Global Journal of Engineering and Technology Advances
eISSN: 2582-5003
Cross Ref DOI: 10.30574/gjeta
Journal homepage: https://gjeta.com/

(RESEARCH ARTICLE)

Check for updates

Security Implications of Time Synchronization Errors in IoT Networks

Justin Njimgou Zeyeum *

Scripps College of Communication, J. Warren McClure School of Emerging Communication Technologies, Ohio University, Athens, Ohio, USA.

Global Journal of Engineering and Technology Advances, 2021, 09(02), 092-100

Publication history: Received on 24 September 2021; revised on 02 November 2021; accepted on 04 November 2021

Article DOI: https://doi.org/10.30574/gjeta.2021.9.2.0151

Abstract

The Internet of Things (IoT) connects billions of devices in healthcare, transportation, manufacturing, and smart cities. For data accuracy, energy efficiency, and low latency, it is very important to keep time in sync. Errors in synchronization slow down performance and make systems more vulnerable to attack. Adversaries can cause clock drift to start denial-of-service (DoS) attacks, use desynchronization to replay exploits, or change timestamps to break industrial control systems. This paper examines the impact of synchronization errors in Narrowband (NB) and Ultra-Narrowband (UNB) IoT systems, modeled through constant, random, and clock skew errors via MATLAB simulations. This study underscores synchronization as a critical IoT security vulnerability, in addition to its impact on technical performance degradation. The results show that when the Bit Error Rate (BER) and the Signal-to-Noise Ratio (SNR) go down, the risks in mission-critical IoT environments go up. The paper ends by suggesting safe ways to sync, such as authenticated protocols, consensus based on blockchain, and lightweight cryptographic timestamping.

Keywords: IoT; Time Synchronization; Security; Clock Error; Narrowband; Ultra-Narrowband; Cybersecurity

1. Introduction

The Internet of Things (IoT) is one of the most important technological ideas of the 21st century. IoT has changed healthcare, industrial automation, supply chain logistics, smart cities, and national defense (Al-Fuqaha et al., 2015). There are already more than 14 billion connected devices around the world, and that number is expected to grow to more than 25 billion by 2030. IoT is defined by the fact that it needs resource-limited devices to work together in different environments to provide services that need to be done quickly.

Accurate time synchronization is a basic requirement for the reliable operation of the Internet of Things (IoT). Timekeeping makes sure that devices stay in sync when sending and receiving messages, which keeps distributed systems in a consistent state. For a long time, engineering literature has seen synchronization as a performance issue that has to do with lowering the bit error rate (BER), packet loss, and energy overhead. People often forget about the cybersecurity point of view.

This mistake is dangerous. Synchronization is not just about getting things right; it is also an important security function. Malicious individuals can take advantage of synchronization errors to insert false information, delay emergency signals, interfere with industrial controllers, or carry out denial-of-service attacks (Narula & Humphreys, 2017; Yiğitler, Badihi, & Jäntti, 2020). GPS spoofing against ships and drones, false-data injections against smart grids, and adversarial jamming against IoT sensor networks are all examples of synchronization attacks (Alghamdi, 2020).

* Corresponding author: Justin Njimgou Zeyeum.

This paper is built on earlier research by Zeyeum (2019), which looked at synchronization errors in NB-BPSK and UNB-BPSK modulation systems, to look more closely at how these errors affect security. The study redefines synchronization as a performance impediment and a cybersecurity weakness by linking engineering simulations to adversarial interpretations.

2. IoT Networks and Security Considerations

IoT networks use many different communication systems, each with its own strengths and security challenges. These include short-range technologies like ZigBee and Bluetooth, low-power wide-area networks such as LoRaWAN and Sigfox, and cellular systems like NB-IoT and LTE-M (Ding, Nemati, Ranaweera, & Choi, 2020). Each one needs some form of time synchronization to work properly, but the way they handle it can also create risks.

- **Sigfox:** works asynchronously, meaning devices send data without strict time alignment. This saves energy and keeps things simple, but it also makes the network easier to attack. Without tight timekeeping, hackers can spoof messages, analyze traffic patterns, or flood the network with fake data to cause denial-of-service (DoS) problems. Because Sigfox has limited security and bandwidth, even small disruptions can take down large parts of the network.
- **LoRaWAN:** uses time slots and special "spreading factors" to manage data transmissions. If security keys or authentication aren't strong, attackers can take advantage of timing problems to replay old packets or forge messages. In other words, they can send fake copies of valid data when the devices are out of sync, tricking the system into accepting them as real.
- **ZigBee:** keeps local devices in sync pretty well, but in bigger networks with multiple hops, clocks tend to drift apart over time. When this happens, attackers can use desynchronization attacks to disrupt communication and coordination between devices. In systems like smart lighting or industrial sensors, even small timing errors can lead to system failures or unsafe conditions.
- **NB-IoT:** which is built on LTE cellular technology, has strong synchronization built in. However, it relies on GPS signals for timing, and that's a problem because GPS can be spoofed. By sending fake GPS signals, hackers could confuse base stations or shift the network's clock, causing large-scale outages or data corruption (Narula & Humphreys, 2017).
- **EC-GSM-IoT:** uses older cellular systems for synchronization, which helps extend coverage but also carries over old security flaws. Attackers could impersonate base stations or manipulate timing to intercept or delay messages.

Overall, synchronization is a two-sided issue. When it works, it makes networks more efficient and reliable. But when it's off or under attack, it becomes a major security weakness. In short, IoT designers need to treat synchronization as both a performance requirement and a potential cybersecurity risk, making sure timekeeping systems are both accurate and protected.

3. Time-Sensitivity and Security Implications

IoT deployments frequently function in mission-critical environments where timing discrepancies can yield significant repercussions:

- **Healthcare IoT:** Wearable monitors, infusion pumps, and implantable devices all depend on signals that are in sync. Drift or delay may turn off alarms, giving enemies the chance to plan deadly attacks.
- **Industrial IoT:** If clock skew causes command misalignment, it can cause problems for factories that use synchronized robotic arms or sensors, which can lead to sabotage by enemies.
- **Smart Grids:** For energy demand-response systems to work, phasor measurement units need to be very well synchronized. Desynchronization can make power flows unstable and cause failures to happen one after the other.

Figure 1 shows that synchronization drift makes the IoT attack surface bigger. Constant errors make it possible to spoof; random errors make denial-of-service attacks more likely; and clock skew makes replay and injection attacks more likely.

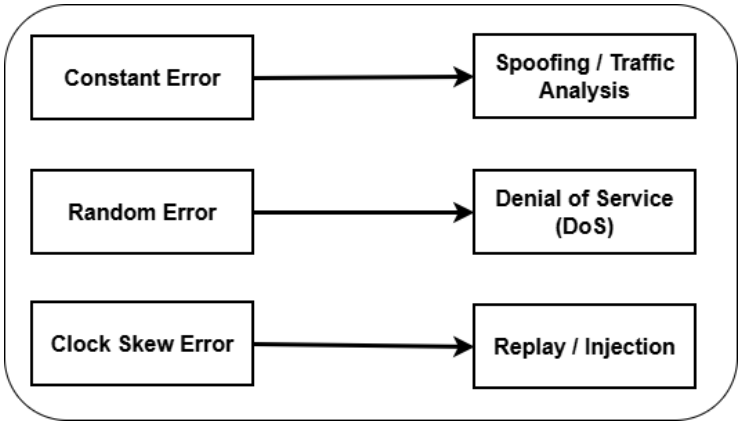

Figure 1 Attack surface of synchronization errors in IoT. Each error type expands adversarial opportunities for spoofing, denial-of-service, and replay/injection

4. Synchronization Solutions and Security Resilience

Different synchronization methods are used in IoT systems, and each one has its own trade-offs between performance, complexity, and security. Synchronization is important not only to make sure devices "speak the same time language," but also to ensure they can't be easily tricked or disrupted. If an IoT system's clock can be influenced by an attacker, it can open the door to data corruption, false readings, or even system shutdowns.

Closed-loop synchronization works by constantly monitoring and correcting the timing difference between devices. It's very effective at keeping clocks aligned because the system can quickly detect and fix drift in real time. From a technical standpoint, this means fewer transmission errors and better coordination between devices. However, the downside is that these systems depend on steady feedback signals between nodes, which makes them vulnerable to jamming. A hacker can block or flood the feedback channel with noise, causing devices to lose sync. In large IoT systems, like smart factories or autonomous vehicle networks, this could stop operations or create dangerous miscommunication between sensors and controllers (Narula & Humphreys, 2017).

One-bit feedback systems are simpler and more energy-efficient. They use very small packets, just one bit of information, to adjust timing. This makes them ideal for IoT networks with low-power sensors or battery-operated devices, such as those used in agriculture, smart homes, or environmental monitoring. The technical advantage is low overhead and high scalability. But security-wise, these systems can be easily tricked if the feedback signal is intercepted or changed by an attacker. Because the signal is so small, it's difficult to detect tampering. An adversary could manipulate timing updates to slowly desynchronize a network or make devices send incorrect data, leading to subtle and hard-to-detect disruptions (Yiğitler, Badihi, & Jäntti, 2020).

Two-way synchronization is more advanced and provides high accuracy by allowing devices to exchange time information in both directions. It's used in protocols like the Precision Time Protocol (PTP) or Network Time Protocol (NTP). The technical benefit is strong alignment and low clock drift, which makes it suitable for critical systems like smart grids or industrial automation. However, without added safeguards, it's still vulnerable to replay attacks, where an attacker records legitimate synchronization messages and resends them later to mislead the system. If cryptographic methods such as message authentication or digital signatures aren't used, the system may not be able to tell the difference between a real synchronization packet and a fake one (Fan & Ren, 2018).

To build secure and resilient synchronization systems, IoT designers must go beyond traditional timing techniques. Several approaches have been proposed:

- **Authenticated synchronization protocols** such as secure versions of NTP and PTP can verify that timing messages come from legitimate sources. This helps prevent spoofing and replay attacks (Alghamdi, 2021).
- **Blockchain-based consensus methods** distribute the trust across multiple nodes, making it almost impossible for one compromised device to alter the network's time. Each node validates time records through a shared ledger, ensuring integrity and tamper resistance (Schmid, 2018).

- **Lightweight timestamping schemes** use small cryptographic checks that fit within the power and bandwidth limits of IoT devices. These schemes offer good protection without the heavy processing costs of full encryption (Jadhav, 2020).

In practice, no single method solves every problem. Closed-loop systems are fast but can be jammed; one-bit systems are efficient but fragile; and two-way systems are accurate but need authentication. For IoT networks to be both reliable and secure, synchronization must be treated as part of the security architecture, not just as an engineering feature. Future designs will likely use a mix of these methods, combining accuracy, low energy use, and cryptographic protection to ensure that time itself cannot be used as a weapon against connected systems.

Figure 2 and Table 1 show the trade-off between security risk and the loss of technical performance. Synchronization increases throughput, but it also makes attacks more likely. Random errors make energy use go up, but they also let denial-of-service attacks happen. Constant errors stay hidden, making it easier to spoof someone. Clock skew makes timestamp-based authentication less secure, which allows replay and injection.

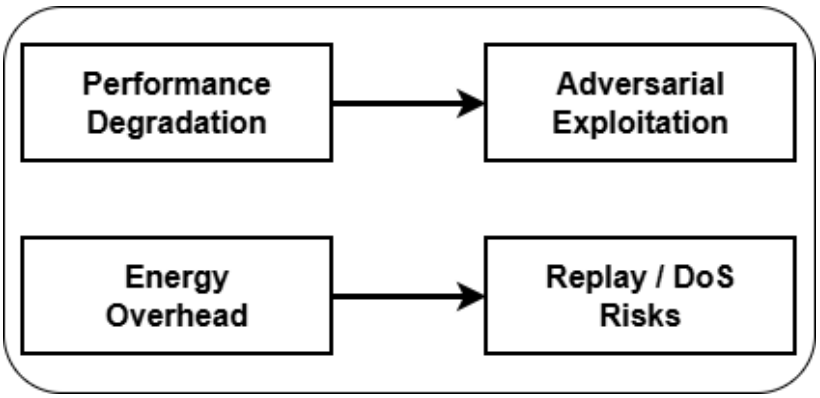

Figure 2 The trade-offs between technical and security in IoT synchronization**.**

Table 1 Synchronization Methods, Efficiency, and Security Resilience

Synchronization Method	Efficiency	Security Resilience
Closed-loop	Low drift, energy intensive	Vulnerable to jamming
One-bit feedback	Efficient, scalable	Susceptible to feedback manipulation
Two-way sync	Robust timing	Resilient but replay-prone without authentication

5. Error Models and Security Attack Vectors

Time synchronization in IoT networks is never perfect, there are always small differences between device clocks caused by temperature changes, signal delay, or processing speed. These timing differences, or synchronization errors, may seem minor but can have major effects on both performance and security. Understanding how these errors appear and how attackers might exploit them is key to designing more resilient IoT systems (Yiğitler, Badihi, & Jäntti, 2020).

Researchers typically describe three main types of synchronization errors in wireless and IoT networks: constant error, random error, and clock skew. Each type represents a unique way timing can drift, and each introduces its own technical and security problems.

5.1. Constant Error

A constant error occurs when all device clocks are consistently offset by a fixed amount, for example, every node being exactly 5 milliseconds behind the network clock. Technically, this leads to a stable but continuous degradation in performance, including reduced bit-error-rate (BER) performance and possible phase mismatches between signals. Because the offset remains predictable, attackers can use that predictability to launch spoofing or traffic-analysis

attacks. For instance, if an adversary knows when a sensor typically sends data, they can inject false packets at just the right moment or monitor timing patterns to infer sensitive operational details (Butun, Österberg, & Song, 2019).

5.2. Random Error

A random error happens when timing variations occur unpredictably caused by interference, fluctuating network load, or hardware noise. Technically, random errors increase variability in BER and latency, which reduces data reliability. From a security standpoint, randomness makes the network unstable and opens it to denial-of-service (DoS) attacks. Attackers can amplify small synchronization instabilities by sending extra or delayed messages, causing congestion and packet collisions that exhaust device resources. Over time, this can result in partial outages or complete service disruptions (Alghamdi, 2020).

5.3. Clock Skew Error

A clock skew error represents a gradual and cumulative drift between device clocks. Instead of being random or fixed, the difference between clocks increases steadily over time, leading to escalating BER misalignment. The danger of clock skew lies in its subtlety; networks often fail to notice it until communication becomes unreliable. Security-wise, clock skew is one of the most serious issues because it enables replay, injection, and impersonation attacks. An attacker can capture legitimate messages and resend them later when the network is out of sync, making the data appear authentic (Narula & Humphreys, 2017). In time-sensitive systems like industrial controllers or smart grids, such manipulation could trigger false readings or safety shutdowns.

These three models, constant, random, and skew, are not just performance concepts; they also define attack surfaces that adversaries can exploit. Together, they explain why timing problems must be viewed as part of IoT's broader security architecture, not merely as engineering noise to be corrected.

As summarized in Table 2, each error model corresponds to a specific family of security threats. Constant errors make spoofing easier; random errors create instability exploitable for DoS attacks; and clock skew produces gradual drifts that support replay or injection attacks. In real deployments, these error types often overlap, creating combined error conditions where multiple vulnerabilities reinforce each other.

Table 2 Mapping synchronization error models to security threats in IoT.

Error Model	Technical Impact	Security Risk	Example Attack Scenario
Constant Error	Predictable, fixed offset in timing	Spoofing / traffic analysis	Adversary predicts timing windows to inject false packets
Random Error	Unstable SNR and BER fluctuations	Denial-of-Service (DoS)	Flooding network during random desync peaks
Clock Skew Error	Gradual drift over time	Replay / injection Attacks	Re-use of time-shifted messages to mislead controllers
Combined Errors	Cumulative desynchronization	False-Data Injection / System Destabilization	Manipulation of timestamps across nodes

These relationships show that time synchronization is both a performance variable and a security vector. Future IoT protocols must monitor and correct all three types of errors dynamically, detecting when drift, noise, or offset patterns deviate from expected ranges to prevent timing from becoming an attacker's easiest weapon.

6. Simulation Studies and Annotated Results

MATLAB simulations created models of BER vs. SNR for NB-BPSK and UNB-BPSK with constant, random, and skew errors. Here are the original plots (Zeyeum, 2019) with new captions that stress the security implications.

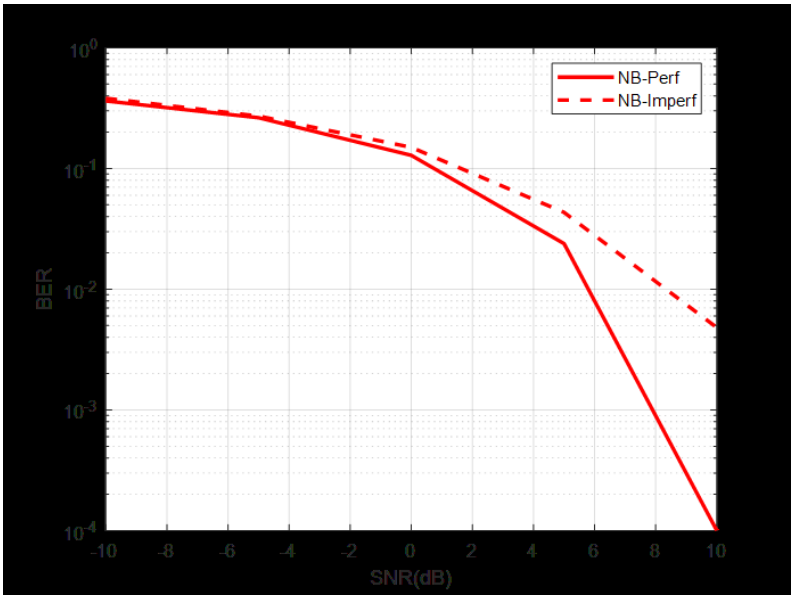

Figure 3 BER compared to SNR when the error is always the same (NB-BPSK). Technical: a predictable offset with a steady drop in BER. Security risk: can be used for spoofing and traffic analysis. Adapted from Zeyeum (2019)

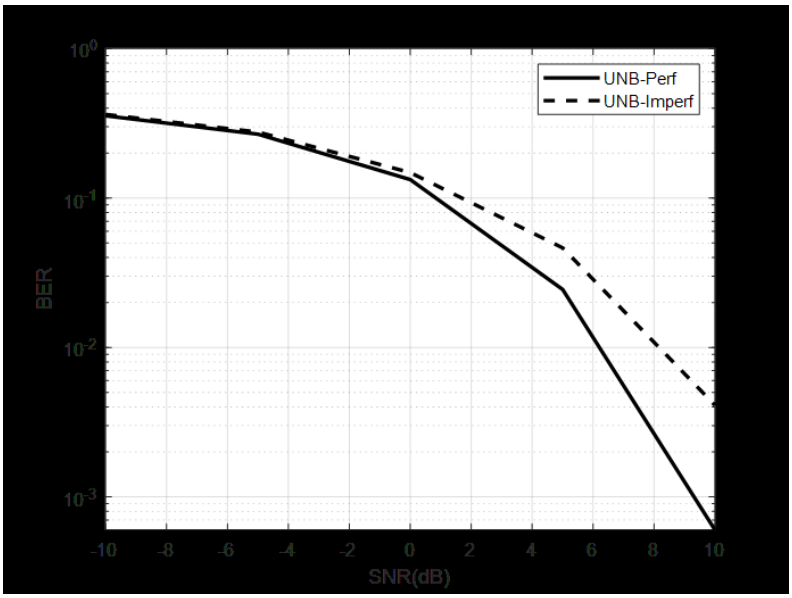

Figure 4 BER vs. SNR with a constant error (UNB-BPSK). Technical: an offset that can be predicted with a steady drop in BER. Security risk: can be used for spoofing and traffic analysis. Adapted from Zeyeum (2019)

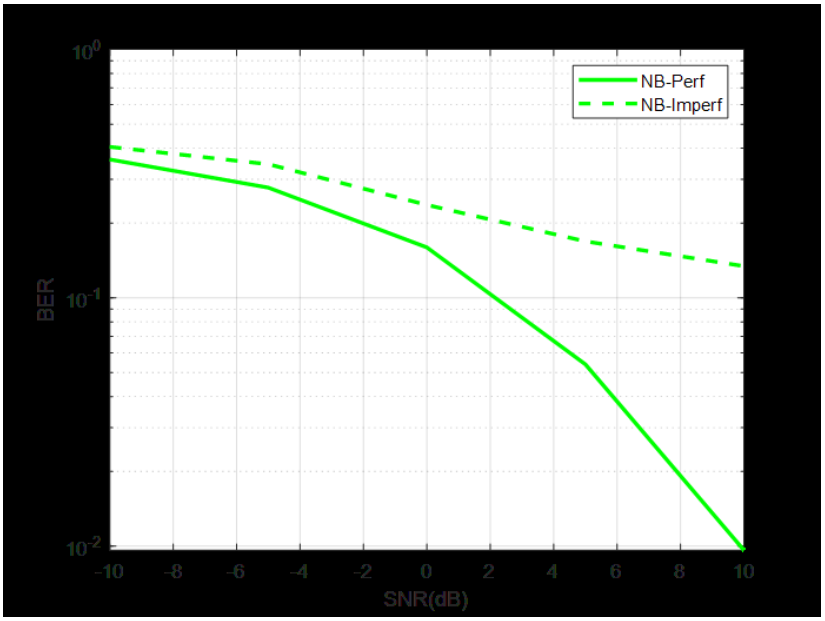

Figure 5 BER and SNR when there is a random error (NB-BPSK). Technical: random changes in the BER. Security risk: can be used to make denial-of-service attacks worse. Adapted from Zeyeum (2019)

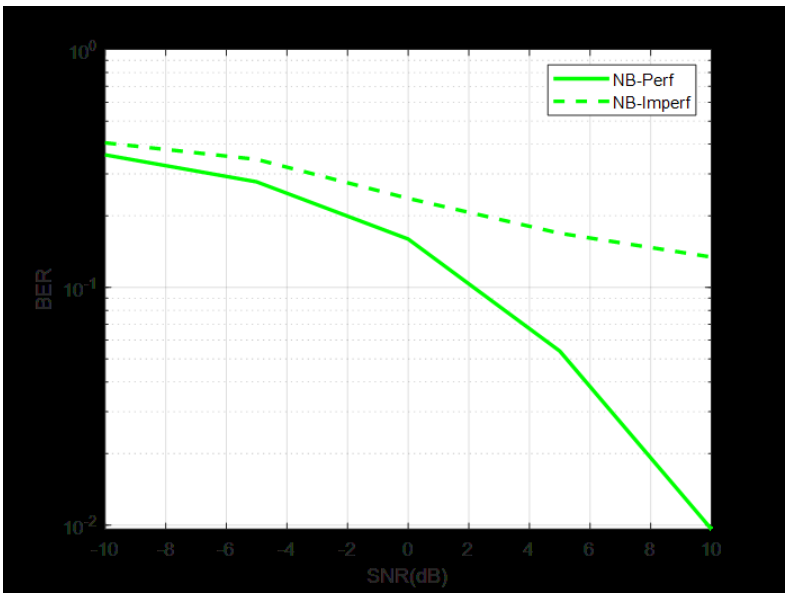

Figure 6 The BER and SNR when there are random errors (UNB-BPSK). Technical: random changes in the BER. Security risk: can be used to make denial-of-service attacks worse. Source: Based on Zeyeum (2019)

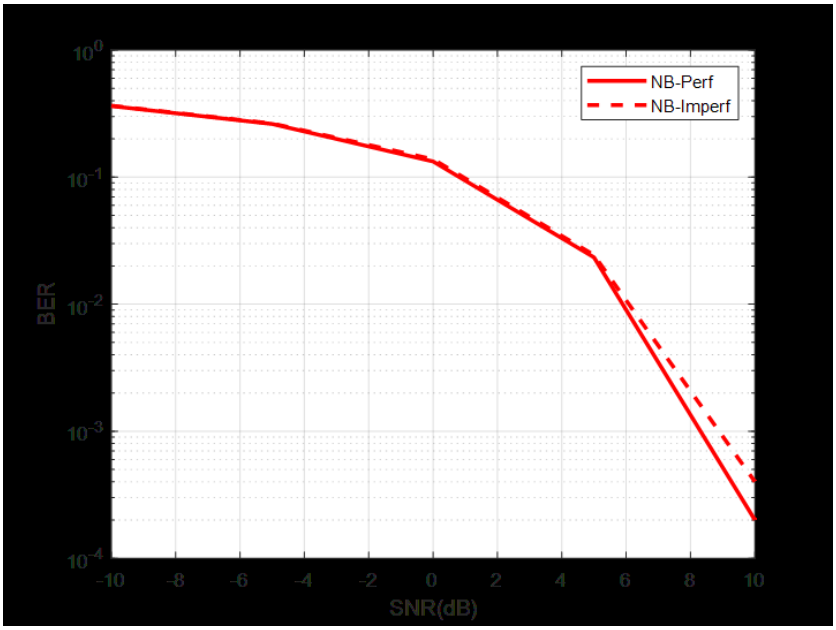

Figure 7 BER compared to SNR with skew error (NB-BPSK). Technical: BER is going up with drift. Security risk: can be used for replay and injection attacks. Adapted from Zeyeum (2019)

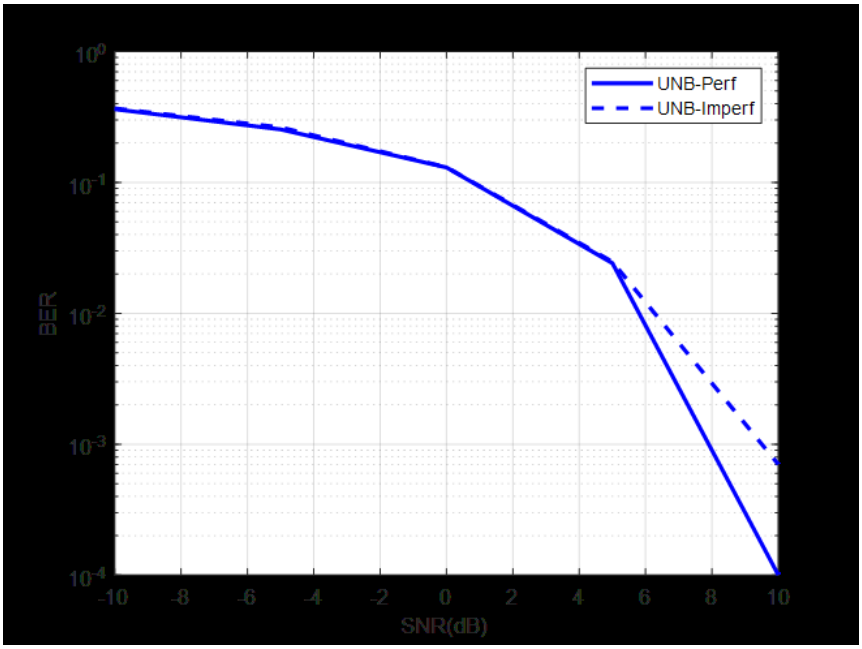

Figure 8 BER and SNR with skew error (UNB-BPSK). Technical: BER is going up with drift. Security risk: can be used for replay and injection attacks. Adapted from Zeyeum (2019)

The results show that constant errors can be predicted and found, but random and skew errors are more dangerous. Their unpredictability or escalation raises both technical degradation and chances for the enemy.

7. Conclusion and Future Work

This paper reframed synchronization as a cybersecurity imperative for IoT networks. By analyzing error models, conducting MATLAB simulations, and integrating adversarial perspectives, it demonstrated that synchronization is both a performance bottleneck and an exploitable vulnerability.

Future work should include:

- Authenticated synchronization protocols (secure NTP/PTP).
- Blockchain-based distributed clocks for tamper resistance.
- Lightweight timestamping schemes for constrained IoT devices.

As IoT expands into healthcare, industry, and energy systems, synchronization must be elevated to the same importance as confidentiality, integrity, and availability.

References

[1] Ding, J., Nemati, M., Ranaweera, C., & Choi, J. (2020). IoT connectivity technologies and applications: A survey. IEEE Access, 8, 67646–67673. https://doi.org/10.1109/ACCESS.2020.2985932

[2] Al-Fuqaha, A., Guizani, M., Mohammadi, M., Aledhari, M., & Ayyash, M. (2015). Internet of Things: A survey on enabling technologies, protocols, and applications. IEEE Communications Surveys & Tutorials, 17(4), 2347–2376.

[3] Alghamdi, W. (2020). Cyber-attacks on Precision Time Protocol networks. Electronics, 9(9), 1398. https://doi.org/10.3390/electronics9091398

[4] Alghamdi, W. (2021). Precision Time Protocol attack strategies and their resistance to existing security extensions. Cybersecurity, 4(1), 15. https://doi.org/10.1186/s42400-021-00080-y

[5] Butun, I., Österberg, P., & Song, H. (2019). Security of the Internet of Things: Vulnerabilities, attacks, and countermeasures. arXiv. https://doi.org/10.48550/arXiv.1910.13312

[6] Fan, K. (2019). Blockchain-based secure time protection scheme in IoT. Digital Commons, University of Memphis. https://digitalcommons.memphis.edu/facpubs/2585/

[7] Fan, K., & Ren, Y. (2018). Secure time synchronization scheme in IoT based on blockchain. In Proceedings of the 2018 IEEE International Conference on Internet of Things. IEEE. https://doi.org/10.1109/Cybermatics_2018.2018.00196

[8] Schmid, E.-G. (2018). Right to Sign: Safeguarding data immutability in blockchain systems with cryptographic signatures over a broad range of available consensus-finding scenarios (arXiv preprint arXiv:1811.05284). https://arxiv.org/abs/1811.05284

[9] Narula, L., & Humphreys, T. (2017). Requirements for secure clock synchronization. arXiv. https://doi.org/10.48550/arXiv.1710.05798

[10] Perry, Y. (2021). How vulnerable is NTP to malicious time servers? NDSS Symposium. https://www.ndss-symposium.org/wp-content/uploads/ndss2021_1A-2_24302_paper.pdf

[11] Jadhav, S. P. (2020). Towards light-weight cryptography schemes for IoT devices: Challenges and approaches. Journal of Mobile Multimedia, 15(1–2), 91-110. https://doi.org/10.13052/jmm1550-4646.15125

[12] Tripathi, N., & Hubballi, N. (2021). Preventing time synchronization in NTP broadcast mode. Computers & Security, 102, 102135. https://doi.org/10.1016/j.cose.2020.102135

[13] Red Hat Product Security. (2020, June 23). CVE-2020-11868: DoS on client ntpd using server mode packet [Security advisory]. Red Hat. https://access.redhat.com/security/cve/cve-2020-11868

[14] National Institute of Standards and Technology. (2020). Replay attack (Glossary entry). Computer Security Resource Center. https://doi.org/10.6028/NIST.SP.800-53r5

[15] Yiğitler, H., Badihi, B., & Jäntti, R. (2020). Overview of time synchronization for IoT deployments: Clock discipline algorithms and protocols. Sensors, 20(20), 5928. https://doi.org/10.3390/s20205928

[16] Zeyeum, J. N. (2019). Effects of time synchronization errors in IoT networks (Master's thesis, Tampere University https://urn.fi/URN:NBN:fi:tty-201905211687). https://doi.org/10.13140/RG.2.2.31305.66408.

[17] National Vulnerability Database. CVE-2018-8956, (2018). https://nvd.nist.gov/vuln/detail/CVE-2018-8956/.

[18] Beke, T., Dijk, E., Ozcelebi, T., & Verhoeven, R. (2020). Time synchronization in IoT mesh networks. In 2020 International Symposium on Networks, Computers and Communications (ISNCC 2020) (Article 9297296). Institute of Electrical and Electronics Engineers. https://doi.org/10.1109/ISNCC49221.2020.9297296