

Technical Challenge

Applicant:

Lee Isbell

[Lee\_Isbell@icloud.com](mailto:Lee_Isbell@icloud.com)

<https://www.linkedin.com/in/lee-isbell>

Applied Position:

Senior Integration Engineer

**Task Summary:** Write a script/program that periodically calls rippled’s server\_info command and records the sequence number of the latest validated ledger along with the current time. Record this data in a file. Then, use this data to construct a plot (time on the x-axis, sequence number on the y-axis) that visualizes how frequently the ledger sequence is incremented over time (i.e. how often new ledgers are validated). Choose a time span and polling interval that can effectively capture and depict this information.

## Introduction

The purpose of this document is to show how I tackled the challenge and ultimately reached the end result. This document should be read by those who have an interest in understanding how frequent ledger validations are performed on the XRP network. This document is aimed at a technical audience.

System pre-requisites include a Linux CentOS virtual machine using command line utilities such as cULR, JQ, Gnuplot and Bash scripts.

## How the script works

Writing a simple bash script, we can send the equivalent of a JSON-RPC request to the XRP node using cURL, whilst at the same time capturing the output from server\_info and writing to a JSON file. Incorporating JQ commands into the script we can date and timestamp each server\_info response and filter for the desired content from within the JSON file, capturing and outputting this data to a CSV file.

Using the CentOS crond utility, we can create a crond job that would execute the bash script and poll the XRP node every 5 minutes.

Sample script is shown in Appendix A.

## Polling intervals

Every few seconds, the XRP network reaches a consensus on a new set of transactions which are applied to the old state of the ledger to create a new “validated ledger” that gets broadcast across the network. One would then assume each “validated ledger” results in an incremental increase of 1 to the sequence number.

I therefore decided to poll the rippled server every 5 minutes for an hour, capturing twelve entry points in time (60/5=12), which can be plotted neatly onto a graph. This would allow the XRP network to receive a subset of new validated ledgers over a prolonged period of time, especially when consensus is reached every few seconds.

## The Results

The results show us a sequence increment of either 76 or 77 ledger validations every 5 minutes. No incremented sequences are equally consistent; the results show us the total number of validated ledgers vary over time, with a variation of +/- one ledger validation from the previous validated ledger in the given time.

We can therefore use simple math to work out that the XRP network validates roughly 15 ledger validations every 1 minute, with on average 1 ledger validated every 4 seconds. (All numbers rounded up to the nearest decimal)

* 1. Sample Data

Below table shows some sample data extracted from rippled server\_info command using the script as described in “How the script works”. An additional column was added to show the sequence difference.



Table1: CSV Sample data

* 1. Graphical Results

Using gnuplot (plotting software) based on the output result data, graphs A and B visualize how frequently the ledger sequence is incremented over a one-hour period. Time on the x-axis and sequence number on the y-axis, I showcase this using both a bar and line graph.

A screenshot of a cell phone

Description automatically generated

Graph A: How frequently the ledger sequence is incremented over time – Bar Graph

A close up of a map

Description automatically generated

Graph B: How frequently the ledger sequence is incremented over time – Line Graph

## Ledger Variations explained

Achieving consensus (all in agreement), the time taken for all participating servers on the network to agree on a group of transactions before creating a new validated ledger. This is based on a set of rules, all validating nodes must comply with these rules and agree on each transaction, this can obviously be time consuming, although relatively fast the network may not reach a consensus equal to the same time as the last. This would explain why we see a variation in the number of new ledgers being validated in the specified period of time.

## Bonus question #1

Unfortunately, I wasn’t able to get my script to work as one would have hoped. After modifying the existing output data, I was able to work out the Minimum, Maximum and Average time that it took for each ledger to be validated:

Minimum: 3.383 Seconds

Maximum: 3.926 Seconds

Average: 3.885 Seconds

## Bonus question #2

Using Public Rippled Methods - Ledger Methods - “Ledger” API call, we can retrieve information about the public ledger.

By specify a ledger\_index parameter: “Validated” (Validated = the most recent ledger that has been validated by the whole network) and using the following fields, we can calculate how long it took to close the ledger.

* ledger.close\_time – (ISO 8601)
* ledger.close\_time\_human
* ledger.close\_time\_resolution
* ledger.closed
* ledger.parent\_close\_time – (ISO 8601)

Reference: <https://xrpl.org/ledger.html#main_content_body>

## Appendix A

#!/bin/bash

service\_url="http://s1.ripple.com:51234"

cd /root/ripple

dt=$(date "+%Y%m%d%H%M")

curl -o server\_info.json -s \

-H "Accept: application/json" \

-H "Content-Type: application/json" \

-d '{"method":"server\_info","params":[{}]}' \

$service\_url

jq . server\_info.json > server\_info\_$dt.json

resp\_seq=$(jq .result.info.validated\_ledger.seq server\_info.json)

resp\_time=$(jq .result.info.time server\_info.json)

echo "$resp\_time;$resp\_seq" >> server\_info\_dist.csv