UNDERSTANDING A BYZANTINE ALGORITHM

J. Misra

Department of Computer Sciences
University of Texas at Austin
Austin, Texas 78712

TR-85-20 September 1985

This work was supported in part by a grant from the Office of Naval Research under grant ONR N00014-85-K-0057.

1. Introduction

The problem of Byzantine Agreement defined in [1] is as follows. There are N processes any pair of which may communicate by messages. Any message sent is received instantly and correctly by the recipient. It is given that exactly t of the processes are faulty and the rest, N-t, are reliable. Each process is initially *off* or *on*. The problem is to devise a scheme whereby all reliable processes agree eventually on a common value, 0 or 1. Furthermore, the common value is 0(1) if all reliable processes are initially *off(on)*. Difficulty arises due to the nature of faulty processes: they may provide conflicting information in a concerted manner to thwart agreement by reliable processes.

We discuss an ingenious algorithm for this problem appearing in literature [2]. This note is intended as a different, and hopefully simpler, exposition of this algorithm and its proof. We believe that simplification is achieved by removing explicit message communication from the algorithm description. It should be easy to see how our scheme may be implemented using synchronous message communications. Our proof uses the major ideas from [2] though the restructuring results in some simplification.

It is known that solutions for this problem exist only if $N \geq 3t+1$. We assume that $N = 3t+1$ and $t > 0$. Let $low = t+1$ and $high = 2t+1$; therefore *high* is the number of reliable processes. Observe that every subset of *low* processes has at least one reliable process and every subset of *high* processes has at least *low* reliable processes.

2. Algorithm

We represent states of processes and their communication histories by a colored directed graph. Every vertex corresponds to a distinct process and a vertex state is *off/on* denoting the current state of the process. An edge (i,j) is directed from process i to j, and has a color, *black* or *white*; we allow edges of the form (i,i).

Initially, there are no edges in the graph and a vertex state is the corresponding process state. The algorithm proceeds in rounds where during the first part of a round processes note the states of all other processes and edges that are present in the graph. Upon completion of these observations, processes recompute their states and may add/color their own outgoing edges. Note that states and outgoing edges of faulty processes may not be observed consistently by different reliable processes; this is the Byzantine aspect of the problem. However, we have:

Axiom:: states and outgoing edges of reliable processes are observed exactly by all processes.

We assume that some unspecified mechanism coordinates the observations and

computations such that all observations precede all computations in a round. Processes cannot observe changes in the graph or process states during the computation phase of a round. In particular, a process observes changes in its own state in the round following the change.

Reliable processes use the following rules to add outgoing edges, color their outgoing edges and change their own states. In the following, p,q are reliable processes and j is an arbitrary process. p,j is the edge from p to j, if it exists. Let $in(j)$ denote the number of incoming edges to j, as observed by p during the round; $white{-}out(p)$ is the number of *white* outgoing edges of p.

We use the following actions with the given meanings in the algorithm.

- **add black edge (p,j):** this has no effect if edge (p,j) {of either color} exists; otherwise a black edge (p,j) is created.
- **Color (p,j) white:** a white edge (p,j) exists, following this operation.
- p **becomes on:** p is on, following this operation.

Black Edge Rule::
p observed j is *on* **or** *high* $> in(j) \geq$ *low* $\rightarrow$ add *black* edge (p,j)

White Edge Rule::
$in(j) \geq$ *high* $\rightarrow$ color (p,j) *white*

State Change Rule::
{Let r be the round number}
white-out(p) $\geq t + r/2 \rightarrow$ p becomes *on*

Observation

1. No reliable process becomes *off* once it is *on*.
2. No edge is ever deleted by a reliable process. No *white* outgoing edge of a reliable process ever becomes *black*.
3. A reliable process creates a *black* edge (p,j) only if j is observed *on* by some reliable process, possibly p. edge (p,j) is *white* only if there are at least *low* reliable processes with edges to j and hence these edges are observed in all subsequent rounds by all reliable processes.

3. Proofs

We have not yet specified the conditions under which processes commit to different values. These conditions become apparent from the results proven below. In the following p,q denote reliable processes and j an arbitrary process. We use "at round r" to mean upon completions of computations of round r and "in round r" to mean prior to computations of that round. "At round 0" will refer to initial conditions.

Lemma 1:

Edge (p,q) is *white* at round $(r+2)$ **iff** q is *on* at round r.

Proof:

If q is *on* at round r, it is observed *on* in round $(r+1)$ by all reliable processes and hence $in(q) \geq high$ at $(r+1)$. Then every reliable process, including p, has a *white* edge to q at round $(r+2)$. Conversely, if q is *off* at round r, it is *off* at all previous rounds and it is observed *off* in round $(r+1)$. Hence $in(q) < low$ at $(r+1)$ and therefore no reliable process has a *white* edge to q at $(r+2)$. □

We show in the following lemma that if any reliable process changes state then every reliable process is *on* two rounds later; furthermore a state change is possible only if at least $r/2$ reliable processes are *on* two rounds earlier. Let $np(r)$ denote the number of reliable processes which are *on* at round r. We note that $np(r)$ is monotone non-decreasing in r.

Lemma 2:

For all $r \geq 2$:

$[np(0) = np(r)]$ **or** $[np(r+2) = high$ **and** $np(r-2) \geq (r-1)/2]$

Proof:

Consider the smallest r, if any, for which $np(0) \neq np(r)$. If no such r exists, the lemma holds. Otherwise, some reliable process p applies the state change rule at round r. For p, $white\text{–}out(p) \geq t + r/2$ in round r; hence p has *white* edges to at least $r/2$ reliable processes and, from lemma 1, all these are *on* at round $(r-2)$, i.e. $np(r-2) \geq r/2$. Also $np(r) > np(r-2)$ and hence $np(r) \geq r/2 + 1$.

Next we show that every reliable process is *on* at round $(r+2)$. We only need to prove this for reliable processes which are *off* at round r; let q be one such process. We first show that if (p,j) is *white* at round r then (q,j) is *white* at round $(r+2)$: for (p,j) to be *white*, $in(j) \geq high$ in some round before or in round r, as observed by p; hence at least *low* reliable processes have edges to j in round r; then every reliable process has at least a *black* edge to j at $(r+1)$ and *white* edge at $(r+2)$. Also, edge (q,p) is *white*

at round $(r+2)$, from lemma 1. Also, from lemma 1, p has no *white* edge to q at round r. Therefore, *white-out(q)* at round $(r+2) \geq 1 +$ *white-out(p)* at round $r \geq t + (r+2)/2$; hence q is *on* at $(r+2)$.

Consider any round $r', r' \geq 2$. For $r' < r$, $np(r') = np(0)$ and hence the lemma holds.

For $r' \geq r$, $np(r'+2) = high$.
For $r' = r+1$, $np(r'-2) \geq np(r-2) \geq r/2 = (r'-1)/2$.
For $r' = r+2$, **or** $r' = r+3$, $np(r'-2) \geq np(r) \geq (r+2)/2 \geq (r'-1)/2$.
For all larger values r', $np(r'-2) = high \geq (r'-1)/2$.

Theorem:

Let $R = 2t+2$.

1. $np(0) = 0 \Rightarrow np(R) = 0$

2. $np(0) = high \Rightarrow np(R) = high$

3. $np(R) < low$ **or** $np(R) = high$

Proof:

1. Suppose $np(0) = 0$. Observe that $np(1) = 0$. We prove that $np(r) = 0$, $r \geq 2$, by induction on r. From inductive hypothesis, for $r \geq 2$, $np(r-2) = 0 < (r-1)/2$. Hence from lemma 2, $np(r) = np(0) = 0$.

2. Follows from the monotonicity of np.

3. Let $np(0) \geq low$. From lemma 1, every reliable process p is *on* at round 2 because *white-out(p)* $\geq low = t + r/2$, at $r = 2$. Hence $np(2) = high$. Therefore, assume that $np(0) < low$. If $np(R) = np(0)$, then the result follows. Hence, consider $np(0) \neq np(R)$. From lemma 2, $np(R-2) \geq (R-1)/2 \geq (2t+1)/2$. Hence $np(R-2) \geq low$. Therefore, $np(0) \neq np(R-2)$. From lemma 2, $np(R) = high$.

Commit Rule:: At round $R+2$,

white–out(p) $\geq$ *high* $\rightarrow$ commit to 1
white–out(p) $<$ *high* $\rightarrow$ commit to 0

Theorem:

All reliable processes commit to the same value. If they are initially *off/on* they commit to 0(1).

Proof:

From the theorem, $np(R) < low$ or $np(R) = high$. If $np(R) < low$ then, from lemma 1, for any reliable process p, $white\text{–}out(p) < high$ at round R. If $np(R) = high$ then, again from lemma 1, $white\text{–}out(p) \geq high$. Hence all reliable processes commit to the same value. Other parts follow trivially from the theorem.

Observation

A decision to commit can be made at round $R+1$. If $out(p)$ denotes the number of outgoing edges of p at round $R+1$ then, $out(p) \geq high \equiv np(R) = high$. Hence p commits to 1 if $out(p) \geq high$ and to 0 otherwise.

4. Discussion

Observations in a round can be implemented by message communications: p observes edge (i,j) in a round only if i sends a message to p informing p of the existence of the edge. Note that colors of edges need not be communicated. Furthermore, since edges are never deleted, only information about newly created edges need be sent.

5. Acknowledgement

I am grateful to Mike Fischer for pointing out the last observation and to Paul Attie for pointing out an error in an earlier manuscript.

6. References

1. Lamport, L., Shostak, R. and Pease, M. "Byzantine Generals Problem", *ACM Transactions on Programming Languages and Systems,* Vol. 4, No. 3, pp.382-401, July 1982.

2. Lynch, N., Fischer, J. and Fowler, R. "A Simple and Efficient Byzantine Generals Algorithm", *Proceedings of the 2nd Symposium on Reliability in Distributed Software and Database Systems,* July 1982.