# Wireless Distributed Consensus for Connected Autonomous Systems

Hao Xu, Yixuan Fan, Wenyu Li and Lei Zhang

*Abstract*—**Connected critical autonomous systems (C-CAS) are envisioned to significantly change our life and work styles through emerging vertical applications such as autonomous vehicles and cooperative robots. However, as the scale of the connected nodes continues to grow, their heterogeneity and cyber-security threats are more eminent, and conventional centralized communications and decision-making methodology are reaching their limit. This paper is the first exploration of a trustworthy and fault-tolerant framework for C-CAS for achieving hyper-reliable global decision-making in a trustless environment, where the connected sensors/nodes are less reliable due to either communication failure or local decision error (e.g., by sensing algorithm/AI, etc.). The proposed framework is based on two iconic distributed consensus (DC) mechanisms, practical Byzantine fault tolerance (PBFT) and Raft, under the proposed PICA (Perception-Initiative-Consensus-Action) protocol with wireless connections among the nodes. We first analytically derived consensus reliability in six different system models. The other fundamental performance metrics such as the consensus throughput and latency, node scalability and reliability gain are also analytically derived. These analytical results provide basic design guidelines for wireless Distributed Consensus (WDC) usage in the C-CAS systems. The results show that WDC significantly improves overall system reliability with the increasing number of participating nodes.**

*Index Terms*—**PBFT, Raft, Byzantine fault tolerance, Wireless distributed consensus, Autonomous system, PICA, Autonomous driving.**

## I. Introduction

Driven by advances in 5G, industry 4.0, cloud/edge computing and artificial intelligence, etc., the Internet of Things (IoT) is extending from home and work environments to critical and complex industrial systems, such as transportation, healthcare, utilities, communications and e-commerce sectors [1] [2] [3]. These vital societal and industrial functions are increasingly interconnected for information exchange through communication networks to complete joint tasks [4] [5] [6]. In such systems, data may be collected from distributed and heterogeneous and trustless sensors located in different places to determine common and critical real-time decisions in order to achieve cooperative actions. Information reliability is of paramount concern since a failure can result in prohibitive costs, such as life loss or natural environment damage [1] [7] [8]. For instance, Connected and Autonomous Vehicles (CAV) with high-level autonomy (L4 and L5 autonomous driving capability) can drive cooperatively for collision avoidance by communicating with nearby vehicles or Road Side Units (RSU) [9]. Based on its locally equipped sensors (e.g., Lidar, mmWave Radar, etc.) and computing resource, the CAV is capable of making an initial decision. However, this decision needs to be informed and agreed upon (and recorded for compliance) by other vehicles in proximity in a real-time manner in order to avoid conflicting local actions between different CAVs. Thus, communications among the vehicles play a key role in such systems, and it should be ultra-reliable with a stringent latency constraint, as any misalignment among the vehicles can cause a disaster.

### A. Background

Centralized communication and decision approach is normally deployed in industry sectors, especially in mobile environments, which requires the connected nodes to transmit their data to a central control station, where critical decisions are made and sent back to nodes for actions, which is named as Perception-Collection-Decision-Action (PCDA) scheme. A representative one is the cloud/edge computing through ultra-reliable ($\geq$99.999%), and low latency ($\leq$ 1ms) communications (URLLC) provided by a cellular base station [10] [11], which is a key feature of 5G networks. However, as the number of connected devices/nodes continues to grow, their heterogeneity and cyber security threats are more significant, and the centralized approaches to these systems that are in use today are reaching their limits [12]. For instance, centralization may suffer from the single point of failure issues as well as cyber security attacks, particularly vulnerable sensors running in open environments. Moreover, in centralized systems, the nodes can only synchronize the information with the central node, and the performance in terms of critical joint decisions for the whole system (e.g., URLLC) can be limited by the worst node connected to the central station. Finally, a centralized communication system can be very costly since it is well-known that high communication reliability is contradictory to low time latency for given spectrum resources [10] [13] [14]. The cost can be unaffordable when the network scales up, e.g., on a busy road in autonomous driving scenarios or in a smart factory with numerous mobile robots.

Recognizing that many new generation mobile applications are discretely distributed in their topology [15], one promising solution for achieving low latency and ultra-reliable joint decisions is to utilize Distributed Consensus (DC) mechanisms (also known as DC algorithms or protocols) [16] [17] [18]. DC protocols are a procedure that only relies on passing messages to reach a common agreement among nodes in a distributed system without a central coordinator [12], and they ensure

All work done at University of Glasgow, Hao Xu and Wenyu Li are with Huawei Technologies, UK; E-mail: hxgla@outlook.com, wyli608@163.com; Yixuan Fan, and Lei Zhang are with the James Watt School of Engineering, University of Glasgow, Glasgow, G12 8QQ, UK; E-mail: y.fan.3@research.gla.ac.uk, Lei.Zhang@glasgow.ac.uk.

consistency of records and integrity of transactions among distributed nodes [19]. Blockchain is a typical DC system, and the proof-based DC protocols, e.g., Proof of Work, is its representative protocol. In addition to blockchain, a novel example of utilizing DC can be vehicles making collective decisions by DC on traffic events with relaxed communication link or node reliability due to the redundancy design in DC mechanisms.

Unlike the PCDA in centralized systems, where information is sent to a central node for decision, we propose a novel Perception-Initiative-Consensus-Action (PICA) framework. The novelty of the PICA is that decision-making does not rely on a central authority but through DC protocols, which enables a node to make the initial decision based on local sensing and computing and then consent through a distributed protocol among the relevant nodes jointly before executing an action. Through this new procedure, issues in the centralized systems listed above can be resolved. On the other hand, the distributed architecture allows decision-makers to sit as close to the end user as possible, reducing the overall end-to-end latency. From this point of view, PICA is also a class of integrated sensing and communication (ISAC) proposed in [20] [21]. However, unlike ISAC, which focuses on joint design of the sensing and communication, which typically works in trustworthy and centralized environments, we consider the multi-sensor joint decision-making in distributed and trustless environments where the communication networks are used to support the DC protocols.

Obviously, DC plays a pivotal role in the PICA approach to make agreements in the correct order among the stakeholders. There are two major types of DC for faults tolerances in distributed systems: Byzantine Fault Tolerance (BFT) [22] for malicious attacks, e.g., Man-in-the-Middle or Sybil attacks under Dolev-Yao attacker model and Crash Fault Tolerance (CFT) [23] for availability attacks, e.g., Distributed Denial of Services. Byzantine fault is distinguished when the misinformation from the malicious (i.e., Byzantine) node or a node with false information is detected; and the crash fault is identified when a member loses the connection with all other nodes and the leader node, a term-time leader of the consensus group, but the information sent by all nodes was assumed correct. Both failures can lead the system to fail in making a successful joint decision. The solutions to avoid such failure are fault tolerance design of consensus, practically used examples are: practical Byzantine Fault Tolerance (PBFT) [22] for BFT [5] [16] and Raft [23] for Crash Fault Tolerance. Both algorithms are voting-based consensuses (a.k.a. message passing consensuses [24]) that require frequent intercommunication to synchronize with peers and replicate the state from the committed nodes.

Thanks to the redundant design, the node or communication network reliability can be much lower but still able to achieve a highly reliable joint decision or consensus. This will give us more flexibility when deploying 5G and beyond URLLC networks since the communication link reliability can be relaxed. The detailed introduction of PBFT and Raft models will be introduced in Section III and Fig. 2.

## *B. Related work*

In recent works, there are few related studies on proposing and analyzing DC protocols in wireless communication scenarios. Examples can be found in a blockchain-based decentralized system architecture with reputation-based consensus proposed for IoT systems in [25], a low consumption consensus mechanism to facilitate the coordination of IoT devices in a lightweight blockchain system proposed in [26], a blockchain-based reward mechanism has been designed for mobile crowdsensing [27], and an efficient and fault tolerance blockchain consensus applied in IoT proposed in [28]. In these studies, their innovations focus on improving the performance of DC protocols in wireless communication scenarios but ignore the impact of wireless communication on DC protocols. Since the original DC protocols are based on stable communication in computer networks and usually contain the assumption of perfect communication, WDC that ignores the uncertainty of wireless communication is dangerous. A few studies focused on wireless communication analysis related to DC are detailed as follows. [29] indicated that communication performance has a significant impact on the consensus of wireless systems and provided a consensus-communication co-design framework, and [30] provides the analysis of SINR and throughput to design the optimal node deployment in a blockchain-enabled wireless IoT mode. However, their analysis scenarios are very limited because they all rely heavily on the overall architecture of the blockchain. Only by focusing on the essential metrics of DC combined with communication conditions can we provide general design guidance for distributed wireless communication.

## *C. Motivations and contributions*

PBFT and Raft are originally designed for running in stable wired networks, where the consensus thresholds (i.e., fault tolerance) are 1/3 and 1/2, respectively [23] [22]. However, unlike wired systems, wireless systems bring extra channel uncertainty [31] [32] [33], scarcity of spectrum and node communication provision (the number of messages a node processes at any given time during the consensus), thus entailing different security thresholds. In particular, original systems consider node failure, and when it happens, all associated communication links are faulty. However, with dynamic wireless communication channels, a node may work fine, but certain links connected with the node might be unstable. Additionally, the traditional design considers a deterministic situation where nodes have a fixed status. However, to extend the distribution system into the engineering applications such as autonomous systems, we have to consider the question from the statistical angle (i.e., failure is a probability). The above issues bring a natural yet important question: **what are the new consensus success probabilities (i.e., consensus reliability) in a wireless connected uncertain network?** As we mentioned above (and more will be given in Section III), DC protocols are designed with redundancy, given wireless connected DC, equivalently, research objectives are conveyed by questions which can be described as: **Is it possible to achieve high reliability and low latency mission-critical wireless distributed consensus (WDC) in a trustless en-**

**vironment through less reliable communications link or even in the presence of malicious users? How does the consensus reliability impact the key performance metrics of DC, such as the consensus throughput and latency?** Such fundamental questions must be answered before WDC can be used in vast critical autonomous systems.

This paper analyzes emerging challenges faced by wireless distributed mission-critical autonomous systems from an engineering perspective and proposes WDC based on PBFT and Raft consensus as a step forward to tackle the demanding reliability and latency with collective efforts in terms of stochastic processes of distributed decision-making analysis for WDC. The main contributions are listed as follows:

- We establish WDC models based on PICA approach and envision the usage of WDC in critical industrial applications with use case examples.
- We derive comprehensive models of the WDC based on PBFT and Raft along with essential synchronizations to further increase system reliability for PICA scheme. Namely, the analytical expressions of PBFT and Raft consensuses reliability with node failure, link failure, and both node and link failure models, respectively. The mathematical derivations provide a fundamental analytical scope looking into the composition of WDC and help to adapt the PICA beyond proposed consensuses with similar approaches.
- We define the concept of reliability gain, a metric of reliability against node quantity, which provides useful guidance for consensus network configuration. We set up criteria for WDC from a perspective of reliability and other quality measures, including latency and throughput. The practical definition of reliability gain offers easy implementation of WDC in the existing industrial system.
- Finally, We provide guidance to WDC deployment with benchmark results of reliability, latency and throughput by simulations.

The remainder of this paper gives the briefing on reliability followed by case studies of proposed WDC in Section II. A detailed PICA framework for WDC is illustrated in Section III, and made comparisons of reliability performance for PBFT and Raft protocols in Section IV, followed by Section V, where we explain the impact of latency, throughput and scalability with regard to the size of the network. The results of reliability validations with simulations are shown in Section VI and discuss the resilience of the system with the measurement of gain. Finally, Section VII summaries the paper.

## II. Reliable decisions and Case Study in critical connected systems

We present our problem statement with the following case study, where we will show readers applications that strongly demand real-time high-reliability WDC using less reliable communication links and in the presence of the failure nodes.

### A. Collision avoidance/ advisory (Clustering Decision)

The revolution of automotive industries brings autonomous driving to everyone, with great risks in its early state [34]. Many catastrophic failures happened due to sensor errors, malicious attacks and AI decision errors [7] [35] [36] [37]. In order to prevent sensors from conflicting with each other and making unreliable decisions, fault tolerance methods are applied to reassure their consistency and reliability. Such time-sensitive information is only solvable locally due to the delay and the single point of failure risk in a centrally managed network.

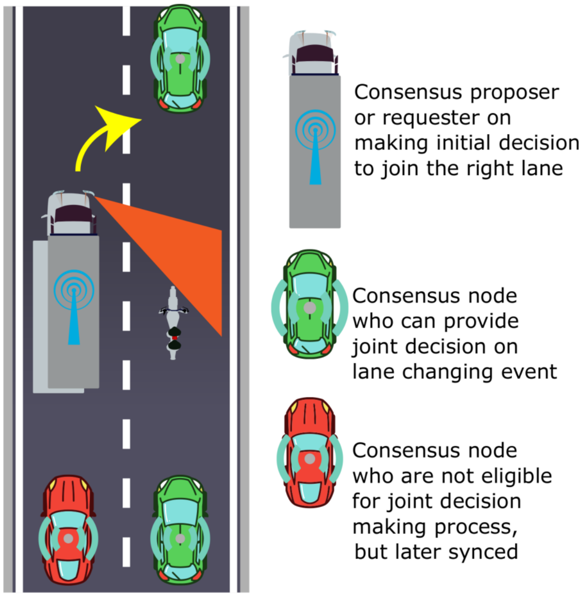

Fig. 1. Wireless distributed consensus for traffic decision with PICA scheme

Modern transportation has regulated collision advisory (CA) to provide traffic, and resolution advice [7]. For example, Traffic Collision Avoidance Systems (TCAS) [7] are widely used in aviation [38], and many emerging AI-based collision advisory systems are on-board new land-based vehicles for autonomous and semi-autonomous driving, though the reliability is well below the real-world requirement and hardly considered usable. Recent traffic accidents caused by self-driving false alarms and miss alarms have caused multiple catastrophic consequences for road users across the world [35]. Thus, a more comprehensive solution to deal with the reliability of self-driving is required, in order to widely adopt autonomous driving in a higher level, in particular the L4 and above, where needs for human interventions are minimized.

### B. X-by-wireless (wireless communication for mission-critical control)

Mission-critical payloads are the leading edge users and developers of real-time high-reliability systems with fault tolerance capacity, such as Fly-by-wire [39] and Drive-by-wire using internal databus (fieldbus) [7], e.g., ARINC 629, ARINC659 (SAFEbus), ARINC 664 Part 4 (AFDX), CAN bus, etc. Wire-based control system suffers from limited flexibility and high implementation cost regarding its installation and dead weight of wires. In the recent search of the next generation control databus, one notable research direction, which may make use of WDC, is the Fly/Drive-by-wireless or simply X-by-wireless. X-by-wireless has been at the center of the next-generation avionics research for many years [40], and the reliability issue is always the top concern for the system designer. In conventional deployments of Fly/Drive-By-Wire, databus is supplied with wired connections, and dual redundancy [7], the reliability is secured by employing duplicates in the system using First-in-First-out (FIFO) queue, which does not take Byzantine fault into consideration, since the physical network is isolated from outside. However, for wireless critical control, malicious activities, such as jamming and spoofing, are more common due to the openness of channels. Therefore, we emphasize the safety and real-time property for such applications. In the following proposed solution, the system is benefited not only from the robust redundancy management but also from taking advantage of DC to avoid high costs on higher-grade commercial-off-the-shelf (COTS) products.

## III. PICA Framework, Raft and PBFT Consensuses

### A. PICA framework

We introduce WDC based on the proposed PICA approach for reliable decisions. The initial decision (i.e., Perception) is made based on the local sensing (e.g., Lidar, Radar in CAV) and potentially combined with state-of-the-art AI techniques. A request (i.e., Initiative) is made by the node based on the initial decision and then sent to the network for a joint decision (i.e., Consensus), where only consented initial decision will be executed (i.e., Action). Under the new scheme, the decision offered by the advisory will not only be made by the standalone actuation controller. Instead, a cluster of nodes with the same visibility will be involved.

An illustration of proposed scenarios can be found in Fig. 1, where a truck will first sense the environment and make an initiative to join the right lane, and feedback will be received from the consensus network formed by the nearby vehicles, for maximum safety and reliability. When making the initiative, at least the truck believes it is safe, but other vehicles nearby may have varied sensing results, and negative feedback from the consensus network can be given. Since the initiative will be verified by other nearby vehicles' established DC networks (thus, with more sensing data), it greatly increases the reliability of the final decision even though the initial decision might be wrong or communication links are unreliable.

A WDC process starts from a decision-making request. For instance, by initiating a decision process, the client who makes the call needs to describe the decision into a statement that can only be answered with Boolean type (Yes/No), such as traffic lights at a given position are green (A simple Yes or No decision, but the decision chain can be extended from here).

It is easy to conclude that the DC, which secures low latency and ultra-reliable joint decisions with relaxed communication link or node reliability, plays a pivotal role in PICA. Next, we will present WDC models based on PBFT and Raft consensuses, in which the fault tolerance design is impacted by the wireless communication performance and node failures. The communication protocols of PBFT and Raft are illustrated in Fig. 2 with detailed steps and phases.

During the deployment, there are also challenges the PICA has considered, in particular from a communications perspective. The challenge in the actual deployment can be classed into two aspects: the network deployment with multiple options in L1 physical layer access and the L2/L3 datalink/network layer protocol. In the physical layer, the actual communication channel may pose a challenge to the WDC network, where the presence of back-scattering signals and variance in the receiver performance may reduce the communication link reliability significantly. Therefore, we have previously modeled the system with various link reliability to mitigate our limitation on the physical layer. The performance of aggregated link reliability is also impacted by the link-layer protocols, e.g., HARQ and CSMA/CA. On the other hand, the communication protocol may induce extra overhead and latency elements into the system, in particular, if the communication protocol is employed as the method to improve the overall link reliability (where the individual link reliability is reinforced by the communication protocol and coding options).

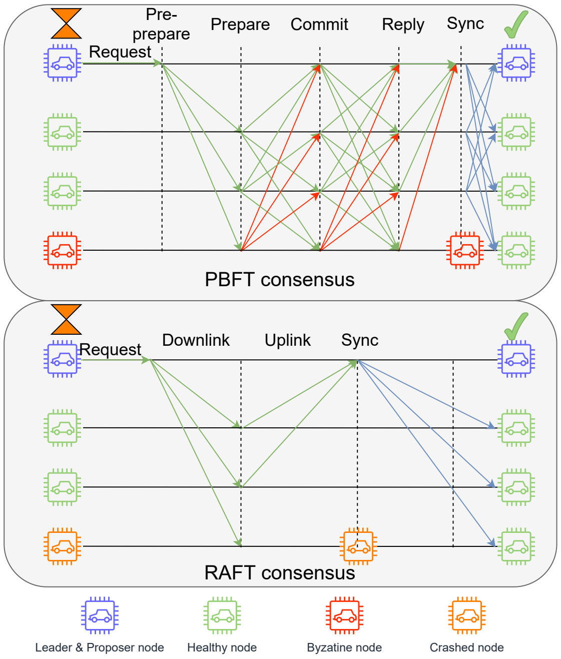

Fig. 2. Consensus protocols of PBFT and Raft with synchronization stage

### B. Wireless distributed consensus model: PBFT

WDC based on PBFT involves actions that may bring conflicts to the consensus parties' interests, such as malicious nodes' presences given malicious feedback, for example, the backup sensors (failed ones are considered as Byzantine nodes) are giving different readings at the same time, where the value can be different, such false information is considered as Byzantine fault. In this paper, we will consider such misinformation (false information unintentionally given by honest node due to the perception/sensing errors) rather than disinformation (false information intentionally given by malicious node), though the same effect will be made from the consensus perspective. In this case, the quorum needs to make a Byzantine-proof decision with additional communications between members, which tolerates less than one-third of Byzantine nodes.

There are three phases of communications that are vital in PBFT protocol for the consensus [22], namely, $pre-prepare$, $prepare$, $commit$ and a $reply$ message critical to the successful operation, as shown in Fig. 2, where we see that PBFT relies on frequent inter-node communications. During $pre - prepare$, the leader node sends a message to all other nodes, and $prepare$ phase, all other nodes duplicate and propagate $prepare$ message to all nodes excludes itself, $commit$ phase does the same communication as the previous phase, and at the $reply$ phase, when leader nodes have received enough $commit$ messages, it replies to the client while synchronizing the latest results with its peer nodes, as shown in Fig. 2. Note that, in a functional PBFT consensus group, a threshold of less than $1/3$ of Byzantine nodes is required to yield correct decisions.

In PICA scheme, the client naturally takes the role of header node when the request is initiated. However, the model built in this paper is also applicable to the case that the client is not part of the consensus network, where the header is the representative of the client.

### C. Wireless distributed consensus models: Raft

The Raft consensus model represents the network with no conflict of interest. All nodes are honest in the system, and such a mutual decision on information fits every node's interest. The leader node is self-elected during this process when the node makes the call and broadcasts it to the perimeter. The protocol of Raft started from receiving the message from the leader during $downlink$, as shown in Fig. 2 lower part, any node within the range that has the ability to make the judgment will provide its opinion to the leader to either confirm it or deny it via $uplink$ communications. Taking Fig. 1 as an example, we can see the truck (leader node) is about to merge into the right lane. By requesting confirmation of obstacles in the blind zone covered in amber, the other vehicles (nodes) are able to tell the truck if it is clear to proceed based on the Raft protocol. The failed node marked in red is not able to give feedback on the situation though it is still part of the consensus group. In this illustration, the red car can only flag itself as failed node due to lack of visibility, which makes the failure a crash fault. Having the following synchronization stage taken into account, such a crash can be mitigated and recoverable if the node is still functional.

Once the leader node receives enough feedback from its follower nodes in both scenarios, it will either note the information has been confirmed or act based on the confirmed information. During the consensus process, there are security thresholds to ensure it has the best decision depending on the reliability and latency requirement. In our case of Raft, more than 50% viable nodes during both uplink and downlink are required, compared to 33% viable nodes required by PBFT, in a combination of communications and nodes reliability.

### D. Full Consensus with synchronization

In addition to the standard PBFT and Raft models introduced above, we propose a synchronization process after completing the decision, which will be detailed in Section IV-C, also shown in Fig. 2 Sync procedures. The sync takes place among failed nodes with committed nodes from the last successful consensus and prepares them in sync before the next consensus commencing, which is called full consensus. It does not affect PBFT but Raft reliability and latency performance at the time of decisions, and it is important to occasions where all nodes should be aware of the joint decision was made, even though part of them did not achieve the final commit stage due to the communication failure.

TABLE I
FREQUENTLY USED NOTATIONS

| Notation | Definition |
|---|---|
| $P_{R-N}$ | Consensus reliability of Raft (node failure) |
| $P_{R-L}$ | Consensus reliability of Raft (link failure) |
| $P_{R-N-L}$ | Consensus reliability of Raft (node and link failures) |
| $P_{P-N}$ | Consensus reliability of PBFT (node failure) |
| $P_{P-L}$ | Consensus reliability of PBFT (link failure) |
| $P_{P-N-L}$ | Consensus reliability of PBFT (node/link failures) |
| $P_n$ | Reliability of nodes |
| $P_l$ | Reliability of communication link |
| $n$ | Number of nodes in the consensus network |
| $f$ | Number of failed/faulty nodes |
| $m$ | Number of successful nodes |
| $P_v$ | Possibility of view change |
| $L$ | Latency of the consensus |
| $T$ | Throughput of the consensus |

## IV. Reliability of wireless distributed consensus

Section III has qualitatively presented why WDC can enhance the joint decision reliability. In this section, the reliability of WDC based on PBFT and Raft consensuses will be derived under non-perfect communication links and nodes. The reliability of a system is critical for mission-critical decision-making, where the process consists of many components, subsystems and external elements, e.g., computing nodes, physical connectors, and wireless channel quality for wireless connected systems. Specifically, in WDC, we consider the following three cases, node failure only, communication link failure only, and both node and communication link failure. A node failure is described as either crashed in Raft or Byzantine-like behavior in PBFT in probability, and the communication link reliability is defined as the statistical probability of success of the point-to-point wireless communication at the given time, which is also a simplified value for a set of network and traffic environment with considerations on channel modeling and interference. The symbols are listed in Table I, where the third model (with subscripts P-N-L for PBFT and R-N-L for Raft) combines the assumptions of the first two models and yields the overall probability of success of the consensus.

As a general assumption, we consider every **node's reliability** $P_n$ in node failure model with a given number of $n$ nodes, and **communication link probability of success** $P_l$ of every wireless channel between nodes. Although, the effect of the initially failed leader node will result differently, which will be discussed in the later section of latency performance analysis regarding consensus view changes ($VC$) (i.e., the process of electing a new leader) [22]. It is assumed that the leader node is always available at the first round of communication for reliability analysis for simplicity, upon the assumption that this very node is the initiator of the consensus.

### *A. Consensus reliability based on PBFT*

PBFT consensus system provides safety and liveness against malicious attacks up to $f = \lfloor \frac{n-1}{3} \rfloor$ faulty nodes [22], where $f$ is the number of faulty nodes, among number of $n$ nodes. According to the consensus requirements of the original PBFT protocol, the number of nodes that successfully participated in all consensus phases should not be less than $n - f$ [22]. The consensus reliability model for PBFT consists of PBFT node failure (P-N) model, PBFT link failure (P-L) model, and PBFT node and link failure (P-N-L) model, respectively, detailed in the following subsections.

**Theorem 1.** *Given $P_n$, $P_l$ and $n$, the probability of success of the DC $P_{P-N-L}$ can be obtained by*

$$\begin{aligned} P_{P-N-L} = \sum_{m=m_{pp}}^{n} \sum_{m_{pp}=m_p}^{n} \sum_{m_p=m_c}^{n} \sum_{m_c=n-f}^{n} & (P_{node}(n,m) \cdot \\ & P_{pp}(m,m_{pp}) \cdot P_p(m_{pp},m_p) \cdot P_c(m_p,m_c)), \end{aligned} \quad (1)$$

*where $P_{node}(n,m)$, $P_{pp}(m,m_{pp})$, $P_p(m_{pp},m_p)$ and $P_c(m_p,m_c)$ are given below.*

$$P_{node}(a,b) = \binom{a-1}{b-1} P_n^{b-1}(1-P_n)^{a-b} \quad (2)$$

$$P_{pp}(a,b) = \binom{a-1}{b-1} P_l^{b-1}(1-P_l)^{a-b} \quad (3)$$

$$P_c(a,b) = \binom{a}{b} P_s(a)^b(1-P_s(a))^{a-b} \quad (4)$$

$$\begin{aligned} P_p(a,b) = & P_s(a)\binom{a-1}{b-1} P_s(a-1)^{b-1}(1-P_s(a-1))^{a-b} \\ & + (1-P_s(a))\binom{a-1}{b} \\ & \times P_s(a-1)^b(1-P_s(a-1))^{a-1-b} \end{aligned} \quad (5)$$

*$P_s$ in equation (4) and (5) are denoting*

$$P_s(a) = \sum_{k=2f}^{a-1} \binom{a-1}{k} P_l^k (1-P_l)^{a-1-k}. \quad (6)$$

*1) PBFT Model with Node and Link Failure (P-N-L):* In this model, node failure (i.e., $1 - P_n$) and link failure (i.e., $1 - P_l$) are considered at the same time. We have the following theorem to show the relationship between them and the consensus reliability $P_{P-N-L}$, as seen in Theorem 1.

Theorem 1 provides a precise equation of the overall reliability of the PBFT system in the real world, where the reliability of nodes or communication links is not guaranteed. Derived by successive summation and multiplication in Theorem 1, the computational complexity of the P-N-L model is $O(n^5)$. The high computational complexity is justified as the derivation equations are precise, and the computational complexity cannot be reduced as the accurate equations cannot be optimized.

The proof of Theorem 1 is given in Appendix A.

*2) PBFT Model with Node Failure (P-N):* In P-N model, we only consider node failure rate $P_n$. It is actually the case that the channel in the consensus system is always reliable in P-N-L model. If we set $P_l = 1$ in equation (1), we have the following remark to show the relationship between $P_n$, number of node $n$ and the WDC reliability:

**Remark 1.** *Given $P_n$, $n$ and $P_l = 1$, the probability of successful consensus of the system $P_{P-N}$ can be obtained by*

$$P_{P-N} = \sum_{m=n-f}^{n} P_{node}(n,m). \quad (7)$$

*The expression of $P_{node}(n,m)$ is denoted in equation (2) in Theorem 1.*

Remark 1 provides a straightforward answer to node failure mode for PBFT consensus, and it is useful to estimate the reliability of the system while the communication link is stable, e.g., wired connected scenario or when sufficient spectrum resource is used in wireless communications (e.g., repeat transmission).

*3) PBFT Model with Link Failure (P-L):* For P-L model, it is also a special case that all the nodes are reliable in P-N-L model. By taking $P_n = 1$, we can illustrate the relation between link probability of success $P_l$, the number of nodes in the system $n$ and the consensus rate of P-L model $P_{P-L}$.

**Remark 2.** *Given $P_l$, $n$ and $P_n = 1$, the probability of successful consensus of the system can be calculated in the following equation:*

$$\begin{aligned} P_{P-L} = \sum_{m_{pp}=m_p}^{n} \sum_{m_p=m_c}^{n} \sum_{m_c=n-f}^{n} & [P_{pp}(m,m_{pp}) \cdot \\ & P_p(m_{pp},m_p) \cdot P_c(m_p,m_c)]. \end{aligned} \quad (8)$$

*The expressions of $P_{pp}(m,m_{pp})$, $P_p(m_{pp},m_p)$ and $P_c(m_p,m_c)$ are denoted in equation (3), equation (5) and equation (4) in Theorem 1.*

The analytical and simulated results of the relationship of consensus failure rates $1 - P_{P-N}$, $1 - P_{P-L}$ or $1 - P_{P-N-L}$ and total number of nodes $n$ are detailed in Section VI, it provides a theoretical mitigation on weak nodes, weak communication links or combined scenarios. Note that in the case of perfect communication links or nodes, Remark 1 and 2 are the special cases of Theorem 1.

### *B. Consensus reliability based on Raft*

Raft consensus mechanism tolerates $\lfloor \frac{n-1}{2} \rfloor$ faulty nodes out of $n$ total nodes during a successful consensus. As indicated by Fig. 2, each follower node casts its vote to the leader via an uplink channel, and the majority wins. The reliability of Raft-based WDC is established under the node failure, communication link failure, and both node and link failure, described in the following sections.

*1) Raft Model with Node and Link Failure (R-N-L):* In Raft node and link failure (R-N-L) model, we set the non-faulty probability of each node and single link reliable rate as $P_n$ and $P_l$ respectively, and aim to obtain the final success consensus rate $P_{R-N-L}$, as seen in Theorem 2, details of the derivation for Theorem 2 is given in Appendix B.

The combined failure mode of Theorem 2 is close to real-world Raft deployment. An illustration of combined models with ascending number of nodes is detailed in Section VI. The computational complexity of R-N-L model is $O(n^4)$. Similar to the P-N-L model, the high computational complexity cannot be reduced as the accurate equations cannot be optimized.

**Theorem 2.** *Given $P_n$, $P_l$ and $n$, $P_{R-N-L}$ can be calculated as*

$$\begin{aligned} P_{R-N-L} = &\sum_{a=\lceil \frac{n-1}{2} \rceil}^{n-1} (\binom{n-1}{a} P_l^a (1-P_l)^{n-1-a} \\ &\sum_{b=\lceil \frac{n-1}{2} \rceil}^{a} (\binom{a}{b} P_n^b (1-P_n)^{a-b} \\ &\sum_{c=\lceil \frac{n-1}{2} \rceil}^{b} \binom{b}{c} P_l^c (1-P_l)^{b-c})). \end{aligned} \tag{9}$$

*2) Raft Model with Node Failure (R-N):* In Raft node failure (R-N) model, nodes have an reliability of $P_n$ in the log replication stage and the links are assumed reliable, where $P_l = 1$, for the number of $n$ nodes, we have following Remark:

**Remark 3.** *By replacing $P_l = 1$ in R-N-L model, the probability of successful consensus of R-N model can be obtained by*

$$P_{R-N} = \sum_{i=\lceil \frac{n-1}{2} \rceil}^{n-1} \binom{n-1}{i} P_n^i (1-P_n)^{n-1-i}. \tag{10}$$

*3) Raft Model with Link Failure (R-L):* In Raft link failure (R-L) model, we assume $P_l$ is the probability of success for every channel and the probability of successful consensus of the system is $P_{R-L}$, for the number of $n$ nodes, we have the following Remark:

**Remark 4.** *Similarly, by replacing $P_n = 1$ in R-N-L model [5], the probability of successful consensus of R-L model can be obtained by*

$$\begin{aligned} P_{R-L} = &\sum_{a=\lceil \frac{n-1}{2} \rceil}^{n-1} (\binom{n-1}{a} P_l^a (1-P_l)^{n-1-a} \\ &\sum_{b=\lceil \frac{n-1}{2} \rceil}^{a} \binom{a}{b} P_l^b (1-P_l)^{a-b}). \end{aligned} \tag{11}$$

Remark 4 provides the analytical equation of Raft link failure model, and it is useful while the estimation is made for a node stable situation, or for a short period of time, where the node failure is less likely, for instance, nodes refreshed after passing the Mean Time Between Failure (MTBF) threshold.

## *C. Reliability of full Consensus with synchronization*

A complete and successful round of consensus requires all non-faulty nodes to sync up to actuate the outcome of consensus. However, the faulty nodes of the current round will be left out and prohibited from entering the next consensus round. Hence, it is important to sync up the previously failed nodes to maintain the liveness of the whole system, shown as Sync in Fig. 2. To achieve full consensus, we add a synchronization phase to help all the nodes who failed due to link failures to update the latest log from the successful nodes and ready them for future requests.

As the communication principles between nodes of PBFT and Raft are different, the *sync* phases added to PBFT and Raft are also different. To extend two protocols uniformly, an alive-node broadcast phase, similar to the original *commit* phase, is added as *sync* phase to PBFT, while a leader broadcast phase, similar to *downlink* phase, is added as *sync* phase to Raft. The following is a detailed description of the *sync* phase in PBFT and Raft.

In the case of PBFT, for the nodes which do not experience any node or communication link failures during the consensus process, they enter the synchronization phase by multicasting synchronization messages *sync* to all other nodes. When a node receives *sync* messages, it will check if it has both *prepare* and *commit* certificates with the same view number, sequence number and request digest as the synchronization message provided. If the request in the synchronization message has not been committed, it accepts the message and waits for a weak certificate via *sync* consensus, which requires fewer message counts than the normal consensus process with at least $f + 1$ *sync* messages with the same view, sequence number and request's digest from different nodes. Otherwise, the node remains unchanged. We call this weak certificate the synchronized certificate. Nodes with this certificate execute the request and update their logs without replying to the client. Similar to the reply certificate in PBFT model, the synchronized certificate with $f + 1$ messages from different nodes aims to ensure the synchronization operation is valid since there is at least one reliable message which indicates that the request has been accepted by a quorum.

As for Raft, only the leader node is able to know whether a consensus process has been completed, so *sync* phase in Raft is simply a broadcast of *sync* messages from the leader to all the other nodes in the system, and *sync* messages are intended to sync up all the failed nodes caused by link failures.

To calculate the reliability of full consensus with synchronization, we conclude the following remarks to show the probability of successful consensus in P-N-L and R-N-L models with *sync* phase.

**Remark 5.** *Given $P_n$, $P_l$ and $n$, the probability of success of the DC with sync phase in P-N-L model can be calculated by*

$$\begin{aligned} P_{P-N-L} = &\sum_{m=m_{pp}}^{n} \sum_{m_{pp}=m_p}^{n} \sum_{m_p=m_c}^{n} \sum_{m_c=n-f}^{n} [P_{node}(n,m) \cdot \\ &P_{pp}(m, m_{pp}) \cdot P_p(m_{pp}, m_p) \cdot P_c(m_p, m_c) \cdot P_{syn}(m, m_c)], \end{aligned} \tag{12}$$

*where $P_{node}(n,m)$, $P_{pp}(m,m_{pp})$, $P_p(m_{pp},m_p)$ and*

*$P_c(m_p, m_c)$ are provided in Theorem 1 while $P_{syn}(m, m_{pp})$ is given below.*

$$P_{syn}(a,b) = (\sum_{k=f+1}^{b} \binom{b}{k} P_l^k (1-P_l)^{b-k})^{a-b} \tag{13}$$

Similarly, by calculating the consensus-reaching rate of the Raft system after the added $sync$ phase in R-N-L model, we have the following Remark.

**Remark 6.** *Given $P_n$, $P_l$ and $n$, we can calculate the probability of success of the DC with $sync$ phase in R-N-L model as*

$$\begin{aligned} P_{R-N-L} = &\sum_{a=\lceil \frac{n-1}{2} \rceil}^{n-1} (\binom{n-1}{a} P_l^a (1-P_l)^{n-1-a} \\ &\sum_{b=\lceil \frac{n-1}{2} \rceil}^{a} (\binom{a}{b} P_n^b (1-P_n)^{a-b} \\ &\sum_{c=\lceil \frac{n-1}{2} \rceil}^{b} \binom{b}{c} P_l^c (1-P_l)^{b-c})) P_l^{n-1-a-c+b}. \end{aligned} \tag{14}$$

Added $sync$ phase adds additional requirements to the consensus completion, which means synchronization has a negative impact on probability of successful consensus. The analysis of the impact is described in Section VI.

By defining the probability of success of consensus synchronizations, the overall probability of success can be concluded. Note that, in all scenarios, when one of the components (node or link) reliability reaches 1 or failure rate reaches 0, it matches up with the conclusion stated in Remark 5 and Remark 6.

## V. Latency, throughput and node scalability analysis

In WDC deployment, in addition to consensus reliability, another two important but reciprocal performance metrics are consensus latency and throughput. Consensus latency is the time cost for a complete consensus, and the throughput is measured by transaction per second (TPS) [19]. On the other hand, WDC systems are also bounded by node scalability, which indicates how well the system can expand without scarifying the consensus latency and throughput of the system, given the network resources.

Latency, throughput and scalability of WDC are jointly decided by DC protocol and communication resource provision [19]. PBFT and Raft are intrinsic with great latency and throughput performance, which is applicable to the high reliability and low latency scenarios (C-CAS). Although it is well known that as voting-based consensus, the PBFT and Raft can only be adopted in small networks, the limited scalability may not cause a major performance concern due to the latency requirement, which restricts the size of the consensus group to be within a real-time-ready range.

As the system grows in size, the latency will increase due to prolonged timeout settings for maintaining the coverage in every phase of communications. The reason is that the greater time-window is required to complete the communications in each stage with a given spectrum (since there are more nodes to communicate in each stage, as shown in Fig. 2). In other words, the increased size of the consensus network sets higher demands for normal operation time, defined as $t_n$ for a successful consensus with sufficient coverage (minimal number of valid nodes) without view changes or leader re-elections. Given the size of the network, WDC selects all valid nodes optimally from the coverage by limiting $t_n$ of time-sensitive tasks, such as real-time decisions. The direct increases in network size lead to extended $t_n$, hence lowering the throughput of the overall network.

In the next, we will analytically derive the consensus latency, throughput, node scalability, and their relationships by providing guidelines for the real system deployment.

### A. Consensus Latency

During the continuous consensus process in previous sections, we have not considered the failure of header node in the first place, nor the $VC$. By including the exceptions, such as, header node failures and view changes in the consensus model with combined node and link failure rate, we can assume the overall latency is the average latency of a set of the consensus plus a set of fixed delay, for instance, packet packing delay, propagation delay and processing delay, which are not significant compared to the overall time [41]. In the WDC communication system, the latency of a consensus mainly consists of two parts: normal operation latency and $VC$ delay, an action to switch to the next consensus round, hence, for a given number of $i$ times of normal operation $t_n$, the overall time is $i \times t_n$. However, for the $VC$, the required time $t_{vc}$ is dependent of $t_n$ and $t_e$, defined by following equation,

$$t_{vc} = 2t_n + j \times t_e, \tag{15}$$

where $j$ is the number of $VC$ took place (which is related to the failure rate, as higher the failure rate, more view changes happen, e.g., with 99.99% probability of success, $j$ will be 1 out of 10000 consensus processes), and $t_e$ is the extra time added to every $VC$ for PBFT or re-election for Raft, note that $t_n$ is also required after the $VC$. From equations (1) and (9).

**Remark 7.** *By defining the final probability of successful consensus as $P_F$, we have the possibility of $VC$ of PBFT (or the chance of header re-election for Raft), $P_v$ of $K+1$ total attempts following geometric distribution $K+1 \sim GE(P_F)$ , which can be written as:*

$$P_v = \sum_{k=1}^{K} (1-P_F)^k P_F. \tag{16}$$

Using Remark (7), we can establish the overall latency regards to $t_n$ of PBFT and Raft.

**Remark 8.** *Given $t_n$, the consensus latency $L$ against $t_n$, regardless of the consensus and actual processing time but reliability of the consensus, is calculated as:*

$$L = \sum_{j=1}^{\infty} [(1-P_F)^j P_F (j \times t_e + 2t_n)] + P_F \times t_n. \tag{17}$$

TABLE II
PERFORMANCE COMPARISON OF FEATURED CONSENSUSES [19]

| Consensus | BFT capability | Transaction Throughput | Scalability | Security Bound | Consensus Communication Complexity [19] | Consensus Communication Provision [19] |
|---|---|---|---|---|---|---|
| PBFT [22] | Yes | High | Low | 33% | $2n^2 + n$ | $2n + 1$ |
| Raft [23] | No | Very High | Medium | 50% | $2n$ | $n + 1$ |

### B. Consensus throughput

Transaction throughput is measured by Transaction Per Second (TPS). Meanwhile, the transaction throughput and latency are also related to the number of nodes in the consensus network. Hence, we can obtain the consensus throughput $T$ as,

$$T = 1/L, \tag{18}$$

where the latency $L$ is positively correlated with failure rate based on equation (17), so is the throughput.

### C. Consensus node scalability

Node scalability is an essential metric to measure the performance of the consensus handling the increasing number of nodes. Consensus varies a lot regarding scalability, for instance, Proof-based consensuses [42] are very scalable thanks to their nature, but when it comes to the voting-based consensus, in our cases, PBFT and Raft are heavy on inter-node communications with time-sensitive stages. With the latency performance requirements raised by certain decision tasks, the scalability is latency constrained in terms of choosing appropriate $t_n$.

As the size of the network grows, the communication complexity of WDC increases rapidly. From Table II, we can see the communication complexity of PBFT and Raft, which are the most iconic consensuses implied in this study, as shown in Fig. 2. Raft has a rather linear growth of message count while the nodes' number increase. However, the scalability of PBFT is troubled by its quadratic expanding complexity [19]. Thus, from the communication complexity perspective, the PBFT-based blockchain hardly scales up, therefore the size of the network is limited while deploying the PBFT consensus. A recent breakthrough of multi-layer PBFT [16] has developed a layered PBFT architecture with weighted participants' methodology, which is a promising solution to the PBFT scalability issue.

Meanwhile, with a quadratic growth trend of PBFT, the communication provision is linearly increased with the number of nodes (as the provision is concluded from one node at the given phase/stage, i.e., at any given time, a node only processes the number of messages defined by communication provision), hence the scalability issue of PBFT for any specific node at the certain time (i.e., during one phase) is less concerned than overall communication complexity, as shown in Table. II. In Fig. 3, we have demonstrated the scalability analysis of PBFT and Raft in terms of communication complexity and communication provision (the number of messages a node processes, i.e., the recourse occupancy, at any given time during the consensus) in Fig. 3. Raft is significantly more scalable than PBFT in terms of communication cost. However, PBFT provides extra security features as it is Byzantine-proof with $1/3$ security bound.

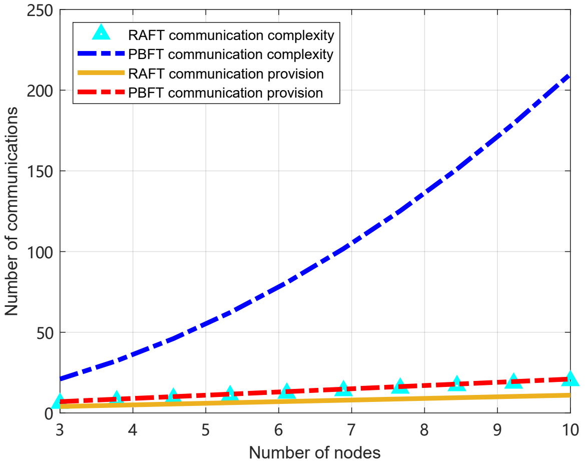

Fig. 3. Comparison of scalability for PBFT and Raft in terms of communication complexity and provision

## VI. SIMULATIONS AND DISCUSSIONS

To verify our analytical results for both PBFT and Raft based models, simulations are carried out to compare the analytical results in reliability, both with and without synchronization scenarios. With the verified results of reliability models, we provide an insightful simulation of reliability gain, which verifies the usefulness of WDC. Moreover, the latency performance is simulated against reliability (failure rate), to indicate the relationship between latency and reliability.

### A. Simulations of WDC reliability

*1) WDC Reliability:* To verify Theorem 1, Theorem 2 and their special cases of Remarks 1 and 2 for PBFT, and Remarks 3 and 4 for Raft. We set up successive sets of values for $n$, $P_n$, $P_l$ to examine the performance of each model and illustrate the relationship between consensus failure rate to the number of nodes for each model, in Fig. 4 for PBFT and Fig. 5 for Raft.

In P-N and R-N models, we assume that a failed node would not respond in any consensus phase. We obtain the result of each simulated consensus by checking if the number of failed nodes exceeds the upper limit $\lfloor \frac{n-1}{3} \rfloor$ for PBFT and $\lfloor \frac{n-1}{2} \rfloor$ for Raft. The consensus failure rate, $1 - P_{P-N}$ and $1 - P_{R-N}$ can be seen in Fig. 4 (b) and Fig. 5 (b), where the reliability of PBFT and Raft nodes are assumed to be 0.99 and 0.9 respectively (a realistic assumption for the consumer-grade product, e.g., iPhone 6 in 2017 has been reported with

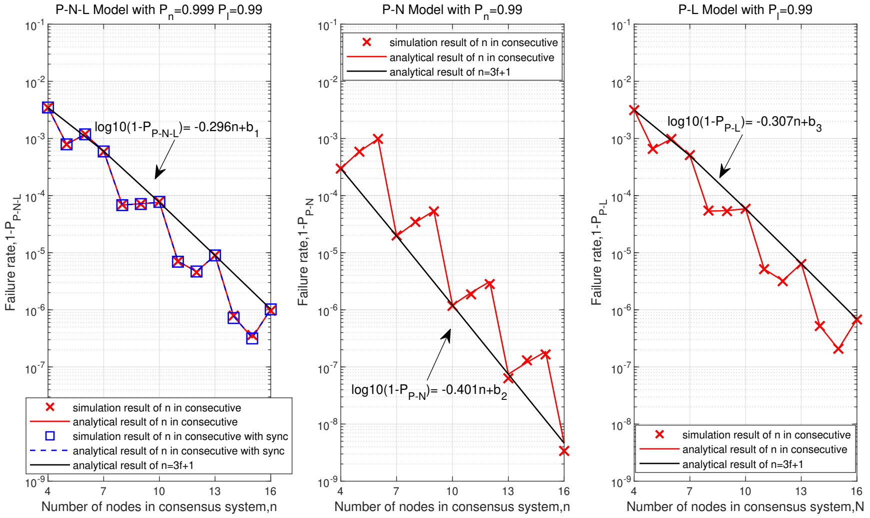

Fig. 4. Reliability performance of PBFT consensus with combined failure rate (a: left), node failure rate (b: middle), link failure rate (c: right)

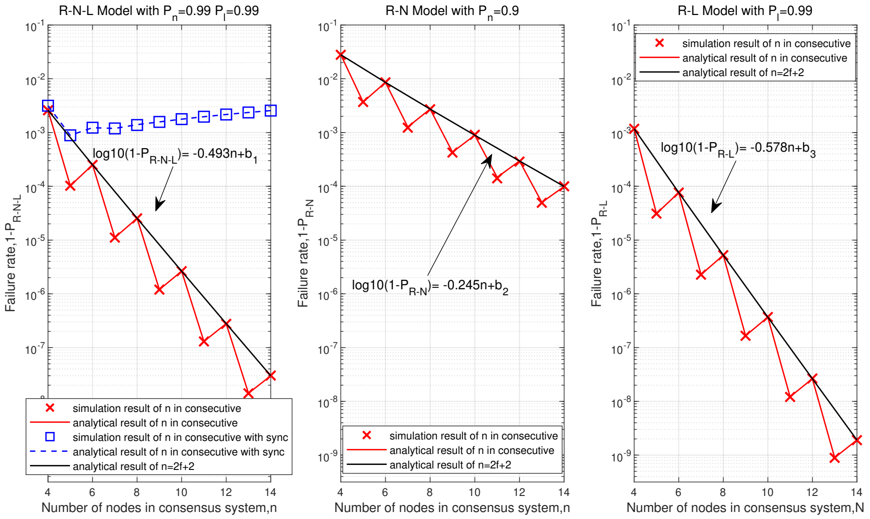

Fig. 5. Reliability performance of Raft consensus with combined failure rate (a: left), node failure rate (b: middle), link failure rate (c: right)

less than 700 hours MTBF of non-self recoverable failures, and less than 10 hours of MTBF for temporarily occurred glitches [43]). The proven remarks of P-N and R-N can be used in a fast validation of WDC when the wireless links are not involved.

In P-L and R-L models, a link failure leads to the failure in the corresponding communication phase, and only the live nodes enter the next round of consensus, as illustrated Remark 2 and Remark 4. According to the behavior of the PBFT and Raft at each step of consensus processes, each link is applied with a uniformly distributed random number to simulate the uncertainty of every transmission. The number of valid messages is counted for every live node in each phase. If the number of the live nodes after the commit/uplink phase is more than $n-f$, the consensus process is successful. In the simulation, we set $P_l = 0.99$ for both PBFT and Raft, as seen in Fig. 4 (c) and Fig. 5 (c). Note that the $P_l$ values are carefully crafted for a comparable WDC reliability range according to earlier results of P-N and R-N models instead of the consistent values from P-N-L and R-N-L simulations. P-L and R-L models can be used in a fast validation of WDC when the nodes are considered reliable, which is particularly useful for transient WDC evaluations that only involve a small amount of time. It can also be deduced that transient evaluations are not sensitive to node reliability but wireless link reliability.

To build the simulation of P-N-L and R-N-L models, we leverage the same method as before and combine P-N/R-N and P-L/R-L models. The relationships between the number

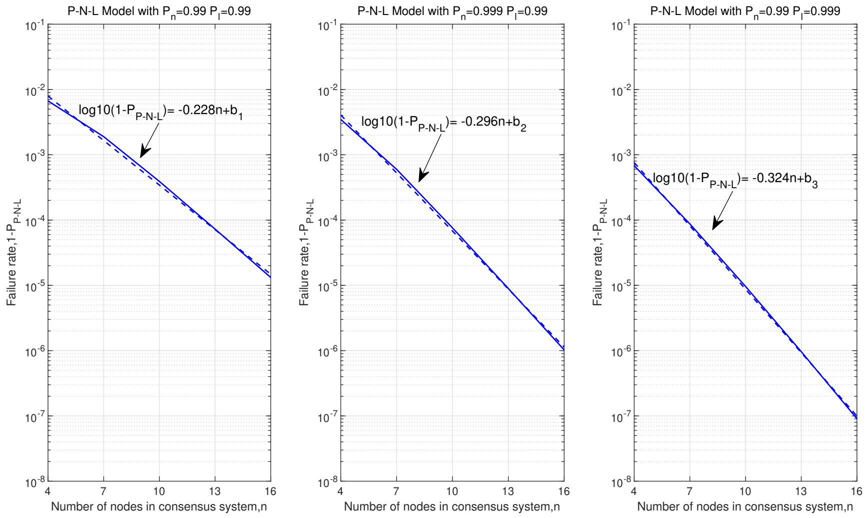

Fig. 6. PBFT reliability amplifier (a: balanced, b: node-heavy, c: link-heavy)

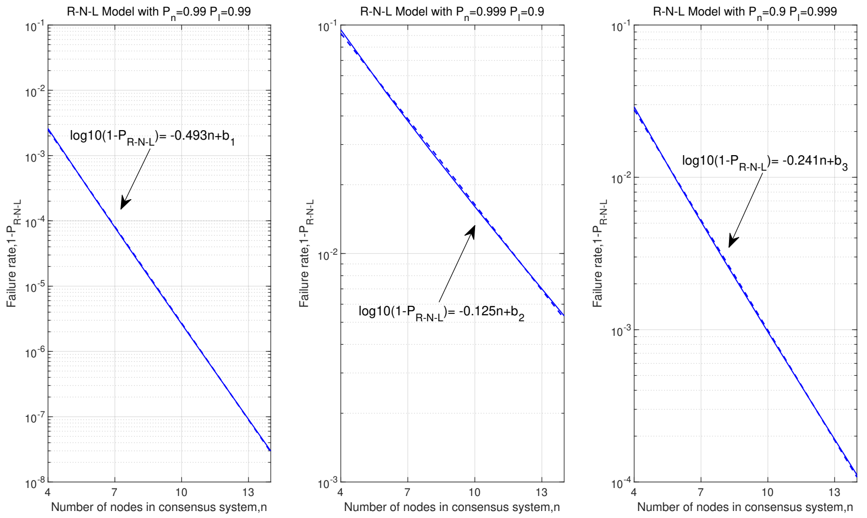

Fig. 7. Raft reliability amplifier (a: balanced, b: node-heavy, c: link-heavy)

of nodes $n$ in WDC and the consensus failure rate in P-N-L model $1-P_{P-N-L}$ and R-N-L model $1-P_{R-N-L}$ are shown in Fig. 4 (a) and Fig. 5 (a) respectively. In these two figures, we set $P_l = 0.99$ and $P_n = 0.999$ for P-N-L model and $P_l = 0.99$ and $P_n = 0.99$ for R-N-L model. The results are of significant importance to the practical system design for two reasons. Firstly, given the communication network or node reliability, we can adjust the size of the consensus network to achieve the required consensus reliability in different application scenarios. Moreover, the confident correlation between analytical and simulated results guides the future distributed network deployment where less reliable COTS and wireless connection may be adopted for high-reliability applications.

It can be seen from Fig. 4 and Fig. 5 that the simulation and analysis results match each other, indicating that the analysis of node and link failure by each model is reliable. It's worth noting that the number of nodes in different cases are grouped by a consistent upper boundary (the worst case). The zigzag shape of red lines are induced by the different remainder of security threshold, for instance, $n = 3f + 1$, $n = 3f + 2$ and $n = 3f + 3$ for PBFT and $n = 2f + 2$, $n = 2f + 2$ for Raft. The number of faulty nodes cannot be divided into consecutive integers, which leads to the discontinuity in the trend of the failure rate of the consensus. According to the relation of the security threshold $f$ and the total number of nodes $n$ in each group with the same remainder, we can observe that the proportion of $f$ in $n$ grows as $n$ increases, which means the faulty tolerance of the consensus system increases as the total number of nodes increases. This matches the tendency of each case in Fig. 4 and Fig. 5, i.e., the failure rate of a consensus process decreases gradually as the number of nodes increases. Moreover, we specifically indicate the group with $n = 3f + 1$ and $n = 2f + 1$ in Fig. 4 and Fig. 5, as the number of nodes shows a linear relationship with the consensus failure rate in log scale.

*2) Simulations of synchronized model:* To analyze the impact of $sync$ on the reliability of consensus systems, we show the analytical results by applying equation (12) and equation (14) in Remarks 5 and 6, with a set of simulations. To compare the result of scenarios with $sync$ phase with P-N-L and R-N-L, we set the same $P_n$, $P_l$, and $n$ as in the original P-N-L and R-N-L models and plot a set of analysis and simulation results in Fig. 4 (a) and Fig. 5 (a) using lines colored in blue.

Comparing the lines of cases with and without $sync$ phase, it is clear that the final probability of successful consensus of P-N-L does not significantly decrease, while R-N-L experiences a decrease when the synchronization is taken into consideration. This is caused by the different $sync$ phases in PBFT and Raft. As the $sync$ phase shown in Fig. 2 and protocol details described in Section IV-C, all live nodes broadcast to help sync up in PBFT case to yield a high chance of success for each link-failed node sync up. However, in Raft scenario, only the leader is able to broadcast $sync$ message. It is much more difficult to help all the link-failed nodes sync up since a single $sync$ message may not reach other nodes as reliable as the PBFT-like group multicasting.

Although the synchronization lowers the probability of successful consensus, it is still meaningful that the added $sync$ phase synchronizes the link failure nodes, which can increase the probability of successful consensus for later requests.

### B. Reliability gain

The feature of the WDC is the resilience that it is capable. A gain of resilience can be obtained by limiting the size of the network and ranging the latency requirement. For instance, the overall resilience can be improved by using higher reliability products or adding nodes to the network and allowing a longer time for response. The reliability gain can reflect the ultimatum performance of WDC, as it can be used as design guidelines for DC group deployment.

Combining the reliability amplification from the consensus, we have demonstrated the capability of achieving highly reliable WDC. In Fig. 4 and Fig. 5, we can see that the fusion has pushed the overall reliability into ultra or hyper reliable state (above 99.9999%), using consumer graded nodes under different link reliability scenarios. And there is a linear relationship while we increase the number of nodes in certain steps, for instance, $3n + 1$ for PBFT and $2n$ for Raft. The correlation of reliability scale and node quantity shows definitive linearity in log space, with a sophisticated mathematical inter-operation that has been partially revealed in [5] for Raft consensus approximation. The linear equations indicated in Fig. 4, Fig. 5, Fig. 6 and Fig. 7 are all obtained by applying polynomial curve fitting and polynomial evaluation to find the quadratic polynomial of least squares fitting.

Fig. 6 and Fig. 7 show that the relationship between system failure rates and node number, which is linear despite the deviation at the start due to the small number of nodes. To examine the generic linearity of the proposed reliability gain model, we have selected three configurations for comparison. The first configuration uses balanced reliability parameters for nodes and links, i.e., the same reliability for both parts. The second configuration is a node-heavy setup, where the node has better reliability than links, with the inverted setups in the third link-heavy configuration.

### C. Consensus latency simulation

Latency, a critical performance indicator, is also an important indicator of how fast the system synchronizes and produces the requested decision results. In equation (17), the latency is established against the failure rate of WDC. Hence the simulation is set to perform random WDC failures based on the failure targets, i.e., counting the average time spent on performing $10^9$ times the consensus process mocked up using a uniform distribution, and the simulated latency can be worked out. Fig. 8 shows the comparison of analytical and simulated results regarding the consensus latency $L$, and the results are strongly correlated. By putting the latency as a performance metric, the better the consensus reliability, the less the latency.

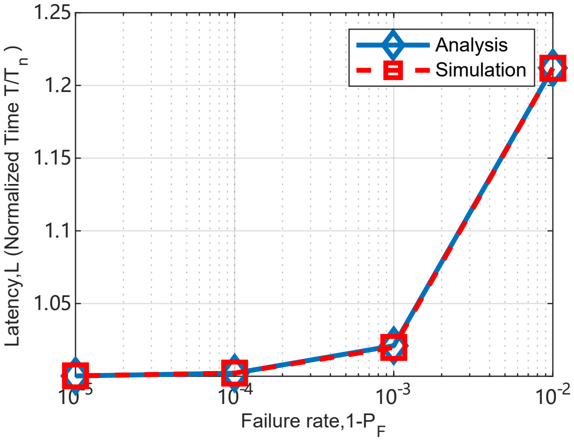

Fig. 8. WDC latency performance to $t_n$

## VII. Conclusions

In this paper, we have investigated a novel concept of using less reliable nodes and communications to power WDC. Starting with the criticism of current C-CAS advancement, we established an innovative model for rapid decision-making regarding the C-CAS requirements and regulated the communication model in conjunction with ISAC potentials. Based on the model established, we provide a detailed analysis of the models with PBFT and Raft consensus regard to their reliability with consideration of node failure and link failure. Meanwhile, we have benchmark results of the latency performance of PBFT and Raft based systems against centralized models. Finally, we investigated the impact of scalability for PBFT/Raft system versus centralized model. The simulations of reliability and latency are provided to validate our model.

In light of the recent development of autonomous driving, wireless critical control system, and many other industrial applications, WDC can help improve efficiency and reliability and enable wireless connectivity in the key fields of future automation. Meanwhile, the search for more efficient evaluations on traffic events with more sensors is projected as a future work for this paper, and the investigation of incentive mechanisms for the WDC-powered C-CAS applications is also planned. In addition to the incentive model, the detailed protocol design has been conducted with multiple access network setups, e.g., consensus over the MAC layer on a Line-of-Sight basis and the consensus group via a local network on the Non-Line-of-Sight basis. Furthermore, the research of blockchain integration is expected to be carried out to bring more features and functionalities making use of hyper reliable consensus. The bespoke blockchain will be studied on its block configurations for best performance.

## Appendix A
## Proof of Theorem 1

To prove Theorem 1, we use equations denoted as $P_{pp}(a,b)$, $P_p(a,b)$, $P_c(a,b)$ to calculate the probability of success of $pre-prepare$ phase, $prepare$ phase and $commit$ phase separately by giving the number of successful nodes $a$ before entering the phase and $b$ after completing the phase. Similarly, the rate of failed nodes in a certain number is calculated in $P_{node}(a,b)$ where $a$ denotes the total number of nodes while $b$ is the number of non-faulty nodes. According to the communication principle shown in Fig. 2 and the assumption of node failure, it is easy to apply binomial distribution method to calculate $P_{pp}(a,b)$ and $P_{node}(a,b)$ by equation (3) and equation (2). As for $P_p(a,b)$ and $P_c(a,b)$, the probability of success of each node should be calculated according to $P_l$ first since the number of broadcast messages each node receives determines whether it can proceed to the next phase. Therefore, by applying binomial distribution with minimum valid messages required, i.e., $2f$ messages (without itself) from different nodes, we have equation (6) to calculate the probability of success of each node in $prepare$ and $commit$ phases. With the probability of success for each node calculated in $prepare$ and $commit$ phase, equation (5) and equation (4) for $P_p(a,b)$ and $P_c(a,b)$ can be regarded as node failure calculation by using binomial distribution as in $P_{node}(a,b)$.

To calculate the probability of successful consensus of a complete $P-N-L$ model with unknown number of success nodes in each phase, we use intermediate notations to replace index $a$ and $b$ in $P_{pp}(a,b)$, $P_p(a,b)$, $P_c(a,b)$ and $P_{node}(a,b)$. $m$ is the number of non-faulty nodes and $m_{pp}$, $m_p$, $m_c$ are used to represent the number of success nodes after $pre-prepare$ phase, $prepare$ phase and $commit$ phase. Therefore, we obtain the rate to complete an entire consensus process with a known number of success nodes in each phase as

$$\begin{aligned} P_{P-N-L_{sub}} = & P_{node}(n,m)\cdot P_{pp}(m,m_{pp})\cdot \\ & P_p(m_{pp},m_p)\cdot P_c(m_p,m_c). \end{aligned} \tag{19}$$

The probability of successful consensus of $P-N-L$ is actually adding all the possible cases of equation (19). To sum up, all the cases that are able to achieve consensus successfully, we have

$$P_{P-N-L} = \sum_{m=m_{pp}}^{n} \sum_{m_{pp}=m_p}^{n} \sum_{m_p=m_c}^{n} \sum_{m_c=n-f}^{n} P_{P-N-L_{sub}}, \tag{20}$$

and the final expansion equation is equation (1).

The core idea of the equation (1) is that if a node fails, it does not participate in the rest of the consensus phase. As we know, failed nodes can be divided into two types. One is caused by node failure, which means the node is unavailable in the entire consensus process, and the other is caused by link failure since the number of messages a node collects in one phase cannot support it entering the next phase. Based on this analysis, we can conclude that, for a successful consensus process, $n \geq m \geq m_{pp} \geq m_p \geq m_c \geq n-f$.

## Appendix B
## Derivation of Raft node link failure model

In Raft node link failure (R-N-L) model, nodes have a reliability of $P_n$ in the log replication stage, and the links have a reliability rate of $P_l$. In this way, in order to reach the final system consensus, we have to ensure that the majority of nodes successfully finish each phase of the log replication. Since the leader node is always regarded as a reliable node in our model, we require that at least $n-1-\lfloor\frac{n-1}{2}\rfloor$, which is also $\lceil\frac{n-1}{2}\rceil$ nodes to respond successfully to the leader node. Therefore, by using the binomial distribution, $P_{R-N-L}$ can be obtained from equation (9).

## References

[1] I. Sarrigiannis, L. M. Contreras, K. Ramantas, A. Antonopoulos, and C. Verikoukis, "Fog-Enabled Scalable C-V2X Architecture for Distributed 5G and Beyond Applications," IEEE Network, vol. 34, no. 5, pp. 120–126, sep 2020.

[2] H. Xu, L. Zhang, E. Sun, and C.-L. I, "BE-RAN: Blockchain-enabled Open RAN with Decentralized Identity Management and Privacy-Preserving Communication," jan 2021. [Online]. Available: http://arxiv.org/abs/2101.10856

[3] C. Savaglio, M. Ganzha, M. Paprzycki, C. Bădică, M. Ivanović, and G. Fortino, "Agent-based internet of things: State-of-the-art and research challenges," Future Generation Computer Systems, vol. 102, pp. 1038–1053, 2020.

[4] K. Wang, Y. Wang, Y. Sun, S. Guo, and J. Wu, "Green Industrial Internet of Things Architecture: An Energy-Efficient Perspective," IEEE Communications Magazine, vol. 54, no. 12, pp. 48–54, dec 2016.

[5] D. Yu, W. Li, H. Xu, and L. Zhang, "Low reliable and low latency communications for mission critical distributed industrial internet of things," IEEE Communications Letters, 2020.

[6] G. Fortino, W. Russo, C. Savaglio, W. Shen, and M. Zhou, "Agent-oriented cooperative smart objects: From iot system design to implementation," IEEE Transactions on Systems, Man, and Cybernetics: Systems, vol. 48, no. 11, pp. 1939–1956, 2017.

[7] I. Moir, A. Seabridge, and M. Jukes, Civil Avionics Systems. Chichester, UK: John Wiley & Sons, Ltd, aug 2013.

[8] RTCA, "DO-178C, Software Considerations in Airborne Systems and Equipment Certification," 2011.

[9] M. Chen, Y. Tian, G. Fortino, J. Zhang, and I. Humar, "Cognitive internet of vehicles," Computer Communications, vol. 120, pp. 58–70, 2018.

[10] P. Popovski, K. F. Trillingsgaard, O. Simeone, and G. Durisi, "5G Wireless Network Slicing for eMBB, URLLC, and mMTC: A Communication-Theoretic View," IEEE Access, vol. 6, pp. 55 765–55 779, 2018.

[11] P. Fan, J. Zhao, and C. I., "5G high mobility wireless communications: Challenges and solutions," China Communications, vol. 13, no. Supplement2, pp. 1–13, 2016.

[12] A. A. Anastasios, F. Nadia, G. Janis, H. Julian, I. Marta, J. Stefan, K.-W. Iwona, M. Manuel, N. Marina, S. Philipp, t. M. Jolanda, and W. Tim, "Convergence of blockchain, AI and IoT," 2019. [Online]. Available: https://www.eublockchainforum.eu/sites/default/files/report_convergence_v1.0.pdf

[13] G. J. Sutton, J. Zeng, R. P. Liu, W. Ni, D. N. Nguyen, B. A. Jayawickrama, X. Huang, M. Abolhasan, Z. Zhang, E. Dutkiewicz, and T. Lv, "Enabling technologies for ultra-reliable and low latency communications: From phy and mac layer perspectives," IEEE Communications Surveys Tutorials, vol. 21, no. 3, pp. 2488–2524, 2019.

[14] C. Feng, Z. Xu, X. Zhu, P. V. Klaine, and L. Zhang, "Wireless distributed consensus in vehicle to vehicle networks for autonomous driving."

[15] T. Taleb, K. Samdanis, B. Mada, H. Flinck, S. Dutta, and D. Sabella, "On multi-access edge computing: A survey of the emerging 5g network edge cloud architecture and orchestration," IEEE Communications Surveys Tutorials, vol. 19, no. 3, pp. 1657–1681, 2017.

[16] W. Li, C. Feng, L. Zhang, H. Xu, B. Cao, and M. A. Imran, "A Scalable Multi-Layer PBFT Consensus for Blockchain," IEEE Transactions on Parallel and Distributed Systems, vol. 32, no. 5, pp. 1146–1160, may 2021.

[17] H.-N. Dai, Z. Zheng, and Y. Zhang, "Blockchain for internet of things: A survey," IEEE Internet of Things Journal, vol. 6, no. 5, pp. 8076–8094, 2019.

[18] Z. Yang, K. Yang, L. Lei, K. Zheng, and V. C. M. Leung, "Blockchain-based decentralized trust management in vehicular networks," IEEE Internet of Things Journal, vol. 6, no. 2, pp. 1495–1505, 2019.

[19] L. Zhang, H. Xu, O. Onireti, M. A. Imran, and B. Cao, "How Much Communication Resource is Needed to Run a Wireless Blockchain Network?" IEEE Network, To appear 2021.

[20] F. Liu, C. Masouros, A. P. Petropulu, H. Griffiths, and L. Hanzo, "Joint Radar and Communication Design: Applications, State-of-the-Art, and the Road Ahead," IEEE Transactions on Communications, vol. 68, no. 6, pp. 3834–3862, jun 2020.

[21] N. Su, F. Liu, and C. Masouros, "Secure Radar-Communication Systems With Malicious Targets: Integrating Radar, Communications and Jamming Functionalities," IEEE Transactions on Wireless Communications, vol. 20, no. 1, pp. 83–95, jan 2021.

[22] M. Castro and B. Liskov, "Practical Byzantine Fault Tolerance," in Proceedings of the Third Symposium on Operating Systems Design and Implementation 1999, ser. OSDI '99. USA: USENIX Association, 1999, pp. 173–186.

[23] D. Ongaro and J. Ousterhout, "In Search of an Understandable Consensus Algorithm," in Proceedings of the 2014 USENIX Annual Technical Conference, USENIX ATC 2014, vol. 22, no. 2, 2014, pp. 305–320.

[24] D. Alistarh, J. Aspnes, V. King, and J. Saia, "Communication-efficient randomized consensus," Distributed Computing, vol. 31, no. 6, pp. 489–501, nov 2018.

[25] A. Asheralieva and D. Niyato, "Reputation-based coalition formation for secure self-organized and scalable sharding in iot blockchains with mobile-edge computing," IEEE Internet of Things Journal, vol. 7, no. 12, pp. 11 830–11 850, 2020.

[26] Y. Liu, K. Wang, Y. Lin, and W. Xu, "LightChain: A lightweight blockchain system for industrial internet of things," IEEE Transactions on Industrial Informatics, vol. 15, no. 6, pp. 3571–3581, 2019.

[27] J. Hu, K. Yang, K. Wang, and K. Zhang, "A blockchain-based reward mechanism for mobile crowdsensing," IEEE Transactions on Computational Social Systems, vol. 7, no. 1, pp. 178–191, 2020.

[28] J. Fu, L. Zhang, L. Wang, and F. Li, "Bct: An efficient and fault tolerance blockchain consensus transform mechanism for iot," IEEE Internet of Things Journal, 2021.

[29] H. Seo, J. Park, M. Bennis, and W. Choi, "Communication and consensus co-design for distributed, low-latency, and reliable wireless systems," IEEE Internet of Things Journal, vol. 8, no. 1, pp. 129–143, 2020.

[30] 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 of Things Journal, vol. 6, no. 3, pp. 5791–5802, 2019.

[31] E. Shi and A. Perrig, "Designing secure sensor networks," IEEE Wireless Communications, vol. 11, no. 6, pp. 38–43, 2004.

[32] M. A. Ferrag and L. Shu, "The performance evaluation of blockchain-based security and privacy systems for the internet of things: A tutorial," IEEE Internet of Things Journal, vol. 8, no. 24, pp. 17 236–17 260, 2021.

[33] K. Peng, M. Li, H. Huang, C. Wang, S. Wan, and K.-K. R. Choo, "Security challenges and opportunities for smart contracts in internet of things: A survey," IEEE Internet of Things Journal, vol. 8, no. 15, pp. 12 004–12 020, 2021.

[34] S. Glaser, B. Vanholme, S. Mammar, D. Gruyer, and L. Nouvelière, "Maneuver-based trajectory planning for highly autonomous vehicles on real road with traffic and driver interaction," IEEE Transactions on Intelligent Transportation Systems, vol. 11, no. 3, pp. 589–606, 2010.

[35] B. Paden, M. Čáp, S. Z. Yong, D. Yershov, and E. Frazzoli, "A survey of motion planning and control techniques for self-driving urban vehicles," IEEE Transactions on Intelligent Vehicles, vol. 1, no. 1, pp. 33–55, 2016.

[36] E. Uhlemann, "Introducing connected vehicles [connected vehicles]," IEEE Vehicular Technology Magazine, vol. 10, no. 1, pp. 23–31, 2015.

[37] J. Kang, R. Yu, X. Huang, M. Wu, S. Maharjan, S. Xie, and Y. Zhang, "Blockchain for secure and efficient data sharing in vehicular edge computing and networks," IEEE Internet of Things Journal, vol. 6, no. 3, pp. 4660–4670, 2019.

[38] E. Shi, Foundations of Distributed Consensus and Blockchains, 2020. [Online]. Available: http://elaineshi.com/docs/blockchain-book.pdf

[39] R. P. G. Collinson, Fly-by-Wire Flight Control. Boston, MA: Springer US, 2003, pp. 159–224. [Online]. Available: https://doi.org/10.1007/978-1-4419-7466-2_4

[40] Dinh-Khanh Dang, A. Mifdaoui, and T. Gayraud, "Fly-by-wireless for next generation aircraft: Challenges and potential solutions," in 2012 IFIP Wireless Days, 2012, pp. 1–8.

[41] A. Perez, "Quality of Service Principles," in Implementing IP and Ethernet on the 4G Mobile Network. Elsevier, jan 2017, pp. 117–135.

[42] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," Tech. Rep., 2008. [Online]. Available: http://www.bitcoin.org/bitcoin.pdf

[43] S. O'Dea, "Failure rate of Apple iPhone worldwide by model 2017-2018," June 2020. [Online]. Available: https://www.statista.com/statistics/804359/iphone-failure-rate-by-model-worldwide/