Assignment No. 9

Name: Rohit Metha Roll No.: TCOD29 Batch: T11

Server Script:

Source Code:

```
# Server code to enable file transfer using UDP socket

# This is server code to send video frames over UDP
import cv2, imutils, socket
import time
import base64

BUFF_SIZE = 65536
server_socket = socket.socket(socket.AF_INET,socket.SOCK_DGRAM)
server_socket.setsockopt(socket.SOL_SOCKET,socket.SO_RCVBUF,BUFF_SIZE)
host_name = socket.gethostname()
host_ip = '192.168.0.2'#  socket.gethostbyname(host_name)
print(host_ip)
port = 9999
socket_address = (host_ip,port)
server_socket.bind(socket_address)
print('Listening at:',socket_address)

vid = cv2.VideoCapture('Working Demo Inshot.mp4') #  replace '*.mp4' with 0 for webcam
fps,st,frames_to_count,cnt = (0,0,20,0)

while True:
	msg,client_addr = server_socket.recvfrom(BUFF_SIZE)
	print('GOT connection from ',client_addr)
	WIDTH=500
	while(vid.isOpened()):
		_,frame = vid.read()
		frame = imutils.resize(frame,width=WIDTH)
		encoded,buffer =
cv2.imencode('.jpg',frame,[cv2.IMWRITE_JPEG_QUALITY,80])
		message = base64.b64encode(buffer)
		server_socket.sendto(message,client_addr)
		frame = cv2.putText(frame,'FPS:
'+str(fps),(10,40),cv2.FONT_HERSHEY_SIMPLEX,0.7,(0,0,255),2)
		cv2.imshow('TRANSMITTING VIDEO',frame)
		key = cv2.waitKey(1) & 0xFF
		if key == ord('q'):
			server_socket.close()
			break
```

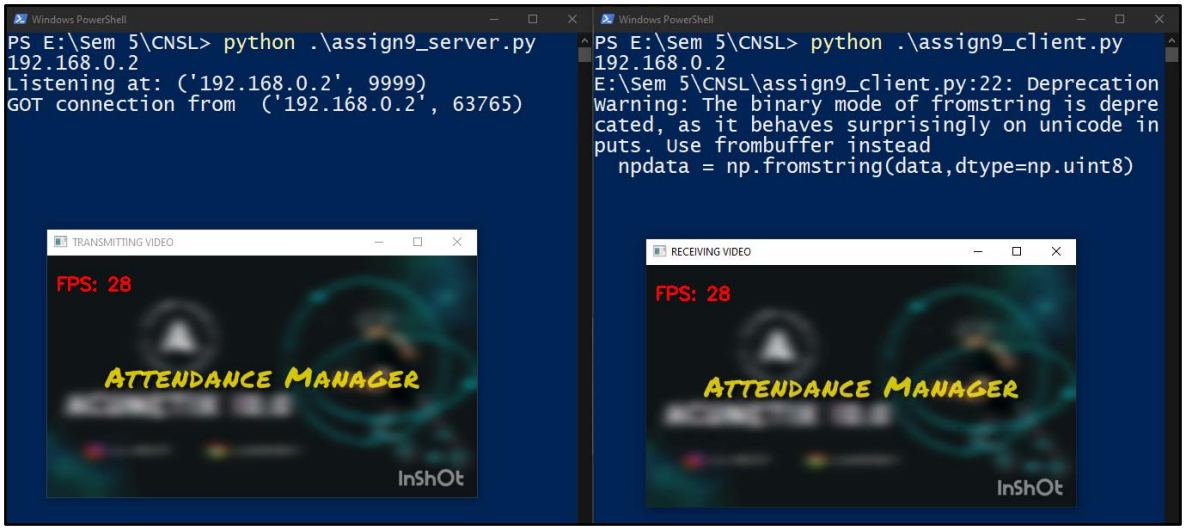
```
if cnt == frames_to_count:
        try:
                fps = round(frames_to_count/(time.time()-st))
                st=time.time()
                cnt=0
        except:
                pass
cnt+=1
```

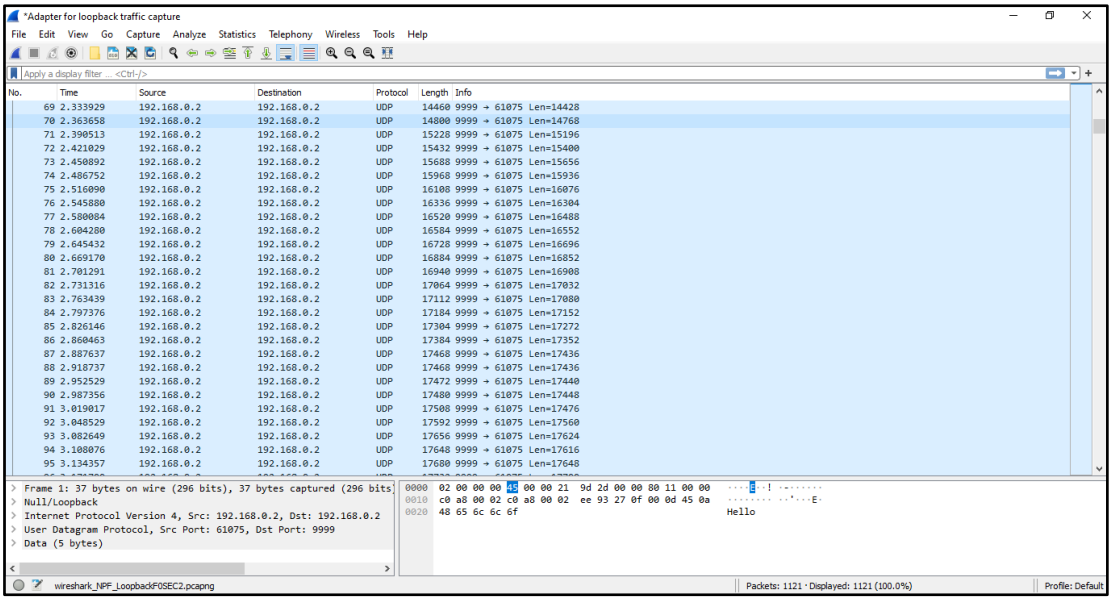
Client Script:

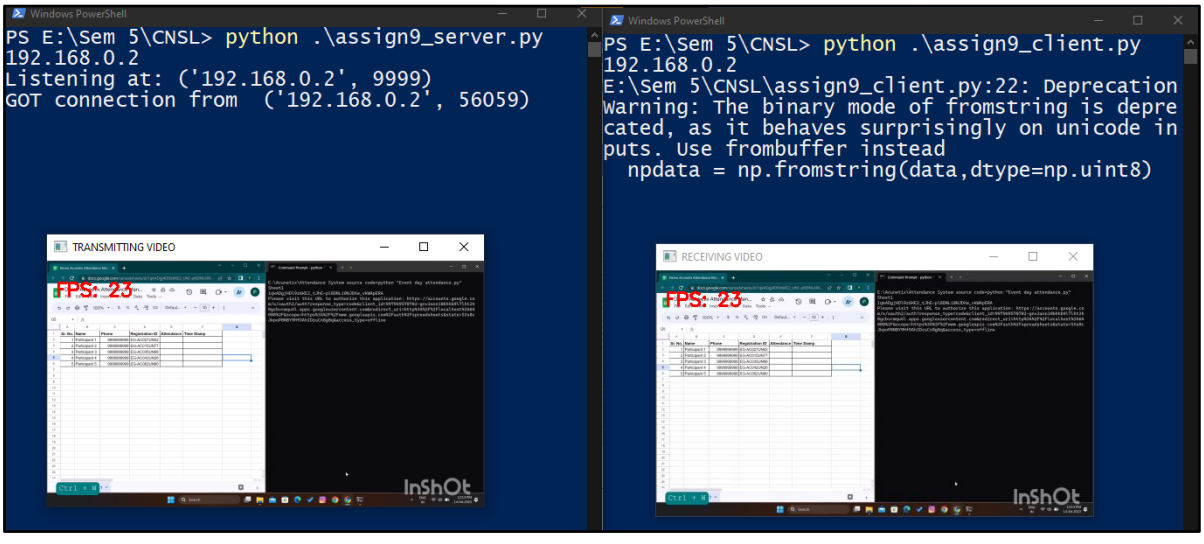
Source Code:

```
# This is client code to receive video frames over UDP
import cv2, socket
import numpy as np
import time
import base64

BUFF_SIZE = 65536
client_socket = socket.socket(socket.AF_INET,socket.SOCK_DGRAM)
client_socket.setsockopt(socket.SOL_SOCKET,socket.SO_RCVBUF,BUFF_SIZE)
host_name = socket.gethostname()
host_ip = '192.168.0.2'#  socket.gethostbyname(host_name)
print(host_ip)
port = 9999
message = b'Hello'

client_socket.sendto(message,(host_ip,port))
fps,st,frames_to_count,cnt = (0,0,20,0)
while True:
	packet,_ = client_socket.recvfrom(BUFF_SIZE)
	data = base64.b64decode(packet,' /')
	npdata = np.fromstring(data,dtype=np.uint8)
	frame = cv2.imdecode(npdata,1)
	frame = cv2.putText(frame,'FPS:
'+str(fps),(10,40),cv2.FONT_HERSHEY_SIMPLEX,0.7,(0,0,255),2)
	cv2.imshow("RECEIVING VIDEO",frame)
	key = cv2.waitKey(1) & 0xFF
	if key == ord('q'):
		client_socket.close()
		break
	if cnt == frames_to_count:
		try:
			fps = round(frames_to_count/(time.time()-st))
			st=time.time()
			cnt=0
		except:
			pass
	cnt+=1
```

Output:


Windows PowerShell
PS E:\Sem 5\CNSL> python .\assign9_server.py
192.168.0.2
Listening at: ('192.168.0.2', 9999)
GOT connection from ('192.168.0.2', 63765)
TRANSMITTING VIDEO
FPS: 28
ATTENDANCE MANAGER
InShOt
Windows PowerShell
PS E:\Sem 5\CNSL> python .\assign9_client.py
192.168.0.2
E:\Sem 5\CNSL\assign9_client.py:22: Deprecation
Warning: The binary mode of fromstring is depre
cated, as it behaves surprisingly on unicode in
puts. Use frombuffer instead
npdata = np.fromstring(data,dtype=np.uint8)
RECEIVING VIDEO
FPS: 28
ATTENDANCE MANAGER
InShOt


*Adapter for loopback traffic capture
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
No. Time Source Destination Protocol Length Info
69 2.333929 192.168.0.2 192.168.0.2 UDP 14460 9999 → 61075 Len=14428
70 2.363658 192.168.0.2 192.168.0.2 UDP 14800 9999 → 61075 Len=14768
71 2.390513 192.168.0.2 192.168.0.2 UDP 15228 9999 → 61075 Len=15196
72 2.421029 192.168.0.2 192.168.0.2 UDP 15432 9999 → 61075 Len=15400
73 2.450892 192.168.0.2 192.168.0.2 UDP 15688 9999 → 61075 Len=15656
74 2.486752 192.168.0.2 192.168.0.2 UDP 15968 9999 → 61075 Len=15936
75 2.516090 192.168.0.2 192.168.0.2 UDP 16108 9999 → 61075 Len=16076
76 2.545880 192.168.0.2 192.168.0.2 UDP 16336 9999 → 61075 Len=16304
77 2.580084 192.168.0.2 192.168.0.2 UDP 16520 9999 → 61075 Len=16488
78 2.604280 192.168.0.2 192.168.0.2 UDP 16584 9999 → 61075 Len=16552
79 2.645432 192.168.0.2 192.168.0.2 UDP 16728 9999 → 61075 Len=16696
80 2.669170 192.168.0.2 192.168.0.2 UDP 16884 9999 → 61075 Len=16852
81 2.701291 192.168.0.2 192.168.0.2 UDP 16940 9999 → 61075 Len=16908
82 2.731316 192.168.0.2 192.168.0.2 UDP 17064 9999 → 61075 Len=17032
83 2.763439 192.168.0.2 192.168.0.2 UDP 17112 9999 → 61075 Len=17080
84 2.797376 192.168.0.2 192.168.0.2 UDP 17184 9999 → 61075 Len=17152
85 2.826146 192.168.0.2 192.168.0.2 UDP 17304 9999 → 61075 Len=17272
86 2.860463 192.168.0.2 192.168.0.2 UDP 17384 9999 → 61075 Len=17352
87 2.887637 192.168.0.2 192.168.0.2 UDP 17468 9999 → 61075 Len=17436
88 2.918737 192.168.0.2 192.168.0.2 UDP 17468 9999 → 61075 Len=17436
89 2.952529 192.168.0.2 192.168.0.2 UDP 17472 9999 → 61075 Len=17440
90 2.987356 192.168.0.2 192.168.0.2 UDP 17480 9999 → 61075 Len=17448
91 3.019017 192.168.0.2 192.168.0.2 UDP 17508 9999 → 61075 Len=17476
92 3.048529 192.168.0.2 192.168.0.2 UDP 17592 9999 → 61075 Len=17560
93 3.082649 192.168.0.2 192.168.0.2 UDP 17656 9999 → 61075 Len=17624
94 3.108876 192.168.0.2 192.168.0.2 UDP 17648 9999 → 61075 Len=17616
95 3.134357 192.168.0.2 192.168.0.2 UDP 17680 9999 → 61075 Len=17648
Frame 1: 37 bytes on wire (296 bits), 37 bytes captured (296 bits)
Null/Loopback
Internet Protocol Version 4, Src: 192.168.0.2, Dst: 192.168.0.2
User Datagram Protocol, Src Port: 61075, Dst Port: 9999
Data (5 bytes)
0000 02 00 00 00 45 00 00 21 9d 2d 00 00 80 11 00 00
0010 c0 a8 00 02 c0 a8 00 02 ee 93 27 0f 00 0d 45 0a
0020 48 65 6c 6c 6f
Hello
wireshark_NPF_Loopback F0SEC2.pcapng
Packets: 1121 · Displayed: 1121 (100.0%)
Profile: Default


Windows PowerShell
PS E:\Sem 5\CNSL> python .\assign9_server.py
192.168.0.2
Listening at: ('192.168.0.2', 9999)
GOT connection from ('192.168.0.2', 56059)
TRANSMITTING VIDEO
FPS: 23
InShOt
Windows PowerShell
PS E:\Sem 5\CNSL> python .\assign9_client.py
192.168.0.2
E:\Sem 5\CNSL\assign9_client.py:22: Deprecation
Warning: The binary mode of fromstring is depre
cated, as it behaves surprisingly on unicode in
puts. Use frombuffer instead
npdata = np.fromstring(data,dtype=np.uint8)
RECEIVING VIDEO
FPS: 23
InShOt