Task 2 – Report (1187 words)

In this assignment, a simple client-server application has been developed in order to demonstrate the communication between the client and server to simulate Go back N and selective repeat modes of sliding windows protocol. This communication is peer to peer based on the UDP protocol [1]. This provides the chance to the others to recover any data by means of the internet irrespective of the information transfer electronic components that they are using. Information can be recovered simply with the exact tools and methods [2].

The idea behind sliding window protocol and Go back N is that both the sender (client) and the receiver (server) keep a window of acknowledgment. Whereas:

- The client keeps the record of expected acknowledgment received from the server.
- The server keeps the record or sequence of the expected receiving frame.

When an acknowledgment is received, the client increments or advances the window, and when the server receives the expected frame, the server increments the window.

JAVA has been selected since the program will run on any platform with java. Also the in-built support of Java, for UDP/IP, via the DatagramPacket and DatagramSocket classes, is a factor towards selecting Java.

Overview of the Solution:

This solution is based on using a single class for a single functionality, up to the extent possible. The solution is designed to work on a network, supporting UDP/IP. Since UDP/IP is an unreliable protocol, the objective is to bring in support for reliability, using the window protocol and Go back N technique to ensure that packets of data are delivered within a certain time frame. If the client does not receive an acknowledgment of receipt of a data packet, then the same is re-transmitted to the client, with all the subsequent packets after that.

The objective is to illustrate how window protocol and Go back N can be used to achieve reliability of packet delivery, even using an unreliable protocol, which is actually how some of the reliable protocols work.

Overview of the Program Design:

The basis of the solution is to use the following classes-

1. The “SD_DataPacket” class
2. The “SD_ClientPacketHelper” class
3. The “PacketReceiverThread” class
4. The “Client” class

5. The “Server” class

1. The “SD_DataPacket” class- It is the class responsible for encapsulating the data to be sent from the client to the server and back. It contains the properties to be sent from client to server and back and also the sequence number of the packet (A better option would also include the total number of packets, like TCP/IP, which can be implemented in the future). It also contains properties to indicate whether the packet is from the server or client or is acknowledged or not. Apart from the general getter/setter methods on the properties, this class also contains two methods –

    a. “getNetworkFormat()”- which returns an array of bytes, after combining the properties into a delimited string. We have chosen the delimiter “#@”, but can be any delimiter by changing the value of the variable “pathSeparator”.

    b. “retriveNetworkFormat()” – which takes a byte array (usually the data from “getNetworkFormat()” sent over the network and received) and checks to see whether it is in the format of the application. If it is, it returns true, all the while setting the appropriate values to the properties of the class object.

2. The “SD_ClientPacketHelper” class- It is the class responsible for initializing the packets on the client side collection, for transmission, finding out the number packets, getting/setting the acknowledgment, getting/setting a particular packet with an SN Number, getting the total number of packets acknowledged, remove a packet from the collection, and so on.

3. The “PacketReceiverThread” class- it is used to set the acknowledgment state and send the packet back to the client, on a separate thread so that the server is free to function, without blocking, during this operation.

4. The “Client” class- it is used to connect to the server, initialize a collection of sending the packets of the collection of “SD_DataPacket” packets, i.e., the number of packets to send to the server, connect to the server, resend the current window, if acknowledgment is not received within a time frame, and send data to the server, as well as receive the data from the server. All the properties, like server address, port, number of packets to send, time to wait for, window size, are all dynamic and input from the user. The only thing fixed is the input file,

which is “input.txt” but can be made dynamic, by replacing the assignment with a “readLine()” of the scanner class.

5. The “Server” class – is the server, which initially binds to a port and waits for client connections. It is responsible for receiving connections on the port and responding appropriately for data packets that meet the format of our application(i.e., of type “SD_DataPacket”). Since we are testing it on “localhost,”

This program is written in Java using UDP sockets.

The server program binds with a particular port and also listens to client requests in its initial phase. And when the server gets a packet, it matches the sequence within the stockpiled sequence. For example, the sequence does not match; the server sends an acknowledgment to the client for the packet received. And, based on some casual no possibility, it tries not to send any acknowledgment to the client so that the client may resend the packet based on gliding window size. Then all the function is applied in a distinct thread maintained by proper print reports. The client database takes the input opinions of the IP address that port not to the server. The client database keeps the way of unrecognized packets [8].
The Java socket API provides the basis of TCP/UDP communication [9]. Java provides Datagram Sockets for network communication via UDP instead of TCP. Datagram Sockets can be used to both send and receive packets over the Internet. Datagram packets are used to implement a connectionless packet delivery service supported by the UDP protocol. Each message is transferred from the source machine to the destination based on information contained within that packet. That means each packet needs to have a destination address, and each packet might be routed differently and might arrive in any order. Packet delivery is not guaranteed. The format of the datagram packet is:

Java supports datagram communication through the following classes:

- Datagram Packet: To implement a connectionless packet transfer service Datagram Packets are used. Each and every message is routed by one engine/machine to the other based merely on the data contained with the packet. And countless packets sent from one engine/machine to the other may be routed in a different way, and it might reach in any order.
- Datagram Socket: A Datagram is the name of an autonomous/self-contained message sent over the network whose entrance time and content are not

guaranteed. The Datagram Socket and Datagram Packet class in JAVA.NET [10] package applied system-free Datagram communication by using UDP.

Java is very suitable to implement a client-server application using either TCP or UDP as it has strong support and implementation of all the required methods.

Task 3 – Report (1892 words)

2.1 Use of Sequence Numbers

UDP protocol sends messages from one program to another with a minimum header and protocol mechanism. The UDP protocol is transaction-oriented. The delivery and duplication of the packet are not ensured in the UDP protocol. The applications that need to deliver reliable, such as data streams should use the TCP protocol [11]. UDP packets keep different source IP, terminus IP, and (scr and dest) PORT numbers. Therefore, each packet initiates at the exact source and is obtained at the exact destination. The order among packets in a flow might be random and depend upon network track and other quirks. The improper order in obtaining data arises when the unique information sent by the source breaks up into several pieces and is sent in distinct UDP packets [12].

Bozakov and Rizk suggest [13] using a queueing and perfect model to describe the controller's crossing point conduct among the control and switch in terms of the number of checked messages over multiple time scales. In another study, a calculus-based method was used to derive and evaluate the consistent service curves. An easy interface extension for the controller outlines was moreover planned to help operators configure the extension restrictions for the changed control message and presented the mathematical outline based on Web calculus to help the scalable SDN distribution/deployment, in which the higher bound of packet delay and usually the buffer size of the controller was appraised beneath the hypothesis of an increasing arrival procedure at the SDN controller [14].

2.2 Reliability and security considerations when transmitting data over UDP

UDP neither guarantees that the packet will reach its destination nor handshakes with the receiving computer as the TCP protocol does. It merely transfers the data and relies on the sending and receiving computers to obtain the data. Many applications need an acknowledgment of the packet sent using the UDP protocol in which if an acknowledgment is not received in a certain time, the application sends the packet

again otherwise it stops resending [15]. The data packets in the UDP protocol may arrive out of order, duplicated, or not received at all without acknowledgment. The major advantage of UDP is a low overhead.
The TCP protocol provides data security by starting a TCP session and handshaking with the other end. The sender and receiver communicate their IPs and have proof of IP ownership, and all the sent data is read reliably. Currently, two systematic models exist for expecting/predicting the average presentation of two dissimilar flow types (TCP & UDP) over SDN designs, which are considered to be packet-level and flow-level arrival, (i.e. UDP/TCP flow/movement). The UDP and TCP are different in nature as TCP is connection-oriented whereas UDP is connectionless [16].
The packet arrivals flow is a Poisson process as the UDP flow arrivals usually follow a Poisson procedure. Moreover, the flow and packet inter-arrival time in every flow follows an exponential supply. The IP address flow inter-arrival time, however, follows the Weibull distribution. Jie et al. [17] demonstrate that the flow arrival procedure in the real-world packet converting networks can never be approached as a Poisson delivery instead considering the Poisson procedure for SDN signaling traffic i.e., the new flow arrivals.

2.3 Evaluate its effectiveness for certain situations; an example might be gaming

Datagram Sockets are Java's mechanism for network communication via UDP instead of TCP. Java provides Datagram Socket to communicate over UDP instead of TCP which is built on top of IP. Datagram Sockets can be used to both send and receive packets over the Internet.
One main example is the live coverage of the TV channels where UDP is favored over TCP [18]. TCP is a reliable/consistent protocol that enhances its own overhead during the broadcast. Another example of UDP preference is in online multiplayer gaming (i.e. calls of duty or counter strikes) whereby it is not essential to transmit all the data but the important portion. UDP makes it beneficial for certain senders where the application may tolerate information loss. [19]
Blow, some set-ups for UDP use are presented:

- In a real-time application where data transfer rate and latency are important and dropped or out-of-order received packets can be afforded.
- In transaction protocols including Network Time Protocol.
- When multiple clients are connected and real-time error correction is not required (i.e. gaming, video conferencing, media streaming).

Following are the applications of UDP:
This study has established two systematic/analytical models for appraising the performance of TCP and UDP flows over SDN where the control and data planes were

jointly measured and the system was epitomized as the four-dimensional state. The steady-state possibilities of both systems [20] (TCP SDN and UDP SDN) were calculated and performance metrics were derived.

1. Lossless data transmission
UDP is used in applications where data transmission is lossless. An application is configured to retransmit the lost packets and arrange the packets at the application level. As per the OSI model, the UDP is available at Layer 4, the transport layer. The UDP typically works with other higher-level protocols to support data transmission services [21]. These protocols include File Transfer Protocol (FTP) or Real-Time Streaming Protocol (RTSP), or Simple Network Management Protocol (SNMP). UDP preserves datagram boundaries between the sender and the receiver. If the buffer is too small, the datagram will be truncated and if too large, only one datagram is returned, and the remaining buffer space is untouched [22].

2. Gaming, voice, and video
UDP is an advantageous protocol in gaming or audio/video communications as such network applications perceive latency as critical. When forward error correction is required, the UDP protocol improves the quality of audio/video in case of packet loss [23].

3. Services that don't need fixed packet transmission
The UDP with its acknowledgment mechanism is useful in applications that need reliable information exchange and are not bound to fixed patterns nor require guaranteed correctness of the sent data. The application decides which response is correct and which is not [24-25].

4. Multicasting and routing update protocols
In multicasting, UDP supports packet switching and is essential when routing updates including Routing Information Protocol (RIP) are required.

5. Fast applications
UDP is vital in applications where speed is a more important factor than reliability.

2.4 Security and ethical implications of protocols and why TCP might be a better solution for data transmission

TCP is widely used for major internet connectivity as it breaks large data into small chunks or individual packets. These packets are checked, the lost ones are resent, and all the data packets are reassembled in a sequence at the receiving end. Compared to

TCP, UDP is a connectionless protocol as it does not need handshaking before any data transfer between sender and receiver. In UDP, packets are routed on different paths between sender and receiver since some lost packets are not received in sequential order.

Summarizing the major characteristics, UDP:

- Is a connectionless protocol.
- Is used for video streaming, Voice over IP (VOIP), gaming applications, and live broadcasts.
- Is faster and requires fewer resources as opposed to TCP.
- May involve out-of-order packets.
- Permits missing packets.
- Is desirable when a fast transmission is required.

Conversely, TCP:

- Is a connection-based protocol.
- The most widely used protocol on the internet.
- Avoids missed packets and ensures the full reception of the data at the other end.
- Facilitates in-order data.
- Is slower than UDP and needs more resources.
- Has a larger header than UDP.
- Is desirable when reliability is favored over speed or transmission time.

2.5 Wireshark and its use in LANs and WANs as a monitoring tool

Wireshark is a widely used leading network traffic analyzer tool applied for security or system administration. Alike “tcdump”, Wireshark” the free package sniffer offers a user-friendly border with filtering features and provides support in capturing packages in “tcpdump”. It can accumulate all packets in a TCP discussion to display the ASCII information/data [26].
Wireshark helps in solving:

- Dropped packets
- Latency issues
- Malicious activity on the network.

What does Wireshark do?
Wireshark is responsible for intercepting trafficking and converts the binary data into a human-readable format supporting over two thousand network protocols. Analyzing IP packets with Wireshark is useful for most security professionals [26]. The packets in a

network are typically TCP, UDP, or ICMP. Wireshark's tools allow filtering the traffic that passes through a business network. This data can be large in volume [27].
The majority of the sniffers are for free and offer high performance, yet some profitable "sniffer" products provide more material and full support through the more accessible interface and media support. Nonetheless, the price is minimal compared to other network monitoring devices). "LAWNWATCH" by Sandstorm Enterprises [LANWATCH06] (i.e. about $200) provides software sniffers with analysis structures and protocols including NETWARE, APPLE TALK, SNA, and VINES, AND NETBIOS, etc. The dominance of mobile computers has moved the beneficial/commercial "sniffers" towards wireless networks while few free applications for wireless systems currently exist. "Wireshark" provides free sniffers for wired networks that offer the goods/products" AirPcap." AirPcap is a USB 2.0 device wireless capture converter for window systems that empowers wireless capture through Wireshark supporting WALAN 802.11 b/g. [28-29]
The network outbreaks may be classified into three main classes:

- Unauthorized access to the resources and data through the network.
- Unauthorized management of data on the network.
- Attacks that lead to service transfer distraction are also called Service Denial.

In the first two, defining an authorized or an unauthorized act is the concern of a network security policy that can be well-defined as an effort by the user to change unacceptable data. Unauthorized access may be the most usual attack on any network.
Network security is a vast term that defines:

- Hardware solutions
- Software solutions
- Process or rules
- Network configurations, accessibility, and threat protection

The network security includes access control, antivirus software, access control, app security, network analysis, encryption, and VPN [30].
Modern network security protections are:

- Firewall
- Network Segmentation
- Access control
- Zero Trust Network Access (ZTNA)
- Remote Access VPN
- Intrusion Prevention Systems (IPS)

Network encryption guarantees that penetrating information conveyed over any computer network is fully encrypted. Confidential information is encrypted end-to-end when conveyed over any of the computer networks. Encryption or cryptography is an important part of network security; an old technique of securing data using codes. This

is used to solely transfer and communicate coded messages between trusted parties, and unauthorized parties cannot read or decode the messages. [31-32].

The current techniques to secure the network:

To secure data/information, advanced mathematical techniques and algorithms are used to ensure confidentiality, integrity, and verification in information/data transmission.

- While cloud technologies and large data applications provide organizations with numerous advantages and allow public authorities and sectors to run effectively, digital procedures, the data connection among data centers and commercial networks are often inadequately protected against manipulation.
- Electrical and optical information lines may still be interrupted with little determination.
- Network encryption includes full data and messages transmitted over a computer network.
- The encryption takes place mostly on layer- 2 (Ethernet Protocol, data link layer) of the OSI models. In encryption measures for IP-based networks, the encryption is carried out through the Internet Protocol (IPsec) encryption mechanism over wide zone networks at layer-3 (protocol: TCP/IP, network layer).

References:

[1] Baeldung, “A Guide to Java Sockets, February 12, 2021”, [Online].

Available:https://www.baeldung.com/a-guide-to-java-sockets [Accessed 10, October 2021].

[2] Balog, P. (2019). Internet integration von Embedded Systems. [Accessed 23, October 2021].

[3] Bozakov, Z., & Rizk, A. (2013, October). Taming SDN controllers in heterogeneous hardware environments. In *2013 Second European Workshop on Software Defined Networks* (pp. 50-55). IEEE. 10.1109/EWSDN.2013.15

[4] CDW, “Types of Network Protocols: The Ultimate Guide, October 2021”, [Online]. Available: https://www.cdw.com/content/cdw/en/articles/networking/types-of-network-protocols.html [Accessed 12, October 2021].

[5] CDW, “Types of Network Protocols: The Ultimate Guide, October 2021”, [Online]. Available: https://www.cdw.com/content/cdw/en/articles/networking/types-of-network-protocols.html [Accessed 12, October 2021].

[6] CDW, “Types of Network Protocols: The Ultimate Guide, October 2021”, [Online]. Available :https://www.cdw.com/content/cdw/en/articles/networking/types-of-network-protocols.html [Accessed 12, October 2021].

[7] Compari Tech, “A guide to UDP (User Datagram Protocol), January 8, 2019”, [Online]. Available :https://www.comparitech.com/net-admin/guide-udp-user-datagram-protocol/ [Accessed 15, October 2021].

[8] CSO, “5 must-have features in a modern network security architecture, July 8, 2019”, [Online]. Available: https://www.csoonline.com/article/3406475/must-have-features-in-a-modern-network-security-architecture.html

[9] CSO, “What is Wireshark? What this essential troubleshooting tool does and how to use it, SEP 17, 2018”, [Online]. Available: https://www.csoonline.com/article/3305805/what-is-wireshark-what-this-essential-troubleshooting-tool-does-and-how-to-use-it.html [Accessed 18, October 2021].

[10] Dastres, R., & Soori, M. (2020). Secure Socket Layer (SSL) in the Network and Web Security. *International Journal of Computer and Information Engineering*, *14*(10), 330-333. Available: https://hal.archives-ouvertes.fr/hal-03128076/document [Accessed 11, Feb 2021]

[11] E. Alothali, N. Zaki, E. A. Mohamed and H. Alashwal, "Detecting Social Bots on Twitter: A Literature Review," *2018 International Conference on Innovations in Information Technology (IIT)*, 2018, pp. 175-180, doi: 10.1109/INNOVATIONS.2018.8605995.

[12] F. Tang, Y. Kawamoto, N. Kato and J. Liu, "Future Intelligent and Secure Vehicular Network Toward 6G: Machine-Learning Approaches," in *Proceedings of the IEEE*, vol. 108, no. 2, pp. 292-307, Feb. 2020, doi: 10.1109/JPROC.2019.2954595.

[13] Force Point, “What is Network Security?”, [Online]. Available: https://www.forcepoint.com/cyber-edu/network-security [Accessed 22, October 2021].

[14] Geeks for Geeks, “Introduction to Wireshark, July 26, 2019”, [Online]. Available: https://www.geeksforgeeks.org/introduction-to-wireshark/ [Accessed 19, October 2021].

[15] KHANAPI ABD GHANI, Mohd et al. (2018) The Design of Flexible Telemedicine Framework for Healthcare Big Data. International Journal of Engineering & Technology, v. 7, n. 3.20, p. 461-468, doi: Available: http://dx.doi.org/10.14419/ijet.v7i3.20.20590

[16] Kumar, P. (2020). QUIC (Quick UDP Internet Connections) --A Quick Study. *arXiv preprint arXiv:2010.03059*. Available: arXiv:2010.03059

[17] Loyola Marymount University, “UDP” [Online]. Available: https://cs.lmu.edu/~ray/notes/udp/ [Accessed 10, October 2021].

[18] Loyola Marymount University, “UDP” [Online]. Available: https://cs.lmu.edu/~ray/notes/udp/ [Accessed 10, October 2021].

[19] Lu, Y., Chen, G., Li, B., Tan, K., Xiong, Y., Cheng, P., ... & Moscibroda, T. (2018). Multi-path transport for {RDMA} in datacenters. In *15th {USENIX} Symposium on Networked Systems Design and Implementation ({NSDI} 18)* (pp. 357-371).

[20] Mastrangelo, L., Ponzanelli, L., Mocci, A., Lanza, M., Hauswirth, M., & Nystrom, N. (2015). Use at your own risk: the Java unsafe API in the wild. *ACM Sigplan Notices*, *50*(10), 695-710. Available: doi.org/10.1145/2858965.2814313

[21] Matveeva, N. (2019). Using Neural Networks programming on Java for solving the problem of signal recognition. *Системні технології*, *1*(120), 124-131.

[22] Mvorganizing.org, “Why do some packets have sequence numbers? May 31, 2021”, [Online]. Available: https://www.mvorganizing.org/why-do-some-packets-have-sequence-numbers/ [Accessed 11, October 2021].

[23] Mvorganizing.org, “Why do some packets have sequence numbers? May 31, 2021”, [Online]. Available: https://www.mvorganizing.org/why-do-some-packets-have-sequence-numbers/ [Accessed 11, October 2021].

[24] Net Wrix, “Network Security Best Practices”, [Online]. Available: https://www.netwrix.com/network_security_best_practices.html [Accessed 23, October 2021].

[25] Security Trails, “Top 10 Common Network Security Threats Explained, April 7, 2021”, [Online]. Available: https://securitytrails.com/blog/top-10-common-network-security-threats-explained [Accessed 20, October 2021].

[26] Security Trails, “Top 10 Common Network Security Threats Explained, April 7, 2021”, [Online]. Available: https://securitytrails.com/blog/top-10-common-network-security-threats-explained [Accessed 20, October 2021].

[27] Shang, W., Yu, Y., Droms, R., & Zhang, L. (2016). Challenges in IoT networking via TCP/IP architecture. *NDN Project*. NDN Technical Report NDN-0038, 2016. Available: http://named-data.net/techreports.html

[28] Tech Target, “User Datagram Protocol (UDP), October 2021”, [Online]. Available: https://www.techtarget.com/searchnetworking/definition/UDP-User-Datagram-Protocol [Accessed 12, October 2021].

[29] Tech Target, “User Datagram Protocol (UDP), October 2021”, [Online]. Available: https://www.techtarget.com/searchnetworking/definition/UDP-User-Datagram-Protocol [Accessed 12, October 2021].

[30] Wire Shark, [Online]. Available :https://www.wireshark.org/ [Accessed 16, October 2021].

[31] Wire Shark, [Online]. Available:https://www.wireshark.org/ [Accessed 16, October 2021].

[32] Wu, R. Q., Jie, H., & Ding, N. (2014). An Improved TCP Congestion Control Algorithm of Based on Bandwidth Estimation in Heterogeneous Networks. *J. Commun.*, *9*(10), 792-797. 10.12720/jcm.9.10.792-797