Resilient Clock Synchronization using Power Grid Voltage

DIMA RABADI, Institute for Infocomm Research (I^2R), A*STAR, Singapore
RUI TAN, Nanyang Technological University, Singapore
DAVID K.Y. YAU, Singapore University of Technology and Design, Singapore
SREEJAYA VISWANATHAN, Advanced Digital Sciences Centre, Singapore
HAO ZHENG, Zhejiang University, China
PENG CHENG, Zhejiang University, China

Many clock synchronization protocols based on message passing, e.g., the Network Time Protocol (NTP), assume symmetric network delays to estimate the one-way packet transmission time as half of the round-trip time. As a result, asymmetric network delays caused by either network congestion or malicious packet delays can cause significant synchronization errors. This paper exploits sinusoidal voltage signals of an alternating current (ac) power grid to limit the impact of the asymmetric network delays on these clock synchronization protocols. Our extensive measurements show that the voltage signals at geographically distributed locations in a city are highly synchronized. Leveraging calibrated voltage phases, we develop a new clock synchronization protocol, which we call Grid Time Protocol (GTP), that allows direct measurement of one-way packet transmission times between its slave and master nodes, subject to an analytic condition that can be easily verified in practice. The direct measurements render GTP resilient against asymmetric network delays under this condition. A prototype implementation of GTP maintains sub-millisecond synchronization accuracy for two nodes tens of kilometers apart in the presence of malicious packet delays. The result has been demonstrated for both Singapore and Hangzhou, China. Simulations driven by real network delay measurements between Singapore and Hangzhou under both normal and congested network conditions also show the synchronization accuracy improvement by GTP. We believe that GTP is suitable for grid-connected distributed systems that are currently served by NTP but desire higher resilience against unfavorable network dynamics and packet delay attacks.

CCS Concepts: • **Networks** → **Time synchronization protocols**; **Cyber-physical networks**; • **Computer systems organization** → *Dependable and fault-tolerant systems and networks.*

Additional Key Words and Phrases: Clock synchronization, power grid, security

A preliminary version of this work appeared in The 12th ACM Asia Conference on Computer and Communications Security (ASIACCS 2017). This work was supported in part by Singapore MOE (Award No. SUTDT12017004), in part by the National Research Foundation, Prime Minister's Office, Singapore under the Energy Programme and administrated by the Energy Market Authority (EP Award No. NRF2014EWT-EIRP002-026), in part by an NTU Start-up Grant, in part by SUTD-ZJU IDEA (Award No. 201805), and in part by NSFC (Award Nos. 61429301 and 61833015).
Authors' addresses: Dima Rabadi, Cyber Security and Intelligence, Institute for Infocomm Research (I^2R), A*STAR, Singapore, (work done while at the Singapore University of Technology and Design), dimadsr@i2r.a-star.edu.sg; Rui Tan, School of Computer Science and Engineering, Nanyang Technological University, Singapore, tanrui@ntu.edu.sg; David K.Y. Yau, Singapore University of Technology and Design, Singapore, david_yau@sutd.edu.sg; Sreejaya Viswanathan, Advanced Digital Sciences Centre, Singapore, Sreejaya.v@adsc.com.sg; Hao Zheng, The State Key Lab of Industrial Control Technology, Zhejiang University, China, zjuzhenghao@163.com; Peng Cheng, The State Key Lab of Industrial Control Technology, Zhejiang University, China, pcheng@iipc.zju.edu.cn.

XXXX-XXXX/2019/2-ART $15.00
https://doi.org/0

ACM Reference Format:
Dima Rabadi, Rui Tan, David K.Y. Yau, Sreejaya Viswanathan, Hao Zheng, and Peng Cheng. 2019. Resilient Clock Synchronization using Power Grid Voltage. *ACM Transactions on Cyber-Physical Systems* 1, 1 (February 2019), 27 pages. https://doi.org/0

1 INTRODUCTION

Secure clock synchronization is critical for many mission-critical distributed system applications. For instance, in common SCADA-controlled infrastructures, various computing nodes and intelligent electronic devices (IEDs) in an advanced manufacturing system monitor collaboratively the system state in real time, in order to run a set of remote terminal units (RTUs) and PLC-enabled actuators. In such systems, for safe and effective operation, the computers, IEDs, and actuators must be tightly synchronized, to within a few milliseconds [3].

Many of today's cyber-physical systems have mainly employed the Network Time Protocol (NTP) in the system's local area networks (LANs) to distribute UTC time from GPS-equipped masters to various slaves. More generally, NTP is a foremost means of network time synchronization that is widely known and adopted. Its design principles are also representative of a large class of synchronization protocols based on message exchanges between the synchronizing nodes. Other examples of these protocols include Precision Time Protocol (PTP) [11] and those for wireless sensor networks such as RBS [4], TPSN [5], and FTSP [16].

In normal operation, NTP's accuracy is generally accepted to be within a few milliseconds. However, NTP is susceptible to a number of attacks. Certain of these attacks can be detected by protocol constructs with cryptographic protection. For example, authenticated sequence numbers can guard against malicious dropping of packets, and signed messages or message digests can ensure the integrity of the message content. A simple but powerful form of attack against NTP (or any synchronization protocols based on message passing), which has evaded satisfactory detection and mitigation so far, is malicious packet delays. In this attack, the adversary on a forwarding path of the NTP packets maliciously delays one direction of the communications between the slave and master. As the malicious delay does not change the message itself, it is immune to cryptographic protection. It is effective, however, because it invalidates a basic *symmetric link assumption* of NTP [22]. Specifically, a malicious delay of d can introduce a synchronization error up to $\frac{d}{2}$; errors on the order of 10 ms up to seconds are eminently feasible. There are various heuristic approaches to detecting and mitigating the delay attack [17, 21, 22], but none of these are completely foolproof. In Section 3, for example, we will demonstrate a subtle attack via ARP spoofing that can evade detection based on tracking historical round-trip times (RTTs) by moving averages [7, 20, 22].

As NTP cannot measure directly one-way transmission times of its synchronization packets for clock offset calculations, it relies on the symmetric link assumption to estimate the one-way transmission time as half of the RTT in a slave-master communication. In this paper, we seek a trustworthy external signal that both the slave and master can observe, so that they can measure directly the one-way transmission times. A new synchronization approach based on this direct measurement will no longer need the symmetric link assumption; it will be resilient against asymmetric network delays caused by network congestion or packet delay attacks. To realize the approach, we exploit the voltage waveforms of an alternating current (ac) power grid for trustworthy and accurate clock synchronization between distributed network nodes. In this paper, we assume that the power grid voltage signal is intact, because tampering with it would require a tremendous amount of energy that is infeasible economically and logistically for would-be attackers. The grid's voltage is location dependent; its values at different monitoring points are different. However, in an ac power grid, the sinusoidal voltage waveforms at all the locations are driven by a same frequency. In existing practice, this frequency is either 60Hz (e.g., in the Americas) or 50Hz (in most

other parts of the world). Hence, the periodicity of the waveforms is synchronized, although the synchronization is imperfect because the phase of the voltage signal changes with location, and the grid's frequency is not truly constant but it is continuously regulated around the nominal value in response to changes in load and generation. An important research question that we seek to answer is whether this synchronization is good enough for practical applications.

To answer the question, we conduct extensive measurements in a city to verify key synchronization properties of the ac grid voltage. Based on the results, we design and implement a new clock synchronization approach, which we call Grid Time Protocol (GTP), that (i) achieves better accuracy than NTP, and (ii) is resilient against asymmetric network delays caused by either network congestion or malicious packet delays. Moreover, we achieve an economical design that can be readily and widely adopted by commodity computing devices with direct utility power access. We make the following main contributions in this paper:

- We verify by real-world experiments that a succinct *phase angle* feature of voltage waveforms exhibits suitable range and stability for accurate and trustworthy distributed clock synchronization.
- Based on the phase angle, we design GTP that achieves sub-ms accuracy in both LAN and city-scale wide-area network (WAN) settings. This accuracy represents a significant improvement over that of NTP, whose errors are often reported to be on the order of ms or even tens of ms [25]. Moreover, unlike NTP, GTP is resilient against malicious packet delays subject to an easy-to-verify condition, which we call the *secure GTP condition*, that is made clear by our analysis.
- We have designed and implemented a working prototype of GTP using PC-class sound cards, general purpose operating system (OS), and a low-cost voltage sensor design. Our experiments demonstrate predominant achievement of the secure GTP condition under diverse settings, including congested city-scale WANs. They also demonstrate ready applicability of GTP to nodes that are connected to the same power grid, and verify GTP's accuracy and robustness in both LAN and WAN scales.
- Unlike NTP, GTP achieves unambiguous trustworthy synchronization under the secure GTP condition. We leverage this property to design a resilience policy for running GTP in practical networks with access to multiple GTP masters. The resilience policy ensures trustworthy clock synchronization when some but not all of these masters are under attack. When the secure GTP condition cannot be satisfied due to long distances between the masters and the slaves, our extensive measurements for a Singapore-Hangzhou link show that GTP still outperforms NTP with high probability in the presence of network congestion.

The rest of the paper is organized as follows. Section 2 reviews related work. Section 3 analyzes the impact of asymmetric network delays on NTP and demonstrates this impact via launching packet delay attacks. Section 4 presents extensive grid voltage measurements to establish the foundation of GTP. Section 5 presents the principle of GTP. Section 6 derives the secure GTP condition and presents empirical evaluation of GTP in LANs and city-scale WANs. Section 7 analyzes the performance of GTP versus NTP over long geographic distances and presents evaluation results based on extensive network delay traces between Singapore and Hangzhou. Section 8 compares GTP with other recent grid-based time synchronization protocols. Section 9 concludes this paper.

2 RELATED WORK

Clock synchronization is a fundamental system function of computer networks. There are two broad categories of clock synchronization approaches based on message passing and external periodic physical signals, respectively. Message passing approaches estimate the clock offset between

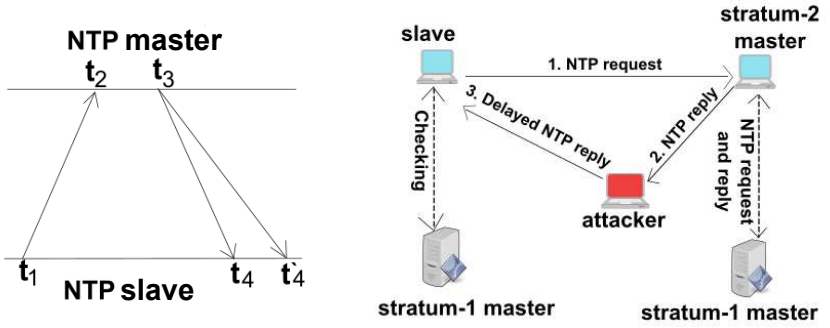
two network nodes by measuring the RTT and one-way transmission times [8]. In NTP, time servers (i.e., masters) are organized into a layered hierarchy, where each layer is called a *stratum* and a smaller stratum number means a layer closer to the groundtruth time sources (e.g., atomic clocks or GPS receivers). A stratum-n master updates its clock according to clock offsets estimated from the RTTs of multiple stratum-n and stratum-(n-1) masters. Various message-passing clock synchronization protocols have also been proposed for wireless sensor networks, such as RBS [4], TPSN [5], and FTSP [16]. As the physical distance between two sensor nodes is often limited, these protocols generally ignore the propagation delays of the radio messages used, but they can still achieve high accuracy due to hardware-level timestamping for the exchanged messages.

Recent work has leveraged various external periodic physical signals to synchronize low-power devices or extract timestamps from recorded data. In [19], Rowe *et al.* propose a hardware device called Syntonistor to sense a periodic electromagnetic signal radiating from powerlines and use it to calibrate[1] the clocks of wireless sensors. In [14], Li *et al.* use light sensors to sense the intensity of a fluorescent light that flickers at a frequency twice that of the ac grid frequency. The periodic flickering is used to calibrate the clocks of nodes. Similarly, other external periodic signals in FM radios [12] and Wi-Fi beacons [6] have been leveraged for clock calibration. Using the above clock synchronization approaches, multiple nodes remain synchronized once they are initially synchronized. The initial synchronization, however, requires the exchange of network messages, which may be adversely affected by asymmetric network delays. However, none of these studies address the asymmetric network delays during this initial synchronization, but we do.

Recent research has studied the security of clock synchronization approaches. NTP is susceptible to integrity and packet delay attacks. An integrity attack that modifies data fields in the synchronization packets can be addressed by cryptographic encryption. A packet delay attack adds malicious time delays to the transmissions of NTP synchronization packets, which invalidates the protocol's symmetric link assumption. Various heuristic approaches have been proposed to detect packet delay attacks, but none of them can provide complete detection. These approaches include setting an upper bound for allowed RTTs [22], comparing the latest RTT with the RTT history [21], and comparing the RTT of NTP with those of other protocols [17]. These heuristic detectors can be bypassed by small attack delays, gradually increased delays, and delays added to all the packets of a victim node. Although more stringent detection thresholds can be used to limit the attack's impact, they will lead to high false alarm rates under dynamic network conditions. This observation will be demonstrated via experiments in Section 3. In contrast, the GTP proposed in this paper exploits an ac electric grid's periodic voltage signal to measure directly one-way packet transmission times between the slave and master. This approach fundamentally disentangles GTP from the symmetric link assumption, and renders it immune to packet delay attacks.

Our prior work [18] presented the design of GTP and evaluated it in Singapore. Based on [18], we make the following two new contributions in this paper. First, we evaluate GTP in Hangzhou, China, at both the LAN and WAN scales. Second, we propose a long-haul GTP to address the cases where the secure GTP condition cannot be met due to long geographic distances. Simulations based on Singapore-Hangzhou network delay traces show that the long-haul GTP outperforms NTP. The secure GTP and the long-haul GTP constitute a GTP hierarchy that increases the clock synchronization resilience.

[1] *Clock calibration* ensures that different clocks will advance at the same speed; *clock synchronization* regulates the clocks to have the same value.



(a) An NTP session. (b) Experimental network.

Fig. 1. NTP in normal operation and under asymmetric network delays, and experimental network of the packet delay attack.

3 MOTIVATION

NTP is the most widely adopted clock synchronization protocol in computer networks. Its design is representative of a large class of the protocols based on message passing. This section reviews NTP and analyzes the impact of asymmetric network delays on its performance. We will demonstrate respectively NTP's performance in normal operation and under asymmetric network delays caused by packet delay attacks.

3.1 Impact of Asymmetric Network Delays on NTP

3.1.1 NTP Principle. As described in Section 2, the nodes running NTP are organized into a layered hierarchy. Each node often runs as both slave and master. For instance, a stratum-n node acts as a slave in synchronizing itself with a stratum-(n-1) node, and as a master when providing its clock values to a stratum-(n+1) node or other stratum-n nodes. Consider a pair of NTP master and slave. Fig. 1(a) illustrates a synchronization session between them. The slave starts by sending a request that contains the time of sending the request based on the slave's clock (t_1); the master receives the request and sends back a reply that contains the time of receiving the NTP request from the slave (t_2) and the time at which this reply is sent (t_3), where both t_2 and t_3 are according to the master's clock. When the slave receives the reply, it records the receive time (t_4) and then computes the offset (Δ) between its and the master's clocks, based on the RTT computed from the quadruple time values ($t_1, t_2, t_3,$ and t_4). The RTT and offset are calculated as $\text{RTT} = (t_4 - t_1) - (t_3 - t_2)$ and $\Delta = t_3 + \frac{\text{RTT}}{2} - t_4$. The above offset calculation is based on a *symmetric link assumption*, i.e., the one-way delays for transmitting the request and the corresponding reply are equal. Based on the computed offset, the slave will calibrate its clock.

3.1.2 Asymmetric Network Delays. Asymmetric links in practice will lead to synchronization errors in NTP. Such asymmetry can be caused by either natural network congestion or malicious time delays introduced to the transmissions of the request and/or reply packets. For the latter case, we formally define the threat model of packet delay attack as follows.

Packet delay attack: We assume that the endpoints (master and slave) of a clock synchronization protocol are trustworthy. However, one or more attackers on a network path of the protocol's packets may delay the transmission of these packets. We assume that the total malicious delay

for a packet is finite. Moreover, we assume that the protocol's packets cannot be tampered with because of cryptographic protection.

We now analyze the impact of the asymmetric network delays on NTP. Fig. 1(a) illustrates a case in which the slave's receive of the NTP reply is delayed from t_4 to t_4' due to a packet delay attack. We assume that the t_4' is still within the NTP's default timeout (normally 1 s [1]). The delay will adversely affect the offset computation. The computed offset in the presence of the network delay is given by $\Delta' = t_3 + \frac{\text{RTT}}{2} - t_4' = t_3 + \frac{t_4' - t_1 - (t_3 - t_2)}{2} - t_4'$. Thus, the extra offset is $\Delta' - \Delta = \frac{t_4 - t_4'}{2}$. Note that the attacker needs to attack only one direction of the communications, because a delay in the other direction would mitigate the effects of the first. Specifically, if the attacker delays the master's receive of the NTP request from t_2 to t_2' and the slave's receive of the NTP reply from t_4 to t_4', the offset computed by the slave, denoted by Δ'', is $\Delta'' = t_3 + \frac{\text{RTT}}{2} - t_4' = t_3 + \frac{t_4' - t_1 - (t_3 - t_2')}{2} - t_4'$. Thus, the added offset is $\Delta'' - \Delta = \frac{(t_4 - t_4') - (t_2 - t_2')}{2}$. We can see that, if the attacker introduces the same delay to the request and reply packets (i.e., $t_4 - t_4' = t_2 - t_2'$), the attack has no effect (i.e., $\Delta'' = \Delta$). Delaying one direction of the communications is the most effective attack.

3.2 Two Packet Delay Attack Experiments

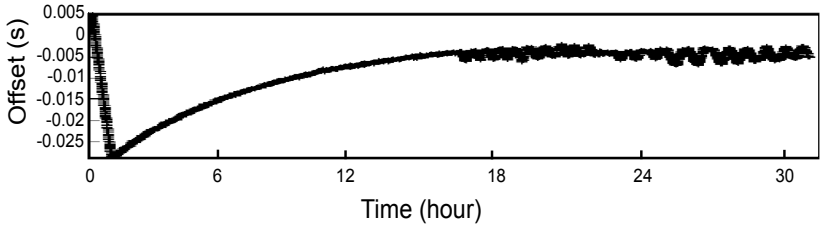
This section presents two experiments to demonstrate the actual impact of packet delay attacks on NTP.

3.2.1 Experiment Setup. The experiment setup is shown in Fig. 1(b). The setup consists of three computers in the same LAN acting as the NTP slave, (stratum-2) NTP master, and attacker, respectively. The NTP daemon `ntpd` running on the slave and master are from the NTP reference implementation `ntp-4.2.8p3`. The NTP slave is configured to synchronize with the NTP master, which synchronizes with higher-level (stratum-1) NTP masters equipped with GPS receivers. The slave's `ntpd` daemon periodically initiates NTP sessions illustrated in Fig. 1(a), based on which it calibrates the slave's clock. To assess the synchronization error, the slave uses the `ntpdate` utility to periodically check its time offsets against the stratum-1 masters with groundtruth GPS time.

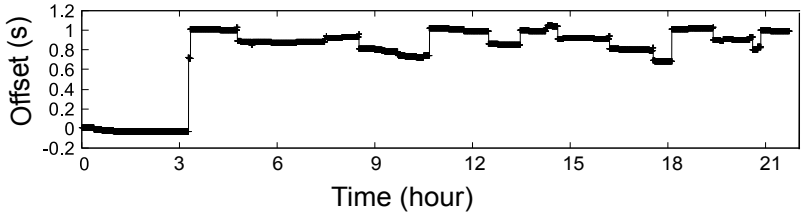
Packet delay attacks on the NTP reply messages are via Address Resolution Protocol (ARP) spoofing, which is practical because ARP generally must be enabled for a LAN's normal operation. The spoofing allows the attacker to intercept the NTP replies. Then, using a traffic control utility `tc`, the attacker delays each reply for a specified amount of time before forwarding it intact to the slave.

3.2.2 Packet Delay Attack with Fixed Delay. We first report NTP's synchronization errors in the absence of attacks. Fig. 2(a) profiles the offsets of the slave's clock from the stratum-1 time masters over time. We can see that the offsets converge to approximately 5 ms, which is a typical synchronization error achieved by NTP in LAN settings. In the second experiment, the slave uses Apache's Java implementation of NTP (`org.apache.commons.net.ntp`) to synchronize with the master. By default, the timeout of this NTP implementation is not enabled. The attacker introduces a malicious delay of 2 s to the NTP reply packets. As shown in Fig. 2(b), the measured offsets are around 1 s, which is half of the malicious delay and consistent with our analysis in Section 3.1.

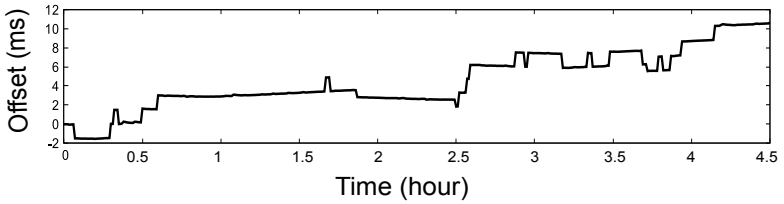
3.2.3 Bypass Moving Average based Detection. Section 2 reviews state-of-the-art methods for detecting packet delay attacks. A widely adopted approach [7, 20, 22] tracks the average RTT of protocol packets (NTP packets in our context) over time windows, and checks that any changes of the average between consecutive windows fall within a predetermined threshold. For example, if the window size is four, the average is calculated based on the last four RTT samples. This average is updated every time a new NTP session occurs and a new RTT sample hence becomes available.



(a) No packet delay attack.



(b) Packet delay attack starts from the 3rd hour.



(c) Gradually increasing delay attack.

Fig. 2. Slave's clock offset from GPS time.

If the difference between the new and old averages exceeds the threshold, the detection declares an attack. The detection guards against abrupt increases in the RTT as evidence of attacks. Thus, to avoid detection, the attacker must limit the malicious delay to a modest value, which in turn limits the attack impact.

Unfortunately, a practical attacker can launch a series of small attacks whose delay increments stay below the detection threshold but add up to a significant cumulative delay over time. To verify its effectiveness, we launch such a gradually increasing delay attack on the NTP using the same ARP spoofing mechanism as before. The window size is four and the detection threshold is 3 ms. Before the attack starts, the last four measured RTTs are 2, 2, 2, and 2.5 ms, respectively, for a logged average of 2.125 ms.

The attacker starts with a small malicious delay of 2 ms, resulting in an observed new RTT sample of 4 ms and, accordingly, a new average RTT of 2.625 ms. The moving average increases by 0.5 ms, well within the threshold, and the clock drifts maliciously by 1 ms, which is half the delay as analyzed in Section 3.1.2. Then, the attacker gradually increases the delay from 2 ms to 20 ms. Fig. 2(c) tracks the offset of the slave's clock from the groundtruth time after each instance of the malicious delay. Note that the achieved offset increases to 10 ms, as shown in Fig. 2(c), while the average RTT differences keep below the detection threshold.

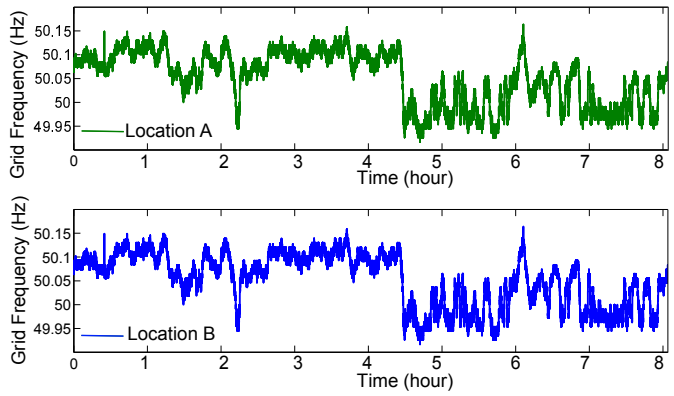


Fig. 3. Power grid frequency measured at two locations 30 km apart.

3.3 Implication on Time-critical Systems

NTP is widely used in various industrial systems. For instance, from our communication with power plant operators and power grid SCADA system integrators, NTP remains the main method used to synchronize various computers and intelligent electronic devices (IEDs) in a power substation with a GPS-based time master. These computers and IEDs monitor the status of power generators and the grid, and control various critical actuators. Because of fast dynamics of electricity, the network congestion and the packet delay attack pose a real danger to the time-critical generator/grid sensing and control. The attack can be launched in the LAN of a power substation by, for example, an insider attacker or a worm that has bypassed the system's air gap.

4 POWER GRID VOLTAGE PROFILES

This paper exploits the power grid voltage to limit the impact of the asymmetric network delays on message passing-based clock synchronization. This section presents the background for power grid voltage and our extensive measurements in a city-scale power grid to provide a design basis for the proposed clock synchronization approach.

4.1 Background

Alternating current (ac) power grids are designed to run at a prescribed nominal frequency (60 Hz in the Americas and 50 Hz in most other places of the world) [9]. This frequency, which is the frequency of the sinusoidal ac voltage, is almost identical across all locations in a grid [9]. Because of constant load dynamics, however, the frequency is not fixed but it must be continuously regulated around the nominal value by, for example, dynamic generation control. This causes persistent albeit small fluctuations of the frequency around the nominal value over time, as Fig. 3 shows for our frequency measurements at two separate locations 30 km apart in Singapore. Previous work [23] uses these frequency fluctuation signatures indicative of time for synchronizing distributed devices. However, to capture the often minute fluctuations, the prior approach [23] employs a highly customized hardware peripheral to achieve high-rate and high-precision sampling, which increases the system cost.

In this paper, we take an alternate economical approach of analyzing the angle of the periodic ac voltage signals. Specifically, the normalized power grid voltage at a location l, denoted by $v_l(t)$, is given by $v_l(t) = \sin(2\pi f_l(t) \cdot t + \psi_l(t) + \phi_l)$, where t is the Newtonian time, $f_l(t)$ is the localized grid

frequency as shown in Fig. 3, $\psi_l(t)$ is the voltage angle, and ϕ_l denotes the grid phase that will be explained shortly. As shown in Fig. 3, $f_l(t)$ changes over time due to transient imbalance between active power generation and active load. It can be slightly different across different locations. The voltage angle often has a small value and changes over time due to the changing geographic distribution of active load in a power grid [9]. Most existing power grids have three phases, i.e., the ϕ_l takes on three possible values of 0, $-2\pi/3$, and $2\pi/3$. In a building, the wall power outlets are often evenly distributed among these three phases, depending on the configuration of the power distribution network. To simplify discussions, we assume that the slave and master at two locations l_c and l_s, respectively, observe the same power grid phase, i.e., $\phi_{l_c} = \phi_{l_s}$. In the experiments conducted in this paper, we select the wall power outlets to match this assumption. In Section 6.3, we will discuss how to address the case in which the slave and master observe different power grid phases.

In this paper, the angle of v_l is defined as $\theta_l(t) = 2\pi f_l(t) \cdot t + \psi_l(t)$, which will be the basis of our new clock synchronization protocol. To use $\theta_l(t)$ for the synchronization, our main concern is whether the difference between the voltage angles of the slave and master (in radians), i.e., $\theta_{l_c}(t) - \theta_{l_s}(t)$, remains near-constant despite the changes of f_l and ψ_l. To simplify discussion, the *angle difference* between the slave and the master in milliseconds is defined as $\gamma(t) = \frac{\theta_{l_c}(t) - \theta_{l_s}(t)}{2\pi} \cdot p$, where p is the nominal ac cycle time duration, e.g., $p = 20$ ms for a 50 Hz power grid. Thus, the $\gamma(t)$ captures the impact of the active load fluctuation and its varying geographic distribution over time. If the angle difference is unknown, it will be part of the synchronization error of the proposed approach; otherwise, we can exclude it from the calculations. As a result, the range and stability of this angle difference are important. A stable angle difference will allow us to calibrate the proposed system using the average value of $\gamma(t)$, denoted by $\bar{\gamma}$. A small $\gamma(t)$ makes sure that the synchronization error is small, when $\bar{\gamma}$ cannot be measured for the system calibration. We note that, as ac voltage is cyclic, the voltage angle difference is within $(-\pi, \pi)$ in radians, which is $(-p/2\,\text{ms}, p/2\,\text{ms})$ in time. Thus, in a 50 Hz grid, the upper bound of synchronization error is 10 ms when the proposed system is not calibrated.

We note that a load that is not purely resistive will increase the angle between voltage and current (i.e., deteriorate the power factor), which is different from the voltage angle difference between two locations that is of concern in this paper. Thus, the power factor does not affect the voltage-based clock synchronization.

To support the integration of renewable energy resources and new power network structures (e.g., microgrids), ac/dc hybrid distribution grids have been proposed as a promising solution [10, 15]. The approach presented in this paper is for the networked nodes that are deployed at the edges of the distribution grids supplying ac power to normal electrical appliances. Thus, when the master and slave nodes are located within two areas separated by a dc link, their measured ac voltage signals will have an angle difference. However, as discussed earlier, the synchronization errors caused by the voltage angle difference are upper-bounded by 10 ms.

4.2 Power Grid Voltage Measurements

In this section, we capture real ac powerline voltage signals to illustrate their synchronization property. In particular, we analyze the angle difference $\gamma(t)$ between two different locations in LAN and WAN settings, respectively. In the LAN setting, we use two nodes located on the same floor of a building. For the WAN setting, we use two nodes that are respectively about 15 km (nodes B and C) and 30 km (nodes A and B) apart in Singapore, as shown in Fig. 4. Note that nodes A, B, and C are within a university, an office building, and a residential building, respectively. The city's grid frequency is 50 Hz. The measurements are conducted using custom hardware that can

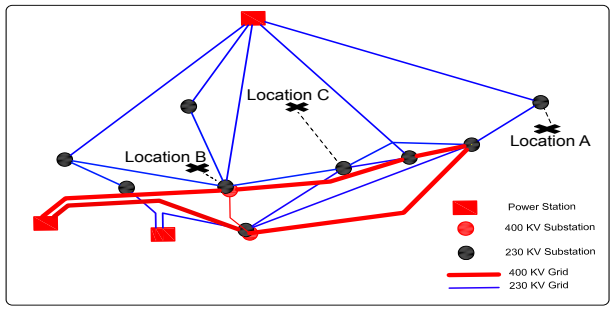
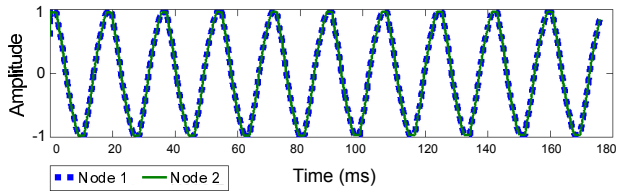
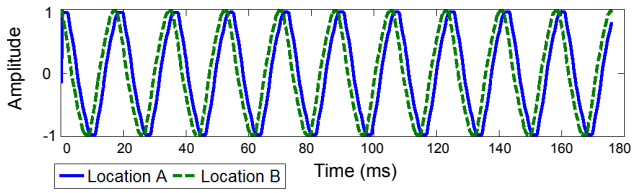


Fig. 4. The locations of three distributed nodes with respect to the transmission system of Singapore. The distance between nodes A and B is 30 km; that between nodes B and C is 15 km. A dashed line connects a location with its transmission system bus.



(a) In the same building floor at Location B



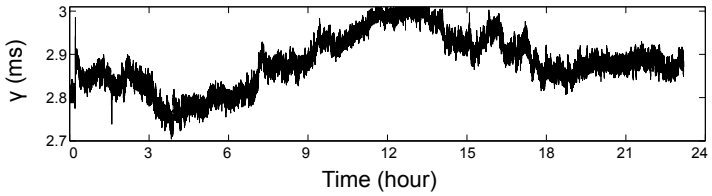
(b) 30 km apart.

Fig. 5. Voltage signals measured by two nodes.

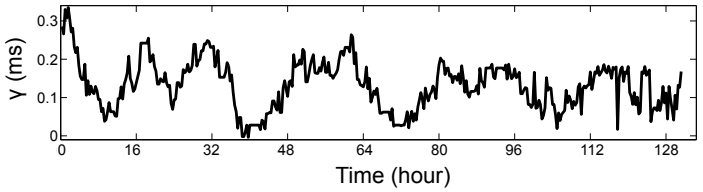
capture the instantaneous grid voltage signal at a high sampling rate. This hardware is used for the background study in this section only, whereas our new clock synchronization approach will use another simple hardware setup described in Section 5.1. The two nodes used in this section are (almost) perfectly synchronized in time using GPS receivers placed at their respective locations. The collected voltage signals are aligned based on their GPS timestamps. As our measurement study presented in this section is conducted in a real-world power grid, it addresses the impact of the active load fluctuation and its time-varying geographic distribution.

4.2.1 LAN-scale Results. The voltage profiles of the two nodes in the same LAN are captured at the same time. They are sinusoidal, and Fig. 5(a) shows their data for ten sample ac cycles among a trace that spans one whole day. For the whole day, the $\gamma(t)$ has an average value of $\bar{\gamma} = 0.009$ ms and a standard deviation of 0.0032 ms. Thus, the angle difference is small and stable. Hence, the two voltage signals are almost perfectly synchronized with each other, as shown in Fig. 5(a).

4.2.2 WAN-scale Results. We now compare the voltage profiles of two nodes that are far apart in Singapore. In the first experiment, the two nodes are located at locations A and B that are



(a) 30 km apart over one day. Mean and standard deviation are 2.879 ms and 0.0609 ms, respectively.



(b) 15 km apart over five day. Mean and standard deviation are 0.1324 ms and 0.0628 ms, respectively.

Fig. 6. Angle difference between two locations.

about 30 km apart. The voltage signals at the two nodes for a sample of ten ac cycles are shown in Fig. 5(b). Notice that, compared with the previous LAN setting, the angle difference between the signals becomes noticeable. Since the angle difference can be affected by changing load distribution of the grid, we ascertain its stability throughout one day, which encompasses say both high load during daytime and low load during late night, as well as concomitant load redistributions. Fig. 6(a) profiles the angle difference over time, with a mean of $\bar{\gamma} = 2.8$ ms and a standard deviation of 0.0609 ms. This shows that the angle difference is small and stable.

We report a further experiment for two nodes deployed at locations B and C that are 15 km apart, as shown in Fig. 4. Fig. 6(b) shows the angle difference over five days, with an average of $\bar{\gamma} = 0.13$ ms and a standard deviation of 0.0628 ms. Thus, the standard deviation is similar to that obtained for locations A and B.

4.2.3 Discussion. A comparison between the LAN setting (Section 4.2.1) and the 15 km and 30 km WAN settings (Section 4.2.2) verifies that the angle difference increases with distance, which is consistent with intuition and knowledge of power engineering [9]. Moreover, the standard deviation of the angle difference remains small (i.e., less than 0.07 ms) over a longer trace that covers different days of the week (specifically, five days that include the weekend), which gives further confirmation that the angle difference is stable.

5 GRID TIME PROTOCOL

We propose a new clock synchronization protocol, Grid Time Protocol (GTP), that utilizes an ac power grid's voltage as a reliable and extrinsic periodic signal to measure asymmetric delays individually, thereby achieving resilient clock synchronization between a master and slave connected to the same grid. Section 5.1 presents a method to capture the ac voltage signal. The core design of GTP, including formal derivations of the one-way delays, is presented in Section 5.2.

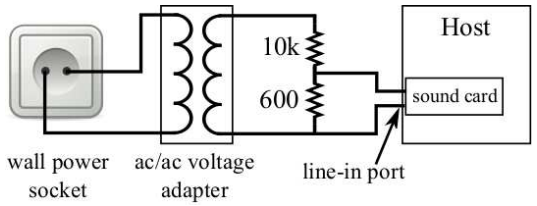


Fig. 7. Device for capturing powerline voltage.

5.1 Voltage Signal Capture

GTP works by leveraging the power grid voltage, a highly accessible and reliable reference signal that is impractical for the adversary to compromise. As Section 4.2 shows, between two geographically distant locations within the same grid, the voltage oscillates uniformly with a near-constant angle difference. Thus, by observing the angle of the grid voltage when a GTP slave sends or a GTP master receives a synchronization packet, we can accurately estimate the one-way packet transmission delays between the two nodes in both directions.

5.1.1 GTP Hardware. Our reference implementation of GTP uses the hardware design shown in Fig. 7. GTP analyzes the sinusoidal voltage signals captured by a commodity PC's sound card. A simple but key hardware device, for both the GTP slave and master, is thus a voltage sensor capable of capturing the subject voltage signal accurately. Off-the-shelf ac/ac voltage adapters can be used as the voltage sensor. However, the output voltages of most ac/ac adapters are higher than the range of the line-in input of PC's sound card. Thus, a voltage divider is needed to interface the ac/ac adapter and the sound card. In our hardware shown in Fig. 7, the voltage divider reduces the peak-to-peak voltage of 17 V given by the ac/ac adapter to 1 V for the sound card. A software application module written in C++ reads the line-in port data from the sound card's driver, which is sampled at 44 kHz in real time. Specifically, the driver continuously samples the line-in signal, and returns a block of data at one-second intervals to GTP.

The prototype hardware device shown in Fig. 7 is mainly for commodity desktop and single-board computers. Using sound card as the sampling device has two salient advantages. First, computers are generally equipped with sound cards and their operating systems already provide unified access interfaces. This ensures that GTP is highly portable. Second, as analyzed in Section 5.2.2, the voltage sampling rate is an important factor in GTP's synchronization error, and sound cards provide a sufficiently high sampling rate for small errors. In contrast, customized sampling devices of comparable sampling rates are often expensive. Single-board computers, e.g., Raspberry Pi and BeagleBone boards, often have add-on and built-in analog-to-digital converters to sample the voltage sensor. For battery-powered nodes without direct access to the power grid, we can use a circuit [19] to sense wirelessly powerline electromagnetic radiations, which are correlated with the ac voltage signal.

5.1.2 Security of Voltage Signal. Although the threat model considered in this paper is the packet delay attack defined in Section 3.1.2, this section discusses the security of the voltage signal as well, since GTP additionally uses this signal. Tampering with power grid voltage signal often raises large economic and logistical barriers for potential attackers. We discuss the following three possible cases where the attacker may affect the voltage signal.

First, the attacker may inject high-frequency noises, similar to the signals generated by power-line communication devices, into related power lines to introduce tiny voltage waveform distortions. However, such high-frequency noises can be removed by a low-pass filter added after the

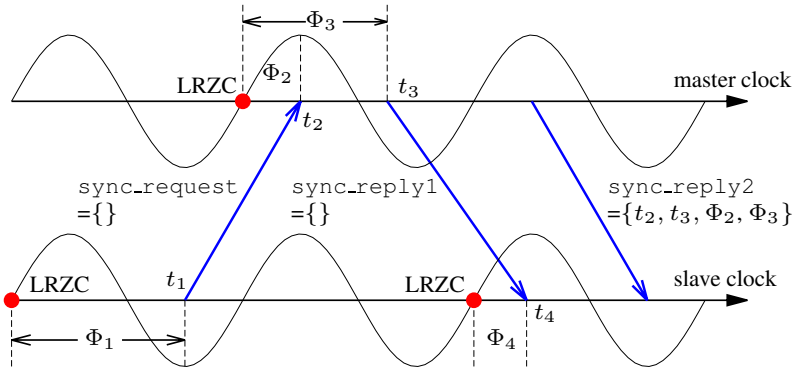


Fig. 8. Illustration of GTP operations. The lower and upper sinusoids represent the voltage signals captured by the slave and the master, respectively. The Φ_2 and Φ_3 are computed after sync_reply1 is sent. The Φ_1 and Φ_4 can be computed after sync_reply2 is received.

voltage divider in Fig. 7. In addition, the attacker may add harmonic sources, such as renewable generators, in the power grid. However, the voltage harmonics often have much lower amplitudes than the base frequency signal. Moreover, the harmonics with frequencies that are integer multiples of the base frequency can be suppressed/removed by a low-pass filter.

Second, by manipulating the geographic distribution of the load of a power grid (e.g., by disconnecting/connecting a sufficiently large load to the grid), the attacker may affect the voltage angle difference between two remote locations in the grid. However, this attack would be extremely challenging economically and logistically, but it can only cause a maximum synchronization error of 10 ms in a 50 Hz grid, because its effect is similar to a poor calibration of the GTP system. In fact, if the attacker is powerful enough to obtain the required physical access, disconnecting the targeted area from the utility grid will likely be more attractive to the attacker than compromising clock synchronization, as the compromise will leave a significant physical footprint immediately.

Third, the attacker may connect an RC circuit in series to a power distribution line to introduce additional voltage angle difference and affect the accuracy of GTP. This attack requires physical access to the power line, which is logistically difficult. Moreover, its effect is also similar to a poor calibration to GTP, which would incur a maximum synchronization error of 10 ms only. If the attacker had gained physical access to the power line, simply cutting it would be a more effective attack.

Given the above considerations, in this paper, we assume that the voltage signal is intact.

5.2 Principle of GTP

This section presents the working principles of GTP.

5.2.1 GTP Operations. The basic operations of GTP are illustrated in Fig. 8. In the figure, the sinusoids in the bottom and upper halves represent the voltage signals captured respectively by the slave and master. A synchronization session is initiated periodically or on an on-demand basis. It consists of the transmissions of a request packet and two reply packets. Specifically, the GTP slave program transmits a `sync_request` packet to the master, and records the slave's clock value t_1 when the packet is transmitted. When the master receives the `sync_request`, it records its current clock value t_2 and constructs a `sync_reply1` packet. The master records its clock value t_3 when the `sync_reply1` is transmitted. When the slave receives the `sync_reply1` packet, it records

its current clock value t_4. After the master has transmitted the `sync_reply1`, the master program identifies the voltage samples that correspond to the *last rising zero crossings* (LRZCs) of the voltage signal prior to the time instants t_2 and t_3, respectively. For instance, in Fig. 8, the time instants t_2 and t_3 share the same LRZC. Note that, if a new voltage cycle starts between t_2 and t_3, these two time instants will have different LRZCs. Then, the master program computes the elapsed times from t_2's LRZC to t_2 and t_3's LRZC to t_3, which are denoted by Φ_2 and Φ_3, respectively. After that, the master program constructs a `sync_reply2` packet containing t_2, t_3, Φ_2, and Φ_3 and transmits it to the slave. After receiving the `sync_reply2`, as illustrated in Fig. 8, the slave program identifies the LRZCs prior to the time instants t_1 and t_4, and the corresponding elapsed times Φ_1 and Φ_4. Finally, the slave program uses the approach presented in Section 5.2.2 to compute its clock offset from the master.

We now discuss several design and implementation issues:

- We use the LRZC as the reference point of the elapsed time calculation because it is a salient feature point that can be easily identified. As the voltage amplitude fluctuations caused by varying reactive loads of a power grid have little (if any) impact on the positions of zero crossing points, GTP is insensitive to the changes of reactive loads. Note that Φ_1, Φ_2, Φ_3, and Φ_4 are obtained by counting voltage samples – they also do not depend on the voltage amplitude.
- If we ignore the angle difference between the slave and master, the zero crossings of their signals are synchronized. Our analysis in Section 5.2.2 addresses the angle difference when computing the slave's clock offset.
- As detailed in Section 5.1, the voltage signal is sampled at 44 kHz by a sound card. Like many sampling devices, the sound card's driver returns data block by block, where each block has 44 K samples. Thus, to identify the LRZC of any given clock value t, the slave/master program needs to wait until the data block covering t is available.
- As we will analyze in Section 5.2.2, GTP uses the packets `sync_request` and `sync_reply1` to measure the slave's clock offset, while the `sync_reply2` is an auxiliary packet to convey the timestamps t_2, t_3 and measurements Φ_2, Φ_3. With this auxiliary packet, we can decouple the tasks of timestamping the sending time instant of `sync_reply1` and the signal processing time of computing Φ_3. As a result, the signal processing will not impact the packet receive and send timestamps. Moreover, since we minimize the payload size of the `sync_reply1` packet by deferring the transmission of t_2 to `sync_reply2`, we minimize `sync_reply1`'s transmission delay. This helps GTP to meet the key condition that we will provide in Section 5.2.2 to accurately estimate the slave's clock offset.
- To simplify the slave program design, we postpone the signal processing of computing Φ_1 and Φ_4 until `sync_reply2` is received. From our measurements, locating the LRZC and computing the elapsed time Φ takes about 2 ms only on commodity computers. Thus, the voltage signal processing imposes little overhead.

5.2.2 Clock Offset Analysis. This section analyzes the offset between the slave's and master's clocks based on $\{t_1, t_2, t_3, t_4\}$ and $\{\Phi_1, \Phi_2, \Phi_3, \Phi_4\}$ that are recorded and measured during a synchronization session illustrated in Fig. 8.

We define Θ_1 and Θ_2 by

$$\Theta_1 = \begin{cases} \Phi_2 - \Phi_1, & \text{if } \Phi_2 - \Phi_1 \geq 0; \\ \Phi_2 - \Phi_1 + p, & \text{otherwise.} \end{cases} \tag{1}$$

$$\Theta_2 = \begin{cases} \Phi_4 - \Phi_3, & \text{if } \Phi_4 - \Phi_3 \geq 0; \\ \Phi_4 - \Phi_3 + p, & \text{otherwise.} \end{cases} \tag{2}$$

When a new ac cycle starts during the transmission of the `sync_request` or the `sync_reply1` packet, the $\Phi_2 - \Phi_1$ or $\Phi_4 - \Phi_3$ can be negative. In this case, we add one ac cycle time p, as given in Eqs. (1) and (2). From the descriptions in Section 5.2.1, Φ_i is the elapsed time from the LRZC to some voltage sample. Thus, we have $0 \leq \Phi_i < p$. From Eqs. (1) and (2), we can verify that $0 \leq \Theta_1 < p$ and $0 \leq \Theta_2 < p$.

The one-way time delays for transmitting `sync_request` and `sync_reply1` can be longer than one ac cycle. To account for this possibility, we use i to denote the non-negative integer number of ac cycles elapsed since the time of sending `sync_request` until the time of receiving it at the master, and j to denote the non-negative integer number of ac cycles elapsed since the time of sending `sync_reply1` until the time of receiving it at the slave. Moreover, as discussed in Section 4.2, a voltage angle difference γ exists between the slave's and master's observations of the signal. A positive angle difference means that the slave's voltage signal leads the master's. Denoting by τ_1 and τ_2 the *actual* one-way delays of transmitting the `sync_request` and `sync_reply1` packets, respectively, we have $\tau_1 = \Theta_1 + i \cdot p + \gamma$ and $\tau_2 = \Theta_2 + j \cdot p - \gamma$. Thus, if i and j can be estimated, we can estimate the one-way delays τ_1 and τ_2, as well as the offset between the slave's and master's clocks as $\Delta = (t_2 - \tau_1) - t_1 = (t_3 + \tau_2) - t_4$. The RTT computed by $\text{RTT} = (t_4 - t_1) - (t_3 - t_2)$ must satisfy

$$\text{RTT} = \tau_1 + \tau_2 = \Theta_1 + \Theta_2 + (i + j) \cdot p. \tag{3}$$

However, estimating the integers i and j from Eq. (3) is an underdetermined problem with *integer ambiguity*. In Section 6, we will derive a condition that solves the ambiguity, thereby ensuring the resilience of GTP against the packet delay attacks.

6 SECURE GTP

This section derives the condition that solves the ambiguity, presents comparative evaluation of GTP and NTP under the packet delay attack, and discusses a resilience policy for GTP.

6.1 Condition for Secure GTP

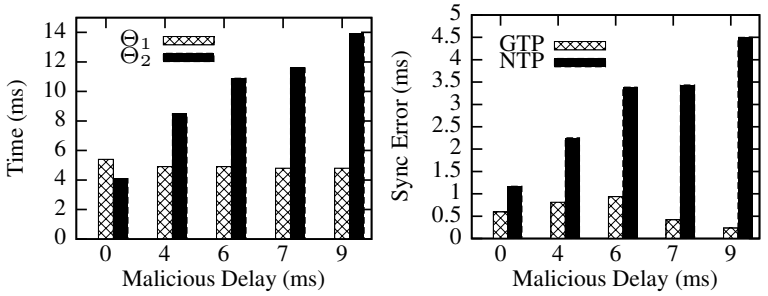
We have the following proposition.

Proposition 1. $\text{RTT} = \Theta_1 + \Theta_2$ *is a sufficient and necessary condition for GTP to unambiguously estimate the two one-way delays as* $\tau_1 = \Theta_1 + \gamma$ *and* $\tau_2 = \Theta_2 - \gamma$*, respectively.*

Proof. If $i \geq 1$ or $j \geq 1$, there is no additional information to help us assign the time $(i + j) \cdot p$ to τ_1 and τ_2. Thus, if and only if $i = 0$ and $j = 0$ (i.e., $\text{RTT} = \Theta_1 + \Theta_2$), GTP can unambiguously estimate τ_1 and τ_2 as stated. □

The GTP programs can run at high CPU priority (e.g., as kernel interrupt handler) to improve the accuracy of the timestamps $\{t_1, t_2, t_3, t_4\}$, making it less prone to delay by other computation. Moreover, as Θ_1 and Θ_2 are obtained through voltage signal processing, their accuracy is subject to the resolution of the voltage signal capture. Thus, in practice, the RTT may not exactly equal $\Theta_1 + \Theta_2$. The slave can check the condition in Proposition 1 by comparing $\text{RTT} - \Theta_1 - \Theta_2$ with a threshold η. From Eq. (3), ideally $\text{RTT} - \Theta_1 - \Theta_2$ is a multiple of p. Thus, we may set η to be $p/2$. If $|\text{RTT} - \Theta_1 - \Theta_2| < \eta$, the slave computes two offsets between the slave's and master's clocks as $\Delta_1 = (t_2 - \tau_1) - t_1$ and $\Delta_2 = (t_3 + \tau_2) - t_4$. Then, the slave uses the average offset $(\Delta_1 + \Delta_2)/2$ to calibrate its own clock.

The sufficient and necessary condition in Proposition 1 is closely related to the voltage measurements. The following proposition gives a necessary condition that depends on the RTT only. The proof is omitted due to space constraints and can be found in Appendix B of [18]. This necessary



(a) Θ_1 and Θ_2 in GTP. (b) Synchronization errors.

Fig. 9. GTP vs. NTP in LAN.

condition allows us to quickly assess whether a cyber network can support GTP without any need to deploy the GTP hardware.

Proposition 2. *RTT $< 2p$ is a necessary condition for GTP to unambiguously estimate the two one-way delays as $\tau_1 = \Theta_1 + \gamma$ and $\tau_2 = \Theta_2 - \gamma$, respectively.*

For a 50 Hz power grid, the necessary condition given by Proposition 2 is RTT $<$ 40 ms. Section 6.4 presents extensive RTT measurements in Singapore in the absence of delay attack. The results show that the RTT rarely exceeds 40 ms. This underlines the GTP's practicality in city-scale networks. In the rest of this paper, by *secure GTP conditions*, we refer to the conditions given by Propositions 1 and 2.

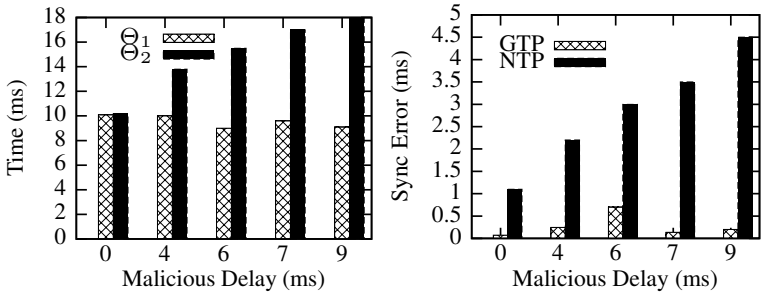
From the proof of Proposition 2, although we cannot unambiguously estimate τ_1 and τ_2 in Case 2.2, Case 2.3, and Case 3, we still have useful information about τ_1 and τ_2. For instance, in Case 2.2 and 2.3, there are two possible solutions for τ_1 and τ_2: $\tau_1 = \Theta_1 + \gamma$, $\tau_2 = \Theta_2 + p - \gamma$; or $\tau_1 = \Theta_1 + p + \gamma$ and $\tau_2 = \Theta_1 - \gamma$. Thus, it is possible to assess the symmetric link assumption up to some uncertainty.

The synchronization error of GTP depends on the accuracy of the timestamps $\{t_1, t_2, t_3, t_4\}$, the calibration of γ, and the sound card's resolution in sampling the voltage signal. First, although hardware-level packet send/receive timestamping by the network interface card (NIC), as prescribed in the Precision Time Protocol [11], will achieve best accuracy, the requirement for special NIC hardware could hinder adoption significantly. Thus, GTP timestamps in software. Commodity PCs generally have $\pm 15\,\mu$s packet timestamp errors [2], which is quite acceptable. Second, from our WAN-scale measurements in Section 4.2, the standard deviation of γ, which characterizes its uncertainty, is about 60 μs. Third, as the sound card adopts a sampling rate of 44 kHz, the time resolution is $\frac{1}{44\,\text{KHz}} = 22\,\mu$s. Thus, we expect that GTP will achieve sub-ms (down to 0.1 ms) synchronization accuracy. This accuracy will be benchmarked in Section 6.2.

6.2 GTP Performance

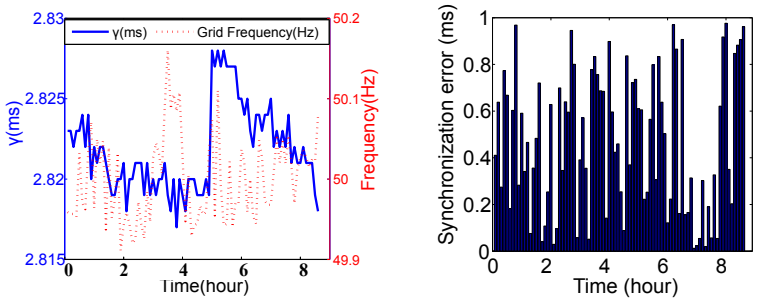
We conduct comparative experiments based on the setup in Fig. 1(b) to measure the synchronization errors of GTP and NTP in the presence of asymmetric network delays. In the experiments, both the slave and the stratum-2 time master are equipped with the GTP hardware shown in Fig. 7. We conduct four sets of experiments in two different countries (Singapore and China) under the LAN and WAN settings, respectively.

6.2.1 GTP Performance in Singapore. In the LAN-scale experiments, all the nodes shown in Fig. 1(b) are located on the same floor of a building. We use ARP spoofing (see Section 3.2) to implement the packet delay attack. The malicious delay increases from zero to 9 ms. As measured in Section 4.2.1,



(a) Θ_1 and Θ_2 in GTP. (b) Synchronization errors.

Fig. 10. GTP vs. NTP in WAN.



(a) Grid frequency and $\gamma(t)$. (b) Synchronization error.

Fig. 11. Grid frequency, $\gamma(t)$, and GTP synchronization error between locations A and B, in the absence of delay attack.

the angle difference γ is near-zero under this setting. Thus, we set $\bar{\gamma} = 0$ for the calculation of the slave clock offset. Fig. 9(a) shows the Θ_1 and Θ_2 measured by GTP when the malicious delay increases from zero to 9 ms. As the attack delays the reply packet, from Fig. 9(a) we can see that Θ_2 increases with the malicious delay. Fig. 9(b) shows the synchronization errors of GTP and NTP. We can see that GTP's synchronization errors are within one ms, which is consistent with our analysis in Section 6.1. Note that we currently run GTP at normal user priority. A higher priority execution will reduce the randomness of packet timestamping and thus GTP's synchronization errors. In contrast, as NTP is unaware of the link asymmetry, its synchronization error is about half of the malicious delay.

In the WAN-scale experiments, the slave and master nodes are at locations A and B shown in Fig. 4. From the angle difference measurements in Section 4.2.2, we set $\bar{\gamma} = 2.8$ ms for computing the slave clock offset. In the first set of WAN-scale experiments, there are no delay attacks. Fig. 11 shows the grid frequency, the $\gamma(t)$ between the two locations, and the GTP synchronization error. We note that the fluctuations of grid frequency and the γ are caused by load fluctuations and varying load distribution. From the figure, during the experiment, the γ is within $(2.81, 2.83)$ ms. Its impact on GTP's synchronization error is largely reduced by the calibration $\bar{\gamma} = 2.8$ ms. Therefore, as shown in Fig. 11(b), GTP maintains sub-ms errors.

Table 1. GTP performance in Hangzhou, China (physical unit: ms)

Scale	Θ_1	Θ_2	$\Theta_1 + \Theta_2$	RTT	GTP error	NTP error
LAN	8.2	3.8	12.0	12.02	0.9	1.3
WAN	5.4	4.8	10.2	9.2	1.8	2.1

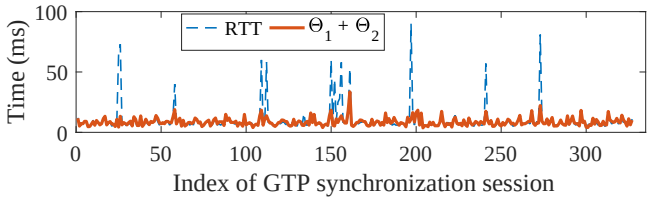


Fig. 12. GTP's $\Theta_1 + \Theta_2$ and RTTs in a WAN-scale experiment in Hangzhou, China. 97% of the synchronization sessions meet the secure GTP conditions.

In the second set of WAN-scale experiments, we launch delay attacks. The ARP spoofing attack is launched in the LAN of the slave node. Fig. 10(a) shows the Θ_1 and Θ_2 measured by GTP when the malicious delay increases from zero to 9 ms. The Θ_2 increases with the delay. Fig. 10(b) shows the synchronization errors of GTP and NTP. Notice that the results are similar to those under the LAN setting.

In all above experiments, the secure GTP conditions given in Propositions 1 and 2 are satisfied. As a result, GTP can maintain sub-ms synchronization errors.

6.2.2 GTP Performance in Hangzhou, China. In the LAN-scale experiments, the slave and master nodes are located in two different laboratories within a university campus. Note that in these experiments conducted in Hangzhou, we do not calibrate the GTP system using $\bar{\gamma}$ and let it be a part of the synchronization error. Table 1 shows the average values of GTP's various measurements and synchronization errors, as well as NTP's synchronization error over 320 synchronization sessions. The secure GTP conditions are satisfied. The GTP's average synchronization error is 0.9 ms only, compared with NTP's average synchronization error of 1.3 ms.

In the WAN-scale experiments, the slave node is 15 km apart from the master node. Table 1 also shows GTP's various internal quantities and synchronization errors in this set of experiments. We note that the RTT in this WAN-scale experiment is smaller than that in the LAN-scale experiment. This is because, to overcome a firewall between the two laboratories, the LAN-scale experiments are conducted in a virtual private network (VPN) that introduces some network delay, whereas the WAN-scale experiments are conducted with direct connections. The GTP's average synchronization error is 1.8 ms, whereas the NTP's is 2.1 ms. Fig. 12 shows GTP's $\Theta_1 + \Theta_2$ and RTTs over time, where 97% of the synchronization sessions meet the secure GTP conditions. Section 6.4 will discuss how to deal with the synchronization sessions that do not meet the secure GTP conditions.

6.3 Impact of Grid Phase

Section 4.1 explains that most power grids have three phases, i.e., ϕ_l has three possible values of 0, $-2\pi/3$, and $2\pi/3$. If the slave and master are on different power grid phases, an additional angle difference of $-\frac{p}{3}$ ms or $\frac{p}{3}$ ms will be present, which is ± 6.67 ms in a 50 Hz grid. This additional angle difference can be considered a part of γ. Hence, GTP and its clock offset analysis in Section 5 still hold.

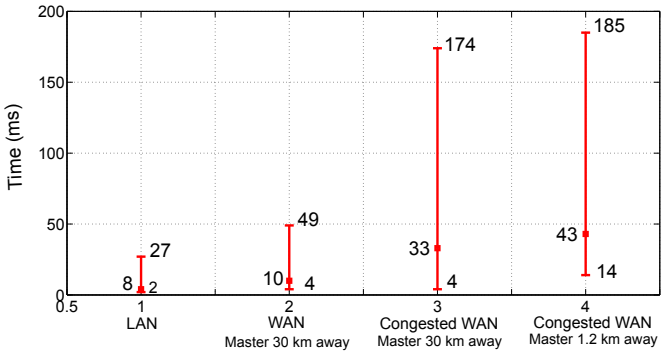


Fig. 13. RTT statistics in a LAN and a city-scale WAN. Square dot represents average; error bar marks the minimum and maximum.

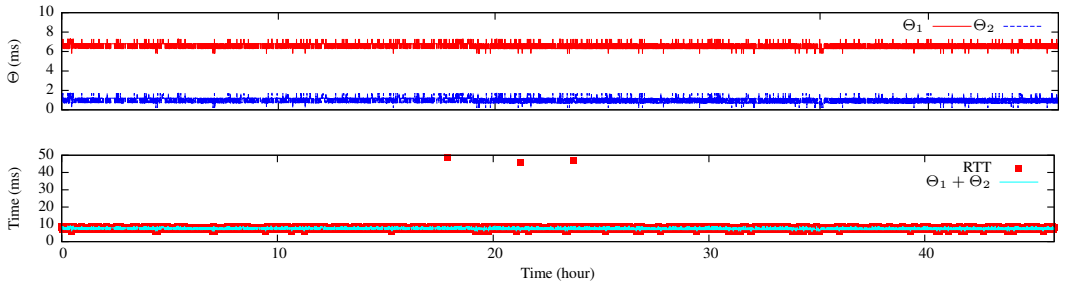


Fig. 14. Θ_1, Θ_2, and RTT measured by GTP for two nodes 30 km apart over two days.

The phase information (i.e., R-, Y-, and B-phase) of power outlets is generally available to facility operators. Viswanathan *et al.* [23] present an approach for a slave node to identify the grid phase of its power supply using sampled ac voltage signal. Based on the phase information, a GTP slave can add -6.67 ms or 6.67 ms to the calibrated γ. When the phase information is not available, GTP may have an additional ± 6.67 ms synchronization error. As GTP on the same grid phase yields a sub-ms synchronization error (cf. Section 6.2), GTP without the grid phase information can achieve a synchronization error bound of ± 8 ms in the presence of link asymmetry. In contrast, NTP in WAN exhibits errors of up to 80 ms [25].

6.4 Resilient GTP

This section presents extensive measurement results to show that the secure GTP conditions (Section 5.2) are mostly satisfied in a city-scale network in diverse natural settings (including congestions), in the absence of packet delay attacks. However, an attack if present could add sufficient delay deliberately to breach the conditions persistently. Thus, we will next present a resilience policy for GTP to deal with these attacks. We will also discuss key differences between GTP and NTP in terms of their security against the packet delay attack.

6.4.1 Validating the Secure GTP Conditions. This section presents our extensive measurements to validate the secure GTP conditions given in Propositions 1 and 2 in the real world.

In the first set of experiments, we measure the RTT, Θ_1, and Θ_2 under normal network conditions. The first error bar in Fig. 13 shows the range of the RTT when the GTP slave and master are in the

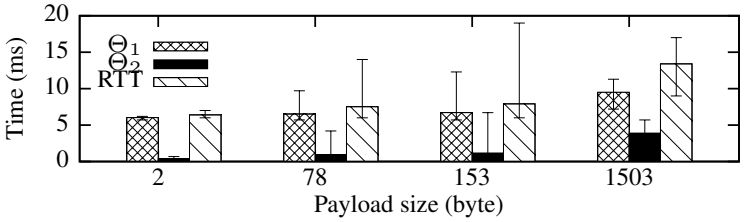


Fig. 15. Impact of payload size on GTP for two nodes 30 km apart.

same LAN. We can see that the RTT is always smaller than 40 ms, which ensures the necessary condition in Proposition 2. Analysis of the measured RTT, Θ_1, and Θ_2 shows that the more stringent sufficient condition in Proposition 1 is also always met. Then, we perform experiments over two days under a WAN setting when the two nodes are located at locations A and B shown in Fig. 4. The slave initiates a GTP session every 20 seconds. The upper half of Fig. 14 shows the Θ_1 and Θ_2 measured by GTP over two days. The bottom half of Fig. 14 shows the values of $\Theta_1 + \Theta_2$ and the RTT measured by GTP. During the two days, the experiment encounters only three instances in which the secure GTP conditions are not satisfied, which are shown as the three RTT outliers in the bottom half of Fig. 14. The second error bar in Fig. 13 shows the range and average of the RTT for the duration of the experiment.

In the second set of experiments, we intentionally create network congestion to understand its impact on GTP. Under the WAN setting, we use Ostinato, a network traffic generator, on two other hosts in the Ethernet of the GTP slave to generate a massive amount of background traffic to congest the network. The third error bar in Fig. 13 shows the range of the RTT. The data traffic clearly causes the RTT to increase. However, a majority of the RTT measurements remain below 40 ms; further analysis shows that the more stringent sufficient condition in Proposition 1 is also satisfied in many of the synchronization sessions. We conduct also a similar experiment for another pair of GTP slave and master that are about 1.2 km apart, where the slave's Ethernet experiences intentionally introduced congestion. We obtain similar results as shown by the fourth error bar in Fig. 13.

In the third set of experiments, we evaluate the impact of the GTP packet's payload size on Θ_1, Θ_2, and RTT. Although our current implementation of GTP carries basic information (e.g., sequence number) only in the `sync_request` and `sync_reply1` packets, a future implementation of GTP with extended features may use larger packets. Hence, Fig. 15 explores the effects of different payload sizes for two nodes at locations A and B. It shows that even when the payload size increases up to 1.5 Kbytes, the secure GTP conditions are always met during the experiments.

6.4.2 GTP Resilience Policy. The previous experiments show that the secure GTP conditions are predominantly satisfied in a real-world city-scale network, even under deliberately introduced severe network congestions. To handle more extreme situations, such as malicious packet delay attacks or extremely persistent congestions that (though highly unlikely) cannot be totally ruled out, this section proposes a resilience policy for a GTP slave to take advantage of multiple GTP masters for resilient results. The policy consists of the following two rules.

First, in each synchronization session, the slave can synchronize with multiple *default* GTP masters, and use the master that satisfies the secure GTP conditions and gives the minimum RTT $-$ Θ_1 $-$ Θ_2 to calibrate its clock. With this mechanism, if one or more default GTP masters satisfy the secure GTP conditions, the synchronization is successful and trustworthy. As discussed in Section 5.2.2, the measurements of RTT, Θ_1, and Θ_2 are subject to packet timestamp and voltage

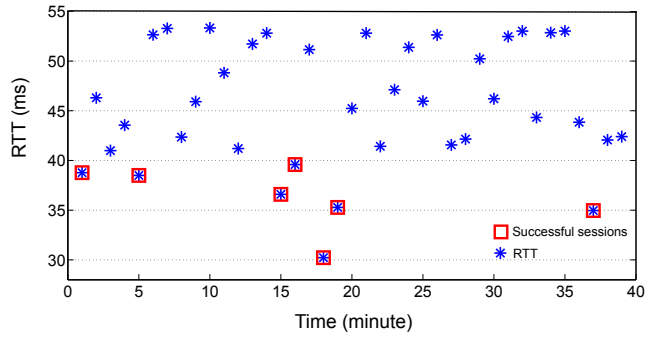


Fig. 16. RTT between slave and master when the master experiences network congestion. 18% synchronization sessions meet secure GTP conditions.

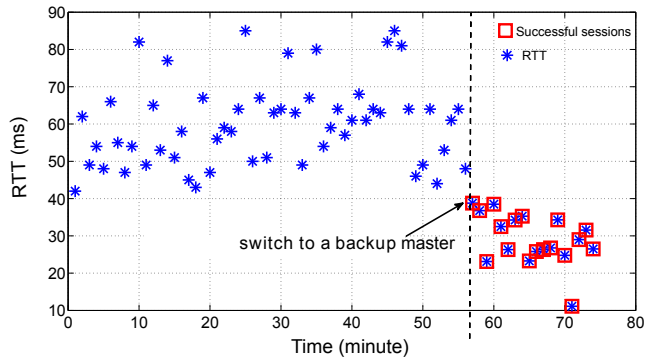


Fig. 17. RTT between slave and default master when the master experiences network congestion and packet delay attack, respectively. In the case of the attack, the slave switches to a backup master after the timeout (55 minutes).

signal processing errors, so that $\mathrm{RTT} - \Theta_1 - \Theta_2$ can be considered a quality metric. Hence, GTP chooses the best master according to the minimum $\mathrm{RTT} - \Theta_1 - \Theta_2$ for the synchronization.

Second, for a default GTP master, in case the secure GTP conditions are persistently not satisfied over an entire *GTP timeout* duration, the slave will replace this default master with a new *backup* GTP master. As shown in Section 6.4.1, even under severe network congestions, we still expect the secure GTP conditions to be satisfied in at least some (according to the experiments, many) synchronization sessions. Hence, the majority of network congestions, particularly transient ones, will not present any problems. But should the synchronization be unsuccessful throughout the timeout duration, GTP gives an unambiguous signal to the slave to react by looking for a better master to replace the non-performing one.

We now illustrate how the second rule of the resilience policy works in practice. In this experiment, the slave resynchronizes with a default master every minute. The GTP timeout is set to be 55 minutes. Similar to the experiment in Section 6.4.1, we use Ostinato to create congested access to the master. In the first example, no packet delay attacks are introduced. Fig. 16 shows the RTT over time. During the experiment, the secure GTP conditions are occasionally satisfied and the

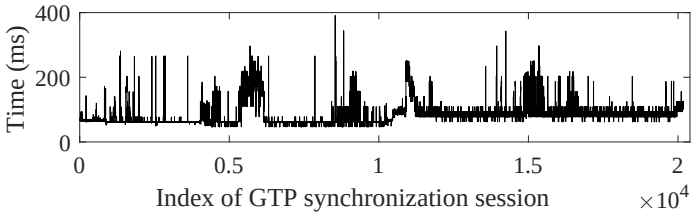


Fig. 18. RTTs over 84 hours between two nodes in Singapore and Hangzhou, China. Mean and standard deviation are 85.7 ms and 33.4 ms, respectively.

corresponding synchronization sessions are successful. Thus, the slave can keep using this default master. In the second example, we create a delay attack in which every GTP reply is maliciously delayed by 22 ms. Fig. 17 shows the RTT over time. As the secure GTP conditions are violated throughout the GTP timeout, the slave switches to a backup master at the end of the timeout. The new master is not under attack, and the slave knows for sure that it can successfully synchronize with the new master because the secure GTP conditions are met.

6.4.3 More Discussions on GTP vs. NTP. In this section, we analytically compare the synchronization errors of GTP and NTP. Denote by τ_1 and τ_2 the two one-way packet transmission delays. For fair comparison, we require RTT $= \tau_1 + \tau_2 < 2p$ for both GTP and NTP, to limit the impact of the link asymmetry. Under this condition, GTP can mostly maintain a sub-ms synchronization error. This error is mainly due to uncertainty of the angle difference γ and the time granularity of voltage sampling. NTP's synchronization error is given by $\frac{\tau_1-\tau_2}{2}$. The asymmetry $\tau_1 \neq \tau_2$ can be caused by natural route asymmetry or dissimilar load between the two network directions, or a packet delay attack. In a LAN, one-way transmission delay is often small. As a result, the synchronization error $\frac{\tau_1-\tau_2}{2}$ may reach its upper bound of p (e.g., 20 ms in a 50 Hz grid). It is significantly larger than the sub-ms error bound achieved by GTP.

7 LONG-HAUL GTP

7.1 Secure GTP Conditions at Inter-city Scale

From Proposition 2, in a power grid, a necessary condition for GTP to be resilient against packet delay attacks is that the RTT is smaller than $2p$ (e.g., 40 ms in a 50 Hz grid). From the evaluation results in Sections 6.2 and 6.4, the secure GTP conditions can be satisfied mostly in a city-scale network. However, such conditions can be hardly satisfied for the nodes that are long distances apart. For instance, in the Western Interconnection, a wide-area synchronous grid in the U.S., the longest geographic distance between two locations is up to 3,000 km. In China's Southern Power Grid, the distance is up to 2,000 km. Over such long distances, the RTT will be most likely longer than $2p$.

We conduct an experiment to measure the RTTs between two nodes that are deployed in Singapore and Hangzhou, China, respectively, with a geographic distance of about 16,000 km. The two nodes are equipped with GPS receivers, so that we can also measure the one-way delays between the two nodes. Note that although Singapore and Hangzhou have different power grids, the RTT measurements give us an understanding about the RTT over long distances, and the one-way delay traces are used to drive the evaluation of GTP in Section 7.3. Fig. 18 shows the RTT measurements over 84 hours that cover three days of a week, including high-traffic periods in daytime, low-traffic periods at night, and a weekend. The minimum, mean, and maximum of the RTTs are 46 ms, 85.7 ms, and 391 ms, respectively. Thus, the secure GTP conditions are not satisfied.

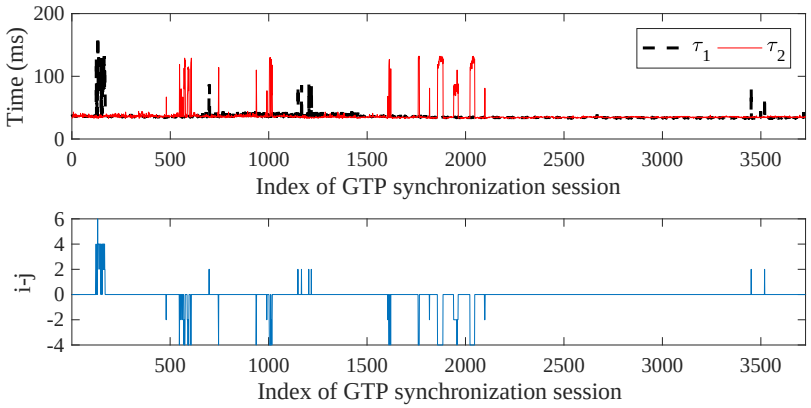


Fig. 19. τ_1, τ_2 and $(i - j)$ for a Singapore-Hangzhou link.

Different from Section 6, which focuses on the resilience of GTP against packet delay attacks, the rest of this section will focus on investigating the performance gain of GTP with respect to NTP in the presence of natural network delays over long distances, i.e., beyond the scale of individual cities. We refer to this long-distance (i.e., inter-city scale) variant of GTP as *long-haul GTP*. Overall, we envision a GTP hierarchy in which (i) long-haul GTP operates over selected long-haul backbone links that are secured against packet delay attacks, and (ii) secure GTP (presented in Section 6) operates over general (i.e., possibly unprotected) network links within cities. Securing the strategic long-haul links requires additional safeguards, e.g., ensuring that the endpoints are directly connected physically so that it is impossible for any intermediate nodes to insert malicious delays. The additional efforts can be feasible, however, since these strategic links are much smaller in number than general network links within cities. Hence, we evaluate the performance of long-haul GTP under non-malicious network conditions only.

7.2 Performance Analysis of Long-haul GTP

When the secure GTP conditions are not satisfied, estimating i and j from Eq. (3) is an underdetermined problem with integer ambiguity. Specifically, with the measured RTT, Θ_1, and Θ_2, we can compute $i + j$ from Eq. (3) but cannot estimate the individual i and j. Long-haul GTP assumes that $i = j$ when the computed $i + j$ is even. Thus, it estimates i and j as

$$\hat{i} = \hat{j} = \frac{\text{RTT} - (\Theta_1 + \Theta_2)}{2 \cdot p}, \quad \text{when } i + j \text{ is even.} \tag{4}$$

With the $\hat{i}$ and $\hat{j}$, the offset between the slave's and master's clocks can be estimated by the approach presented in Section 5.2.2. When $i + j$ is odd, the long-haul GTP will skip the current synchronization session and start a new one.

We use the one-way network delay traces between Singapore and Hangzhou to evaluate the assumption of $i = j$ when $i + j$ is even. Fig. 19 shows the measured τ_1, τ_2, and $i - j$ in 3,953 synchronization sessions over 22 hours. We can see that most of the time, i and j have the same value. The link experiences bursts of increased one-way delays. The $i + j$ is even and odd for 88% and 12% of the time, respectively. Among the synchronization sessions for which $i + j$ is even, the probability that the assumption of $i = j$ holds is 96.3%. Thus, Eq. (4) will yield correct i and j estimates with high probability in the presence of natural network delays.

Proposition 3. *When $i = j$ and the voltage angle difference between the slave and the master (i.e., γ) is calibrated,*

$$|\varepsilon_{NTP}| - |\varepsilon_{GTP}| = \left|\frac{\Theta_1 - \Theta_2}{2}\right| \in \left[0, \frac{p}{2}\right], \tag{5}$$

where ε_{NTP} and ε_{GTP} are the NTP's and GTP's synchronization errors, respectively.

Proof. When $i = j$, NTP's synchronization error is given by

$$\varepsilon_{NTP} = \frac{\tau_1 - \tau_2}{2} = \frac{\Theta_1 - \Theta_2}{2} + \frac{(i-j)p}{2} + \gamma = \frac{\Theta_1 - \Theta_2}{2} + \gamma.$$

GTP's synchronization error is $\varepsilon_{GTP} = \gamma$. Thus, if γ is calibrated (i.e., $\gamma = 0$), we have Eq. (5). □

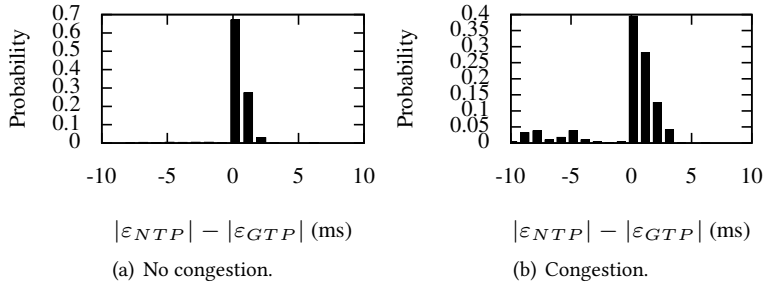
Since the long-haul GTP assumes $i = j$ if $i + j$ is even and this assumption holds with high probability (e.g., 96.3% for the Singapore-Hangzhou link), from Proposition 3, the long-haul GTP outperforms NTP with high probability.

7.3 Performance Evaluation of Long-haul GTP

To evaluate the result given by Proposition 3, we conduct simulations driven by the one-way network delay traces collected by the two nodes deployed in Singapore and Hangzhou over a time period of 22 hours. Moreover, based on the traces of τ_1 and τ_2, we generate traces of i, j, Θ_1, and Θ_2 by assuming $p = 20$ ms and $\gamma = 0$ as follows: $i = \lfloor \tau_1/p \rfloor$, $j = \lfloor \tau_2/p \rfloor$, $\Theta_1 = \tau_1 \bmod p$, and $\Theta_2 = \tau_2 \bmod p$. The collected and generated data traces are used to drive the simulations to evaluate NTP's and long-haul GTP's synchronization errors. Fig. 20(a) shows the distribution of $|\varepsilon_{NTP}| - |\varepsilon_{GTP}|$ when $i + j$ is even. We can see that most $|\varepsilon_{NTP}| - |\varepsilon_{GTP}|$ records are within $[0, 3]$ ms. When $i + j$ is even and $i \neq j$, the $|\varepsilon_{NTP}| - |\varepsilon_{GTP}|$ can be negative, which indicates that NTP outperforms GTP. However, this situation occurs with a low probability of 2.6% only when $i + j$ is even. Note that because the Singapore-Hangzhou link is quite symmetric, in this simulation, the performance gain of GTP is small (i.e., within 3 ms). In Fig. 20(b), we intentionally create network congestion to understand its impact on long-haul GTP's synchronization errors. For the same Singapore-Hangzhou link, we use Ostinato to congest severely the LAN of the Singapore synchronization node. We can observe more negative values of $|\varepsilon_{NTP}| - |\varepsilon_{GTP}|$, where better performance for NTP than GTP is indicated. However, when $i + j$ is even, these values are still the minority (15.4% only). Despite the congestion, GTP outperforms NTP most of the time, since most of the recorded $|\varepsilon_{NTP}| - |\varepsilon_{GTP}|$ are within $[0, 4]$ ms.

8 COMPARISON BETWEEN GRID-BASED TIME SYNCHRONIZATION PROTOCOLS

In this section, we compare GTP with other recent grid-based time synchronization protocols (e.g., [13, 24]) in terms of cost, coverage, and applicability. As discussed in Section 4.1, the possibility of deriving secure time information from electric network voltage (ENV) has been studied [24]. They use continuous fluctuations of the ENF to derive a time fingerprint (TiF), where the TiF captured by the slave is time-aligned within a trace of ac cycle lengths captured by the master. They also present an automatic phase identification approach for the slave node to identify its grid phase (i.e., R-, Y-, or B- phase of the power outlet), using a sampled ac voltage signal. Wireless extraction of accurate and robust time information from power line electromagnetic radiation (EMR), induced mainly by power line voltage oscillations at the rate of the ENF, has also been proposed [13]. They present a robust signal processing pipeline to overcome challenges of attenuation and noise arising in the wireless setting. Their validation shows that the EMR provides natural timestamps with median errors as low as 50 ms.



(a) No congestion. (b) Congestion.

Fig. 20. Distribution of $|\varepsilon_{NTP}| - |\varepsilon_{GTP}|$ for a Singapore-Hangzhou link.

GTP is different from the aforementioned grid-based time synchronization protocols [13, 24] in aspects of cost, coverage, and applicability. First, GTP uses a low-cost voltage sensor, rather than a highly customized hardware peripheral [24] for high-rate and high-precision sampling. Second, long-haul GTP achieves synchronization for two inter-city nodes connected to different power grids, whereas the prior work has demonstrated performance for nodes connected to the same grid only. Third, unlike the use of EMR [13], GTP is not susceptible to interference from proximate electromagnetic energy sources.

9 CONCLUSION

This paper exploited the ac power grid voltage as an extrinsic signal to limit the impact of asymmetric network delays on message-passing-based clock synchronization. The asymmetric delays can be caused by either natural asymmetric network conditions or packet delay attacks. Based on the synchronization properties of the voltages in a power grid, we designed GTP and analyzed the condition to unambiguously measure the one-way packet transmission delays, thereby achieving resilience against asymmetric network delays.

We showed that GTP admits a resilience policy that can significantly enhance its performance when multiple GTP masters are available. Specifically, if communications with any of the masters satisfy the secure GTP conditions, a slave's synchronization session will be successful with demonstrably trustworthy sub-ms accuracy. This desirable property is in contrast to the utility of multiple masters under NTP. GTP has promise for wide adoption in grid-connected distributed systems. In particular, it can meet the demands of applications that are currently served by NTP but desire better robustness against unfavorable network dynamics or resilience against malicious attacks. Examples of these applications include a utility's metering infrastructures for end users and advanced manufacturing processes that are based on Internet of Things technologies.

LIST OF ABBREVIATIONS

LRZCs	Last rising zero crossings in the sine voltage signal
Φ_i	The elapsed times from t_i's LRZC to t_i; $0 \le \Phi_i < p$
Θ_i	The two one-way time delays for transmitting `sync_request` and `sync_reply1`; $0 \le \Theta_i < p$
γ	The voltage angle difference between two different locations
Δ	Average clock offset. A positive or negative number
p	The nominal ac cycle time duration, e.g., 20 ms for a 50 Hz power grid

τ_i	The actual two one-way delays of transmitting GTP's `sync_request` and `sync_reply1` packets
RTT	Round-trip time
GTP	Grid Time Protocol
NTP	Network Time Protocol

REFERENCES

[1] 2019. NTP Security Analysis. https://www.eecis.udel.edu/~mills/security.html.

[2] Grenville Armitage, Mark Claypool, and Philip Branch. 2006. *Networking and online games: understanding & engineering multiplayer Internet games.* Wiley.

[3] Don Von Dollen. 2009. *Report to NIST on the Smart Grid Interoperability Standards Roadmap.* Technical Report. Electric Power Research Institute.

[4] Jeremy Elson, Lewis Girod, and Deborah Estrin. 2002. Fine-grained network time synchronization using reference broadcasts. *ACM SIGOPS Operating Systems Review* 36, SI (2002), 147–163.

[5] Saurabh Ganeriwal, Ram Kumar, and Mani B Srivastava. 2003. Timing-sync protocol for sensor networks. In *The 1st ACM Conference on Embedded Networked Sensor Systems (SenSys).*

[6] Tian Hao, Ruogu Zhou, Guoliang Xing, Matt W Mutka, and Jiming Chen. 2014. Wizsync: Exploiting Wi-Fi infrastructure for clock synchronization in wireless sensor networks. *IEEE Transactions on Mobile Computing* 13, 6 (2014).

[7] James Joshi. 2008. *Network Security: Know It All: Know It All.* Morgan Kaufmann.

[8] Hermann Kopetz and Wilhelm Ochsenreiter. 1987. Clock synchronization in distributed real-time systems. *IEEE Trans. Comput.* 100, 8 (1987).

[9] Prabha Kundur. 1994. *Power system stability and control.* McGraw-Hill.

[10] Kyohei Kurohane, Tomonobu Senjyu, Atsushi Yona, Naomitsu Urasaki, Tomonori Goya, and Toshihisa Funabashi. 2010. A hybrid smart AC/DC power system. *IEEE Transactions on Smart Grid* 1, 2 (2010), 199–204.

[11] Kang B Lee and J Eldson. 2004. Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems. In *Conference on IEEE 1588, Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems.*

[12] Liqun Li, Guoliang Xing, Limin Sun, Wei Huangfu, Ruogu Zhou, and Hongsong Zhu. 2011. Exploiting FM radio data system for adaptive clock calibration in sensor networks. In *The 9th Annual International Conference on Mobile Systems, Applications, and Services (MobiSys).*

[13] Yang Li, Rui Tan, and David KY Yau. 2018. Natural Timestamps in Powerline Electromagnetic Radiation. *ACM Transactions on Sensor Networks (TOSN)* 14, 2 (2018), 13.

[14] Zhenjiang Li, Wenwei Chen, Cheng Li, Mo Li, Xiang-Yang Li, and Yunhao Liu. 2012. Flight: Clock calibration using fluorescent lighting. In *The 18th Annual International Conference on Mobile Computing and Networking (MobiCom).*

[15] Xiong Liu, Peng Wang, Poh Chiang Loh, et al. 2011. A Hybrid AC/DC Microgrid and Its Coordination Control. *IEEE Trans. Smart Grid* 2, 2 (2011), 278–286.

[16] Miklós Maróti, Branislav Kusy, Gyula Simon, and Ákos Lédeczi. 2004. The flooding time synchronization protocol. In *The 2nd ACM Conference on Embedded Networked Sensor Systems (SenSys).*

[17] Tal Mizrahi. 2012. A game theoretic analysis of delay attacks against time synchronization protocols. In *International Symposium on Precision Clock Synchronization for Measurement, Control and Communication (ISPCS).*

[18] Dima Rabadi, Rui Tan, David KY Yau, and Sreejaya Viswanathan. 2017. Taming Asymmetric Network Delays for Clock Synchronization Using Power Grid Voltage. In *ACM Asia Conference on Computer and Communications Security (ASIACCS).*

[19] Anthony Rowe, Vikram Gupta, and Ragunathan Raj Rajkumar. 2009. Low-power clock synchronization using electromagnetic energy radiating from ac power lines. In *The 7th ACM Conference on Embedded Networked Sensor Systems (SenSys).*

[20] Jean-Charles Tournier and Otmar Goerlitz. 2009. Strategies to secure the IEEE 1588 protocol in digital substation automation. In *The 4th International Conference on Critical Infrastructures (CRITIS).*

[21] Jeanette Tsang and Konstantin Beznosov. 2006. A security analysis of the precise time protocol. In *International Conference on Information and Communications Security.* 50–59.

[22] Markus Ullmann and M Vogeler. 2009. Delay attacks Implication on NTP and PTP time synchronization. In *International Symposium on Precision Clock Synchronization for Measurement, Control and Communication (ISPCS).*

[23] Sreejaya Viswanathan, Rui Tan, and David Yau. 2016. Exploiting Power Grid for Accurate and Secure Clock Synchronization in Industrial IoT. In *The 37th IEEE Real-Time Systems Symposium (RTSS).*

[24] Sreejaya Viswanathan, Rui Tan, and David KY Yau. 2018. Exploiting Electrical Grid for Accurate and Secure Clock Synchronization. *ACM Transactions on Sensor Networks (TOSN)* 14, 2 (2018), 12.

[25] Lei Wang, Javier Fernandez, Jon Burgett, Richard W Conners, and Yilu Liu. 2008. An evaluation of network time protocol for clock synchronization in wide area measurements. In *PES General Meeting*.