

Department of Computer Science

Computer and Mobile Networks

I. Module Learning Outcomes

1. Critically analyse the core concepts in modern computer networks such as LANs and WANs, network architecture, communication protocols and their design principles, the layered organisation of computer networks, and mobile networks;
2. Apply network concepts and design principles, design/communicate and implement a networked application;
3. Critically evaluate and apply tools for computer network performance analysis;
4. Critically evaluate network security techniques;
5. Critically evaluate the legal and ethical impact of computer networks and Internet.

II. Assessment Background/Scenario

For devices to communicate on a WAN or distributed system, protocols need to be established. The OSI and TCP architectures and suites of protocols enable all devices to communicate with each other using a variety of protocols and services. TCP and UDP are the two packet types used on an 802.x network.

Given this, the aim is to create a simple client server system to demonstrate (via output statements): the communications between the client and server and the actions within each. Your system should include the following functionality:

Functional Requirements

- Your program must set and close sessions of communication
- Write a simple STOP and WAIT protocol for UDP
- In the packet, the byte payload will include both the original data and a sequence number for the packet (SN=x)

- The data in the file to be sent between the client/server should contain the string 'umbrella'
- You need to identify and include UDP ports for the communication endpoints
- Your client must take an IP address or Hostname and port number as arguments
- The port number is the port number the server must listen on
- The sender must read the line of data from the file and combine that text with the sequence number and send the packet to the receiver.

Outputs

Your system should output to the command console clear statements to demonstrate the interactions and functionality.

Non-Functional Requirements

Your code must be commented throughout

Best practice should also be applied to your code.

Technical Requirements

The application is built using Java (Liberica JDK 11)

The application runs from the Command Console

III. Assessment Tasks

Task 1 (20% weighting) - Programme

Given the scenario above, you are going to design and code a program to communicate between two devices – a client and a server (sender and receiver) over UDP.

1. Write the code that will act as the sender (client) and another program to act as the receiver (server) – these will send over UDP to communicate.
2. You must write the code to perform a STOP and WAIT – this will be a sliding window size = 1.
3. Once this runs, you are going to expand on your program to increase the sliding window to size=5 and use GO-BACK-N for retransmission

4. Your program should have a sequence number = SN and the data should be part of the data packet.
5. The server, upon receipt, should examine the SN and if it matches the SN it is expecting, print the data to the screen or file and transmit ACK back to the sender, remembering that the sliding window is size=5.

Task 2 (20% weighting) – Report

As part of a structured report, briefly discuss the programme you have produced, making sure you include the following:

1. An overview of your solution - a discussion of how your design and the selected constructs support the functional requirements.
2. An overview of the program design, and how the main components support client and server interactions and communications.
3. Briefly discuss the suitability of Java for demonstrating port and socket communications over UDP.

As a guideline, you should use around 500 words of your 2000-word limit for this section.

Task 3 (60% weighting) – Report

Continue your report with a discussion on “why, when using UDP, you need to use sequence numbers”, ensuring you consider the following:

- I. Discuss the reliability and security considerations when transmitting data over UDP.
- II. Evaluate its effectiveness for certain situations; an example might be gaming.
- III. You will need to do some background research and clearly state any assumptions you make.
- IV. Your discussion should be supported by appropriate references to the literature and should use accurate technical terms.

Finally, critically evaluate the following:

- I. Security and ethical implications of protocols and why TCP might be a better solution for data transmission.
- II. Wireshark and its use in LANs and WANs as a monitoring tool.
- III. You must include a discussion on modern security over LANs and the Internet, including modern encryption methods.

As a guideline, you should use around 1500 words of your 2000-word limit for this section.

IV. Deliverables

You should submit the following to the Canvas submission point:

- A zipped file containing your Java source code (not executables), in .zip format ONLY
- A separate file containing your report, in either .docx or .pdf ONLY

If you are submitting multiple files, you **must** upload all files simultaneously to ensure that they are marked as a single submission. If you want to resubmit one component of your work, you need to re-upload all other files at the same time: every submission must include **all** of the deliverables listed in the assessment brief.

NOTE: Failure to submit all the required elements will result in a grade of **ZERO**. Work NOT submitted in the requested format will **NOT** be considered.

Report Format

Your report should be no more than **2000 words in total**. For each task contributing to your report, we have provided *guidelines* above on the distribution of the word count. However, it is your choice how to use the word limit across the whole of this report.

You should also adhere to the following formatting instructions.

- You should submit a single word-processed file as **.docx** or **.pdf**. Other formats are NOT acceptable and are not accessible by your marker.
- Word counts that exceed the overall limits will not be graded. A line will be drawn at the limit as indicated above.
- You must state on the front page of your report the number of words used and this will be checked.
- The main text should be written using a consistent **sans serif font** and font size.
- All images and diagrams must be clear and viewable on the page without scaling. They should be accompanied by appropriate captions and be referred to and discussed in the main body of the text. Those that are not will NOT be considered.
- Your report should be justified OR left justified, NOT centrally justified
- You should not exceed more than 3 levels of section headings. i.e. main heading, sub-heading 1, sub-heading 2. Your title is not classed as a heading.

- All source material that is used, whether by direct quotation or not, must be acknowledged, following the IEEE referencing style. See the University of York Academic Integrity site.
- Appendices may be used but should **not exceed 5 additional pages** and all content must be referred to and discussed in the main body of the text. Those that are not will NOT be considered.
- Appendices should ONLY be used for supportive information, such as over-large figures or tables of data. They are NOT a device to incorporate material that would otherwise cause you to exceed the word limit. These are not included in the word count.
- Your reference list should come after any appendices and is not included in the word count. It should be formatted using the IEEE guidelines.

DO NOT include your examination number, student ID or your name anywhere within your report.

General presentation guidance

- Coloured text and highlighting should not be used for accessibility reasons, and ease of reading.
- Bullet points should only be used where the content is a definitive list and should be used sparingly.
- Tables should only be used for tabulated data and not as a means of structuring passages of text.
- Paragraphs must not be excessively long. “Wall of text” answers which do not contain breaks at logical points are not acceptable.
- Paragraphs should not consist of a single sentence.
- You are advised to use direct quotes minimally. They should only be used to support your points, and do not attract marks directly.
- You are expected to undertake further research to answer assessment questions. It is not sufficient to use only the module notes/content.

NOTE: Any work submitted beyond the word-count or stated limits will not be considered when grading the work.

V. Marking Criteria

Students should be able to use this as a guide to the elements they need to cover, without it giving them the answer to the task/question. Where academic quality is required there should be a separate and clear criterion for this (e.g. 10%). The criteria should be clearly linked to the module level learning outcomes (via a number). This can be at the task or element level.

MLO	Criteria	Available marks
Task 1 – Coded Solution		
	Code compiles. Code that does not compile will be marked as ZERO for Task 1 only.	Pass/Fail
1, 2	Using Java coding tool, student can demonstrate a clear understanding and application of protocols, UDP, packets and ports Appropriate code and designs are given, demonstrating the overall effectiveness of the Java program meeting the requirements of the brief	20
Task 2 – Evaluation of Solution (Report)		
1, 2	Evaluate the effectiveness of design for the application using UDP A rational argument for Java being effective in terms of ease of development. Critical evaluation demonstrated and appropriate technical language	20
Task 3 – Evaluation of Networking Protocols, Tools, and Security (Report)		
1	Evaluate the relationship of UDP with other protocols/services	10
3	Discuss whether TCP is more suitable for data transmission on both a LAN and the Internet, and give examples	10
3	Evaluate whether Wireshark is an appropriate tool in large networks as a main security/monitoring network application, giving examples.	20
4, 5	Critically evaluate the security and ethical considerations for protocols across a data sensitive network.	20
	TOTAL:	100