

Low Reliable and Low Latency Communications for Mission Critical Distributed Industrial Internet of Things

Dachao Yu, Wenyu Li, Hao Xu¹, and Lei Zhang², *Senior Member, IEEE*

Abstract—Achieving ubiquitous ultra-reliable low latency consensus in centralized wireless communication systems can be costly and hard to scale up. The consensus mechanism, which has been widely utilized in distributed systems, can provide fault tolerance to the critical consensus, even though the individual communication link reliability is relatively low. In this article, a widely used consensus mechanism, Raft, is introduced to the Industrial Internet of Things (IIoT) to achieve ultra-reliable and low latency consensus, where the consensus reliability performance in terms of nodes number and link transmission reliability is investigated. We propose a new concept, Reliability Gain, to show the linear relationship between consensus reliability and communication link transmission reliability. We also find that the time latency of consensus is contradictory to consensus reliability. These conclusions can provide guides to deploy Raft protocol in distributed IIoT systems.

Index Terms—Distributed industrial Internet of Things, consensus mechanism, raft, reliability, latency, fault tolerance, URLLC.

I. INTRODUCTION

DRIVEN by advances in the fifth generation (5G) mobile network, industry 4.0, cloud computing and artificial intelligence, etc., the Internet of Things (IoT) is extending from home and work environments to missing critical industrial sectors such as transportation, public infrastructure and utilities [1]. In Industrial IoT (IIoT) systems, data may be collected from distributed sensors located in different places to determine common and critical real-time decisions for achieving cooperative tasks among the connected components. A typical example is the connected autonomous driving [2], where a vehicle can make an initiative (e.g., acceleration/deceleration, merging into a lane, etc.) based on the sensors (e.g., Lidar, Radar, etc.) equipped locally. However, the initiative has to be consented by other vehicles in proximity through a safe and secure consensus protocol since any nonalignment among the vehicles may cause a disaster. In such a distributed IIoT system, communication plays a pivotal role in the information exchange among the connected components. Especially in mobile environments, where the connection among the nodes (e.g., cars, robots, or any other type of equipment) is wireless, the uncertainty of wireless channels and scarcity of communication resources can be critical factors to limit the IIoT performance in terms of decision reliability and latency.

Manuscript received August 12, 2020; accepted August 28, 2020. Date of publication September 2, 2020; date of current version January 8, 2021. The paper was supported in part by UK EPSRC project under Grant EP/S02476X/1. The associate editor coordinating the review of this letter and approving it for publication was D. Feng. (*Corresponding author: Lei Zhang.*)

The authors are with the James Watt School of Engineering, University of Glasgow, Glasgow G12 8QQ, U.K. (e-mail: d.yu.2@research.gla.ac.uk; h.xu.2@research.gla.ac.uk; 23574761@student.gla.ac.uk; lei.zhang@glasgow.ac.uk).

Digital Object Identifier 10.1109/LCOMM.2020.3021367

Ultra-reliable and low latency communications (URLLC) has been recognized as a key feature of 5G to meet the stringent requirements of industrial or personal applications [3]. According to [4], in some critical IIoT application scenarios, URLLC needs to provide an end-to-end latency lower than 1 ms and exceedingly high reliability more than $1 - 10^{-9}$. Centralized communication systems are normally deployed in industry sectors, which requires the connected IoT nodes to transmit their data to a central control station, where the critical decisions will be made and send back to actuators for processing. However, a large number of new generation mobile IIoT applications are discretely distributed in their topology, which means the scheme of centralized systems may hardly be implemented in these applications. Additionally, the centralized system suffers from an ever-present single point of failure issues. Moreover, in a centralized IIoT system, the IIoT nodes can only synchronize the information with the central station, which means the system's reliability performance heavily relies on the central station, and the performance can be limited by the worst node connection with the central station. Any wireless communication link failure can cut off the synchronization, which may cause disaster or loss of human life in extreme cases. Finally, the centralized communication system can be very costly since it is well-known that high communication reliability is contradictory to low time latency with given spectrum resources. The cost can be unaffordable when the network scales-up [5], e.g., on a busy road of autonomous driving scenarios or a smart factory with a large number of mobile robots. Therefore, from algorithms and protocols perspectives, an alternative low-cost solution should be investigated on how to improve the overall network's critical decision reliability and latency with low individual link transmission reliability.

Distributed systems can achieve such stringent requirements with relaxed communication link reliability by using a consensus mechanism (CM) to achieve the necessary agreement on a single state of the network. As one of the most recently successful applications, CM is a key element of blockchain networks to ensure the synchronization among the distributed nodes [6]. Unlike the traditional centralized communication system that requires all communication links are reliable under a time delay constraint to make correct decisions for IIoT, CMs in a distributed system can tolerate a certain ratio of link transmission failure, i.e., it can achieve a high reliability critical decision with relatively low reliable communication links. Raft [7] is such a typical crash tolerating CM to manage log duplicate. However, up to this point, CM (and corresponding applications such as blockchain) is primarily designed in stable wired communication environments. Unlike wired networks,

wireless communication channels are unstable, scarce, and prone to interference. In particular, original Raft considers that when the node failure happens, all associated communication links are faulty. However, with dynamic wireless communication channels, a node may work well, but some links connected with the node might be unstable. It is worth to derive the reliability in the presence of link failure to adopt the distributed IIoT in wireless environments. Moreover, it is unclear how such a distributed protocol affects the overall delay in the wireless environment. These concerns in the Raft consensus should be investigated to instruct the deployment of the CM in mission critical distributed IIoT applications.

For the first time, this article discusses how to use Raft to achieve highly reliable consensus for mission critical distributed IIoT, where the communication links can be low latency but low reliable. We first introduce a Raft CM link failure model to analyze the mathematical relationship between the communication link reliability and system decision reliability. Based on this derived relationship, the letter proposes an essential concept called *Reliability Gain*. It presents the mathematical relationship between consensus reliability and communication link reliability. We also find that the *Reliability Gain* is in a linear form of nodes number approximately. Additionally, the derivation reveals that the consensus reliability contradicts to delay, which provides design guidance for consensus to the distributed IIoT systems.

II. RAFT PROTOCOL IN IIoT

Section II introduces the concept of distributed system based on the Raft consensus. A Raft network is composed of a number of consensus nodes, as shown in Fig. 1, where the leader node needs to pack the commands in log entries and replicate these entries to all followers ceaselessly in every term through downlink communications; depending on the successful reception of the request, the followers confirm and send back the log to the leader by using the uplink communications. Consensus nodes can be actuators or work as a group to provide consensus to the actuator(s) in the IIoT. The actuators can only take actions if the critical decision is a consensus of the CM network. In the next, we assume the consensus nodes are actuators for simplicity. A successful Raft consensus represents that more than 50% of overall followers can receive the log entries and send confirmation back to the leader successfully in one term. In realistic cases, the ratio of followers with successful communication is flexible, which should fit in the requirement of scenarios. Thus, communication plays a key role in such a system and determines the consensus performance. The followers/actuators who cannot receive log entries or send back the confirmation because of communication link failure need to synchronize their state through other normal followers/actuators. Eventually, all actuators can get the correct log state to process these critical decisions made by consensus in the distributed system to accomplish complex manufacturing tasks. The leader can be selected by simple rotation or under the criteria of maximizing the system performance (e.g., select the node with the best communication connection with others), which is not the focus of this article.

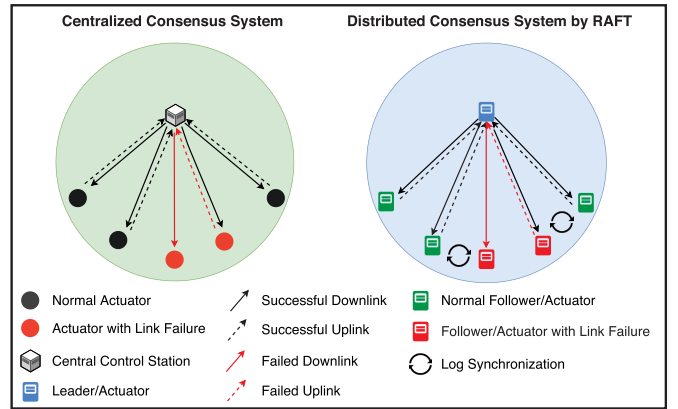


Fig. 1. Centralized consensus system and distributed consensus system by Raft.

As shown in Fig. 1, a centralized system is compared with a distributed system with Raft. In the centralized communication system, any communication link failure related to IIoT devices can cause the failure of critical decisions to the actuator. However, in the distributed communication system with Raft, the consensus can be made even though there are some unstable communication links, who cannot have reliable communications with the leader. However, the normal follower with completed logs may become a backup for synchronization to guarantee that all followers can get the consensus state.

Moreover, Raft protocol does not concern potential malicious nodes' effects on the distributed network [8]. The autonomous driving and other critical IIoT also fit in this case because the probability of the malicious users in this system is in low risks, or the nodes are under high-security level protection. Even if the malicious nodes in systems cannot be ignored, other similar consensus mechanisms like Practical Byzantine Fault Tolerance (PBFT) [9] can be adopted, and our following derivations can be extended accordingly.

III. RAFT RELIABILITY AND LATENCY ANALYSIS

In this Section, we first establish a wireless communication model with Raft protocol to analyze the consensus reliability performance. Then, we investigate the Raft properties in terms of consensus reliability and time latency.

A. Reliability of Communication System With Raft

Considering there are N nodes in a distributed system with Raft CM, theoretically, a reliable critical decision requires that over $\frac{N-1}{2}$ followers can receive log entries from the leader and send the confirmed messages back to the leader to achieve the commitment of log replication, which means the number of nodes with both successful downlink and uplink transmissions should be more than half nodes (i.e., $\frac{N-1}{2}$ followers and the leader) to accomplish the consensus progress. It is worth to mention that 50% is the fault tolerance of Raft [7], however, this value can be higher in an environment with unstable communication links. Nevertheless, this value will not affect our derivations.

We assume that the communication link success rate is P_l . Mathematically, the consensus success rate of the system

P_C is accumulated by the probability of every case in the successful consensus progress, which is in the form of two summations of probability in a binomial distribution. The accurate probability of success consensus P_C in the distributed IIoT communication system can be derived in the following equation

$$P_C = \sum_{i=\frac{N-1}{2}}^{N-1} \binom{i}{N-1} P_l^i (1-P_l)^{N-1-i} \times \sum_{j=\frac{N-1}{2}}^i \binom{j}{i} P_l^j (1-P_l)^{i-j}. \quad (1)$$

where the symbol $\binom{x}{y}$ denotes the combination of y choose x with $y \geq x$ and both x and y being non-negative integers. The first summation represents the probability that the majority of followers can download the log entry from the leader. The second summation equals to the probability that the majority of followers can upload their confirmation back to the leader. Because the downlink transmission happens before uplink transmission in Raft, the number of successful uplink transmission is never larger than the number of successful downlink transmissions. Therefore, the probability of a successful consensus term is the product of these two summations. It is worth to mention that consensus success rate P_C increases monotonically with the nodes number N . Though this property cannot be revealed by equation (1) straightforwardly, however, our following simplification in Section III-B and the simulation result in Section IV can show this property explicitly.

Remark 1: According to the equation (1), to satisfy the most stringent reliability requirement in IIoT, i.e., the consensus failure rate $1 - P_C$ is less than 10^{-9} , the nodes number N should not be less than 69, 31, 12, 5, when the link success rate P_l is 90%, 95%, 99% and 99.9%, respectively.

B. Reliability Gain

The remark in Section III-A indicates the fact that even if the link success rate P_l is undesirable, the consensus success rate of Raft can still be improved to the standard of IIoT, and the nodes number N can influence this reliability improvement. Therefore, we introduce a parameter called *Reliability Gain* (also can be interpreted as reliability amplification factor) to represent the quantitative relationship between the reliabilities of consensus and communication link.

Theorem 1: When the link success rate P_l is reasonably large¹, it has a linear relationship with consensus failure rate $1 - P_C$ in logarithm

$$\log(1 - P_C) = k \cdot \log(1 - P_l) + h, \quad (2)$$

where the *Reliability Gain* $k = \frac{N+1}{2}$ and the intercept $h = \log\left(\binom{\frac{N-3}{2}}{N-1}\right) + \Delta h$, with Δh being given in Table I.

Proof: See Appendix A ■

From the definition of *Reliability Gain* k , we can find that consensus failure rate $\log(1 - P_C)$ and link failure rate

¹Our results show that $P_l = 90\%$ is large enough to make the conclusion, while this communication link reliability can be achieved in most of the communication environments.

TABLE I
ESTIMATED Δh

N	Δh	N	Δh	N	Δh
5	0.472	6	0.704	7	0.797
8	1.081	9	1.214	10	1.509
11	1.668	12	1.965	13	2.144
14	2.441	15	2.635	16	2.931
17	3.136	18	3.431	19	3.645

$\log(1 - P_l)$ are in linear relation when the nodes number N is constant. With fixed link reliability, the increasing nodes number rises up the consensus reliability, which proves the increasing monotonicity of consensus success rate P_C with the nodes number N . Compared to the equation (1), this equation shows a simple relationship between link reliability and consensus reliability. Thus, it can provide a valid guide for the real Raft CM deployment in the IIoT systems. Table I shows the estimated Δh when nodes number N increases from 5 to 19, where Δh remains constant with a fixed nodes number N . The simulation result in Section IV shows that the consensus failure rate $\log(1 - P_C)$ satisfies the linear relationship in equation (2) when P_l is as low as 90%.

C. Relationship Between Latency and Reliability

In this Subsection, we will show that the consensus reliability and the consensus latency are contradictory. A wireless communication model, which aims to analyze the packet error probability of the wireless short package transmissions in URLLC [10], is used to find out the relationship between consensus success rate P_C and the consensus latency T , which we assume it is caused by downlink and uplink transmission delay, i.e., Raft consensus latency T only composes of communication transmission delay to show the communication impacts on the overall consensus latency. This model is an illustrational case and other models can be used without affecting the main conclusion of the letter. According to [10], the link failure rate $1 - P_l$ used in equation (1) and (2) can be written as a function of T as follow

$$1 - P_l = f_Q\left(\frac{B \frac{T}{2N} (C - R) + \frac{\log_2(B \frac{T}{2N})}{2}}{(B \frac{T}{2N})^{\frac{1}{2}} \log_2(e)}\right), \quad (3)$$

where B is the available spectrum bandwidth. R and C are the uplink or downlink transmission rate and channel capacity, respectively. Note that here we assume both uplink and downlink transmissions are time divisioned, i.e., given the overall consensus delay, T , each transmission can have $t = \frac{T}{2N}$ transmission internal since there are N transmissions in both uplink and downlink. Therefore, with a constant N , the increasing consensus delay T can provide more time t for each link transmission, which intuitively can reduce the link failure rate $1 - P_l$. By substituting equation (3) into equation (1) or (2), we can obtain the relationship of reliability $1 - P_C$ with the latency T . The contradiction of consensus reliability $1 - P_C$ and time delay T can be proved in mathematics by calculating the derivative of the variable $Q = \frac{B \frac{T}{2N} (C - R) + \frac{\log_2(B \frac{T}{2N})}{2}}{(B \frac{T}{2N})^{\frac{1}{2}} \log_2(e)}$ in

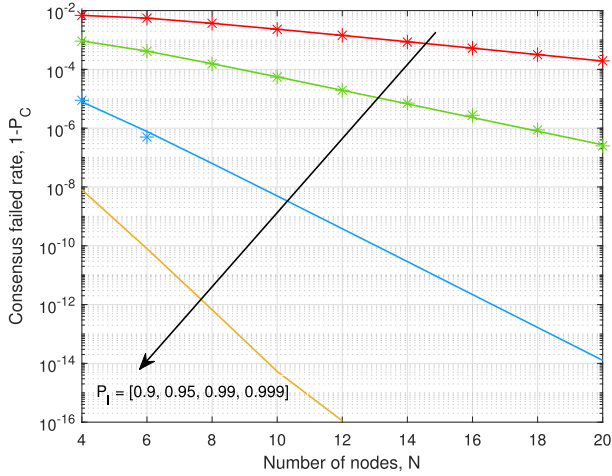


Fig. 2. Consensus failure rate $1 - P_C$ vs. Nodes number N [Lines: analytical result from equation (1), Asterisks: simulated].

Q-function

$$\frac{\partial Q}{\partial T} = \frac{\frac{B}{2N}(C - R) - \frac{1}{2T} \log_2\left(\frac{T}{2N}B\right) + \frac{1}{T}(\ln 2)}{2\sqrt{\frac{T}{2N}B \log_2(e)}}. \quad (4)$$

The derivative $\frac{\partial Q}{\partial T}$ in equation always keeps positive, which means the variable Q increases monotonically along with T . Based on the decreasing monotonicity of Q-function $f_Q(*)$ along with Q and the increasing monotonicity of P_C along with P_l , the time delay of consensus T and consensus reliability $1 - P_C$ are contradictory.

According to the conclusion in Section III-A, the consensus reliability $1 - P_C$ increases monotonically with the nodes number. However, given fixed consensus delay T , increasing node number will also result in a shorter transmission time $t = \frac{T}{2N}$ for each link, thus causes a smaller P_l , which may turn out a less reliable consensus according to equation (1) or (2). Thus, it is expected that there is an optimal N to achieve maximum consensus reliability.

IV. SIMULATION RESULTS AND DISCUSSION

Simulations are conducted to validate the proposed consensus communication model and its derivations. The given bandwidth for link transmission B is set as 18 kHz, and the SINR (signal-to-interference-plus-noise ratio) is set to 10 dB. The uplink and downlink capacity R is assumed 50% of the channel capacity, which is calculated by $C = \log(1 + \text{SINR})$.

Fig. 2 indicates the consensus success rate in the model with an increasing number of nodes N in the Raft link failure model. The consensus failure rate $1 - P_C$ declines as nodes number increases with relatively low communication link success rate $P_l = 90\%, 95\%, 99\%, 99.9\%$, respectively. The simulated results (in asterisks) of the consensus failure rate $1 - P_C$ is overlapped to their analytical curves (in lines) when the link success rate $P_l = 90\%$ and 95% , which proves the correctness of the equation (1). The analytical curves shows the property that the consensus success rate P_C increases monotonically with the nodes number N . Because the consensus failure rate is extremely low for a larger P_l and

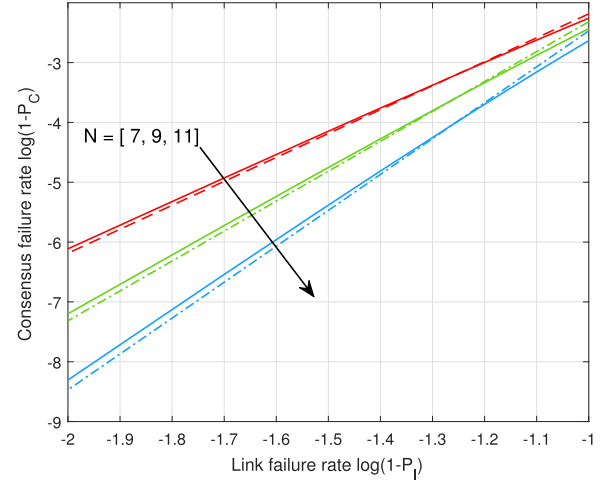


Fig. 3. Consensus failure rate $\log(1 - P_C)$ vs. Link failure rate $\log(1 - P_l)$ [Solid lines: analytical results from equation (1), Broken lines: simplified analytical results from equation (2)].

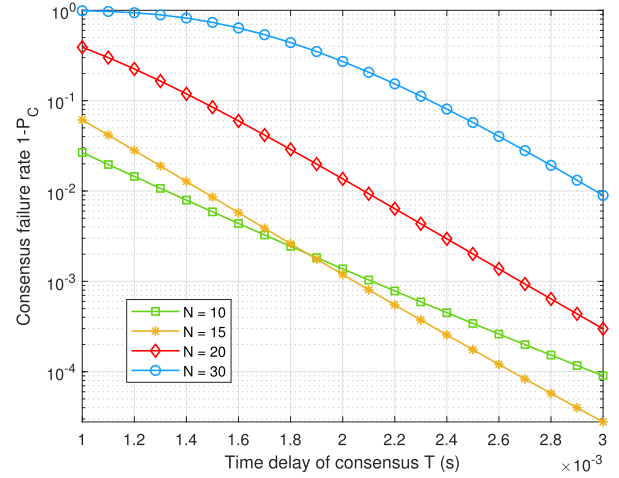


Fig. 4. Consensus failure rate $(1 - P_C)$ vs. Consensus delay T .

MATLAB compute power is limited, the simulated result of consensus failure rate $1 - P_C$ cannot be completely presented in the Fig.1 when P_l is 99% and 99.9%.

Fig. 3 shows the consensus reliability tendency along with the link success rate P_l . The analytical result represents the original consensus reliability $1 - P_C$ in logarithm in equation (1). The simplified result represents the consensus failure rate $\log(1 - P_C)$ in equation (2). Analytical lines and simplified lines are highly matched, which support the accuracy of the linear relation in equation (2). The slopes of lines are equivalent to the value of *Reliability Gain* $k = \frac{N+1}{2}$, which become steeper when the nodes number N rises up. The result shown here suggests that we can use a simplified model to guide the real deployment of Raft for distributed systems.

The simulation in Fig. 4 reveals the contradiction between consensus reliability $1 - P_C$ and consensus delay T . Four curves are corresponding to different nodes number $N = 10, 15, 20, 30$, respectively. All curves in Fig. 4 show that with the constant nodes number, the consensus reliability $1 - P_C$ reduces when the time delay T rises up, which proves the contradiction of the consensus reliability and time latency.

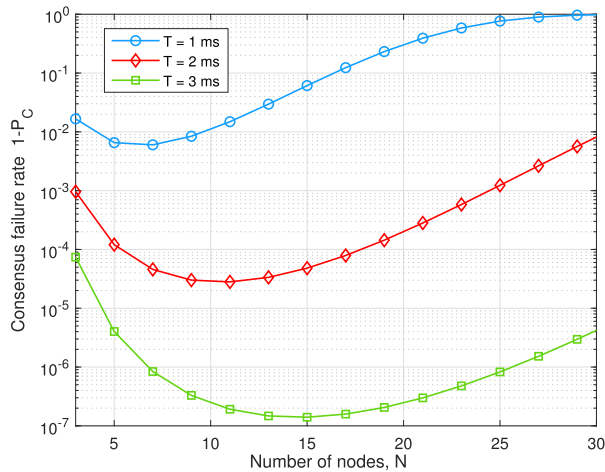


Fig. 5. Consensus failure rate ($1 - P_C$) vs. Nodes number (N).

The tendency of consensus failure rate at $N = 15$ drops more dramatically than the consensus failure rate at $N = 10$, along with the increase of time delay, which causes the interception of two curves. It implies that the consensus reliability does not have monotonicity along with the nodes number N if consensus delay T 's effect on the link transmission reliability is considered. Therefore, further investigation of this phenomenon is performed.

Fig. 5 indicates the change in the consensus reliability $1 - P_C$ along with nodes numbers N and a constant consensus time latency T . The curves show that the consensus reliability $1 - P_C$ fluctuates when nodes number increases, and there is maximum consensus reliability. By modifying the time delay in the consensus system, the maximum value of the reliability curve will be shifted to a higher value with the larger corresponding nodes number. The reason of this phenomenon is that the consensus failure rate follows the monotonicity in equation (1) when N is small; when N becomes large, given the fact that communication resource (i.e., the communication time T) is limited, the time latency in each link transmission will be reduced, and $1 - P_l$ will increase dramatically along N based on the property of Q function in equation (3), which causes the rise of $1 - P_C$. Therefore, the nodes number N has both positive and negative effects on consensus reliability. The shifting of curves indicates that the optimization of consensus reliability by allocating communication resources can be implemented to reach the requirements of different scenarios in IIoT.

V. CONCLUSION

The analysis of consensus reliability in the distributed IIoT system with Raft concludes that with a low communication link reliability, the consensus reliability in Raft can achieve ultra reliability by increasing the nodes number. The relationship of consensus reliability with communication link reliability is interpreted in a linear form for simplicity. Meanwhile, the results show that the time latency is contradictory to the consensus reliability in Raft. Therefore, this article provides a valuable guide for the design and deployment of Raft in distributed systems.

APPENDIX A

PROOF OF EQUATION (2)

In the summations of the binomial distributions of the consensus success rate P_C , the largest term dominates the summation if the link success rate P_l increases to reasonably large. Thus, the inner summation of the binomial distributions can be replaced by the largest term of it for simplification

$$\sum_{j=\frac{N-1}{2}}^i \binom{j}{i} P_l^j (1 - P_l)^{i-j} \approx \binom{i}{i} P_l^i (1 - P_l)^{i-i} = P_l^i. \quad (5)$$

The result of equation (5) can be substituted into the equation (1). According to the cumulative distribution function of binomial distribution, the consensus failure rate $1 - P_C$ is

$$1 - P_C = \sum_{i=0}^{\frac{N-3}{2}} \binom{i}{N-1} P_l^{2i} (1 - P_l)^{N-1-i} \approx \binom{\frac{N-3}{2}}{N-1} (1 - P_l)^{\frac{N+1}{2}}. \quad (6)$$

The largest term in the summation of the consensus failure rate in equation (6) is also dominating. Since P_l is reasonably large, the summation in equation (6) can be simplified in the same way as equation (5). And when the consensus failure rate $1 - P_C$ is converted to logarithm form, it will correspond to the linear relation in equation (2)

$$\log(1 - P_C) = \left(\frac{N+1}{2}\right) \log(1 - P_l) + \log\left(\binom{\frac{N-3}{2}}{N-1}\right) + \Delta h, \quad (7)$$

where Δh is the corrected value of the intercept in equation (2) to get the minimum error between equation (1) and equation (2).

REFERENCES

- [1] S. Mumtaz, A. Alshaily, Z. Pang, A. Rayes, K. F. Tsang, and J. Rodriguez, "Massive Internet of Things for industrial applications: Addressing wireless IIoT connectivity challenges and ecosystem fragmentation," *IEEE Ind. Electron. Mag.*, vol. 11, no. 1, pp. 28–33, Mar. 2017.
- [2] L. Hobert, A. Festag, I. Llatser, L. Altomare, F. Visintainer, and A. Kovacs, "Enhancements of V2X communication in support of cooperative autonomous driving," *IEEE Commun. Mag.*, vol. 53, no. 12, pp. 64–70, Dec. 2015.
- [3] G. Brown, "Ultra-reliable low-latency 5G for industrial automation," in *Heavy Reading White Paper for Qualcomm*, 2018.
- [4] Y. Rao *et al.*, "New services & applications with 5G ultra-reliable low latency communication," 5G Americas, Bellevue, WA, USA, Tech. Rep., 2018.
- [5] M. Bennis, M. Debbah, and H. V. Poor, "Ultrareliable and low-latency wireless communication: Tail, risk, and scale," *Proc. IEEE*, vol. 106, no. 10, pp. 1834–1853, Oct. 2018.
- [6] Y. Sun, L. Zhang, G. Feng, B. Yang, B. Cao, and M. A. Imran, "Blockchain-enabled wireless Internet of Things: Performance analysis and optimal communication node deployment," *IEEE Internet Things J.*, vol. 6, no. 3, pp. 5791–5802, Jun. 2019.
- [7] D. Ongaro and J. Ousterhout, "In search of an understandable consensus algorithm," in *Proc. USENIX ATC*, Jul. 2014, pp. 305–319.
- [8] H. Xu, L. Zhang, Y. Liu, and B. Cao, "RAFT based wireless blockchain networks in the presence of malicious jamming," *IEEE Wireless Commun. Lett.*, vol. 9, no. 6, pp. 817–821, Jun. 2020.
- [9] M. Castro and B. Liskov, "Practical Byzantine fault tolerance," in *Proc. 3rd Symp. Operating Syst. Design Implement.*, 1999, pp. 173–186.
- [10] B. Chang, L. Zhang, L. Li, G. Zhao, and Z. Chen, "Optimizing resource allocation in URLLC for real-time wireless control systems," *IEEE Trans. Veh. Technol.*, vol. 68, no. 9, pp. 8916–8927, Sep. 2019.