# Pre-trained Time Synchronization Algorithm for Distributed Networks


Alexander Tu
9000538@student.munster.us
Munster High School

Rui Zhang
zhan5097@pnw.edu
Purdue University Northwest

Supervisor: Dr. Wei Dai (Senior Member, IEEE)
weidai@pnw.edu
Purdue University Northwest


## Abstract


Time synchronization is a critical task in distributed systems, especially those that require precise ordering of events across multiple computer servers. Existing time synchronization algorithms, including the Berkeley's algorithm and the Network Time Protocol (NTP), do not perform well when network delays are unstable. However, stable networks do not always exist. Here, we develop a state-of-the-art algorithm that balances time accuracy, and energy consumption, especially in large networks. We run the new algorithm on a 1024-node simulation and a 14-node real-world network. Compared to NTP, the new algorithm reduces the average time error by 37.96%, improves the energy efficiency by 18.30%, and increases the total energy consumption by only 1.54%. Experimental tests also show that the new algorithm reduces the average time error by 10.70%. We expect the new algorithm to be a starting point for better time synchronization algorithms in RTC environments. In addition, our time-sync algorithm could be used in data centers or RTC systems to synchronize time without using GPS time sources or atomic clocks.


## Problem Statement

Synchronizing time without relying on GPS time or atomic clocks is an interesting challenge. The Berkeley's algorithm and the Network Time Protocol (NTP) do not balance time accuracy and energy consumption in distributed networks, and these existing algorithms do not work well in unstable network, especially RTC environments.

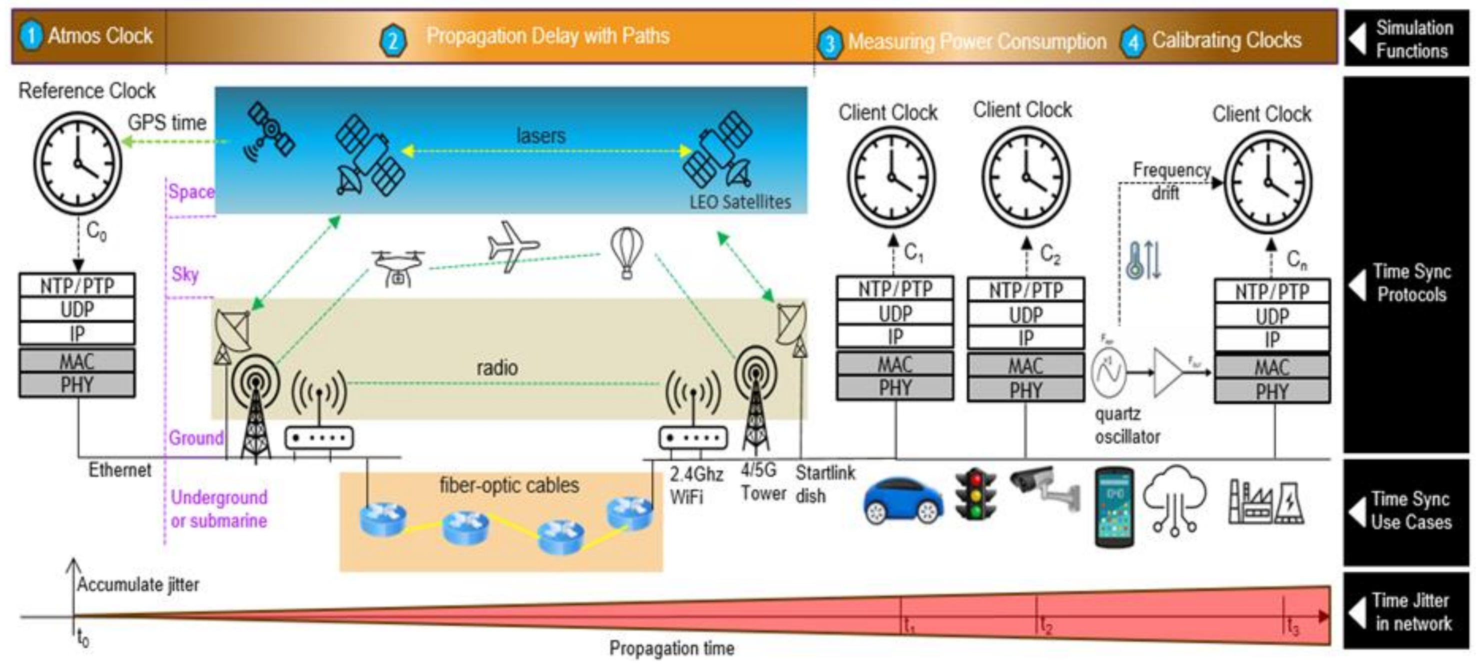


Fig. 1. Clock synchronization in model society

## Methodology and Simulation

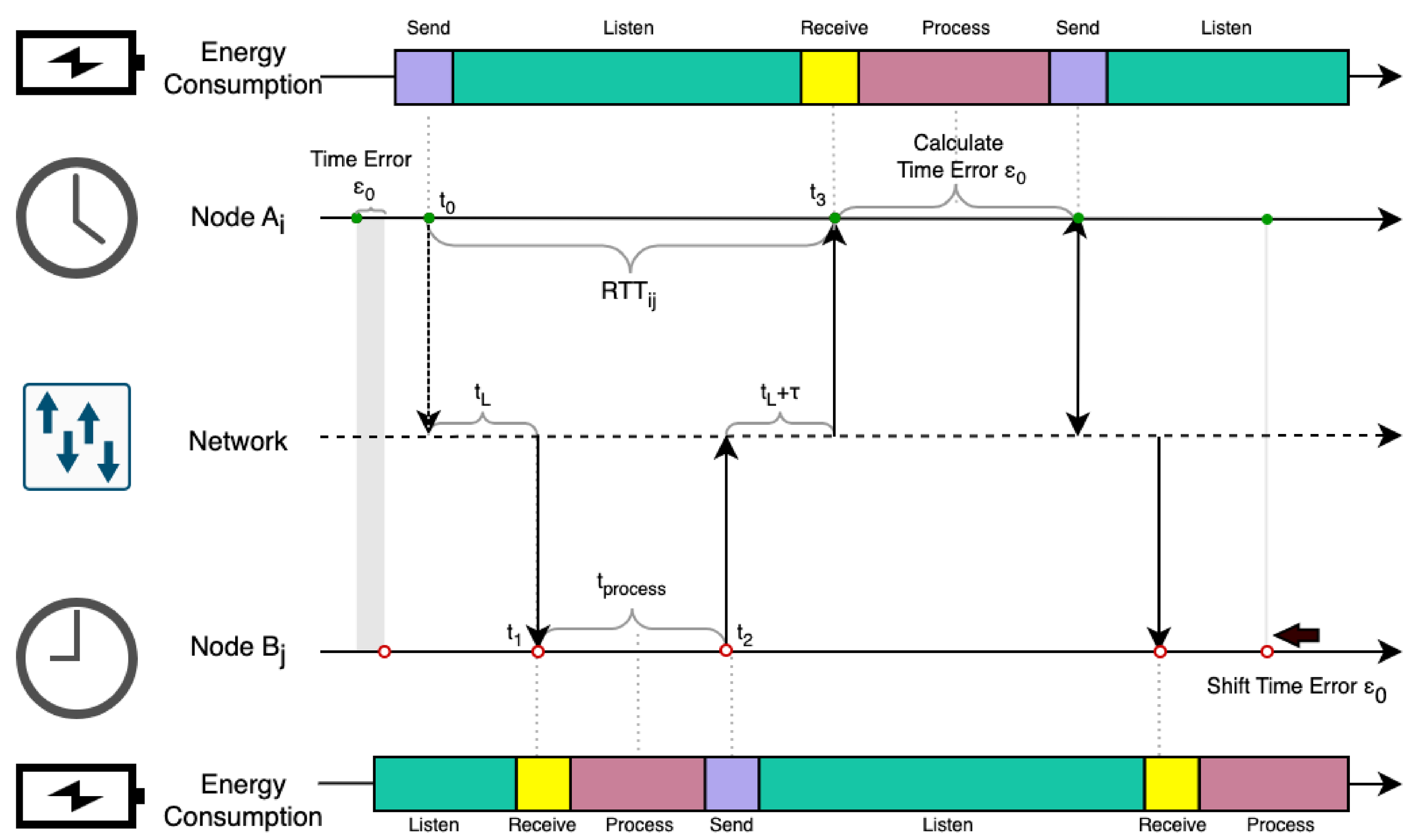


Fig. 2. Two-Way Time Synchronization between Node A and Node B with Energy Consumption.

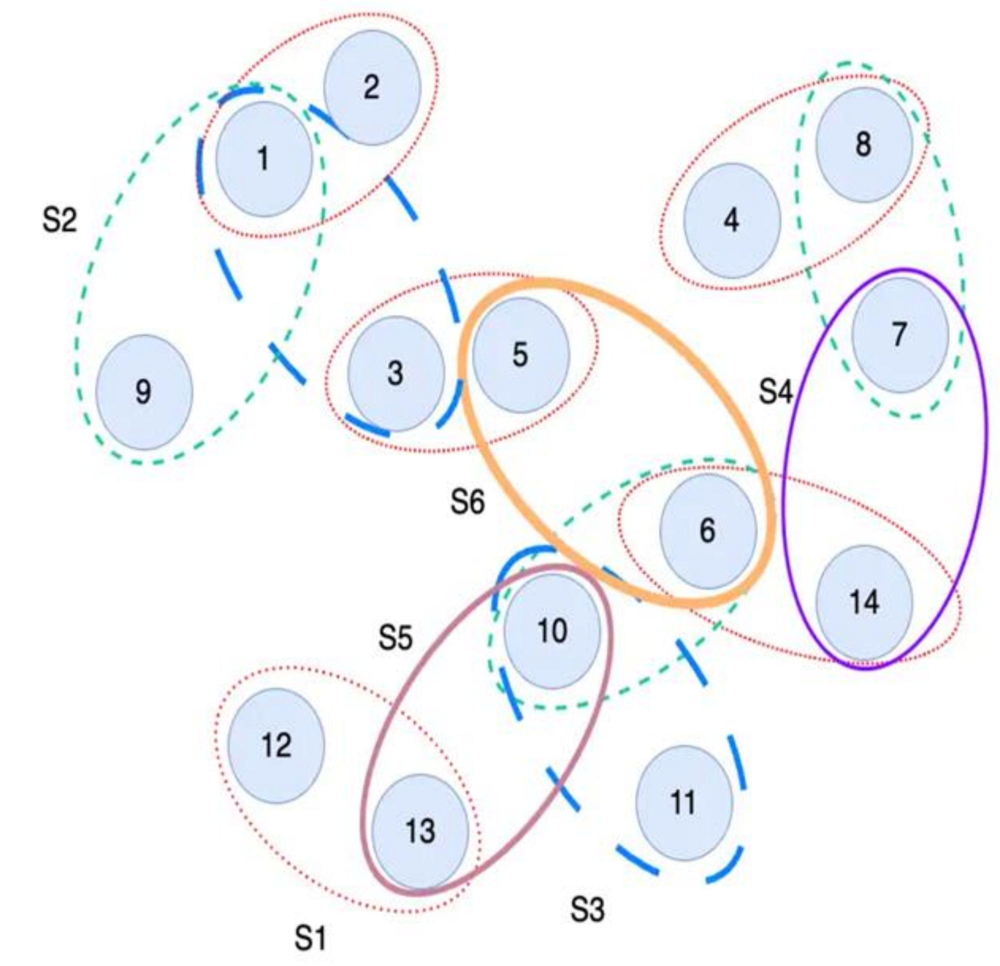


Fig 3. Demonstration of the Gallagher, Humblet, and Spira construction of a minimum spanning tree.

$$t_{b.\text{receive}} = t_{a.\text{send}} + \tau_L + \epsilon_0$$

$$t_{a.\text{receive}} = t_{b.\text{send}} + \tau_L + \tau - \epsilon_0$$

$$\epsilon_0 = \frac{(t_{a.\text{receive}} - t_{b.\text{send}}) + (t_{a.\text{send}} - t_{b.\text{receive}})}{2} + \frac{\tau}{2}$$

Formulas 1-3: Time Error

$$Error_{\text{n}} = \frac{t_\tau}{2}$$

$$Error_{\text{c}} = f_{\epsilon_{A_i}}(\text{RTT}) + f_{\epsilon_{B_j}}(T_{\text{process}})$$

$$Error_{\text{total}} = Error_{\text{n}} + Error_{\text{c}}$$

Formulas 4-6: Synchronization Error

$$\text{Total}_{\text{Energy}} = \sum_j \sum_i \text{Energy}_{i,j}$$

$$i = \{\text{Send}, \text{Receive}, \text{Listen}, \text{Process}\}$$

$$j = \{\text{Node}_1, \text{Node}_2, \text{Node}_3, \dots\}$$

Formulas 7: Energy Consumption

When generating the MST, we used hybrid weights for calculation. First, we explored all weight combinations on a network of 256 sensors. Then, using a multi-objective linear optimization method, we found the optimal weight combination of 7:1.5:1:1.5. We tested this combination on networks ranging from 16 to 2048 nodes on simualtion. The results are listed Table-1.

| Sensor # | Reduced Error | Total Energy | Max Energy |
|---|---|---|---|
| 16 | 4.98% | -0.12% | -0.60% |
| 32 | 17.90% | -0.10% | 1.57% |
| 64 | 37.96% | -0.09% | 2.25% |
| 128 | 30.05% | -0.06% | 2.02% |
| 256 | 23.14% | -0.04% | 3.16% |
| 512 | 12.20% | -0.02% | 8.07% |
| 1024 | 17.72% | -1.54% | 18.30% |
| 2048 | 2.35% | -0.02% | 0.40% |

Table 1: Simulation Results

## Experiment Results

| Node ID | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|---|---|---|---|---|---|---|---|
| New Algorithm(sec) | 9.9E-06 | 1.8E-05 | 4.7E-05 | 1.8E-06 | 3.2E-06 | 3.1E-06 | 1.4E-06 |
| NTP (sec) | 7.4E-07 | 2.4E-06 | 2.4E-05 | 7.9E-06 | 3.4E-06 | 1.8E-05 | 1.4E-05 |
| **Reduced Error** | **20.7%** | **1.4%** | **3.2%** | **3.5%** | **1.2%** | **43.3%** | **24.8%** |
| Node ID | 8 | 9 | 10 | 11 | 12 | 13 | Average |
| New Algorithm (sec) | 1.8E-06 | 5.8E-07 | 8.9E-07 | 7.1E-07 | 2.0E-07 | 2.6E-06 | 2.9E-06 |
| NTP (sec) | 6.7E-06 | 1.2E-05 | 1.0E-05 | 3.8E-06 | 7.1E-06 | 4.4E-06 | 3.1E-06 |
| **Reduced Error** | **9.0%** | **10.3%** | **7.4%** | **2.4%** | **8.0%** | **3.8%** | **10.7%** |

Table 2 : Experiment results between the new algorithm and NTP

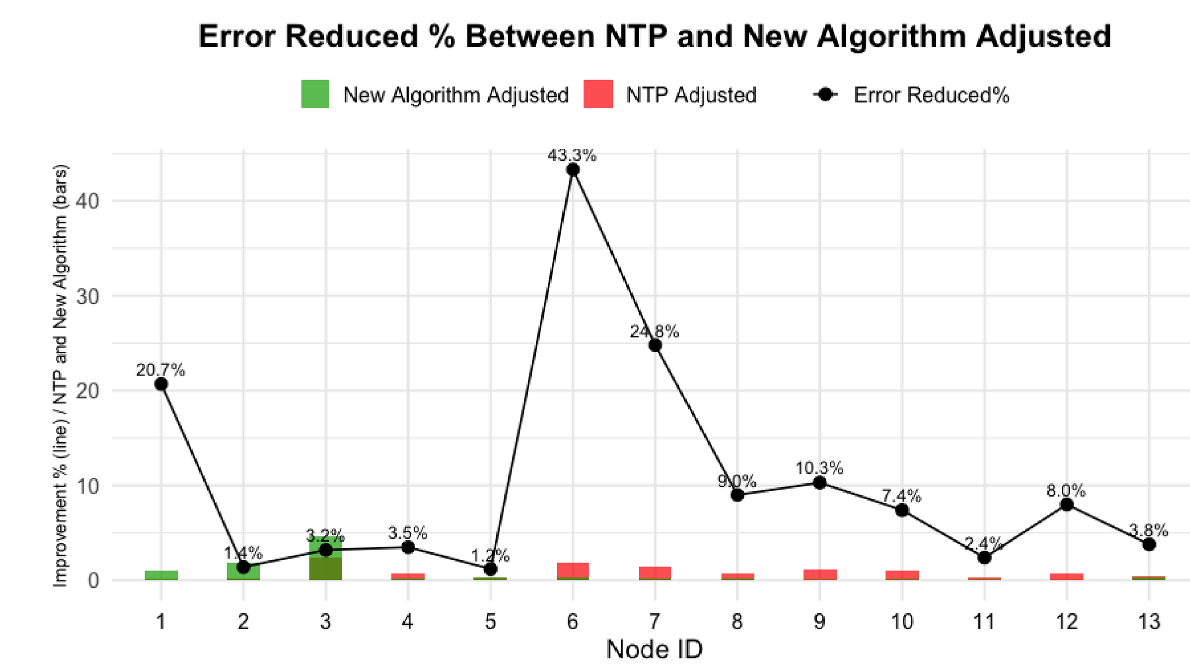


Fig. 4. Visualization of NTP and the New Algorithm

## Conclusion & Future Work

The research has demonstrated effectiveness in scenarios requiring high accuracy in time synchronization, increasing its applicability in unstable network environments. The newly developed time-synchronization algorithm demonstrated a 37.96% reduction in the average time error, an 18.30% improvement in energy efficiency, and a mere 1.54% increase in total energy consumption. Furthermore, experimental tests substantiated that the novel algorithm reduced the average time error by 10.70%.

**Future Work:**

- Evolve experimental tests in a data center for collecting field test results.
- Automating the determination and application of optimal weights based on varying network conditions.

## Open Source

All codes on the Github at
https://github.com/Astrowebdeving/fullDistributedSys

## Acknowledge


Purdue Northwest Provost Office and Indiana Space Grant Consortium (NASA-controlled) have funded the research project.