# CT420 Assignment 1 – NTP Benchmarking Report

Author: Ayodeji Ali StudentID: 20343733

# Introduction

In this assignment, I explore the performance of Network Time Protocol (NTP) servers by measuring and analysing key metrics: **delay**, **offset**, and **jitter**. NTP is essential for synchronizing system clocks across distributed networks, and factors such as **geographical distance**, **number of network hops**, and **time of day** can significantly impact its performance. The approach involved selecting multiple NTP servers from various regions (Ireland, UK, Mainland Europe, United States, Australia, and Asia) and polling them at **20-minute intervals** for approximately **eight hours**. I then performed **traceroutes** to these servers to gather additional information about hop counts, allowing us to investigate potential correlations between route complexity, distance, and observed NTP metrics.

# **Experimental Setup**

I used **Meinberg NTP** on a Windows system to query six carefully chosen NTP servers. Each query recorded:

- **Delay**: The average time (in milliseconds) for an NTP packet to travel between client and server.
- Offset: The difference in clock times between the client and the server.
- **Jitter**: The variability or fluctuation in the packet delay over successive measurements.

These queries were performed every **20 minutes** for roughly **eight hours**, producing a time-series dataset of delay, offset, and jitter values. After this data-collection phase, I executed **traceroutes** to each server to determine the **number of network hops**, which could help explain differences in latency or jitter across servers.

# Data Collection & Analysis

All logged data was stored in a file named **ntp\_output.txt**, with each entry tagged with a **Coordinated Universal Time (UTC)** timestamp. This method ensured I could investigate whether network conditions changed at certain hours of the day. To process the data, I used a **Python script** that parsed the logs and computed statistical metrics—namely the **minimum**, **maximum**, **mean**, and **standard deviation** for **delay** and **jitter** across each server. The script also generated **two separate plots** to visually illustrate how these metrics evolved over time.

- 1. Delay vs. Time
- 2. Jitter vs. Time

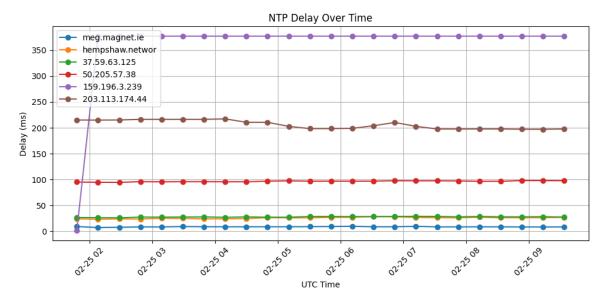
By examining these plots, I aimed to determine if **distance**, **hop count**, or **time of day** had any discernible impact on NTP performance.

# **Results & Discussion**

After analysing the dataset, I found that **servers located farther away** from the client generally exhibited higher **average delay**. For instance, an Australian or Asian server typically showed delays exceeding 200 ms, whereas local servers in Ireland or the UK maintained delays of around 10–30 ms. These observations suggest that **geographical distance** plays a significant role in determining NTP performance. Meanwhile, traceroute results indicated that servers with **more network hops** often correlated with higher delays, implying that **route complexity** can further compound latency issues.

Regarding **jitter**, certain servers displayed spikes at specific times, hinting at **time-of-day effects**. These spikes could coincide with local peak traffic hours in the server's region, potentially increasing network congestion. Some servers, however, maintained relatively stable jitter throughout the observation period, indicating more consistent routing paths or less congestion.

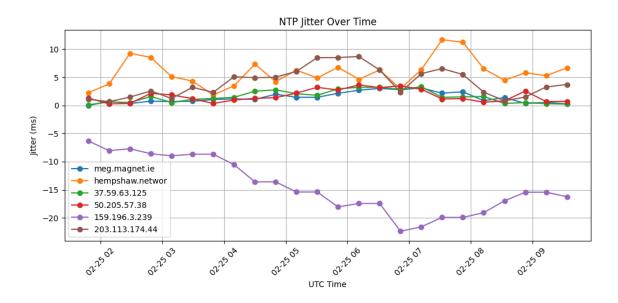
# **Delay Over Time**



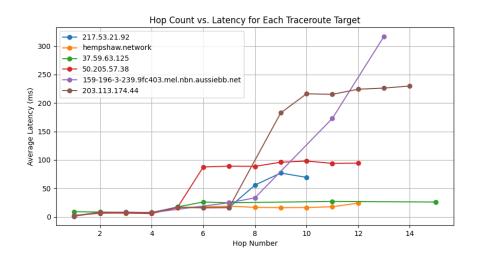
This graph displays how **delay** (in ms) varied over the eight-hour logging window for each NTP server, plotted against **UTC time** on the x-axis. I can observe that servers closer to the client consistently exhibit lower delays, while more distant servers show higher baseline latency. Any noticeable spikes in the graph may point to temporary network congestion or route changes at those specific timestamps.

#### **Jitter Over Time**

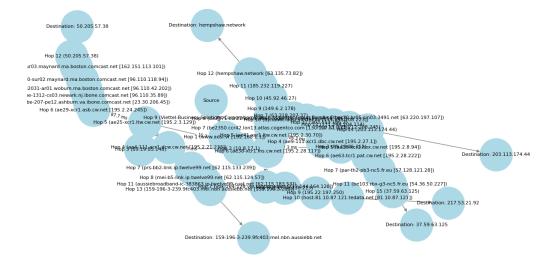
This graph tracks the **jitter** (in ms) for each server over the same eight-hour period, again aligned with **UTC time** on the x-axis. Jitter represents the variability in the delay measurements over successive queries. Higher jitter values may indicate increased route fluctuation or congestion, particularly during local peak usage hours. Some servers maintain steady jitter, suggesting more stable routing paths, while others experience distinct spikes.



#### **Traceroute Visuals**



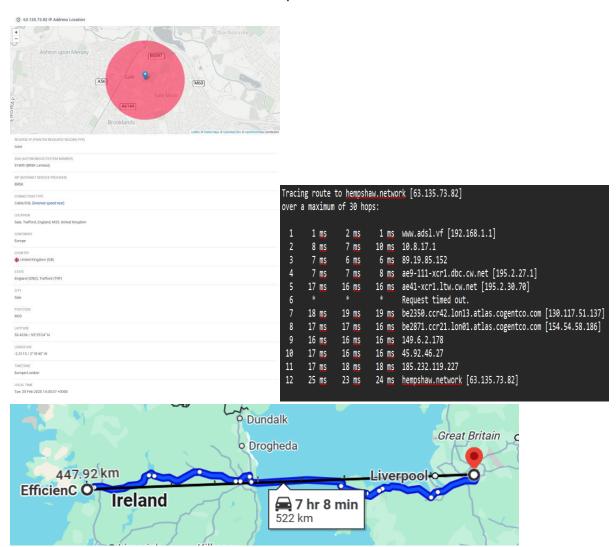
This visualization outlines the **number of network hops** from the client to each server, as determined by traceroute. Generally, servers requiring more hops correlate with higher observed delay in the NTP logs. The traceroute data reinforces the idea that both **distance** and **route complexity** can significantly impact NTP performance.



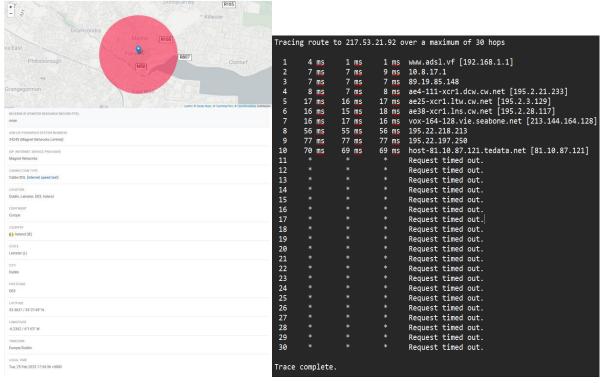
This diagram presents a **graph-based visualization** of the traceroutes from the local machine ("Source") to each NTP server ("Destination"). Each **blue circle** (node) represents either an intermediate hop labelled with its IP address (for example, "Hop 1 (x.x.x.x)") or the final server labelled "Destination: <server>." The **arrows** (edges) connecting these nodes typically indicate the average latency in milliseconds needed to reach that hop. Following the path from "Source" outward to "Destination" reveals how many intermediate hops the traffic traverses, indicating whether the route is short and direct or long and complex.

In general, **longer paths** with more intermediate hops can result in **higher overall latency** to the destination server. For instance, a path involving ten or more hops often signals a more complex route, which can lead to increased delay in NTP queries. Conversely, servers requiring fewer hops may exhibit **lower latency**, suggesting that geographical or topological proximity can substantially benefit NTP performance. Overall, this graph underscores how both **route complexity** (number of hops) and **network topology** influence the reliability and speed of NTP synchronization.

# Traceback and Distance for hempshaw network

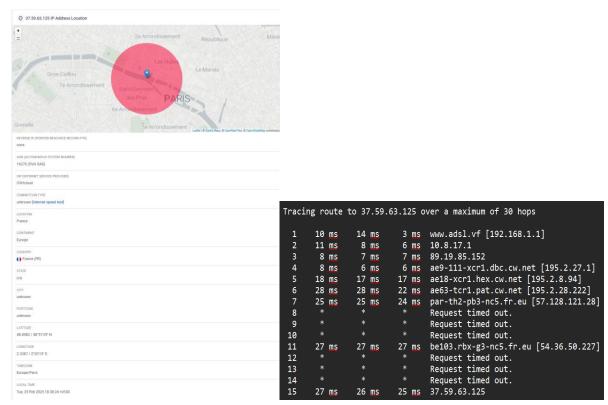


# Traceback and Distance for Magnet Network



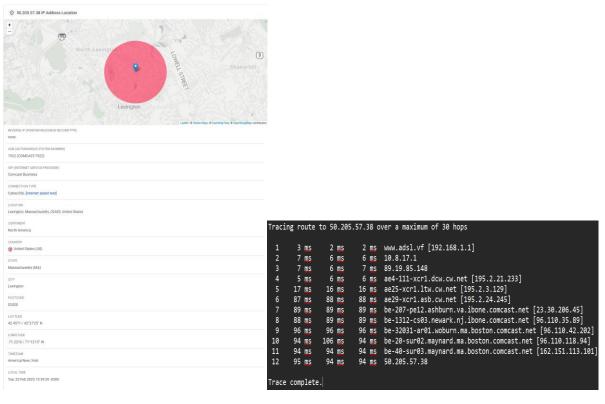


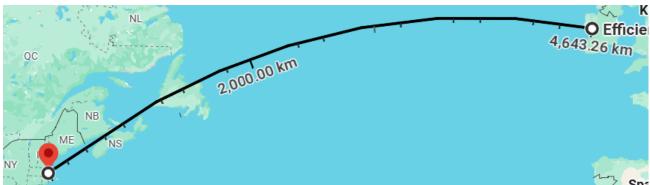
## Traceback and Distance for 37.59.63.125



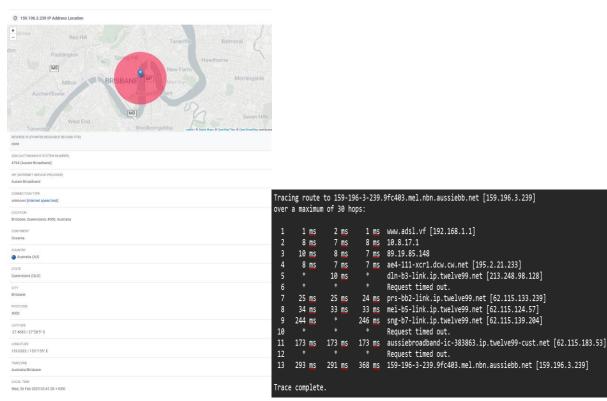


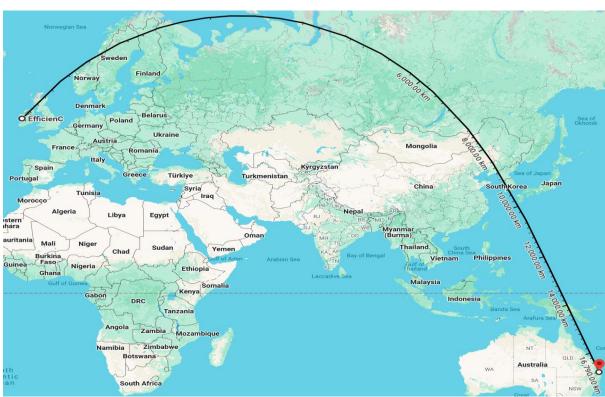
### Traceback and Distance for 50.205.57.38





# Traceback and Distance for 159.196.3.239





# Traceback and Distance for 203.113.174.44

