Journal of Theoretical and Applied Information Technology
30th June 2023. Vol.101. No 12
© 2023 Little Lion Scientific

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

NEW SERVICES AND APPLICATIONS CAN LEVERAGE THE POWER OF LOW RELIABLE AND LATENCY COMMUNICATION FOR MISSION CRITICAL DISTRIBUTED INDUSTRIAL INTERNET OF THINGS

VUTUKURI SARVANI DUTI REKHA[2], SWARUPA RANI BONDALAPATI[2], RATNA KUMARI VEMURI[3], RAMARAO GUDE[4], PRAVEEN TUMULURU[5] Dr. SURYA PRASADA RAO[6] BORRA[6]

[1,3]Assistant Professor, Department of Electronics and Communication Engineering,

Prasad V. Potluri Siddhartha Institute of Technology, Vijayawada, Andhra Pradesh, India.

[2]Assistant Professor, Department of Electrical and Electronics Engineering,

Velagapudi Ramakrishna Siddhartha Engineering College, Andhra Pradesh, Vijayawada, India.

[4]Associate Professor, Department of Electronics and Communication Engineering,

G. Pullaiah College of Engineering and Technology, Kurnool, Andhra Pradesh, India.

[5]Department of Computer Science Engineering,

Koneru Lakshmaiah Education Foundation, Vaddeswaram, Andhra Pradesh, India

[6]Associate Professor, Department of Electronics and Communication Engineering,

Prasad V. Potluri Siddhartha Institute of Technology, Vijayawada, Andhra Pradesh, India.

E-mail: vsdrekha@gmail.com , swarupabondalapti@gmail.com , vemuriratna2005@gmail.com, ramaraog19@gmail.com, praveenluru@gmail.com, suryaborra1679@gmail.com

ABSTRACT

It can be expensive and difficult to build up centralised wireless communication systems that achieve ubiquitous, ultra-reliable, low-latency consensus. Consensus mechanisms have been used extensively in distributed systems, and they can provide fault tolerance for the critical consensus even when the reliability of the individual communication links is low. This paper introduces Raft, a popular consensus mechanism, to the Industrial Internet of Things with the goal of achieving an ultra-reliable and low-latency consensus, and it examines the consensus reliability performance in terms of node number and link transmission reliability. By introducing the notion of reliability gain, we demonstrate the linear relationship between consensus reliability and the reliability of the transmission of information through the communication links. Also, we discover that the time latency of consensus undermines its validity. The findings can be used as guidelines for implementing the Raft protocol in decentralized IIoT environments.

Keywords: *Distributed industrial Internet of Things, consensus mechanism, raft, reliability, latency, fault tolerance.*

1. INTRODUCTION

The Internet of Things (IoT) is rapidly expanding beyond the domestic and commercial spheres into vitally important sectors like transportation, public infrastructure, and utilities, thanks to developments in the fifth generation (5G) mobile network, industry 4.0, cloud computing, artificial intelligence, etc. [1]. Data from dispersed sensors in various locations may be collected in Industrial IoT (IIoT) systems in order to make standard and crucial real-time decisions in order to accomplish cooperative tasks with the interconnected components. One common type is "connected autonomous driving" [2], in which a car makes decisions on its own (such as whether to speed up or slow down, change lanes, etc.) using data from on-board sensors. Any discord among nearby vehicles, however, could have disastrous consequences, so the initiative must be approved by all of them using a fool proof consensus protocol. Communication plays a crucial role in the information exchange between the connected components of a distributed IIoT system. Especially in environments where the nodes (e.g., cars, robots, or any other type of equipment) are connected wirelessly, the unpredictability of wireless channels and the scarcity of communication resources can be critical factors limiting the performance of the IIoT in terms of decision reliability and latency.

The ability of 5G to provide ultra-reliable and low-latency communications (URLLC) is seen as crucial for meeting the demanding needs of commercial or individual use cases [3]. Some of the most important use cases for IIoT require URLLC to have a latency of less than 1 millisecond and a reliability of greater than 1109, as stated in [4]. Many industrial sectors use centralized communication systems in which data from IoT nodes is sent to a control hub, where crucial decisions are made, and then sent back to actuators for implementation. However, many of the latest generation of mobile IIoT applications have a distributed topology, making it difficult to implement the scheme of centralized systems. The centralized system has the additional problem of a constant single point of failure. In addition, the IIoT nodes can only synchronize the information with the central station in a centralized IIoT system, which means the system's reliability performance is highly dependent on the central station and can be limited by the worst node connection with the central station. Disruption to the synchronization can occur if any wireless communication link fails, which could lead to catastrophic results or even the loss of human life. Finally, the centralized communication system can be expensive because it is well understood that high communication reliability is incompatible with low time latency with limited spectrum resources. When the network grows in size [5], as it would on a busy street with autonomous driving scenarios or in a smart factory with a large number of mobile robots, the price can become prohibitive. Since the individual links in a network often have low reliability, it is important to look into algorithmic and protocol-level ways to improve the overall network's reliability and latency in the face of such low transmission reliability.

Using a consensus mechanism (CM) to reach the required agreement on a single state of the network, distributed systems can meet such stringent requirements with relaxed communication link reliability. In order to maintain consistency across multiple nodes in a distributed network, CM has emerged as one of the most promising new applications [6]. In contrast to the conventional centralised communication system for the IIoT, which necessitates all communication links to be reliable under a time delay constraint to make correct decisions, CMs in a distributed system can tolerate a certain ratio of link transmission failure, i.e., it can achieve high reliability for critical decisions with relatively low-reliability communication links. A popular crash-tolerant CM for handling log duplication is Raft [7]. CM (and related applications such as block chain) was, however, primarily developed for dependable wired communication environments until recently. Wireless networks, in contrast to their wired counterparts, are unreliable, scarce, and susceptible to interference. It is important to note that the original Raft assumes that all connected communication links will be broken at the time of a node failure. Nonetheless, a node may function well, but some linked nodes may be unstable when using dynamic wireless communication channels. For the distributed IIoT to be successful in wireless settings, it is important to derive reliability even in the face of link failure. In addition, it is not known how the overall delay in a wireless environment is affected by a distributed protocol of this type. These Raft consensus issues must be studied before the CM can be used in mission-critical distributed IIoT applications.

In this article, we'll look at how to use Raft to reach consensus over low-latency, low-reliability communication links for mission-critical distributed IIoT applications. To begin exploring the mathematical connection between communication link reliability and system decision reliability, we present the Raft CM link failure model. The letter proposes a crucial concept called reliability gain based on this inferred relationship. It connects the two concepts of consensus reliability and communication link reliability mathematically. Additionally, we find that the reliability gain follows a linear pattern, with nodes roughly corresponding to integers. The proof-of-concept also shows that the reliability of the consensus is at odds with the delay, which can be used as a reference when designing distributed IIoT systems.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

2. THE RAFT PROTOCOL USED IN THE IIOT

In Section 2, we are presented with the idea of a raft-based distributed system. As shown in Fig. 1, a Raft network is made up of a group of consensus nodes; the leader node is responsible for encoding commands into log entries and sending them to all followers continuously over the course of each term via downlink communications; if the request is successfully received, the followers confirm and send the log back to the leader via uplink communications. Actuators in the IIoT can also serve as consensus nodes or collaborate with one another to reach a consensus on behalf of the actuators. If the CM network agrees on the crucial decision, then the actuators will take action. In what follows, for the sake of clarity, we will assume that the consensus nodes are actuators. More than half of all followers can receive the log entries and send confirmation back to the leader successfully in one term, which is what constitutes a successful Raft consensus. The number of followers necessary for effective communication can vary in real life, depending on the circumstances. Therefore, the success or failure of such a system depends heavily on the quality of its communication. In the event of a broken communication link, any followers or actuators that are unable to receive log entries or send back confirmations will need to bring their states into sync with those of normally functioning followers or actuators. Sooner or later, all actuators will have access to the right log state to process these crucial decisions made by consensus in the distributed system. Although not the focus of this article, choosing a leader by simple rotation or according to criteria designed to maximise system performance (for example, choosing the node with the best communication connection with others) are both viable options.

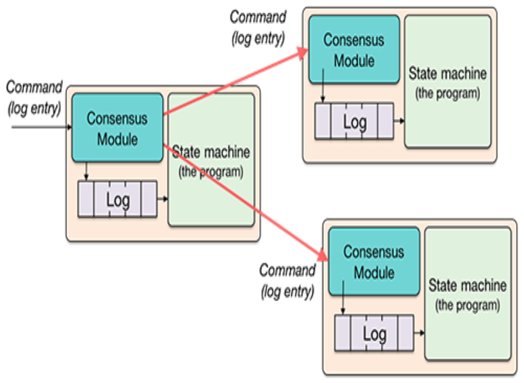

Figure 1: The Raft consensus system

In Figure 1, the advantages of a decentralized system like Raft are laid out in contrast to a centralized one. When using a centralized communication system, if an IIoT-related communication link fails, it could prevent the actuator from receiving vital decisions. While some communication links may be unstable and unable to contact the leader, a consensus can still be reached in a raft-based distributed communication system. To ensure that all followers can reach the consensus state, the typical follower with complete logs may serve as a backup for synchronization.

The Raft protocol also cares little about how the presence of malicious nodes might influence the overall network [8]. Autonomous vehicles and other mission-critical IIoT can benefit from this setup because the likelihood of malicious users is low and the nodes are protected to a high security level. Practical Byzantine Fault Tolerance (PBFT) [9] and other similar consensus mechanisms can be used if malicious nodes in a system can't be ignored. Our later derivations can be changed to account for this.

3. ANALYSIS OF THE RELIABILITY AND LATENCY OF THE RAFT

To evaluate the effectiveness of the consensus reliability mechanism, we first set up a wireless communication model using the Raft protocol. Consensus reliability and time latency are then explored as they pertain to the raft properties.

3.1 Raft Communication Security

Raft CM's commitment of log replication necessitates, in theory, that more than half of the nodes (i.e., (N-1)/2 followers and the leader) successfully downlink log entries from the leader and uplink confirmed messages to the leader in order to reach consensus on a critical decision. Raft's fault tolerance [7] is stated to be 50%, but in a network with unreliable connections, this value may be higher. However, this value will not have any bearing on our calculations.

The proportion of successful communication links is assumed to be P. In mathematics, the system's PC's consensus success rate is calculated by adding the probabilities of each case in the consensus progress, where each case is represented by two summations of probabilities in a binomial distribution. The following equation can be used to accurately calculate the success rate of the consensus PC in a distributed IIoT communication system.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

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

where (x, y) represents the set of all possible combinations where y is greater than or equal to x and x is positive, and y is the integer chosen by y. The first tally shows how likely it is that most followers will be able to get the log entry from the leader. The second total equals the likelihood that most followers can transmit confirmation to the leader via upload. Because downlink transmission occurs first, the number of successful uplink transmissions in Raft is never greater than the number of successful downlink transmissions. In this way, the likelihood of a successful consensus term can be calculated by multiplying the two sums together. The success rate of reaching a consensus, denoted by PC, steadily rises as N increases. This property is not immediately apparent from equation (1) but is made clear by our simplification in Section III-B and the simulation result in Section 4

As shown in equation (1), when the link success rate Pl is 91%, 95%, 99.5%, and 99.8%, the minimum number of nodes N that can be used to guarantee a failure rate of 1-PC of less than 109 is 69, 31, 12, and 5, respectively.

3.2 Enhanced Overall reliability

The section III-A remark shows that the consensus success rate of Raft can be improved to IIoT standards even if the link success rate Pl is low, and that the nodes N have some say in this. To quantify the connection between the reliabilities of consensus and communication link, we introduce the reliability gain parameter (also called a reliability amplification factor).

When the link success rate Pl is high enough, the consensus failure rate 1-PC is proportional to its logarithm.

$$\log(1 - P_C) = k.\log(1 - P_l) + h \qquad (2)$$

$$\sum_{j=\frac{N-1}{2}}^{i} \binom{j}{i} P_l^j (1-P_l)^{i-j} \approx \binom{i}{j} P_l^i (1-P_l)^{i-I} = P_l^i \quad (3)$$

$$1-P_C = \sum_{i=0}^{\frac{N-3}{2}} \binom{i}{N-1} P_l^{2i} (1-P_l)^{N-1-i}$$
$$\approx \binom{\frac{N-3}{2}}{N-1} (1 - P_l)^{\frac{N+1}{2}} \qquad (4)$$

$$\log(1 - P_C) = \left(\frac{N+1}{2}\right) \log(1 - P_l) + \log\binom{\frac{N-3}{2}}{N-1} + \Delta h \qquad (5)$$

Constant node count (N) leads to a linear relationship between the logs of the consensus failure rate (1-Pl) and the link failure rate (1-Pl), as defined by the reliability gain k. Consensus reliability, under the assumption of constant link reliability, increases as N increases, illustrating the monotonicity of the PC as N increases. This equation illustrates a more direct relationship between link reliability and consensus reliability than equation (1). That means it's a solid resource for real-world Raft CM implementation in IIoT environments. The estimated value of Δh is for a range of N values from 5 to 19, where Δh is constant regardless of N. From the simulation results in Section 4, we can see that even at Pl = 90%, the consensus failure rate log (1-P_C) follows the linear relationship in equation (2).

3.3 Reliability and quality and Its Correlation with Latency

In this section, we'll demonstrate that the two concepts, consensus reliability and consensus latency, are mutually exclusive. To demonstrate the effects of communication on the overall consensus latency, we use a wireless communication model developed to analyse the packet error probability of the wireless short package transmissions in URLLC [10] to determine a relationship between the consensus success rate PC and the consensus latency T, which we assume is caused by downlink and uplink transmission delay. This is just an example, and the letter's main conclusion will hold with any number of alternative models. This paper shows that the link failure rate 1 Pl used in equations (1) and (2) can be expressed as a function of temperature, as shown in [10]:

$$1-P_l = f_Q\left(\frac{B\frac{T}{2N}(C-R) + \frac{\log_2 B\frac{T}{2N}}{2}}{\left(B\frac{T}{2N}\right)^{1/2} \log_2(e)}\right) \qquad (6)$$

Where B represents the total amount of usable spectrum. The uplink or downlink transmission rate, R, and the channel capacity, C, are defined as follows: Keep in mind that here we assume that both uplink and downlink transmissions are time-divided; that is, that each transmission can have t=T/2N transmission internals, since there are a total of N transmissions in both directions. Since N remains constant, an increase in the consensus delay T affords more time t for each link transmission, which should theoretically result in a lower link failure rate (1-P$_l$). By plugging Equation (3) into Eqns. (1) or (2), we can obtain the relationship between reliability 1-P$_C$ and 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.

Q-function

$$\frac{\partial Q}{\partial T} = \frac{\frac{B}{2N}(C-R) - \frac{1}{2T}\log_2\left(\frac{T}{2N}B\right) + \frac{1}{T}(\ln 2)}{2\sqrt{\frac{T}{2N}B\log_2(e)}} \quad (5)$$

The derivative ∂Q/∂T in equation always keeps positive, which means the variable Q increases monotonically along with T. Based on the decreasing monotonicity of Q-function fQ(*)along with Q and the increasing monotonicity of PC along with P$_l$, the time delay of consensus T and consensus reliability 1−P$_C$ are contradictory. According to the conclusion in Section 3.1, 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=T/2N for each link, thus causes a smaller Pl, 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.

4. SIMULATION RESULTS

The proposed consensus communication model and its derivations are put through simulations for verification.

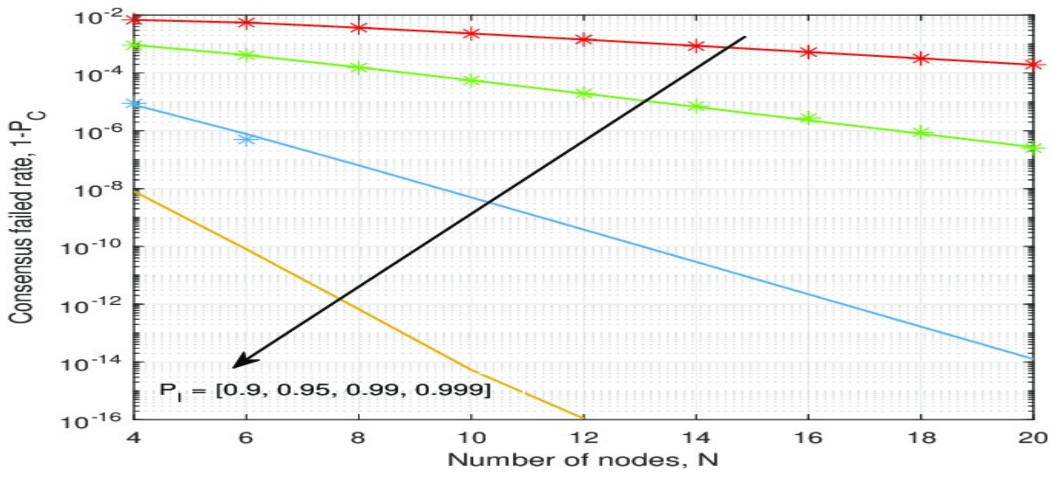

Figure 2: analytical results

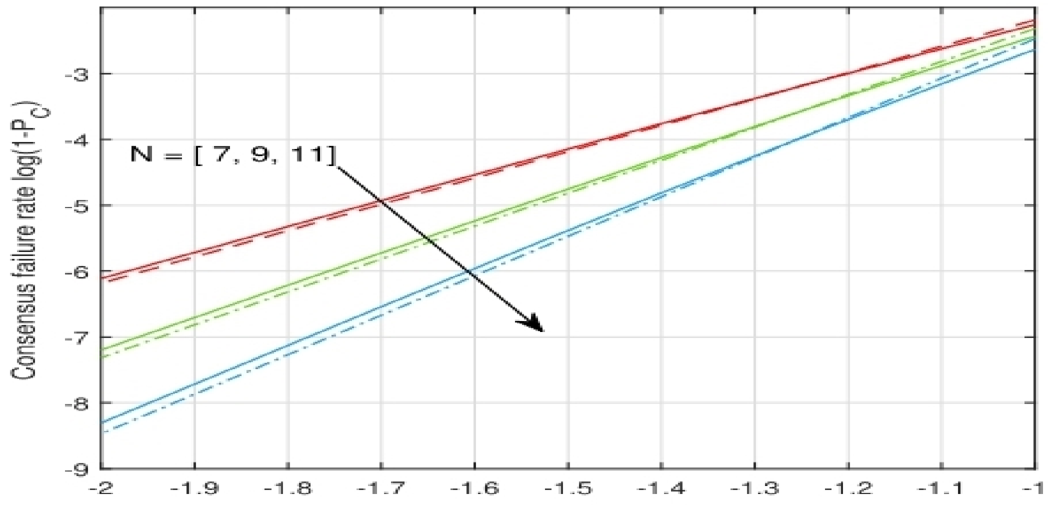

Figure 3: Simplified analytical results

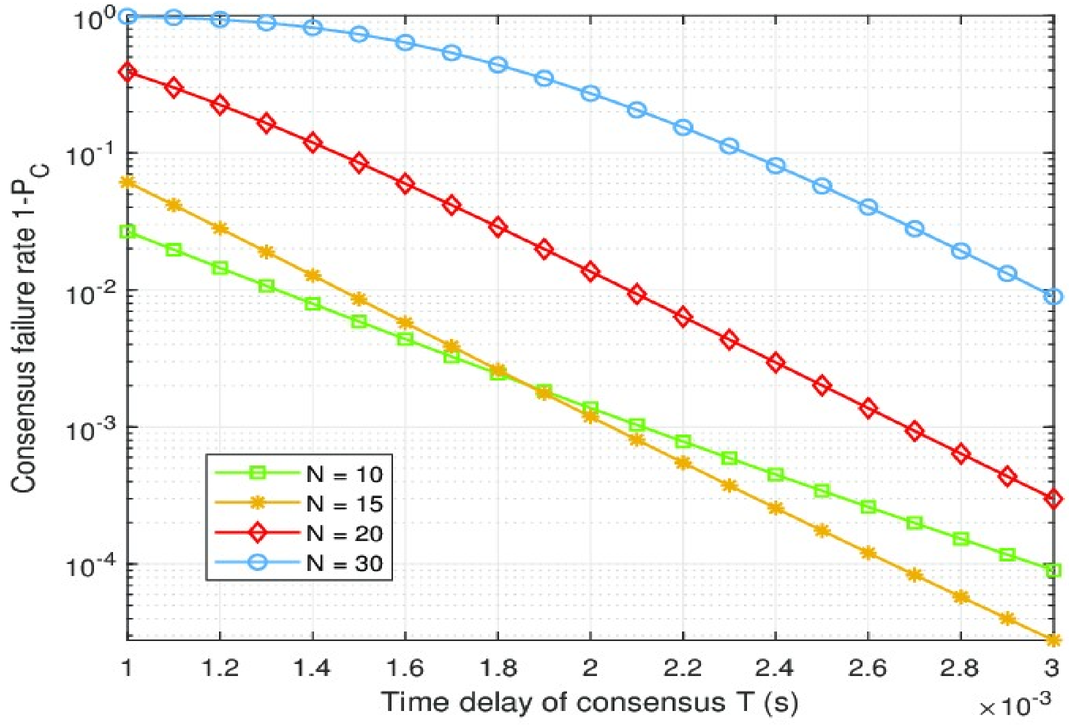

Figure 4: Consensus failure rate vs Consensus delay analysis

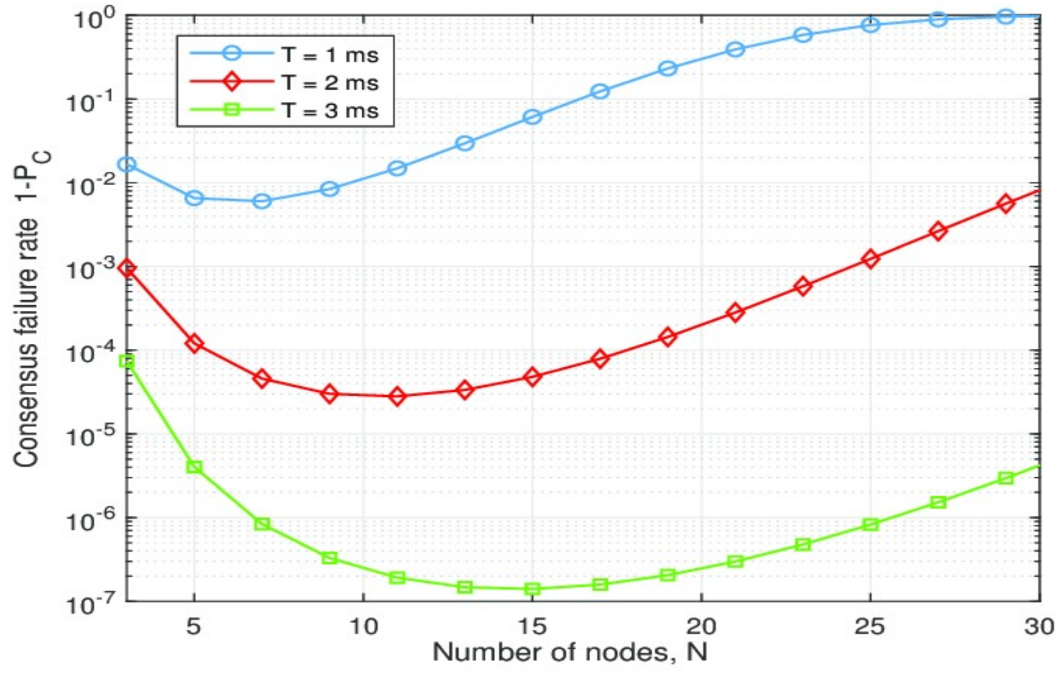

Figure 5: Consensus failure rate vs Nodes number

The signal-to-interference-plus-noise ratio (SINR) is set to 10 dB, and the bandwidth for link B transmission is 18 kHz. The formula for determining channel capacity is C=log(1+SINR), which is assumed to be 50% of the uplink and downlink capacities R.

As shown in Figure 2, as the number of nodes N in the Raft link failure model grows so does the percentage of successful consensus attempts. While the communication link success rate P_l increases from 91% to 96% to 98.9% to 99.8% as the number of nodes grows, the consensus failure rate 1-PC remains relatively constant throughout. When the link success rate P_l=91% and 96%, the simulated results (in asterisks) for the consensus failure rate 1-P_C are superimposed with their respective analytical curves (in lines), demonstrating the correctness of the equation (1). The consensus success rate P_C is shown to increase monotonically with the number of nodes N, as depicted by the analytical curves. Due to the fact that the consensus failure rate is extremely low for a larger P_l and the fact that the computational power of MATLAB is limited, the full simulated result of the consensus failure rate 1-P_C cannot be shown in Figure 1 for P_l values of 98.8% and 99.6%.

The consensus reliability trend and link success rate are displayed in Figure 3. The analytic answer to (1) is the logarithmic representation of the initial consensus reliability of 1-P_C. The logarithm (1-P_C) in equation (2) gives the reduced form of the failure rate at reaching consensus. The close agreement between the analytical and simplified lines demonstrates that the linear relation in the equation is indeed correct (2). As the number of nodes increases, the reliability gain, denoted by the slope, increases to k=(N+1)/2. This finding hints that a simplified model can be used to direct the actual deployment of Raft in distributed systems.

The conflict between consensus reliability (1/P_C) and consensus delay (T) is illustrated by the simulation in Fig. 4. There are four distinct curves that can be drawn using the values 10, 15, 20, and 30 for N. Figure 4's straight lines demonstrate that, for a given number of nodes, the consensus reliability 1-P_C decreases as the time delay T increases, thereby demonstrating the incompatibility of the two concepts.

When the time delay increases, two curves intersect, and the consensus failure rate at N=15 tends to decrease more dramatically than the consensus failure rate at N=10. Given the impact that consensus delay T has on link transmission reliability, it follows that consensus reliability does not exhibit monotonicity with the number of nodes N. As a result, more research into this phenomenon is required.

There is no change in the consensus time latency T, and Figure 5 shows how the number of nodes affects the consensus reliability 1-P_C. The graphs demonstrate that there is a maximum level of consensus reliability and that it fluctuates as the number of nodes increases. Changing the time delay in the consensus system will cause the reliability curve's maximum value to move upwards towards a larger value as the number of nodes increases. When N is small, the consensus failure rate follows the monotonicity in equation (1); however, when N is large, 1-P_l increases dramatically along N due to the property of the Q function in equation (3), which causes the rise of 1-P_C. This is because the time latency in each link transmission decreases as a result of the scarcity of communication resources (i.e., the communication time T). As a result, node N has both beneficial and detrimental effects on the consistency of the consensus. The bending of lines shows that in order to meet the needs of various IIoT scenarios, it is possible to optimize consensus reliability by allocating communication resources.

5. CONCLUSION

Consensus reliability in Raft can be made ultra-reliable even with low communication link reliability, according to the analysis of consensus reliability in the distributed IIoT system. A linear interpretation of the relationship between consensus reliability and communication link reliability is used for the sake of clarity. Meanwhile, the data indicates that the consensus reliability in Raft is at odds with the time latency. Thus, this article serves as a helpful reference for those planning to implement Raft in their distributed system's architecture.

REFERENCES:

[1] G. Brown, "Ultra-reliable low-latency 5G for industrial automation," in Heavy Reading White Paper for Qualcomm, 2018.

[2] S. Mumtaz, A. Alsohaily, 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.

[3] Y. Rao et al., "New services & applications with 5G ultra-reliable low latency communication," 5G Americas, ellevue, WA, USA, Tech. Rep., 2018.

[4] 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.

[5] 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.

[6] 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.

[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] 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.

[10] M. Castro and B. Liskov, "Practical Byzantine fault tolerance," in Proc. 3rd Symp. Operating Syst. Design Implement., 1999, pp. 173–186.