Ch 14 – Consensus

- The problem and its relevance
- The Binary Consensus
- Generalization to multivalued Consensus
- Basic lower bounds
**Def - Consensus**

**Definition 14.1 (Consensus).** Each node \( v \in V_g \) is given an input \( x_v \in X \). To solve Consensus, an algorithm must compute output values \( o_v \in X \) at all correct nodes \( v \in V_g \) meeting the following conditions:

- **Agreement:** There is \( o \in X \) so that \( o_v = o \) for all \( v \in V_g \). We refer to \( o \) as the output of the Consensus algorithm.
- **Validity:** If there is \( x \in X \) so that for all \( v \in V_g \) it holds that \( x_v = x \), then \( o = x \).
- **Termination:** There is \( r \in \mathbb{N} \) satisfying that each \( v \in V_g \) terminates and outputs \( o_v \) by the end of round \( r \).

The algorithm has round complexity \( R \in \mathbb{N} \), if it terminates in \( R \) rounds in all executions.

If \( X = \{0, 1\} \), we refer to the task as Binary Consensus, and denote the input of node \( v \) by \( b_v \) (to indicate that it is a bit).
Breakout room:

The power of Consensus
-- Safe Broadcast –
a node is forced to send the same value to every other node

-- \((n-f \geq) 2f+1; f+1 ; 1\)

-- Lynch-Welch ?
Byzantine case
Selfstabilized version

-- Other examples?
Consensus (with at most f Byzantine nodes) is not solvable if:
1) fully asynchronous system
2) more than f faulty in total (including self-stabilized)
3) $n \leq 3f$
4) nodes connectivity $\leq 2f$
5) less than f+1 rounds

Other remarks:
1) with cryptography
   • no limit of ration n to f
   • node connectivity $> f$; round complexity f+1
2) in the literature sometimes "agreement" is used for "safe broadcast" – agreeing on the leaders value
3) the problem was studies for many many different models
4) we will assume full synchrony, and no cryptography
Consensus – universal (complete graphs)

**Theorem 14.4.** Suppose that $G$ is a complete graph on $n$ nodes, $X$ is a set of feasible messages, and $\mathcal{A}$ is a Consensus algorithm for input set $X \cup \{\perp\}$ on $G$ of round complexity $R$. Then we can simulate communication by Safe Broadcast for messages in $X$ on $G$, where $R + 1$ rounds are required for each simulated round. Denoting by $M$ the maximum message size of $\mathcal{A}$, the maximum message size of the simulation is $n \cdot \max\{M, \lceil\log(|X| + 1)\rceil\}$.

Simulate each message sending by a safe broadcast
- each communication exchange is simulated by
- $n$ concurrent safe broadcasts
- takes $R+1$ rounds per sending

By the Consensus properties:
all correct nodes ends each wave of sending holding an identical "message value" associated to each sender
Consensus – universal (general graphs)

Theorem 14.11. Suppose that $\mathcal{A}$ is a Consensus algorithm for a complete graph on $n$ nodes with up to $f$ Byzantine faults. Fix an arbitrary $(2f + 1)$-node connected $n$-node graph $G$. Then there is an algorithm simulating $\mathcal{A}$ on $G$. Its round complexity is at most factor $n$ larger and it uses messages of size $O(n^2(M + \log n))$, where $M$ is the maximum message size of $\mathcal{A}$.

- Simulate each message sending by sending along $(2f+1)$ node independent paths (Menger's Theorem)
  - $f+1$ support for a message or null ($\perp$)
  - feed algorithm $\mathcal{A}$ with that message
  - for $v, w \in V_g$ $w$ accepts message $m$ from $v$ iff $v$ sent it in that specific round
Definition 14.1 (Consensus). Each node $v \in V_g$ is given an input $x_v \in X$. To solve Consensus, an algorithm must compute output values $o_v \in X$ at all correct nodes $v \in V_g$ meeting the following conditions:

- **Agreement:** There is $o \in X$ so that $o_v = o$ for all $v \in V_g$. We refer to $o$ as the output of the Consensus algorithm.
- **Validity:** If there is $x \in X$ so that for all $v \in V_g$ it holds that $x_v = x$, then $o = x$.
- **Termination:** There is $r \in \mathbb{N}$ satisfying that each $v \in V_g$ terminates and outputs $o_v$ by the end of round $r$. 
The Phase King Alg

- Binary Consensus
- Leader per phase
- $f+1$ phases
- Each phase is composed of 3 broadcasts

The idea: once we have a correct leader a value will be set and will never be changed again

---

Algorithm 18 Phase King Algorithm at node $i \in V_g$. Note that for convenience the code assumes that $i$ also receives its own broadcasts and all messages are consistent with the format required by the algorithm (i.e., invalid or missing messages by faulty nodes are replaced by valid default values).

<table>
<thead>
<tr>
<th>Line</th>
<th>Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>$op \leftarrow b_i$</td>
</tr>
<tr>
<td>2</td>
<td>for $j = 1 \ldots f + 1$ do</td>
</tr>
<tr>
<td>3</td>
<td>\quad strong $\leftarrow 0$</td>
</tr>
<tr>
<td>4</td>
<td>\quad broadcast $op$ $\quad \triangleright$ first broadcast</td>
</tr>
<tr>
<td>5</td>
<td>\quad if received at least $n - f$ times $op$ then</td>
</tr>
<tr>
<td>6</td>
<td>\quad \quad strong $\leftarrow 1$</td>
</tr>
<tr>
<td>7</td>
<td>\quad end if</td>
</tr>
<tr>
<td>8</td>
<td>\quad if strong $= 1$ then</td>
</tr>
<tr>
<td>9</td>
<td>\quad \quad broadcast $op$ $\quad \triangleright$ second broadcast</td>
</tr>
<tr>
<td>10</td>
<td>\quad end if</td>
</tr>
<tr>
<td>11</td>
<td>\quad if received fewer than $n - f$ times $op$ then</td>
</tr>
<tr>
<td>12</td>
<td>\quad \quad strong $\leftarrow 0$</td>
</tr>
<tr>
<td>13</td>
<td>\quad end if</td>
</tr>
<tr>
<td>14</td>
<td>\quad if $i = j$ then $\quad \triangleright$ king’s broadcast</td>
</tr>
<tr>
<td>15</td>
<td>\quad \quad if received at least $f + 1$ times $0$ then</td>
</tr>
<tr>
<td>16</td>
<td>\quad \quad \quad broadcast $0$</td>
</tr>
<tr>
<td>17</td>
<td>\quad \quad else</td>
</tr>
<tr>
<td>18</td>
<td>\quad \quad \quad broadcast $1$</td>
</tr>
<tr>
<td>19</td>
<td>\quad \quad end if</td>
</tr>
<tr>
<td>20</td>
<td>\quad end if</td>
</tr>
<tr>
<td>21</td>
<td>\quad if strong $= 0$ and received $b \in {0, 1}$ from node $j$ then</td>
</tr>
<tr>
<td>22</td>
<td>\quad \quad $op \leftarrow b$ $\quad \triangleright$ if not sure, obey the king</td>
</tr>
<tr>
<td>23</td>
<td>\quad end if</td>
</tr>
<tr>
<td>24</td>
<td>end for</td>
</tr>
<tr>
<td>25</td>
<td>return $op$</td>
</tr>
</tbody>
</table>
Phase King Algorithm - 2

1: \( \text{op} \leftarrow b_i \)
2: \textbf{for} \( j = 1 \ldots f + 1 \) \textbf{do}
3: \hspace{1em} \text{strong} \leftarrow 0
4: \hspace{1em} \text{broadcast} \ \text{op} \quad \triangleright \text{ first broadcast}
5: \hspace{1em} \textbf{if} \ \text{received at least} \ n - f \ \text{times} \ \text{op} \ \textbf{then}
6: \hspace{2em} \text{strong} \leftarrow 1
7: \hspace{1em} \textbf{end if}
Phase King Algorithm - 3

8: if strong = 1 then
9:    broadcast op  \text{ \texttt{\textgreater} second broadcast }  \\
10:   end if
11:   if received fewer than $n - f$ times $op$ then
12:      strong $\leftarrow 0$
13:   end if

- only correct nodes with strong=1 participate in this round
- there can be only \textbf{a single value with support $\geq f+1$}
- if we have had a consistent $op$ value, all correct would get at least $n-f$ support
- if any correct sees support to $op$, since $n \geq 3f+1$ each correct sees at least $f+1$ support to $op$ (if it holds $op$)
- If we wouldn't be in a consistent $op$ state some may set to 0
Phase King Algorithm - 4

14: if \( i = j \) then
15:     if received at least \( f + 1 \) times 0 then
16:         broadcast 0
17:     else
18:         broadcast 1
19:     end if
20: end if

• if any correct saw \( n-f \) support to \( op \) in the previous stage
  the leader would obtain that value
• thus, if we would be in a consistent \( op \) state, that would
  be the value the leader would see at this stage
• in such a case a correct leader would broadcast that value
• in any case, a correct leader broadcasts a value to all correct
Phase King Algorithm - 5

if \( i = j \) then
  if received at least \( f + 1 \) times 0 then
    broadcast 0
  else
    broadcast 1
  end if
end if

if \( \text{strong} = 0 \) and received \( b \in \{0, 1\} \) from node \( j \) then
  \( \text{op} \leftarrow b \)
end if

- if we would be in a consistent op state, all correct are with \( \text{strong}=1 \) and wouldn't do anything
- if any correct is with \( \text{strong}=1 \), that would be the value a correct leader would send and all others will adopt it
- if the leader is correct we move into a consistent op state
- a faulty leader can affect only those without \( \text{strong}=1 \)
**Definition 14.1 (Consensus).** Each node \( v \in V_g \) is given an input \( x_v \in X \). To solve Consensus, an algorithm must compute output values \( o_v \in X \) at all correct nodes \( v \in V_g \) meeting the following conditions:

- **Agreement:** There is \( o \in X \) so that \( o_v = o \) for all \( v \in V_g \). We refer to \( o \) as the output of the Consensus algorithm.

- **Validity:** If there is \( x \in X \) so that for all \( v \in V_g \) it holds that \( x_v = x \), then \( o = x \).

- **Termination:** There is \( r \in \mathbb{N} \) satisfying that each \( v \in V_g \) terminates and outputs \( o_v \) by the end of round \( r \).
Phase King Algorithm - 6

14: if $i = j$ then
15:     if received at least $f + 1$ times 0 then
16:         broadcast 0
17:     else
18:         broadcast 1
19:     end if
20: end if
21: if strong = 0 and received $b \in \{0, 1\}$ from node $j$ then
22:     $op \leftarrow b$
23: end if
24: end for
25: return $op$

• Validity (or having identical input values) and Termination hold
• once we hit a correct leader – agreement is reached
• within $f+1$ rounds it will happen
• all correct return an identical value
Proof 1

Lemma 14.13. If, for some $b \in \{0, 1\}$ and all $i \in V_g$, $op_i = b$ at the beginning of a phase of Algorithm 18, then the same holds at the end of the phase.

1: $op \leftarrow b_i$
2: for $j = 1 \ldots f + 1$ do
3: \hspace{1em} strong $\leftarrow 0$
4: \hspace{1em} broadcast $op$ \hspace{2em} $\triangleright$ first broadcast
5: \hspace{1em} if received at least $n - f$ times $op$ then
6: \hspace{2em} strong $\leftarrow 1$
7: \hspace{1em} end if
8: \hspace{1em} if strong = 1 then
9: \hspace{2em} broadcast $op$ \hspace{2em} $\triangleright$ second broadcast
10: \hspace{1em} end if
11: if received fewer than $n - f$ times $op$ then
12: \hspace{1em} strong $\leftarrow 0$

throughout the $f+1$ rounds:
- for all $v \in V_g$ op = b.
- strong = 1 and remains 1
Proof 1

Lemma 14.13. If, for some \( b \in \{0, 1\} \) and all \( i \in V_g \), \( op_i = b \) at the beginning of a phase of Algorithm 18, then the same holds at the end of the phase.


```
21:     if strong = 0 and received \( b \in \{0, 1\} \) from node \( j \) then
22:         \( op \leftarrow b \) \hspace{1cm} \triangleright \text{if not sure, obey the king}
23:     end if
24: end for
25: return \( op \)
```

throughout the \( f+1 \) rounds:
- for all \( v \in V_g \) \( \text{op} = b \).
- \( \text{strong} = 1 \) and remains 1
- no leader can change that
- all return that value
Proof 2

Lemma 14.15. Fix a phase $j \in \{1, \ldots, f + 1\}$. There is a $b \in \{0, 1\}$ satisfying that each $i \in V_g$ holding $\text{strong} = 1$ after the first broadcast of phase $j$ has $\text{op}_i = b$.

2: for $j = 1 \ldots f + 1$ do  
3: \hspace{1em} \text{strong} \leftarrow 0  
4: \hspace{1em} \text{broadcast op} \hspace{1em} \triangleright \text{first broadcast}  
5: \hspace{1em} \text{if received at least } n - f \text{ times } \text{op} \text{ then}  
6: \hspace{2em} \text{strong} \leftarrow 1  
7: \hspace{1em} \text{end if}

Count how many different pairs $(i, b_i)$ are sent in total in line 4
- Each $i \in V_g$ contributes a single pair $(i, b_i)$, since they send the same to all
- Each $j \in V \setminus V_g$ may contribute up to two such pairs
- the total $\leq n-f+2f = n+f$
- if there are two different values in line 5 there are at least $2(n-f)$ pairs
- $2(n-f) \leq \text{total number of pairs} \leq n+f$

which implies $n \leq 3f$ – a contradiction