Lecture 5: Road Map

- Bound on $f$
- Impossibility for BA with $n = 3$ and $f = 1$
- Impossibility for BA with $f \geq \frac{n}{3}$
- Bounding $f$ in the Authenticated Case
- Weak and Strict BA
- Synchronous, Un-authenticated, BA
- Impossibility of BA for $f = 1$ and $n > 2$
- Impossibility of BA for $f = 2$ and $n > 3$
- General Impossibility proof
- Reading for Next Lecture
Bound on $f$

We now show that there is no protocols that solves BA without authentication if $f \geq \frac{n}{3}$.

We first establish the claim for the case of $n = 3$, showing that there is no BA if $f \geq 1$. We restrict to the case that the possible values are 0 and 1.

Assume that there is a solution for three processes, say 1, 2, and 3.

Let $r$ be the maximal round number in which a decision is made in the two executions where the all start with 0 and 1, and no failure occurs.
Impossibility for BA with $n = 3$ and $f = 1$

Connect $2 \cdot r$ copies of the processes, using the solution (with $n = 3$ and $f = 1$), into a new system. We make no requirements on the behavior of the new system. Note, however, that it is well defined: Every process in the new system follows its well defined protocol (state-machine.)

Consider now the following execution $\alpha$ where all processes at top start with 0 and those at bottom start with 1.
Impossibility for BA with $n = 3$ and $f = 1$

Claim: Each two consecutive processes reach the same decision.

Proof: Consider a pair of adjacent processes $L$ and $R$.

Case a: Both have same value, and each has $r$ neighbors on both sides with the same initial value. These obviously cannot tell apart $\alpha$ from the execution of the triangle in which all are correct and start with same value, and therefore decide on their initial value. (No information revealing the existence of another value can make to the pair in $r$ rounds or less, and they decide by round $r$.)}
Impossibility for BA with $n = 3$ and $f = 1$

Case b: Both have same initial value and one has less than $r$ neighbors on that side with same initial value. Then the pair cannot tell this execution from an execution of the triangle where their third partner is faulty. They both reach the same decision.

Case c: Each process has a different initial value. Again, there exists an execution of the triangle in which they are both correct, start with the same initial values, and their third partner is faulty (sends to $L$ in triangle whatever $L$’s left neighbor in sends it in ring, and sends to $R$ in triangle whatever $R$’s right neighbor sends it in ring.) Since both $L$ and $R$ reach a decision in the triangle, they will reach this decision in the ring.
Note on “Triangle”

If $v_i = v_j = v_k = 0$ and $v_{k'} = 1$, then $i$ and $j$ cannot tell execution of right structure from execution of left structure (triangle) in which $k$ is faulty and behaves the same as in right structure towards both $i$ and $j$. Since we assume triangle reaches BA, both $i$ and $j$ on left eventually reach agreement. They must therefore reach the same agreement on right. Whatever this agreement is, the only thing we care about is that eventually $d_i = d_j \neq \bot$. 
Impossibility for BA with $n = 3$ and $f = 1$

From the claim it follows that eventually all $6 \cdot r$ processes on the ring reach the same decision value. However, all $r$ processes on the center top (case a) must decide on 0 since they cannot distinguish $\alpha$ from an execution of the triangle where they all start with 0 and there are no faults. Similarly, all $r$ processes on the center bottom (case a) must decide on 1 since they cannot distinguish $\alpha$ from an execution of the triangle where they all start with 1 and there are no faults. Hence, we obtained a contradicting to the claim that they all reach the same value. We must therefore concluded that there is no solution to the triangle, i.e., there is not solution to BA with $f = 1$ and $n = 3$. 
Impossibility for BA with $f \geq \frac{n}{3}$

To show that no BA is attainable with $f \geq \frac{n}{3}$, it suffices to assume that there is such a protocol and show how to derive from it a solution to $n = 3$, $f = 1$. This is done by (1) partitioning the set of $n$ processes into three sets, each containing no more than $f$ processes with all faulty processes in the same set, and (2) letting each process in “the triangle” mimic the behavior of the group in the original solution.
Bounding $f$ in the Authenticated Case

Once again, it seems that we have two contradictory results: On one hand, we showed how to attain authenticated BA with any $f < n$ faulty processes. On the other hand, we proved that no solution is possible if $f \geq \frac{n}{3}$ (where, of course, we meant for the unauthenticated case.)

Where does the impossibility proof fail once we allow authentication?
Weak and Strict BA

There are two notions of BA: One (Strict BA) in which if all correct processes start with the same value then this is the decision value, and the other (Weak BA) in which if all processes start with the same value then this is the decision value.

Obviously, for possibility results, the strict notion is harder, and for impossibility results, the weak notion is harder. We showed the impossibility results for the weak notion (which imply the impossibility of the strong notion, though proving it directly is somewhat easier), and with the possibility results we have focused on the weak notion, though some of our protocols do, or can easily be made to, attain strict BA. (The notable exception are the more efficient authenticated BA protocols.)
Synchronous, Un-authenticated, BA

Assumptions:

- Each process can tell the sender of each message it receives.
- The only decision values are 0 and 1 (see textbook how to extend the solution to other decision values.\(^1\))
- We have a Consistent Broadcast mechanism, in which messages are of the form \((m, r, i)\) that guarantees:
  1. If \((m, i, r)\) is broadcast by a correct \(i\) at round \(r\), it is accepted by all correct processes by round \(r + 1\).
  2. If \((m, i, r)\) is not broadcast by a correct \(i\) at round \(r\), it is never accepted by any correct process.
  3. If a message is accepted by some correct process, it is accepted by all correct processes at most one round later.

---

\(^1\) See also in textbook how to “correct” the EIG protocol to apply to the unauthenticated case.
Synchronous, Un-authenticated, BA

The algorithm proceeds in \( f + 1 \) phases, each consisting of an odd round followed by an even round. The only messages sent are of the form \((r, i)\), and they are only sent on odd rounds. At round 1, all processes \( i \) whose initial value is 1 (consistent) broadcast \((1, i)\). At an odd round \( r \), \( i \) broadcasts a message if:

1. it hadn’t done that before; and
2. it had accepted messages from at least \( f + \frac{r-1}{2} \) different processes.

At the end, a process decides 1 if it had accepted messages from \( 2f + 1 \) different processes, and 0 otherwise.
Correctness of Protocol

Claim: The protocol solves strict BA if $n > 3f$.

Proof: Termination is obvious. For validity, if all correct processes start with 1, then, according to property 1 of consistent broadcast, all correct processes accept $n - f \geq 2f + 1$ messages by round 2; thus, they all decide 1. Similarly, if they all start with 0, it is easy to show, by induction on number of rounds using property 2 of consistent broadcast, that no correct every broadcasts since no message is ever accepted by a correct process.

It remains to prove agreement. Suppose that a correct $i$ decides 1...
Correctness of Protocol (cont.)

Then \(i\) accepted messages from \(2f + 1\) different processes. Let \(\mathcal{C}\) denote the set of correct processes from whom \(i\) accepted messages. Obviously, \(|\mathcal{C}| \geq f + 1\). Let \(\mathcal{C}_b\) denote the set of \(\mathcal{C}\)'s processes whose initial value is \(b\) (recall that \(\mathcal{C}_1\) includes at least one process!) Every \(j \in \mathcal{C}_1\) b'casts at phase 1, so all correct processes accept \(j\)'s message by the end of the first phase. (Property 1 of b'cast.)

If \(|\mathcal{C}_1| \geq f + 1\), then all correct processes accept \(f + 1\) messages by the end of the first phase, and the all b'cast by the second phase. Thus, they all accept messages from at least \(n - f\) processes by the end of the second phase, and decide 1.
Correctness (end of Pf.)

Otherwise $|C_0| > 0$. Let $j \in C_0$ and assume $j$ b’casts at phase $p$ (the fact that it must b’cast follows the definition of $C$ and property 2 of b’cast.) It thus follows that $j$ accepts messages from at least $f + p - 1$ non-$j$, processes by the beginning of phase $p - 1$. Thus (property 3), all correct processes accept from $f + p - 1$ non-$j$ processes by the end of phase $p$. Since $j$ is correct and b’casts at phase $p$, the all have accepted from $f + p$ processes by the end of phase $p$.

If $p < f + 1$, then they all b’cast by phase $p + 1$ and accept messages from $n - f$ processes by the end of that phase. Else $f + p = 2f + 1$ and they all accept the required number of messages by the end of phase $f + 1$. In any event, all correct processes decide 1.
Complexity of Algorithm

It’s easy to see that the protocol requires $2(f + 1)$ rounds, and $n$ broadcast invocations.

Each invocation of consistent broadcast requires $O(n^2)$ messages, thus the communication complexity is $O(n^3)$ messages.

See Pages 126–127 for description of consistent Broadcast.
Impossibility of BA with $f$ rounds or less

We will assume:

- A full communication graph.
- Possible values are 0 and 1.
- Decisions are made only at round $f$.
- Stopping failures only.
- At every round, every process sends a message to every other process.

Note that all assumptions simplify the task of obtaining BA, thus obtaining impossibility result under them does not impair generality. (I.e, solutions impossible even under relaxed assumptions.)
Impossibility of 1-round BA for $f = 1$, $n > 2$

We consider a sequence of execution, each constructed from the previous one by a removal or addition of a single message. Thus, the only processes who can distinguish between the two executions are the sender and receiver of the message. It therefore follows that all correct processes reach the same decision in each two consecutive executions, and, consequently, the reach the same decision in every execution.

The first execution is one in which all process start with 0 and no fault occurs. The last execution is one in which all start with 1 and no fault occurs. We thus obtain a contradiction.

The sequence can be subdivided to phases, each “belonging” to one process (thus there are $n$ phases.) Each phase, “belonging” to process $p$, can be further divided to two subphases: A “corruption” subphase in which $p$’s messages are removed (one by one), and, after an execution in which $p$ sends no messages obtains, its initial value is changed from 0 to 1 and a the “correction” subphase begins. In the fixing subphase, $p$’s messages are added, one by one, until the end of the phase, where $p$ is correct and its initial value is 1.
Impossibility of BA 2-round BA for 
\( f = 2, \ n > 3 \)

In principle, the proof is similar to the previous one. Only here we need to work harder to corrupt (and correct) a process: to corrupt \( p \) at the second round, we can just eliminate its messages one by one. However, corrupting it at the first round, we need to apply more care since if we just remove a round 1 message it sends to \( q \), then this may impact \( q \)’s round 2 messages.

Thus, after removing \( p \)’s round 2 messages, we remove \( q \)’s round 2 messages (one by one.) We can do that since we assumed that \( f = 2 \), thus we can corrupt 2 processes at a time. After we are done, we remove \( p \)’s round 1 message to \( q \), and then, one by one, restore \( q \)’s round 2 messages. Thus, at the end of this process, \( p \)’s round 1 message to \( q \) is removed, and, after we do that for every \( q \), all of \( p \)’s messages are removed. This is the end of the corruption case of \( p \), so its initial value can change, and, in a similar manner, it can be corrected.
General Impossibility proof

The above proofs are the two first steps in the general claim no BA is possible in \( \leq f \) rounds for any \( f \geq 1 \) and \( n > f + 2 \). The general proof follows the same ideas. Thus, at each corruption phase it is necessary to corrupt \( f \) processes. Details can be found in the textbook (Section 6.7). Note that at each execution, we need to leave two processes correct and intact so that we can argue they cannot distinguish between the two consecutive executions and therefore reach the same decision.

Since we assumed stopping failures only, the impossibility proof applies for both strong and strict BA.
Reading for Next Lecture

Chapter 7 of textbook, focusing on Section 7.3.

Another good reference for this material is Chapter 7 in *Concurrency control and Recovery in Database Systems*, by Bernstein, Hadzilacos, and Goodman. Addison Wesley, 1987.