Fast Quantum Byzantine Agreement[*]

[Extended Abstract]

Michael Ben-Or[†]
benor@cs.huji.ac.il

Avinatan Hassidim
avinatan_h@hotmail.com

School of Computer Science and Engineering
The Hebrew University
Jerusalem 91904, Israel

ABSTRACT

We present a fast quantum Byzantine Agreement protocol that can reach agreement in $O(1)$ expected communication rounds against a strong full information, dynamic adversary, tolerating up to the optimal $t < n/3$ faulty players in the synchronous setting, and up to $t < n/4$ faulty players for asynchronous systems. This should be contrasted with the known classical synchronous lower bound of $\Omega(\sqrt{n/\log n})$ [3] when $t = \Omega(n)$.

Categories and Subject Descriptors

F.0 [**Theory of Computation**]: General

General Terms

Theory

Keywords

Byzantine Agreement, Quantum Computation

1. INTRODUCTION

Reaching agreement in the presence of faults is a fundamental problem in distributed computation. Here a group of n players must agree on a bit despite the faulty behavior of some of the players. Ruling out trivial solution (such as always deciding on the same bit), this problem, first suggested by Pease, Shostak and Lamport [15], as the Byzantine Agreement Problem, has been studied in synchronous/asynchronous and deterministic/randomized computation models and under various fault models or Adversaries: Fail-Stop and Byzantine, Static and Adaptive, Computationally Bounded and Unbounded Adversaries - just to name a few.

Studying this problem has revealed deep and sometimes surprising differences between various distributed models of computation. This problem shows the power of randomization in distributed computation allowing faster (expected time) protocols in synchronous systems, and for asynchronous systems, randomization allows solutions were no deterministic protocol, tolerating even one fail-stop fault, can exist. In this context it is natural to study what quantum protocols can do.

Previous results

It is well known that when up to t players may fail no deterministic algorithm can solve the synchronous problem in less than $t + 1$ rounds [14] even for the Fail-Stop model, and Garay and Moses showed in [13] a fully polynomial deterministic agreement protocol which works in $t + 1$ rounds even for the Byzantine case, tolerating the optimal $t < n/3$ faults.

Randomized agreement protocols which manage to reach agreement in a constant expected number of rounds are known both for the Fail-Stop (for example the protocol given by Chor, Merrit and Shmoys in [9] for non adaptive adversaries) and the Byzantine model (see [16, 12] in the private channels model). These protocols all assume some bounds on the possible behavior of the adversary.

The problem of a Byzantine, computationally unbounded, adaptive adversary was studied by Feldman and Micali [12] which found an ingenious expected constant round protocol for this problem. However, their protocol relies heavily on the privacy of the good players, i. e. on the limited knowledge of the adversary. For a Byzantine, computationally unbounded, adaptive adversary that has full information, an $O(1)$ randomized protocol was shown by Ben-Or only for $t = O(\sqrt{n})$[4]. We note that all randomized agreement protocol rely, either explicitly or implicitly, on the ability of the players to generate in constant numbers of rounds a "weak global coin". To achieve a constant expected time agreement protocol the weak global coin prodedure can be quite biased or even undefined provided that both outcomes of 0 and 1 have probability of at least some constant $c > 0$ [16, 11, 12].

A later result by Bar-Joseph and Ben-Or shows that some bounds on the adversary are indeed necessary to achieve constant expected time randomized agreement. Studying

[*] Research suppoprted by an Israel Defence Ministry research grant and by the Israel Science Foundation.

[†] Incumbent of the Jean and Helena Alfassa Chair in Computer Science.

STOC'05, May 22-24, 2005, Baltimore, Maryland, USA.
Copyright 2005 ACM 1-58113-960-8/05/0005 ...$5.00.

Table 1: Full information adversary, synchronous agreement time bounds for $t = \theta(n)$

Fault Model	Deterministic	Randomized/Static Adv.	Randomized/Adaptive Adv.	New Quantum
Fail-Stop	$t+1$	$O(1)$	$\theta(\sqrt{n/\log n})$	$O(1)$
Byzantine	$t+1$	$O(\log n)^a$	$\Omega(\sqrt{n/\log n})^b$	$O(1)$

a. See [5]. The best published protocol [8] gives a bound of $O(n/\log n)$.
b. The best upper bound is the deterministic $t+1$ round protocol.

the case of a Fail-Stop computationally unbounded, full information, adaptive adversary they prove a lower bound of $\Omega(t/\sqrt{n \log(2 + t/\sqrt{n})})$ on the expected number of rounds, and present a protocol which has this complexity [3].

We note that the well known bound of $t < n/3$ Byzantine faults holds also for quantum protocols[1].

Our results

This paper presents two results. The first is a quantum protocol for synchronous consensus in the presence of an adaptive, full information and computationally unbounded, fail-stop adversary, which acts in an expected constant number of rounds for any number of bad players $t < \frac{n}{3}$. The second, and the main result of this paper, is a protocol for synchronous Byzantine agreement, in the presence of an adaptive, full information and computationally unbounded adversary which acts in an expected constant number of rounds tolerating the optimal $t < \frac{n}{3}$ faulty players. This second result is clearly a stronger than the first, but the solution of the first is presented because it is much simpler and easier to understand. We caution the reader not to take our quantum fail-stop adversary model too seriously as its main purpose is just to provide an easy yet interesting and nontrivial example to the advantage of quantum protocols on just classical randomization in distributed computation.

Both protocols use the same basic idea of postponing coin flips, using instead quantum superpositions, until after the adversary has chosen her actions in a certain round. This idea will be expanded a little in section 3. In section 4 we will present the protocol for the fail-stop adversary, and the Byzantine adversary will follow in section 5. In this extended abstract we briefly explain how to extend our results to a constant expected number of communication round *asynchronous* quantum Byzantine agreement protocol tolerating $t < n/4$ faults.

2. PRELIMINARIES

The basic model we deal with is a group of n processes, where each pair of processes are connected via a separate, two way, quantum communication channel. We will sometimes pass classical information through these channels, so we may also add pairwise classic channels (but this is not necessary of course). In this extended abstract we discuss in detail only the Synchronous communication model in which communication between the players happens in well defined communication rounds[2]. Formally, each round will consist of two phases:

Phase A - The communication phase:

In this phase all processes send the messages they wish to send in this round. They then receive all messages sent to them in this round.

Phase B - The computation phase:

In this phase all the processes process the messages they received, and choose what to send in the next round. We assume that the computation phase begins only after the communication phase has ended.

We model the faulty behavior of the system by an Adversary. We will consider Byzantine and Fail-Stop adversaries, but both will always be adaptive, computationally unbounded, and will have full information on all the local variables of all processes. Using the "quantum" notation, a full information adversary knows at each point the exact pure state of the system. In many ways these are the ultimate adversaries, which can coordinate their actions in the most effective way. We will only limit the number of processes the adversary can control (in the Byzantine model) or stop (in the Fail-Stop model). Other than the adversary, we will assume that the communication network is absolutely reliable (e.g. any link failure or corruption is attributed as a fault of the sender).

We will also allow the adversary to synchronize each phase of a round in any way she pleases. For example, in the Byzantine case, she may choose to let a process P receive some of its messages, then corrupt P and measure P's state and based on this information decide whether to corrupt some other process Q who sent P a message, and change some of Q's messages.

This paper deals with the Consensus (or Byzantine Agreement) problem, in which n different processes $P_1, \ldots, P_n$ must decide upon the value of a bit. We will assume that each process is given an input bit $x_i \in \{0, 1\}$ and all processes must agree on a common output bit despite the intervention of the adversary. The protocol will be over after all processes chose their output. Once a process has chosen the output bit it can no longer change it. The protocol must obey three conditions:

- **Agreement:** All non faulty processes (the meaning of this depends on the adversary) choose the same value with probability 1.
- **Validity:** If the input of all processes is v, then v will be chosen.
- **Termination:** All non faulty processes decide upon a value with probability 1.

We measure the complexity of the algorithm by the maximal expected number of rounds it takes to reach agreement. The maximum is taken over all adversaries, and all initial inputs.

The Fail-Stop Adversary We will now describe the capabilities of the Fail-Stop adversary, of section 4. We assume

[1] Byzantine Agreement should not be confused with "weak agreement", sometimes called "detectable broadcast" where a single faulty player may force the system to abort even if all good players received the same input.

[2] Extending our results to the asynchronous setting is straightforward. See section 6

that during each round of the protocol, and based on the the known pure state of the system the adversary has the ability to stop any number of processes. The processes killed by the adversary will no longer take part in the protocol. The adversary will choose which subset of the messages sent by them in their dying round will reach their destination. A stopped process will be also called "faulty" or "dead" and will no longer play a part in future rounds. Our fail-stop adversary cannot apply any quantum operation on the system - she can only halt certain processes based on the known pure state of the system. Our pure state assumption forces us to require that the state of the dead processes is not traced-out as this would amount to a measurement and the adversary will know the resulting pure state[3].

The Byzantine Adversary

The Byzantine adversary we consider here is very much like the Fail-Stop adversary in her knowledge, knowing the pure state of the system and has unbounded computational capabilities.

The Byzantine adversary will have the ability to take over processes, operate on their work space, and send whatever messages (quantum or classic) the adversary wishes to send. A process which was taken over will obey the adversary until the end of the protocol. The only limitation on the adversary is that she can not take over more than t processes. A process which was taken over will be considered faulty and the three conditions stated before (Agreement, Validity and Termination) will not apply to it.

3. THE TRICK

As mentioned in the introduction, a constant expected time agreement protocol tolerating up to $t < n/3$ faults can be easily reduced to the task of generating within a constant number of rounds a weak global coin (see [16, 11, 12]).

Definition: Let G be a protocol for n players (with no input) where each player P_i outputs a (classical) bit $v_i \in \{0, 1\}$. We say that the protocol G is a t-resilient weak global coin protocol with fairness $p > 0$, if for any t-adversary and any value $b \in \{0, 1\}$, with probability at least p, $v_i = b$, for all good players P_i.

The standard reduction gives

Theorem: For $t < n/3$, a t-resilient, constant round, weak global coin protocol, with fairness $p > 0$, implies a t-resilient Byzantine Agreement protocol with $O(1/p)$-expected number of rounds.

The protocols described in sections 4 and 5 are variations on well known classical weak global coin protocols, which deal with weaker adversaries - mainly adversaries which are either not adaptive and thus have to choose the bad players in advance, or have only partial knowledge and thus can't choose the bad players according to an efficient strategy. This is done mostly by sending entangled (usually error corrected) qubits instead of classical bits. The sender will not choose the value $|0\rangle$ or $|1\rangle$ but rather send a superposition $(|0\rangle + |1\rangle)/\sqrt{2}$ of the values. The adversary will know the exact pure state the sender is sharing between the processes but this provides no information what will be the value of a later measurement. Special care is needed in the Byzantine case since a message sent by a good process to a faulty process can be measured immediately and give the adversary information whether to corrupt the sender, or some other process during this communication round. This information leakage can be avoided by careful use of quantum error correcting coding, or quantum secret sharing.

This way the adversary will not have any information before the value is measured at a later stage by the "good" processes and therefore she will not have enough information to act correctly to postpone the agreement. In a way, our protocol for the fail-stop model is a "purified" version of the classical (non adaptive) fail-stop protocol of Chor, Merritt and Shmoys [9], and the quantum Byzantine protocol is a purified version of the classical (secure channels) protocol by Feldman and Micali [12].

4. THE FAIL-STOP PROTOCOL

In this section we present our two round quantum weak global coin protocol for the fail stop model.

Protocol QuantumCoinFlip for P_i:

1. **Round I:**
   Generate the state
   $$|Coin_i\rangle = \frac{1}{\sqrt{2}}|0, 0, \ldots, 0\rangle + \frac{1}{\sqrt{2}}|1, 1, \ldots, 1\rangle$$
   on n qubits and send the k-th qubit to the k-th player (keeping one part to yourself).

2. Generate the state
   $$|Leader_i\rangle = \frac{1}{n^{3/2}} \sum_{a=1}^{n^3} |a, a, \ldots, a\rangle$$
   on n-qudits, an equal superposition of the numbers between 1 and n^3. Distribute the n qudits between all the players.

3. receive the the quantum messages from all players and wait for the next communication round, thus forcing the adversary to choose which messages were passed.

4. **Round II:**
   Measure (in the standard base) all $Leader_j$ qudits received in round I. Select the player with the highest Leader value (ties broken arbitrarily) as the "leader" of the round.
   Measure the leader's coin in the standard base.

5. Set the output of the QuantumCoinFlip protocol:
   v_i = measurement outcome of the leader's coin.

Lemma 1: For $t < n/3$ the protocol QuantumCoinFlip is a weak global coin protocol with fairness $p \approx 1/3$, for any fail-stop t-adversary.

Proof: The adversary has to decide which players to halt during the first round of the protocol. Note that this is before the measurement assigns random values to the $Leader_i$ and the $Coin_i$ variables on the second round. Since during the first round all the players are symmetric the best the adversary can do is to stop up to t players from transmitting all their messages during the first round. The adversary

[3] This assumption is added just to simplify the presentation of our fail-stop protocol. We could also "trace out" dead process at the expense of replacing the simple GHZ type states of our protocol with quantum error erasure codes.

will succeed only if she happens to stop the player which is going to be the leader, but this will happen with probability less than 1/3. The probability of a tie for the maximum in $Leader_i$ values is negligible (actually less than 4% even for $t = 1$) and therefore we will dismiss it.

Given that the leader's messages were delivered during the first round, the probability for any value b to be the outcome of the protocol for all the (non faulty) players is exactly 1/2, so the probability that all the players will output b is at least 1/3.

With this at hand we immediately obtain.

Theorem 1: There is a constant expected number of rounds quantum protocol for synchronous agreement in the presence of a full information computationally unbounded adaptive fail-stop adversary tolerating up to $t < n/3$ failures.

5. THE BYZANTINE PROTOCOL

The simple QuantumCoinFlip protocol of the previous section is clearly useless in the Byzantine fault model for the following obvious reasons:

- Controlling a single faulty player the adversary can measure and collapse the states of all the players during the first round and thus know which process to stop (the leader if its coin is not the value sought by the adversary).
- With a single faulty process at hand, the adversary can make sure that this process is chosen by distributing the maximal value as its *Leader* value instead of the superposition of all the values. Therefore setting its coin value, the adversary has essentially complete control on the outcome.

We can easily prevent the first problem by sharing the information using quantum error correcting codes. Handling the second requires a few more tools.

The protocol we present will be based largely on the constant expected time Byzantine agreement protocol of Feldman and Micali in [12], replacing the classical "Graded Verifiable Secret Sharing Protocol" with a "graded" variant of the quantum verifiable secret sharing protocol of Crépeau, Gottesman and Smith in [10].

To understand our protocol we briefly describe the ideas behind the *ObliviousCoin* protocol of [12]. To generate a random coin they use the following procedure: We first want to assign a random integer in the range $[0, n-1]$ to each player. As we can not let the faulty players select their own value, each player P_k selects a random integer s_i^k, $0 \leq s_i^k < n$ for each other player P_i. P_k uses a verifiable secret sharing scheme to distribute s_i^k.

After the end of this phase the players agree which secrets were properly shared. At this stage we open all the properly shared secrets and assign the value

$$s_i = \sum \left\{ s_i^k \middle| \begin{array}{l} \text{for all } k \text{ for which } s_i^k \\ \text{was properly shared} \end{array} \right\} \pmod{n}$$

Note[4] that s_i is a random number since the sum contains secrets randomly selected by the good players and care was taken to assure that values selected by faulty players do not depend on the values selected by the honest players.

This could be done in constant number of rounds, if we could implement the agreements required. This by itself would require a constant round protocol for Byzantine Agreement which is exactly the problem we are trying to solve. This is ellgantly solved in [12] by replacing the agreements by their constant round Grade-Cast agreement protocol. With these weakened agreements they prove that a random number will be assigned to all players (except for bad players that were publically caught cheating during the process), and that during the opening phase all that bad players can do is to have the random number assigned to them revealed to only a subset of the good players. In any case all good players will receive the random numbers assigned to all the good players. As all assigned numbers are random, with probability $\geq 2/3$ the minimum value has been assigned to an honest player. Thus selecting the parity bit of the minimum value will provide a random enough bit shared by all the good players. This provides a constant round weak global coin protocol with fairness $> 1/4$, and with this we obtain their classical constant expected time Byzantine Agreement protocol. We refer the reader to [12] for further details.

The scheme sketched above requires private channels which we can not afford in our full information strong adversary model. Therefore we replace all random secrets by the superposition

$$|\phi\rangle = \frac{1}{\sqrt{n}} \sum_{a=0}^{n-1} |a\rangle$$

We cannot distribute the state $|\phi, \phi, \ldots, \phi\rangle$ since the bad players can collpase the state. To prevent bad players from doing so we encode the state using the Quantum Verifiable Secret Sharing (QVSS) of [10] and send each player their share of the secret. Here again the verification requires Byzantine Agreement, but replacing the agreement by the Grade-Cast protocol is good enough.

For $t < n/4$ the verification stage of the QVSS protocol guarantees that for a good dealer the correct state will be encoded, and that for any, possibly faulty dealer, some particular state will be recovered during the recovery stage.

We note that for the purpose of our Byzantine quantum coin flip protocol the recovery stage is much simpler. Each player measures his share of the QVSS and sends the classical value to all other players. The verification stage guarantees, with high probability, that in the presence of up to $t < n/4$ faulty players all the good players will recover the same classical value (which is the same value that would result from a direct measurement of the encoded state).

Thus replacing the graded-VSS in the constant round Byzantine agreement protocol of [12] with the graded version of the QVSS of [10], while sharing the known state ϕ instead of sharing random classical numbers, we obtain:

Theorem 2: There is a constant expected number of rounds quantum protocol for synchronous Byzantine agreement in the presence of a full information computationally unbounded adaptive Byzantine adversary tolerating up to $t < n/4$ failures.

[4] This is a very subtle point in [12], and they have a beautiful proof showing that in their protocol the adversary is not able to choose her secrets such that they are dependent on those of the good players. We will not have this problem, as our "secrets" will be the known superposition of all the numbers between 0 and $n-1$.

Improving the Fault-Tolerance to $t < n/3$

We note that for the simple use of the QVSS in our protocol we need to protect against bit-flip errors but do not mind phase flip errors. In such case the scheme of [10] can handle up to $t < n/3$ faults giving us the opitmal tolerance of $t < n/3$. We briefly sketch the arguments how and why this works.

The QVSS shares the secret using (a two layered variant of) the polynomial quantum error correcting codes of Aharonov and Ben-Or [1]. Sharing the information using degree $2t$ polynomials when $t < n/4$ protects the state against t bit-flip and against t phase-flip errors. For $t < n/3$ we share the information using polynomials of degree t. This can protect the quantum state against t bit flip errors but cannot protect against t phase flip errors. Note however that the code does protect against t-erasures! Therefore controlling at most t shares the Byzantine adversary cannot collapse the state of the good players.

During the verification stage we simply skip the phase checking stage. The verification we do perform guarantees that the state held by the good players is supported with high probability by the space spanned by the standard base vectors which represent polynomials of degree $\leq t$. Thus by the end of a successful verification stage the dealer is committed to a particular distribution on the values that will be recovered during the measurement and reconstruction steps. With these modifications we obtain

Theorem 3: There is a constant expected number of rounds quantum protocol for synchronous Byzantine agreement in the presence of a full information computationally unbounded adaptive Byzantine adversary tolerating up to the optimal $t < n/3$ failures.

6. THE ASYNCHRONOUS CASE

Feldman has extended the fast synchronous Byzantine agreement protocol of [12] to the asynchronous secure channels environment tolerating $t < n/4$ faults. For details see Canetti and Rabin [7]. Our Byzantine protocol can easily be extended to this setting as well. We note that for the QVSS in the asynchronous setting we must handle t-errors and t-erasures. A straight forward extension of the QVSS of [10] will work with polynomial codes of degree $3t$ when $t < n/6$. Since in our application we can neglect the phase-flip errors we can use the degree t polynomial codes that can correct up to t erasures, as well as t bit flips in the presence of t-erasures for $t < n/4$.

Note that randomized agreement with probability 1 is possible in this model as long as $t < n/3$ leaving an intriguing open question whether we can close this gap.

7. REFERENCES

[1] D. Aharonov and M. Ben-Or, "Fault tolerant quantum computation with constant error rate," quantph/9906129. *Preliminary version in STOC '97. Submitted to SIAM J. Comp.*, June 1999.

[2] J. Aspnes, "Lower bounds for distributed coin-flipping and randomized consensus", *Journal of the ACM* 45(3):415-450, May 1998.

[3] Ziv Bar-Joseph and Michael Ben-Or, "A Tight Lower Bound for Randomized Synchronous Computing ," *proceedings of the Seventeenth Annual ACM Symposium on Principles of Distributed Computing (PODC)*, Puerto Vallarta, Mexico, pp. 193–199, 1998.

[4] M. Ben-Or, "Another advantage of free choice: Completely asynchronous agreement protocols," In *In Proceedings of the 2nd Annual ACM Symposium on the Principles of Distributed Computing*, pp. 27–30, 1983.

[5] M. Ben-Or and E. Pavlov, unpublished manuscript.

[6] Gabriel Bracha , "An asynchronous $[(n-1)/3]$-resilient consensus protocol," In *Proceedings of the third annual ACM symposium on Principles of distributed computing* , JACM, pp. 154–162, 1984.

[7] Ran Canetti and Tal Rabin, "Fast asynchronous Byzantine agreement with optimal resilience." STOC 1993: 42-51.

[8] Benny Chor and Brian A. Coan, "A Simple and Efficient Randomized Byzantine Agreement Algorithm," *IEEE Trans. Software Eng.* 11(6): 531-539, 1985.

[9] Benny Chor, Michael Meritt and David B. Shmoys, "Simple Constant Time Consensus Protocols In Realistic Failure Models" *Journal of the ACM*, vol. 36, no. 3, pp. 591–614, Jan. 1989.

[10] Claude Crépeau, Daniel Gottesman and Adam Smith, "Secure Multi-party Quantum Computation," In *34th ACM Symposium on the Theory of Computing*, STOC, pp. 643–652, 2002.

[11] Cynthia Dwork, David B. Shmoys, Larry J. Stockmeyer, " Flipping Persuasively in Constant Time," SIAM J. Comput. 19(3): 472-499 (1990)

[12] P. Feldman and S. Micali, "An Optimal Protocol for Synchronous Byzantine Agreement, " *SIAM Journal of Computing*, vol. 26, pp. 873–933, 1997.

[13] J. A. Garay and Y. Moses, "Fully Polynomial Byzantine agreement in t + 1 rounds.," In *Proceedings of th 25th Annual ACM Symposyum on Theory of Computing*, ACM, pp. 31–41, 1993.

[14] Nancy A. Lynch, *Distributed Algorithms*, Morgan Kaufman, 1996.

[15] M. Pease , R. Shostak , L. Lamport, "Reaching Agreement in the Presence of Faults", Journal of the ACM (JACM), v.27 n.2, p.228-234, April 1980

[16] M. O. Rabin, "Randomized Byzantine Generals," In *Proceedings of the 24th Annual IEEE Symposyum on the Foundations of Computer Science*, IEEE, pp. 403–409, 1983.