Δ-EIG PROTOCOL FOR BYZANTINE AGREEMENT

Dr. Arpita Patra , Ishita Mandal, Shikha Panwar
Department of Computer Science and Automation
Indian Institute of Science, Bengaluru

ABSTRACT

The standard **EIG (Exponential Information Gathering) protocol** for Byzantine Agreement requires exactly **t+1** rounds for achieving consensus between n processors where n = total number of processors and t = upper bound on the total number of faulty processors. Each processor maintains its copy of EIG tree[1] .

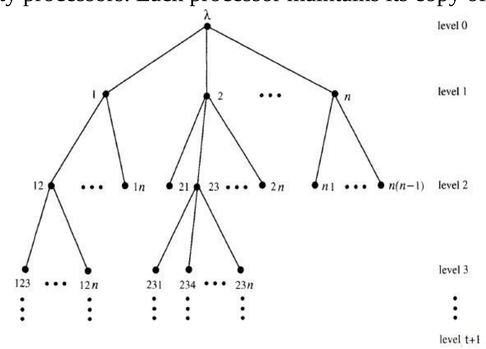

Fig 1 EIG TREE
Source: Lynch, Nancy A. *Distributed algorithms*

By using the **Δ-EIG protocol** the number of rounds required to reach consensus can be reduced further where Δ is the lower bound on the number of initially disabled faulty processors.

INTRODUCTION

•In an instance of Δ-agreement, each nonfaulty processor i starts with an initial value $v_i \in \{0,1\}$ and a set $\mathcal{F}_i$ of processors that i has already detected as faulty. Let Let $\mathcal{D} = \bigcap_i \mathcal{F}_i$ denote the set of initially disabled processors. If all nonfaulty processors starts with initial vote of 0, parameter Δ has no role to play, otherwise, if vote = 1, then atleast Δ faulty processors are initially disabled and $\# D \geq \Delta$. Δ-agreement has two additional parameters, n and t, where n and t have their usual meanings and n and t satisfy $n \geq 3t+1$ while $\Delta < t$. The Decision, Agreement and Validity requirements remain same as in case of standard byzantine agreement i.e.

•**Decision**: Every nonfaulty processor i eventually irreversibly decides on a value {0,1}.

•**Agreement**: The nonfaulty processors all decide on the same value.

•**Validity**: If the initial values v_i of all nonfaulty processors are identical then $d_i = v_i$ for all nonfaulty processors i.

The standard EIG protocol for byzantine agreement is an instance of Δ-agreement where $\Delta = 0$ and $\mathcal{F}_i = \phi$ for every nonfaulty processor i.

THE PROTOCOL

The Δ-EIG protocol[2] for a single instance of Δ-agreement is based on some components like it operates on an EIG tree of depth t + 1 - Δ(contrary to depth t + 1 in standard EIG protocol) and every processor i maintains a set of processors it has detected as faulty. Let $\mathcal{F}i(r)$ denote the set of faulty processors detected by i in first r rounds which grows monotonically over time i.e. $\mathcal{F}_i(r+1) \supseteq \mathcal{F}_i(r)$ and $\mathcal{F}_i(0) = \mathcal{F}_i$. These sets are useful for both masking values in its own tree as well as for reporting on masked nodes.

Sending: In a given round r + 1, a processor i sends a message to all other processors consisting of two components:

1. The message contains reports *mask(i, z)* where $z \in \mathcal{F}_i(r) \setminus \mathcal{F}_i(r-1)$ i.e. i has just discovered z as faulty. For completeness, let $\mathcal{F}_i(-1) = \phi$ so that in first round i reports that it is masking the processors in $\mathcal{F}_i(0) = \mathcal{F}_i$.
2. The message consists of pairs <σj; v> where $v = tree_i(\sigma j)$, ∀ nodes σj of depth r such that $j \notin \mathcal{F}i(r)$.

Recording and Masking: Processor i appends every *mask(i, j)* report it receives in round r to the list of mask reports that it maintains. The values are recorded in $tree_i$ by processor i in following two ways:

1. If $j \notin \mathcal{F}i(r-1)$ then $tree_i(\sigma j)$ is the value reported by j for σ in round r.
2. If $z \in \mathcal{F}i(r-1)$ then $tree_i(\sigma j) = par(|\sigma|)$ i.e. values of nodes corresponding to initially detected failures are always masked.

Fault Detection: $\mathcal{F}_i(r)$ is obtained by adding to $\mathcal{F}_i(r-1)$ new failed processor discovered by applying the fault detection rules given below. Processor i detects z as faulty at the end of round if one of the following conditions holds true :

FD0: z sends ill-formated message in round r.

FD1:By end of round r, processor i received $\geq t + 1$ *mask(j, z)* from distinct processors j.

FD2:By end of round r, some node σz that was not closed in $tree_i$ by end of round |σz| is committed to both 0 and 1 in $tree_i$.

FD3:By end of round r, some node σaz and value v such that (a) r = |σaz|+1, and (b) σaz is not closed in $tree_i$ by end of round r; we have that

(i) σaz is committed to v in $tree_i$.

(ii) $\geq 2(t + 1 - |\sigma a|) + 1$ of the nodes σaz are committed to 1 – v in $tree_i$ by end of round r; and

(iii) z does not mask a in round |σaz| +1.

The failures discovered in round r will affect processor's messages and processing from round r + 1 onwards.

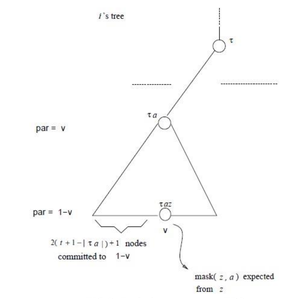

FIG 2: Fault detection using FD3
Source: Garay, Juan A., and Yoram Moses. "Fully polynomial byzantine agreement for processors in rounds." *SIAM Journal on Computing* 27.1 (1998): 247-290.

Fixing nodes: A node σ is fixed in $tree_i$ to value v at end of round r if σ was not closed at end of round r – 1 and one of following rules holds true:
Fx1: $r = |\sigma| = t + 1 - \Delta$ and $tree_i(\sigma) = v$
Fx2: $r = |\sigma| + 1$, $par(|\sigma|) = v$ and

$$tree_i(\sigma j) = v \text{ for atleast } \begin{cases} n - t \text{ nodes } \sigma j & \text{if } \sigma = \lambda; \\ n - t - 1 \text{ nodes } \sigma j & \text{if } |\sigma| = 1 \text{ and;} \\ n - t - 2 \text{ nodes } \sigma j & \text{if } |\sigma| \geq 1 \end{cases}$$

Fx3:Rules Fx1 and Fx2 do not apply and either
(a)$par(|\sigma|) = v$ and $\geq F(|\sigma|)$ of σj's are fixed to v; or
(b)$par(|\sigma|) = 1 - v$ and $\geq n - |\sigma| - F(|\sigma|) + 1$ of σj's are fixed to v.
Here, F is an admissible function.

RESULTS

This protocol in addition to saving in communication also allows a processor to detect failures based on reports received by it and also estimate the number of disabled processors.

Crucial property of this protocol is that two rounds after a node σ is closed in one nonfaulty processor's tree, it will be closed in all processors' tree and once σ is closed, processor i has no use of its descendants.

$tree_i(\sigma) = tree_j(\sigma)$ holds for every correct node σ of depth atmost t + 1 - Δ and nonfaulty processors i and j.

FUTURE WORKS

Since, i needs to relay vaules in the subtree rooted at σ for atmost 2 rounds after σ is closed in $tree_i$ thus, Δ-EIG protocol can be further modified to obtain an early stopping protocol called Δ-Es protocol. This protocol would lead to further decrease in number of nodes in $tree_i$.

Applying Δ-Es protocol, sliding-flip protocol together with monitor voting[3] would lead to fully polynomial byzantine agreement for n > 3t in t + 1 rounds.

CONTACT

Ishita Mandal
Computer Science & Technology
Indian Institute of Engineering Science &Technology, Shibpur
E-mail: imandal.dd2014@cs.iiests.ac.in
Phone: 9836303955

REFERENCES

[1] Lynch, Nancy A. *Distributed algorithms*. Morgan Kaufmann, 1996.
[2] Garay, Juan A., and Yoram Moses. "Fully polynomial byzantine agreement for processors in rounds." *SIAM Journal on Computing* 27.1 (1998): 247-290.
[3] Berman, Piotr, Juan A. Garay, and Kenneth J. Perry. "Towards optimal distributed consensus." *Foundations of Computer Science, 1989., 30th Annual Symposium on*. IEEE, 1989.