Byzantine Vector Consensus in Complete Graphs

Authors: Nitin H. Vaidya, Vijay K. Garg

Presented by: Chaoqi Wang

Department of Computer Science,
The University of Toronto

7, Dec 2017

Byzantine Vector Consensus Model

Configurations:

1. fully connected network.
2. n processes, with at most f Byzantine processes.
3. d-dimensional real-valued vector as input.

Byzantine Vector Consensus Model

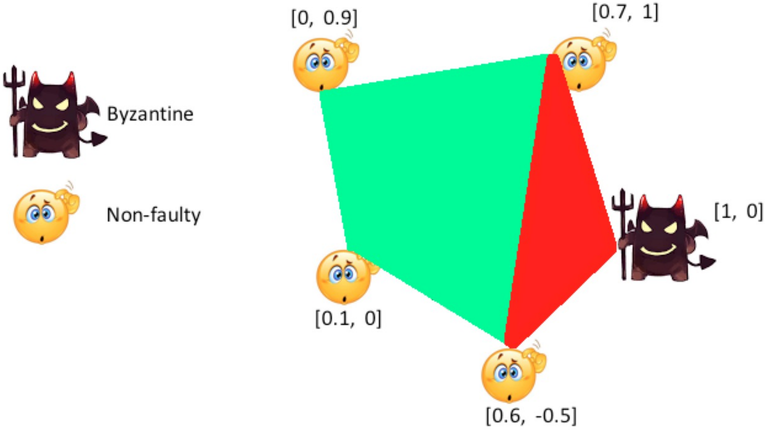
Configurations:

1. fully connected network.
2. n processes, with at most f Byzantine processes.
3. d-dimensional real-valued vector as input.

Conditions:

1. **Termination:** Each non-faulty process must terminate after a finite amount of time.
2. **Agreement:** The decision (or output) vector at all the non-faulty processes must be identical.
3. **Validity:** The decision vector at each non-faulty process must be in the **convex hull** of the input vectors at the non-faulty processes.

Byzantine Vector Consensus Model



An example with 4 non-faulty processes and 1 byzantine process.

Solution?

Can we simply perform Byzantine Agreement on each dimension of the input vectors independently?

Solution?

Can we simply perform Byzantine Agreement on each dimension of the input vectors independently?
No!

Solution?

Can we simply perform Byzantine Agreement on each dimension of the input vectors independently?

No!

Counterexample:

We have 4 process, and only one is faulty.

p_0 : is faulty.

$p_1 : [1, 0, 0]$

$p_2 : [0, 1, 0]$

$p_3 : [0, 0, 1]$

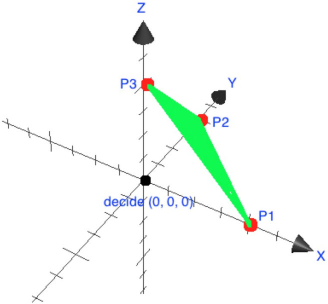
Solution?

Can we simply perform Byzantine Agreement on each dimension of the input vectors independently?

No!

Counterexample:

p_0 : is faulty.
p_1 : $[1, 0, 0]$
p_2 : $[0, 1, 0]$
p_3 : $[0, 0, 1]$



if we perform Byzantine Agreement on each dimension of the vectors separately, then the processes may possibly agree on $[0, 0, 0]$. In this case, the *Validity* condition is violated!

Results in the paper

In this paper, the authors obtain the following two results for BVC in *complete graph* while tolerating up to f Byzantine failures when the input is a d-dimensional vector:

1. For a synchronous system, $n > max(3f, (d+1)f)$ is necessary and sufficient for achieving BVC.

Results in the paper

In this paper, the authors obtain the following two results for BVC in *complete graph* while tolerating up to f Byzantine failures when the input is a d-dimensional vector:

1. For a synchronous system, $n > max(3f, (d+1)f)$ is necessary and sufficient for achieving BVC.
2. For an asynchronous system, $n > (d+2)f$ is necessary and sufficient to achieve *approximate* BVC.

Necessary condition

Theorem 1 *$n > max(3f, (d+1)f)$ is necessary for BVC in a synchronous system.*

Proof (Necessary)

Theorem 1 *$n > max(3f, (d+1)f)$ is necessary for BVC in a synchronous system.*

1. When $d = 1$, $n > 3f$ is a necessary condition for achieving Byzantine agreement in presence of up to f faults. (Already proved in the textbook!)

Proof (Necessary)

Theorem 1 *$n > max(3f, (d+1)f)$ is necessary for BVC in a synchronous system.*

1. When $d = 1$, $n > 3f$ is a necessary condition for achieving Byzantine agreement in presence of up to f faults. (Already proved in the textbook!)
2. When $d \geq 2$, $n > (d+1)f$ is also a necessary condition.

Proof (Necessary)

Theorem 1 $n > max(3f, (d+1)f)$ *is necessary for BVC in a synchronous system.*

Consider the validity condition: **decision vector should be in the convex hull of non-fault processes' inputs !**

Proof (Necessary)

Theorem 1 *$n > max(3f, (d+1)f)$ is necessary for BVC in a synchronous system.*

Consider the validity condition: **decision vector should be in the convex hull of non-fault processes' inputs !**

Claim: Suppose f $= 1$, since none of the non-faulty process know which process is faulty, the decision vector v must be in the convex hull of each multiset containing the input vectors of $n-1$ of the processes (there are n such multiset, let its convex hull be Q_i for $1 = 1, 2, ..., n$).

Proof (Necessary)

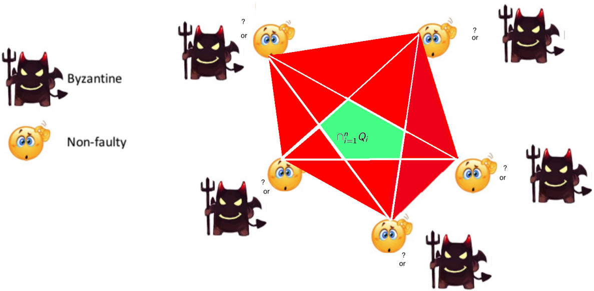
Theorem 1 *$n > max(3f, (d+1)f)$ is necessary for BVC in a synchronous system.*

Consider the validity condition: **decision vector should be in the convex hull of non-fault processes' inputs !**

Claim: Suppose $f = 1$, since none of the non-faulty process know which process is faulty, the decision vector v must be in the convex hull of each multiset containing the input vectors of $n-1$ of the processes (there are n such multiset, let it be Q_i for $1 = 1, 2, ..., n$).

1. p_i is the input of process i
2. $\mathcal{P} = \{p_1, p_2, ..., p_n\}$
3. $H(\mathcal{P})$ is the convex hull of $\mathcal{P}$.
4. $Q_i = \mathcal{H}(\mathcal{P} - \{p_i\})$
5. $v \in \cap_{i=1}^{n} Q_i$

Proof (Necessary)



Proof (Necessary)

Theorem 1 *$n > max(3f, (d+1)f)$ is necessary for BVC in a synchronous system.*

Proof:

1. $f = 1$, input vector $p_i, 1 \leq i \leq d$, is a vector whose i-th element is 1 and the remaining are 0. p_{d+1} is the all-0 vector.

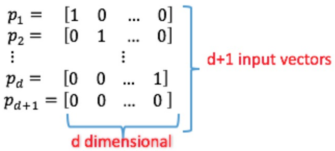
$$\left.\begin{array}{ll} p_1 = & [1 \quad 0 \quad \dots \quad 0] \\ p_2 = & [0 \quad 1 \quad \dots \quad 0] \\ \vdots & \qquad\qquad \vdots \\ p_d = & [0 \quad 0 \quad \dots \quad 1] \\ p_{d+1} = & \underbrace{[0 \quad 0 \quad \dots \quad 0]}_{\text{d dimensional}} \end{array}\right\} \text{d+1 input vectors}$$

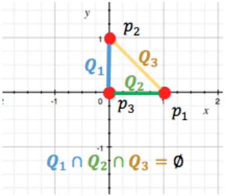
Proof (Necessary)

Theorem 1 *$n > max(3f, (d+1)f)$ is necessary for BVC in a synchronous system.*

Proof:

1. $f = 1$, input vector $p_i, 1 \le i \le d$, is a vector whose i-th element is 1 and the remaining are 0. p_{d+1} is the all-0 vector.





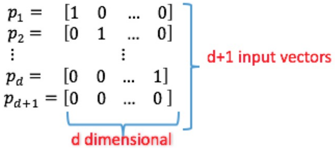
Let $\mathcal{P} = \{p_1, .., p_{d+1}\}$ and Q_i is the convex hull of $\mathcal{P} - \{p_i\}$, then $\cap_{i=1}^{d+1} Q_i = \emptyset$.

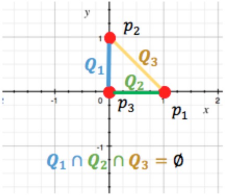
Proof (Necessary)

Theorem 1 *$n > max(3f, (d+1)f)$ is necessary for BVC in a synchronous system.*

Proof:

1. $f = 1$, input vector $p_i, 1 \leq i \leq d$, is a vector whose i-th element is 1 and the remaining are 0. p_{d+1} is the all-0 vector.





Let $\mathcal{P} = \{p_1, .., p_{d+1}\}$ and Q_i is the convex hull of $\mathcal{P} - \{p_i\}$, then $\cap_{i=1}^{d+1} Q_i = \emptyset$.
Which leads to $n \leq d+1$ is not sufficient. Therefore, $n \geq d+2$ is necessary for the case when $f = 1$.

Proof (Necessary)

Theorem 1 *$n > max(3f, (d+1)f)$ is necessary for BVC in a synchronous system.*

For the following two cases.

1. $f = 1$, the input vector for $p_i, 1 \leq i \leq d$, is a vector whose i-th element is 1 and the remaining are 0. p_{d+1} is the all-0 vector.
2. $f > 1$, we can use the simulation approach, and thus f simulated process are implemented by a single process.

Proof (Necessary)

1. ...
2. $f > 1$, we can use the simulation approach. That is, we use a single process to simulate f processes. Therefore, if a correct algorithm were to exist for tolerating up to f faults among $(d+1)f$ processes, then we can obtain a correct algorithm to tolerate a single failure among $d+1$ processes. Contradict to the case $f = 1$.

Proof (Necessary)

1. ...
2. $f > 1$, we can use the simulation approach. That is, we use a single process to simulate f processes. Therefore, if a correct algorithm were to exist for tolerating up to f faults among $(d+1)f$ processes, then we can obtain a correct algorithm to tolerate a single failure among $d+1$ processes. Contradict to the case $f = 1$.

Therefore, $n > max(3f, (d+1)f)$ is necessary for achieving BVC in a synchronous system.$\square$

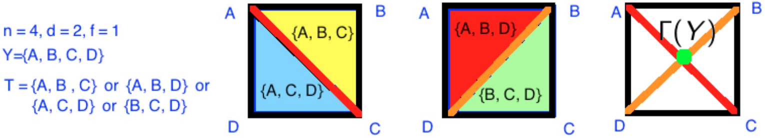
Proof of Sufficient Condition

Theorem 2 *$n > max(3f, (d+1)f)$ is sufficient for achieving BVC in a synchronous system.*

Define:

- Y: a multiset of points. (e.g. $\mathrm{Y} = \{1, 1, 2, 3\}$)
- $\mathcal{H}(T)$: the convex hull of a multiset T.
- $\Gamma(Y) = \cap_{T \subseteq Y, |T| = |Y| - f} \mathcal{H}(T)$: intersection of convex hulls of all subsets of Y of size $|Y| - f$.

.



Proof (Sufficient)

Define:

- Y: a multiset of points.
- $\mathcal{H}(T)$: the convex hull of a multiset T.
- $\Gamma(Y) = \cap_{T \subseteq Y, |T|=|Y|-f} \mathcal{H}(T)$: intersection of convex hulls of all subsets of Y of size $|Y| - f$.

Algorithm: ($n \geq max(3f+1, (d+1)f+1)$)

1. Each process use the Byzantine agreement algorithm to decide the d elements one by one of all the n processes. Non-faulty processes can agree on the d elements of the input vector at each of the n processes, and thus collect such n vectors as S.
2. Each process chooses as its decision vector a point in $\Gamma(S)$ using a determinstic algorithm.

Proof of Termination

- *Termination:* The Byzantine agreement algorithm terminates in finite time.
-
-

Proof of Agreement

- *Termination:* The Byzantine agreement algorithm terminates in finite time.
- *Agreement:* Agreement condition holds because all the non-faulty processes have identical multiset S, thus we can use a deterministic algorithm to pick the decision vector.

Proof of the correctness of the algorithm

- *Termination:* The Byzantine agreement algorithm terminates in finite time.
- *Agreement:* Agreement condition holds because all the non-faulty processes have identical multiset S, thus we can use a deterministic algorithm to pick the decision vector.
- *Validity:*

Proof of Validity

Theorem 3 (Tverberg's Theorem) *For an integer $f \geq 1$, and for every multiset Y containing at least $(d+1)f+1$ points in R^d, there exists a partition $Y_1, ..., Y_{f+1}$ of Y into $f+1$ non-empty multisets such that $\cap_{l=1}^{f+1} \mathcal{H}(Y_l) \neq \emptyset$*

Proof of Validity

Theorem 3 (Tverberg's Theorem) *For an integer $f \geq 1$, and for every multiset Y containing at least $(d+1)f+1$ points in R^d, there exists a partition $Y_1, ..., Y_{f+1}$ of Y into $f+1$ non-empty multisets such that $\cap_{l=1}^{f+1} \mathcal{H}(Y_l) \neq \emptyset$*

1. f: an integer and ≥ 1.
2. Y: a multiset, and $|Y| = (d+1)f+1$.
3. $\cap_{l=1}^{f+1} H(Y_l) \neq \emptyset$, for some partition $Y_1, ..., Y_{f+1}$ of Y. (Note: $\cup_{l=1}^{f+1} Y_l = Y$)

Proof of Validity

Theorem 3 (Tverberg's Theorem) *For an integer $f \geq 1$, and for every multiset Y containing at least $(d+1)f+1$ points in R^d, there exists a partition $Y_1, ..., Y_{f+1}$ of Y into $f+1$ non-empty multisets such that $\cap_{l=1}^{f+1} \mathcal{H}(Y_l) \neq \emptyset$*

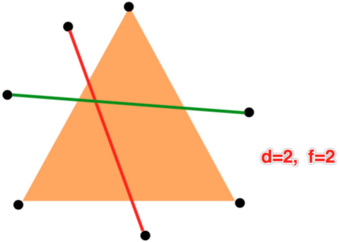


Figure 1: *Illustration of a Tverberg partition.*
Acknowledgment: *The above example is inspired by an illustration authored by David Eppstein, which is available in the public domain from* Wikipedia Commons.

Proof of Validity

Lemma 1 *For any multiset* Y *containing at least* $(d+1)f+1$ *points in* $\mathbb{R}^d$, $\Gamma(Y) \neq \emptyset$.
(Recall that: $\Gamma(Y) = \cap_{T \subseteq Y, |T|=|Y|-f} \mathcal{H}(T)$)

Proof of Validity

Lemma 1 *For any multiset Y containing at least $(d+1)f+1$ points in $\mathbb{R}^d$, $\Gamma(Y) \neq \emptyset$*

(Recall that: $\Gamma(Y) = \cap_{T \subseteq Y, |T|=|Y|-f} \mathcal{H}(T)$)

Proof:

1. By Tverberg's theorem, there exists partition of Y ($|Y| \geq (d+1)f+1$) into non-empty subsets $Y_1, ..., Y_{f+1}$, such that $\cap_{l=1}^{f+1} \mathcal{H}(Y_l) \neq \emptyset$.

Proof of Validity

Lemma 1 *For any multiset Y containing at least $(d+1)f+1$ points in $\mathbb{R}^d$, $\Gamma(Y) \neq \emptyset$*

(Recall that: $\Gamma(Y) = \cap_{T \subseteq Y, |T|=|Y|-f} \mathcal{H}(T)$)

Proof:

1. By Tverberg's theorem, there exists partition of Y ($|Y| \geq (d+1)f+1$) into non-empty subsets $Y_1, ..., Y_{f+1}$, such that $\cap_{l=1}^{f+1} \mathcal{H}(Y_l) \neq \emptyset$.

2. Consider $\Gamma(Y) = \cap_{T \subseteq Y, |T|=|Y|-f} \mathcal{H}(T)$.
We have $|T| = |Y| - f$, and there are $f+1$ subsets in the Tverberg's partition of Y. Therefore, at least one subset Y_i is in T. Hence, $\cap_{l=1}^{f+1} \mathcal{H}(Y_l) \subseteq \Gamma(Y)$, and thus $\Gamma(Y) \neq \emptyset$. $\square$

Proof of Validity

- *Termination:* The Byzantine agreement algorithm terminates in finite time.
- *Agreement:* Agreement condition holds because all the non-faulty processes have identical multiset S, thus we can use a deterministic algorithm to pick the decision vector.
- *Validity:* With at most f faulty process, there at least one multiset T^* must contain the inputs of only non-faulty processes. Thus, $\Gamma(S)$ is in the convex hull of the inputs of non-faulty processes. Hence, validity is satisfied.

Conclusion

For a synchronous system, $n > max(3f, (d+1)f)$ is necessary and sufficient for achieving BVC.

Further Reading

See next slide for the Asynchronous case.

Background

How about in an asynchronous system?

Background

How about in an asynchronous system?

- In an asynchronous system, *exact* consensus is impossible in the presence of faulty processes.
- But we can prove that $n \geq (d+2)f + 1$ is necessary and sufficient to achieve *approximate* Byzantine vector consensus.

Proof (Necessary)

Theorem: $n \geq (d+2)f + 1$ is necessary for approximate BVC in an asynchronous system.

Necessary (Cnt.)

Theorem: $n \geq (d+2)f + 1$ is necessary for approximate BVC in an asynchronous system.

We only need to consider the case when $f = 1$, and for the cases that $f \geq 2$, we can use a simulation similar to the proof of Theorem 1.

Necessary (Cnt.)

Theorem: $n \geq (d+2)f + 1$ is necessary for approximate BVC in an asynchronous system.

Suppose that $f = 1$, and $n = d + 2$. We can see that the following d+1 scenarios cannot be ditinguished by processes $p_1, p_2, ..., p_{d+1}$.

Necessary (Cnt.)

Theorem: $n \geq (d+2)f + 1$ is necessary for approximate BVC in an asynchronous system.

Suppose that $f = 1$, and $n = d + 2$. We can see that the following d+1 scenarios cannot be ditinguished by processes $p_1, p_2, ..., p_{d+1}$.

- Process p_{d+2} has crashed.
- Process p_j $(j \neq i, 1 \leq j \leq d+1)$ is faulty, and process p_{d+2} is slow.

Necessary (Cnt.)

- Process p_{d+2} has crashed.
- Process p_j ($j \neq i, 1 \leq j \leq d+1$) is faulty, and process p_{d+2} is slow.

In order to meet the validity condition, the decided vector of p_i must be in the intersection of convex hull of all non-faulty processes' vectors. For the first case, it should be in the convex hull of X_i^{d+2}. For the other d cases, it should be in the convex hull of X_i^j. Where,

$$X_i^j = \{x_k : k \neq j \textit{ and } 1 \leq k \leq d+1\}$$

.

Besides, we have:

$$\mathcal{H}(X_i^j) \subseteq \mathcal{H}(X_i^{d+2})$$

Necessary (Cnt.)

Besides, we have:

$$\mathcal{H}(X_i^j) \subseteq \mathcal{H}(X_i^{d+2})$$

Therefore, the decision vector of p_i must be in

$$\cap_{j \neq i, 1 \leq j \leq d+1} \mathcal{H}(X_i^j)$$

so as to meet the validity condition.

Necessary (Cnt.)

Besides, we have:

$$\mathcal{H}(X_i^j) \subseteq \mathcal{H}(X_i^{d+2})$$

Therefore, the decision vector of p_i must be in

$$\bigcap_{j \neq i, 1 \leq j \leq d+1} \mathcal{H}(X_i^j)$$

so as to meet the validity condition. Consider the following example:

$$\left.\begin{array}{ll}
x_1 = & [4\epsilon \quad 0 \quad \dots \quad 0] \\
x_2 = & [0 \quad 4\epsilon \quad \dots \quad 0] \\
\vdots & \qquad \vdots \\
x_d = & [0 \quad 0 \quad \dots \quad 4\epsilon] \\
x_{d+1} = & [0 \quad 0 \quad \dots \quad 0] \\
x_{d+2} = & \underbrace{[0 \quad 0 \quad \dots \quad 0]}_{\text{d dimensional}}
\end{array}\right\} \text{d+2 input vectors}$$

Necessary (Cnt.)

Besides, we have:

$$\mathcal{H}(X_i^j) \subseteq \mathcal{H}(X_i^{d+2})$$

Therefore, the decision vector of p_i must be in

$$\cap_{j \neq i, 1 \leq j \leq d+1} \mathcal{H}(X_i^j)$$

so as to meet the validity condition. Consider the following example:

$$\left.\begin{array}{rl}
x_1 = & [4\epsilon \quad 0 \quad \dots \quad 0] \\
x_2 = & [0 \quad 4\epsilon \quad \dots \quad 0] \\
\vdots & \qquad\quad \vdots \\
x_d = & [0 \quad 0 \quad \dots \quad 4\epsilon] \\
x_{d+1} = & [0 \quad 0 \quad \dots \quad 0] \\
x_{d+2} = & \underbrace{[0 \quad 0 \quad \dots \quad 0]}_{\text{d dimensional}}
\end{array}\right\} \text{d+2 input vectors}$$

The decision vector of p_i must be x_i, and thus for each pair of processes in $p_1, \dots, p_{d+1}$ differ by 4ϵ in at least one element!!

Necessary (Cnt.)

So far, we have proved that $n \leq d+2$ is not sufficient, and for the case when $f > 1$, we can use a simulation similar to the proof of Theorem 1, and show that $n \leq (d+2)f$ is also not sufficient. Thus, $n \geq (d+2)f+1$ is necessary for $f \geq 1$.

In the following, we will present a prove that $n \geq (d+2)f+1$ is sufficient by proving the correctness of an algorithm.

Abraham, Amit and Dolev's (AAD) algorithm

Abraham, Amit and Dolev's (AAD) algorithm aims to solve the approximate *scalar* consensus problem in an asynchronous system. It could be viewed as consisting of three components:

- **AAD componend #1**: each process communicate its state vector $v_i[t-1]$ to other processes. AAD guarantees that for each non-faulty process p_i in round t obtains a set $B_i[t]$ containing **at least** $n-f$ tuples of the form (p_j, w_j, t) such that the following properties hold:
  - If p_i, p_j are non-faulty, then $|B_i[t] \cap B_j[t]| \geq n-f$

Abraham, Amit and Dolev's (AAD) algorithm

Abraham, Amit and Dolev's (AAD) algorithm aims to solve the approximate *scalar* consensus problem in an asynchronous system. It could be viewed as consisting of three components:

- **AAD componend #1**: each process communicate its state vector $v_i[t-1]$ to other processes. AAD guarantees that for each non-faulty process p_i in round t obtains a set $B_i[t]$ containing **at least** $n-f$ tuples of the form (p_j, w_j, t) such that the following properties hold:
  - If p_i, p_j are non-faulty, then $|B_i[t] \cap B_j[t]| \geq n-f$
  - If (p_l, w_l, t) and (p_k, w_k, t) are both in $B_i[t]$, then $p_l \neq p_k$.

Abraham, Amit and Dolev's (AAD) algorithm

Abraham, Amit and Dolev's (AAD) algorithm aims to solve the approximate *scalar* consensus problem in an asynchronous system. It could be viewed as consisting of three components:

- **AAD component #1**: each process communicate its state vector $v_i[t-1]$ to other processes. AAD guarantees that for each non-faulty process p_i in round t obtains a set $B_i[t]$ containing **at least** $n-f$ tuples of the form (p_j, w_j, t) such that the following properties hold:
  - If p_i, p_j are non-faulty, then $|B_i[t] \cap B_j[t]| \geq n-f$
  - If (p_l, w_l, t) and (p_k, w_k, t) are both in $B_i[t]$, then $p_l \neq p_k$.
  - If p_k is non-faulty, and $(p_k, w_k, t) \in B_i[t]$, then $w_k = v_k[t-1]$.

Abraham, Amit and Dolev's (AAD) algorithm

- **AAD component #1**: each process communicate its state vector $v_i[t-1]$ to other processes. AAD guarantees that for each non-faulty process p_i in round t obtains a set $B_i[t]$ containing **at least** $n-f$ tuples of the form (p_j, w_j, t).
- **AAD component #2**: Process p_i, having obtained $B_i[t]$, computes its new state $v_i[t]$ as a function of the tuples in $B_i[t]$.

Abraham, Amit and Dolev's (AAD) algorithm

- **AAD component #1**: each process communicate its state vector $v_i[t-1]$ to other processes. AAD guarantees that for each non-faulty process p_i in round t obtains a set $B_i[t]$ containing **at least** $n-f$ tuples of the form (p_j, w_j, t)
- **AAD component #2**: Process p_i, having obtained $B_i[t]$, computes its new state $v_i[t]$ as a function of the tuples in $B_i[t]$.
- **AAD component #3**: AAD also includes a sub-algorithm that allows the non-faulty processes to determine when to terminate their computation.

Asynchronous Approximate BVC algorithm ($n \geq (d+2)f+1$)

1. In the t-th round, each non-faulty process uses the mechanism in *Component #1* of the AAD algorithm to obtain a set $B_i[t]$ containing at least $n-f$ tuples, such that $B_i[t]$ satisfies properties 1, 2, and 3 described earlier for AAD. While these properties were proved in [1] for scalar states, the correctness of the properties also holds when $\mathbf{v}_i$ is a vector.

Asynchronous Approximate BVC algorithm ($n \geq (d+2)f+1$)

1. In the t-th round, each non-faulty process uses the mechanism in *Component #1* of the AAD algorithm to obtain a set $B_i[t]$ containing at least $n-f$ tuples, such that $B_i[t]$ satisfies properties 1, 2, and 3 described earlier for AAD. While these properties were proved in [1] for scalar states, the correctness of the properties also holds when $\mathbf{v}_i$ is a vector.

2. In the t-th round, after obtaining set $B_i[t]$, process p_i computes its new state $\mathbf{v}_i[t]$ as follows. Form a multiset Z_i using the steps below:

   - Initialize Z_i as empty.
   - For each $C \subseteq B_i[t]$ such that $|C| = n - f \geq (d+1)f+1$, add to Z_i one deterministically chosen point from $\Gamma(\Phi(C))$. Since $|\Phi(C)| = |C| \geq (d+1)f+1$, by Lemma 1, $\Gamma(\Phi(C))$ is non-empty.

Asynchronous Approximate BVC algorithm $(n \geq (d+2)f+1)$

1. In the t-th round, each non-faulty process uses the mechanism in *Component #1* of the AAD algorithm to obtain a set $B_i[t]$ containing at least $n-f$ tuples, such that $B_i[t]$ satisfies properties 1, 2, and 3 described earlier for AAD. While these properties were proved in [1] for scalar states, the correctness of the properties also holds when $\mathbf{v}_i$ is a vector.
2. In the t-th round, after obtaining set $B_i[t]$, process p_i computes its new state $\mathbf{v}_i[t]$ as follows. Form a multiset Z_i using the steps below:
   - Initialize Z_i as empty.
   - For each $C \subseteq B_i[t]$ such that $|C| = n-f \geq (d+1)f+1$, add to Z_i one deterministically chosen point from $\Gamma(\Phi(C))$. Since $|\Phi(C)| = |C| \geq (d+1)f+1$, by Lemma 1, $\Gamma(\Phi(C))$ is non-empty.

   Note that $|Z_i| = \binom{|B_i[t]|}{n-f} \leq \binom{n}{n-f}$. Calculate

$$\mathbf{v}_i[t] = \frac{\sum_{\mathbf{z} \in Z_i} \mathbf{z}}{|Z_i|} \tag{9}$$

3. Each non-faulty process terminates after $1 + \lceil \log_{1/(1-\gamma)} \frac{U-\nu}{\epsilon} \rceil$ rounds, where γ $(0 < \gamma < 1)$ is a constant defined later in (11). Recall that ϵ is the parameter of the ϵ-agreement condition.

Note: $\Phi(B) = \{w_k : (p_k, w_k, t) \in B\}$

Proof of the correctness

Without loss of generality, suppose that m processes, $p_1, p_2, ..., p_m$ are non-faulty, where $m \geq n - f$, and the remaining $n - m$ processes are faulty.

Definition 1: A point $\mathbf{r}$ is said to be valid if there exists a representation of $\mathbf{r}$ as a convex combination of $v_k[t-1]$, $1 \leq k \leq m$. That is, there exists constants β_k, such that $0 \leq \beta_k \leq 1$ and $\sum_{1 \leq k \leq m} \beta_k = 1$, and

$$r = \sum_{1 \leq k \leq m} \beta_k v_k[t-1]$$

.

In general, there may exits multiple such convex combination representations of a valid point r. Moreover, it's obvious that at least one of the weights in any such convex combination must be $\geq \frac{1}{m} \geq \frac{1}{n}$.

Proof of the correctness

In the following, we will break the proof into three parts.

Proof of the correctness

In the following, we will break the proof into several parts.

1. For any non-faulty process p_i, consider any $C \subseteq B_i[t]$, such that $|C| = n - f$. Because $n \geq (d+2)f + 1$, and thus $|\Phi(C)| = |C| = n - f \geq (d+1)f + 1$, by Lemma 1, $\Gamma(\Phi(C)) \neq \emptyset$. Therefore, Z_i will contain a point from $\Gamma(\Phi(C))$ for each C.

Proof of the correctness

In the following, we will break the proof into several parts.

1. For any non-faulty process p_i, consider any $C \subseteq B_i[t]$, such that $|C| = n - f$. Because $n \geq (d+2)f + 1$, and thus $|\Phi(C)| = |C| = n - f \geq (d+1)f + 1$, by Lemma 1, $\Gamma(\Phi(C)) \neq \emptyset$. Therefore, Z_i will contain a point from $\Gamma(\Phi(C))$ for each C.
   Because there are at most f faulty processes. Then there exists at least one $(n-2f)$-size subset of $\Phi(C)$ must be a subset of $\{v_1[t-1], v_2[t-1], ..., v_m[t-1]\}$. Therefore, all points in $\Gamma(\Phi(C))$ must be valid, and thus all the points in Z_i computed in *Step* 2 must be valid.

Proof of the correctness

In the following, we will break the proof into several parts.

1. All the points in Z_i computed in *Step* 2 must be valid.
2. Because $|B_i[t] \cap B_j[t] \geq n - f$. Therefore, there exists a set $C_{ij} \subseteq B_i \cap B_j$ such that $|C_{ij}| = n - f$. Therefore, Z_i and Z_j both contain one **identical** point from $\Gamma(\Phi(C_{ij}))$. Suppose the point is z_{ij}, as shown in Part 1, z_{ij} must be valid. Therefore, there must exists a non-faulty process, say $p_{g(i,j,t)}$, such that the weight associated with $v_{g(i,j,t)}[t-1]$ in the convex combination for z_{ij} is $\geq \frac{1}{m} \geq \frac{1}{n}$.

Proof of the correctness

In the following, we will break the proof into several parts.

1. All the points in Z_i computed in *Step* 2 must be valid.
2. Because $|B_i[t] \cap B_j[t] \geq n - f$. Therefore, there exists a set $C_{ij} \subseteq B_i \cap B_j$ such that $|C_{ij}| = n - f$. Therefore, Z_i and Z_j both contain one **identical** point from $\Gamma(\Phi(C_{ij}))$. Suppose the point is z_{ij}, as shown in Part 1, z_{ij} must be valid. Therefore, there must exists a non-faulty process, say $p_{g(i,j,t)}$, such that the weight associated with $v_{g(i,j,t)}[t-1]$ in the convex combination for z_{ij} is $\geq \frac{1}{m} \geq \frac{1}{n}$.
3. Because $v_i[t]$ is computed as the average of the points in Z_i, and $|Z_i| = \binom{|B_i|}{n-f} \leq \binom{n}{n-f}$. Therefore, the weight of $v_{g(i,j,t)}[t-1]$ in $v_i[t]$ is $\geq \frac{1}{n\binom{n}{n-f}}$. Define

$$\gamma = \frac{1}{n\binom{n}{n-f}}$$

Proof of the correctness

Because we have proved that, at time step t, every z_k in Z_i is valid, and z_k can be treated as a convex combination of non-faulty processes' state vectors. Therefore, $v_i[t] = \frac{\sum_{z \in Z_i} z}{|Z_i|}$ can also be represented as a convex combination of non-faulty processes' state vectors (i.e. $\{v_1[t-1], ..., v_m[t-1]\}$).

Proof of the correctness

Because we have proved that, at time step t, every z_k in Z_i is valid, and z_k can be treated as a convex combination of non-faulty processes' state vectors. Therefore, $v_i[t] = \frac{\sum_{z \in Z_i} z}{|Z_i|}$ can also be represented as a convex combination of non-faulty processes' state vectors (i.e. $\{v_1[t-1], ..., v_m[t-1]\}$).

Therefore, we can get that for any non-faulty process's $v_i[t]$ is a convex combination of $\{v_1[0], v_2[0], ..., v_m[0]\}$, implying that the proposed algorithm satisfies the **validity** condition for approximate consensus. (note, $v_k[0]$ is the process p_k's input vector.)

Proof of the correctness

Therefore, we can get that for any non-faulty process's $v_i[t]$ is a convex combination of $\{v_1[0], v_2[0], ..., v_m[0]\}$, implying that the proposed algorithm satisfies the **validity** condition for approximate consensus. (note, $v_k[0]$ is the process p_k's input vector.)

Let $v_{il}[t]$ denote the $l-th$ element of the vector of the state $v_i[t]$ of process p_i. Define $\Omega_l[t] = max_{1 \leq k \leq m} v_{kl}[t]$, the maximum value of l-th element of the vector state of non-faulty processes. Similiarily, $\mu_l[t] = min_{1 \leq k \leq m} v_{kl}[t]$. We have that:

$$\Omega_l[t] - \mu_l[t] \leq (1-\gamma)(\Omega_l[t-1] - \mu_l[t-1])$$

Proof of the correctness

Let $v_{il}[t]$ denote the $l-th$ element of the vector of the state $v_i[t]$ of process p_i. Define $\Omega_l[t] = max_{1 \leq k \leq m} v_{kl}[t]$, the maximum value of l-th element of the vector state of non-faulty processes. Similiarily, $\mu_l[t] = min_{1 \leq k \leq m} v_{kl}[t]$. We have that:

$$\Omega_l[t] - \mu_l[t] \leq (1-\gamma)(\Omega_l[t-1] - \mu_l[t-1])$$

Then we can repeat until $(1-\gamma)^t(\Omega_l[0] - \mu_l[0]) < \epsilon$, thus we can get

$$t > log_{1/(1-\gamma)} \frac{\Omega_l[0] - \mu_l[0]}{\epsilon}$$

.

Proof of the correctness

Let $v_{il}[t]$ denote the $l-th$ element of the vector of the state $v_i[t]$ of process p_i. Define $\Omega_l[t] = max_{1 \le k \le m} v_{kl}[t]$, the maximum value of l-th element of the vector state of non-faulty processes. Similiarily, $\mu_l[t] = min_{1 \le k \le m} v_{kl}[t]$. We have that:

$$\Omega_l[t] - \mu_l[t] \le (1-\gamma)(\Omega_l[t-1] - \mu_l[t-1])$$

Then we can repeat until $(1-\gamma)^t(\Omega_l[0] - \mu_l[0]) < \epsilon$, thus we can get

$$t > log_{1/(1-\gamma)} \frac{\Omega_l[0] - \mu_l[0]}{\epsilon}$$

.

Simply assume that U is the upper bound of the input value, and v is the lower bound of the input value. We can eventually get that for each non-faulty process, it will terminate after $1 + log_{1/(1-\gamma)} \frac{U-v}{\epsilon}$, and $\epsilon-$agreement is ensured.

Proof of the correctness

In conclusion, the proposed algorithm can terminate within finite steps and the validity and the ϵ-agreement are both satisfied. Therefore, $n \geq (d+2)f + 1$ is sufficient for approximate consensus in asynchronous systems.

Conclusion

1. For a synchronous system, $n \geq max(3f+1, (d+1)f+1)$ is necessary and sufficient for achieving Byzantine vector consensus.
2. For an asynchronous system, $n \geq (d+2)f+1$ is necessary and sufficient to achieve *approximate* BVC.

QA

Thanks! QA?