ETH zürich

Asymptotically Optimal Early Termination for Dishonest Majority Broadcast

Giovanni Deligios, ETH Zurich
Ivana Klasovitá, ETH Zurich
Chen-Da Liu-Zhang, Lucerne University of Applied Sciences and Arts
& Web3 Foundation

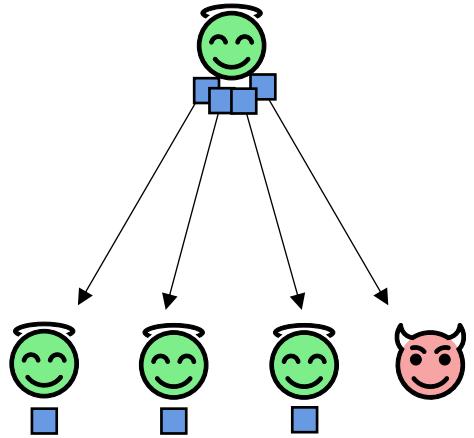
Broadcast

- Set of n players $\mathcal{P}$, designated sender P^*
- Sender P^* holds input x
- Every player P_i determines output y_i
- At most t corruptions

- Protocol needs to satisfy:
  - *Validity*: If P^* is honest, then for each honest player P_i the output is $y_i = x$
  - *Agreement*: For each honest players P_i, P_j, the outputs satisfy $y_i = y_j$
  - *Termination*: All honest players terminate

Broadcast

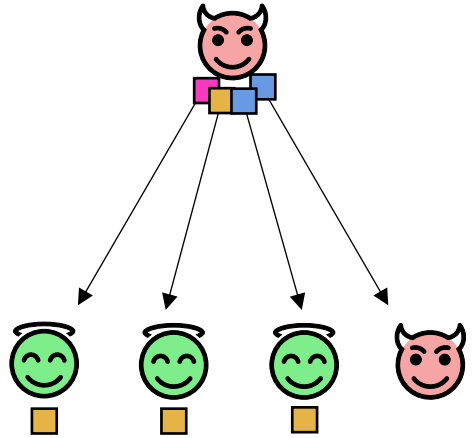
- Set of n players $\mathcal{P}$, designated sender P^*
- Sender P^* holds input x
- Every player P_i determines output y_i
- At most t corruptions



- Protocol needs to satisfy:
  - *Validity*: If P^* is honest, then for each honest player P_i the output is $y_i = x$
  - *Agreement*: For each honest players P_i, P_j, the outputs satisfy $y_i = y_j$
  - *Termination*: All honest players terminate

Broadcast

- Set of n players $\mathcal{P}$, designated sender P^*
- Sender P^* holds input x
- Every player P_i determines output y_i
- At most t corruptions



- Protocol needs to satisfy:
  - *Validity*: If P^* is honest, then for each honest player P_i the output is $y_i = x$
  - *Agreement*: For each honest players P_i, P_j, the outputs satisfy $y_i = y_j$
  - *Termination*: All honest players terminate

General Model

- Deterministic Protocol
- Complete network
- Synchronized Model
- Strongly adaptive, rushing adversary

Broadcast Background

- Pease et al. give upper bounds on the number of byzantine corruptions [PSL80] :
  - $t < n/3$ without authentication
  - $t < n$ with authentication

- Fischer and Lynch proved the lower bound on runtime of $t + 1$ rounds [FL82].
  - ➢ Can we do better, when $f \ll t$ players are corrupted?

Early Termination Broadcast

- Dolev et al. distinguish two types of termination [DRS82]:
  - *Simultaneous termination*: All honest players terminate in the same round.
  - *Eventual termination*: All honest players eventually terminate.

- Let f be the number of actual corruptions.
  - In the worst case, any broadcast protocol with *simultaneous termination* will run for $t + 1$ rounds, no matter the actual number of corruptions [DRS82].
  - In the worst case, any broadcast protocol with *eventual termination* will run for $\min\{f + 2, t + 1\}$ rounds [DRS90].

Early Termination Broadcast Results
Protocols without Authentication

Runtime	Corruption Resilience	Source
$\min\{2f+5, 2t+3\}$	$\boldsymbol{t < n/3}$	[DRS82]
$2f+3$	$t < n/6$	[Rei85]
$\min\{2f+4, 2t+2\}$	$\boldsymbol{t < n/3}$	[TPS87]
$\min\{f(1+1/d)+5, t(1+1/d)\}$, for any constant $d > 0$	$\boldsymbol{t < n/3}$	[BGP92]
$\boldsymbol{\min\{f+2, t+1\}}$	$n > 2t^2 + 3t + 5$	[DRS82]
$\boldsymbol{\min\{f+2, t+1\}}$	$n > \max\{4t, 2t^2 - 2t + 2\}$	[DRS90]
$\boldsymbol{\min\{f+2, t+1\}}$	$n > \lceil\sqrt{t}\rceil \cdot \lfloor 4t + \sqrt{t} + 1\rfloor$	[Coa93]
$\boldsymbol{\min\{f+2, t+1\}}$	$t < n/8$	[GM98]
$\boldsymbol{\min\{f+2, t+1\}}$, with exponential message complexity	$\boldsymbol{t < n/3}$	[BGP92]
$\boldsymbol{\min\{f+2, t+1\}}$	$\boldsymbol{t < n/3}$	[AD15]

Early Termination Broadcast Results

Protocols with Authentication (Signatures)

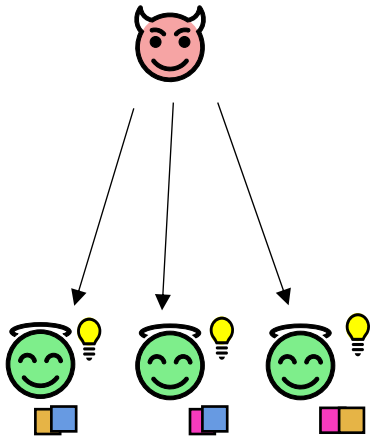
Runtime	Corruption Resilience	Source
$2f + 4$	$t < n/2$	[PT84]
$(d + 5) \cdot (\lfloor f/d \rfloor + 2) + 2$, for a fixed constant d	$t < n/2$	[ELP25]
$O(\min\{f^2, t\})$	$\boldsymbol{t < n}$	[LN24]

- Our contribution: broadcast protocol resilient against $t < n$ corruptions running in $O(\min\{f^2, t\})$ rounds.

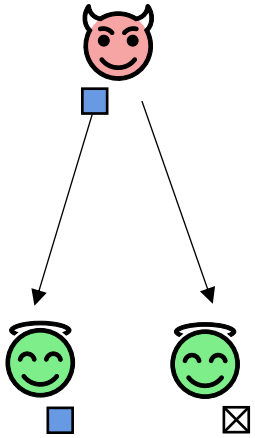
  If $t < (1 - \varepsilon)n$ for some $\varepsilon > 0$, then the protocol runs in $\boldsymbol{O(f)}$ **rounds**.

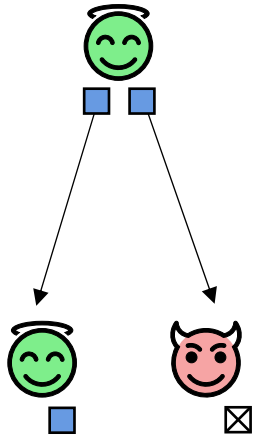
Possible Sender Behavior

- Contradicting messages



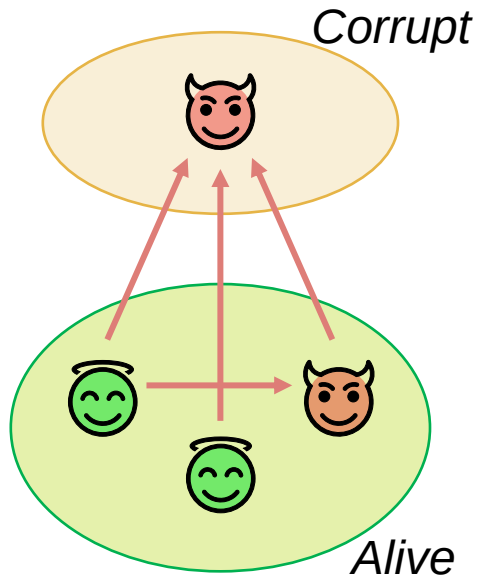
- No message





Polarisers as Certificates

- Loss and Nielsen [LN24] present the idea of polarisers to use instead of certificates, when the sender does not share a message with a player. We say *Pol = (Alive, Corrupt, Accuse)* is a polariser, if
  - All players $\mathcal{P}$ into two disjoint sets: *Alive* and *Corrupt*
  - For each player P_i in *Alive* and each player P_j in *Corrupt,* there is an accusation $Acc_{i,j}$ from the player P_i towards the player P_j in *Accuse*
  - Accusation soundness: honest players do not accuse each other
    - It follows, that all honest players are either in *Alive* or in *Corrupt*
    - Thus, a player P will accept polariser *Pol = (Alive, Corrupt, Accuse)* if
      - *Pol* is a valid polariser
      - Player $P \in$ *Alive*



From Polarisers to Broadcast

- Loss and Nielsen [LN24] propose the following approach to build broadcast using the mechanic of polarisers

	Runtime:
1. Define polariser-cast such that a sender P_j can send their input to all players, and enables honest players to construct a polariser *Pol = (Alive, Corrupt, Accuse)* such that $P_j \in$ *Corrupt*	$O(f)$
2. Use the given polariser-cast to build graded broadcast with justifiable outputs, which means the output of an honest player comes with a proof	$O(f)$
3. Use the king-phase paradigm with rotating kings running the graded broadcast to achieve broadcast	$f \cdot O(f) = \boldsymbol{O(f^2)}$

Generalized Polarizers

- Loss and Nielsen [LN24] present the idea of polarisers to use instead of certificates, when the sender does not share a message with a player. We extend this definition and say that the tuple *Pol = (Alive, Corrupt, Accuse, MakeGraph)* is a polariser, if
  - *MakeGraph* is an algorithm that takes as input *Accuse* and produces the graph $G = (\mathcal{P}, E)$
  - All players $\mathcal{P}$ are divided into two disjoint sets: *Alive* and *Corrupt*
  - For each player P_i in *Alive* and each player P_j in *Corrupt,* there is no path in G between player P_i and player P_j
  - Accusation soundness: honest players do not accuse each other
    - Require all honest players to be either in *Alive* or in *Corrupt*

Example 1: Simple MakeGraph Algorithm

1. Initialize $G \leftarrow \mathrm{K}_{\mathcal{P}}$
2. For each valid accusation $\mathrm{Acc}_{i,j}$ remove edge $\{P_i, P_j\}$ from G
3. Return G

- The graph is constructed starting from a complete graph on $\mathcal{P}$ and removing edges for any accusation
- The generalized definition of Polariser with this MakeGraph algorithm coincides with the definition given by Loss and Nielsen [LN24]

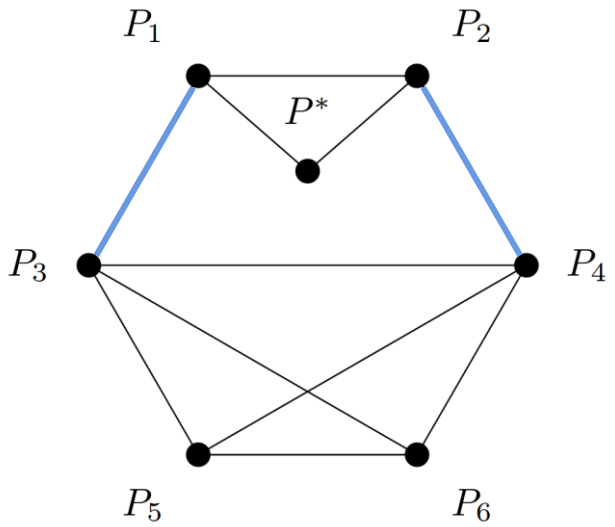
Shortcoming of Simple MakeGraph

1. Initialize $G \leftarrow \mathrm{K}_{\mathcal{P}}$
2. For each valid accusation $\mathrm{Acc}_{i,j}$ remove edge $\{P_i, P_j\}$ from G
3. Return G

- We can analyze the shortcoming of this simple algorithm using the following example: Suppose $n = 7$ with players $\mathcal{P} = \{P^*, P_1, P_2, P_3, P_4, P_5, P_6\}$, and $t < 5$
- Assume P_6 has observed the following accusations:

$$Acc_{3,*},\ Acc_{3,2},\ Acc_{4,*}, Acc_{4,1},\ Acc_{5,*},\ Acc_{5,1},\ Acc_{5,2},\ Acc_{6,*},\ Acc_{6,1}, Acc_{6,2}$$

- Using the simple MakeGraph algorithm, P_6 computes the following graph:



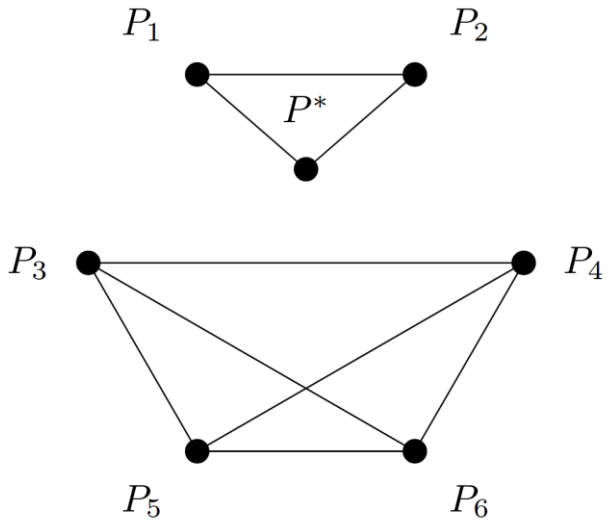
- Given that $t \leq 4$, the number of honest players $h \geq 7 - 4 = 3$
- Honest players don't accuse each other
- Honest players are guaranteed to be part of a clique of size 3
- Edges that are not part of any clique of size 3 do not connect honest players

Example 2: Clique MakeGraph Algorithm

- Let $h = n - t$ be the lower bound on the number of honest players in the following algorithm:

1. **Initialize** $G \leftarrow \mathrm{K}_{\mathcal{P}}$
2. **For each valid accusation** $\mathrm{Acc}_{i,j}$ **remove edge** $\{P_i, P_j\}$ **from** G
3. **Find an edge** e **in** G **which is not part of any clique of size** h **and remove** e **from** G. **Repeat until no such edges exist.**
4. **Return** G

- Suppose $n = 7$ with players $\mathcal{P} = \{P^*, P_1, P_2, P_3, P_4, P_5, P_6\}$, and $t < 5$
- Assume P_6 has observed the following accusations:
  $Acc_{3,*}$, $Acc_{3,2}$, $Acc_{4,*}$, $Acc_{4,1}$, $Acc_{5,*}$, $Acc_{5,1}$, $Acc_{5,2}$, $Acc_{6,*}$, $Acc_{6,1}$, $Acc_{6,2}$
- Using the clique MakeGraph algorithm, P_6 computes the following graph:

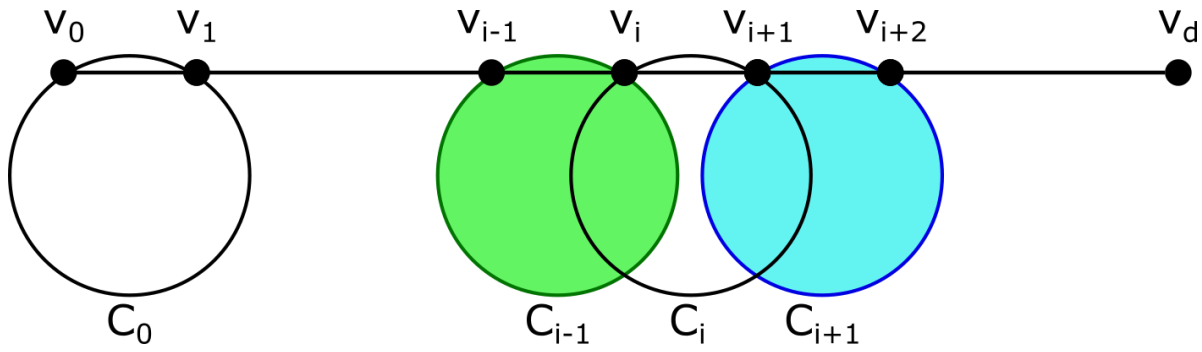


Bounding the Diameter of Resulting Clique Graphs

Lemma: Let G be a graph with n in which every edge is contained in a clique of size h**.** Then G has diameter at most $d \leq 2n/h$**.**

Proof: Let d be the diameter of G.

$$
\begin{aligned}
2n &\geq |C_0 \cup C_2 \cup \cdots| + |C_1 \cup C_3 \cup \cdots| \\
&= |C_0| + |C_2| + \cdots + |C_1| + |C_3| + \cdots \\
&= |C_0| + |C_1| + |C_2| + \cdots + |C_d| \\
&\geq d \cdot h
\end{aligned}
$$



Corollary: Let G be a graph with n in which every edge is contained in a clique of size $h = n - t$**.** Assuming that $t < (1 - \varepsilon)n$ for some constant $\varepsilon > 0$, then the diameter of G is constant.

Proof: $d \leq 2n/h \leq 2n/\varepsilon \cdot n = 2/\varepsilon$

From Polarisers to Broadcast

- Utilize approach proposed by Loss and Nielsen [LN24] to build broadcast using the generalization of polarisers

	Runtime:
1. Construct polariser-cast such that a sender P_j can send their input to all players, and allows honest players that have not received the value from the sender to construct a polariser *Pol = (Alive, Corrupt, Accuse, MakeGraph)* with $P_j \in$ *Corrupt*	$O(1)^*$
2. Use the given polariser-cast to build graded broadcast with justifiable outputs, which means the output of an honest player comes with a proof	$O(1)^*$
3. Use the king-phase paradigm with rotating kings running the graded broadcast to achieve broadcast	$f \cdot O(1) = \boldsymbol{O(f)}^*$

*$t < (1 - \varepsilon)n$ for some constant $\varepsilon > 0$

References

[AD15] Ittai Abraham and Danny Dolev. “Byzantine agreement with optimal early stopping, optimal resilience and polynomial complexity”. In: Proceedings of the forty-seventh annual ACM symposium on Theory of Computing. 2015, pp. 605–614.

[BGP92] Piotr Berman, Juan A Garay, and Kenneth J Perry. “Optimal early stopping in distributed consensus”. In: Distributed Algorithms: 6th International Workshop, WDAG’92 Haifa, Israel, November 2–4, 1992 Proceedings 6. Springer. 1992, pp. 221–237.

[Coa93] Brian A Coan. “Efficient agreement using fault diagnosis”. In: Distributed computing 7 (1993), pp. 87–98

[DRS82] Danny Dolev, Ruediger Reischuk, and H Raymond Strong. “’Eventual’ is earlier than immediate’”. In: 23rd Annual Symposium on Foundations of Computer Science (sfcs 1982). IEEE. 1982, pp. 196–203

[DRS90] Danny Dolev, Ruediger Reischuk, and H Raymond Strong. “Early stopping in Byzantine agreement”. In: Journal of the ACM (JACM) 37.4 (1990), pp. 720–741.

[ELP25] Fatima Elsheimy, Julian Loss, and Charalampos Papamanthou. “Early Stopping Byzantine Agreement in (1 + ε)· f Rounds”. In: International Conference on the Theory and Application of Cryptology and Information Security. Springer. 2025, pp. 398–424.

References

[FL82] Michael J. Fischer and Nancy A. Lynch. “A lower bound for the time to assure interactive consistency”. In: Inf. Process. Lett. 14.4 (1982), pp. 183–186.

[GM98] Juan A Garay and Yoram Moses. “Fully polynomial Byzantine agreement for 3t<n processors in t+1 rounds”. In: SIAM Journal on Computing 27.1 (1998), pp. 247–290.

[LN24] Julian Loss and Jesper Buus Nielsen. “Early Stopping for Any Number of Corruptions”. In: Annual International Conference on the Theory and Applications of Cryptographic Techniques. Springer. 2024, pp. 457–488.

[PSL80] Marshall Pease, Robert Shostak, and Leslie Lamport. “Reaching agreement in the presence of faults”. In: Journal of the ACM (JACM) 27.2 (1980), pp. 228–234.

[PT84] Kenneth J Perry and Sam Toueg. An authenticated Byzantine generals algorithm with early stopping. Tech. rep. Cornell University, 1984.

[Rei85] Rüdiger Reischuk. “A new solution for the Byzantine generals problem”. In: Information and Control 64.1-3 (1985), pp. 23–42.

[TPS87] Sam Toueg, Kenneth J Perry, and TK Srikanth. “Fast distributed agreement”. In: SIAM Journal on Computing 16.3 (1987), pp. 445–457.