

# CPSC 559 – W2021 – Project Iteration 3 Optional Requirements

## Contents

CPSC 559 – W2021 – Project Iteration 3 Optional Requirements .....	1
System Description.....	1
Requirements for a Peer Process .....	1
Group management requirements .....	1
User interface and snippets requirements ( <i>snip</i> UDP/IP messages) .....	2
Updated Registry Communication Protocol .....	2
1) Updated Report Request.....	2
Testing Your Solution .....	4
Submission .....	4
Collaboration Opportunities and Limitations.....	5

## System Description

We can improve the snippet system by adding additional failure handling for dropped *snip* messages and by improving our group management. Note that marks for optional features are counted directly to your course grade, not to your project grade. You only qualify for these grades if the required components are implemented correctly.

You can submit your solution for this on Friday April 1 or on Friday April 15 (submission date for iteration 4).

## Requirements for a Peer Process

See iterations 2 and 3 for detailed requirements. Existing communication between peer and registry stays the same as does communication between peers. Some additional messages management of peers is required to improve the fault tolerance of your peer process.

## Group management requirements

1. Maintain the status of peers in your list. If you have not received any *peer* messages from a peer in your list for 3 to 5 minutes (exact time is your choice), mark the peer as inactive due to lack of peer messages. In your report, add this status to the peer in your peer list. Also stop sending messages to this peer as soon as you mark it as inactive.
2. When you add a new peer to your list or you re-activate a peer in your peer list, send this peer all *snip* messages that you are aware of to help this peer catch up on any messages they may have missed. This does mean that you may receive the same *snip* message multiple times. Make sure that *snip* messages are not duplicated in the messages you display to the end user nor in your report to the Registry. Only report the first time you receive a snippet.

When a peer is added or re-activated, send one *catch-up* message to this peer for each snippet you are aware of. The format for catch-up messages is:

```
<catch-up snippet> ::=      ctch<original sender><space><snippet  
time><stamp><space><content>  
<original sender> ::= <ip>:<port>  
<snippet time> ::= <num>
```

The snippet time should be the snippet's timestamp expressed as an integer. ~~the number of milliseconds since Epoch. (This is the default way to store time in systems and date objects. All other date and time representations are derived from this time in milliseconds since Epoch.)~~

There should be no acks for catch-up messages.

### User interface and snippets requirements (*snip* UDP/IP messages)

On receipt of a snip message, respond to the peer you received the message from with an ack message using the following format:

```
<ack snippet> ::= ack<space><snippet timestamp>
```

Use the timestamp of the snippet your received in the ack.

The snippet *acks* you have received should be added to your report.

Remember, it is possible that you receive the same snip message multiple times. Make sure that snip messages are not duplicated in the messages you display to the end user nor in your report to the Registry.

If you do not receive an ack message within 10 seconds after sending a snip, send the snip message again. If you do not receive an ack after sending three times, mark the peer inactive due to lack of ack messages. Include this information in your peer list in your report.

### Updated Registry Communication Protocol

The report should be updated to include peer status and snippet ack messages. Note that the symbol  $\lambda$  represents the empty string.

#### 1) Updated Report Request

```
<report request> ::= "get report"<newline>  
<report response> ::= <peer list2><peer list sources><peers recd><peers  
sent><snippet list><acks recd>  
<peer list2> ::= <peer3>|<peer3><peer list2>|  $\lambda$   
<peer3> ::= <ip><colon><port><space><aliveness><newline>  
<aliveness> ::= "alive"|"silent"|"missing_ack"  
<acks recd> ::= <num><acks list>|0
```

```

<acks list> := <ack received>|<ack received><acks list>| λ
<ack received> := <timestamp><space><source peer><newline>

```

For example: the following is a possible updated response to a 'get report' request.

<i>Report</i>	<i>Explanation</i>
4 10.0.0.148:62035 <b>alive</b> 10.0.0.148:55794 <b>alive</b> 10.0.0.148:48945 <b>alive</b> 10.0.0.148:30373 <b>alive</b> 1 localhost:55921 2021-03-27 12:09:12 3 10.0.0.148:62035 10.0.0.148:48945 10.0.0.148:30373 10 10.0.0.148:48945 10.0.0.148:30373 2021-03-27 12:09:12 10.0.0.148:55794 10.0.0.148:48945 2021-03-27 12:09:12 10.0.0.148:30373 10.0.0.148:48945 2021-03-27 12:09:19 10.0.0.148:62035 10.0.0.148:62035 2021-03-27 12:09:21 10.0.0.148:48945 10.0.0.148:48945 2021-03-27 12:09:22 10.0.0.148:55794 10.0.0.148:48945 2021-03-27 12:09:22 10.0.0.148:30373 10.0.0.148:62035 2021-03-27 12:09:29 10.0.0.148:62035 10.0.0.148:48945 2021-03-27 12:09:31 10.0.0.148:48945 10.0.0.148:55794 2021-03-27 12:09:32 10.0.0.148:55794 10.0.0.148:48945 2021-03-27 12:09:32 0 7 1 test1 test message 0 10.0.0.148:48945 4 hello 10.0.0.148:30373 7 test0 test message 0 10.0.0.148:62035 11 test2 test message 0 10.0.0.148:55794 14 hello 10.0.0.148:30373 16 test0 test message 0 10.0.0.148:62035 20 test2 test message 0 10.0.0.148:55794 <b>2</b> <b>1 10.0.0.148:62035</b> <b>1 10.0.0.148:30373</b>	<p>The reporting peer has a list of 4 peers <b>note added status when can have value <i>alive</i>, <i>silent</i>, or <i>missing_ack</i>.</b></p> <p>There was one source that provided a list of peers (that would have been the registry)  The date this list was received  There were three peers in the list  The three peer that was received</p> <p>There were 10 peers that were received via a UDP</p> <p>There were 0 peer messages sent via UPD  This peer is aware of 7 snippets in the system</p> <p><b>This peer received 2 ack messages</b></p>

## Testing Your Solution

Before the iteration due date, a Registry server will be running (mostly) continuously. By default, it will be running at 136.159.5.22:12955. There will also always be at least three peer processes running that will generate random snippets and will participate in the group management as required.

The code for the Registry and the automated test peers will be provided if you want to run them locally before joining the rest of the class to test your peer process. If the registry is not running at the default location or the location last communicated on the discussion board, post a note on the discussion board and I will get the Registry restarted as quickly as possible.

The Registry that will be used to submit your solution will be the same as the one used for testing, but it will be restarted before submission starts.

For your convenience, the source code of this test registry is available on D2L.

## Submission

This third iteration is due on Friday April 1 at 3pm. Alternatively, you can submit it on Friday April 15 (for iteration 4 of the project), but you won't get a chance to resubmit.

There are three requirements for submission:

1. Class diagram of your solution to the D2L dropbox for this iteration.
2. A video that gives a brief explanation of your implementation. Make sure to pay special attention to explaining your implementation of the optional features. The length of your video should be between 5 and 10 minutes. Only the first 10 minutes of your video will be viewed when grading. Use the same D2L dropbox to either upload the video or provide a link to your video.
3. Running your peer on
  - a. Friday April 1 between 3pm and 4pm, or

Grading will be based on:

1. Required: managing dropped *stop* messages.
2. Optional: managing dropped *snip* messages. Implementing this will increase your course grade by up to 3%.
3. Optional: improving group management including improving detection of peers that left the group and helping peers that join late to catch up on lost snip messages. Implementing this will increase your course grade by up to 3%.
4. Your ability to connect with the submission registry server and your ability to send your source code and report when required and requested. If this portion is not functioning, your submission will NOT be graded.
5. The accuracy of the report that your process sends to the registry sever during shut down.
6. The quality of the code. (Design, legibility and documentation)
  - a. If no class diagram is provided, there is no marks for design.

- b. If the video does not give a brief overview of the design and organization of the code, there are no marks available for design.
- 7. The ability for an end user to send snippets and read snippets. If the video does not show this functionality clearly, no marks are available for the interface.

### Collaboration Opportunities and Limitations

You may continue with the same partner, work with a different partner, or work alone for this iteration. Each team must have a unique team name. Please contact the instructor with a preferred team name and the list of team members.

No other individuals should work directly on the code you write, except for the instructor and TAs. Do not send your entire code base to any individuals nor make it available in a public repository, a discussion board or any other forums.

You may use any other resources you find useful. You may ask questions in the course discussion boards and any public discussion boards. Do make sure you cite any sources you use and any suggestions you received on discussion boards and other forums. An example on citing code using code documentation can be found at: <https://runestone.academy/runestone/books/published/StudentCSP/CSPCreativity/referencing.html> Do make it clear exactly which of your submitted code the citation relates to.

If you find an algorithm at a source other than the materials provided for class, make sure you cite this as well in the code documentation.

If you are not sure if a certain level of help or collaboration from individuals outside the team is allowed, make sure to contact the instructor for clarification.