Network Algorithms and Complexity
(NTUA-MPLA)

Reliable Broadcast

Aris Pagourtzis, Giorgos Panagiotakos, Dimitris Sakavalas

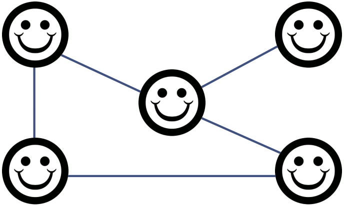
Slides are partially based on the joint work of Christos Litsas, Aris Pagourtzis, Giorgos Panagiotakos and Dimitris Sakavalas

Introduction

Secure Distributed Computing

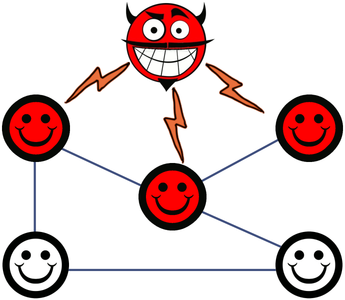
- Several interacting entities (players/agents) that cooperate to achieve a common goal in the absence of a central authority.

Secure Distributed Computing



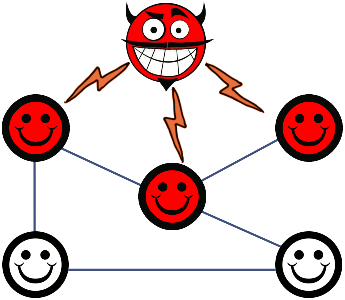
- Several interacting entities (players/agents) that cooperate to achieve a common goal in the absence of a central authority.
- Players arranged in a communication network.

Secure Distributed Computing



- Several interacting entities (players/agents) that cooperate to achieve a common goal in the absence of a central authority.
- Players arranged in a communication network.
- Adversarial Behavior: Corrupted players controlled by a central adversary.

Secure Distributed Computing



- Several interacting entities (players/agents) that cooperate to achieve a common goal in the absence of a central authority.
- Players arranged in a communication network.
- Adversarial Behavior: Corrupted players controlled by a central adversary. Cope with corruption.

Agreement in Unreliable Distributed Systems

Two Major (equivalent) variations of the problem [Lamport, Shostak, Pease 1982].

Broadcast (Byzantine Generals)

The goal is to have some designated player, called the dealer, consistently send a message to all other players.

Agreement in Unreliable Distributed Systems

Two Major (equivalent) variations of the problem [Lamport, Shostak, Pease 1982].

Broadcast (Byzantine Generals)

The goal is to have some designated player, called the dealer, consistently send a message to all other players.

Consensus (Byzantine Agreement)

Goal: Make all players agree on the same output value (decision) given that every player starts with an input value.

If all correct players hold the same input value then the decision is required to be the same as this input value.

Agreement in Unreliable Distributed Systems

Two Major (equivalent) variations of the problem [Lamport, Shostak, Pease 1982].

Broadcast (Byzantine Generals)

The goal is to have some designated player, called the dealer, consistently send a message to all other players.

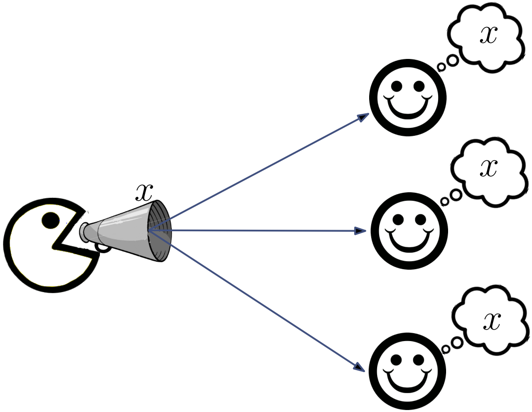
Consensus (Byzantine Agreement)

Goal: Make all players agree on the same output value (decision) given that every player starts with an input value.

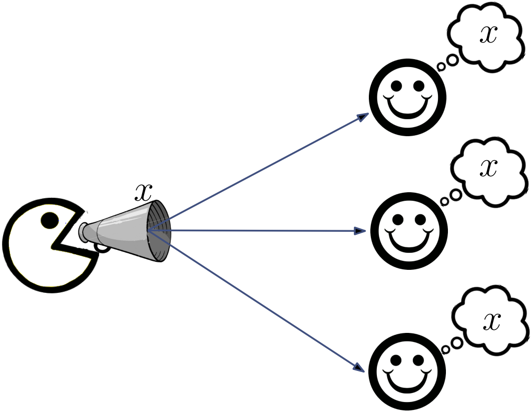
If all correct players hold the same input value then the decision is required to be the same as this input value.

Polynomially equivalent
(for $t < n/2$, where n number of players, t: number of corruptions).

Ideal Broadcast

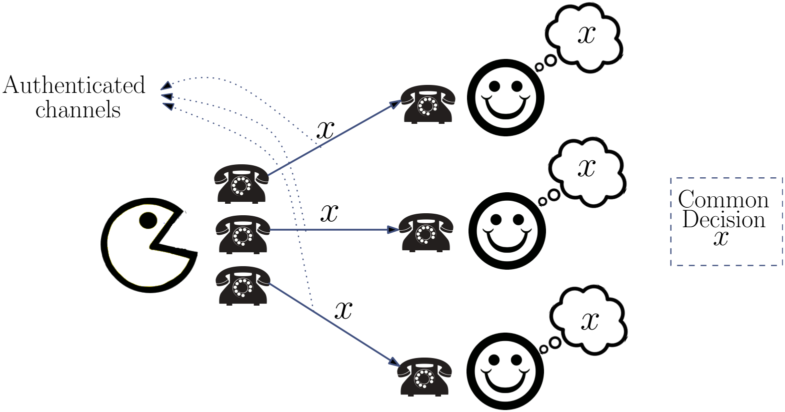


Ideal Broadcast

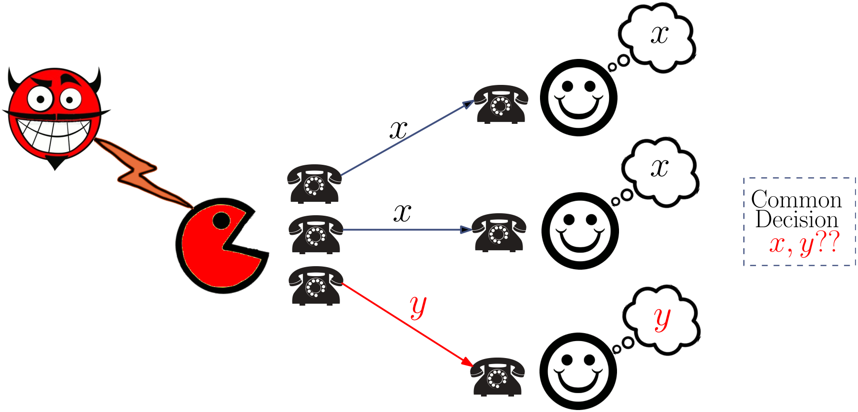


Common
Decision
x

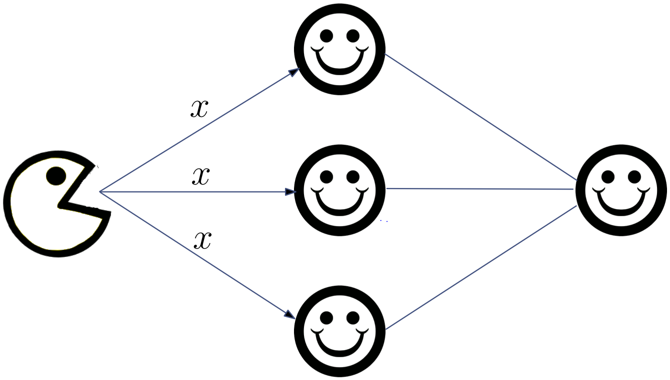
Real Broadcast



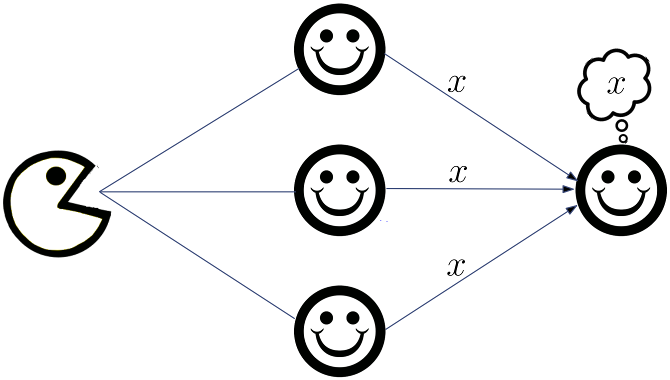
Real Broadcast with Corrupted Dealer



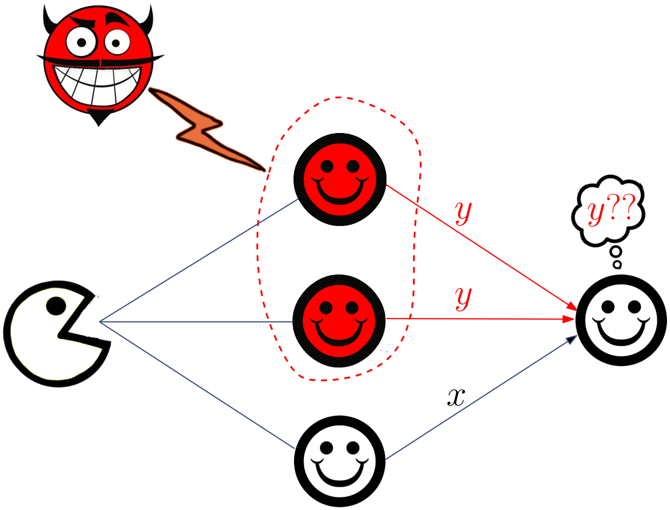
Broadcast in Incomplete Networks



Broadcast in Incomplete Networks II



Broadcast in Incomplete Networks III



Problem Definition

Player Set: $\mathcal{V} = \{v_1, v_2, \cdots, v_n\}$, **Corrupted Players Set:** $T \subseteq \mathcal{V}$.
Each $v \in \mathcal{V}$ finally outputs (decides on) a value **decision**(**v**).

Problem Definition

Player Set: $\mathcal{V} = \{v_1, v_2, \cdots, v_n\}$, **Corrupted Players Set:** $T \subseteq \mathcal{V}$.
Each $v \in \mathcal{V}$ finally outputs (decides on) a value **decision(v)**.

Broadcast (Byzantine Generals)

Dealer $D \in \mathcal{V}$ with **input value** $\mathbf{x_D}$.
Π is a Broadcast protocol for $\mathcal{V}$ if it satisfies:

1. **(Consistency)**
   All honest players decide on the same value *decision*(v).
2. **(Validity)**
   If D is honest then all honest players decide on the dealer's value x_D.

Problem Definition

Player Set: $\mathcal{V} = \{v_1, v_2, \cdots, v_n\}$, **Corrupted Players Set:** $T \subseteq \mathcal{V}$.
Each $v \in \mathcal{V}$ finally outputs (decides on) a value **decision**(**v**).

Broadcast (Byzantine Generals)

Dealer $D \in \mathcal{V}$ with **input value** $\mathbf{x_D}$. Π is a Broadcast protocol for $\mathcal{V}$ if it satisfies:

1. **(Consistency)**
   All honest players decide on the same value *decision*(v).
2. **(Validity)**
   If D is honest then all honest players decide on the dealer's value x_D.

Consensus (Byzantine Agreement)

Every player $v \in \mathcal{V}$ has an **input value** $\mathbf{x_v}$. Π is a Consensus protocol for $\mathcal{V}$ if it satisfies:

1. **(Consistency)**
   All honest players decide on the same value *decision*(v).
2. **(Validity)**
   If all honest players have the same input value x then all honest players decide x.

Adversary Model

Corruption Type

- **Passive:** Obtains all internal data of corrupted players.

Adversary Model

Corruption Type

- **Passive:** Obtains all internal data of corrupted players.
- **Active (Byzantine):** Full control of corrupted players.

Adversary Model

Corruption Type

- **Passive:** Obtains all internal data of corrupted players.
- **Active (Byzantine):** Full control of corrupted players.
- **Fail-Stop (Fault):** Makes corrupted players crash at any time.

Adversary Model

Corruption Type

- **Passive:** Obtains all internal data of corrupted players.
- **Active (Byzantine):** Full control of corrupted players.
- **Fail-Stop (Fault):** Makes corrupted players crash at any time.
- **Static/Adaptive/Mobile**

Adversary Model

Corruption Type

- **Passive:** Obtains all internal data of corrupted players.
- **Active (Byzantine):** Full control of corrupted players.
- **Fail-Stop (Fault):** Makes corrupted players crash at any time.
- **Static/Adaptive/Mobile**

Adversary's Computing Power

- **Unlimited**

Adversary Model

Corruption Type

- **Passive:** Obtains all internal data of corrupted players.
- **Active (Byzantine):** Full control of corrupted players.
- **Fail-Stop (Fault):** Makes corrupted players crash at any time.
- **Static/Adaptive/Mobile**

Adversary's Computing Power

- **Unlimited**
- **Computationally Bounded**
  (to probabilistic polynomial time computations in a security parameter κ).

Adversary Model

Corruption Type

- **Passive:** Obtains all internal data of corrupted players.
- **Active (Byzantine):** Full control of corrupted players.
- **Fail-Stop (Fault):** Makes corrupted players crash at any time.
- **Static/Adaptive/Mobile**

Adversary's Computing Power

- **Unlimited**
- **Computationally Bounded**
  (to probabilistic polynomial time computations in a security parameter κ).

t-**Threshold Adversary:** Can corrupt all player subsets of size at most t.

Adversary Model

Corruption Type

- **Passive:** Obtains all internal data of corrupted players.
- **Active (Byzantine):** Full control of corrupted players.
- **Fail-Stop (Fault):** Makes corrupted players crash at any time.
- **Static/Adaptive/Mobile**

Adversary's Computing Power

- **Unlimited**
- **Computationally Bounded**
  (to probabilistic polynomial time computations in a security parameter κ).

t-**Threshold Adversary:** Can corrupt all player subsets of size at most t.
General Adversary: Characterized by the *adversary structure* $\mathcal{Z}$ which enumerates all possible subsets of corrupted players.

Communication Model

Communication Channels

- **Authenticated**

Communication Model

Communication Channels

- **Authenticated**
- **Synchronous/Asynchronous**
  (No deterministic protocol can achieve asynchronous fault-tolerant Broadcast [FLP85]).

Communication Model

Communication Channels

- **Authenticated**
- **Synchronous/Asynchronous**
  (No deterministic protocol can achieve asynchronous fault-tolerant Broadcast [FLP85]).
- **Complete/Incomplete Communication Networks**

Communication Model

Communication Channels

- **Authenticated**
- **Synchronous/Asynchronous**
  (No deterministic protocol can achieve asynchronous fault-tolerant Broadcast [FLP85]).
- **Complete/Incomplete Communication Networks**

Asynchronous Model: Honest players cannot wait for messages from more than $n - t$ players in each round, where n is the number of players and t the number of corruptions tolerated.

Security

Security is defined with respect to a security parameter κ, allowing an error probability ϵ that is negligible in function of κ.

- **Computational/Cryptographic:** Security against a computationally bounded adversary.

Security

Security is defined with respect to a security parameter κ, allowing an error probability ϵ that is negligible in function of κ.

- **Computational/Cryptographic:** Security against a computationally bounded adversary.
- **Unconditional/Information-Theoretic:** Security against an unlimited adversary.

Security

Security is defined with respect to a security parameter κ, allowing an error probability ϵ that is negligible in function of κ.

- **Computational/Cryptographic:** Security against a computationally bounded adversary.
- **Unconditional/Information-Theoretic:** Security against an unlimited adversary.
- **Perfect Security:** Unconditional Security with zero error probability.

Security

Security is defined with respect to a security parameter κ, allowing an error probability ϵ that is negligible in function of κ.

- **Computational/Cryptographic:** Security against a computationally bounded adversary.
- **Unconditional/Information-Theoretic:** Security against an unlimited adversary.
- **Perfect Security:** Unconditional Security with zero error probability.

Consistently shared data: Typically a PKI.

Efficiency and Resiliency

Communication round: All players in parallel receive the latest messages from their neighbors, perform arbitrary local computation and finally send new messages to their neighbors.

Efficiency and Resiliency

Communication round: All players in parallel receive the latest messages from their neighbors, perform arbitrary local computation and finally send new messages to their neighbors.

Efficiency and Resiliency

We want to optimize

- **Bit/Message Complexity:** Total number of bits/messages sent by all honest players.

Efficiency and Resiliency

Communication round: All players in parallel receive the latest messages from their neighbors, perform arbitrary local computation and finally send new messages to their neighbors.

Efficiency and Resiliency

We want to optimize

- **Bit/Message Complexity:** Total number of bits/messages sent by all honest players.
- **Round Complexity:** Maximum number of rounds required by any honest player.

Efficiency and Resiliency

Communication round: All players in parallel receive the latest messages from their neighbors, perform arbitrary local computation and finally send new messages to their neighbors.

Efficiency and Resiliency

We want to optimize

- **Bit/Message Complexity:** Total number of bits/messages sent by all honest players.
- **Round Complexity:** Maximum number of rounds required by any honest player.
- **Local Computation Complexity:** Maximum over the local computational worst-case complexities of all honest players.

Efficiency and Resiliency

Communication round: All players in parallel receive the latest messages from their neighbors, perform arbitrary local computation and finally send new messages to their neighbors.

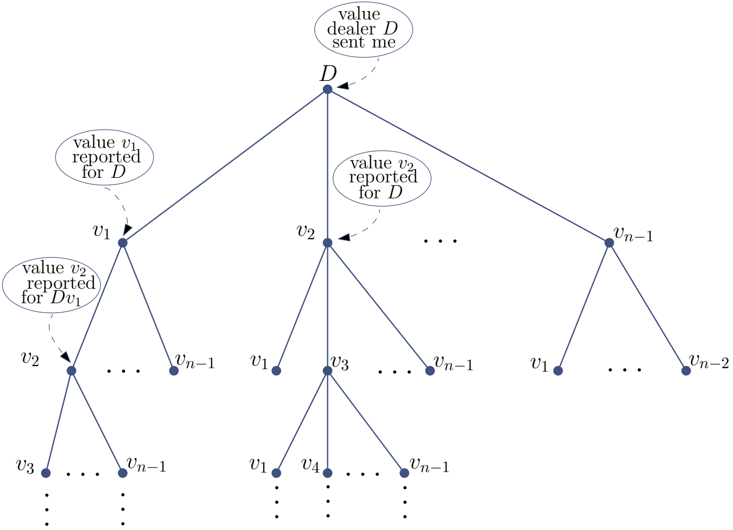
Efficiency and Resiliency

We want to optimize

- **Bit/Message Complexity:** Total number of bits/messages sent by all honest players.
- **Round Complexity:** Maximum number of rounds required by any honest player.
- **Local Computation Complexity:** Maximum over the local computational worst-case complexities of all honest players.
- **Resiliency:** Number of corrupted players t a protocol can tolerate.

Exponential Information Gathering

EIG Tree


value
dealer D
sent me
D
value v_1
reported
for D
value v_2
reported
for D
v_1
v_2
. . .
v_{n-1}
value v_2
reported
for Dv_1
v_2
. . .
v_{n-1}
v_1
v_3
. . .
v_{n-1}
v_1
. . .
v_{n-2}
v_3
. . .
v_{n-1}
v_1
v_4
. . .
v_{n-1}

EIG Algorithm I - Information Gathering

Information Gathering

Round 1

1. Dealer sends its initial value x_D to the $n-1$ other players and decides on x_D.
2. Each v stores value x_D in the root of $tree_v$ ($tree_v(D) := x_D$). A special default value of $\perp$ is stored if the Dealer failed to send a legitimate value in X.

EIG Algorithm I - Information Gathering

Information Gathering

Round 1

1. Dealer sends its initial value x_D to the $n-1$ other players and decides on x_D.
2. Each v stores value x_D in the root of $tree_v$ ($tree_v(D) := x_D$). A special default value of $\perp$ is stored if the Dealer failed to send a legitimate value in X.

Round h, $2 \leq h \leq t+1$

1. Each v broadcasts the leaves of its round $(h-1)$ tree.
2. Every v adds a new level to its tree, storing at node $D \ldots qr$ the value that r claims to have stored in node $D \ldots q$ in its own $tree_r$. Again, $\perp$ is used for inappropriate messages.

EIG Algorithm I - Information Gathering

Information Gathering

Round 1

1. Dealer sends its initial value x_D to the $n-1$ other players and decides on x_D.
2. Each v stores value x_D in the root of $tree_v$ ($tree_v(D) := x_D$). A special default value of $\perp$ is stored if the Dealer failed to send a legitimate value in X.

Round h, $2 \leq h \leq t+1$

1. Each v broadcasts the leaves of its round $(h-1)$ tree.
2. Every v adds a new level to its tree, storing at node $D \ldots qr$ the value that r claims to have stored in node $D \ldots q$ in its own $tree_r$. Again, $\perp$ is used for inappropriate messages.

Intuitively, v stores in node $D \ldots qr$ the value that "r says q says ... the source said".

EIG Algorithm *II* - Data Conversion

After $t+1$ rounds o Information Gathering, each player v computes a the commonly agreed-upon recursive function *resolve*() in order to decide.

Resolve Function

(Recursive majority of descendants of node a)
For all a sequences of $tree_v$:

$$resolve_v(a) = \begin{cases} tree(a) & \text{, if } a \text{ is a leaf;} \\ m & \text{, If } m \text{ is the majority of } resolve \text{ applied} \\ & \quad \text{to the children of } a; \\ \bot & \text{, If } a \text{ is not a leaf and no majority exists.} \end{cases}$$

EIG Algorithm II - Data Conversion

After $t+1$ rounds o Information Gathering, each player v computes a the commonly agreed-upon recursive function *resolve*() in order to decide.

Resolve Function

(Recursive majority of descendants of node a)
For all a sequences of $tree_v$:

$$resolve_v(a) = \begin{cases} tree(a) & \text{, if } a \text{ is a leaf;} \\ m & \text{, If } m \text{ is the majority of } resolve \text{ applied} \\ & \quad \text{to the children of } a; \\ \bot & \text{, If } a \text{ is not a leaf and no majority exists.} \end{cases}$$

Decision

Player v decides on the value $resolve_v(D)$.

Complexity of the EIG Algorithm

Proposition 2.1 (Lamport, Shostak, Pease 1982).

The EIG Algorithm achieves Broadcast in $t + 1$ rounds provided that $n \geq 3t + 1$

Bit Complexity

For any $1 \leq h \leq t + 1$, the h-round EIG tree has $O(n^{h-1})$ leaves, yielding messages of size $O(n^{h-1})$ in round $h + 1$. Thus, BC and LCC grow exponential in t.

Complexity of the EIG Algorithm

Proposition 2.1 (Lamport, Shostak, Pease 1982).

The EIG Algorithm achieves Broadcast in $t+1$ rounds provided that $n \geq 3t+1$

Bit Complexity

For any $1 \leq h \leq t+1$, the h-round EIG tree has $O(n^{h-1})$ leaves, yielding messages of size $O(n^{h-1})$ in round $h+1$. Thus, BC and LCC grow exponential in t.

[GM98]: First $(t+1)$-round fully polynomial, optimal resilience Broadcast protocol.

Complexity of the EIG Algorithm

Proposition 2.1 (Lamport, Shostak, Pease 1982).

The EIG Algorithm achieves Broadcast in $t+1$ rounds provided that $n \geq 3t+1$

Bit Complexity

For any $1 \leq h \leq t+1$, the h-round EIG tree has $O(n^{h-1})$ leaves, yielding messages of size $O(n^{h-1})$ in round $h+1$. Thus, BC and LCC grow exponential in t.

[GM98]: First $(t+1)$-round fully polynomial, optimal resilience Broadcast protocol.
[Coa87]: Binary Consensus can be used to achieve General Consensus with an overhead of 2 extra rounds and $O(n^2 \cdot b)$ extra communication bits, where b : maximum length of a message.

Parameter Lower Bounds

Threshold Adversary Model

t-Threshold Adversary

Can corrupt all player subsets of size at most t.

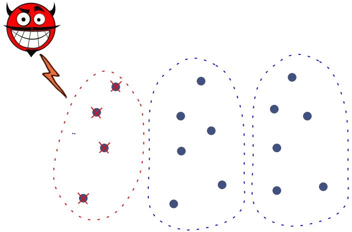
Threshold Adversary Model

t-Threshold Adversary

Can corrupt all player subsets of size at most t.

Complete Networks

Broadcast Necessary and Sufficient Condition:
$t < n/3$ [LSP82]



Threshold Adversary Model

t-Threshold Adversary

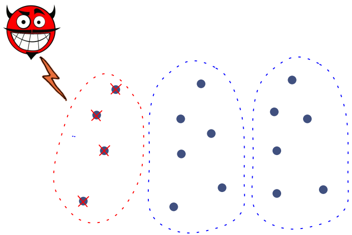
Can corrupt all player subsets of size at most t.

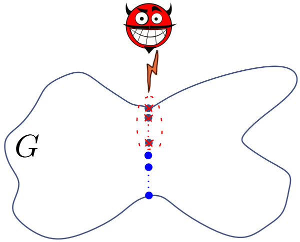
Complete Networks

Broadcast Necessary and Sufficient Condition:
$t < n/3$ [LSP82]

Incomplete Networks

Broadcast Necessary and Sufficient Condition [Dol82]:
$(t < n/3)$ *AND* $(t < conn(G)/2)$





Parameter Lower Bounds -Overview

- Resiliency: $n > 3t$ (Interactive Consistency) [PSL80]
- Bit Complexity: $BC \geq n(t+1)/4$ [DR85]
- Round Complexity: $RC \geq t+1$ [FL82, DS83]
- Connectivity of Network G: $conn(G) > 2t$ [Dol82]

Scenarios

- **State Assignment** C_i**:** An assignment of states to each player.
- **Message assignment** M_i**:** An assignment of a message to each channel.

A Scenario is defined to be an infinite sequence:

$$\sigma = C_0, M_1, C_1, M_2, C_2, \ldots$$

Indistiguishable Scenarios ($\sigma \overset{v}{\sim} \sigma'$)

Two scenarios σ, σ' are indistiguishable with respect to player v, $\sigma \overset{v}{\sim} \sigma'$ if v has the same sequence of states, outgoing and incoming messages ($view(v)$).

Scenarios

- **State Assignment** C_i**:** An assignment of states to each player.
- **Message assignment** M_i**:** An assignment of a message to each channel.

A Scenario is defined to be an infinite sequence:

$$\sigma = C_0, M_1, C_1, M_2, C_2, \ldots$$

Indistiguishable Scenarios ($\sigma \overset{v}{\sim} \sigma'$)

Two scenarios σ, σ' are indistiguishable with respect to player v, $\sigma \overset{v}{\sim} \sigma'$ if v has the same sequence of states, outgoing and incoming messages (*view(v)*). Scenarios σ, σ' may be scenarios of different systems.

Scenarios

- **State Assignment** C_i**:** An assignment of states to each player.
- **Message assignment** M_i**:** An assignment of a message to each channel.

A Scenario is defined to be an infinite sequence:

$$\sigma = C_0, M_1, C_1, M_2, C_2, \ldots$$

Indistiguishable Scenarios ($\sigma \overset{v}{\sim} \sigma'$)

Two scenarios σ, σ' are indistiguishable with respect to player v, $\sigma \overset{v}{\sim} \sigma'$ if v has the same sequence of states, outgoing and incoming messages (*view(v)*). Scenarios σ, σ' may be scenarios of different systems.

decision(**v**): deterministic function of *view*(v) (Perfect Security).

Connectivity Lower Bound ($conn(G) > 2t$)

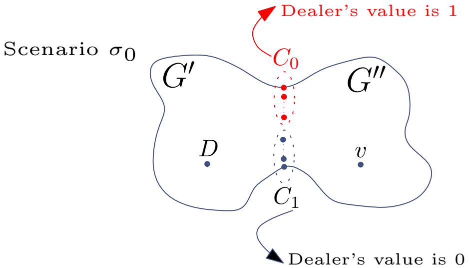
σ_0	σ_1
$x_D = 0$ $T = C_0$	$x_D = 1$ $T = C_1$

Corrupted players C_i of scenario σ_i act like in σ_{1-i}.

Connectivity Lower Bound ($conn(G) > 2t$)

σ_0	σ_1
$x_D = 0$ $T = C_0$	$x_D = 1$ $T = C_1$

Corrupted players C_i of scenario σ_i act like in σ_{1-i}.



Then,

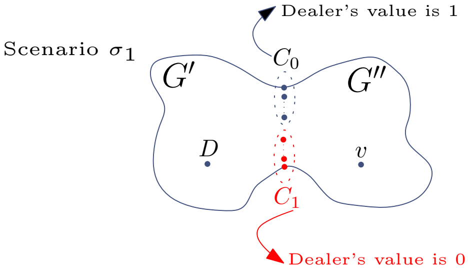
$$\forall v \in G'',\ \sigma_0 \overset{v}{\sim} \sigma_1 \Rightarrow decision_{\sigma_0}(v) = desicion_{\sigma_1}(v)$$

and thus validity is violated. □

Connectivity Lower Bound ($conn(G) > 2t$)

σ_0	σ_1
$x_D = 0$ $T = C_0$	$x_D = 1$ $T = C_1$

Corrupted players C_i of scenario σ_i act like in σ_{1-i}.



Then,

$$\forall v \in G'',\ \sigma_0 \overset{v}{\sim} \sigma_1 \Rightarrow decision_{\sigma_0}(v) = desicion_{\sigma_1}(v)$$

and thus validity is violated. □

Resiliency-Example I

Assume that v_0, v_1, v_2 solve Broadcast in two rounds given that $t = 1$:

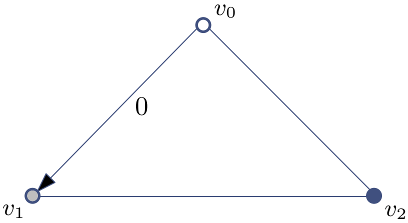
1. The dealer v_0 sends value
2. Each player reports the dealer's value

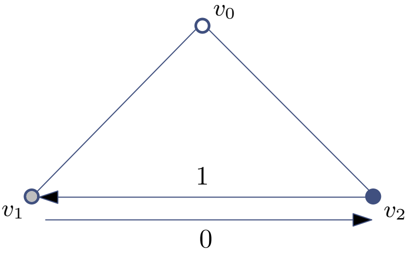
Resiliency-Example I

Assume that v_0, v_1, v_2 solve Broadcast in two rounds given that $t = 1$:

1. The dealer v_0 sends value
2. Each player reports the dealer's value

Honest player v_1, knowing that at most one of the v_0, v_2 is corrupted, has to decide on a value that satisfies both conditions of the Broadcast problem. Consider the following *view*(v_1).





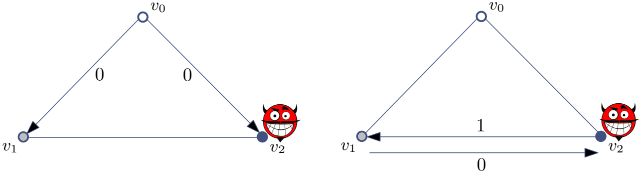
Resiliency-Example II

Two possible scenarios σ_1(corrupted v_2) and σ_2(corrupted v_0) s.t. $\sigma_1 \overset{v_1}{\sim} \sigma_2$ (indistinguishable with respect to v_1):

Resiliency-Example II

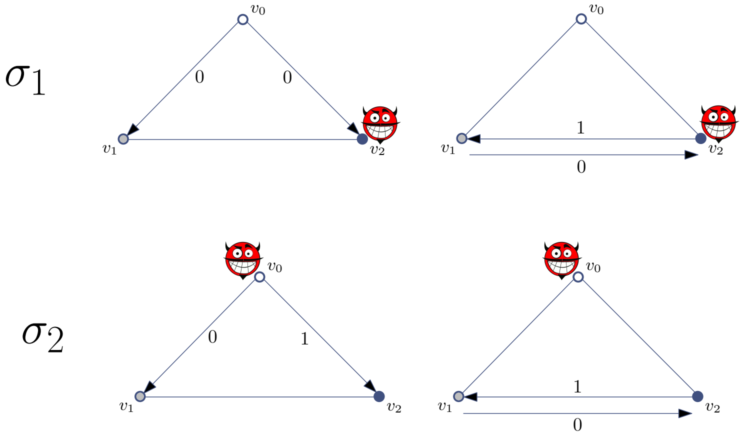
Two possible scenarios σ_1(corrupted v_2) and σ_2(corrupted v_0) s.t. $\sigma_1 \overset{v_1}{\sim} \sigma_2$ (indistinguishable with respect to v_1):

σ_1



Resiliency-Example II

Two possible scenarios σ_1(corrupted v_2) and σ_2(corrupted v_0) s.t. $\sigma_1 \overset{v_1}{\sim} \sigma_2$ (indistinguishable with respect to v_1):



Resiliency-Example III

Impossibility of Broadcast

If $decision(v_1) = 1$ and σ_1 holds, then validity is violated, thus

$$decision(v_1) = 0 \quad (1)$$

Resiliency-Example III

Impossibility of Broadcast

If $decision(v_1) = 1$ and σ_1 holds, then validity is violated, thus

$$decision(v_1) = 0 \quad (1)$$

If σ_2 holds then by symmetry v_2 should decide on 1

$$decision(v_1) = 1 \quad (2)$$

$(1), (2) \Rightarrow$ Consistency is violated.

Resiliency-Example III

Impossibility of Broadcast

If $decision(v_1) = 1$ and σ_1 holds, then validity is violated, thus

$$decision(v_1) = 0 \tag{1}$$

If σ_2 holds then by symmetry v_2 should decide on 1

$$decision(v_1) = 1 \tag{2}$$

$(1), (2) \Rightarrow$ Consistency is violated.

The algorithm uses only two rounds and particular types of messages.

Resiliency Lower Bound I

Lemma 3.1.

Three players cannot solve the Broadcast problem in the presence of one fault ($n = 3$ and $t = 1$).

Resiliency Lower Bound I

Lemma 3.1.

Three players cannot solve the Broadcast problem in the presence of one fault ($n = 3$ and $t = 1$).

Proof. Assume the existence of algorithm $\mathcal{A}$ that achieves Broadcast in system T in the presence of a corrupted player. Construct system H using two copies of T,

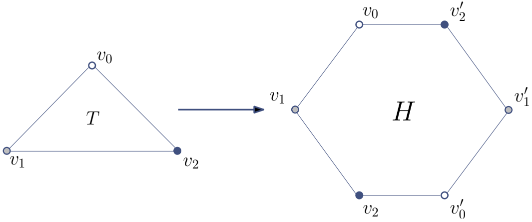


Figure: Identical copy $v_k' = v_{k+3}$ of v_k. Connect $v_{k \bmod 6}$ with $v_{(k+1) \bmod 6}$ and $v_{(k-1) \bmod 6}$

Resiliency Lower Bound II

In H all players run $\mathcal{A}$ and have only local names for their two neighbors.

Claim

For all σ_H scenario of H without adversary and $\forall k \in \{0, \ldots, 5\}$, $\exists \sigma_T$ scenario of T in which $v_{(k+2) \bmod 3}$ is corrupted s.t.

$$\sigma_H \overset{v_k}{\sim} \sigma_T \text{ and } \sigma_H \overset{v_{k+1 \bmod 6}}{\sim} \sigma_T$$

Resiliency Lower Bound II

In H all players run $\mathcal{A}$ and have only local names for their two neighbors.

Claim

For all σ_H scenario of H without adversary and $\forall k \in \{0, \dots, 5\}$, $\exists \sigma_T$ scenario of T in which $v_{(k+2) \bmod 3}$ is corrupted s.t.

$$\sigma_H \overset{v_k}{\sim} \sigma_T \text{ and } \sigma_H \overset{v_{k+1 \bmod 6}}{\sim} \sigma_T$$

For v_k and $v_{k+1 \bmod 6}$, their views are indistinguishable from their views as players $v_{k \bmod 3}$ and $v_{(k+1) \bmod 3}$ in T where the adversary corrupts $v_{(k+2) \bmod 3}$ by simply simulating all the remaining players of H.

Resiliency Lower Bound II

In H all players run $\mathcal{A}$ and have only local names for their two neighbors.

Claim

For all σ_H scenario of H without adversary and $\forall k \in \{0, \ldots, 5\}$, $\exists \sigma_T$ scenario of T in which $v_{(k+2) \bmod 3}$ is corrupted s.t.

$$\sigma_H \overset{v_k}{\sim} \sigma_T \text{ and } \sigma_H \overset{v_{k+1 \bmod 6}}{\sim} \sigma_T$$

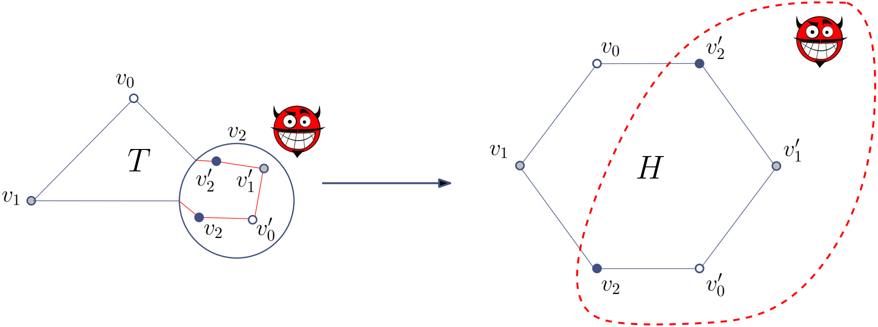
For v_k and $v_{k+1 \bmod 6}$, their views are indistinguishable from their views as players $v_{k \bmod 3}$ and $v_{(k+1) \bmod 3}$ in T where the adversary corrupts $v_{(k+2) \bmod 3}$ by simply simulating all the remaining players of H.

Thus, every such pair executes $\mathcal{A}$ in H without adversary and achieves Broadcast. If H exhibits contradictory behavior then $\mathcal{A}$ cannot exist.

Resiliency Lower Bound III

Example.

The adversary corrupts v_2 in T by simulating the subsystem of H encircled



Resiliency Lower Bound IV

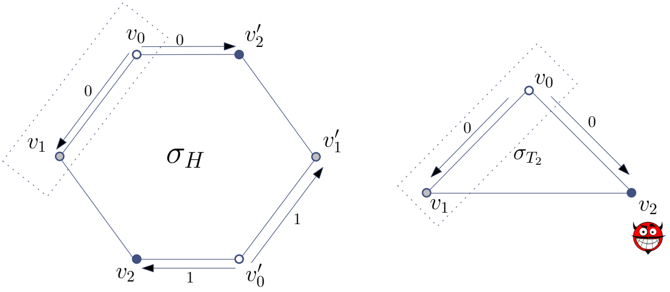
Contradictory behavior of H

H involves two players v_0, v_0' of the type corresponding to the Dealer. Suppose they have inputs $x_0 \in \{0,1\}$ and $x_0' = 1 - x_0$ respectively.

Resiliency Lower Bound IV

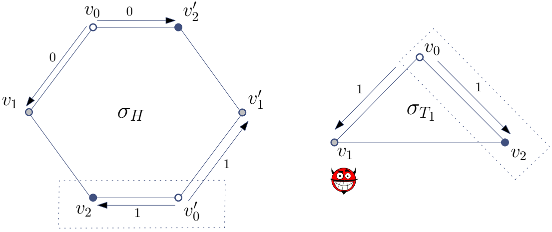
Contradictory behavior of H

H involves two players v_0, v_0' of the type corresponding to the Dealer. Suppose they have inputs $x_0 \in \{0,1\}$ and $x_0' = 1 - x_0$ respectively.

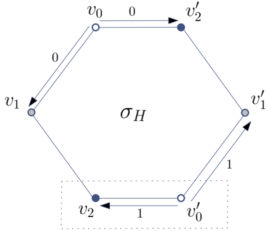


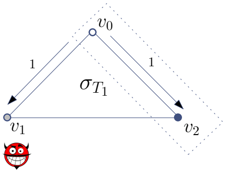
$$\sigma_H \overset{v_0}{\sim} \sigma_{T_2} \text{ and } \sigma_H \overset{v_1}{\sim} \sigma_{T_2} \Rightarrow \mathit{decision}(v_1) = 0 \qquad (1)$$

Resiliency Lower Bound V



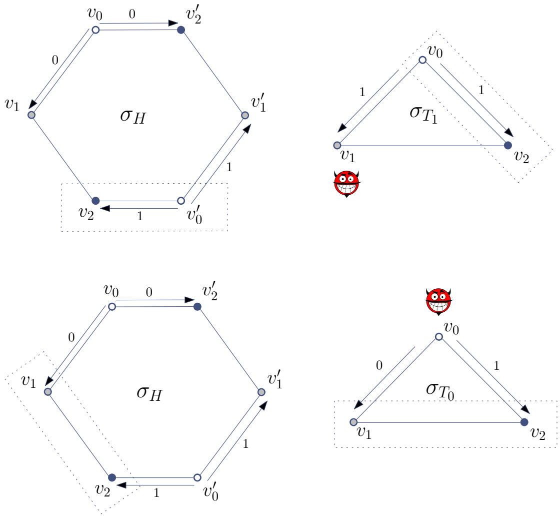
Resiliency Lower Bound V





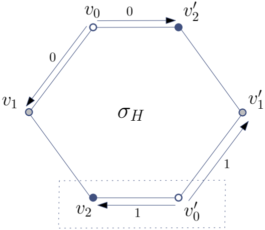
$$\sigma_H \overset{v_0'}{\sim} \sigma_{T_1} \text{ and } \sigma_H \overset{v_2}{\sim} \sigma_{T_1} \Rightarrow$$
$$\Rightarrow decision(v_2) = 1 \qquad (2)$$

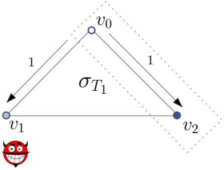
Resiliency Lower Bound V



$$\sigma_H \overset{v_0'}{\sim} \sigma_{T_1} \text{ and } \sigma_H \overset{v_2}{\sim} \sigma_{T_1} \Rightarrow$$
$$\Rightarrow \textit{decision}(v_2) = 1 \qquad (2)$$

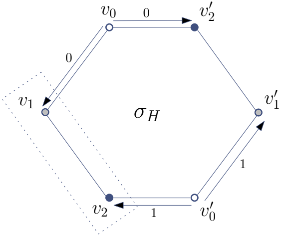
Resiliency Lower Bound V

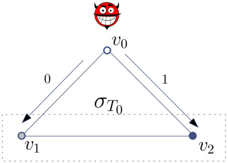




$$\sigma_H \overset{v_0'}{\sim} \sigma_{T_1} \text{ and } \sigma_H \overset{v_2}{\sim} \sigma_{T_1} \Rightarrow$$

$$\Rightarrow \mathit{decision}(v_2) = 1 \qquad (2)$$

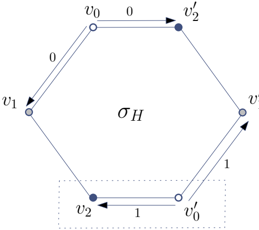


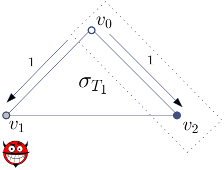


$$\sigma_H \overset{v_1}{\sim} \sigma_{T_0} \text{ and } \sigma_H \overset{v_2}{\sim} \sigma_{T_0} \Rightarrow$$

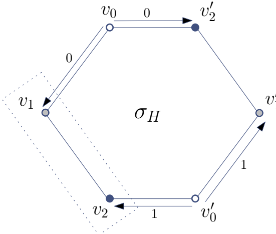
$$\Rightarrow \mathit{decision}(v_1) = \mathit{decision}(v_2) \qquad (3)$$

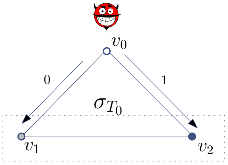
Resiliency Lower Bound V





$$\sigma_H \overset{v_0'}{\sim} \sigma_{T_1} \text{ and } \sigma_H \overset{v_2}{\sim} \sigma_{T_1} \Rightarrow$$
$$\Rightarrow decision(v_2) = 1 \qquad (2)$$





$$\sigma_H \overset{v_1}{\sim} \sigma_{T_0} \text{ and } \sigma_H \overset{v_2}{\sim} \sigma_{T_0} \Rightarrow$$
$$\Rightarrow decision(v_1) = decision(v_2) \qquad (3)$$

Relations (1), (2) and (3) yield a contradiction. □

Resiliency Lower Bound VI

Theorem 3.2.

There is no solution to the Broadcast problem for n players in the presence of t corrupted players, if $3 \leq n \leq 3t$

Resiliency Lower Bound VI

Theorem 3.2.

There is no solution to the Broadcast problem for n players in the presence of t corrupted players, if $3 \leq n \leq 3t$

Proof.

Idea: Assume Broadcast protocol $\mathcal{A}$ with dealer v_0 for $|\mathcal{V}| = n, |T| \geq n/3$.
Transform $\mathcal{A}$ into B Broadcast protocol for players v_0, v_1, v_2 and $|T| = 1$.

Resiliency Lower Bound VI

Theorem 3.2.

There is no solution to the Broadcast problem for n players in the presence of t corrupted players, if $3 \leq n \leq 3t$

Proof.

Idea: Assume Broadcast protocol $\mathcal{A}$ with dealer v_0 for $|\mathcal{V}| = n, |T| \geq n/3$. Transform $\mathcal{A}$ into B Broadcast protocol for players v_0, v_1, v_2 and $|T| = 1$. Partition $\mathcal{V}_0 \cup \mathcal{V}_1 \cup \mathcal{V}_2 = \mathcal{V}$ s.t. $\forall i$, $1 \leq |\mathcal{V}_i| \leq t$. We let each v_i simulate every $v \in \mathcal{V}_i$ (messages and computation steps).

Resiliency Lower Bound VI

Theorem 3.2.

There is no solution to the Broadcast problem for n players in the presence of t corrupted players, if $3 \leq n \leq 3t$

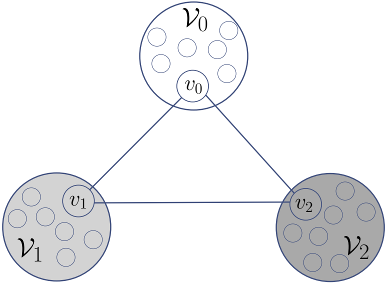
Proof.

Idea: Assume Broadcast protocol $\mathcal{A}$ with dealer v_0 for $|\mathcal{V}| = n, |T| \geq n/3$.
Transform $\mathcal{A}$ into B Broadcast protocol for players v_0, v_1, v_2 and $|T| = 1$.
Partition $\mathcal{V}_0 \cup \mathcal{V}_1 \cup \mathcal{V}_2 = \mathcal{V}$ s.t. $\forall i$, $1 \leq |\mathcal{V}_i| \leq t$. We let each v_i simulate every $v \in \mathcal{V}_i$ (messages and computation steps).

Protocol $\mathcal{B}$

Player v_0: dealer in protocol $\mathcal{B}$.
If in $\mathcal{A}$: $v \in \mathcal{V}_i$ sends m to $u \in \mathcal{V}_j$, $i \neq j$, then
$\mathcal{B}$: v_i sends m to v_j along with the identities of v, u.
If in $\mathcal{A}$: $v \in \mathcal{V}_i$ decides on m, then
$\mathcal{B}$: v_i decides on the value m. (If there are multiple values chooses one)

Resiliency Lower Bound VII



For any execution a of $\mathcal{B}$ with $T_{\mathcal{B}} = v_j$.

Let a' be the simulated execution of $\mathcal{A}$, with $T_{\mathcal{A}} = \mathcal{V}_j$ $(|T_{\mathcal{A}}| \leq t)$.

Validity: From Validity in $\mathcal{A}$.
Consistency: From Consistency in $\mathcal{A}$.

□

Bit Complexity

Theorem 3.3 (Dolev, Reischuk 1985).

Every Broadcast protocol which handles up to t corruptions ($t < n - 1$), requires at least $n(t+1)/4$ messages to be sent.

Bit Complexity

Theorem 3.3 (Dolev, Reischuk 1985).

Every Broadcast protocol which handles up to t corruptions ($t < n - 1$), requires at least $n(t+1)/4$ messages to be sent.

Proof.

Assume scenarios: σ_0 with honest dealer D and $x_D = 0$
σ_1 with honest dealer D and $x_D = 1$, and let

Bit Complexity

Theorem 3.3 (Dolev, Reischuk 1985).

Every Broadcast protocol which handles up to t corruptions ($t < n-1$), requires at least $n(t+1)/4$ messages to be sent.

Proof.

Assume scenarios: σ_0 with honest dealer D and $x_D = 0$
σ_1 with honest dealer D and $x_D = 1$, and let

$A(v) = \{u \in \mathcal{V} \mid \exists\ i \in \mathbb{N},\ \exists j \in \{0,1\}\ s.t.\ \sigma_j(v,u,i) \neq \varnothing \text{ or } \sigma_j(u,v,i) \neq \varnothing\}$
(Players that communicate with v in at least one scenario).

Bit Complexity

Theorem 3.3 (Dolev, Reischuk 1985).

Every Broadcast protocol which handles up to t corruptions ($t < n - 1$), requires at least $n(t+1)/4$ messages to be sent.

Proof.

Assume scenarios: σ_0 with honest dealer D and $x_D = 0$
σ_1 with honest dealer D and $x_D = 1$, and let

$A(v) = \{u \in \mathcal{V} \mid \exists\ i \in \mathbb{N},\ \exists j \in \{0,1\}\ s.t.\ \sigma_j(v,u,i) \neq \emptyset \text{ or } \sigma_j(u,v,i) \neq \emptyset\}$
(Players that communicate with v in at least one scenario).

Let $\exists v \in \mathcal{V},\ s.t.\ |A(v)| < t + 1.$ Consider scenario
σ': The scenario σ_1 with every $u \in A(v)$ behaving towards v as in σ_0.

Bit Complexity

Theorem 3.3 (Dolev, Reischuk 1985).

Every Broadcast protocol which handles up to t corruptions ($t < n - 1$), requires at least $n(t+1)/4$ messages to be sent.

Proof.

Assume scenarios: σ_0 with honest dealer D and $x_D = 0$
σ_1 with honest dealer D and $x_D = 1$, and let

$A(v) = \{u \in \mathcal{V} \mid \exists\ i \in \mathbb{N},\ \exists j \in \{0,1\}\ s.t.\ \sigma_j(v,u,i) \neq \varnothing\ or\ \sigma_j(u,v,i) \neq \varnothing\}$
(Players that communicate with v in at least one scenario).

Let $\exists v \in \mathcal{V},\ s.t.\ |A(v)| < t+1$. Consider scenario
σ': The scenario σ_1 with every $u \in A(v)$ behaving towards v as in σ_0.

$$\sigma' \overset{v}{\sim} \sigma_0 \Rightarrow decision_v(\sigma') = 0,\ and$$
$$\sigma' \overset{u}{\sim} \sigma_1 \Rightarrow decision_u(\sigma') = 1,\quad \forall u \in \{\mathcal{H} \setminus \{v\}\}$$

Bit Complexity

Theorem 3.3 (Dolev, Reischuk 1985).

Every Broadcast protocol which handles up to t corruptions ($t < n - 1$), requires at least $n(t+1)/4$ messages to be sent.

Proof.

Assume scenarios: σ_0 with honest dealer D and $x_D = 0$
σ_1 with honest dealer D and $x_D = 1$, and let

$A(v) = \{u \in \mathcal{V} \mid \exists\ i \in \mathbb{N},\ \exists j \in \{0,1\}\ s.t.\ \sigma_j(v,u,i) \neq \varnothing \text{ or } \sigma_j(u,v,i) \neq \varnothing\}$
(Players that communicate with v in at least one scenario).

Let $\exists v \in \mathcal{V},\ s.t.\ |A(v)| < t + 1.$ Consider scenario
σ': The scenario σ_1 with every $u \in A(v)$ behaving towards v as in σ_0.

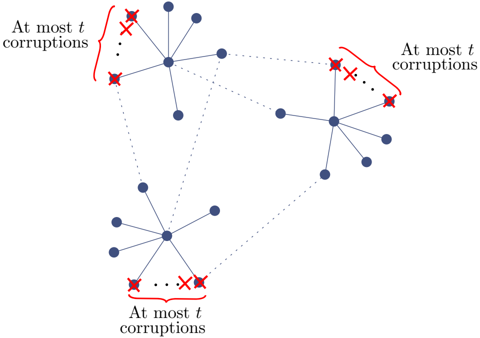
$$\sigma' \overset{v}{\sim} \sigma_0 \Rightarrow decision_v(\sigma') = 0,\ and$$
$$\sigma' \overset{u}{\sim} \sigma_1 \Rightarrow decision_u(\sigma') = 1,\quad \forall u \in \{\mathcal{H} \setminus \{v\}\}$$

Hence $|A(v)| \geq t + 1 \Rightarrow n(t+1)/2$ overall messages in both scenarios
$\Rightarrow$ At least $n(t+1)/4$ messages in σ_0 or σ_1. □

Locally Bounded Adversary

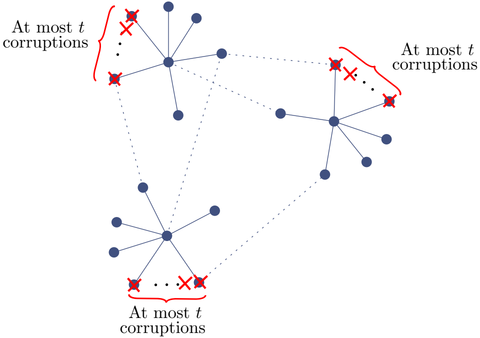
Locally Bounded Adversary Model

t-Locally Bounded Adversary [Koo04]: Can corrupt at most t players in each neighborhood.



Locally Bounded Adversary Model

t-Locally Bounded Adversary [Koo04]: Can corrupt at most t players in each neighborhood.

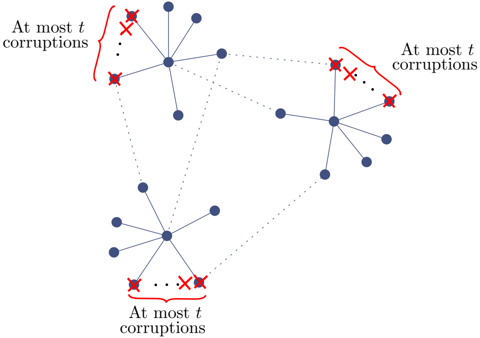


Assumptions

- Honest Dealer
- Incomplete Network
- Byzantine Adversary
- Perfect Security
- Synchronous Channels
- Authenticated Channels

Locally Bounded Adversary Model

t-Locally Bounded Adversary [Koo04]: Can corrupt at most *t* players in each neighborhood.



Assumptions

- Honest Dealer
- Incomplete Network
- Byzantine Adversary
- Perfect Security
- Synchronous Channels
- Authenticated Channels

Results for Broadcast with honest dealer directly apply in the wireless *Ad Hoc* model due to consistency of local Broadcasts.

Broadcast With Locally Bounded Adversary

Topological restrictions on the adversary's corruption capacity

- Tolerate more corruptions
- Local restrictions → local criteria for *Ad Hoc* network Broadcast.

Broadcast With Locally Bounded Adversary

Topological restrictions on the adversary's corruption capacity

- Tolerate more corruptions
- Local restrictions → local criteria for *Ad Hoc* network Broadcast.

Definitions

- **t-Local Set**: A set W, s.t. $|W \cap \mathcal{N}(v)| \leq t, \ \forall v \in \mathcal{V}$.

Broadcast With Locally Bounded Adversary

Topological restrictions on the adversary's corruption capacity

- Tolerate more corruptions
- Local restrictions → local criteria for *Ad Hoc* network Broadcast.

Definitions

- **t-Local Set**: A set W, s.t. $|W \cap \mathcal{N}(v)| \leq t, \ \forall v \in \mathcal{V}$.
- **t-Locally Safe Algorithm**: Never causes a node to decide on an incorrect message under any t-local corruption set.

Broadcast With Locally Bounded Adversary

Topological restrictions on the adversary's corruption capacity

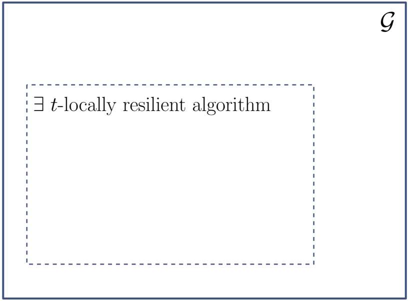
- Tolerate more corruptions
- Local restrictions → local criteria for *Ad Hoc* network Broadcast.

Definitions

- **t-Local Set**: A set W, s.t. $|W \cap \mathcal{N}(v)| \leq t, \ \forall v \in \mathcal{V}$.
- **t-Locally Safe Algorithm**: Never causes a node to decide on an incorrect message under any t-local corruption set.
- **t-Locally Resilient Algorithm**: Achieves Broadcast under any t-local set of corrupted players (locally tolerates t-corruptions).

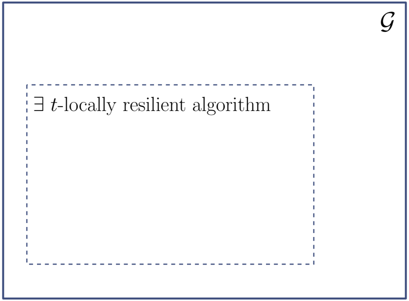
Main Question

Define the class of graphs where achieving Broadcast in the t-locally bounded model is possible (for a given $t \in \mathbb{N}$).



Main Question

Define the class of graphs where achieving Broadcast in the t-locally bounded model is possible (for a given $t \in \mathbb{N}$).



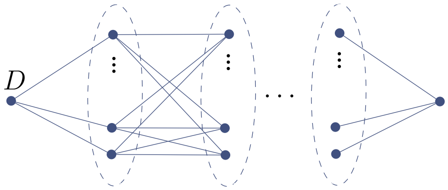
Main Question Rephrased

- For a given graph and dealer determine the maximum number of corruptions $\mathbf{t_{max}}$ that can be locally tolerated.
- To this end: Introduce graph parameters to bound $\mathbf{t_{max}}$.

The Certified Propagation Algorithm

Certified Propagation Algorithm (CPA) [Koo04]

1. The dealer D sends its initial value x_D all of its neighbors, decides on x_D and terminates.
2. If a node decides on a value through a **decision rule**, it sends it to all its neighbors and terminates.



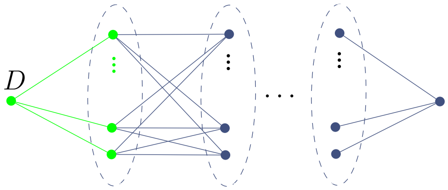
The Certified Propagation Algorithm

Certified Propagation Algorithm (CPA) [Koo04]

1. The dealer D sends its initial value x_D all of its neighbors, decides on x_D and terminates.
2. If a node decides on a value through a **decision rule**, it sends it to all its neighbors and terminates.

Decision Rules

(i) (Neighbors of the dealer) Upon receiving the message x_D from the dealer, decide on x_D.



The Certified Propagation Algorithm

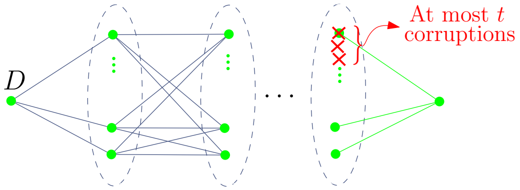
Certified Propagation Algorithm (CPA) [Koo04]

1. The dealer D sends its initial value x_D all of its neighbors, decides on x_D and terminates.
2. If a node decides on a value through a **decision rule**, it sends it to all its neighbors and terminates.

Decision Rules

(i) (Neighbors of the dealer) Upon receiving the message x_D from the dealer, decide on x_D.

(ii) Upon receiving message m from $t+1$ distinct neighbors, decide on m.



The Certified Propagation Algorithm

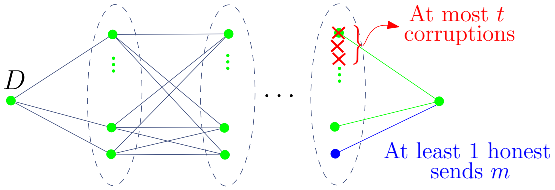
Certified Propagation Algorithm (CPA) [Koo04]

1. The dealer D sends its initial value x_D all of its neighbors, decides on x_D and terminates.
2. If a node decides on a value through a **decision rule**, it sends it to all its neighbors and terminates.

Decision Rules

(i) (Neighbors of the dealer) Upon receiving the message x_D from the dealer, decide on x_D.

(ii) Upon receiving message m from $t + 1$ distinct neighbors, decide on m.



The Certified Propagation Algorithm

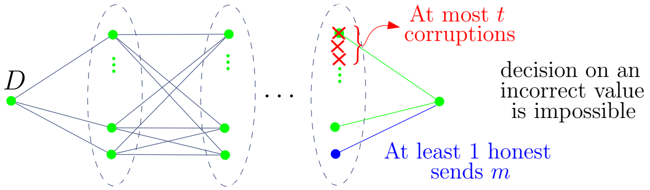
Certified Propagation Algorithm (CPA) [Koo04]

1. The dealer D sends its initial value x_D all of its neighbors, decides on x_D and terminates.
2. If a node decides on a value through a **decision rule**, it sends it to all its neighbors and terminates.

Decision Rules

(i) (Neighbors of the dealer) Upon receiving the message x_D from the dealer, decide on x_D.

(ii) Upon receiving message m from $t+1$ distinct neighbors, decide on m.



Resilience of CPA

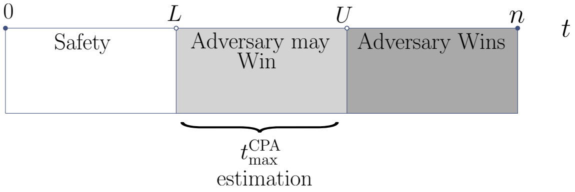
Definition 4.1 (Max CPA Resilience).

$t_{\max}^{\mathrm{CPA}}(G, D)$: The maximum number of corruptions that can be locally tolerated by CPA, for a G and dealer D.

Resilience of CPA

Definition 4.1 (Max CPA Resilience).

$t_{\max}^{\mathrm{CPA}}(G, D)$: The maximum number of corruptions that can be locally tolerated by CPA, for a G and dealer D.



A first goal: Approximate the value $t_{\max}^{\mathrm{CPA}}$ by computing upper and lower bounds.

A Lower Bound on $t_{\max}^{\mathrm{CPA}}$

Graph parameter of [PP05]

For a graph G and dealer D,
$\mathcal{X}(G, D)$: Maximum integer x s.t. every node v has at least x neighbors closer to D than v is.

A Lower Bound on $t_{\max}^{\mathrm{CPA}}$

Graph parameter of [PP05]

For a graph G and dealer D,
$\mathcal{X}(G, D)$: Maximum integer x s.t. every node v has at least x neighbors closer to D than v is.

Theorem 1 (Sufficient Condition [PP05]).

For every graph G, dealer D and integer $t < \mathcal{X}(G, D)/2$, CPA is t-locally resilient

A Lower Bound on $t_{\max}^{\mathrm{CPA}}$

Graph parameter of [PP05]

For a graph G and dealer D,
$\mathcal{X}(G, D)$: Maximum integer x s.t. every node v has at least x neighbors closer to D than v is.

Theorem 1 (Sufficient Condition [PP05]).

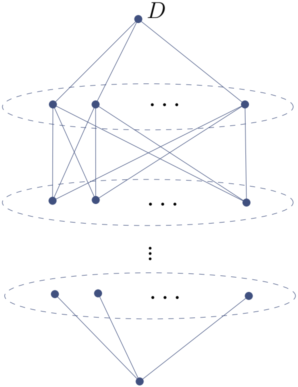
For every graph G, dealer D and integer $t < \mathcal{X}(G, D)/2$, CPA is t-locally resilient $\Rightarrow t_{\max}^{\mathrm{CPA}} \geq \lceil \mathcal{X}/2 \rceil - 1$

Proof Sketch

Observation

The criterion implies a **level ordering** of the nodes w.r.t. the distance from the dealer. In a synchronous setting, information is propagated one level in each round.

$t < \mathcal{X}(G,D)/2 \Rightarrow \mathcal{X}(G,D) \geq 2t+1$

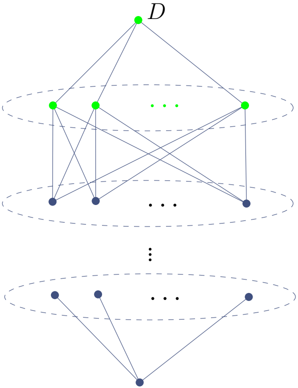


Proof Sketch - CPA Round 1

Observation

The criterion implies a **level ordering** of the nodes w.r.t. the distance from the dealer. In a synchronous setting, information is propagated one level in each round.

$t < \mathcal{X}(G, D)/2 \Rightarrow \mathcal{X}(G, D) \geq 2t + 1$

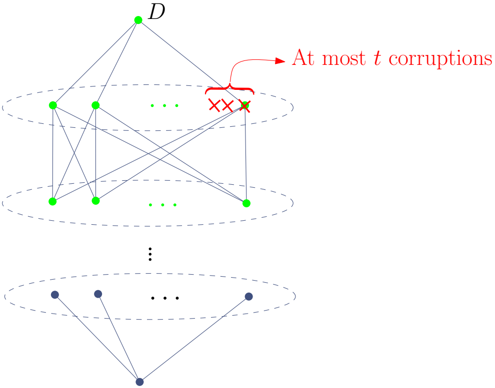


Proof Sketch - CPA Round 2

Observation

The criterion implies a **level ordering** of the nodes w.r.t. the distance from the dealer. In a synchronous setting, information is propagated one level in each round.

$t < \mathcal{X}(G, D)/2 \Rightarrow \mathcal{X}(G, D) \geq 2t + 1$

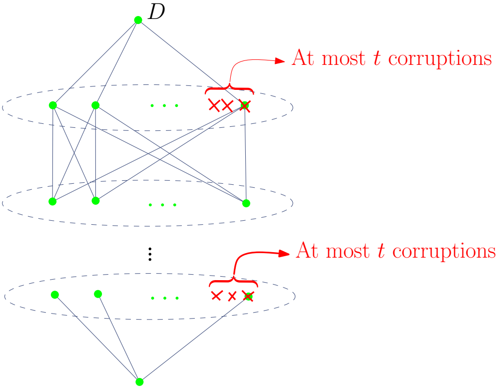


Proof Sketch - CPA Round k

Observation

The criterion implies a **level ordering** of the nodes w.r.t. the distance from the dealer. In a synchronous setting, information is propagated one level in each round.

$t < \mathcal{X}(G,D)/2 \Rightarrow \mathcal{X}(G,D) \geq 2t+1$

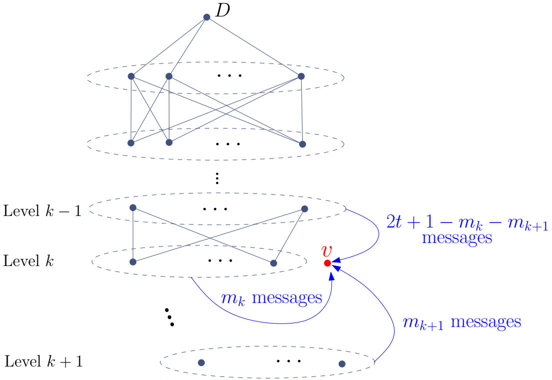


Non-Tightness of the Lower Bound

Condition $t < \mathcal{X}(G, D)/2$ is not necessary for CPA.

Non-Tightness of the Lower Bound

Condition $t < \mathcal{X}(G, D)/2$ is not necessary for CPA.



Node v with $distance(v, D) = k$ may collect $t + 1$ identical values from decided neighbors in distance k and $k + 1$ as well.

A Better Topological Parameter for CPA

Condition of [PP05]

A player will decide if he has at least $2t + 1$ decided neighbors in smaller distance from the dealer than he is.

A Better Topological Parameter for CPA

New Condition

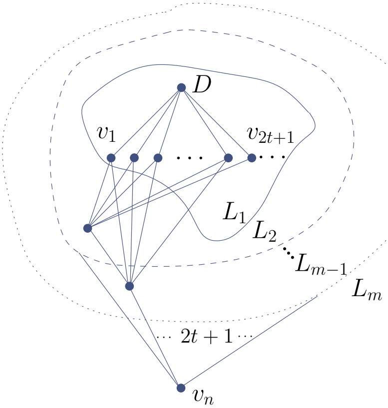
A player will decide if he has at least $2t+1$ decided neighbors ~~in smaller distance from the dealer than he is~~.

A Better Topological Parameter for CPA

New Condition

A player will decide if he has at least $2t+1$ decided neighbors ~~in smaller distance from the dealer than he is~~.

Generalized Notion of Levels



A New Parameter for Bounding $t_{\max}^{\mathrm{CPA}}$

Definitions [LPS13]

For a graph $G = (V, E)$ with dealer-node D,

Minimum k-Level Ordering $\mathcal{L}_k(G, D)$:
A partition $V = \bigcup_{i=1}^{m} L_i, m \in \mathbb{N}$, s.t. $L_1 = \mathcal{N}(D)$ and each level L_i contains all the nodes that have at least k neighbors in the union of previous levels.

A New Parameter for Bounding $t_{\max}^{\mathrm{CPA}}$

Definitions [LPS13]

For a graph $G = (V, E)$ with dealer-node D,

Minimum k-Level Ordering $\mathcal{L}_k(G, D)$:
A partition $V = \bigcup_{i=1}^{m} L_i, m \in \mathbb{N}$, s.t. $L_1 = \mathcal{N}(D)$ and each level L_i contains all the nodes that have at least k neighbors in the union of previous levels.

$\mathcal{K}(G, D) \stackrel{def.}{=} \max\{k \in \mathbb{N} \mid \exists$ Minimum k-Level Ordering $\mathcal{L}_k(G, D)\}$

Lower Bound on $t_{\max}^{\mathrm{CPA}}$

Theorem 5.1 (Sufficient Condition).

For every graph G, dealer D and $t \in \mathbb{N}$, if $t < \mathcal{K}(G, D)/2$ then CPA is t-locally resilient.

Lower Bound on $t_{\max}^{\mathrm{CPA}}$

Theorem 5.1 (Sufficient Condition).

For every graph G, dealer D and $t \in \mathbb{N}$, if $t < \mathcal{K}(G, D)/2$ then CPA is t-locally resilient.

$\Rightarrow t_{\max}^{\mathrm{CPA}} \geq \lceil \mathcal{K}(G, D)/2 \rceil - 1$

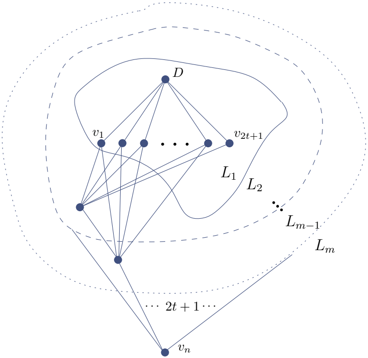
Lower Bound on $t_{\max}^{\mathrm{CPA}}$

Theorem 5.1 (Sufficient Condition).

For every graph G, dealer D and $t \in \mathbb{N}$, if $t < \mathcal{K}(G, D)/2$ then CPA is t-locally resilient. $\Rightarrow t_{\max}^{\mathrm{CPA}} \geq \lceil \mathcal{K}(G, D)/2 \rceil - 1$

Proof Sketch.

$\exists \mathcal{L}_k(G, D)$ with $k \geq 2t + 1$.



Lower Bound on $t_{\max}^{\mathrm{CPA}}$

Theorem 5.1 (Sufficient Condition).

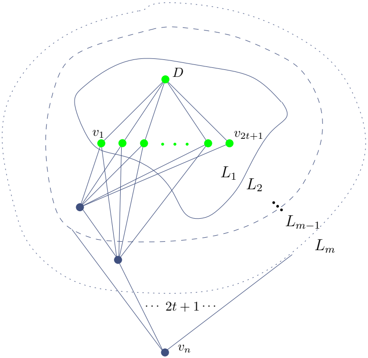
For every graph G, dealer D and $t \in \mathbb{N}$, if $t < \mathcal{K}(G, D)/2$ then CPA is t-locally resilient. $\Rightarrow t_{\max}^{\mathrm{CPA}} \geq \lceil \mathcal{K}(G, D)/2 \rceil - 1$

Proof Sketch.

$\exists \mathcal{L}_k(G, D)$ with $k \geq 2t + 1$.

Decided

Round 1: $L_1 = \mathcal{N}(D)$



Lower Bound on $t_{\max}^{\mathrm{CPA}}$

Theorem 5.1 (Sufficient Condition).

For every graph G, dealer D and $t \in \mathbb{N}$, if $t < \mathcal{K}(G,D)/2$ then CPA is t-locally resilient. $\Rightarrow t_{\max}^{\mathrm{CPA}} \geq \lceil \mathcal{K}(G,D)/2 \rceil - 1$

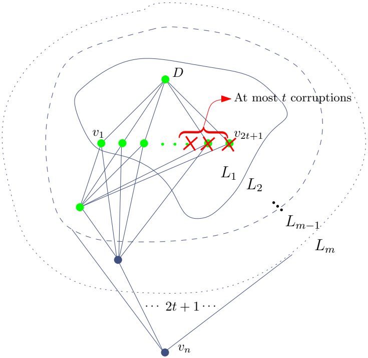
Proof Sketch.

$\exists \mathcal{L}_k(G,D)$ with $k \geq 2t+1$.

Decided

Round 1: $L_1 = \mathcal{N}(D)$

Round 2: $L_1 \cup L_2$

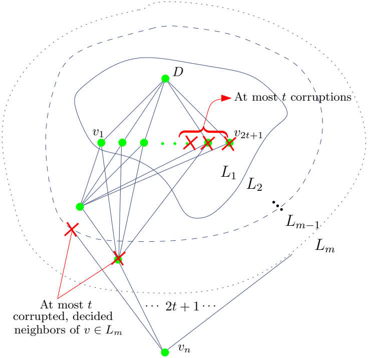


Lower Bound on $t_{\max}^{\mathrm{CPA}}$

Theorem 5.1 (Sufficient Condition).

For every graph G, dealer D and $t \in \mathbb{N}$, if $t < \mathcal{K}(G,D)/2$ then CPA is t-locally resilient. $\Rightarrow t_{\max}^{\mathrm{CPA}} \geq \lceil \mathcal{K}(G,D)/2 \rceil - 1$

Proof Sketch.
$\exists \mathcal{L}_k(G,D)$ with $k \geq 2t+1$.



Decided

Round 1: $L_1 = \mathcal{N}(D)$
Round 2: $L_1 \bigcup L_2$
⋮
Round m: $\bigcup_{j=1}^{m} L_j = \mathcal{V}$

An equivalent Parameter [IS10]

Observation

Parameter $\mathcal{K}(G, D)$ equals $\widetilde{\mathcal{X}}(G, D)$ of [IS10], which is defined using different kind of orderings.

Definition of $\mathcal{K}(G, D)$ implies improved complexity, namely,

$$\text{[IS10]: } O(E \cdot V)$$

$$\mathcal{K}(G, D)\text{: } O(E \log \delta)$$

where $\delta = \min_{v \in \mathcal{V} \setminus \mathcal{N}(D)} \ deg(v)$.

Non-Tightness of the Lower Bound

Proposition 5.2.

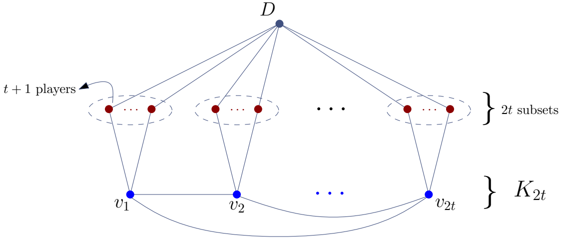
There exists a family of instances, s.t. CPA is $(\mathcal{K}(G,D)-1)$*-locally resilient.*

Non-Tightness of the Lower Bound

Proposition 5.2.

There exists a family of instances, s.t. CPA is $(\mathcal{K}(G,D)-1)$-locally resilient.

$\mathcal{K}(G,D) = t+1$

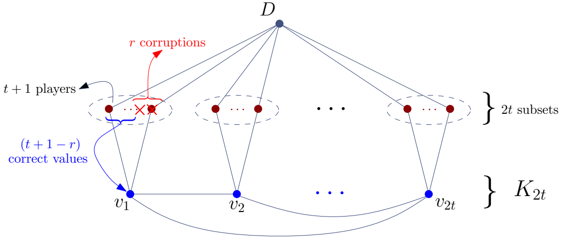


Non-Tightness of the Lower Bound

Proposition 5.2.

There exists a family of instances, s.t. CPA is $(\mathcal{K}(G,D)-1)$-locally resilient.

$\mathcal{K}(G,D) = t+1$



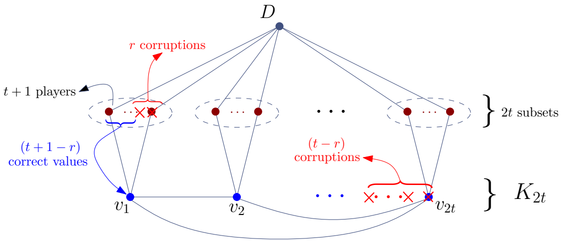
Proof Sketch. Due to trade-off of corruptions in the interconnected neighborhoods, each player receives at least $t+1$ correct messages, thus CPA is t-locally resilient.

Non-Tightness of the Lower Bound

Proposition 5.2.

There exists a family of instances, s.t. CPA is $(\mathcal{K}(G, D) - 1)$-locally resilient.

$\mathcal{K}(G, D) = t + 1$



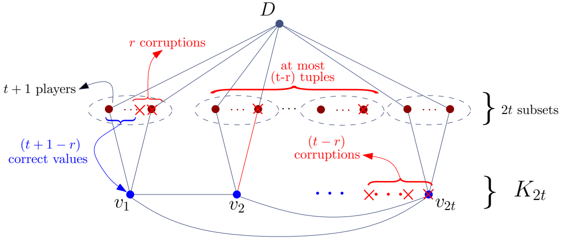
Proof Sketch. Due to trade-off of corruptions in the interconnected neighborhoods, each player receives at least $t + 1$ correct messages, thus CPA is t-locally resilient.

Non-Tightness of the Lower Bound

Proposition 5.2.

There exists a family of instances, s.t. CPA is $(\mathcal{K}(G,D)-1)$-locally resilient.

$\mathcal{K}(G,D) = t+1$



Proof Sketch. Due to trade-off of corruptions in the interconnected neighborhoods, each player receives at least $t+1$ correct messages, thus CPA is t-locally resilient.

Upper Bound on $t_{\max}^{\mathrm{CPA}}$

Theorem 5.3 (Necessary Condition).

For any graph G, dealer D and $t \geq \mathcal{K}(G,D)$, CPA is not t-locally resilient
$\Rightarrow t_{\max}^{\mathrm{CPA}} \leq \mathcal{K}(G,D) - 1$

Upper Bound on $t_{\max}^{\mathrm{CPA}}$

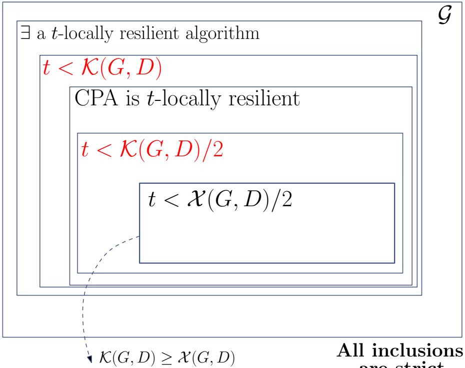
Theorem 5.3 (Necessary Condition).

For any graph G, dealer D and $t \geq \mathcal{K}(G,D)$, CPA is not t-locally resilient
$\Rightarrow t_{\max}^{\mathrm{CPA}} \leq \mathcal{K}(G,D) - 1$

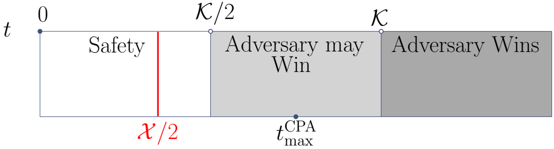
Observation (Proof Sketch)

If $t \geq \mathcal{K}(G,D) \Rightarrow \nexists \mathcal{L}_{t+1}(G,D)$. Even with no corruption at all there will always be a player who doesn't get $t+1$ messages from decided neighbors.

Condition/Bounds Overview I



All inclusions are strict



2-Approximation of $t_{\text{max}}^{\text{CPA}}$

Existence check of $\mathcal{L}_k(G, D)$ with BFS variation in $O(|E|)$ time.

Approximation Algorithm for Optimal t

1. Compute $\mathcal{K}(G, D)$ ($\log \delta$ existence checks) $O(|E| \log \delta)$.
2. Return $\lceil \mathcal{K}(G, D)/2 \rceil - 1 > \lceil t_{\text{max}}^{\text{CPA}}/2 \rceil - 1$

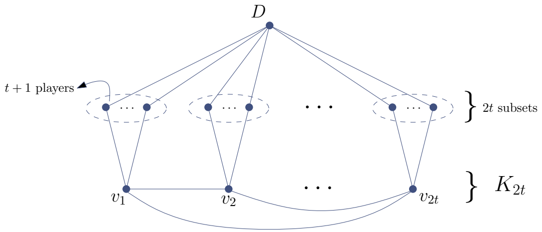
2-Approximation of $t_{\max}^{\mathrm{CPA}}$

Existence check of $\mathcal{L}_k(G, D)$ with BFS variation in $O(|E|)$ time.

Approximation Algorithm for Optimal t

1. Compute $\mathcal{K}(G, D)$ ($\log \delta$ existence checks) $O(|E| \log \delta)$.
2. Return $\lceil \mathcal{K}(G, D)/2 \rceil - 1 > \lceil t_{\max}^{\mathrm{CPA}}/2 \rceil - 1$

Tight Example.



Determining $t_{\max}^{\mathrm{CPA}}$ Exactly

With $\mathbf{G}_{\bar{\mathbf{T}}}$ we denote the **node induced subgraph** of G on the node set $V \setminus T$.

Definition 5.4 (t-safety threshold).

For graph G, dealer D and positive integer t, the *t-safety threshold* is the quantity

$$\mathcal{M}(G, D, t) = \min_{T:\ t\text{-local set}} \mathcal{K}(G_{\bar{T}}, D).$$

Determining $t_{\max}^{\mathrm{CPA}}$ Exactly

With $\mathbf{G}_{\bar{\mathbf{T}}}$ we denote the **node induced subgraph** of G on the node set $V \setminus T$.

Definition 5.4 (t-safety threshold).

For graph G, dealer D and positive integer t, the *t-safety threshold* is the quantity

$$\mathcal{M}(G, D, t) = \min_{T:\ t\text{-local set}} \mathcal{K}(G_{\bar{T}}, D).$$

Theorem 5.5 (Necessary and Sufficient Condition).

For a graph $G = (V, E)$ and dealer D, CPA is t-locally resilient iff $\mathcal{M}(G, D, t) \geq t + 1$.

Determining $t_{\max}^{\mathrm{CPA}}$ Exactly

With $\mathbf{G}_{\bar{\mathbf{T}}}$ we denote the **node induced subgraph** of G on the node set $V \setminus T$.

Definition 5.4 (t-safety threshold).

For graph G, dealer D and positive integer t, the *t-safety threshold* is the quantity
$$\mathcal{M}(G, D, t) = \min_{T:\ t\text{-local set}} \mathcal{K}(G_{\bar{T}}, D).$$

Theorem 5.5 (Necessary and Sufficient Condition).

For a graph $G = (V, E)$ and dealer D, CPA is t-locally resilient iff $\mathcal{M}(G, D, t) \geq t + 1$.

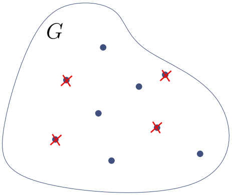
Corollary 5.6.

$$\mathcal{T}(G, D) = \max\{t \in \mathbb{N} \mid \mathcal{M}(G, D, t) \geq t + 1\} = t_{\max}^{\mathrm{CPA}}(G, D)$$

Determining $t_{\max}^{\mathrm{CPA}}$ Exactly

Proof Sketch.

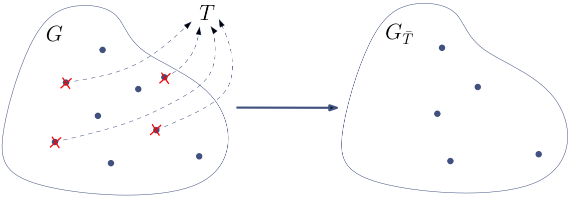
Since decision on an incorrect value is impossible, we can assume wlog that the corrupted players send nothing.



Determining $t_{\max}^{\mathrm{CPA}}$ Exactly

Proof Sketch.

Since decision on an incorrect value is impossible, we can assume wlog that the corrupted players send nothing.



$\mathcal{K}(G_{\bar{T}}, D) \geq \mathcal{M}(G, D, t) \geq t+1$,
and all players are honest

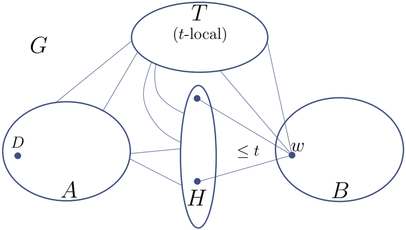
"$\Leftarrow$" If $\mathcal{M}(G, D, t) \geq t+1$, each player has at least $t+1$ decided neighbors in all possible $G_{\bar{T}}$.

"$\Rightarrow$" If $\mathcal{M}(G, D, t) \leq t$, then there exists a player that won't have $t+1$ decided neighbors in all possible $G_{\bar{T}}$.

A Simpler Characterization of $t_{\text{max}}^{\text{CPA}}$

Definition 5.7 (t-Partial Local Pair Cut).

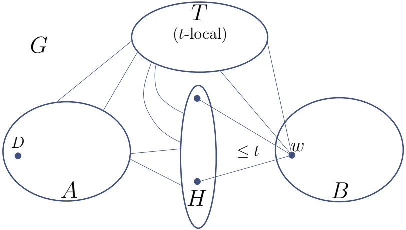
Let C be a node-cut of G, partitioning $V \setminus C$ into sets $A, B \neq \emptyset$ s.t. $D \in A$. C is a *t-partial local pair cut* (*t-plp cut*) in G, D if there exists a partition $C = T \cup H$ where T is t-local and $\forall w \in B, |\mathcal{N}(w) \cap H| \leq t$ (H is t-local w.r.t. B).



A Simpler Characterization of $t_{\text{max}}^{\text{CPA}}$

Definition 5.7 (t-Partial Local Pair Cut).

Let C be a node-cut of G, partitioning $V \setminus C$ into sets $A, B \neq \emptyset$ s.t. $D \in A$. C is a *t-partial local pair cut* (*t-plp cut*) in G, D if there exists a partition $C = T \cup H$ where T is t-local and $\forall w \in B, |\mathcal{N}(w) \cap H| \leq t$ (H is t-local w.r.t. B).



Equivalent Necessary and sufficient condition

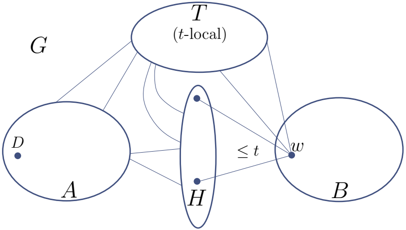
Theorem 5.8.

For G, D, CPA is t-locally resilient iff no t-plp cut exists.

A Simpler Characterization of $t_{\max}^{\mathrm{CPA}}$

Definition 5.7 (t-Partial Local Pair Cut).

Let C be a node-cut of G, partitioning $V \setminus C$ into sets $A, B \neq \emptyset$ s.t. $D \in A$. C is a *t-partial local pair cut* (*t-plp cut*) in G, D if there exists a partition $C = T \cup H$ where T is t-local and $\forall w \in B, |\mathcal{N}(w) \cap H| \leq t$ (H is t-local w.r.t. B).



Equivalent Necessary and sufficient condition

Theorem 5.8.

For G, D, CPA is t-locally resilient iff no t-plp cut exists.

$$t_{\max}^{\mathrm{CPA}}(G, D) = \max\{t \in \mathbb{N} \mid \nexists t-plp \text{ cut in } G, D\}$$

CPA Uniqueness in *Ad Hoc* Networks

Ad Hoc Network Model

Nodes know only their own labels, the labels of their neighbors and the label of the dealer. An *ad hoc* algorithm operates under these assumptions.

CPA Uniqueness in *Ad Hoc* Networks

Ad Hoc Network Model

Nodes know only their own labels, the labels of their neighbors and the label of the dealer. An *ad hoc* algorithm operates under these assumptions.

CPA Uniqueness Conjecture

No *ad hoc* algorithm can locally tolerate more traitors than CPA.

CPA Uniqueness in *Ad Hoc* Networks

Ad Hoc Network Model

Nodes know only their own labels, the labels of their neighbors and the label of the dealer. An *ad hoc* algorithm operates under these assumptions.

CPA Uniqueness Conjecture

No *ad hoc* algorithm can locally tolerate more traitors than CPA.

Observation: There exists a non-safe algorithm (*Relaxed Propagation algorithm* [PP05]) which locally tolerates more traitors than CPA in certain families of graphs.

CPA Uniqueness in *Ad Hoc* Networks

Ad Hoc Network Model

Nodes know only their own labels, the labels of their neighbors and the label of the dealer. An *ad hoc* algorithm operates under these assumptions.

CPA Uniqueness Conjecture

No *ad hoc* algorithm can locally tolerate more traitors than CPA.

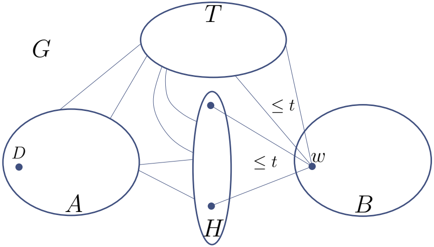
Observation: There exists a non-safe algorithm (*Relaxed Propagation algorithm* [PP05]) which locally tolerates more traitors than CPA in certain families of graphs.

Theorem 5.9.

Let $\mathcal{A}$ be a t-locally safe ad hoc Broadcast algorithm. If $\mathcal{A}$ is t-locally resilient for a graph G with dealer D then CPA is t-locally resilient for G, D.

Proof Sketch

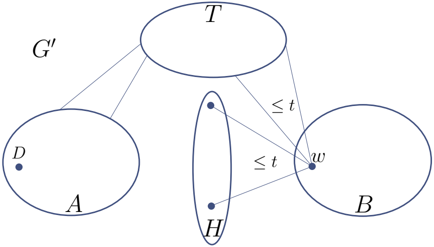
Assume that CPA is not t-locally resilient in G, D, then there exists a t-plp cut $C = T \cup H$ in G, D.



Proof Sketch

Assume that CPA is not t-locally resilient in G, D, then there exists a t-plp cut $C = T \cup H$ in G, D.

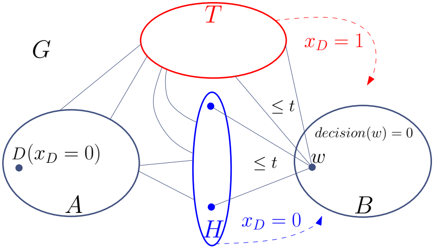
Let G' be the graph which results if we remove the edges that connect the set $A \cup T$ with H. Then H is t-local in G'.



Proof Sketch

Assume that CPA is not t-locally resilient in G, D, then there exists a t-plp cut $C = T \cup H$ in G, D.

Let G' be the graph which results if we remove the edges that connect the set $A \cup T$ with H. Then H is t-local in G'.



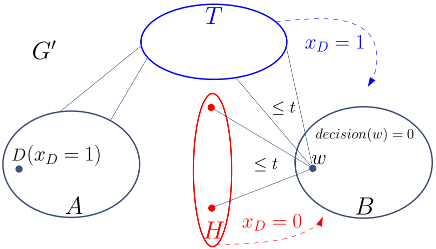
Execution of $\mathcal{A}$	σ_0	σ_1
Dealer's value x_D	0	1
Corruption set	T	H
Graph	G	G'

Corrupted players of σ_i act as honest in σ_{1-i}.

Proof Sketch

Assume that CPA is not t-locally resilient in G, D, then there exists a t-plp cut $C = T \cup H$ in G, D.

Let G' be the graph which results if we remove the edges that connect the set $A \cup T$ with H. Then H is t-local in G'.



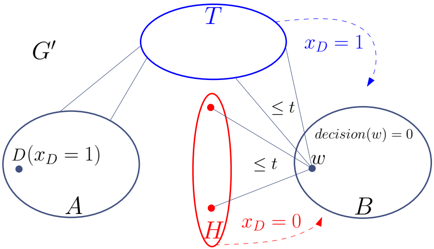
Execution of $\mathcal{A}$	σ_0	σ_1
Dealer's value x_D	0	1
Corruption set	T	H
Graph	G	G'

Corrupted players of σ_i act as honest in σ_{1-i}.

Proof Sketch

Assume that CPA is not t-locally resilient in G, D, then there exists a t-plp cut $C = T \cup H$ in G, D.

Let G' be the graph which results if we remove the edges that connect the set $A \cup T$ with H. Then H is t-local in G'.



Execution of $\mathcal{A}$	σ_0	σ_1
Dealer's value x_D	0	1
Corruption set	T	H
Graph	G	G'

Corrupted players of σ_i act as honest in σ_{1-i}.

Using $\mathcal{A}$, w decides on the same value in σ_0, σ_1, thus $\mathcal{A}$ is not t-locally safe. □

Complexity of Computing $t_{\mathsf{max}}^{\mathrm{CPA}}$

To show that the computation of $t_{\mathsf{max}}^{\mathrm{CPA}}$ is NP-hard it suffices to show that the following decisional problem is NP-hard.

pLPC Problem

Given a graph G, a dealer-node D and integer t determine whether there exists a t-plp cut in G, D.

Complexity of Computing $t_{\mathsf{max}}^{\mathrm{CPA}}$

To show that the computation of $t_{\mathsf{max}}^{\mathrm{CPA}}$ is NP-hard it suffices to show that the following decisional problem is NP-hard.

pLPC Problem

Given a graph G, a dealer-node D and integer t determine whether there exists a t-plp cut in G, D.

Theorem 5.10.

pLPC is NP*-hard.*

Complexity of Computing $t_{\max}^{\mathrm{CPA}}$

To show that the computation of $t_{\max}^{\mathrm{CPA}}$ is NP-hard it suffices to show that the following decisional problem is NP-hard.

pLPC Problem

Given a graph G, a dealer-node D and integer t determine whether there exists a t-plp cut in G, D.

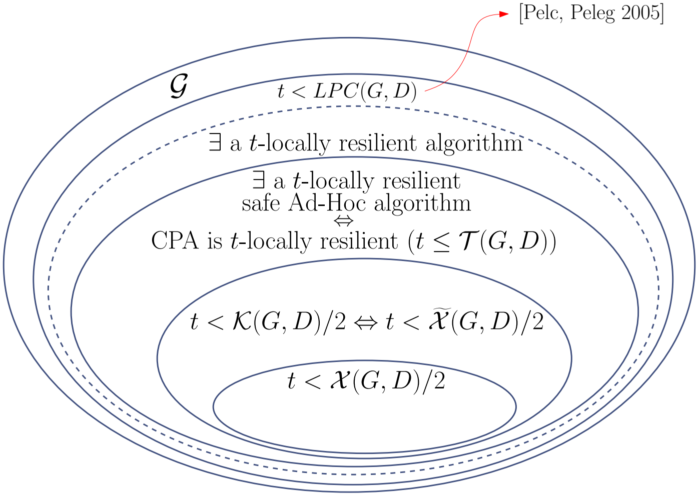
Theorem 5.10.

pLPC is NP-hard.

Observation

A polynomially bounded adversary is unable to design an optimal attack unless $\mathrm{P} = \mathrm{NP}$.

Overview of Conditions II



Conclusions and Open Problems

Better approximation of $t_{\mathsf{max}}^{\mathrm{CPA}}$

What is the best attack a polynomially bounded adversary could deploy? In other words,

- Obtain a better approximation algorithm (ideally a PTAS) for $t_{\mathsf{max}}^{\mathrm{CPA}}$.
- A graph parameter more accurate than $\mathcal{K}$.

Conclusions and Open Problems

Better approximation of $t_{\max}^{\mathrm{CPA}}$

What is the best attack a polynomially bounded adversary could deploy?
In other words,

- Obtain a better approximation algorithm (ideally a PTAS) for $t_{\max}^{\mathrm{CPA}}$.
- A graph parameter more accurate than $\mathcal{K}$.

Model Variations

- Global/Partial Knowledge of Topology [PPS14].
- General Adversary.
- Computation of $t_{\max}^{\mathrm{CPA}}$ in specific network topologies.
- Wireless Networks (Collision Avoidance).

References I

B. A. Coan.
Achieving consensus in fault-tolerant distributed computer systems: protocols, lower bounds, and simulations.
PhD thesis, Cambridge, MA, USA, 1987.

Danny Dolev.
The byzantine generals strike again.
J. Algorithms, 3(1):14–30, 1982.

Danny Dolev and Rüdiger Reischuk.
Bounds on information exchange for byzantine agreement.
J. ACM, 32(1):191–204, 1985.

Danny Dolev and H. Raymond Strong.
Authenticated algorithms for byzantine agreement.
SIAM J. Comput., 12(4):656–666, 1983.

References II

Michael J. Fischer and Nancy A. Lynch.
A lower bound for the time to assure interactive consistency.
Inf. Process. Lett., 14(4):183–186, 1982.

Michael J. Fischer, Nancy A. Lynch, and Mike Paterson.
Impossibility of distributed consensus with one faulty process.
J. ACM, 32(2):374–382, 1985.

Juan A. Garay and Yoram Moses.
Fully polynomial byzantine agreement for $n > 3t$ processors in $t+1$ rounds.
SIAM J. Comput., 27(1):247–290, 1998.

Akira Ichimura and Maiko Shigeno.
A new parameter for a broadcast algorithm with locally bounded byzantine faults.
Inf. Process. Lett., 110(12-13):514–517, 2010.

References III

Chiu-Yuen Koo.
Broadcast in radio networks tolerating byzantine adversarial behavior.
In Soma Chaudhuri and Shay Kutten, editors, *PODC*, pages 275–282. ACM, 2004.

Chris Litsas, Aris Pagourtzis, and Dimitris Sakavalas.
A graph parameter that matches the resilience of the certified propagation algorithm.
In Jacek Cichon, Maciej Gebala, and Marek Klonowski, editors, *ADHOC-NOW*, volume 7960 of *Lecture Notes in Computer Science*, pages 269–280. Springer, 2013.

Leslie Lamport, Robert E. Shostak, and Marshall C. Pease.
The byzantine generals problem.
ACM Trans. Program. Lang. Syst., 4(3):382–401, 1982.

References IV

Andrzej Pelc and David Peleg.
Broadcasting with locally bounded byzantine faults.
Inf. Process. Lett., 93(3):109–115, 2005.

Aris Pagourtzis, Giorgos Panagiotakos, and Dimitris Sakavalas.
Optimal resilience broadcast against locally bounded and general adversaries.
IACR Cryptology ePrint Archive, 2014:290, 2014.

Marshall C. Pease, Robert E. Shostak, and Leslie Lamport.
Reaching agreement in the presence of faults.
J. ACM, 27(2):228–234, 1980.