



**Faculty of Engineering  
Computer & Systems Eng. Dept.**

## **Computer Networks**

# **Project Documentation**

**Name:** Enas Hosny sa'ed  
Sarah abdelaziz mahmoud  
Heba Allah Essam Eldein  
Hoda Abdelbasit Mohammed

**Presented to:** Dr : Ayman Bahaa & Eng :Aly Osama

**Submission Date:** 23/12/2017

# Project Description

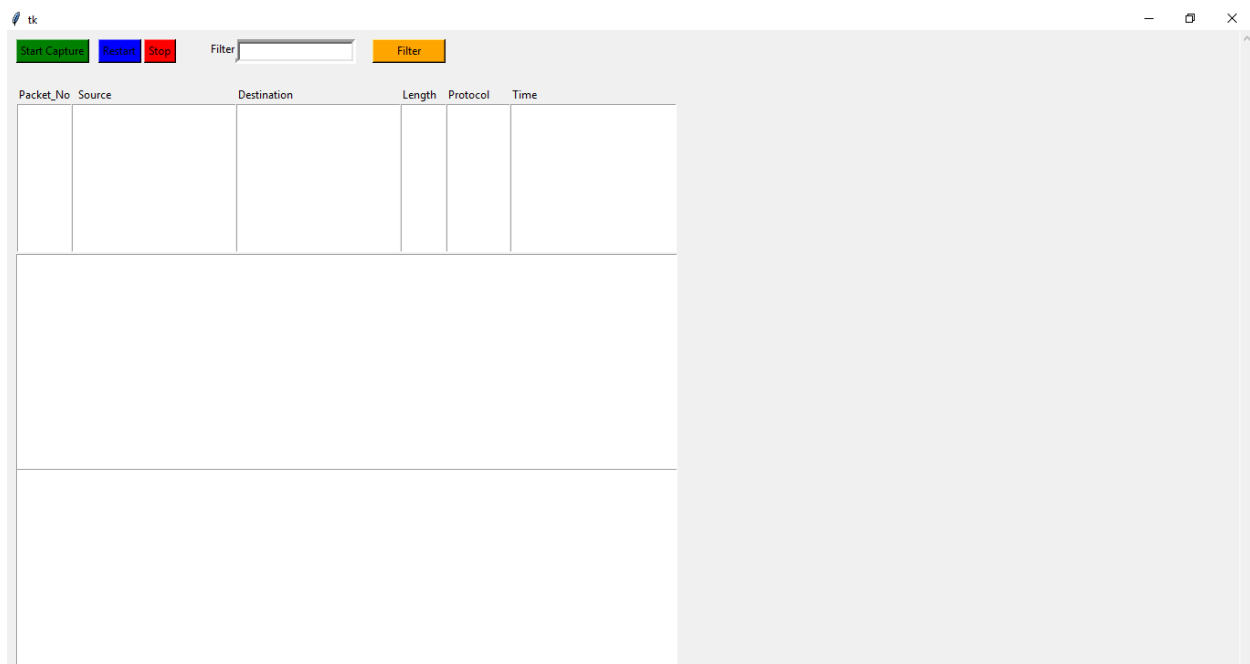
The objective of the project is to make simulation for Wireshark program that is the world's foremost and widely-used network protocol analyzer. It lets you see what's happening on your network at a microscopic level.

The application can :

- 1- Capture packets
- 2- Filter it by filter type ( HTTP,TCP,.....etc)
- 3- Show data for any packets and its details
- 4- Stop capture at any time and restart it also

Application written by python programming language and made GUI with thinker, code of it uploaded on GitHub.

## GUI



# Application Functions details:

## Capture packets

tk					
<div>Start CaptureRestartStopFilter<input type="text"/>Filter</div>					
Packet_No	Source	Destination	Length	Protocol	Time
1	192.168.1.8	104.244.42.3	41	TCP	1513250547.416218
2	104.244.42.3	192.168.1.8	52	TCP	1513250547.603172
3	157.240.21.16	192.168.1.8	284	TCP	1513250549.099973
4	157.240.21.16	192.168.1.8	151	TCP	1513250549.10085
5	192.168.1.8	157.240.21.16	40	TCP	1513250549.10104
6	192.168.1.8	157.240.21.16	519	TCP	1513250549.149939
7	157.240.21.16	192.168.1.8	82	TCP	1513250549.396784
8	192.168.1.8	157.240.21.16	40	TCP	1513250549.438102
9	192.168.1.8	157.240.21.35	348	TCP	1513250550.727579
10	192.168.1.8	157.240.21.35	722	TCP	1513250550.727664

## Show packet details and data

Spyder (Python 3.6)

File Edit Search Source Run Debug Consoles Projects Tools View Help

```
Editor - C:\Users\Hoda Abdelbasi\Downloads\untitled0 (3).py

untitled0 (3).py

184     destination = pkt[scapy.IPv6].dst
185     length = str(pkt[scapy.IPv6].plen)
186     proto_field = pkt[scapy.IPv6].get_field('nh')
187     protocol = proto_field.i2s[pkt.nh]
188     elif(pkt.haslayer(scapy.IP)):
189         source = pkt[scapy.IP].src
190         destination = pkt[scapy.IP].dst
191         length = str(pkt[scapy.IP].len)
192         proto_field = str(pkt[scapy.IP].proto)
193         if(proto_field == "6"):
194             if(str(pkt[scapy.TCP].sport) == "80" or str(pkt[scapy.TCP].dport) == "80"):
195                 protocol = "HTTP"
196             else:
197                 protocol = "TCP"
198         elif(proto_field == "17"):
199             protocol = "UDP"
200         else:
201             protocol = "ip other"
202     elif(pkt.haslayer(scapy.ARP)):
203         source = pkt[scapy.ARP].psrc
204         destination = pkt[scapy.ARP].pdst
205         length = str(pkt[scapy.ARP].plen)
206         protocol = "ARP"
207
208     else:
209         source = "OTHER"
210         destination = "OTHER"
211         length = "OTHER"
212         protocol = "OTHER"
213
214     Source.insert(i, source)
215     Destination.insert(i, destination)
216     Protocol.insert(i, protocol)
217     Length.insert(i, length)
218     Time.insert(i, time)
219     pktnum.insert(i, pkt_no)
220     i=i+1
221     print("SRC = " + source + " DESTINATION = " + destination + " LENGTH = " + length + " TIME = " + time)
```

tk

Start Capture Restart Stop Filter Filter

Packet_No	Source	Destination	Length	Protocol	Time
1	192.168.1.8	104.244.42.3	41	TCP	1513250547.416218
2	104.244.42.3	192.168.1.8	52	TCP	1513250547.603172
3	157.240.21.16	192.168.1.8	284	TCP	1513250549.099973
4	157.240.21.16	192.168.1.8	151	TCP	1513250549.10085
5	192.168.1.8	157.240.21.16	40	TCP	1513250549.10104
6	192.168.1.8	157.240.21.16	519	TCP	1513250549.149939
7	157.240.21.16	192.168.1.8	82	TCP	1513250549.396784
8	192.168.1.8	157.240.21.16	40	TCP	1513250549.438102
9	192.168.1.8	157.240.21.35	348	TCP	1513250550.727579
10	192.168.1.8	157.240.21.35	722	TCP	1513250550.727664

### [ Ethernet ] ###

dst = 70:9f:2d:83:a3:d0  
src = 2c:6e:85:ed:e6:0d  
type = 0x800

### [ IP ] ###

version = 4  
ihl = 5  
tos = 0x0  
len = 41  
id = 2551  
flags = DF  
frag = 0  
ttl = 128  
proto = tcp  
checksum = 0x9a30

0000 709f2d83a3d02c6e85ede60d08004500 p.-...,n.....E.  
0010 002909f7400080069c30c0a8010868f4 .).....0....h.  
0020 2a03fd3d01bb5401fe5cb3a9a40b5010 \*....T..\\....P.  
0030 0041b1de000000 .A.....

## Filtering packets with filter type

tk

Start Capture Restart Stop Filter ARP Filter

Packet_No	Source	Destination	Length	Protocol	Time
24	192.168.1.1	192.168.1.8	4	ARP	1513250551.854356
26	192.168.1.8	192.168.1.1	4	ARP	1513250551.854579
244	192.168.1.1	192.168.1.8	4	ARP	1513250588.382037
245	192.168.1.8	192.168.1.1	4	ARP	1513250588.382117
999	192.168.1.1	192.168.1.8	4	ARP	1513250627.152597
1000	192.168.1.8	192.168.1.1	4	ARP	1513250627.152676
1259	192.168.1.1	192.168.1.8	4	ARP	1513250667.143169

```
###[ Ethernet ]###
dst      = 70:9f:2d:83:a3:d0
src      = 2c:6e:85:ed:e6:0d
type     = 0x800
###[ IP ]###
version  = 4
ihl      = 5
tos      = 0x0
len      = 41
id       = 2551
flags    = DF
frag     = 0
ttl      = 128
proto    = tcp
checksum = 0x9c30
0000  709FD83A3D02C6E85EDE60D08004500 p.-...,n.....E.
0010  002909F7400080069C30C0A8010868F4 .)..<...0....h.
0020  2A03FD3D01BB5401FE5CB3A9A40B5010 *..=.T..\....P.
0030  0041B1DE0000000 .A.....
```

```
184 destination = p
185 length = str(pk
186 proto_field = p
187 protocol = prot
188 elif(pkt.haslayer(sc
189 source = pkt[sca
190 destination = pk
191 length =str(pkt[
192 proto_field = st
193 if(proto_field =
194     if(st
195     p
196     else:
197     p
198 elif(proto_field
199     protocol =
200 else:
201     protocol =
202 elif(pkt.haslayer(sc
203 source = pkt[s
204 destination = p
205 length = str(p
206 protocol = "ARP
207
208 else:
209     source = "OTHER"
210     destination = "O
211     length = "OTHER"
212     protocol = "OTHE
213
214 Source.insert(i, sou
215 Destination.insert(i
216 Protocol.insert(i,pr
217 Length.insert(i,leng
218 Time.insert(i,time)
219 pktnum.insert(i,pkt_
220 i=i+1
221 print("<SR< = " + sou
```