**Grading Guidelines:**

A right answer will get full credit when:

1. It is right (worth 25%)
2. It is right **AND** neatly presented making it easy and pleasant to read. (worth an **extra** 15%)
3. There is an **obvious and clear link** between 1) the information provided in the exercise and in class and 2) the final answer. A clear link is built by properly writing, justifying, and documenting an answer (worth an **extra** 60%).
4. Calculation mistakes will be minimally penalized (2 to 5% of full credit) while errors on units will be more heavily penalized.

**Late Submission** : as specified in the syllabus. Days counting starts one minute after the deadline.

You are welcome/encouraged to discuss exercises with other students or the instructor. But, ultimately, **personal** writing is expected.

* USE THIS FILE AS THE STARTING DOCUMENT YOU WILL TURN IN. **KEEP IN THE QUESTIONS** AND INSERT YOUR ANSWERS.
* IF USING HAND WRITING (STRONGLY DISCOURAGED), REWRITE THE QUESTIONS.
* FAILING TO FOLLOW TURN IN DIRECTIONS /GUIDELINES WILL COST A 30% PENALTY.

**Objectives of this assignment:**

* to learn independently about an important topic
* to answer questions about the independently studied topic
* to empower you: you can learn any networking topic on your own
* to learn independently new concepts

**What you need to do:**

Answer the questions and/or solve the exercises described below.

**KEEP THE GRADING GUIDELINES ABOVE TO REMEMBER THE DIRECTIONS AND HOW THE HOMEWORK IS GRADED.**

**Objective:** The objective of this assignment is to learn **independently** about *transport issues* not discussed in class. You must research and read about *delayed acknowledgements* and *Nagle's algorithm*, and then answer the questions.

**Resources**:

1. **Textbook:** Tanenbaum, Andrew S. and David J. Wetherall. *Computer Networks*.
2. Module 6 lectures
3. Your instructor (Through Piazza)
4. RFC 896
5. Wikipedia
6. Internet

Note that the textbook, Module 6 material, and your instructor are sufficient to answer all questions in this homework as well as the related self-study questions.

**Questions**: (**hint**: read all questions before answering. This will avoid you repeating or overlapping answers).

**Delayed Acknowledgements**

1) **(10 points)** What is a delayed acknowledgement?

A delayed acknowledgment is implemented by TCP, and uses the technique of improving network performance by combining together multiple acknowledgments into a single response. The acknowledgements are delayed for up to 500 msec to acquire some data on which to hitch a free ride, cutting the packet count and bandwidth usage. Delayed acknowledgements reduce the load placed on the network by the receiver, but when senders send multiple short packets the sender is still operating inefficiently.

2) **(15 points)** What is the purpose of delayed acknowledgements?

The purpose of delayed acknowledgements is to improve the packet count and the bandwidth use. Delayed acknowledgements also give the application an opportunity to update the window to send an immediate response. In particular, in the case of character-mode remote login, a delayed acknowledgement can reduce the number of segments sent by the server by a factor of 3 (acknowledgement, window update, and echo character all combined in one segment) In addition, on some large multi-user hosts, a delayed acknowledgement can substantially reduce protocol processing overhead by reducing the total number of packets to be processed.

3) **(10 points)** When using delayed acknowledgements, does it mean that EVERY acknowledgement will acknowledge at least two segments? If not, provide a scenario where an acknowledgement acknowledges only one segment.

Using delayed acknowledgments does not mean every acknowledgement will acknowledge at least two segments. For example, if the client sends a request and the response to that request takes longer than 500 msec, then the delayed acknowledgement timer will run out and the acknowledgement will be sent by itself and the response will be sent later, thus the acknowledgment does not acknowledge two segments.

**Nagle's algorithm**

4) **(15 points)** What is Nagle's algorithm? Describe it

Nagle’s algorithm uses the technique that when data comes into the sender in small pieces, send the first piece and buffer all of the rest until the first piece is acknowledged, then send all other buffered data in one TCP segment and start buffering again until the next segment is acknowledged. Thus only one short packed back be outstanding at a time. If any pieces of data are sent by the application in one round-trip time, Nagle’s algorithm will put the many pieces in one segment.

5) **(15 points)** What is the purpose of Nagle's algorithm?

The purpose of Nagle’s algorithm is to reduce the bandwidth used and reduce traffic congestion on a network by limiting the transmission of small datagrams and controlling the size of the TCP sending window. It also is used to increase the efficiency of routers by reducing the latency of the routing process. The Nagle’s algorithm effectively forces TCP into stop-and-wait behavior when it has outstanding data that has not yet been acknowledged and the small segments waiting are still smaller than the maximum segment size.

6) **(10 points)** Do delayed acknowledgements and Nagle's algorithm interfere?

Yes, because if Nagle’s algorithm is being used by the sender, the sender will queue data until an acknowledgement is received, but if the sender does not send enough data to the receiver to fill the maximum segment size and the receiver is using delayed acknowledgements, the receiver will pause until the acknowledgement delay timeout is reached, which causes a temporary deadlock. In other words, the receiver waits for data on which to piggyback an acknowledgement and the sender waits on the acknowledgement to send more data.Therefore, they shouldn’t both be used at the same time.

7) **(15 points)** Are there situations where Nagle's algorithm is not desirable?

Yes, for example in interactive games that are run over the internet when the players want a rapid stream of short update packets. If a rapid stream of update packets are desired, gathering the updates to send them in bursts can make erratic responses. This is common in networked video games with multiplayer capabilities and live video streaming that isn’t slightly delayed by default.

8) **(10 points)** Can you disable Nagle's algorithm on your machine? Describe the operating system you are using to answer this question.

Yes, in the Windows 10 operating system disabling Nagle’s Algorithm can be very helpful and can be done through these steps.

1. In the search box (under start) type ‘regedit’ and press enter to open the Windows Registry Editor.
2. Click the drop down menu until you see ‘HKEY\_LOCAL\_MACHINE\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters\Interfaces’
   1. the key may contain multiple NICs, so to disable Nagle’s Algorithm for the correct internet profile you must access the correct NIC-ID.
   2. To do this, click any interface key and locate then match the IP address with the one that is your IPv4 Address.
3. Once the correct IP address has been found, right click on the folder and select ‘New’ then ‘DWORD (32 bit) Value’ to create two DWORD values.
   1. name one of them ‘TcpAckFrequency’ (sending out packets immediately) and the other ‘TCPNoDelay’ (disabling Nagle-Algorithm)
4. Double click on both entries to modify them.
   1. to activate them, put 1 as their ‘Value data’ and click ‘OK’ to save the changes.
5. Close ‘Registry Editor’ and Reboot your computer.

If you ever want to reactivate Nagle’s Algorithm you can simply change the ‘TcpAckFrequency’ and ‘TCPNoDelay’ values to 0.