Clock Exercise

Q: How did you calculate your clients average clock drift rate?

After each successful request to the NTP server I divided the calculated clock offset by $10 + (10 * numfails)$ where *numfails* is the number of failed attempts since the last successful attempt. Since a request to the server is made every 10 seconds this gives a good approximation for the drift per second (in microseconds).

I kept an array of these values and calculated the average at each iteration. The final average was $12.032 miliseconds$

Picking the Timeout

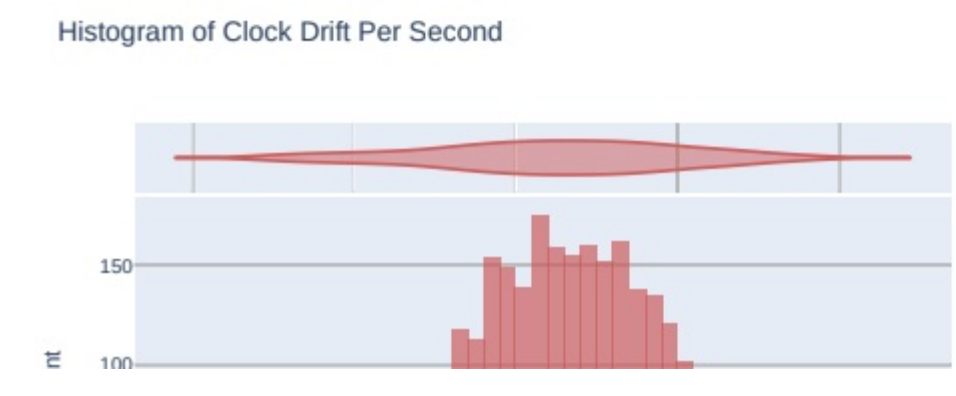
Q: what timeout did you pick to detect a failed interaction? What happens if the server's response packet arrives after that timeout

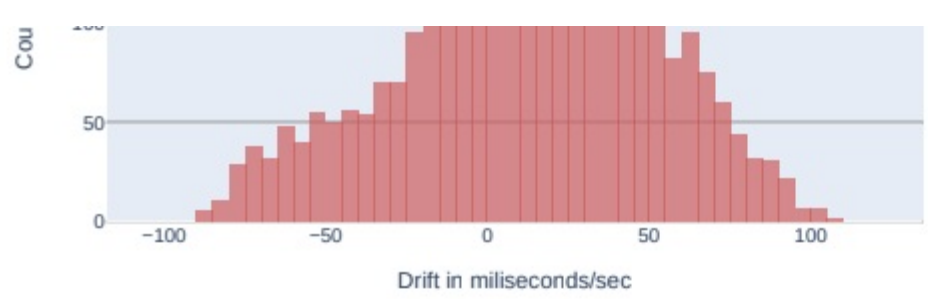
I picked **10 seconds** as the timeout. The resulting packet loss rate was 4. When a timeout occurred the failure was counted and no other stats were calculated

```
while True:
        ...
        try:
                t3 = calc_time()
                data, address = client.recvfrom( 1024 )
                t0 = calc_time()
                succs += 1
        except Exception as ex:
                losses += 1
                logger.debug(f'timeout: {losses}')
                continue
        ...
        stat = {
                'offset':off,
                'RTT':rtt,
                'smoothed_offset':smoff,
                'smoothed_RTT':smrtt,
                'drop_rate':round(losses/(succs+losses), 2)*100,
                'current_system_time':now,
                'adjusted_system_time':adjusted,
                'current_drift':drift,
                'average_drift':sum(drifts)/len(drifts),
                'average_RTT':sum(rtts)/len(rtts),
        }
        logger.debug(stat)
        time.sleep(10)
```

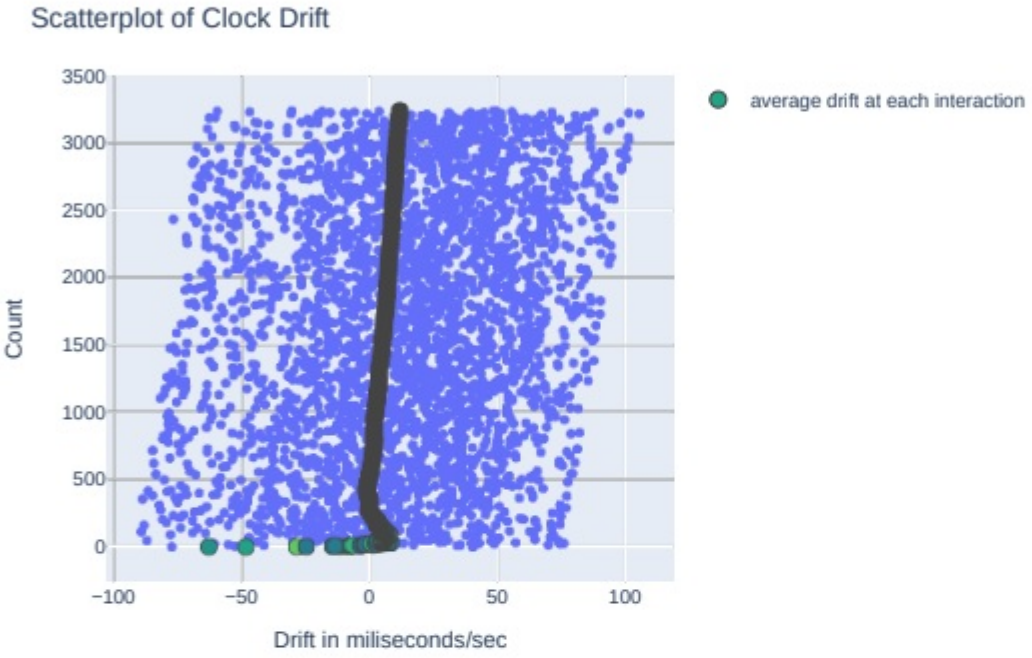
Graphing the Clock Drift

I created this histogram of the clock drift per second (in miliseconds) for each successful interaction with the server. Note that there were 3200 interactions logged





As you can see most of the time the drift was positive, meaning that my machine's clock was running faster than the NTP server's. The histogram reflects a fairly normal distribution.



The scatter plot shows that the clock drift skews slightly to the right over time, but the width (range of the values) stays mostly constant. Adding in the plot of the average drift (calculated at each interaction) you can get a better picture of the overall trend. The average slowly increases over time which means that the client machine is getting out of sync with the NTP server at a faster rate.

Python Code for the NTP client

```
import socket
import struct
import sys
import time
import logging
import math
NTP_SERVER = "0.uk.pool.ntp.org"
TIME1970 = 2208988800

logger = logging.getLogger('rtt_and_offset')
logger.setLevel(logging.DEBUG)
handler = logging.FileHandler('rtt_offset.log')
handler.setLevel(logging.DEBUG)
logger.addHandler(handler)


SOCKET_TIMEOUT = 10
NANOS = 1000000000

def calc_time():
    t = time.time()
```

```
    return (t//1, (t%1)*NANOS//1)

def diff(a, b):
    secs = b[0]-a[0]
    nanos = b[1]-a[1]
    nanos = (NANOS)*secs + nanos

    return nanos

def addnanos(it, nanos):
    newnanos, secs = it[1]+nanos, it[0]
    if newnanos >= NANOS:
        secs += 1
        newnanos -= NANOS
    return (secs, newnanos)

stats, drifts, rtts = [], [], []
succs, losses, cur_fails = 0, 0, 0
while True:
    data = '\x1b' + 47 * '\0'
    data = data.encode('utf-8')
    client = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
    client.settimeout(SOCKET_TIMEOUT)
    client.sendto( data,( NTP_SERVER, 123 ))
    try:
        t3 = calc_time()
        data, address = client.recvfrom( 1024 )
        t0 = calc_time()
        succs += 1
    except Exception as ex:
        losses += 1
        logger.debug(f'timeout: {losses}')
        continue

    resp = struct.unpack( '!12I', data )

    reference = (resp[4]-TIME1970, resp[5]%NANOS)
    originate = (resp[6]-TIME1970, resp[7]%NANOS)
    receive   = (resp[8]-TIME1970, resp[9]%NANOS)
    transmit  = (resp[10]-TIME1970, resp[11]%NANOS)

    t1, t2 = transmit, receive

    off, rtt = (diff(t3,t2) - diff(t1,t0))/2, diff(t3, t0)
    drift = off/((10+(rtt/NANOS))*(losses-cur_fails+1))
    drifts.append(drift)
    cur_fails = losses

    rtts.append(rtt)

    stats.append((rtt, off))
    if len(stats) >= 8:
        stats = stats[1:]

    smrtt, smoff = min(stats, key=lambda st: st[0])

    now = calc_time()
    adjusted = addnanos(now, smoff)

    stat = {
            'offset':off,
            'RTT':rtt,
            'smoothed_offset':smoff,
            'smoothed_RTT':smrtt,
            'drop_rate':round(losses/(succs+losses), 2)*100,
            'current_system_time':now,
            'adjusted_system_time':adjusted,
            'current_drift':drift,
            'average_drift':sum(drifts)/len(drifts),
            'average_RTT':sum(rtts)/len(rtts),
```

```
    }

    logger.debug(stat)

    time.sleep(10)
```

Python Code for Parsing the Log and Producing the Graphs

```
import re
import pandas
import numpy
import plotly.express as px
import plotly.graph_objects as go
from types import SimpleNamespace

with open('rtt_offset.log') as log:
    data = re.findall('{.*?}', log.read())

stats = [eval(it) for it in data]

avg_RTT = stats[-1]['average_RTT']/1000000000
packet_loss_rate = stats[-1]['drop_rate']
avg_drift_in_milis = stats[-1]['average_drift']/1000000


for stat in stats:
    stat['current_drift'] /= 1000000
    stat['average_drift'] /= 1000000

frame = pandas.DataFrame(stats)

fig = px.histogram(frame, x='current_drift',
        marginal = 'violin',
        title='Histogram of Clock Drift Per Second',
        labels={'current_drift':'drift_in_miliseconds_per_second', 'y':'percent of r
        opacity=0.7,
        color_discrete_sequence=['indianred'],
        hover_data=frame.columns)
fig.update_layout(xaxis_title="Drift in miliseconds/sec", yaxis_title="Count")
fig.write_html('fig1.html')

fig2 = px.scatter(frame, x='current_drift')
fig2.add_trace(go.Scattergl(
    x=frame.average_drift,
    mode='markers',
    name='average drift at each interaction',
     marker=dict(
        size=10,
        color=numpy.random.randn(1000), #set color equal to a variable
        colorscale='Viridis', # one of plotly colorscales
        line_width=1
    )
))
fig2.update_layout(title='Scatterplot of Clock Drift',
        xaxis_title="Drift in miliseconds/sec",
        yaxis_title="Count")
fig2.show()
fig2.write_html('fig2.html')
```

The Head of the Log Output

```
{'offset': -487609773.0, 'RTT': 153340816.0, 'smoothed_offset': -487609773.0, 'smoot
{'offset': -782231681.5, 'RTT': 165259122.0, 'smoothed_offset': -487609773.0, 'smoot
{'offset': 405187118.5, 'RTT': 21826267.0, 'smoothed_offset': 405187118.5, 'smoothed
{'offset': -134287537.5, 'RTT': 151166439.0, 'smoothed_offset': 405187118.5, 'smooth
{'offset': 413104544.0, 'RTT': 156281471.0, 'smoothed_offset': 405187118.5, 'smoothe
{'offset': 97151269.5, 'RTT': 162478208.0, 'smoothed_offset': 405187118.5, 'smoothed
{'offset': -524623306.5, 'RTT': 18080711.0, 'smoothed_offset': -524623306.5, 'smooth
{'offset': -65638281.0, 'RTT': 145982980.0, 'smoothed_offset': -524623306.5, 'smooth
{'offset': 414353482.0, 'RTT': 186136484.0, 'smoothed_offset': -524623306.5, 'smooth
{'offset': -34700086.0, 'RTT': 164871931.0, 'smoothed_offset': -524623306.5, 'smooth
{'offset': 345384341.0, 'RTT': 27503729.0, 'smoothed_offset': -524623306.5, 'smoothe
{'offset': -156930524.5, 'RTT': 153172493.0, 'smoothed_offset': -524623306.5, 'smoot
{'offset': -415798577.5, 'RTT': 152435779.0, 'smoothed_offset': -524623306.5, 'smoot
{'offset': 53230894.5, 'RTT': 14305830.0, 'smoothed_offset': 53230894.5, 'smoothed_R
{'offset': 363647429.0, 'RTT': 175778627.0, 'smoothed_offset': 53230894.5, 'smoothed
{'offset': 701310640.0, 'RTT': 16103983.0, 'smoothed_offset': 53230894.5, 'smoothed_
{'offset': 177379613.0, 'RTT': 152553558.0, 'smoothed_offset': 53230894.5, 'smoothed
{'offset': -152859552.5, 'RTT': 145633221.0, 'smoothed_offset': 53230894.5, 'smoothe
{'offset': -558811530.0, 'RTT': 178144931.0, 'smoothed_offset': 53230894.5, 'smoothe
{'offset': -67971236.0, 'RTT': 19111633.0, 'smoothed_offset': 53230894.5, 'smoothed_
{'offset': 88386480.5, 'RTT': 155122757.0, 'smoothed_offset': 701310640.0, 'smoothed
{'offset': 755776305.5, 'RTT': 136873484.0, 'smoothed_offset': 701310640.0, 'smoothe
{'offset': 497872607.5, 'RTT': 172153711.0, 'smoothed_offset': -67971236.0, 'smoothe
{'offset': -70138038.5, 'RTT': 14465094.0, 'smoothed_offset': -70138038.5, 'smoothed
{'offset': -612678082.0, 'RTT': 151287556.0, 'smoothed_offset': -70138038.5, 'smooth
{'offset': 33432399.0, 'RTT': 137347460.0, 'smoothed_offset': -70138038.5, 'smoothed
{'offset': 406811993.5, 'RTT': 169964552.0, 'smoothed_offset': -70138038.5, 'smoothe
{'offset': 762719560.0, 'RTT': 16429663.0, 'smoothed_offset': -70138038.5, 'smoothed
{'offset': 438755302.5, 'RTT': 186074257.0, 'smoothed_offset': -70138038.5, 'smoothe
{'offset': 138644104.0, 'RTT': 158298493.0, 'smoothed_offset': -70138038.5, 'smoothe
{'offset': -77067563.5, 'RTT': 177956820.0, 'smoothed_offset': 762719560.0, 'smoothe
{'offset': 121172430.5, 'RTT': 19437313.0, 'smoothed_offset': 762719560.0, 'smoothed
{'offset': 582050109.0, 'RTT': 161080122.0, 'smoothed_offset': 762719560.0, 'smoothe
{'offset': 280531996.0, 'RTT': 159139872.0, 'smoothed_offset': 762719560.0, 'smoothe
{'offset': 49005456.5, 'RTT': 173752308.0, 'smoothed_offset': 121172430.5, 'smoothed
{'offset': -480102159.5, 'RTT': 14315844.0, 'smoothed_offset': -480102159.5, 'smooth
{'offset': 9694540.5, 'RTT': 164414883.0, 'smoothed_offset': -480102159.5, 'smoothed
{'offset': 296811347.5, 'RTT': 158821582.0, 'smoothed_offset': -480102159.5, 'smooth
{'offset': 77841943.5, 'RTT': 180118561.0, 'smoothed_offset': -480102159.5, 'smoothe
{'offset': -436547178.0, 'RTT': 14151335.0, 'smoothed_offset': -436547178.0, 'smooth
{'offset': 21572393.0, 'RTT': 161961556.0, 'smoothed_offset': -436547178.0, 'smoothe
{'offset': -299498435.5, 'RTT': 150073528.0, 'smoothed_offset': -436547178.0, 'smoot
{'offset': 55238804.5, 'RTT': 176194430.0, 'smoothed_offset': -436547178.0, 'smoothe
{'offset': -205749929.5, 'RTT': 147642612.0, 'smoothed_offset': -436547178.0, 'smoot
{'offset': 483638346.0, 'RTT': 155375719.0, 'smoothed_offset': -436547178.0, 'smooth
{'offset': 85250723.5, 'RTT': 153871059.0, 'smoothed_offset': -436547178.0, 'smoothe
{'offset': -302849320.5, 'RTT': 150412083.0, 'smoothed_offset': -205749929.5, 'smoot
{'offset': -868528025.5, 'RTT': 20198583.0, 'smoothed_offset': -868528025.5, 'smooth
{'offset': 236866666.5, 'RTT': 146574497.0, 'smoothed_offset': -868528025.5, 'smooth
{'offset': 750359446.0, 'RTT': 157105684.0, 'smoothed_offset': -868528025.5, 'smooth
{'offset': 313121457.0, 'RTT': 152260542.0, 'smoothed_offset': -868528025.5, 'smooth
{'offset': -239663296.5, 'RTT': 19502402.0, 'smoothed_offset': -239663296.5, 'smooth
{'offset': 216086952.0, 'RTT': 159087419.0, 'smoothed_offset': -239663296.5, 'smooth
{'offset': -197002738.5, 'RTT': 157872677.0, 'smoothed_offset': -239663296.5, 'smoot
{'offset': 206487350.5, 'RTT': 146981955.0, 'smoothed_offset': -239663296.5, 'smooth
{'offset': 649291139.5, 'RTT': 14567137.0, 'smoothed_offset': 649291139.5, 'smoothed
{'offset': 185639439.5, 'RTT': 180225372.0, 'smoothed_offset': 649291139.5, 'smoothe
{'offset': -139314524.0, 'RTT': 154194832.0, 'smoothed_offset': 649291139.5, 'smooth
{'offset': -242123242.5, 'RTT': 164503336.0, 'smoothed_offset': 649291139.5, 'smooth
{'offset': 197241393.5, 'RTT': 163100958.0, 'smoothed_offset': 649291139.5, 'smoothe
{'offset': -87333994.0, 'RTT': 164438009.0, 'smoothed_offset': 649291139.5, 'smoothe
{'offset': 509138939.5, 'RTT': 163611650.0, 'smoothed_offset': 649291139.5, 'smoothe
{'offset': 179175615.5, 'RTT': 152207375.0, 'smoothed_offset': 179175615.5, 'smoothe
{'offset': -162709884.0, 'RTT': 161403180.0, 'smoothed_offset': 179175615.5, 'smooth
{'offset': 255811840.5, 'RTT': 163250923.0, 'smoothed_offset': 179175615.5, 'smoothe
{'offset': -156234651.5, 'RTT': 167948961.0, 'smoothed_offset': 179175615.5, 'smooth
{'offset': 515074420.5, 'RTT': 150323868.0, 'smoothed_offset': 515074420.5, 'smoothe
{'offset': 248508170.5, 'RTT': 168063879.0, 'smoothed_offset': 515074420.5, 'smoothe
{'offset': -8456188.5, 'RTT': 170471192.0, 'smoothed_offset': 515074420.5, 'smoothed
{'offset': 425825272.5, 'RTT': 157263041.0, 'smoothed_offset': 515074420.5, 'smoothe
```

```
{'offset': 72035808.0, 'RTT': 152787685.0, 'smoothed_offset': 515074420.5, 'smoothed
{'offset': -187373118.5, 'RTT': 163704395.0, 'smoothed_offset': 515074420.5, 'smooth
{'offset': -475859837.5, 'RTT': 166577578.0, 'smoothed_offset': 515074420.5, 'smooth
{'offset': 31350232.0, 'RTT': 161027193.0, 'smoothed_offset': 72035808.0, 'smoothed_
{'offset': 718650508.0, 'RTT': 151880264.0, 'smoothed_offset': 718650508.0, 'smoothe
{'offset': 415337125.5, 'RTT': 157495737.0, 'smoothed_offset': 718650508.0, 'smoothe
{'offset': -109994064.0, 'RTT': 15164375.0, 'smoothed_offset': -109994064.0, 'smooth
{'offset': -24046608.5, 'RTT': 14882088.0, 'smoothed_offset': -24046608.5, 'smoothed
{'offset': 268532808.5, 'RTT': 152490139.0, 'smoothed_offset': -24046608.5, 'smoothe
{'offset': -17704612.0, 'RTT': 166549921.0, 'smoothed_offset': -24046608.5, 'smoothe
{'offset': 185301863.0, 'RTT': 19432306.0, 'smoothed_offset': -24046608.5, 'smoothed
{'offset': 414446718.5, 'RTT': 14759779.0, 'smoothed_offset': 414446718.5, 'smoothed
{'offset': -99173936.0, 'RTT': 150211096.0, 'smoothed_offset': 414446718.5, 'smoothe
{'offset': -416108219.0, 'RTT': 160335303.0, 'smoothed_offset': 414446718.5, 'smooth
{'offset': 277005776.0, 'RTT': 153583765.0, 'smoothed_offset': 414446718.5, 'smoothe
{'offset': -49116659.5, 'RTT': 136053801.0, 'smoothed_offset': 414446718.5, 'smoothe
{'offset': 193957640.0, 'RTT': 158528567.0, 'smoothed_offset': 414446718.5, 'smoothe
{'offset': 733268835.0, 'RTT': 157902002.0, 'smoothed_offset': 414446718.5, 'smoothe
{'offset': 384095708.5, 'RTT': 147869348.0, 'smoothed_offset': -49116659.5, 'smoothe
{'offset': 69552276.5, 'RTT': 162010432.0, 'smoothed_offset': -49116659.5, 'smoothed
{'offset': -220914545.0, 'RTT': 151873112.0, 'smoothed_offset': -49116659.5, 'smooth
{'offset': -775664538.0, 'RTT': 14700889.0, 'smoothed_offset': -775664538.0, 'smooth
{'offset': 394448633.0, 'RTT': 147518396.0, 'smoothed_offset': -775664538.0, 'smooth
{'offset': -139897244.5, 'RTT': 19578934.0, 'smoothed_offset': -775664538.0, 'smooth
{'offset': 304019759.0, 'RTT': 148239374.0, 'smoothed_offset': -775664538.0, 'smooth
{'offset': -249607332.5, 'RTT': 17404318.0, 'smoothed_offset': -775664538.0, 'smooth
{'offset': 206205084.0, 'RTT': 155303001.0, 'smoothed_offset': -775664538.0, 'smooth
{'offset': -316986578.5, 'RTT': 19504309.0, 'smoothed_offset': -775664538.0, 'smooth
{'offset': -1866035.0, 'RTT': 151319504.0, 'smoothed_offset': -249607332.5, 'smoothe
{'offset': 163701399.0, 'RTT': 19475937.0, 'smoothed_offset': -249607332.5, 'smoothe
```