ETSI GR PDL 020 V1.1.1 (2023-06)

Permissioned Distributed Ledger (PDL); Wireless Consensus Network

Disclaimer

The present document has been produced and approved by the Permissioned Distributed Ledger (PDL) 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.

Reference
DGR/PDL-0020_Wireless_consens

Keywords
network management, PDL, wireless,
wireless ad-hoc 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

Contents

List of Tables

List of Figures

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/).

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.

BLUETOOTH® is a trademark registered and owned by Bluetooth SIG, Inc.

Foreword

This Group Report (GR) has been produced by ETSI Industry Specification Group (ISG) 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 (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 present document presents the fundamentals and potential applications of decentralized identification that can benefit various public and private services. Further present document also discusses a set of PDL services that can together enable a PDL based Wireless Consensus Network framework.

Introduction

Consensus is a fundamental component of PDL, critical when updating ledgers with new transactions and ensuring ledgers are synchronized 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 [i.3]. However, in practical terms communication errors may occur during consensus process because 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 provides 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 realize WCNs are analysed. In addition, this study also demonstrates some use cases based on WCNs.

A consensus network is used to achiever two primary goals:

a) to ensure a consensus on content of data among nodes in a distributed system exists; and

b) to reach an agreement on a proposal.

It is expected to be 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 also serves as the backbone of distributed systems such as PDL. The present 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 the following aspects related to wireless consensus network:

- Use cases of wireless consensus networks.
- Wireless consensus network architecture.
- Methods to construct wireless consensus networks:
  - MAC and physical layers.
  - Decentralized/Centralized communication.
- Performance metrics of consensus mechanisms/protocols.
- 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] Xu H., Fan Y., Li W. & Zhang L. (2022): "Wireless Distributed Consensus for Connected Autonomous Systems". IEEE™ Internet of Things Journal, doi: 10.1109/JIOT.2022.3229746.

[i.2] Sae International (2018): "Taxonomy and definitions for terms related to driving automation systems for on-road motor vehicles".

[i.3] Shi Y., Zhou Y., & Shi, Y. (2021, July): "Over-the-air decentralized federated learning". In 2021 IEEE International Symposium on Information Theory (ISIT) (pp. 455-460). IEEE™.

[i.4] Hu Z., Shen J., Guo S., Zhang X., Zhong Z., Chen Q. A. & Li K. (2022, January): "Pass: A system-driven evaluation platform for autonomous driving safety and security". In NDSS Workshop on Automotive and Autonomous Vehicle Security (AutoSec).

[i.5] Feng C., Xu Z., Zhu X., Klaine P. V. & Zhang L. (2023): "Wireless Distributed Consensus in Vehicle to Vehicle Networks for Autonomous Driving", IEEE™ Transactions on Vehicular Technology.

[i.6] Sun Y., Zhang L., Feng G., Yang B., Cao B. & Imran M. A. (2019): "Blockchain-enabled wireless Internet of Things: Performance analysis and optimal communication node deployment", IEEE™ Internet of Things Journal, 6(3), 5791-5802.

[i.7] Zhang L., Xu H., Onireti O., Imran M. A. & Cao, B. (2021): "How much communication resource is needed to run a wireless blockchain network?", IEEE™ network, 36(1), 128-135.

[i.8] Li W., Feng C., Zhang L., Xu H., Cao B. & Imran M. A. (2020): "A scalable multi-layer PBFT consensus for blockchain", IEEE™ Transactions on Parallel and Distributed Systems, 32(5), 1146-1160.

[i.9] Williamson T. & Spencer N. A. (1989): "Development and operation of the traffic alert and collision avoidance system (TCAS)", Proceedings of the IEEE™, 77(11), 1735-1744.

[i.10] Isermann R., Schwarz R. & Stolzl S. (2002): "Fault-tolerant drive-by-wire systems", IEEE™ Control Systems Magazine, 22(5), 64-81.

[i.11] Patterson D. A., Gibson G. & Katz R. H. (1988, June): "A case for redundant arrays of inexpensive disks (RAID)". In Proceedings of the 1988 ACM SIGMOD international conference on Management of data (pp. 109-116).

[i.12] Vukadinovic V., Bakowski K., Marsch P., Garcia I. D., Xu H., Sybis M., ... & Thibault I. (2018): "3GPP C-V2X and IEEE™ 802.11 p for Vehicle-to-Vehicle communications in highway platooning scenarios". Ad Hoc Networks, 74, 17-29.

[i.13] McKeen F., Alexandrovich I., Anati I., Caspi D., Johnson S., Leslie-Hurd R. & Rozas C. (2016): "Intel® software guard extensions (intel® sgx) support for dynamic memory management inside an enclave". In Proceedings of the Hardware and Architectural Support for Security and Privacy 2016 (pp. 1-9).

[i.14] Gervais A., Karame G. O., Wüst K., Glykantzis V., Ritzdorf H. & Capkun, S. (2016, October): "On the security and performance of proof of work blockchains". In Proceedings of the 2016 ACM SIGSAC conference on computer and communications security (pp. 3-16).

[i.15] Menon A. A., Saranya T., Sureshbabu S. & Mahesh A. S. (2022): "A Comparatıve Analysis on Three Consensus Algorithms: Proof of Burn, Proof of Elapsed Time, Proof of Authority". In Computer Networks and Inventive Communication Technologies: Proceedings of Fourth ICCNCT 2021 (pp. 369-383). Springer Singapore.

[i.16] Samuel C. N., Glock S., Verdier F. & Guitton-Ouhamou P. (2021, May): "Choice of ethereum clients for private blockchain: Assessment from proof of authority perspective". In 2021 IEEE International Conference on Blockchain and Cryptocurrency (ICBC) (pp. 1-5). IEEE™.

[i.17] IEEE 802.11p™: "IEEE Standard for Information technology -- Local and metropolitan area networks -- Specific requirements -- Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications Amendment 6: Wireless Access in Vehicular Environments".

[i.18] IEEE 802.15.4™: "IEEE Standard for Low-Rate Wireless Networks".

[i.19] DSRC vs. C-V2X for Safety Applications.

[i.20] ARINC 629: "Airlines Electronic Engineering Committee, 629 Part 2-2 Multi-Transmitter Data Bus, Part 2-Application Guide", February 1999.

[i.21] ARINC 659: "Airlines Electronic Engineering Committee, 659 Backplane Data Bus", December 1993.

[i.22] ARINC 664: "Airlines Electronic Engineering Committee, 664P4-2 Aircraft Data Network, Part 4 - Internet-Based Address Structure Assigned Numbers", December 2007.

3 Definition of terms, symbols and abbreviations

3.1 Terms

Void.

3.2 Symbols

Void.

3.3 Abbreviations

For the purposes of the present document, the following abbreviations apply:

3GPP	3rd Generation Partnership Project
4G	4th Generation of mobile communication technology standards
5G	5th Generation of mobile communication technology standards
AI	Artificial Intelligence
API	Application Programming Interface
BFT	Byzantine Fault Tolerance
CA	Collision Advisory
CAN	Controller Area Network
CFT	Crash Fault Tolerance
CM	Consensus Mechanism
CP	Consensus Protocol
CPU	Central Processing Unit
CSMA	Carrier-Sense Multiple Access
CSMA/CA	Carrier-Sense Multiple Access with Collision Avoidance
CSMA/CD	Carrier-Sense Multiple Access with Collision Detection
CSS	Chirp Spread Spectrum
DCN	Distributed Consensus Network
DDoS	Distributed Deny of Service
DSRC	Dedicated Short Range Communication
FIFO	First In First Out
GPS	Global Positioning System
ID	Identity
IIoT	Industrial Internet of Things
IoT	Internet of Things
IP	Internet Protocol
IPFS	InterPlanetary File System
ITS	Intelligent Transportation Systems
LoRa	Long Range
LTE	Long Term Evolution
MAC	Medium Access Control
MCU	MicroController Unit
NAND	Not AND
OFDM	Orthogonal Frequency Division Multiplexing
PBFT	Practical Byzantine Fault Tolerance
PCDA	Perception-Collection-Decision-Action
PDL	Permissioned Distributed Ledger
PoA	Proof of Authority
PoS	Proof of Stake
PoW	Proof of Work
PoX	Proof-based Algorithms
PSU	Power Supply Unit
QoS	Quality of Service
RAID	Redundant Arrays of Independent Disks
RAM	Random Access Memory
RF	Radio Frequency
RISC	Reduced Instruction Set Computer
ROP	Return Oriented Programming
RREP	Routing Response message
RREQ	Routing Request message
SAE	Society of Automotive Engineers
SC-FDMA	Single-Carrier Frequency-Division Multiple Access
SGX	Software Guard eXtensions
SNR	Signal to Noise Ratio

SPOF	Single Point Of Failure
TCAS	Traffic Collision Avoidance Systems
TEE	Trusted Execution Environment
TPM	Trusted Platform
TPS	Transaction Per Second
UAF	Use After Free
URLLC	Ultra-Reliable and Low Latency Communication
V2I	Vehicle to Infrastructure
V2N	Vehicle to Network
V2P	Vehicle to Pedestrian
V2V	Vehicle to Vehicle
V2X	Vehicle to Everything
WCN	Wireless Consensus Network
WLAN	Wireless Local Area Network
WPAN	Wireless Personal Area Network
XGS	intel Software Guard eXtensions

4 Overview of Wireless Consensus Networks

4.1 Background

Permissioned Distributed Ledger (PDL) is built on a decentralized network that relies on frequently direct communications between distributed nodes. Compared with centralized data records as presented in Table 1, PDL is more receptive to enabling numerous participants to share data in an autonomous and uncoordinated manner. The Consensus Mechanisms (CMs), which play a pivotal role in PDL, are resource-demanding both in terms of computation and in terms of communication overheads. The CMs would often determine security requirements (i.e. fault tolerances, identity) and other key performance metrics such as transaction throughput, latency thresholds and scalability to achieve the data consistency required for proper PDL functions.

Table 1: Comparison of centralized vs. decentralized

Property	Centralized	Decentralized
Meaning	The retention of power 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 preservation of authority.	Disintermediation. Systematic dispersal of authority.
Communication	Vertical.	Open and Free.
Decision Making	Made by single entity - SPOF. Fast.	Consensus by participants to prevent SPOF. May be slow depending on consensus mechanism.
Advantage	Clear coordination and leadership.	Sharing of burden and responsibility.
Power of decision making	Managing authority (not necessarily the operator of the ledger).	Decentralization. Disintermediation. 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.	Multiple (all) participants.

Most PDL systems are designed and operated 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 rendering them Wireless Consensus Networks (WCN) [i.1]. 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 methods such as CSMA/CA. The transceiver has no knowledge if the frame has been received. On the other hand, wired systems can detect transmission failures easily through collision detections techniques, such as, CSMA/CD. Hence this study investigates consensus mechanisms and protocols that can potentially be used in WCN in the future and discusses characteristics, metrics, and use cases of WCN.

4.2 Need for Wireless Consensus Networks

4.2.1 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, an increasing number of 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. It is infeasible to rely on wired connectivity between such mobile devices. Thus, achieving consensus in open wireless channels involving mobile devices becomes a necessity and needs to be further investigated.

4.2.2 Consensus for distributed automation

Consensus for distributed automation is best demonstrated through a use case of autonomous vehicles. Considering Table 2, autonomous vehicles are currently at SAE L2 of Autonomy heading towards SAE L3 and further based on a framework defined by the Society of Automotive Engineers (SAE) [i.2], as shown in Table 2. 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 centralized, with limited availability and challenging trustworthiness, reliability, scalability, privacy, and security.

Compared with centralized 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 2: SAE Automation Levels

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							
L0	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
L1	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
L2	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							
L3	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
L4	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
L5	Full Automation		Under all roadway and environmental conditions that can be managed by a human driver				All driving modes

4.3 Motivations

Compared with wireless networks, one of the significant advantages of wired networks is their reliability and stability. Wired networks offer faster and more reliable connectivity compared to wireless networks. Since the data is transmitted through physical wires, the chances of data loss or interruption are significantly lower. Wired networks are also less susceptible to interference from other electronic devices, making them a more stable option for industries that require stable and continuous connectivity, such as hospitals and data centres. However, one of the major disadvantages of wired networks is their lack of mobility. Wired networks require physical cabling between nodes, which restricts the mobility of devices connected to them. An additional disadvantage of wired networks is the time, cost, and effort required to physically install and connect nodes to the wired network.

On the other hand, wireless networks provide mobility and flexibility, which are significant advantages for many users, and a necessity for certain types of nodes, especially those that cannot be physically connected to a wired network (e.g. a vehicle). Wireless network nodes are convenient and easy to set up since they do not require physical cabling between the nodes. Additionally, wireless networks offer users the freedom to move around while still being connected to the network. This feature is particularly beneficial for individuals who require mobility, such as mobile workers, students, and travellers. However, wireless networks are more prone to interference and signal loss, which can lead to slower and less reliable connectivity. Therefore, reaching consensus in wireless networks is more difficult than that in wired networks in terms of communication.

It is important to note that while setting up and using a wireless node is simpler than setting up a wired node and offers the benefits of mobility, setting up the wireless network infrastructure that allows such wireless connectivity (e.g. mobile cell towers and the connectivity/backhaul between them and the mobile network core, or a home/office Wi-Fi® network) is in many cases more complex and expensive than setting up and operating a wired networks. This discussion, however, is out of scope of the present document.

Although wireless networks have been widely deployed based on various protocols and standards to meet different scenarios, WCN introduces new challenges that may lead to additional requirements to both the network architecture and hardware, as well as the applied CMs and protocols. Therefore, the WCN paradigm should be analysed by stakeholders to identify its applied requirements and potential use cases in the future. The present document discusses possible requirements and use cases of WCN for PDL in terms of architecture, hardware, consensus mechanisms and protocols, defining an overall perception of how WCNs should be constructed and what the key components of WCNs should be.

5 Opportunities and Use Cases of Wireless Consensus Network

5.1 Opportunities

5.1.1 Background

Recently, there is a growing use of IoT devices in critical applications, such as industrial environments and Intelligent Transportation Systems (ITS) aiding these processes to make critical real-time decisions [i.1]. For example, between 60 to 100 sensors are built into a typical car produced today that collect data and help the driver or an autonomous system to make decisions [i.1]. Despite their benefits in terms of driver safety and information provided, most of the devices in a car today only use information collected by themselves or by other locally installed sensors for their decision making.

The safety and operational functions of a car only react to information captured by the built-in sensors. Such systems can only react to the behaviour of other cars by sensing their movement and action. E.g. when the sensor measuring the distance from the car in front senses that the distance is diminishing it will assume it is slowing down. Another sensor may be looking for break-lights on the car in front which may be used to verify the assumption. The information may then be processed by the car's safety functions which may trigger an alarm to the driver and in certain cases may even activate the braking system to avoid collision. Consider a scenario where the safety functions on both cars were connected (wirelessly): In such case the car in front could alert the car behind it "my driver has removed the leg from the acceleration pedal and placed it on the brake pedal" or "there's a pothole ahead and I am about to slow down and circumvent it from the left". Such wireless connectivity between vehicles could increase safety by giving the vehicle in the back precious time to make better decisions. 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 decisions taken by different cars are based entirely on algorithms using sensory data where false sensor readings 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 [i.4]. In order to overcome such 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 [i.5]. 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 clause.

5.1.2 Centralized

In centralized implementations, vehicles send their sensory 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 industrial sectors, especially in mobile environments, where 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 Perception-Collection-Decision-Action (PCDA) scheme [i.1].

With the continuous growth of IoT devices and connected vehicles, centralized approaches are expected to serve a growing number of autonomous cars in the near future. Although centralized systems are simpler to design and operate and bring more control over the decisions, they have certain disadvantages [i.6]:

- They are based on a Single Point Of Failure (SPOF). If the central server fails or is unreachable some cars, or the entire fleet, become inoperable.
- It may suffer computational/processing overhead when scaling.
- Data pollution affects the entire system.
- Data delivery latency in time-sensitive scenarios (such as mobility) may introduce new risks.
- Network congestion causing data delay or loss.
- They are intermediated by a single entity that is given full authority based on trust by all users. However, it may be challenging to achieve such trust and potential users may be reluctant to put their safety in the hands of an untrusted third party.

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 operational constraints. 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 short of expected values, or even in complete link failures, leading to asynchronization between vehicles, yielding unpredictable behaviour potentially resulting in accidents leading to loss of human lives [i.4].

5.1.3 Decentralized

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

- Communication link reliability, especially in wireless communication environment [i.7].
- 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 useful approach overall, its performance and robustness should be further considered and improved.

To address those challenged Distributed Consensus Networks (DCNs) can be a potential alternative to be combined with decentralized V2X systems. A DCNs is a variation of a centralized approach where the central server is replaced by a distributed ledger, which is not controlled by any single party. This disintermediation approach may resolve issues associated with mistrust in a centralized intermediator. Such DCNs would typically use a variant of the well-known types of consensus protocols:

a) *crash fault tolerance* (only tolerating communication or node failure); or

b) *Byzantine tolerance* (also tolerating malicious attacks) [i.8].

However, despite its advantages, the consensus performance can become 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 [i.7]. The well-known Practical Byzantine Fault Tolerance (PBFT) based consensus protocol, is very simple to implement compared with the PoW-based consensus, but only applicable to small scale consensus networks since it is very communication resource demanding [i.7]. In addition, unlike wired systems, wireless systems introduce channel uncertainty and scarcity of spectrum provisioning, thus entailing different security thresholds. In particular, the PBFT systems gauge node failure and would consider all associated communication links as faulty when such node failure is detected. However, when dynamic wireless communication channels are used, a node in good working order may be rendered as failed due to instable wireless links connecting that node. Moreover, traditional PBFT algorithms will consider a failed node as abstained from a vote thus critical objections hindered due to wireless communication link errors 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 consensus-based agreed upon 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 is qualified to address many of the use cases of interest to the industry and governmental institutions from both technical and legal aspects. The cost of transaction and consensus, and the fairness properties among participants can be controlled. Legal aspects, including governance support, through external legal agreements and regulatory enforcement in critical sectors can be effectively resolved. However, the construction of networks to achieve consensus for PDL in wireless environments still needs further research. Therefore, two use cases of achieving consensus for PDL in wireless environments are introduced hereunder to facilitate the study of WCN.

5.3 Use case 1: Autonomous vehicle

5.3.1 Collision avoidance and advisory (clustering decision)

The evolution of the automotive industry brings autonomous vehicles to public, with great risks in its early state [i.1]. Many catastrophic failures happened due to sensor errors, malicious attacks, and AI decision errors [i.4]. 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 define the expected behaviour of traffic sensors and the resulting advice. For example, Traffic Collision Avoidance Systems (TCAS) [i.9] 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. Land-based autonomous driving is still experimental and yet to become commercially usable [i.1]. Recent traffic accidents caused by self-driving false alarms and missed alarms have caused multiple catastrophic consequences for road users across the world [i.4]. Thus, a more comprehensive solution to deal with the reliability of self-driving is required, in order to widely adopt autonomous driving, in particular SAE L4 and above (as defined in Table 2), where needs for human interventions are minimized.

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

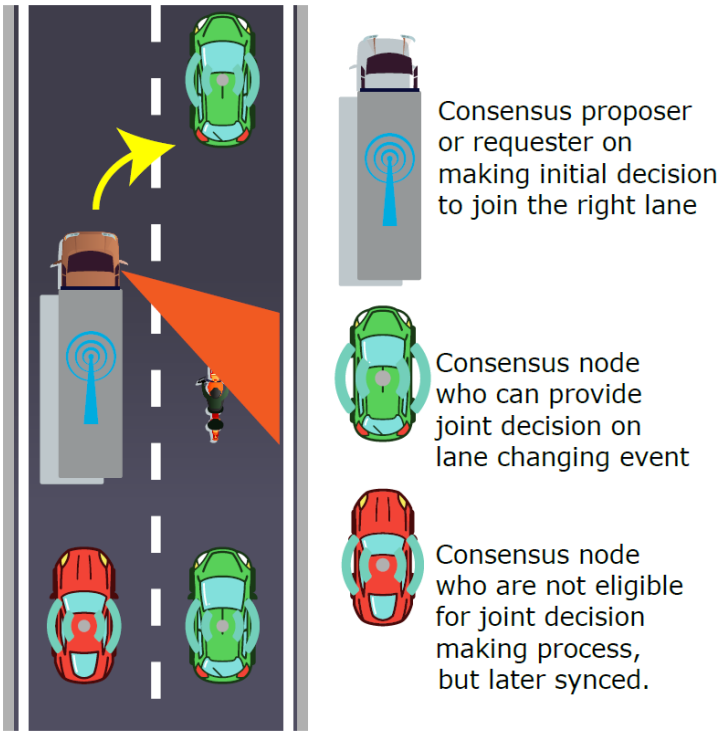

Figure 1: Wireless distributed consensus for traffic decision

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

Mission-critical payload users are the main drive for development of real-time high-reliability systems with fault tolerance capacity, such as Fly-by-wire and Drive-by-wire using internal databus (fieldbus) [i.10], e.g. ARINC 629 [i.20], ARINC 659 [i.21] (SAFEbus), ARINC 664 Part 4 [i.22] (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 recent research of 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 centre of next-generation avionics research for some years, and the reliability aspects of wireless connectivity had always been a major concern for system designers. In conventional deployments of Fly/Drive-By-Wire, databus is implemented using wired connections with dual redundancy. Reliability is guaranteed by employing duplicates in the system using First-In-First-Out (FIFO) queue. Since the physical network is isolated from outside there is no need to use Byzantine fault tolerance (BFT) or other integrity assurance methods. When critical control is operated using wireless connectivity a variant of BFT should be considered due to the open and unstable nature of wireless communications. Therefore, WCN could be a possible solution enabling nodes to achieve consensus on the data transmitted in the wireless network using BFT based protocols.

5.4 Use case 2: Industrial IoT

5.4.1 Background

Industrial Internet of Things (IIoT) refers to interconnected sensors, instruments, manufacturing and energy management devices networked together with computerized industrial applications. IIoT systems is typically defined as a layered architecture involving sensors, user devices, communication components, applications, etc. as shown in Table 3. 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 3: 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 analyse data and transform it into actionable information)
Network layer	Communications protocols e.g. Wired, Wi-Fi®, Bluetooth®, LoRa and cellular network
Device layer	Hardware (e.g. cyber-physical systems, machines and sensors)

This architecture enables data collection, exchange, and analysis, potentially facilitating improvements in productivity and efficiency as well as other economic benefits. 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.

5.4.2 Operation synchronization

In an IIoT system, multiple machines and sensors may collaborate in manufacturing by receiving operational commands from the operator (user). For example, when handling hazardous chemicals, chemical plants use machines for automated batch processing (e.g. filling bottles with a corrosive acid) and sensors to monitor the process, monitor environmental metrics, detect possible leakage and trigger alarms. Allowing a machine to perform such tasks reduces risk to employees by keeping them physically away from possible harm. It also reduces the number of employees required to be present at the manufacturing areas as some, if not all, of their tasks can now be performed by an automated machine. Some or all the machines and sensors may be teleoperated using wireless communications, reducing the risk of failure of wired communications as 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. Therefore, in such scenarios ensuring integrity of communications between operators and machines is critical to the IIoT system.

Employing WCNs could be a solution to enable sensors and machines to cross-check the received commands and reach consensus in a distributed and autonomous manner. Sensors and machines in the same category can self-organize a WCN to cross-check and synchronize the commands received in case some of the devices receive distorted instructions due to unstable communication channels. When consensus is reached the devices that have received different (distorted) commands can accept the consensus-based commands from other devices.

5.4.3 Data service

By employing WCNs in IIoT systems, PDLs can record the consensus results of the received commands for operators to record and monitor. Furthermore, running status data collected by machines, sensors, and other devices 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 participating devices can achieve a consensus on the abnormal status, the organizer can trigger an alarm to the operator. If the detected abnormal status is false (e.g. caused by environmental factors such as radioactivity and vibrations only affecting a few nodes), the consensus voting process based on WCN will identify it as such and will not trigger an alarm.

6 Functionalities and Considerations for Wireless Consensus Network Framework

6.1 Background

WCN can function as a backbone of a PDL to achieve consensus for the information recorded therein. However, the fundamental framework of WCN and the functions and considerations when constructing WCNs should be further discussed. Overall, there are two main points to be considered when designing WCNs: communication and consensus. Therefore, the following WCN framework, functions and considerations are focussed on these two points.

6.2 WCN Framework

6.2.1 Access network based WCN framework

The first type of WCN framework involves an access network as shown in Figure 2. The (four in this figure) Nodes communicate 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 executing the consensus protocol communicating with each other using said network. Consensus can be reached in this WCN framework according to protocol. Furthermore, all the nodes in the WCN can act as PDL nodes to store the consensus results (transactions) and maintain the PDL's integrity. For access network based WCN, the following components may be involved:

- Consensus node (PDL node): performs consensus protocol and maintains PDL using computational and communication resources.
- Access point: accepts connections from consensus nodes to allow them to join the network.
- Wireless communication infrastructure: supports communications among consensus nodes.
- Membership service provider: issues membership certifications to consensus nodes and verifies the memberships of consensus nodes when they try to connect through access points.
- Storage: provided by each consensus node to ensure it can keep consensus status and PDL transactions.
- Power supply: Batteries and PSU (power supply unit to obtain power from wired grids) providing power for computation, communication, and storage hardware.

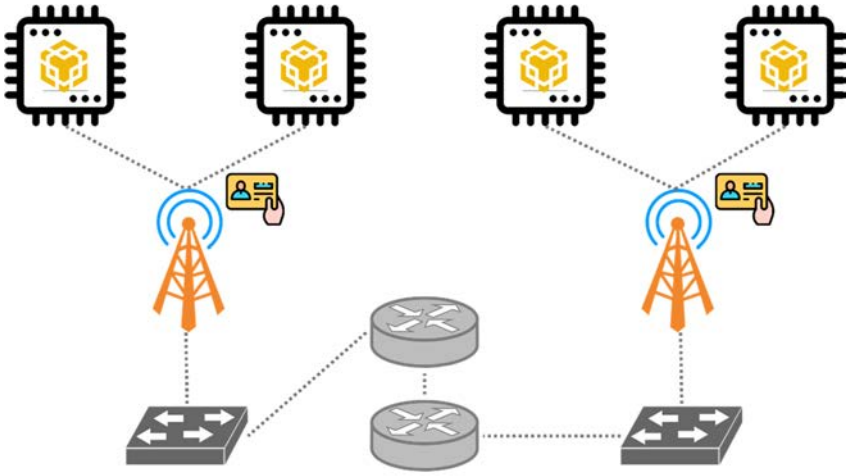

Figure 2: WCN framework based on access network

6.2.2 Self-organizing WCN framework

The second type of WCN frame is self-organizing where all consensus nodes establish direct connections with other nodes to perform consensus protocol communications. Compared with the former WCN framework, there is no access point or wireless communication infrastructure to support the communications between consensus nodes as shown in Figure 3. Therefore, consensus nodes should have peer-to-peer communication capabilities that will allow them to self-organize WCNs and perform a consensus protocol to maintain the PDL. In a self-organizing WCN, each node should have the following components:

- Computational resource: Used to process consensus data and perform communication tasks.
- Communication resource: Wireless communication hardware used to establish connections with other nodes, join and organize WCN with other consensus nodes.
- Storage: Used by consensus nodes to store consensus status and PDL transactions. Should be non-volatile.
- Power supply: Batteries and PSU (power supply unit to obtain power from wired grids) providing power for computation, communication, and storage hardware.

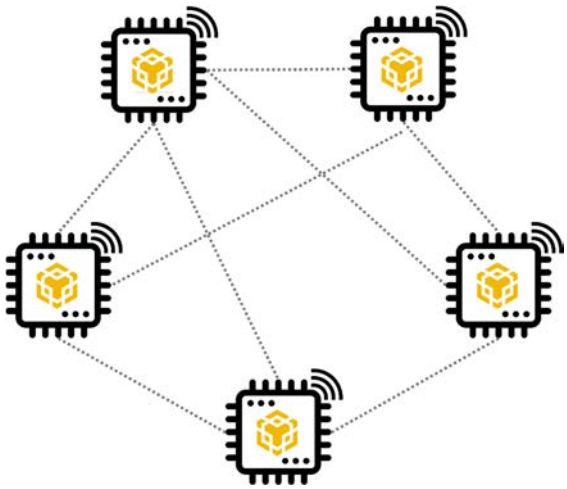

Figure 3: WCN framework based on self-organizing networks

6.3 Functionalities and Considerations

6.3.1 Membership management (network peer arrangement)

6.3.1.1 Node join

A WCN can allow new nodes to join it. All existing nodes should be made aware of newly joined nodes. After a new node joins the WCN, it should be included in the consensus process going forward. Thus, the governance/leader should update the quorum and majority settings of the PDL to accommodate the new number of nodes in the WCN in accordance with the consensus protocol.

6.3.1.2 Node quit

An existing node may leave a WCN for various reasons. All existing nodes should be made aware of the nodes that have left the PDL. After a node leaves the WCN it should be excluded from the consensus process going forward. Thus, the governance/leader should update the quorum and majority settings of the PDL to accommodate the new number of nodes in the WCN in accordance with the consensus protocol.

6.3.1.3 Faulty node detection

Node failures may be caused by failed hardware, signal loss or other reasons. Strategies should be developed to detect and manage faulty nodes in WCNs. When faulty nodes are detected, the governance/leader should exclude them from participating in consensus tasks. When faulty nodes recover, they should notify the governance/leader. When the governance/leader confirms these faulty nodes are in good working order, these nodes can be included in upcoming consensus tasks.

6.3.1.4 Leader change

In WCNs where the PDL type requires a leader, in the event that the leader node leaves the WCN or becomes a faulty node, the WCN initializes a consensus task to elect a new leader for the WCN. The ongoing consensus tasks should be paused during change of leadership. When the leader change is complete, the paused tasks continue (or 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.

NOTE: Depending on PDL type the roles of leader and governance may be complementary or interchangeable. Certain types of PDLs do not require a leader to perform governance tasks.

6.3.1.5 Access control (identity)

6.3.1.5.1 Access network based WCN

a) **Issue:** when a node applies to join a WCN, the membership service provider should issue a valid identity for a node to join the WCN via the access network for a certain period. The validity requirements of an identity may vary depending on the case. Certain implementations may require a certified identity issued by a CA. Others may require a less stringent identity as long as it is unique to other nodes in the WCN.

b) **Withdraw:** when the identity of the node expires, 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.

c) **Renew:** if the node applies to renew its identity, the membership service provider may extend the validity of the identity.

6.3.1.5.2 Self-organizing WCN

Basic identity: In a self-organizing WCN, which does not have means to realize authentication or authorization, each node can only possess a basic identity such as a unique hash ID to represent itself in the self-organizing WCN.

6.3.1.5.3 Requirements of access control methods

a) **Encryption:** the payloads for access control should be encrypted to mitigate sensitive data leakage.

b) **Identity verifiability:** the identities used in access control should be verifiable as a single node or a group of nodes.

c) **Lightweight:** the applied access control methods for WCNs should be lightweight enough especially for computational resource-constrained nodes in WCNs. Therefore, access control methods requiring complex cryptographic operations and interactions should be avoided.

6.3.2 Reliability management

6.3.2.1 Self-converged loop

The WCN should prevent unconverging communication loops 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 event of timeout, the system should trigger heartbeat signals and conduct leader rotations or re-elections. In the event of leader change failure, the WCN should perform additional attempts to change the leader and produce warning/error messages to the system control panel, the participants messages interfaces and store all necessary logs on any reachable nodes in a distributed manner.

6.3.2.2 Jamming resilience

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

6.3.2.3 Firewall

The system should have a distributed firewall configuration among all peer nodes in the consensus group. In the event of initialization, the governance/leader 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 nodes joining or leaving, the governance/leader node should broadcast the amended configuration to the group for approval.

6.3.2.4 Channel stability

Wireless channels should be monitored frequently to select less busy channels for the group consensus protocol. In the event of reduction in channel stability, the system may increase the wireless 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 majority of consensus nodes, the WCN deployment should consider employing licensed spectrum for its usages.

6.3.2.5 Streaming bandwidth

A minimum streaming bandwidth should be defined and made available to all nodes. This is to ensure the quality of services and constant performance among all nodes. The governance/leader node should measure and monitor nodes' signal strength and their live streaming bandwidth to make sure the QoS can be supported. Governance/leader nodes should periodically broadcast the minimum streaming bandwidth between cycles of consensus.

6.3.2.6 Storage

The minimum reliability of single node storage should not drop below a given criteria. Specifically, the memory dynamically used by CPU/MCU should be large and stable enough for general computation to process consensus tasks and network packets. Meanwhile, the external storage should be large and stable enough to store transactions in PDL and synchronize transactions with other nodes in the WCN. To improve the storage reliability, Redundant Arrays of Independent Disks (RAID) [i.11] settings can be considered in a node. Furthermore, when WCN nodes operate in certain extreme environments (e.g. extreme temperatures or vibration), cold backups can be considered to allow restoration of data when a node becomes physically extinct together with its local storage.

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 ultimate performance of WCN, as it can be used as design guidelines for WCN deployment.

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 can support the computation functional requirements of consensus and communication and the choice between the two depends on the specific scenario as the volume of computations MCUs and CPUs can perform significantly differs. For example, if power is limited but the volume of computations is low, an MCU could be a better choice as it is more energy-efficient than a CPU. On the other hand, when the WCN scales, consensus nodes may require higher computational capabilities to process the consensus computations and the increased volume of communication data packets. This may render MCUs insufficient for the task and thus require a CPU. This is notable especially for consensus nodes in self-organizing WCNs.

MicroController Unit (MCU)

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.

The computational power, communication speeds, and storage capabilities of MCUs are typically limited. They are designed to perform simple tasks in an economical and efficient manner.

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 analogy 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.

Central Processing Unit (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). There are two major differences between CPUs and MCUs:

a) the CPU typically uses external storage, which is practically unlimited in size (additional storage can always be added), while the MCU uses whatever limited storage is embedded therein; and

b) CPUs apply large-scale electronic circuits which allows them to perform computations significantly faster than MCUs.

Also the power consumption of CPUs is much higher than that of MCUs and their use may also produce more heat. WCN scenarios where power supply and/or cooling capacity is limited but strong computation capabilities are required can benefit from use of RISC-based energy-efficient CPUs.

7.1.2 Communication capability

7.1.2.1 Common wireless communication protocols

To achieve consensus in a WCN, each consensus node should have sufficient 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 available cellular or Wi-Fi® modules to communicate with other nodes. Such modules are commercially available, and their capabilities are well known thus the communication capability of such nodes is not discussed in this clause. In a self-organizing WCN framework, consensus nodes communicate with others to organize a WCN without Wi-Fi® access points, cellular networks, or other wireless communication infrastructure. Hence, the communication modules in consensus node for such scenarios should support customized protocols of self-organizing networks such as Radio Frequency (RF) modules LoRa and Zigbee®.

7.1.2.2 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.

7.1.2.3 Zigbee®

Zigbee® is an IEEE 802.15.4-based [i.18] 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 to 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.

7.1.2.4 Vehicle specific WCN technologies

7.1.2.4.1 Introduction

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:

a) Dedicated Short Range Communication (DSRC based on radio communication provided by IEEE 802.11p [i.17]); and

b) C-V2X based on 3GPP LTE.

7.1.2.4.2 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 [i.17]) 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 IEEE 802.11p [i.17]. 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.

7.1.2.4.3 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 % [i.12] but its transmission time and time synchronization requirement are higher than that of DSRC with an increasing energy cost for long communication range [i.19]. C-V2X is designed to operate in two modes:

a) **Device-to-network:** communication using conventional cellular links for Vehicle to Network (V2N) applications such as cloud services in end-to-end solutions.

b) **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 [i.12].

7.1.3 Storage capability

7.1.3.1 Storage requirements

Each WCN node requires storage space for two purposes:

a) temporary storage for consensus and communication purposes; and

b) permanent storage for consensus results (transactions).

7.1.3.2 Storage for computing

To reach consensus and communicate with other nodes, a node should possess enough storage space for computation. This storage may be volatile and does not need to survive a reboot. It does not need to be retained for future use.

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.

7.1.3.3 Storage for transaction persistence

Access network based WCN nodes can use a small storage space to keep the consensus transactions temporarily before transmitting them to other storage nodes (e.g. clouds and IPFS) or store the transactions by themselves on directly connected storage. Self-organizing WCN need to keep the transactions locally as they typically would not have network access to remote storage nodes. Be that remote or directly connected storage, each node should be equipped with or have access to sufficient external storage space to store transaction data. Such storage should be non-volatile and should survive a reboot of the node as it may be needed for future use by the consensus algorithms or users. When a WCN consists of mobile or battery operated nodes, where energy efficiency and size may play a significant factor, NAND flash may be a desirable choice for such storage because of its small size, lightweight and energy efficiency.

7.2 Hardware security and threats

7.2.1 Hardware security

7.2.1.1 Secure booting

The booting system of a WCN node should verify all hardware modules and operation system modules to ensure they are not replaced or distorted.

7.2.1.2 Trusted computing environment

A trusted computing environment can prevent memory exploitation (e.g. stack/heap overflow, heap spray, ROP, and UAF incurred by vulnerable applications). Therefore, the trusted computing environment should be supported by the WCN node by employing TPM modules and TEE technology such as Intel SGX [i.13].

7.2.1.3 Invasion detection and physical protection

To avoid physical sabotage of hardware in the WCN node, the stationary hardware such as base-stations, servers, edge devices should be shielded in solid containers. Some alarms can be deployed in the container to detect physical invasion. Moreover, network firewall should be applied in WCN nodes to detect invasion and block malicious connections.

NOTE: Physical protection of nodes applies primarily to stationary devices such as base station, server racks, antennas, edge devices. Mobile devices are expected to be lightweight and portable, and their physical protection should be designed accordingly.

7.2.1.4 Environmentally safe and storage encryption

Some protecting measures and material should be considered for the hardware to resist negative environmental factors such as water, fire and electromagnetic waves. In addition, hardware encryption (e.g. Bitlocker) should be supported for the storage to prevent attackers reading data from the storage chip directly.

7.2.2 Hardware threats

7.2.2.1 Trusted Platform (TPM) intrusion

Intrusion into the Trusted Platform (TPM) of a node in a WCN may cause damage or access sensitive data. Such attacks on the TPM may void confidentiality and integrity of the code and data it contains. The sensitive data at risk of intrusion is user key, session key, user identity. The resulting risk may be sibyl attack, double spending, privacy leakage, illegal interception.

The vulnerabilities of TPM in WCN nodes are:

a) **Hardware vulnerabilities** including unprotected Debug API, hardware backdoor, etc.

b) **Software vulnerabilities** including cache attack, failure in memory isolation, etc.

7.2.2.2 WCN underlay network intrusion

Intrusion into the WCN underlay network may cause service failure. Attacks on the WCN underlay network, which provides the basic communication capabilities for WCN and PDL, may cause outage of WCN.

WCN underlay network is vulnerable to the following network threats:

a) Node routing table.

b) Network DDoS.

c) Node identity.

d) Network routing.

By attacking the WCN underlay network, the attackers may corrupt the inherent WCN system security and global consensus by abusing computing power or oversaturating bandwidth.

7.2.2.3 Environmental factors and physical invasion

a) **Fire, water, strong electromagnetic waves** may break the hardware of a WCN node resulting in a faulty WCN node.

b) **Physical invasions** by attackers may sabotage hardware or introduce some malicious hardware modules into the WCN node that leads to a faulty or an evil/malicious node.

8 Consensus Protocol for WCN

8.1 Background

The 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-based systems such as WCN and PDL. CM largely determines WCN security bounds (i.e. fault tolerances) and performance such as transaction throughput, delay, and node scalability. 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.

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 to restrict them to be applied in WCN. For instance, recent study estimates that Bitcoin's electricity consumption range between 0,1 % to 0,3 % of global electricity use in 2018 and rises rapidly to 0,55 % in 2021 [i.7]. Also, these CMs have other limitations such as long transaction confirmation latency and low throughput.

Unlike public chains, the private and consortium blockchains prefer to adopt lighter protocols such as Raft and Practical Byzantine Fault Tolerance (PBFT) [i.8] to reduce computational power demand and improve the transaction throughput. This property is critically important to WCNs, which are typically composed of low-cost and low-power devices. Raft, which is used by private chains, does not protect the integrity of transactions from malicious attacks, but enables Crash Fault Tolerance (CFT) for the applying system [i.8]. To protect the system from malicious users, PBFT was proposed in as an improved and practical protocol based on original BFT.

In the following sections, different consensus protocols that could be used in WCN are introduced briefly. Meanwhile, several metrics of consensus protocols which are critical to WCN performance are analysed and compared.

8.2 Proof based consensus

8.2.1 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 spent. A key feature of PoW is the asymmetry, i.e. the 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 to 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 Figure 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) [i.14]:

a) Hash sequences.

b) Integer square root modulo a large prime.

c) Weaken Fiat-Shamir signature.

d) Ong-Schnorr-Shamir signature broken by Pollard.

e) Partial hash inversion.

f) Diffie-Hellman-based puzzle.

g) Moderate.

h) Mbound.

i) Hokkaido.

j) Cuckoo Cycle.

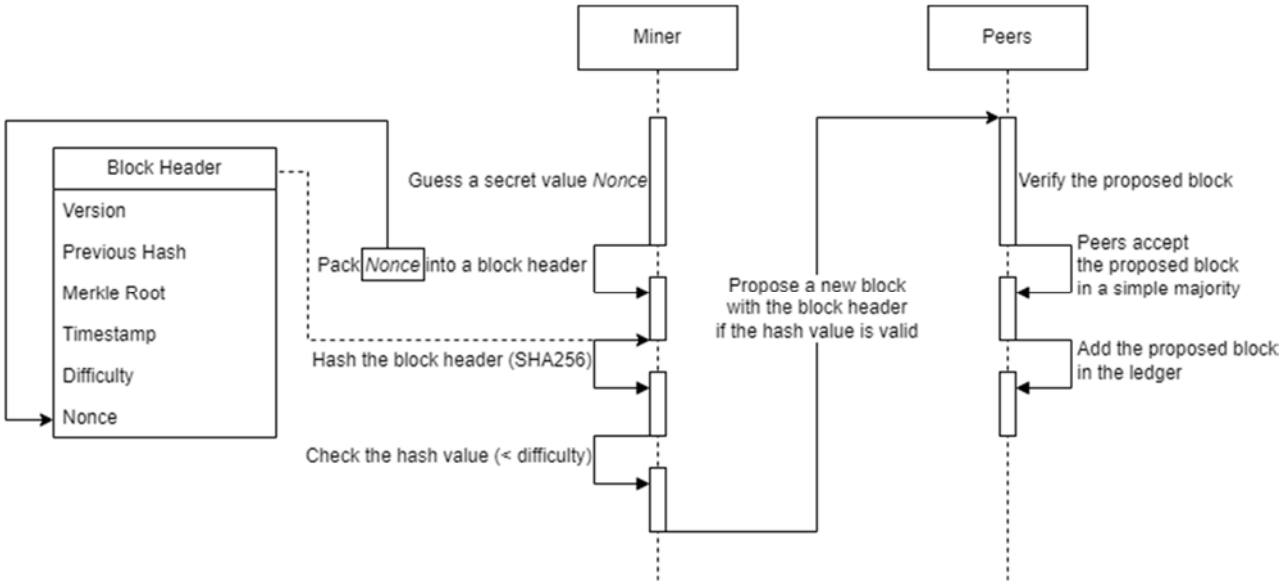

Figure 4: Process of guessing a secret value in Bitcoin™

If a miner can find the solution in a given time threshold, which is designated by the level of difficulty of said puzzle, 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 process 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 protocol 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 4. 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 more than 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 TWh to 80 TWh annually, or about 0,1 % to 0,3 % of global electricity consumption [i.8]. 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 [i.7]. Therefore, considering the resource constraints of WCN both in terms of number of nodes and in terms of transmission capacity, PoW may not be best suited.

8.2.2 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. In a wireless network there is an increased risk that nodes will become faulty due to transmission errors. As a result, the number of PDL nodes in a WCN may be significantly less than that in wired networks. Thus, using PoS in WCN may lead to fewer nodes possessing a stake greater than 50 % of the whole stake. As a result, new blocks may be only appended by those few nodes, which beats the purpose of a decentralized design.

8.2.3 Proof of Authority

Proof of Authority (PoA) is a reputation-based consensus protocol that introduces a practical and efficient solution for distributed ledgers (especially the private ones like PDL). The PoA consensus protocol 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 protocol 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 [i.15]. 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 protocol is being implemented [i.16]. 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 protocol 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 WCN and 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.

Similar to PoS, the PoA protocol is at risk of losing its decentralized nature when the number of nodes decreases, thus the use of PoA in a WCN is subject to the same shortcomings as PoS.

8.2.4 Other proof-based consensus protocols

Proof of burn and proof of space are two other kinds of proof-based consensus protocols 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 mining process, the proof of space algorithm generates many large datasets called plots on the hard disk. The more plots a node has, the more chance the miner can get to mine a new *block.Proof* of elapsed time is another consensus protocol proposed by Intel [i.15] 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.

Similar to protocols discussed earlier Proof of Space suffers from decentralization shortcomings in a WCN environment. On top of that it also suffers shortcomings related to storage space:

a) the space itself, which may be scarce on some wireless devices; and

b) transmission reliability when storage is accessed through the network.

An additional aspect relevant to WCN is leadership, in protocols that require a leader. Transmission limitations/errors may lead to frequent change of leadership that negatively impacts the performance of the PDL.

8.3 Voting based consensus

8.3.1 PBFT

There are three phases of communications that are vital in PBFT protocol for consensus, namely, *pre-prepare*, *prepare*, *commit* and a *reply* message. They are critical to successful operation, as shown in Figure 5, where 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 Figure 5.

NOTE: 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.

8.3.2 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 Figure 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 Figure 5 as an example, it can be seen that 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 the 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.

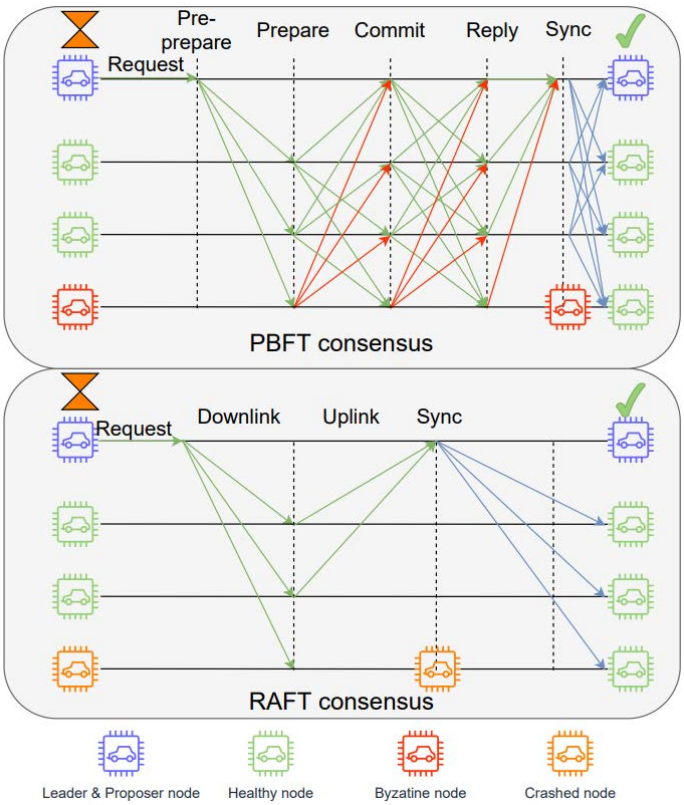

Figure 5: PBFT and Raft consensus protocols with synchronization stages

In both PBFT and Raft, nodes can reach consensus by counting the received packets and checking if the number of received packets is over 2/3 of the number of total nodes. From the view of computation cost, such consensus protocols require less computational resources than proof-based consensus protocols. Therefore, voting based consensus protocols could be more suitable to those resource-constrained scenarios such as WCN, where nodes may have limited computational resources (self-organizing WCN) or power supplement (both access network based WCN and self-organizing WCN).

8.4 Performance metrics

8.4.1 Background

Security bound, node scalability, transaction throughput and latency are the four most important metrics to measure CM performance in WCN. These metrics are largely determined by the ledger data structure design and CM selection to eventually affect the actual performance of WCN, 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 when building WCNs in order to balance the requirements of different prospects. The performance comparison of the CMs is summarized in Table 4.

Table 4: 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	$3f + 1$	$2N^2 + N$	$2N + 1$	Hyperledger Fabric	Medium	Low
Raft	Permissioned	Very high	Medium	$2f + 1$	$2N$	$N + 1$	Quorum	Low	Medium
PoW	Permissionless	Low	High	$2f + 1$	$2N$	2	Bitcoin, Ethereum	High	High
PoS	Permissionless	Low	High	$2f + 1$	$2N$	2		High	High
PoA	Permissioned	Medium	High	$2f + 1$	$2N$	2		High	High

8.4.2 Security Bound

The lifeline of CM as a security measure should be guaranteed to validate the transactions stored in blocks. In the WCN, security bound can be defined as the maximum number/ratio of faulty or byzantine nodes versus valid nodes supported/tolerated by the consensus protocol. Hence, CMs provide strategies of defence against malicious activities and byzantine attacks with security bounds. A typical security bound for PoW is considered as $2f + 1$, which means the consensus will be compromised if more than 50 % of the WCN'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 WCN). Under the assumption of perfect communications, generic PBFT allows up to a third of overall nodes to be either byzantine (malicious) or faulty. Raft allows up to 50 % fault tolerance capability but cannot tolerate any malicious node. However, the assumption of perfect communications may not be suitable for WCN and the tolerance may need to be adjusted accordingly.

8.4.3 Node Scalability

This is a metric to measure the capacity of the system to handle the increasing number of nodes in WCN. As shown in Table 4, proof-based CMs are designed with excellent node scalability performance through nodes competition. In theory, PoW can hold 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 which then suffers greater risks when it is applied in WCNs. 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 on a wired network and likely even less than that on WCN.

8.4.4 Transaction Throughput and Latency

These are two important but reciprocal performance metrics. Transaction throughput is measured as Transaction Per Second (TPS), and transaction latency describes the time duration from transaction request to confirmation. In general, proof-based consensus suffers from low throughput, due to its time guarded characteristics. On the other hand, a vote-based CP 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 TPS to 1 000 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 to be reached. The unpredictability of transmission performance in WCNs then poses an additional risk and a possible bottleneck. Transaction throughput and latency are also dependent on the number of nodes in the WCN.

9 Raft as a Protocol for WCN

9.1 Background

To illustrate how WCN works, the Raft protocol is demonstrated as an exemplar consensus protocol to be used in a WCN. Raft, as a distributed consensus protocol, is normally deployed in distributed computing system to tolerate node crash fault shown in Figure 5. This feature can also help nodes in a WCN to tolerate node crash fault or transmission failure when they make and execute critical decisions. In distributed consensus, a valid node that engages in the consensus protocol process should have the capabilities to complete the process, which includes broadcasting, multicast, peer-to-peer communication, and the verification of request call. Applying Raft to build a WCN should consider all phases including the network construction, consensus protocol and state synchronization after the consensus.

9.2 Protocol description

9.2.1 Number of nodes

Since the overall number of nodes is unknown prior to network construction in the WCN, the number of nodes that engage in the consensus protocol process should be determined during construction. The determined group of nodes can start the consensus protocol process 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 in the WCN. 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 WCN. 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 WCN 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 shown in Figure 6. The state on the node includes:

a) 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*

b) 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*

c) 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)*

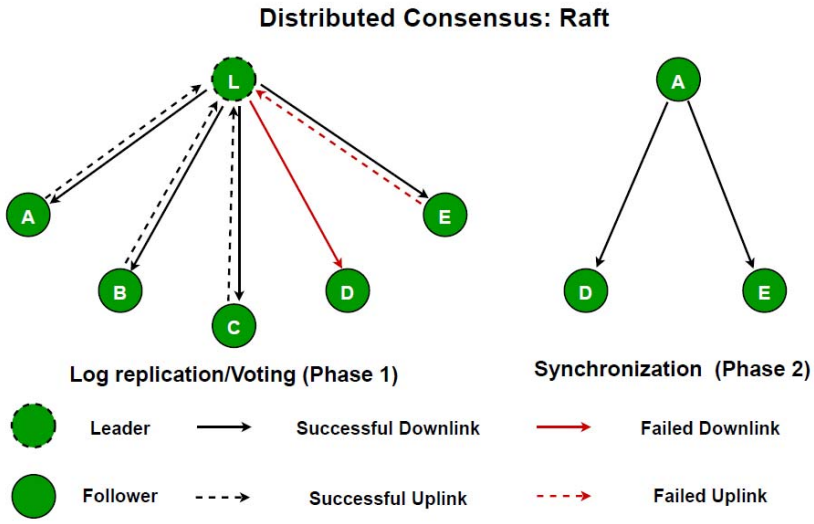

Figure 6: Communication topology of Raft

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:*

  a) The follower will reply false in SR if LT < FT or the log does not contain the entry at NLI whose term matches NLT.

  b) If existing entry conflicts with a new entry, delete the existing entry and all nodes follow the new one.

  c) Append all new entries that are not in the log.

  d) If *LC* < *CI*, set *CI = min (LC, index of last new entry).*

9.2.5 Rules for node

a) 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.

b) 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.

c) 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.

d) 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*.

9.3 Routing and synchronization

The routing protocols in the WCN are implemented in two cases:

a) A new node that wants to join the WCN but cannot build the direct connection to the current leader.

b) 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 Figure 7.

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

- ***SA:*** *Address of Source Node*
- ***PR:*** *Permission Request*
- ***HN:*** *Hopping Number*
- ***RID:*** *RREQ Identity*

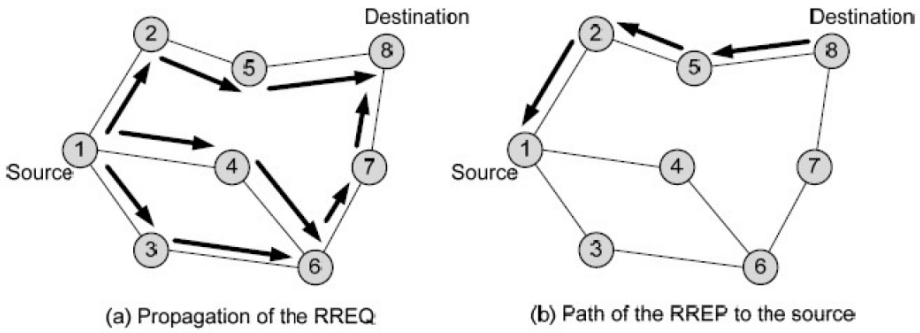

Figure 7: Routing protocol

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 protocol 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.

9.4 On-boarding and withdrawal of nodes

A WCN may encounter on-boarding and withdrawal of nodes, which may have critical influence on consensus protocol performance.

When a new node wants to join an existing WCN, it needs to send the engagement request to the leader of the current term. If the leader replies positively to the request, the new node can participate in the next round of log replication. Otherwise, it should follow the routing protocol mentioned in case a) of clause 9.3.

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

a) When a follower wants to withdraw from the WCN, it sends the withdrawal request to the leader and the leader will not send *AppendEntriesCall* to this follower anymore will delete 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.

b) When a leader wants to withdraw from the WCN, it needs to multicast an election command to all followers. The followers that receive the command can start an election timer and become candidates when the timer runs out. All routing paths should expire, and new routing paths can be set up after the election.

c) When a faulty node wants to withdraw from the WCN, 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 the 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 below 50 % of overall nodes, Raft will end the current term and start an election for a new leader.

9.5 Recommendation

Clause 9 discusses the suitability of Raft as a protocol for WCN. It is recommended that this is further studied as the indications are that it is more suitable than other protocols currently available.

PBFT should also be further studied as some of its derivatives may be suitable for WCN.

10 Conclusion and recommendation

10.1 Conclusion

The present document discusses wireless consensus networks. It first describes the background and two use cases of WCN. Then, two WCN architectures are presented with consideration to their functionalities. The hardware and consensus protocols that may be used in WCNs are discussed. Next, a protocol for WCN is presented. Finally, recommendations for the next step are included.

10.2 Recommendations for the Next Step

Since different consensus protocols and hardware for wireless networks with different computing and communication overhead are still evolving, it is out of the scope of ETSI ISG PDL to define a particular wireless consensus network with specific technology. More creative and lightweight approaches should be developed for PoS based consensus, such as Proof of Honesty (putting reputation as stake) and PBFT consensus such as PBFT with multiple layers. However, the following aspects could be considered for standardization by ETSI ISG PDL:

- Specifications on the architecture of wireless consensus network could be developed.
- Specifications on the functions and protocols of wireless consensus network could be developed.
- Specifications on the access control of wireless consensus network could be developed.

History

Document history		
V1.1.1	June 2023	Publication