| CMSC 858K — Advanced Topics in Cryptography | | May 6, 2004 |
|---|---|---|
| | Lecture 27 | |
| *Lecturer: Chiu Yuen Koo* | *Scribe(s):* *Omer Horvitz, John Trafton, Zhongchao Yu, Kavitha Swaminathan* | |

# 1 Introduction

In a previous lecture, we defined the Byzantine agreement/broadcast problems and showed that there is no protocol solving these problems when the fraction of corrupted players is 1/3 or larger. Today, we prove the converse by showing a protocol for broadcast (and hence Byzantine agreement; cf. the previous lecture) when the fraction of corrupted players is less than 1/3. The protocol we will show is called the *Exponential Information Gathering (EIG)* protocol, and was essentially the first known protocol for this task (see [2, 4, 1, 3]). We stress at the outset that the protocol is not very efficient — as the name suggests, its complexity is exponential in the number of players — but it forms an important feasibility result. Since the protocol was introduced, much work has focused on improving various parameters of the protocol, and fully polynomial Byzantine agreement/broadcast protocols with optimal resilience are now known.

# 2 The EIG Protocol

Let $n$ be the number of players, and $t$ be the upper-bound on the number of malicious players (for simplicity, let the number of faults be exactly $t$). We show how to achieve broadcast when $t < n/3$. To gain intuition, we describe how the protocol works for the case of four players, when only one is assumed to be faulty. Let the players be denoted $A, B, C, D$, and assume the sender $A$ holds a value $v$. $A$ begins by sending $v$ to $B, C, D$, who proceed to send the value they received from $A$ to each other. At this point, each player has a value that it received from $A$ and values that the other two players report to have received from $A$. Each player maintains this information in a tree (cf. Fig. 1). Each player $i \in \{B, C, D\}$ stores the value it received from $A$ in the root of its tree, and the value that node $j \in \{B, C, D\}$ (including itself) says it received from $A$ in node $Aj$. Each player $i \in \{B, C, D\}$ decides on the majority value of the latter three values (i.e., the values stored at the leaves of the tree). If no majority value exists, the players decide on some default value $v_0$.

We proceed with the analysis: if $A$ is non-faulty then one of $B, C, D$ *is* faulty; without loss of generality, assume it is $B$. Looking at the trees maintained by $C$ and $D$, we see that in each of their trees the value stored at leaves $AC$ and $AD$ is exactly $v$, the initial input of the sender $A$. Thus, regardless of what $B$ sends to $C$ and $D$ (i.e., regardless of what appears at leaf $AB$ in $C$'s tree and lead $AB$ in $D$'s tree), $C$ and $D$ will decide on $v$, ad required.

On the other hand, if $A$ is faulty then $B, C$, and $D$ are not faulty. Let $v_b, v_c, v_d$ denote the values that $A$ sent to $B, C, D$, respectively. Looking at the trees maintained by $B, C$,

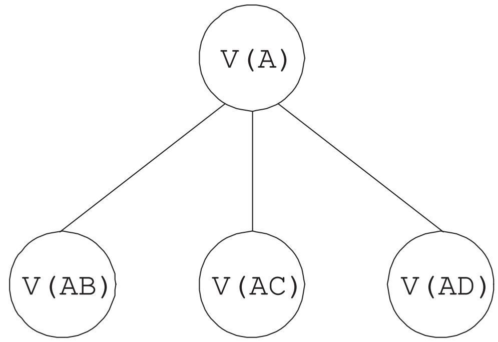


Figure 1: Information Gathering Tree for $n = 4, k = 1$.

and $D$, we see that each will have the value $v_b$ stored at leaf $AB$, the value $v_c$ stored at leaf $AC$, and the value $v_d$ stored at leaf $AD$. Thus, they will all decide on some common value (whatever that value may be) as required.

More generally, assume the network consists of $n$ players $A, B, C, \ldots$, of which $t < n/3$ are faulty (equivalently, $n > 3t$). The EIG algorithm, described below, involves the maintenance of a tree of height $t$ (i.e., having $t + 1$ levels) by each player. The root of the tree is labeled with the sender $A$. Every internal node labeled $\ell$ (where $\ell$ is a string) has one child for each player $s$ that does not appear in $\ell$; the label of this child is $\ell s$, the concatenation of the label of the parent with the name of the aforementioned player. The node labeled $\ell s$ is said to *correspond* to player $s$. For example, if we have players $A, B, C, D, E, F, G$, then the root has children $AB, AC, AD, AE, AF, AG$; the node labeled $AE$ has children $AEB, AEC, AED, AEF, AEG$; and nodes $AEF, AEG$ are said to correspond to $F, G$, respectively.

The protocol proceeds in $t + 1$ rounds. In the first round, $A$ sends its input $v$ to all other players. Each player stores the value it received from $A$ in the root of its tree. In each subsequent round, each player (except $A$, who no longer needs to take part in the protocol) broadcasts the most-recently-filled level of its tree. Upon receiving these messages, each player $P$ fills the next level of its tree by storing at node $\ell X$ the value that player $X$ claims to have stored at node $\ell$ in its own tree.[1] Intuitively, player $P$ stores in node $A \cdots YX$ the value that "$X$ says that $Y$ says ... that the sender $A$ said". We refer to the value stored at node $\ell$ in $P$'s tree as $v_P(\ell)$. A generic information-gathering tree is depicted in Fig 2 (the identity of the particular player maintaining the tree is omitted).

After completion of the $t + 1$ rounds, we define a *reduced value* for each node. For a player $P$ and a node labeled $\sigma$, the reduced value $v'_P(\sigma)$ is defined as follows:

- If $\sigma$ is a leaf, then $v'_P(\sigma) = v_P(\sigma)$.
- If $\sigma$ is not a leaf, then $v'_P(\sigma)$ is the majority value of the *reduced values* of its children (in $P$'s tree). If no majority exists, $v'_P(\sigma)$ is assigned the default value $v_0$.

Each player computes and decides on the reduced value of the root of its tree.

[1] *Note*: this includes messages that player $P$ sends "to itself". Of course, no such message is actually sent but $P$ can certainly simulate the sending of such a message.

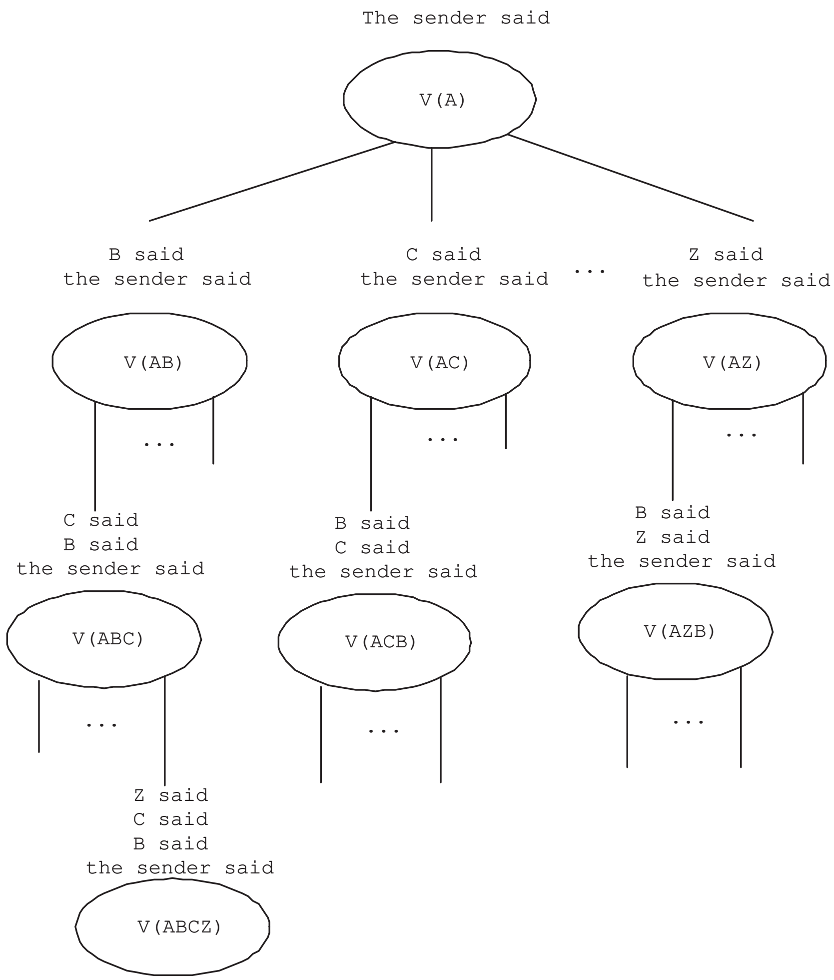


Figure 2: A Generic Information Gathering Tree.

We now prove the correctness of this protocol. Each (honest) player maintains a tree with $t+1$ levels (numbered 1 through $t+1$), and each node at level $k$ has $n-k$ children. It follows that in the tree maintained by each player, every internal node has at least $n-t \geq 2t+1$ children (of which at most $t$ are faulty). The correctness of the protocol follows from the following lemmas:

**Lemma 1** *If a node $\sigma$ corresponds to a non-faulty player, then $v_P(\sigma) = v_Q(\sigma)$ for all non-faulty players $P, Q$.*

**Proof** Let $\sigma = \sigma' R$, where $R$ is a non-faulty player. The lemma follows easily from the fact that $R$ sends the same value $v_R(\sigma')$ to all other players. (Note that the lemma holds even when $R \in \{P, Q\}$.) ■

**Lemma 2** *If a node $\sigma$ corresponds to a non-faulty player, then there is a value $v$ such that $v'_P(\sigma) = v_P(\sigma) = v$ for any non-faulty player $P$.*

**Proof** By backward induction on the level of $\sigma$ in the information-gathering tree.

**Base case:** Here, $\sigma$ is a leaf. Then $v'_P(\sigma) = v_P(\sigma)$ for any honest $P$ by the definition of reduced values. Moreover, Lemma 1 shows that for any non-faulty player $Q$ we have $v_P(\sigma) = v_Q(\sigma)$, and hence $v'_P(\sigma) = v'_Q(\sigma)$ as well.

**Inductive step:** Assume the claim holds for all nodes at level $k+1$, and consider a node $\sigma$ at level $k$. Lemma 1 shows that all non-faulty processes $P$ have the same value $v_P(\sigma)$. Call this value $v$. Each non-faulty process will send $v$ to all other processes in round $k+1$, so $v_Q(\sigma P) = v$ for all non-faulty $P, Q$. The inductive hypothesis now implies that $v'_Q(\sigma P) = v_Q(\sigma P) = v$ for all non-faulty $P, Q$. Since a majority of the children of $\sigma$ are non-faulty (by the argument above), this implies that $v'_Q(\sigma) = v = v_Q(\sigma)$ for all non-faulty $Q$.

■

If $A$ is honest, the above lemma immediately implies that all non-faulty players decide on the sender's initial value. All that is left to be shown is that all non-faulty players reach agreement when the sender is faulty. We prove a slightly stronger lemma:

**Lemma 3** *If all paths from a node $\sigma$ to a leaf contain at least one node corresponding to a non-faulty player, then $v'_P(\sigma) = v'_Q(\sigma)$ for all non-faulty players $P, Q$.*

**Proof** By backward induction on the level of $\sigma$ in the information-gathering tree.

**Base case:** $\sigma$ is a leaf. The claim reduces to Lemma 2.

**Inductive step:** Assume the claim holds for all nodes at level $i+1$, and consider a node $\sigma$ at level $i$.

- If $\sigma$ corresponds to a non-faulty node, the claim reduces to Lemma 2.
- If $\sigma$ corresponds to a faulty node, consider a child $\sigma'$ of $\sigma$. It must be the case that all paths from $\sigma'$ to a leaf contain at least one node corresponding to a non-faulty player. By the induction hypothesis, $v'_P(\sigma') = v'_Q(\sigma')$ for all non-faulty players $P, Q$. By the definition of the reduced value, it follows that $v'_P(\sigma) = v'_Q(\sigma)$ too.

■

Notice that every path from the root of the information-gathering tree to a leaf contains a node corresponding to a non-faulty player, because the length of any such path is $t+1$ and there are at most $t$ faulty players. Applying Lemma 3 to the root completes our proof.

We sum up the result with the following theorem:

**Theorem 4 ($n > 3t$ is sufficient for Byzantine agreement)** *There exists a protocol that achieves Byzantine agreement when $n > 3t$.*

## References


[1] A. Bar-Noy, D. Dolev, C. Dwork and R. Strong. Shifting Gears: Changing Algorithms on the Fly to Expedite Byzantine Agreement. In *Proc. 6th Annual ACM Symposium on Principles of Distributed Computing (PODC)*, pp. 42–51, 1987.

[2] L. Lamport, R. Shostak, and M. Pease. The Byzantine Generals Problem. *ACM Trans. Program. Lang. Syst.* 4(3): 382–401 (1982).

[3] Nancy Lynch. Distributed Algorithms. Morgan Kaufmann, 1996.

[4] M. Pease, R. Shostak, and L. Lamport. Reaching Agreement in the Presence of Faults. *J. ACM* 27(2): 228–234 (1980).