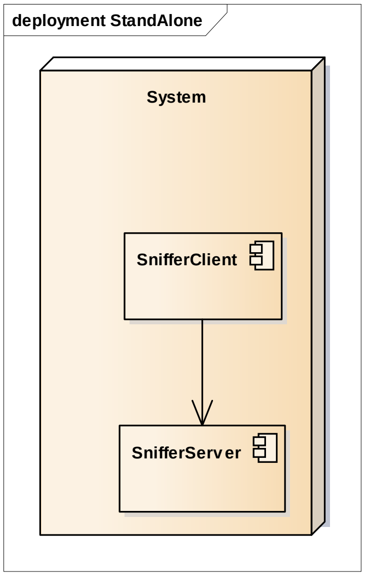

class System
«Enumerati...
PacketType
TCP = 1
UDP = 2
IP = 3
ICMP = 4
TcpPacket
+ GetHeaderInfo(): string
+ GetPacketDataInfo(): string
+ ValidateCheckSum(): bool
UdpPacket
+ GetHeaderInfo(): string
+ GetPacketDataInfo(): string
+ ValidateCheckSum(): bool
IcmpPacket
+ GetHeaderInfo(): string
+ GetPacketDataInfo(): string
+ ValidateCheckSum(): bool
GenericPacket
- checkSum: string = null
- dataLength: int = 0
- headerLength: int = 0
- packetStream: byteArray = null
- packetType: string
+ GetHeaderInfo(): string
+ GetPacketDataInfo(): string
+ GetPacketType(): PacketType
+ SetPacketType(PacketType): void
+ ValidateCheckSum(): bool
«use»
+analyzer
«abstract»
PacketAnalyzer
- queue: PacketQueue
+ AnalyzePacket(byteArray): GenericPacket
TcpPacketAnalyzer
+ AnalyzePacket(byteArray): GenericPacket
UdpPacketAnalyzer
+ AnalyzePacket(byteArray): GenericPacket
IcmpPacketAnalyzer
+ AnalyzePacket(byteArray): GenericPacket
«use»
+rawPacketQueue
SnifferClient
- channel: SnifferServer
- ipaddress: string
- port: int
+ ActivateStreaming(bool): void
+ ConnectToServer(string, int): bool
+ IsServerOnAir(string, int): bool
+ SendPackets(byteArray, int)
+clients
*
+server
1
SnifferServer
- clients: List<string> = null
- queue: PacketQueue
- server_ip: string
- server_port: int = 8888
+ ReceivePackets(byteArray, int): void
+ StartCapture(string): bool
+ StartListen(): bool
+ StartMonitoring(): bool
+ StopCapturing(): bool
+ StopListen(): bool
+ StopMonitoring(): bool
+rawPacketQueue
«use»
PacketQueue
- processedPackets: Queue<GenericPacket>
- psyncLock: Mutex
- rawPackets: Queue<byteArray>
- rsyncLock: Mutex
+ GetProcessedPacket(): GenericPacket
+ GetRawPacket(): byteArray
+ IsProcessedQueueEmpty(): bool
+ IsRawQueueEmpty(): bool
+ PutProcessedPacket(GenericPacket): void
+ PutRawPacket(byteArray): void
+server
1
+processedPacketQueue
+model
1
SnifferModel
- importExport: ImportExportData
- pQueue: PacketQueue
- server: SnifferServer
- view: SnifferUI
+ Attach(SnifferUI): void
+ Capture(bool): void
+ Detach(SnifferUI): void
+ Listen(bool): void
+ Monitor(bool): void
+ NotifyAll(): void
SnifferUI
- controller: SnifferController
- displaystatsButton: Button
- selectClient: PopdownMenu
- startStopButton: Button
+ SelectClient(string): void
+ StartCapture(string): void
+ StartMonitor(): void
+ StopCapture(): void
+ StopMonitor(): void
+view
1
+model
1
+model
+ui
1
+controller
1
SnifferController
- model: SnifferModel
+ DisplayStats(): void
+ ShowHeaderInfo(): void
+ ShowPacketInfo(): void
+ StartCapture(string): bool
+ StartMonitor(): bool
+ StopCapture(): bool
+ StopMonitor(): bool
+importExport
«interface»
ImportExportData
+ ExportData(List<GenericPacket>, string): void
+ ImportData(string): void
XmlImportExport
+ ExportData(List<GenericPacket>, string): void
+ ImportData(string): void
TextImportExport
+ ExportData(List<GenericPacket>, string): void
+ ImportData(string): void
DbImportExport
+ ExportData(List<GenericPacket>, string): void
+ ImportData(string): void

6.3 Deployment View

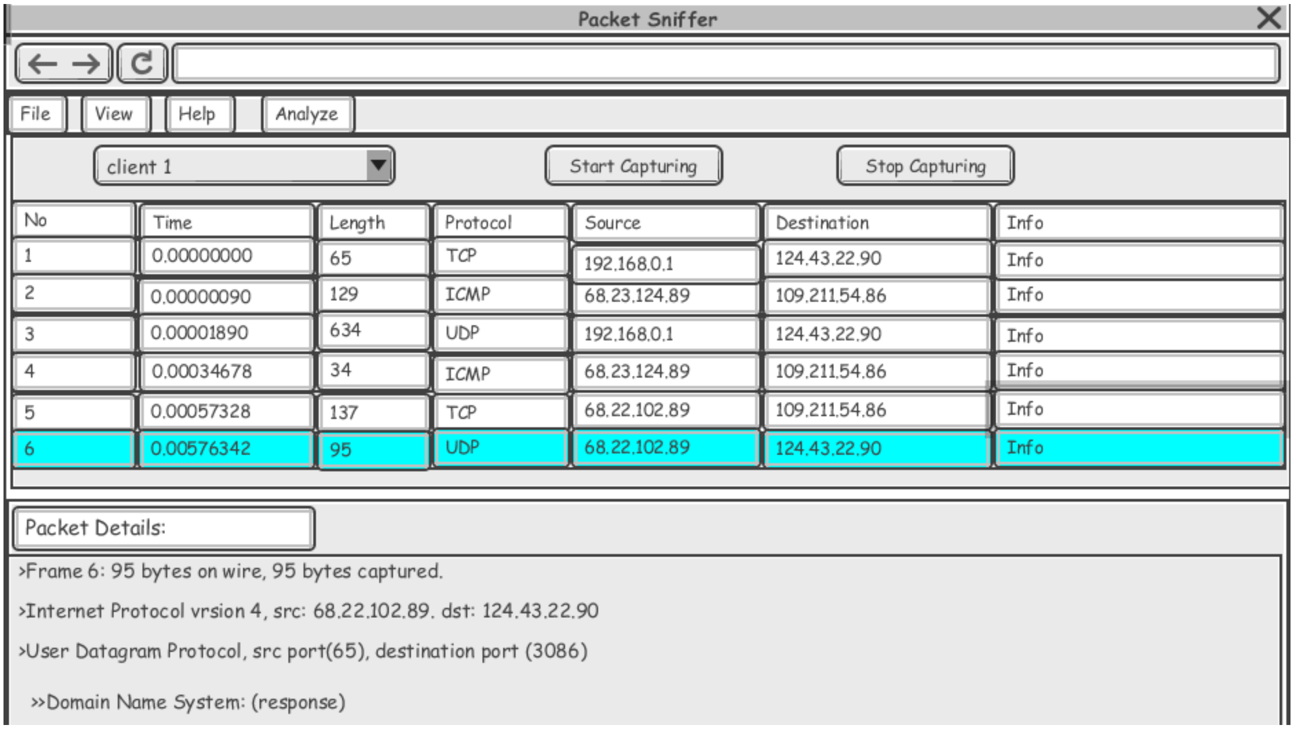
6.3.1 Multi-client deployment



6.3.2 Stand-Alone deployment



5. UI Mock-ups



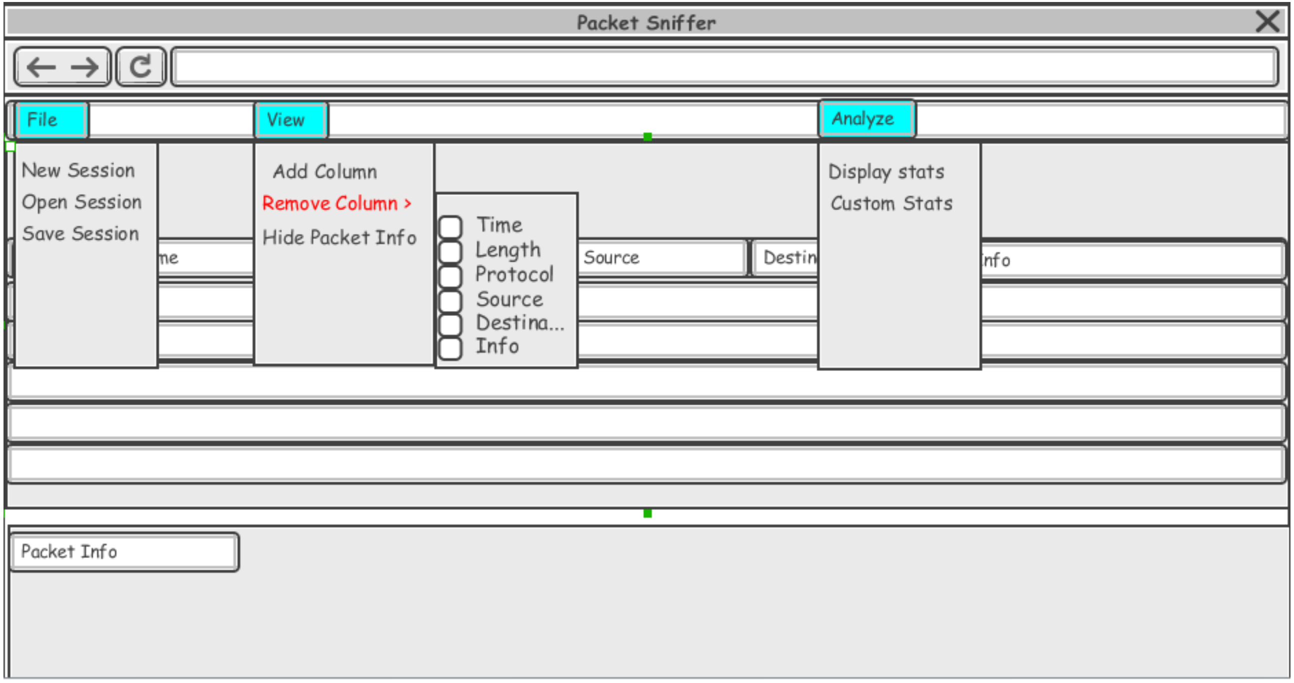
No	Time	Length	Protocol	Source	Destination	Info
1	0.00000000	65	TCP	192.168.0.1	124.43.22.90	Info
2	0.00000090	129	ICMP	68.23.124.89	109.211.54.86	Info
3	0.00001890	634	UDP	192.168.0.1	124.43.22.90	Info
4	0.00034678	34	ICMP	68.23.124.89	109.211.54.86	Info
5	0.00057328	137	TCP	68.22.102.89	109.211.54.86	Info
6	0.00576342	95	UDP	68.22.102.89	124.43.22.90	Info

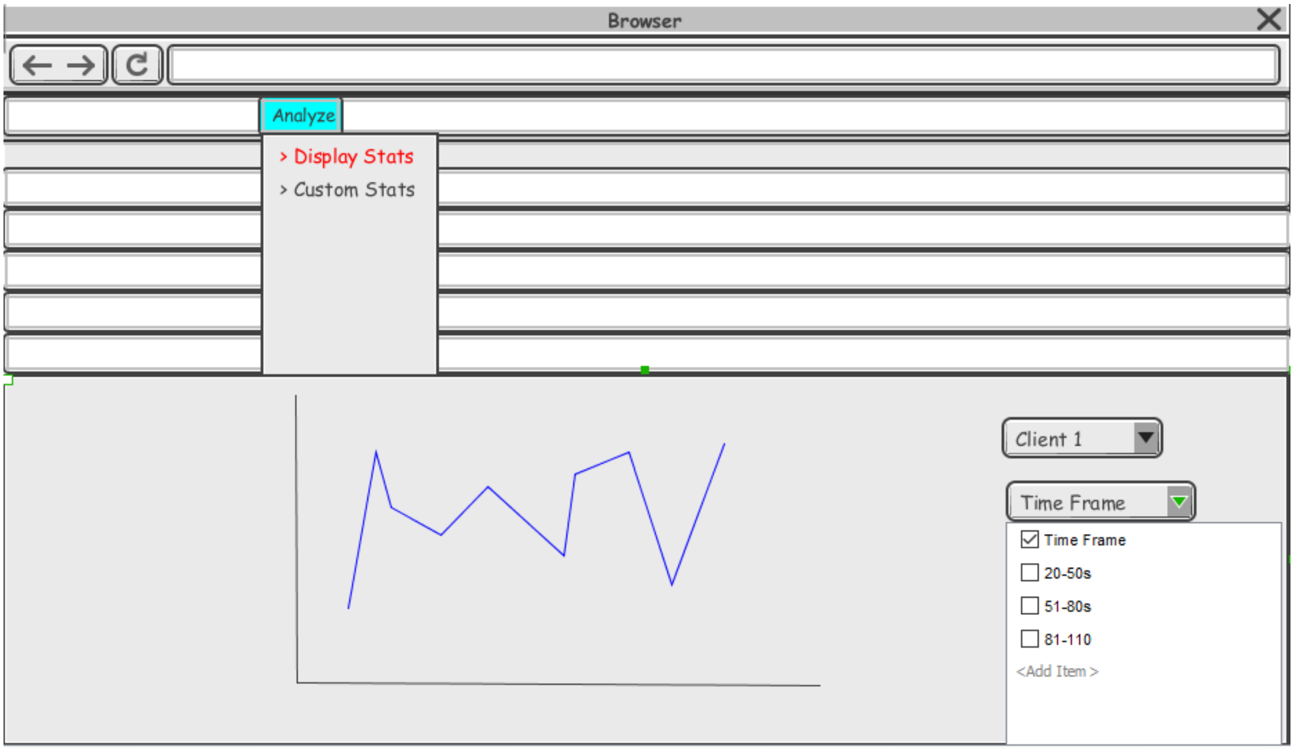
>Frame 6: 95 bytes on wire, 95 bytes captured.

>Internet Protocol vrsion 4, src: 68.22.102.89. dst: 124.43.22.90

>User Datagram Protocol, src port(65), destination port (3086)

>>Domain Name System: (response)





☑ Time Frame
□ 20-50s
□ 51-80s
□ 81-110