

Design & Professional Skills — Pacman Protocol Specification Assignment: Marking Guidance

Motivation

Why are we getting you to mark each other's coursework? The aim here is that you have tried hard to write a correct, complete and unambiguous specification. That is a valuable skill for a computer scientist, but you only really learn whether a specification is sufficient when you try to implement someone else's. Thus we want you to read and critique each other's specification. Ideally your written feedback would be sufficient to help your classmate remedy a deficiency in their specification, thus you learn by trying to interpret a specification that is not your own, and they learn where their efforts fell short from the viewpoint of someone who might have to code from their implementation. This only works well though if everyone puts in effort trying to constructively critique each other's specifications.

General Guidance

As stated in the assignment sheet, the main criteria for marking are:

- Conciseness. Don't waffle. Be specific.
- Correctness. Will the protocol fail if implemented as specified?
- Unambiguous. Do you understand how to code what is specified in all cases?
- Completeness. Are some things missing?

The primary goal of a specification is that someone else should be able to code an implementation of the protocol that will interoperate with other implementations.

Broadly, you should assign marks for Correctness, Unambiguousness (is that a word?), and Completeness.

Conciseness is not a primary marking criterion, but you may deduct one mark (out of 30 total) for excessive waffle that does not contribute to an understanding of the protocol. Please do not deduct marks for lack of conciseness unless it really is excessive.

Correctness

You are assigning marks regarding whether you believe the protocol will perform the task assigned correctly. A protocol can be incorrect for many reasons. For example:

- it may omit sending information that is needed
- it may encode information in a way that loses information when that is not acceptable (sometimes losing information can be acceptable, but any protocol that deliberately does so should explain why, or you can consider this to be unintentional).
- it may fail to behave correctly under some circumstances. This is more likely with UDP-based protocols that fail to correctly handle packet loss.

Generally, you're reading to see if you can see any obvious flaws. You are not expected to need to implement the protocol to spot such flaws though.

Asssign a mark out of 10 for correctness .

Rough guidance:

- 10: I can see no error in this protocol, and it looks like it will work in all circumstances I can see.
- 7: (threshold for an A grade) It looks like there's a minor error, but it would be easily corrected.
- 5: (threshold for a C grade) Protocol contains a few errors, but most of it is there.
- 4: (throughhold for a pass) Protocol contains quite a few errors, but there's a reasonable attempt at a design.
- 0: Not enough design submitted to be able to determine correctness.

Unambiguousness

For this section, don't concern yourself with parts of the protocol that have not been specified, or whether the parts that are specified are correct. What you are attempting to do is to decide whether you could unambiguously implement the specified protocol (even if the resulting implementation has correctness errors or is incomplete).

- Do you know when each message should be sent? If it's not explicit, but is obvious, that's not a problem.
- When a message is received, do you know how to decode it using software?

- Is it clear how to separate a TCP bytestream into messages? This may not be explicitly written, but if it is obvious how to do it, such as the fixed sized messages in pong v1 identified by the type field, then that is not ambiguous.
- With a text-based encoding, could you write a parser that can tell a legal message from an illegal one?

Sometimes the specification may not be perfect, but there is an example that makes clear what was intended. In such cases, do not mark down if the example is sufficient to disambiguate.

Assign a mark out of 10 for umambiguousness. Rough guidance:

- 10: I know exactly how to decode all the messages, how to tell a legal message from an illegal message, when I should send each message, and what to do about packet loss (UDP only).
- 7: This is a good and largely unambiguous specification, but falls slightly short on one or two points. I could implement it, but I wouldn't be 100% sure I'd done it the same way someone else would.
- 5: This is an OK protocol, but I couldn't be sure how to implement it without asking quite a few clarifying questions.
- 4: I generally understand roughly what is intended, but I don't know unambiguously how to interpret most of the messages.
- 0: There's too little protocol here to determine if its unambiguous.

Completeness

Are there any gaps in what has been specified? Are all the messages specified? Is it specified when to send them (if this not explicit, but is obvious enough that you could implement a complete protocol, don't mark down for it)? If the protocol uses UDP, does it indicate how to handle packet loss for each message type that uses UDP?

Generally, completeness is easier to assess than Correctness or Unambiguity, because gaps in completeness will be more obvious.

Assign a mark out of 10 for completeness. Rough guidance:

- 10: Specification appears to be complete in every aspect. I can't think of anything else that is necessary to add to achieve interoperability.
- 7: Specification is largely complete. There are one or two omissions, but I think I could largely implement a complete protocol despite the omissions.

- 5: There are significant gaps in the specification. I could implement part of the protocol, but would need more information to complete it.
- 4: Part of the protocol is specified, but more than half is not. I could not implement most of this protocol without more information.
- 0: There's no actual specification here - I couldn't implement any of the protocol without more information.

Again, if there is a gap in completeness, but an example explains well enough that you could implement this part anyway, this may offset the missing specification.

Overall Mark

For each marking section (Correctness, Unambiguousness and Completeness), you must write a short explanation of why you allocated your mark. Normally one paragraph will be sufficient, but you can write more if you wish. Try to be specific: what precisely is missing; which message is ambiguous, and why; what is an example where the protocol would behave incorrectly?

Try to write in a constructive style that would help your classmate improve their specification. If it helps, imagine you had to stand up at the microphone in a large standards meeting to make your comment to the document author, with the aim of improving the draft standard document. If a comment would not be appropriate in such a setting, it is probably not appropriate as feedback here either.

Add together your marks out of 10 for correctness, unambiguousness, and completeness to obtain a mark out of 30.

You may amend this mark if necessary, if you think it does not reflect a fair grade for the specification.

- You may add up to 2 marks for elegance of design, if you think the design is really cute in some way, so long as this does not take the total mark above 30.
- You may subtract up to 1 mark for lack of conciseness, as discussed in the intro to this guidance. **Do not do this unless the specification contains very excessive waffle.**
- You may subtract up to 5 marks for other flaws not listed, **but you cannot do this unilaterally** - subtraction of marks needs to be clearly justified, and needs to be approved by me via a Piazza private message.

The final mark will be the median mark of the student marks for the coursework, as this is generally more robust to outliers than the mean.

If anyone feels the marks they received are not justified by the feedback, there will be a process by which this can be contested. If marks are contested, the final mark will be allocated by the course staff, but such contested marks

will involve a re-mark of the whole coursework, and so marks may move either up or down.

If any vindictive feedback is reported to us, the person who wrote that feedback may themselves have marks deducted.