RAFT Consensus Reliability in Wireless Networks: Probabilistic Analysis

Yuetai Li, Yixuan Fan, Lei Zhang, *Senior Member, IEEE* and Jon Crowcroft, *Fellow, IEEE*

Abstract—**The centralized system becomes less efficient, secure, and resilient as the network size and heterogeneity increase due to its inherent single point of failure issues. Distributed consensus mechanisms characterized by decentralization, autonomy, parallelism and fault-tolerance can meet the increasing demands of safety and security in critical interconnected systems. This paper establishes a Node and Link probabilistic failure model in the presence of node and communication link failures for a representative crash fault tolerant distributed consensus protocol: RAFT. The analytical results in terms of the probability density function and the mean value of consensus reliability are derived. Two important reliability performance indicators, Reliability Gain and Tolerance Gain are proposed to indicate the linear relationship between the consensus reliability and two basic parameters, i.e. the joint failure rate and the maximum number of tolerant faulty nodes, which provide the theoretical guidance for quickly deploying a RAFT system. The special case of a distributed consensus network with already a certain number of failures and its adverse impact are evaluated. The Markov probabilistic models, definitions of Reliability Gain and Tolerance Gain, and the analysis methods proposed in this paper can be extended to other consensus mechanisms.**

Index Terms—**Internet of Things, Distributed Consensus Mechanism, RAFT, Reliability, Fault Tolerance**

I. Introduction

Driven by advances in 5G, industry 4.0, cloud/edge computing and artificial intelligence, etc., the Internet of Things (IoT) is envisioned to be extensively applied to critical and complex systems, such as transportation, healthcare, automation, supply chain and finance sectors [1]. These vital societal and industrial functions are increasingly interconnected for information exchange through communication networks to complete joint tasks for achieving a safer, securer and more efficient digital society. For instance, connected autonomous vehicles may exchange information and make joint decisions with proximity in a real-time manner based on the data collected by sensors equipped locally (e.g., infrared sensor, inductive sensor, etc.).

Centralized architectures have been widely deployed in interconnected systems in the industry. Under such an architecture, the central node is highly resource demanding since all other nodes can only synchronize the states with the central one. A failure of the central node leads to a system crash. Thus, the reliability performance of the network is determined by the condition of the central node [2]. In addition, such an architecture limits the topology of network communication to some extent, putting enormous pressure on important links [3]. On the other hand, the central node has higher privileges than other nodes. In some privacy-sensitive situations, the central node may trigger privacy issues because of its dominance over the system. The issues are worse with the increase of network size, nodes heterogeneity and security threaten.

Distributed systems, on the other hand, can provide fault-tolerant, transparent and robust solutions that maintain the efficiency, security, and scalability of large systems. Research on distributed systems has led to a large number of applications such as distributed computing [4], federated learning [5], blockchain [6], and distributed sensor networks [7]. However, network delay, time and clock issues are the main challenges in distributed systems [8]. Distributed consensus algorithms are designed to ensure that most of the network nodes can achieve the necessary agreement on states and tolerate certain types of faults in the progress [9]. As such, they are considered as a core of the distributed system to provide liveness, safety and fault-tolerance capability [10].

A. Background: distributed consensus algorithm

Consensus algorithms can be categorised as Byzantine fault tolerance (BFT) and crash fault tolerant (CFT) protocols. BFT protocols like Practical Byzantine Fault Tolerance (PBFT) [11] and Hotstuff [12] are introduced in decentralized networks against the Byzantine failure, which refers to the malicious behaviors given by an adversary, including contradictory commands to the progress, communication abort, and lengthy intentional delay to critical messages. For decentralized systems with high openness, the byzantine fault-tolerant design is necessary because the nodes in the network are not trusted. However, for most critical systems, the nodes can be assumed trusted by authentications, thus, it is redundant to consider Byzantine tolerance in the majority of scenarios. Moreover, the tolerance of Byzantine nodes can significantly increase the complexity of the algorithm and reduce the system throughput and scalability, which may result in the loss of suitability for IoT systems composed of low-cost and low-power devices.

In the trust distributed systems, the CFT protocol with node crash tolerance feature is sufficient to manage reliable state duplication and prevent the system breakdown. The first CFT consensus algorithm, Paxos, was proposed by Lamport in 1998

Yuetai Li, is with Glasgow College, University of Electronic Science and Technology of China, Sichuan, Chengdu, No.2006, Xiyuan Ave, West Hi-Tech Zone, China (e-mail: yuetail9@163.com).

Yuetai Li, Yixuan Fan and Lei Zhang (corresponding author) are with James Watt School of Engineering, University of Glasgow, Glasgow, G12 8QQ, United Kingdom (e-mail: 2510959L@student.gla.ac.uk, y.fan.3@research.gla.ac.uk, Lei.Zhang@glasgow.ac.uk).

Jon Crowcroft is with Computer Lab, University of Cambridge, Cambridge, CB2 1TN, United Kingdom (e-mail: jon.crowcroft@cl.cam.ac.uk).

[13], while its correctness and efficiency have been proved in subsequent work. However, the original Paxos algorithm has two issues [14]. Firstly, it is admittedly too hard to understand even with several papers emerging to explain Paxos in simpler terms [15], [16]. Additionally, Lamport's description focuses on single Paxos but lacks many details about multi-Paxos, which is more likely to be applied in the real world. Therefore, Paxos does not provide a good foundation for building an actual system [14].

Consensus algorithms Viewstamped Replication (VR) [17], Zookeeper atomic broadcast protocol (ZAB) [18] and RAFT [14] were proposed successively as variants of Paxos. Compared with Paxos, they provide detailed protocol details to facilitate the real-world deployment of distributed systems. However, VR requires a large number of different message types to achieve consensus, resulting in a relatively complex protocol. In addition, some messages need to carry entire log information, leading to high network transmission costs and reduces the efficiency of the algorithm. ZAB consensus algorithm is an efficient atomic broadcast protocol adopted by the Zookeeper system. However, ZAB was not widely deployed since its complex modules are difficult to be abstracted as a general library, while the system is communication resource demanding in the stage of leader selection.

RAFT was proposed in 2014 [14], thanks to its simplicity, a lot of attention has been drawn from academic community on validation [19], improvement [20], and application [21]. Additionally, RAFT is rapidly accepted by the industry due to its convenient modular deployment with independent functions, low communication complexity and high throughput. It is worth noting that the property of RAFT significantly matches the scenarios of IoT systems, in which the devices are typically low-cost and low-power.

B. Motivations and State of the Art

The tolerance to fault nodes is the most important feature of the CFT algorithm. For most of the CFT consensus algorithms, more than half of the nodes need to survive to ensure a consensus [14], [15], [18]. However, due to inevitable natural damage, subsystem malfunction, limited lifespan, inadequate energy supplement and jammer attacks [22]–[24], the connected nodes are easy to crash. If more than half of the nodes are crashed, the consensus cannot be achieved. Therefore, node reliability is an important factor for the consensus reliability of IoT systems.

Additionally, the distributed consensus is originally designed in stable wired communication networks, where the performance degradation caused by communication failures is negligible. Nevertheless, to support large-scale systems and allow connections among IoT devices and other clients, wireless communication is mandatory in operations of consensus [25], [26]. A wireless IoT network may suffer from various levels of transmission link failure due to channel fading or spectrum jamming in an open wireless channel [27], [28]. Since a consensus protocol relies on information exchange, the link failure may cause consensus transaction failure depending on the communication link reliability. Additionally, network communication failures could lead to network partitions [29], [30], insulating part of the cluster to access to the majority. Even though protocol improvements such as RAFT's leader election strategy [31] can effectively maintain consensus, partitioning a portion of the cluster may still reduce consensus reliability in wireless networks. This is because the system redundancy and fault-tolerant ability are reduced due to partition.

Thus, the integrated influence of communication link failure and node crash to the RAFT consensus algorithms may significantly affect the probability of achieving consensus, which is of importance to any type of mission critical interconnected distributed system deployed in real-world.

State of the art research considers the consensus in a determinate manner, i.e., the link or node is either faulty or non-faulty. However, in the real world applications, in particular in engineering field, it is unrealistic to make such assumption since all nodes and links connecting them are unreliable, thus should be modeled as a probability question. On the other hand, availability of quorum system [31], [32] characterizes how reliable the service provided by a quorum system is from the probabilistic perspective. However, they do not consider the communication reliability in the quorum system, and also lack reliability analysis for consensus protocols.

Some recent work has studied the applications of wireless distributed consensus systems [33]–[38]. For instance, [33] proposed a distributed reputation-based blockchains with consensus mechanism in the mobile-edge computing network for IoT. Additionally, [39]–[42] proposed RAFT-based blockchain applications in IoT. But most of the preceding works on wireless distributed consensus only concentrate on specific application scenarios including blockchain, edge computation, dynamic spectrum access, federated learning, etc., and the relationship between wireless communication and consensus protocol operations are not clarified. [26] and [25] analyze the impact of wireless communication on consensus reliability and latency in detail, but their models are based on specific distance information and cannot be effectively extended to a generic case.

Based on the link reliability, RAFT in a wireless communication network has been investigated to show high reliability critical decision making with relatively low link reliability in IoT [3], [43]. Reliability Gain was firstly proposed in [3], which refers to the linear relationship between consensus reliability and link reliability. However, the corrected index in the linear relationship of Reliability Gain in [3] is ambiguous since it only presents some specific values of the corresponding index to correct the large deviation between their approximation and theoretical result. Additionally, the analysis in [3] does not consider the impact of node reliability. [43] proposed a RAFT-based Internet of Vehicles, and considers both node reliability and link reliability. But it lacks with more analytical result to demonstrate the property of consensus reliability. Moreover, the results of [3], [43] only analyze the consensus reliability assuming the reliability of all the links to be the same, but different links might have different quality of communication channels in practical application scenarios. More general and explicit mathematical results should be revealed to calculate the consensus reliability and provide a generic explanation of the Reliability Gain or other properties in RAFT and other consensus protocols.

C. Contributions

This article proposes step forward approaches to calculate the consensus reliability of RAFT in probability in the presence of node and communication link failures. It provides theoretical guidance to deploy the RAFT consensus protocol in the real-world decentralized critical networks. The main contributions are listed below.

- We propose a node and link probabilistic failure model using the Markov property to calculate the probability density function (p.d.f.) and mean value of the consensus reliability. The methods of joint failure and power series are proposed to simplify the calculation. The proposed approach is generic thus can be applied to other consensus protocols.
- We derive generic and accurate Reliability Gain and defined the Tolerance Gain to indicate the linear relationship between the consensus reliability and two basic parameters, the joint reliability and the threshold of faulty nodes.
- We derive analytical relationship to show that the number of deterministic failure nodes in the network will increase the order of magnitude of consensus failure rate linearly, which may cause disastrous results when the joint reliability are not perfect.

The structure of this paper is explained as follows. The system model of the RAFT based distributed consensus is given in Section II. Section III proposes the consensus reliability in the view of a random variable, while Section IV analyzes the properties of the mean value of the consensus reliability under certain assumptions. In Section V, further potential explorations of wireless consensus protocols are briefly discussed. Simulation and results are demonstrated in Section VI to verify the analysis of the consensus in probability. Finally, Section VII makes a conclusion.

II. System model

The RAFT network is composed of consensus nodes with full functions, including synchronizing the state of other nodes or exchanging the voting messages while acting as a leader or follower. The roles of leader and followers are exchangeable during the RAFT leader election [14]. As the primary consensus node, the leader needs to pack the commands in log entries and replicate these entries to all followers successively through downlink, which is defined as the communication from the leader to the followers [26]; the followers need to send back the confirmation to the leader through uplink, which is defined as the communications from followers to the leader [26] when they receive the package from the leader successfully. A successful RAFT transaction consensus represents that the leader receives the confirmations of more than 50% of followers successfully in one transaction consensus.

As discussed above, both nodes and links may fail in a wireless connected network at any stage of the uplink or downlink in the RAFT network. A generic probabilistic failure model where the node failure and link failure are measured by probability will be derived to explore the impacts of node and link reliability on the reliability of RAFT consensus in wireless environments. The log replication process from the leader to followers in RAFT is analyzed in detail in this paper since this is the core of the consensus and the most frequent process of system operation. Other process such as leader election are not considered but might be further modeled based on the analysis in this paper. Such a probabilistic model is critically important to the real engineering where all entities are not 100% reliable, limited by cost, size and complexity etc. Additionally, the leader is assumed to be reliable since RAFT has a robust leader election mechanism to deal with the crash of the leader.

In this paper, the reliability of each node and the reliability of each link are assumed as random variables determined by the actual practical application scenarios to evaluate their impacts on the final transaction consensus reliability in Section III. The assumption of random variables is because the quality of a certain channel or the reliability of a certain node may vary at different observation time. This provides detailed analysis of consensus reliability when links have different quality of communication channel or nodes have different reliability, thus other analysis with special assumptions are the subset and simplified form of the analysis in Section. III. The model could also be linked with the conventional network reliability theory (NRT) [44], [45], where the overall feasibility of a network is measured by the reliabilities of nodes and edges from the statistical angle.

Additionally, the link reliability can also be further modeled through classic signal to interference plus noise ratio (SINR) in physical layer, while the node reliability can be modeled as a specific model such as Weibull distribution [46]. However, this is not the focus of this article.

Moreover, if the malicious attacks to the IoT network cannot be ignored in the application scheme, byzantine fault tolerance consensus like Hotstuff [12] can be implemented with validly extended derivations.

All the frequently used notations are indicated in Table I.

TABLE I: Frequently used notations

Notation	Definition
P_C	Consensus successful rate in RAFT consensus system
P_F	Consensus failure rate in RAFT consensus system
P^{DL}	Success rate of downlink communication in RAFT consensus system
P^{UL}	Success rate of uplink communication in RAFT consensus system
P^N	Node reliability in RAFT consensus system
N	Number of total nodes in RAFT consensus system
n	Number of followers in RAFT consensus system
f	Maximum number of faulty nodes the RAFT consensus system can tolerate
s	Number of already deterministic faulty nodes in RAFT consensus system

III. Consensus Reliability

We will derive the closed-form expression of the RAFT consensus in probability (probability density function and mean value) by assuming each node's reliability and each communication link's reliability are random variables. Then, the methods of joint failure and power series are proposed to simplify the analysis.

A. Derivation of consensus reliability in the view of random variable

Let the number of all the nodes in RAFT system be N, and the number of followers be $n = N - 1$. Since RAFT can tolerate half of the followers crashed, the threshold of number of faulty nodes f satisfies $f = \lfloor n/2 \rfloor$. We consider the phase that the leader send messages to the followers as one downlink communication and the phase that a follower responds the leader as one uplink communication. Let $\Omega = \{N_1, N_2, N_3, \ldots, N_n\}$ represent the set of n followers connected with the leader. Additionally, let the reliability of node i ($i = N_1, N_2, ...N_n \in \Omega$) is a random variable P_i^N, the reliability of the downlink between the node i and the leader is a random variable P_i^{DL}, and the reliability of the uplink between the node i and the leader is a random variable P_i^{UL}. In the log replication phase of a consensus transaction, let $S_{1,x}, S_{2,y}, S_{3,z} \subseteq \Omega$ be the set of non-faulty nodes, the set of followers that successfully receive the message by the leader through downlink, the set of followers whose messages are successfully received by the leader in uplink, with the number of elements in them satisfying $|S_{1,x}| = x, |S_{2,y}| = y, |S_{3,z}| = z$, respectively. Let $P\left(S_{1,x}, S_{2,y}, S_{3,z}\right)$ represents the probability that all the followers in set $S_{1,x}$ are non-faulty, all the followers in set $S_{2,y}$ have successful downlink communication and all the followers in set $S_{3,z}$ have successful uplink communication.

In the RAFT consensus protocol, only the non-faulty nodes can receive messages from the leader by downlink communication, and only the nodes that receives the leader's message can send the response back to the leader by uplink communication. Therefore, $S_{3,z} \subseteq S_{2,y} \subseteq S_{1,x} \subseteq \Omega$ and $z \leq y \leq x \leq n$. In addition, for the next state, the last state contains all the information of previous states for the state transitions, e.g. the conditional probability $P(S_{3,z}|S_{1,x}, S_{2,y})$ is equal to $P(S_{3,z}|S_{2,y})$. Thus the state in the next stage is only dependent with the state in the last stage and the consensus process considering node failures and link failures satisfies Markov property. Thus $P\left(S_{1,x}, S_{2,y}, S_{3,z}\right)$ can be calculated as

$$P\left(S_{1,x}, S_{2,y}, S_{3,z}\right) = P(S_{1,x}) P(S_{2,y}|S_{1,x}) P(S_{3,z}|S_{2,y}) \tag{1}$$

where $P(S_{1,x})$ represents the joint probability that the nodes in $S_{1,x}$ are non-faulty while in $\complement_{\Omega} S_{1,x}$ are faulty; $P(S_{2,y}|S_{1,x})$ represents the joint probability that downlink with nodes in $S_{2,y}$ are successful while that in $\complement_{\mathrm{S_{1,x}}} S_{2,y}$ are faulty; $P(S_{3,z}|S_{2,y})$ represents the joint probability that uplink with nodes in $S_{3,z}$ are successful while that in $\complement_{\mathrm{S_{2,y}}} S_{3,z}$ are faulty.

If the reliability of node, uplink and downlink of different followers are not independent (e.g., correlation loss of wireless communication), chain rule can be used to calculate $P(S_{1,x}) P(S_{2,y}|S_{1,x})$ and $P(S_{3,z}|S_{2,y})$. For example, as for the uplink communication, $P(S_{3,z}|S_{2,y})$ is a joint probability of y events (z events for communication success and $y - z$ events for communication failure). If all these events are noted as $A_1, A_2, \ldots, A_y$, $P\left(S_{3,z}|S_{2,y}\right)$ can be expressed according to the chain rule as:

$$\begin{aligned} P\left(S_{3,z}|S_{2,y}\right) = & Pr\left(A_1\right) Pr\left(A_2|A_1\right) Pr\left(A_3|A_1 A_2\right) \\ & \ldots Pr\left(A_y|A_1 A_2 \ldots A_{y-1}\right) \end{aligned} \tag{2}$$

Consider a simple case, where the nodes N_1 and N_2 are in $S_{3,z}$ for a certain Markov chain. Assuming only the uplink between N_1 and N_2 are correlated, thus we have,

$$\begin{aligned} P\left(S_{3,z}|S_{2,y}\right) = & Pr\left(Z_{N_1}\right) Pr\left(Z_{N_2}|Z_{N_1}\right) \\ & \prod_{u \in S_{3,z} - N_1 - N_2} P_u^{UL} \prod_{v \in \complement_{S_{2,y}} S_{3,z}} 1 - P_v^{UL} \end{aligned} \tag{3}$$

Therefore, based on similar conditional probabilities, the correlation between wireless communications can be fully characterized as long as there is sufficient prior information about each channel.

To facilitate the derivation, assuming node reliability, uplink reliability and downlink reliability of different followers are independent, we have

$$P\left(S_{1,x}\right) = \prod_{u \in S_{1,x}} P_u^N \prod_{v \in \complement_{\Omega} S_{1,x}} 1 - P_v^N \tag{4}$$

$$P(S_{2,y}|S_{1,x}) = \prod_{u \in S_{2,y}} P_u^{DL} \prod_{v \in \complement_{S_{1,x}} S_{2,y}} 1 - P_v^{DL} \tag{5}$$

$$P\left(S_{3,z}|S_{2,y}\right) = \prod_{u \in S_{3,z}} P_u^{UL} \prod_{v \in \complement_{S_{2,y}} S_{3,z}} 1 - P_v^{UL} \tag{6}$$

According to the RAFT protocol, when the number of messages the leader receives from the followers z is no less than $n - f$, where $f = \lfloor n/2 \rfloor$, the cluster will reach consensus. Therefore, the probability that the cluster successfully reaches a consensus P_C is the sum of probabilities of all the Markov chain $(S_{1,x}, S_{2,y}, S_{3,z})$ satisfying $n \geq x \geq y \geq z \geq n - f$ and $\Omega \supseteq S_{1,x} \supseteq S_{2,y} \supseteq S_{3,z}$

$$P_C = \sum_{n \geq x \geq y \geq z \geq n-f} \; \sum_{\Omega \supseteq S_{1,x} \supseteq S_{2,y} \supseteq S_{3,z}} P\left(S_{1,x}, S_{2,y}, S_{3,z}\right) \tag{7}$$

Note that the second summation is to traverse over all combinations of possible node sets for given x, y and z. In fact, P_C is a function of random variables of node reliability and link reliability. According to Eq. (1)-(7), we can theoretically analyze the influence of the reliability of each node/link on the final consensus reliability. It implies that Eq. (1)-(7) can be universally applied to arbitrary practical situation as long as the node reliability P_i^N, the downlink reliability P_i^{DL} and the uplink reliability P_i^{UL} of all nodes in Ω are given.

In addition, using the Markov property to analyze consensus reliability considering node reliability and link reliability can also be applied to other BFT consensus protocols, such as PBFT [11] and Hotstuff [12], to carefully examine the impacts of node reliability and link reliability on each stage of the consensus protocol. However, it is beyond the scope of this paper.

B. Joint failure method

Through mathematical derivations, we transform Eq. (1), (4) - (7) as:

$$P_C = \sum_{n\geq k\geq n-f} \sum_{\Omega\supseteq S_{J,k}} \prod_{u\in S_{J,k}} P_u^J \prod_{v\in \complement_\Omega S_{J,k}} 1-P_v^J \quad (8)$$

where $S_{J,k}$ is a running variable of subset of nodes with $|S_{J,k}|$ equal to k and $P_i^J = P_i^N P_i^{DL} P_i^{UL}$.

The identity between Eq. (8) and Eq. (1), (4) - (7) is proven in Appendix A.

In fact, Eq. (8) has obvious physical meaning. Eq. (1), (4) - (7) derived consensus reliability following the consensus process which reflects the influence of each failure factor at different stages. However, it is worth noting that the log replication of RAFT or most of other CFT consensus is so concise and light since it only involves one uplink and one downlink communication. Based on this feature, either the failure happens on the node on its own, or its communication channel with the leader (the link) will result in the failure of this node to contribute to the consensus. Consider the non-faulty nodes which both receive the leader's message successfully and send their responses to the leader successfully as contributors. The random variable, probability of node i becoming a contributor (also refers to the joint success rate) P_i^J, can be determined as $P_i^J = P_i^N P_i^{DL} P_i^{UL}$. Let $S_{J,k} \subseteq \Omega$ represent the set of contributors. As long as the number of contributors (i.e. k) is more than $n-f$, the consensus will be achieved. Thus we have Eq. (8).

C. Power series of consensus reliability

Although the joint failure view is proposed to simplify the consensus reliability, Eq. (8) seems still complicated because it is the summation of $\sum_{i=n-f}^{n} \binom{n}{i}$ terms (equal to 2^{n-1} for odd n and $2^{n-1} + \binom{n}{f}/2$ for even n) with each term being the product of n random variables. In this section, the nsensus reliability is further simplified by the method of power series.

Firstly take the view of consensus failure. Let the consensus failure rate be $P_F = 1 - P_C$ and joint failure rate of the node i be $P_i^{JF} = 1 - P_i^J$. Since no less than $f+1$ nodes not becoming contributors will cause the consensus failure, P_F can be calculated as:

$$P_F = \sum_{n\geq k\geq f+1} \sum_{\Omega\supseteq S_{JF,k}} \prod_{u\in S_{JF,k}} P_u^{JF} \prod_{v\in \complement_\Omega S_{JF,k}} 1-P_v^{JF} \quad (9)$$

Note that the term $\prod_{u\in S_{JF,k}} P_u^{JF}$ is the product of k joint failure rates since $|S_{JF,k}| = k$. Actually, in practical application scenarios, the joint failure rate P_i^{JF} is usually small i.e. closer to 0. Otherwise, a large number of nodes and links would fail so that the system might not carry out any consensus transaction. Taking advantage of this characteristic, we propose a power series expression of the consensus failure rate P_F based on the joint failure rate P_i^{JF} as

$$P_F = \sum_{t=f+1}^{n} a_t Q_t \quad (10)$$

where $Q_t = \sum_{\Omega\supseteq S_{JF,t}} \prod_{u\in S_{JF,t}} P_u^{JF}$ is the summation of $\prod_{u\in S_{JF,t}} P_u^{JF}$, and $\prod_{u\in S_{JF,t}} P_u^{JF}$ is the product of t joint failure rates which can be considered as power of t, and $a_t = (-1)^{t-f-1}\binom{t-1}{f}$ is the coefficient of the series expansion.

The proof is given in Appendix B.

Note that the term Q_t will become smaller with the increment of t, since larger t leads to both the product of the joint failure rates $\prod_{u\in S_{JF,t}} P_u^{JF}$ smaller and the summation number $\binom{n}{t}, t \geq f+1$ (how many products are summed in Q_t) smaller. Therefore, similar to the Taylor series, the high-order power terms (Q_t with larger t) can be allowed to be omitted to achieve the purpose of simplifying P_F. This provides a flexible way to simplify P_F according to actual approximate accuracy requirements. For example, after retaining only the first non-zero term, the reliability consensus can be further simplified as

$$P_F \approx Q_{f+1} = \sum_{\Omega\supseteq S_{JF,f+1}} \prod_{u\in S_{JF,f+1}} P_u^{JF} \quad (11)$$

Theoretically, an approximation of Eq. (10) with arbitrary precision for any P_{JF} can be obtained as long as more power series terms Q_t are retained. Thus the power series expansion form of Eq. (10) could provide great convenience and flexibility for accurate approximations in engineering.

D. p.d.f. of the logarithmic consensus failure rate

Considering the actual application scenario, we generally pay attention to the order of magnitude of the probability of transaction failure (e.g., in some safety critical applications, the reliability requirement could be as high as $1-10^{-9}$). We let $H = \log(P_F)$ represent the logarithmic consensus failure rate. According to Eq. (9), H is a function of the random variables of the joint failure rate of different followers P_i^{JF}. Thus the cumulative distribution function (c.d.f.) of H is the integration of the joint p.d.f. of followers not becoming contributors. Let $f_{P_i^{JF}}\left(p_i^J\right)$ represents the p.d.f. of the follower i not becoming a contributor. Since different nodes are independent, $F(H)$ can be written as

$$F_H(h) = \oint_A \prod_{i\in\Omega} f_{P_i^{JF}}(p_i^{JF}) dp_i^{JF} \quad (12)$$

where A is the integration area of $H \leq h$, which can be determined according to Eq. (9):

$$A: \begin{cases} \sum\limits_{n\geq k\geq f+1} \sum\limits_{\Omega\supseteq S_{JF,k}} \prod\limits_{u\in S_{JF,k}} p_u^{JF} \prod\limits_{v\in \complement_\Omega S_{JF,k}} 1-p_v^{JF} \leq 10^h \\ 0 \leq p_i^{JF} \leq 1 \end{cases} \quad (13)$$

and if the approximation of Eq. (11) is used to simplify H, the integration area will change into:

$$\widetilde{A}: \begin{cases} \sum\limits_{\Omega\supseteq S_{JF,f+1}} \prod\limits_{u\in S_{JF,f+1}} p_u^{JF} \leq 10^h \\ 0 \leq p_i^{JF} \leq 1 \end{cases} \quad (14)$$

Finally, p.d.f. of H is:

$$f_H(h) = \frac{dF_H(h)}{dh} \quad (15)$$

Eq. (12)-(15) represent that the p.d.f. of H can be obtained according to the p.d.f. of $f_{P_i^{JF}}\left(p_i^{JF}\right)$, which is determined by the application scenarios. Particularly, we take several representative p.d.f. of the node reliability and link reliability (including uniform, truncated Gaussian and exponential distribution) as examples and our numerical results of Eq. (12)-(15) show that the p.d.f. of H is close to Gaussian distribution, which is described in detail in Section VI. With the help of p.d.f. of H, the probability that H distributed at arbitrary interval can be calculated by integrating the p.d.f. of H on the corresponding interval to evaluate that how reliable the consensus protocol is for transaction consensus or critical decision making when links have different quality of communication channel or nodes have different reliability.

IV. Consensus Reliability Mean Value and Properties

In this section, we analyze the theoretical mean value of the consensus reliability given the mean value of the node and link reliability. Similarly, simplification analysis of joint failure view and power series are also used to explore properties of the consensus reliability in Sec. IV-A. In particular, in Sec. IV-B we proposed the concepts of Reliability Gain and Tolerance Gain and derived the closed-form equations to reveal the two linear relations of the consensus reliability mean value. Additionally, the impact of a special case, networks with already s failures, on the consensus reliability is analyzed in Section IV-D.

A. Consensus Reliability Mean Value

The mean value of Eq. (8)-(11) can be easily obtained under the independence condition of reliability of different node and link. For instance, Eq. (9) can be transformed into:

$$E(P_F) = \sum_{n \geq k \geq f+1} \sum_{\Omega \supseteq S_{JF,k}} \prod_{u \in S_{JF,k}} E(P_u^{JF}) \prod_{v \in \complement_\Omega S_{JF,k}} 1 - E(P_v^{JF}) \tag{16}$$

Although the joint failure method (Eq. (8)) and power series (Eq .(10) and Eq. (11)) are proposed to simplify Eq. (4) - (7), Eq. (11) might still be sophisticated for quick deployment because it is the summation of $\binom{n}{f+1}$ terms with each term being the product of $f+1$ variables. In order to conveniently deploy the wireless RAFT system based on consensus reliability in engineering, we give a more intuitive and concise analysis of the mean value of consensus reliability under certain assumptions in this section.

Assume that the link reliability and node reliability are independent, the node reliability P_i^N of each node are i.i.d with the mean value satisfying $E(P_i^N) = p_N$ and that the uplink and downlink reliability of each different link P_i^{UL}, P_i^{DL} are i.i.d with the mean value satisfying $E\left(P_i^{UL}\right) = E\left(P_i^{DL}\right) = p_L$. Let the mean value of consensus reliability $E(P_C) = p_C$. Then Eq. (1), (4) - (7) will be degenerated as:

$$p_C = \sum_{x=n-f}^{n} \binom{n}{x} p_N^x (1-p_N)^{n-x} \sum_{y=n-f}^{x} \binom{x}{y} p_L^y (1-p_L)^{x-y} \sum_{z=n-f}^{y} \binom{y}{z} p_L^z (1-p_L)^{y-z} \tag{17}$$

The authors in [43] considered RAFT-based Internet of Vehicles and also proposed Eq. (17). However, they did not consider the case that different nodes or links have different reliability, and there is no further simplification and analytical results for the consensus reliability in [43].

According to the method of joint failure view proposed in section III-B, let the mean value of the joint success rate of each node is $E(P_i^J) = p_J = p_N p_L^2$, then p_C can be calculated as:

$$p_C = \sum_{x=n-f}^{n} \binom{n}{x} p_J^x (1-p_J)^{n-x} \tag{18}$$

Let the mean value of the consensus failure rate be $p_F = E(P_F)$, the mean value of node failure rate is $p_{NF} = 1-p_N$, the mean value of link failure rate is $p_{LF} = 1-p_L$, and the mean value of joint failure rate is

$$p_{JF} = 1 - p_J = 1 - (1-p_{NF})(1-p_{LF})^2 \tag{19}$$

Similar with the power series analysis in Sec. III-C, since the joint failure rate p_{JF} is usually small in practical application scenarios, the mean value of the consensus failure rate p_F can be simplified as

$$p_F = \sum_{t=f+1}^{n} b_t (p_{JF})^t \tag{20}$$

where $b_t = (-1)^{t-f-1}\binom{t-1}{f}\binom{n}{t}$ is the coefficient of the series expansion. Note that Q_t in Eq. (10) is degenerated to become that $\binom{n}{t}(p_{JF})^t$ so that the coefficient b_t is multiplied with $\binom{n}{t}$ than a_t. After retaining only the first non-zero term, the mean value of consensus failure rate can be simplified as

$$p_F \approx \binom{n}{f+1} p_{JF}^{f+1}. \tag{21}$$

B. Reliability Gain and Tolerance Gain

According to simplified Eq. (21) preserving the first non-zero term in the power series method, it is obvious that approximated the consensus failure rate and the joint failure rate are linear in logarithmic form. This linear relationship is conceptually defined as the Reliability Gain, k_p, which is described in detail as follows:

Theorem 1. *The logarithmic consensus failure rate* $\log p_F$ *can be expressed in a linear relation of* $\log p_{JF}$ *with an error term, i.e.,*

$$\log p_F = k_p \cdot \log p_{JF} + h_p + \varepsilon \tag{22}$$

where the Reliability Gain $k_p = f+1$*, the intercept* $h_p = \log(\binom{n}{f+1})$ *and the error term* $\varepsilon \leq \log(\frac{(1-p_{JF})^{n-f} - p_{JF}^{n-f}}{(1-p_{JF})-p_{JF}})$.

Remark 1. *When the joint failure rate* p_{JF} *is reasonably small*[1]*, the error term in Eq.(22) is approximate to 0.*

The proof of Theorem 1 is shown in Appendix C. Theorem 1 shows for a consensus network with a fixed number of nodes, the logarithmic consensus failure rate $\log p_F$ will approximately decrease linearly by increasing the wireless

[1] Our results show that $p_{JF} = 0.1$ is small enough to make the conclusion, while this condition is achieved in most of the application environments.

communication quality or using higher reliability products to reduce $\log p_{JF}$. Therefore, Reliability Gain provides an intuitive indicator for evaluating the consensus reliability of the system with a deterministic size. For example, if the size of the consensus network is 13 (i.e., $N = 13, n = 12, f = 6$), when p_F is required to be less than $10^{-5}, 10^{-7}, 10^{-9}$, p_{JF} should be less than $10^{-1.15}, 10^{-1.45}, 10^{-1.75}$, reflecting the decrease of $\log p_{JF}$ by 0.3 makes $\log p_F$ decrease by 2.

Particularly, according to Eq. (19), if we take further approximation of p_{JF} as

$$p_{JF} \approx 1 - (1 - p_{NF})(1 - 2p_{LF}) \approx p_{NF} + 2p_{LF} \tag{23}$$

we can see the impact of communication reliability has a larger weight on the consensus failure. This is because there are two communication process, i.e. downlink and uplink. Thus increasing link reliability can obtain larger improvement of consensus reliability than increasing node reliability. Also the wireless link reliability can further modeled to characterize how transmission distance, communication source, noise, etc., affect consensus reliability. In addition, if one of node failure and link failure can be neglected, the joint failure rate will degenerate to another failure factor, thus $\log p_F$ will has the linear relation with $\log p_{NF}$ or $\log p_{LF}$.

Note that although the authors in [3] propose the concept of Reliability Gain for the first time, they do not consider node reliability, and in spite of only considering the link reliability p_L, there is a huge deviation in their approximation result, which will be discussed in Section VI.

If further transformations of the consensus reliability are carried, Tolerance Gain k_f can be obtained, which refers to $\log p_F$ is approximately linear to the maximum number of tolerant faulty nodes f. The following theorem gives a detailed mathematical expression of the Tolerance Gain.

Theorem 2. *When the joint failure rate p_{JF} is reasonably small*[2]*, The logarithmic consensus failure rate $\log p_F$ can be expressed in a linear relation of the faulty node threshold f with an error term*

$$\log p_F = \begin{cases} k_f \cdot f + h_f + \varepsilon \\ \text{(when the number of followers } n \text{ is even)} \\ k_f \cdot f + h_f + \log 2p_J + \varepsilon \\ \text{(when the number of followers } n \text{ is odd),} \end{cases} \tag{24}$$

where the Tolerance Gain $k_f = (\log p_{JF} + \log p_J + 2\log 2)$, the intercept $h_f = \log(\frac{p_{JF}}{p_J\sqrt{\pi}}) + \Delta f$, in which $\Delta f = -\frac{1}{2}\log(f)$ is the no-linear complementary term to decrease the approximation error, and the error term $\varepsilon \leq \log(\frac{(1-p_{JF})^{n-f} - p_{JF}^{n-f}}{(1-p_{JF})^{n-f} - p_{JF}(1-p_{JF})^{n-f-1}})$

Remark 2. *When the joint failure rate p_{JF} is reasonably small*[3]*, the error term in Eq. (24) is approximate to 0.*

The proof of Theorem 2 is shown in Appendix D.

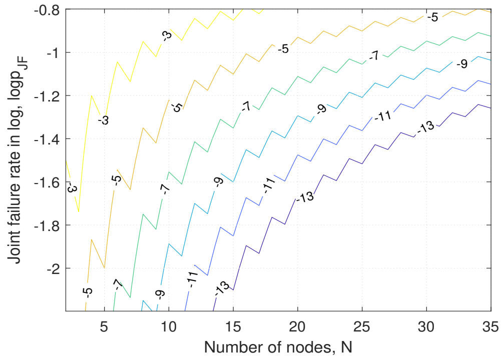

Fig. 1: The consensus network size N versus to $\log p_{JF}$ for a fixed $\log p_F$

Fault tolerance in wireless RAFT allowing at most f nodes failed infers one feature of the distributed consensus system is the resilience. Theorem 2 shows that $\log p_F$ will decrease linearly with the increment of f, which intuitively reflects the impact of the resilience of the wireless RAFT system on the transaction consensus reliability. Additionally, according to Eq. (24), p_F is $2P_J$ times larger when the number of followers n is odd than that when n is even. This is because there are fewer nodes required to become the contributors when n is even (i.e. $n - f = f$) than that when n is odd (i.e. $n - f = f + 1$). Therefore deploying even followers in a wireless RAFT system is recommended (Note that here n represents the number of followers). With the analysis of Tolerance Gain, the probability of reaching consensus can be adjusted by not only changing p_{JF} but also enlarging the size of the network to modify f. For instance, if p_{JF} in consensus network is $10^{-1.6}$, when p_F is required to be less than $10^{-5}, 10^{-7}, 10^{-9}$, the maximum number of tolerant faulty nodes f should be more than $3, 5, 7$, which reflects the increase of f by 2 makes $\log pF$ decrease by 2.

Reliability Gain and Tolerance Gain are complementary, which reveals that $\log p_F$ is linear with two significant parameters, $\log p_{JF}$ and f, in consensus network, respectively. With the help of Reliability Gain and Tolerance Gain, the arrangement of the number of total nodes and the joint reliability can be quickly calculated to satisfy the stringent reliability requirement in the system, which is shown in Fig. 1. Note that in Fig. 1, if one of these two factors is fixed, $\log p_F$ will be linear with another factor. It is suggested that the conceptions of Reliability Gain and Tolerance Gain can be applied to quickly check the consensus reliability and be the design guideline towards the real RAFT consensus mechanism deployment in the IoT systems.

In spite of the approximations in the derivations of Eq. (22) and Eq. (24), the consensus reliability results obtained from Reliability Gain and Tolerance Gain are sufficiently accurate as of the results without any approximation, which is shown in simulations in Sec. VI. Additionally, it is worth noting that besides the significantly light consensus protocol RAFT, the analysis and conceptions of Reliability Gain and Tolerance Gain can be similarly applied to other complicated consensus algorithms or can provide a reference for models in the network reliability theory.

[2] Our results show that $p_{JF} = 0.1$ is small enough to make the conclusion, while this condition is achieved in most of the application environments.

[3] Our results show that $p_{JF} = 0.1$ is small enough to make the conclusion, while this condition is achieved in most of the application environments.

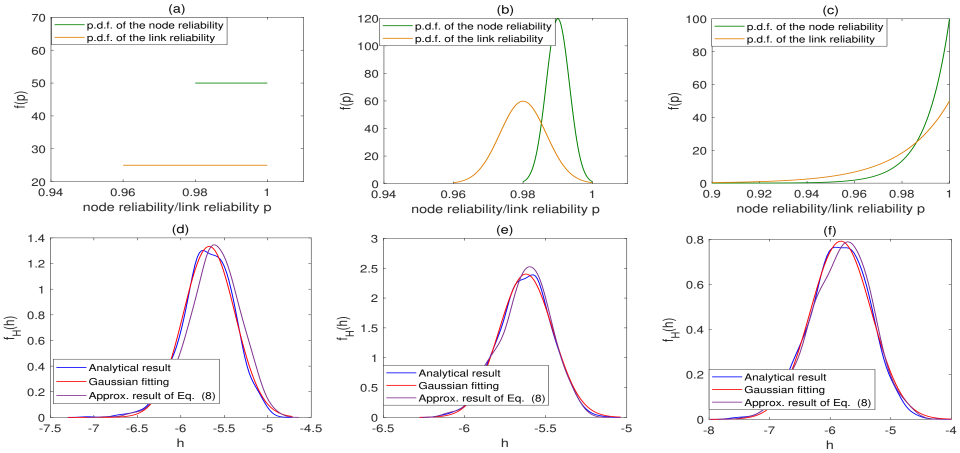

Fig. 2: The p.d.f. of the logarithmic consensus failure rate H. (a) the node reliability and link reliability are uniform distribution, and the corresponding p.d.f. of H is shown in (d); (b) the node reliability and link reliability are truncated Gaussian distribution, and the corresponding p.d.f. of H is shown in (e); (c) the node reliability and link reliability are exponential distribution, and the corresponding p.d.f. of H is shown in (f).

C. Dominant Analysis of node failure and link failure

In terms of different application scenarios, node reliability and link reliability have different impacts on consensus. If the wireless RAFT system suffers from much more link failures rather than node failures, it is reasonable to ignore node failure and vice versa. However, it is difficult to directly assess and reasonably interpret whether an influencing factor is so minor to be ignored. Since the reliability of nodes and links are usually discussed separately according to different application scenarios [45], we propose a method of dominant analysis to discuss in what conditions link failure rate, p_{LF}, or node failure rate, p_{NF} is dominant.

Let the consensus failure rate of neglecting link failure and that of neglecting node failure be p_{NLF} and p_{NNF}, respectively. To theoretically assess in what circumstances link reliability or node reliability play a dominant role, let $\varepsilon_L = \log p_F - \log p_{NLF}$ be the error of ignoring link failure and $\varepsilon_N = \log p_F - \log p_{NNF}$ be the error of ignoring node failure. Let $k = \frac{p_{NF}}{p_{LF}}$.

The threshold of neglecting non-dominant failure can be obtained as follow. Please see Appendix E for proof.

Remark 3. *For given ε_L and ε_N, if $k > \frac{2}{10^{\frac{\varepsilon_L}{f+1}}-1}$, link failure can be ignored in Node Link Failure model and the error is smaller than ε_N. If $k < 2(10^{\frac{\varepsilon_N}{f+1}}-1)$, node failure can be ignored in Node Link Failure model and the error is smaller than ε_N.*

D. Special case: network with deterministic s failures

A specific situation is considered in this section with a prior condition: there are already $s (s \leq f)$ deterministic equivalent failed nodes. Note that the equivalent failed nodes here not only includes the physical failure of the devices, the links of one node can be experienced eclipse attack [47] or some nodes can be partitioned by the network [29], which will make the leader unable to connect them for a period thus these followers are considered to be equivalent failed nodes. [4] For example, in a vehicular network system, vehicles may run to remote areas without a communication connection to the majority, or there is no communication authority between some of the vehicles. Despite the RAFT protocol can theoretically tolerate half crashes, the lack of some nodes in the cluster may affect the performance of achieving consensus due to the decreased redundancy.

Following the definitions and similar derivations in the section IV-B, the Reliability Gain k_p and intercept h_p in Theorem 1 can be obtained as:

$$k_p = f+1-s, h_p = \log\binom{n-s}{f+1-s} \tag{25}$$

while the Tolerance Gain k_f and intercept h_f in Theorem 2 can be calculated as:

$$\begin{aligned} k_f &= \log p_{JF} + \log p_J + 2lg2, \\ h_f &= \log(\frac{p_{JF}}{p_J\sqrt{\pi}}) - (\log 2 + \log p_{JF})s + \Delta f \end{aligned} \tag{26}$$

It is notable that in this case h_f in Eq. (26) is linear to s, which means by putting Eq. (26) into Eq. (24), the following property can be obtained: .

Remark 4. $\log p_F$ *has an approximation linearship with s with the coefficient of* $\log 2 + \log p_{JF}$

This implies that the consensus failure rate p_F might increase by orders of magnitude with the additive increment of s. It can be inferred that the impact of large s may be disastrous for the consensus system, especially when $s = f$ although this is theoretically allowed in the wireless consensus system. In fact, section VI shows that p_F is greater than 0.5 when $p_{JF} = 0.1, n \geq 14$ and $s = f$, which infers that most transactions cannot reach a consensus.

[4] This is different from the concept of link failure mentioned before, which is only temporarily invalid in one transaction consensus and may succeed in the next.

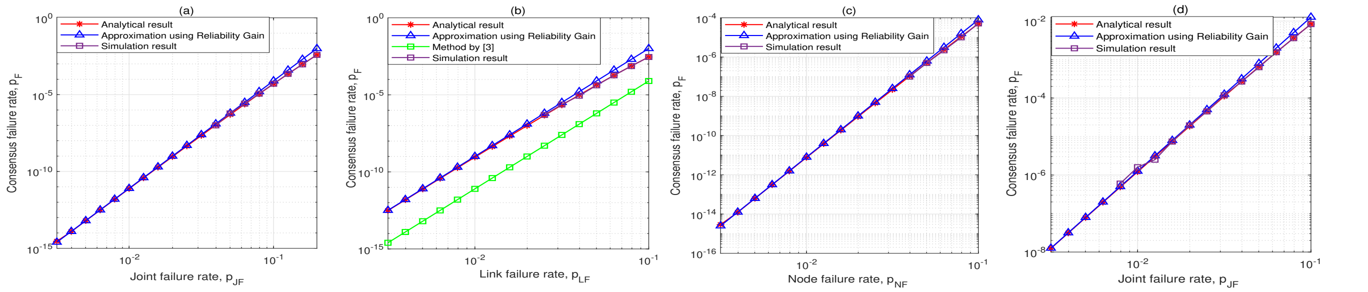

Fig. 3: Approximation of Reliability Gain when $n = 12$ of (a) considering both node failure and link failure, (b) no node failure ($p_{NF} = 0$), (c) no link failure ($p_{LF} = 0$), (d) the number of deterministic failures $s = 3$.

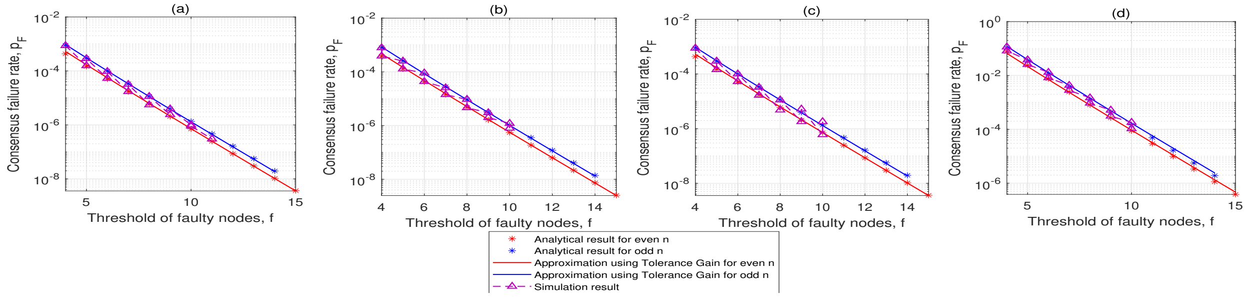

Fig. 4: Approximation of Tolerance Gain. The threshold of number of faulty nodes f versus to the consensus failure rate p_F when (a) $p_{JF} = 0.1$, (b) $p_{NF} = 0$ and $p_{LF} = 0.05$, (c) $p_{LF} = 0$ and $p_{NF} = 0.1$ (d) $p_{JF} = 0.1$ and $s = 3$.

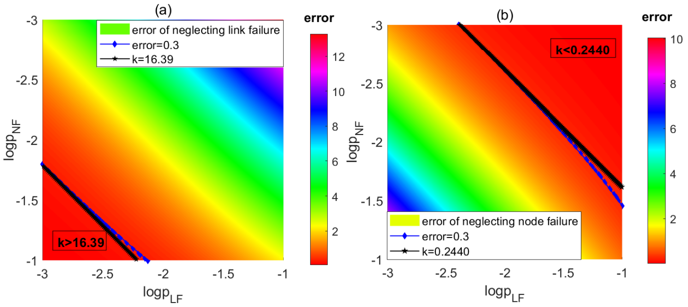

Fig. 5: For different link failure rate p_{LF} and node failure rate p_{NF}, error of neglecting (a) link failure; (b) node failure.

E. Consensus reliability for multiple instances

We consider consensus reliability of single instance in above. In this section, assuming different instances are independent, we show that the consensus reliability of consecutive multiple instances can be easily extended based on that of single instance.

Firstly, we introduce that link failures and node failures have different status as for multiple instances. Node failure can be considered as permanent (compared with the latency of consensus) while link failure is transient since re-transmission can be conducted. If the node fails, all the subsequent instances will be affected, thus node failure is more catastrophic in multiple instances. To obtain consensus reliability of consecutive instances, we can firstly calculate the conditional probability of consensus of multiple instances for a given number of non-faulty nodes, then use full-partition formula to obtain the overall consensus reliability. Formally, we can get the mean of consensus reliability of consecutive M instances p_C^{Mul} as

$$p_C^{Mul} = \sum_{x=n-f}^{n} \binom{n}{x} p_N^x (1-p_N)^{n-x} (p_C^*)^M \tag{27}$$

where p_C^* is the conditional probability of single instance consensus for a given number of non-faulty nodes, satisfying:

$$p_C^* = \sum_{y=n-f}^{x} \binom{x}{y} (p_L^2)^y (1-(p_L)^2)^{x-y} \tag{28}$$

Through the simplification methods of joint failure and power series proposed in section III-B and III-C, we get the approximated form of consensus failure rate p_F^{Mul}:

$$p_F^{Mul} \approx Mp_F - (M-1)\binom{n}{f+1} p_{NF}^{f+1}$$

(29)

where p_F is the consensus failure rate of single instance, which has been analyzed in detail before. See Appendix F for proof.

Thus through Eq. (IV-E) the consensus reliability of consecutive M instances can be obtained based on that of single instance. We can approximately evaluate that in how many instances, at least one consensus instance failure will occur with probability 1.

V. Discussion and Future exploration

In this section, three aspects are discussed, which could be further explored in the future.

1). Consensus reliability for other protocols: As for the log replication process or the process with the same function in other protocols, RAFT has the similar design with most of other CFT consensus such as ZAB [18] and Multi-Paxos: the leader sends messages in the downlink while followers give responses in the uplink, and the leader considers consensus achieved with more than 50% followers' feedback. Thus the analysis proposed in this paper can also be extended to most of other CFT protocols when only considering the log replication process. For BFT consensus protocols with more complex communication processes, the Markov property (first-order or high-order) can still be used to model and analyze the impacts of faulty nodes and links, since the state transition probability of the cluster due to the node or link failure at different stages still depends on the previous state. Additionally, the concept of Reliability Gain and Tolerance Gain might be obtained in other consensus protocols due to their physical meanings.

2). Latency, throughout and wireless communication optimization: We do not focus on the models of consensus performance in terms of time delay and throughput. However, the results derived in this paper can be served as a foundation for investigating the relationship between latency or throughput and consensus reliability. It is worth noting that RAFT does not allow holes in the log, (i.e., the consensus of the next instance can only be achieved once that of the previous one is achieved.) thus consensus failure of one instance might simultaneously increase the latency of this instance itself and the subsequent instances. In addition, in practical applications, different wireless channels may have different channel gains. In the condition of limited communication resources (bandwidth or power), the consensus reliability represented by Eq. (16) or (9) (or other indicators such as latency and throughput) can be further optimized through the reasonable allocation of communication resources.

3). Wireless consensus protocol design and supplement: Since we consider wireless link failures, the consensus protocol is not sufficient for this situation. For example, under the assumption of a synchronous or weak synchronization network with link failures, it is possible that the leader receives fewer than $n - f$ messages from followers thus causing consensus failure. Therefore, th exact recovery mechanism along with its impact and the synchronization method specified for followers who experienced link failures should be carefully considered.

In addition, we did not consider the impact of the follower-side commit operation. The downlink communication packet of the next transaction can be designed to contain the leader's commit instruction to the previous transaction to carry out the commit phase in a more effective way. By this way, the nodes that were not synchronized due to the wireless link failures in the last transaction can also directly replicate the state and submit it according to the commit command.

As for the possible influence of wireless scenarios on the leader, if the consensus process fails once or more times on one leader, it is very likely that the wireless communication resources possessed by the leader are scarce, or the wireless communication environment becomes hazardous due to the dynamic change of the network. Therefore, when electing or changing the leader, the wireless link reliability between the leader and followers and the corresponding consensus reliability should be considered as the indicators to minimize the delays and maximize the throughput.

VI. Simulation

The consensus reliability are analyzed mathematically in previous sections. Since the theoretical model is for the log replication, we demonstrate simulations of log replication by using the Monte Carlo method in this section, not only verifying the correctness of the theoretical expressions but also visualizing the characteristics of the relations and the results.

A. p.d.f. of the logarithmic consensus failure rate

To simplify the calculation process, the p.d.f. of the logarithmic consensus failure rate H, i.e. Eq. (12)-(15), is carried out assuming that the mean value of the reliability of different nodes are identical and the mean value of the reliability of different uplink and downlink are identical. Node reliability P_i^N of each node, uplink reliability P_i^{UL} and downlink reliability P_i^{DL} of each link are assumed to follow particular distributions to demonstrate their influences on the p.d.f. of H with the constant number of followers $n = 10$. The p.d.f. of node reliability and link reliability in three scenarios are shown in Fig. 2(a)-(c), which are uniform distribution, truncated Gaussian distribution, exponential distribution, respectively. Their corresponding H are shown in Fig. 2(d)-(f). The approximated results of power series for retaining the first term (Eq. (11)) are sufficiently close to the analytical results. Despite the different p.d.f. of P_i^N, P_i^{DL} and P_i^{UL}, the numerical results of the p.d.f. of logarithmic consensus failure rate, H, are always approximate to Gaussian distribution.

B. Reliability Gain and Tolerance Gain

Reliability Gain and Tolerance Gain clearly show the linear relationship between consensus failure rate and its two factors, p_{JF} and f. We illustrate the results of consensus from Reliability Gain and Tolerance Gain in Fig. 3 and Fig. 4 respectively to verify the linearity. $n = 12, f = 6$ is used as an example in Fig. 3 while $p_{JF} = 0.1$ is used as an example in Fig. 4. The two figures exemplify four scenarios that includes the general node and link probabilistic failure model, the scenario in which the node failure rate p_{NF} is 0, the scenario in which the link failure rate p_{LF} is 0, and scenario in which the number of already deterministic failures s is 3.

The results from Reliability Gain in Fig. 3 indicates that the accuracy increases along with the drop of p_{JF}, and the deviation is almost negligible from p_{JF} is less than 0.1. The analytical result of Reliability Gain can match the simulation result, which shows that Reliability Gain is valid and accurate. Fig. 3(a) reveals that when p_{JF} is 0.1, the consensus failure rate p_F is 0.0001, which means the consensus reliability can still reach perfect even with imperfect node and link reliability. Additionally, the approximation results from Reliability Gain proposed in [3] is indicated as a green line in Fig. 3 (b), which clearly indicates that the accuracy of Reliability Gain proposed in this paper is higher than that in [3]. Note that the difference

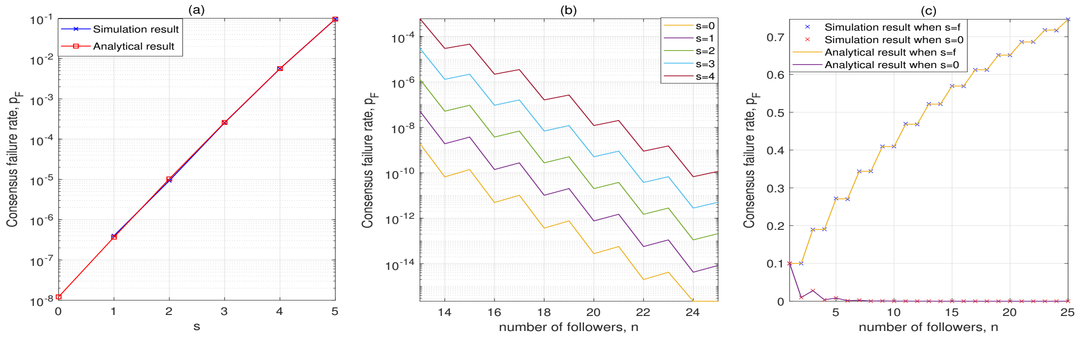

Fig. 6: Consensus failure rate p_F with already deterministic s failures of (a) $n = 10$ and $p_{JF} = 0.02$, (b) different n with $s = 0, 1, 2, 3, 4$ and $p_{JF} = 0.02$, (c) different n with $s = 0, f$ and $p_{JF} = 0.1$

between our approximation and the approximation in [3] is nearly 7log2, which approaches the difference between our expression of Reliability Gain and their expression, i.e. $(f+1)\log(1-(1-p_{LF})^2)-(f+1)\log(p_{LF}) \approx (f+1)\log 2$.

Theorem 2 demonstrates the linearity of Tolerance Gain when n is odd or even. This feature is evident in Fig. 4 as well. In terms of Tolerance Gain performance in Fig. 4, results from Tolerance Gain overlap with the original analytical results and the simulation results.

C. Dominant analysis

The simulation results of dominant analysis when $n = 11$ are demonstrated in Fig. 5 and matches Remark 3. In Fig. 5(a), it can be seen that in the area of $\log k = \log p_{NF} - \log p_{LF} > 1.21$, which is equal to $k > 16.39$, the error of neglecting link failures ε_L is less than 0.3. We can conclude that when $k > 16.39$, node failures are considered dominant, and the link failures can be ignored with ε_L less than 0.3. Similarly, in Fig. 5(b), it can be seen that in the area of $\log k = \log p_{NF} - \log p_{LF} < -0.6126$, which is equal to $k < 0.2440$, the error of neglecting node failures ε_N is less than 0.3. Thus when $k < 0.2440$, node failures can be ignored with ε_N less than 0.3.

D. Consensus with already deterministic s failures

The simulation processes the RAFT protocol with a certain number of fault nodes to verify the linear relationship between $\log p_F$ and s,. In Fig. 6(a) and (b), when $p_{JF} = 0.02$, the linear relationship is indicated between $\log p_F$ and s. Note that p_F increases by 1.4 orders of magnitude with the increase of s in Fig. 6 (a) and (b), which obviously matches analysis in Sec. IV-D. Due to the large impact of a number of unavailable nodes on the consensus reliability, we explore the scenario with a large s. We demonstrate two extreme cases of s, one is $s = 0$, the other is $s = f$ when $p_{JF} = 0.1$ in Fig. 6 (c). Fig. 6 shows that excessive unavailable nodes can be devastating to system performance. Therefore, even though the distributed consensus can theoretically tolerate f failures, p_C with f nodes unavailable is much lower than that with zero unavailable nodes according to the results in Fig. 6 (c). Particularly, p_F is greater than 0.5 when $n \geq 14$ and $s = f$, which infers that most transactions cannot reach a consensus and it is disastrous to the system.

VII. Conclusion

To evaluate the consensus of the RAFT system, a Node and Link probabilistic failure model is built, and the methods of joint failure and power series are presented to simplify the calculation process. The p.d.f. of the consensus reliability of the RAFT system is derived. Additionally, two reliability indicators, Reliability Gain and Tolerance Gain are defined and presented with general and accurate analytical expressions, which clearly indicate the linearity of the joint reliability and threshold of faulty nodes with the mean value of consensus reliability. Finally, the special case of a network with already s failure and its adverse impact are evaluated. Simulations are provided based on the model to validate the theoretical analysis. Simulations show that despite the different p.d.f. of node reliability and link reliability, the p.d.f. of logarithmic consensus failure rate is always approximate to Gaussian distribution. The linear relations of Reliability Gain and Tolerance Gain are effectively validated, which both provide benchmark guidelines for designing or evaluating distributed consensus IoT systems that meet specified consensus reliability. Similar approaches can be used to analyze IoT systems with other consensus mechanisms for future research.

Acknowledgement

The work is partially supported by Glasgow College Student Science and Technology Innovation Fund.

Appendix A
identity between Eq. (8) and Eq. (1), (4) - (7)

Change the summation area of Eq. (7), we have:

$$P_C = \sum_{n \geq z \geq n-f, \Omega \supseteq S_{3,z}} \sum_{n \geq y \geq z, \Omega \supseteq S_{2,y} \supseteq S_{3,z}} \sum_{n \geq x \geq y, \Omega \supseteq S_{1,x} \supseteq S_{2,y}} P\left(S_{1,x}, S_{2,y}, S_{3,z}\right) \tag{30}$$

Substitute Eq. (1) and (4) into Eq. (30), we can get:

$$\begin{aligned} P_C = & \sum_{n \geq z \geq n-f, \Omega \supseteq S_{3,z}} \sum_{n \geq y \geq z, \Omega \supseteq S_{2,y} \supseteq S_{3,z}} P\left(S_{3,z} | S_{2,y}\right) \sum_{n \geq x \geq y, \Omega \supseteq S_{1,x} \supseteq S_{2,y}} P\left(S_{1,x}\right) P\left(S_{2,y} |, S_{1,x}\right) \\ = & \sum_{n \geq z \geq n-f, \Omega \supseteq S_{3,z}} \sum_{n \geq y \geq z, \Omega \supseteq S_{2,y} \supseteq S_{3,z}} P\left(S_{3,z} | S_{2,y}\right) \sum_{n \geq x \geq y, \Omega \supseteq S_{1,x} \supseteq S_{2,y}} \prod_{u \in S_{1,x}} P_u^N \prod_{v \in \complement_\Omega S_{1,x}} (1 - P_v^N) \prod_{s \in S_{2,y}} P_s^{DL} \prod_{t \in \complement_{S_{1,x}} S_{2,y}} (1 - P_t^{DL}) \end{aligned} \tag{31}$$

Now consider the last summation, since $S_{1,x} = S_{2,y} \cup \complement_{S_{1,x}} S_{2,y}$ and $S_{2,y} \cap \complement_{S_{1,x}} S_{2,y} = \emptyset$, Eq. (31) can be transformed as:

$$\begin{aligned} & \sum_{n \geq x \geq y, \Omega \supseteq S_{1,x} \supseteq S_{2,y}} \prod_{u \in S_{1,x}} P_u^N \prod_{v \in \complement_\Omega S_{1,x}} (1 - P_v^N) \prod_{s \in S_{2,y}} P_s^{DL} \prod_{t \in \complement_{S_{1,x}} S_{2,y}} (1 - P_t^{DL}) \\ = & \sum_{n \geq x \geq y, \Omega \supseteq S_{1,x} \supseteq S_{2,y}} \prod_{v \in \complement_\Omega S_{1,x}} (1 - P_v^N) \prod_{s \in S_{2,y}} P_s^N P_s^{DL} \prod_{t \in \complement_{S_{1,x}} S_{2,y}} P_t^N (1 - P_t^{DL}) \\ = & \prod_{s \in S_{2,y}} P_s^N P_s^{DL} (\sum_{n \geq x \geq y, \Omega \supseteq S_{1,x} \supseteq S_{2,y}} \prod_{v \in \complement_\Omega S_{1,x}} (1 - P_v^N) \prod_{t \in \complement_{S_{1,x}} S_{2,y}} P_t^N (1 - P_t^{DL})) \end{aligned} \tag{32}$$

Since $\complement_\Omega S_{2,y} = \complement_\Omega S_{1,x} \cup \complement_{S_{1,x}} S_{2,y}$, $\complement_\Omega S_{1,x} \cap \complement_{S_{1,x}} S_{2,y} = \emptyset$ and the summation traversed all sets from Ω to $S_{2,y}$, so

$$\begin{aligned} & \prod_{s \in S_{2,y}} P_s^N P_s^{DL} (\sum_{n \geq x \geq y, \Omega \supseteq S_{1,x} \supseteq S_{2,y}} \prod_{v \in \complement_\Omega S_{1,x}} (1 - P_v^N) \prod_{t \in \complement_{S_{1,x}} S_{2,y}} P_t^N (1 - P_t^{DL})) \\ = & \prod_{s \in S_{2,y}} P_s^N P_s^{DL} \prod_{t \in \complement_\Omega S_{2,y}} (1 - P_t^N) + P_t^N (1 - P_t^{DL}) = \prod_{s \in S_{2,y}} P_s^N P_s^{DL} \prod_{t \in \complement_\Omega S_{2,y}} (1 - P_t^N P_t^{DL}) \end{aligned} \tag{33}$$

Similar with Eq. (32)-(33), P_C can be further transformed as:

$$\begin{aligned} P_C = & \sum_{n \geq z \geq n-f,\ \Omega \supseteq S_{3,z}} \sum_{n \geq y \geq z,\ \Omega \supseteq S_{2,y} \supseteq S_{3,z}} \prod_{u \in S_{3,z}} P_u^{UL} \prod_{v \in \complement_{S_{2,y}} S_{3,z}} \left(1 - P_v^{UL}\right) \prod_{s \in S_{2,y}} P_s^N P_s^{DL} \prod_{t \in \complement_\Omega S_{2,y}} \left(1 - P_t^N P_t^{DL}\right) \\ = & \sum_{n \geq z \geq n-f,\ \Omega \supseteq S_{3,z}} \sum_{n \geq y \geq z,\ \Omega \supseteq S_{2,y} \supseteq S_{3,z}} \prod_{u \in S_{3,z}} P_u^N P_u^{DL} P_u^{UL} \prod_{v \in \complement_{S_{2,y}} S_{3,z}} P_v^N P_v^{DL} \left(1 - P_v^{UL}\right) \prod_{t \in \complement_\Omega S_{2,y}} \left(1 - P_t^N P_t^{DL}\right) \\ = & \sum_{n \geq z \geq n-f,\ \Omega \supseteq S_{3,z}} \prod_{u \in S_{3,z}} P_u^N P_u^{DL} P_u^{UL} (\sum_{n \geq y \geq z,\ \Omega \supseteq S_{2,y} \supseteq S_{3,z}} \prod_{v \in \complement_{S_{2,y}} S_{3,z}} P_v^N P_v^{DL} \left(1 - P_v^{UL}\right) \prod_{t \in \complement_\Omega S_{2,y}} \left(1 - P_t^N P_t^{DL}\right)) \\ = & \sum_{n \geq z \geq n-f,\ \Omega \supseteq S_{3,z}} \prod_{u \in S_{3,z}} P_u^N P_u^{DL} P_u^{UL} \prod_{v \in \complement_\Omega S_{3,z}} P_v^N P_v^{DL} \left(1 - P_v^{UL}\right) + \left(1 - P_v^N P_v^{DL}\right) \\ = & \sum_{n \geq z \geq n-f,\ \Omega \supseteq S_{3,z}} \prod_{u \in S_{3,z}} P_u^N P_u^{DL} P_u^{UL} \prod_{v \in \complement_\Omega S_{3,z}} 1 - P_v^N P_v^{DL} P_v^{UL} \end{aligned} \tag{34}$$

Consider $P_u^N P_u^{DL} P_u^{UL}$ as the joint success rate P_u^J, Eq. (8) is identical to Eq. (1), (4) - (7).

Appendix B
Derivation of Power series of Eq. (10)

By observation of Eq. (9), each term after its full expansion is always the product of joint failure rate P_i^{JF}. In fact, we only care about how many joint failure rates are multiplied for each term of the expansion of Eq. (9), and the higher product is the higher power, while the lower product is the lower power. From this perspective, we might as well consider that the joint failure rate P_i^{JF} of each node is the same and denoted as p_{JF}, and the product of t joint failure rate P_i^{JF} at this time corresponds to $(p_{JF})^t$. For Eq. (9), $\prod_{u \in S_{J,k}} P_u^J$ will be transformed into $(1 - p_{JF})^k$,$\prod_{v \in \complement_\Omega S_{J,k}} 1 - P_v^J$ becomes $(p_{JF})^{n-k}$, thus Eq. (9) can be transformed as:

$$P_F = \sum_{x=f+1}^{n} \binom{n}{x} (1 - p_{JF})^{n-x} p_{JF}^x \tag{35}$$

By expanding $(1-p_{JF})^{n-x}$, we have

$$P_F = \sum_{x=f+1}^{n} \sum_{k=0}^{n-x} \binom{n}{x} \binom{n-x}{k} (-1)^k p_{JF}^{x+k} \tag{36}$$

Substituting $x+k$ as t, we have

$$P_F = \sum_{x=f+1}^{n} \sum_{t=x}^{n} \binom{n}{x} \binom{n-x}{t-x} (-1)^{t-x} p_{JF}^{t} \tag{37}$$

After changing the summation order,

$$P_F = \sum_{t=f+1}^{n} \sum_{x=f+1}^{t} \binom{n}{x} \binom{n-x}{t-x} (-1)^{t-x} p_{JF}^{t} \tag{38}$$

Replacing $\binom{n}{x}\binom{n-x}{t-x}$ as $\binom{n}{t}\binom{t}{x}$, P_F can be calculated as

$$P_F = \sum_{t=f+1}^{n} \binom{n}{t} p_{JF}^{t} (\sum_{x=f+1}^{t} \binom{t}{x} (-1)^{t-x}) \tag{39}$$

Consider $\sum\limits_{x=f+1}^{t} \binom{t}{x}(-1)^{t-x}$, let $m = t-x$, we have

$$\begin{aligned} &\sum_{x=f+1}^{t} \binom{t}{x} (-1)^{t-x} \\ &= \sum_{m=0}^{t-f-1} \binom{t}{t-m} (-1)^m = \sum_{m=0}^{t-f-1} \binom{t}{m} (-1)^m \end{aligned} \tag{40}$$

Through $\binom{t}{m} = \binom{t-1}{m} + \binom{t-1}{m-1}$, it can be further transformed as:

$$\begin{aligned} \sum_{m=0}^{t-f-1} \binom{t}{m} (-1)^m &= \sum_{m=0}^{t-f-1} (\binom{t-1}{m} + \binom{t-1}{m-1}) (-1)^m \\ &= \binom{t}{0} (-1)^0 + \sum_{m=1}^{t-f-1} (\binom{t-1}{m} + \binom{t-1}{m-1}) (-1)^m \\ &= \sum_{m=0}^{t-f-1} \binom{t-1}{m} (-1)^m + \sum_{r=0}^{t-f-2} \binom{t-1}{r} (-1)^{r+1} \\ &= \binom{t-1}{t-f-1} (-1)^{t-f-1} = \binom{t-1}{f} (-1)^{t-f+1} \end{aligned} \tag{41}$$

Thus P_F with respect to p_{JF} can be written as:

$$P_F = \sum_{t=f+1}^{n} (-1)^{t-f+1} \binom{t-1}{f} \binom{n}{t} p_{JF}^{t} \tag{42}$$

Note here p_{JF}^t corresponds to some t products of joint failure rate P_i^{JF}. In fact, since the summation in Eq. (9) is rotationally symmetric to the nodes, every possible combination of t products of P_i^{JF} will appear and with the same coefficients, so $\binom{n}{t} p_{JF}^t$ can be converted to $\sum_{\Omega \supseteq S_{JF,t}} \prod_{u \in S_{JF,t}} P_u^{JF}$, so we get:

$$P_F = \sum_{t=f+1}^{n} (-1)^{t-f+1} \binom{t-1}{f} \sum_{\Omega \supseteq S_{JF,t}} \prod_{u \in S_{JF,t}} P_u^{JF} \tag{43}$$

Appendix C
Proof of Theorem 1

According to Eq. (18), the consensus failure rate can be expressed as:

$$p_F = \sum_{x=f+1}^{n} \binom{n}{x} p_{JF}^{x} (1-p_{JF})^{n-x} \tag{44}$$

Thus the error term could be obtained as:

$$\begin{aligned} \varepsilon &= \log(\frac{\sum\limits_{x=f+1}^{n} \binom{n}{x} p_{JF}^{x} (1-p_{JF})^{n-x}}{\binom{n}{f+1} p_{JF}^{f+1}}) \\ &= \log(\frac{(1-p_{JF})^{n-f-1} \sum\limits_{x=f+1}^{n} \binom{n}{x} p_{JF}^{x} (1-p_{JF})^{n-x}}{\binom{n}{f+1} p_{JF}^{f+1} (1-p_{JF})^{n-f-1}}) \\ &\leq \log(\frac{(1-p_{JF})^{n-f-1} \binom{n}{f+1} \sum\limits_{x=f+1}^{n} p_{JF}^{x} (1-p_{JF})^{n-x}}{\binom{n}{f+1} p_{JF}^{f+1} (1-p_{JF})^{n-f-1}}) \\ &= \log((1-p_{JF})^{n-f-1} (1 + \frac{p_{JF}}{1-p_{JF}} + ... + (\frac{p_{JF}}{1-p_{JF}})^{n-f-1})) \\ &= \log(\frac{(1-p_{JF})^{n-f} - p_{JF}^{n-f}}{(1-p_{JF}) - p_{JF}}) \end{aligned} \tag{45}$$

Here $\binom{n}{x} \leq \binom{n}{f+1}$when $x \geq f+1$ is used to obtain the upper bound of ε. Thus Theorem 1 has been proved.

Appendix D
Proof of Theorem 2

With the assumptions in Section IV, the degeneration form of Eq. (9) can be written as:

$$\begin{aligned} p_F &= \sum_{x=f+1}^{n} \binom{n}{x} p_{JF}^{x} (1-p_{JF})^{n-x} \\ &\approx \binom{n}{f+1} p_{JF}^{f+1} (1-p_{JF})^{n-f-1} \end{aligned} \tag{46}$$

Similar wiht Eq. (45) the approximation error term could be obtained as:

$$\begin{aligned} \varepsilon &= \log(\frac{\sum\limits_{x=f+1}^{n} \binom{n}{x} p_{JF}^{x} (1-p_{JF})^{n-x}}{\binom{n}{f+1} p_{JF}^{f+1} (1-p_{JF})^{n-f-1}}) \\ &\leq \log(\frac{1}{(1-p_{JF})^{n-f-1}} \frac{(1-p_{JF})^{n-f} - p_{JF}^{n-f}}{(1-p_{JF}) - p_{JF}}) \\ &= \log(\frac{(1-p_{JF})^{n-f} - p_{JF}^{n-f}}{(1-p_{JF})^{n-f} - p_{JF}(1-p_{JF})^{n-f-1}}) \end{aligned} \tag{47}$$

If the number of followers is even, i.e. $n = 2f$,

$$\begin{aligned} \log p_F =& \log p_{JF} \cdot (f+1) + \log(1-p_{JF}) \cdot (f-1) \\ &+ \log(\frac{(2f)!}{(f+1)!(f-1)!}) \end{aligned} \tag{48}$$

According to the Stirling formula, $n! \approx \sqrt{2\pi n}(\frac{n}{e})^n$, Eq. (48) can be transformed into

$$\begin{aligned} \log p_F \approx (f+1)(\log p_{JF}) + (f-1)\log(1-p_{JF}) + \log(\frac{1}{\sqrt{2\pi}}) \\ + \log(\sqrt{\frac{2f}{f^2-1}}) + \log(\frac{(2f)^{2f}}{(f+1)^{f+1}(f-1)^{f-1}}) \end{aligned} \tag{49}$$

The fifth term in logarithm form can be simplified as

$$\begin{aligned} &\log(\frac{(2f)^{2f}}{(f+1)^{f+1}(f-1)^{f-1}}) = (f+1)\log(\frac{2f}{f+1}) \\ &+ (f-1)\log(\frac{2f}{f-1}) \approx 2f\log(2) \end{aligned} \tag{50}$$

By using Eq. (50) and arranging Eq. (49), we get an equation consisting of linear and nonlinear parts as

$$\begin{aligned} \log p_F \approx (2\log 2 + \log p_{JF} + \log p_J) \cdot f \\ + \log(\frac{p_{JF}}{(p_J)\sqrt(\pi)}) - \frac{1}{2}\log f \end{aligned} \tag{51}$$

It is obvious that $(2\log 2+\log p_{JF}+\log p_J)\cdot f+\log(\frac{p_{JF}}{(p_J)\sqrt(\pi)})$ is the linear part of Eq. (50) while $-\frac{1}{2}\log f$ is the non-linear part. The derivative of $\log p_F$ with respect to f can be calculated as: $2\log 2 + \log p_{JF} + \log p_J - \frac{1}{2f}$. Since p_{JF} is close to 0, the impact of $-\frac{1}{2f}$ on the linearity is minor. Thus the tolerance gain when n is even is $2\log 2+\log p_{JF}+\log p_J$ and the linearity will be more obvious when p_{JF} is smaller and f is larger. The non-linear part $-\frac{1}{2}\log f$ is considered as the complement term to increase the approximation accuracy when f is relatively large.

If $n = 2f + 1$, we have similar proof process. After transformation and Stirling approximation,

$$\begin{aligned} \log p_F \approx (\log p_{JF}) \cdot (f+1) + (\log(1-p_{JF})) \cdot f + \log(\frac{1}{\sqrt{2\pi}}) \\ + \log(\sqrt{\frac{2f+1}{f(f+1)}}) + \log(\frac{(2f+1)^{2f+1}}{(f+1)^{f+1}(f)^f}) \\ \approx (2\log 2 + \log p_{JF} + \log p_J)f + \log(\frac{p_{JF}}{p_J\sqrt{\pi}}) \\ + \log 2p_J - \frac{1}{2}\log f \end{aligned} \tag{52}$$

Thus, when n is odd, the linear relation can be obtained as equation (24).

Appendix E
Proof of Remark 3

According to the definitions of ε_L and ε_N in section IV-C, they can be calculated as:

$$\begin{aligned} \varepsilon_L = \log\frac{p_F}{p_{NLF}} \approx (f+1)\log(\frac{1-(1-p_{NF})(1-p_{LF})^2}{p_{NF}}) \\ \approx (f+1)\log(\frac{p_{NF}+2p_{LF}}{p_{NF}}) = (f+1)\log(1+\frac{2}{k}) \end{aligned} \tag{53}$$

$$\begin{aligned} \varepsilon_N = \log\frac{p_F}{p_{NNF}} \approx (f+1)\log(\frac{1-(1-p_{NF})(1-p_{LF})^2}{1-(1-p_{LF})^2}) \\ \approx (f+1)\log(\frac{p_{NF}+2p_{LF}}{2p_{LF}}) = (f+1)\log(1+\frac{k}{2}) \end{aligned} \tag{54}$$

According to Eq. (53), k can be represented as $k = \frac{2}{10^{\frac{\varepsilon_L}{f+1}}-1}$. Since Eq. (53) is a decreasing function about k, for a given ε_L, if k is more than $\frac{2}{10^{\frac{\varepsilon_L}{f+1}}-1}$, the neglecting error will be less than ε_L. The situation of neglecting node failure is the same with that of neglecting link failure.

Appendix F
Proof of Eq. (IV-E)

Since $1 - p_C^*$ is close to 0, we have:

$$(p_C^*)^M = (1-(1-p_C^*))^M \approx Mp_C^* - (M-1) \tag{55}$$

According to Eq. (55), (28) and the simplification method of joint failure, Eq. (28) can be further transformed as

$$\begin{aligned} p_C^{Mul} &\approx M\sum_{x=n-f}^{n}\binom{n}{x}p_N^x(1-p_N)^{n-x}p_C^* \\ &- (M-1)\sum_{x=n-f}^{n}\binom{n}{x}p_N^x(1-p_N)^{n-x} \\ &= Mp_C - (M-1)\sum_{x=n-f}^{n}\binom{n}{x}p_N^x(1-p_N)^{n-x} \end{aligned} \tag{56}$$

where p_C is the consensus reliability of a single instance. Thus $p_F^{Mul} = 1 - p_C^{Mul}$ can be further transformed based on the method of power series proposed in section III-C as:

$$\begin{aligned} p_F^{Mul} &\approx 1 - Mp_C + (M-1)\sum_{x=n-f}^{n}\binom{n}{x}(p_N^x(1-p_N)^{n-x} \\ &= M(1-p_C) - (M-1)(1-\sum_{x=n-f}^{n}\binom{n}{x}(p_N^x(1-p_N)^{n-x}) \\ &\approx Mp_F - (M-1)\binom{n}{f+1}p_{NF}^{f+1} \end{aligned} \tag{57}$$

Thus we get Eq. (IV-E).

References

[1] Waleed Ejaz and Alagan Anpalagan. Dimension reduction for big data analytics in internet of things. In *Internet of Things for Smart Cities*, pages 31–37. Springer, 2019.

[2] Yao Sun, Lei Zhang, Gang Feng, Bowen Yang, Bin Cao, and Muhammad Ali Imran. Blockchain-enabled wireless internet of things: Performance analysis and optimal communication node deployment. *IEEE internet of things journal*, 6(3):5791–5802, 2019.

[3] Dachao Yu, Wenyu Li, Hao Xu, and Lei Zhang. Low reliable and low latency communications for mission critical distributed industrial internet of things. *IEEE Communications Letters*, 25(1):313–317, 2021.

[4] Ajay D Kshemkalyani and Mukesh Singhal. *Distributed Computing: Principles, Algorithms, and Systems.* Cambridge University Press, Cambridge, 2008.

[5] Wenchao Xia, Tony Q. S Quek, Kun Guo, Wanli Wen, Howard H Yang, and Hongbo Zhu. Multi-armed bandit-based client scheduling for federated learning. *IEEE transactions on wireless communications*, 19(11):7108–7123, 2020.

[6] Elad Elrom. *The Blockchain Developer: A Practical Guide for Designing, Implementing, Publishing, Testing, and Securing Distributed Blockchain-Based Projects*. Apress L. P, Berkeley, CA, 2019.

[7] Admar Ajith Kumar Somappa, Knut Øvsthus, and Lars M Kristensen. An industrial perspective on wireless sensor networks - a survey of requirements, protocols, and challenges. *IEEE Communications surveys and tutorials*, 16(3):1391–1412, 2014.

[8] Leslie Lamport. Time, clocks, and the ordering of events in a distributed system. 21(7):558–565, 1978.

[9] Heidi Howard. Distributed consensus revised (doctoral thesis), 2019.

[10] L. Lamport. Proving the correctness of multiprocess programs. *IEEE Transactions on Software Engineering*, SE-3(2):125–143, 1977.

[11] Miguel Castro, Barbara Liskov, et al. Practical byzantine fault tolerance. In *OSDI*, volume 99, pages 173–186, 1999.

[12] Maofan Yin, Dahlia Malkhi, Michael K Reiter, Guy Golan Gueta, and Ittai Abraham. Hotstuff: Bft consensus with linearity and responsiveness. In *Proceedings of the 2019 ACM Symposium on Principles of Distributed Computing*, pages 347–356, 2019.

[13] Leslie Lamport. The part-time parliament. *ACM transactions on computer systems*, 16(2):133–169, 1998.

[14] Diego Ongaro and John Ousterhout. In search of an understandable consensus algorithm. In *2014 USENIX Annual Technical Conference (USENIX ATC 14)*, pages 305–319, Philadelphia, PA, June 2014. USENIX Association.

[15] Leslie Lamport et al. Paxos made simple. *ACM Sigact News*, 32(4):18–25, 2001.

[16] Butler Lampson. The abcd's of paxos. In *Proceedings of the twentieth annual ACM symposium on principles of distributed computing*, PODC '01, page 13. ACM, 2001.

[17] Brian Oki and Barbara Liskov. Viewstamped replication: A new primary copy method to support highly-available distributed systems. In *Proceedings of the seventh annual ACM Symposium on principles of distributed computing*, PODC '88, pages 8–17. ACM, 1988.

[18] F. P Junqueira, B. C Reed, and M Serafini. Zab: High-performance broadcast for primary-backup systems. In *2011 IEEE/IFIP 41st International Conference on Dependable & Systems Networks (DSN)*, pages 245–256. IEEE, 2011.

[19] Heidi Howard, Malte Schwarzkopf, Anil Madhavapeddy, and Jon Crowcroft. Raft refloated: Do we have consensus? *Operating systems review*, 49(1):12–21, 2015.

[20] Sebastian Pedersen, Hein Meling, and Leander Jehl. An analysis of quorum-based abstractions: A case study using gorums to implement raft. In *Proceedings of the 2018 Workshop on advanced tools, programming languages, and platforms for implementing and evaluating algorithms for distributed systems*, ApPLIED '18, pages 29–35. ACM, 2018.

[21] Ermin Sakic and Wolfgang Kellerer. Response time and availability study of raft consensus in distributed sdn control plane. *IEEE eTransactions on network and service management*, 15(1):304–318, 2018.

[22] Junyi Xu, Xiaohui Yuan, Zhenchun Wei, Jianghong Han, Lei Shi, and Zengwei Lyu. A wireless sensor network recharging strategy by balancing lifespan of sensor nodes. In *2017 IEEE Wireless Communications and Networking Conference (WCNC)*, pages 1–6. IEEE, 2017.

[23] Joseph Habiyaremye, Marco Zennaro, Chomora Mikeka, Emmanuel Masabo, Santhi Kumaran, and Kayalvizhi Jayavel. Gprs sensor node battery life span prediction based on received signal quality: Experimental study. *Information (Basel)*, 11(524):524, 2020.

[24] Alberto Signori, Federico Chiariotti, Filippo Campagnaro, and Michele Zorzi. A game-theoretic and experimental analysis of energy-depleting underwater jamming attacks. *IEEE internet of things journal*, 7(10):9793–9804, 2020.

[25] Hyowoon Seo, Jihong Park, Mehdi Bennis, and Wan Choi. Communication and consensus co-design for distributed, low-latency, and reliable wireless systems. *IEEE internet of things journal*, 8(1):129–143, 2021.

[26] Hao Xu, Lei Zhang, Yinuo Liu, and Bin Cao. Raft based wireless blockchain networks in the presence of malicious jamming. *IEEE wireless communications letters*, 9(6):817–821, 2020.

[27] B.R Vojcic and R.L Pickholtz. Performance of direct sequence spread spectrum in a fading dispersive channel with jamming. *IEEE journal on selected areas in communications*, 7(4):561–568, 1989.

[28] K.C Teh, A.C Kot, and K.H Li. Multitone jamming rejection of ffh/bfsk spread-spectrum system over fading channels. *IEEE transactions on communications*, 46(8):1050–1057, 1998.

[29] Lucianna Kiffer, Dave Levin, and Alan Mislove. Stick a fork in it: Analyzing the ethereum network partition. In *Proceedings of the 16th ACM Workshop on Hot Topics in Networks*, HotNets-XVI, page 94?100, New York, NY, USA, 2017. Association for Computing Machinery.

[30] Mohammed Alfatafta, Basil Alkhatib, Ahmed Alquraan, and Samer Al-Kiswany. Toward a generic fault tolerance technique for partial network partitioning. In *14th USENIX Symposium on Operating Systems Design and Implementation (OSDI 20)*, pages 351–368. USENIX Association, November 2020.

[31] David Peleg and Avishai Wool. The availability of quorum systems. *Inf. Comput.*, 123:210–223, 1995.

[32] Dahlia Malkhi, Michael K. Reiter, and Avishai Wool. The load and availability of byzantine quorum systems. *SIAM Journal on Computing*, 29(6):1889–1906, 2000.

[33] Alia Asheralieva and Dusit Niyato. Reputation-based coalition formation for secure self-organized and scalable sharding in iot blockchains with mobile-edge computing. *IEEE internet of things journal*, 7(12):11830–11850, 2020.

[34] Hyowoon Seo, Jihong Park, Mehdi Bennis, and Wan Choi. Consensus-before-talk: Distributed dynamic spectrum access via distributed spectrum ledger technology. In *2018 IEEE International Symposium on Dynamic Spectrum Access Networks (DySPAN)*, pages 1–7. IEEE, 2018.

[35] Hyesung Kim, Jihong Park, Mehdi Bennis, and Seong-Lyun Kim. Blockchained on-device federated learning. *IEEE communications letters*, 24(6):1279–1283, 2020.

[36] Yinqiu Liu, Kun Wang, Yun Lin, and Wenyao Xu. mathsf: A lightweight blockchain system for industrial internet of things. *IEEE transactions on industrial informatics*, 15(6):3571–3581, 2019.

[37] Wenbo Wang, Dinh Thai Hoang, Peizhao Hu, Zehui Xiong, Dusit Niyato, Ping Wang, Yonggang Wen, and Dong In Kim. A survey on consensus mechanisms and mining strategy management in blockchain networks. *IEEE access*, 7:22328–22370, 2019.

[38] Lei Zhang, Hao Xu, Oluwakayode Onireti, Muhammad Ali Imran, and Bin Cao. How much communication resource is needed to run a wireless blockchain network? *IEEE network*, pages 1–8, 2021.

[39] Jintian Fu, Lupeng Zhang, Leixin Wang, and Fengqi Li. Bct: An efficient and fault tolerance blockchain consensus transform mechanism for iot. *IEEE Internet of Things Journal*, pages 1–1, 2021.

[40] Zhiming Liu, Lu Hou, Kan Zheng, Qihao Zhou, and Shiwen Mao. A dqn-based consensus mechanism for blockchain in iot networks. *IEEE Internet of Things Journal*, pages 1–1, 2021.

[41] Xiaojun Xu, Lu Hou, Yankai Li, and Yunxin Geng. Weighted raft: An improved blockchain consensus mechanism for internet of things application. In *2021 7th International Conference on Computer and Communications (ICCC)*, pages 1520–1525, 2021.

[42] Lu Hou, Xiaojun Xu, Kan Zheng, and Xianbin Wang. An intelligent transaction migration scheme for raft-based private blockchain in internet of things applications. *IEEE Communications Letters*, 25(8):2753–2757, 2021.

[43] Hao Xu, Yixuan Fan, Wenyu Li, and Lei Zhang. Wireless distributed consensus for connected autonomous systems.

[44] Charles J Colbourn. Combinatorial aspects of network reliability. *Annals of Operations Research*, 33(1):1–15, 1991.

[45] H Frank and I Frisch. Analysis and design of survivable networks. *IEEE transactions on communication technology*, 18(5):501–519, 1970.

[46] Li Juan Qiu, Cheng Ming Jin, and Ming Ma. Research on modeling of sensor nodes reliability. *Applied mechanics and materials*, 203:247–251, 2012.

[47] Ethan Heilman, Alison Kendler, Aviv Zohar, and Sharon Goldberg. Eclipse attacks on bitcoin's peer-to-peer network. In *Proceedings of the 24th USENIX Conference on Security Symposium*, SEC'15, page 129?144, USA, 2015. USENIX Association.