ENERGY EFFICIENT BYZANTINE AGREEMENT PROTOCOLS FOR CYBER PHYSICAL RESILIENCE

A Thesis

Submitted to the Faculty

of

Purdue University

by

Manish Nagaraj

In Partial Fulfillment of the

Requirements for the Degree

of

Master of Science

May 2019

Purdue University

West Lafayette, Indiana

THE PURDUE UNIVERSITY GRADUATE SCHOOL
STATEMENT OF THESIS APPROVAL

Dr. Saurabh Bagchi, Chair

Electrical and Computer Engineering

Dr. Aniket Kate

Computer Science

Dr. Xiaojun Lin

Electrical and Computer Engineering

Approved by:

Dr. Pedro Irazoqui

Head of Graduate Program

I dedicate this thesis to my parents, Nagaraj and Hemalatha
and my grandparents, Gangamma and Gangahanumaiah,
whose love and faith in me has supported me from across the oceans.

ACKNOWLEDGMENTS

I would like to thank Prof. Bagchi, whose guidance and mentoring has always encouraged me to strive for more. It has been an honor to learn from him and I look forward to working for him in the coming years.

I would like to thank Prof. Kate, for his insightful comments which motivated me to widen my research from various perspectives. I

I would also like to thank Adithya Bhat, for stimulating discussions and for the sleepless nights we were working together before deadlines.

My sincere thanks also goes to Naif S Almakhdhub for the quick replies and help with the conduction of the experiments.

Last but not least, I would like to thank my family and friends, whose moral support has always stood by me.

TABLE OF CONTENTS

Page

LIST OF TABLES

LIST OF FIGURES

ABSTRACT

Nagaraj, Manish M.S., Purdue University, May 2019. Energy Efficient Byzantine Agreement Protocols for Cyber Physical Resilience. Major Professor: Saurabh Bagchi.

Cyber physical systems are deployed in a wide range of applications from sensor nodes in a factory setting to drones in defense applications. This distributed setting of nodes or processes often needs to reach agreement on a set of values. Byzantine Agreement protocols address this issue of reaching an agreement in an environment where a malicious entity can take control over a set of nodes and deviates the system from its normal operation. However these protocols do not consider the energy consumption of the nodes. We explore Byzantine Agreement protocols from an energy efficient perspective providing both energy resilience where the actions of the Byzantine nodes can not adversely effect the energy consumption of non-malicious nodes as well as fairness in energy consumption of nodes over multiple rounds of agreement.

1. INTRODUCTION

In many contexts of Cyber Physical Systems (CPS), it is often required for the nodes or devices in the system to agree upon a set of values. Byzantine failures can cause nodes to send conflicting values or withhold values to disrupt the system from achieving an agreement. *Byzantine Agreement* protocols ensure that the honest (or non-faulty nodes) agree upon a value in the presence of faulty (or byzantine nodes). The authors in [1] provide an overview of Byzantine Agreement protocols. If the design parameters are met, these protocols must provide two properties as described in [2].

- **Safety:** A property which is false for a behaviour if and only if it is false for some finite initial prefix of the behaviour is said to provide safety. It ensures nothing bad would happen in the system and puts an upper bound on the number of faulty nodes a system can tolerate.

- **Liveness:** Any property in which a finite behaviour can be extended to a finite or infinite behaviour that still satisfies the property is said to provide liveness. It ensures that something good will eventually happen and the system and the protocol eventually succeeds.

Most CPS settings involve low power embedded systems. These devices are resource constrained. Existing works do not consider this aspect while designing protocols. Energy expensive protocols may drain the nodes completely of energy. Since many applications need these systems to be deployed in areas not accessible to people, the information collected by these nodes may be completely lost or infiltrated. Hence it is important to ensure that any protocol in the CPS settings is energy efficient.

This work aims to improve the energy efficiency of Byzantine Agreement algorithms particularly in the setting for wireless embedded nodes. One example setting

in which our protocol could be put to use is in industrial control systems which comprise of multiple CPS devices that collect and process information. These devices need to agree upon the values that are collected. Another example of such systems could be in the case of a factory room, where the temperature needs to be monitored. Different sensors are deployed through the room, the nodes need to reach an agreement on a set of values and the determination of whether the temperature is optimum would be made based on that.

Existing works on consensus algorithms do not address the issue of energy constraints in adversaries and the energy complexity of the overall system. Since most of the devices used in practical CPS settings are low power embedded systems, practical implementation of these protocols need to take into consideration their energy constraints as well. Protocols must also take into consideration the fact that the communication setting in such systems is not very reliable.

We aim to design an agreement protocol for practical implementations in CPS settings with the following properties:

1. **Byzaninte Fault Tolerance:** In a system of n nodes, upto which t are faulty; the protocol must perform byzantine agreement.

2. **Energy Efficiency:** The protocol must be consume the least amount of energy without compromising safety of the protocol.

3. **Fariness:** The maximum energy consumed by any non faulty node must not differ by more than δ (a protocol parameter) from the minimum energy consumed by non faulty nodes.

4. **Energy Resilience:** The faulty nodes must not increase the energy consumption of the non faulty nodes.

2. CYBER PHYSICAL SYSTEMS

Cyber Physical Systems (CPS) refers to a class of distributed systems with integrated cyber and physical components. The cyber aspects of the system are computational in nature and the physical aspects include sensors, actuators and mechatronic components. These aspects interact with each other and exhibit multiple modalities. Examples of CPS systems include smart grids, automated avionics and collision avoidance systems.

Most CPS systems are typically embedded systems with physical inputs connected over a network. There are protocols and mechanisms that enable communication between these devices. One of the main advantages of CPS systems is that they provide an increase in computational resources as multiple devices coordinate with each other. The decentralization of the system also provides an increase in reliability in terms of device failures.

With advancements in the dependability and reconfigurability in both hardware and software components, CPS systems have been found to be increasingly find more applications. Many government and industry sectors have begun investing in these technologies. The European Union has a joint technology initiative across many European countries and the Advanced Research and Technology for Embedded Intelligence Systems (ARTEMIS). The U.S. National Science Foundation (NSF) has also funded many CPS based research projects. These systems are often targeted and there is a need for strengthening their security aspects.

3. FAULT MODELS IN DISTRIBUTED SYSTEMS

CPS systems are a class of distributed systems and like all distributed systems are prone to failures. Fig. 3.1 shows a general fault model in distributed systems.

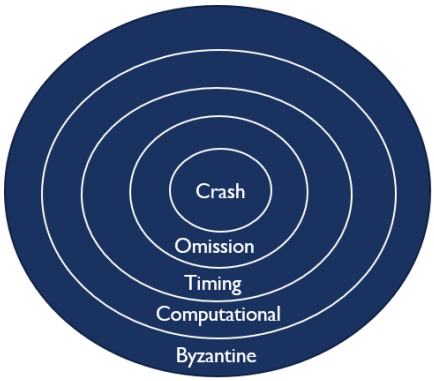

Fig. 3.1. Fault models in distributed systems

1. **Crash Failures:**

   The simplest type of failures in distributed systems are crash failures. Here the faulty device stops functioning completely after a certain point of time. To handle t such errors in a system, we need to have atleast $t+1$ number of nodes. Hence, even if t nodes have crashed, there is 1 node which is not faulty and the protocol can proceed.

2. **Omission Failures:**

   Omission failures occur when faulty devices omit sending information at a particular round. These devices may resume normal functioning there after. To handle t such errors in a system, we need to have atleast $t+1$ number of nodes. Hence, even if t nodes omit sending messages, the 1 honest node sends the message and the protocol continues.

3. **Timing Failures:**

   Timing failures occur when there is a mismatch in the clock synchronization of devices in the system. Hence messages may not be received by the intended receiver. To tolerate t timing failures, we need to have atleast $t + 1$ number of nodes in the system. Hence if there are t clock synchronization mismatches, there is another node with the correct clock synchronization and the protocol can proceed.

4. **Computational Failures:**

   The node may respond incorrectly due to computational errors. These errors may be due to a misread or due to small errors while computing the value. To tolerate t such failures, the system needs $2t + 1$ devices. Hence $t + 1$ correct values outnumber the t incorrect values and the system can make a decision based on a majority of the values.

5. **Byzantine Failures:**

   A superset of all of these errors are referred to as *Byzantine Failures*. Here, a malicious adversary takes control of the device and disrupts the protocol. These errors can be of various types but prevent the system protocol from proceeding. To tolerate t such faults, there needs to be $3t+1$ number of nodes in the system.

3.1 Byzantine Nodes

Malicious adversaries can gain access to devices within CPS systems and take control over these devices. These well resourced attacks are becoming more common with the intention of infiltrating organizations. Such nodes which are controlled or corrupted by adversaries are called Byzantine nodes.

The problem of Byzantine failures was first explained in [3]. The authors showed that in order for the protocol to be resilient to a certain number of t faulty nodes, the total number of nodes in the system must be atleast $3t + 1$ nodes. Byzantine nodes can do one of the following:

1. The node may omit information. The device may simply choose to not participate in a particular round of the algorithm or completely stop participating in the algorithm after a certain point of time.

2. The nodes may send faulty information. These may be conflicting values to what the correct value is.

3. The node may forge information. It may either pretend to be another node or claim that it has received messages from other nodes.

4. The node may equivocate. It sends conflicting values to different nodes.

Since Byzantine faults can be of different forms, it is important for protocols and algorithms in distributed systems to be resilient to all possible faults. Such systems are referred to as Byzantine Fault Tolerant (BFT) systems.

4. BROADCAST AND AGREEMENT PROTOCOLS

Devices in distributed systems can communicate either through point to point communication channels referred to as unicast or one-to-many communication channels known as multicast. Broadcast is a communication primitive where the sender sends the message to all other devices or nodes in the systems. Since the systems are prone to faults as discussed previously, this is achieved through mechanisms referred to as *broadcast protocols*. [4] provides a detailed description of reliable broadcast protocols. A reliable broadcast is one which provides the following properties:

- **Validity:** When a correct sender broadcasts a message m, then all correct nodes will eventually deliver the message m.
- **Integrity:** A message is delivered by a correct node only once and if such a message is delivered, then it was broadcast by some node.
- **Agreement:** If a message m is delivered by a correct node, then all correct nodes will eventually deliver the message m.

Once the messages from each node are broadcasted, the nodes need to agree or commit to the same set of values. This is done through mechanisms known as *agreement protocols*. Agreement protocols address many sets of problems. The problem at consideration is known as the *Interactive Consistency Problem* where correct nodes agree upon a set of values. Agreement protocols addressing this setting provide the following properties:

- **Agreement:** All correct nodes must agree upon the same set of values at the end of the protocol.
- **Validity:** If a correct node has an initial value v, then at the end of the protocol the set of values agreed upon by all correct nodes will include v.

5. LITERATURE SURVEY

There are several existing works that address the issue of agreement protocols in Byzantine environments. However none of them address the following issues.

- The energy expenditure for the protocol by the honest users.
- The energy expenditure of the honest nodes when the network is under attack.
- The adversary is also constrained in terms of energy and computational power available.

Table 5.1. summarizes some of the existing works and the parameters considered. Here n is the total number of nodes in the system, t is the number of Byzantine faulty nodes. PKI (Public Key Infrastructure) represents the cryptographic scheme used.

Three key factors that we observed when we looked at agreement protocols were the message complexity, cryptographic scheme used and mode of transmission. To the best of our knowledge no existing works makes use of multicasting. All of the works with a few exceptions had a message complexity greater than $O(n^2)$ and used expensive asymmetric key cryptographic schemes.

[7] is an extension of HotStuff [6], an SMR protocol, where a synchronous setting is considered rather than a partially synchronous setting. The protocol assumes a public key infrastructure with optimistic responsiveness. A vote for a proposal from a leader is considered as a reproposal, and the authors show that the protocol terminates in 2Δ delay, where Δ is the optimal latency of the network.The protocol assumes full connectivity and assumes a strongly rushing adaptive adversary.

There are several works on minimizing the power of broadcast that make use of the *wireless multicast* property. This is based on the property of wireless networks where omnidirectional antennae are usually used, and a single transmission can reach

Table 5.1.
Comparison of existing works

Protocol	**Network**	**Message Complexity**	**Resilience**	**PKI**
Dolev Strong [1]	Synchronous	$O(n^3)$	$n-1$	Yes
PBFT [5]	Partial	$O(n^3)$	$\lfloor \frac{n-1}{3} \rfloor$	Yes
Hotstuff [6]	Partial	$O(n)$	$\lfloor \frac{n-1}{3} \rfloor$	Yes
Sync Hotstuff [7]	Synchronous	$O(n)$	$\lfloor \frac{n-1}{2} \rfloor$	Yes
Zyzzyva [8]	Partial	$O(n^2)$	$\lfloor \frac{n-1}{3} \rfloor$	Optional
Tendermint [9]	Partial	$O(n^2)$	$\lfloor \frac{n-1}{3} \rfloor$	Yes
PaLa [10]	Partial	$O(n^2)$	$\lfloor \frac{n-1}{3} \rfloor$	Yes
Abraham et al [11]	Synchronous	$O(n^2)$	$\lfloor \frac{n-1}{2} \rfloor$	Yes
PiLi [12]	Synchronous	$O(n^2)$	$\lfloor \frac{n-1}{2} \rfloor$	Yes

many receivers. The authors in [13] propose a mixed integer programming model in determining the optimal transmission power of a sender in a graph. The authors in [14] provide an algorithm to compute broadcasting and multicasting protocols taking into account the routing topology and making efficient use of the resources. [15] considers the algorithms for two classes of optimization, namely MEB (minimum energy broadcast/multicast) and MLB(maximum lifetime broadcast). While MEB aims in minimizing the total transmission power consumed by the system, MLB aims to maximize the operation time until a node experiences battery depletion. [16] solves a generalization of the MEB problem known as MEBRA(minimum energy broadcast in realistic antennae). The authors in [16] propose an ant colony optimization problem

in solving MEBRA. The main drawbacks of these works is that malicious Nodes are not considered.

[17] shows that by utilizing broadcast channels with three nodes; a special case of $(2, 3)$ uniform hypergraph over n nodes, the bound on the number of Byzantine nodes that a fully connected synchronous network is strengthened from $n/3$ to $n/2$. The authors assume that the partial broadcast is reliable and equivocation in a broadcast is not possible. In [**?**], it is shown that $2n/h < k + 1$ can be achieved if there are k-cast links available between any subset of size k among n nodes and h of them are honest. $2n/h < k+1$ implies $n/t > (k+1)/(k-1)$. They also show the impossibility of $2n/h >= k + 1$. These works however assume that any set of 3 or k nodes in the system have access to a channel where the sender cannot equivocate with the receivers. We consider a heterogeneous network where the k is not fixed throughout the network.

[18] considers broadcast in radio networks with a Byzantine presence. A prescheduled clock cycle ensures that there is no collision during the broadcast protocols. The message from the original sender is relayed in hops. A message is forwarded if the node receives more that $f + 1$ copies of the same message, where f is the number of faulty nodes. Although this paper talks about achieving secure broadcast in radio networks, it assumes a strong adaptive adversary that can choose and strategically coordinate and choose byzantine nodes in a radio network. The paper then specifies the bounds on faulty nodes and a protocol that can tolerate those faults. It does not take into account the energy bounded adversary or the energy efficiency of the protocol.

There exist a few related works that consider k-casts. Their assumptions differ from our requirements. The novelty in our approach will be in employing k-cast for Byzantine Agreement and providing energy-efficient protocol that tolerates an energy bounded adversary. There exists a leaderless protocol which is inefficient in terms of energy complexity (due to a large number of messages exchanged per round: $d \cdot f + 1 = O(n^2)$) , with resilience of $n > 2f$ and the requirement of digital signatures.

A modification to this algorithm to use HMACs (hash based message authentication codes), a symmetric key cryptography will result in the resilience being $n > 3f$ and a corresponding increase in the number of messages to $O(n^3)$.

In this work, we aim to build a leader based byzantine agreement protocol that uses $O(n^2)$ messages and is more energy efficient than existing works. The novelty in our work consists of an energy efficient, leader based byzantine agreement protocol for a system of n nodes with a weaker model for the adversary bounded by computational power and energy.

6. SYSTEM MODEL

Most practical settings of CPS systems can be generalized to form a tiered architecture. An upper tier where devices are not energy constraint and a lower tier where devices are energy constraint and where the sensing or reading of data occurs. Fig. 6.1 illustrates the two tiered architecture considered.

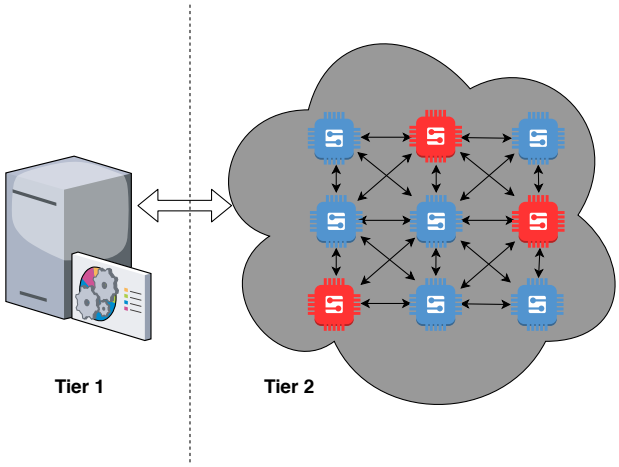

Fig. 6.1. Two tiered architecture of CPS systems

6.1 Tier 1 Nodes

Tier 1 nodes are powerful devices that are not constrained in resources. These correspond to servers and desktops that are more powerful in nature. The nodes in Tier 1 can run existing fault tolerant consensus algorithm since they are not constrained like the Tier 1 nodes. The nodes in Tier 1 communicate with their peers using any communication medium that is available between them. They communicate with the lower tier devices through energy expensive communication links such as WiFi.

6.2 Tier 2 Nodes

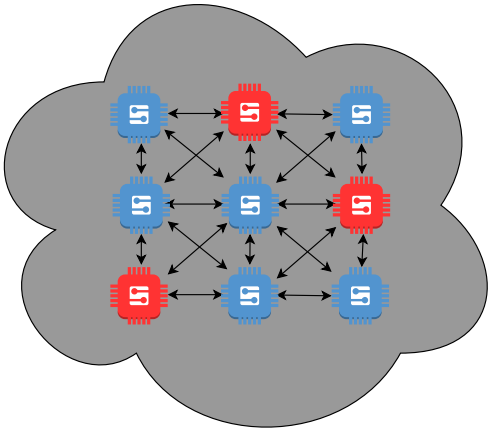

Fig. 6.2. A cluster in tier 2

Tier 2 nodes are weaker nodes that have computational constraints (such as energy, computation and bandwidth). These nodes correspond to sensor nodes that are deployed on low power embedded systems. The devices in this tier are more proximal to each other and hence have energy efficient communication links such as Bluetooth Low Energy (BLE) between them. Fig. 6.2 depicts a cluster in Tier 2. The blue nodes are honest or non-malicious nodes while the red ones are Byzantine nodes.

These devices form clusters based on proximity. The objective of each cluster is to achieve an agreement on values such as sensor information, and to communicate the agreed upon values to the Tier 1 devices. For each cluster, if the total number of nodes is n, t nodes can be byzantine. The values n and t follow the relation $n > 3t$.

We will work under the closed world assumption. That is, the number of nodes in the system is fixed. The nodes in each cluster do not form a completely connected graph, but there is enough connectivity in the system to ensure that there is no network partitioning. This implies that a failure of a device does not split the network into smaller topologies.

6.3 Adversary Model

The malicious nodes are assumed to have the following properties:

1. The total energy available to the adversary is the same as the energy available to any of the non faulty nodes.

2. The cost to violate the safety conditions must be large in terms of energy requirements.

3. The adversary can be modeled as a user who has to violate safety with access to E_f amount of energy from the system.

6.4 Network Assumptions

The network considered will be synchronous in nature. This implies that there is an upper bound on the time delay between message communications. Any message sent by any transmitter reaches the intended receiver within this upper bound.

6.4.1 Clock Synchronization

There are several works that address the issue of clock synchronization in embedded devices. We assume that the clock synchronization takes place before the agreement protocol begins to execute and that there is a prescheduled clock cycle available to the nodes. One particular example of such clock synchronizing protocols is provided in [19]. Here the authors propose HARMONIA, a clock synchronization protocol that executes in very little time.

A round is defines as the time between two consecutive transmission slots for a particular transmitter. Hence each round consists of multiple transmission slots for different transmitter-receiver/s sets. This definition of rounds provide these properties:

1. If a message is sent in round k, it will reach the recipient before round $k + 1$.

2. If a node v_i does not receive a message from another node v_j in round $k+1 \implies$ node v_j did not send the message in round k.

We will work with the assumption that all the nodes are connected such that the degree of every node is at least $2t + 1$ where t is the number of faulty nodes in the system. Formally, if the set of nodes is $N = \{v_1, v_2, \cdots, v_n\}$,

$$deg(v_i) \geq t + 1 \quad , \forall v_i \in N$$

6.4.2 k-cast Links

We will take advantage of the multicast property of transmitters presented in the literature survey. Here the node connections are heterogeneous and the network consists of a mixture of unicast and k-cast links. The k or the number of recipients in these multicasts may also vary. Let $N = \{v_1, v_2, \cdots, v_n\}$ be the set of all nodes in Tier 2. A k-cast link $(v_i, L = \{v_{i_1}, v_{i_2}, \cdots, v_{i_k}\})$ is a link with the sender $v_i \in N$. The sender v_i is connected to k nodes $v_{i_1}, v_{i_2}, ..v_{i_k}$ with $L \subseteq N\}$ and $|L| = k$ and the links possess the following properties:

1. *k-Connectivity*: If all the k nodes $\in L$ are active, then all the nodes can listen to what the sender v_i sends.

2. *Reliability*: If two nodes $v_i, v_j \in L$ of the link receive values v and v', then

$$v = v'$$

7. COMPARISON OF ENERGY CONSUMPTION

The main energy consuming aspects in agreement protocols are the communications between nodes and the cryptographic scheme used. To understand the energy consumptions of these two aspects we compared the energy consumed by various alternatives for these aspects.

7.1 Communication Mediums

Many communication mechanisms are available in CPS settings. In order to achieve maximum energy efficiency, it is essential to compare the energy consumption of different mechanisms. [20] provides an overview of the energy consumption of different mechanisms which is shown in Fig. 7.1.

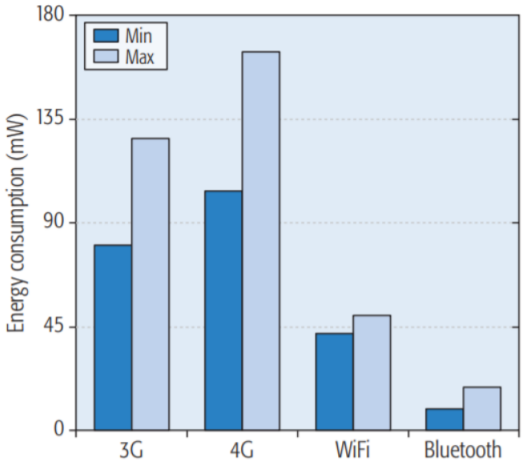

Fig. 7.1. Energy consumption using different communication techniques in mobile devices [20]

[21] provides energy consumption in Bluetooth LE for receiving and sending messages. The energy measurements were performed using a Monsoon Power Monitor

Table 7.1.
BLE and WiFi energy consumption

Size	BLE		Wifi	
	Send	Receive	Send	Receive
256B	0.1720 mJ	0.1351 mJ	1.284 mJ	0.2703 mJ
512B	0.3440 mJ	0.2703 mJ	2.568 mJ	0.5406 mJ
1kB	0.6881 mJ	0.5405 mJ	5.136 mJ	1.0813 mJ
2kB	1.3762 mJ	1.0813 mJ	10.2 mJ	2.1626 mJ
4kB	2.7525 mJ	2.1626 mJ	20.5 mJ	4.3253 mJ
8kB	5.5050 mJ	4.3253 mJ	41.0 mJ	8.6507 mJ

which can directly be connected to a module. [22] provides energy consumption values of WiFi modules. Table 7.1 compares the values of energy consumption for Bluetooth LE and RN171 WiFi module that are transmitting at the same rate provided in [21, 22]. Fig. 7.2 provides a graphical representation of the same.

7.2 Cryotographic Schemes

The most commonly used cryptographic scheme in existing works are public key infrastructure. One example and a popular scheme is ECDSA (Elliptic curve digital signature algorithm). We compared the energy consumption of such schemes to a symmetric key cryptography algorithm, HMAC (Hash based message authentication code).

The energy comparisons were made through energy measurements on a STM32F407VG micro-controller using Salea's Logic 8 Pro-analyzer. It was found that the energy used

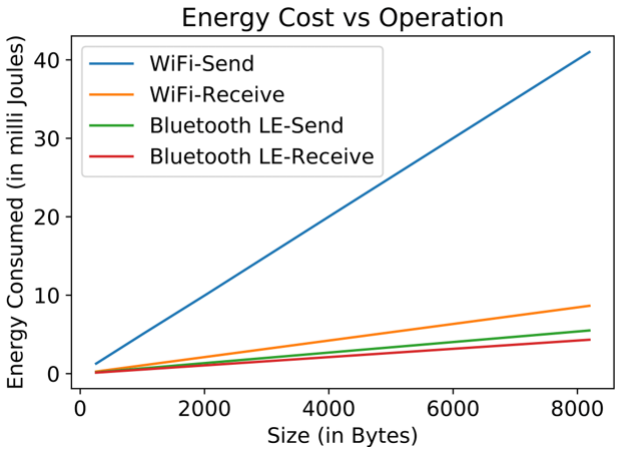

Fig. 7.2. WiFi and BLE energy consumption

for signature techniques is nearly 3 orders of magnitude more than the energy consumed for hash based message authentication codes on the messages of the same size. Table 7.2 shows the energy consumption of ECDSA and HMACs on messages of the same size.

Table 7.2.
Energy consumption of ECDSA and HMAC

Size	ECDSA		HMAC
	Sign	Verify	
256B	324 mJ	531 mJ	0.1946 mJ
512B	325 mJ	537 mJ	0.1947 mJ
1kB	332 mJ	544 mJ	0.1949 mJ
2kB	341 mJ	551 mJ	0.1949 mJ

Although HMACs are computationally and hence more energy efficient, they do not provide certain security features that signatures provide. More specifically, signatures provide the *non-repudiation* that HMACs do not. This property assures that the sender of a message cannot deny or dispute the authenticity of the message sent at an earlier point of time. This property allows protocols using signature schemes to have a lower message complexity.

8. CHALLENGES AND APPROACHES

As stated previously, one of the novelties in our work lies in the consideration of energy consumption of nodes in CPS settings and making it more efficient. The key issues in existing works in this aspect are as follows:

1. Asymmetric cryptographic schemes are used to provide security properties and these schemes are more energy expensive.

2. They do not leverage multicast features of the network. The works that do, do not consider the heterogeneous nature of the topology.

3. The message complexity of these protocols is directly proportional to the energy consumption, and most protocols have a message complexity of $O(n^2)$ or more in a system with n nodes.

We provide a baseline protocol in a CPS setting and further build upon it using techniques that address the issues stated above to make it more energy efficient.

8.1 Baseline Protocol

We consider a system with the clock synchronized as explained in the previous chapters. Each node i has an initial value v_i and all correct nodes need to agree upon a set of values from the set of initial values. The protocol requires each node to have $2t + 1$ connectivity where t is the maximum number of Byzantine faulty nodes in the system. Hence each node is connected to atleast $t + 1$ honest nodes.

Protocol Overview

1. Each node transmits it's value to its neighboring nodes in the transmission slot allotted to it. This can be done using unicasts and/or k-casts.

2. Each node maintain an array with element i corresponding to node n_i. The array for node n_i initially is filled with $\perp$ values for elements $j \neq i$ and v_i for element i.

3. The nodes exchange these arrays to their neighbors every round.

4. At the end of each round, the nodes fill each element of the array with the majority value from all received array elements.

5. At the end of d rounds (where d is the diameter of the graph) all nodes would have values for all other nodes.

Next we begin optimizing this baseline protocol in order to make it more energy efficient.

8.2 Providing Energy Efficient Security Properties

Although PKI cryptographic schemes like signatures are more energy expensive, they provide security properties which are important. In order to provide these security properties, we use the broadcast primitive provided in [23]. Here the authors provide an alternative primitive which provides the security properties of a signature without implementing signatures. Fig. 8.1 shows an overview of the broadcast primitive used. Here each round is split into two phases phase $2k$ and phase $2k + 1$. The *Echo* primitive represents the node forwarding the received message to all its neighbors, while the *Accept* primitive represents the node "commiting" or agreeing upon that message.

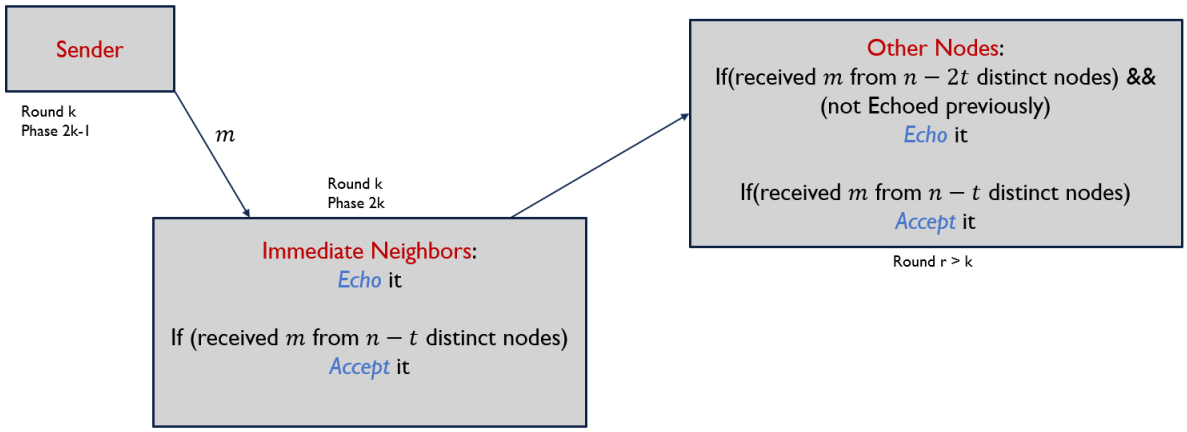

Fig. 8.1. Overview of the broadcast primitive in [23]

This mechanism however requires a completely connected network and does not leverage k-casts. It has a message complexity of $O(n^3)$ which needs to be improved upon.

8.3 Leveraging k-casts

K-casts are multicasts links from one transmitter to k receivers. K-casts provide many advantages. Consider the illustration shown in Fig. 8.2. Here v_l k-casts a message to $\{v_1, v_2, v_3\}$. If the nodes v_l and v_3 are honest, then a message forwarded to a node outside the k-cast such as v_4, is equivalent to having the nodes v_1 and v_2 also forwarding this message. This is due to the fact that the possibility of equivocation is lost in a k-cast.

The authors in [17] proved mathematically that in a system with k-casts, the number of nodes required to tolerate t Byzantine errors reduced. They showed that if any subset of k nodes can form a set with 1 transmitter and $k-1$ receivers of multicast, then the number of nodes required n to tolerate t faults is represented as:

$$n > \frac{k+1}{k-1}t$$

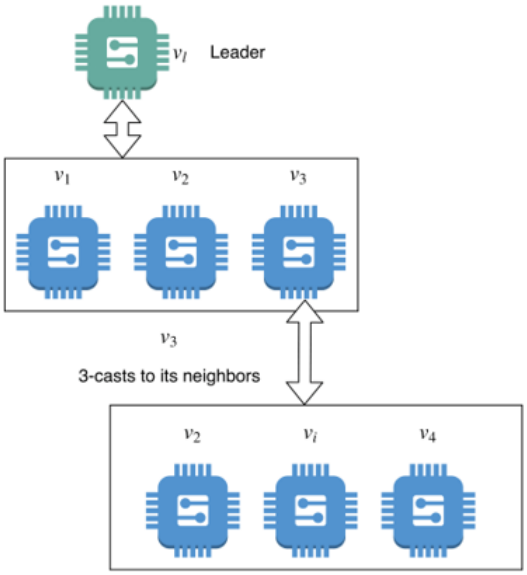

Fig. 8.2. An illustration of k-casts in a Tier 2 topology

Hence if any subset of 3 nodes can form such a set, $n > 2t$ which is reduced from $n > 3t$. This property requires the entire network topology to be known in advance. We consider a system which not only contains heterogeneous k-casts but unicasts as well.

8.4 Reducing Message Complexity

Message complexity of protocols is directly proportional to the energy consumption of the protocols. This is because the energy consumed increases with every communication between nodes. The protocol proposed in [6] provides both the linearity property as well as the optimistic responsiveness property. The linearity property ensures that the protocol has a linear message complexity. The optimistic responsiveness allows the protocol to progress after listening to only a subset of the nodes and not wait for all the nodes to respond. This is proposed in a partially synchronous setting. An extension of this work in a synchronous setting is provided in [7]. Fig. 8.3 illustrates the overview of the protocol.

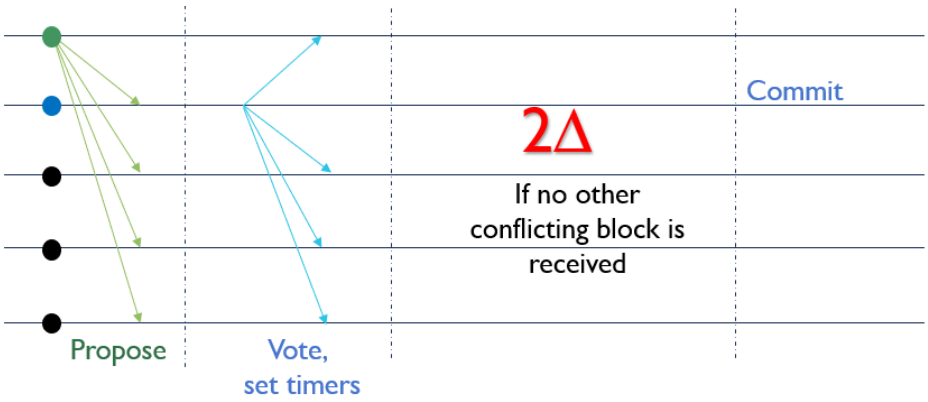

Fig. 8.3. An overview of the protocol in [7]

In the protocol, a vote from a node is not just sent to the sender of the message (referred to as *proposal*) but it is broadcasted to all other nodes as well. Once a node casts its vote, it sets timers to *commit* or agree upon the message. The timer is set to 2Δ where Δ is the maximum network delay in the network. If no other conflicting proposal is received within this time and if no other node reports an equivocation, the node commits to the message. This protocol considers a network with $2t + 1$ connectivity. This protocol however makes use of signatures and does not leverage multicasts available in the system.

By taking the above primitives and solutions into account, we aim to propose a more energy efficient protocol which leverages k-casts, does not use signatures and has a lower message complexity.

9. FUTURE WORK

Using the approaches mentioned in the previous chapter, we aim to formulate an optimization problem for the protocol.

1. The objective of the problem would be to minimize the energy consumed by all the Tier 2 nodes in order to reach an agreement. Let $\mathcal{C}$ be the set of all nodes that are not honest. Let $E_i(r)$ be the energy consumed by the i^{th} node in round r. Then the objective of the protocol is to minimize E_r (9.1).

$$E_r = \sum_{i \notin \mathcal{C}} E_i(r) \tag{9.1}$$

2. The constraints this problem include the energy constraints of the devices in the CPS setting.

3. The secondary constraints would be to assert fairness on the system. That is, after n rounds, all the honest nodes should have consumed the same amount of energy (9.2).

$$\sum_{r=0}^{n} E_i(r) = \sum_{r=0}^{n} E_j(r), i, j \notin \mathcal{C} \tag{9.2}$$

4. The protocol would consider a heterogeneous network topology with a mixture of k-casts and unicasts.

5. Security primitives would be provided either through a restricted use of signatures or alternative schemes to simulate signatures.

10. CONCLUSION

The objective of this work was to emphasize the need for more energy efficient agreement protocols in Byzantine environments. No existing works considers this aspect of the protocol. In a practical setting, it is important to take into consideration the energy constraints of the devices being used.

We compared the energy consumption of communication paradigms used and found Bluetooth LE to be the most energy efficient. We also compared the energy consumed by different cryptographic schemes used and found that asymmetric schemes schemes such as signatures were nearly 3 orders of magnitude more expensive than symmetric key schemes. However asymmetric key cryptographic schemes provide security properties like non-repudiation that symmetric key schemes don't.

We use a generalized system model consisting of two tiers. The upper tier consists of powerful nodes which do not have energy constraints and the lower tier consists on weaker nodes which are energy-wise bounded. the nodes in the lower tier devices use Bluetooth LE to communicate with each other while the communication between the lower and upper tier takes place through a more energy expensive WiFi. The nodes in the lower tier reach an agreement and pass on the agreed upon value to the upper tier nodes.

In order to provide an energy efficient solution for the nodes to reach an agreement we propose a baseline protocol and optimize it in order to make it more energy efficient. Some of the approaches we have considered are simulating signatures in order to provide security properties while consuming less energy, leveraging k-casts and reducing the message complexity. An energy efficient Byzantine agreement protocol can be proposed using these approaches while ensuring fairness in energy consumption of honest nodes.

REFERENCES

REFERENCES

[1] D. Dolev and H. R. Strong, "Authenticated algorithms for byzantine agreement," *SIAM Journal on Computing*, vol. 12, no. 4, pp. 656–666, 1983.

[2] L. Lamport and N. Lynch, "Chapter on distributed computing," MASSACHUSETTS INST OF TECH CAMBRIDGE LAB FOR COMPUTER SCIENCE, Tech. Rep., 1989.

[3] L. Lamport, R. Shostak, and M. Pease, "The byzantine generals problem," *ACM Transactions on Programming Languages and Systems (TOPLAS)*, vol. 4, no. 3, pp. 382–401, 1982.

[4] J.-M. Chang and N. F. Maxemchuk, "Reliable broadcast protocols," *ACM Transactions on Computer Systems (TOCS)*, vol. 2, no. 3, pp. 251–273, 1984.

[5] M. Castro and B. Liskov, "Practical byzantine fault tolerance and proactive recovery," *ACM Transactions on Computer Systems (TOCS)*, vol. 20, no. 4, pp. 398–461, 2002.

[6] M. Yin, D. Malkhi, M. K. Reiter, G. G. Gueta, and I. Abraham, "Hotstuff: Bft consensus in the lens of blockchain," *arXiv preprint arXiv:1803.05069*, 2018.

[7] I. Abraham, D. Malkhi, K. Nayak, L. Ren, and M. Yin, "Sync hotstuff: Synchronous smr with 2 latency and optimistic responsiveness."

[8] R. Kotla, L. Alvisi, M. Dahlin, A. Clement, and E. Wong, "Zyzzyva: speculative byzantine fault tolerance," in *ACM SIGOPS Operating Systems Review*, vol. 41, no. 6. ACM, 2007, pp. 45–58.

[9] E. Buchman, "Tendermint: Byzantine fault tolerance in the age of blockchains," Ph.D. dissertation, 2016.

[10] T. H. Chan, R. Pass, and E. Shi, "Pala: A simple partially synchronous blockchain," 2018.

[11] I. Abraham, S. Devadas, D. Dolev, K. Nayak, and L. Ren, "Synchronous byzantine agreement with expected o (1) rounds, expected o (n2) communication, and optimal resilience," *Financial Cryptography and Data Security (FC)*, 2019.

[12] T. H. Chan, R. Pass, and E. Shi, "Pili: An extremely simple synchronous blockchain," 2018.

[13] R. Montemanni, L. M. Gambardella, and A. K. Das, "The minimum power broadcast problem in wireless networks: a simulated annealing approach," in *IEEE Wireless Communications and Networking Conference, 2005*, vol. 4. IEEE, 2005, pp. 2057–2062.

[14] J. Cohen, P. Fraigniaud, J.-c. Konig, and A. Raspaud, "Optimized broadcasting and multicasting protocols in cut-through routed networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 9, no. 8, pp. 788–802, 1998.

[15] S. Guo and O. W. Yang, "Energy-aware multicasting in wireless ad hoc networks: A survey and discussion," *Computer Communications*, vol. 30, no. 9, pp. 2129–2148, 2007.

[16] H. Hernández and C. Blum, "Distributed ant colony optimization for minimum energy broadcasting in sensor networks with realistic antennas," *Computers & Mathematics with Applications*, vol. 64, no. 12, pp. 3683–3700, 2012.

[17] M. Fitzi and U. Maurer, "From partial consistency to global broadcast," in *STOC*. Citeseer, 2000, pp. 494–503.

[18] C.-Y. Koo, "Broadcast in radio networks tolerating byzantine adversarial behavior," in *Proceedings of the twenty-third annual ACM symposium on Principles of distributed computing*. ACM, 2004, pp. 275–282.

[19] J. Koo, R. K. Panta, S. Bagchi, and L. Montestruque, "A tale of two synchronizing clocks," in *Proceedings of the 7th ACM Conference on Embedded Networked Sensor Systems*. ACM, 2009, pp. 239–252.

[20] J. Wang, Y. Wang, D. Zhang, and S. Helal, "Energy saving techniques in mobile crowd sensing: Current state and future opportunities," *IEEE Communications Magazine*, vol. 56, no. 5, pp. 164–169, 2018.

[21] M. Siekkinen, M. Hiienkari, J. K. Nurminen, and J. Nieminen, "How low energy is bluetooth low energy? comparative measurements with zigbee/802.15. 4," in *2012 IEEE wireless communications and networking conference workshops (WCNCW)*. IEEE, 2012, pp. 232–237.

[22] M. S. Mahmoud and A. A. Mohamad, "A study of efficient power consumption wireless communication techniques/modules for internet of things (iot) applications," 2016.

[23] T. Srikanth and S. Toueg, "Simulating authenticated broadcasts to derive simple fault-tolerant algorithms," *Distributed Computing*, vol. 2, no. 2, pp. 80–94, 1987.