Word count = 1998

Section 2. Evaluation of Solution Feedback

Section (2.1): An overview of your solution - a discussion of how your design and the selected constructs support the functional requirements.

I implemented the stop-and-wait protocol and the go-back-N protocol.

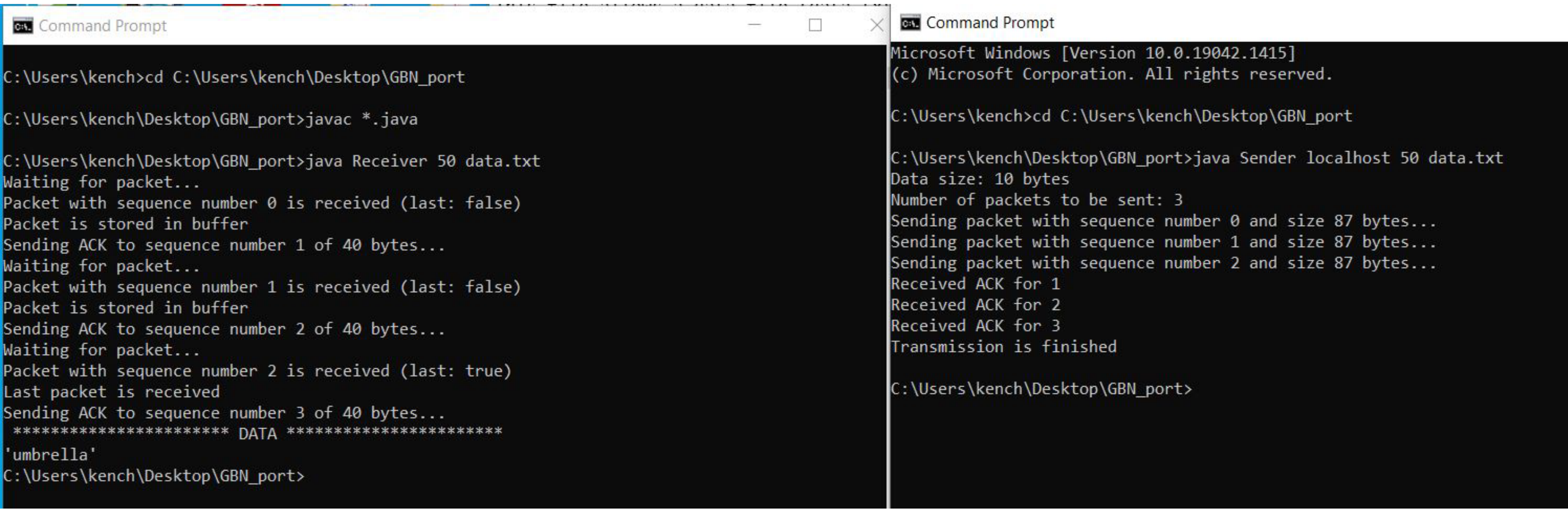
There are 8 files in the zipped folder "GBN_submit":
readme.txt: the readme document containing information about how to compile and run the code.
Sender.java: the sender (client) side of the client server system
Receiver.java: the receiver (server) side of the client server system
Serializer.java: there are: toBytes (to convert an object to byte array) and toObject (to convert a byte array to an object)
RDTAck.java: the file deals with the methods such as getPacket()
RDTPacket.java: the file deals with current sequence number, the data and a Boolean (last) (to determine whether the packet is the last one)
Data.txt: the file to be sent between the client/server should contain the string 'umbrella'
Capture.JPG: the picture showing the interaction between the sender and receiver

Principle of the solution

The main difference between go-back-N and stop-and-wait protocol is that the window size of the sender of stop-and-wait is 1, but the window size of the sender of stop-and-wait is larger than 1. Both go-back-N and stop-and-wait protocols have window size of 1 for receiver. In other words, stop-and-wait is go-back-N when the window size of the sender is 1.

After reading the data from the text file, we run the "whilefile send" two times in the sender side. The window size of the first and second time of the "whilefile send" are 1 and 5 respectively, which corresponds to stop-and-wait with sliding window size=1 and go-back-N with sliding window size=5.

Section (2.2): An overview of the program design, and how the main components support client and server interactions and communications, Evaluation of Solution Feedback



```
C:\Users\kench>cd C:\Users\kench\Desktop\GBN_port

C:\Users\kench\Desktop\GBN_port>javac *.java

C:\Users\kench\Desktop\GBN_port>java Receiver 50 data.txt
Waiting for packet...
Packet with sequence number 0 is received (last: false)
Packet is stored in buffer
Sending ACK to sequence number 1 of 40 bytes...
Waiting for packet...
Packet with sequence number 1 is received (last: false)
Packet is stored in buffer
Sending ACK to sequence number 2 of 40 bytes...
Waiting for packet...
Packet with sequence number 2 is received (last: true)
Last packet is received
Sending ACK to sequence number 3 of 40 bytes...
 ********************** DATA **********************
'umbrella'
C:\Users\kench\Desktop\GBN_port>
```

```
Microsoft Windows [Version 10.0.19042.1415]
(c) Microsoft Corporation. All rights reserved.

C:\Users\kench>cd C:\Users\kench\Desktop\GBN_port

C:\Users\kench\Desktop\GBN_port>java Sender localhost 50 data.txt
Data size: 10 bytes
Number of packets to be sent: 3
Sending packet with sequence number 0 and size 87 bytes...
Sending packet with sequence number 1 and size 87 bytes...
Sending packet with sequence number 2 and size 87 bytes...
Received ACK for 1
Received ACK for 2
Received ACK for 3
Transmission is finished

C:\Users\kench\Desktop\GBN_port>
```

Figure 1: client and server interactions and communications

Figure 1 illustrates the client and server interactions and communications. The method applied is appropriate for the data size and data type in the exercises. The data size is 10 bytes and data type is string. The string data in the file is read using an InputStream and byte array. UDP (User Datagram Protocol) is used to transfer large files over the Internet. UDP can be used because the size of the data being transferred is small; the UDP protocol breaks up a large set of data into individual packets that are then sent over the Internet. The packets hop from node to node, eventually arriving at their destination. When the packet arrives, the data set is reconstructed and the file is restored to its original state. [1]

The exercises/code/task can be achieved with a more effective method, that is, to use C++ as a language for a server-client application that use UDP. In general, C++ programs are always faster than the equivalent Java programs because C++ code is already native to the operating system, whereas Java is compiled into bytecode. [2]

Section (2.3): Briefly discuss the suitability of Java for demonstrating port and socket communications over UDP.
Java is suitable for showing port and socket communications over UDP. Java provides unreliable datagram communication for UDP [3]

The code relies on the Java programming environment to provide a true object-oriented message system. Therefore, the implementation in Java that provides reliability and security is based on a collection of objects that implement key interfaces. This allows the data abstraction and encapsulation of the message objects and message constructor/processor objects. [4]

Java is a robust, object-oriented programming language that is platform independent. The Java programming language has become very popular recently. Java provides many benefits to distributed systems. These benefits include platform independence, object-oriented programming paradigm, dynamic class loading, elimination of pointers, security and multithreading. [5]

Section 3. Evaluation of networking protocols, tools and security: Why, when using UDP, you need to use sequence numbers?

Section (3.1): Evaluation of UDP with other protocols

Section (3.1a): Discuss the reliability and security considerations when transmitting data over UDP.

UDP is an unreliable protocol. In computer networks, a reliable protocol is a communication protocol that informs the sender whether data has successfully reached the intended recipient. [1] UDP does not support the forwarding of streams, but sends blocks of messages (called datagrams) in any order, with no guarantee that they will be received UDP does not start a session with a handshake. Handshaking allows the two endpoints communicating to prove ownership of the IP, or at least the ability to read whatever is sent to the claimed IP. There is no concept of a session in UDP. It simply accepts any packet received from a given IP. This means that an out-of-the-way attacker can send packets that appear to come from any IP, effectively spoofing the IP of an endpoint or anyone else on the network. Malicious actors on the Internet can exploit this to launch amplification attacks, also known as denial of service attacks. [6]

However, there are a number of encryption standards available for UDP. The main option for direct UDP security is the Datagram Transport Layer Security Protocol or DTLS.

DTLS is available in many free open source libraries, so there is no need to look up protocol definitions or write open programs to implement it. The open source library OpenSSL is the most common source for implementing transport layer security and is the most widely implemented security system for TCP. This library also contains an implementation of DTLS, so you should be able to encounter secure UDP options in applications that provide secure TCP connections. [7]

Section (3.1b): Evaluate its effectiveness for certain situations; an example might be gaming.

Isochronous data (e.g. multimedia or video playback) becomes obsolete as soon as the time limit expires, so there is no advantage in retransmitting it to ensure reliability. [3] In many cases, interpolation or statistical algorithms are used to predict the missing parts, which is very effective in hiding intermittently missing frames. This makes it very suitable for use in multicast. For example, VoD.

UDP offers the following advantages for different types of applications [8]:

1. no retransmission delay - UDP is suitable for time-sensitive applications where retransmission delay due to packet dropping cannot be tolerated. Examples include VoIP (Voice over IP), online gaming and media streaming.
2. speed - UDP's speed is suitable for query response protocols such as DNS, where data packets are small and transactions are required.
3. suitable for broadcast - UDP does not involve end-to-end communication, it is suitable for broadcast, where outgoing data packets are addressed so that they can be received by all devices on the Internet. It can be received by many clients.

Section (3.2): Discuss whether TCP is more suitable for data transmission on both a LAN and the Internet, and give examples

TCP is a connection-oriented network and is more reliable than UDP because it ensures that the packets it sends reach their destination. UDP sends only one datagram and does not handle retransmissions, sequencing or connections. In terms of error checking mechanisms, TCP provides extensive error checking mechanisms. This is because it provides data flow control and acknowledgement. UDP, however, has only one basic error checking mechanism, namely checksum. In terms of handshaking techniques, TCP uses handshakes such as SYN, ACK and SYN-ACK. UDP, however, is a connectionless protocol and has no handshake capability. TCP is very reliable and is used for everything from browsing the internet (HTTP) to sending email (SMTP) to transferring files (FTP). TCP is used when you want all data sent from one device to be received in its entirety by another device. [9]

There are several advantages to using TCP over UDP [1]:
1. TCP ensures that packets arrive at their destination without duplication and that the data is in the same order.
2. Data transmission over TCP is more reliable than data transmission over UDP

TCP has many real-life applications, for example, email. For example, in e-mail, if a packet (word or sentence) is missing, its content cannot be understood. TCP applications include the World Wide Web (HTTP), e-mail (SMTP TCP), File Transfer Protocol (FTP) and Secure Shell (SSH). [10]

There are other protocols such as SCTP. [11] In general, SCTP can provide greater flexibility for certain applications that require reliable, message-oriented data transfer, such as Voice over IP (VoIP). For these types of applications, SCTP may be more suitable than TCP or UDP. TCP is reliable and provides strictly ordered data transfer. For applications that require reliability but allow for unordered or partial data transfer, TCP may introduce unnecessary delays due to head-of-line blocking. SCTP provides logical separation of different data streams due to its concept of having multiple data streams within a single connection. The concept of multiple streams within a single connection allows SCTP to provide strictly ordered delivery within a single stream while providing logical separation of data from different streams.

Section (3.3): Evaluate Wireshark and its use in LANs and WANs as a monitoring tool.
Wireshark has more benefits than drawbacks. The main usage of the Wireshark program is to detect the extent to which a system is susceptible to security vulnerabilities. Wireshark has the following advantages:

1. Easy to use and well-documented. Firstly, the program can be downloaded for free from the website. Secondly, there are numerous manuals available on the Internet that disclose the mechanics of operating the software product and publications describing how to optimise Wireshark.

2. this program supports many protocols: ARP [12], IP, TCP, UDP, DCCP, HTTP, HTTP2, FTP, ATM, etc.

3. This application allows you to capture packets from the network, record the data online and analyse it offline. The functionality of this application allows you to prevent possible intrusions and react quickly. [12].

An important feature of this program is its ability to be optimised. This is because many other libpcap-based analysers are also optimisable, including
Wireshark has a weakness in that it loses packets at gigabit Ethernet speeds. To address this weakness, it is possible to increase the amount of buffer memory at the kernel level [13] and make it multi-threaded. This would reduce packet loss, support buffering regardless of libpcap, speed up application response times, and generally improve interception performance. wireshark's optimizations are a good alternative to expensive commercial alternatives such as Viavi and Observer Gigastor.

On top of this, the Wireshark program allows you to identify security flaws in your system at the user authentication level. You can easily capture unprotected packets that have been sent to a webcam [14] communication between the webcam [3] and the server, thus detecting possible use of backdoors [15],[16].

HTTP POST requests have user credentials. This packet is easily detected by Wireshark. If the application tracks the TCP stream options, the entire content can be shown. As a result, you can easily find your username - email and password - in plain text [15],[16].

Section (3.4): Critical Evaluation of Security and Ethics: Discussion on modern security over LANs and the Internet, including modern encryption methods

Cyber security involves important ethical issues:

1. Harm to privacy [17]:

   The most common cyber threats to privacy include identity theft, where personal identity information is stolen for illegal purposes, such as impersonating a victim in a financial transaction (e.g. taking out a loan in the victim's name or using a credit card to make unauthorised purchases) or providing a stolen identity to criminals. These include Hacking and other network intrusions may be used to obtain sensitive information about individuals and their activities, which may be used for extortion, blackmail, or to manipulate people's will in unethical ways. Such invasions of privacy are often used to induce victims to harm the interests of third parties. Threats may be used to pressure compromised employees to sell confidential customer information, trade secrets, or to engage in corporate or government fraud. The risk of privacy breaches due to unethical cyber security measures is further magnified by the continued growth of a chaotic global data ecosystem where most people have little or no ability to personally manage or control the storage or disclosure of their personal information.

2. harm to property [18]:

   It is noted that breaches of data privacy can indirectly threaten property through mechanisms such as extortion. In many cases, property can be directly targeted through cyber intrusions designed to misuse electronic funds.

   They can steal valuable intellectual property, such as trade secrets, obtain bank account numbers and passwords, or cause remote damage or destruction to a person's or organisation's digital or physical property. The motivation for such damage may vary, and property may be targeted by profit-seeking criminal enterprises, politically motivated non-state actors, agents of corporate espionage, hostile foreign military or intelligence agents, or the aggressive impulses of lone hackers or groups seeking to demonstrate their destructive power. Property harms those depending on that property to secure a good life. Rarely, it can be argued that unauthorised damage to property is morally justified because of a higher moral obligation, such as the interests of national security. For example, by agents of the state that used the Stuxnet worm to cripple Iran's centrifuges used to enrich uranium in 2010. [19]

Reference

[1] J. Kurose and K. Ross, “Computer Networks: A Top Down Approach Featuring the Internet,” Computer Networking, Pearson Education, 2005.

[2] L. Gherardi, D. Brugali, and D. Comotti, “A java vs. c++; performance evaluation: A 3d modeling benchmark,” in Proc. of the Third Int. Conf. on Simulation, Modeling, and Programming for Autonomous Robots, ser. SIMPAR’12. SpringerVerlag, 2012, pp. 161–172.

[3] TCP/UDP Communication in Java. [Online]. Available at: https://link.springer.com/chapter/10.1007%2F0-387-23840-9_6 [Accessed: 30 Oct 2021]

[4] E. R. Harold, Java Network Progranuning: O’Reilly & Assoc., 1997

[5] H. Schildt, Java The Complete Reference, Seventh Edition. McGraw-Hill Publishing, 2006.

[6] C.-H. J. Wu and J. D. Irwin, Introduction to Computer Networks and Cybersecurity. Boca Raton, FL: CRC Press, 2013.

[7] A guide to UDP (User Datagram Protocol) [Online]. Available at: https://www.comparitech.com/net-admin/guide-udp-user-datagram-protocol/#UDP_vs_TCP [Accessed: 09 Jan 2022]

[8] User datagram protocol (UDP) [Online]. Available at: https://www.imperva.com/learn/ddos/udp-user-datagram-protocol/ [Accessed: 30 Oct 2021]

[9] Differences between TCP and UDP]. Available at: https://www.geeksforgeeks.org/differences-between-tcp-and-udp/ [Accessed: 09 Jan 2022]

[10] Kevin R. Fall, W. Richard Stevens, "TCP/IP Illustrated, Volume 1 Second Edition," Addison-Wesley, pp. 729-730, 2012.

[11] Stream Control Transmission Protocol. Available at: https://www.ibm.com/docs/en/aix/7.1?topic=protocol-stream-control-transmission [Accessed: 09 Jan 2022]

[12] Shaoqiang Wang, DongSheng Xu and ShiLiang Yan, "Analysis and application of Wireshark in TCP/IP protocol teaching," 2010 International Conference on E-Health Networking Digital Ecosystems and Technologies (EDT), 2010, pp. 269-272, doi: 10.1109/EDT.2010.5496372.

[13] A. Dabir and A. Matrawy, "Bottleneck Analysis of Traffic Monitoring using Wireshark," 2007 Innovations in Information Technologies (IIT), 2007, pp. 158-162, doi: 10.1109/IIT.2007.4430446.

[14] R. Das and G. Tuna, "Packet tracing and analysis of network cameras with Wireshark," 2017 5th International Symposium on Digital Forensic and Security (ISDFS), 2017, pp. 1-6, doi: 10.1109/ISDFS.2017.7916510.

[15] S. Sandhya, S. Purkayastha, E. Joshua and A. Deep, "Assessment of website security by penetration testing using Wireshark," 2017 4th International Conference on Advanced Computing and Communication Systems (ICACCS), 2017, pp. 1-4, doi: 10.1109/ICACCS.2017.8014711.

[16] P. Navabud and C. -L. Chen, "Analyzing the Web Mail Using Wireshark," 2018 14th International Conference on Natural Computation, Fuzzy Systems and Knowledge Discovery (ICNC-FSKD), 2018, pp. 1237-1239, doi: 10.1109/FSKD.2018.8686871.

[17] M. Manjikian, “Cybersecurity Ethics: An Introduction First Edition”, Routledge, 2017, https://doi.org/10.4324/9781315196275

[18] R. Spinello, “Cyberethics: Morality and Law in Cyberspace Fifth Edition”, Jones & Bartlett, 2014.

[19] S. Karnouskos, "Stuxnet worm impact on industrial cyber-physical system security," IECON 2011 - 37th Annual Conference of the IEEE Industrial Electronics Society, 2011, pp. 4490-4494, doi: 10.1109/IECON.2011.6120048.