



HAL
open science

Reaching Approximate Byzantine Consensus in Partially-Connected Mobile Networks

Chuanyou Li, Michel Hurfin, Yun Wang

► **To cite this version:**

Chuanyou Li, Michel Hurfin, Yun Wang. Reaching Approximate Byzantine Consensus in Partially-Connected Mobile Networks. [Research Report] RR-7985, INRIA. 2012, pp.17. hal-00703111

HAL Id: hal-00703111

<https://inria.hal.science/hal-00703111>

Submitted on 31 May 2012

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Reaching Approximate Byzantine Consensus in Partially-Connected Mobile Networks

Chuanyou Li , Michel Hurfin , Yun Wang

**RESEARCH
REPORT**

N° 7985

May 2012

Project-Team Cidre



Reaching Approximate Byzantine Consensus in Partially-Connected Mobile Networks

Chuanyou Li ^{*} [†] [‡], Michel Hurfin [§], Yun Wang [†] [‡]

Project-Team Cidre

Research Report n° 7985 — May 2012 — 17 pages

Abstract: We consider the problem of approximate consensus in mobile networks containing Byzantine nodes. We assume that each correct node can communicate only with its neighbors and has no knowledge of the global topology. As all nodes have moving ability, the topology is dynamic. The number of Byzantine nodes is bounded by f and known by all correct nodes. We first introduce an approximate Byzantine consensus protocol which is based on the linear iteration method. As nodes are allowed to collect information during several consecutive rounds, moving gives them the opportunity to gather more values. We propose a novel sufficient and necessary condition to guarantee the final convergence of the consensus protocol. The requirement expressed by our condition is not "universal": in each phase it affects only a single correct node. More precisely, at least one correct node among those that propose either the minimum or the maximum value which is present in the network, has to receive enough messages (quantity constraint) with either higher or lower values (quality constraint). Of course, nodes' motion should not prevent this requirement to be fulfilled. Our conclusion shows that the proposed condition can be satisfied if the total number of nodes is greater than $3f + 1$.

Key-words: agreement problem, approximate consensus, Byzantine fault, distributed system, necessary and sufficient condition, dynamic topology, mobility, ad-hoc network

* PhD student invited in the research team CIDre from December 2011 to November 2012.

† School of Computer Science and Engineering, Southeast University, Nanjing, China.

‡ Key Lab of Computer Network & Information Integration, Ministry of Education.

§ INRIA Rennes Bretagne Atlantique, EPI CIDre, Rennes, France.

RESEARCH CENTRE
RENNES – BRETAGNE ATLANTIQUE

Campus universitaire de Beaulieu
35042 Rennes Cedex

Résoudre le problème du consensus approximatif en présence de Byzantins dans un réseau mobile partiellement connecté

Résumé : Nous considérons le problème du consensus approximatif dans des réseaux mobiles contenant des nœuds byzantins. Nous supposons que chaque nœud correct ne peut communiquer qu'avec ses voisins et n'a pas connaissance de la topologie globale. Comme tous les nœuds ont la possibilité de se déplacer, la topologie est dynamique. Le nombre de nœuds byzantins est borné par f et est connu de tous les nœuds corrects. Nous présentons tout d'abord un protocole de consensus approximatif byzantine qui est fondé sur la méthode d'itération linéaire. Comme les nœuds sont autorisés à collecter des informations lors de plusieurs tours consécutifs, le fait de se déplacer leur donne l'occasion de recueillir plus de valeurs. Nous proposons une nouvelle condition nécessaire et suffisante pour garantir la convergence finale du protocole de consensus. La contrainte exprimée par notre condition n'est pas "universelle": lors de chaque phase, elle ne concerne qu'un seul nœud correct. Plus précisément, au moins un nœud correct parmi ceux qui proposent la valeur minimale ou la valeur maximale présente dans le réseau, doit recevoir suffisamment de messages (contrainte sur la quantité) contenant des valeurs supérieures ou inférieures (contrainte sur la qualité). Bien entendu, les déplacements des nœuds doivent permettre à cette condition d'être remplie. Notre conclusion montre que la condition proposée peut être satisfaite si le nombre total de nœuds est plus grand que $3f + 1$.

Mots-clés : problème d'accord, consensus approximatif, faute Byzantine, système réparti, condition nécessaire et suffisante, topologie dynamique, mobilité, réseau ad-hoc

1 Introduction

We consider a distributed system where nodes are mobile and form an ad hoc network characterized by a dynamic topology. When a node changes its physical location by moving around, it also changes the set of its neighbors with whom it can communicate directly (roughly speaking, nodes that are physically nearby). The system is unreliable. Nodes may suffer from Byzantine faults and messages may be lost. A Byzantine node, also called a malicious node, may stop its activity or execute arbitrary code. In particular, it may send messages with fake values. Nodes that are not malicious are said to be correct.

Consensus is recognized as a basic paradigm for fault-tolerance in distributed systems. According to the application's needs, several variants of the consensus problem have been proposed. Among these agreement abstractions, one is called the *Approximate consensus* problem and has been presented for the first time in [1]. Each node begins to participate by providing a real value called its initial value. Eventually all correct nodes must obtain final values that are different from each other within a maximum value denoted ϵ (convergence property) and must be in the range of initial values proposed by the correct nodes (validity property). Approximate consensus can be used in applications (clock synchronization, distributed data fusion, . . .) that do not require to achieve exact agreement on a single outcome value.

Several protocols have been proposed to solve this problem in the presence of Byzantine nodes. Some protocols [1, 2] assume that the network is fully connected: during the whole execution, a correct node should be able to communicate by message passing with any other correct node. Obviously, this property is not satisfied in our context. Other protocols [3, 4, 5] consider partially connected networks but require an additional constraint: any correct node must know the whole topology. Again, such a global information is impossible to obtain in our context. Based on the linear iterative consensus strategy [6], recent protocols [7, 8, 9] also assume that the network is partially connected but do not require any global information. At each iteration, a correct node broadcasts its value, gathers values from its neighborhood and updates its own value. Its new value is an average of its own previous value and those of some of its neighbors. Like in [1], before computing its new value, a correct node must ignore some of the values it has collected. These removed values may have been proposed by Byzantine nodes and may invalidate the validity property. In order to achieve convergence, the proposed solutions rely on additional conditions that have to be satisfied by the topology. In [7, 9], the proposed conditions are proved to be sufficient and necessary in the case of an arbitrary directed graph.

The solution presented in this paper addresses the approximate Byzantine consensus problem in Partially-Connected Mobile Networks. It follows the general strategy proposed in [7, 8, 9]. However, it differs from these previous works for two main reasons. First we modify the iterative protocol to cope more efficiently with mobility. Each node still follows an iteration scheme and repeatedly executes rounds. Yet a round is now decomposed into two parts: a moving step followed by a computing step. Furthermore, during the computing step, a node still broadcasts its value, gathers values and updates its values but now the values used to compute its new value have not necessarily been received during the current round. In other words, a correct node can now take into account values contained in messages sent during consecutive rounds. An integer parameter (denoted R_c hereafter) is used to fix the maximal number of rounds during which values can be gathered and stored while waiting to be used. Thanks to this flexibility, a node can use its ability to travel to collect enough values. The second difference is the most important one. While the solutions proposed in [7, 8, 9] define conditions that refer only to the topology, we present a condition that considers also the values proposed by correct nodes. To understand the interest of our approach, let us consider the following example. One correct node p_i proposes an initial value v_a while all the other correct nodes propose an initial value v_b . In this particular scenario, if the node p_i can receive values from a sufficient number of correct neighbors, approximate consensus can be reached even if all the other nodes are isolated and receive no message. This example suggests that the location of values is just as important as the network topology. In [7, 9], constraints on the topology ensure that each node has enough neighbors. These constraints are "universal" because they affect all nodes

in the network. In a mobile environment, it is difficult to ensure that no node is never isolated from (or insufficiently connected to) the rest of the network. Furthermore, the above example shows that a strong universal constraint is not always necessary. In this paper, a novel sufficient and necessary condition is proposed. In this condition, topology and values proposed by correct nodes are both taken into account. The condition affects only a subset of nodes that can change from one round to another. More precisely, the condition focuses only on the correct nodes that propose either the maximum or the minimum value and imposes no obligation on the other nodes. To achieve consensus, from time to time, at least one of these particular nodes must receive enough messages (quantity requirement) with values different from its current value (quality requirement). Obviously, the constraint is weaker and not universal as it has to be satisfied by a single node.

The rest of this paper is organized as follows. Section 2 introduces the model and provides a formal definition of the approximate consensus problem. In Section 3, we present our protocol based on linear iteration and we prove that correct nodes will never violate the validity property by adopting illegal values. Section 4 sketches out some related works. To ensure convergence, Section 5 proposes a sufficient condition. Then this condition is slightly modified to obtain a sufficient and necessary version. Section 6 brings our concluding remarks.

2 Model and Problem Definition

2.1 Model

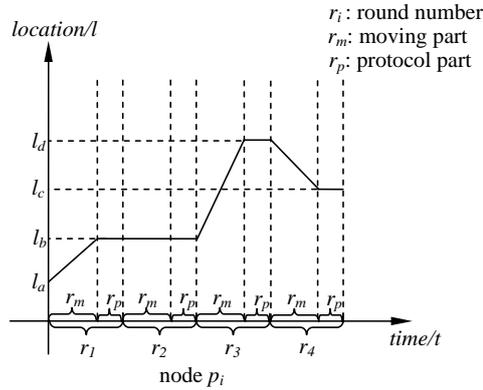
We consider a mobile distributed system composed of n nodes $V = \{p_1, p_2, \dots, p_n\}$. During the entire period of computation, each node p_i can move towards any direction and at any speed within a limited geographical area. Nodes communicate with each other only by exchanging messages. A node can only communicate with its close neighbors. Therefore the topology (*i.e.*, the communication graph) is dynamic. When receiving a message, the receiver knows the correct identity of the sender. The communication is synchronous. Messages can be lost but there are no duplicate messages and each channel is FIFO.

Nodes are divided into two subsets denoted C_n and F_n . The set C_n contains the correct nodes which always follow the protocol's specification. The nodes of the set F_n are Byzantine nodes. They behave arbitrarily and can collude together. In particular, each of them can stop its computation or send messages with different fake values to different neighbors. No assumption restricts their possible behaviors. However, the total number of Byzantine nodes is limited by f .

The protocol described in Section 3 is based on an iterative process. A sequence of rounds is carried out by each node. A round is identified by a round number r that belongs to the set $R = \{1, 2, \dots\}$. For simplicity, the schedulers of all correct nodes are assumed to be fully synchronous. Each round r is divided into two parts denoted r_m (mobility part) and r_p (protocol part). During r_m , a node can either move to a new location or stay in the same place. During r_p , a node p_i broadcasts its value v , gathers values, and updates its state: a consensus protocol (such as the one proposed in Section 3) describes the computation performed by p_i during a round.

The behavior of a correct node p_i during 4 consecutive rounds is described in Figure 1. Just before executing round r_1 , p_i is located in l_a . During the first part of round r_1 , p_i moves to another location l_b and executes the protocol. The node p_i remains in location l_b during round r_2 and executes again the protocol. It moves to l_d during round r_3 and executes the protocol. In round r_4 , p_i moves to location l_c and it executes the protocol for the fourth time.

During a round r , a *simple directed graph* $G_r(V, E_r)$ is used to model the dynamic topology. If during round r , node p_i can receive a message from a node p_j located in its neighborhood then there is a directed link from p_j to p_i : $(p_j, p_i) \in E_r$. In the proposed protocol, a correct node p_i can receive a value during round r , keep it during several consecutive rounds, and use it in a future round $r + k$. Therefore,


 Figure 1: Path followed by node p_i during 4 rounds

the concept of *joint graph* [10] is also used within this paper. A joint graph is defined as the union of the graphs corresponding to several well-identified consecutive rounds. Figure 2 illustrates this concept in the particular case of two consecutive rounds r_1 and r_2 . The graphs $G_{r_1}(V, E_{r_1})$ and $G_{r_2}(V, E_{r_2})$ are depicted on the left side. The corresponding joint graph appears on the right side.

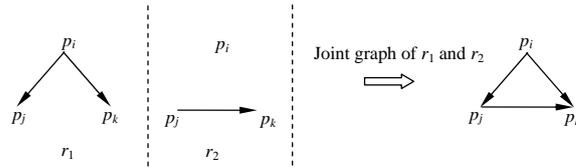


Figure 2: Two graphs and the associated joint graph

2.2 Definition of the Agreement Problem

Within this paper, the value of a correct node p_i at the beginning of round r is denoted $v_i(r)$. Consequently the initial value of p_i is denoted $v_i(1)$. The minimum (respectively maximum) value proposed by correct nodes during round r is denoted $v_{min}(r)$ (respectively $v_{max}(r)$).

Definition 1. The approximate Byzantine consensus problem is formally defined by two properties:

Validity property:

During any round r , the value of a correct node is in the range of initial values of correct nodes:

$$\forall p_i \in C_n, \forall r \geq 1, v_i(r) \in [v_{min}(1), v_{max}(1)]$$

Convergence property:

Eventually all correct nodes have values which are different from each other within a maximum predefined value denoted ϵ and such that $\epsilon > 0$.

$$\forall p_i, p_j \in C_n, \exists N > 0, \forall r > N, |v_i(r) - v_j(r)| < \epsilon$$

3 The Protocol and its Safety Proof

Algorithm 1 Linear Approximate Byzantine Consensus

```

1:  $r \leftarrow 1$ ;
2:  $v_i(1) \leftarrow$  the initial value proposed by  $p_i$ ;
3:  $Neb_i(1) \leftarrow null$ ;

4: for any node  $p_i$  in round  $r$ ;
5: do;
6:  $p_i$  sends  $v_i$  to its neighbors;
7:  $p_i$  waits for receiving messages;
8:  $Neb_i(r) \leftarrow \{\text{new values from neighbors}\} \cup Neb_i(r)$ ;
9:  $Neb_i(r) \leftarrow \text{sort}(Neb_i(r))$ ;
10:  $x \leftarrow$  the number of values bigger or equal than  $v_i(r)$ ;
11:  $y \leftarrow$  the number of values less or equal than  $v_i(r)$ ;
12: if  $(x \geq f + 1$  or  $y \geq f + 1)$  then
13:    $Neb_i(r) \leftarrow \text{reducing}(Neb_i(r), f, x, y)$ ;
14:    $v_i(r + 1) \leftarrow \text{average}(Neb_i(r), v_i(r))$ ;
15:    $Neb_i(r + 1) \leftarrow null$ ;
16: else
17:    $v_i(r + 1) \leftarrow v_i(r)$ ;
18:   if  $(r \bmod R_c$  equals to 0) then
19:      $Neb_i(r + 1) \leftarrow null$ ;
20:   else
21:      $Neb_i(r + 1) \leftarrow Neb_i(r)$ ;
22:   end if
23: end if
24:  $r \leftarrow r + 1$ ;
25: enddo;

26: Procedure  $\text{reducing}(Neb_i(r), f, x, y)$ ;
27: do;
28:  $B \leftarrow$  the set of  $f$  largest values in  $Neb_i(r)$ ;
29:  $S \leftarrow$  the set of  $f$  smallest values in  $Neb_i(r)$ ;
30: if  $x > y$  then
31:   Suppress all the values of  $B$ ;
32:   Suppress the values  $v_j \in S$  such that  $v_j < v_i(r)$ ;
33: else
34:   Suppress all the values of  $S$ ;
35:   Suppress the values  $v_j \in B$  such that  $v_j > v_i(r)$ ;
36: end if
37: enddo;

38: Procedure  $\text{average}(Neb_i(r), v_i(r))$ 
39: do;
40:  $n_i \leftarrow |Neb_i(r)|$ ;
41:  $v_{new} \leftarrow \frac{v_i(r) + \sum_j v_j}{n_i + 1}, v_j \in Neb_i(r)$ ;
42: return  $v_{new}$ ;
43: enddo;

```

3.1 An Iterative Protocol

The pseudo-code (See Algorithm 1) is executed by all the correct nodes during the second part of each round $r \geq 1$.

The execution of the three first lines initializes the three main variables managed by a node p_i : its current round number r , its current value $v_i(r)$ and a multi-set Neb_i which is used to store values received from neighbors. From time to time, Neb_i is reset to *null*, in accordance with a strategy explained later. The rest of the code is divided into two main stages called *gathering* (line 6-11) and *updating* (line 12-24).

Node p_i and its neighbors exchange their values (line 6-7). The received values are logged into the multi-set Neb_i (line 8). A received value can be kept in Neb_i during at most R_c rounds (See the test at line 18). During a round r , p_i receives at most one value from each (correct or Byzantine) node. But if Neb_i has not been reset for several rounds, p_i can receive a value $v_j(r)$ during round r while a value $v_j(r - k)$ previously provided by the same node p_j is already in Neb_i . In that case, p_i keeps only the most recent value. When p_i stops collecting values, all values of Neb_i are sorted into ascending order (line 9).

To guarantee the validity property, a correct node p_i must gather enough values to be allowed to compute a new value (line 14). Otherwise, p_i has to start the next round with the same value (line 17). During round r , the test evaluated by p_i at line 12 defines two favorable cases: either p_i has received values that are greater than or equal to $v_i(r)$ from at least $f + 1$ different nodes, or p_i has received values that are smaller than or equal to $v_i(r)$ from at least $f + 1$ different nodes. If p_i has received less than $f + 1$ values from different nodes, the test cannot be satisfied. If p_i has received values from at least $2f + 1$ different nodes, the test is necessarily satisfied. If p_i has gathered more than f values but less than $2f + 1$ values, the test can be satisfied or not.

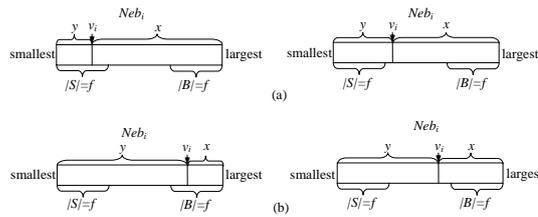


Figure 3: The reducing procedure

When the test of line 12 is satisfied, p_i executes sequentially the *reducing* procedure (line 13) and the *average* procedure (line 14). Reducing operation has been introduced in [1]. To ensure the validity property, a few values have to be removed from the multi-set $Neb_i(r)$. The strategy used in this paper leads to suppress between f and $2f$ values while the strategy used in [1] leads to always ignore exactly $2f$ values. To choose the removed values, p_i compare the received ones with its own current value v_i . Within the set of values $Neb_i(r)$, B is defined as the subset that contains the f largest values (line 28) while S is defined as the subset that contains all the f smallest values (line 29). Due to the fact that the test of line 12 is satisfied, either at least f values in Neb_i are greater than or equal to v_i (case a) or at least f values in Neb_i are smaller than or equal to v_i (case b). The two cases (a and b) are depicted in Figure 3 where the sorted set Neb_i is represented by a rectangle. Note that the two cases are not mutually exclusive. Thus, the two representations of $Neb_i(r)$ that appear on the right side of Figure 3 are equivalent and may lead to suppress $2f$ values (*i.e.* all the values of B and S) if v_i belongs neither to B nor S . Less values will be removed if we consider the two representations on the left side. In case a, only the values of B and the values v_j of S such that $v_j < v_i(r)$ are suppressed from Neb_i . In case b, only the values of S and the values v_j of B such that $v_j > v_i(r)$ are removed.

After reducing, p_i executes the average procedure. p_i considers only the remaining values of $Neb_i(r)$. It calculates average with the values in $Neb_i(r)$ and $v_i(r)$. The weight is simply set to $|Neb_i(r)| + 1$ and $v_i(r + 1)$ is assigned to the computing result. Then $Neb_i(r + 1)$ is reset to *null* (line 15).

3.2 Resetting the Log of Values Neb_i

The variable Neb_i is initialized to *null* (line 3) and can be reset to *null* in two different cases (line 15 and 19). As indicated in the previous paragraph, all the gathered values are suppressed when the node p_i computes a new value during the round r (line 15). Therefore, in the future, this node will only use values issued during a round higher than r .

In a mobile environment, a node is sometimes isolated or at least weakly connected to the rest of the network. The number of values collected during a given round is sometimes low. If a reset is made systematically at the end of each round, the test of line 12 (used to control if enough values have been gathered) is rarely satisfied. By reducing the frequency of reset operations, a node can collect more values over several consecutive rounds. Thus it may take advantage of mobility to increase the number of discovered neighbors. Consequently, the probability that it can frequently calculate a new value increases. Yet the reset operation is very important and is a key element in the proof of the convergence property. A periodic reset operation cleans the system of old values. If a mobile node p_i moves far away and keep some very old values in Neb_i for a long period of time (which is not bounded by a number of rounds), a negative impact on the convergence can be observed.

In the proposed protocol, a general reset is performed by all the correct nodes every R_c rounds (line 18 and 19). By construction, line 19 is executed during a round r such that $r = kR_c$ with $k \geq 1$. The execution of line 3 during the initialization phase can also be considered as a general reset performed during a fictive round numbered 0. We define the set S^c as the set of all the rounds r_c such that the instruction " $Neb_i(r_c) \leftarrow null$ " as been executed either at line 3 or at line 19 of round $r_c - 1$. These rounds are called *common new starting* rounds. By definition, $S^c = \{r_c \mid r_c = kR_c + 1 \text{ with } k \geq 0\}$. The common new starting rounds allow to divide the computation into phases. Each phase is identified by the value of the integer k and is composed of R_c rounds. We define also the concept of *local new starting round* as follows. From the point of view of a correct node p_i , r is a local new starting round, if either $r = 1$ or p_i resets Neb_i to *null* during round $r - 1$. The set of all local new starting rounds of p_i is denoted S_i . Obviously, for any correct node p_i , $S^c \subseteq S_i$. By definition, during a phase, a correct node p_i executes a reset operation at least once (during the last round of the phase) and at most R_c times (each time a new value is computed). The local new starting rounds of a correct node p_i are used to identify some particular joint graphs (See Section 2). Let r be a round executed by p_i . By definition, there exist a unique local new starting round $r_{s_1} \in S_i$ such that $r_{s_1} \leq r$ and for any $r_{s_2} \in S_i$ either $r_{s_2} \leq r_{s_1}$ or $r < r_{s_2}$. Round r_{s_1} is p_i 's latest new starting round. The joint graph corresponding to the union of the communication graphs observed during the non empty sequence of consecutive rounds beginning with r_{s_1} and ending with r is used to identify the nodes which have communicated their values to p_i during this period. In this paper, the notation JN_i^r is used to represent the *joint neighbor set* of p_i at round r .

3.3 Validity Property and Legal Values

Theorem 1. *The proposed protocol satisfies the validity property.*

Proof. Obviously, the property is satisfied during the first round: $\forall p_i \in C_n, v_i(1) \in [v_{min}(1), v_{max}(1)]$. Let us consider that the property is satisfied during any round smaller than or equal to r . To violate the property during round $r + 1$, at least one correct process p_i must modify its value during the execution of the average procedure and must adopt a new value with is either smaller than $v_{min}(1)$ or greater than $v_{max}(1)$. Due to the properties of the average function, at least one value that is either smaller than $v_{min}(1)$ or greater than $v_{max}(1)$ must appear in the multi-set $Neb_i(r)$. A value v contained in this set is either proposed by a correct node or by a Byzantine node. In the first case, due to the induction assumption, v belongs to the range $[v_{min}(1), v_{max}(1)]$. In the second case, v can remains in $Neb_i(r)$ after the execution the reducing procedure only if at least $f + 1$ fake values have been gathered. As the number of Byzantine nodes is bounded by f , the *validity* property is always satisfied. \square

Some works [7] adopt a property which is stronger than the above validity property. During the whole computation, the maximum value proposed by a correct node has to be monotonically non-increasing and similarly the minimum value has to be monotonically non-decreasing. More precisely, for any round $r \geq 1$, the conditions $v_{min}(r) \leq v_{min}(r+1)$ and $v_{max}(r) \geq v_{max}(r+1)$ must hold. The proposed protocol can satisfy this stronger property if and only if $R_c = 1$. When $R_c > 1$, as a correct node p_i may keep old values in its log $Neb_i(r)$, the above conditions are not always true. The new value computed by p_i during round r , namely $v_i(r+1)$, may be less than $v_{min}(r)$ or bigger than $v_{max}(r)$. To take this possibility into account, we define first the concept of *legal value* and then we propose a *safety* property which is stronger than our original *validity* property.

Definition 2. Let r be a round number such that $r = kR_c + m$ with $k \geq 0$ and $1 \leq m \leq R_c$. The value $v_i(r)$ of a correct node p_i is legal if the two conditions $v_i(r) \geq v_{min}(d)$ and $v_i(r) \leq v_{max}(d)$ are satisfied when the round number d is defined as follows:

1. $(k = 0) \wedge (m = 1)$: $d = 1$
2. $(m \neq 1)$: $d = kR_c + 1$
3. $(k \neq 0) \wedge (m = 1)$: $d = (k - 1)R_c + 1$

Lemma 1. $\forall p_i \in C_n, \forall r \geq 1, v_i(r)$ is legal.

Proof. Depending on the round number $r = kR_c + m$, three cases that are mutually exclusive have to be considered. When $m = 1$ and $k = 0$, the value $v_i(1)$ of a correct process p_i is in the range $[v_{min}(1), v_{max}(1)]$. In the two remaining cases, we prove that $v_i(r) \leq v_{max}(d)$. A similar demonstration can be done to conclude that $v_i(r) \geq v_{min}(d)$.

If $m \neq 1$, then r is not a common new starting round. The nearest previous common new starting round is $d = kR_c + 1$. As $m > 1$, we have $r > d$. By definition, at the beginning of round d , every correct node p_j has no value in its set $Neb_j(d)$. Furthermore, at that time, for any correct node p_j , the property $v_j(d) \leq v_{max}(d)$ holds. Now the proof is by contradiction. Let us consider that r is the very first round greater than d during which at least one correct node p_i violates the property. Thus, we have $v_i(r) > v_{max}(d)$. The computation of the value $v_i(r)$ has been done by p_i during the previous round $r - 1$. All the values used during the execution of the average procedure by p_i have been received by p_i during round $r - 1$ and may be during rounds $r - 2, \dots, d + 1$ and d . In all the possible cases, any value v received from a correct node is such that $v \leq v_{max}(d)$. To have still a value greater than $v_{max}(d)$ and thus greater than $v_i(r - 1)$ in its log $Neb_j(r - 1)$ after the execution of the reducing procedure, p_i must gather $f + 1$ fake values. This contradicts both the fact that the network contains at most f Byzantine nodes and the fact that a node (correct or not) cannot insert two different values in the multi-set Neb_i of a correct node p_i .

If $m = 1$ and $k > 0$, then r is a common new starting round. The value $v_i(r)$ has been computed by p_i during the round $r - 1 = kR_c$ and $Neb_i(r - 1)$ may contain values proposed during the R_c previous rounds. As the round $d = (k - 1)R_c + 1$ is also a common new starting round, a similar reasoning leads to conclude that $v_i(r) \leq v_{max}(d)$. \square

Note that after the execution of the reducing procedure, all the remaining values are legal. The following corollary focuses on the common new starting rounds that identify the beginning of phases. This corollary can be considered as our new *safety* property.

Corollary 1. *Safety property:*

$\forall r \in S^c, \forall p_i \in C_n, \forall x \geq 0, v_i(r + x) \geq v_{min}(r)$ and $v_i(r + x) \leq v_{max}(r)$

Proof. As $r \in S^c$, there exists an integer $k \geq 0$ such that $r = kRc + 1$. When $x = R_c$, we have $r + x = (k + 1)Rc + 1$. Due to lemma 1, we conclude directly that the two conditions holds. When x is a multiple of R_c , the proof is also obvious. Finally, when $x = k'R_c + m'$ with $1 \leq m' \leq R_c - 1$, we have $r + x = (k + k')Rc + m' + 1$. Again the proof relies on Lemma 1. \square

4 Related works

Dolev et al. are the first to address the approximate consensus problem in the presence of failures[1]. Under the assumptions that the network is fully connected and the total number of nodes is known, [1] proposes *reducing*, *selecting* and *average* operations and then presents two consensus protocols in a synchronous and an asynchronous environment, separately. In [2], Abraham et al. improve the protocol proposed in [1]: only $3f + 1$ nodes are needed in an asynchronous environment.

Azadmanesh et al. extend approximate consensus to partially connected networks [11, 12]. However without using flooding, they did not completely achieve global convergence. Approximate consensus problem is also addressed in multi-agent system [10, 13, 14, 3, 4]. These protocols are called linear iterative consensus and mainly based on linear control theory and matrix theory. Without Byzantine failure, [10] indicates that in an undirected graph a sufficient and necessary condition for *convergence* consists in having adequate *joint* connected graphs. For a directed graph, [13] points out that a sufficient and necessary condition consists in having a spanning tree contained in adequate *joint* connected graphs. When no Byzantine failure occurs, the speed of *convergence* was analyzed in [14]. Based on the knowledge of the global topology, [3] and [4] address approximate consensus problem in systems where nodes suffer from Byzantine faults.

Without flooding and global topology information, to our knowledge, [8] is the first paper where a solution to the approximate Byzantine consensus problem based on the linear iteration method is proposed. A sufficient condition on the network topology is proposed. When this condition is satisfied, *convergence* is ensured.

While [8] only shows a sufficient condition, [7] and [9] define a sufficient and necessary condition almost simultaneously. Their new arguments are also related to topology. Yet their conditions are static and can not be adapted directly to mobile environments.

Convergence and gathering problems in environments with mobile robots are also similar with approximate consensus. Each robot needs to make the next moving action according to the results returned by its sensors [15]. However they did not consider any topology requirements: each robot can sense all the other ones.

5 Sufficient & Necessary Condition

The sufficient and necessary conditions proposed in previous works [7, 8, 9] consider a static topology. In our mobile system, the topology is not fixed and changes each time a node moves. From *Corollary 1*, we know that each time a common new starting round r is reached, v_{max} can no more increase and v_{min} can no more decrease in the future. But, for example, if the network is partitioned into two disconnected sub-networks, an approximate agreement cannot be reached: nodes that belong to the first group may converge to a value v_1 while the others may converge to a value v_2 . Even if, during each round r , $v_{max}(r)$ can continue to decrease or $v_{min}(r)$ can continue to increase, this does not guarantee that the convergence property will be satisfied.

In this paper a sufficient and necessary condition is proposed. This condition is compatible with the fact that the topology is always changing. Moreover, by its very definition, the proposed condition consider the dynamic evolution of the distribution of values within the system. More precisely, it focuses on the particular correct nodes that have currently either the value v_{min} or the value v_{max} . At least one

of these nodes has to receive from its neighborhood enough messages (quantity constraint) that contain values different from its own current value (quality constraint).

To formally define what is expected in terms of quality, we first provide the definition of a *proper value*. The problem definition (See Section 2) refers to a parameter ϵ which sets the level of precision that needs to be obtained to consider that an agreement is reached. We introduce a second parameter called δ whose range of possible values is $(0, \frac{\epsilon}{2}]$. This parameter, which is not used in the protocol, is necessary to define the condition and the notion of proper value on which the condition relies. This non-zero positive integer (whose value may be very small) allows us to define five intervals of values as follows. When a common new starting round $r \in S^c$ begins, all correct nodes have values in the range $[v_{min}(r), v_{max}(r)]$. Five value intervals are defined. *Minimum value* corresponds to the value $v_{min}(r)$. *Maximum value* corresponds to the value $v_{max}(r)$. *Nearly minimum value* represents the value interval $(v_{min}(r), v_{min}(r) + \delta)$. Symmetrically, *Nearly maximum value* represents the value interval $(v_{max}(r) - \delta, v_{max}(r))$. Finally, *Middle value* represents the value interval $[v_{min}(r) + \delta, v_{max}(r) - \delta]$. These three intervals are defined at the beginning of a new phase (*i.e.* just before a round $r = kR_c + 1$ begins) and will not change during R_c consecutive rounds. During a round r' of the phase k ($r \leq r' < r + R_c$), a correct node p_i may change its value v_i . Depending on its value $v_i(r')$, a correct node is classified in one of the following five groups: $CMin(r')$, $CNin(r')$, $CMid(r')$, $CNax(r')$, and $CMax(r')$. The letter C at the beginning of the name of a group indicates that the members of the group are correct nodes. Throughout a phase k , the same interval of value is associated to a group. During a round r' , the rule for assigning a correct node p_i to a group is simple: the value $v_i(r')$ must belong to the corresponding interval. During a round, a correct node belongs to exactly one group. By definition, the distribution of the nodes into the five groups may change at each round of a phase. Figure 4(a) summarize the above discussion.

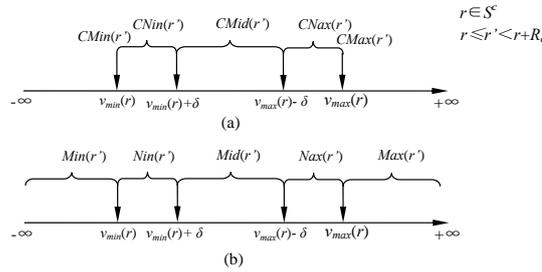


Figure 4: Value intervals and node sets

Byzantine nodes may exist in the system. They can propose values that belong to $[v_{min}(r), v_{max}(r)]$ but also values that are beyond this interval. If Byzantine nodes propose legal values, they also make sense. We use $Max(r')$, $Min(r')$, $Nax(r')$, $Nin(r')$ and $Mid(r')$ to represent groups that mix correct nodes and Byzantine nodes. Due to the Byzantine nodes, the value intervals corresponding to $Min(r')$ and $Max(r')$ are enlarged respectively to $(-\infty, v_{min}(r)]$ and $[v_{max}(r), +\infty)$. Of course, $CMax(r') \subseteq Max(r')$, $CNax(r') \subseteq Nax(r')$, $CMid(r') \subseteq Mid(r')$, $CNin(r') \subseteq Nin(r')$, and $CMin(r') \subseteq Min(r')$. Node that a Byzantine node can belong to several group during the same round. These five new groups are depicted in Figure 4(b). Except Max and Min , the three other groups can be empty. Nin and Nax are two special groups which are used to distinguish legal values that are sufficiently different from either the current minimal value v_{min} or the current maximal value v_{max} .

The proposed condition only affects the nodes that have either the minimum value or the maximum value when a phase k begins. Let $r = kR_c + 1$ be the first round of this phase. The targeted nodes belong to $CMin(r)$ or $CMax(r)$.

Definition 3. For any common new starting point $r \in S^c$ and for any round r' such that $r \leq r' < r + R_c$,

a proper value, from the point of view of a correct node that belongs to $CMin(r)$ is a value $v_j(r')$ that belongs to the interval $[v_{min}(r) + \delta, +\infty)$ while a proper value, from the point of view of a correct node that belongs to $CMax(r)$ is a value $v_j(r')$ that belongs to the interval $(-\infty, v_{max}(r) - \delta]$.

Note that a proper value is not necessarily a legal value. A proper value can be a fake value proposed by a Byzantine node. Based on the above definition, we can now express the proposed condition.

Theorem 2. *The convergence property is satisfied by the proposed protocol if the following sufficient condition always holds:*

$$\begin{aligned} &\forall r \in S^c \text{ such that } v_{max}(r) - v_{min}(r) \geq \epsilon, \\ &\exists p_i \in C_n \text{ such that } p_i \in CMax(r) \cup CMin(r), \\ &\exists r' \text{ such that } r \leq r' < r + R_c, \\ &\exists V_q \in V \text{ such that } V_q \subseteq JN_i^{r'}, \\ &|V_q| \geq f + 1 \text{ and } \forall p_j \in V_q, v_j(r') \text{ is a proper value.} \end{aligned}$$

As long as the convergence test is not satisfied, for each phase (characterized by its associated common new starting round r), at least one correct node p_i among those which are members of the groups $CMin(r)$ or $CMax(r)$ must, at least once during the phase (*i.e.* during a round r'), compute a new value using $f + 1$ (quantity constraint) proper values (quality constraint) received during the current phase. As mentioned before, a proper value is not a legal value. But, due to the fact that at least $f + 1$ are gathered by p_i , at least one of them is not removed during the reducing procedure and is a legal value. Note that the above condition does not ensure that either v_{min} increases or v_{max} decreases during a phase. In fact, several correct nodes may have the minimum value or the maximum value when the phase begins. The condition just ensures that at least one of them will increase or decrease its value.

Now the reason why Nax and Nin have been defined is explained. The requirements expressed in *Theorem 2* only focus on the nodes of the groups $CMin(r)$ and $CMax(r)$: the other correct nodes are not concerned. Suppose $v_{max}(r) - v_{min}(r) \geq \epsilon$ and values from Nax and Nin are not excluded. In this situation, if the nodes that belong to $CMin(r)$ and $CMax(r)$ only receive values from respectively $(v_{min}(r), v_{min}(r) + \delta)$ and $(v_{max}(r) - \delta, v_{max}(r))$ and if all the other correct nodes do not change their values, then there may exist a value $\mu \geq \epsilon$, such that $\lim_{r \rightarrow +\infty} v_{max}(r) - v_{min}(r) = \mu$.

Mobility has a strong impact on the fact that the condition can be satisfied or not. If some correct nodes are always moving far away, convergence can not be obtained. In fact, a correct node can remain isolated as long as it is neither in $CMax(r)$ nor in $CMin(r)$. But, of course, in many applications, a correct node can not always determine if it is currently concerned or not by the condition. The fact that the nodes in $CMin(r)$ or $CMax(r)$ can obtain enough proper values depends not only on the trajectory and the speed of the correct nodes. It depends also on the cardinality of the system (*i.e.* the cardinality of the five sets). For example, if the cardinality of the union set $\bigcup_{r'} (Max(r') \cup Nax(r') \cup Mid(r'))$ ($r' \in \{r, r + 1, \dots, r + R_c - 1\}$) is smaller than $f + 1$, no node of $CMin(r)$ has the possibility to meet both the quantity and quality constraint. However, in that case, the cardinality of the union set $\bigcup_{r'} (Min(r') \cup Nin(r') \cup Mid(r'))$ should be sufficient to ensure that at least one node that belongs to $CMax(r)$ can collect enough proper values.

Lemma 2. *To ensure that at least one node (either in $CMin$ or in $CMax$) can collect enough proper values, the cardinality of the system must satisfy the following constraint: $n \geq 3f + 1$.*

Proof. Suppose that there is only $3f$ nodes in the network. Think about this situation $|CMax| = 1$, $|CMin| = 1$, $|CNax| = f - 1$, $|CNin| = f - 1$ and $|CMid| = 0$. Moreover, the f Byzantine nodes propose values bigger than v_{max} to the nodes in $CMax$ and, at the same time, the f byzantine nodes propose values smaller than v_{min} to nodes in $CMin$. In that case only f proper values can be seen

by nodes in $CMax$ and $CMin$. The cardinality of the system is not sufficient to provide "quantity" and "quality" simultaneously.

While $v_{max} - v_{min} \geq \epsilon$, if $n = 3f + 1$, there is always a chance to satisfy at least one node in $CMax$ or $CMin$. The proof is by contradiction. Suppose there is no chance to gather enough proper values neither for the nodes in $CMax$ nor for those in $CMin$. Suppose Byzantine nodes send f illegal values bigger than v_{max} to nodes in $CMax$ and send f values smaller than v_{min} to nodes in $CMin$ or just keep silent. In that way the Byzantine nodes do not contribute to the satisfaction of the condition. The remaining $2f + 1$ are all correct nodes. Suppose $|CMid| = 0$, because any nodes belongs to $CMid$ helps both the nodes of $CMin$ and $CMax$ to satisfy the constraints. So according to the pigeonhole principle, at least one of the following two inequalities must be true: $|CMax + CMax| \geq f + 1$ or $|CMin + CMin| \geq f + 1$. A contradiction. \square

Let us now consider that the system is populated with a sufficient number of correct nodes: $n \geq 3f + 1$. Even if the nodes travel arbitrarily within the system, we assume that the condition is always satisfied. First we prove two general lemmas related to the convergence property. By definition, the convergence property is a stable property. Once the convergence is reached, this property remains true.

Lemma 3. *Let r be a common new starting point. If $v_{max}(r) - v_{min}(r) < \epsilon$ then convergence is already reached when round r starts.*

Proof. Let us assume that $v_{min}(r)$ is proposed by a correct node p_i while $v_{max}(r)$ is proposed by a correct node p_j . By definition, for any correct node p_k , $v_{min}(r) \leq v_k(r) \leq v_{max}(r)$. As $r \in S^c$, due to Corollary 1, $\forall r' \geq r$, $v_{min}(r) \leq v_k(r') \leq v_{max}(r)$. Therefore, $\forall r' \geq r$, $v_{min}(r) \leq v_{min}(r')$ and $v_{max}(r') \leq v_{max}(r)$. Consequently, as $v_{max}(r) - v_{min}(r) < \epsilon$, $\forall r' \geq r$, $v_{max}(r') - v_{min}(r') < \epsilon$. Thus convergence is already reached when round r begins. \square

Note that the fact that r is a common starting round is essential in the proof of Lemma 3. If r is not a common starting round, it could be the case that $v_{max}(r) - v_{min}(r) < \epsilon$ while $v_{max}(r+1) - v_{min}(r+1) \geq \epsilon$.

Lemma 4. *When a common new starting round r begins, convergence is already reached if and only if either $CMin(r) = CMax(r)$ or $CMax(r) \cap CMin(r) \neq \emptyset$.*

Proof. By definition, if convergence is already reached during a round r (that belongs or not to S^c), either all the values of the correct nodes are equal or they differ by at most ϵ . In the first case, $CMin(r) = CMax(r)$ and $CMax(r) \cap CMin(r) \neq \emptyset$. In the second case, $CMin(r) \neq CMax(r)$ and $v_{max}(r) - v_{min}(r) < \epsilon$. As $\delta \leq \frac{\epsilon}{2}$, we have $v_{max}(r) - \delta < v_{min}(r) + \delta$. Thus $CMax(r) \cap CMin(r) \neq \emptyset$. The first implication holds. To prove the second implication, let us first consider that $CMin(r) = CMax(r)$. Due to Corollary 1, all the correct nodes will keep the common value in the future. Now if $CMax(r) \cap CMin(r) \neq \emptyset$, there exist at least one correct node p_i who has proposed a value $v_i(r)$ which belongs both to $(v_{min}(r), v_{min}(r) + \delta)$ and $(v_{max}(r) - \delta, v_{max}(r))$. Thus $v_{max}(r) - \delta < v_{min}(r) + \delta$. By assumption, δ belongs to $(0, \frac{\epsilon}{2}]$. Therefore, $v_{max}(r) - v_{min}(r) < 2\delta \leq \epsilon$. Again, due to Corollary 1, convergence is already reached when round r begins. \square

Lemma 5. *Let $r \in S^c$ be a common new starting round such that convergence is not yet reached when r starts. Let ω_1 be a positive integer ($\omega_1 \geq 1$). Let ς_1 and ς_2 be two reals that belong to $(0, 1]$ and such that:*

$$v_{min}(r + \omega_1 R_c) = \varsigma_1 v_{min}(r)$$

$$v_{max}(r + \omega_1 R_c) = \varsigma_2 v_{max}(r)$$

During the ω_1 phases, if neither the minimal value increases ($\varsigma_1 = 1$) nor the maximal value decreases ($\varsigma_2 = 1$) then the two following predicates are both satisfied:

1. ($| CMin(r + \omega_1 R_c) | < | CMin(r) |$) or ($| CMax(r + \omega_1 R_c) | < | CMax(r) |$)
2. $\omega_1 < n$

Proof. Due to Corollary 1, it is obvious that we can rewrite $v_{min}(r + R_c)$ and $v_{max}(r + R_c)$ using the defined ς_1 and ς_2 . Let us assume that the minimal and the maximal values are stable during the $\omega_1 R_c$ rounds: $\varsigma_1 = 1$ and $\varsigma_2 = 1$. We demonstrate that, after each phase, the cardinality of at least one of the two sets decrease. The fact that the property holds when $\omega_1 = 1$ allows us to conclude that the property holds for any value of ω_1 . During the R_c numbered $r, r + 1, \dots, r + R_c - 1$, due to the necessary condition expressed in Theorem 2, there exists at least one round during which a "good" phenomena occurs. Let us consider the highest round r' during which the condition is true and let p_i be a node such that p_i has gathered enough proper values: $\exists V_q \subseteq JN_i^{r'}$ such that $|V_q| \geq f + 1$ and $\forall p_j \in V_q, v_j(r')$ is a proper value. Without loss of generality, let us assume that p_i belongs to $CMin(r)$. Due to Corollary 1, $v_i(r') \geq v_{min}(r)$. Moreover, due to the condition, the reducing procedure and the average procedure are executed by node p_i . During the reducing procedure, at least one value greater or equal to $v_{min}(r) + \delta$ is not removed. In the worst case, all the other values used during the computation are equal to $v_{min}(r)$. Even in that case, the new computed value of p_i is such that $v_i(r' + 1) > v_{min}(r)$. If r' is not the last round of the phase (and despite the fact that r' is the highest round of the phase during which the condition is true), it could be the case that p_i computes again its new value during rounds r'' such that $r' < r'' < r + R_c$. In the worst case, p_i will compute the average between its own value ($> v_{min}(r)$) and a set of gathered values all equal to $v_{min}(r)$. Thus, when the next phase begins, $v_i(r + R_c) > v_{min}(r)$. Consequently, if $v_{min}(r + R_c) = v_{min}(r)$, at least one node (namely p_i) belongs to $CMin(r)$ but not to $CMin(r + R_c)$. A similar reasoning can be adopted if p_i belongs to $CMax(r)$. As the number of nodes is finite, and as at least one node per phase is removed from either $CMin$ or $CMax$, we can conclude that v_{min} and v_{max} both remain stable during at most $n - 1$ phases if convergence was not yet reached during round r . It is always the case that $v_{min}(r + nR_c) > v_{min}(r)$ or $v_{max}(r + nR_c) < v_{max}(r)$. Thus the second predicate $\omega_1 < n$ also holds. □

We prove now Theorem 2.

Proof. (Theorem 2)

Let r be a starting round during which convergence is not yet achieved. From Lemma 5, we can conclude that there exist a positive integer ω_1 such that either $v_{min}(r + \omega_1 R_c) = \varsigma_1 v_{min}(r)$ with $\varsigma_1 \in (0, 1)$ or $v_{max}(r + \omega_1 R_c) = \varsigma_2 v_{max}(r)$ with $\varsigma_2 \in (0, 1)$. Let $d = v_{max}(r) - v_{min}(r)$. In the first case we have:

$$d > v_{max}(r + \omega_1 R_c) - v_{min}(r).$$

In the second case,

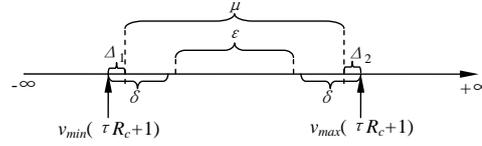
$$d > v_{max}(r) - v_{min}(r + \omega_1 R_c). \text{ Due to Corollary 1, } v_{min}(r) \leq v_{min}(r + \omega_1 R_c) \text{ and } v_{max}(r) \geq v_{max}(r + \omega_1 R_c). \text{ Therefore, in both cases: } d > v_{max}(r + \omega_1 R_c) - v_{min}(r + \omega_1 R_c).$$

The difference $v_{max} - v_{min}$ will always decrease. Yet this is not sufficient to prove that eventually convergence is reached. To prove this last point, we have to show the existence of a finite integer τ , such that: $v_{max}(\tau R_c + 1) - v_{min}(\tau R_c + 1) < \epsilon$.

The proof is by contradiction. Suppose that the above condition is never satisfied. In other words, whatever the value of the integer τ , the difference $v_{max}(\tau R_c + 1) - v_{min}(\tau R_c + 1)$ only approaches a value μ but $\mu \geq \epsilon$. Figure 5 depicts such a scenario. In this representation we assume that there exists always a real Δ such that: $\Delta \geq 0$ and $v_{max}(\tau R_c + 1) - v_{min}(\tau R_c + 1) = \Delta + \mu$.

In Figure 5, Δ is represented by the sum of Δ_1 and Δ_2 .

Due to Lemma 5, when τ approaches to infinity, Δ approaches to 0. We will show that μ is not a limit: the value of Δ may become less than zero and never become positive again. First we will show that there exists a particular positive value of Δ such that after a single execution of the average procedure


 Figure 5: Example of value μ

during a round r , the value of Δ becomes negative. Then we show that there exists a particular positive value of Δ such that after a finite number of execution of the procedure average, the value of Δ remains negative forever.

First let us consider a particular phase k . Let $r = (k - 1)R_c + 1$ be the first round of this phase. Let d denotes the difference $v_{max}(r) - v_{min}(r)$. Let us assume that at the beginning of phase k , the value of Δ is positive and equal to $d - \mu$. Due to Lemma 5, there exists a round r' such that: $CMin(r) = CMin(r')$ and $CMax(r) = CMax(r')$ and $(CMin(r) \neq CMin(r' + 1) \text{ or } CMax(r) \neq CMax(r' + 1))$.

Let d' be the difference $v_{max}(r' + 1) - v_{min}(r' + 1)$. We have $d > d'$. The value of Δ decreases by $d - d'$ between round r and round $r' + 1$. The value of Δ is equal to $d' - \mu$ during round $r' + 1$. This value is negative if $d' < \mu$. First we compute an estimation of the minimal value $d - d'$ that can be observed. Obviously, to ensure that the difference $d - d'$ is as small as possible, either just the minimal value has to increase or just the maximal value has to decrease (but not both during the same round r'). As the two cases are symmetric, let us consider that the minimal value increases while the maximal one remains stable. Let us consider a node p_i such that p_i has the minimal value during round r' . This node may have again the minimal value during round $r' + 1$. In that case, $v_{min}(r' + 1) > v_{min}(r')$ and $d - d' = v_{min}(r' + 1) - v_{min}(r')$. Our aim is to obtain an estimation (more precisely an under-estimation) of the difference $d - d'$. To be allowed to compute a new value, the node p_i must gather at least $f + 1$ proper values (in the worst case, these value can be equal to $v_{min}(r') + \delta$). After the reducing procedure, p_i keeps at most $n - f$ values. Furthermore at least one correct node propose a value greater than μ (otherwise this contradict the fact that the limit μ is respected between round r and r'). Yet as our goal is just to provide an under-estimation, we consider an (unrealistic) worst case. The new value of node p_i computed during round r' during the execution of the average procedure is the average between n values where $n - 1$ are equal to $v_{min}(r')$ and a single one is a proper value (more precisely, the minimal proper value, namely $v_{min}(r') + \delta$). In that case, we have:

$$v_{min}(r' + 1) > \frac{(n-1)v_{min}(r') + (v_{min}(r') + \delta)}{n}$$

As the right part of the above formula is an under-estimation, we use the symbol " $>$ " rather than the symbol " \geq ". Based on the previous formula, we conclude that:

$$d - d' > v_{min}(r') + \frac{\delta}{n} - v_{min}(r')$$

We have proved the existence of a particular positive value of Δ (during round r') that is small enough to imply that the value of Δ can be negative during the next round. If during round r' , the value of Δ (which is equal to $d - \mu$) is strictly less than $\frac{\delta}{n}$, then the value of Δ can be negative during round $r' + 1$. This contradict the fact that μ is a limit for $v_{max} - v_{min}$ which is never violated.

At this stage, we have just demonstrated that μ is not a limit. We now show that after some time, this limit will be breached permanently. To understand why a violation of the limit μ is sometimes transient, let us consider that during a phase a single node p_i has the smallest value $v_{min}(r)$. During the first round of this phase, it may broadcast this value to all the other nodes and then compute a new value $v_{min}(r')$ which violates the limit μ . Unfortunately any other node may compute again a new value based on its own value and old values contained in their log that are less than $v_{min}(r')$ and possibly very closed from $v_{min}(r)$. As a consequence the value of p_i may decrease again and respect again the limit μ .

Let us consider a common new starting round r such that the limit μ is respected. Due to Lemma 5, after $n - 1$ phases, either all the nodes that have the minimal value or all the nodes that have the maximal

value during round r have now adopted either an higher value or respectively a smaller value. Again without loss of generality, let us consider the worst case where all the nodes (except one) were sharing the minimal value $v_{min}(r)$ during round r while a single node has a value equal to $v_{min}(r) + \mu$. Again, our goal is to identify a limit x (even if this one is under-estimated) that shows that at a beginning of round $r + nR_c + 1$, no value less than $v_{min}(r) + x$ remains in the system. Again, during each phase, at least one node p_i , which has the minimal value when the phase begins modifies its value and adopts during a round of the phase, a value which is at least equal to $v_{min}(r) + \frac{\delta}{n}$ (See the above discussion). In the worst case, this change occurs during the first round of the first phase denoted r . Then p_i may compute again its value during the next $nR_c - 1$ following rounds. If it receives only values that are equal to v_{min} during these rounds, its value at the end of the phase is strictly greater than $v_{min}(r') + \frac{\delta}{n^{nR_c}}$. Once this last phase ends, the value of p_i can no more decrease. Therefore, after at most n phases, all the correct nodes have a value that will remain greater than $v_{min}(r) + \frac{\delta}{n^{nR_c}}$. Consequently there exists a positive value of Δ such that the violation of the limit μ is permanent.

The condition is sufficient to ensure convergence. □

Regarding the fact that the condition is necessary, we identify a weaker condition. Indeed, the condition does not have to be satisfied in each phase but only infinitely often. This modification of the condition has no major impact on the way we prove that the condition is a sufficient condition. Some properties are no more satisfied "at the end of each phase" but "after a finite number of phases".

To prove that the condition is necessary, we show that the quantity constraint and the quality constraint are both necessary. Within the set of n nodes, let us assume that half of the $n - f > 2f$ correct nodes share a same value v_{min} while the second half share the value v_{max} . We assume that v_{min} and v_{max} are such that the convergence is not yet reached. If a correct node gathers only f proper values before computing its new value, it could be the case that the values that remain after the execution of the reducing procedure are all equal to its own value. Thus no correct node will change its value. If a correct node gathers $f + 1$ values but at least one of them is not a proper value, it is also possible that all the proper values will be removed during the reducing procedure. Again, the values of the correct nodes will be stable and convergence is not ensured.

6 Conclusion

In this paper, we addressed approximate Byzantine consensus problem in partially connected mobile networks. An architecture for both moving and consensus protocol has been proposed. Then an approximate consensus protocol based on a linear iteration method has been presented. In order to take advantage of mobility, in this protocol, nodes are allowed to collect messages during at most R_c consecutive rounds. Afterwards, we have defined a sufficient and necessary condition that allows to satisfy *convergence*. Compared to existing papers, this novel condition is dynamic and not "universal". It only focuses on the correct nodes which propose the maximum or the minimum value and requires that, from time to time, at least one of them should receive enough (quantity constraint) proper (quality constraint) values. Our analysis shows that if $n \geq 3f + 1$, the condition has chances to be satisfied and consensus can be reached. We are now working on particular mobility scenarios where either the existence of some meeting points or a predefined trajectory and scheduling allow to prove that the condition is satisfied. Simulations are also conducted to analyze the impact of a tuning of the R_c parameter.

Acknowledgment

This work is partially supported by Natural Science Foundation, China under grant 60973122 and National 863 Hi-Tech Program, China under grant 2011AA040502. This work is partially supported by the ANR French national program for Security and Informatics (grant #ANR-11-INSE-010, project AMORES).

References

- [1] D. Dolev, A. N. Lynch, S. Pinter, W. E. Stark, and E. W. Weihl. Reaching approximate agreement in the presence of faults. In *Proc. of 3rd IEEE Symp. on Reliability in Distributed Software and Database Systems*, pages 145–154, 1983.
- [2] I. Abraham, Y. Amit, and D. Dolev. Optimal resilience asynchronous approximate agreement. In *Proc. of the 8th Int. Conf. on Principles of Distributed Systems*, volume 3544 of *LNCS*, pages 229–239, 2005.
- [3] S. Sundaram and C. N. Hadjicostis. Distributed function calculation via linear iterations in presence of malicious agents - part i: Attacking the networks. In *Proc. of the American Control Conference*, pages 1350–1355, 2008.
- [4] S. Sundaram and C. N. Hadjicostis. Distributed function calculation via linear iterations in presence of malicious agents - part ii: Overcoming malicious behavior. In *Proc. of the American Control Conference*, pages 1356–1361, 2008.
- [5] F. Pasqualetti, A. Bicchi, and F. Bullo. Consensus computation in unreliable networks: A system theoretic approach. *IEEE Trans. on Automatic Control*, 57(1):90–104, 2012.
- [6] Wei Ren, Randal W. Beard, and Ella M. Atkins. A survey of consensus problems in multi-agent coordination. In *Proc. of American Control Conference*, pages 1859–1864, 2005.
- [7] N. Vaidya, L. Tseng, and G. Liang. Iterative approximate byzantine consensus in arbitrary directed graphs. In *Proc. of 31st Symp. on Principles of Distributed Computing*, 2012.
- [8] H. Zhang and S. Sundaram. Robustness of information diffusion algorithms to locally bounded adversaries. *CoRR*, abs/1110.3843, 2011.
- [9] H. LeBlanc, H. Zhang, S. Sundaram, and X. Koutsoukos. Consensus of multi-agent networks in the presence of adversaries using only local information. In *Proc. of the 1st int. conf. on High Confidence Networked Systems*, pages 1–10, 2012.
- [10] A. Jadbabaie, J. Lin, and A. S. Morse. Coordination of groups of mobile autonomous agent using nearest neighbor rules. *IEEE Trans. on Automatic Control*, 48(6):988–1001, 2003.
- [11] M. H. Azadmanesh and A. W. Krings. A step toward global convergence in partially connected networks. In *Proc. of Conf. on Parallel and Distributed Computing Systems*, pages 234–241, 1997.
- [12] M. H. Azadmanesh and H. Bajwa. Global convergence in partially fully connected networks (pfcn) with limited relays. In *Proc. of the 27th Conf. of the IEEE Industrial Electronics Society*, pages 2022–2025, 2001.
- [13] W. Ren and R. W. Beard. Consensus seeking in multi-agent systems under dynamically changing interaction topologies. *IEEE Trans. on Automatic Control*, 50(5):655–661, 2005.
- [14] Y. Kim and M. Mesbahi. On maximizing the second smallest eigenvalue of a state-dependent graph laplacian. *IEEE Trans. of Automatic Control*, 51(1):116–120, 2006.
- [15] T. Izumi, Z. Bouzid, S. Tixeuil, and K. Wada. The bg-simulation for byzantine mobile robots. In *Proc. of DISC*, volume 6950 of *LNCS*, pages 330–331, 2011.

Contents

1	Introduction	3
2	Model and Problem Definition	4
2.1	Model	4
2.2	Definition of the Agreement Problem	5
3	The Protocol and its Safety Proof	5
3.1	An Iterative Protocol	6
3.2	Resetting the Log of Values Neb_i	8
3.3	Validity Property and Legal Values	8
4	Related works	10
5	Sufficient & Necessary Condition	10
6	Conclusion	16



**RESEARCH CENTRE
RENNES – BRETAGNE ATLANTIQUE**

Campus universitaire de Beaulieu
35042 Rennes Cedex

Publisher
Inria
Domaine de Voluceau - Rocquencourt
BP 105 - 78153 Le Chesnay Cedex
inria.fr

ISSN 0249-6399