

Feedback on Lab1

Using feedback provided largely by the TAs, this note provides a list of take-aways / reminders inspired by the lab1 submissions. The items in the **first list** below will be **mandatory** for subsequent labs. Points were not deducted for most of them this time, but will be in future labs.

Mandatory Items

1. **[wireshark]** When taking screenshots of wireshark, make sure all three sub-windows are visible. The borders in between the sub-windows are draggable, so you can get it just right to fit the requirements of the report.
2. **[wireshark]** Again, when taking screenshots of wireshark, resize the wireshark window or crop the screenshot to cut off unnecessary empty space on the right side of the window. We only need where the text stops. In future labs, there may be a lot more text in the "Info" section. We only need to see the port numbers unless we tell you otherwise.
3. **[wireshark]** Start a fresh capture when you are acquiring your screenshots for the report. Follow the steps outlined in "Proper Packet Capturing in Wireshark" to perform your capture. This will also make your lives easier when you are counting packets, looking at timing, etc.
4. **[wireshark]** Just as in lab1, there are times you'll be asked to paste a screenshot of a wireshark capture on the client, and then do the same but on the server. Maybe sometimes it will be multiple hosts. Start these captures at the same time, so they are capturing the same experiment. This will again make your life easier (provides more for you to verify the proper behavior). Also, later labs inherently have some randomness to them, and your captures will be unrelated if you don't do them at the same time.
5. **[report]** Please only include the information that we ask for / is necessary to explain your answer, and nothing more. Here are some examples: extra wireshark screenshots and extra output (like logging into the terminal, navigating to the directory, compiling the java, etc...). There may be some exceptions to this, just use your critical thinking skills. The above examples were not necessary for the answer.
6. **[report]** Make sure answers in later sections match your wireshark screenshots in previous sections. For example, the times at which packets were sent / received, port numbers, IP and ethernet addresses should all match.
7. **[report]** Print two-sided. This is a **must**.
8. **[report]** Remove any unnecessary whitespace. If there are a bunch of blank lines that don't serve any purpose, remove them. These reports are supposed to be professional, they should look as neat and tight as possible.
9. **[report]** In this vein, use the pre-formatted fonts in the lab document. If something is askew, here is the recipe: (a) questions are in Book Antiqua, size 11 (b) Answers are the same as questions but italicized (c) Code sections are Courier size 10. Note this is not Courier New (d) All sections have 1.0 line spacing. If you have problem with line space when cut-and-pasting between different editors, it may be because of different CR/LF encoding that creates different paragraphs. If you check the "paragraph style" in effect, paragraphs are typically preceded and followed by additional white space, which can therefore produce an output that looks like double-space.
10. **[report/code]** Take your debug statements out of your code before you perform your final commit and paste it in your lab report. We are already looking through lots of code, so wading through a minefield of System.out.println's just means your labs take longer to grade. Debug printouts also should not be in your copied results (like parts C and D of lab1).

11. **[report/code]** Please do not change the test scripts that are provided. There may be some instances where this is justified, *e.g.*, to make the script run (better) in a slightly different environment than the one for which it was designed, but even in those cases check with the TAs first. This way, if there is indeed a need for a change, it can then be pushed to everyone's repositories and ensure that grading can still be done in a consistent fashion.
12. **[code]** Document your protocol. Look at the comments in the solution. That is around the level of detail that is expected. Someone should be able to look only at the code description, and understand how to program the protocol themselves. This was specified in the lab report in bold and all caps. When grading this, we used the following scale: -3 points, no comment at all; -2 points, a comment was left, but one that did not mention the basic commands of the protocol (get, put, remove); -1 point, this command mentioned the basic commands of the protocol, but didn't elaborate on them at all. Full points on the comment was when there were more details provided. Note that this will not be the scale for future labs, because now you know the level of detail required.

There are also things that are not mandatory, but they are recommendations to make your life easier or ensure you get the most points on the reports.

Optional Recommendations

1. **USE PIAZZA** to its fullest extent, *i.e.*, to post questions as well as see what types of questions others are posting and whether the answers that are provided might be relevant to some of the issues you are facing. There were several instances of lab1 reports where a student could not complete a question, and lost points, because of an issue that another student had run into and for which a solution was posted on Piazza.
2. You don't have to write complex code. Your life will be easier if you fulfill the assignment requirements and nothing more. We aren't trying to trick you. The development time of this first project was small, but some of the later labs require a more substantial effort.
3. Pay close attention to the instructions, both in the report and in the assignment document. The majority of points lost on the first lab were the result of not following instructions. Similarly pay close attention to your answers. Did you really paste the server wireshark window into the "paste your server window here" prompt? Convince yourself.
4. Don't forget that you can use hostnames instead of IP addresses. It is much easier to type "URB216-06" than an IP address.
5. Check to see if your answers make sense. For example, in the first lab it was very easy to compare the command the testScript was running versus the output your program returned. Did the output make sense? If so, you are more confident that your program works correctly. Also, use the questions to your benefit.

Lastly, from our initial assessment, the lab grades actually seem pretty good. Take note of our advice/standards, and keep it up!