## ***Disclaimer***

The present document has been produced and approved by the <long ISGname> (<short ISGname>) ETSI Industry Specification Group (ISG) and represents the views of those members who participated in this ISG.  
It does not necessarily represent the views of the entire ETSI membership.

ETSI GR ISG-PDL 020 V0.0.4 (2023-01)

**Group REPORT**

PDL Services for Wireless Consensus Network

<

Reference

PDL-020

Keywords

PDL, Consensus, Wireless network

***ETSI***

650 Route des Lucioles

F-06921 Sophia Antipolis Cedex - FRANCE

Tel.: +33 4 92 94 42 00 Fax: +33 4 93 65 47 16

Siret N° 348 623 562 00017 - APE 7112B

Association à but non lucratif enregistrée à la

Sous-préfecture de Grasse (06) N° w061004871

***Important notice***

The present document can be downloaded from:  
<http://www.etsi.org/standards-search>

The present document may be made available in electronic versions and/or in print. The content of any electronic and/or print versions of the present document shall not be modified without the prior written authorization of ETSI. In case of any existing or perceived difference in contents between such versions and/or in print, the prevailing version of an ETSI deliverable is the one made publicly available in PDF format at [www.etsi.org/deliver](http://www.etsi.org/deliver).

Users of the present document should be aware that the document may be subject to revision or change of status. Information on the current status of this and other ETSI documents is available at <https://portal.etsi.org/TB/ETSIDeliverableStatus.aspx>

If you find errors in the present document, please send your comment to one of the following services:  
<https://portal.etsi.org/People/CommiteeSupportStaff.aspx>

If you find a security vulnerability in the present document, please report it through our

Coordinated Vulnerability Disclosure Program:

<https://www.etsi.org/standards/coordinated-vulnerability-disclosure>

***Notice of disclaimer & limitation of liability***

The information provided in the present deliverable is directed solely to professionals who have the appropriate degree of experience to understand and interpret its content in accordance with generally accepted engineering or

other professional standard and applicable regulations.

No recommendation as to products and services or vendors is made or should be implied.

No representation or warranty is made that this deliverable is technically accurate or sufficient or conforms to any law and/or governmental rule and/or regulation and further, no representation or warranty is made of merchantability or fitness for any particular purpose or against infringement of intellectual property rights.

In no event shall ETSI be held liable for loss of profits or any other incidental or consequential damages.

Any software contained in this deliverable is provided "AS IS" with no warranties, express or implied, including but not limited to, the warranties of merchantability, fitness for a particular purpose and non-infringement of intellectual property rights and ETSI shall not be held liable in any event for any damages whatsoever (including, without limitation, damages for loss of profits, business interruption, loss of information, or any other pecuniary loss) arising out of or related to the use of or inability to use the software.

***Copyright Notification***

No part may be reproduced or utilized in any form or by any means, electronic or mechanical, including photocopying and microfilm except as authorized by written permission of ETSI.  
The content of the PDF version shall not be modified without the written authorization of ETSI.  
The copyright and the foregoing restriction extend to reproduction in all media.

© ETSI 2022.

All rights reserved.

Contents

Intellectual Property Rights 5

Foreword 5

Modal verbs terminology 5

Executive summary 5

Introduction 5

1 Scope 7

2 References 7

2.1 Normative references 7

2.2 Informative references 7

3 Definition of terms, symbols and abbreviations 7

3.1 Terms 7

3.2 Symbols 7

3.3 Abbreviations 7

4 Overview of Wireless Consensus Networks 9

4.1 Need for Wireless Consensus Network 9

4.1.1 General problem statement 9

4.1.2 Consensus for distributed automation 10

4.2 Motivations 11

5 Opportunities and Use Cases of Wireless Consensus Network 11

5.1 Opportunities 11

5.1.1 Background 11

5.1.2 Centralized 11

5.1.3 Decentralized 12

5.2 Use Case Background 12

5.3 Use case 1: Autonomous vehicle 13

5.3.1 Collision avoidance and advisory (clustering decision) 13

5.3.2 X-by-wireless (wireless communication for mission-critical control) 13

5.4 Use case 2: Industrial IoT 14

5.4.1 Operation synchronization 14

5.4.2 Data service 15

6 Functionalities and Considerations for Wireless Consensus Network Framework 15

6.1 Background 15

6.2 WCN Framework 15

6.2.1 Access network based WCN framework 15

6.2.2 Self-organizing WCN framework 16

6.3 Functionalities and Considerations 16

6.3.1 Membership management (network peer arrangement) 16

6.3.2 Reliability management 16

6.3.3 Situational awareness improvement 16

7 Hardware Definition 16

7.1 Hardware requirement 16

7.1.1 Processing capability for consensus 16

7.1.2 Communication capability 17

7.1.3 Storage and power supply 17

7.2 Hardware security and threats 17

8 Consensus Mechanism 17

8.1 Proof based consensus 18

8.1.1 Proof of Work 18

8.1.2 Proof of Stake 19

8.1.3 Proof of Authority 19

8.1.4 Other proof-based consensus mechanisms 20

8.2 Voting based consensus 20

8.2.1 PBFT 20

8.2.2 Raft 20

8.3 Performance metrics 21

8.3.1 Security Bound 22

8.3.2 Node Scalability 22

8.3.3 Transaction Throughput and Latency 22

9 Protocol for Wireless Consensus Network 22

9.1 Background 22

9.2 Protocol description 23

9.2.1 Number of nodes 23

9.2.2 Node state of consensus 23

9.2.3 Leader election 24

9.2.4 Log replication 24

9.2.5 Rules for node 25

9.3 Routing and synchronization 25

9.4 On-boarding and withdrawal of nodes 26

10 Conclusion and Recommendation 27

Annex A: Title of annex 28

Annex B: Title of annex 29

B.1 First clause of the annex 29

B.1.1 First subdivided clause of the annex 29

Annex: Bibliography 30

Annex : Change History 31

History 32

# Intellectual Property Rights

Essential patents

IPRs essential or potentially essential to normative deliverables may have been declared to ETSI. The declarations pertaining to these essential IPRs, if any, are publicly available for **ETSI members and non-members**, and can be found in ETSI SR 000 314: *"Intellectual Property Rights (IPRs); Essential, or potentially Essential, IPRs notified to ETSI in respect of ETSI standards"*, which is available from the ETSI Secretariat. Latest updates are available on the ETSI Web server ([https://ipr.etsi.org](https://ipr.etsi.org/)).

Pursuant to the ETSI Directives including the ETSI IPR Policy, no investigation regarding the essentiality of IPRs, including IPR searches, has been carried out by ETSI. No guarantee can be given as to the existence of other IPRs not referenced in ETSI SR 000 314 (or the updates on the ETSI Web server) which are, or may be, or may become, essential to the present document.

Trademarks

The present document may include trademarks and/or tradenames which are asserted and/or registered by their owners. ETSI claims no ownership of these except for any which are indicated as being the property of ETSI, and conveys no right to use or reproduce any trademark and/or tradename. Mention of those trademarks in the present document does not constitute an endorsement by ETSI of products, services or organizations associated with those trademarks.

**DECT™**, **PLUGTESTS™**, **UMTS™** and the ETSI logo are trademarks of ETSI registered for the benefit of its Members. **3GPP™**and **LTE™** are trademarks of ETSI registered for the benefit of its Members and of the 3GPP Organizational Partners. **oneM2M™** logo is a trademark of ETSI registered for the benefit of its Members and of the oneM2M Partners. **GSM**® and the GSM logo are trademarks registered and owned by the GSM Association.

# Foreword

This Group Report (GR) has been produced by ETSI Industry Specification Group Permissioned Distributed Ledger (PDL).

# Modal verbs terminology

In the present document "**should**", "**should not**", "**may**", "**need not**", "**will**", "**will not**", "**can**" and "**cannot**" are to be interpreted as described in clause 3.2 of the [ETSI Drafting Rules](https://portal.etsi.org/Services/editHelp!/Howtostart/ETSIDraftingRules.aspx) (Verbal forms for the expression of provisions).

"**must**" and "**must not**" are **NOT** allowed in ETSI deliverables except when used in direct citation.

# Executive summary

The Group Report presents the fundamentals and potential applications of decentralised identification that can benefit various public and private services. Further the group report also discusses a set of PDL services that can together enable a PDL based Wireless Consensus Network framework.

# Introduction

As a fundamental of PDL, consensus is critical to update ledgers with new transactions and ensure ledgers are synchronised and consistent. Current studies related to PDL and consensus have not considered the network infrastructure (i.e., wired or wireless) and assume network communications is reliable and error-free [ref placeholder]. However, in practical terms communication errors may occur during consensus process as a result of network infrastructure conditions especially when wireless networks are in use. Wireless networks are less stable and less reliable than wired networks due to interferences and obstacles in space. Meanwhile, compared with wired networks, wireless networks can be more dynamic since wireless nodes (such as mobile devices) can join or leave a network without the need for physical connections or disconnection of devices. Therefore, the use of wireless consensus networks (WCNs) for consensus between nodes (which can be a mix of mobile and static devices) could pose challenges. This study introduces an overview of wireless consensus network approaches that can offer benefits to certain services. Various factors such as the requirements and architectures of WCNs, consensus mechanisms, hardware, protocols used to realise WCNs are analysed. In addition, this study also demonstrates some use cases based on WCNs.

A consensus network is used to ensure a consensus on content of data among nodes in a distributed system exists or to reach an agreement on a proposal. It is fault tolerant, scalable, secure, democratic, and privacy-preserving to serve as an auditable tool in scenarios where data integrity should be preserved and recorded (e.g., when investigating events related to autonomous driving). Furthermore, a consensus network is also the backbone technique of distributed systems such as PDL. This document discusses the challenges of maintaining sufficient quality of the above metrics when the consensus network is operated over fully or partially wireless infrastructure, hence becoming a WCN.

# 1 Scope

The present document investigates wireless consensus network related to the following aspects:

* Use cases of wireless consensus networks
* Wireless consensus network architecture
* Ways to construct wireless consensus networks
  + MAC and physical layers
  + Decentralised/Centralised communication
* Performance metrics of consensus mechanisms
* Protocols to construct wireless consensus networks

# 2 References

## 2.1 Normative references

Normative references are not applicable in the present document.

## 2.2 Informative references

References are either specific (identified by date of publication and/or edition number or version number) or non‑specific. For specific references, only the cited version applies. For non-specific references, the latest version of the referenced document (including any amendments) applies.

NOTE: While any hyperlinks included in this clause were valid at the time of publication, ETSI cannot guarantee their long term validity.

The following referenced documents are not necessary for the application of the present document but they assist the user with regard to a particular subject area.

[i.1] <Standard Organization acronym> <document number><version number/date of publication>: "<Title>".

[i.2] etc.

# 3 Definition of terms, symbols and abbreviations

## 3.1 Terms

For the purposes of the present document, the [following] terms [given in ... and the following] apply:

## 3.2 Symbols

For the purposes of the present document, the [following] symbols [given in ... and the following] apply:

## 3.3 Abbreviations

For the purposes of the present document, the [following] abbreviations [given in ... and the following] apply:

PDL Permissioned Distributed Ledger

WCN Wireless Consensus Network

CMs Consensus Mechanism(s)

PBFT Practical Byzantine Fault Tolerance

PoW Proof of Work

PoS Proof of Stake

CSMA/CA Carrier-sense Multiple Access with Collision Avoidance

CSMA/CD Carrier-sense Multiple Access with Collision Detection

SPOF Single Point of Failure

# 4 Overview of Wireless Consensus Networks

Permissioned distributed ledger (PDL) is built on a decentralised network that relies on frequent communications among distributed nodes. Compared with centralised data records as presented in Table I, PDL is more receptive to enabling numerous participants to share data in an autonomous manner. The consensus mechanisms (CMs), which plays a pivotal role in PDL, is resource-demanding both in terms of calculations and in terms of communication overheads. The CMs would often determine security requirements (i.e., fault tolerances) and other key performance metrics such as transaction throughput, latency thresholds and scalability to achieve the data consistency required for proper PDL functions.

**TABLE I.** Comparison of centralized and decentralized

|  |  |  |
| --- | --- | --- |
| **Property** | **Centralized** | **Decentralized** |
| Meaning | The retention of powers and authority with respect to planning and decisions, with the top management, is known as centralized or centralization. | The dissemination of authority, responsibility and accountability to the various management levels, is known as decentralized or decentralization. |
| Geographical Distribution | Located at a centralized location (with possible mirrors/replication). | Geographically distributed. |
| Node Ownership | All nodes are owned by a single entity. | Each node is owned by a different entity. |
| Involves | Systematic and consistent reservation of authority. | Systematic dispersal of authority. |
| Communication | Vertical | Open and Free |
| Decision Making | Made by single entity fast but cannot prevent SPOF | Comparatively faster for reaching consensus in large groups by all participants to prevent SPOF |
| Advantage | Proper coordination and Leadership | Sharing of burden and responsibility |
| Power of decision making | Lies with the top management. | Multiple participants have the power of decision making. |
| Best suited for | Small-sized networks/organizations  Data that is owned by a single entity. | Large-sized networks/organizations.  Data that is shared between multiple entities. |
| Authority | Single entity | All participants |

Most PDL systems are considered and designed in a stable wired communication network connecting advanced devices under the assumption of sufficient communication resource availability and quality. However, in reality a growing number of PDL node peers are connected through wireless networks turning them into wireless consensus networks (WCN) [ref placeholder]. Constrained by the unpredictable behaviour of wireless channels and frequency spectrum limitations, communications can significantly affect the key performance metrics of WCN. Moreover, wired communications systems can quickly detect transmission failure, while wireless systems may not be able to detect faults as quickly. In wireless systems, transmission failures are not sensed by the transmitters and receivers. Wireless nodes can only sense if the channel is occupied during transmissions, and back-off for a random period to avoid collisions using with methods such as CSMA/CA. The transceiver has no knowledge if the frame has been transmitted or received. On the other hand, wired systems can detect transmission failures easily with collision detections techniques, such as, CSMA/CD. Hence, in this study, we investigate consensus mechanisms and protocols that can be potentially used in WCN in the future as well as the suggested requirements and use cases of WCN.

## 4.1 Need for Wireless Consensus Networks

### 4.1.1 General problem statement

Driven by advances in 5G, industry 4.0, cloud/edge computing and artificial intelligence, 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. Meanwhile, more and more mobile devices and applications are emerging to serve people in their daily tasks such as wearables and autonomous driving. These vital societal and industrial functions are increasingly interconnected for information exchange through communication networks to complete joint tasks. Since it is infeasible to rely on wired networks to enable such mobile devices to communicate, achieving consensus in open wireless channels involving mobile devices should be further investigated.

[possible for future use][In such systems, data from mobile devices may be collected from distributed, heterogeneous and trustless sensors located in different places to determine common and critical real-time decisions in order to achieve cooperative actions over the air.]

### 4.1.2 Consensus for distributed automation

Consensus for distributed automation is best demonstrated through a use case of autonomous vehicles. Considering Table II below autonomous vehicles are currently at L2 of Autonomy heading towards L3 and further based on a framework defined by the Society of Automotive Engineers (SAE) as shown in Table II [ref placeholder]. Current autonomous vehicles detect other vehicles by identifying them as obstacles, which is not optimal in terms of safety and efficiency. One step forward is that all driver-less vehicles are connected, communicating with each other, knowing each other’s intention in advance, and jointly reach optimal decisions in a cooperative manner. However, existing solutions are centralised, with limited availability and challenging trustworthiness, reliability, scalability, privacy and security.

Compared with centralised solutions, PDLs could be a promising technical route for a distributed scenario such as connected autonomous vehicles. It requires solutions that are fault-tolerant, scalable, ultra-reliable, flexible, democratic and privacy-preserving, operated over a wireless network. Therefore, a WCN that meets the above requirements can serve as an enabling technology to bring the autonomous driving to reality.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Table II.** SAE (J3016) Automation Levels [ref placeholder] | | | | | | | |
| **SAE Level** | **Name** | **Narrative definition** | | **Execution of steering and acceleration/ deceleration** | **Monitoring of driving environment** | **Fallback performance of dynamic driving task** | **System capability (driving modes)** |
| ***Human driver monitors the driving environment*** | | | | | | | |
| 0 | No Automation | The full-time performance by the human driver of all aspects of the dynamic driving task, even when “enhanced by warning or intervention systems” | | Human driver | Human driver | Human driver | N/A |
| 1 | Driver Assistance | The driving mode-specific execution by a driver assistance system of either steering or acceleration/deceleration | Using information about the driving environment and with the expectation that the human driver performs all remaining aspects of the dynamic driving task | Human driver and system | Some driving modes |
| 2 | Partial Automation | The driving mode-specific execution by one or more driver assistance systems of both steering and acceleration/deceleration | System |
| ***Automated driving system monitors the driving environment*** | | | | | | | |
| 3 | Conditional Automation | The driving mode-specific performance by an automated driving system of all aspects of the dynamic driving task | With the expectation that the human driver will respond appropriately to a request to intervene | System | System | Human driver | Some driving modes |
| 4 | High Automation | Even if a human driver does not respond appropriately to a request to intervene the car can pull over safely by guiding system | System | Many driving modes |
| 5 | Full Automation | Under all roadway and environmental conditions that can be managed by a human driver | All driving modes |

## 4.2 Motivations

Although wireless networks have been widely deployed based on various protocols and standards to meet different scenarios, WCN introduces a process of reaching consensus among mobile nodes that may bring additional requirements to not only the network architecture and hardware, but also the applied CMs and protocols. Therefore, such a new paradigm should be analysed by stakeholders to clarify its suggested requirements and potential use cases in the future. Motivated by this, we investigate in this study the suggested requirements and use cases of WCN for PDL and ETSI members in terms of architecture, hardware, consensus mechanism and protocol to define an overall perception of how to practically construct WCNs and what the key components of WCNs should be.

Wired vs wireless unique features/challenges/benefits

List a table for the comparison

A small subsection for each core part to introduce the features of wireless

# 5 Opportunities and Use Cases of Wireless Consensus Network

## 5.1 Opportunities

### 5.1.1 Background

Recently, there is an increased use of IoT devices in critical applications, such as industrial environments and intelligent transportation systems (ITS) in order to aid these processes to make critical real-time decisions [ref placeholder]. For example, today inside a car there are around 60 to 100 sensors that collect data and help the driver or an autonomous system to make decisions [ref placeholder]. However, despite all their benefits in terms of driver safety and information provided, currently the devices inside a car only aid in local decision making, or in other words, other vehicles may not be aware of a specific vehicle’s decision. In addition, sensors are also prone to fail, which can lead to unintended actions. As such, local decision making and sensor faults may have negative impacts on driving safety, especially in autonomous transportation systems where human lives can be lost, as local decisions taken by different cars or false sensor readings can be conflicting and may lead to accidents. For example, in a fatal crash of an autonomous car, in which a car’s sensor failed to recognize a large truck and trailer crossing the highway, leading the vehicle to collide with the truck [ref placeholder]. Thus, in order to overcome these issues, vehicle to vehicle (V2V) networks, or a broader concept of vehicle to everything (V2X) networks were introduced, in which communication networks, such as cellular networks, can be used to exchange reliable information provided by PDL between vehicles as well as infrastructures to improve the decision making leading to safer driving [ref placeholder]. In this context, V2X communications can be implemented in two distinct manners, either centralized or distributed. The choice between the two structures depends on the use cases and requirements as discussed later in this section.

### 5.1.2 Centralized

In centralized approaches, vehicles send their collected data to a central server, such as a base station, which is then responsible for making decisions. These decisions are then sent back to the vehicles, which act accordingly. The centralized communication and decision approach is typically 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. This is named as Perception-Collection-Decision-Action (PCDA) scheme [ref placeholder].

With the continuous growth of IoT devices and connected vehicles, centralized approaches are expected to serve more and more autonomous cars in the near future. Although centralized systems are simpler and bring more control over the decisions, it comes with its disadvantages [ref placeholder].

* Single point of failure (SPOF)
* Computational/processing overhead
* Data pollution
* Data delivery latency in time-sensitive scenarios such as mobility
* Network congestion causing data delay or loss

Moreover, due to the critical nature of V2X communications and the vehicle’s speed, ultra-reliable and low latency communications (URLLC) proposed in 5G is often required in order to meet the stringent constraints posed in V2X. However, in centralized systems, since the vehicles need to send the information to a central authority, the performance in terms of a system’s reliability and latency will be limited by the node with the worst connection to the server. This can result in parameters such as latency and reliability falling below (or above, in case of latency) expected values, or even in complete link failures, leading to asynchronization between vehicles, potentially resulting in accidents and the loss of human lives [ref placeholder].

### 5.1.3 Decentralized

Another approach for V2X communications would be adopting a decentralized and distributed approach, in which vehicles share information to one another and then make decisions jointly, instead of relying on a central authority. However, despite distributed solutions solving the issues faced by centralization, it still faces some challenges. For example,

* Communication link reliability, especially in wireless communication environment [ref placeholder]
* Asynchronization in information sharing
* Trust/Authentication among participants

Besides, vehicles may make decisions based on incorrect sensor readings. In this situation, if asynchronization occurs in decentralized systems, nodes can send conflicting information to each other causing some vehicles to rethink their decisions while others might have already taken actions. Thus, despite decentralization being a good approach, its performance and robustness should be further considered and improved.

In this regard, distributed consensus networks (DCNs) can be a potential alternative to be combined with decentralized V2X systems. By implementing DCNs, a central server is replaced with a distributed ledger, which is not controlled by any single party. Two well-known types of consensuses are crash fault tolerance (only tolerating communication or node failure) and Byzantine tolerance (also tolerating malicious attacks) [ref placeholder]. However, despite its advantages, the consensus performance can also be a bottleneck in V2X systems, since it can significantly be affected by the performance of the wireless communication network, especially in terms of latency, reliability and throughput [ref placeholder]. In participial, the well-known practical Byzantine fault tolerance (PBFT) based consensus, is very simple to implement compared with the PoW-based consensus, but only applicable to small scale of consensus networks since it is very communication resource demanding [ref placeholder]. In addition, unlike wired systems, wireless systems bring extra channel uncertainty, scarcity of spectrum provision, thus entailing different security thresholds. In particular, the PBFT 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 some links connected with the node might be unstable. Moreover, traditional PBFT algorithms do not consider negative votes for a transaction, only abstentions are available, thus few but critical objections may be overlooked. Thus, there is a need to adapt existing consensus mechanisms to wireless environments.

## 5.2 Use Case Background

Distributed ledgers have become one of the most distinctive applications stemming from blockchain. Their ability to store any kind of data as a consensus of replicated, shared, and synchronized digital records distributed across multiple sites, without depending on any central administrator, together with their properties regarding immutability (and therefore non-repudiation) and multi-party verifiability opens a wide range of applications, and new interaction models among those entities willing to record the transactions associated to those interactions through these ledgers. PDL requires nodes to be approved to validate the transactions and record them on the ledger. Therefore, PDL are best qualified to address most of the use cases of interest to the industry and governmental institutions due to the considerations from both technical and legal aspects. As for the technical side, the cost of transaction and consensus, and the fairness properties among participants can be controlled. In addition, the legal matters include the support from external legal agreements or the regulatory enforcement in critical sectors. However, the construction of networks to achieve consensus for PDL in wireless environments still needs further research. Therefore, we introduce two use cases of achieving consensus for PDL in wireless environments as follows to facilitate the study of WCN.

## 5.3 Use case 1: Autonomous vehicle

In this part, we present two scenarios in autonomous vehicle, in which real-time high-reliability WCNs are strongly needed in the presence of the failure nodes to ensure the PDL can reach consensus.

### 5.3.1 Collision avoidance and advisory (clustering decision)

The revolution of automotive industries brings autonomous vehicles to public, with great risks in its early state [ref placeholder]. Many catastrophic failures happened due to sensor errors, malicious attacks and AI decision errors [ref placeholders]. 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.

Modern transportation has regulated collision advisory (CA) to provide traffic, and resolution advice. For example, Traffic Collision Avoidance Systems (TCAS) [ref placeholder] are widely used in aviation, 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 [ref placeholder]. Recent traffic accidents caused by self-driving false alarms and miss alarms have caused multiple catastrophic consequences for road users across the world [ref placeholder]. 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 minimised.

Fig. 1 presents an example of using WCNs in autonomous vehicle. It is notable that the motorbike drives in the blind area of the truck. When the truck needs to join the right lane, the collision may occur if there is no assistance from other three cars to check the right lane. If the truck and three cars can construct a WCN, this WCN can reach a consensus to the occupation situation of the right lane to decline the request of joining the right lane from the truck. Meanwhile, such status information of road/lane occupation can be recorded in PDL for all vehicles nearby.

图形用户界面, 应用程序

描述已自动生成

**Fig. 1.** Wireless distributed consensus for traffic decision.

### 5.3.2 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 [ref placeholder] and Drive-by-wire using internal databus (fieldbus) [ref placeholder], e.g., ARINC 629, ARINC659 (SAFEbus), ARINC 664 Part 4 (AFDX), CAN bus, etc. However, current 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 WCN, 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 [ref placeholder], 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, the reliability is secured by employing duplicates in the system using First-in-First-out (FIFO) queue, which does not take the Byzantine fault into consideration, since the physical network is isolated from outside. However, for wireless critical control, Byzantine faults should be considered due to the open and unstable channels for communications. Therefore, WCN could be a promising solution to enable nodes to achieve consensus to the transmitted data in the wireless network.

## 5.4 Use case 2: Industrial IoT

The industrial internet of things (IIoT) refers to interconnected sensors, instruments, and other devices networked together with industrial applications using computers, including manufacturing and energy management. In general, an IIoT system is conceived as a 5-layered architecture of involving sensors, user devices, communication components, applications, etc. as shown in Table III. The device layer refers to the physical components: cyber-physical systems, sensors or machines. The network layer consists of physical network buses, cloud computing and communication protocols that aggregate and transport the data to the service layer, which consists of applications that manipulate and combine data into information that can be displayed on the user dashboard. The top of the IIoT architecture is the content layer or the user interface to provide users with information they require.

**TABLE III.** Layered Architecture of IIoT

|  |  |
| --- | --- |
| Content layer | User interface devices e.g., computer screens, PoS stations, tablets, smart glasses and smart surfaces |
| Service layer | Applications e.g., software to analyze data and transform it into actionable information |
| Network layer | Communications protocols e.g., [Wi-Fi](https://en.wikipedia.org/wiki/Wi-Fi" \o "Wi-Fi), [Bluetooth](https://en.wikipedia.org/wiki/Bluetooth" \o "Bluetooth), [LoRa](https://en.wikipedia.org/wiki/LoRa" \o "LoRa) and [cellular](https://en.wikipedia.org/wiki/Cellular_network" \o "Cellular network) |
| Device layer | Hardware e.g., cyber-physical systems, machines and sensors |

This connectivity enables data collection, exchange, and analysis, potentially facilitating improvements in productivity and efficiency as well as other economic benefits [ref placeholder]. The IIoT is an evolution of a distributed control system that allows for a higher degree of automation by using cloud computing to refine and optimize the process controls.

### 5.4.1 Operation synchronization

In an IIoT system, multiple machines and sensors may work together in manufacturing by receiving operational commands from the operator (user). For example, to fill certain hazardous chemicals, chemical plants use filling machines for batch filling automatically and sensors to monitor the environment to detect leakage and send alarms instead of workers. All the machines and sensors are teleoperated via wireless communications since some chemicals may be corrosive to cables. Another example is mining radioactive minerals underground. Drills and conveyor belts are also teleoperated and sensors can be deployed to monitor the temperature of drills and the running status of belts. However, radioactive minerals may affect wireless and even wired communications for the teleoperation. Therefore, in these scenarios, how to ensure received commands from operators are correct is critical to the IIoT system.

Employing WCNs could be a solution to enable sensors and machine to cross-check the received commands and reach a consensus for them in a distributed and autonomous manner. For the sensors and machines in the same category, when they receive commands from the operator, they can self-organize a WCN to cross-check and synchronize the received commands in case the received commands of some devices are distorted due to unstable communication channels. After the consensus is reached for the received commands, the devices receiving different (distorted) commands can synchronize the consensus commands from other devices without requesting the operator.

### 5.4.2 Data service

By employing WCNs in IIoT systems, PDL can be built up to record the consensus results of the received commands for operators to check. Furthermore, running status data collected by machines, sensors, etc. can be recorded in the PDL by using consensus mechanisms to avoid false alarms and provide a more robust data service for operators. For example, when a sensor detects an abnormal status, it can organize a consensus voting with other related sensors and devices via the WCN. If the participated devices can achieve a consensus for the abnormal status, the organizer can send an alarm to the operator. If the detected abnormal status is a false alarm (e.g., caused by environmental factors such as radioactivity and vibrations), the consensus voting process based on the WCN can identify the false alarm.

# 6 Functionalities and Considerations for Wireless Consensus Network Framework

### 6.1 Background

WCN can be a backbone of PDL to achieve consensus for the information recorded in PDL. The potential applications of WCNs are introduced in the last section. However, the (general) fundamental framework of WCN and the functions and considerations in constructing WCNs should be further discussed. Overall, there are two points to be considered for WCNs: communication and consensus. Therefore, the following WCN framework, functions and considerations are illustrated around the two points.

### 6.2 WCN Framework

#### 6.2.1 Access network based WCN framework

The first type of WCN framework is constructed on access networks as shown in Fig. 2. Four nodes can communicate with others via wireless access networks such as cellular networks (4G, 5G, etc.) and Wi-Fi networks. When a node sends a consensus request to other nodes via the access network, all nodes can start the consensus mechanism via communications among nodes. Therefore, the consensus can be reached in this WCN framework. Furthermore, all the nodes in the WCN can act as PDL nodes to store the consensus results (transactions) and maintain the PDL together. For the access network based WCN, the following components can be involved.

* Consensus node (PDL node): can perform consensus mechanisms and maintain PDL with the computational and communication resources.
* Access point: can accept connections from consensus nodes to allow them joining the same network.
* Wireless communication infrastructure: is used to support communications among consensus nodes.
* Membership service provider: should issue memberships for consensus nodes and verify the memberships of consensus nodes when they try to connect access points.
* Storage: should be provided for each consensus node to ensure it can keep consensus status and PDL transactions.
* Power supply: is needed in each consensus node to provide power for computation, communication and storage hardware such as batteries and PSU (power supply unit to obtain power from wired grids).

图示

中度可信度描述已自动生成

**Fig 2.** WCN framework based on access networks.

#### 6.2.2 Self-organizing WCN framework

The second type of WCN frame is self-organizing i.e., all consensus nodes need to establish connections with other nodes to communicate for reaching consensus by themselves. Compared with the former WCN framework, there is no access point or wireless communication infrastructure to support the communications among consensus nodes as shown in Fig. 3. Therefore, consensus nodes should have communication hardware to self-organize WCNs. Multiple consensus nodes construct the self-organizing WCN to perform consensus mechanisms and maintain PDL. In the self-organizing WCN, each node should have the following components.

* Computational resource: is used to process consensus data and communication packets.
* Communication resource: should be involved in the consensus node such as wireless communication hardware to enable it to join and organize WCN with other consensus nodes.
* Storage: should be included in each consensus node for keeping consensus status and PDL transactions such as memory and flash disk.
* Power supply: is needed in each consensus node to provide power for computation, communication and storage hardware such as batteries and PSU (power supply unit to obtain power from wired grids).

图片包含 文本

描述已自动生成

**Fig 3.** WCN framework based on self-organizing networks.

### 6.3 Functionalities and Considerations

#### 6.3.1 Membership management (network peer arrangement)

* Node join

A WCN can allow new nodes to join in it. Meanwhile, all other existed nodes should be aware of the new joined nodes. Furthermore, after the new nodes join the WCN, all nodes should process the upcoming consensus tasks together. After new nodes join the WCN, the leader should update the quorum setting depending on the new number of nodes in the WCN for upcoming consensus tasks.

* Node quit

A WCN can accept existed nodes to leave it. Meanwhile, all other existed nodes should be aware of the left nodes. Furthermore, after some nodes leave the WCN, all other nodes should process the upcoming consensus tasks together. After some nodes leave the WCN, the leader should update the quorum setting depending on the new number of nodes in the WCN for upcoming consensus tasks.

* Faulty node detection

Some strategies should be involved in WCN communication to detect and manage faulty nodes caused by failed hardware, signal loss, etc. If some faulty nodes are detected, the leader should exclude them in consensus tasks. On the other hand, if faulty nodes recover to work normally, they should notify the leader. If the leader confirms these faulty nodes can work normally after check, these nodes can be involved in the upcoming consensus tasks by the leader.

* Leader change

If the leader node leaves the WCN or becomes a faulty node, the WCN is able to initialize a consensus task to elect a new leader for the WCN. The ongoing consensus tasks should be paused during changing the leader. Then, when the leader change is complete, the paused tasks continue (or is restarted by the new leader). If the leader change fails, the WCN should attempt the leader change for a few times depending on the strategies set in the WCN. In addition, such failure should be recorded in the PDL on reachable nodes in the WCN.

* Access control (identity)
  + Accessed network based WCN
    - Issue: when a node applies to join a WCN, the membership service provider should issue a legal identity for a node to join the WCN via the accessed network in a period.
    - Withdraw: After the identity of the node expired, the membership service provider should notify the access points that the identity has been withdrawn and the identity should be declined by the access points.
    - Renew: If the node applies to renew its identity, the membership service provider should extend the valid period of the identity. The updated identity should be accepted by the access points.
  + Self-organizing WCN
    - Basic identity: In a self-organizing WCN, as it does not have the fundamental to realize authentication or authorization, each node can only possess a basic identity such as a hash ID to represent itself in the self-organizing WCN.
* Requirements of access control methods
  + Encryption
  + Identity verifiability
  + Lightweight (lightweight/more required features…)

#### 6.3.2 Reliability management

The WCN system should prevent the unconverging communication loop by making the communication process a self-converged loop. By preventing open-loop or unconverging communication outcome, the system should attempt to re-transmit messages within the timeout setting of the consensus. In the case of timeout, the system should trigger heartbeat signals and conduct leader rotations or re-elections. In the case of leader change fails, the system should have more attempts on leader changes, and produce warning/error messages to system control panel, participants messages interfaces and store all necessary logs on any reachable nodes in a distributed manner.

* Jamming resilience

WCN uses wireless consensus to prevent jamming attacks. The system resilience is secured by a valid quorum made by multiple survival nodes with consistent replies to the leader node. In the event of not sufficient quorum within the consensus group, the system should resize the group and adjust the service area dynamically based on the black out area distribution.

* Firewall

The system shall have a distributed firewall configuration among all peer nodes in the consensus group. In the event of initialization, the header is responsible for enabling designated ports for nodes who have been previously accepted by registering their network identifiers to the firewall configuration. In the event of new member joining or node leaving, the header node shall broadcast the amended configuration to the group for approval.

* Channel stability

Channels should be monitored frequently to select less busy channels for the group consensus protocol. In the event of worsen channel stability, the system shall increase the transmission power for better SNR performance and make proper channel hopping decision between all nodes within the consensus group. If the channel is considered not stable for the major consensus nodes, the system deployment should consider employing licensed spectrum for its usages.

* Streaming bandwidth

The system should have a minimum streaming bandwidth requirement for all members. This is to ensure the quality of services and constant performance among all group members. The header node shall measure nodes signal strength and their live streaming bandwidth to make sure the QoS is to be supported. Header nodes should periodically broadcast the minimum streaming bandwidth between cycles of consensus.

* Storage (TBC)

1. Minimum reliability of single node storage, shall not below a given criteria.

2. Storage RAID settings.

3. Storage copies in hot and cold backups.

#### 6.3.3 Reliability gain

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 WCN, as it can be used as design guidelines for WCN deployment. (TBC with Hao)

# 7 Hardware Definition

## 7.1 Hardware requirement

### 7.1.1 Processing capability for consensus

Processing capability or consensus capability means the computational capability of hardware in a consensus node to process consensus data and communication data packets. There are two common types of hardware in current embedded systems and computer systems: Microcontroller Unit (MCU) and Central Processing Unit (CPU). Both of them can support the computation raised in consensus and communication. The better solution for WCN could be different in varied scenarios. For example, if the power supply is limited, MCU could be a better choice as it is more energy-saving than CPU. On the other hand, when the WCN scale grows, each consensus node may use processing hardware with higher computational capability to process more communication data packets, especially for consensus nodes in self-organizing WCNs.

* MCU (Microcontroller Unit):

An MCU can be regarded as a small computer on a single chip containing one or more CPUs (processor cores) along with memory and programmable input/output peripherals. Program memory is also often included on a chip, but its size is fixed, as well as a small amount of RAM. MCUs are designed for embedded applications, in contrast to the CPUs used in personal computers or other general purposes applications.

Microcontrollers are used in automatically controlled products and devices, such as automobile engine control systems, implantable medical devices, remote controls, office machines, appliances, power tools, toys and other embedded systems. By reducing the size and cost compared to a design that uses a separate microprocessor, memory, and input/output devices, microcontrollers make it economical to digitally control even more devices and processes. Mixed signal microcontrollers are common, integrating analog components needed to control non-digital electronic systems. In the context of the internet of things, microcontrollers are an economical and popular means of data collection, sensing and actuating the physical world as edge devices. Some common MCUs for WCN are MSP430 series, STM32 series, AVR series and TMS.

* CPU

A central processing unit (CPU) is the electronic circuitry that executes instructions comprising a computer program. The CPU performs basic arithmetic, logic, controlling, and input/output (I/O) operations i.e., general computation, specified by the instructions in the program. Compared with the MCU, the main difference between CPU and MCU is the CPU usually reads and writes programs in the external storage. In addition, since modern CPUs apply large-scale electronic circuits, the power consumption of CPUs is higher than MCUs. For certain WCN scenarios with limited power supply, energy-saving CPUs may be more suitable to process consensus and communication such as RISC-based CPUs.

### 7.1.2 Communication capability

To achieve consensus in the WCN, each consensus node should have communication capability to communicate with other nodes to exchange consensus data and synchronize PDL transactions. In the access network based WCN framework, each node can use cellular modules or Wi-Fi modules, which have been widely introduced [ref placeholder], to communicate with other nodes. Therefore, the communication capability of each consensus node is not discussed in this section. In the self-organizing WCN framework, each consensus node needs to communicate with others to organize a WCN with other nodes by itself without access points, cellular networks, or other wireless communication infrastructures. Hence, the communication module in the consensus node for such a scenario should support customized protocols of self-organizing networks such as radio frequency (RF) modules LoRa and Zigbee.

* LoRa

LoRa (Long Range) is a physical proprietary radio communication technique. It is designed on spread spectrum modulation techniques derived from chirp spread spectrum (CSS) technology. Based on LoRa on the physical layer, LoRaWAN defines the software communication protocol (upper network layers). LoRaWAN is a cloud-based medium access control (MAC) layer protocol but acts mainly as a network layer protocol for managing communication between end-node devices as a routing protocol.

While the LoRa physical layer enables the long-range communication link, LoRaWAN is responsible for managing the communication frequencies, data rate, and power for all devices. Devices in the network are asynchronous and transmit when they have data available to send. Data transmitted by an end-node device can be received by multiple devices, which forward the data packets to the target device using the routing protocol.

* Zigbee

Zigbee is an IEEE 802.15.4-based specification for a suite of high-level communication protocols used to create personal area networks with small, low-power digital radios, such as for home automation, medical device data collection, and other low-power low-bandwidth needs, designed for small scale projects which need wireless connection. Hence, Zigbee is a low-power, low data rate, and close proximity wireless ad hoc network.

The technology defined by the Zigbee specification is intended to be simpler and less expensive than other wireless personal area networks (WPANs), such as Bluetooth or more general wireless networking such as Wi-Fi. Applications include wireless light switches, home energy monitors, traffic management systems, and other consumer and industrial equipment that requires short-range low-rate wireless data transfer.

Its low power consumption limits transmission distances to 10-100 meters line-of-sight, depending on power output and environmental characteristics. Zigbee has a defined rate of up to 250 kbit/s. Meanwhile, Zigbee networks are secured by 128-bit symmetric encryption keys. Zigbee devices can transmit data over long distances by passing data through a mesh network of intermediate devices to reach more distant ones. Zigbee is typically used in low data rate applications that require long battery life and secure networking.

Furthermore, there are two types of V2X (vehicle-to-everything) communication technology depending on different underlying technology (WLAN and cellular) for implementing WCN in vehicles: Dedicated Short Range Communication (DSRC based on radio communication provided by IEEE 802.11p) and C-V2X based on 3GPP LTE.

* DSRC

DSRC) is a wireless communication technology designed to allow automobiles in the intelligent transportation system (ITS) to communicate with other automobiles or infrastructures. IEEE first published the specification of WLAN-based V2X (IEEE 802.11p) in 2010. It supports direct communication between vehicles (V2V) and between vehicles and infrastructure (V2I). This technology is referred to as Dedicated Short-Range Communication (DSRC) using the underlying radio communication provided by 802.11p. The original V2X communication uses WLAN technology and works directly between vehicles (V2V) as well as vehicles and traffic infrastructure (V2I), which form a vehicular ad-hoc network as two V2X senders come within each other's range. Hence, it does not require any communication infrastructure for vehicles to communicate, which is key to assure safety in remote or little-developed areas. DSRC technology operates on the 5.9 GHz band of the radio frequency spectrum and is effective over short to medium distances. WLAN is particularly well-suited for V2X communication due to its low latency. DSRC can support interoperability and receive very little interference, even in extreme weather conditions, because of the short range that it spans. This makes it ideal for communication to and from fast-moving vehicles.

* C-V2X

In 2016, 3GPP published V2X specifications based on LTE as the underlying technology, which is generally referred to as “Cellular V2X” (C-V2X). Cellular V2X uses 3GPP standardized 4G LTE or 5G mobile cellular connectivity to exchange messages between vehicles, pedestrians, and wayside traffic control devices such as traffic signals. It commonly uses the 5.9 GHz frequency band, which is the officially designated ITS frequency in most countries. C-V2X can function without network assistance and exceeds the range of DSRC by about 25% [ref placeholder]. C-V2X is designed to operate in two modes:

* **Device-to-network:** communication using conventional cellular links for vehicle-to-network (V2N) applications such as cloud services in end-to-end solutions.
* **Device-to-device:** direct communication without the use of network scheduling for vehicle-to-vehicle (V2V), vehicle-to-infrastructure (V2I), and vehicle-to-pedestrian (V2P) applications such as vulnerable road user protection and tolling [ref placeholder].

Table. IV highlights the commonalities and differences between the usability and general properties of DSRC and C-V2X.

**Table IV.** The property comparison of DSRC and C-V2X [ref placeholder].

|  |  |  |
| --- | --- | --- |
| Property | DSRC | C-V2X |
| Goal | Direct real-time wireless communication between vehicles | |
| Underlying technology | IEEE 802.11p | Cellular |
| Communication modulation | OFDM with CSMA to offer robust communication dense and dynamic environment without dependency on GPS signal | SC-FDM with semi-persistent sensing and coding gain |
| Transmission time | 0.4 mS typically but varying from packet length | 1 mS typically but increasing energy cost per bit for long communication rage |
| Symbol duration | 8 uS with fast channel tracking | 71 uS |
| Link coding | Convolution | Turbo code |
| Transmission scheduling | CSMA: Transmit when no ongoing reception  No pre-determined transmission slots fitting facilities layer per-cycle decision whether to transmit | Semi-persistent sensing of least occupied resource  Collisions are not sensed with slow response to changing environment |
| Retransmission | No | Yes |
| Time synchronization requirement | Loose | Tight |
| Security | Public key cryptography and infrastructure  Lacking V2X isolation from non-safety domain may raise a cybersecurity risk | |

### 7.1.3 Storage capability

In a WCN, each node need some space to storge information temporarily for consensus and communication. Meanwhile, the consensus results (transactions) may be kept by nodes in a WCN. Therefore, the storage capability is considered from two aspects: storage for computing and storage for transaction persistence.

* Storage for computing

To reach consensus and communicate with other nodes, a node should possess enough storage space for computation. For example, the internal memory of the embedded device in the WCN node should be enough to load consensus applications, its operating systems, communication modules, and data packets to be processed by the processor and so on.

* Storage for transaction persistence

For nodes in access network based WCN, they can use a small storage space to keep the consensus transactions temporarily and then transmit them to other storage nodes (e.g., clouds and IPFS) or store the transactions by themselves. On the other hand, nodes in self-organizing WCN need to keep the transactions locally. In this case and the second case in access network based WCN, each node should be equipped with an enough external storage space. Due to the considerations of WCN’s mobility and energy saving, NAND flash may be a desirable choice as the external storage because of its small size, light weight and low energy cost.

## 7.2 Hardware security and threats

* Secure booting
* Trusted computing environment (TPM TEE)
* Invade detection
* Safe storage and encryption

TPM TEE Trusted computational environment

# 8 Consensus Mechanism

The consensus mechanism (CM, a.k.a. distributed consensus, consensus algorithm or consensus protocol), which ensures an unambiguous ordering of transactions and guarantees the integrity and consistency of blockchain across geographically distributed nodes, plays a key role in blockchain systems such as PDL. In a permissioned network like PDL, nodes should be authenticated to access the network whilst nodes are allowed to join/leave the network without permission and authentication in a permissionless public chain. Nevertheless, CM largely determines blockchain system security bounds (i.e., fault tolerances) and performance such as transaction throughput, delay, and node scalability. Depending on application scenarios and performance requirements, different CMs can be used. Therefore, proof-based algorithms (PoX) such as Proof-of-Work (PoW), Proof-of-Stake (PoS) and their variants are commonly used in many blockchain applications (e.g., Bitcoin, and Ethereum). PoX algorithms are designed with excellent node scalability performance through nodes competition. However, they could be very resource-demanding. For instance, recent study estimates of Bitcoin’s electricity consumption range between 0.1 percent - 0.3 percent of global electricity use in 2018 and rises rapidly to 0.55 percent in 2021[ref placeholder]. Also, these CMs have other limitations such as long transaction confirmation latency and low throughput.

Unlike the public chain, the private and consortium blockchains prefer to adopt lighter protocols such as Raft [ref placeholder] and practical Byzantine Fault Tolerance (PBFT) [ref placeholder] to reduce computational power demand and improve the transaction throughput. This property is critically important to the application scenarios of blockchain-enabled IoT ecosystems, which are typically composed of low-cost and low-power devices. Raft, which is used by the private chain, does not protect the integrity of transactions from malicious attacks, but enables the Crash Fault Tolerance (CFT) for the applying system [ref placeholder]. To protect the system from malicious users, PBFT was proposed in as an improved and practical protocol based on original BFT.

## 8.1 Proof based consensus

### 8.1.1 Proof of Work

Proof of Work (PoW) is a form of proof in which one party (the prover) proves to others (the verifiers) that a certain amount of a specific computational effort has been expended. A key feature of PoW is the asymmetry, i.e., the work or computation should be moderately hard (yet feasible) on the prover or requester side but easy to check for the verifier. To get the consensus between all nodes about the newly added block in the distributed ledger, the PoW requires each node to solve a difficult puzzle with adjusted difficulty by the applied consensus network, to get the right to append a new block to the distributed ledger. The first node who solves the puzzle will have this right.

Proof of work was popularized by Bitcoin as a foundation for consensus in permissionless decentralized network, in which nodes compete to append blocks and mint new currency, each miner experiencing a success probability proportional to the computational effort expended. Before solving a puzzle, all the verifying nodes would have to put their verified transactions into a block. Then, they start solving this puzzle, by guessing a secret value as a solution to a specific cryptographic problem. An example of guessing the secret value applied in Bitcoin is shown in Fig. 4. Some common cryptographic problems used in different PoW systems are listed as follows. Most of them are hash-based problems except the modulo problem and the discrete logarithm problems (Sharmir signatures and Diffie-Hellman puzzles) [ref placeholder].

* Hash sequences
* Integer square root modulo a large prime
* Weaken Fiat-Shamir signature
* Ong-Schnorr-Shamir signature broken by Pollard
* Partial hash inversion
* Diffie-Hellman-based puzzle
* Moderate
* Mbound
* Hokkaido
* Cuckoo Cycle

图示

描述已自动生成

**Fig. 4.** The process of guessing a secret value in Bitcoin.

If a miner can find the solution in a given time threshold, which is designated by the difficulty, the secret value can be accepted. Otherwise, the node has to make another guess of the secret value, until it gets the answer timely. The difficulty of the puzzle will be adjusted after a certain number of blocks are appended, so that the average speed for adding a new block can be stable. Because of the efforts paid for guessing the right value, this work is called the PoW. In addition, the node joining the PoW network can be called a miner, and the action of finding a suitable solution is called mining.

PoW consensus mechanism needs simple majority (>50%) of the total nodes to reach a consensus on a proposed solution, which is a relatively high fault tolerance capability as shown in Table III. However, the PoW consumes resources such as computation such as energy and CPU time. Furthermore, when the total number of nodes is small, the PoW network may have higher possibility to incur “51% attack” i.e., the malicious nodes are over half of the total nodes. In addition, the transaction confirmation latency and throughput of PoW systems are quite limited. For instance, recently published estimates of bitcoin’s electricity consumption are wide-ranging, on the order of 20-80 TWh annually, or about 0.1% to 0.3% of global electricity consumption [ref placeholder]. Also, the Transaction Per Second (TPS) is generally limited to 7 in Bitcoin and about 15 in Ethereum, while the transaction confirmation delay is typically as considerable as 10 minutes in Bitcoin and 15 seconds in Ethereum [ref placeholder]. Therefore, for PDL with fewer nodes than other permissionless distributed ledgers, PoW may not be suitable to be applied.

### 8.1.2 Proof of Stake

The PoW is supposed to be unfair: while some miners owning modern and powerful equipment could find the suitable solution easier, others with poorer condition could find it very difficult to be the first one to find a suitable solution to the puzzle. PoS could be a potential solution to deal with this inequality. The basic idea of PoS is using the stake to decide who can get the chance to mine the next block of the distributed ledger. Using stake as a proof has an advantage that anyone who owns much stake would be more trustful. This node would not want to perform any fraudulent actions to attack the distributed ledger that contains much of its profits. Furthermore, using PoS would require any attackers to own at least 50% of all stakes in the network to perform a double spending attack, which is very difficult.

However, the limitation of PoS in PDL could be the centralization caused by the stake. Since the number of nodes in PDL may be much less than that in permissionless distributed ledgers, using PoS in PDL may lead to a few nodes possessing the stake greater than 50% of the whole stake. As a result, new blocks may be only appended by these nodes, which is not a decentralized design.

### 8.1.3 Proof of Authority

Proof of Authority (PoA) is a reputation-based consensus mechanism that introduces a practical and efficient solution for distributed ledgers (especially the private ones like PDL). The PoA consensus mechanism leverages the value of identities, which means that block validators are not staking coins like PoS but their own reputation instead. Therefore, PoA distributed ledgers are secured by the validating nodes that are arbitrarily selected as trustworthy entities. The Proof of Authority model relies on a limited number of block validators and this is what makes it a highly scalable distributed system. Blocks and transactions are verified by pre-approved participants, who act as moderators of the PoA system.

PoA consensus mechanism may be applied in a variety of scenarios and is deemed a high-value option for logistical applications. When it comes to supply chains, for example, PoA is considered as an effective and reasonable solution [ref placeholder]. Furthermore, the PoA model enables companies to maintain their privacy while availing the benefits of PDL. Microsoft Azure is another example where the PoA consensus mechanism is being implemented [ref placeholder]. The Azure platform provides solutions for private networks, with a system that does not require a native currency like the Ether or gas in Ethereum, since there is no need for mining.

Although the conditions may vary from system to system, the PoA consensus mechanism is usually reliant on:

* valid and trustworthy identities: validators need to confirm their real identities.
* difficulty to become a validator: a candidate can be willing to invest money and put the reputation at stake. A tough process reduces the risks of selecting questionable validators and incentivize a long-term commitment.
* a standard for validator approval: the method for selecting validators should be equal to all candidates.

The essence behind the reputation mechanism is the certainty behind a validator’s identity. This cannot be an easy process nor one that would be readily given up. In a PoA system, the identity and reputation mechanisms should be capable of weeding out bad players. Finally, ensuring that all validators go through the same procedure ensures the PoA system’s integrity and reliability.

Although PoA is more suitable to PDL when compared with PoW, it still has some drawbacks. To maintain the reliability of the identities for the participants in PoA, extra security measures for access control may be applied to increase the system complexity. PoA consensus systems use reputation strategies which still cost computation resources. Furthermore, consensus could be out of service if the leader node (with the highest reputation) is compromised in a consensus process.

Another common criticism is that the identities of PoA validators are visible to anyone. The argument against this is that only established players capable of holding this position would seek to become a validator (as a publicly known participant). Still, knowing the validators’ identities could potentially lead to third-party manipulation. For instance, if a competitor wants to disrupt a PoA-based network, it may try to influence public known validators to act dishonestly in order to compromise the system from within.

### 8.1.4 Other proof-based consensus mechanisms

Proof of burn and proof of space are two other kinds of proof-based consensus mechanisms different to PoW, PoS and PoA. In proof of burn, miners have to send their coins to an address to “burn” them, which means these coins could not be used anymore. The miner who burns the largest amount of coin during a duration can be the one getting the right to mine a new block.

With Proof of Space, miners have to invest their money on hard disk, which is much cheaper than computing devices for PoW. During the work, the proof of space algorithm generates many large datasets called plots on the hard disk. The more plots a node has, the more chance he can get to mine a new block.Proof of elapsed time is another consensus mechanism proposed by Intel [ref placeholder] using trusted execution

environment (TEE) and Intel Software Guard Extensions (XGS) technology to perform the consensus process. In the TEE provided by XGS, each node requests a (random) waiting time from the trusted leader. After receiving the wating time, each node waits until the received waiting time elapses. When a node waits enough time, but has found no one has finished the waiting match, it can broadcast to all other nodes that it is the winner, which provides it a chance to mine new block.

## 8.2 Voting based consensus

### 8.2.1 PBFT

There are three phases of communications that are vital in PBFT protocol for the consensus, 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 in the *prepare* phase, all other nodes duplicate and propagate *prepare* message to all nodes excluding itself, *commit* phase does the same communication as the previous phase, and at the *reply* phase, when the leader node have received enough *commit* messages, it replies to 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 are required to yield correct decisions.

WCN based on PBFT involves actions that may bring conflicts to the consensus parties' interests, as malicious nodes' presences given malicious feedback, for example, the back-up 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 can be considered as Byzantine fault.

### 8.2.2 Raft

The Raft consensus model represents the network that has no conflict of interest, and all nodes are honest in the system, and such a mutual decision on information that fits every node's interest. The leader node is self-elected during this process when the node makes the call and broadcast it to the peers. The protocol of Raft started from receiving the message from the leader during *downlink*, as shown in Fig. 5 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. 5 as an example, we can see the truck (leader node) is about merging 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 to 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 as 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, it will either note the information has been confirmed, or act based on the confirmed information. During the consensus process, depends on the reliability and latency requirement, there are security thresholds, in order to assure it has the best decision, which 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 communication and node’s reliability.

图示, 折线图

描述已自动生成

**Fig. 5.** Consensus protocols of PBFT and Raft with synchronization stages.

## 8.3 Performance metrics

Security bound, node scalability, transaction throughput and latency are the four most important metrics to measure CM performance. These metrics are largely determined by the ledger data structure design and CM selection, although those metrics are contradictory to each other to some degree. For instance, in Bitcoin, transactions packed in each block can be confirmed only if six or more blocks are generated afterward. This protocol design is to prevent the double-spending issues (thus maximizing the security performance), which can significantly degrade the transaction throughput and latency performance. From the CM perspective, each CM has its unique privileges and drawbacks, which makes it a tangled choice in real-world applications in order to balance the needs of different prospects. The performance comparison of the CMs is summarized in Table IV.

**TABLE IV.** Performance comparison of commonly used CMs.

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **CM** | **Ledger type** | **Transaction throughput** | **Scalability** | **Security bound** | **Communication complexity** | **Spectrum requirement** | **Representative project** | **Latency** | **Sensitivity to communication fault** |
| PBFT | Permissioned | High | Low |  |  |  | Hyperledger Fabric | Medium | Low |
| RAFT | Permissioned | Very high | Medium |  |  |  | Quorum | Low | Medium |
| PoW | Permissionless | Low | High |  |  |  | Bitcoin, Ethereum | High | High |
| PoS | Permissionless | Low | High |  |  |  |  | High | High |
| PoA | Permissioned | Medium | High |  |  |  |  | High | High |

### 8.3.1 Security Bound

It is the lifeline of CM as the security should be guaranteed to validate the transactions stored in blocks. Security bound can be defined as the maximum faulty or byzantine nodes f supported/tolerated by the consensus protocol. Hence, CMs provide strategies of defence against in-activities and byzantine attacks with security bounds. Typical security bound for PoW is considered as , which means the consensus will compromise if more than 50 percent of the network’s resource capacity is possessed by a single party, under perfect communication and non-interruptive service. Differently, the voting-based CMs define the number of faulty nodes as either inactive or malicious, which sends misinformation to imperil the whole network. Under the assumption of perfect communications, generic PBFT allows 1/3 of overall nodes are either byzantine (i.e., malicious user) or faulty, and Raft gives a fair performance with 50 percent fault tolerance capability but cannot tolerate any malicious node.

### 8.3.2 Node Scalability

This is a metric to measure the capacity of the system to handle the increasing number of nodes. As shown in Table III, proof-based CMs are designed with excellent node scalability performance through nodes competition. In theory, PoW can hold as many users within the networks without considering the communication burden. However, in practice, considering that all transactions and mining results should be broadcasted and received by all nodes, the spectrum demand in WCN can be unaffordable when the network is extremely large. When it comes to the voting-based CMs, for instance, PBFT relies on heavy inter-node communications. As the size of the node number grows, the required communication resource provision increases rapidly, resulting in low efficiency and poor scalability. Thus, from the communication resource provision perspective, the PBFT-based blockchain hardly scales up to 100 nodes.

### 8.3.3 Transaction Throughput and Latency

These are two important but reciprocal performance metrics. Transaction throughput is measured by transaction per second (TPS), and transaction latency describes the time duration from transaction request to confirmation. In general, the proof-based consensus suffers from low throughput, due to its time guarded characteristics. On the other hand, a vote-based CM has better liveness, and it can conclude the consensus in a rapid manner; hence it yields greater throughput. For instance, the TPS is normally limited to 7 in Bitcoin and 20 to 30 in Ethereum. The transaction confirmation delay is typically as large as 60 minutes in Bitcoin and three minutes in Ethereum. On the other hand, a voting-based blockchain network can achieve a transaction throughput in the range of 100 to 1000 TPS with the current physical communication limits. Note that the communication throughput can be a bottleneck to transaction throughput since a large amount of message exchanges are required for consensus achievement. Hence, transaction throughput and latency are also dependent on the number of nodes in the consensus network.

# 9 Protocol for Wireless Consensus Network

## 9.1 Background

To illustrate how WCN is applied in PDL networks, a concrete protocol RAFT is demonstrated as an example. RAFT, as a distributed consensus protocol, is normally deployed in distributed computing system to tolerate node crash fault shown in Fig. 2. This feature can also help nodes in a wireless distributed network to tolerate node crash fault or transmission failure when they make and execute critical decisions. In distributed consensus, a valid node that engage in the consensus should have full functions to complete the consensus, which includes broadcasting, multicast, peer-to-peer communication, and the verification of request call. Applying RAFT in wireless consensus network should consider several phases including the new network construction, consensus protocol and state synchronization after the consensus.

## 9.2 Protocol description

### 9.2.1 Number of nodes

Because the overall number of nodes is unknown before the network construction. The number of nodes that engage in the consensus should be determined in this phase. Then, the determine group of nodes can start the consensus from the stage of leader election. The client node can count the number of nodes that want to construct a new distributed network and support a consent type of consensus protocol.

Firstly, the client broadcasts a network construction command to other nodes. The command includes the information about IP address of the client, type of consensus and sequence of network. Every node that supports the consensus can send an acknowledgement including its IP address and sequence of network to join the network if they can support the given consensus and have not join another network. The client can retry this procedure several times and count the number of acknowledges, which also refers to the number of nodes that join in the new distributed network after the last round of broadcasting. After nodes counting, the client can multicast a permission of consensus to the IP address of nodes that respond their acknowledgements. The nodes can start a timer when they receive the permission and convert to candidate when the timer runs out.

### 9.2.2 Node state of consensus

The state of nodes in RAFT should be synchronized in a stable storage before any type of calls. The state on the node includes:

1. Persistent state on all nodes:

***T****: Current Term of consensus*

***VF****:* *Candidate ID that this node vote for*

***LOG****:* *Log entries which contain the commands and corresponding term*

2. Volatile state on all states of nodes/candidates:

***CI***: *Index of highest log entry that is committed*

***AI****:* *Index of highest log entry applied to state*

3. Volatile state on leader (Reinitialized after the stage of election):

***NI****:* *Index of next log entry send to nodes*

***MI***: *Index of highest log entry replicated on server (matched)*

图表

描述已自动生成

**Fig. 3.** Communication topology of RAFT.

### 9.2.3 Leader election

The commands from candidates in leader election refers to *VoteRequestCall*, which includes:

***CT****:* *Candidate Term*

***CA****:* *Candidate Address*

***LLI****:* *Index of Last Log Entry from candidate*

***LLT****:* *Term of Last Log Entry from candidates*

The commands from followers in the leader election refers to *VoteRequestResponse*, which includes:

***FT****:* *Follower Term*

***VOTE****:* *Vote from Follower*

*VOTE* reply *False* when *CT* < *FT*. Otherwise, it will reply *True* and the candidate can receive the vote from this follower. If the candidate can receive votes from over half of the overall nodes in the network before the timeout of its election, it will win the election and become the leader of this consensus term.

### 9.2.4 Log replication

The commands from the leader in the stage of log replication refers to *AppendEntriesCall*, which includes:

***LT****: Leader Term*

***LA****: Leader Address*

***NLI****: Index of New Log Entry*

***NLT****: Term of New Log Entry*

***NLE****: New Log Entry for Storage*

***LC****: Leader Commit Index*

The commands from the followers in the stage of log replication refers to *AppendEntriesResponse*, which includes:

***FT****: Follower Term*

***SR****: Success Response*

1. The follower will reply false in SR if LT < FT or the log does not contain the entry at NLI whose term matches NLT.
2. If existing entry conflicts with a new entry, delete the existing entry and all nodes follow the new one.
3. Append all new entries that are not in the log.
4. If *LC < CI*, set *CI = min (LC, index of last new entry).*

### 9.2.5 Rules for node

1. For servers:

If *CI > AI*, increase *AI* by 1 and apply *LOG[AI]* to the state.

If the request or response call contains the term *CT,* or *LT* is larger than the current term *T*, it sets *T=CT/LT* and convert to followers.

1. For followers:

It can only response the call from candidates and leader. If the election timeout elapses without receiving any *AppendEntriesCall* from current leader or *VOTE* request from candidate, it converts to a candidate.

1. For candidates:

Increase term *T* by 1 at the start of election.

Vote for itself. Reset election timer.

Send *VOTE* request to all other nodes.

If the candidate receives votes from majority of nodes, it becomes the leader in current term.

If it receives *AppendEntriesCall* from the new leader, it converts to follower.

If election timeout elapses, it starts a new election.

1. For leaders:

Send initial *AppendEntriesCall* to each node and repeat it to prevent election timeout.

Append log entry to state when it receives commands from client and respond after the command is synchronized to all followers.

If the last log index *LLI* is not less than the next index *NI* for a follower, send *AppendEntriesCall* with log entries starting at NI. When the call is successful, it can update *NI* and *MI* for follower. Otherwise, it decreases *NI* and retries the progress.

If there exists an index *X* that *X > CI*, the majority of followers’ *MI* is not less than *X* and the term of *LOG[X]* is equal to the current term *T*, set *CI=X*.

Customized actions can be extended by different applications.

## 9.3 Routing and synchronization

The routing protocol are implemented in two cases:

1. A new node that wants to join the distributed network but cannot build the direct connection to the current leader.
2. The state of a node that failed in the stage of log replication need to be synchronized after the log replication. The routing protocol can be implemented to connect the failed node with the leader through several hopping nodes as shown in Fig. 4.

In the case 1, the new node wants to join the network but cannot connect to the current leader directly acts as the source node in the routing protocol. The source node needs to broadcast RREQ (Routing Request message) to neighbour nodes. The RREQ message contains:

***SA****: Address of Source Node*

***PR****: Permission Request*

***HN****: Hopping Number*

***RID****: RREQ Identity*

图示

描述已自动生成

**Fig. 4.** Routing protocol.

The node that receives the RREQ can check whether it receives the request for the first time base on the message. If it is, the node records corresponding routing message including *SA*, *RID*, increase *HN* by 1 and send this RREQ to its neighbour nodes. Meanwhile, this node sends back a RREP (Routing Response Message) to the last node that sends the RREQ message to set up a temporary reverse routing path. Otherwise, the RREQ is ignored. When the destination node is the leader in current term, it receives the RREQ and checks out if the source node has the permission to join the network. If so, it registers the source node in the current distributed network for the next round of log replication and sends back the acknowledgement.

The mechanism of RAFT ensure that a follower has a direct connection to the leader. Therefore, if any follower receives the RREQ message, it can send the request to the leader directly and set up the valid path. The threshold of fault tolerance in RAFT requests the number of nodes that implemented routing protocol cannot exceed half of the overall number.

Mentioning some Routing algorithms for WCN and PDL as information

## 9.4 On-boarding and withdrawal of nodes

The WCN may encounter the on-boarding and withdrawal of nodes, which can have critical influence on the performance of consensus.

When a new node wants to join the existing distributed network, it needs to send the engagement request to the leader of current term. If the leader replies to its request, this new node can participate in the next round of log replication. Otherwise, it should follow the routing protocol mentioned in the case 1 of Section 9.3.

The procedure that nodes withdraw from the network have several cases:

1. If a follower wants to withdraw from the network, it sends the withdraw request to the leader and leader does not send *AppendEntriesCall* to this follower anymore and deletes all routing paths through this follower. The destination nodes on these deleted paths need to restart the routing protocol to update new routing paths for future state synchronization.
2. If a leader wants to withdraw from the network, it needs to multicast an election command to all followers. The followers that receive the command can start an election timer and convert to the candidates when the timer runs out. All routing paths should be expired, and new routing paths can be set up after the election.
3. If consensus fault node wants to withdraw from the network, it needs to send the request through current routing path to the leader and the leader can delete the routing path and not synchronize the state of this failed node after next round of log replication.

The node engagement and withdrawal can change the ratio of followers and consensus failed nodes in the distributed network. Once the ratio of followers drops under 50% of overall nodes, RAFT will end the current term and start an election for the new leader.

# 10 Conclusion and Recommendation

Annex A:  
Title of annex

Annex B:  
Title of annex

# B.1 First clause of the annex

## B.1.1 First subdivided clause of the annex

Annex:  
Bibliography

Annex :  
Change History

| Date | Version | Information about changes |
| --- | --- | --- |
| <Month year> | <#> | <Changes made are listed in this cell> |
|  |  |  |
|  |  |  |
|  |  |  |

# History

|  |  |  |
| --- | --- | --- |
| **Document history** | | |
| <Version> | <Date> | <Milestone> |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

*Latest changes made on 2022-07-06*