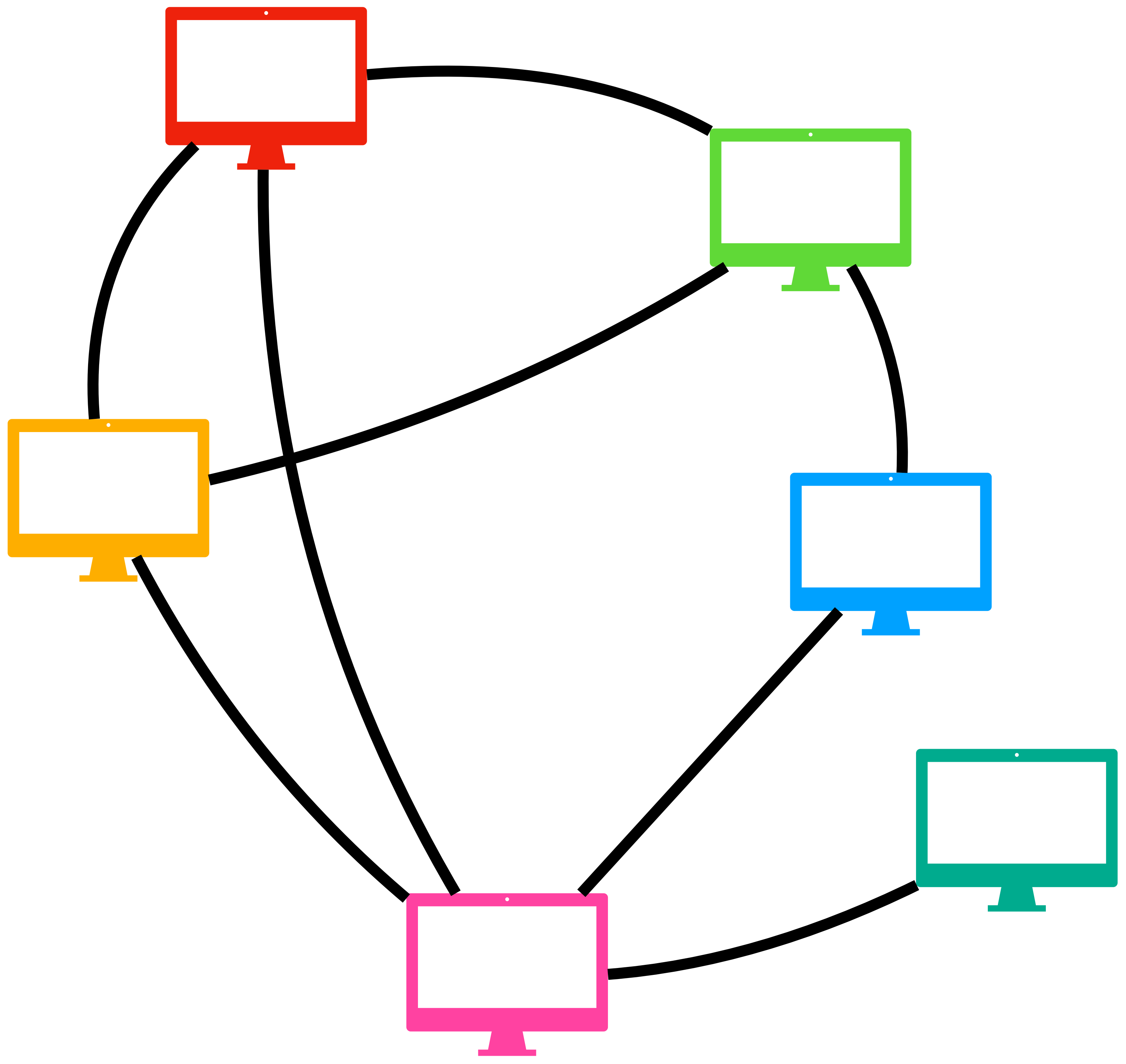
Message Complexity of Distributed Graph Algorithms

Shreyas Pai
IIT Madras

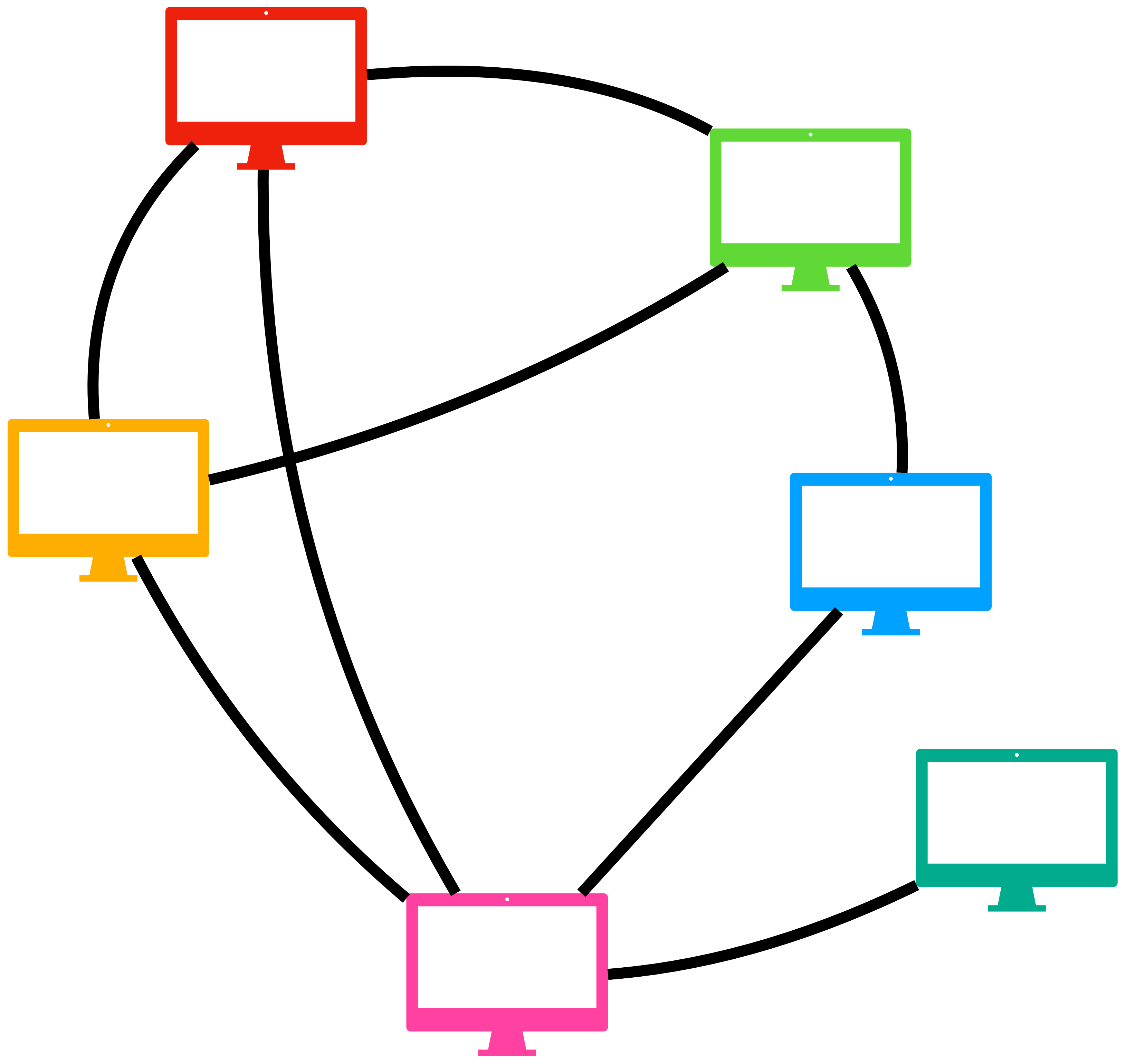
Advances in Distributed Graph Algorithms, ADGA 2024

The CONGEST Model



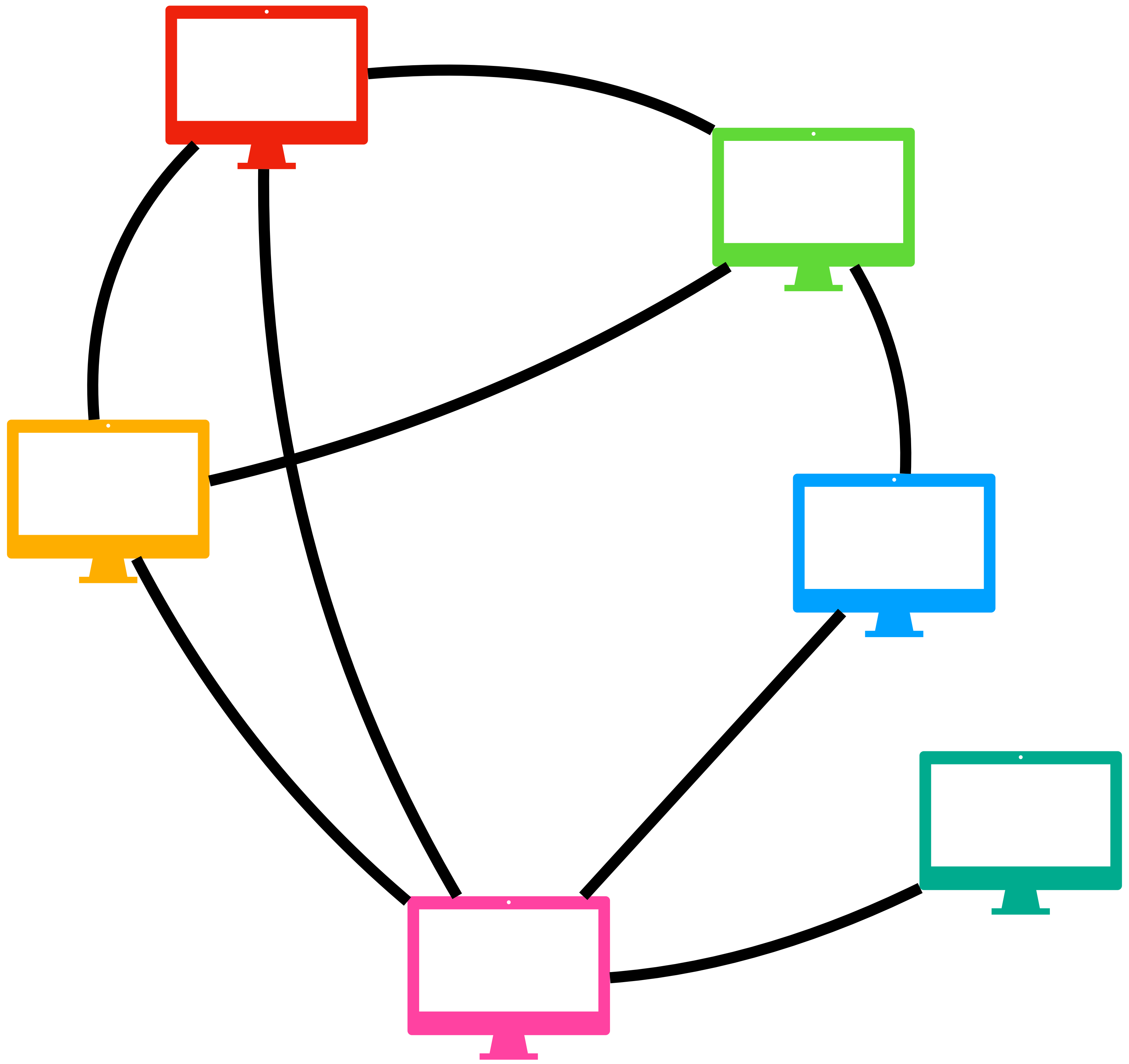
The CONGEST Model

- Given a graph G which is the communication network.



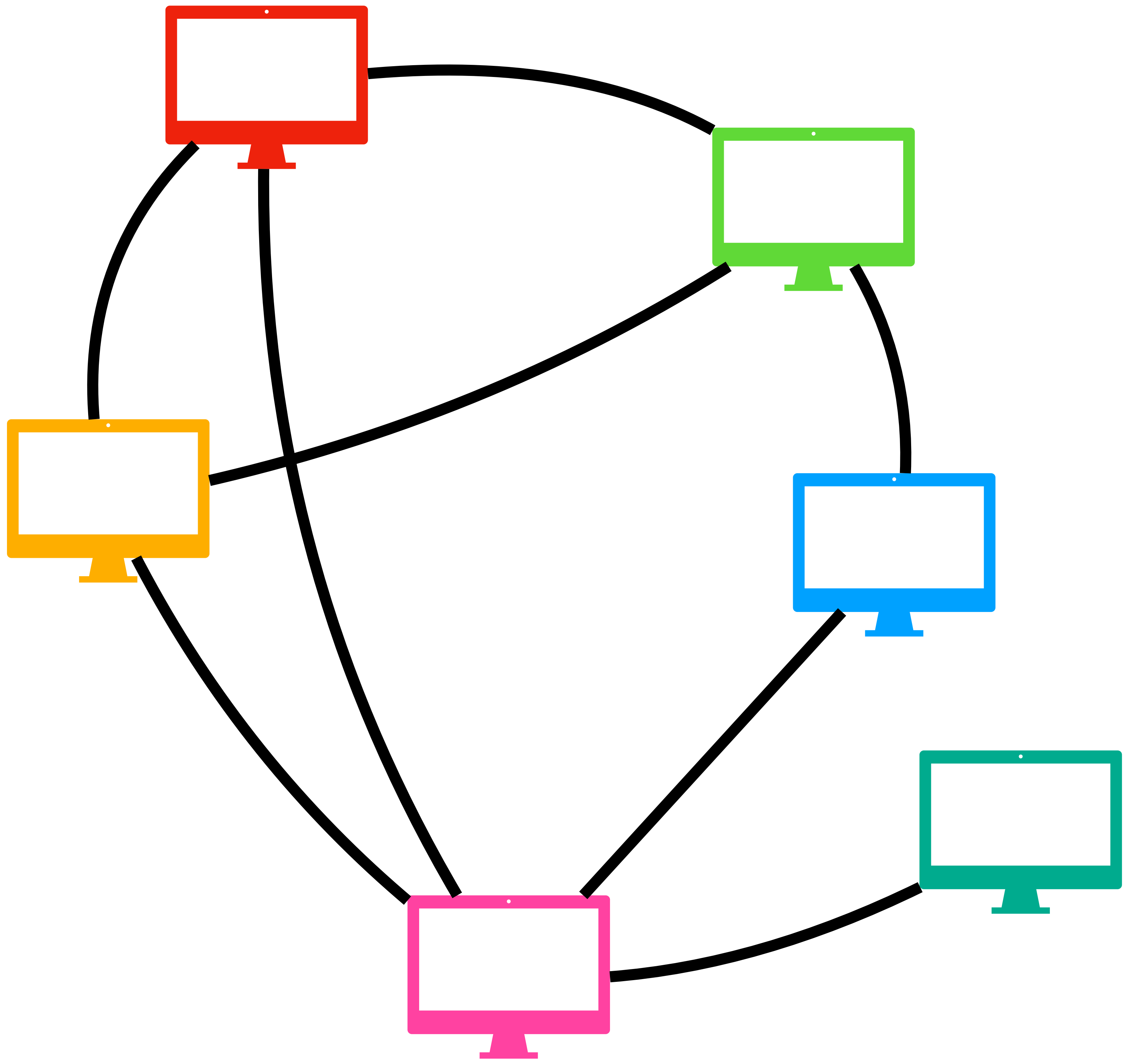
The CONGEST Model

- Given a graph G which is the communication network.
  - n nodes and m edges.



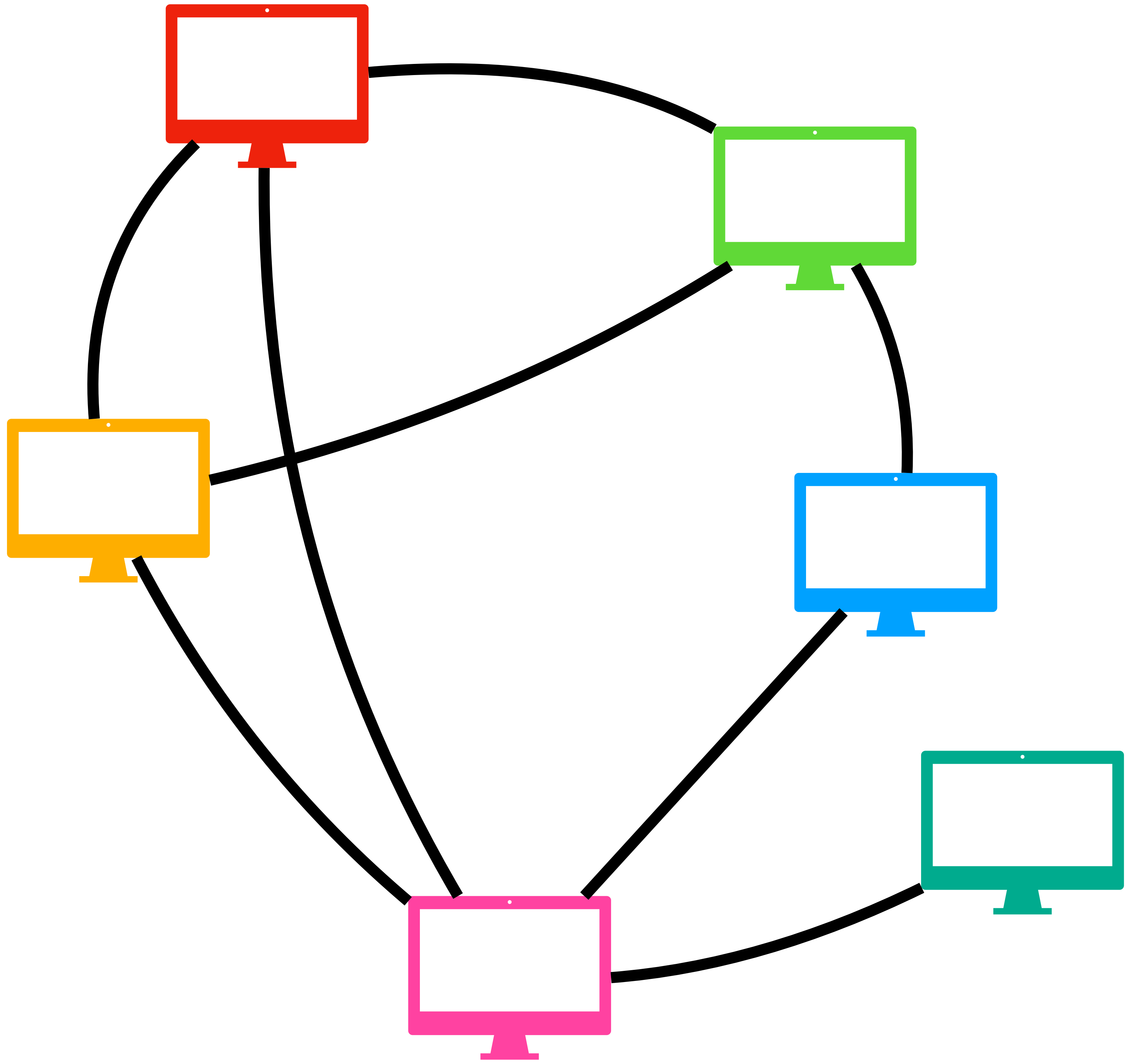
The CONGEST Model

- Given a graph G which is the communication network.
    - n nodes and m edges.
    - Goal: solve a problem on G, like broadcast or coloring.

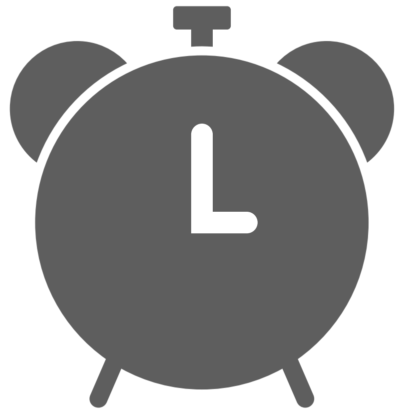


The CONGEST Model

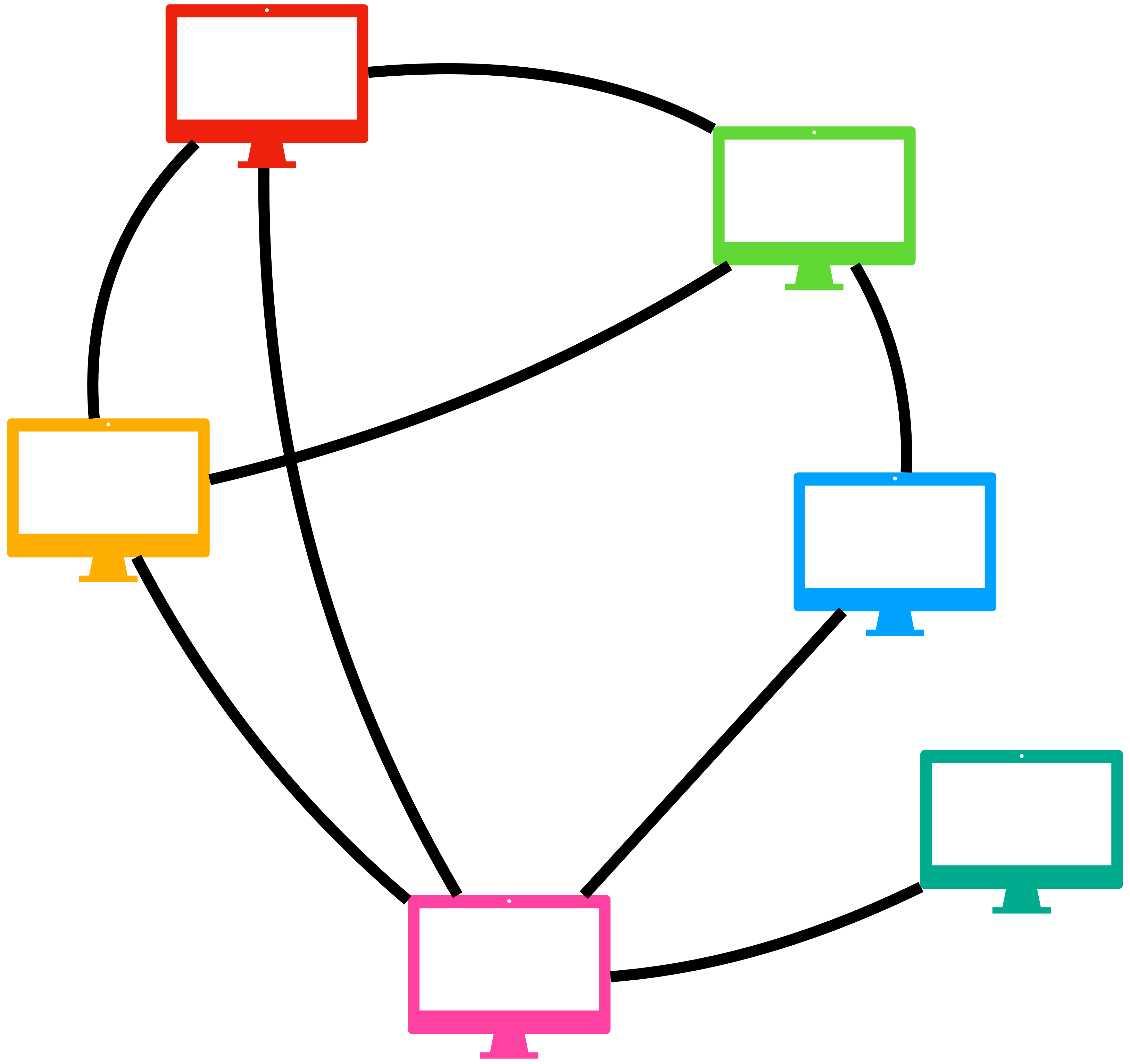
- Given a graph G which is the communication network.
    - n nodes and m edges.
    - Goal: solve a problem on G, like broadcast or coloring.
- Synchronous message-passing model.



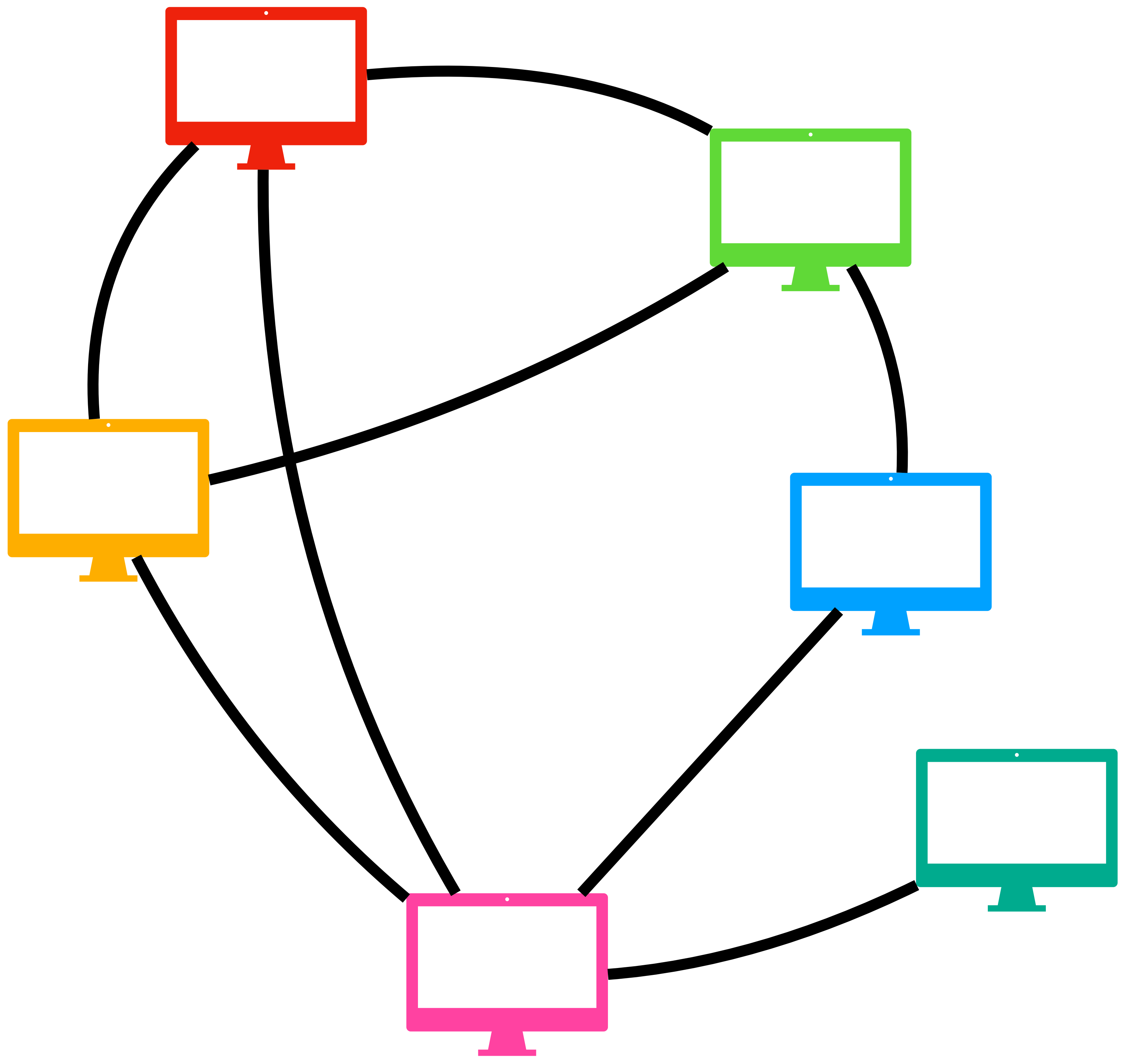
The CONGEST Model



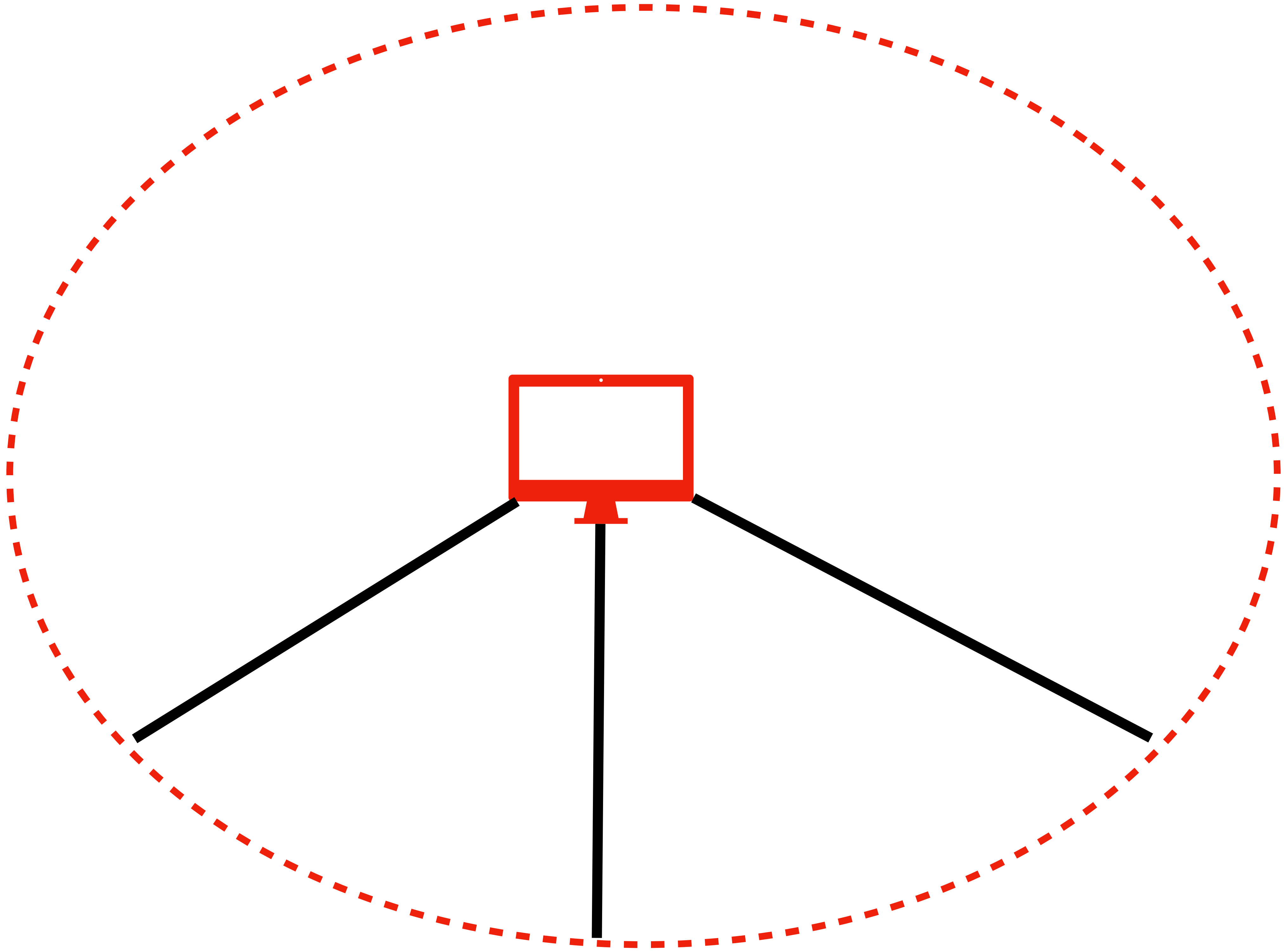
- Given a graph G which is the communication network.
  - n nodes and m edges.
  - Goal: solve a problem on G, like broadcast or coloring.
- Synchronous message-passing model.
- Nodes send $O(\log n)$-bit messages per neighbor per round.

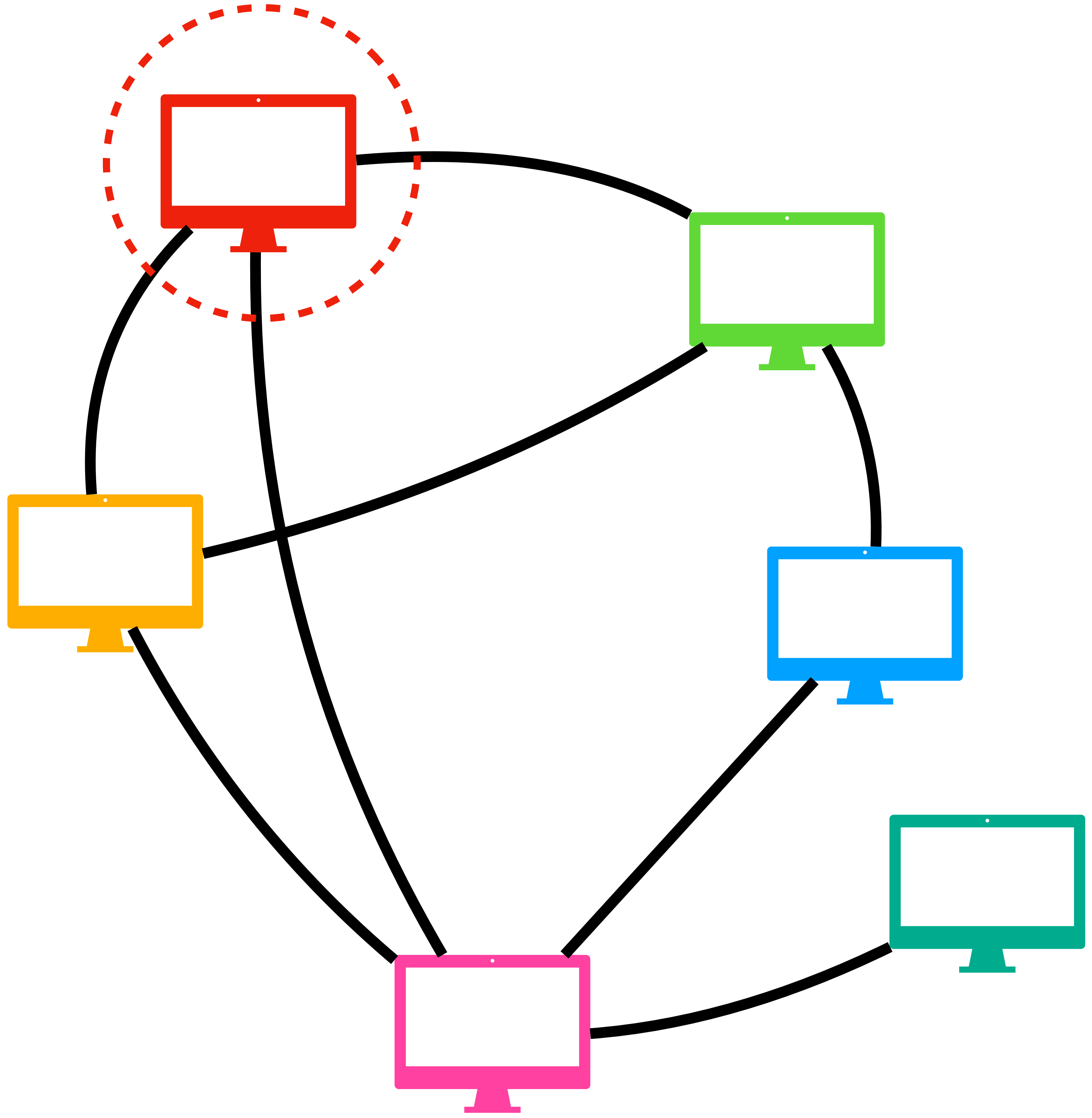


A Synchronous Round

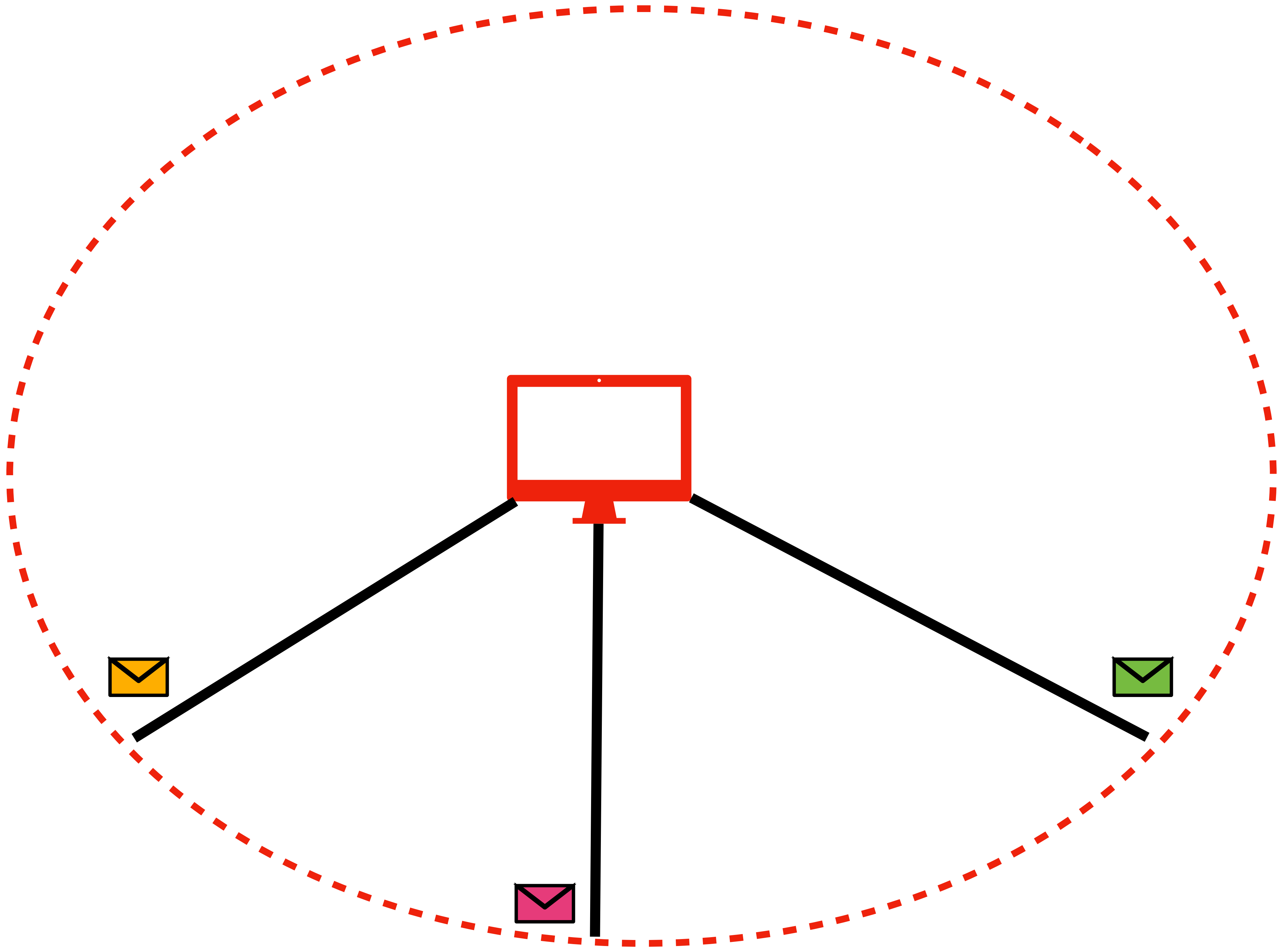


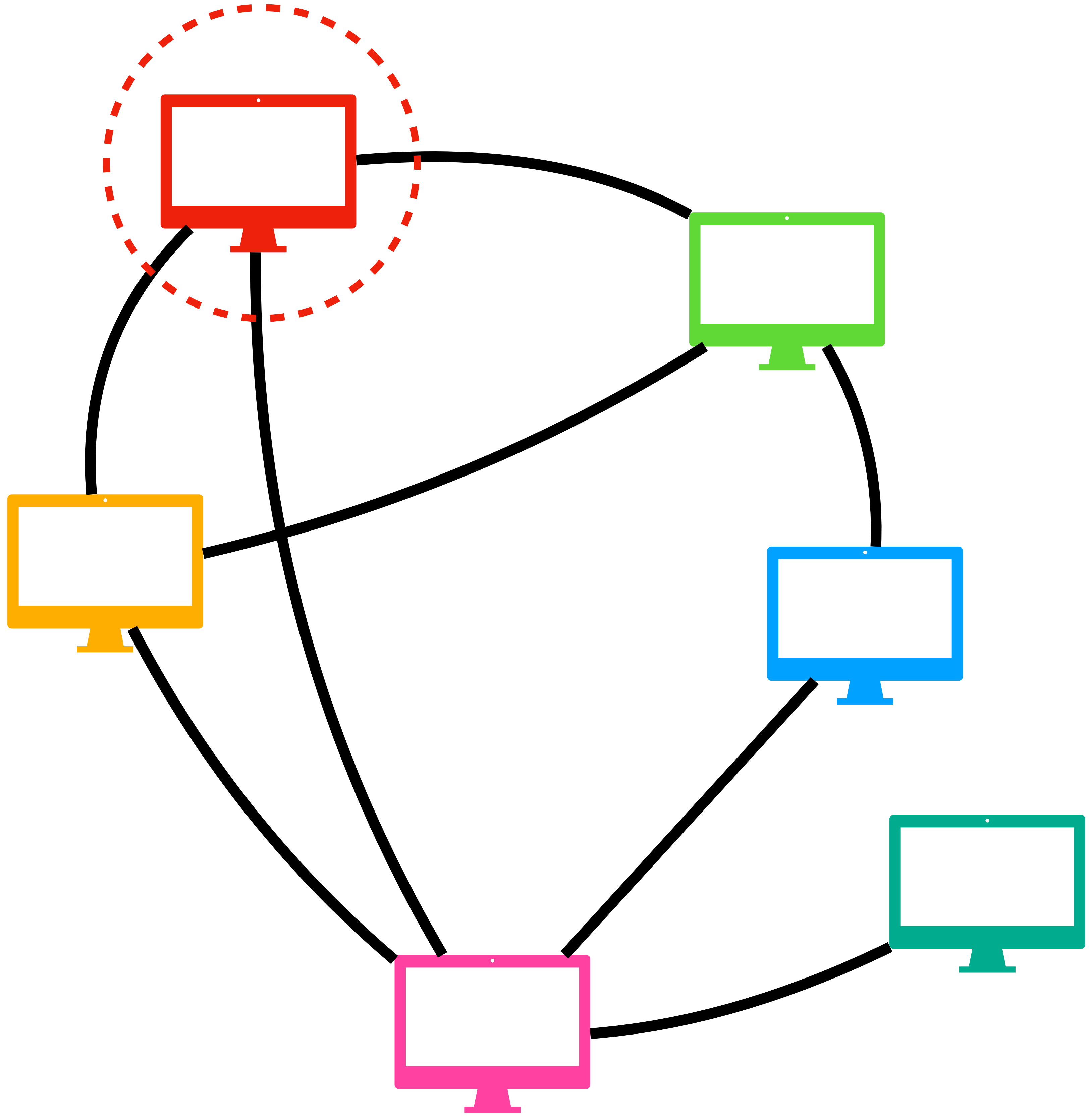
A Synchronous Round



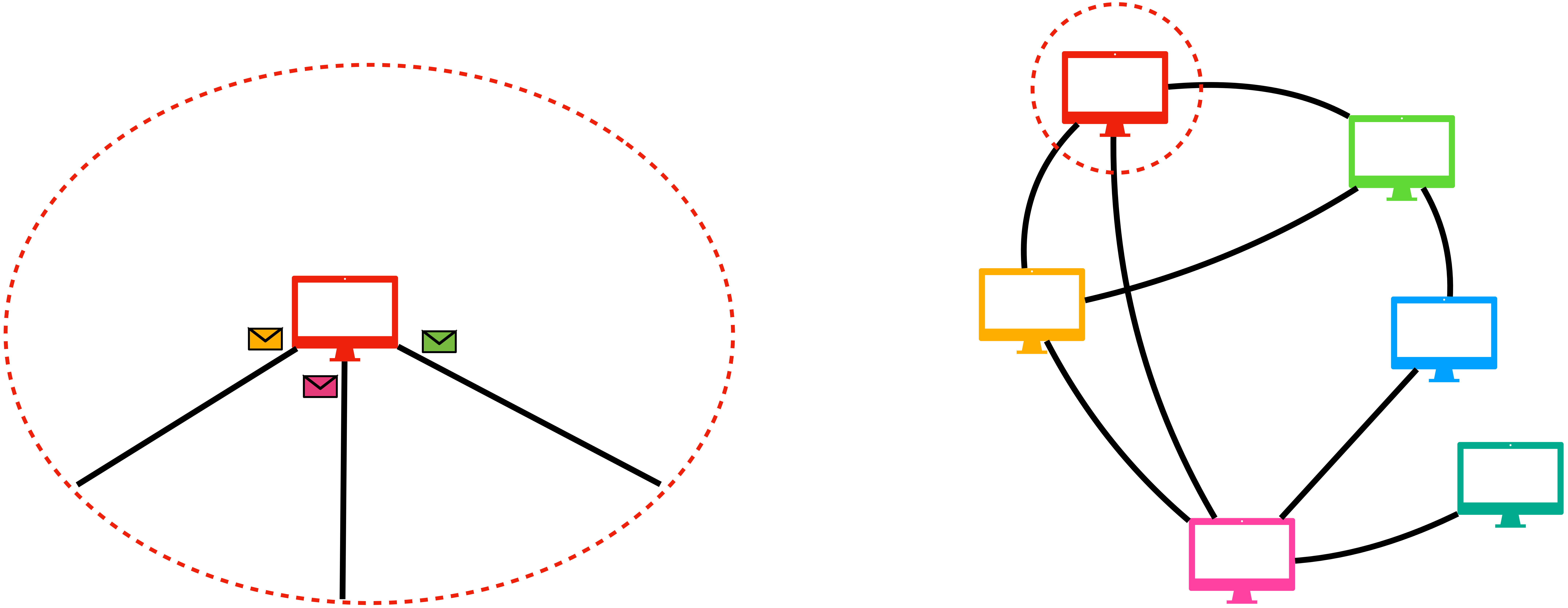


A Synchronous Round

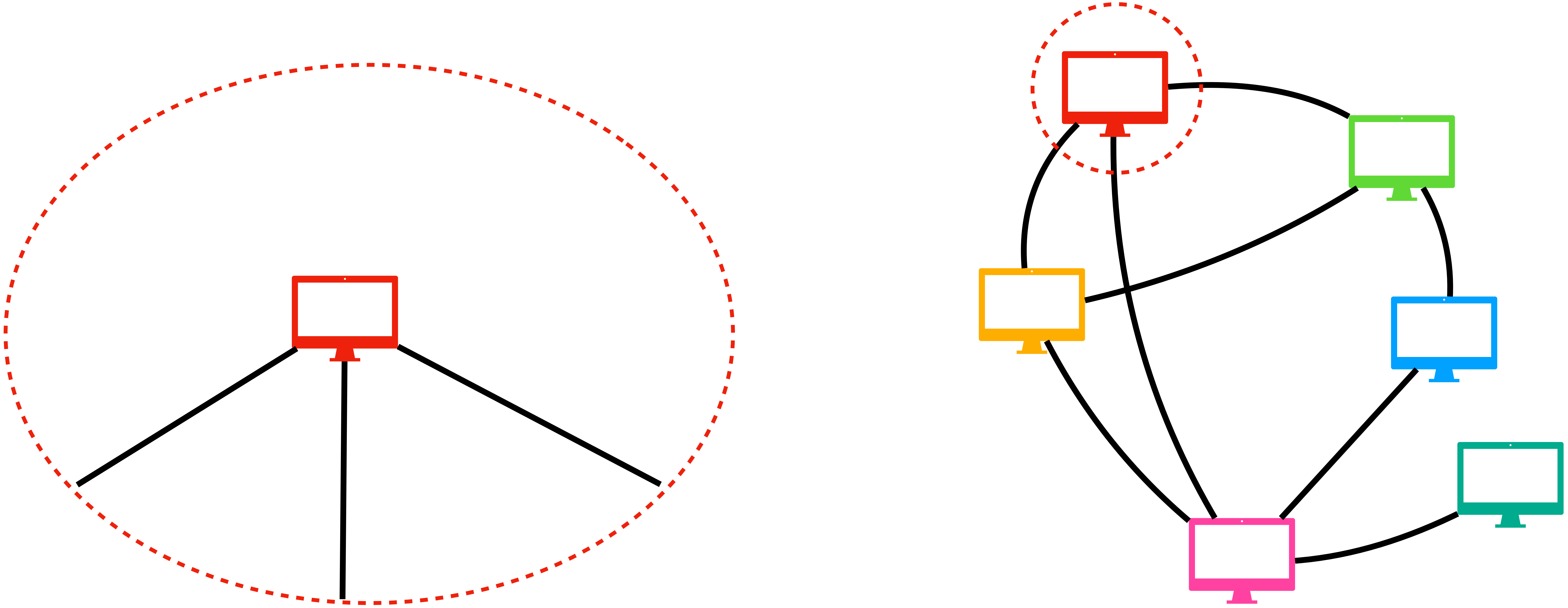




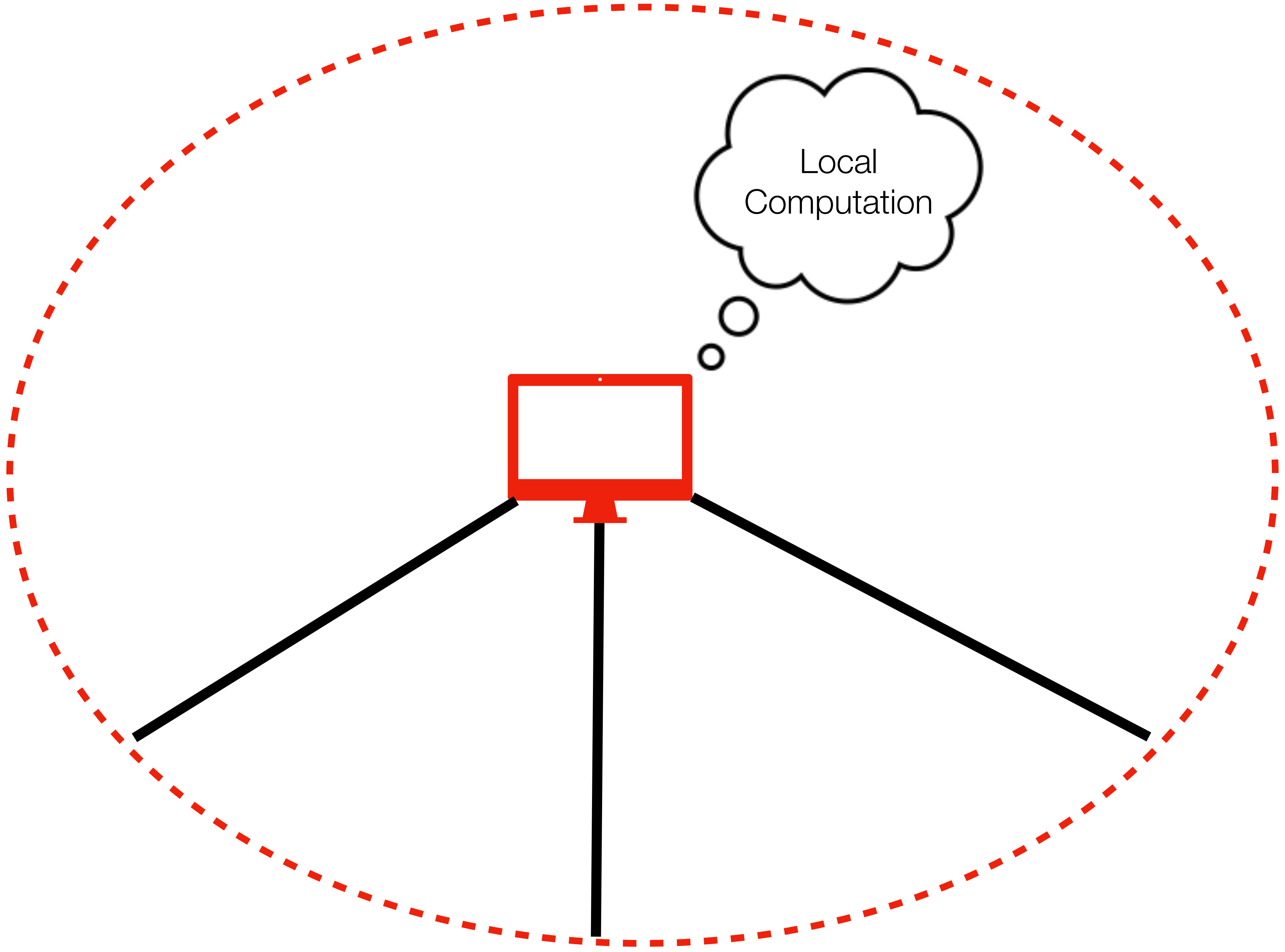
A Synchronous Round

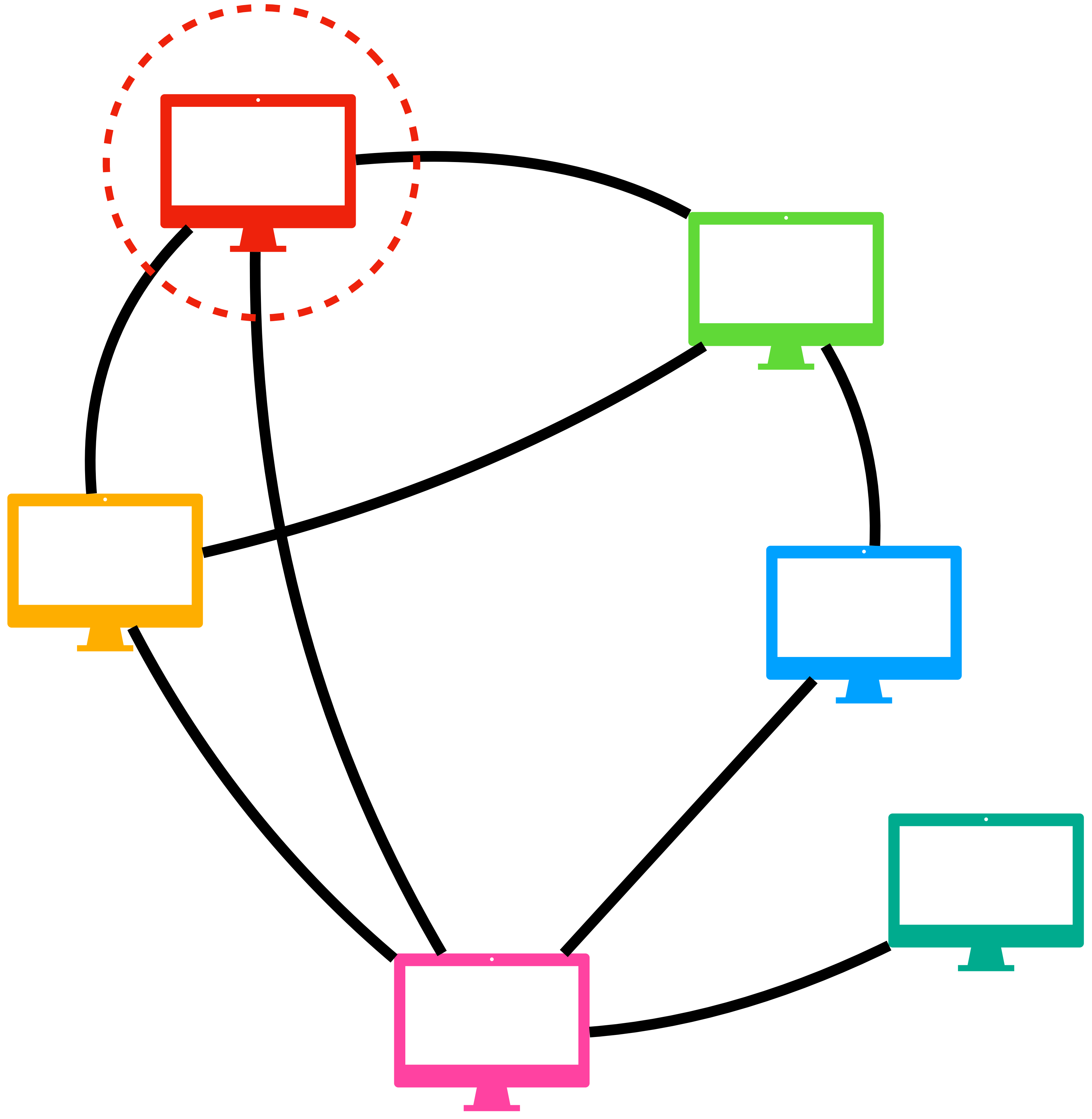


A Synchronous Round

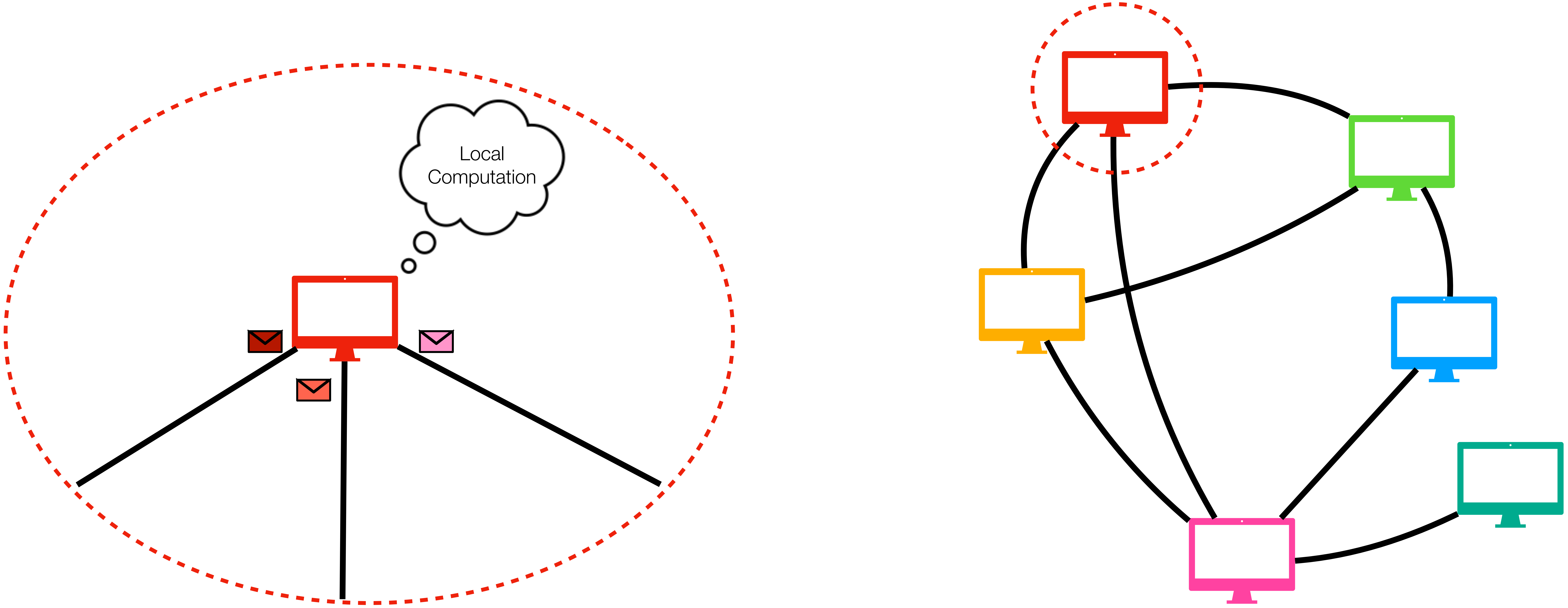


A Synchronous Round

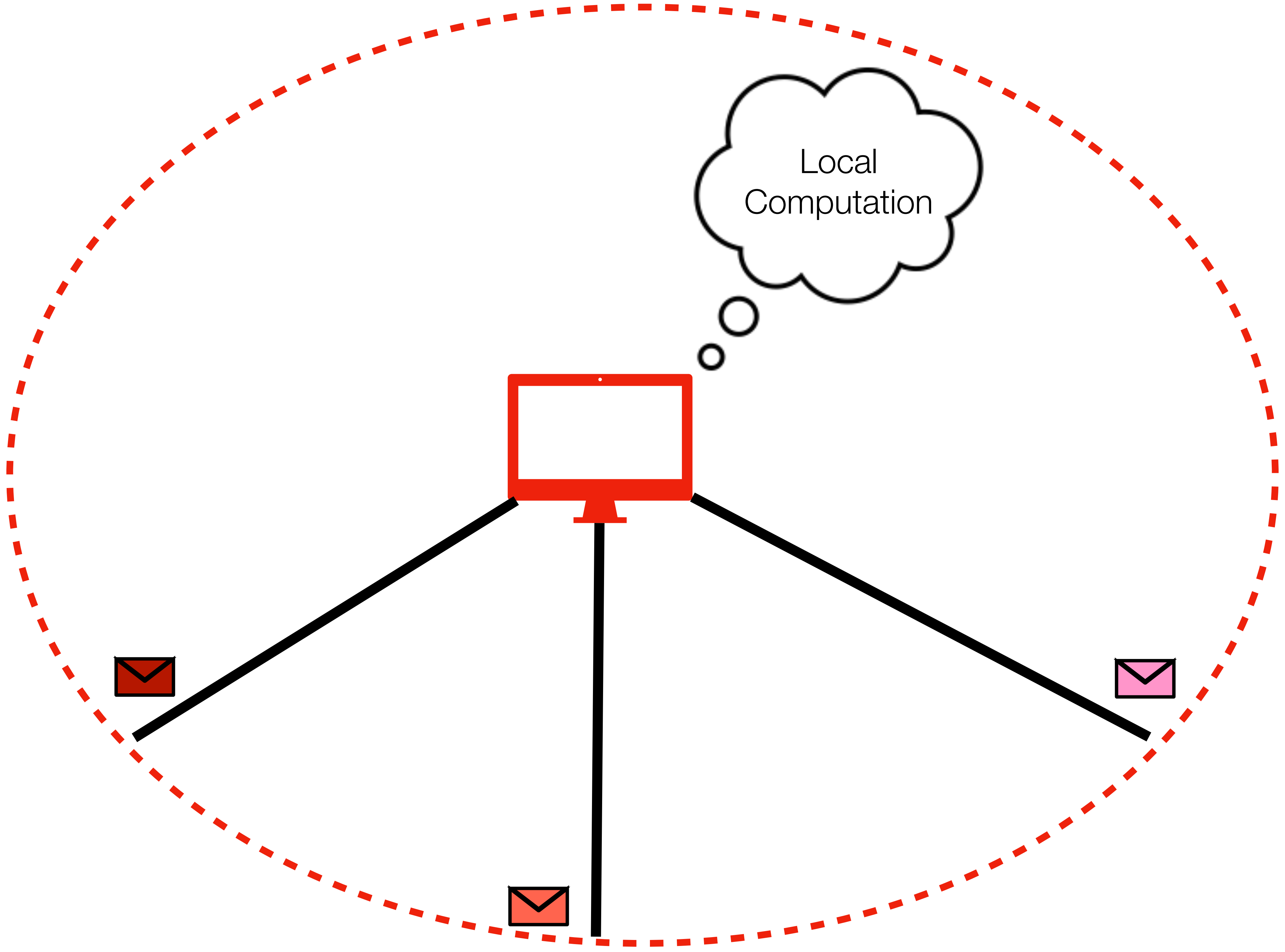


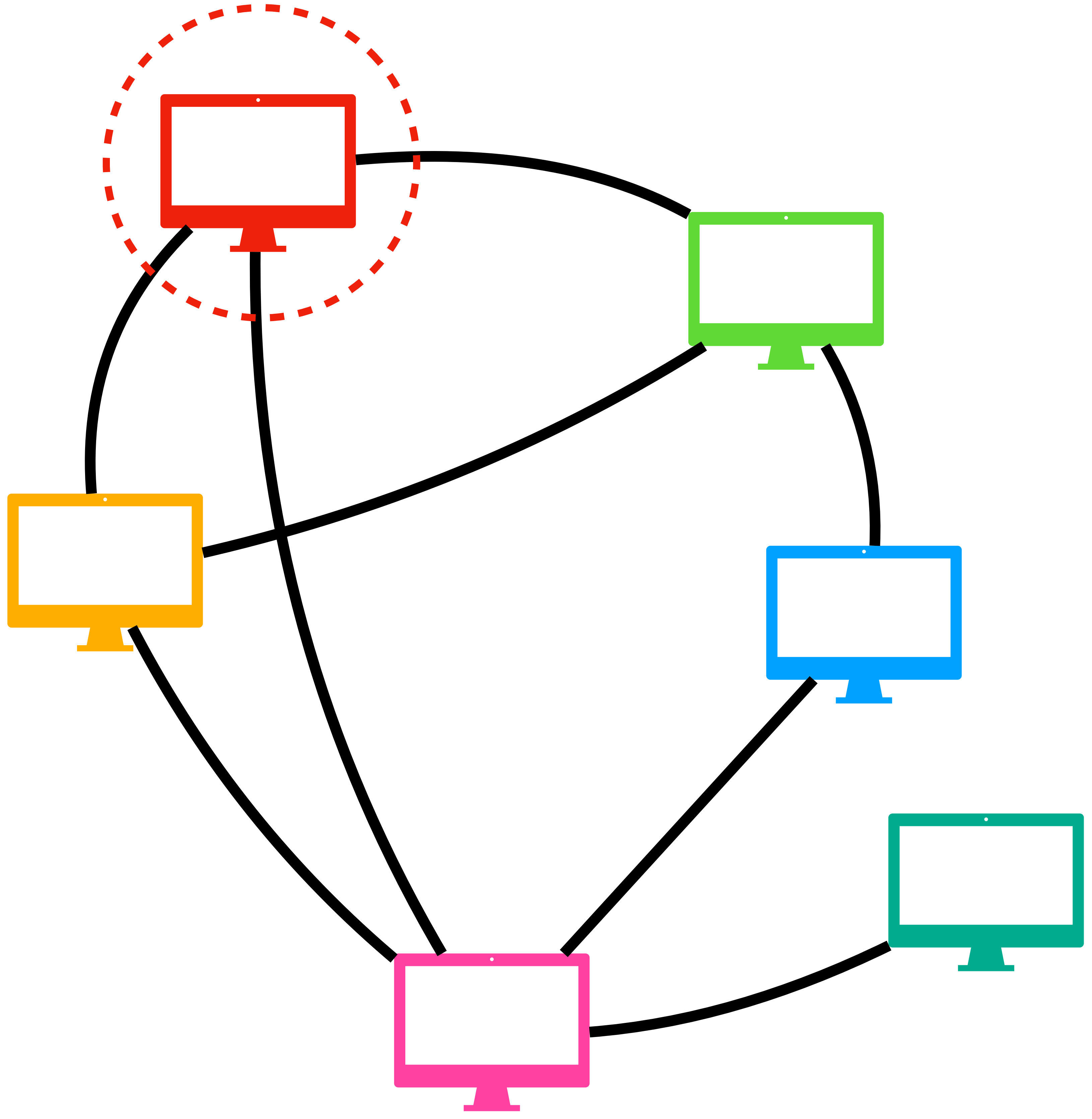


A Synchronous Round

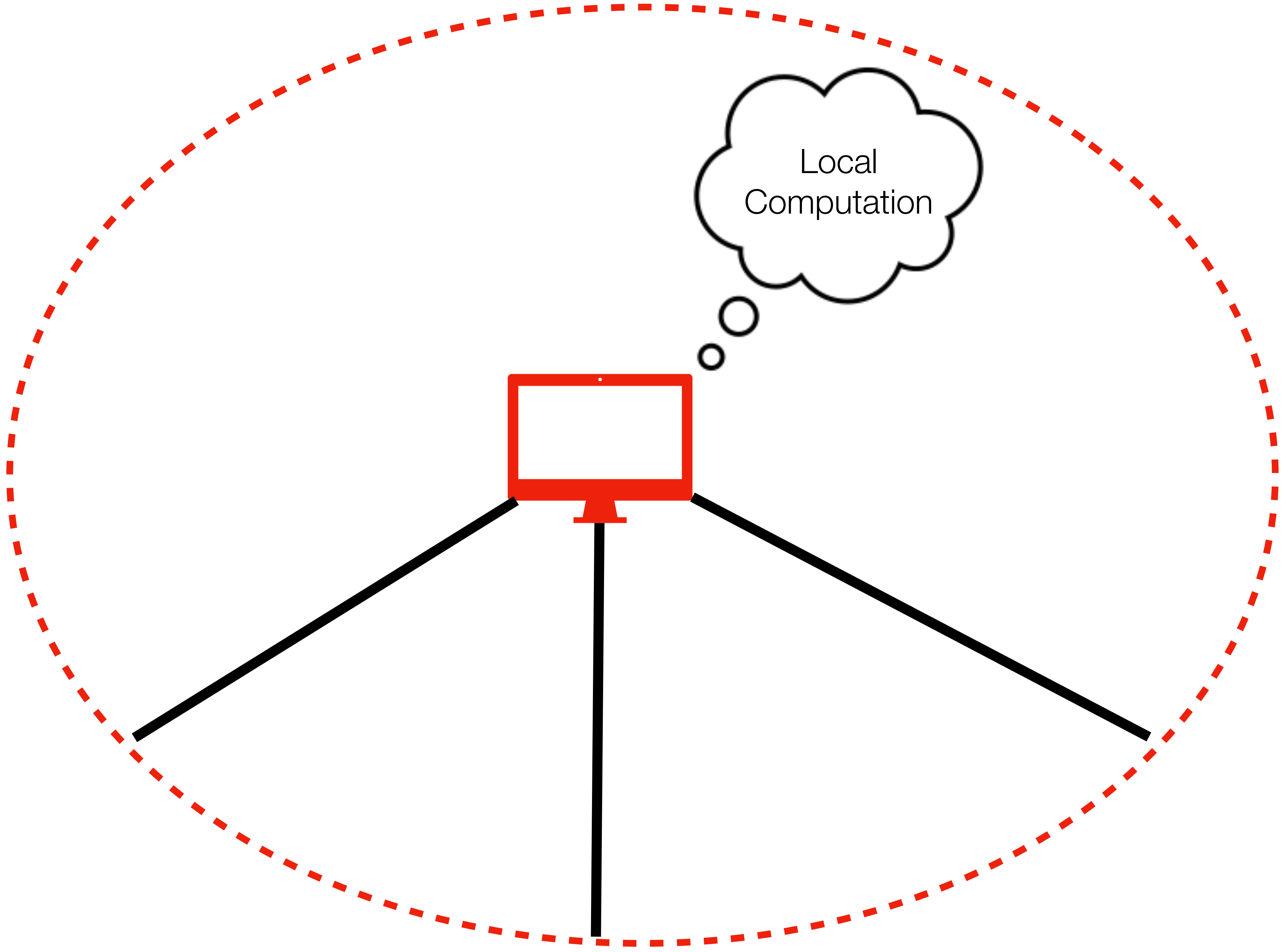


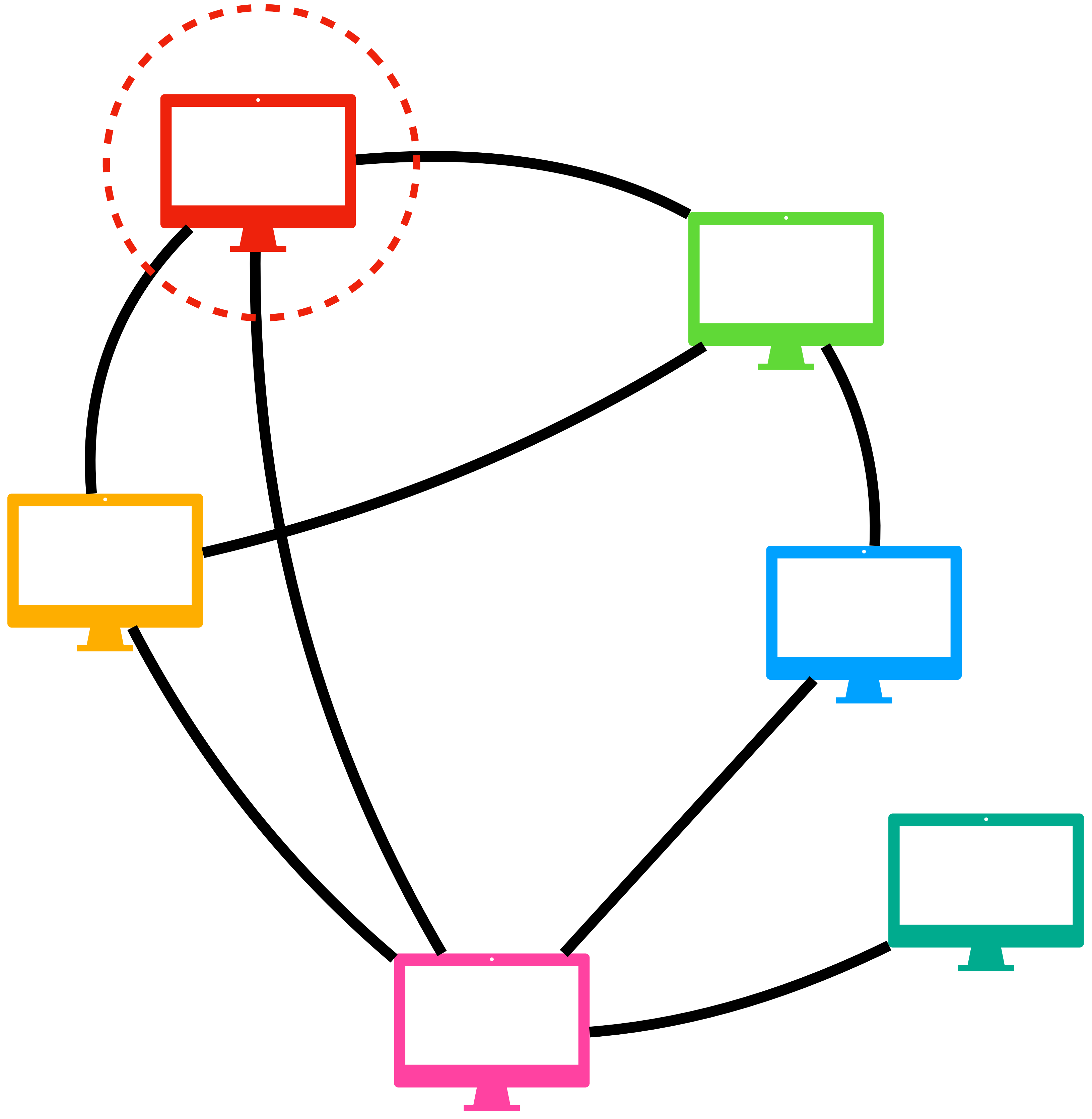
A Synchronous Round





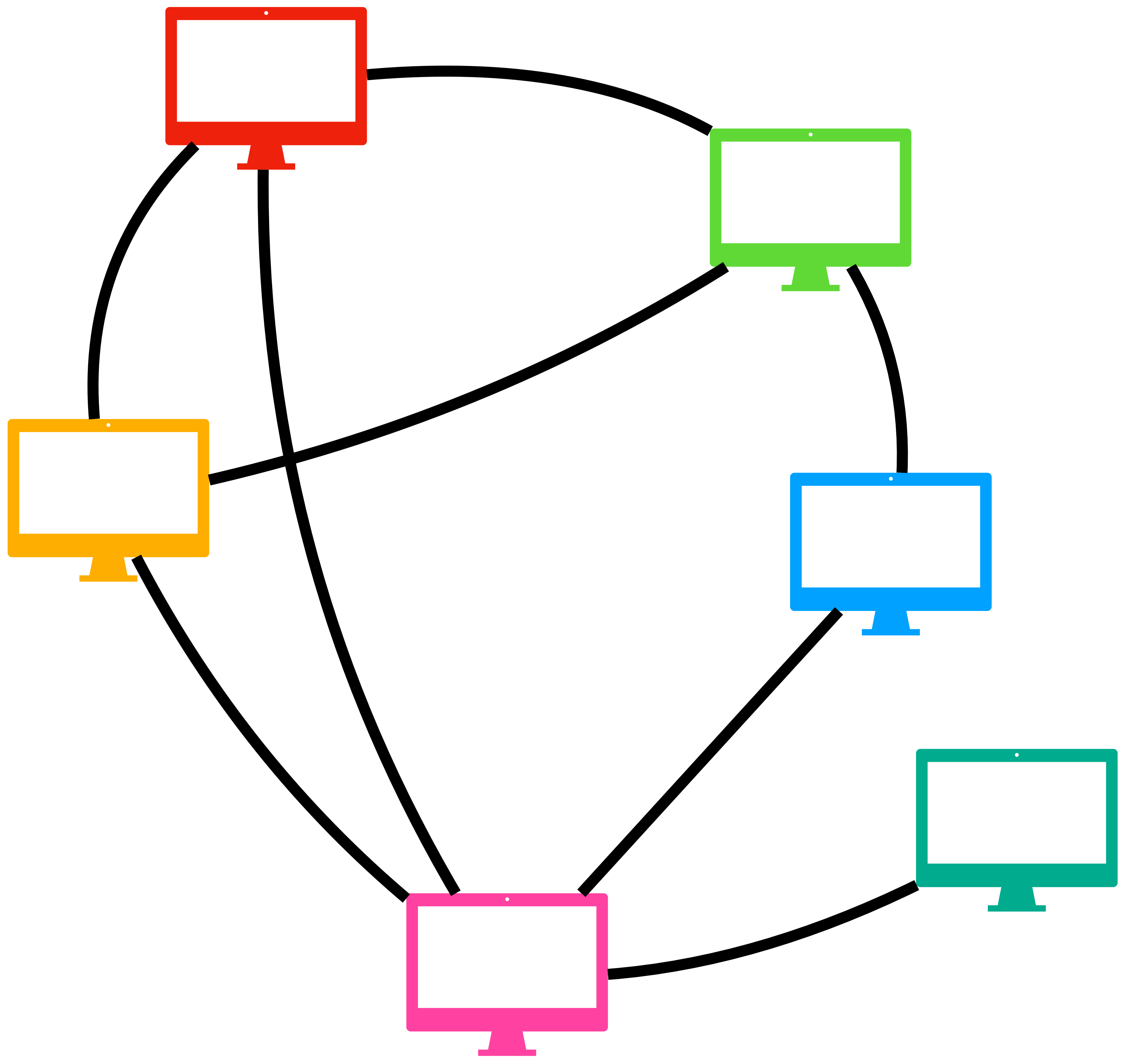
A Synchronous Round



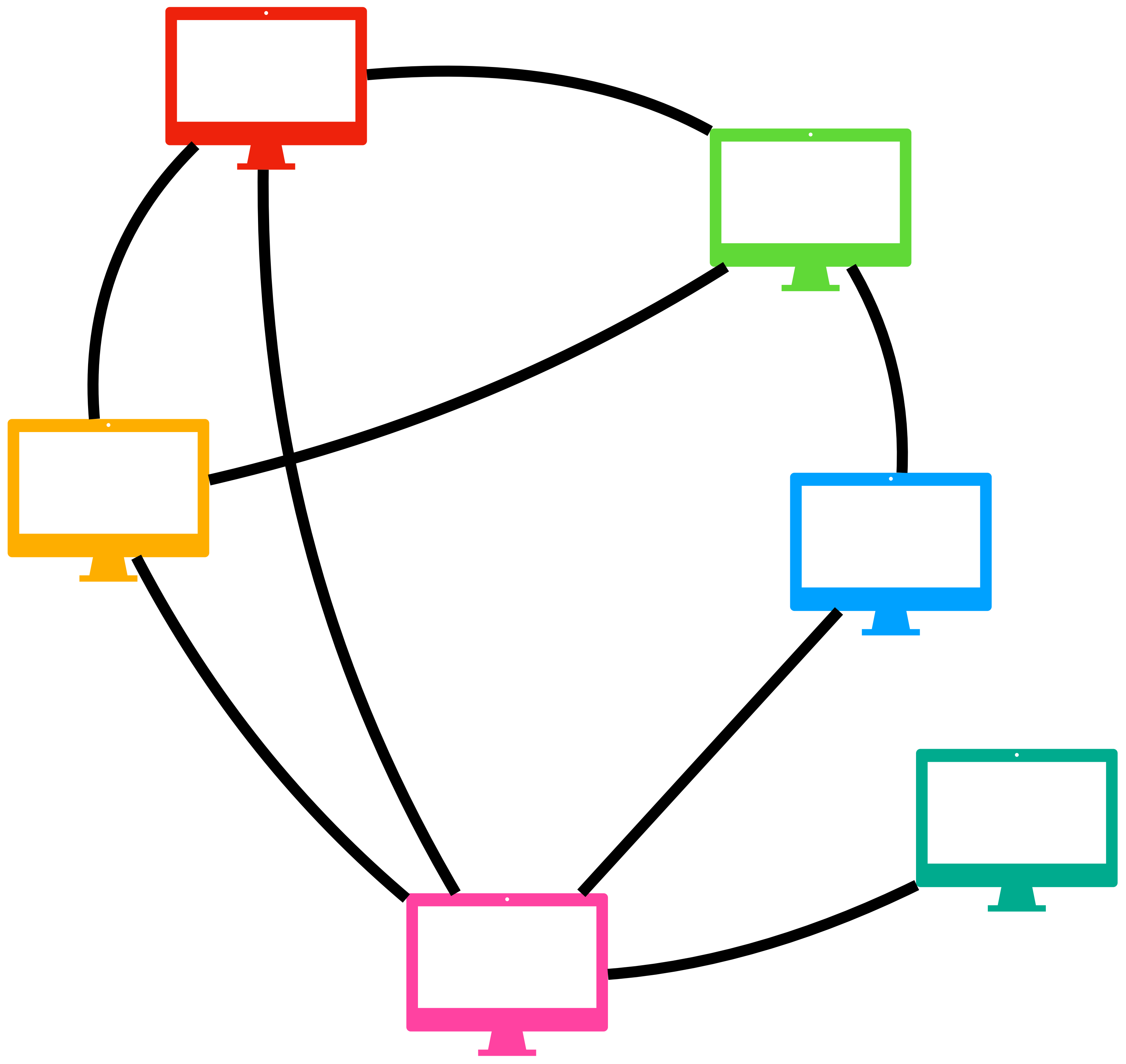


The CONGEST Model

- Other model assumptions:
  - The machines have unique ID's
  - Perfect Synchrony
  - No Faults
  - Lossless Message Passing
  - Infinite Local Computation Power

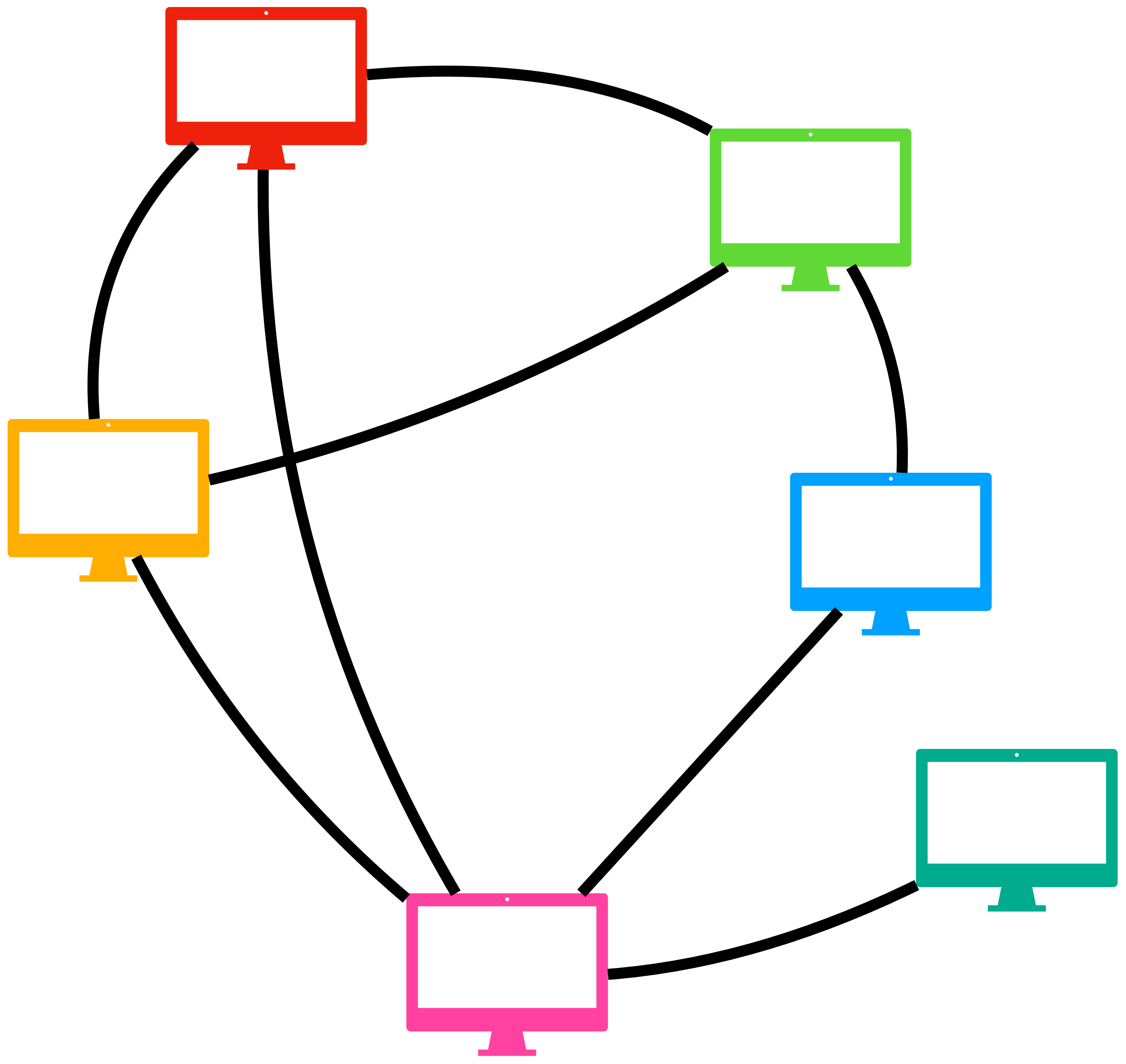


The CONGEST Model



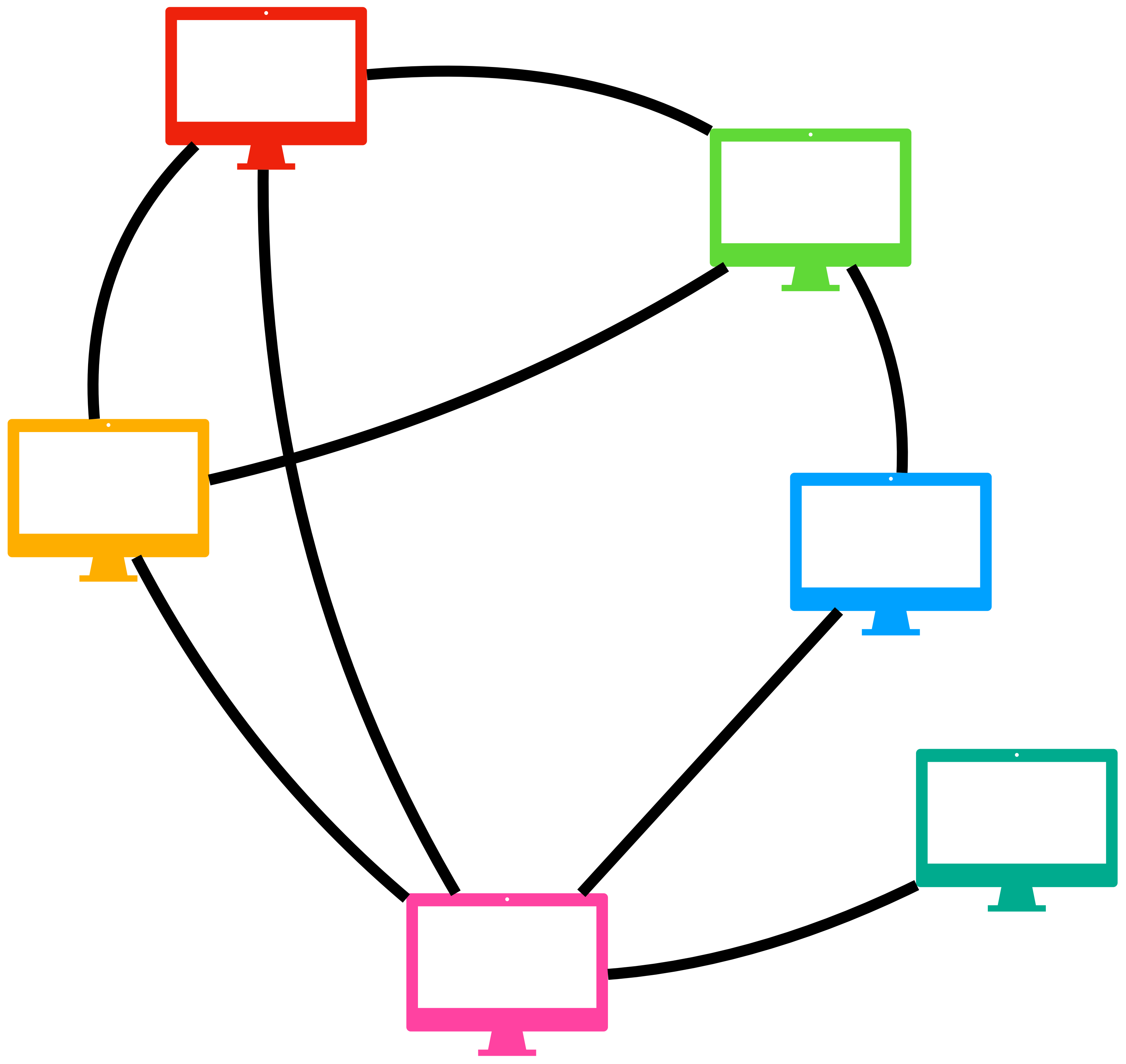
The CONGEST Model

- **Output:** each node will compute a part of the output, eg, pass/fail, its own color.



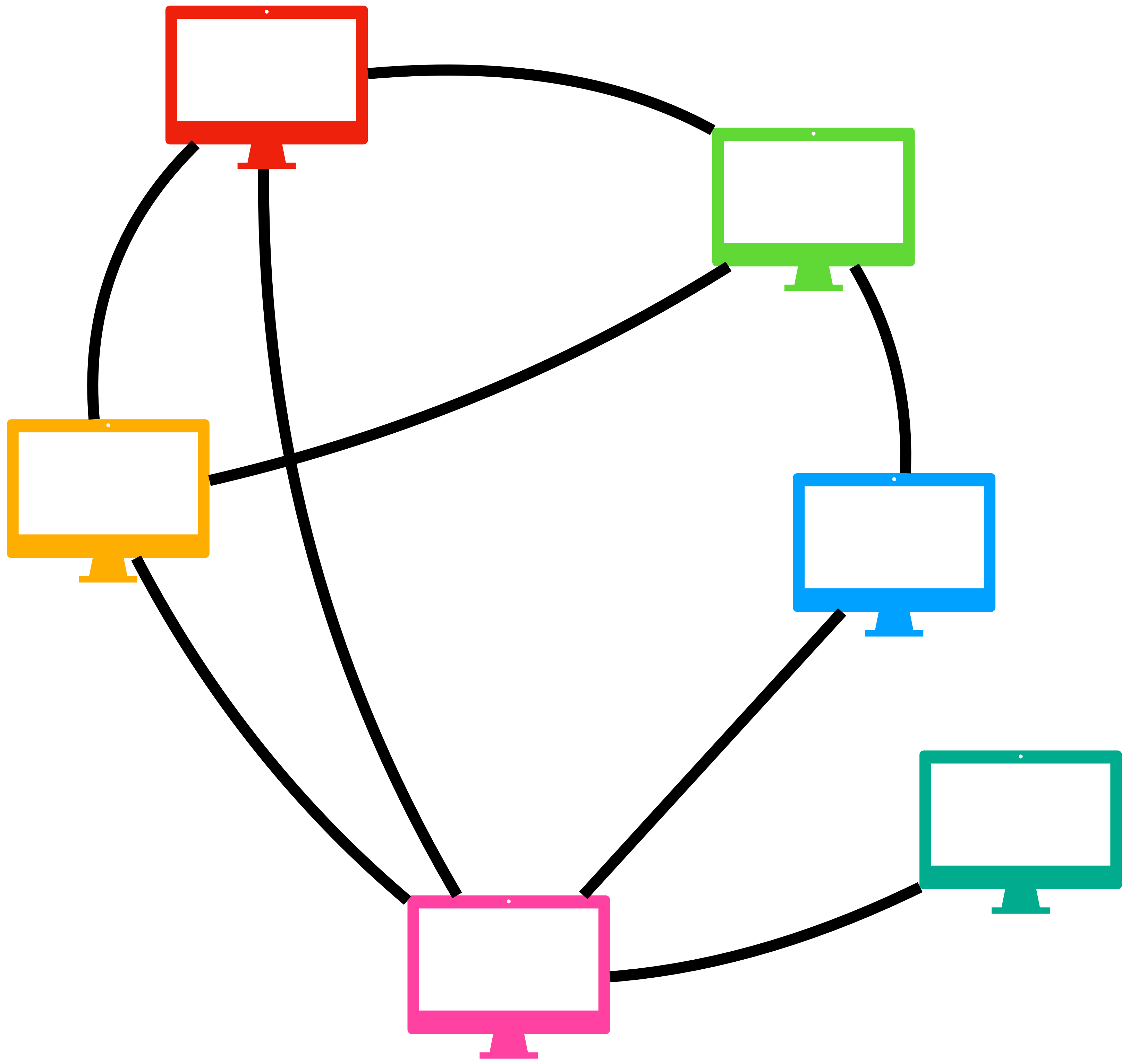
The CONGEST Model

- **Output:** each node will compute a part of the output, eg, pass/fail, its own color.
- **Round Complexity:** total number of rounds used by an algorithm.

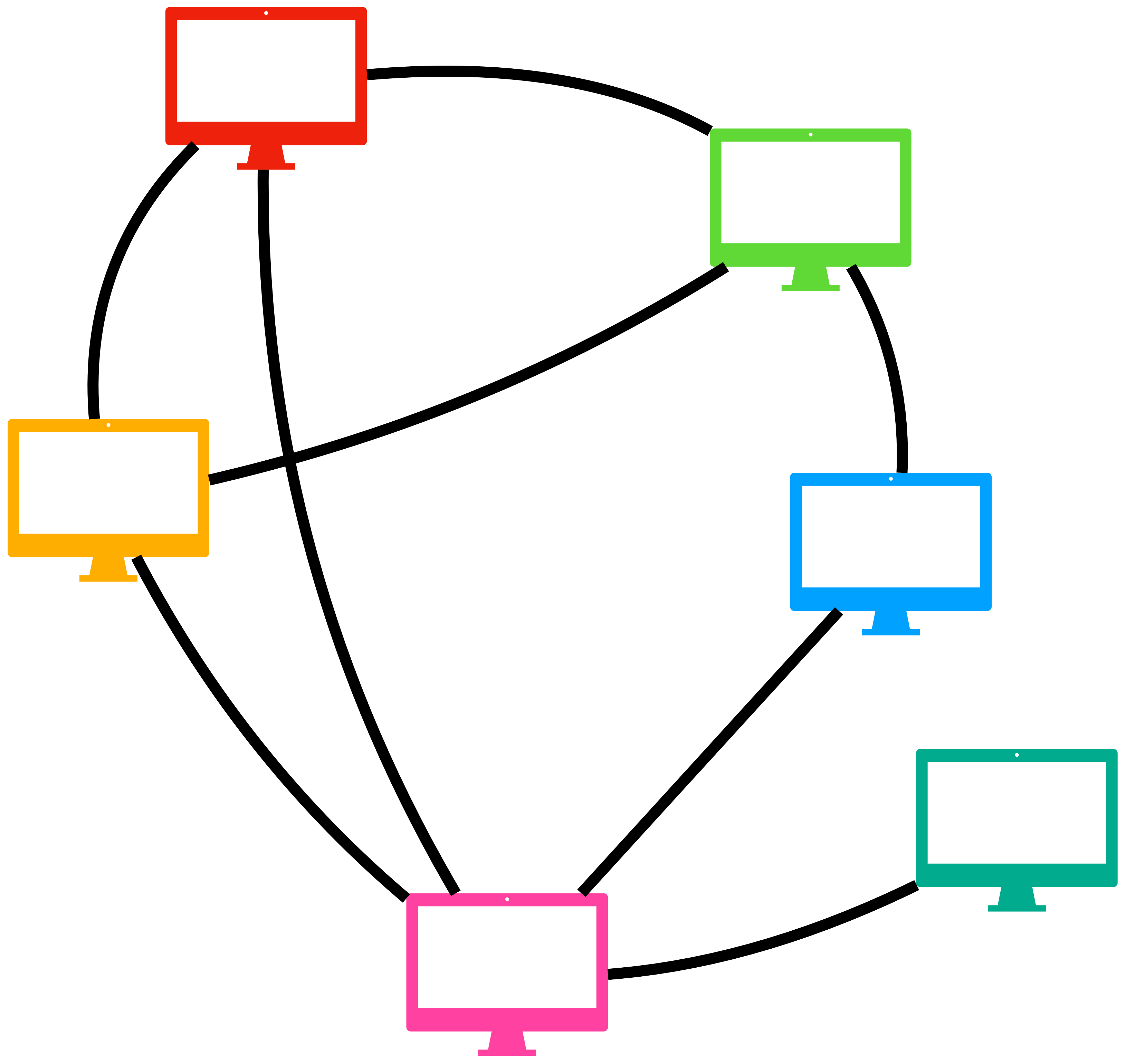


The CONGEST Model

- **Output:** each node will compute a part of the output, eg, pass/fail, its own color.
- **Round Complexity:** total number of rounds used by an algorithm.
- **Message Complexity:** total number of messages sent/received in the network.

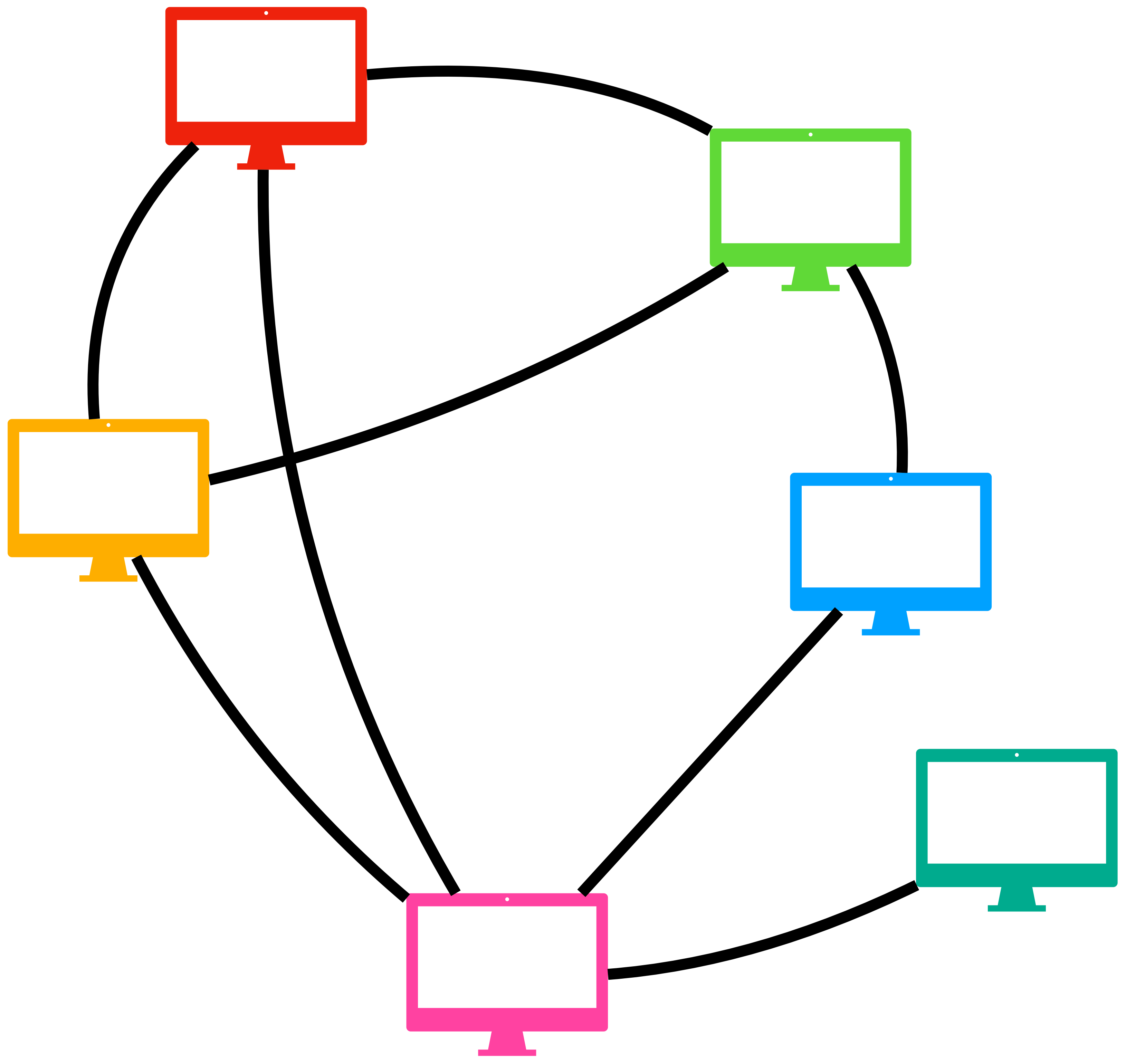


Why CONGEST?



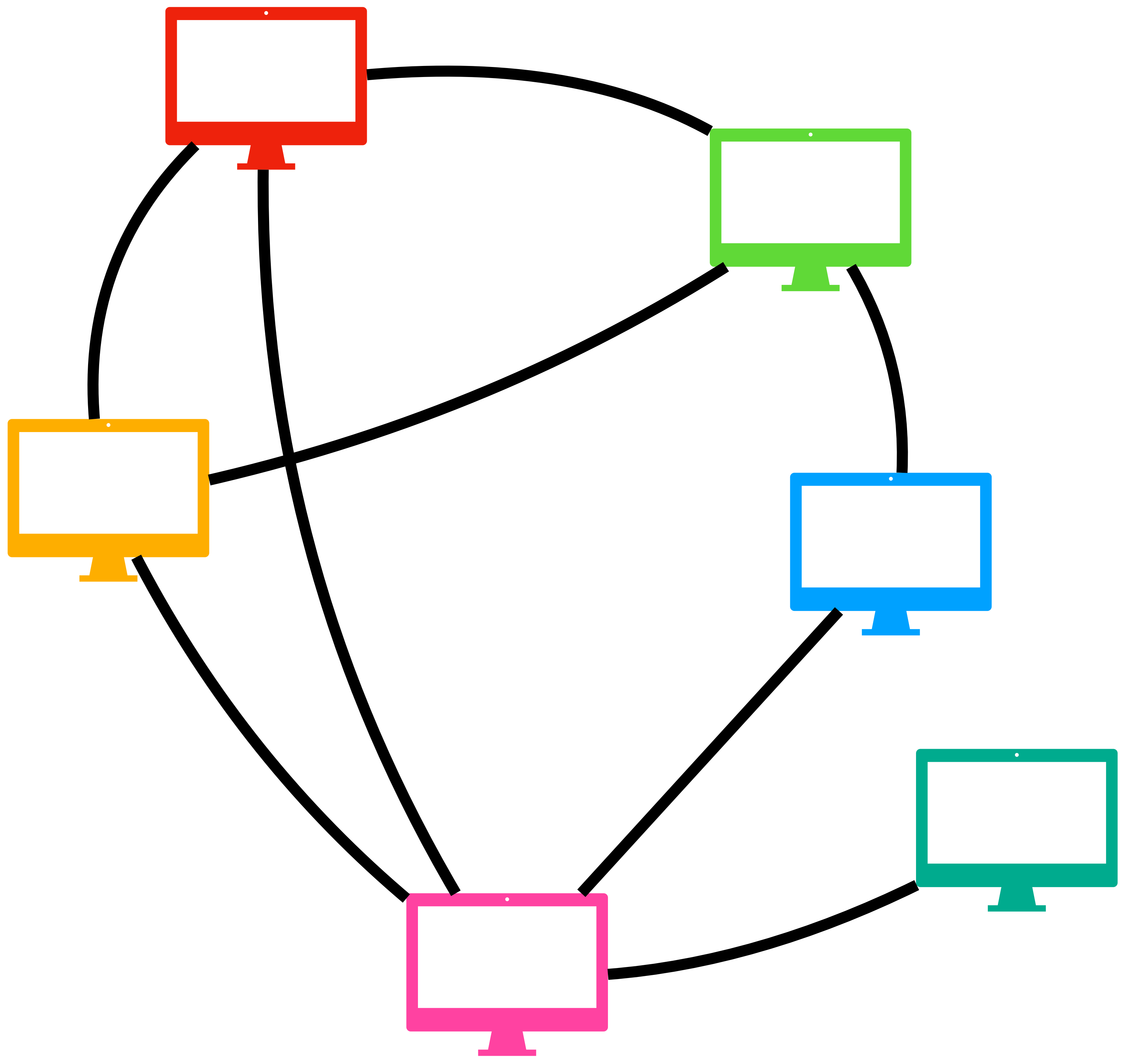
Why CONGEST?

- Node centric computing simplifies algorithm design.



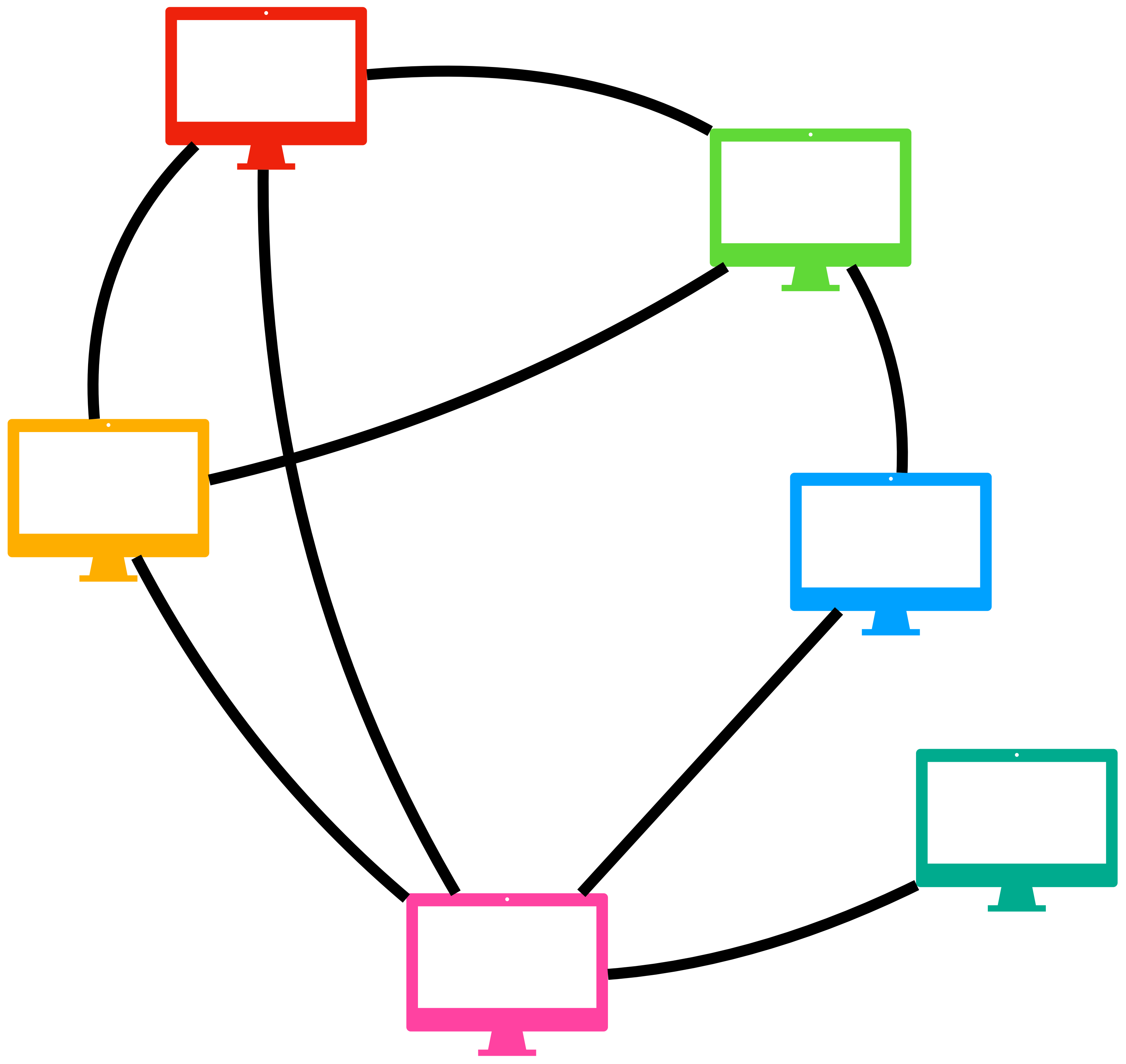
Why CONGEST?

- Node centric computing simplifies algorithm design.
- Captures two important aspects of distributed computing:



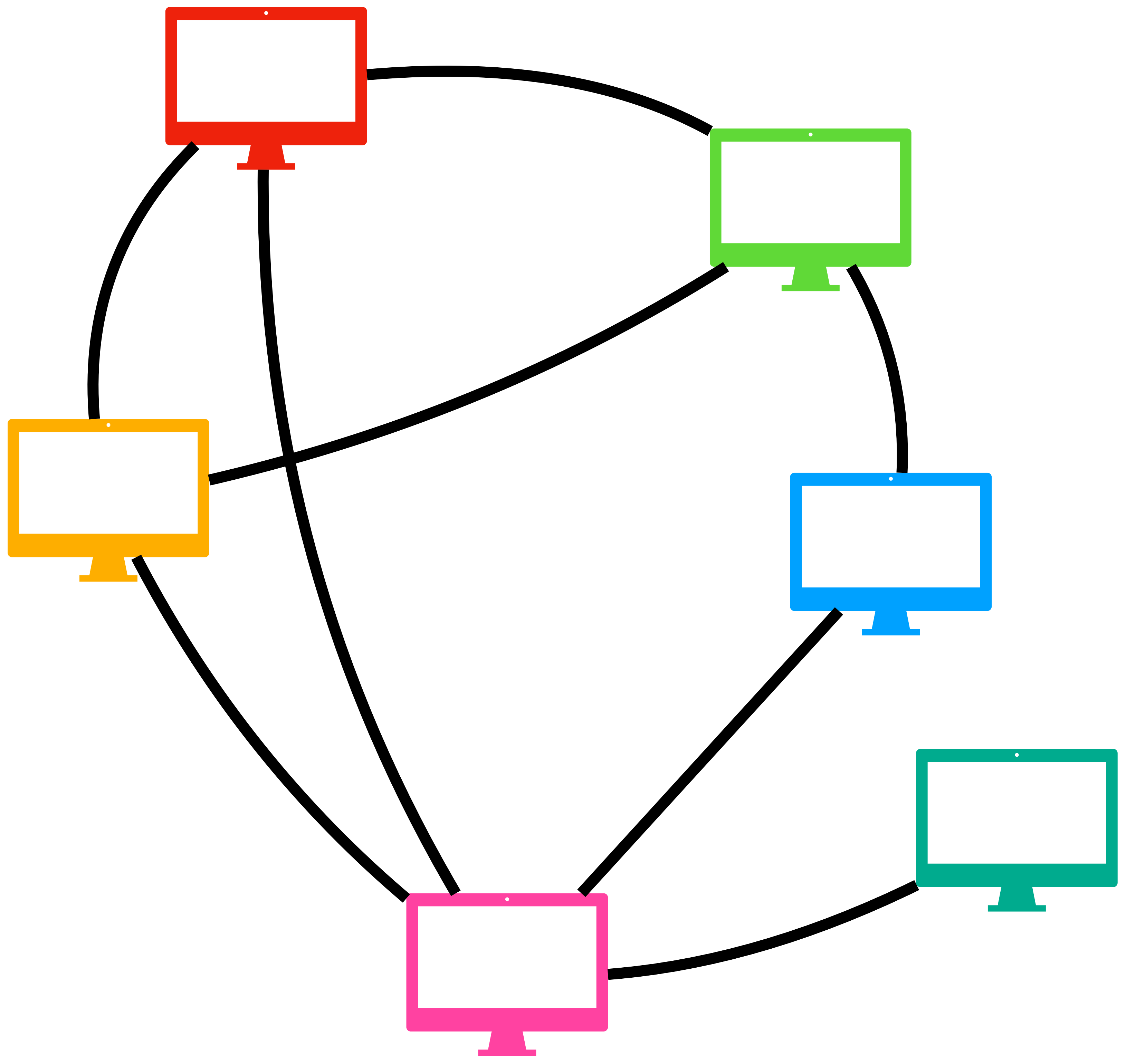
Why CONGEST?

- Node centric computing simplifies algorithm design.
- Captures two important aspects of distributed computing:
    - **Locality:** the information required is far away in the network.



Why CONGEST?

- Node centric computing simplifies algorithm design.
- Captures two important aspects of distributed computing:
  - **Locality:** the information required is far away in the network.
  - **Congestion:** bandwidth constraints cause bottlenecks in the network.



Why not LOCAL?

Why not LOCAL?

- Notion of message complexity also makes sense in the LOCAL model.

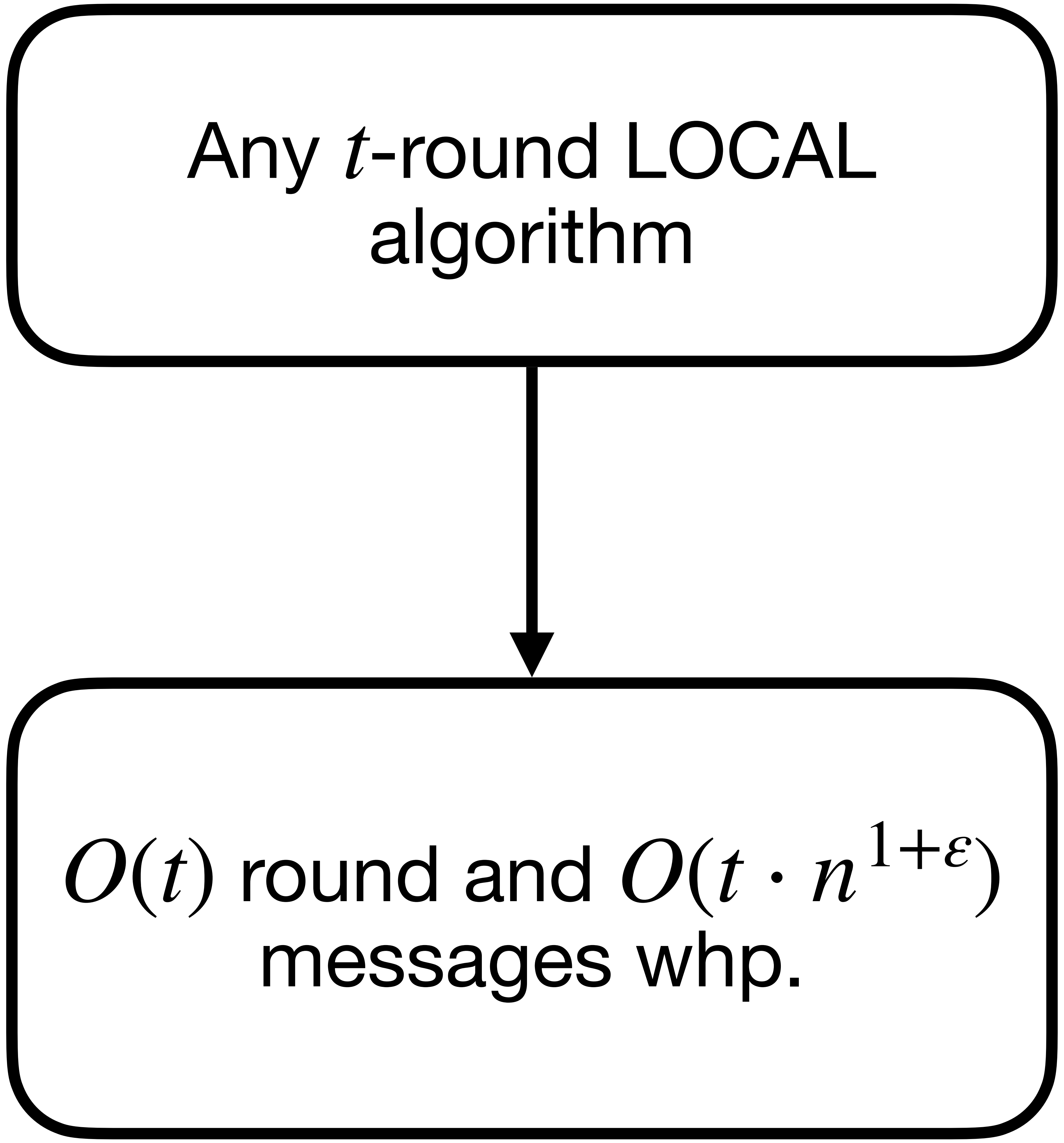
Why not LOCAL?

- Notion of message complexity also makes sense in the LOCAL model.
- Generic message reduction schemes are known. For example [BEI+19].

[BEI+19] Bitton, Emek, Izumi, Kutten. DISC 2019

Why not LOCAL?

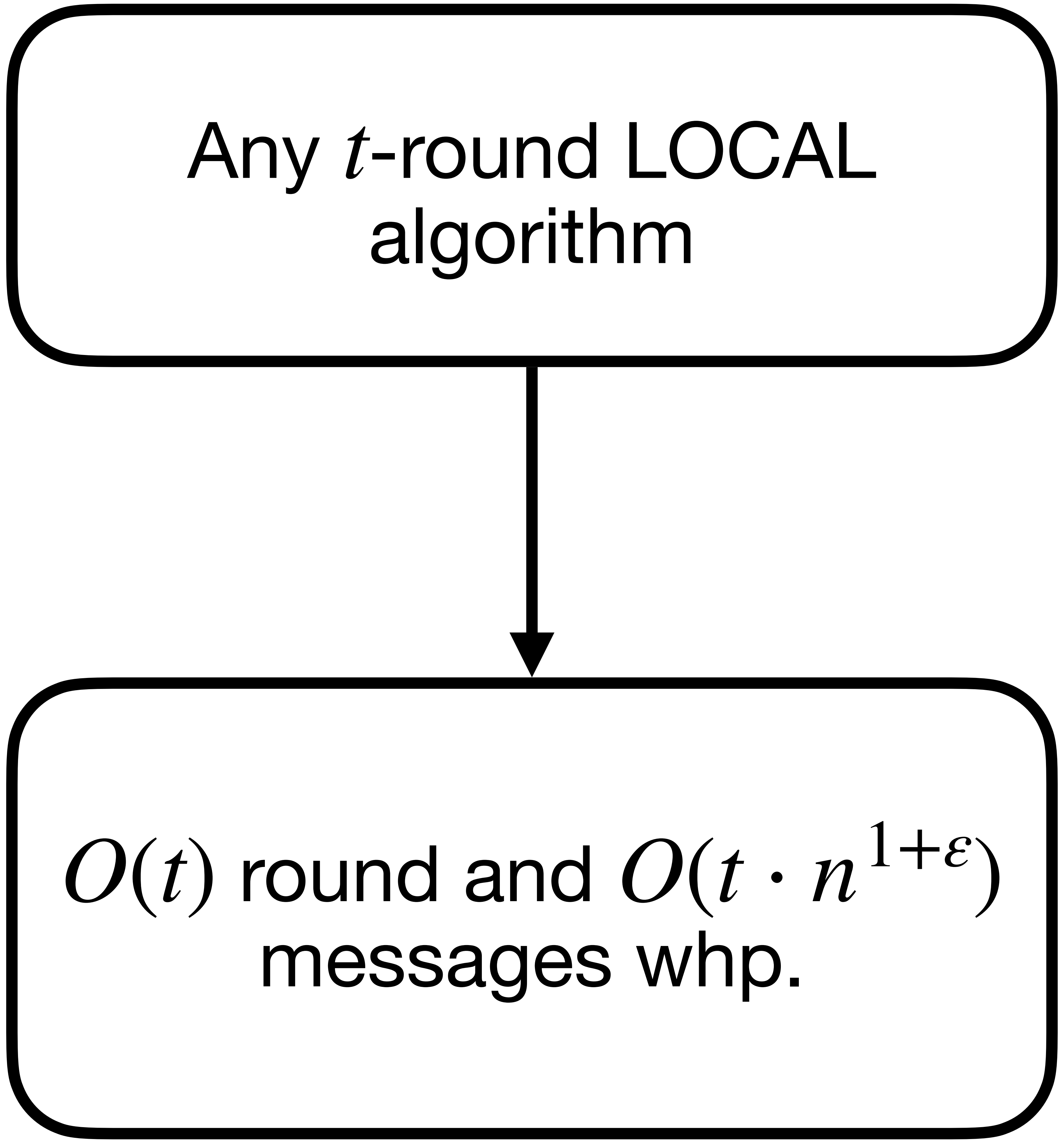
- Notion of message complexity also makes sense in the LOCAL model.
- Generic message reduction schemes are known. For example [BEI+19].



[BEI+19] Bitton, Emek, Izumi, Kutten. DISC 2019

Why not LOCAL?

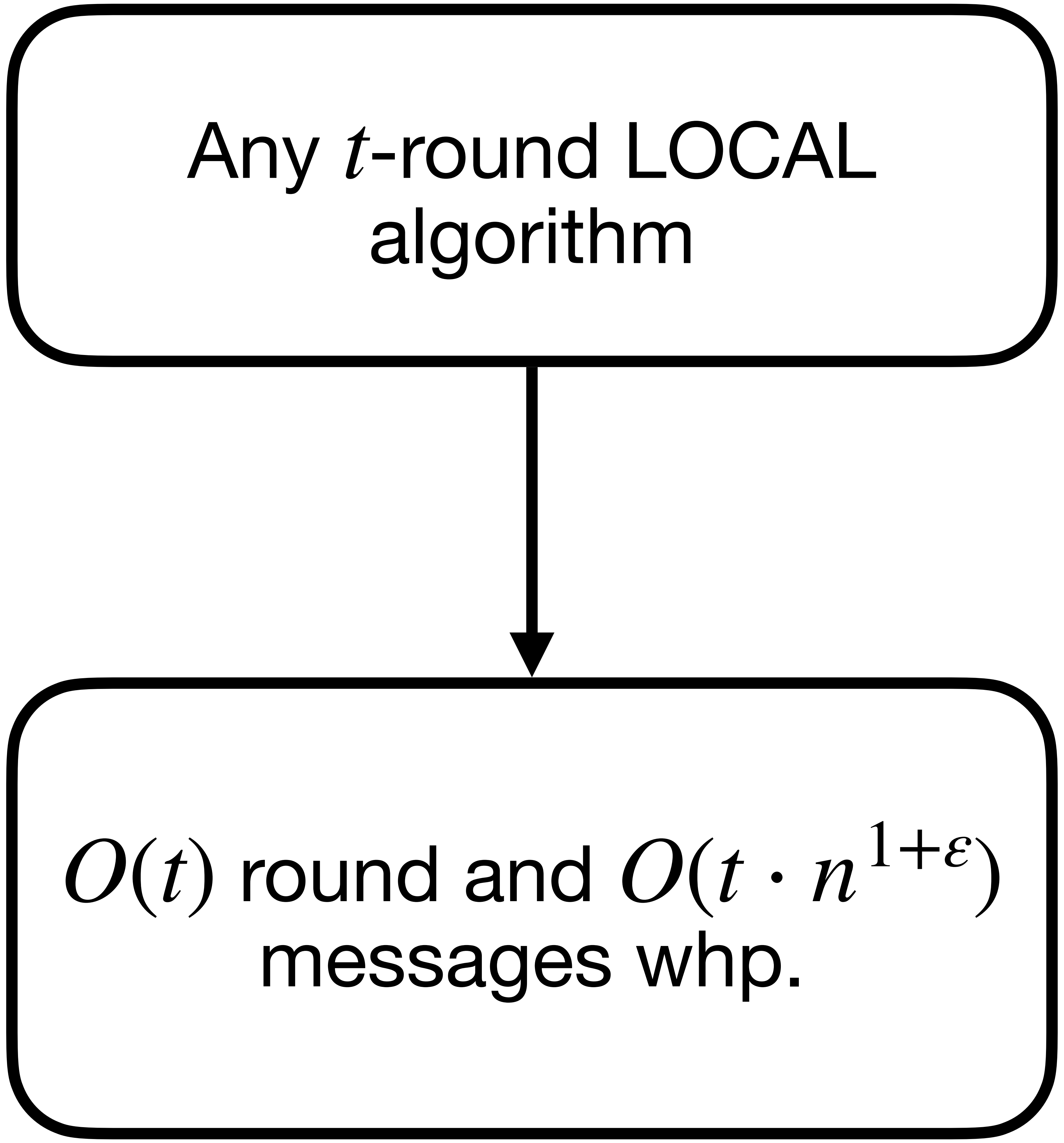
- Notion of message complexity also makes sense in the LOCAL model.
- Generic message reduction schemes are known. For example [BEI+19].
  - Need to send very large sized messages.



[BEI+19] Bitton, Emek, Izumi, Kutten. DISC 2019

Why not LOCAL?

- Notion of message complexity also makes sense in the LOCAL model.
- Generic message reduction schemes are known. For example [BEI+19].
  - Need to send very large sized messages.
- Will not be the primary focus in this talk.



[BEI+19] Bitton, Emek, Izumi, Kutten. DISC 2019

Why Message Complexity?

Why Message Complexity?

- Studying the interplay between round and message complexity is a fundamental theoretical question.

Why Message Complexity?

- Studying the interplay between round and message complexity is a fundamental theoretical question.
- Communication is costly and consumes a lot of energy.

Why Message Complexity?

- Studying the interplay between round and message complexity is a fundamental theoretical question.
- Communication is costly and consumes a lot of energy.
- Lead to efficient algorithms in big data computing models.

Why Message Complexity?

- Studying the interplay between round and message complexity is a fundamental theoretical question.
- Communication is costly and consumes a lot of energy.
- Lead to efficient algorithms in big data computing models.
    - k-machine model
    - Massively Parallel Computing (MPC)

Why Message Complexity?

- Studying the interplay between round and message complexity is a fundamental theoretical question.
- Communication is costly and consumes a lot of energy.
- Lead to efficient algorithms in big data computing models.
    - k-machine model
    - Massively Parallel Computing (MPC)

Broadcast

Broadcast

- One node u has a special message M of $O(\log n)$ bits.

Broadcast

- One node u has a special message M of $O(\log n)$ bits.
- All nodes connected to u must receive M at least once.

Broadcast

- One node u has a special message M of $O(\log n)$ bits.
- All nodes connected to u must receive M at least once.

- Flooding Algorithm:
  - Node u sends M to all its neighbors.
  - If v receives M for the first time, it sends M to all neighbors.

Analysis of Flooding Algorithm

Analysis of Flooding Algorithm

- Takes D rounds and $2m$ messages.

Analysis of Flooding Algorithm

- Takes D rounds and $2m$ messages.
- Is this the best we can do for broadcast?

Analysis of Flooding Algorithm

- Takes D rounds and $2m$ messages.
- Is this the best we can do for broadcast?
- Broadcast must take $\Omega(D)$ rounds.

Analysis of Flooding Algorithm

- Takes D rounds and $2m$ messages.
- Is this the best we can do for broadcast?
- Broadcast must take $\Omega(D)$ rounds.
    - Otherwise there are not enough rounds for M to reach all nodes.

Analysis of Flooding Algorithm

- Takes D rounds and $2m$ messages.
- Is this the best we can do for broadcast?
- Broadcast must take $\Omega(D)$ rounds.
  - Otherwise there are not enough rounds for M to reach all nodes.
- How do you formally prove that broadcast requires $\Omega(m)$ messages?

Initial Knowledge

Initial Knowledge

KT-0

Nodes just know their own ID

Initial Knowledge

KT-0

Nodes just know their own ID

KT-1

Nodes know ID’s of neighbors

Initial Knowledge

KT-0

Nodes just know their own ID

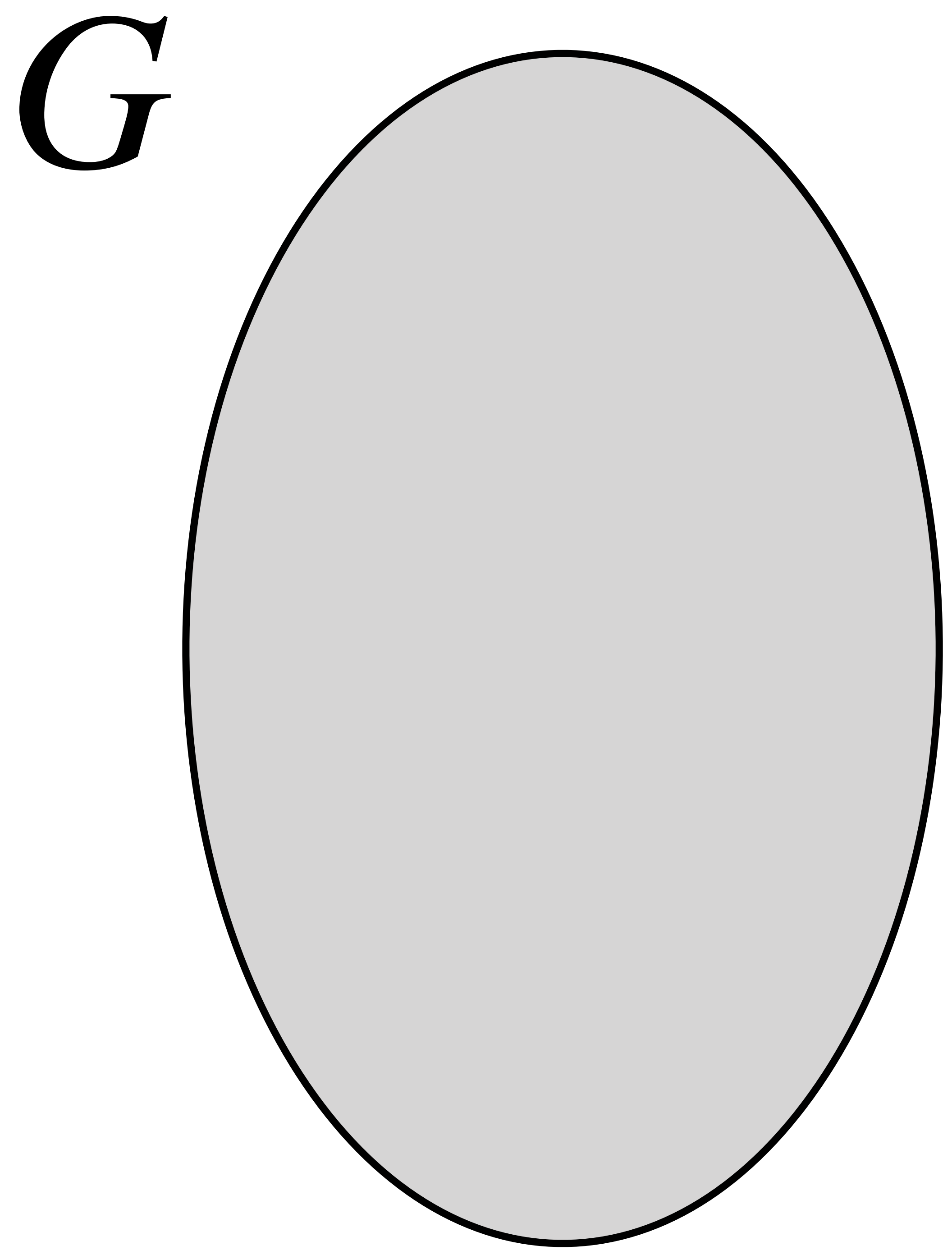
KT-1

Nodes know ID's of neighbors

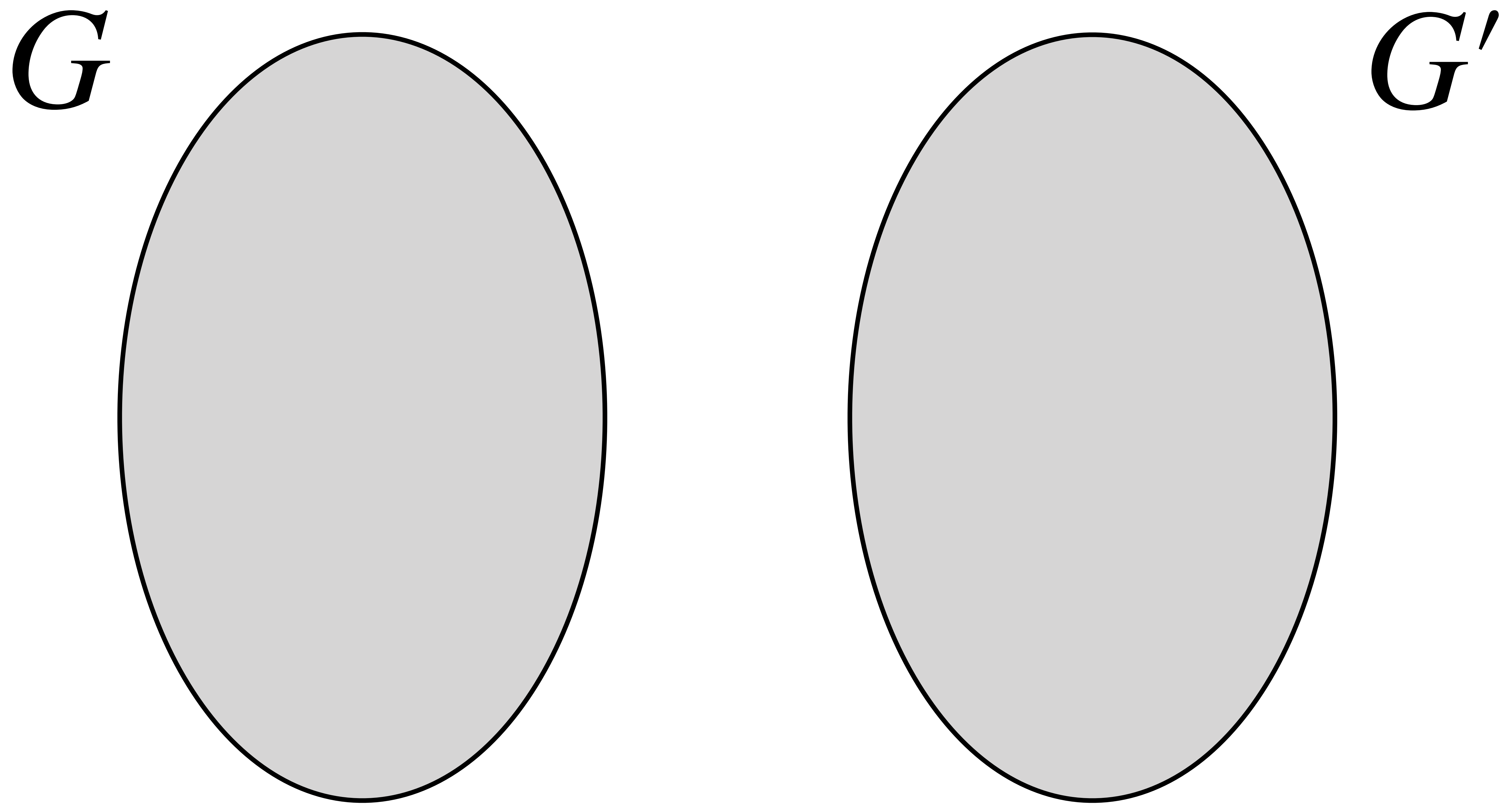
- Going from KT-0 to KT-1 requires only one round, but $O(m)$ messages.

KT-0 Indistinguishability

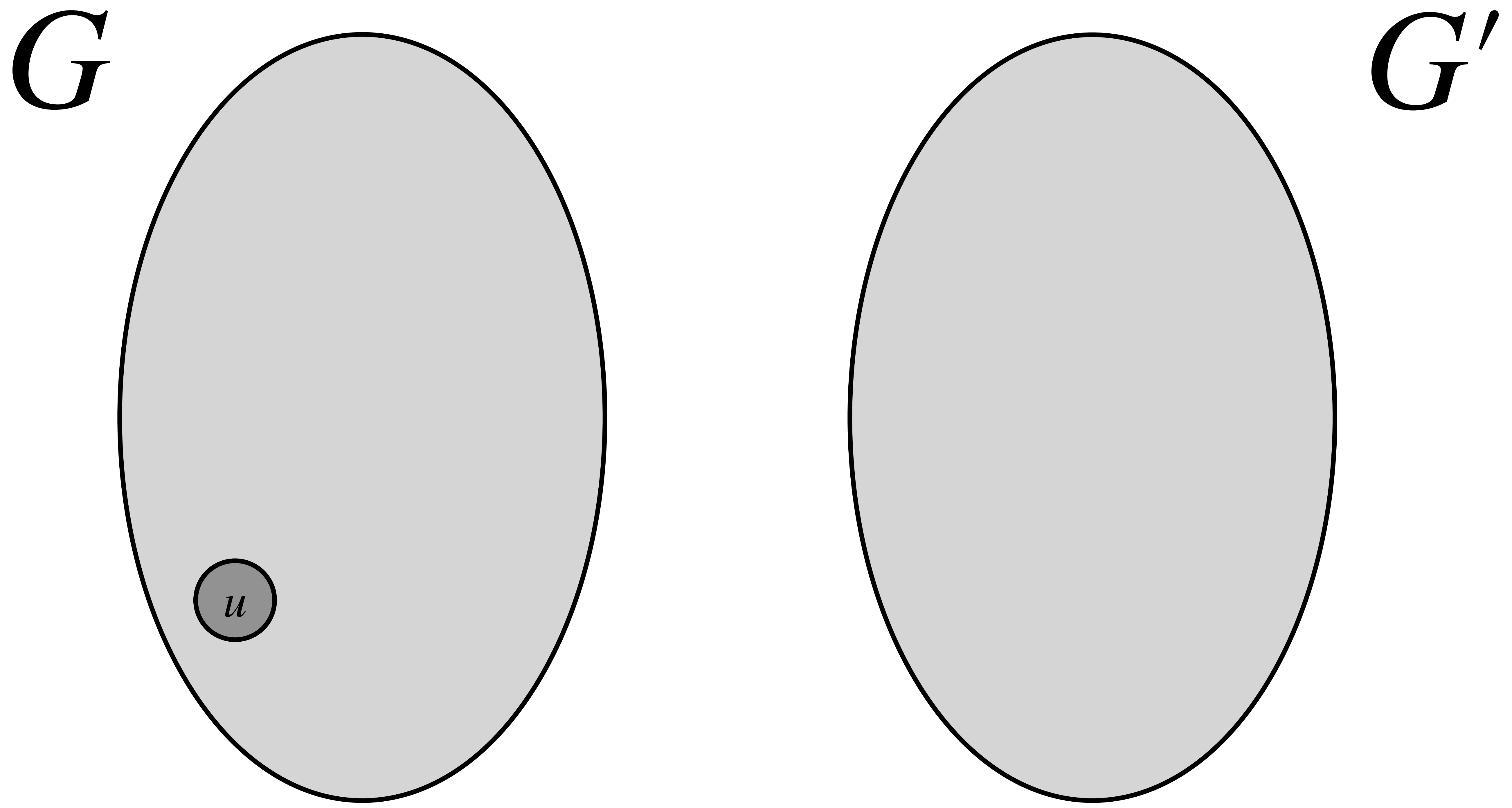
KT-0 Indistinguishability



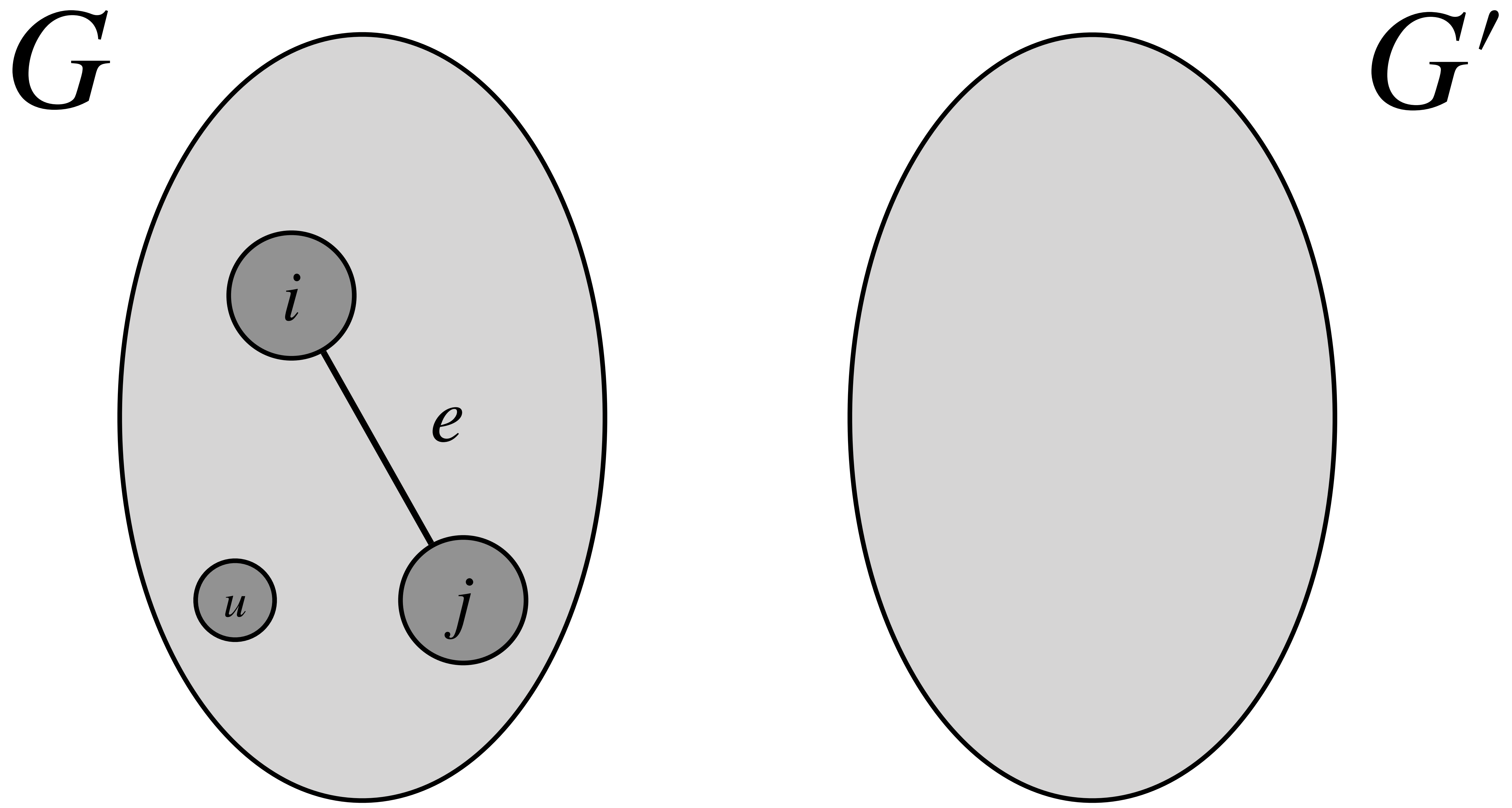
KT-0 Indistinguishability



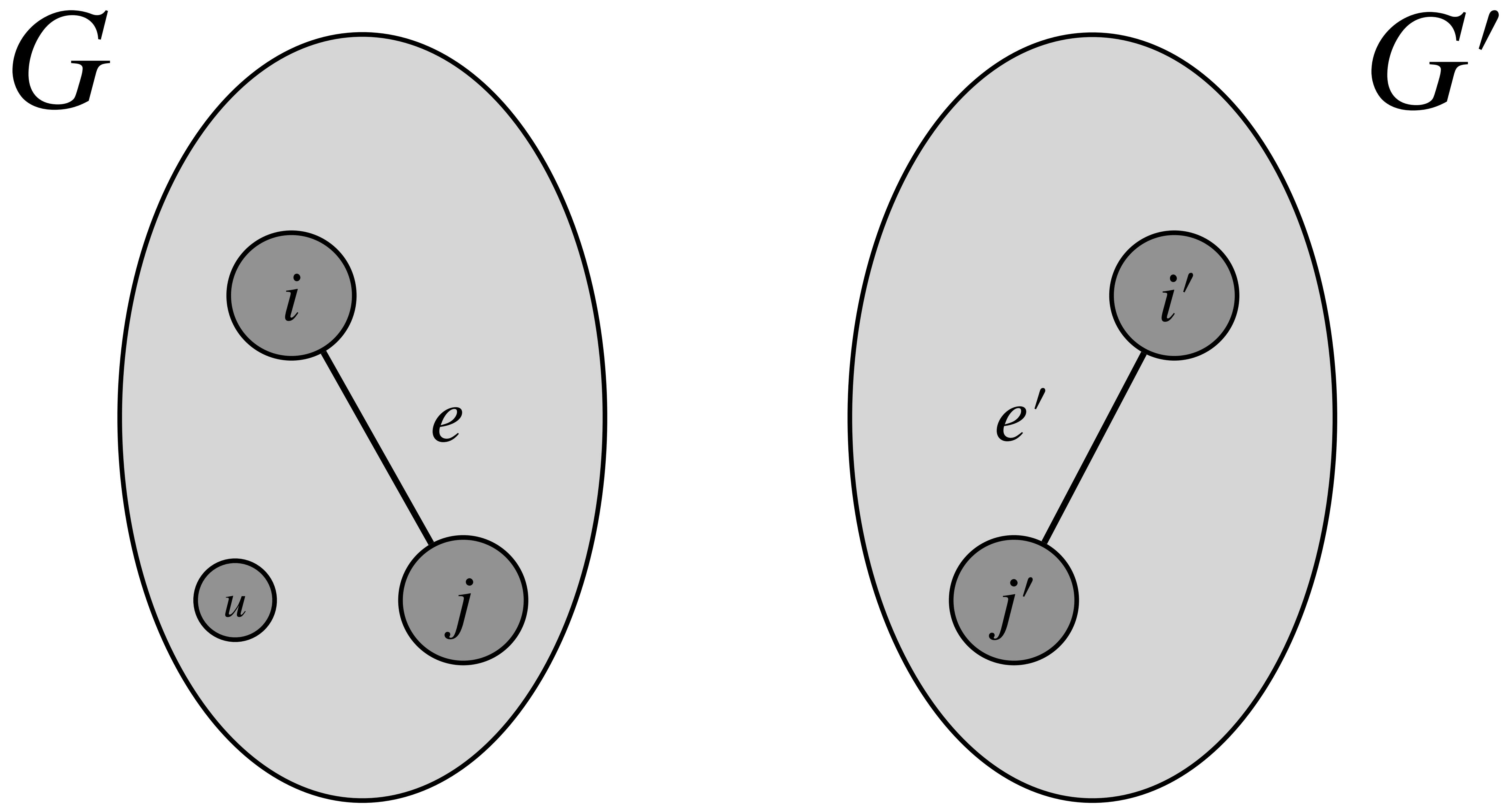
KT-0 Indistinguishability



KT-0 Indistinguishability

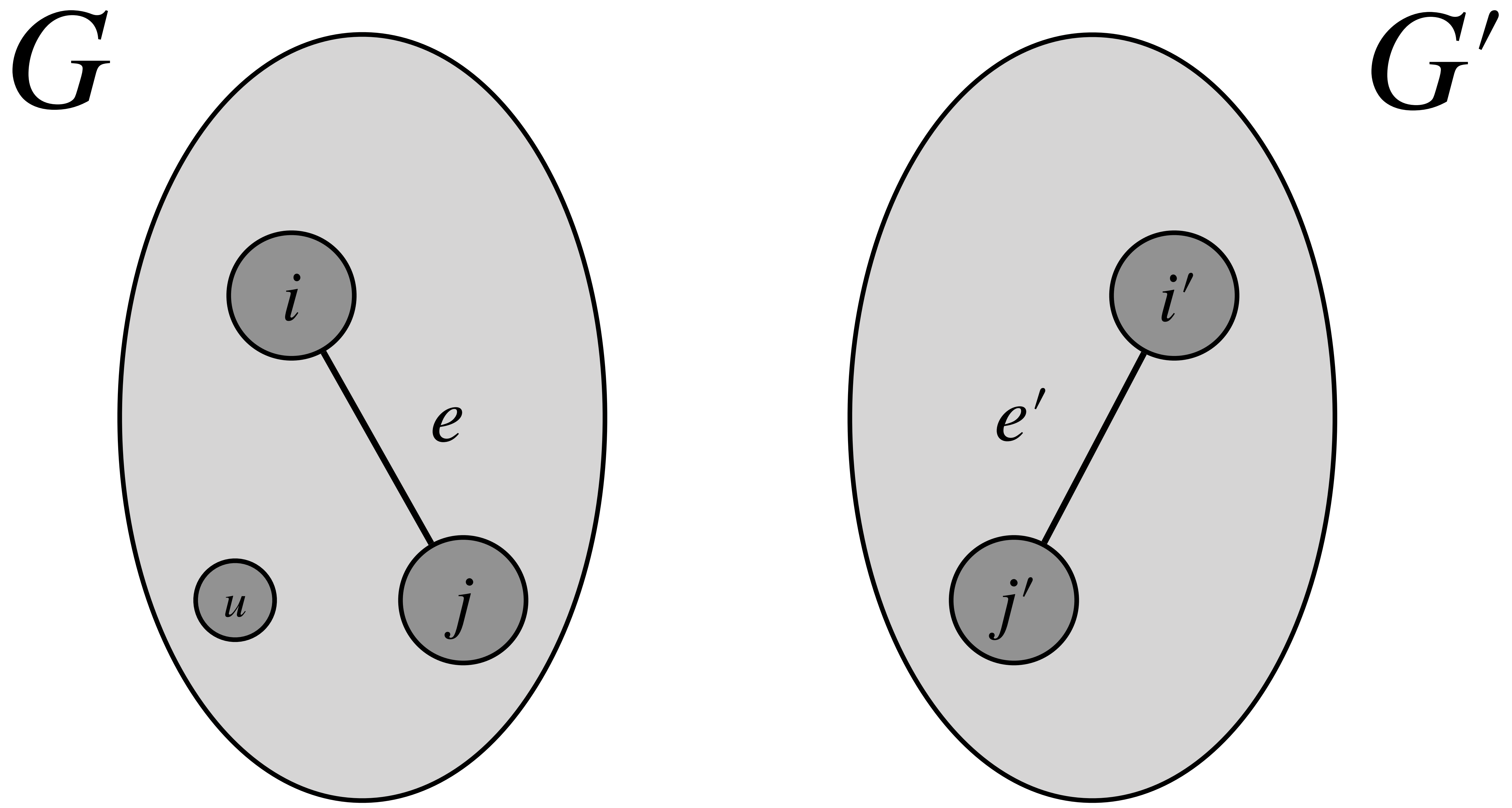


KT-0 Indistinguishability



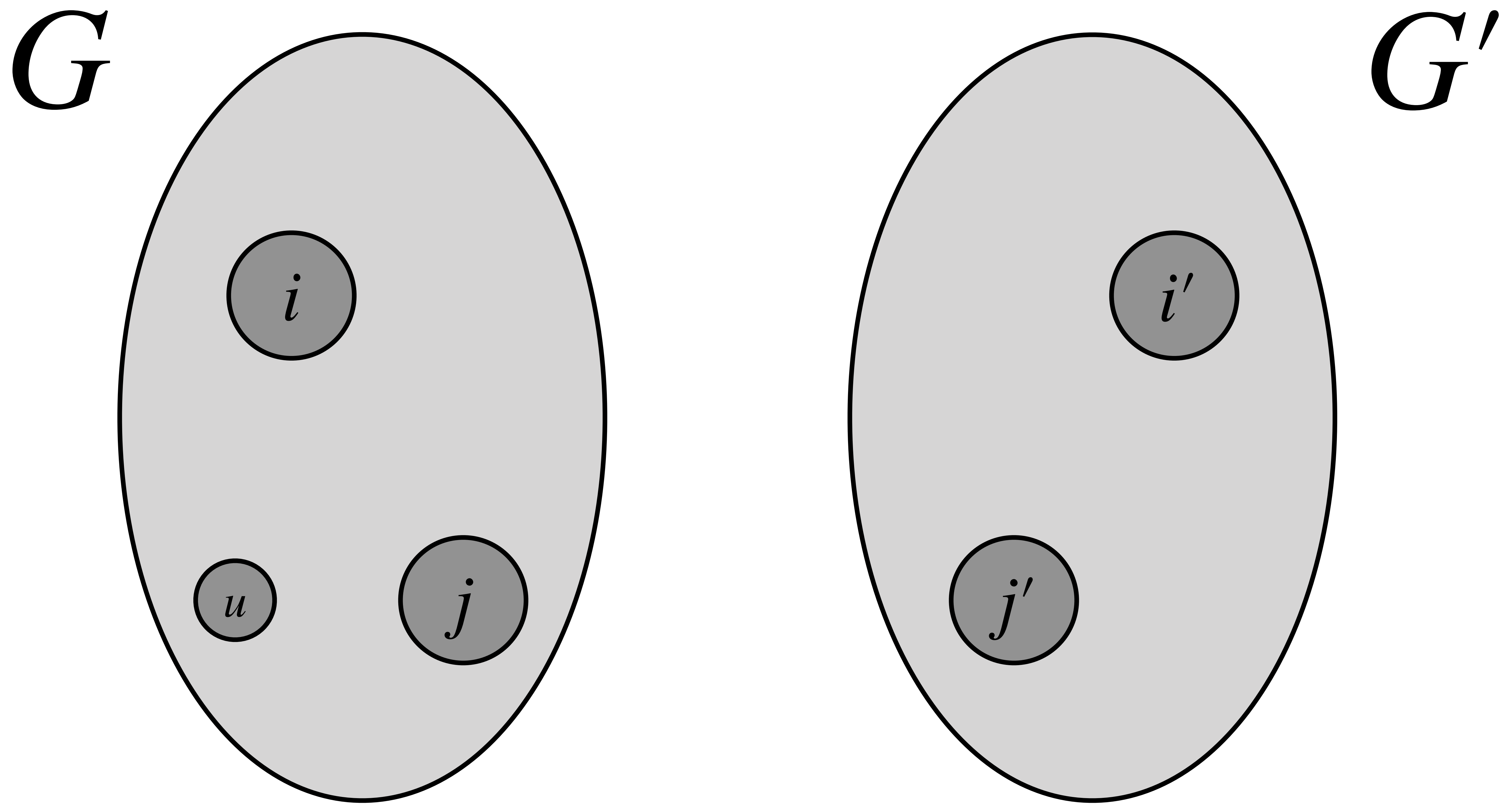
KT-0 Indistinguishability

- Assume no message passes through e and e'.



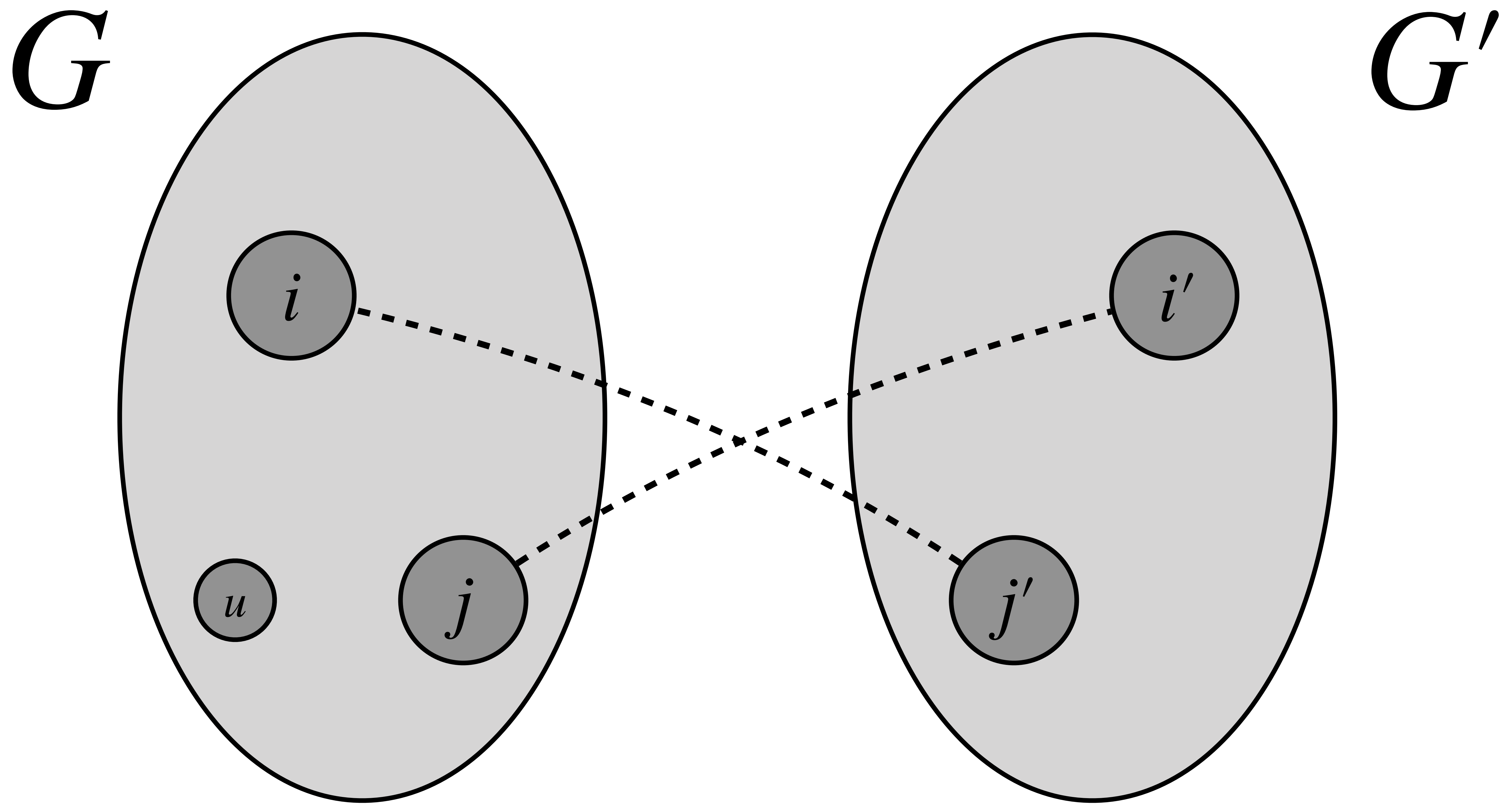
KT-0 Indistinguishability

- Assume no message passes through e and e'.



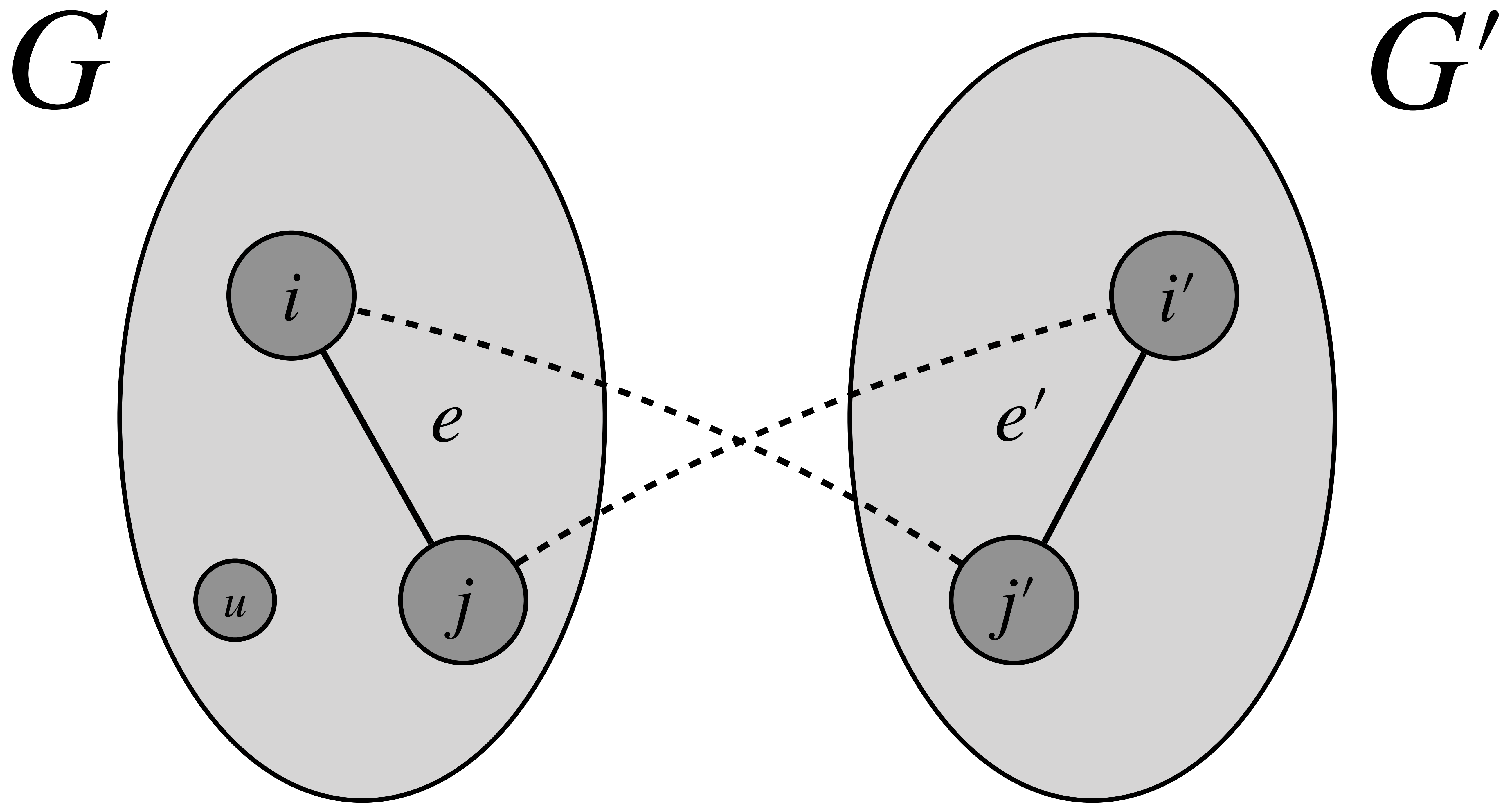
KT-0 Indistinguishability

- Assume no message passes through e and e'.



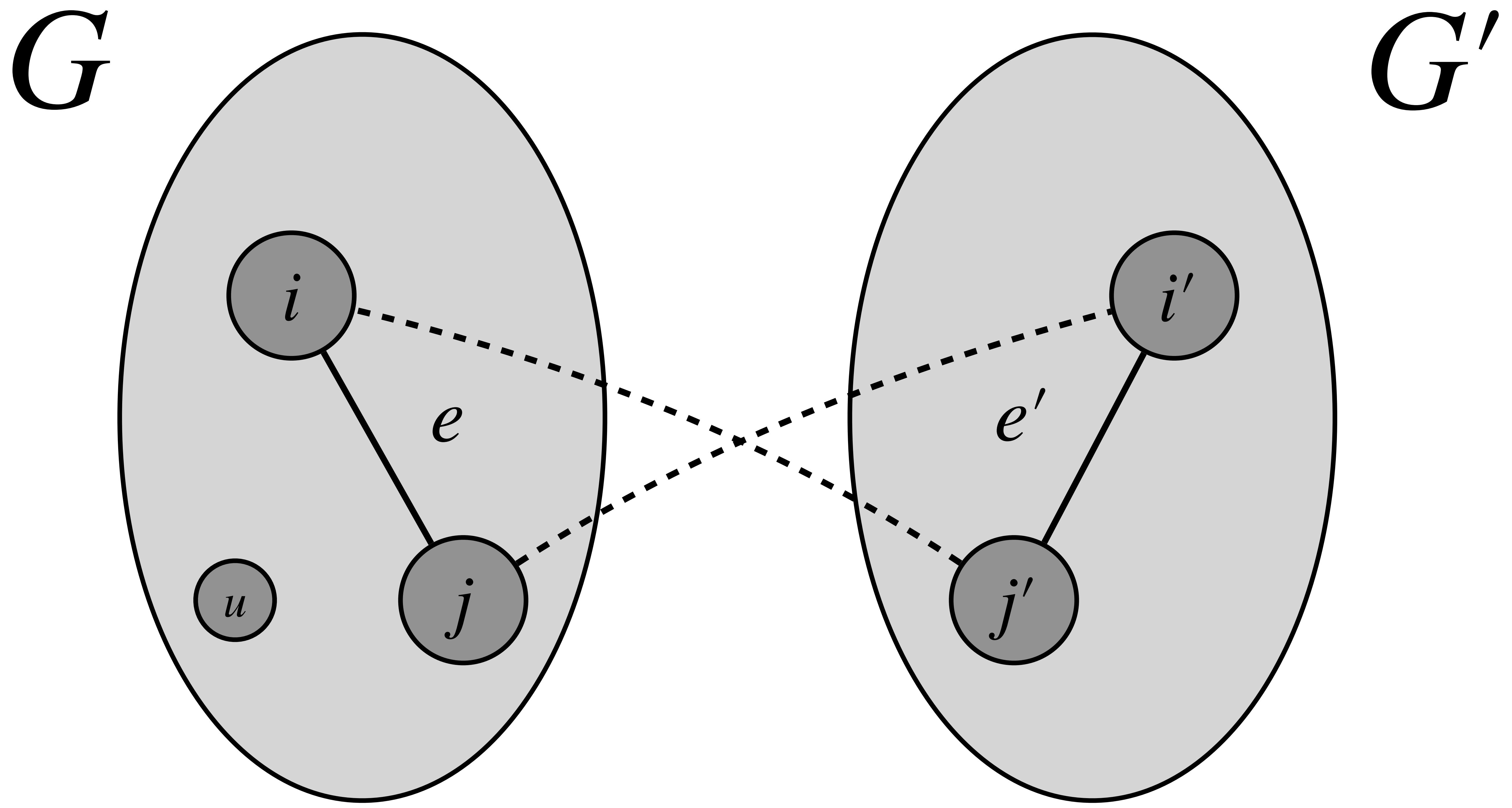
KT-0 Indistinguishability

- Assume no message passes through e and e'.



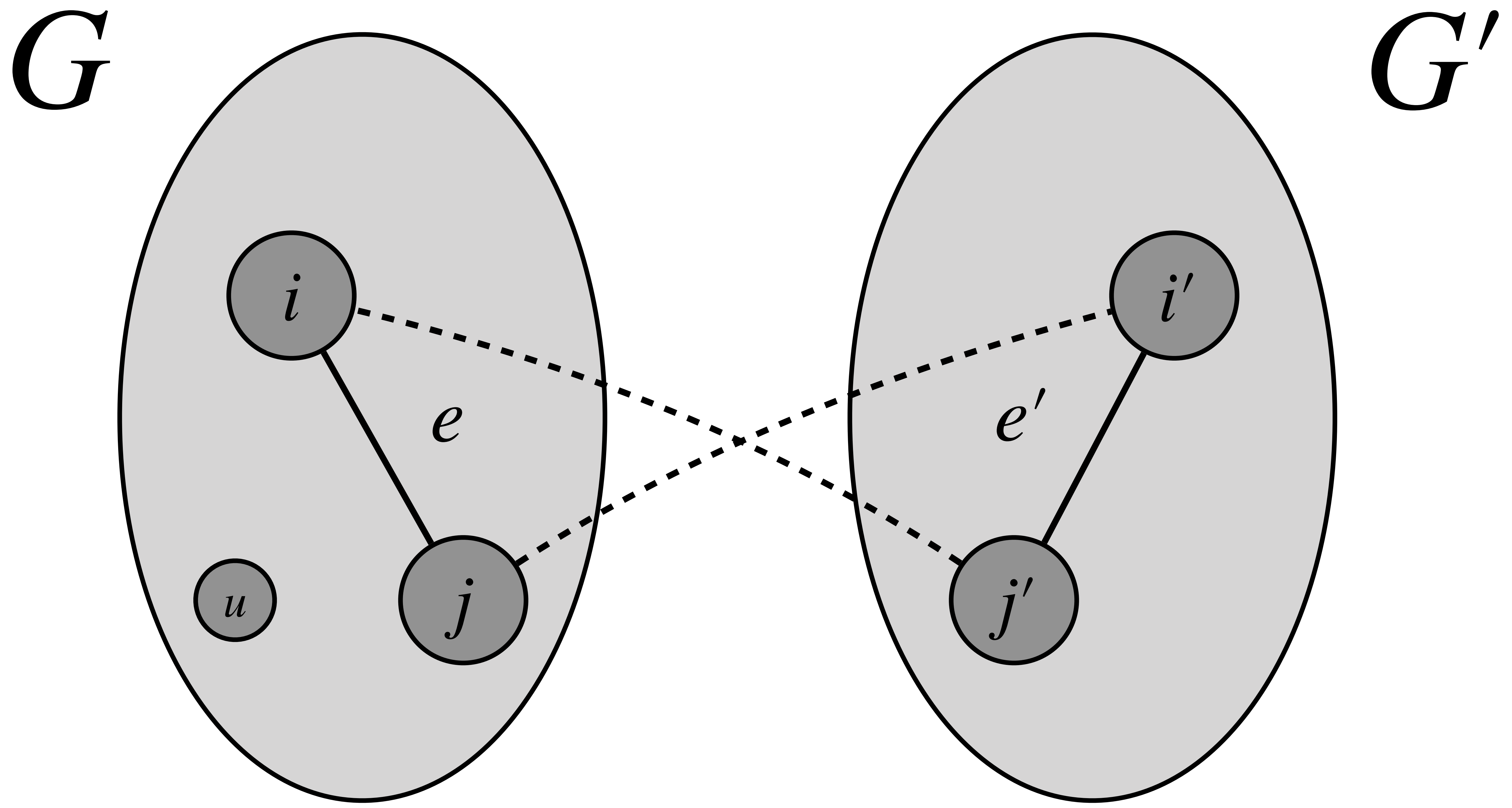
KT-0 Indistinguishability

- Assume no message passes through e and e'.
- For every pair of edges, at least one must be used to send a message.



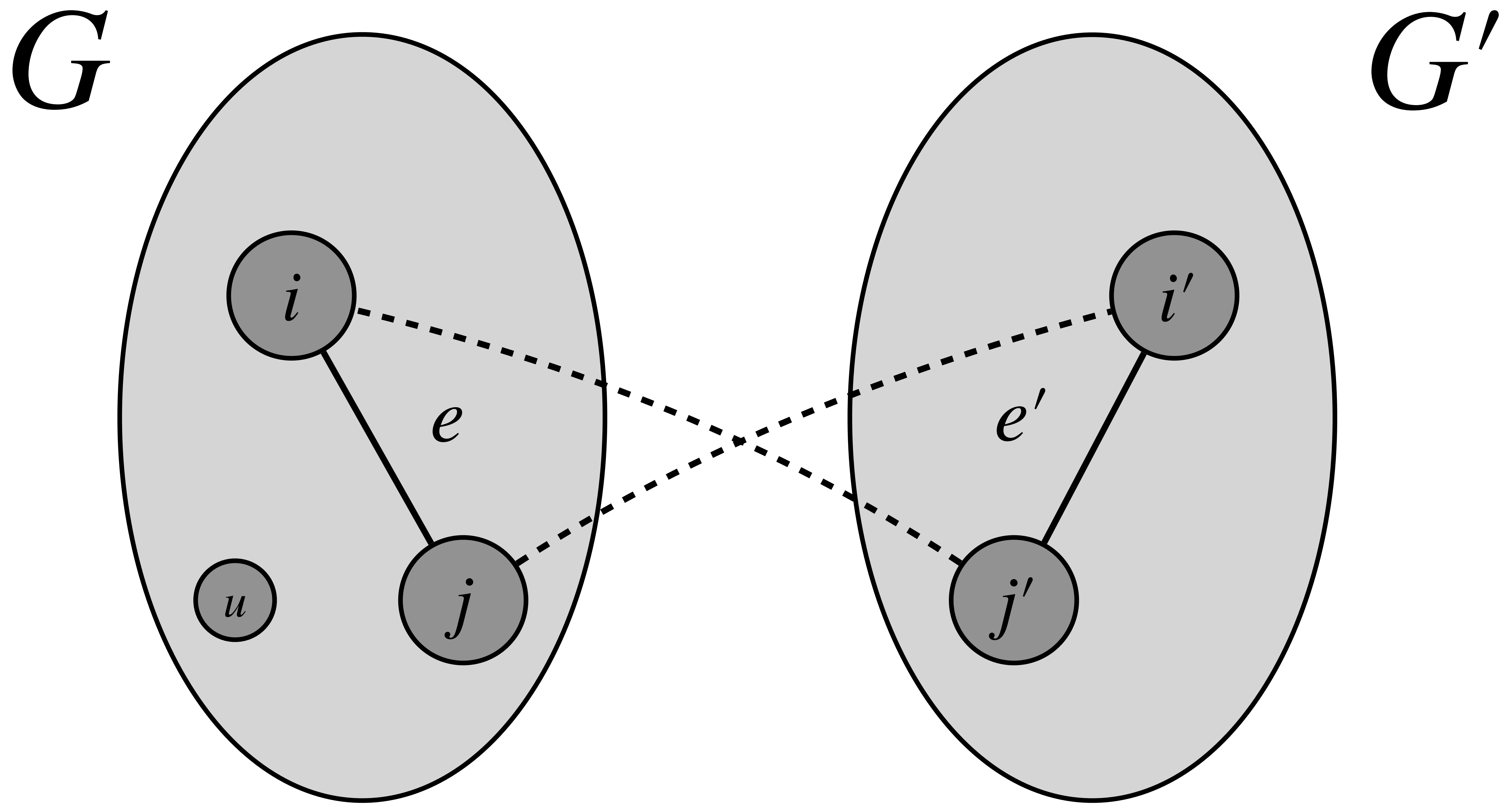
KT-0 Indistinguishability

- Assume no message passes through e and e'.
- For every pair of edges, at least one must be used to send a message.
- Therefore, message complexity of broadcast is $\Omega(m)$.

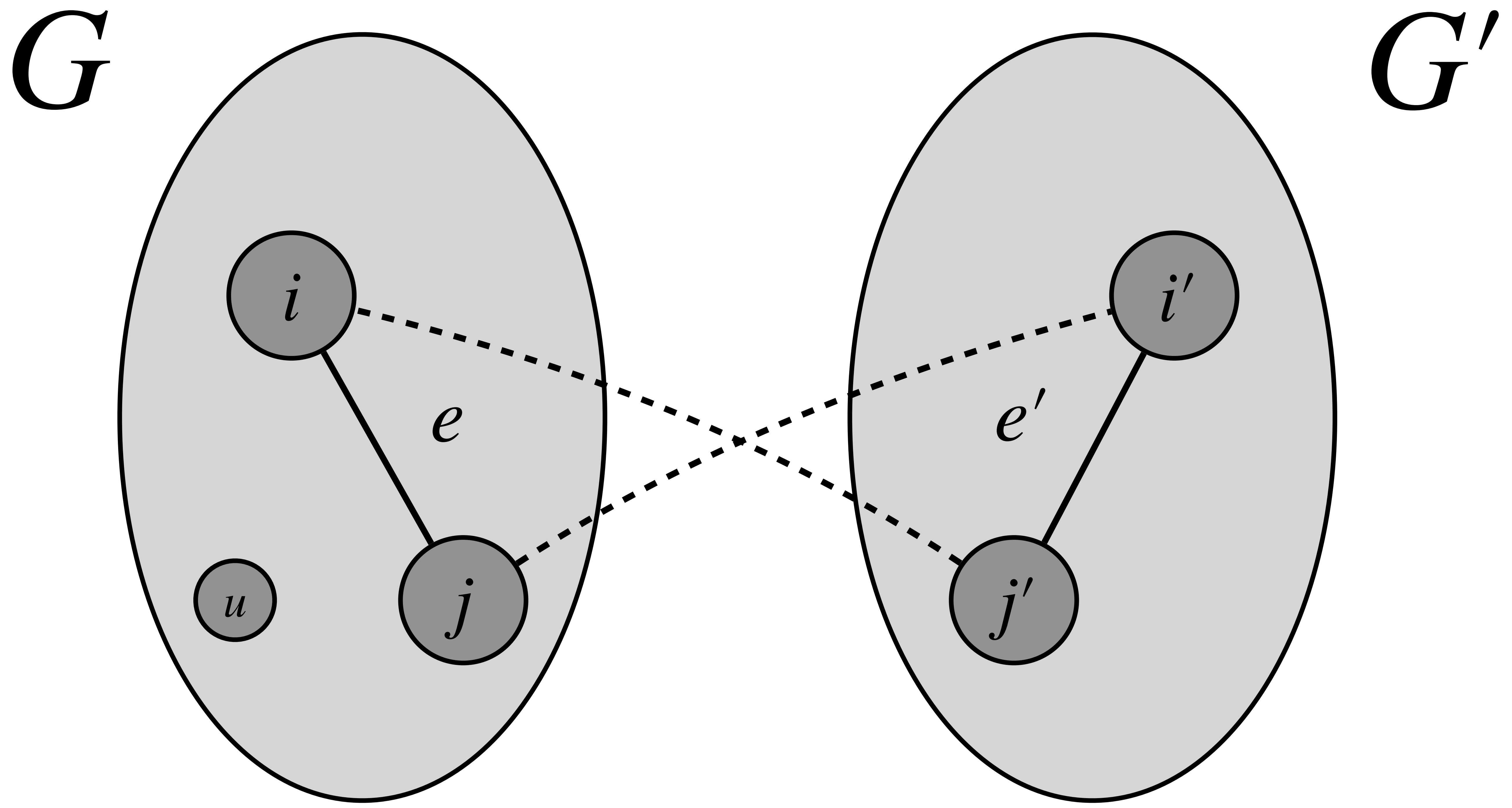


KT-0 Indistinguishability

- Assume no message passes through e and e'.
- For every pair of edges, at least one must be used to send a message.
- Therefore, message complexity of broadcast is $\Omega(m)$.
- Even works with infinite bandwidth!

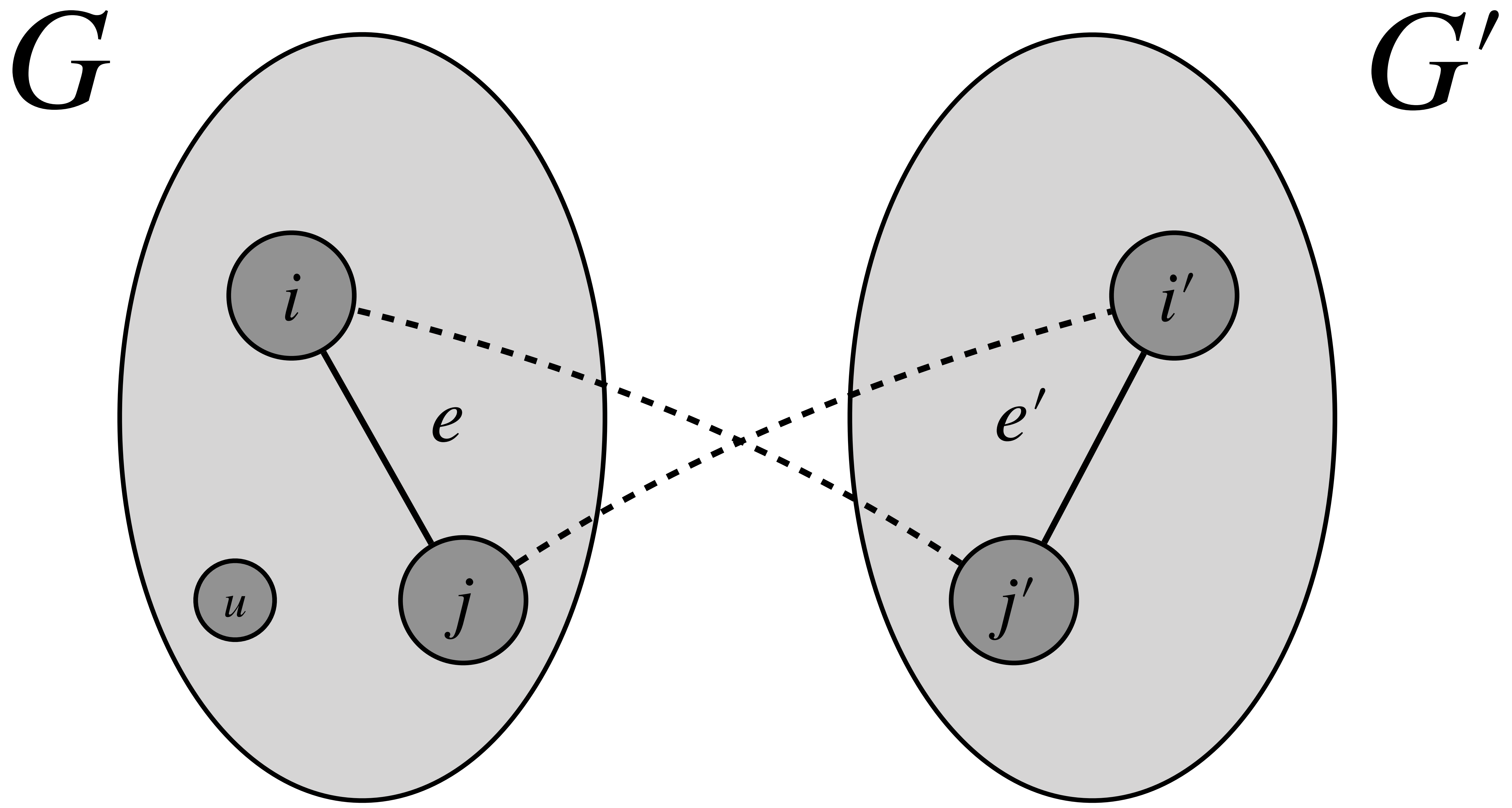


What about KT-1?



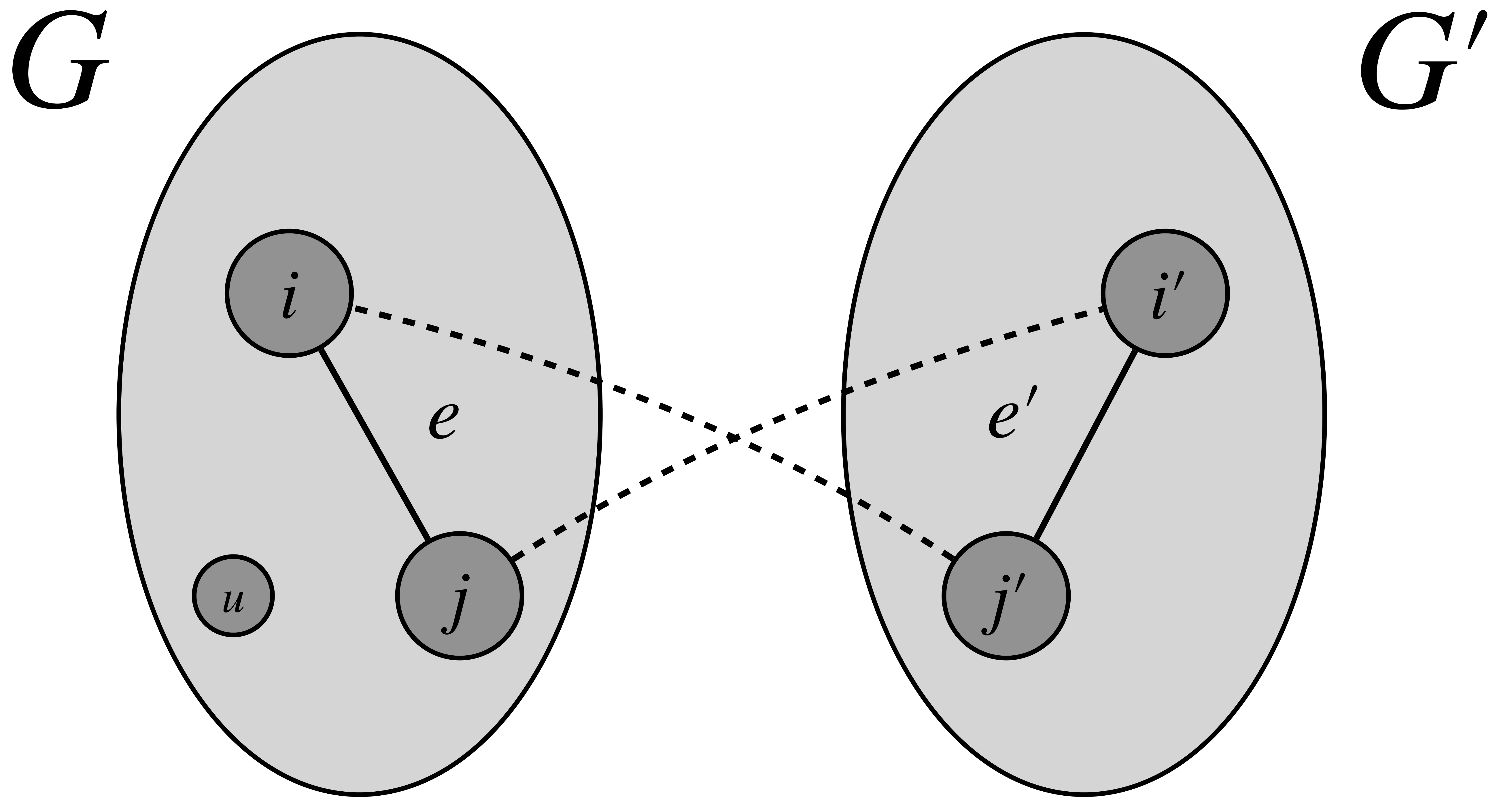
What about KT-1?

- Initial knowledge itself is different!



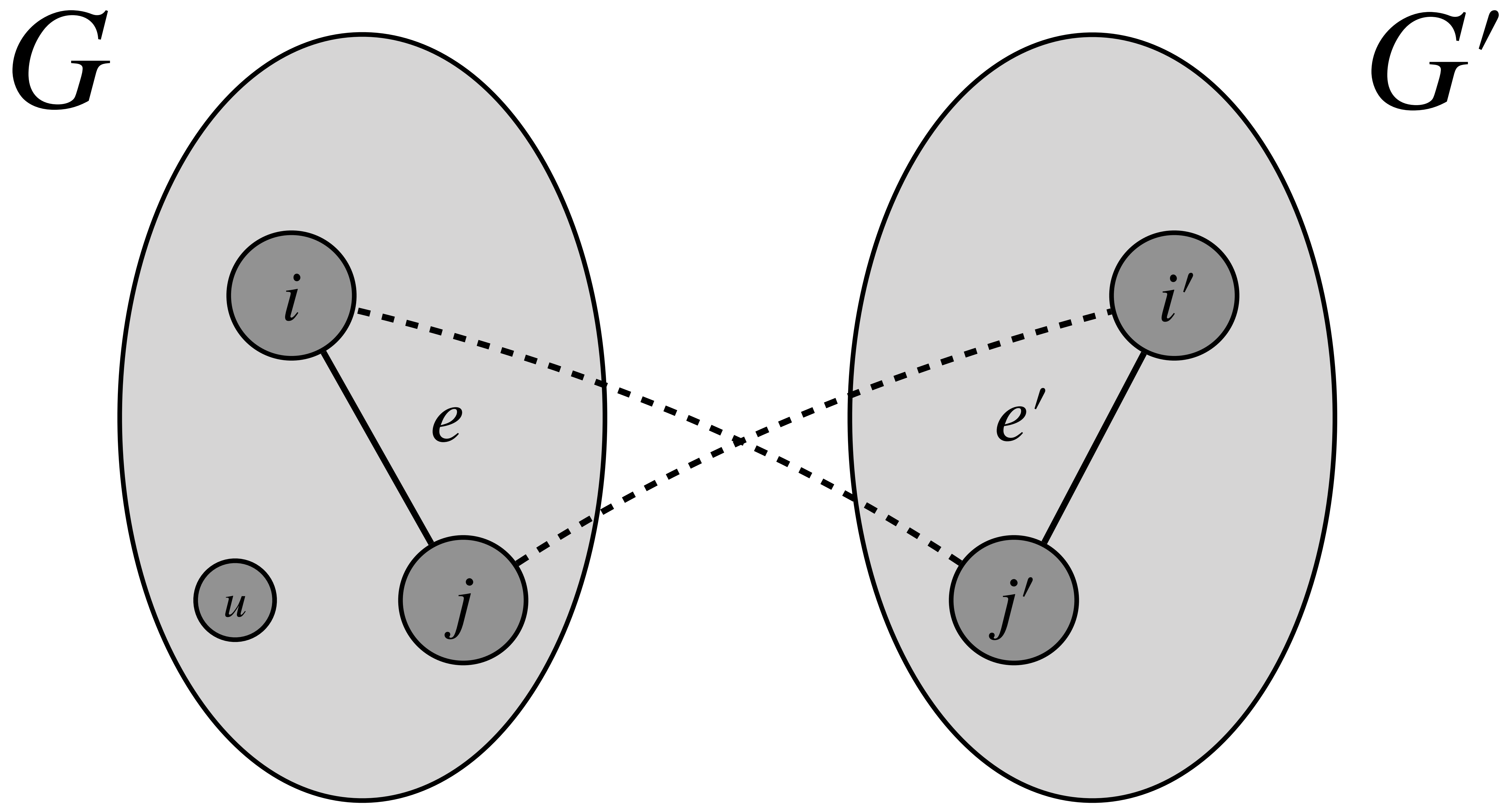
What about KT-1?

- Initial knowledge itself is different!
- Node i sees j in one graph and j' in the other.

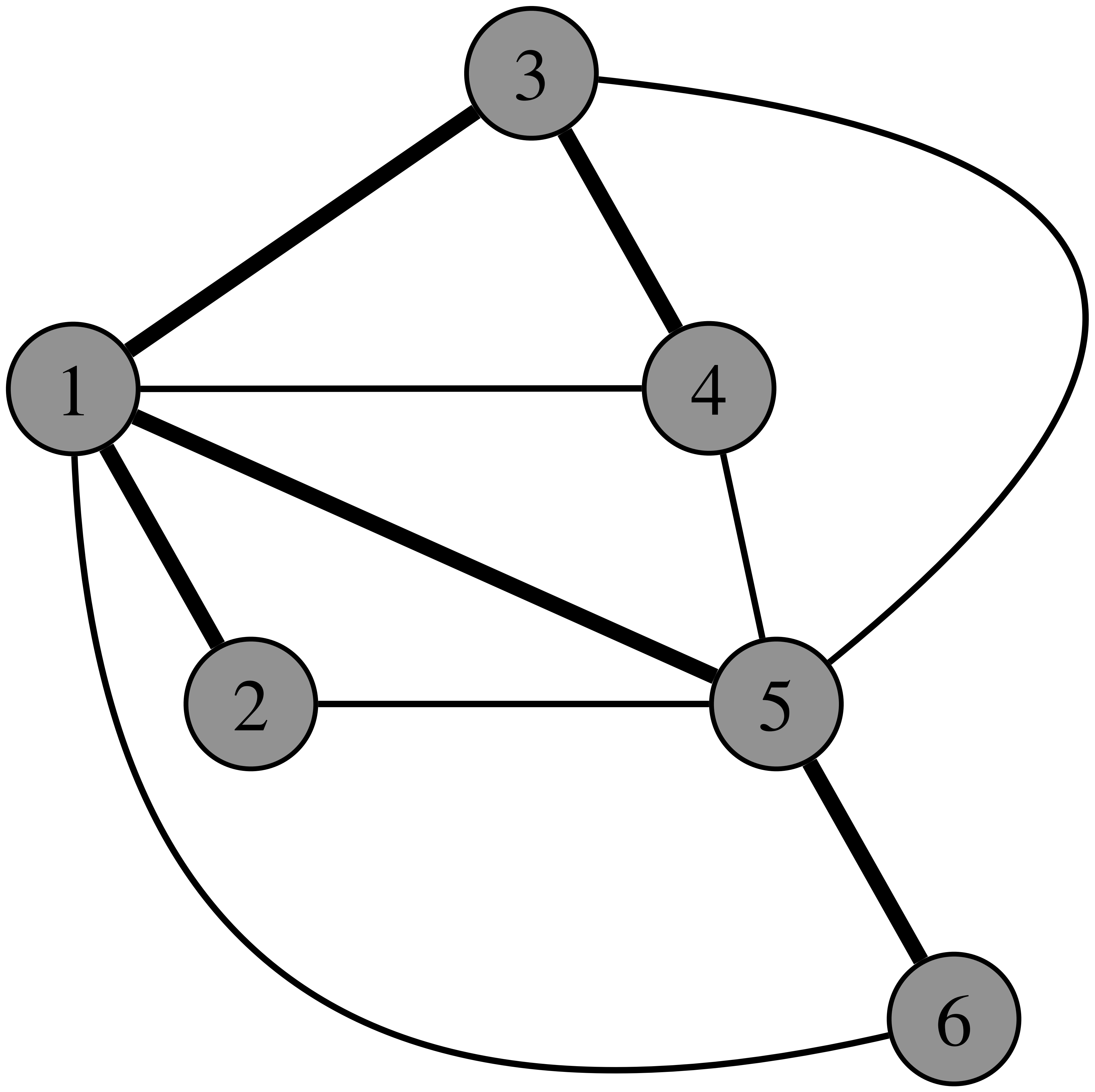


What about KT-1?

- Initial knowledge itself is different!
- Node i sees j in one graph and j' in the other.
- Is there any hope for an $\Omega(m)$ lower bound?

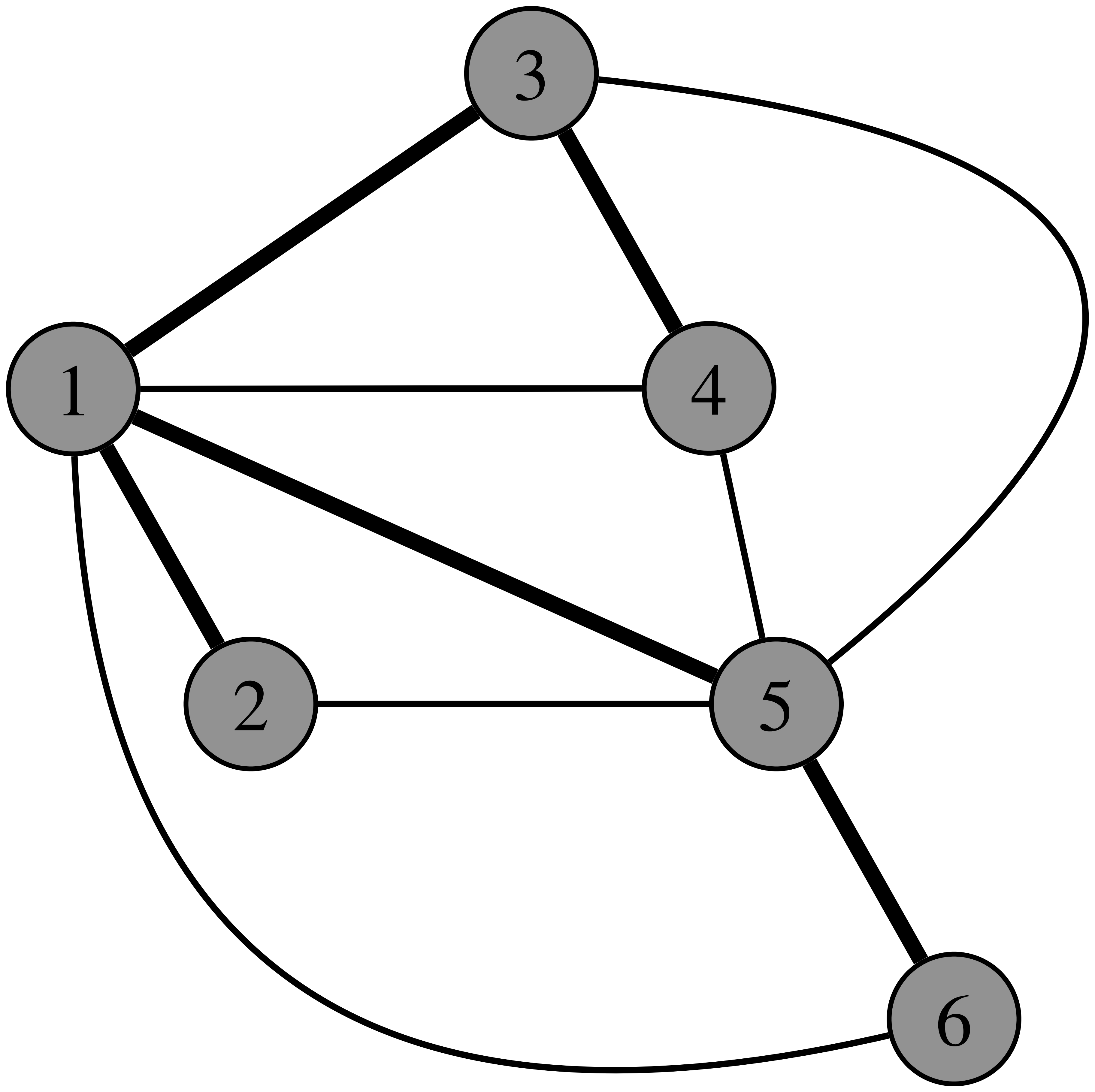


Message Efficient Broadcast



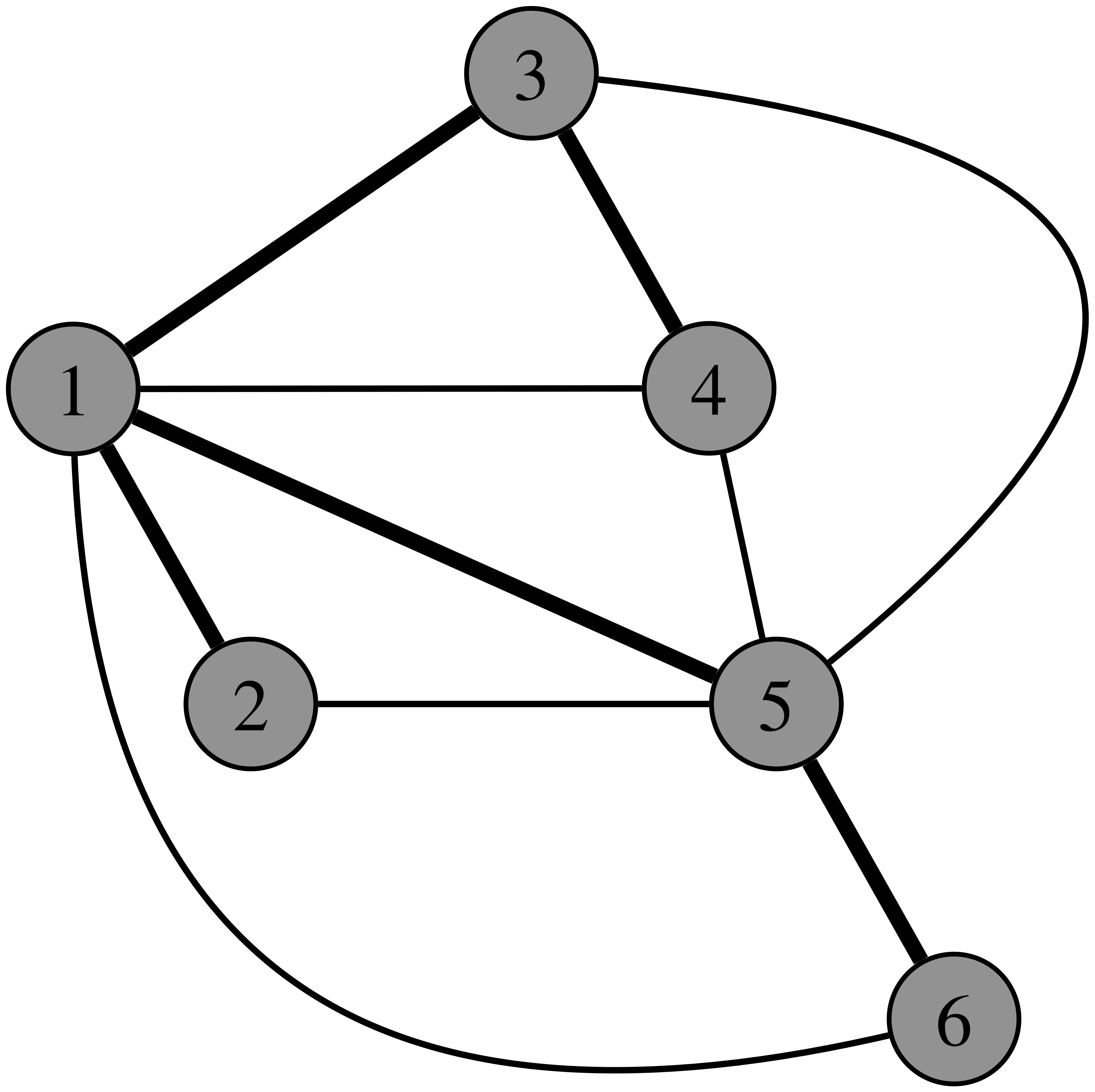
Message Efficient Broadcast

- We can compute a **spanning tree** with $\tilde{O}(n)$ rounds and $\tilde{O}(n)$ messages.



Message Efficient Broadcast

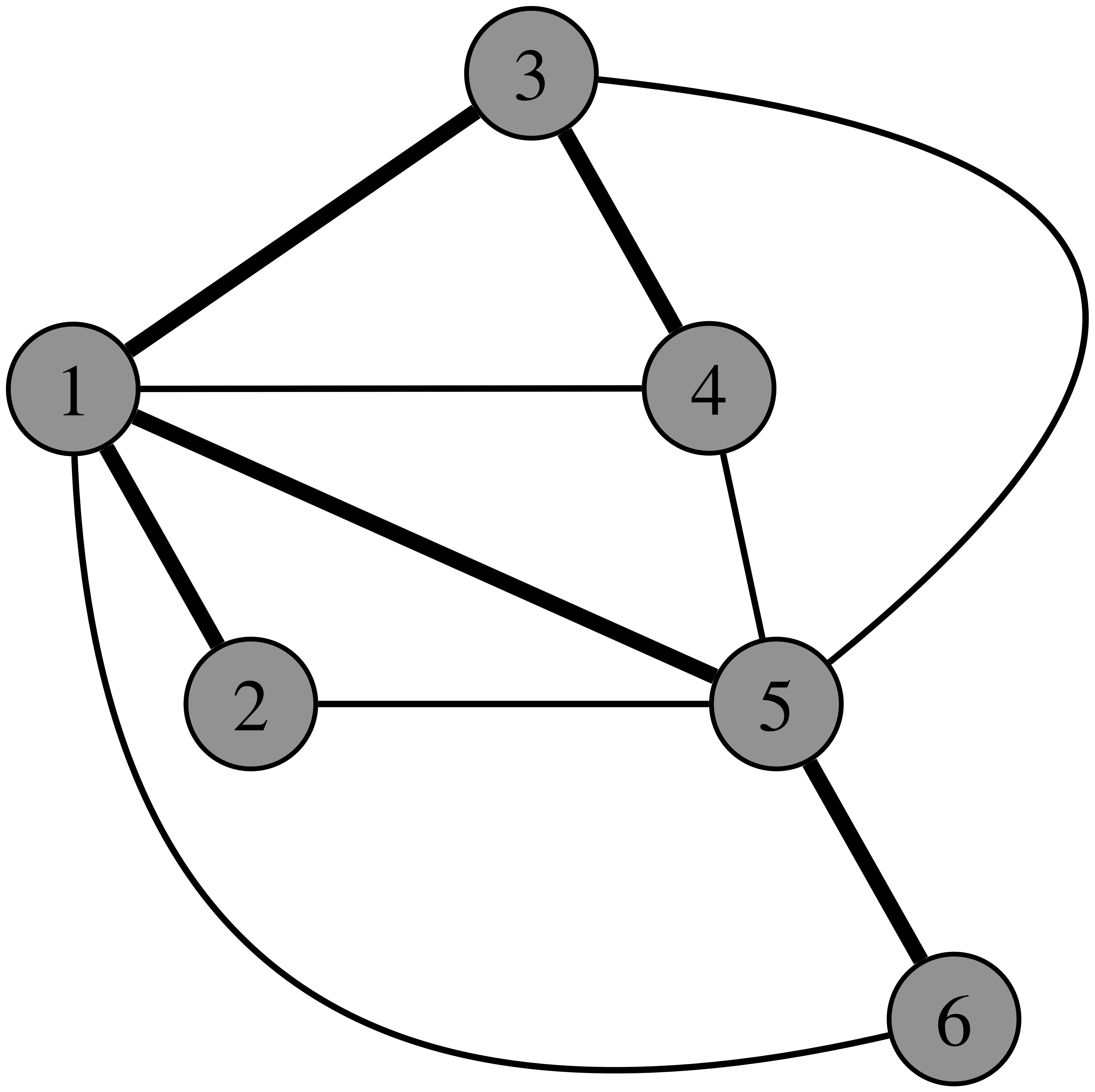
- We can compute a **spanning tree** with $\tilde{O}(n)$ rounds and $\tilde{O}(n)$ messages.
    - [KKT15] key ingredient: randomized linear sketches.



[KKT15] King, Kutten, Thorup. PODC 2015

Message Efficient Broadcast

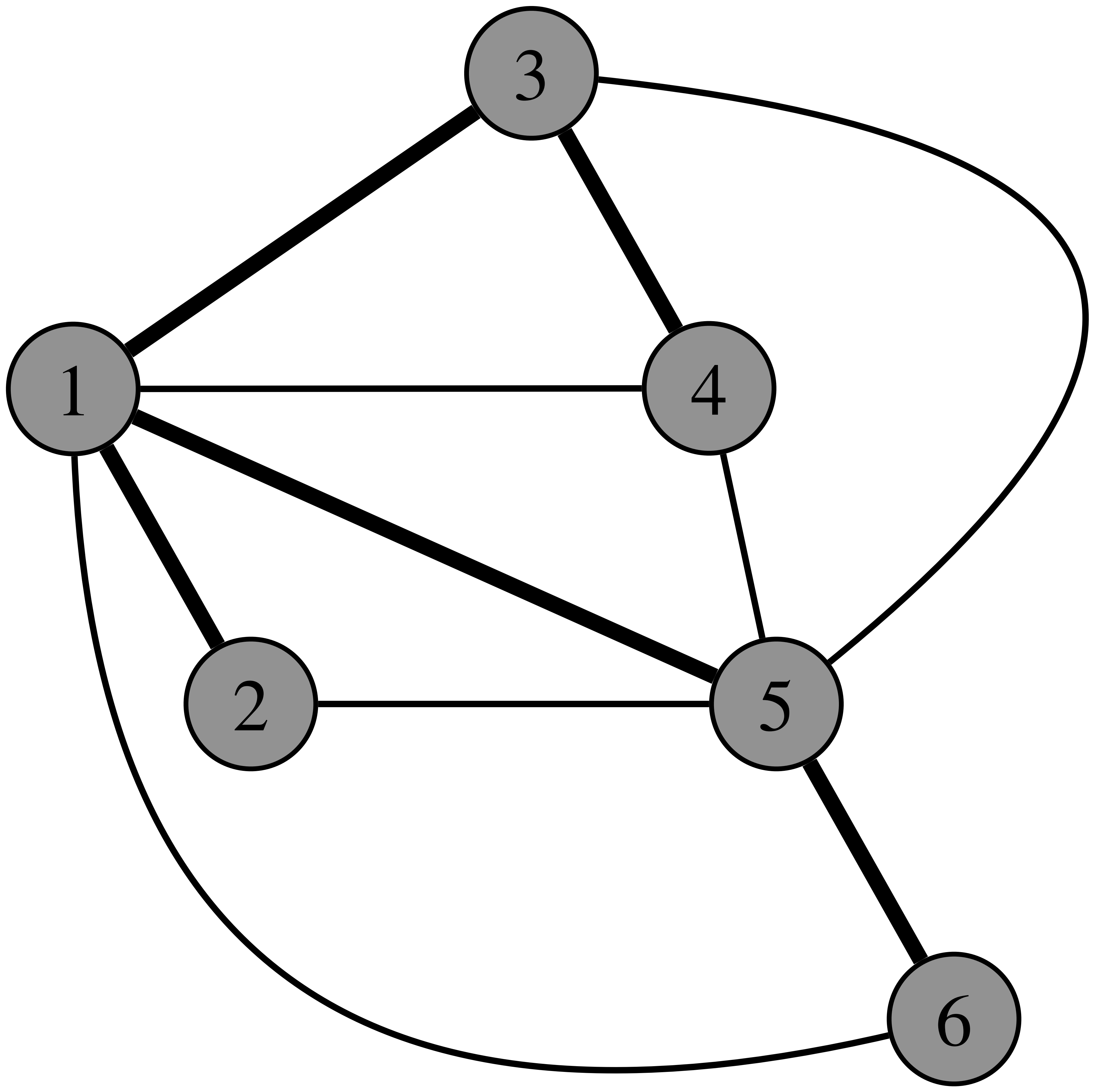
- We can compute a **spanning tree** with $\tilde{O}(n)$ rounds and $\tilde{O}(n)$ messages.
    - [KKT15] key ingredient: randomized linear sketches.
- Broadcast by flooding on just the spanning tree edges.



[KKT15] King, Kutten, Thorup. PODC 2015

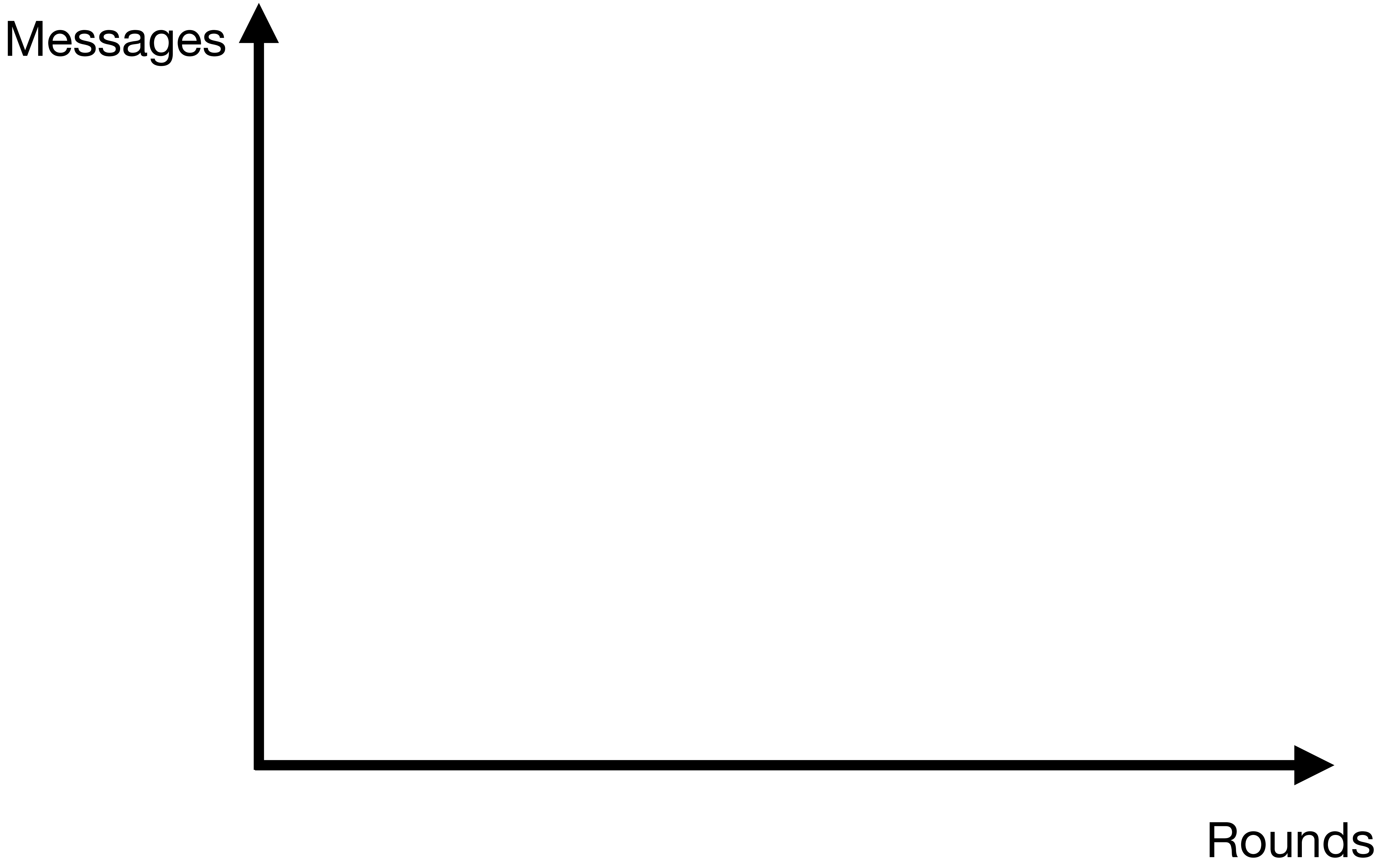
Message Efficient Broadcast

- We can compute a **spanning tree** with $\tilde{O}(n)$ rounds and $\tilde{O}(n)$ messages.
    - [KKT15] key ingredient: randomized linear sketches.
- Broadcast by flooding on just the spanning tree edges.
- Message efficient but not round efficient.

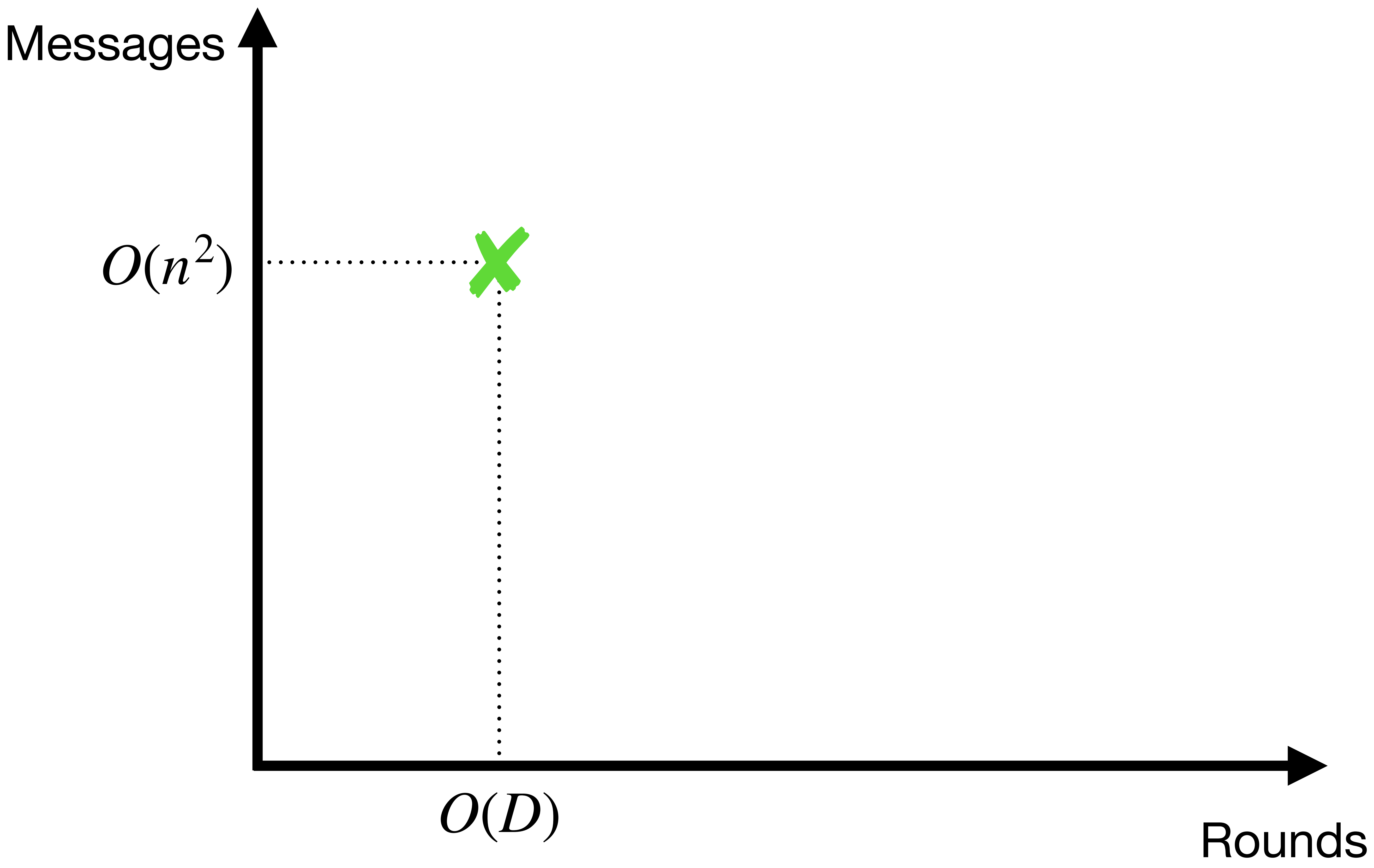


[KKT15] King, Kutten, Thorup. PODC 2015

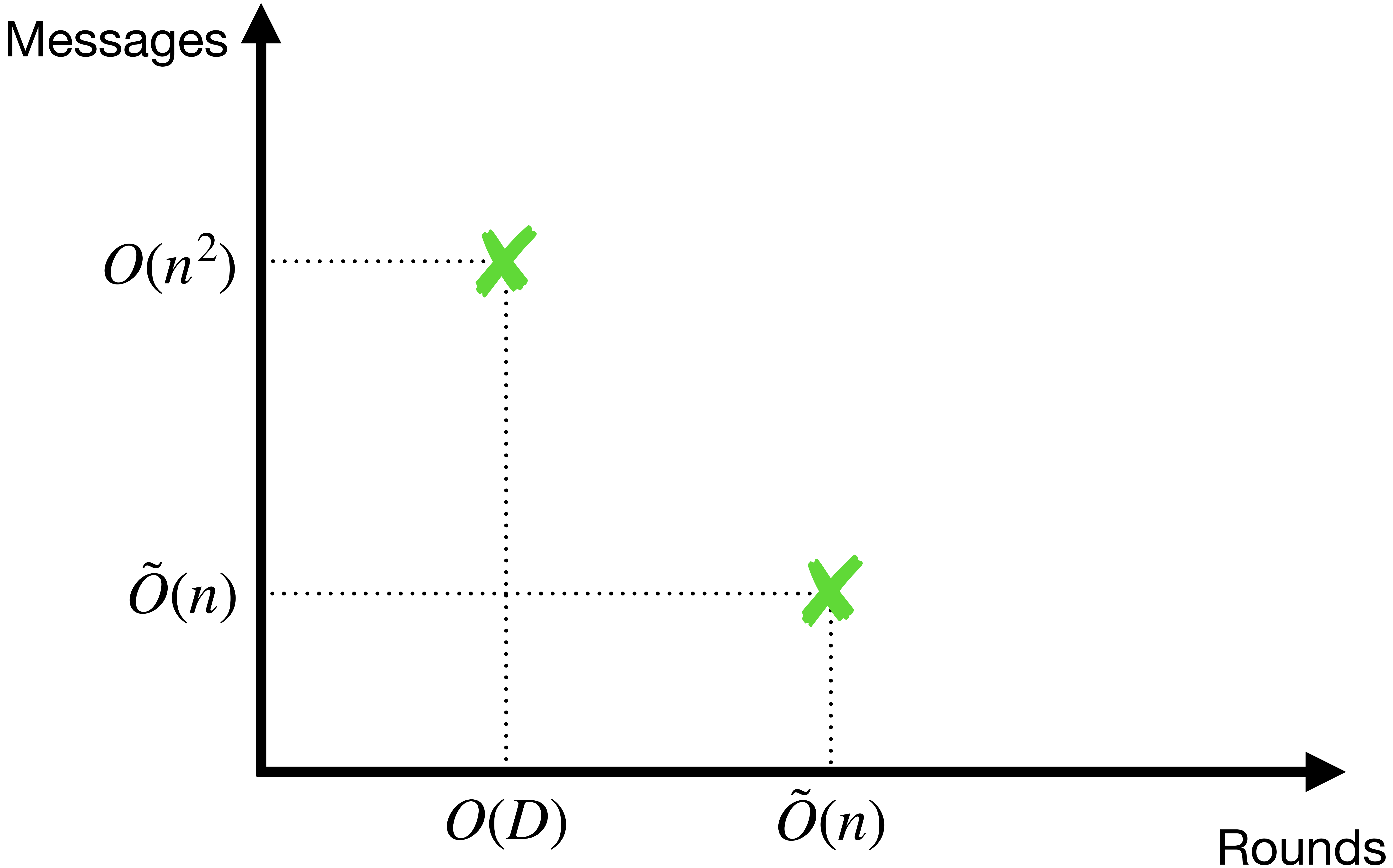
Rounds vs Messages



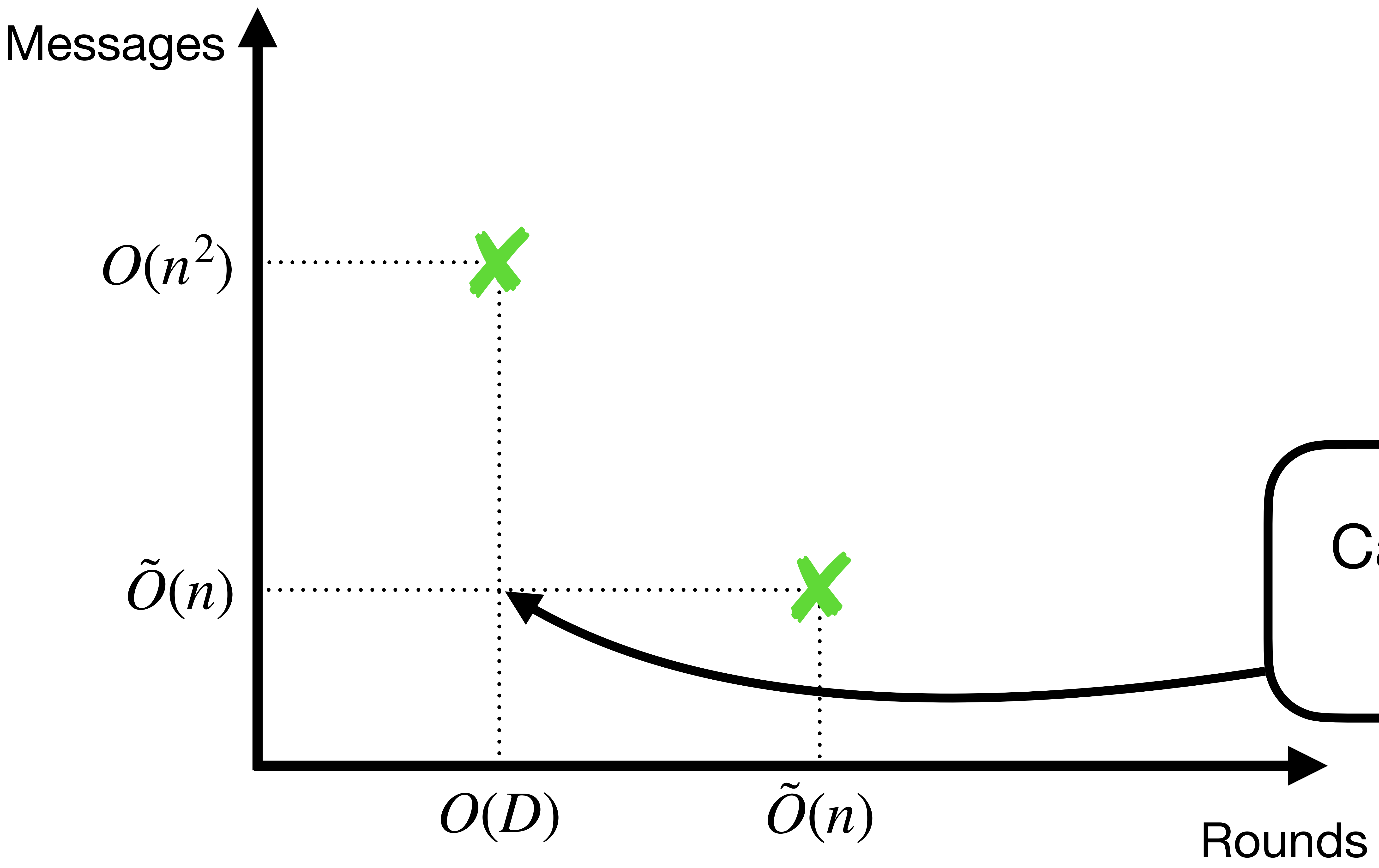
Rounds vs Messages



Rounds vs Messages

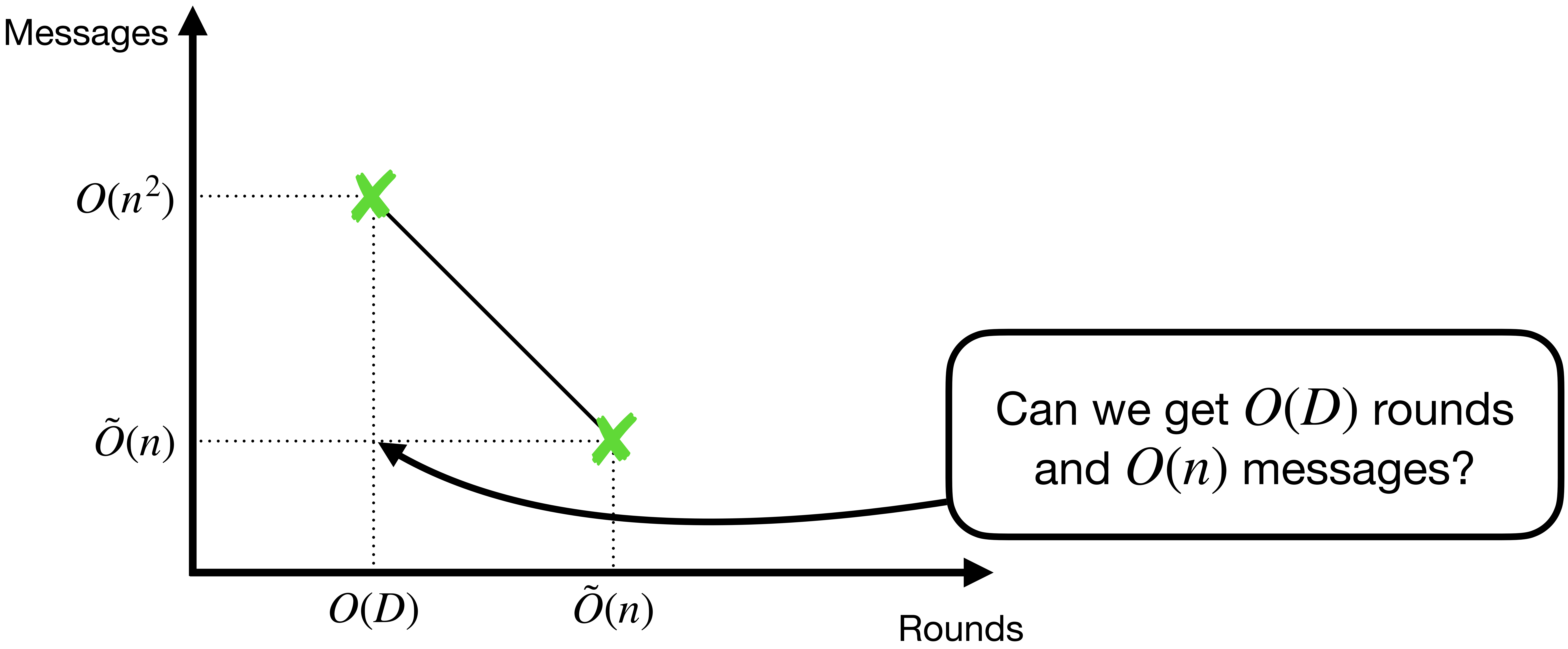


Rounds vs Messages

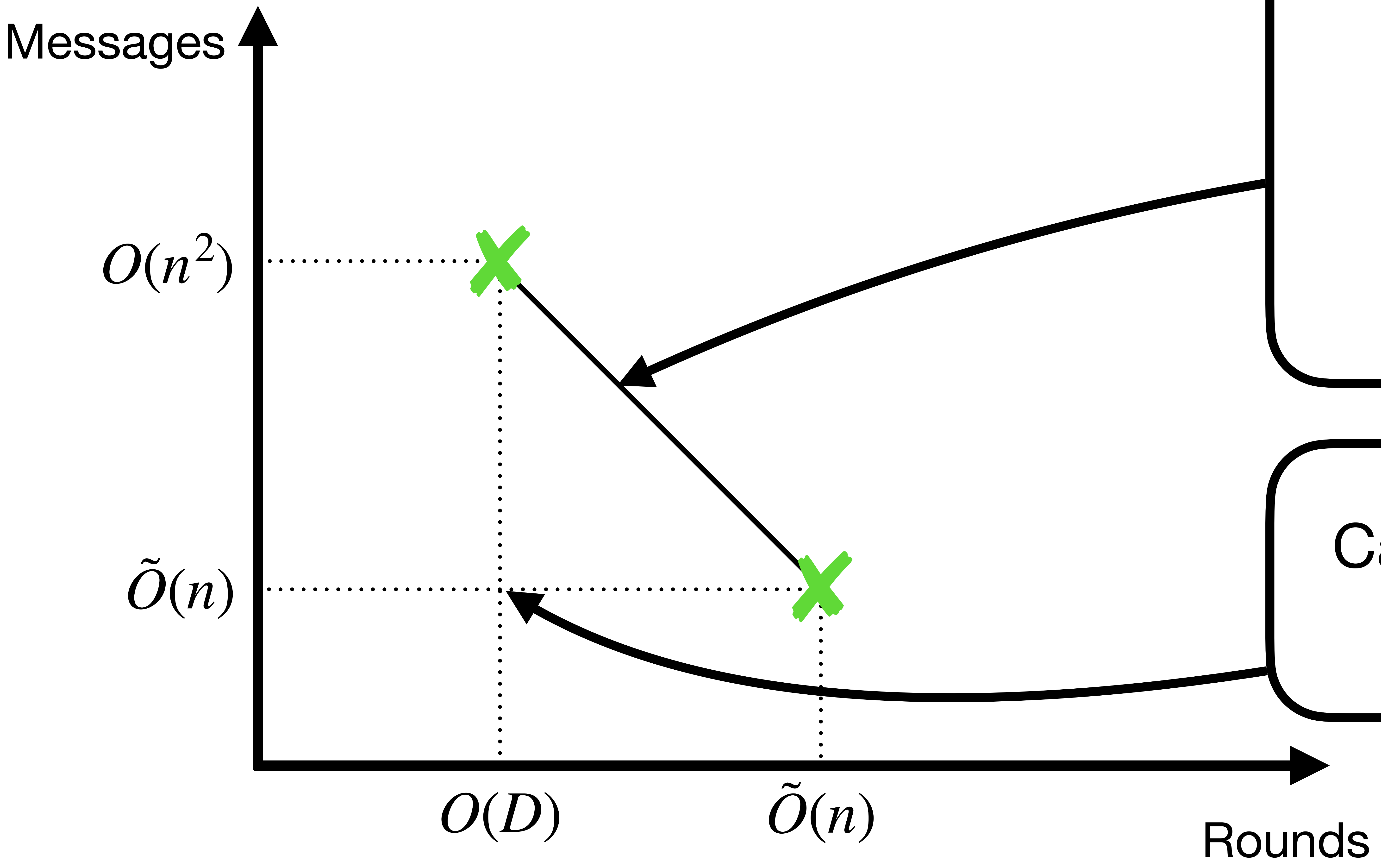


Can we get $O(D)$ rounds and $O(n)$ messages?

Rounds vs Messages



Rounds vs Messages



$\tilde{O}(D + n^{1-\delta})$ rounds

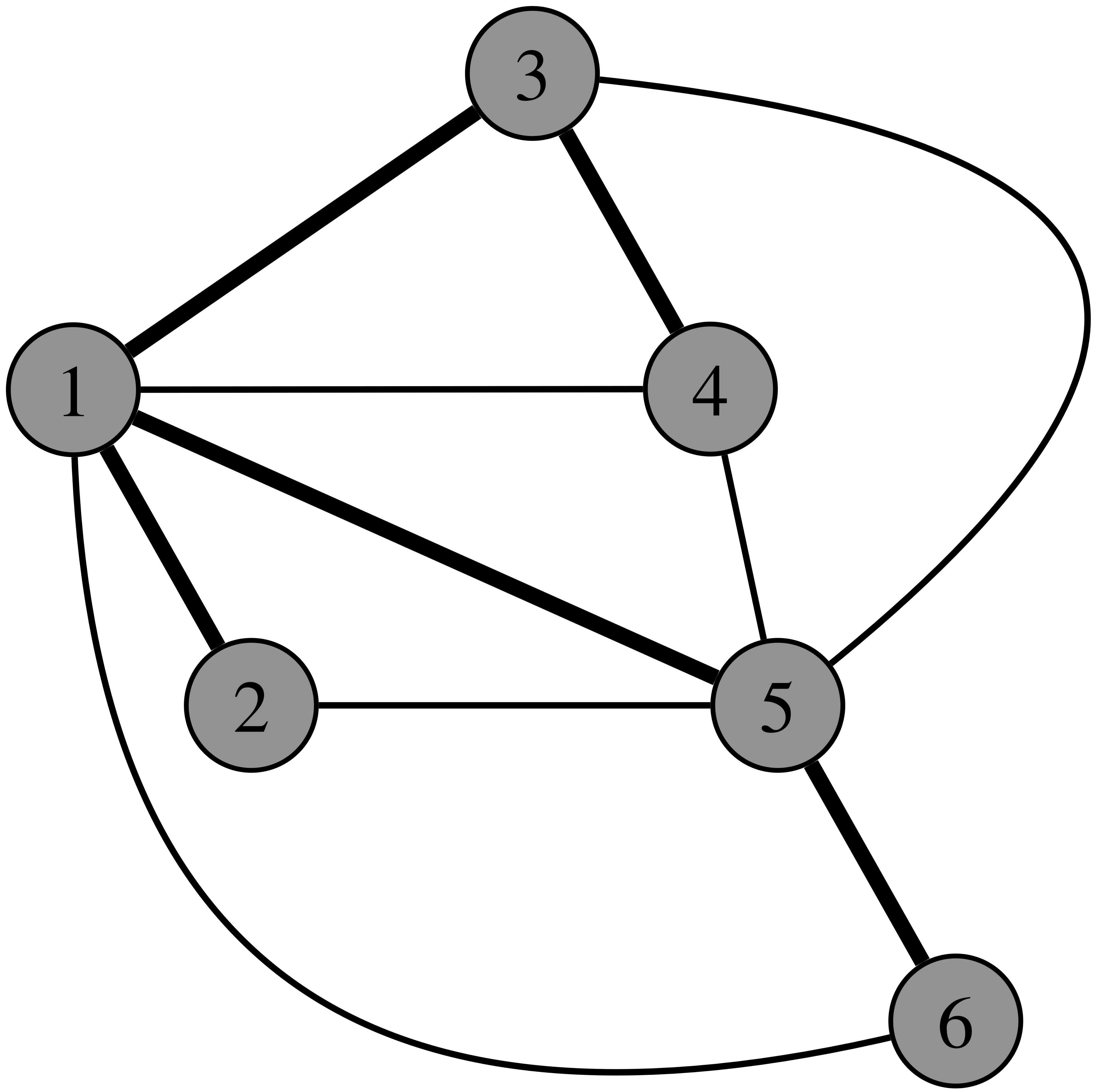
$\tilde{O}(n^{1+\delta})$ messages

for $0 < \delta < 1$ [GP18]

Can we get $O(D)$ rounds and $O(n)$ messages?

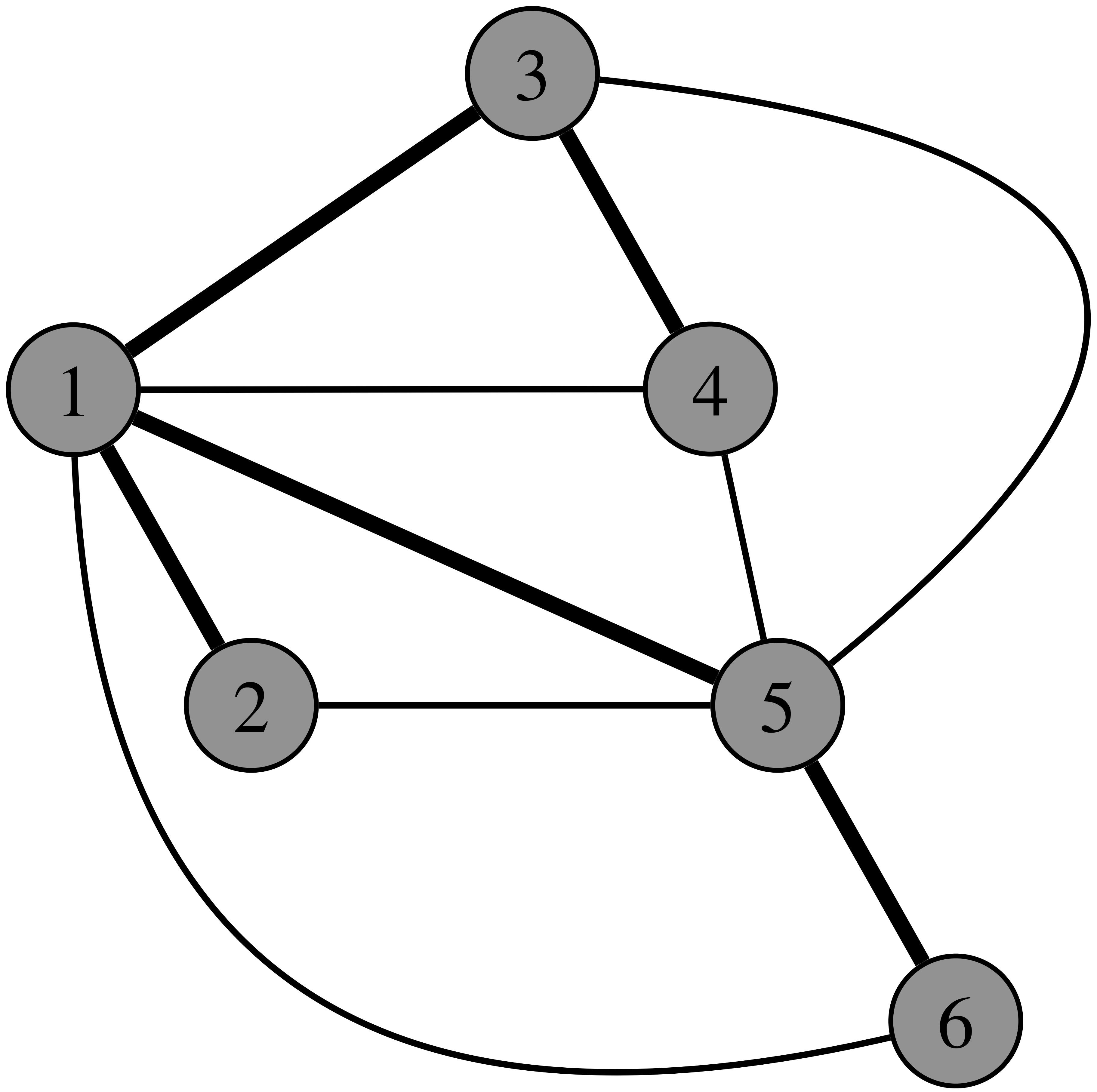
[GP18] Gmyr, Pandurangan. DISC 2018

Silence Conveys Information



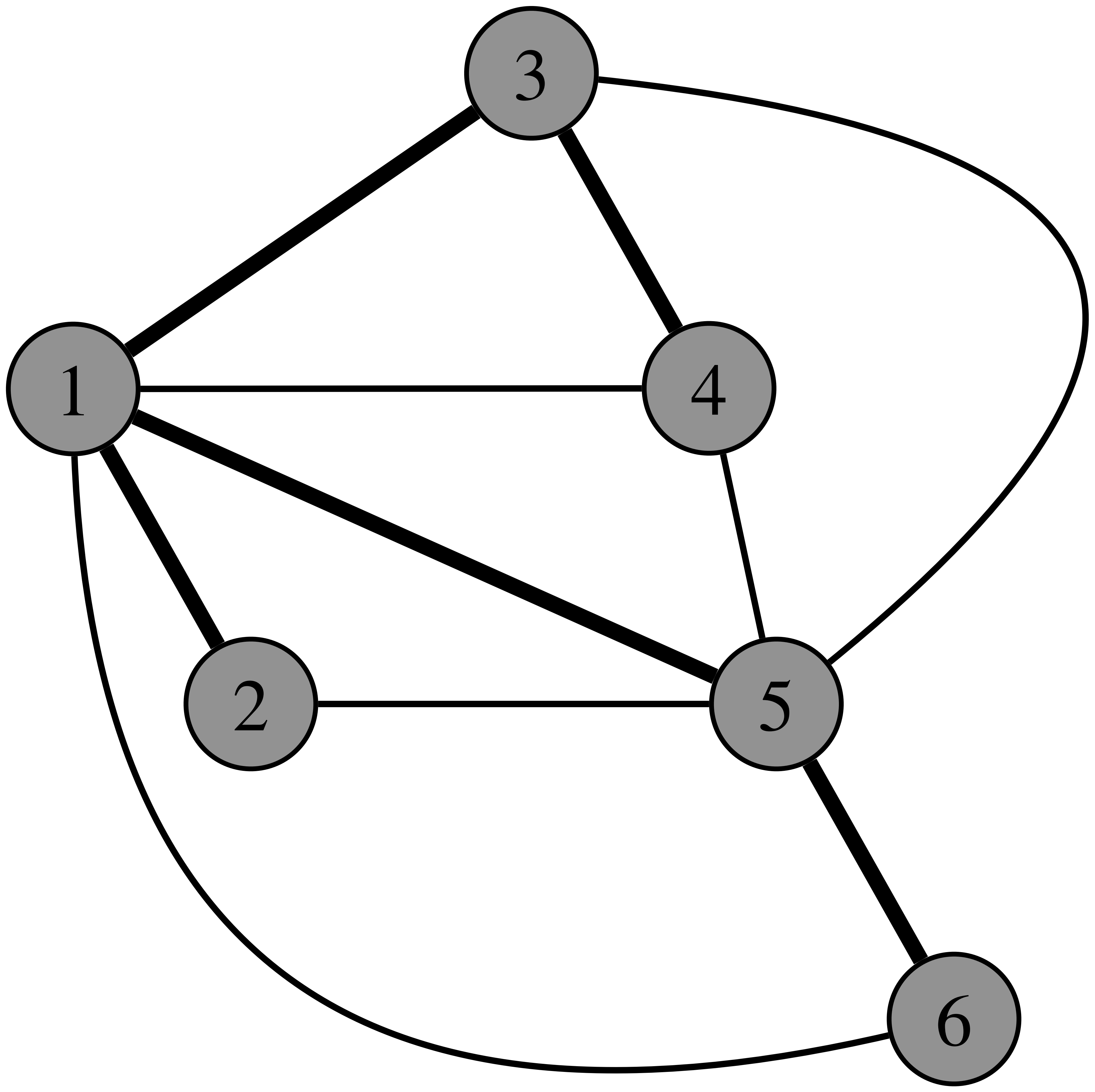
Silence Conveys Information

- We can compute a spanning tree using $\tilde{O}(n)$ messages.



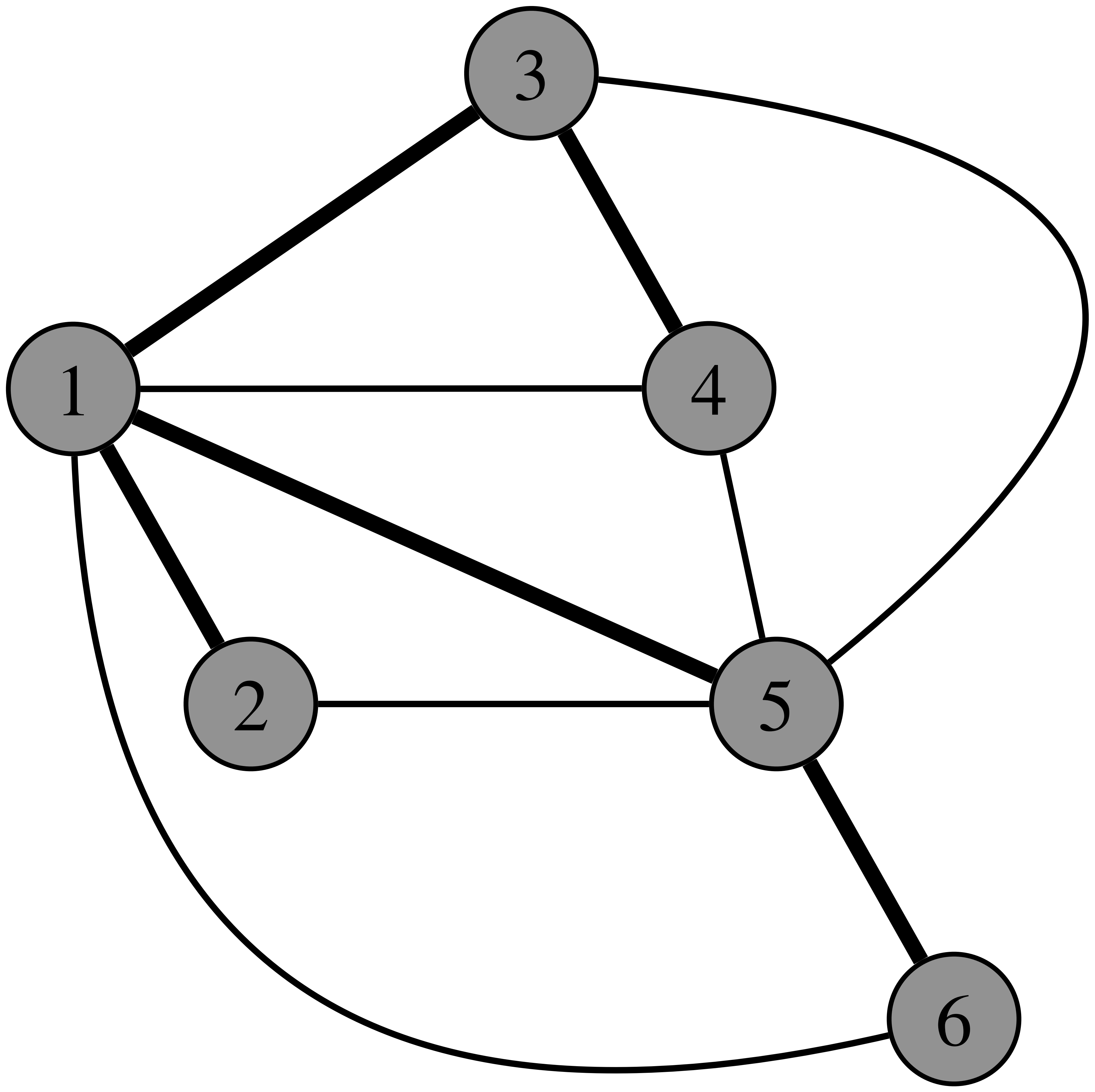
Silence Conveys Information

- We can compute a spanning tree using $\tilde{O}(n)$ messages.
- This allows us to solve *any* graph problem using $\tilde{O}(n)$ messages!

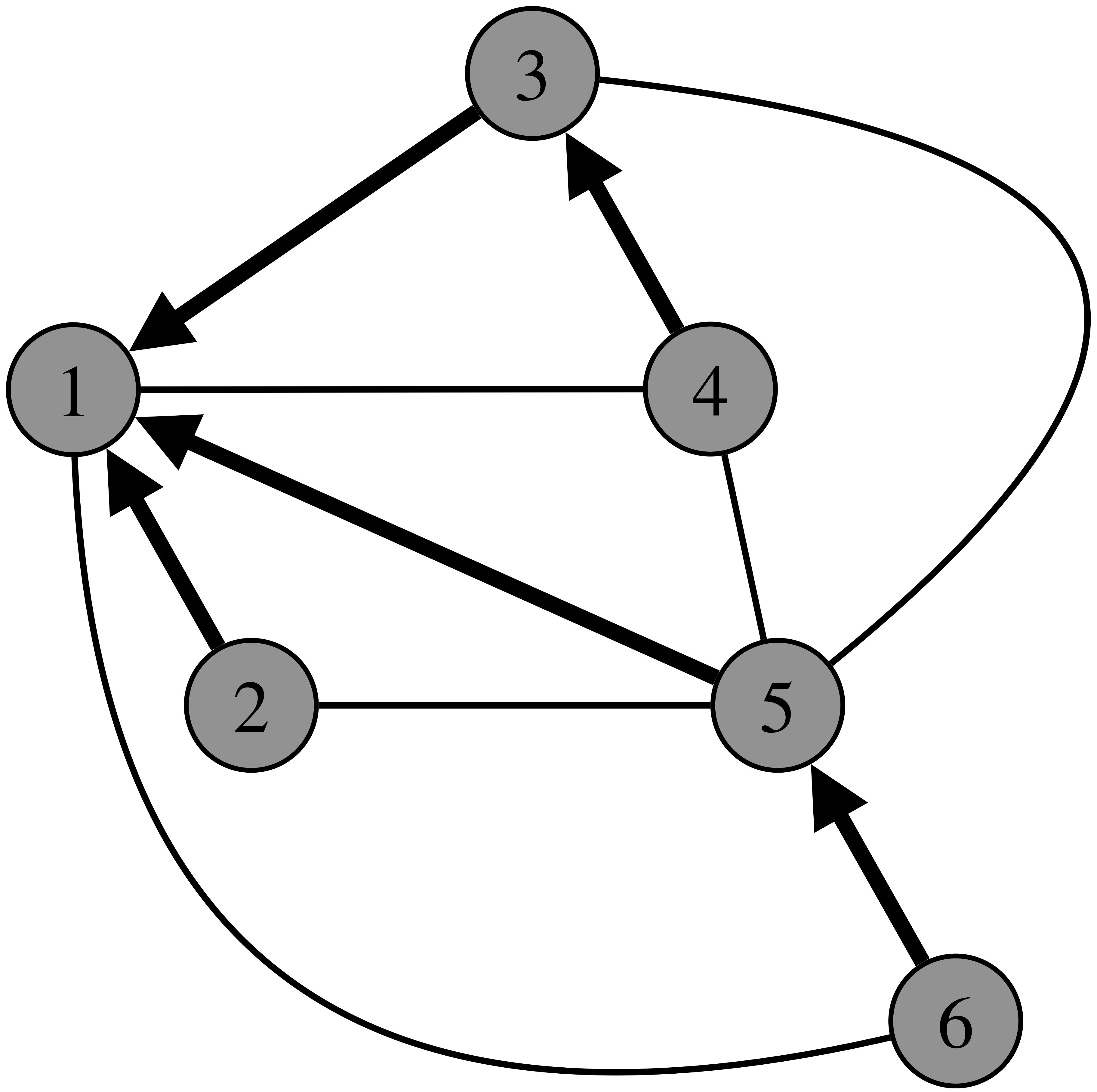


Silence Conveys Information

- We can compute a spanning tree using $\tilde{O}(n)$ messages.
- This allows us to solve *any* graph problem using $\tilde{O}(n)$ messages!
- Trick: nodes use clock values to send topology information up the spanning tree.



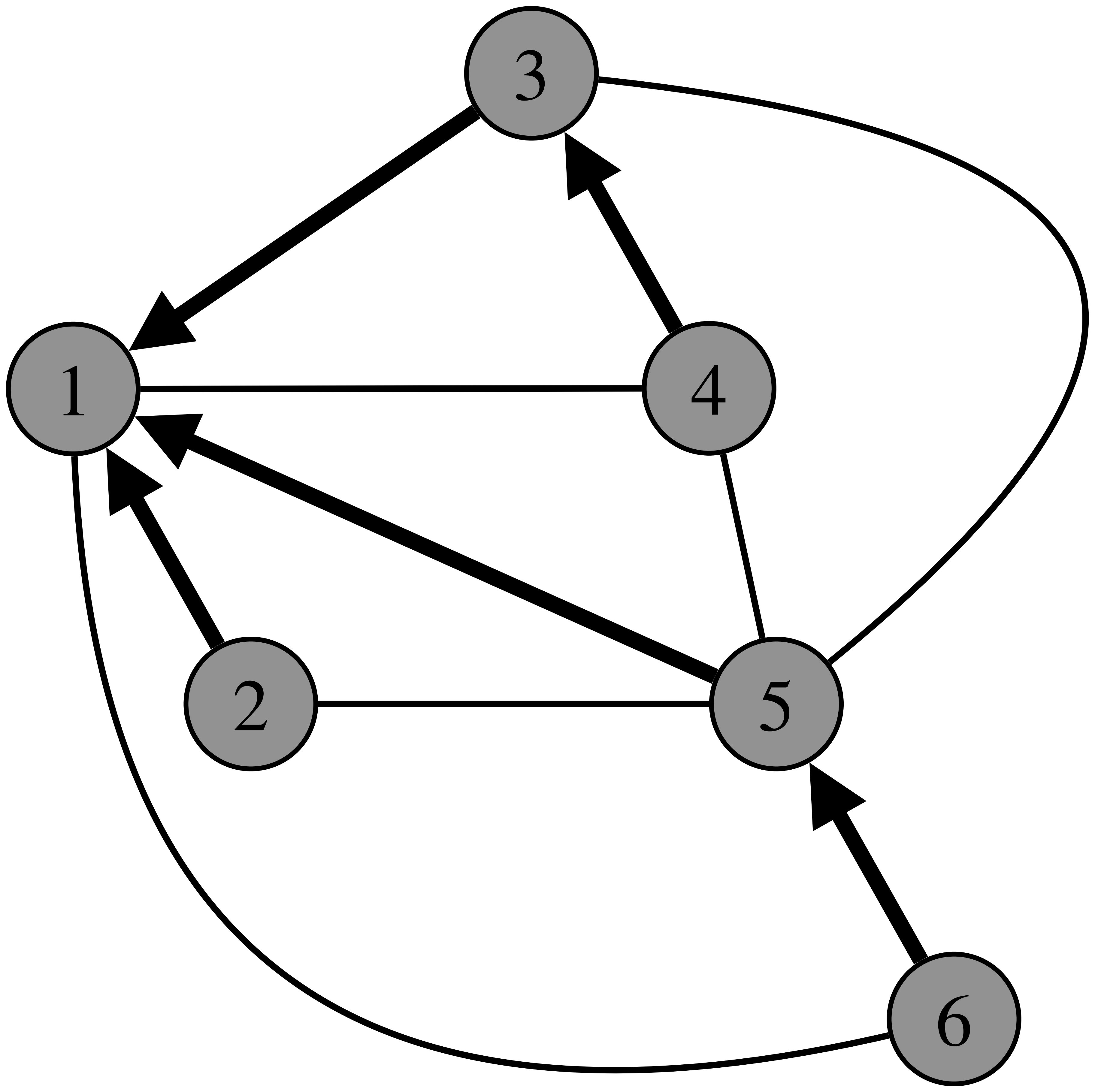
Silence Conveys Information



Silence Conveys Information

List $\ell = 2^{\binom{n}{2}}$ graphs on n nodes.

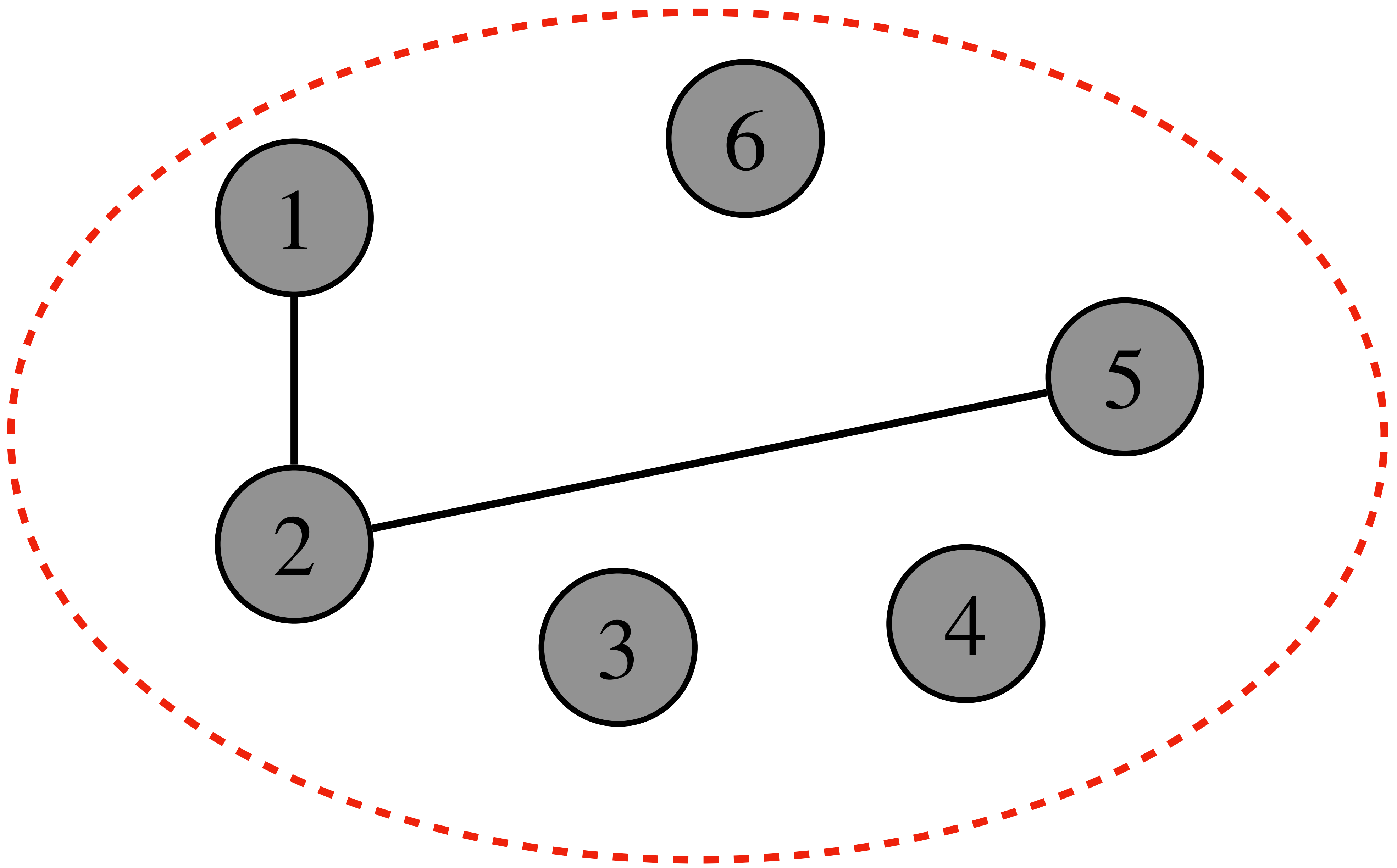
Round $T = i \pmod{\ell}$ denotes the i^{th} graph.

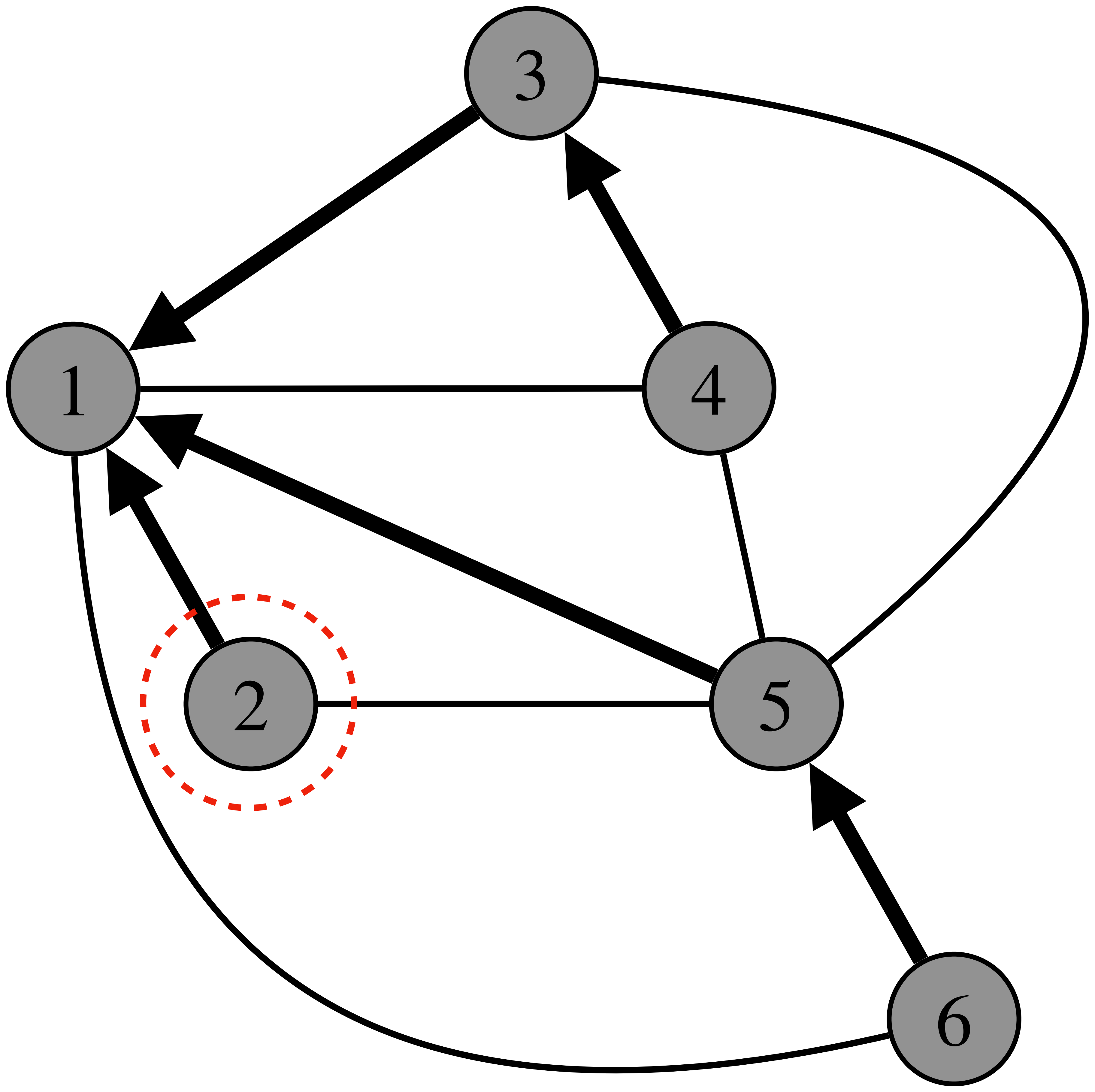


Silence Conveys Information

List $\ell = 2^{\binom{n}{2}}$ graphs on n nodes.

Round $T = i \pmod{\ell}$ denotes the i^{th} graph.

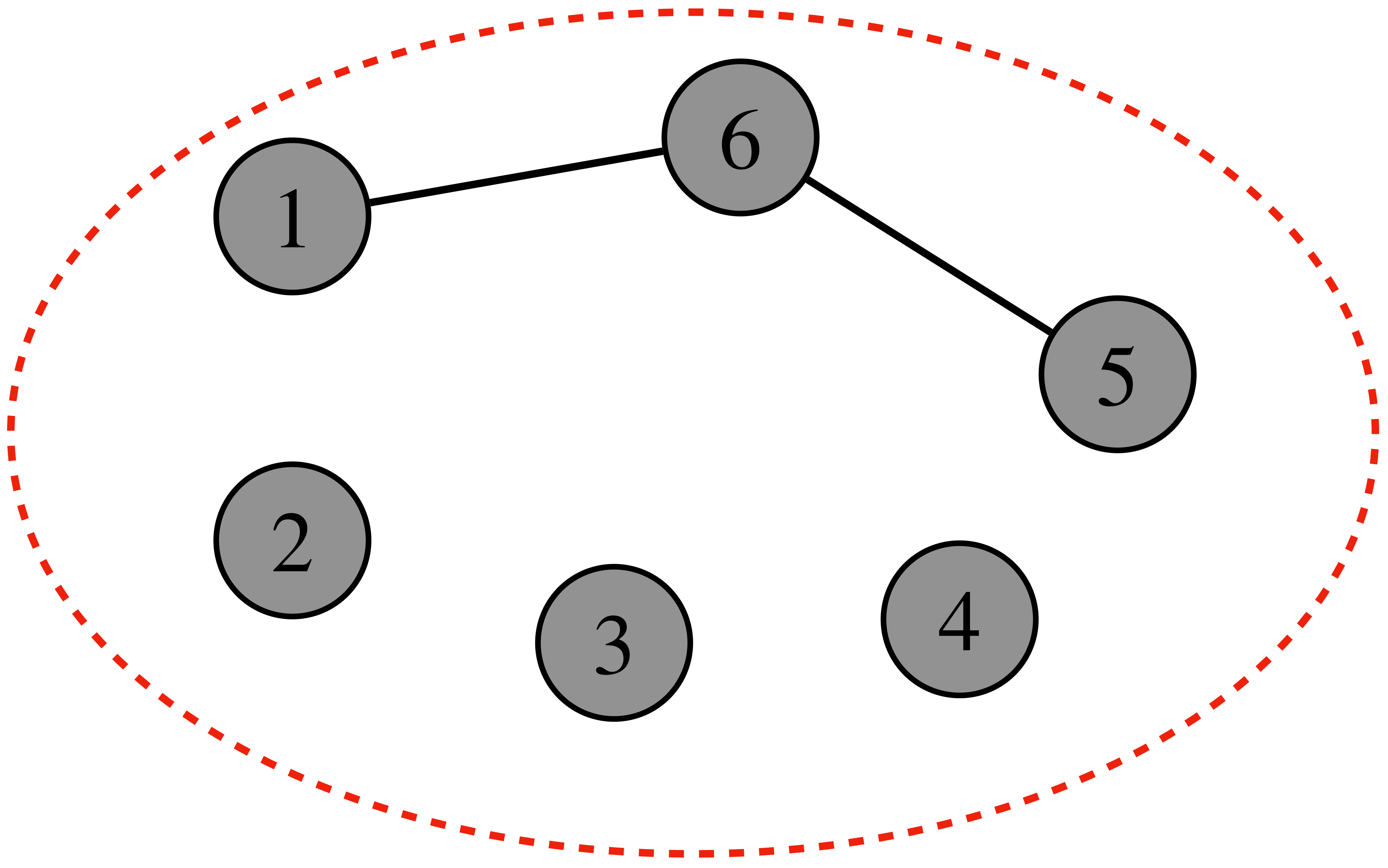


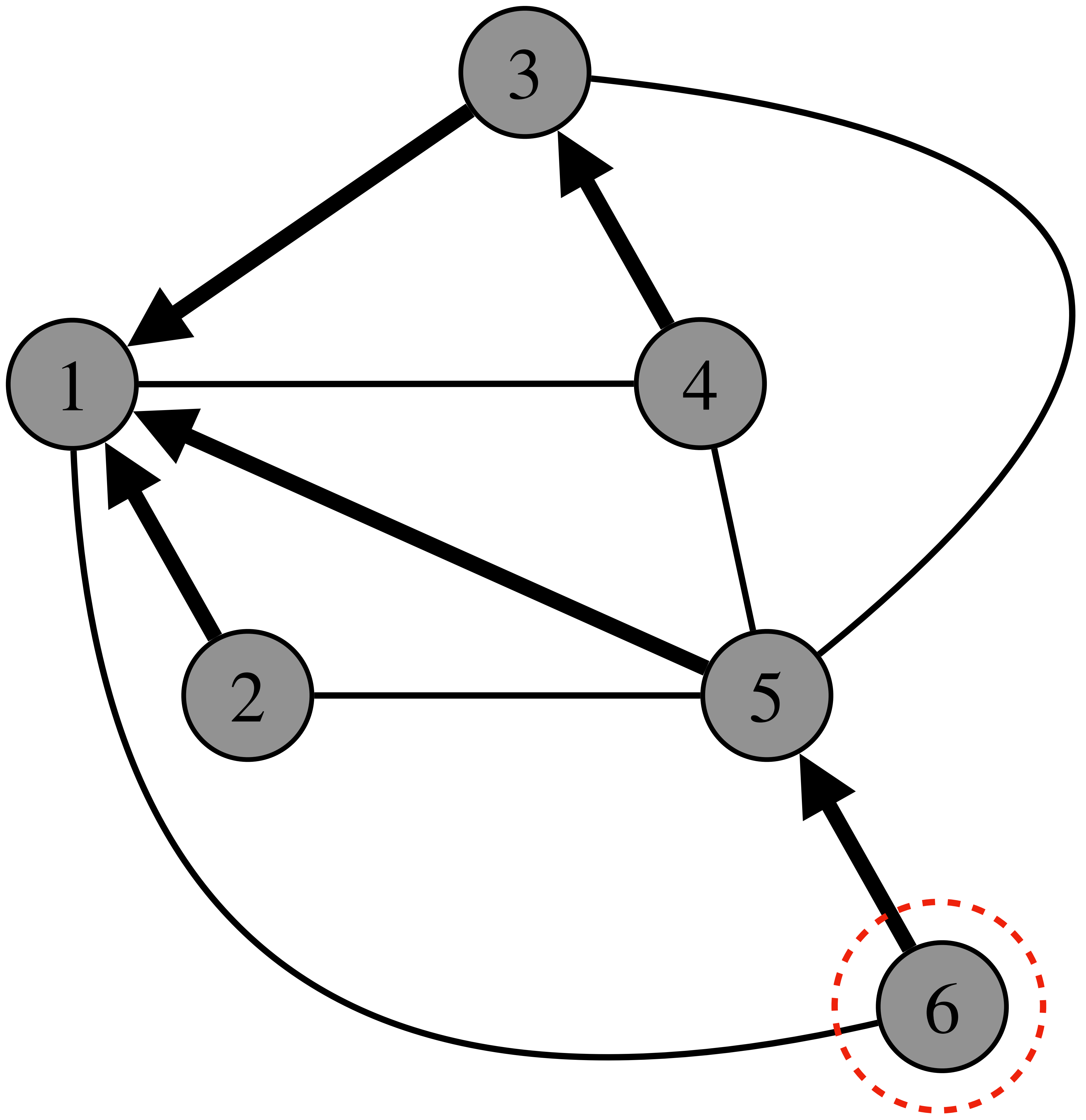


Silence Conveys Information

List $\ell = 2^{\binom{n}{2}}$ graphs on n nodes.

Round $T = i \pmod{\ell}$ denotes the i^{th} graph.

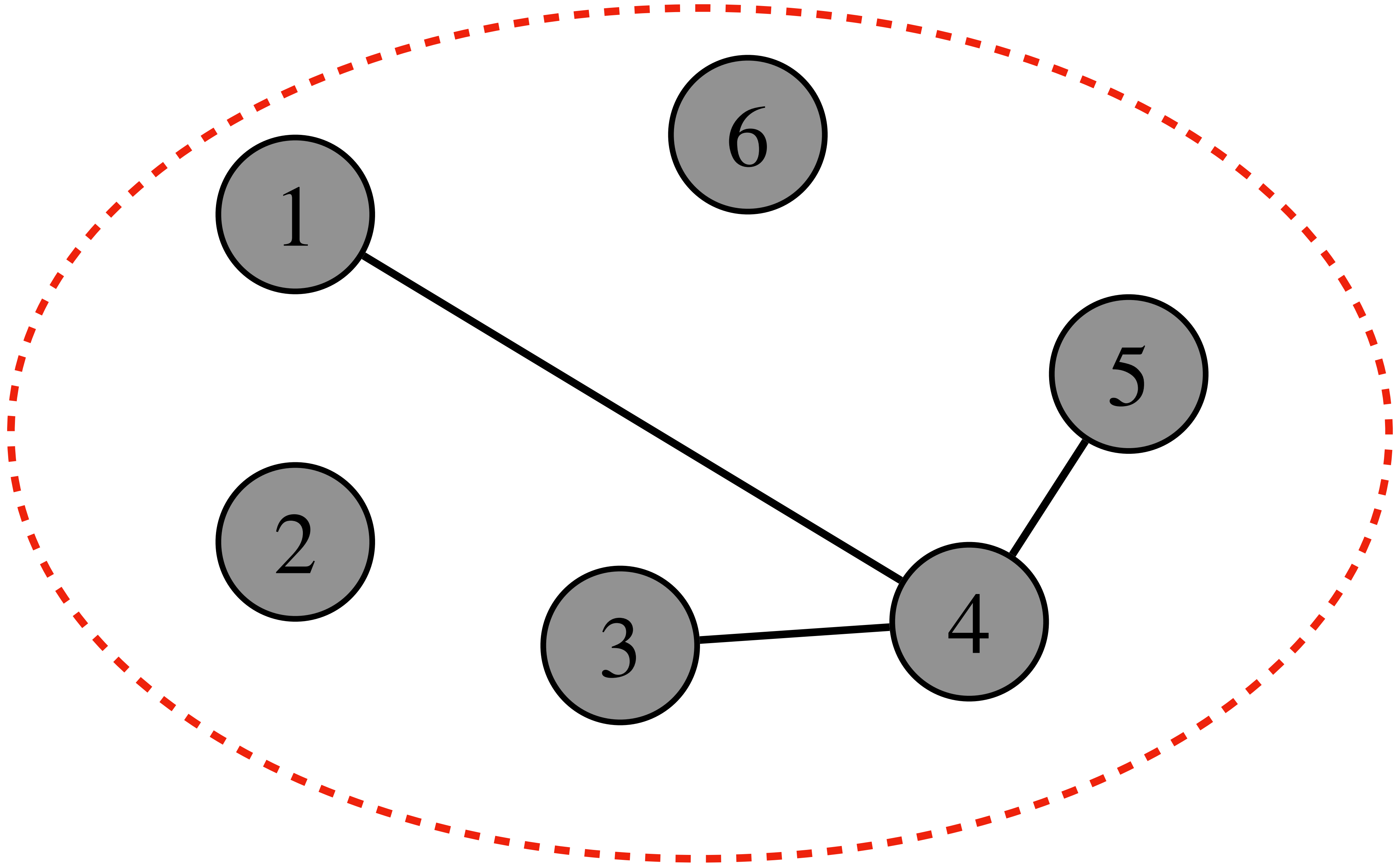


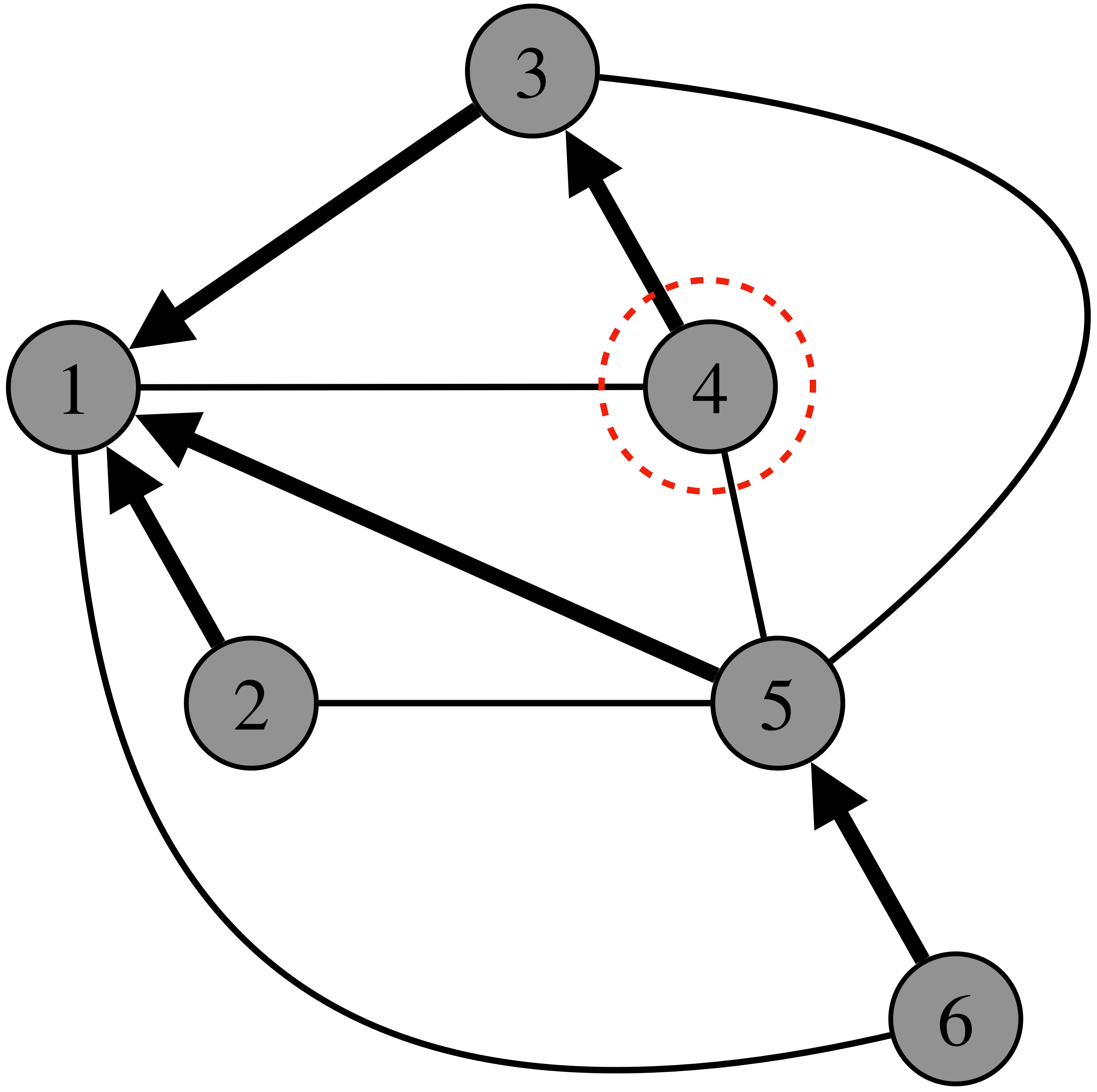


Silence Conveys Information

List $\ell = 2^{\binom{n}{2}}$ graphs on n nodes.

Round $T = i \pmod{\ell}$ denotes the i^{th} graph.

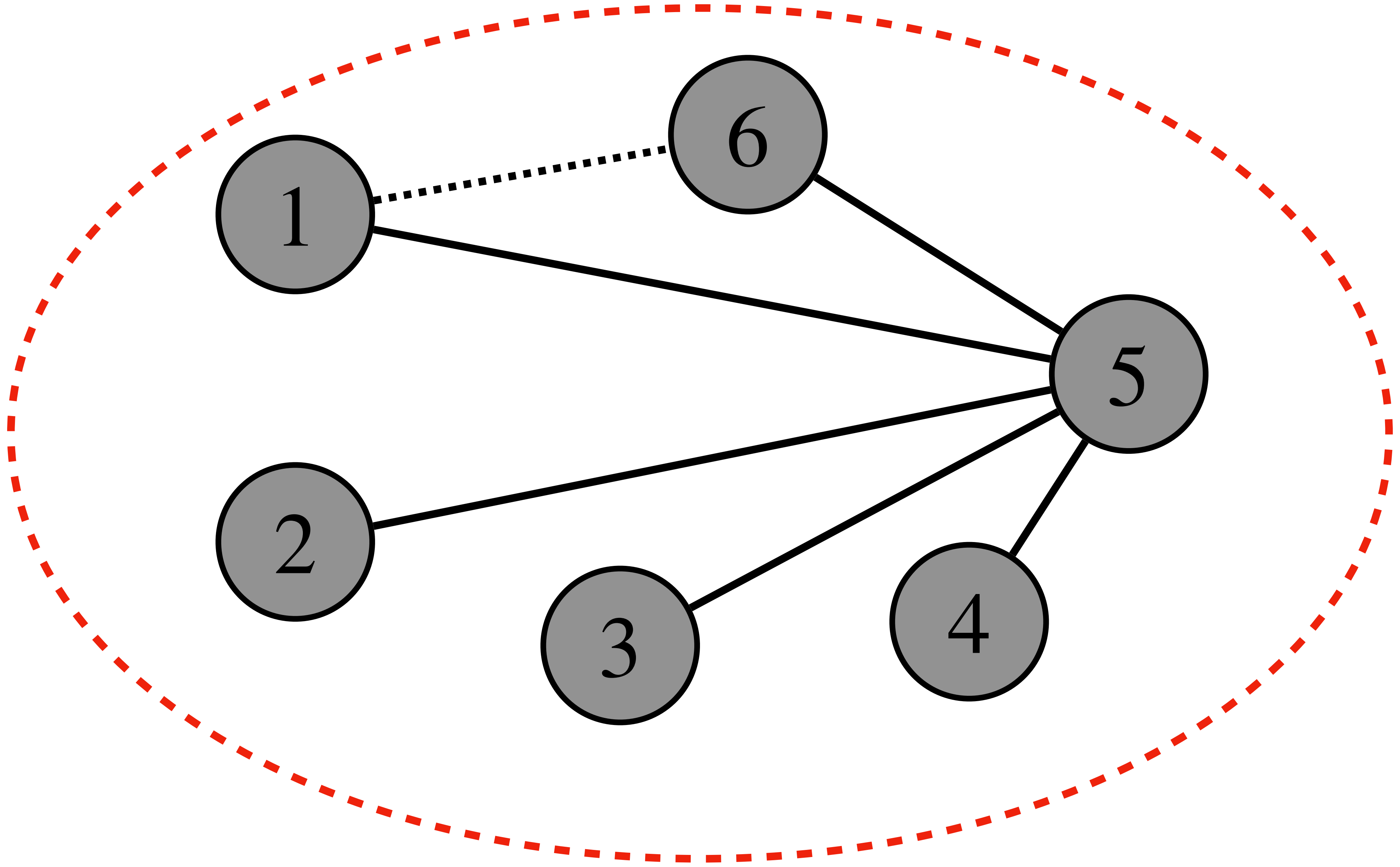


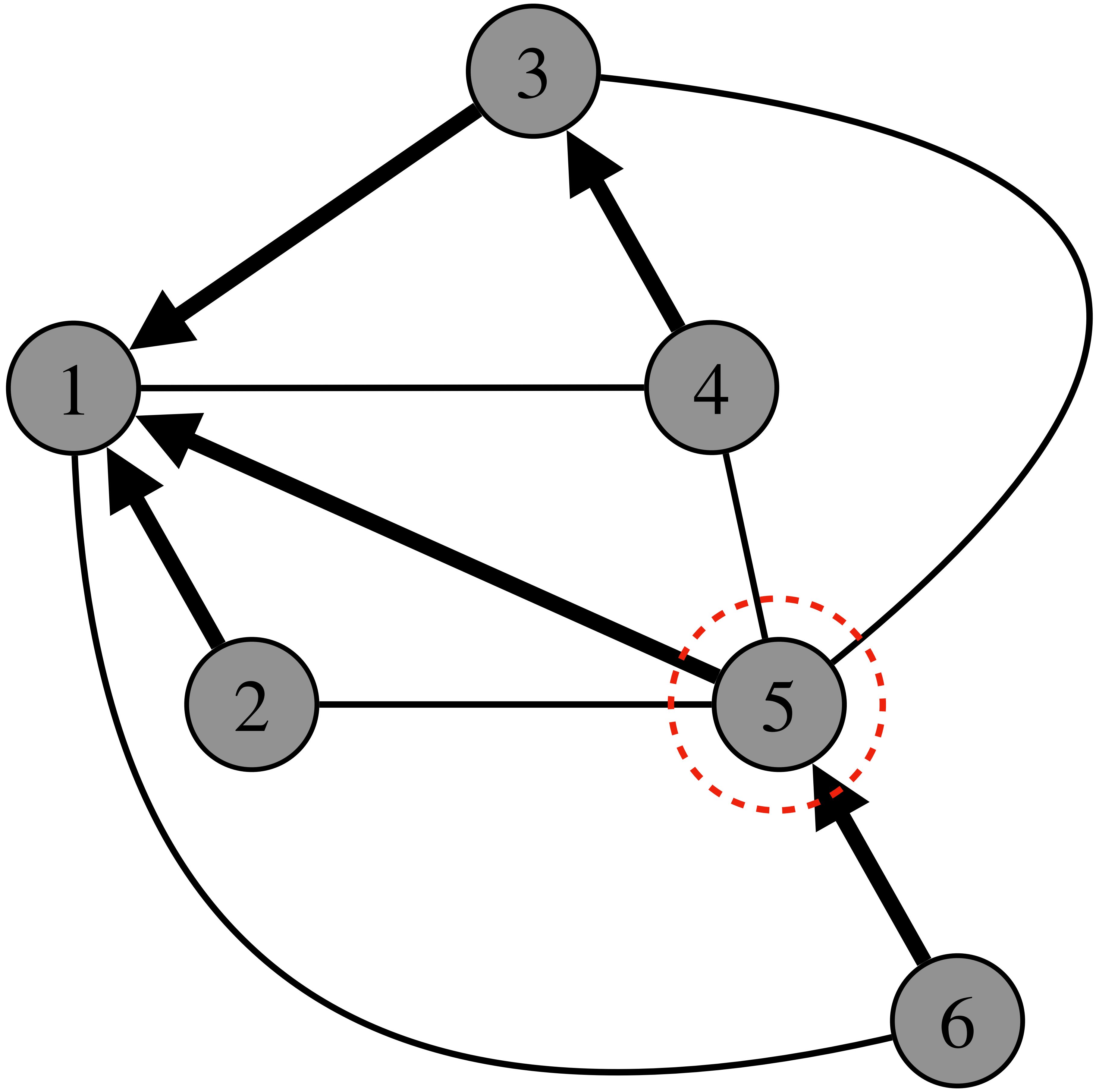


Silence Conveys Information

List $\ell = 2^{\binom{n}{2}}$ graphs on n nodes.

Round $T = i \pmod{\ell}$ denotes the i^{th} graph.

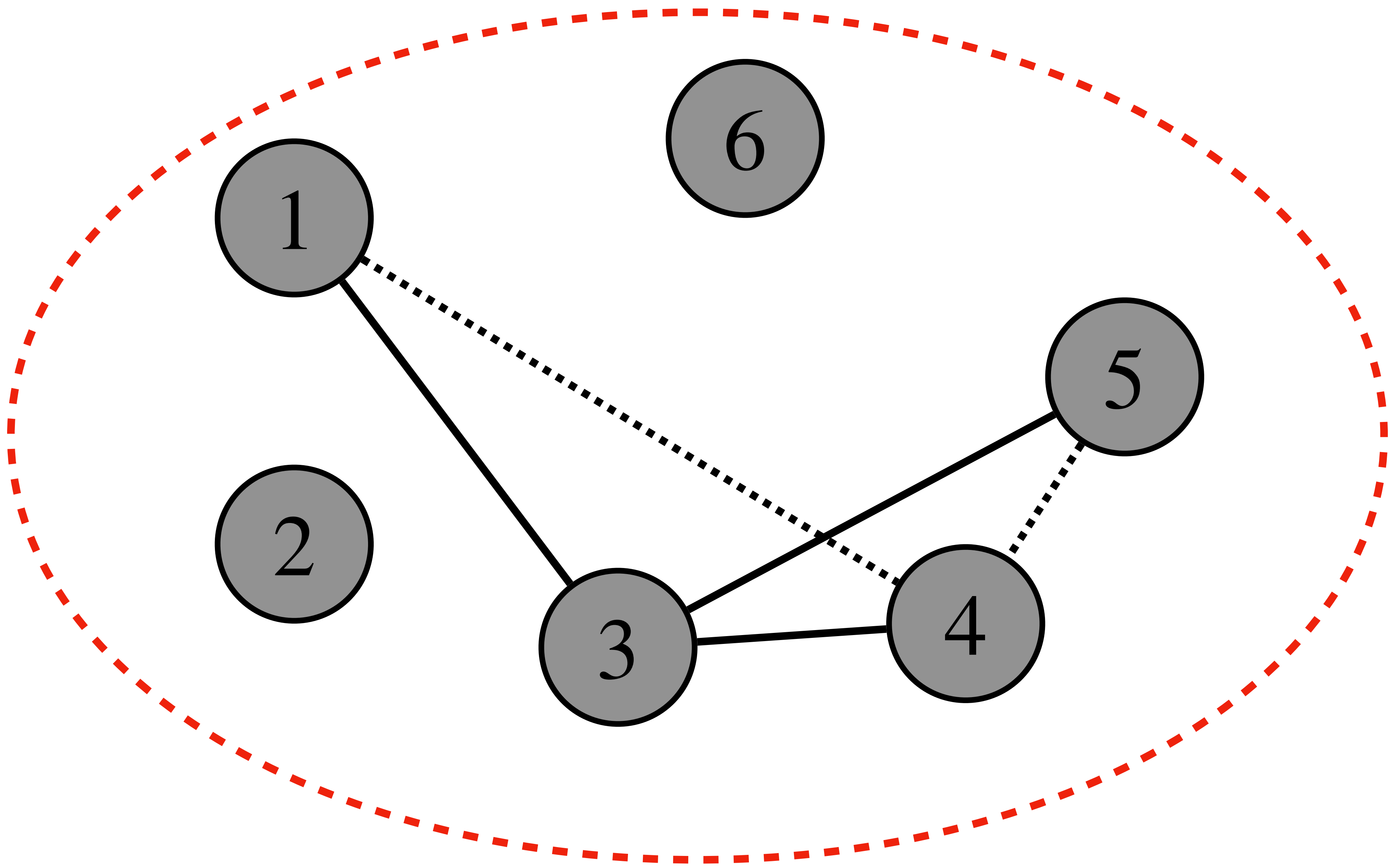


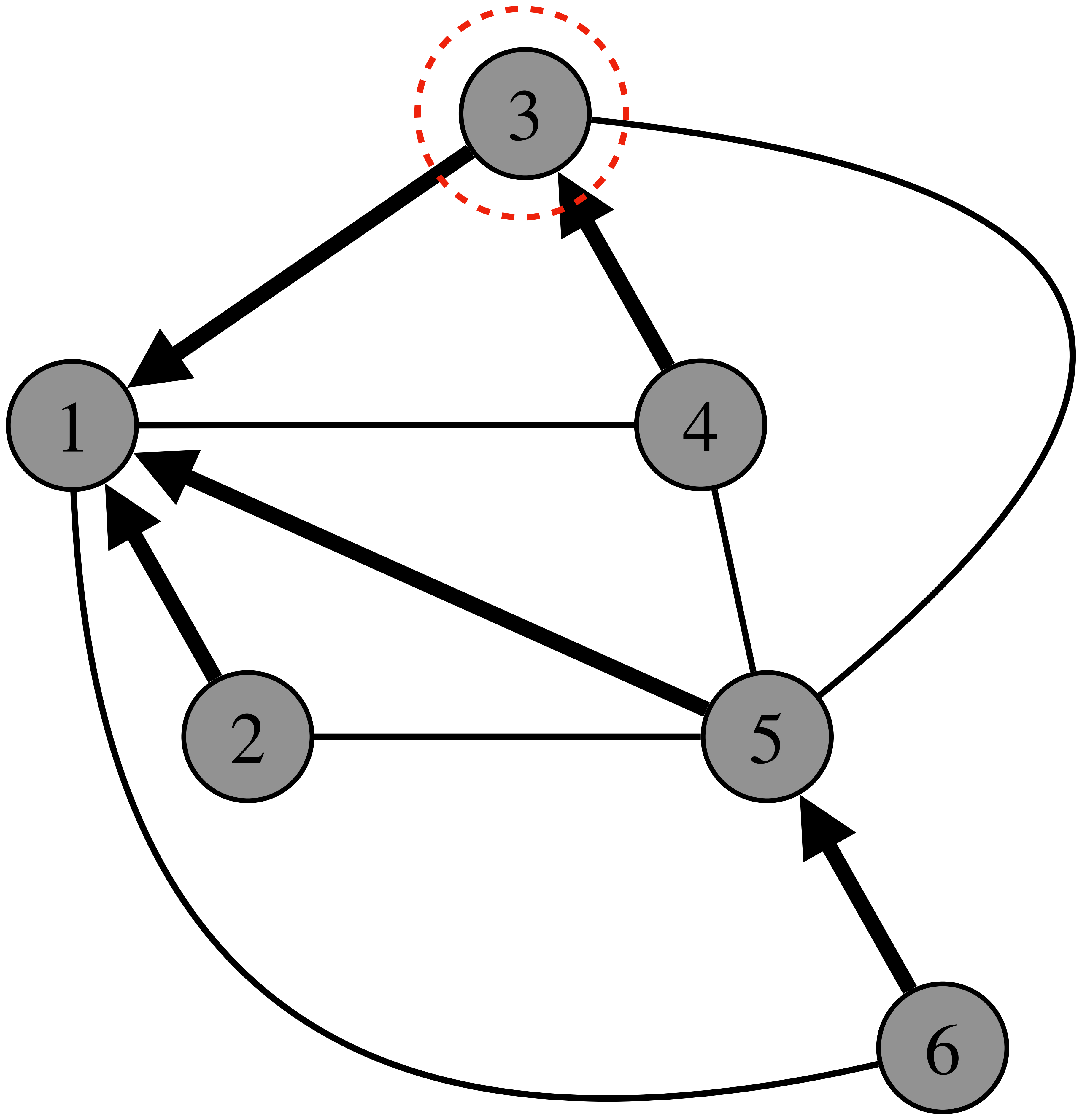


Silence Conveys Information

List $\ell = 2^{\binom{n}{2}}$ graphs on n nodes.

Round $T = i \pmod{\ell}$ denotes the i^{th} graph.

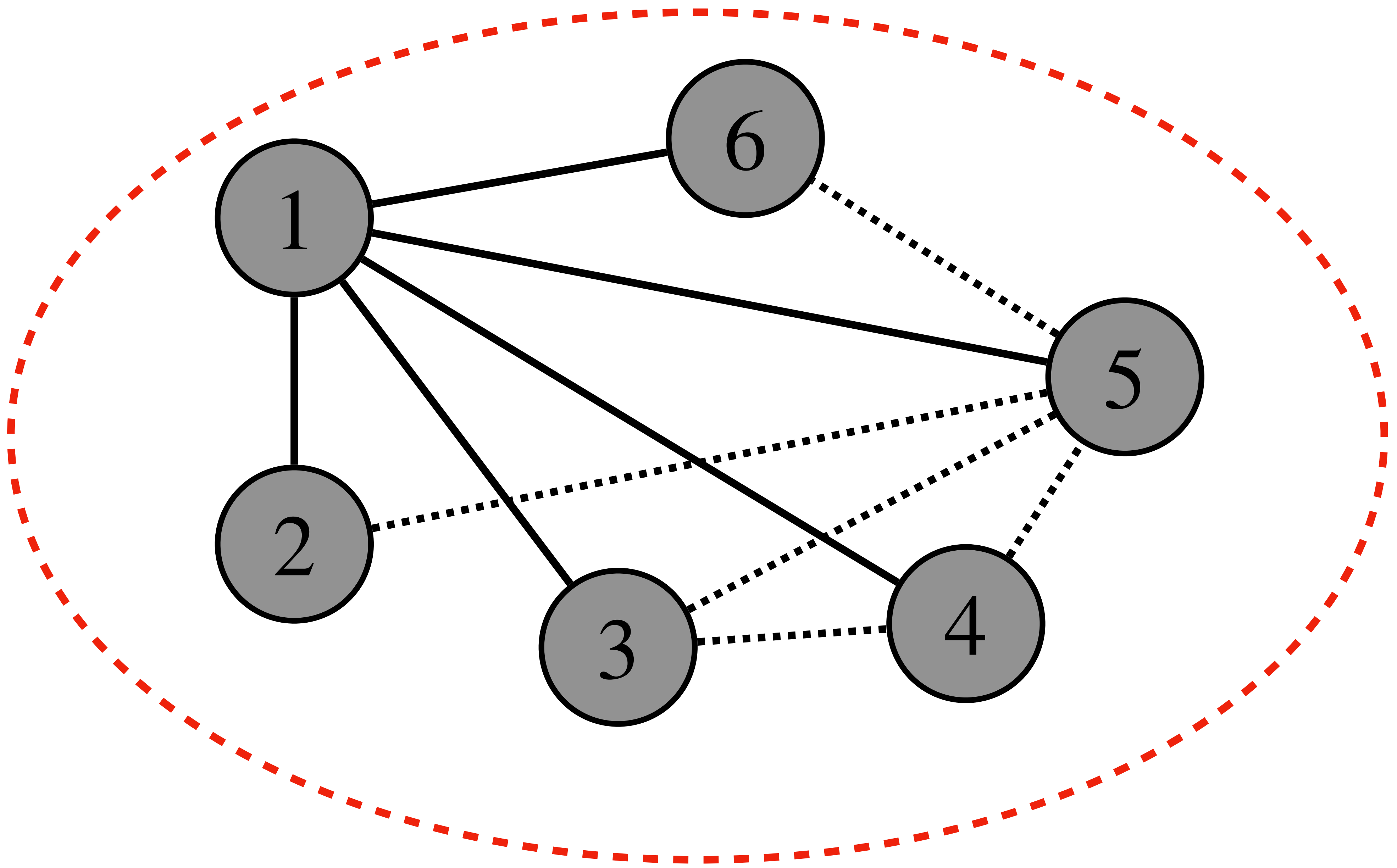


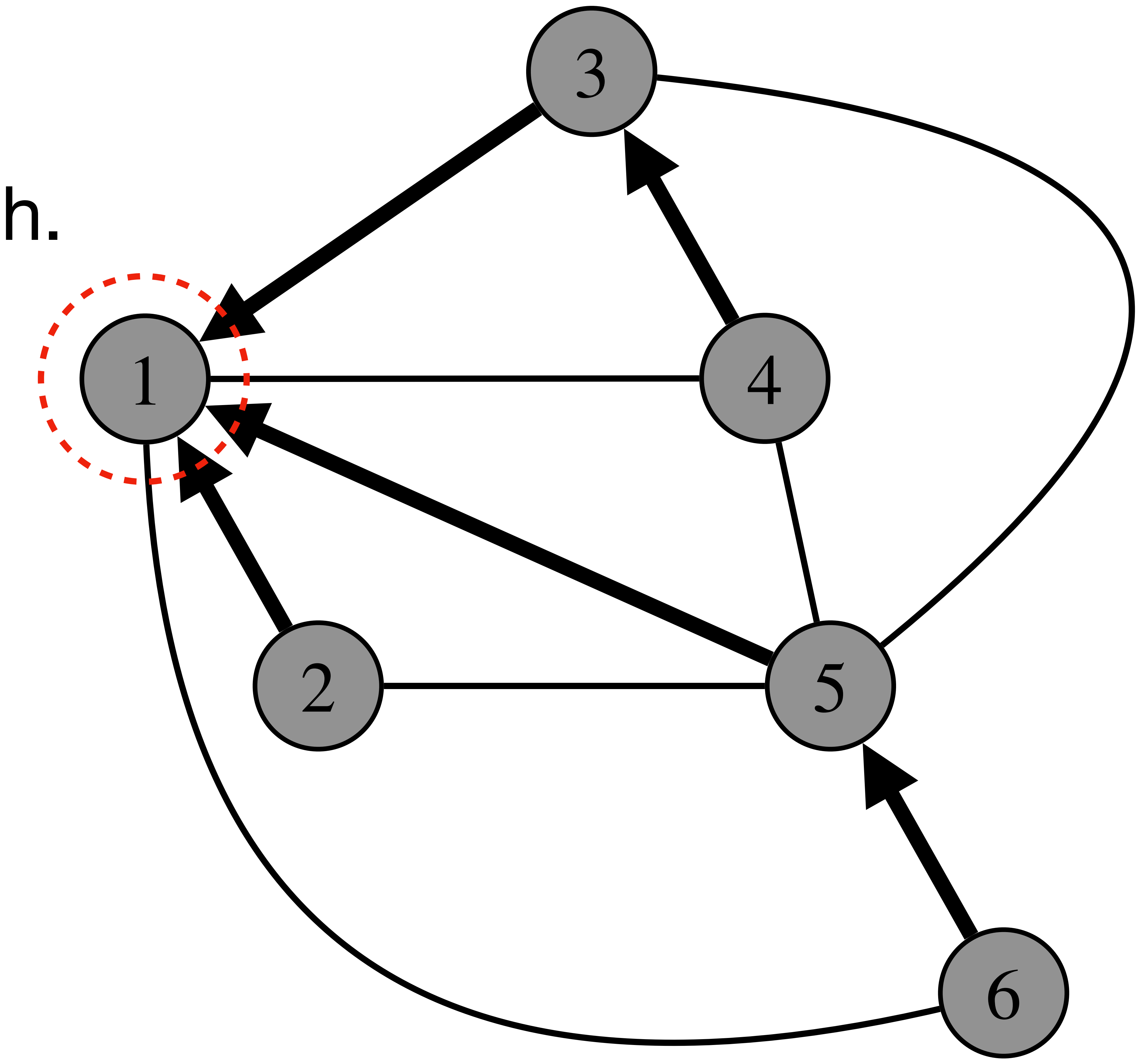


Silence Conveys Information

List $\ell = 2^{\binom{n}{2}}$ graphs on n nodes.

Round $T = i \pmod{\ell}$ denotes the i^{th} graph.

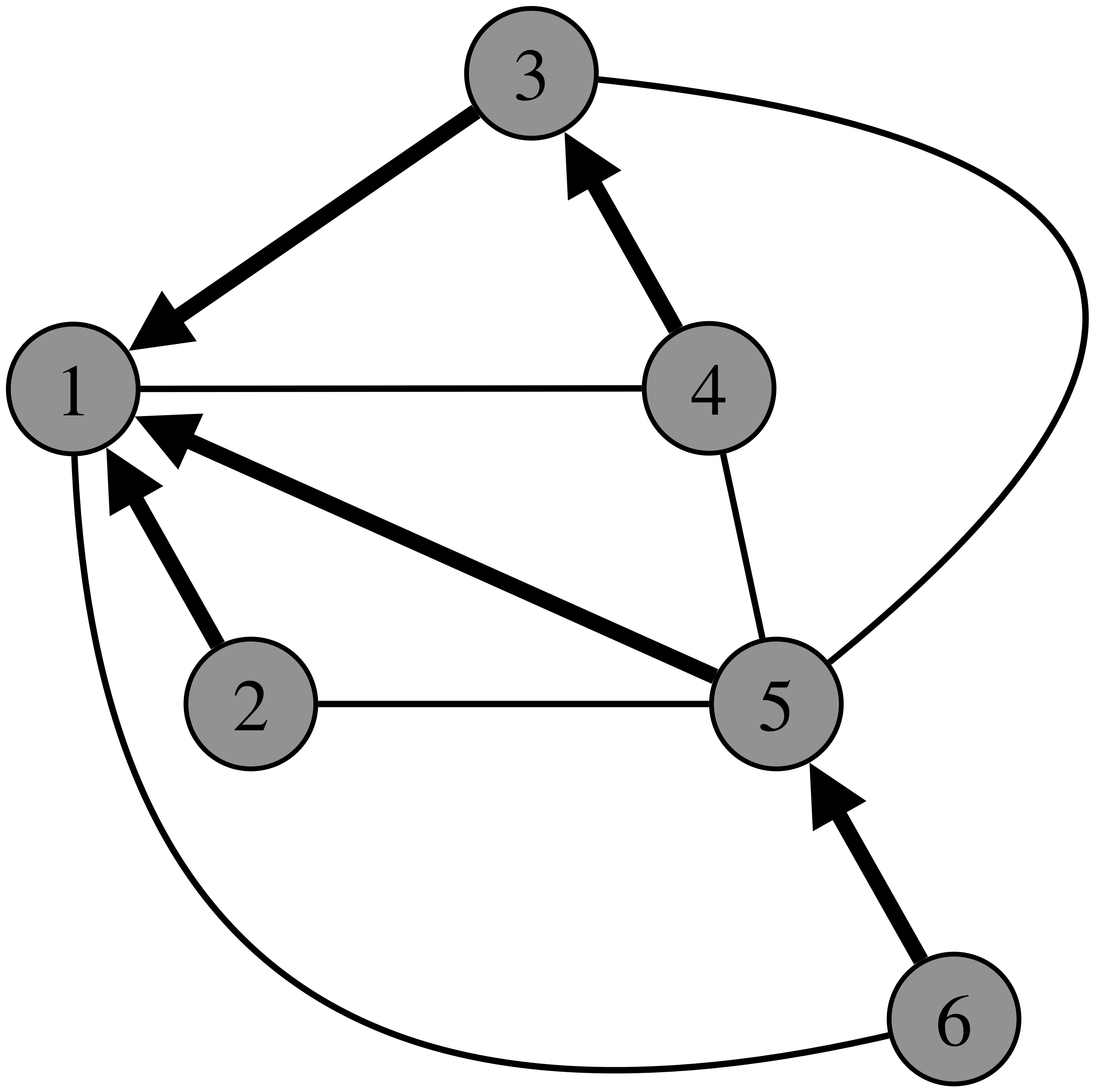




Silence Conveys Information

List $\ell = 2^{\binom{n}{2}}$ graphs on n nodes.

Round $T = i \pmod{\ell}$ denotes the i^{th} graph.



Silence Conveys Information

Silence Conveys Information

- If allowed $\exp(n^2)$ rounds, I can solve any graph problem in $\tilde{O}(n)$ messages.

Silence Conveys Information

- If allowed $\exp(n^2)$ rounds, I can solve any graph problem in $\tilde{O}(n)$ messages.
    - This can also be made deterministic!

Silence Conveys Information

- If allowed $\exp(n^2)$ rounds, I can solve any graph problem in $\tilde{O}(n)$ messages.
    - This can also be made deterministic!
- Cannot get good message lower bounds *unconditionally* in KT-1 CONGEST!

Silence Conveys Information

- If allowed $\exp(n^2)$ rounds, I can solve any graph problem in $\tilde{O}(n)$ messages.
    - This can also be made deterministic!
- Cannot get good message lower bounds *unconditionally* in KT-1 CONGEST!
- Under what conditions can we hope for message lower bounds?

KT-1 Lower Bounds

KT-1 Lower Bounds

- Restrict computation on KT-1 information.

KT-1 Lower Bounds

- Restrict computation on KT-1 information.
    - Nodes can only do comparison operations on the ID’s.

KT-1 Lower Bounds

- Restrict computation on KT-1 information.
  - Nodes can only do comparison operations on the ID's.
    - This is the **comparison-based KT-1** model [AGPV90].

[AGPV90] Awerbuch, Peleg, Goldreich, Varnish. JACM

KT-1 Lower Bounds

- Restrict computation on KT-1 information.
    - Nodes can only do comparison operations on the ID's.
        - This is the **comparison-based KT-1** model [AGPV90].
        - Here, we can still do indistinguishability type arguments.

[AGPV90] Awerbuch, Peleg, Goldreich, Varnish. JACM

KT-1 Lower Bounds

- Restrict computation on KT-1 information.
    - Nodes can only do comparison operations on the ID's.
        - This is the **comparison-based KT-1** model [AGPV90].
        - Here, we can still do indistinguishability type arguments.
- Restrict to algorithms that use few rounds (maybe poly(n) or $\log n$ rounds)

[AGPV90] *Awerbuch, Peleg, Goldreich, Varnish. JACM*

KT-1 Lower Bounds

- Restrict computation on KT-1 information.
  - Nodes can only do comparison operations on the ID's.
    - This is the **comparison-based KT-1** model [AGPV90].
    - Here, we can still do indistinguishability type arguments.
- Restrict to algorithms that use few rounds (maybe poly(n) or $\log n$ rounds)
  - Can be achieved by communication complexity reductions.

[AGPV90] Awerbuch, Peleg, Goldreich, Varnish. JACM

To $\Omega(m)$ and Beyond

To $\Omega(m)$ and Beyond

To $\Omega(m)$ and Beyond

- For comparison based KT-1 algorithms, $\Omega(m)$ message lower bound for:
    - $(\Delta + 1)$-coloring.
    - Maximal Independent Set.

[PPP+21] Pai, Pandurangan, Pemmaraju, Robinson. PODC 2021

To $\Omega(m)$ and Beyond

- For comparison based KT-1 algorithms, $\Omega(m)$ message lower bound for:
  - $(\Delta + 1)$-coloring.
  - Maximal Independent Set.

[PPP+21] Pai, Pandurangan, Pemmaraju, Robinson. PODC 2021

To $\Omega(m)$ and Beyond

- For comparison based KT-1 algorithms, $\Omega(m)$ message lower bound for:
  - $(\Delta + 1)$-coloring.
  - Maximal Independent Set.

[PPP+21] Pai, Pandurangan, Pemmaraju, Robinson. PODC 2021

- For poly(n) round KT-1 algorithms, $\Omega(m \cdot D)$ message lower bound for:
  - Minimum Vertex Cover.
  - Maximum Independent Set.
  - Minimum Dominating Set.

[DPP+24] Dufoulon, Pai, Pandurangan, Pemmaraju, Robinson. ITCS 2024

To $\Omega(m)$ and Beyond

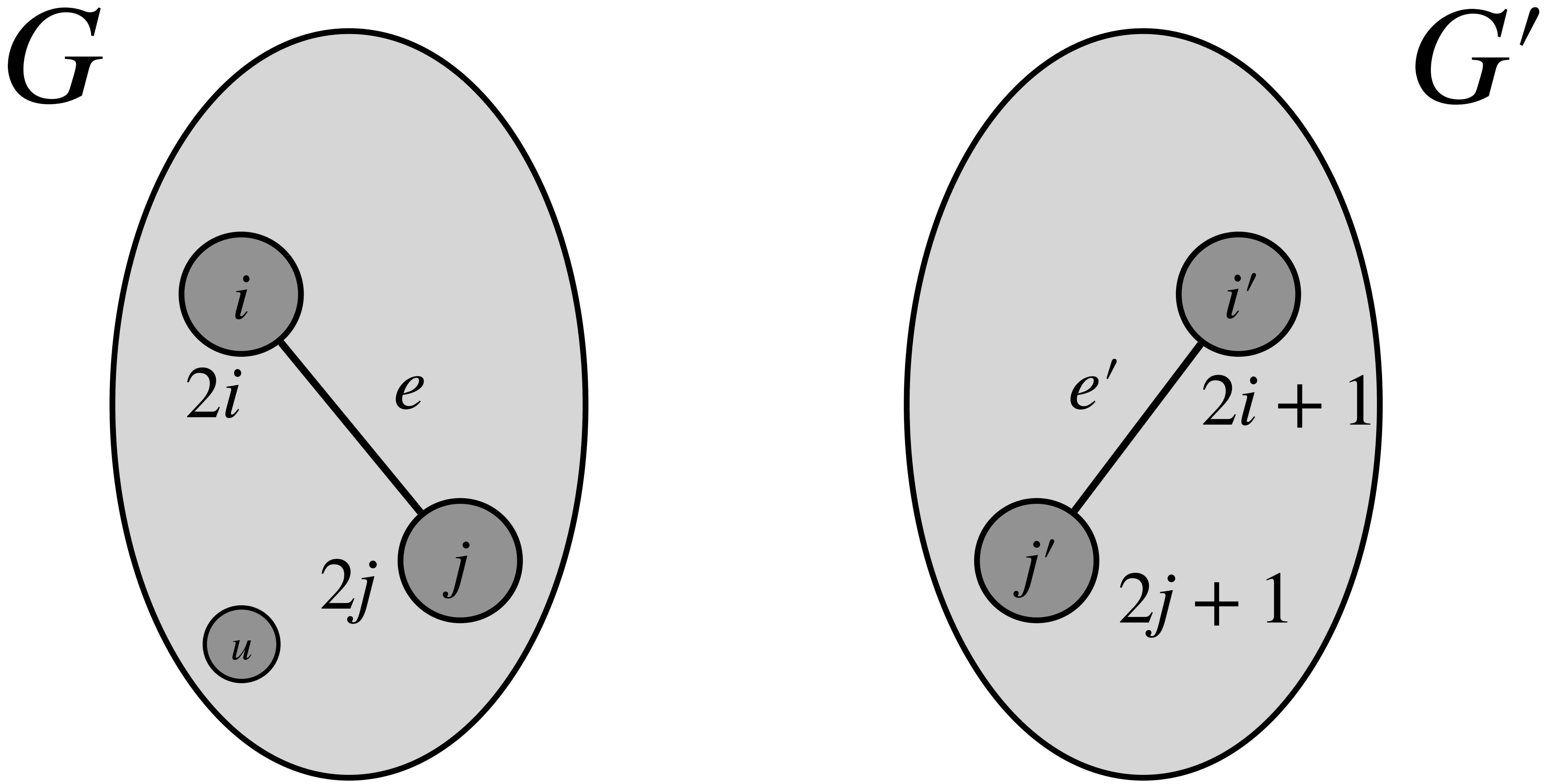
- For comparison based KT-1 algorithms, $\Omega(m)$ message lower bound for:
  - $(\Delta + 1)$-coloring.
  - Maximal Independent Set.

[PPP+21] Pai, Pandurangan, Pemmaraju, Robinson. PODC 2021

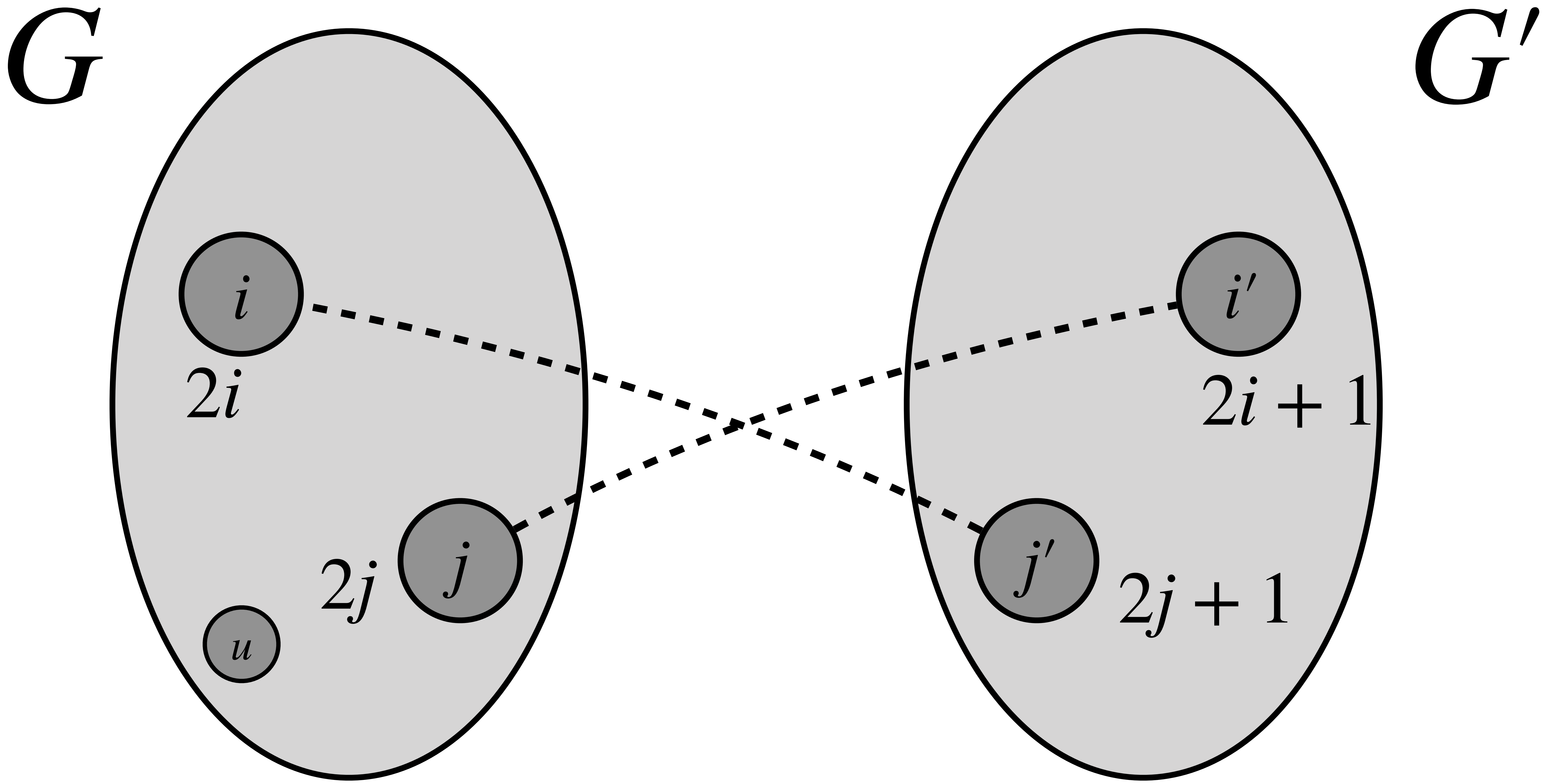
- For poly(n) round KT-1 algorithms, $\Omega(m \cdot D)$ message lower bound for:
  - Minimum Vertex Cover.
  - Maximum Independent Set.
  - Minimum Dominating Set.

[DPP+24] Dufoulon, Pai, Pandurangan, Pemmaraju, Robinson. ITCS 2024

Comparison Based Broadcast

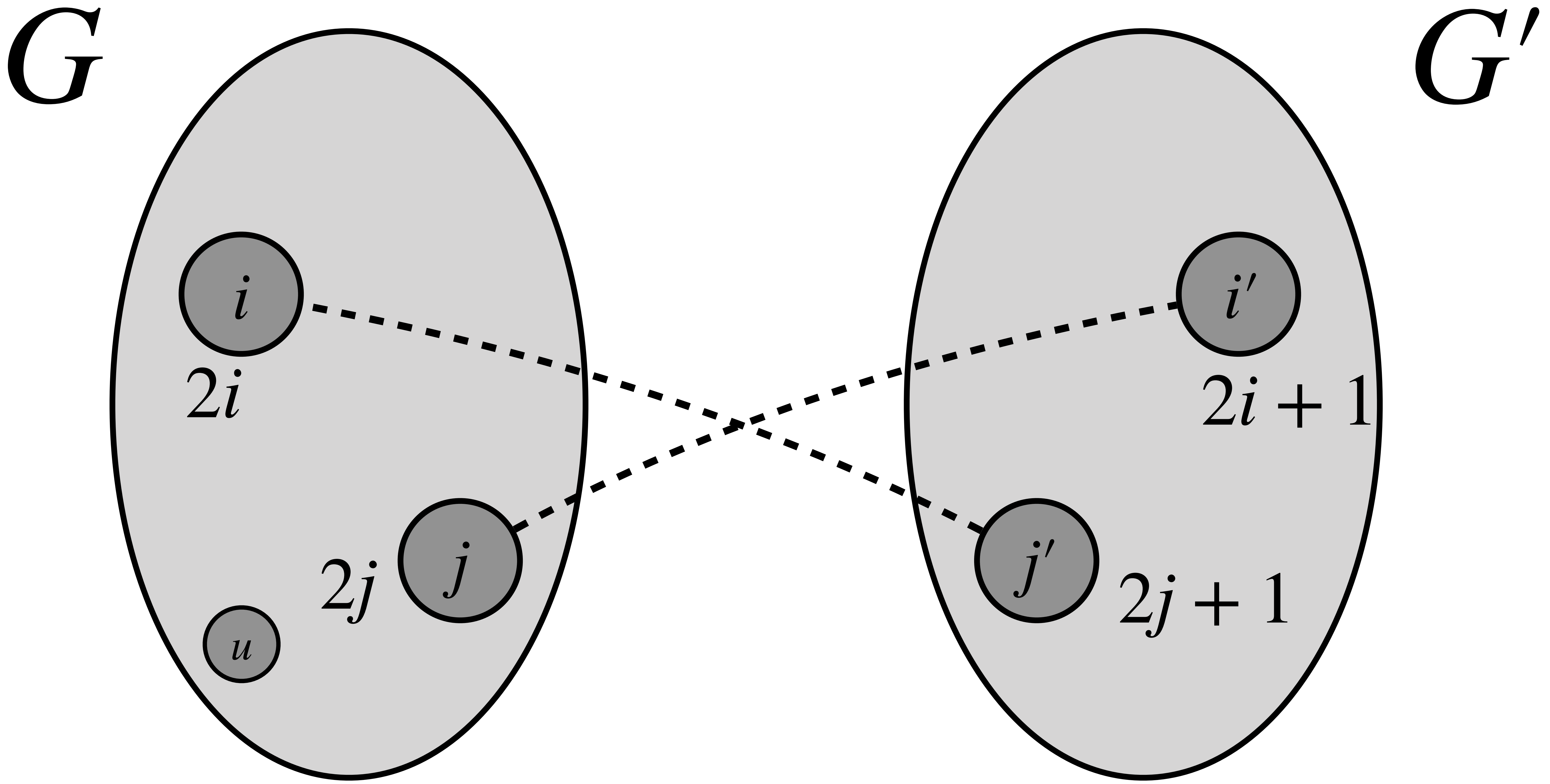


Comparison Based Broadcast



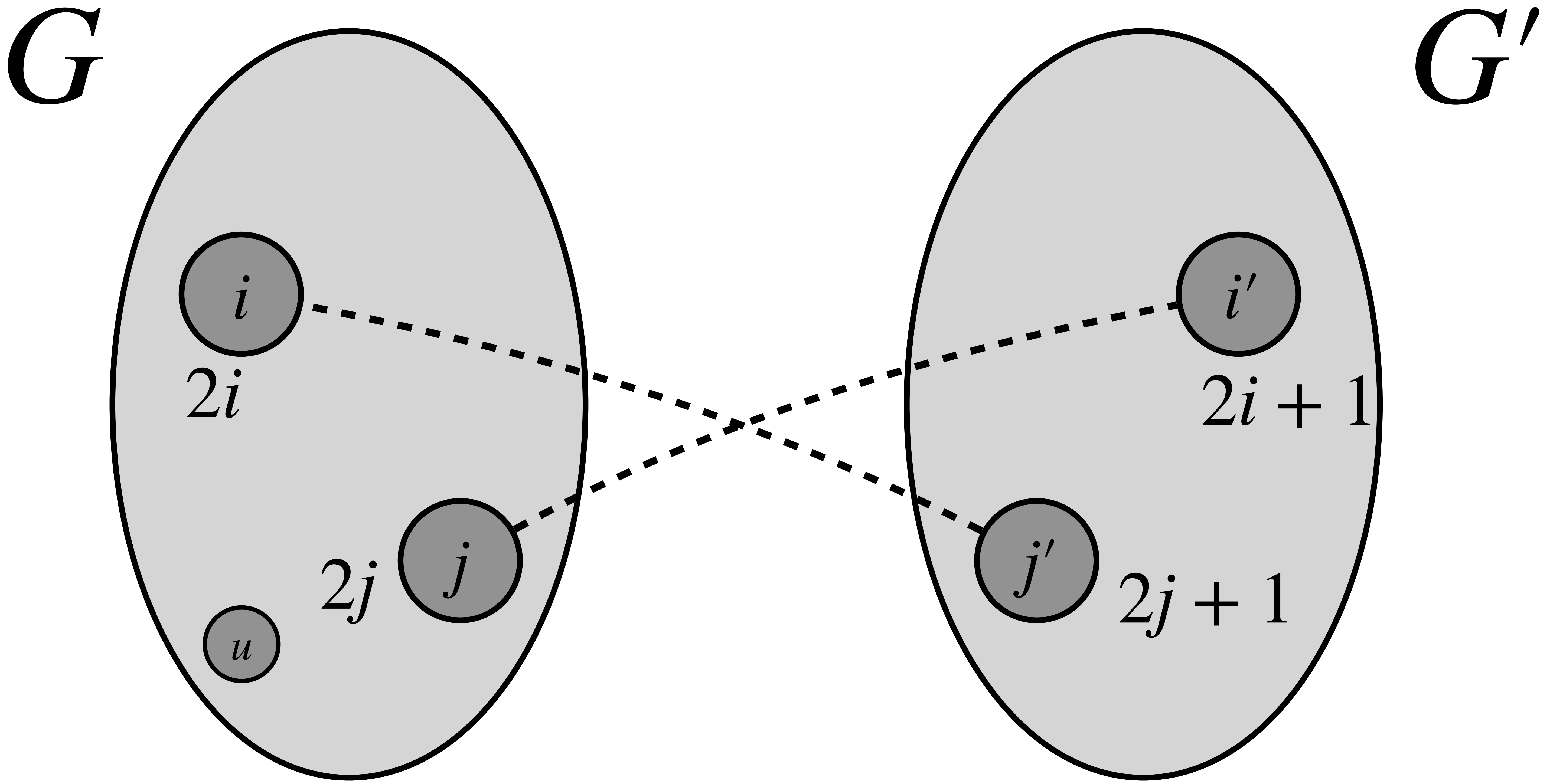
Comparison Based Broadcast

- Initial knowledge is different!



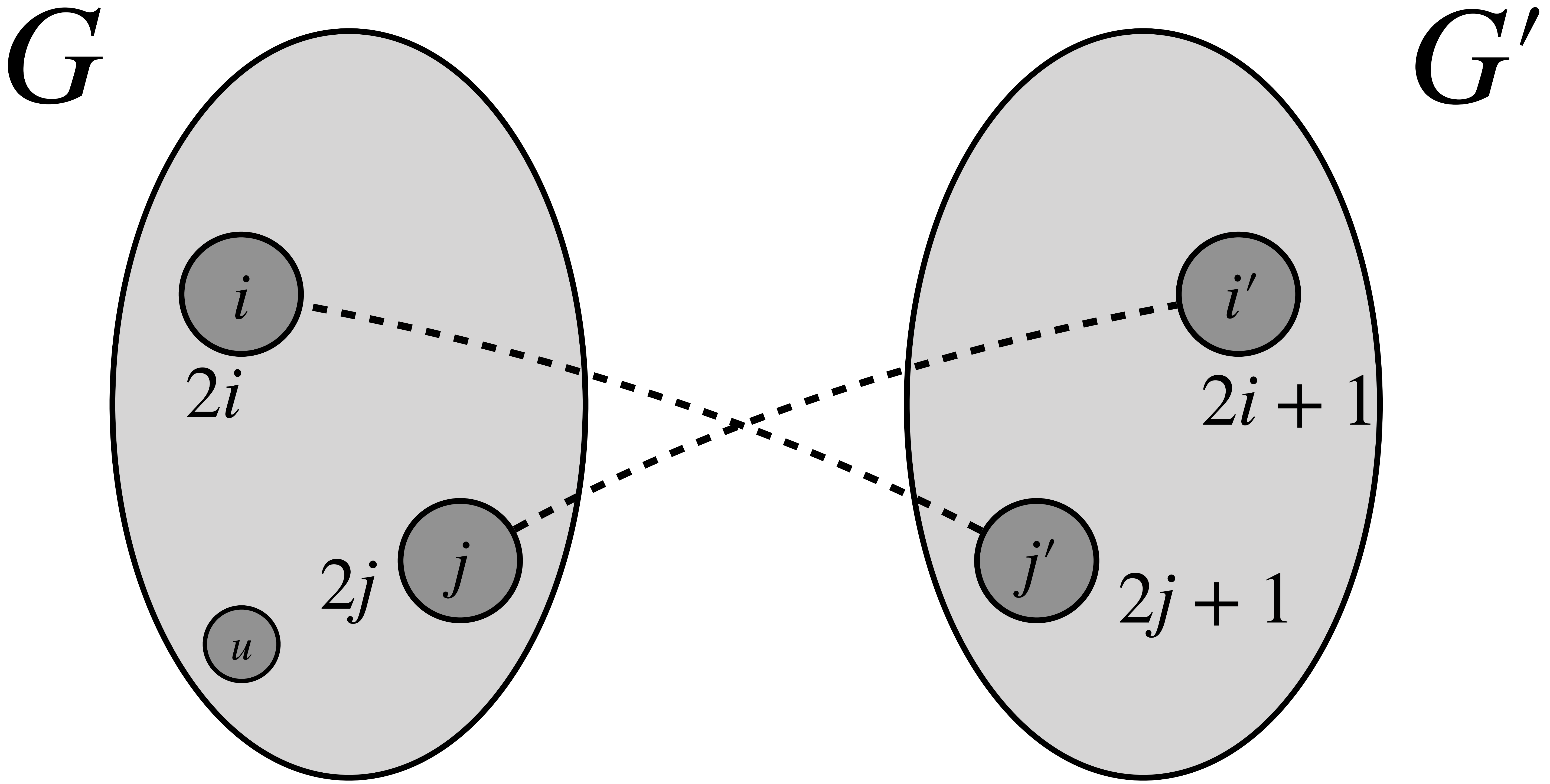
Comparison Based Broadcast

- Initial knowledge is different!
- Node i sees ID $2j$ in one graph and ID $2j + 1$ in the other.



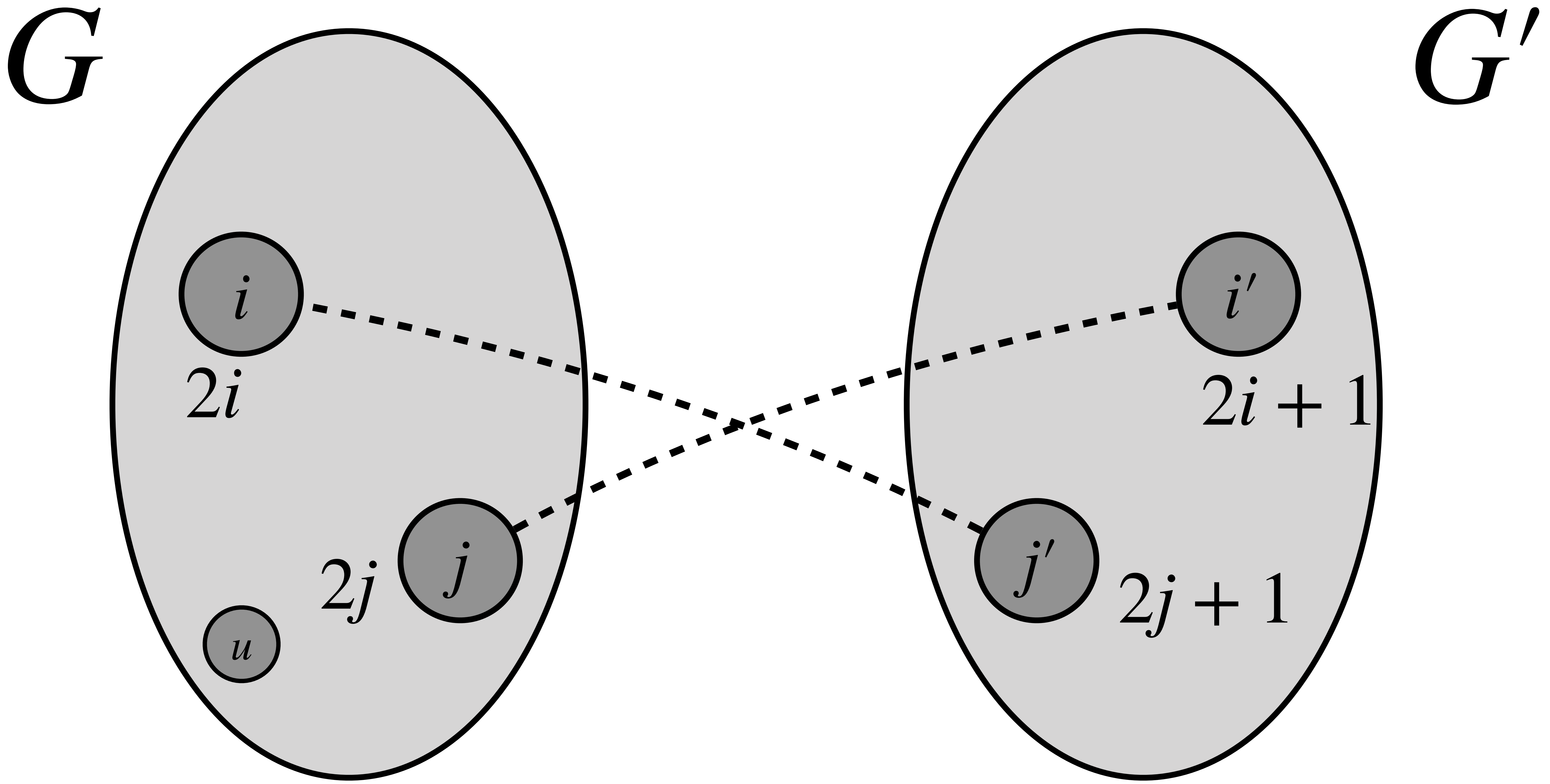
Comparison Based Broadcast

- Initial knowledge is different!
- Node i sees ID $2j$ in one graph and ID $2j+1$ in the other.
- Order of IDs is the same!



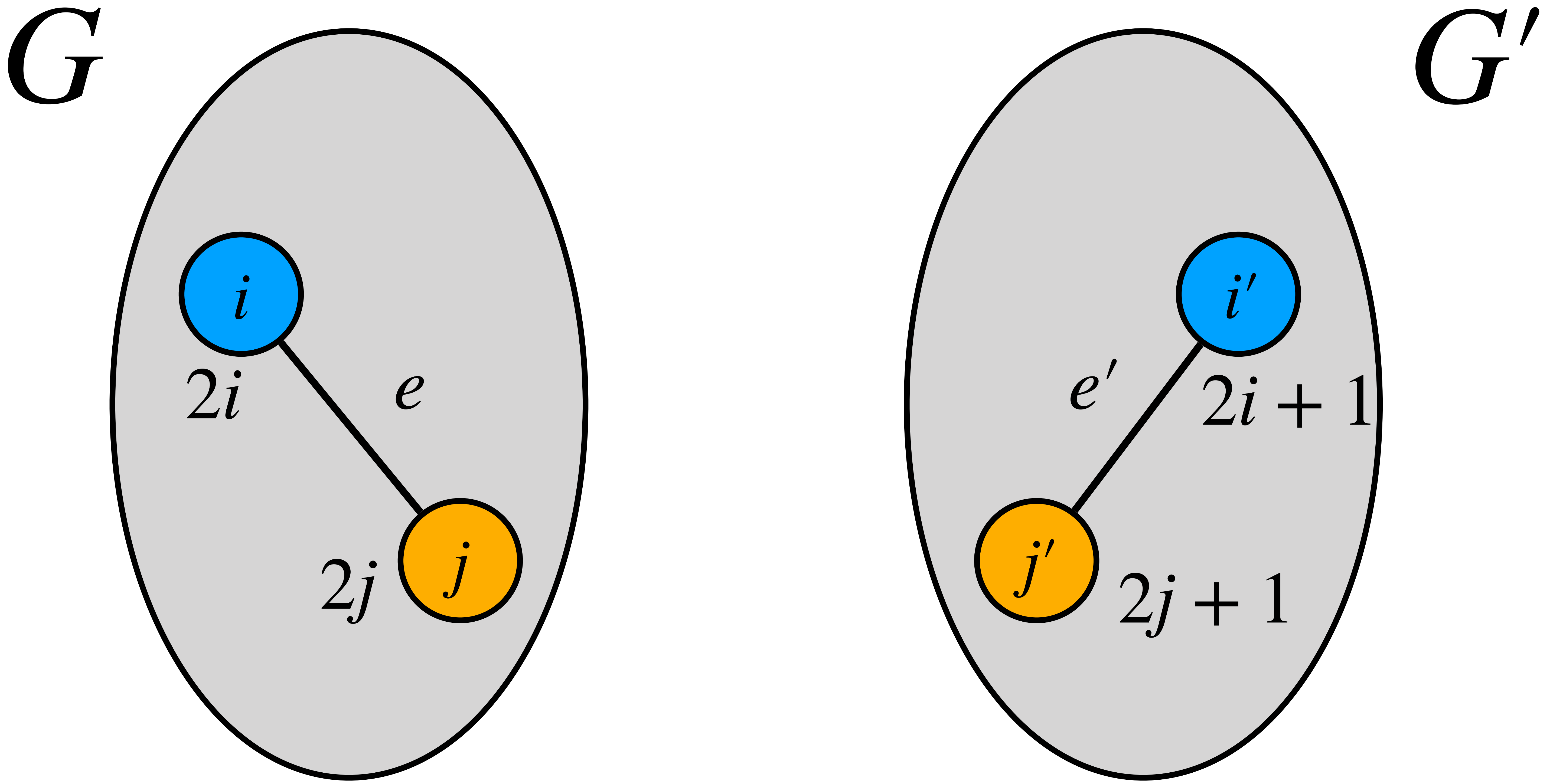
Comparison Based Broadcast

- Initial knowledge is different!
- Node i sees ID $2j$ in one graph and ID $2j+1$ in the other.
- Order of IDs is the same!
- So we can get an $\Omega(m)$ message lower bound.

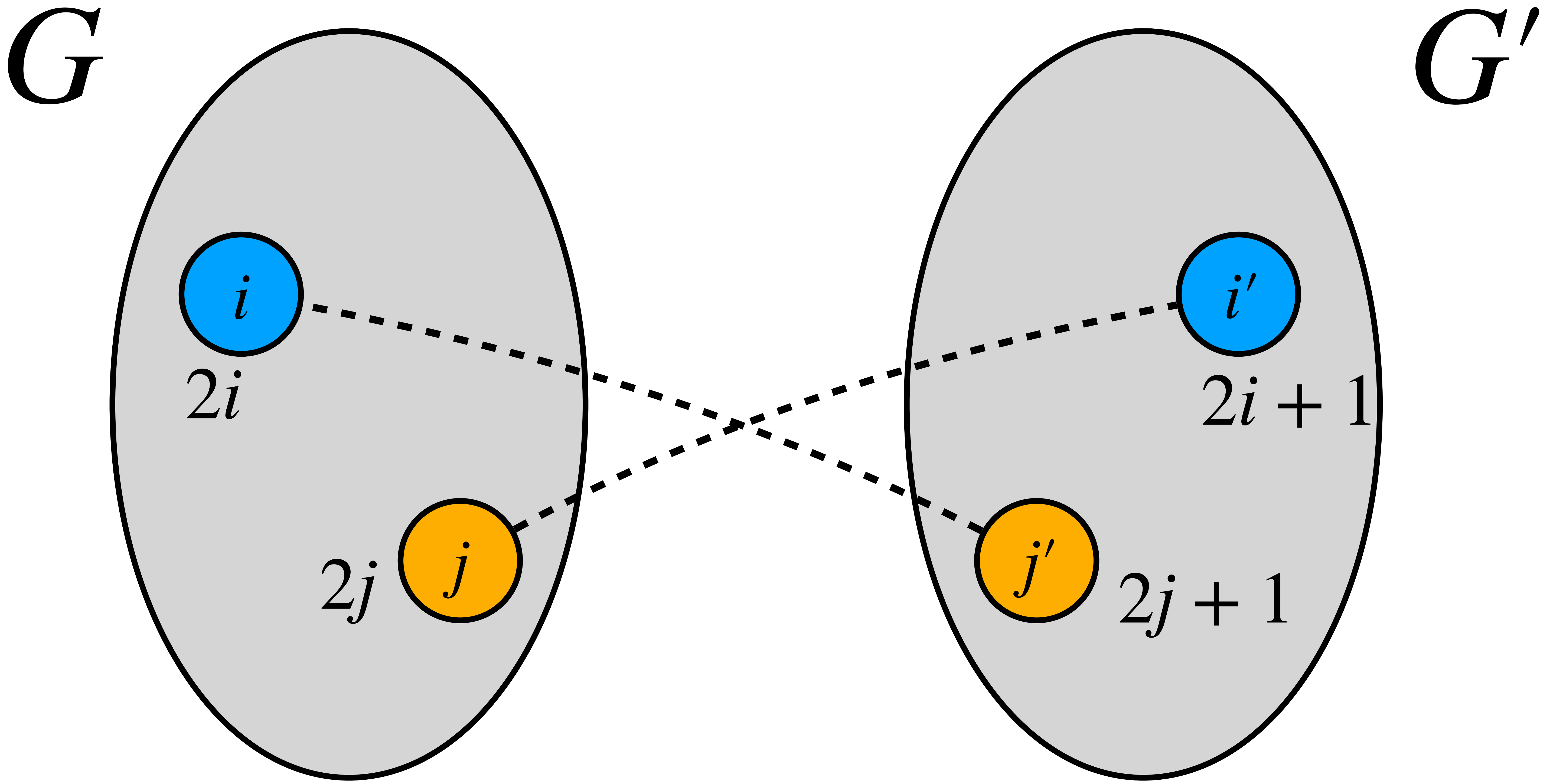


[AGPV90] *Awerbuch, Peleg, Goldreich, Varnish. JACM*

Extending to Coloring

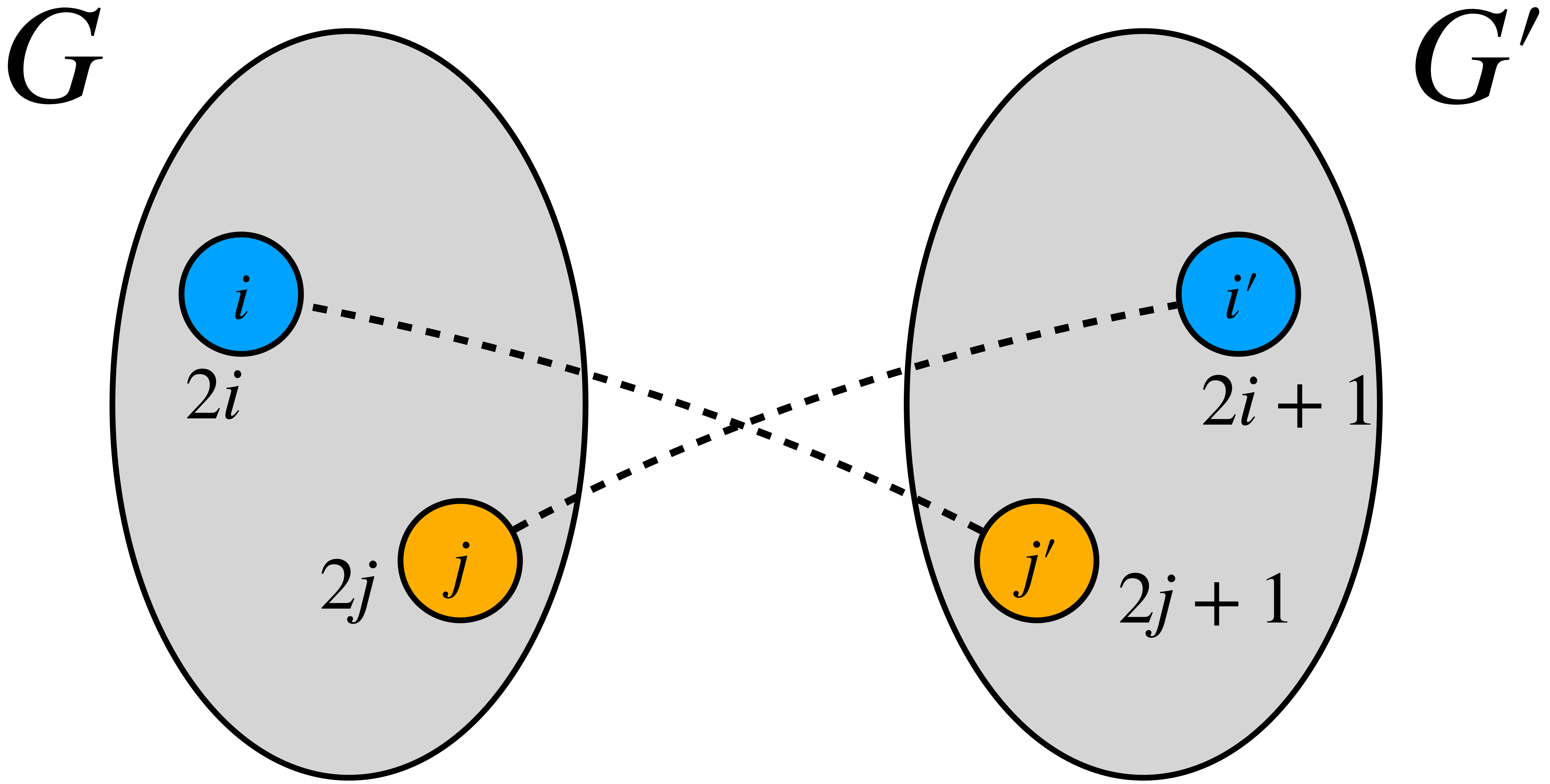


Extending to Coloring



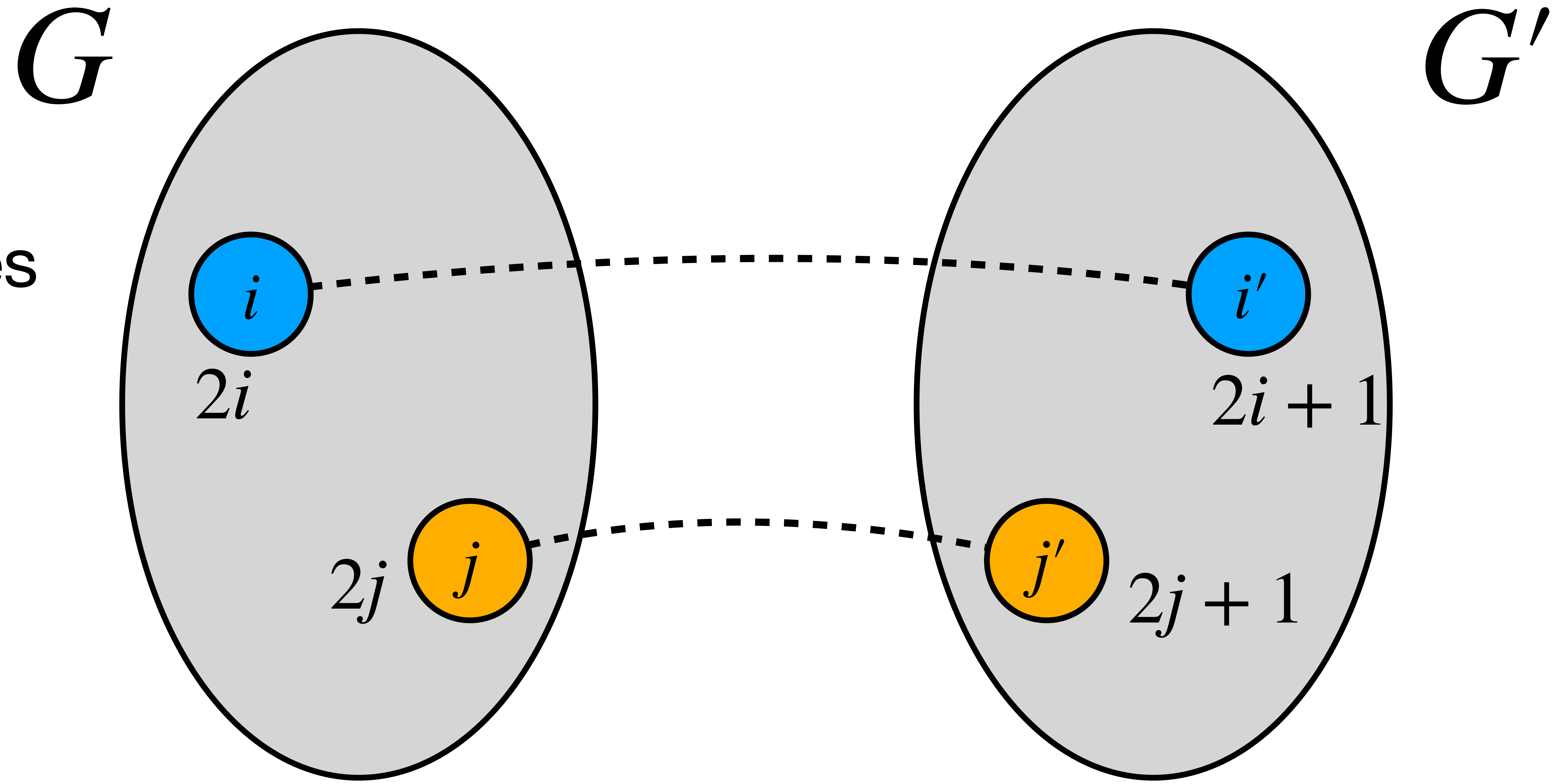
Extending to Coloring

- Don't get any contradiction :(



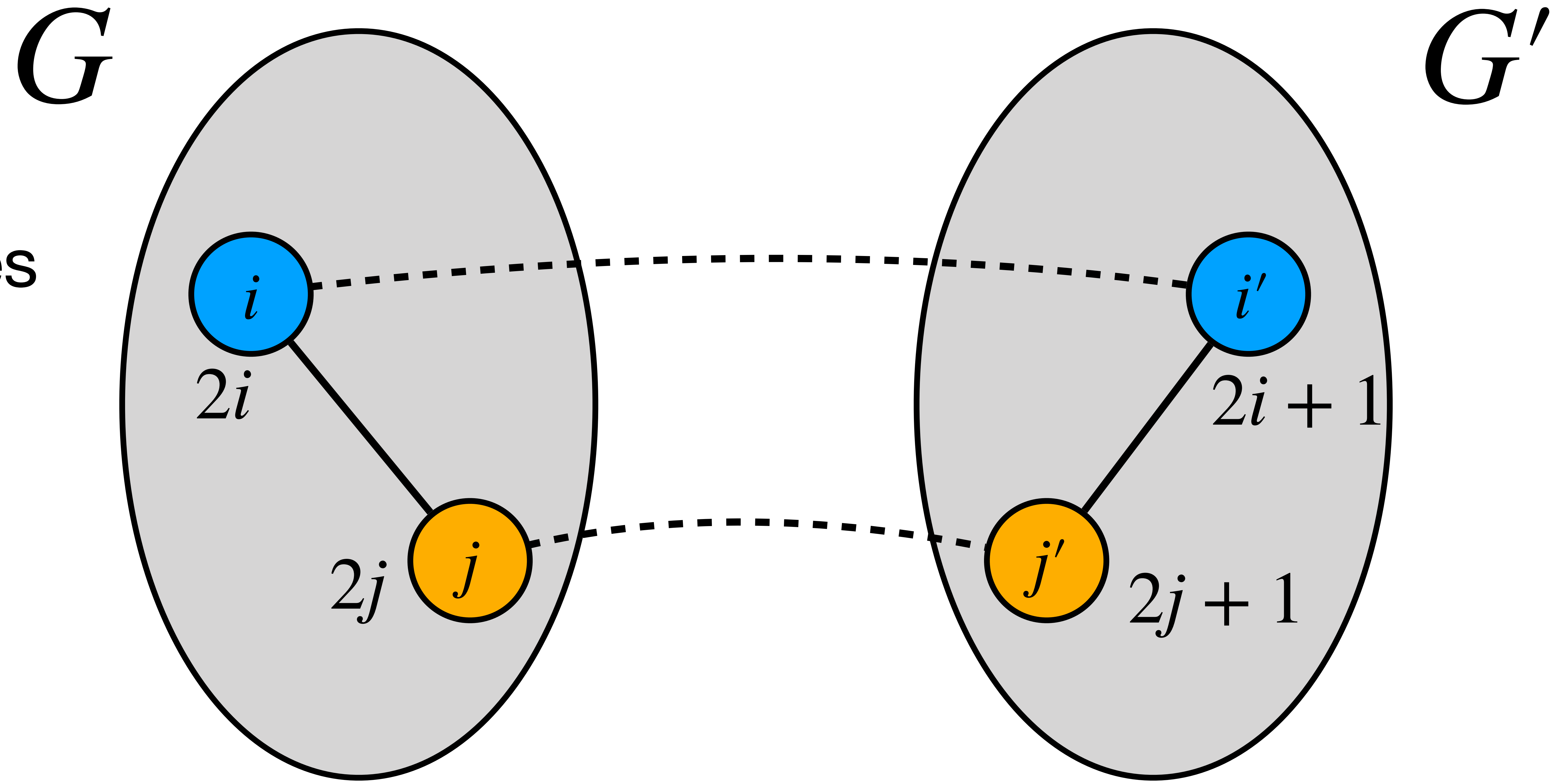
Extending to Coloring

- Don’t get any contradiction :(
- Want to add the “parallel” edges to get contradiction.

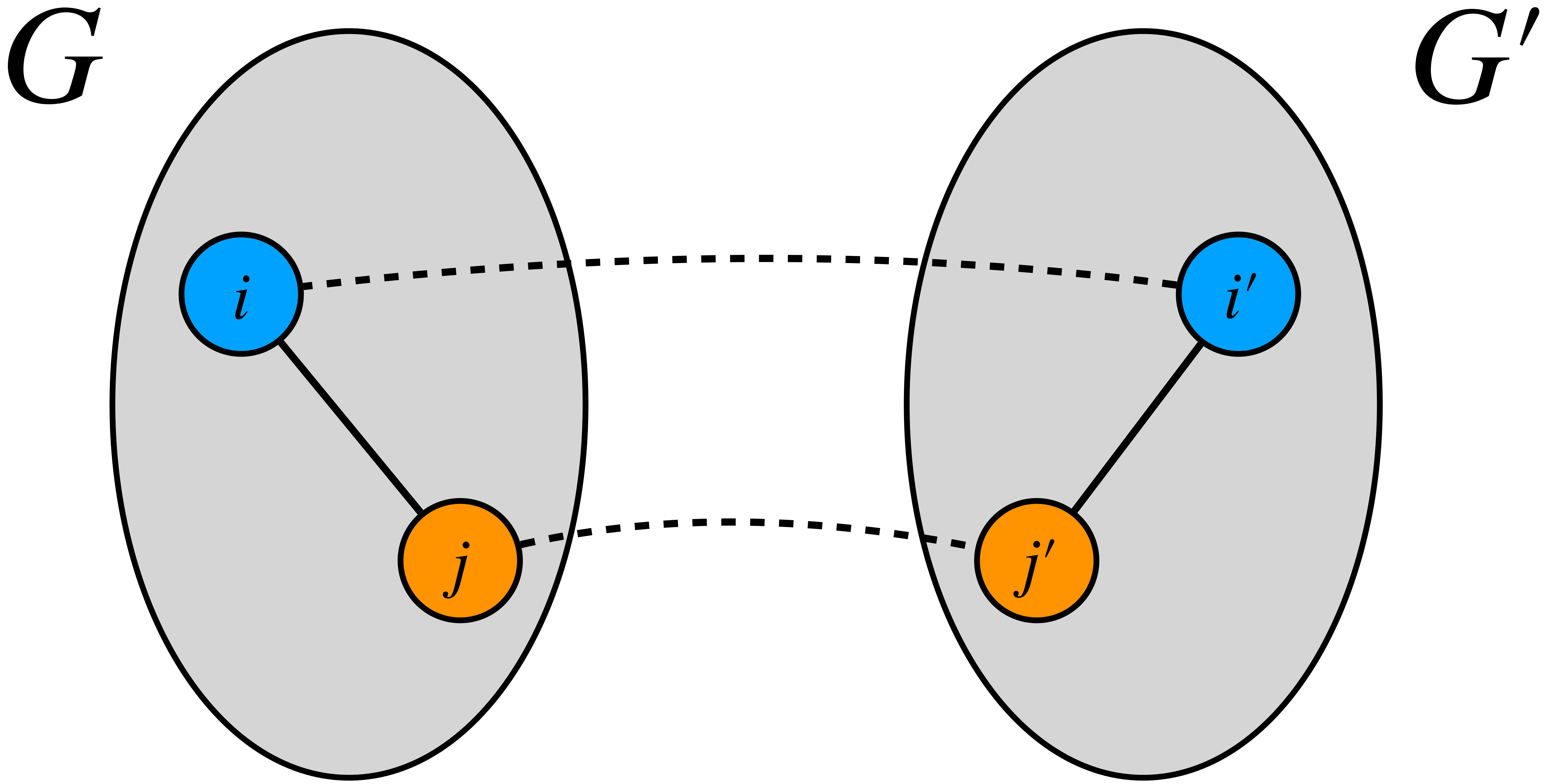


Extending to Coloring

- Don't get any contradiction :(
- Want to add the "parallel" edges to get contradiction.
- But $2i + 1$ can be way out of order compared to $2j$...

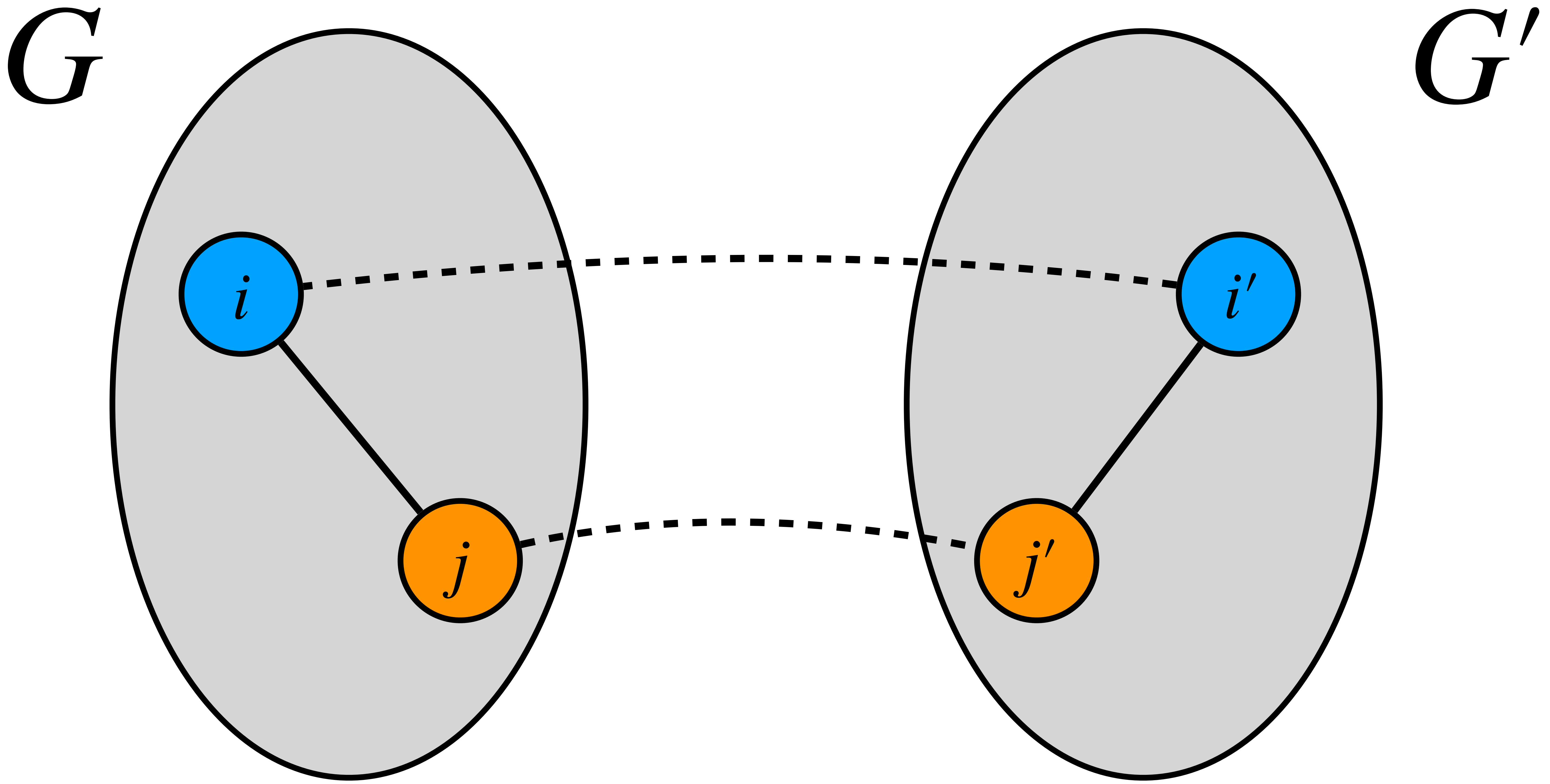


Comparison Based Coloring



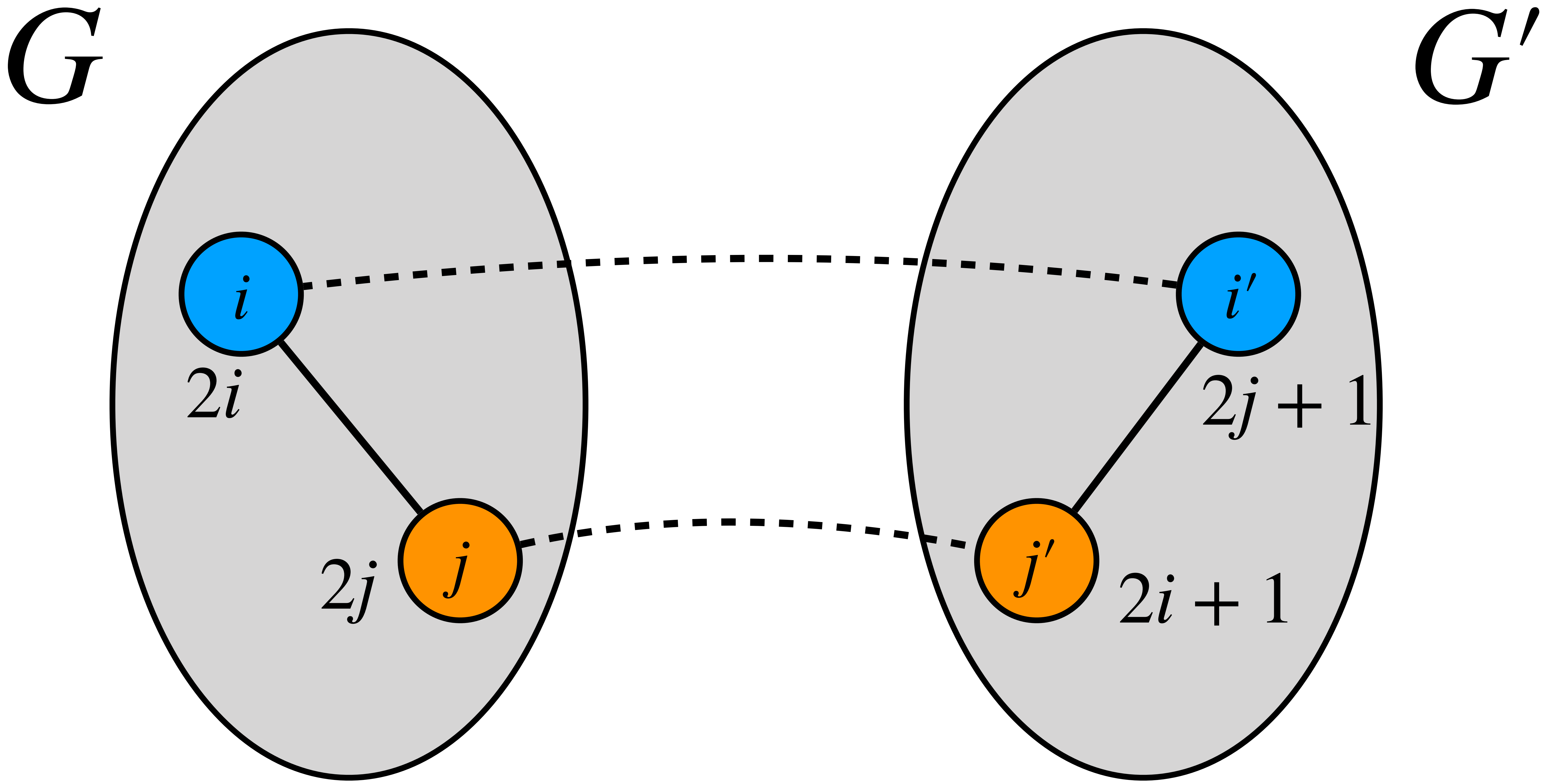
Comparison Based Coloring

- “Shift” the IDs in G' such that



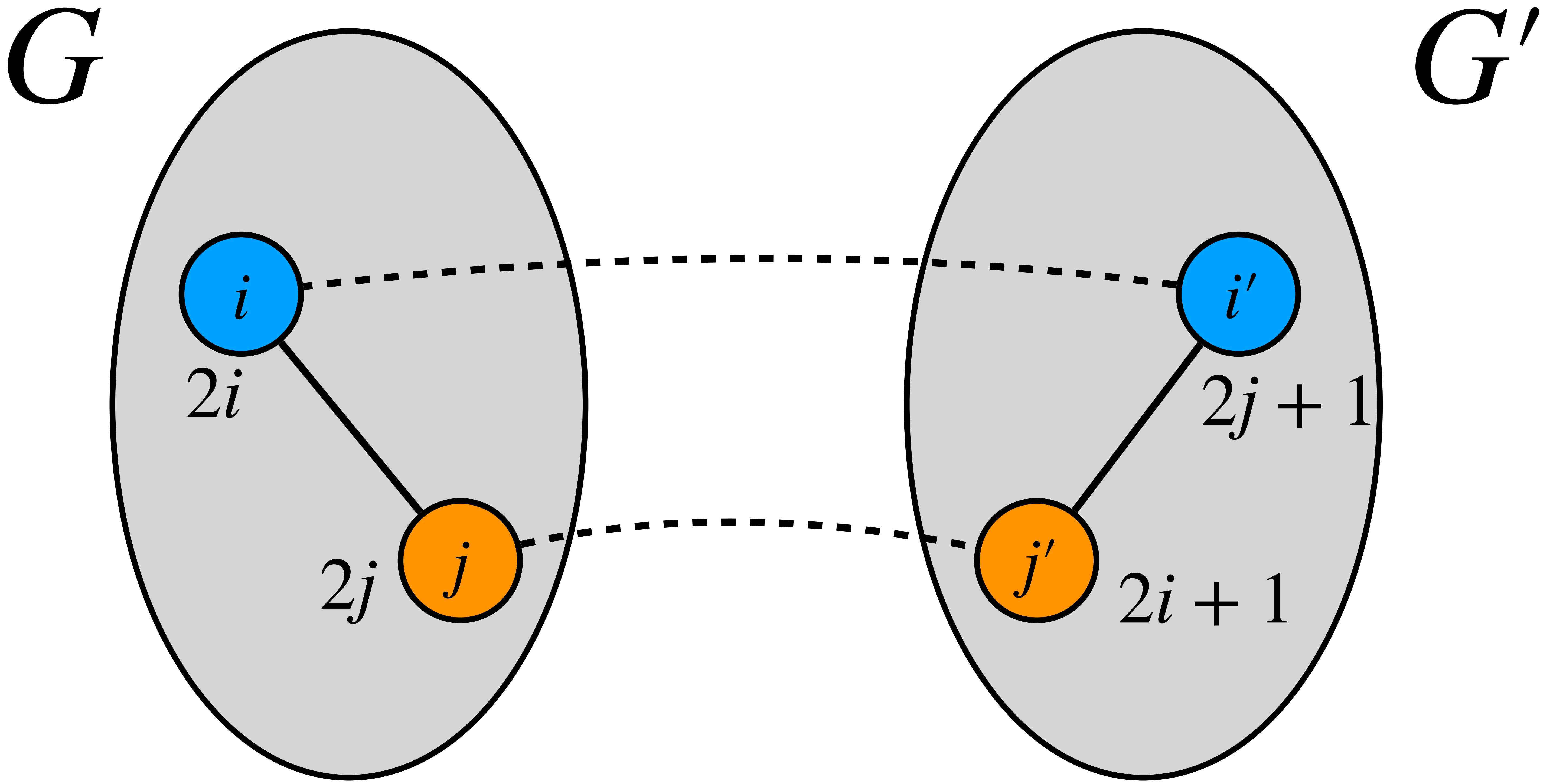
Comparison Based Coloring

- “Shift” the IDs in G' such that
  - i' gets ID $2j + 1$ and
  - j' gets ID $2i + 1$.



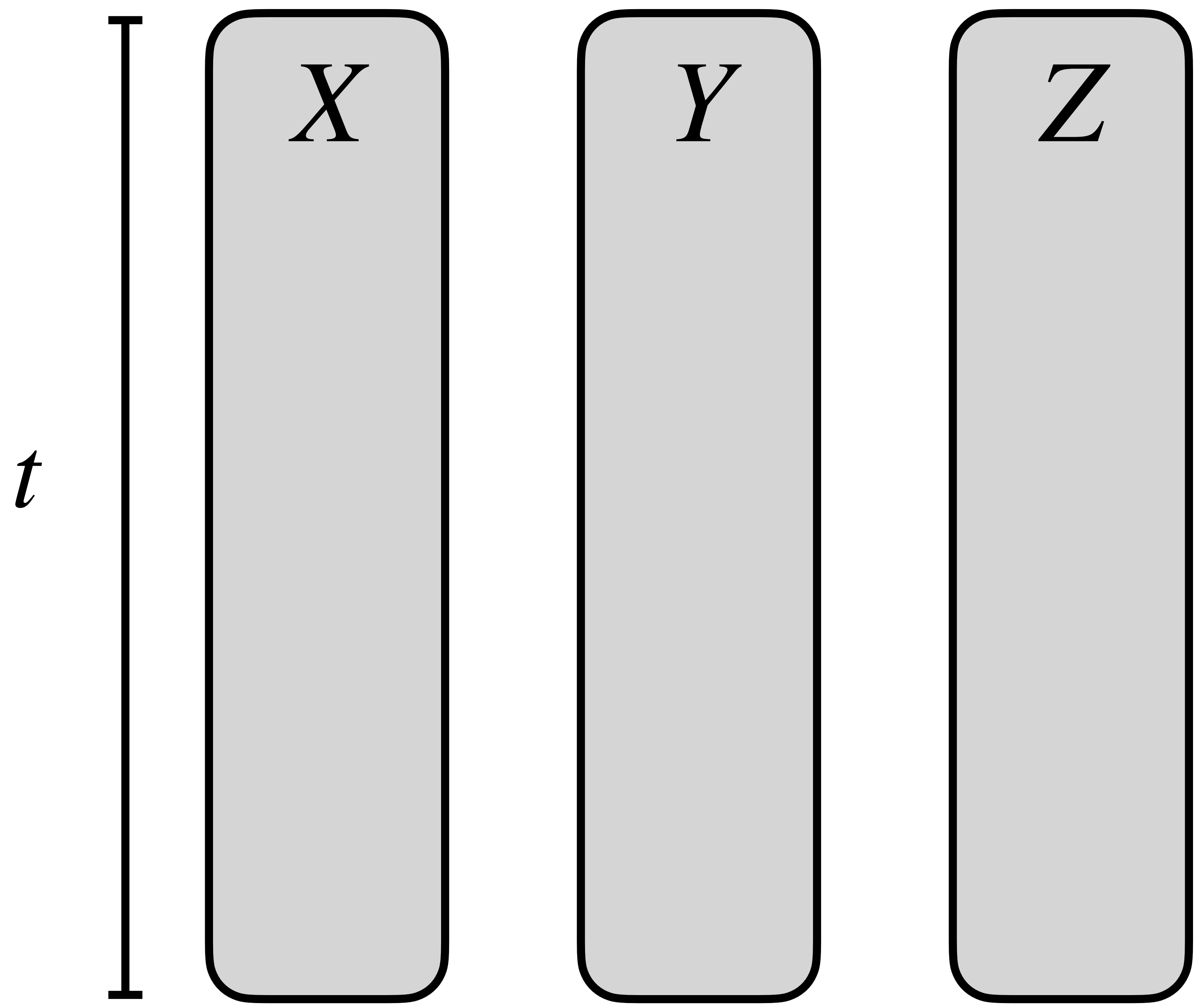
Comparison Based Coloring

- "Shift" the IDs in G' such that
  - i' gets ID $2j + 1$ and
  - j' gets ID $2i + 1$.
- Is it always possible?

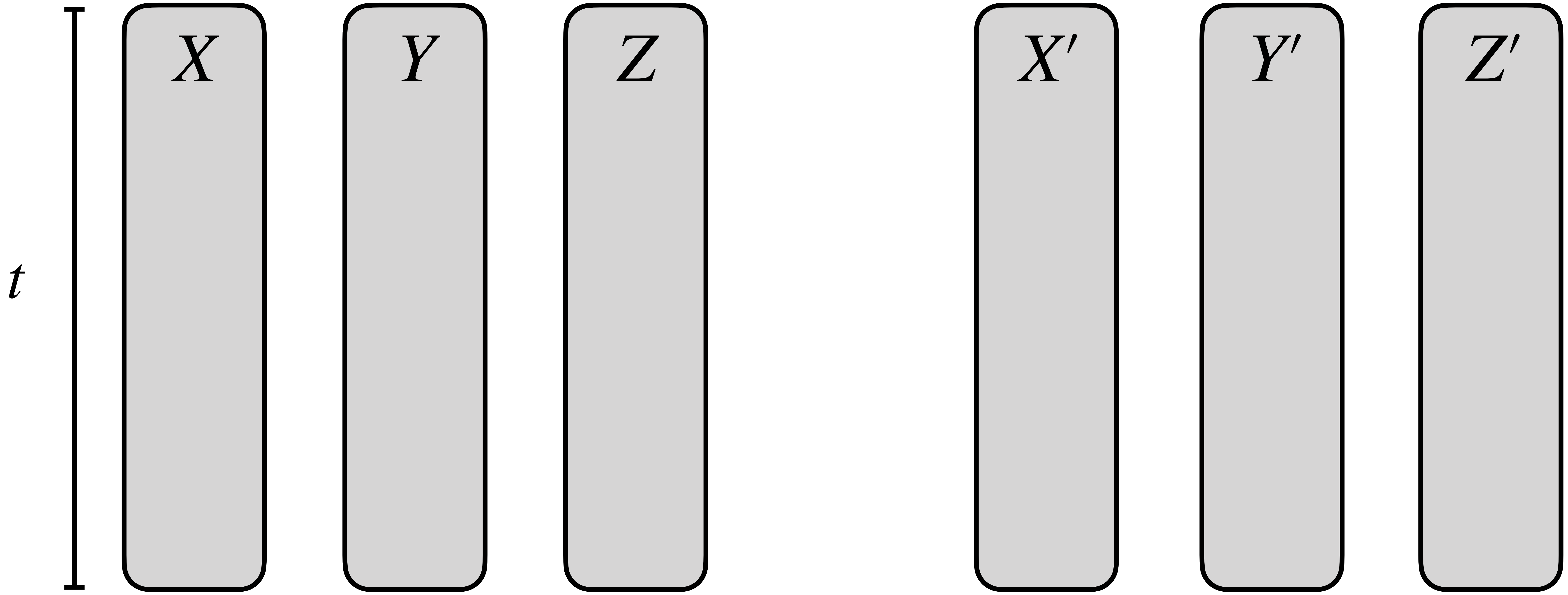


ID Assignments

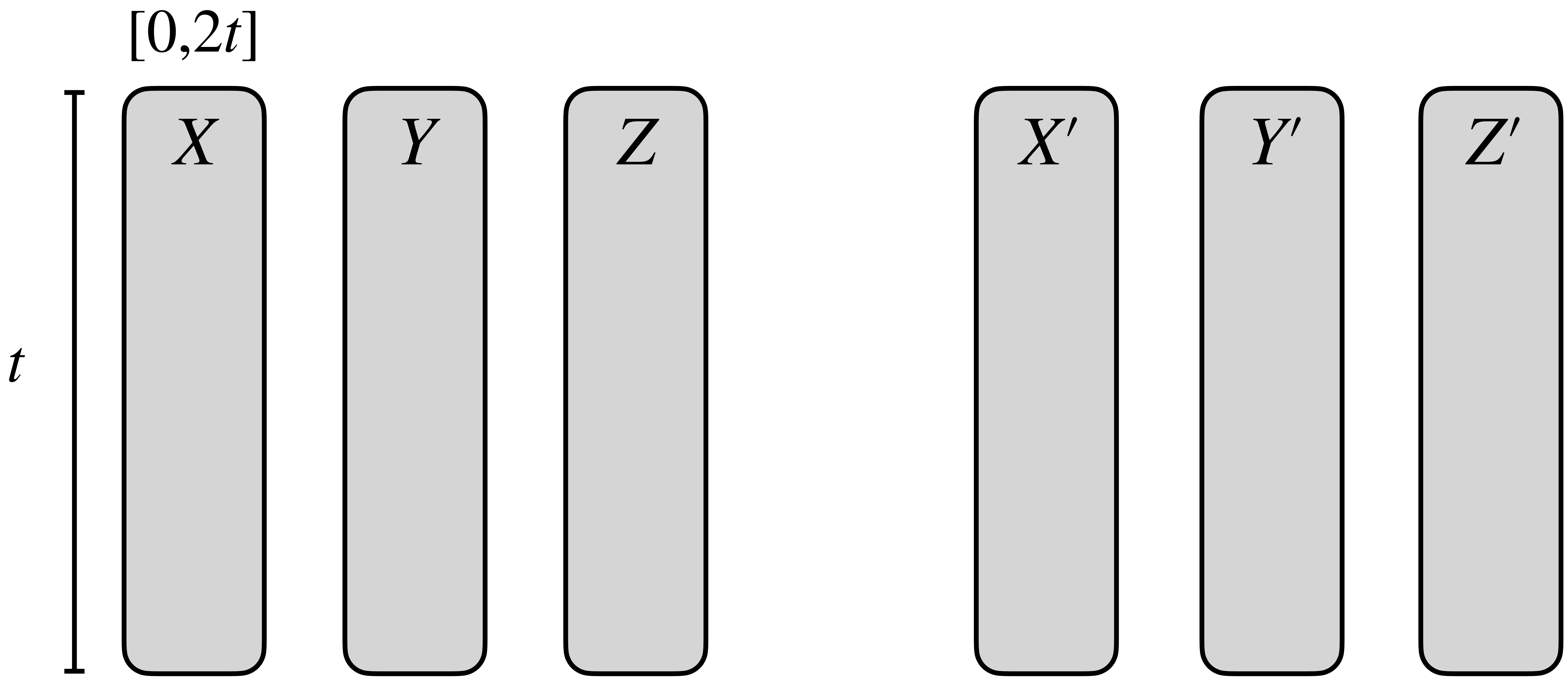
ID Assignments



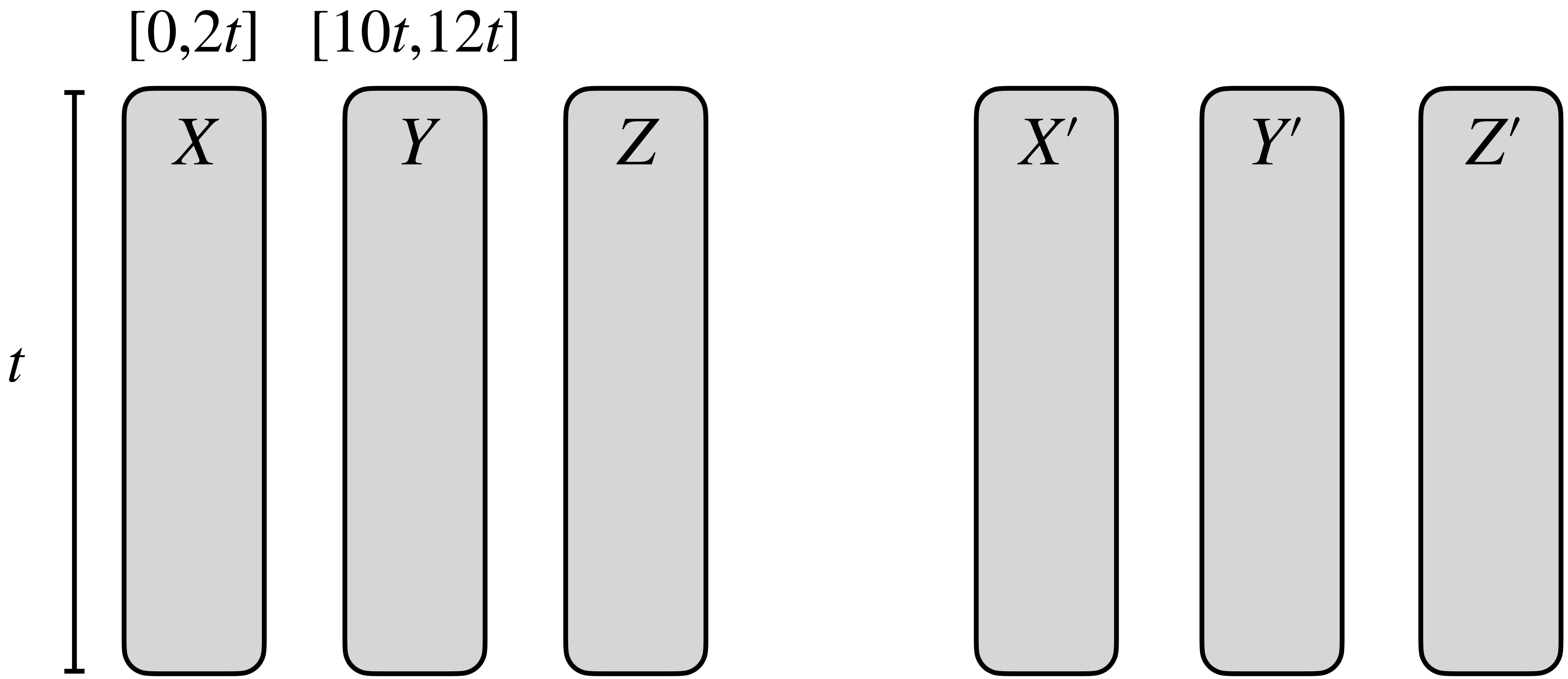
ID Assignments



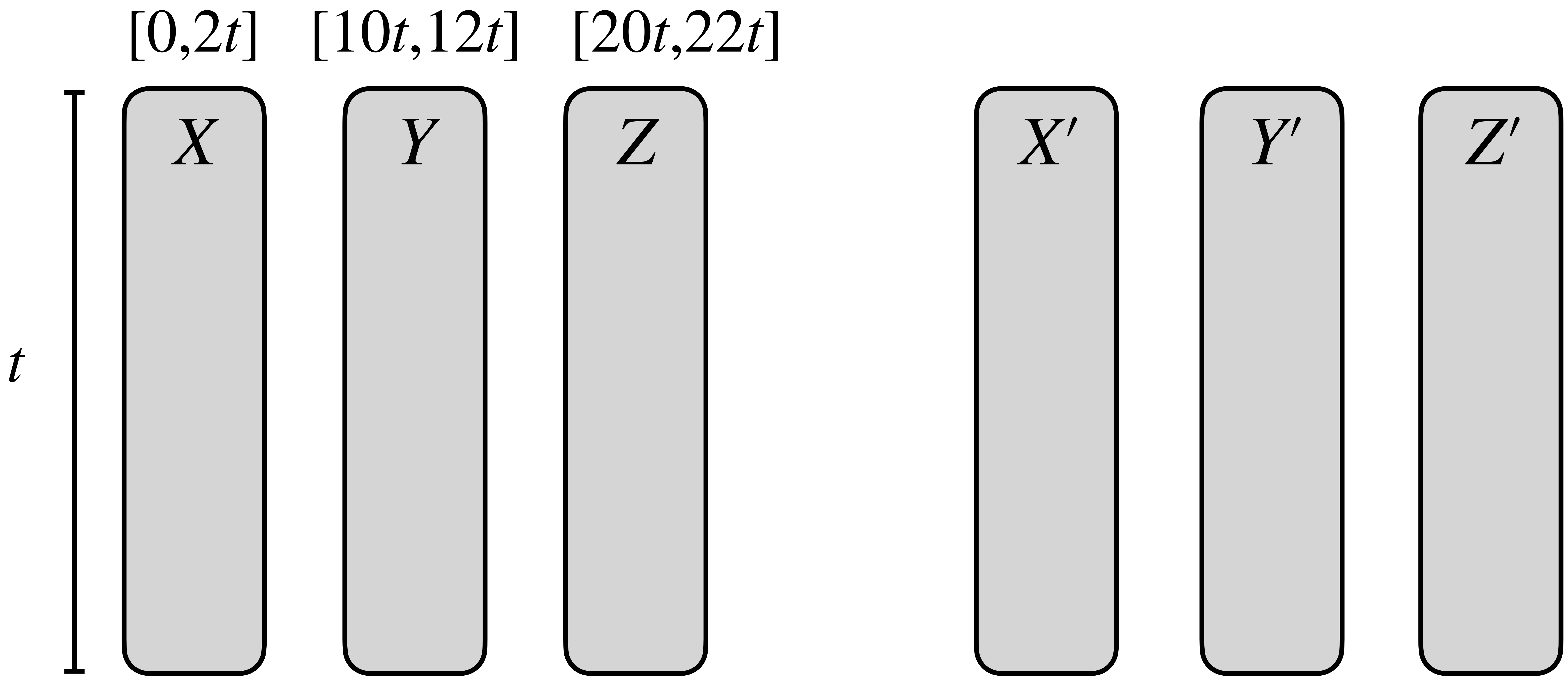
ID Assignments



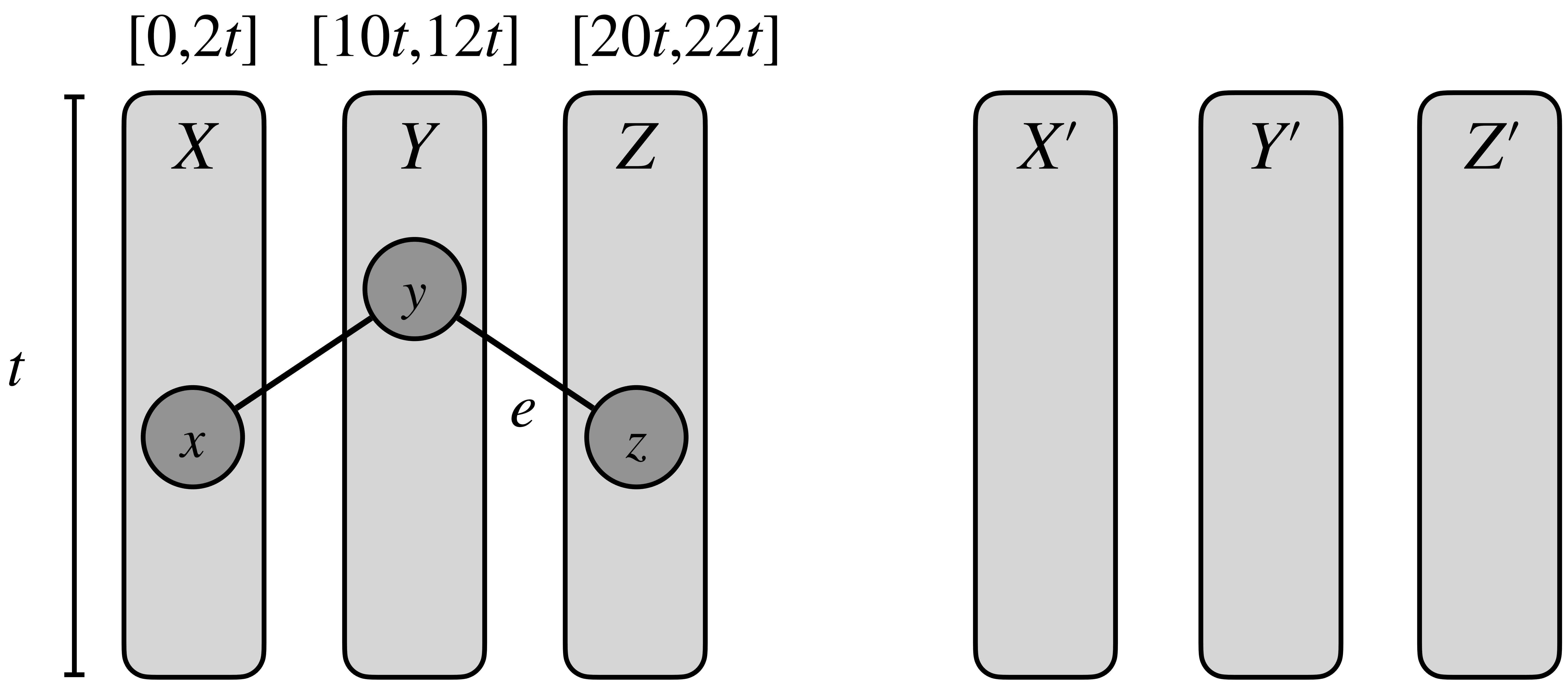
ID Assignments



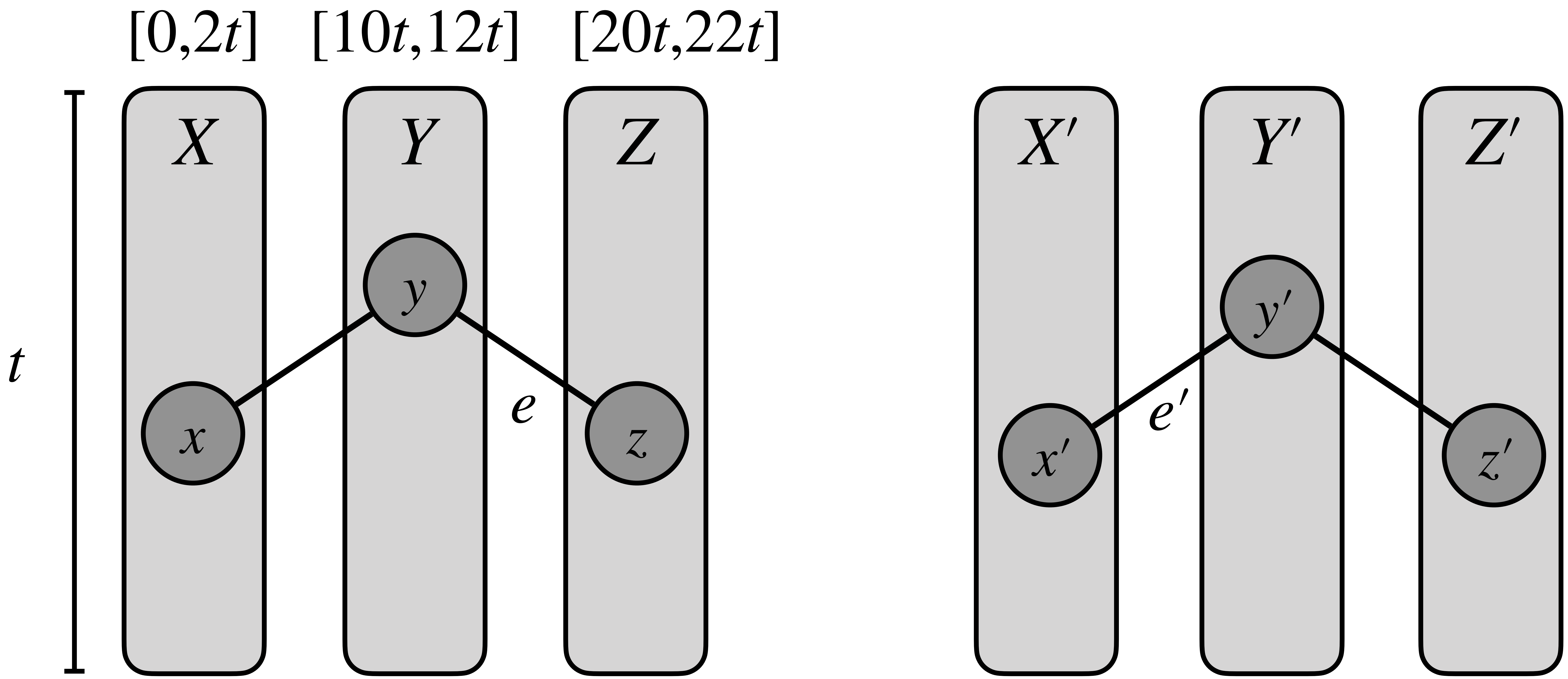
ID Assignments



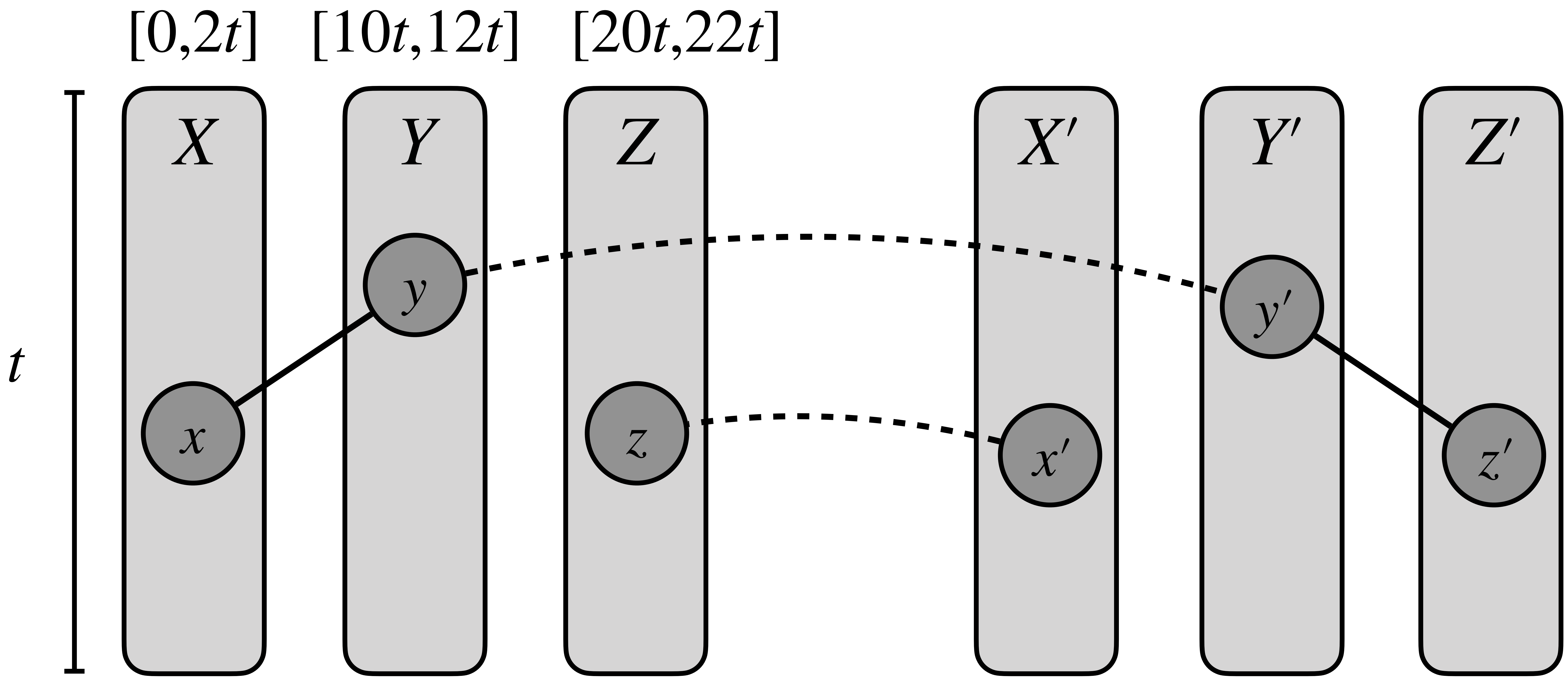
ID Assignments



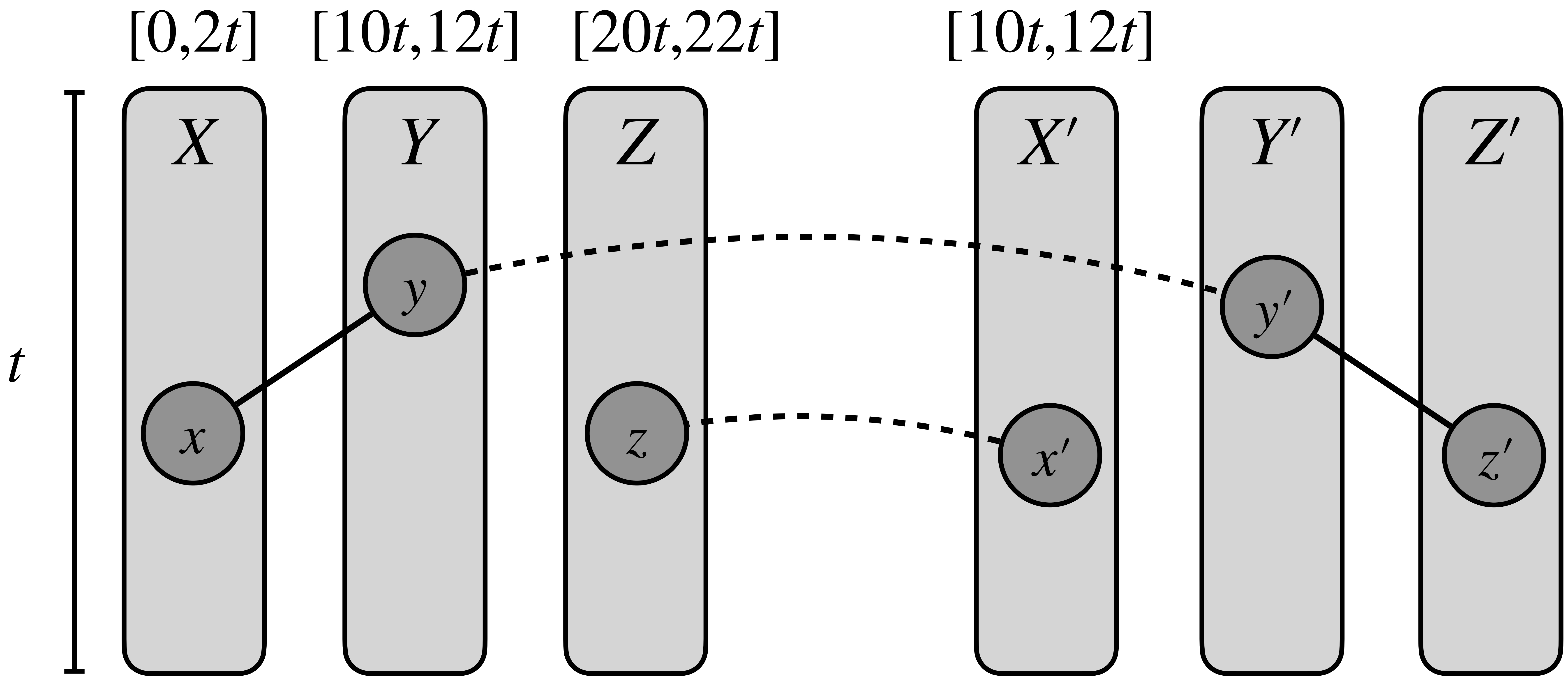
ID Assignments



ID Assignments

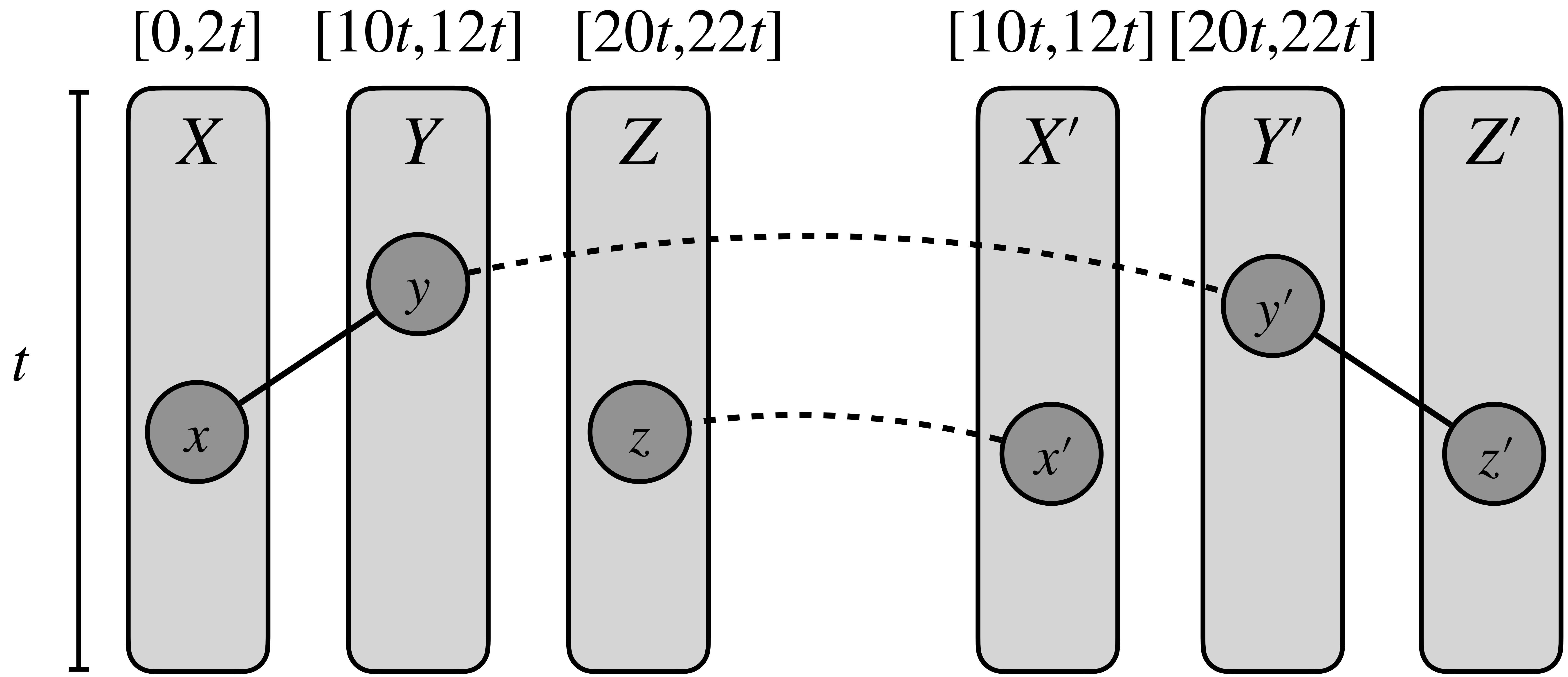


ID Assignments



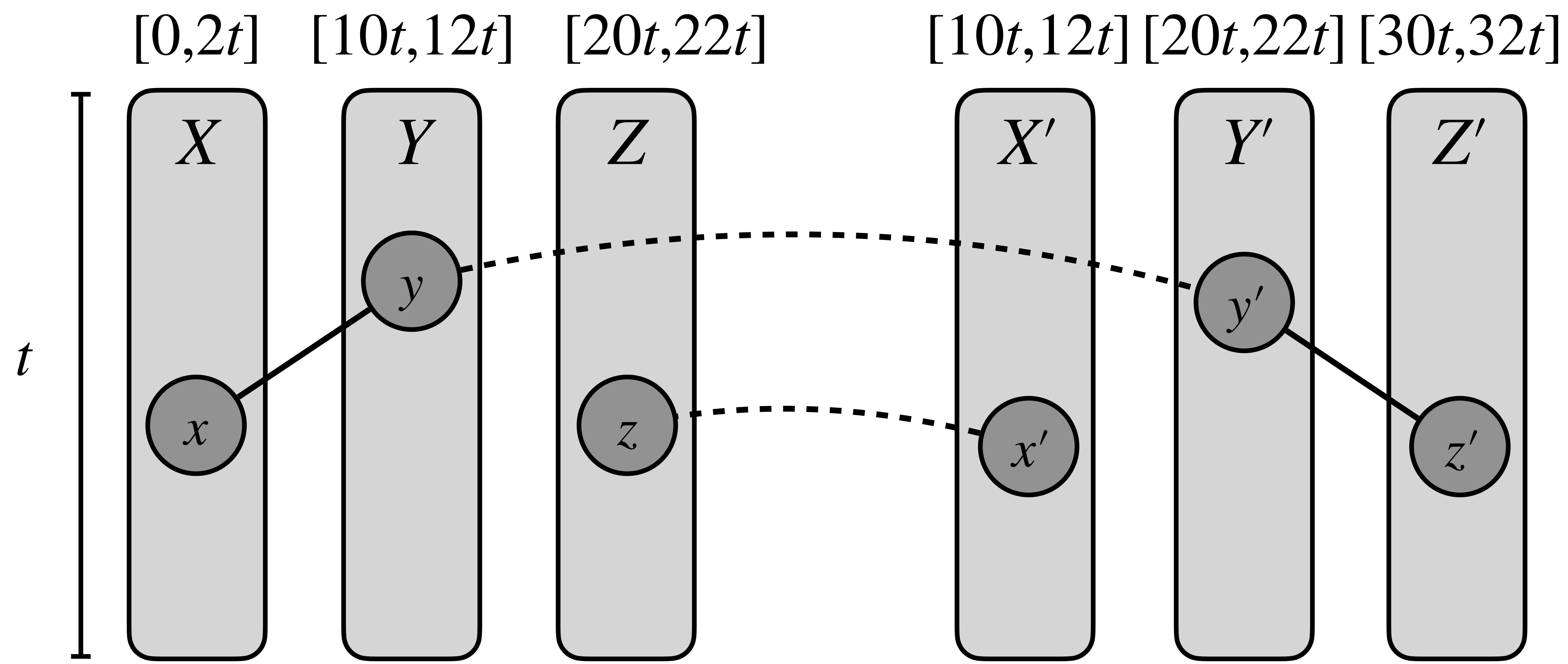
- Shift the ID's in X' to get $ID(x') = ID(y) + 1$

ID Assignments



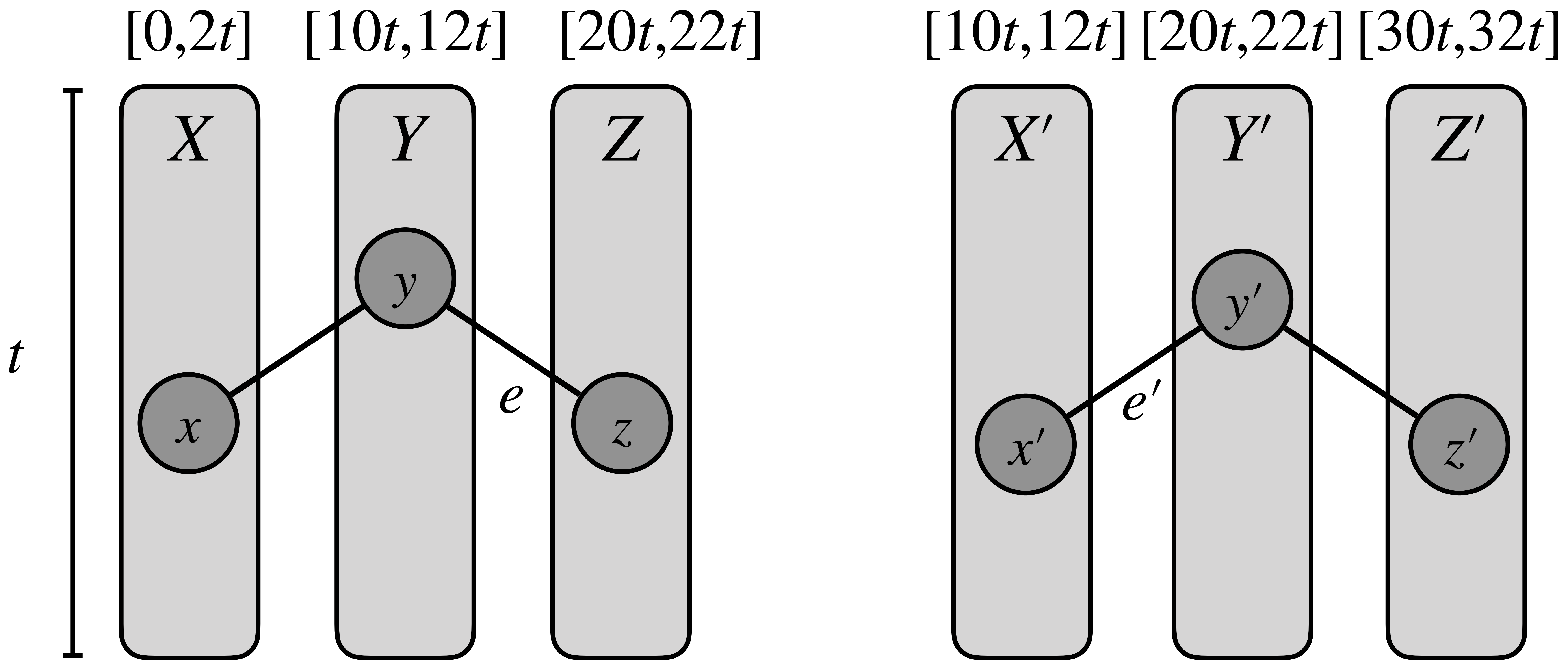
- Shift the ID's in X' to get $ID(x') = ID(y) + 1$
- Shift the ID's in Y' to get $ID(y') = ID(z) + 1$

ID Assignments



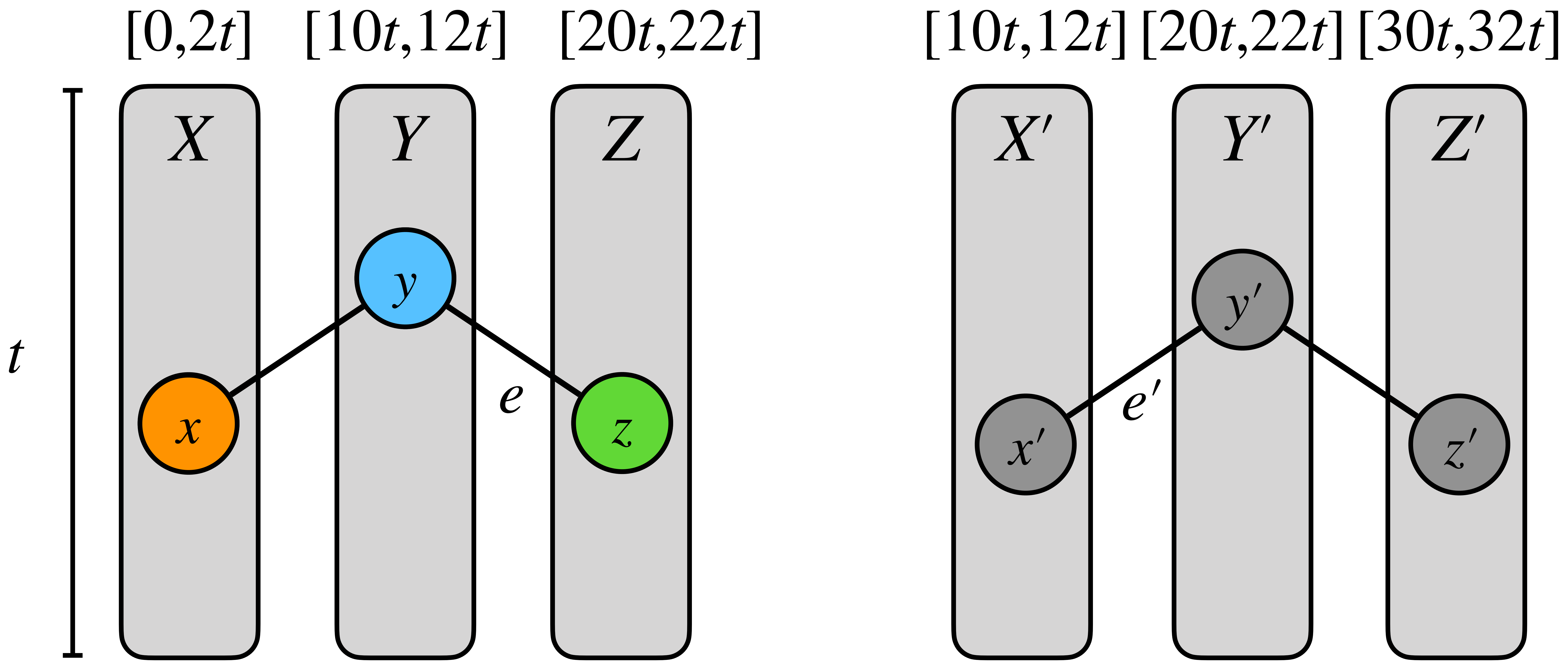
- Shift the ID's in X' to get $ID(x') = ID(y) + 1$
- Shift the ID's in Y' to get $ID(y') = ID(z) + 1$

ID Assignments



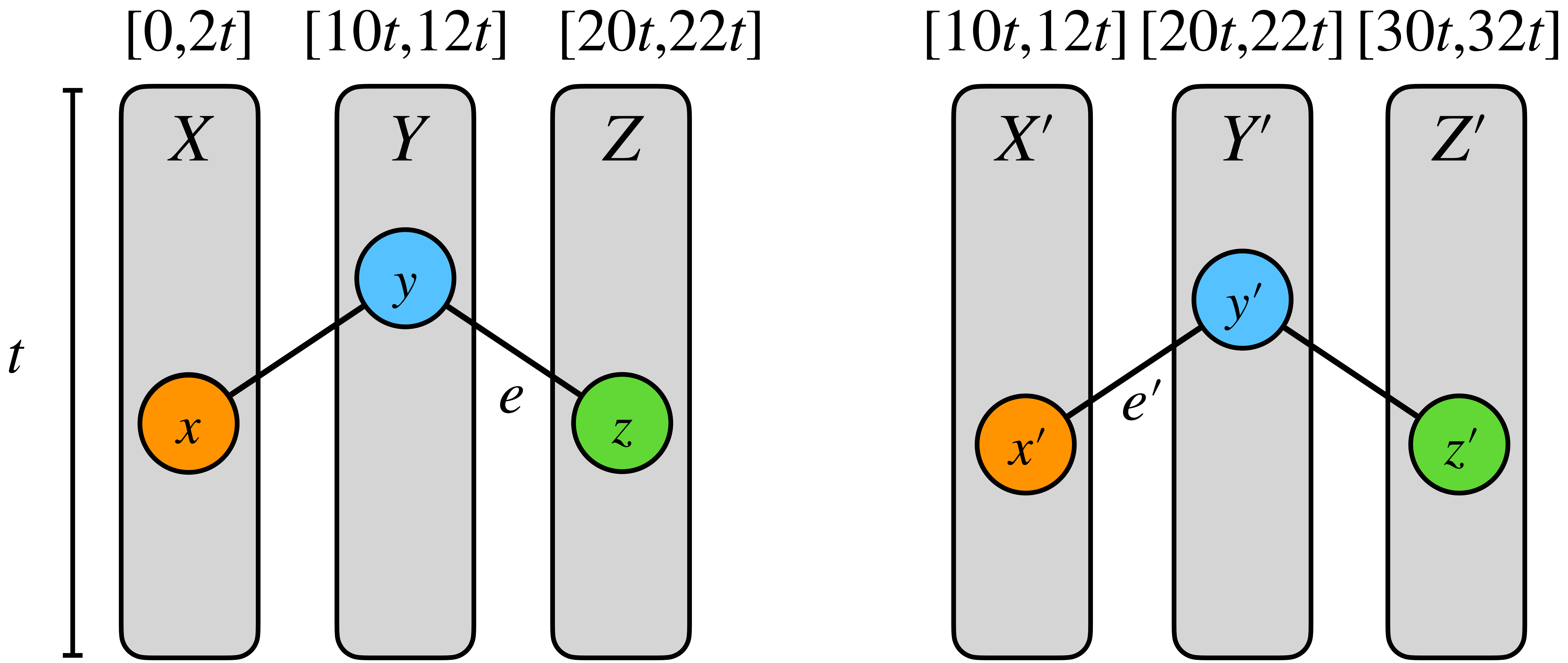
- Shift the ID's in X' to get $ID(x') = ID(y) + 1$
- Shift the ID's in Y' to get $ID(y') = ID(z) + 1$

ID Assignments



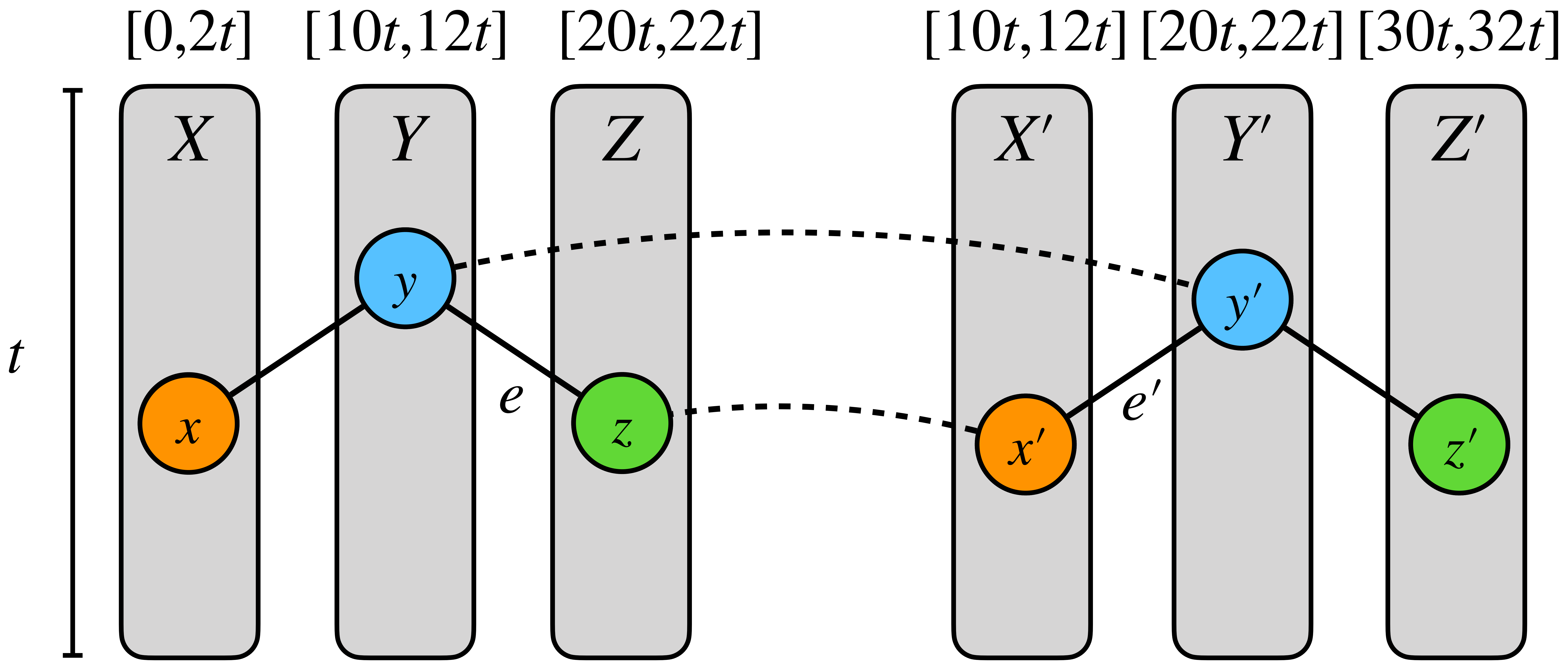
- Shift the ID's in X' to get $ID(x') = ID(y) + 1$
- Shift the ID's in Y' to get $ID(y') = ID(z) + 1$

ID Assignments



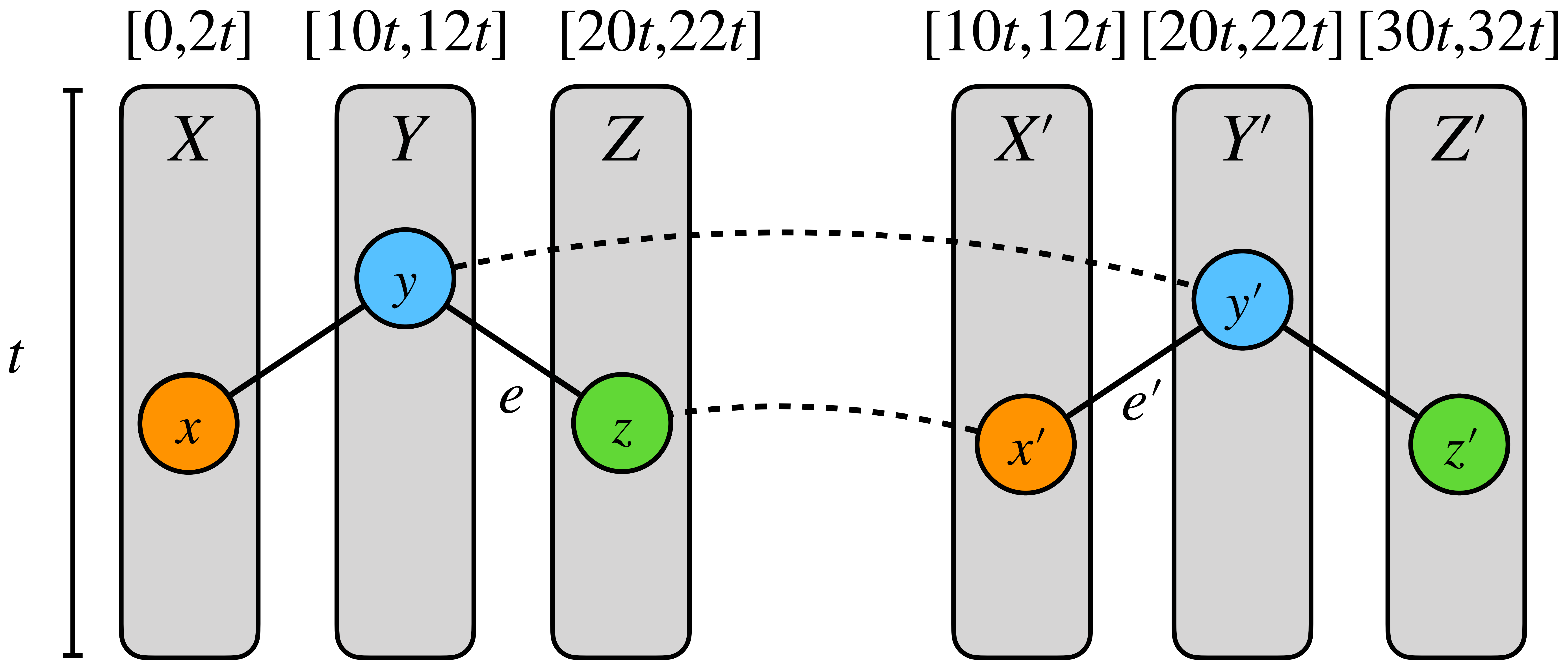
- Shift the ID's in X' to get $ID(x') = ID(y) + 1$
- Shift the ID's in Y' to get $ID(y') = ID(z) + 1$

ID Assignments

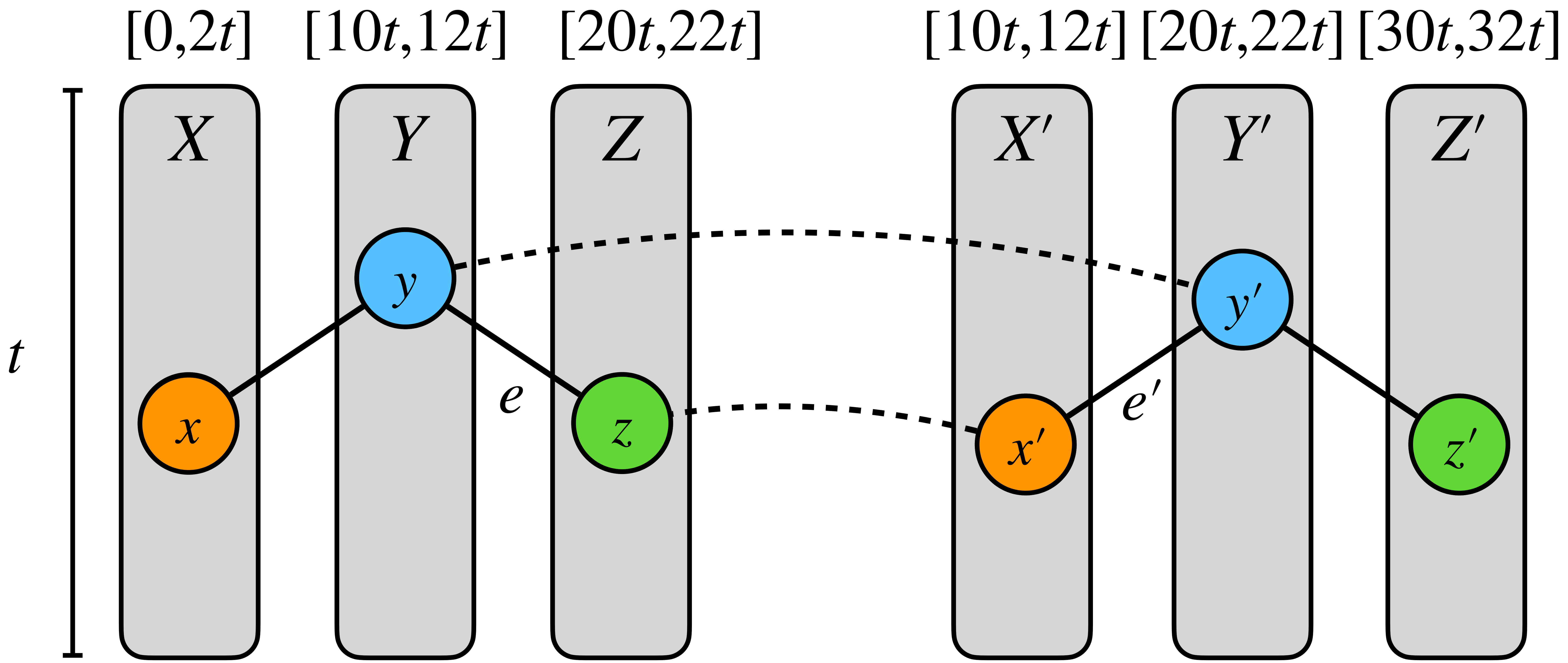


- Shift the ID's in X' to get $ID(x') = ID(y) + 1$
- Shift the ID's in Y' to get $ID(y') = ID(z) + 1$

ID Assignments

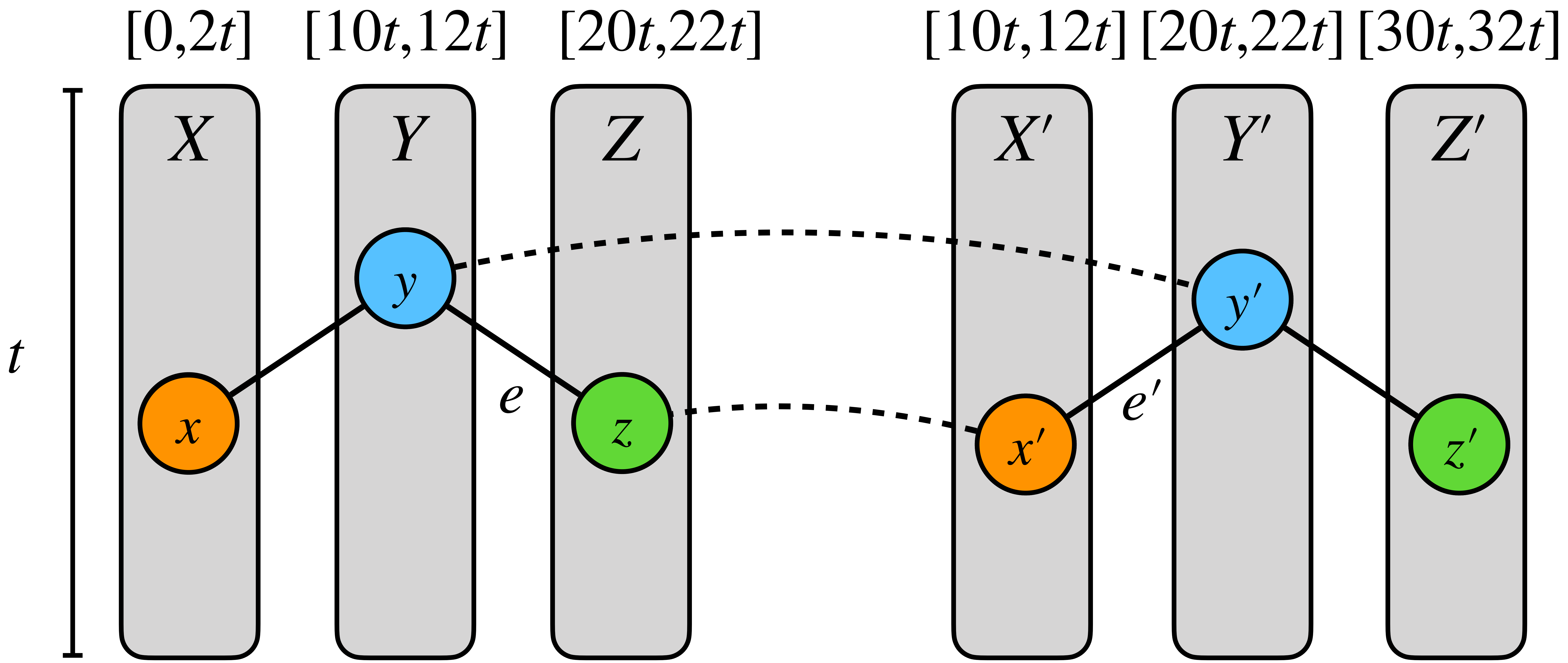


ID Assignments



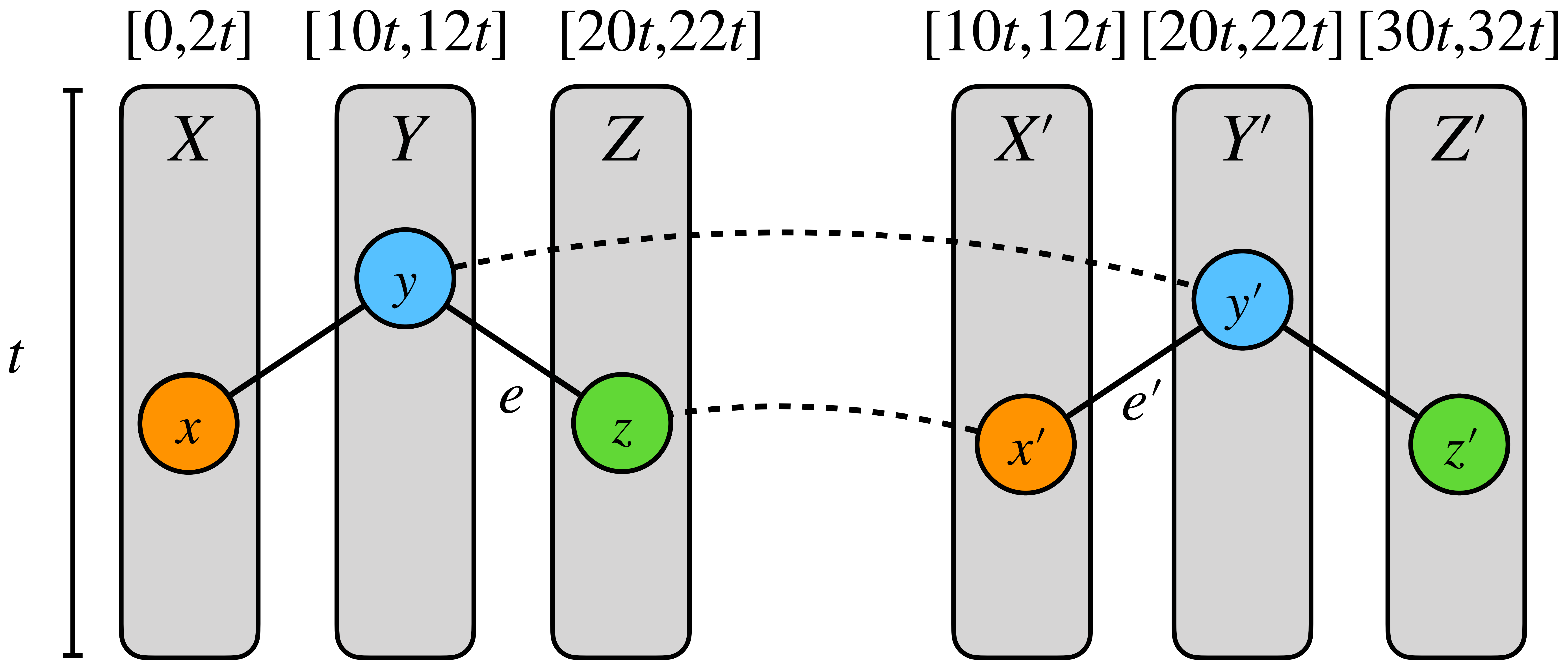
- If there is a $o(m)$ message algorithm, we can find many such edges e, e'.

ID Assignments



- If there is a $o(m)$ message algorithm, we can find many such edges e, e'.
- $\Omega(m) = \Omega(n^2)$ message lower bound for $(\Delta + 1)$-coloring.

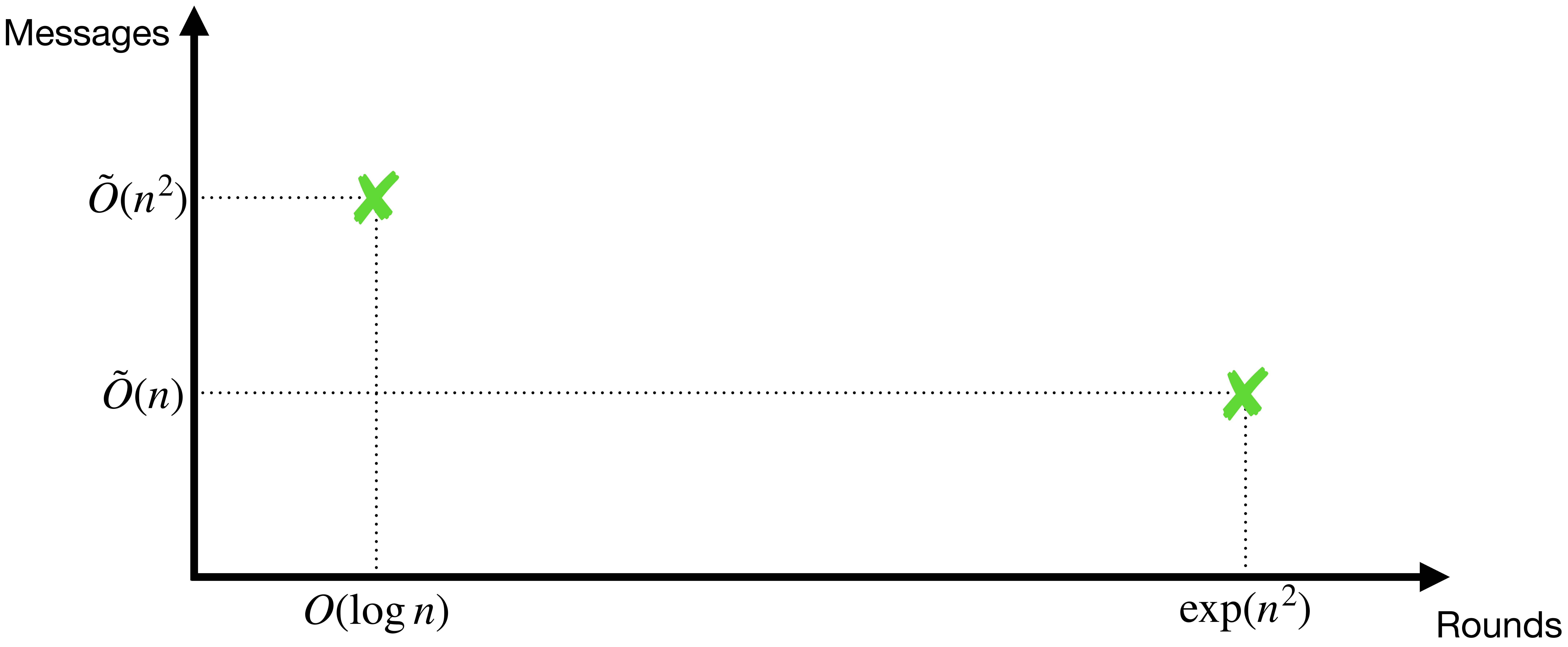
ID Assignments



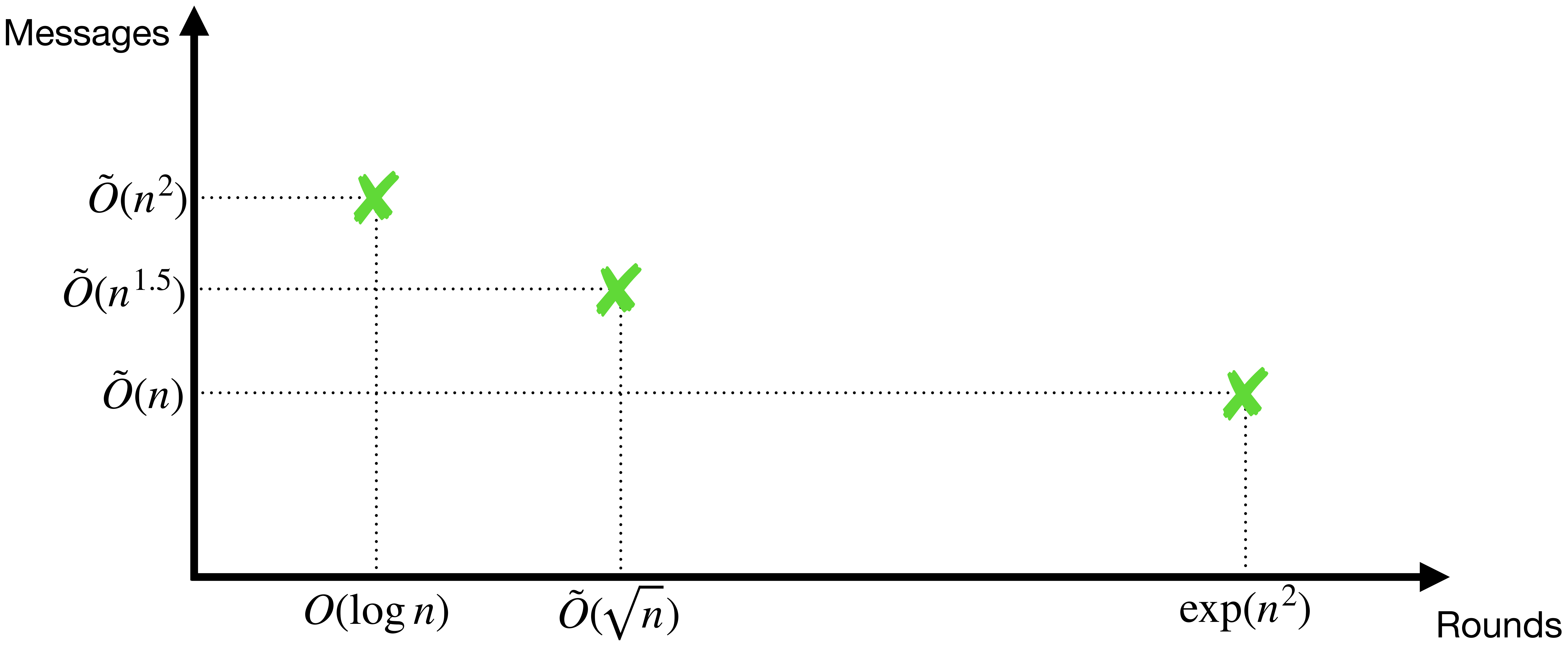
KT-1 Coloring Algorithms

KT-1 Coloring Algorithms

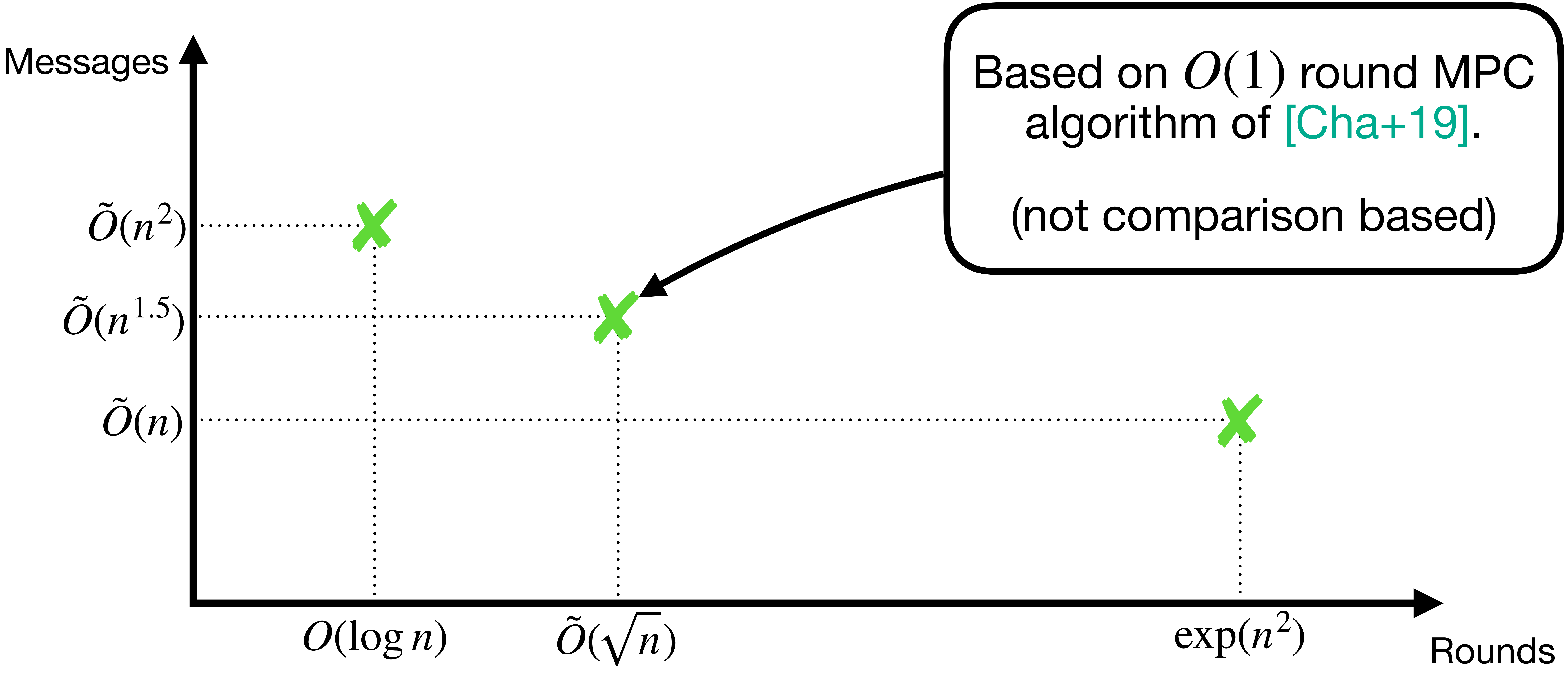
KT-1 Coloring Algorithms



KT-1 Coloring Algorithms

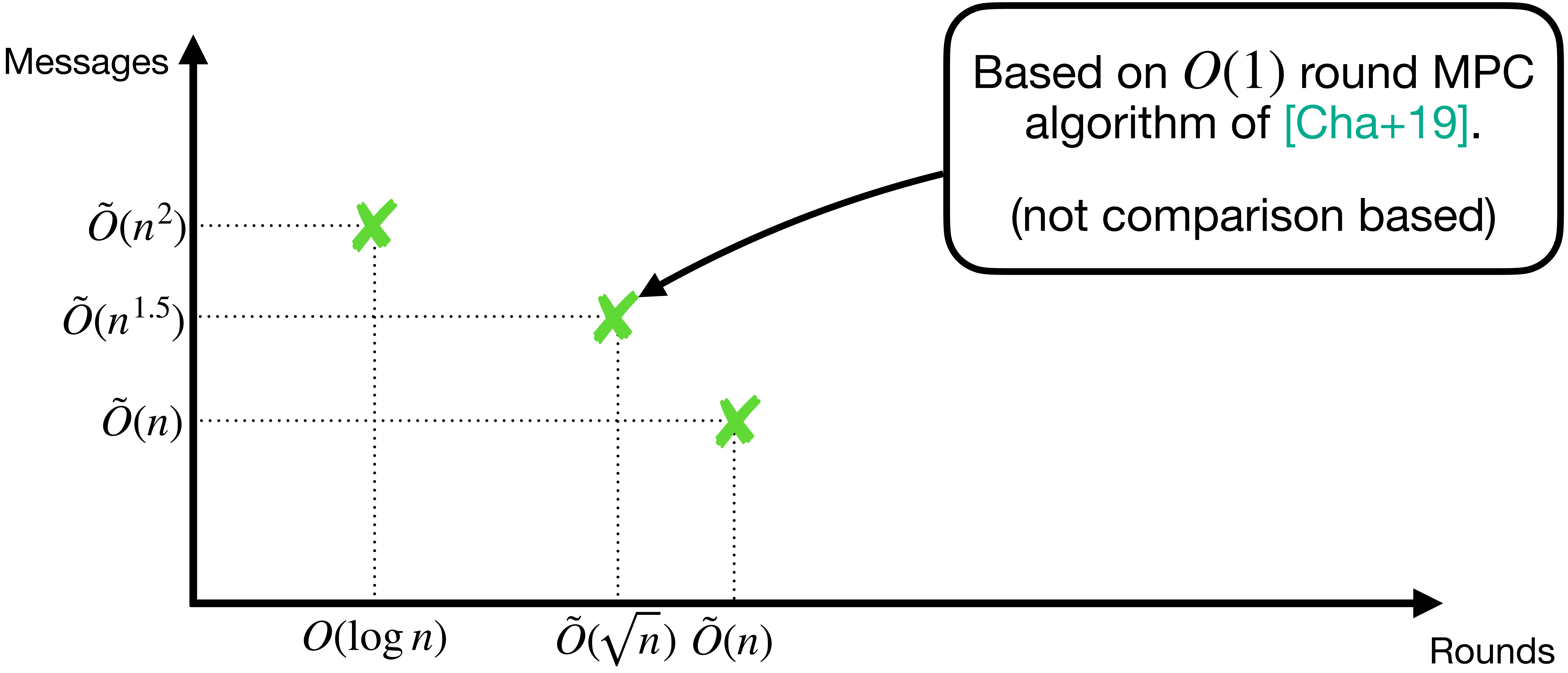


KT-1 Coloring Algorithms



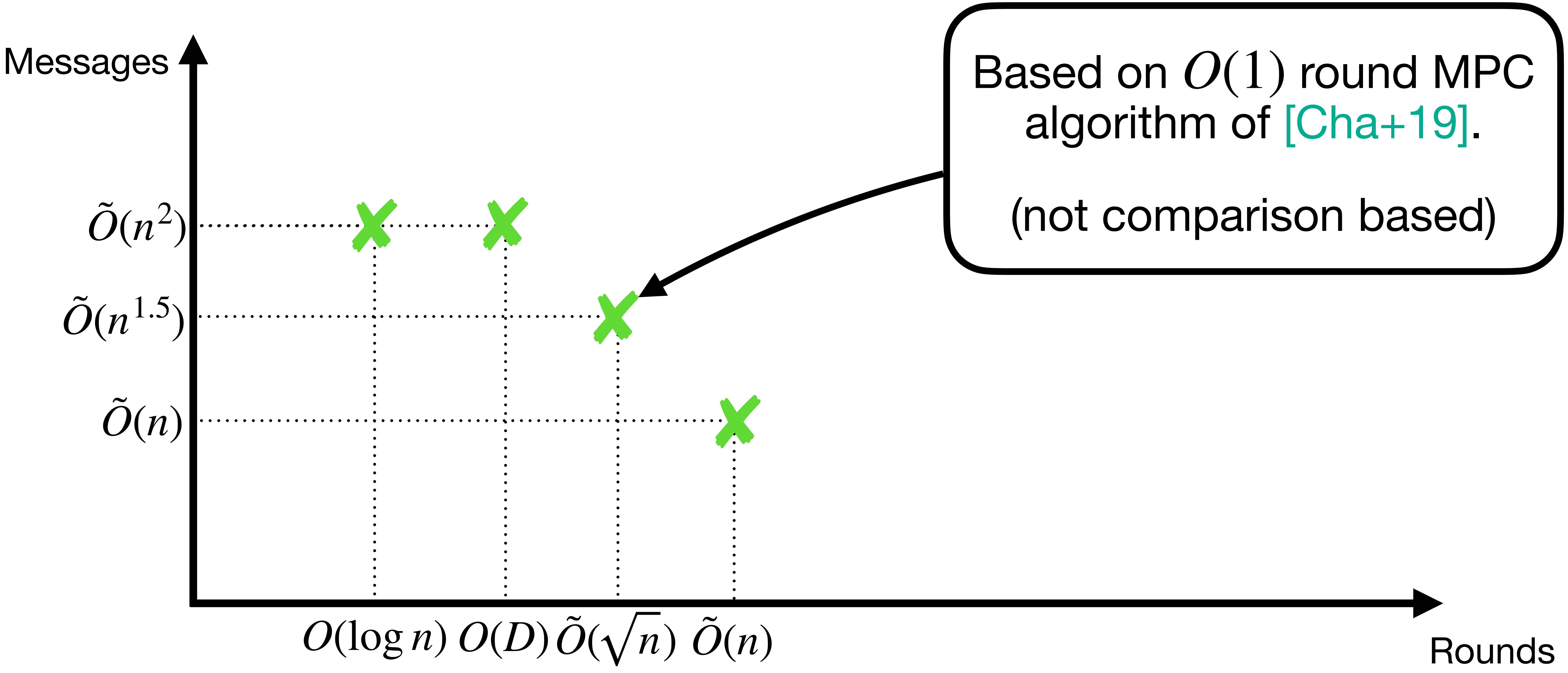
[Cha+19] Chang, Fischer, Ghaffari, Uitto, and Zheng. PODC 2019

KT-1 Coloring Algorithms



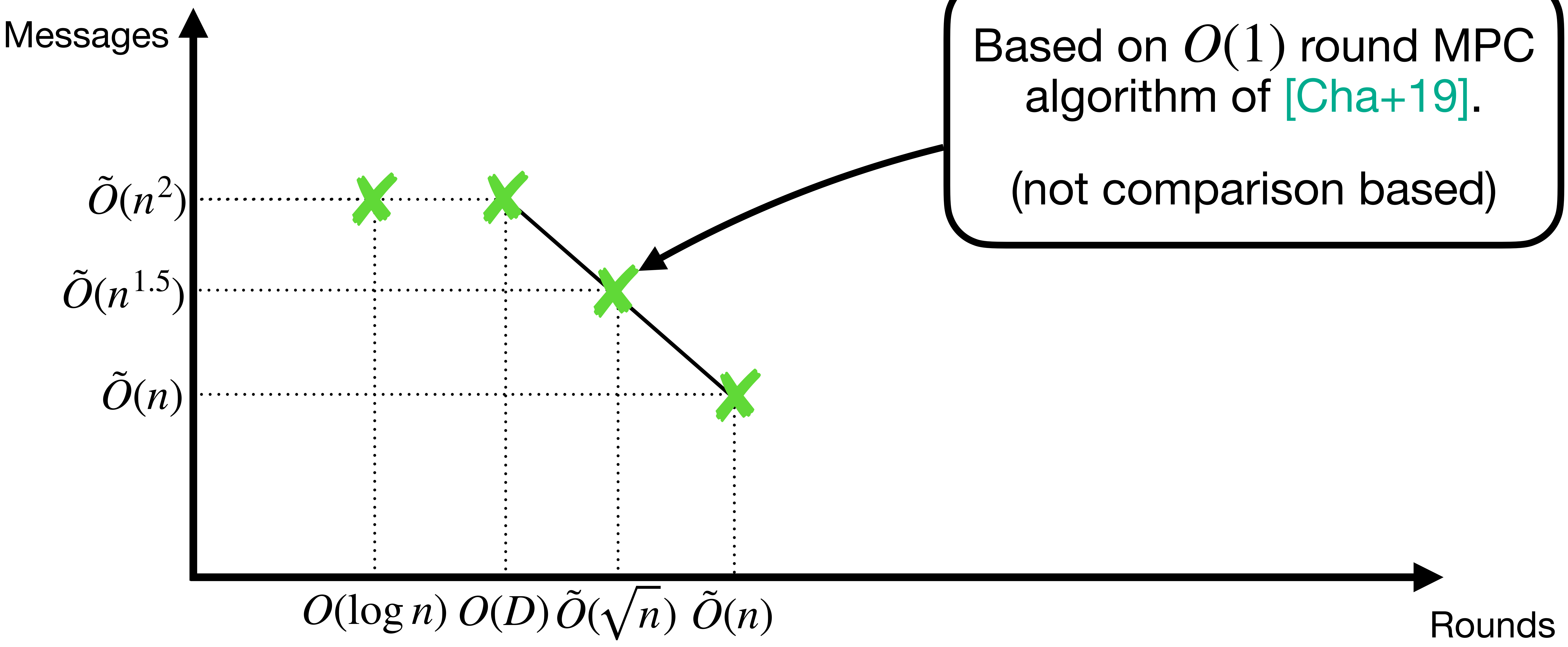
[Cha+19] Chang, Fischer, Ghaffari, Uitto, and Zheng. PODC 2019

KT-1 Coloring Algorithms



[Cha+19] Chang, Fischer, Ghaffari, Uitto, and Zheng. PODC 2019

KT-1 Coloring Algorithms



Based on $O(1)$ round MPC algorithm of [Cha+19].

(not comparison based)

[Cha+19] *Chang, Fischer, Ghaffari, Uitto, and Zheng. PODC 2019*

Open Questions

- Can we get singularly optimal algorithms for local symmetry breaking problems like MIS, $(\Delta + 1)$-coloring, Maximal Matching?
    - Such algorithms are known for problems like leader election.
- Can we rule out singularly optimal algorithms in KT-1 CONGEST?
- Can we design an algorithm for MIS in KT-1 CONGEST that uses poly(n) rounds and $o(m)$ messages?

To $\Omega(m)$ and Beyond

- For comparison based KT-1 algorithms, $\Omega(m)$ message lower bound for:
  - $(\Delta + 1)$-coloring.
  - Maximal Independent Set.

[PPP+21] Pai, Pandurangan, Pemmaraju, Robinson. PODC 2021

- For poly(n) round KT-1 algorithms, $\Omega(m \cdot D)$ message lower bound for:
  - Minimum Vertex Cover.
  - Maximum Independent Set.
  - Minimum Dominating Set.

[DPP+24] Dufoulon, Pai, Pandurangan, Pemmaraju, Robinson. ITCS 2024

To $\Omega(m)$ and Beyond

- For comparison based KT-1 algorithms, $\Omega(m)$ message lower bound for:
  - $(\Delta + 1)$-coloring.
  - Maximal Independent Set.

[PPP+21] Pai, Pandurangan, Pemmaraju, Robinson. PODC 2021

- For poly(n) round KT-1 algorithms, $\Omega(m \cdot D)$ message lower bound for:
  - Minimum Vertex Cover.
  - Maximum Independent Set.
  - Minimum Dominating Set.

[DPP+24] Dufoulon, Pai, Pandurangan, Pemmaraju, Robinson. ITCS 2024

Lower Bound for MVC

Lower Bound for MVC

- Based on the 2-party communication complexity reduction framework of [CKP17].

[CKP17] Censor-Hillel, Khoury, Paz. DISC 2017

Lower Bound for MVC

- Based on the 2-party communication complexity reduction framework of [CKP17].

- Typically reduce from Set Disjointness.

[CKP17] Censor-Hillel, Khoury, Paz. DISC 2017

Lower Bound for MVC

- Based on the 2-party communication complexity reduction framework of [CKP17].
- Typically reduce from Set Disjointness.

[CKP17] *Censor-Hillel, Khoury, Paz. DISC 2017*

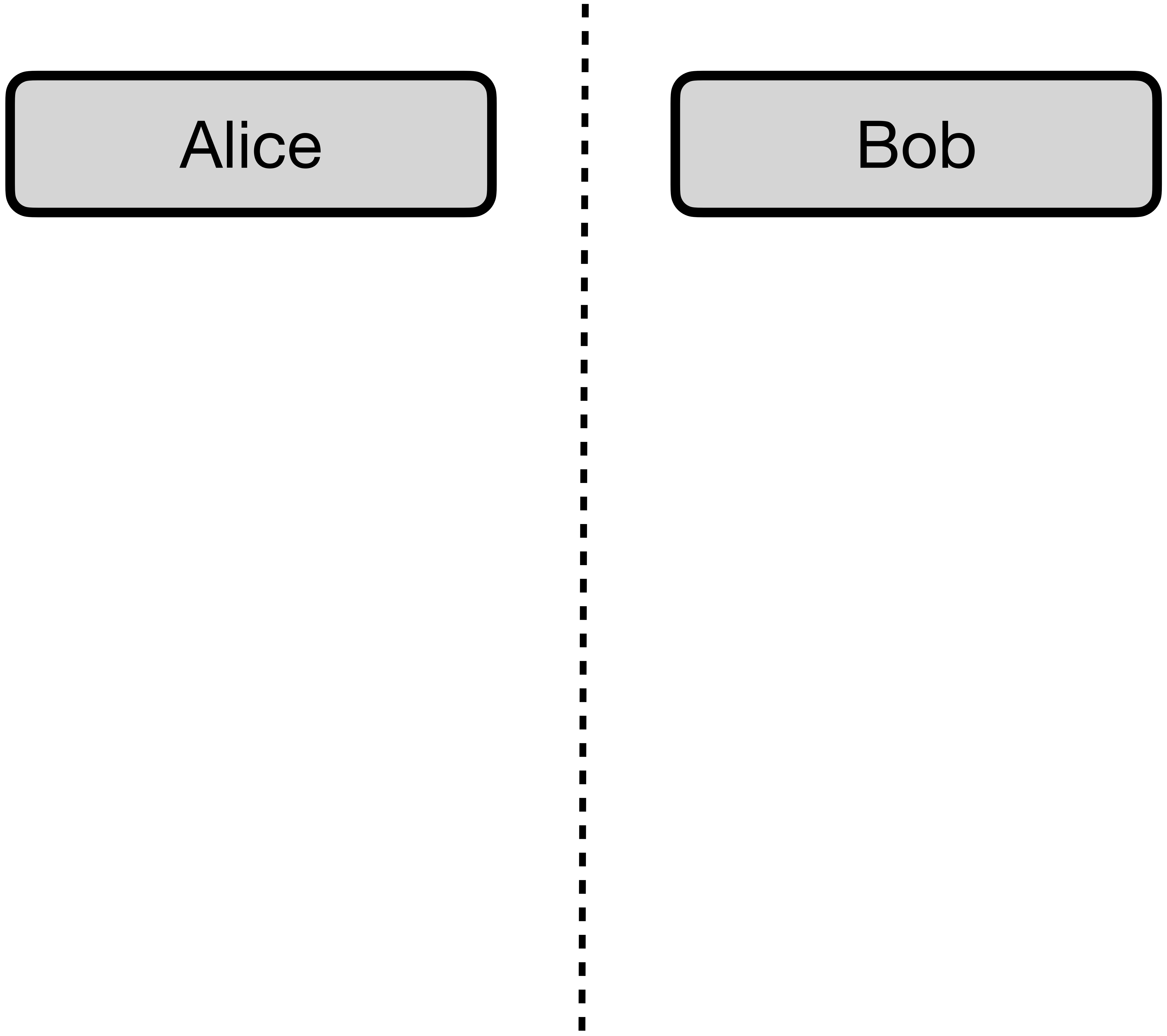
Lower Bound for MVC

- Based on the 2-party communication complexity reduction framework of [CKP17].
- Typically reduce from Set Disjointness.

[CKP17] Censor-Hillel, Khoury, Paz. DISC 2017

Lower Bound for MVC

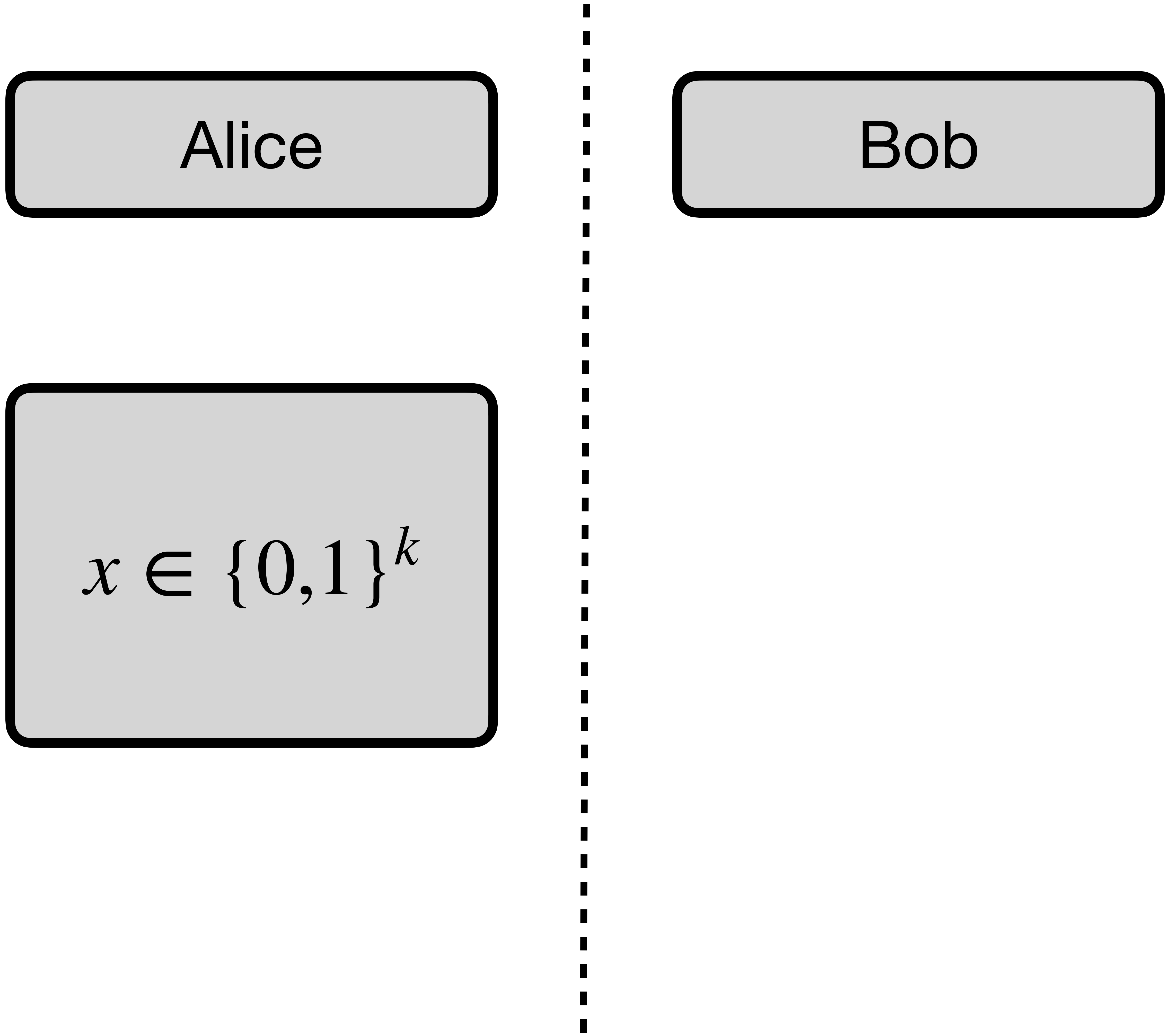
- Based on the 2-party communication complexity reduction framework of [CKP17].
- Typically reduce from Set Disjointness.



[CKP17] *Censor-Hillel, Khoury, Paz. DISC 2017*

Lower Bound for MVC

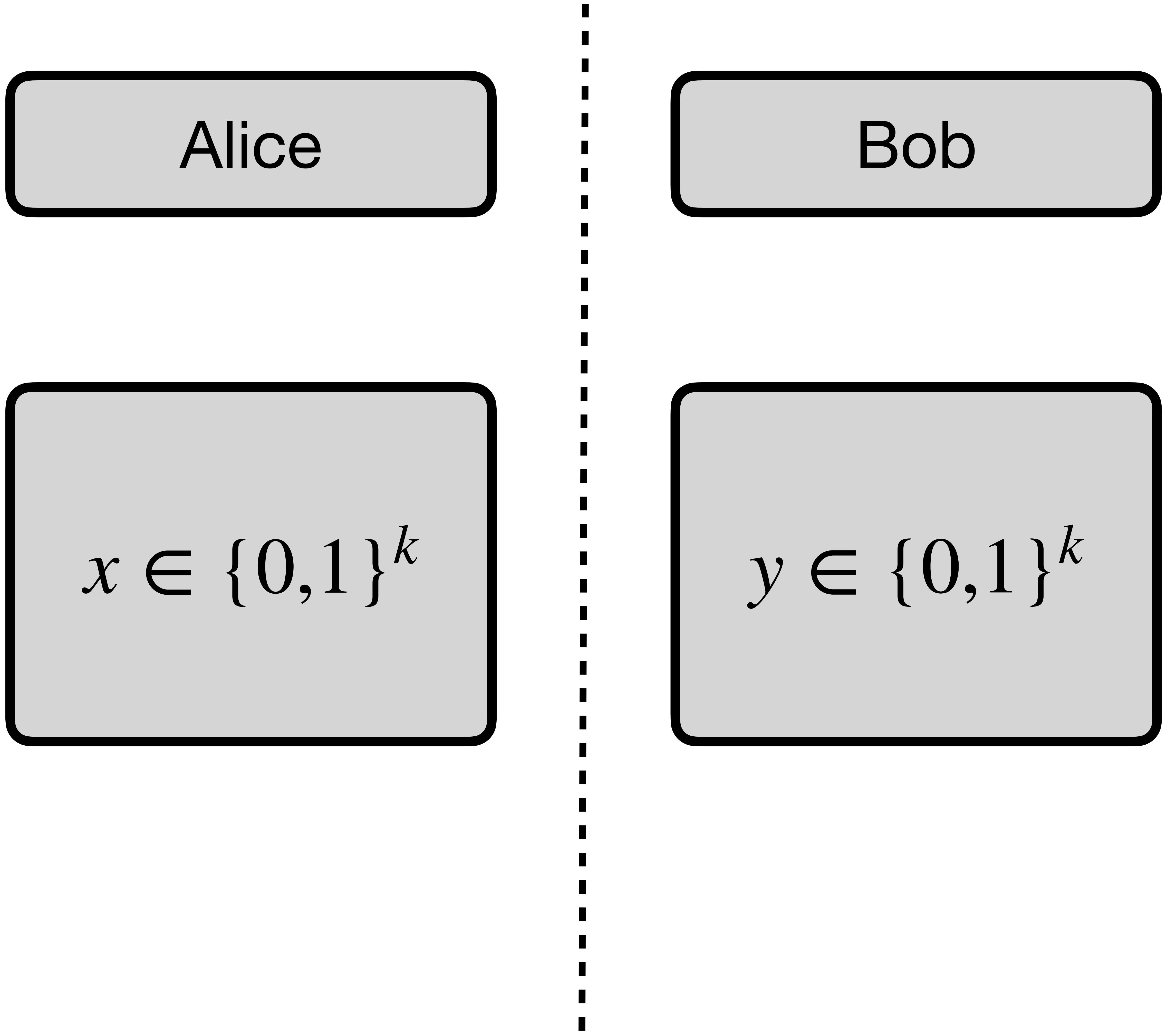
- Based on the 2-party communication complexity reduction framework of [CKP17].
- Typically reduce from Set Disjointness.



[CKP17] *Censor-Hillel, Khoury, Paz. DISC 2017*

Lower Bound for MVC

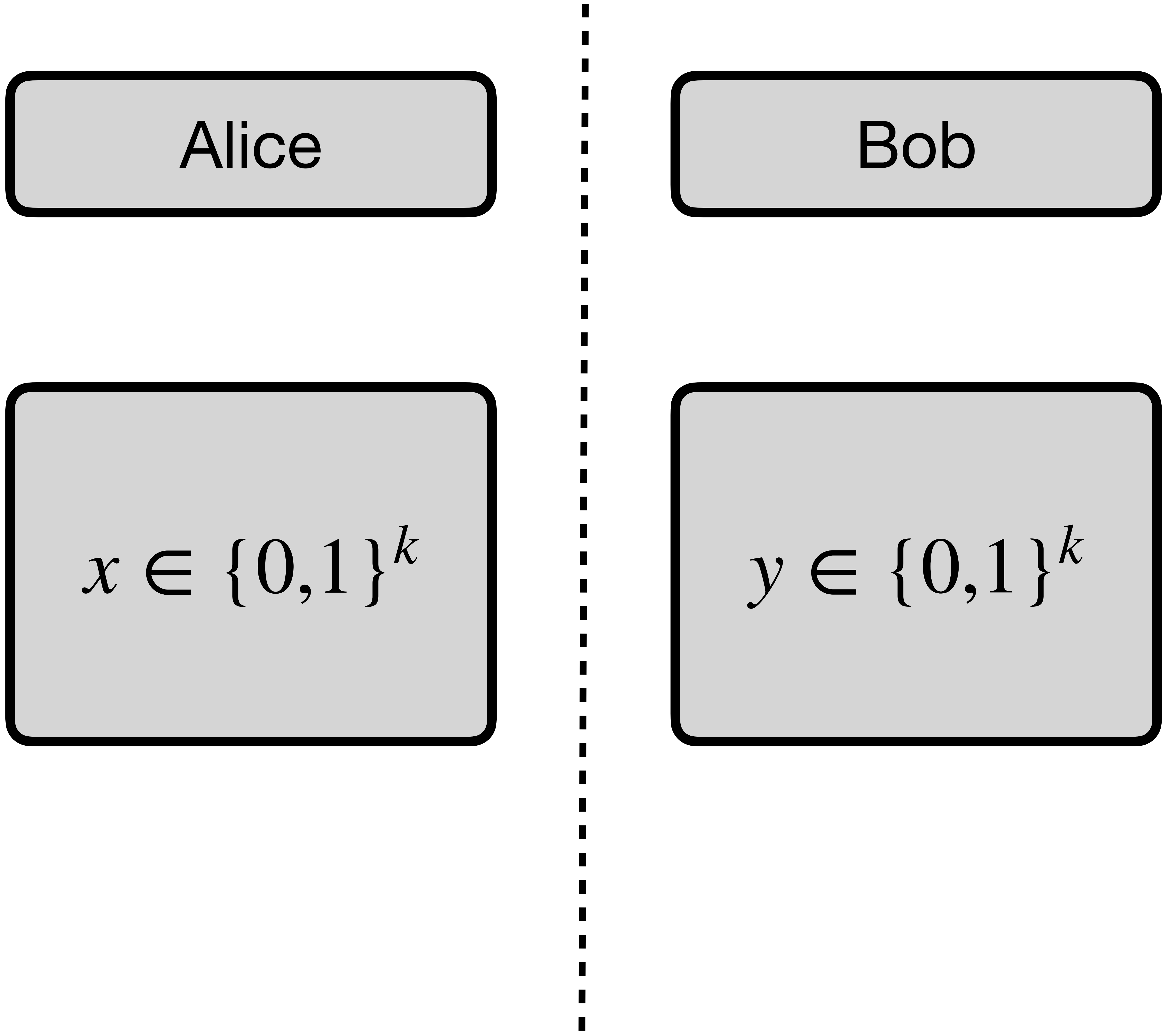
- Based on the 2-party communication complexity reduction framework of [CKP17].
- Typically reduce from Set Disjointness.



[CKP17] Censor-Hillel, Khoury, Paz. DISC 2017

Lower Bound for MVC

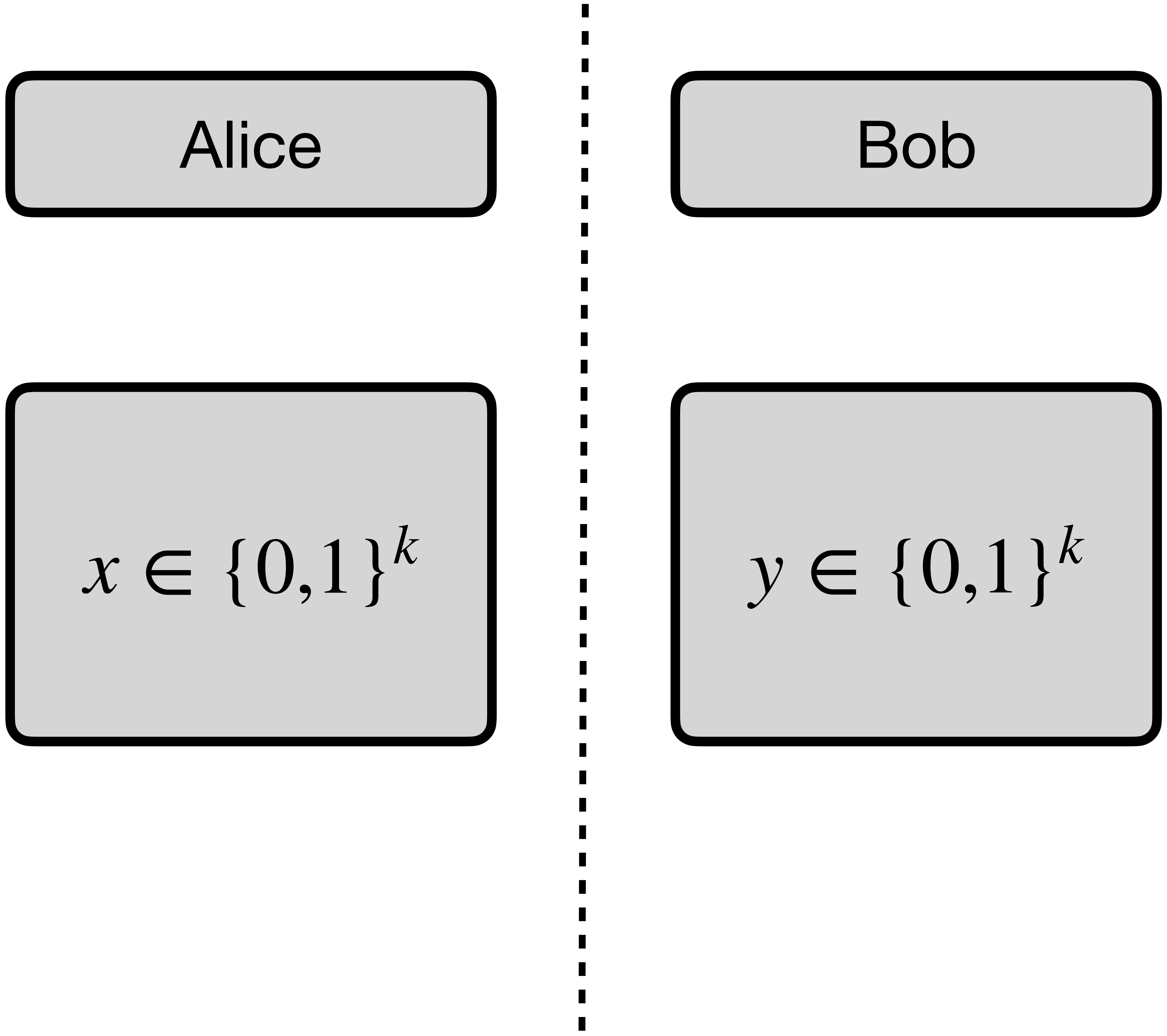
- Based on the 2-party communication complexity reduction framework of [CKP17].
- Typically reduce from Set Disjointness.
    - SD(x, y) = False if $x_i = y_i = 1$.



[CKP17] Censor-Hillel, Khoury, Paz. DISC 2017

Lower Bound for MVC

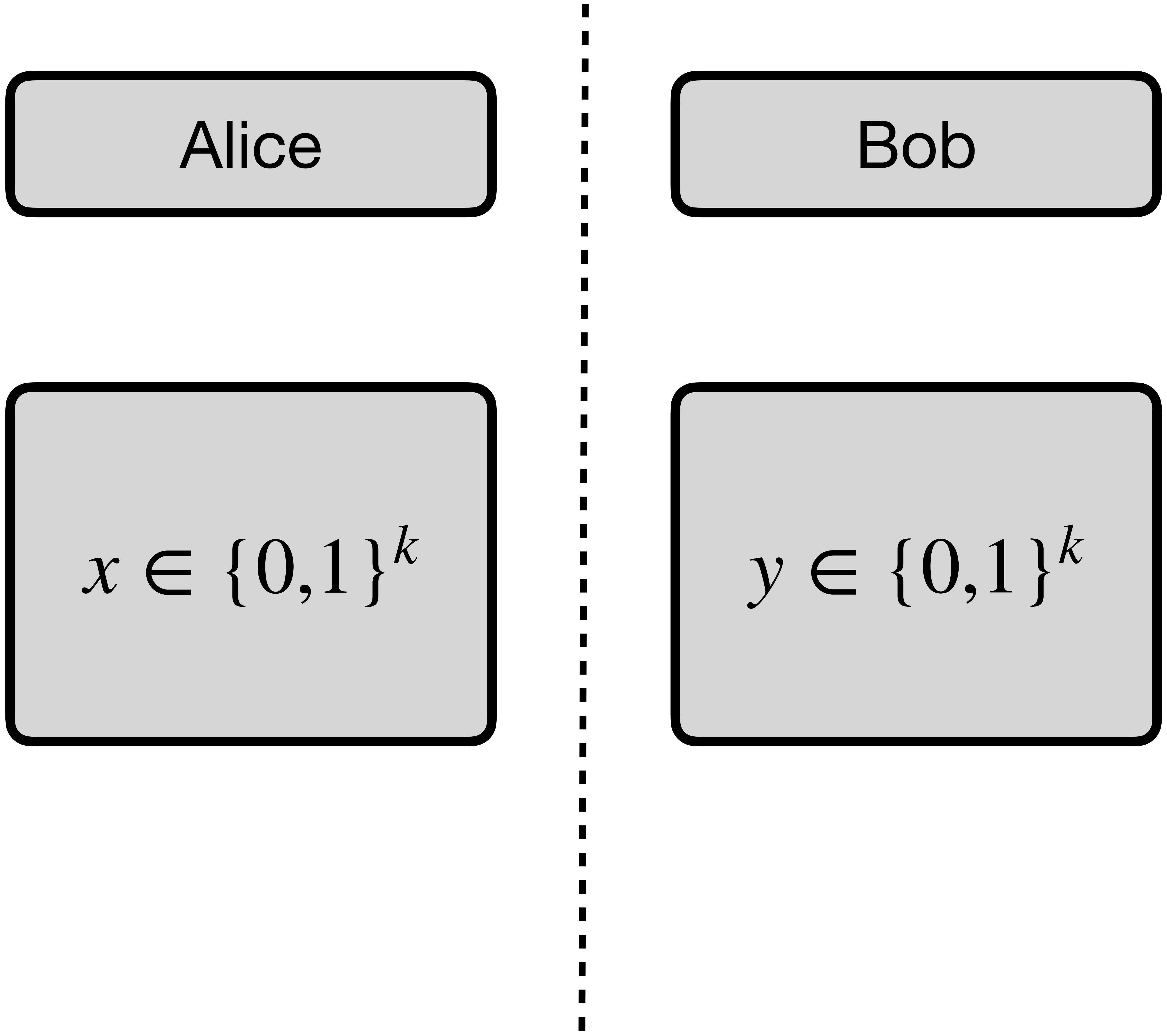
- Based on the 2-party communication complexity reduction framework of [CKP17].
- Typically reduce from Set Disjointness.
  - $\text{SD}(x, y) = $ False if $x_i = y_i = 1$.
  - $\text{SD}(x, y) = $ True otherwise



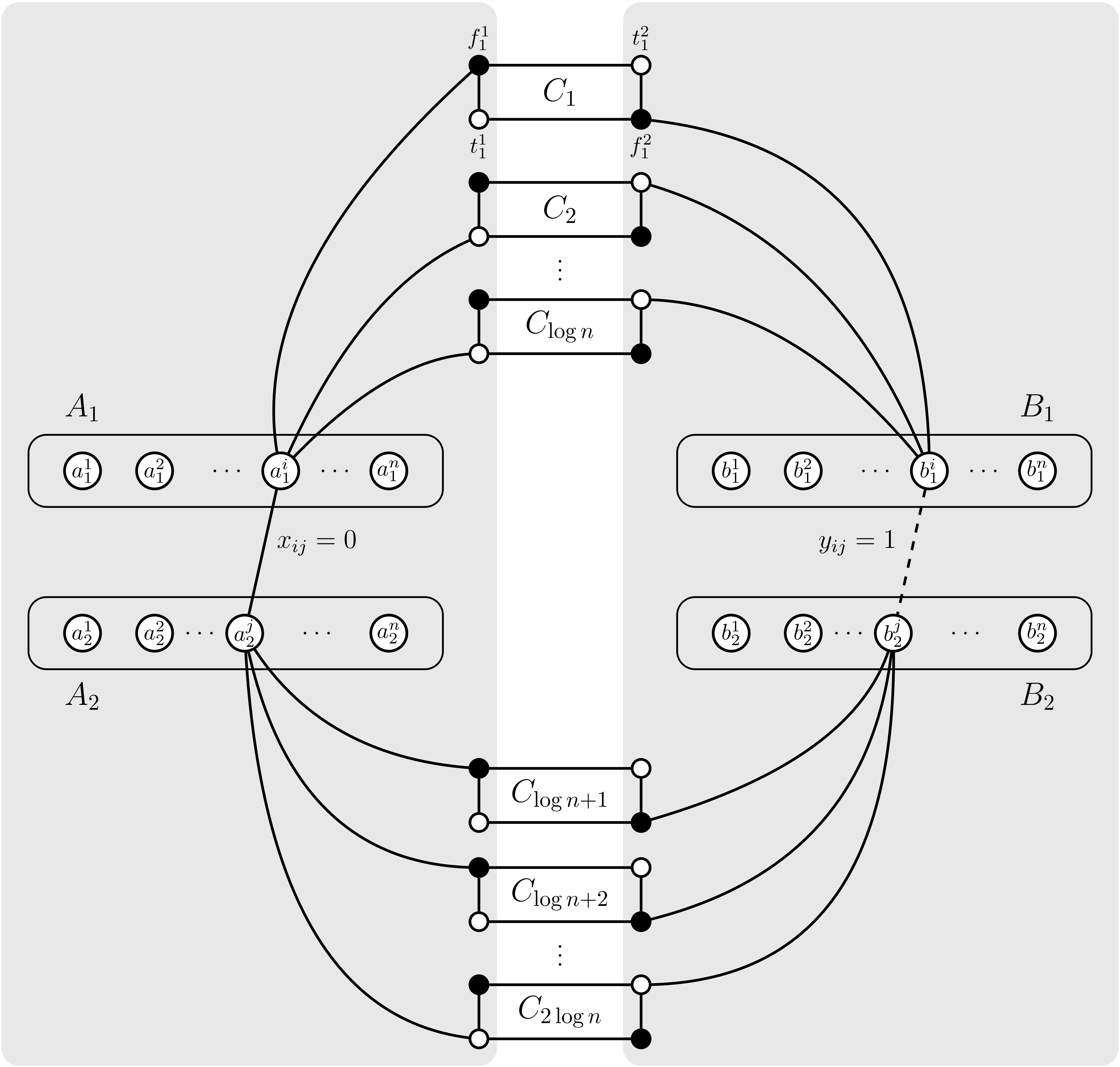
[CKP17] *Censor-Hillel, Khoury, Paz. DISC 2017*

Lower Bound for MVC

- Based on the 2-party communication complexity reduction framework of [CKP17].
- Typically reduce from Set Disjointness.
  - SD(x, y) = False if $x_i = y_i = 1$.
  - SD(x, y) = True otherwise
- Alice and Bob need to exchange $\Omega(k)$ bits to compute SD(x, y).

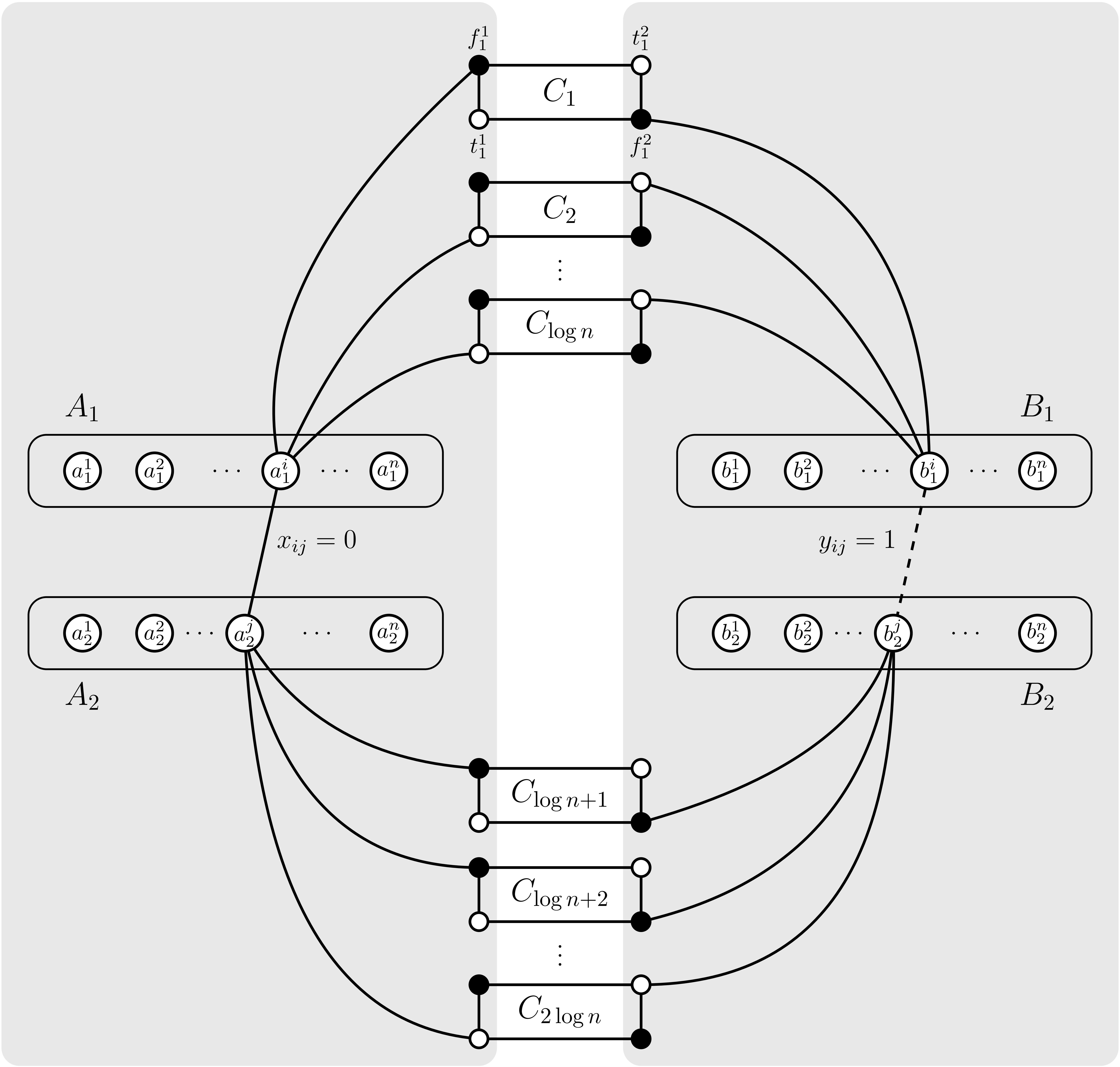


[CKP17] Censor-Hillel, Khoury, Paz. DISC 2017



f_1^1
t_1^2
C_1
t_1^1
f_1^2
C_2
$C_{\log n}$
A_1
B_1
a_1^1 a_1^2 ⋯ a_1^i ⋯ a_1^n
b_1^1 b_1^2 ⋯ b_1^i ⋯ b_1^n
$x_{ij} = 0$
$y_{ij} = 1$
a_2^1 a_2^2 ⋯ a_2^j ⋯ a_2^n
b_2^1 b_2^2 ⋯ b_2^j ⋯ b_2^n
A_2
B_2
$C_{\log n+1}$
$C_{\log n+2}$
$C_{2 \log n}$

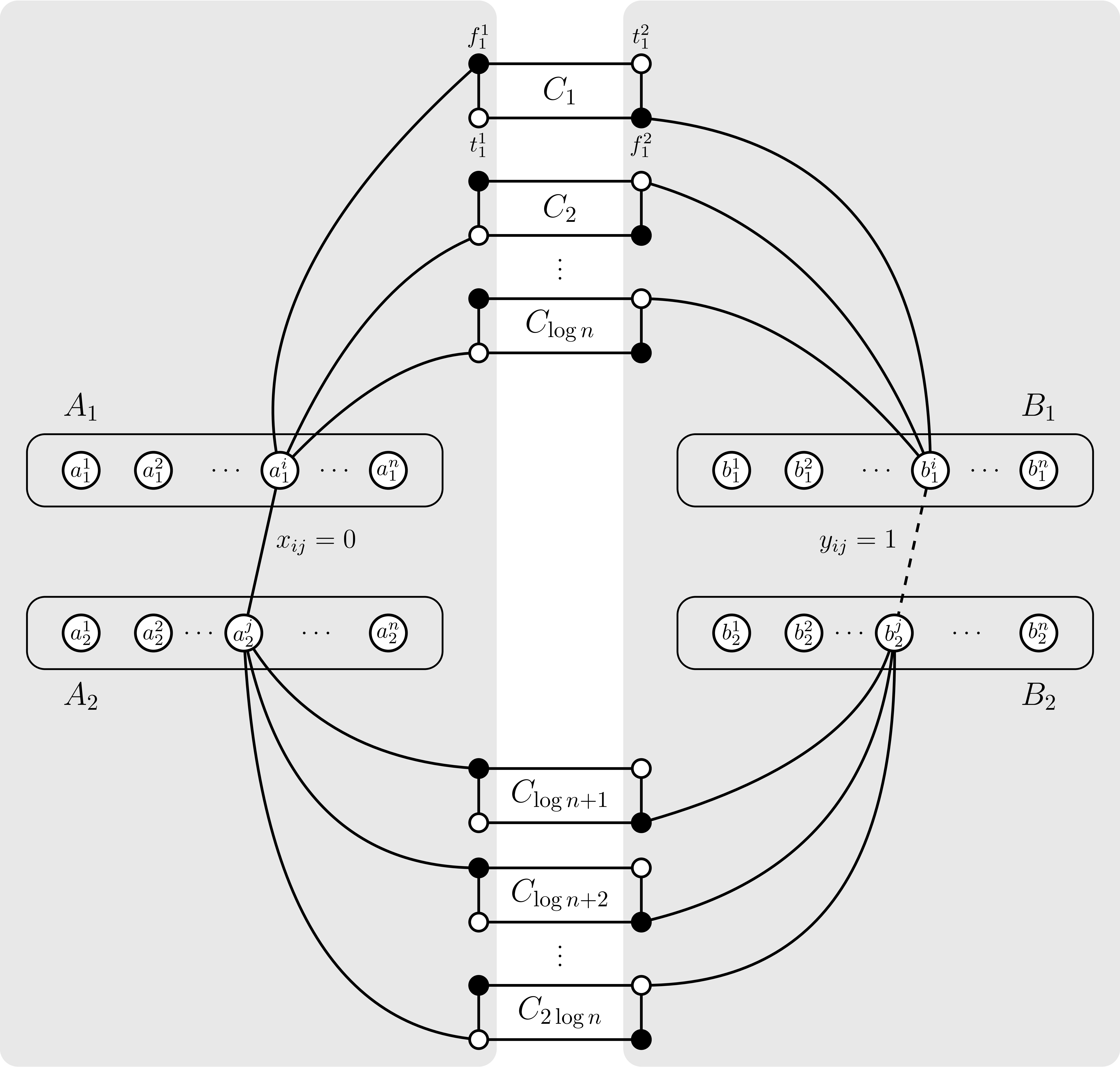
SD(x, y) = True iff $G_{x,y}$ has an MVC of size at least $C(n)$.



SD(x, y) = True iff $G_{x,y}$ has an MVC of size at least $C(n)$.

Computing MVC of $G_{x,y}$ requires $\tilde{\Omega}(n^2)$ rounds.

[CKP17] Censor-Hillel, Khoury, Paz. DISC 2017



Round Lower Bound

Alice

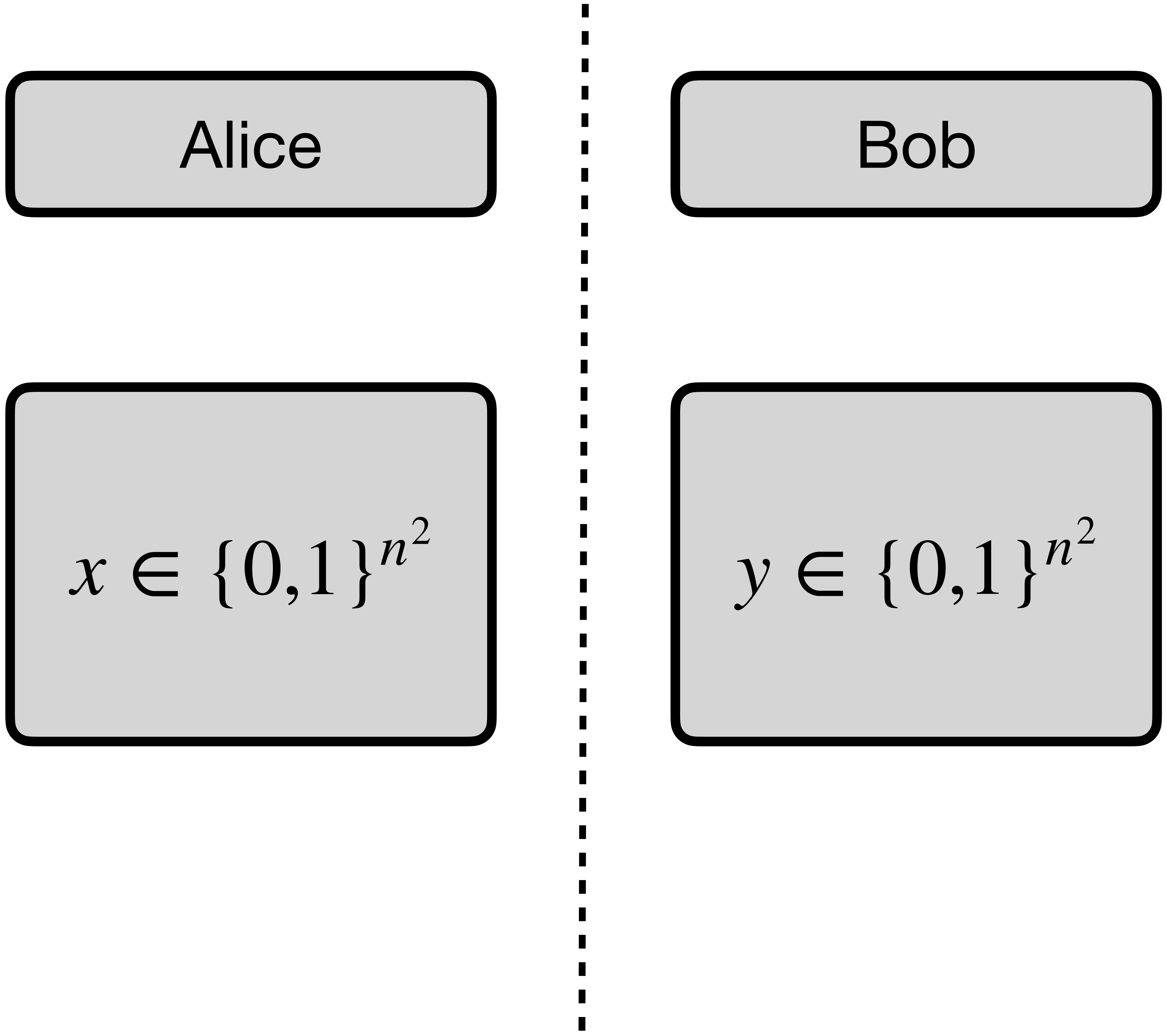
Bob

$x \in \{0,1\}^{n^2}$

$y \in \{0,1\}^{n^2}$

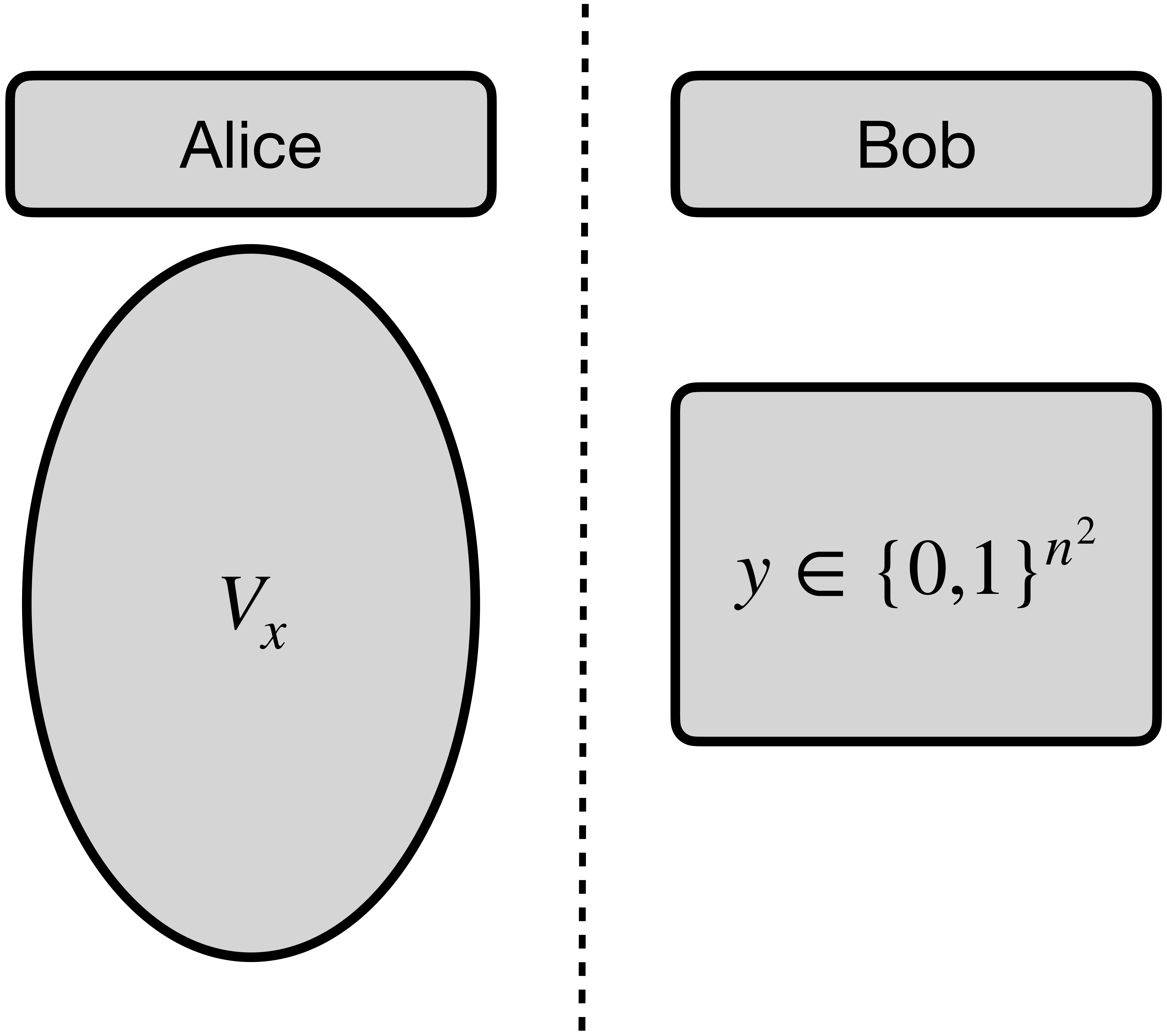
Round Lower Bound

- Alice and Bob first construct $G_{x,y}$



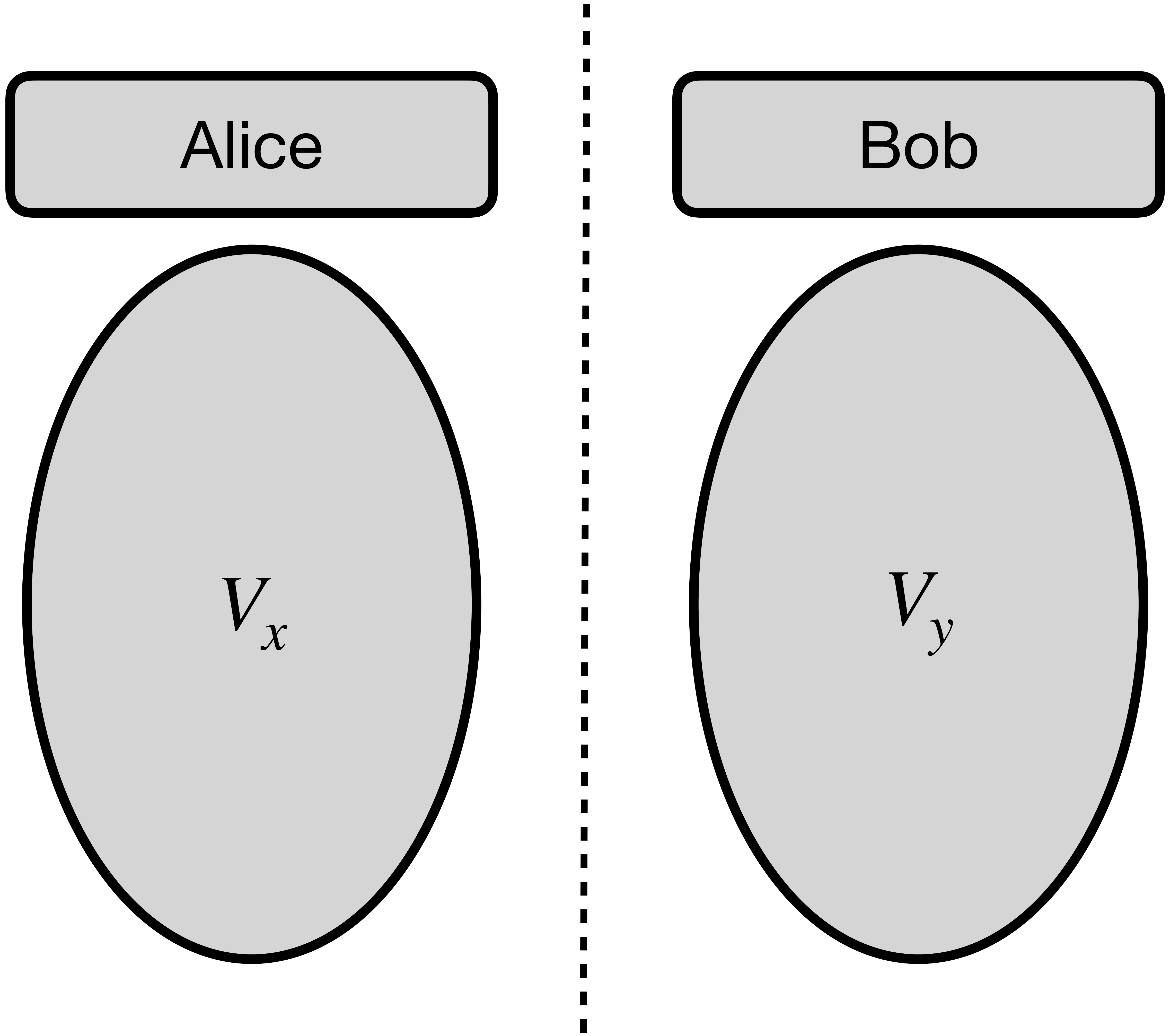
Round Lower Bound

- Alice and Bob first construct $G_{x,y}$



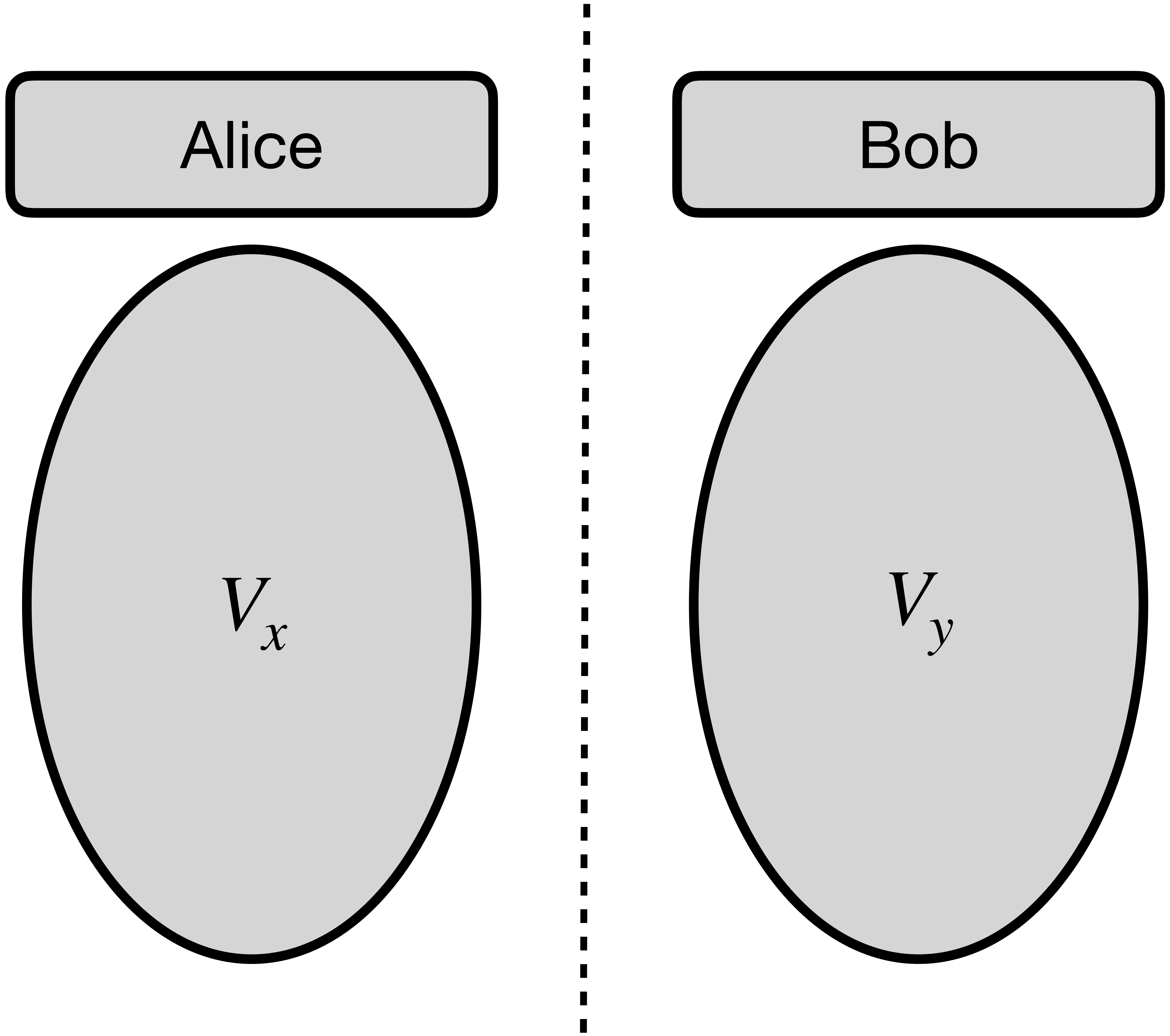
Round Lower Bound

- Alice and Bob first construct $G_{x,y}$



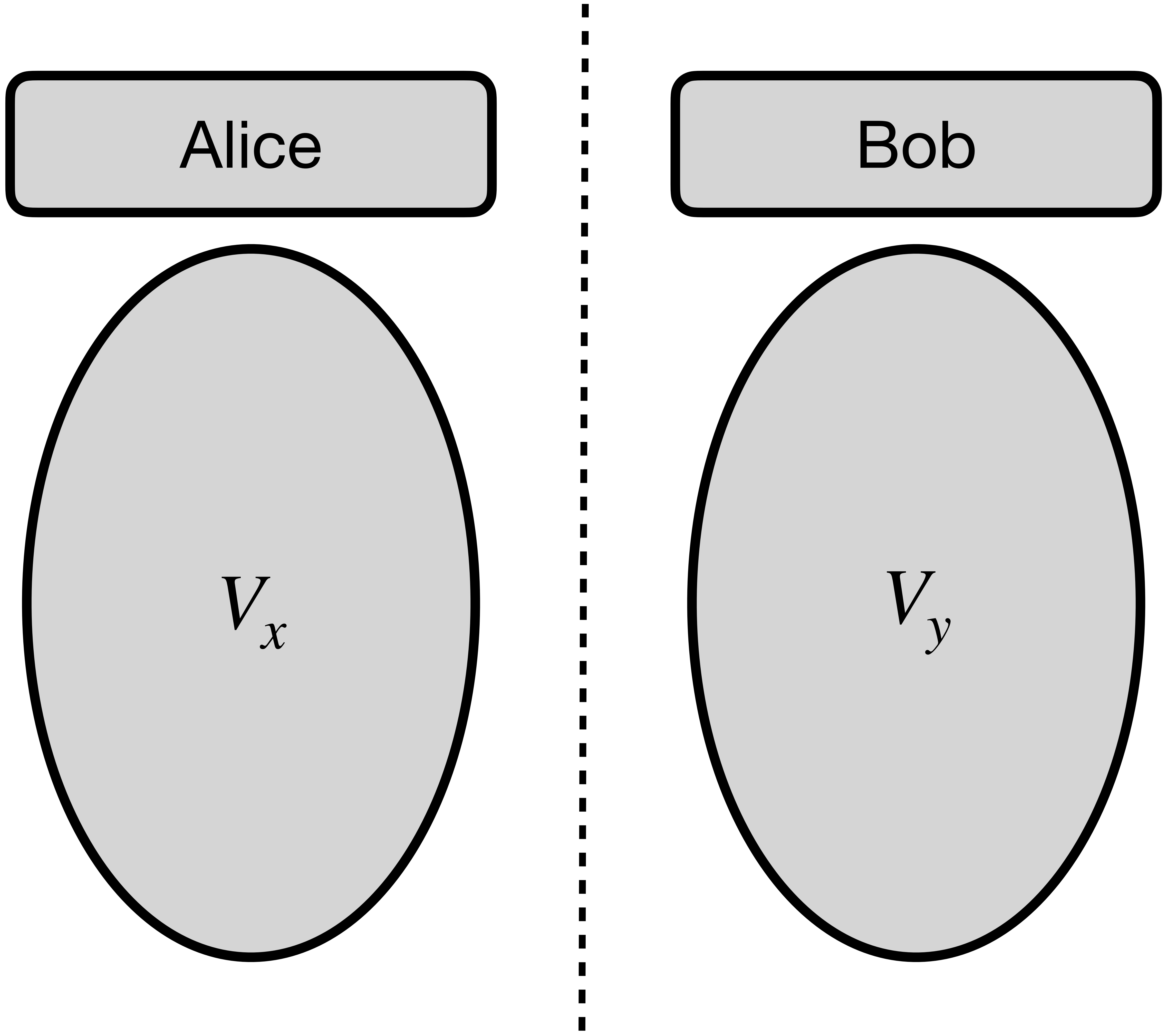
Round Lower Bound

- Alice and Bob first construct $G_{x,y}$
- Then simulate an r-round CONGEST algorithm that computes size of MVC.



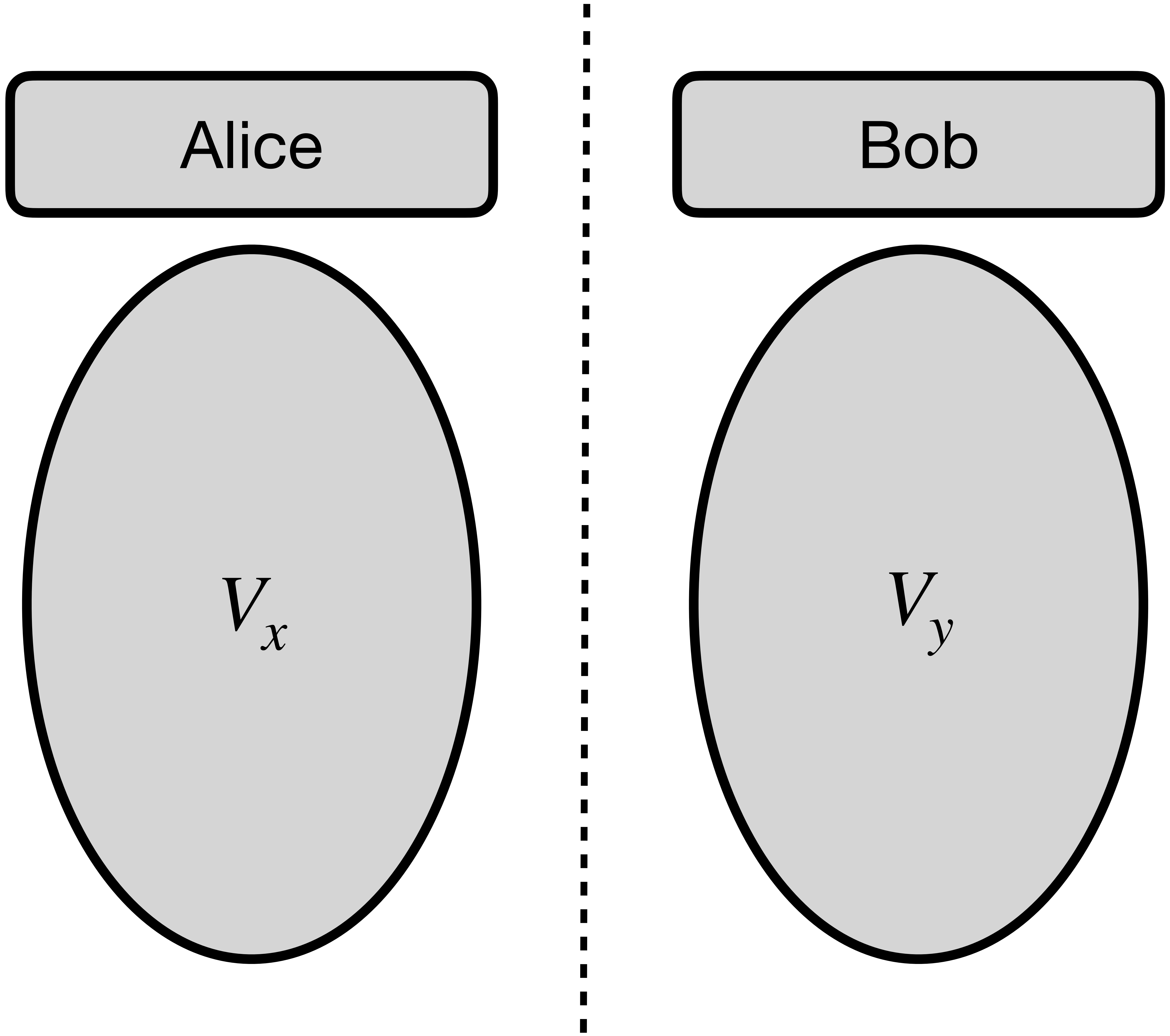
Round Lower Bound

- Alice and Bob first construct $G_{x,y}$
- Then simulate an r-round CONGEST algorithm that computes size of MVC.
    - Each round requires Alice and Bob to exchange $O(\log^2 n)$ bits.

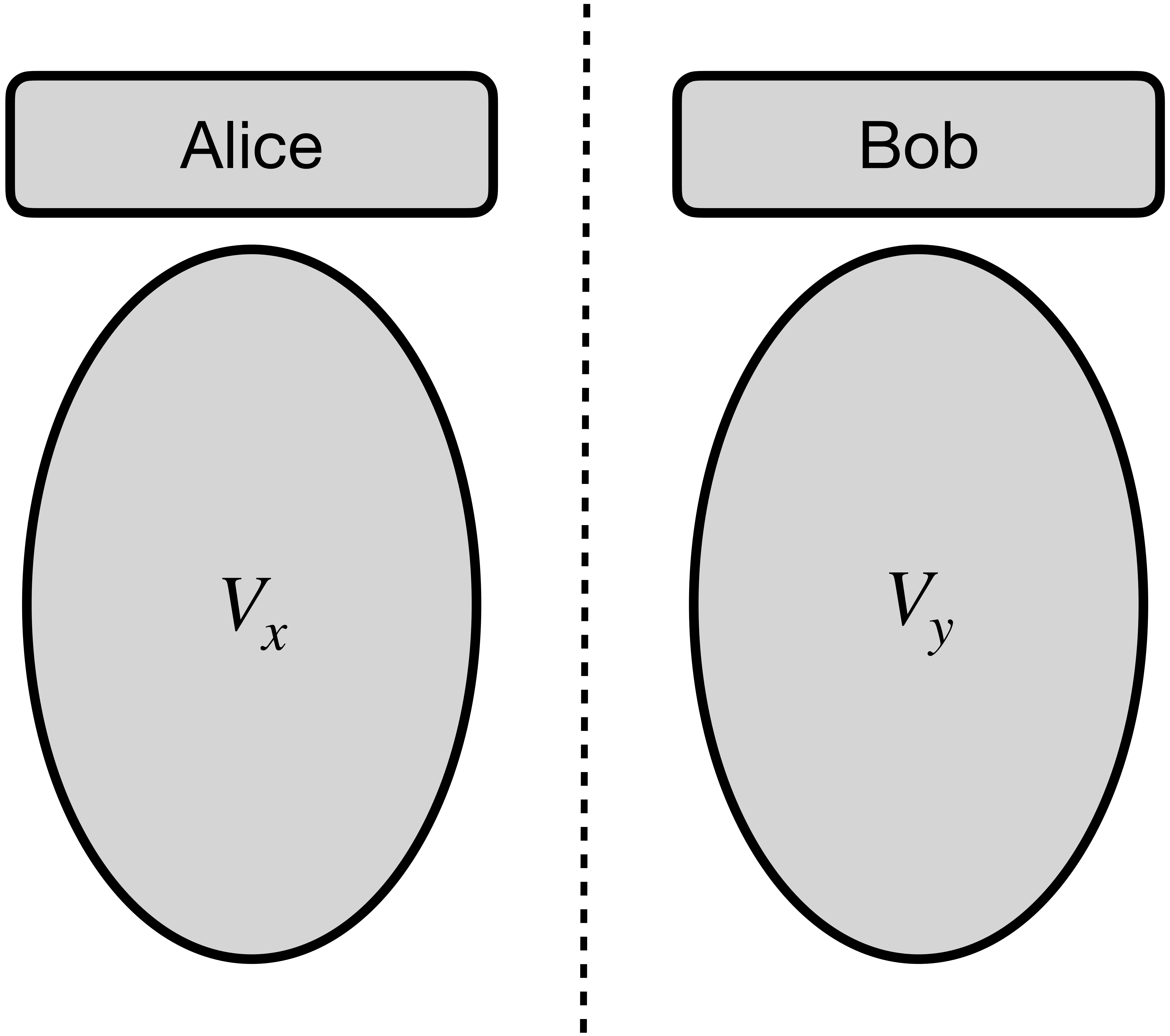


Round Lower Bound

- Alice and Bob first construct $G_{x,y}$
- Then simulate an r-round CONGEST algorithm that computes size of MVC.
    - Each round requires Alice and Bob to exchange $O(\log^2 n)$ bits.
- By the SD lower bound we must have $r \cdot \log^2 n \geq n^2$.

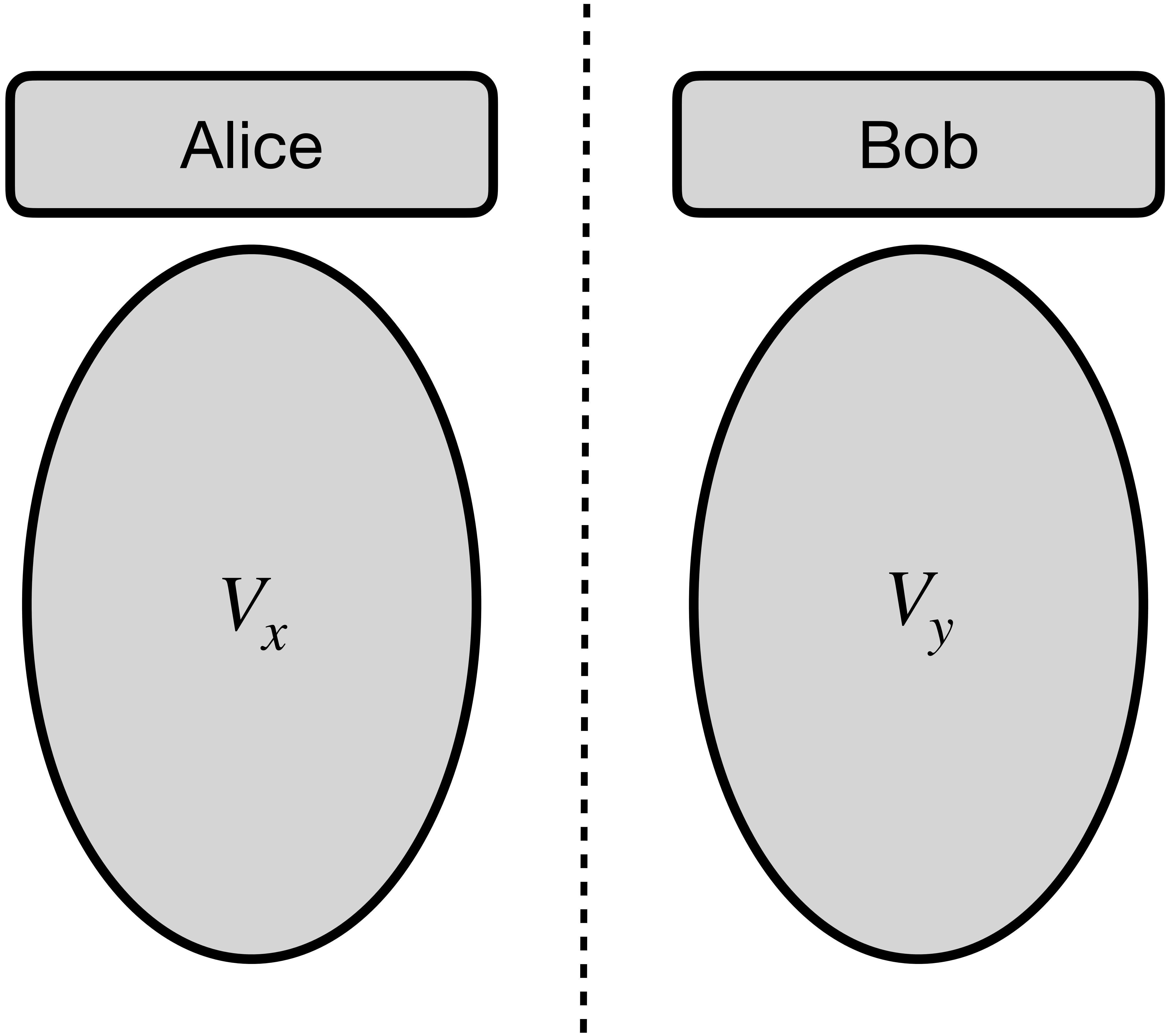


Message Lower Bound?



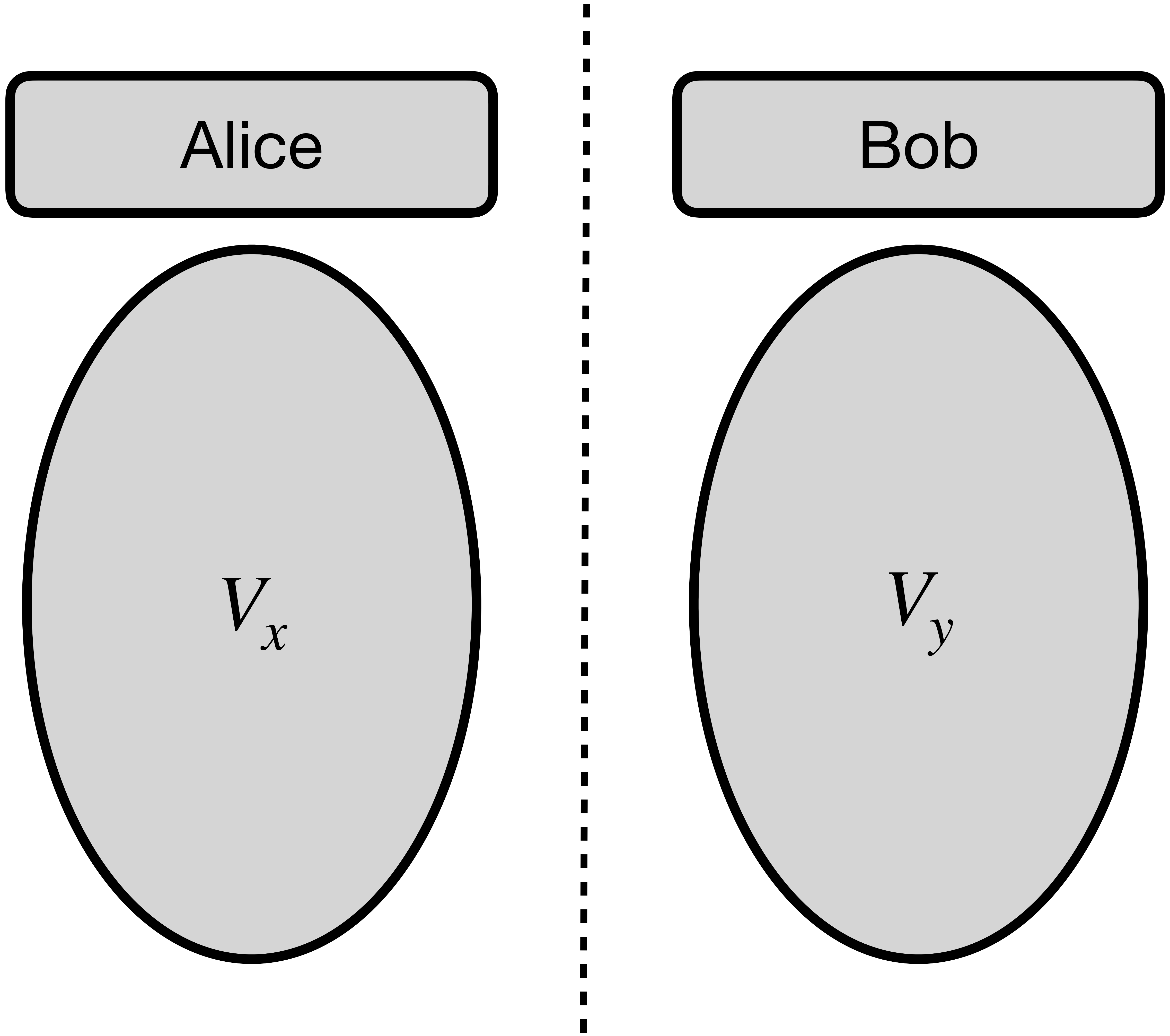
Message Lower Bound?

- The 2-party communication complexity model is asynchronous.



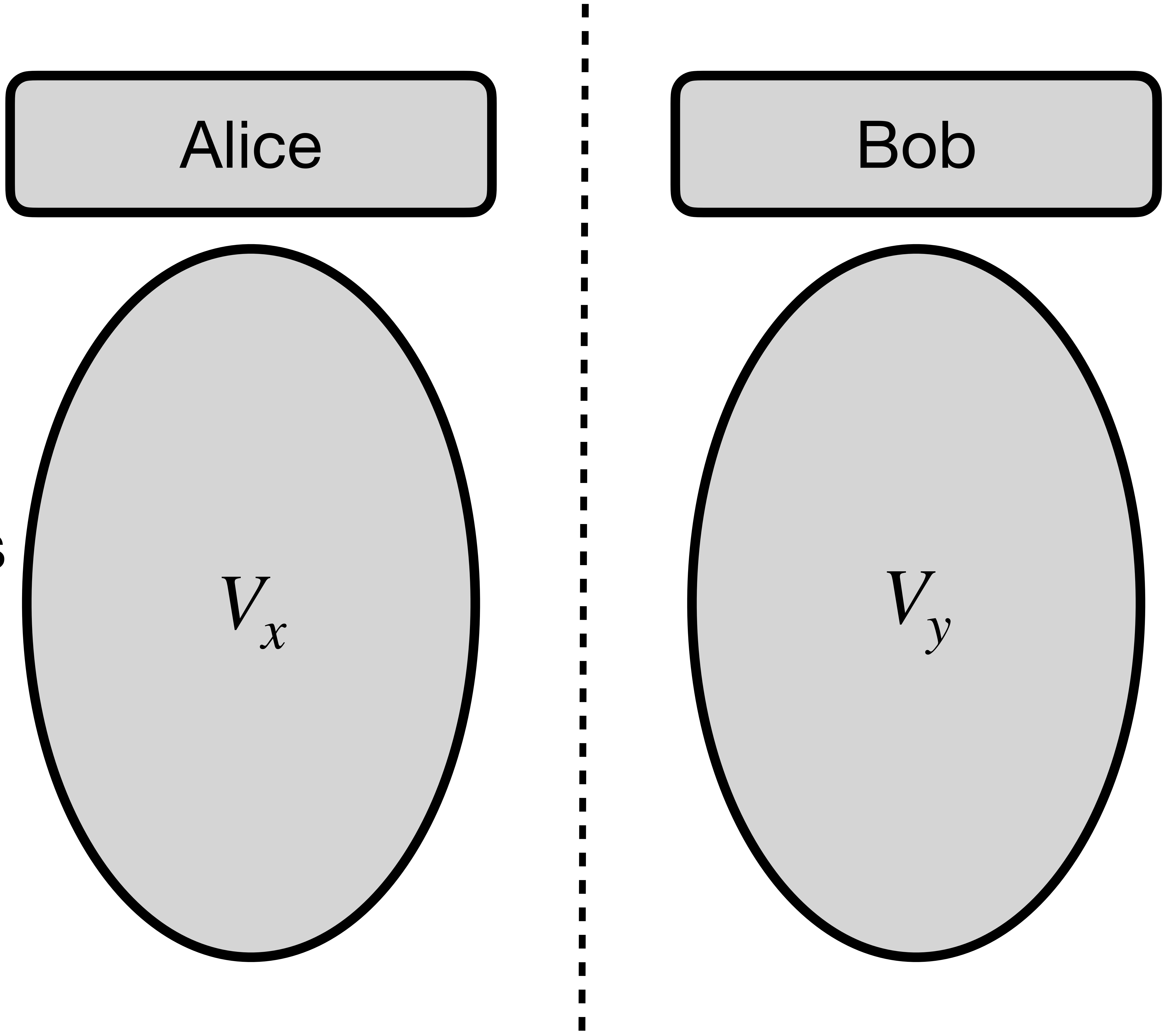
Message Lower Bound?

- The 2-party communication complexity model is asynchronous.
- Naive simulation requires Alice and Bob to send at least one bit for each round.



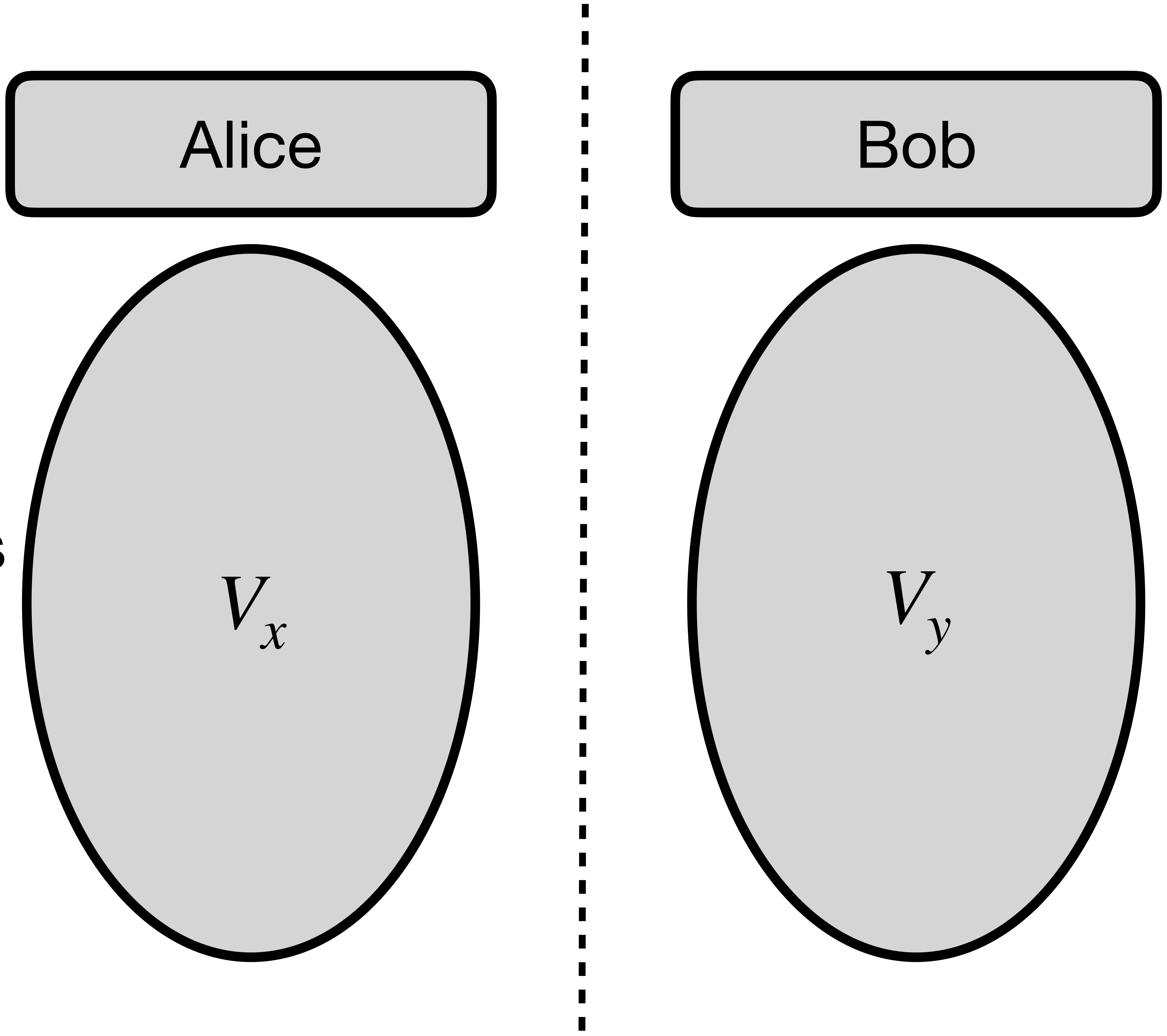
Message Lower Bound?

- The 2-party communication complexity model is asynchronous.
- Naive simulation requires Alice and Bob to send at least one bit for each round.
    - Even though the CONGEST algorithm sends no messages across the cut!



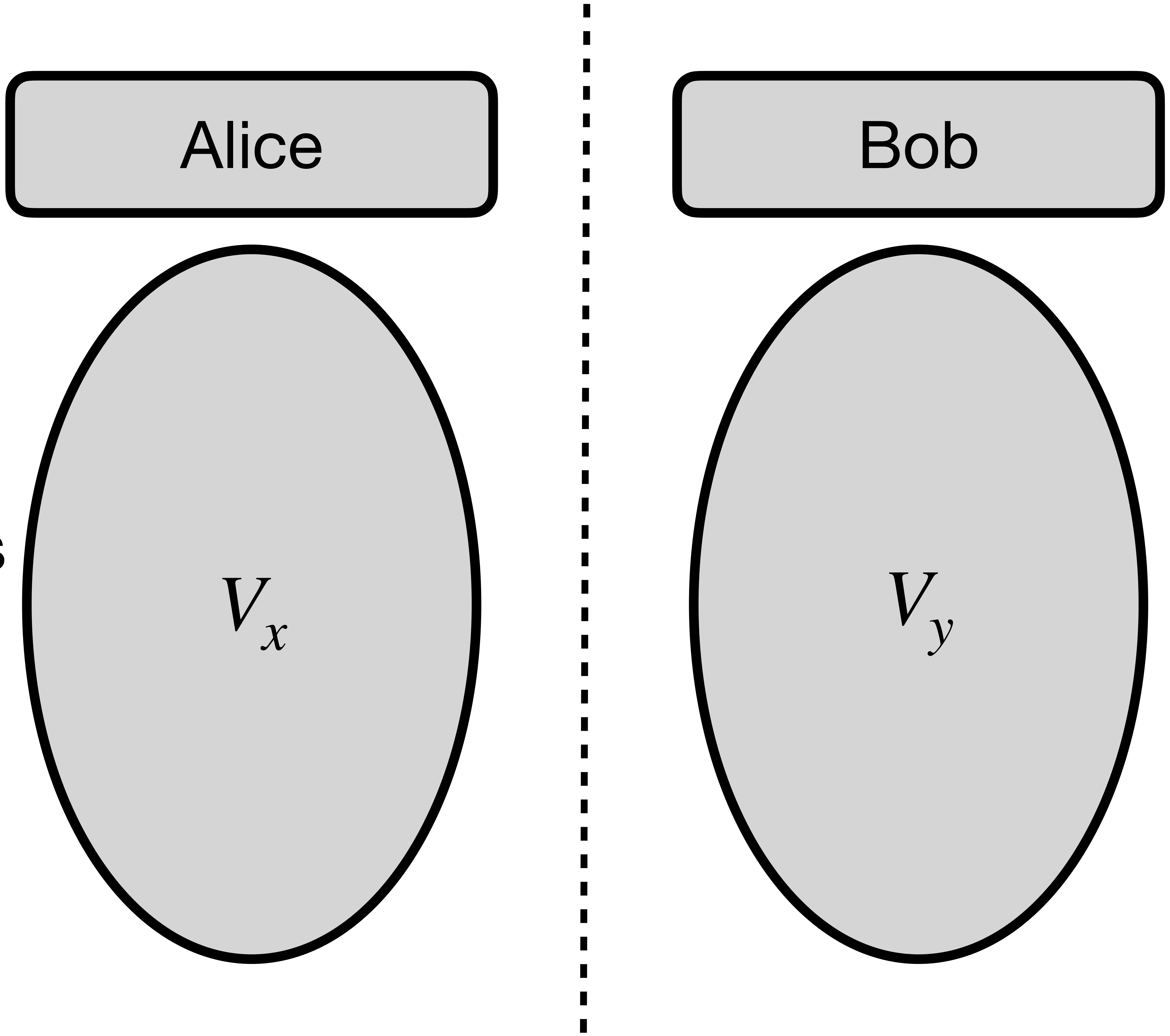
Message Lower Bound?

- The 2-party communication complexity model is asynchronous.
- Naive simulation requires Alice and Bob to send at least one bit for each round.
  - Even though the CONGEST algorithm sends no messages across the cut!
  - Need $o(n^2)$ round budget.

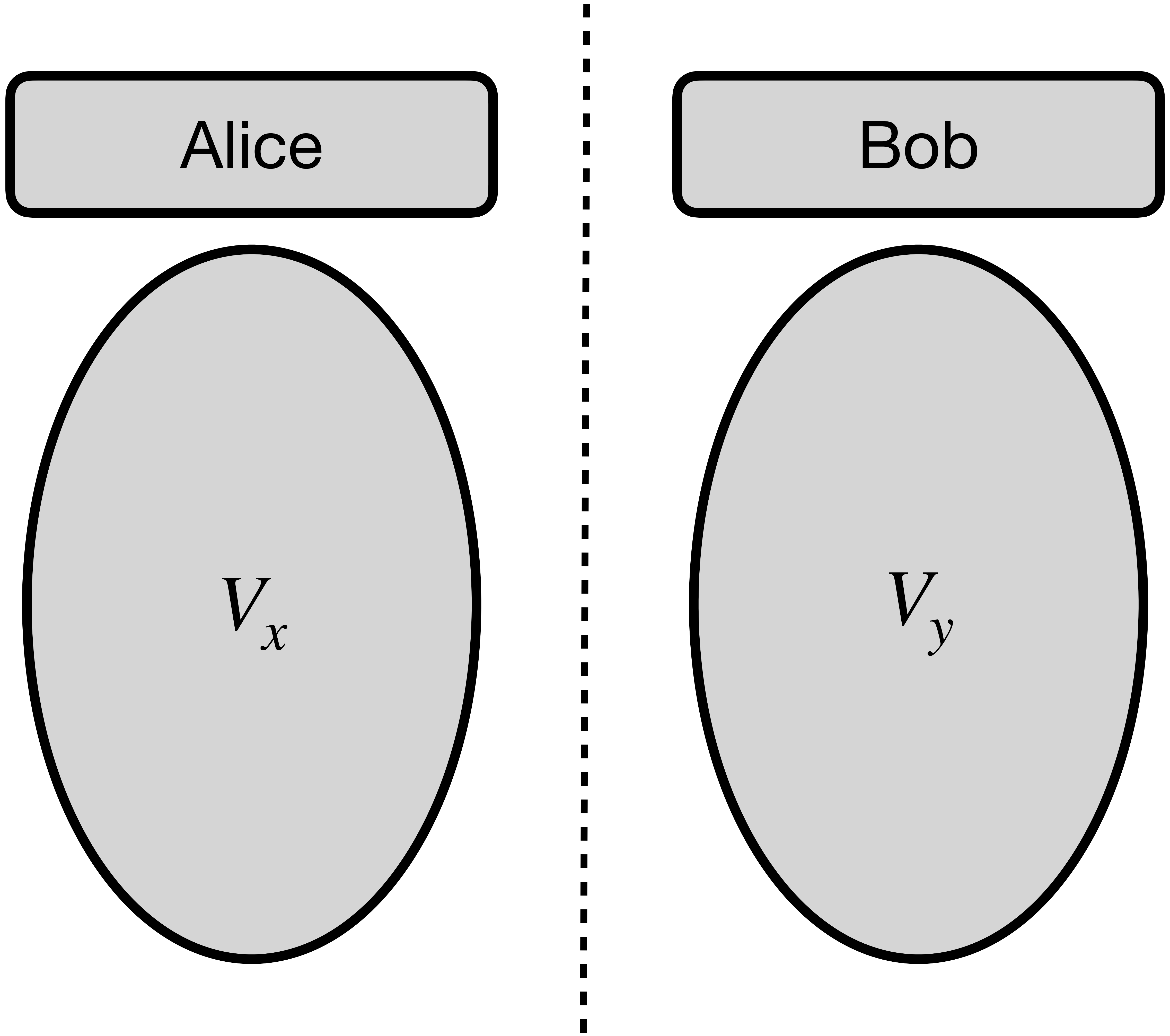


Message Lower Bound?

- The 2-party communication complexity model is asynchronous.
- Naive simulation requires Alice and Bob to send at least one bit for each round.
    - Even though the CONGEST algorithm sends no messages across the cut!
    - Need $o(n^2)$ round budget.
    - Can we do better?

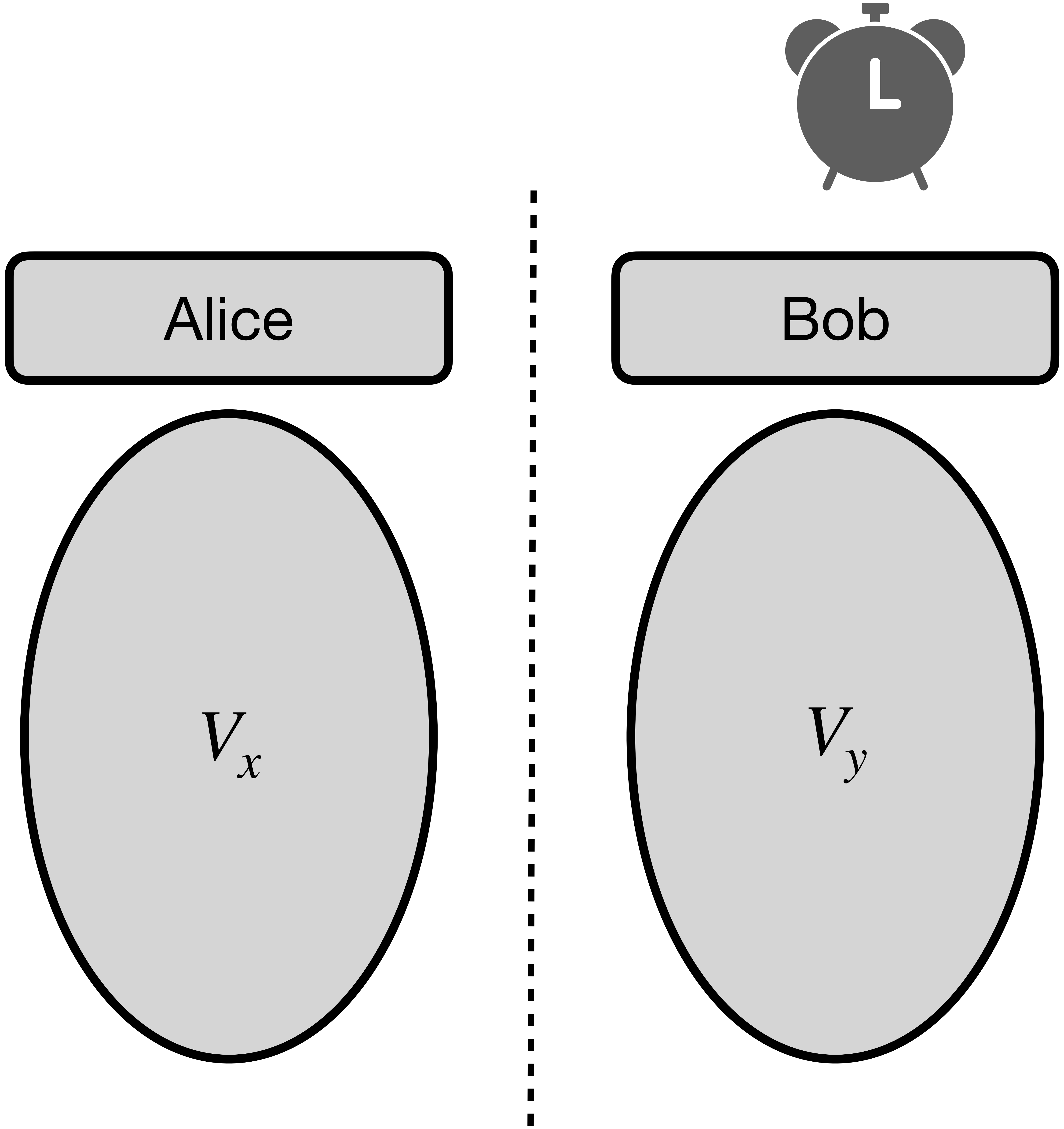


Synchronous CC



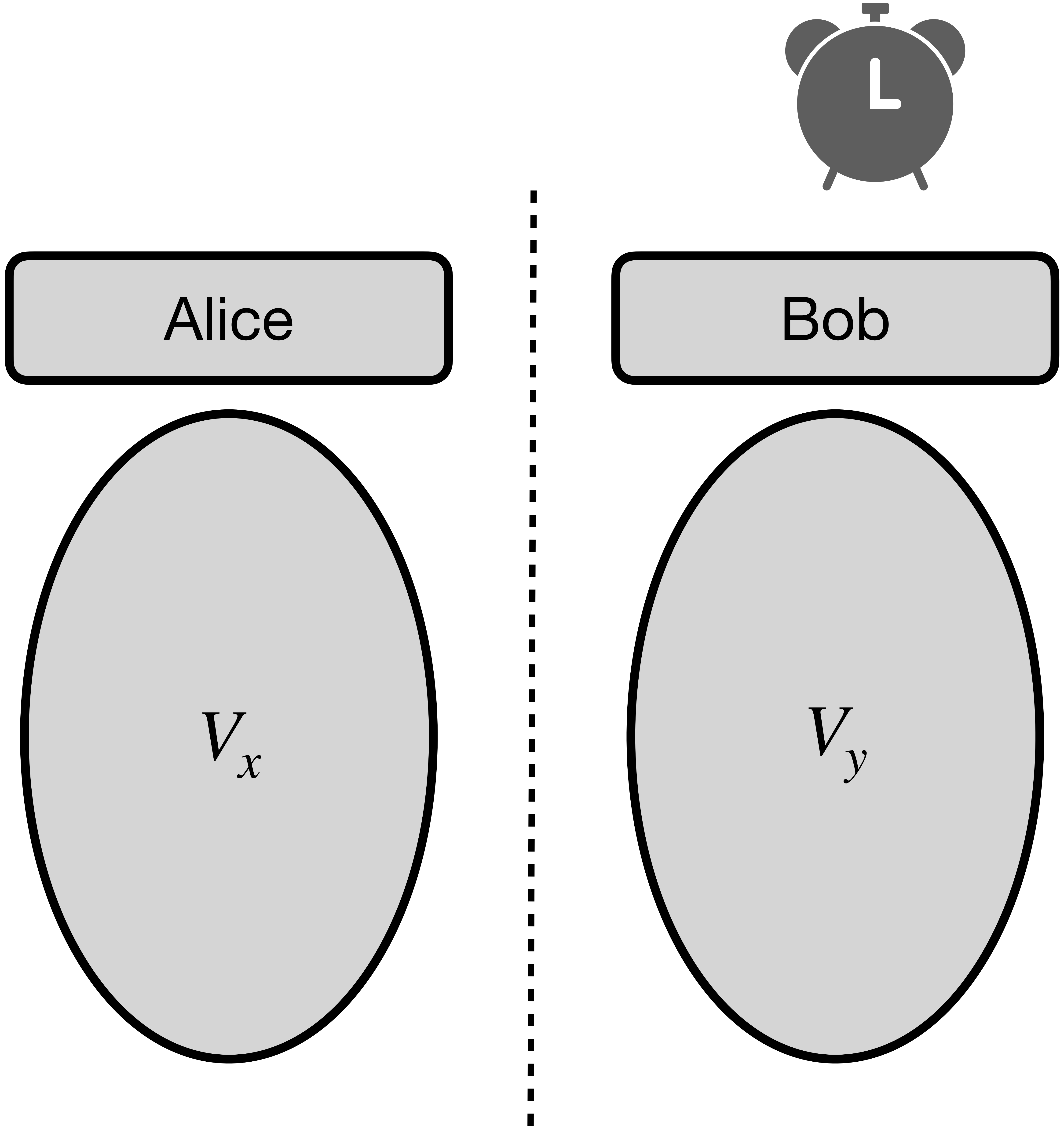
Synchronous CC

- If Alice and Bob had a clock, then they can simulate any R round CONGEST algorithm.



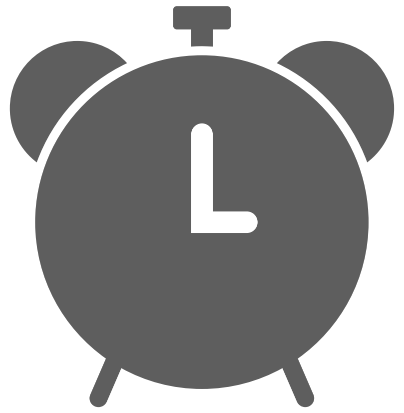
Synchronous CC

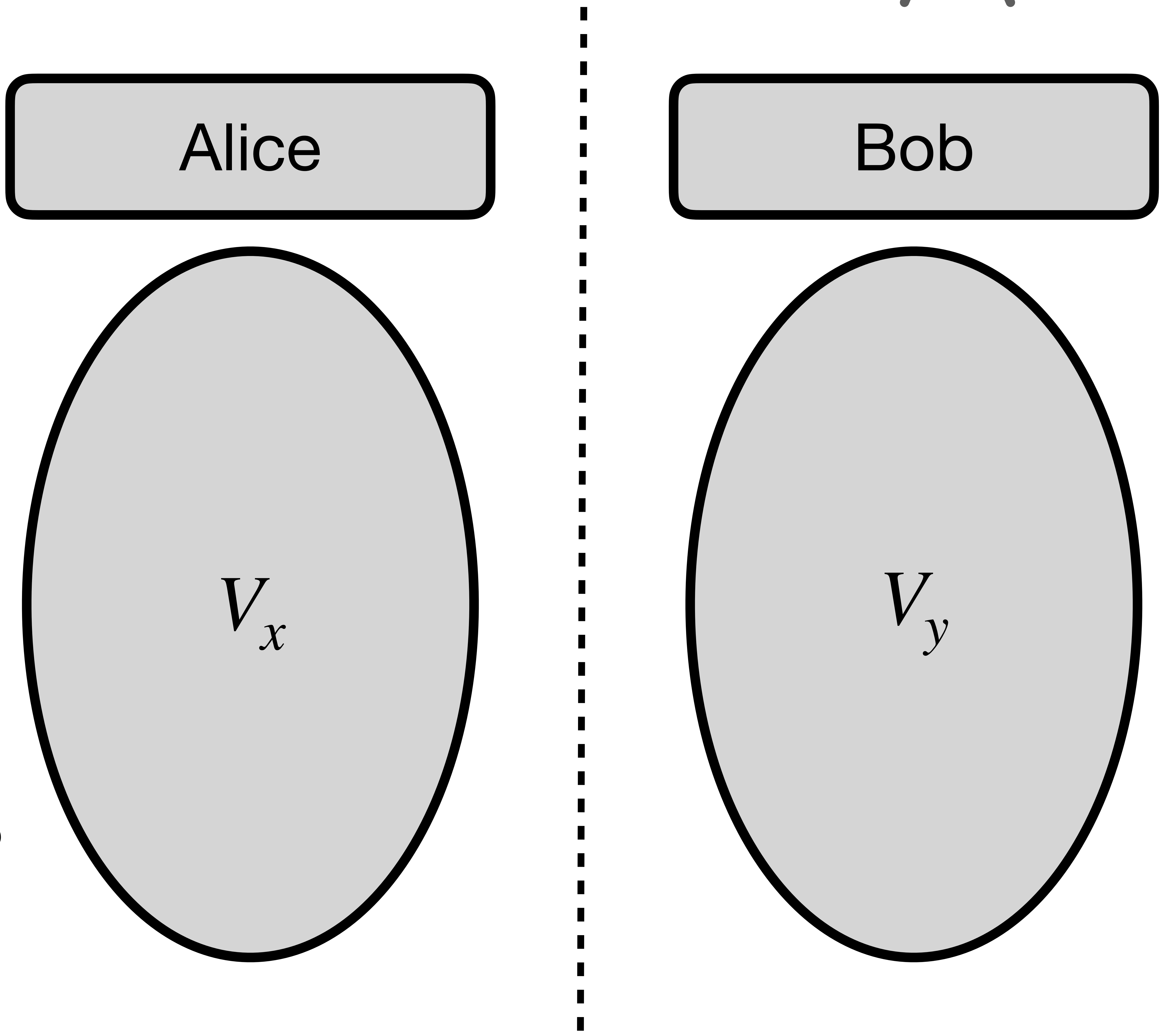
- If Alice and Bob had a clock, then they can simulate any R round CONGEST algorithm.
    - Alice and Bob don't need to communicate at all if the CONGEST algorithm does not send any messages from V_x to V_y.



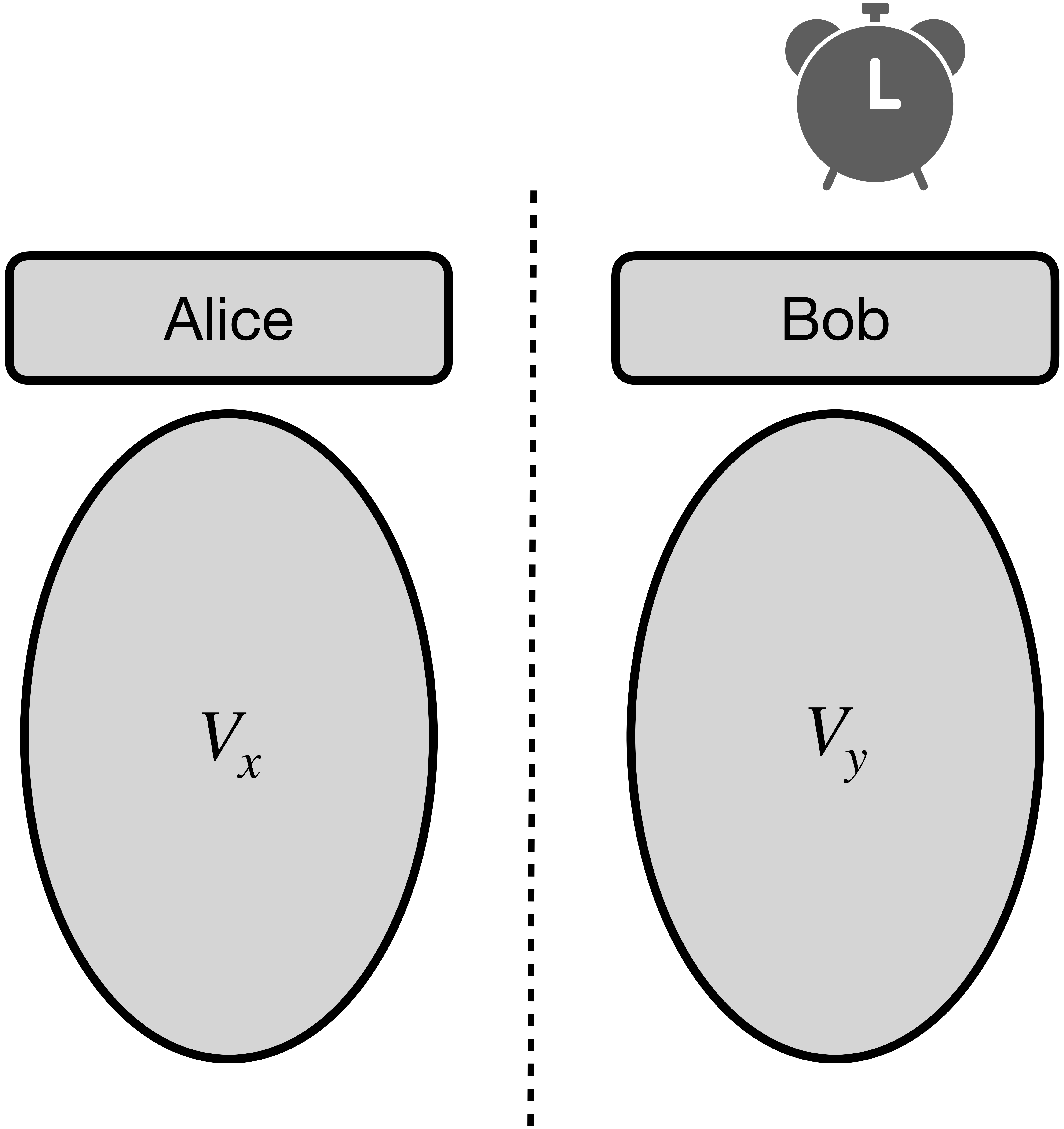
Synchronous CC

- If Alice and Bob had a clock, then they can simulate any R round CONGEST algorithm.
    - Alice and Bob don't need to communicate at all if the CONGEST algorithm does not send any messages from V_x to V_y.
- How strong are the lower bounds for synchronous communication complexity?



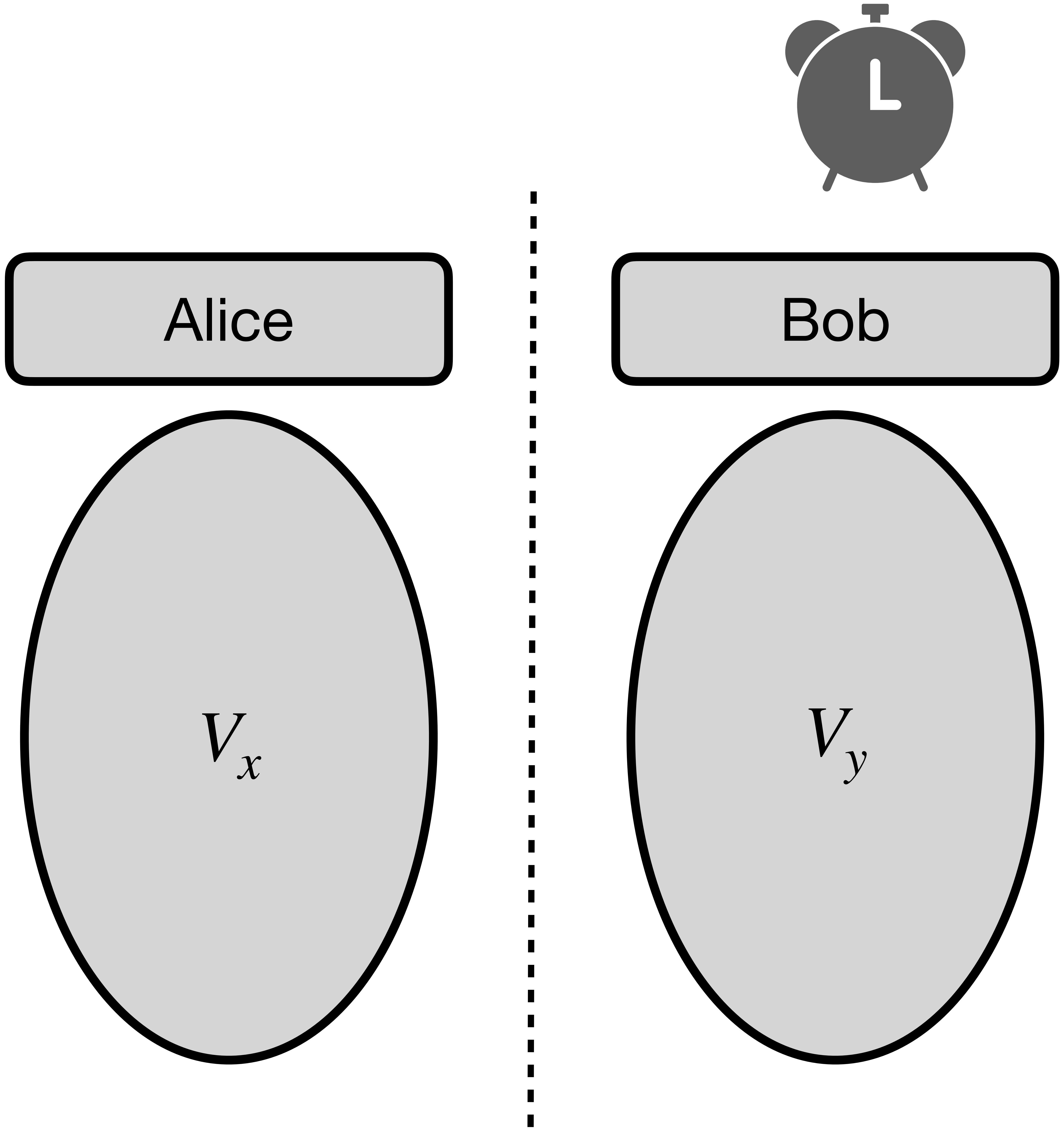


Synchronous CC



Synchronous CC

- [PPS20] give a message efficient synchronizer for clique networks which implies:

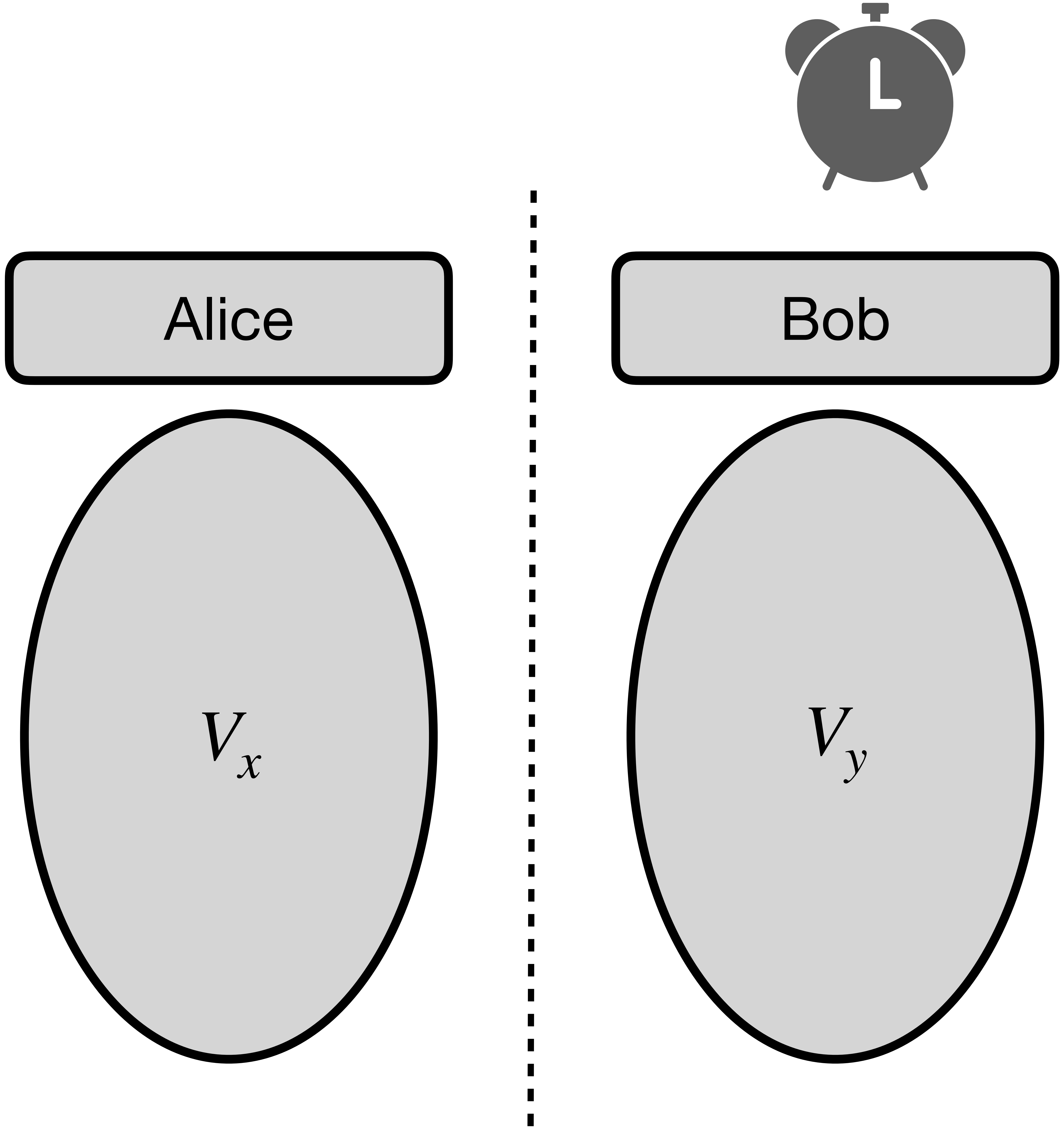


[PPS20] *Pandurangan, Peleg, Squizatto. TCS 2020*

Synchronous CC

- [PPS20] give a message efficient synchronizer for clique networks which implies:

$$SCC_r(f) \geq \Omega\left(\frac{CC(f)}{1 + \log r}\right)$$



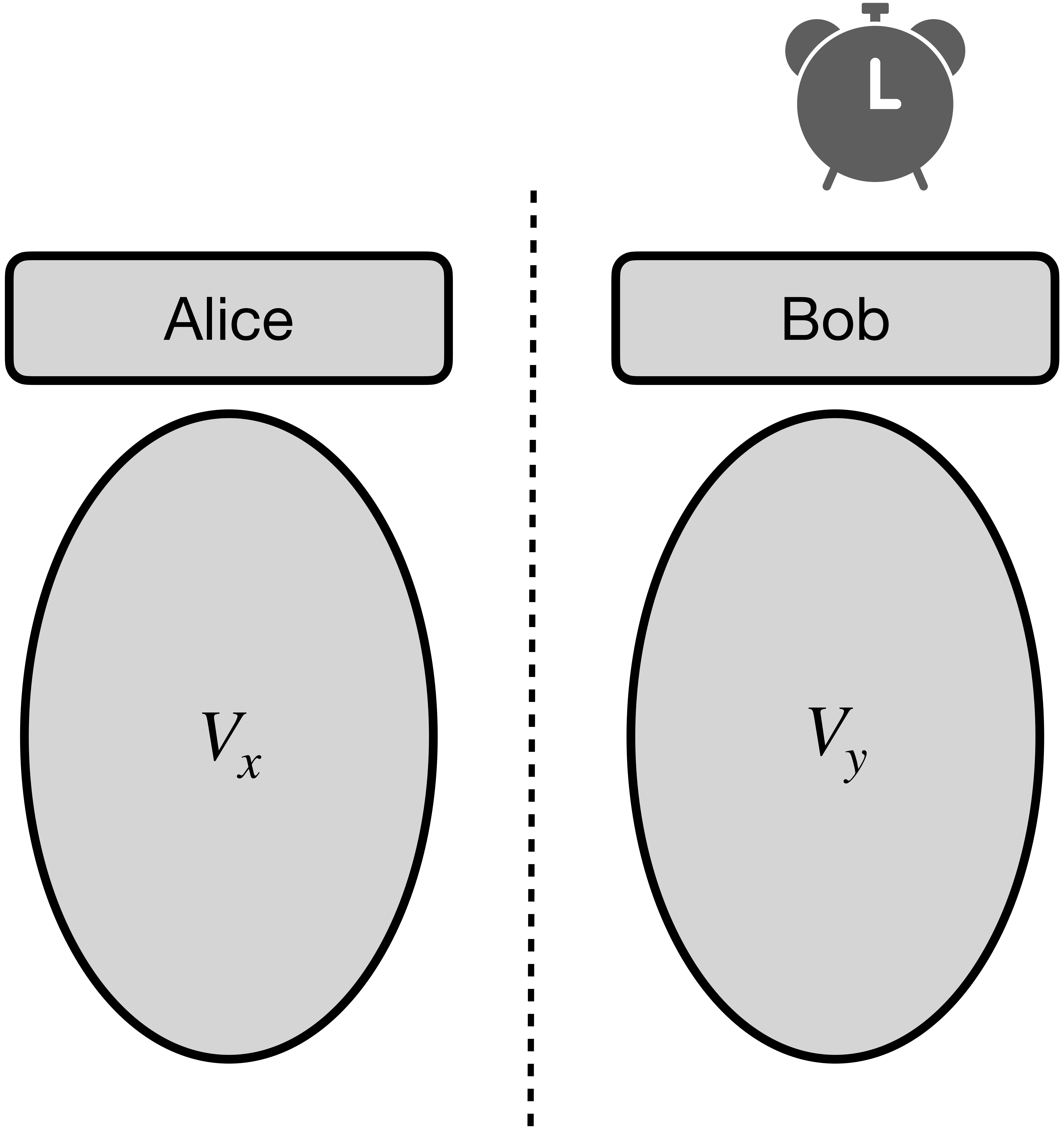
[PPS20] Pandurangan, Peleg, Squizatto. TCS 2020

Synchronous CC

- [PPS20] give a message efficient synchronizer for clique networks which implies:

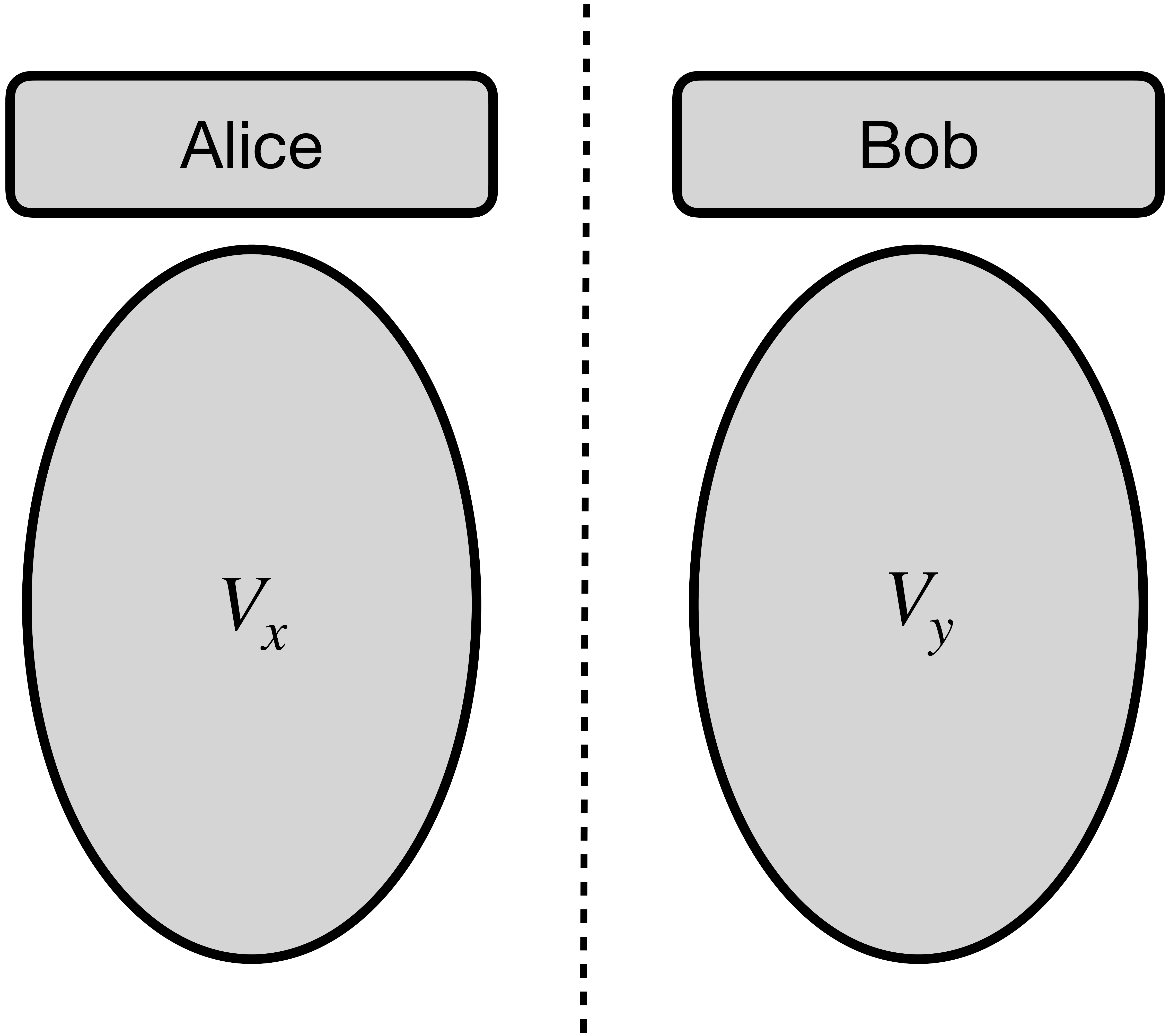
$$SCC_r(f) \geq \Omega\left(\frac{CC(f)}{1 + \log r}\right)$$

- So for poly(n) round algorithms we do not pay much.



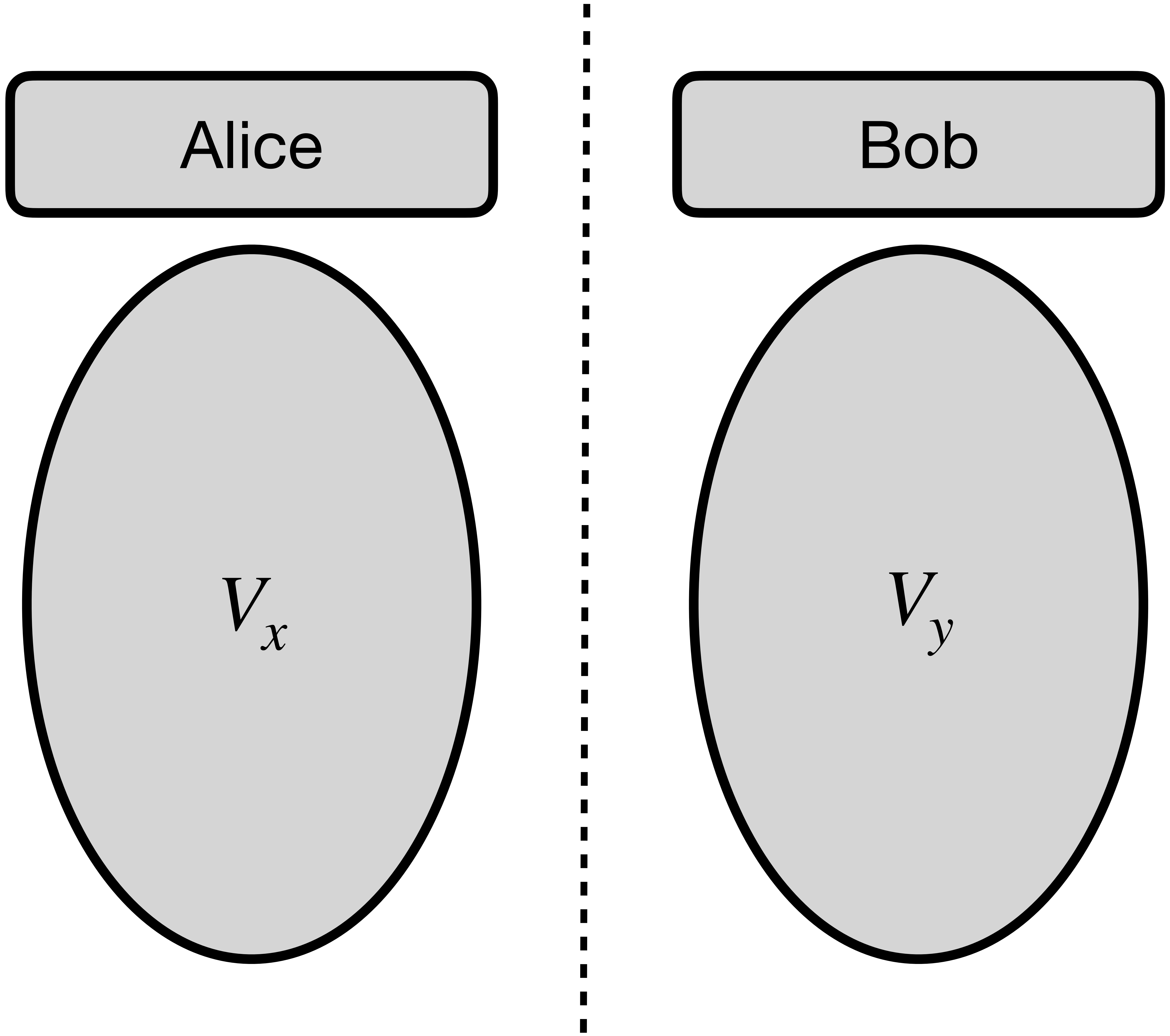
[PPS20] Pandurangan, Peleg, Squizatto. TCS 2020

Message Complexity of MVC



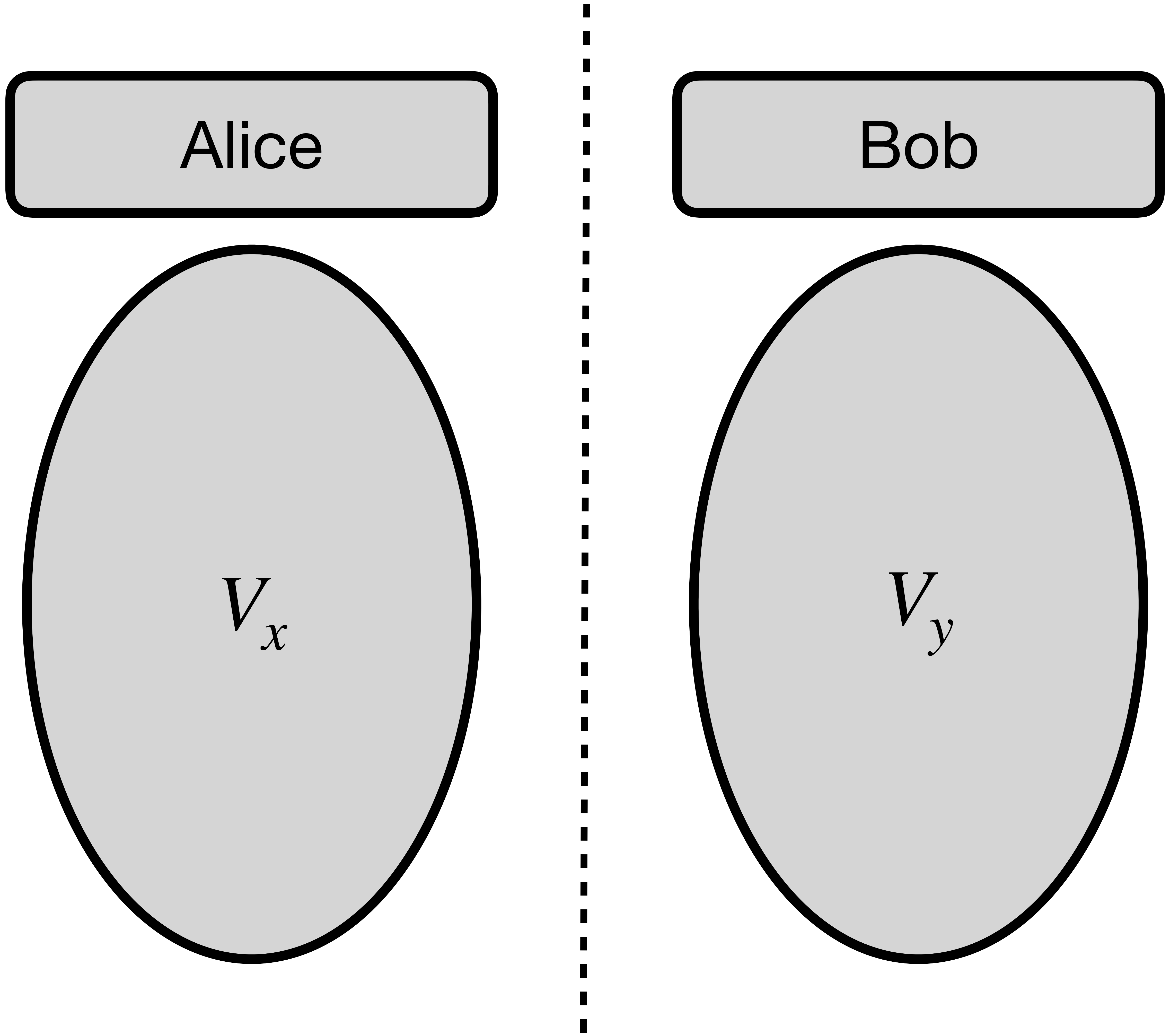
Message Complexity of MVC

- Any poly(n) round CONGEST algorithm requires $\tilde{\Omega}(n^2)$ messages.



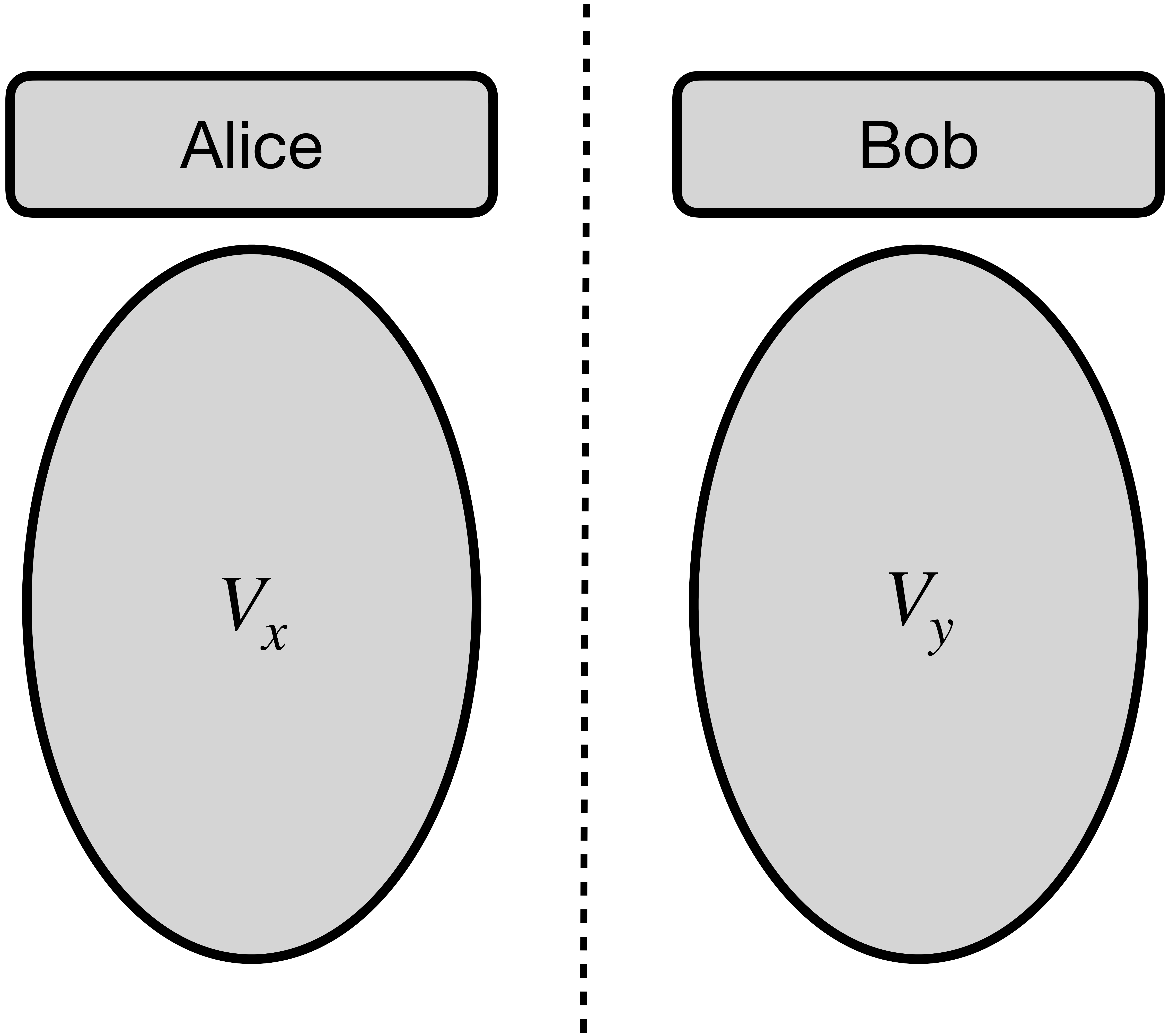
Message Complexity of MVC

- Any poly(n) round CONGEST algorithm requires $\tilde{\Omega}(n^2)$ messages.
- But we were aiming for $\Omega(m \cdot D)$ which can be as large as n^3.



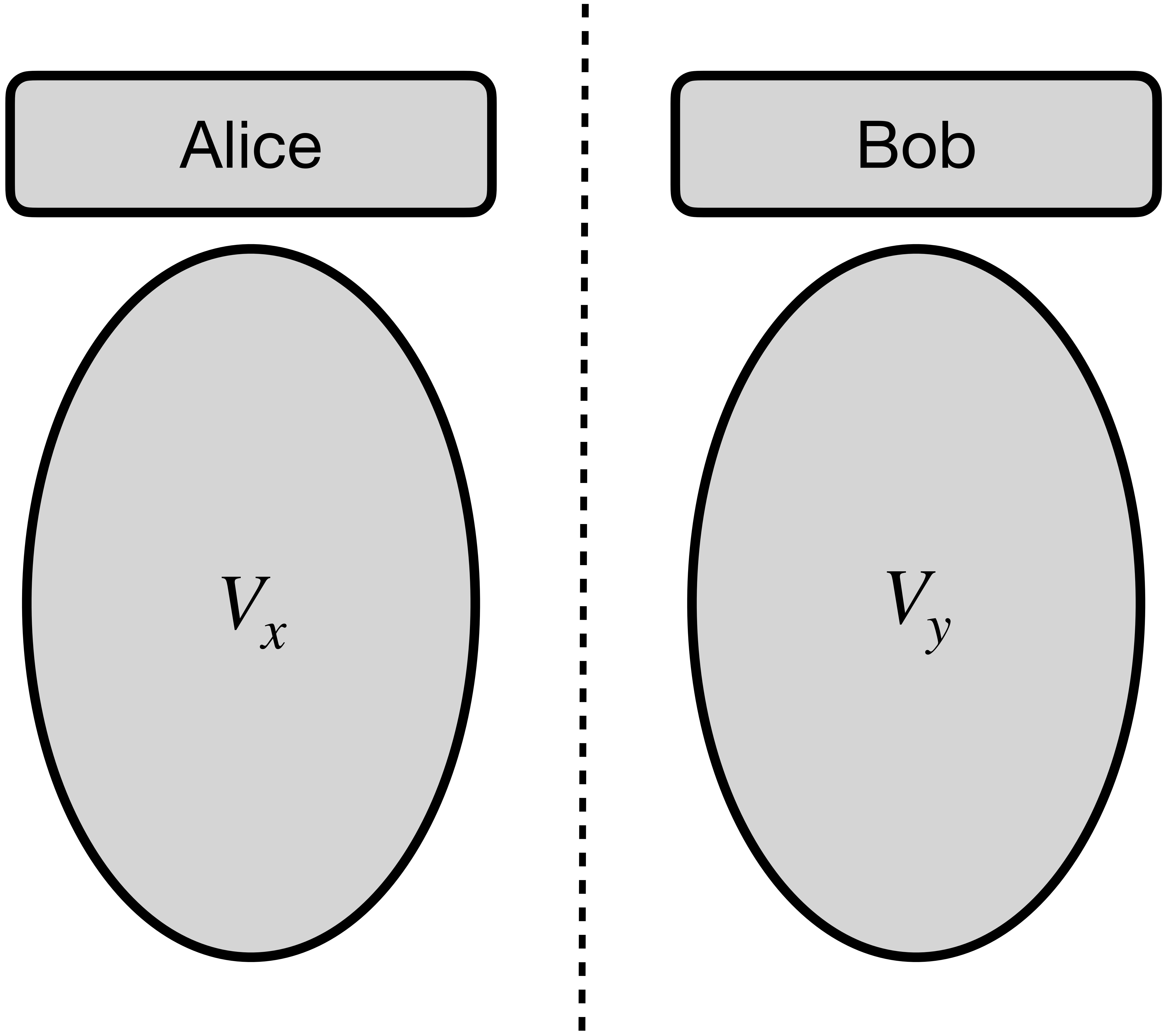
Message Complexity of MVC

- Any poly(n) round CONGEST algorithm requires $\tilde{\Omega}(n^2)$ messages.
- But we were aiming for $\Omega(m \cdot D)$ which can be as large as n^3.
- $G_{x,y}$ as described has constant diameter.

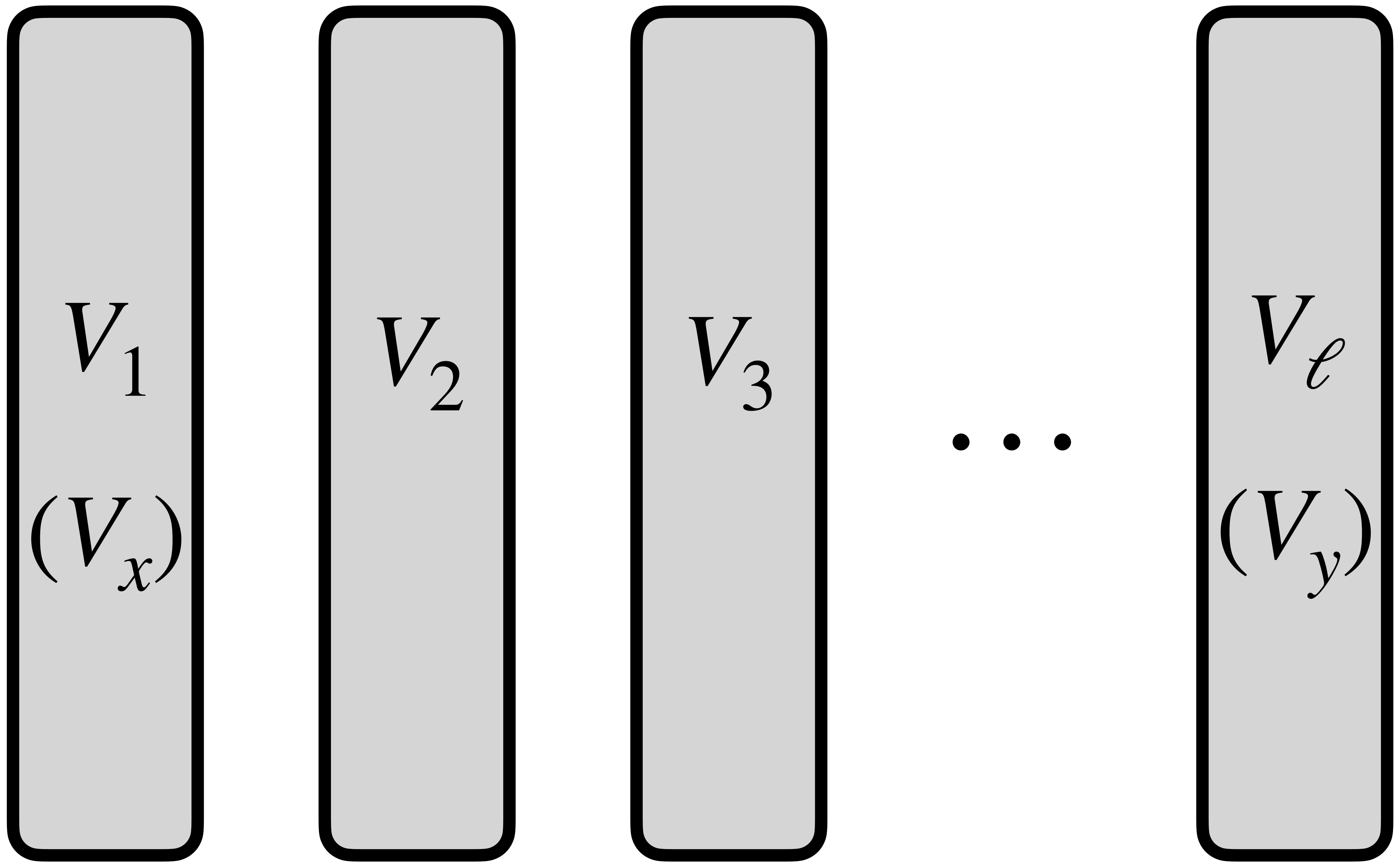


Message Complexity of MVC

- Any poly(n) round CONGEST algorithm requires $\tilde{\Omega}(n^2)$ messages.
- But we were aiming for $\Omega(m \cdot D)$ which can be as large as n^3.
- $G_{x,y}$ as described has constant diameter.
- Can we "stretch" the graph to get a higher lower bound?

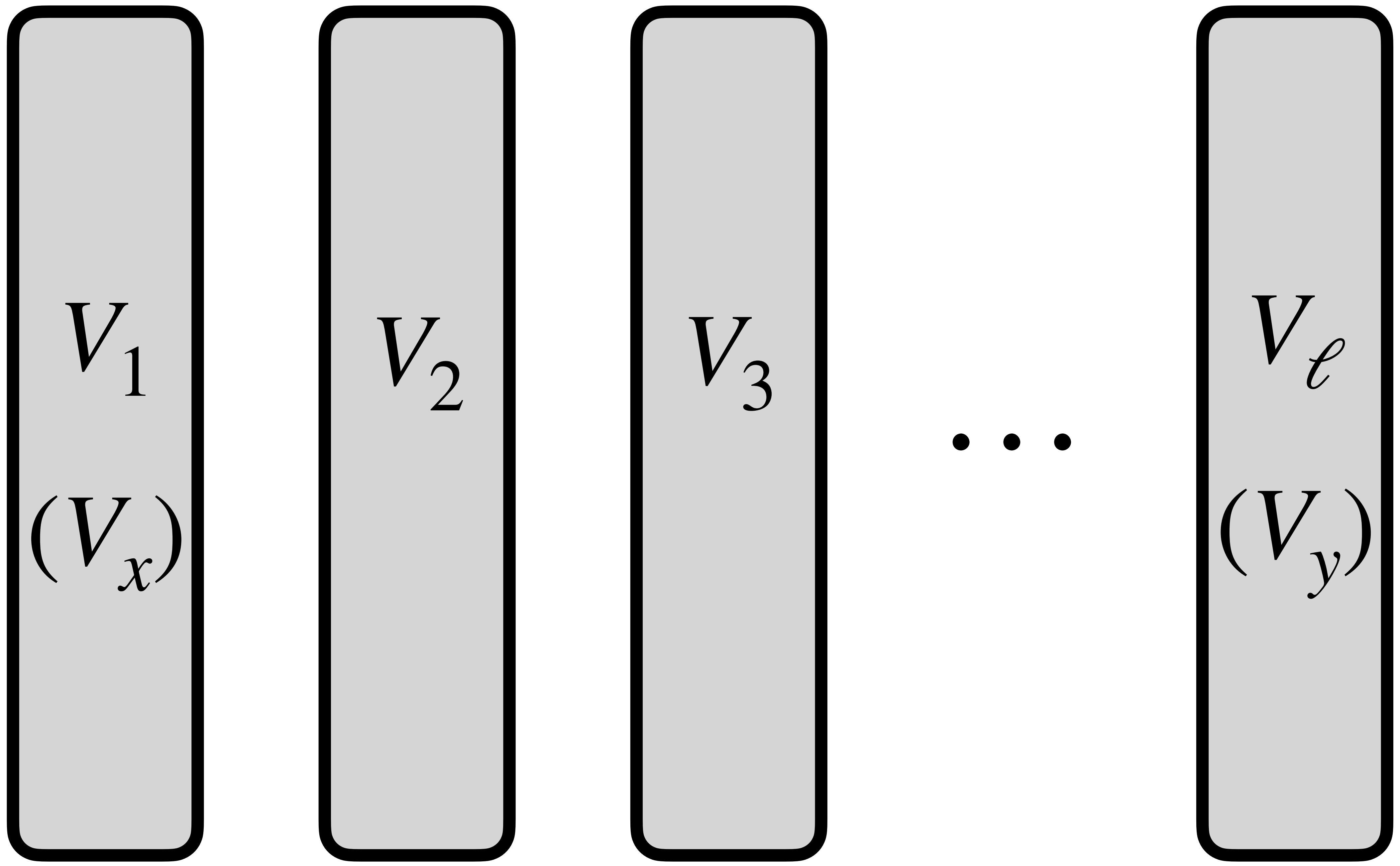


Message Complexity of MVC



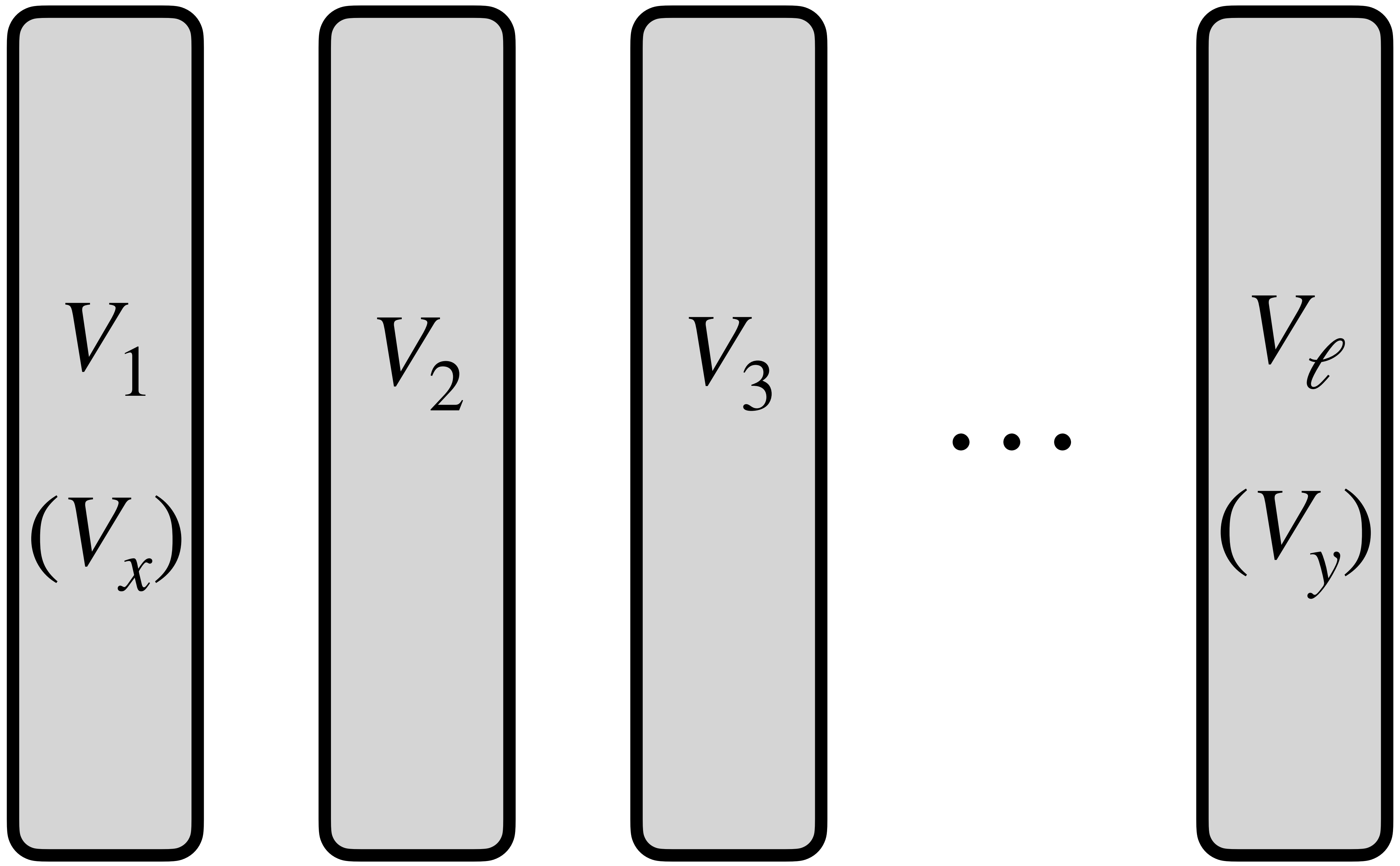
Message Complexity of MVC

- Instead of two parts, we have ℓ parts.



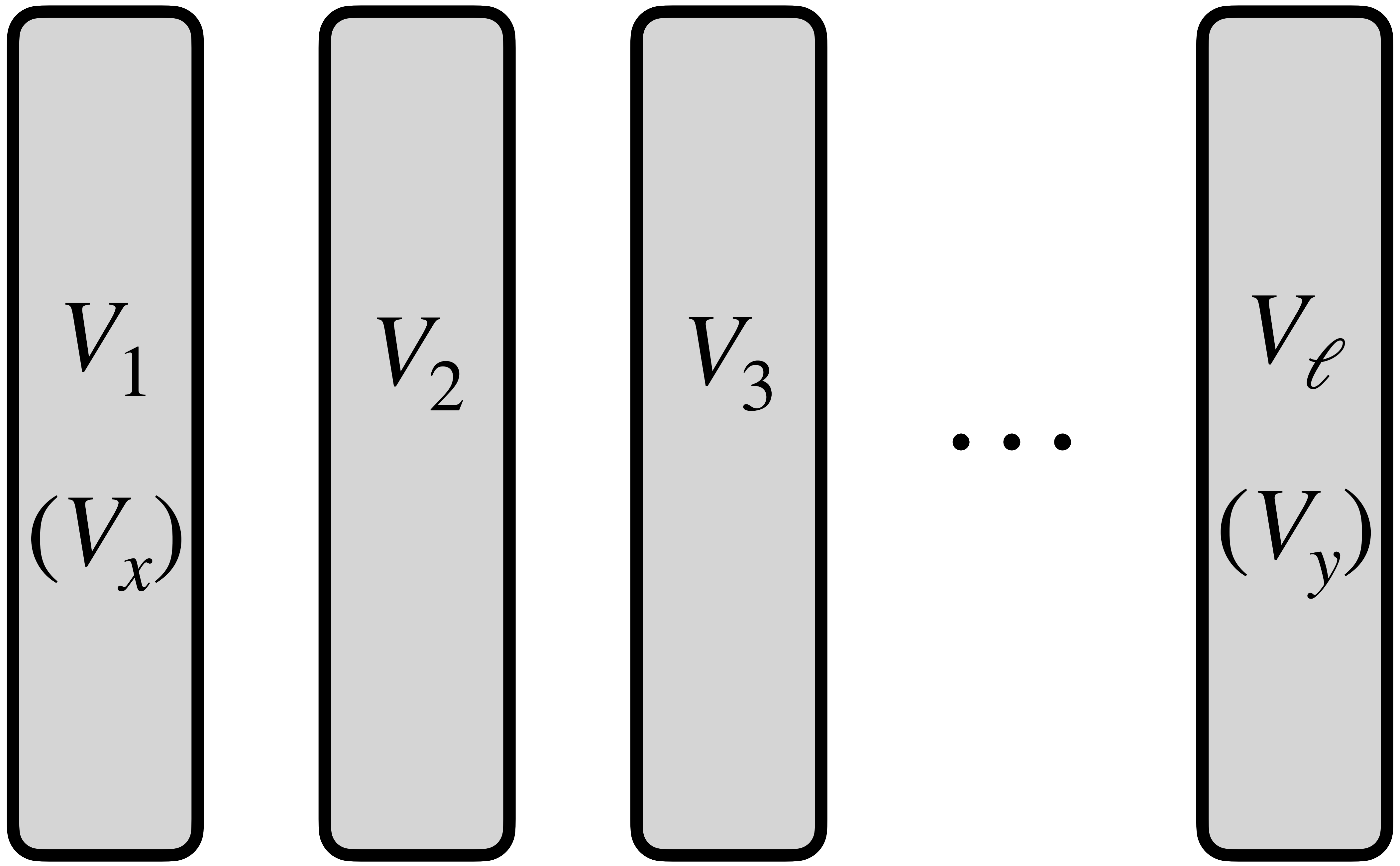
Message Complexity of MVC

- Instead of two parts, we have ℓ parts.
  - V_i only has edges to V_{i-1}, V_i, V_{i+1}.



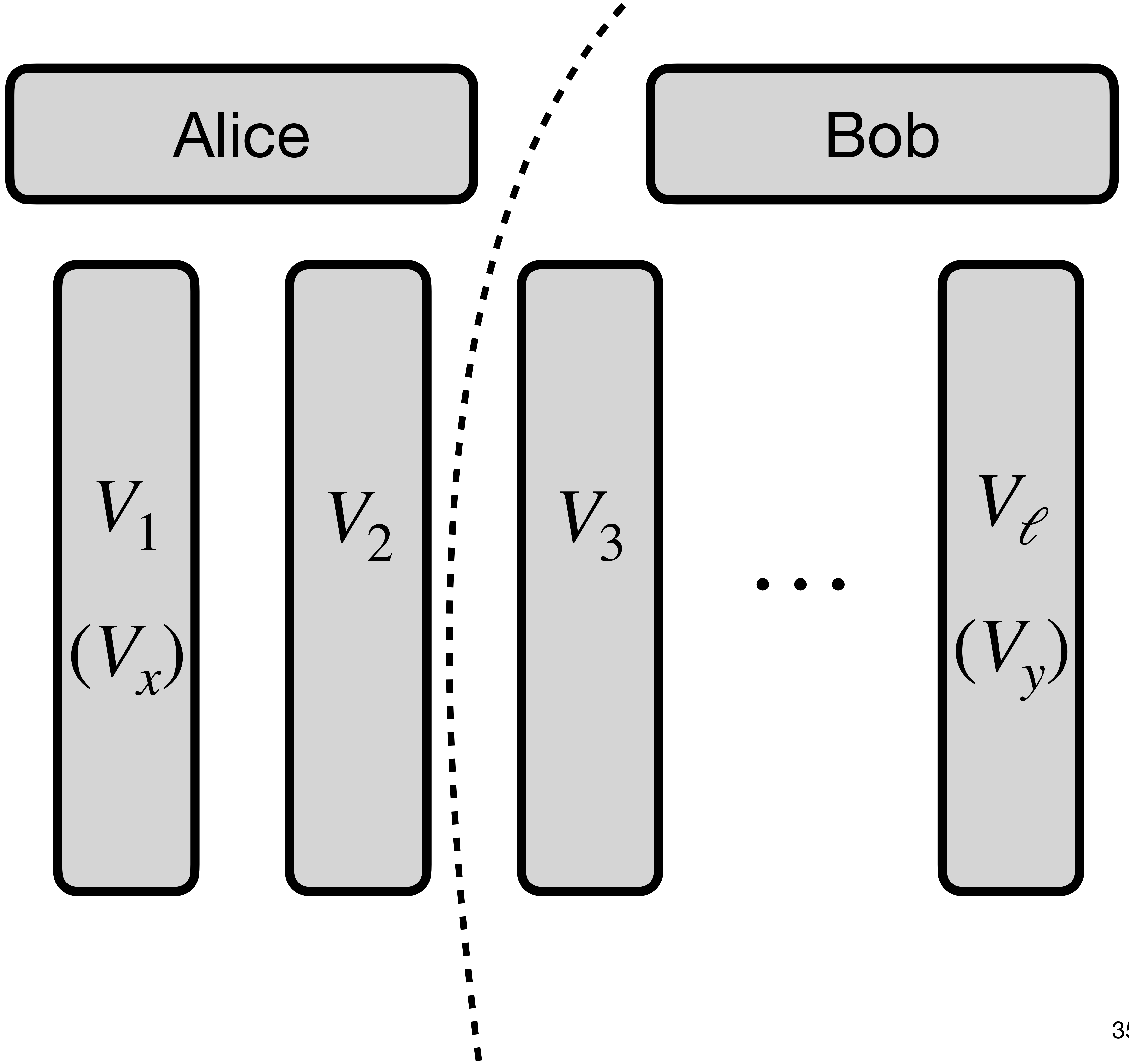
Message Complexity of MVC

- Instead of two parts, we have ℓ parts.
  - V_i only has edges to V_{i-1}, V_i, V_{i+1}.
  - CONGEST algorithm better send many messages across all cuts…



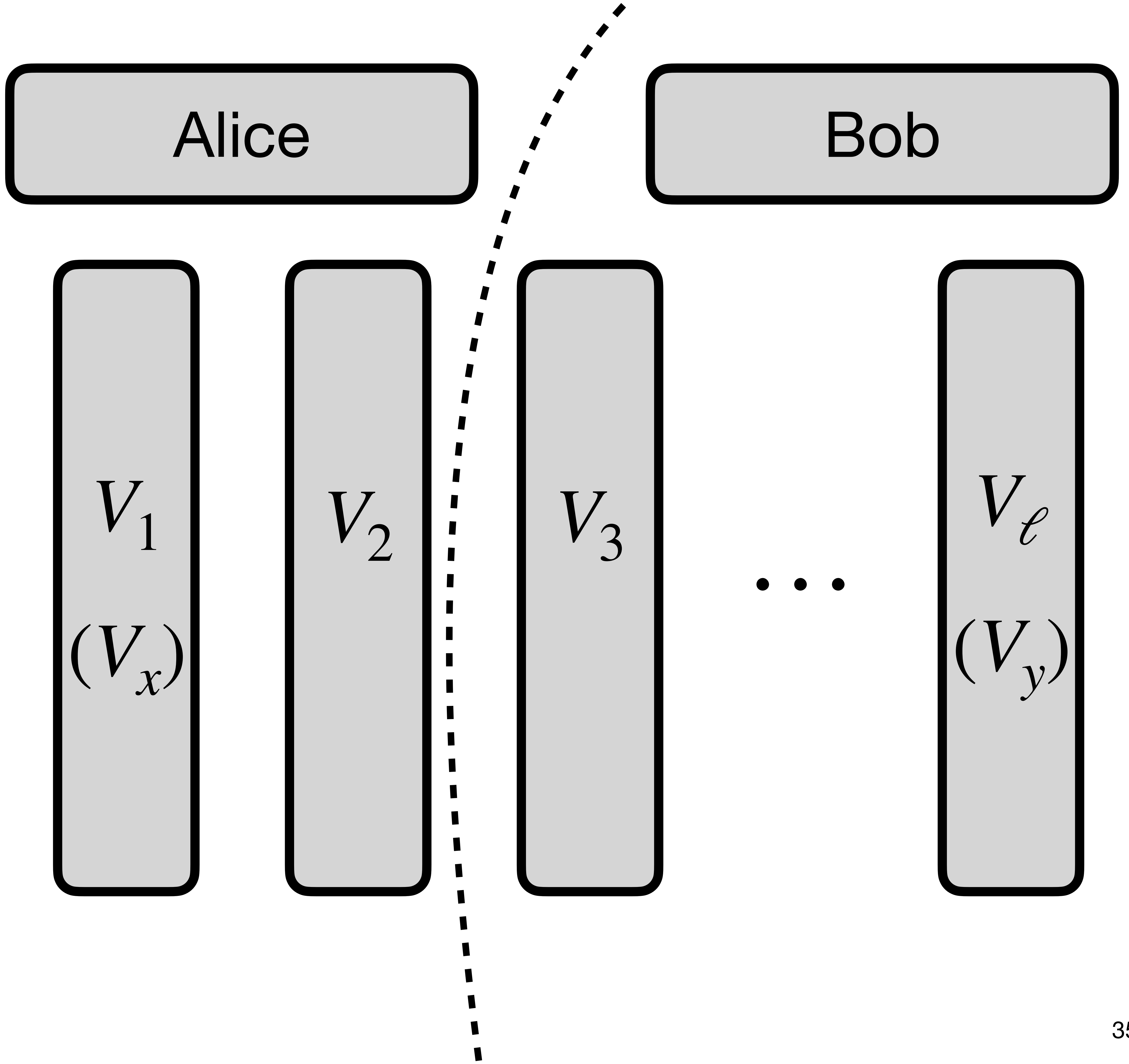
Message Complexity of MVC

- Instead of two parts, we have ℓ parts.
  - V_i only has edges to V_{i-1}, V_i, V_{i+1}.
  - CONGEST algorithm better send many messages across all cuts…

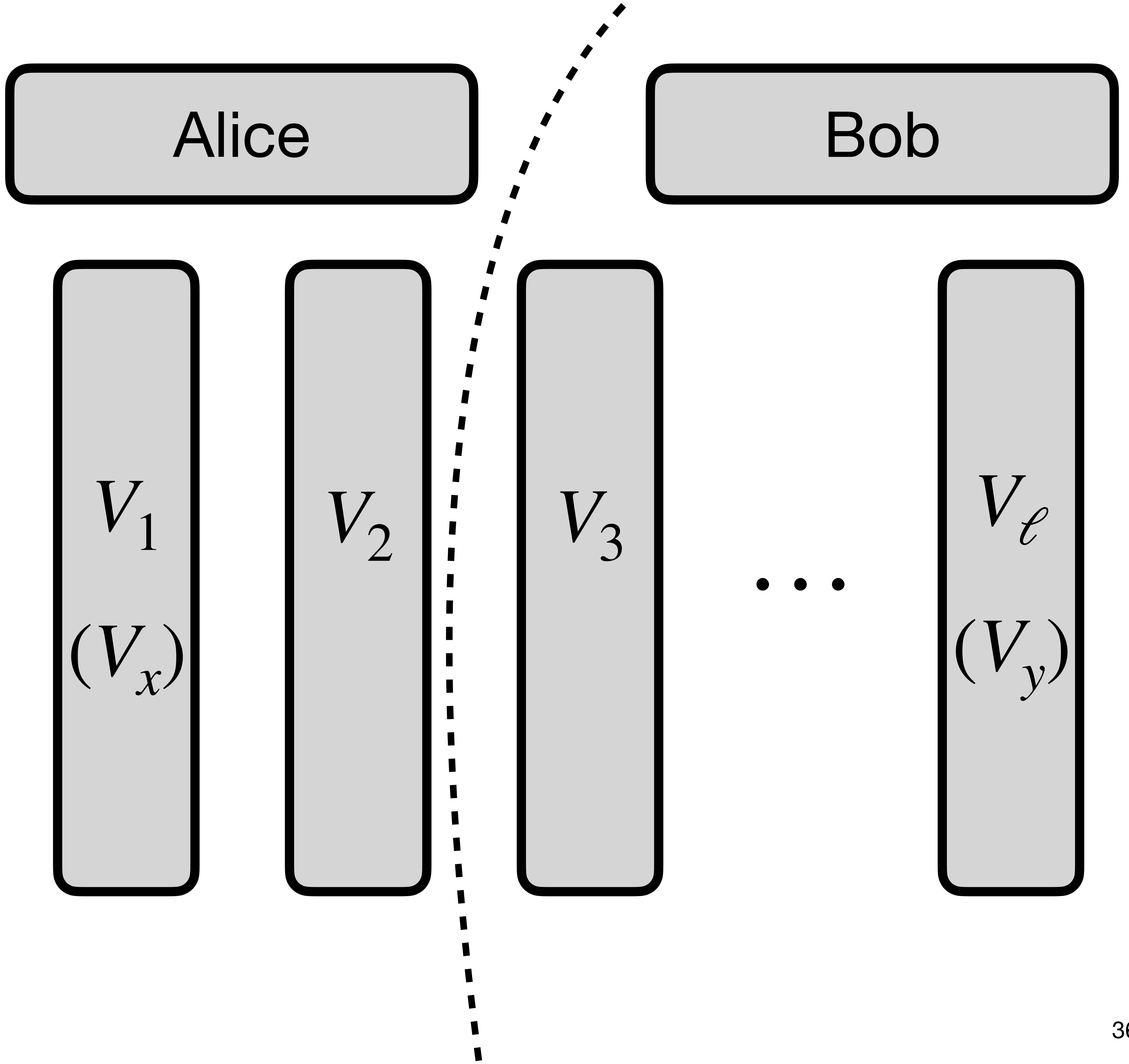


Message Complexity of MVC

- Instead of two parts, we have ℓ parts.
  - V_i only has edges to V_{i-1}, V_i, V_{i+1}.
  - CONGEST algorithm better send many messages across all cuts…
- Message complexity will be increased by an ℓ factor!

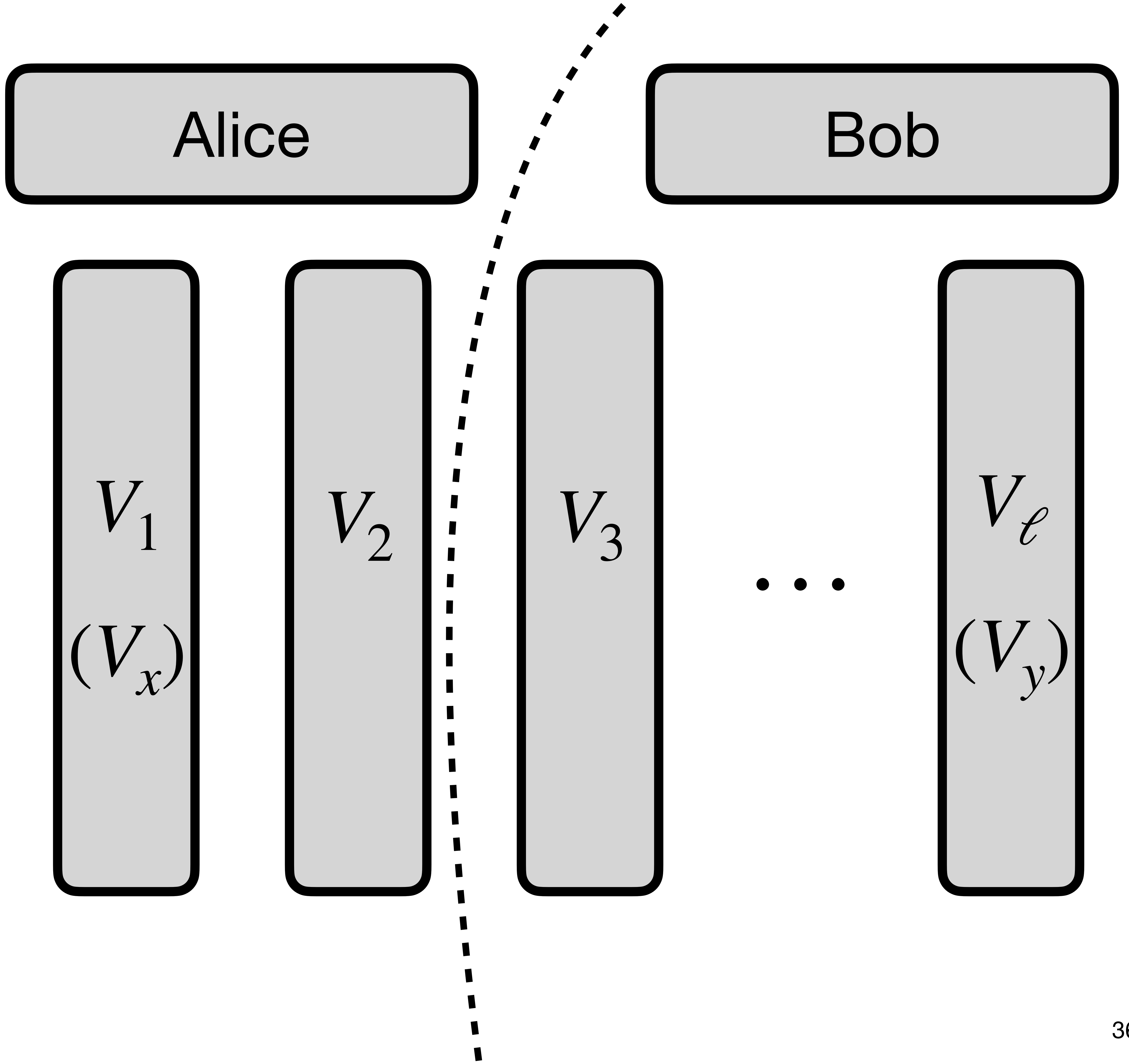


Message Complexity of MVC



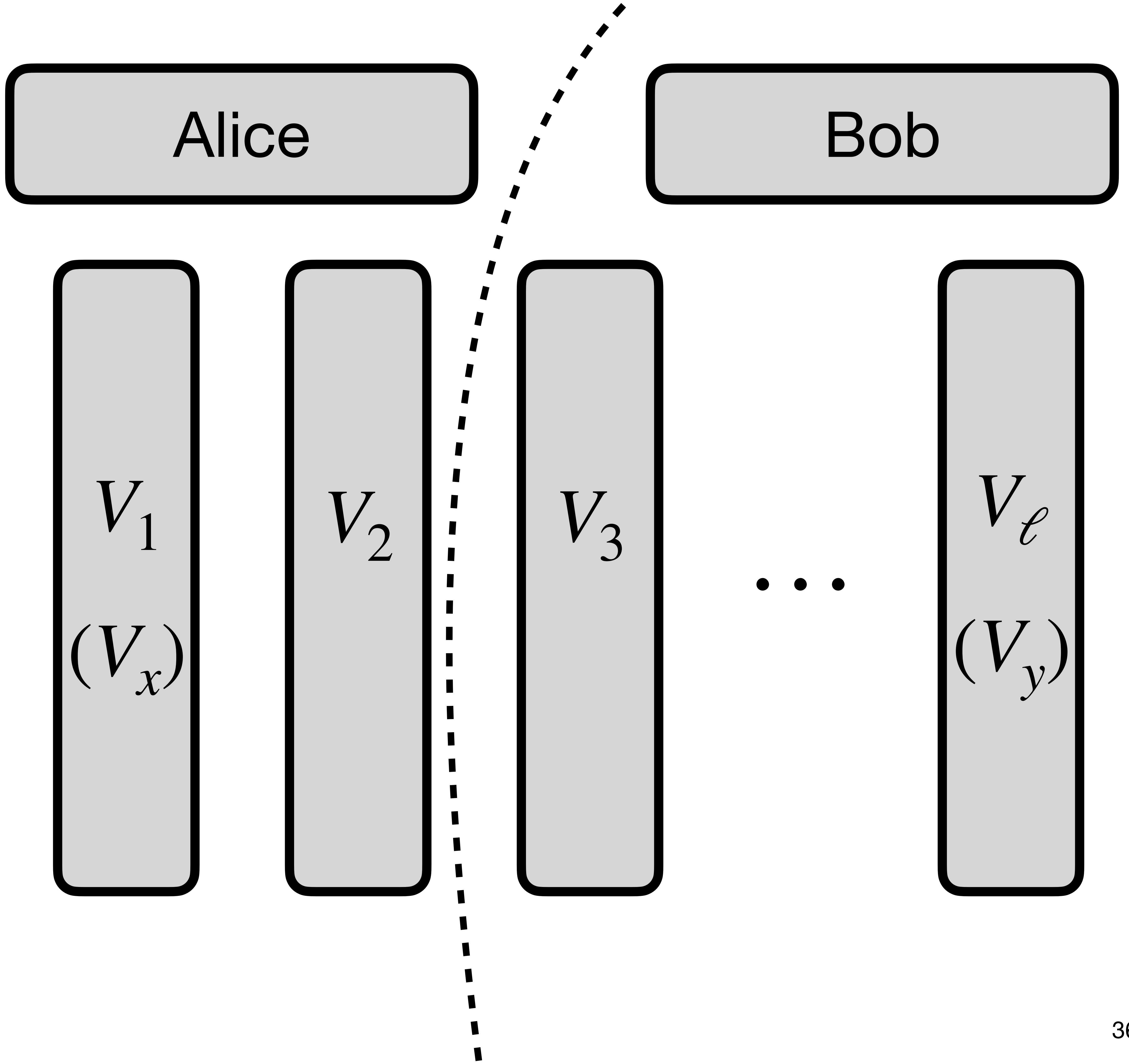
Message Complexity of MVC

- How do Alice and Bob determine this "low message" cut before simulating the CONGEST algorithm?



Message Complexity of MVC

- How do Alice and Bob determine this “low message” cut before simulating the CONGEST algorithm?
- If the CONGEST algorithm is message efficient, a constant fraction of the cuts will be “low message”.

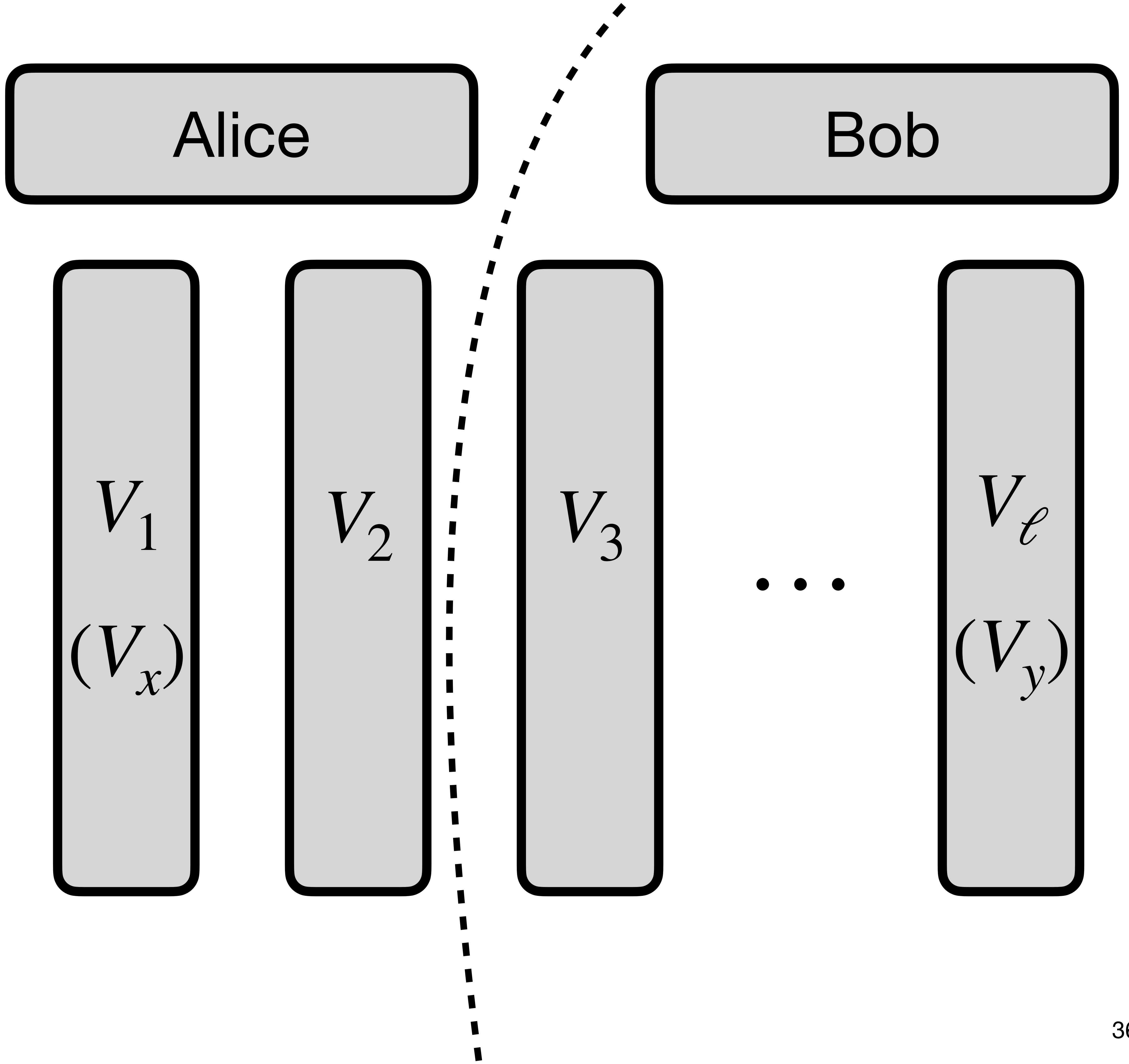


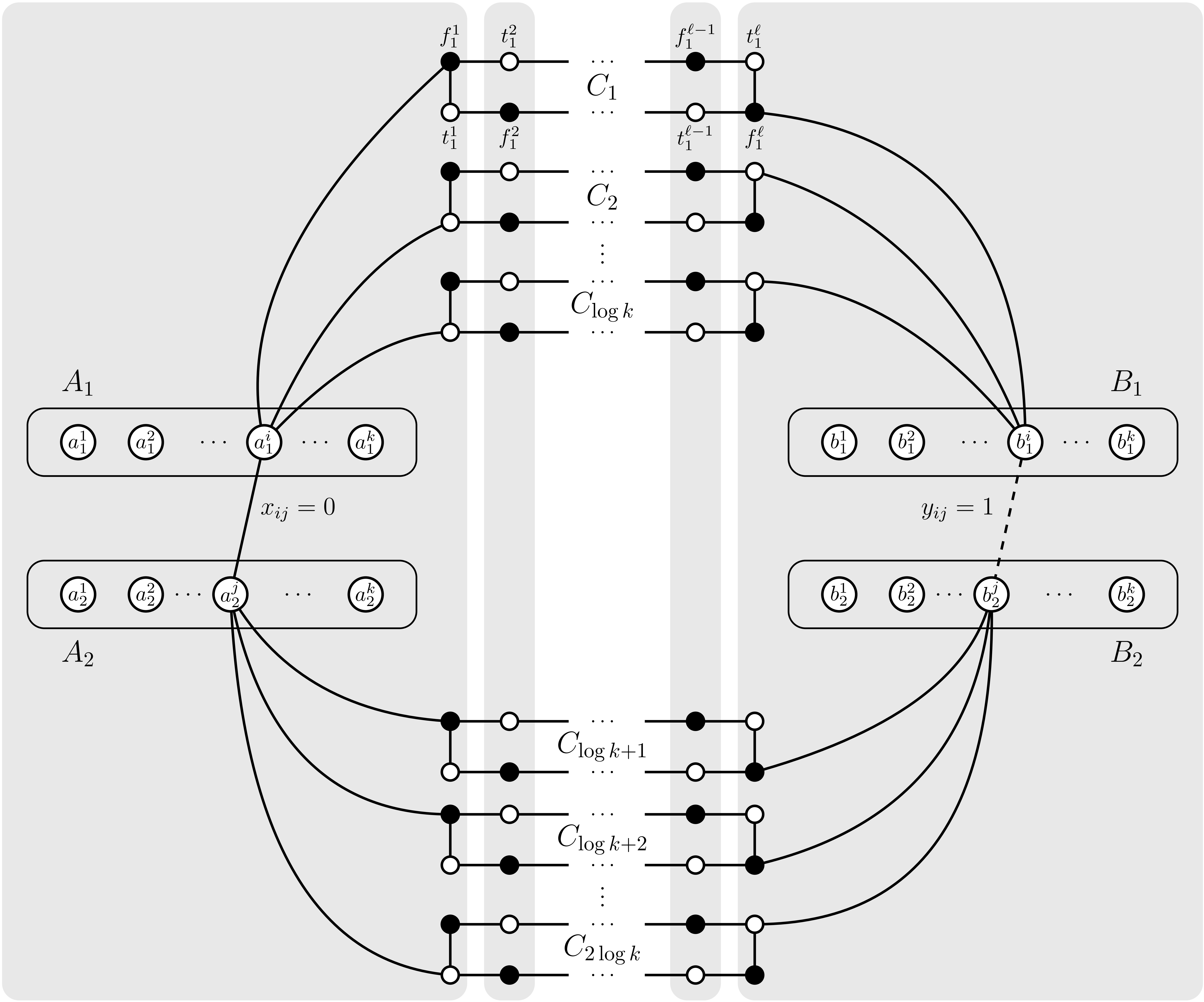
Message Complexity of MVC

- How do Alice and Bob determine this "low message" cut before simulating the CONGEST algorithm?

- If the CONGEST algorithm is message efficient, a constant fraction of the cuts will be "low message".

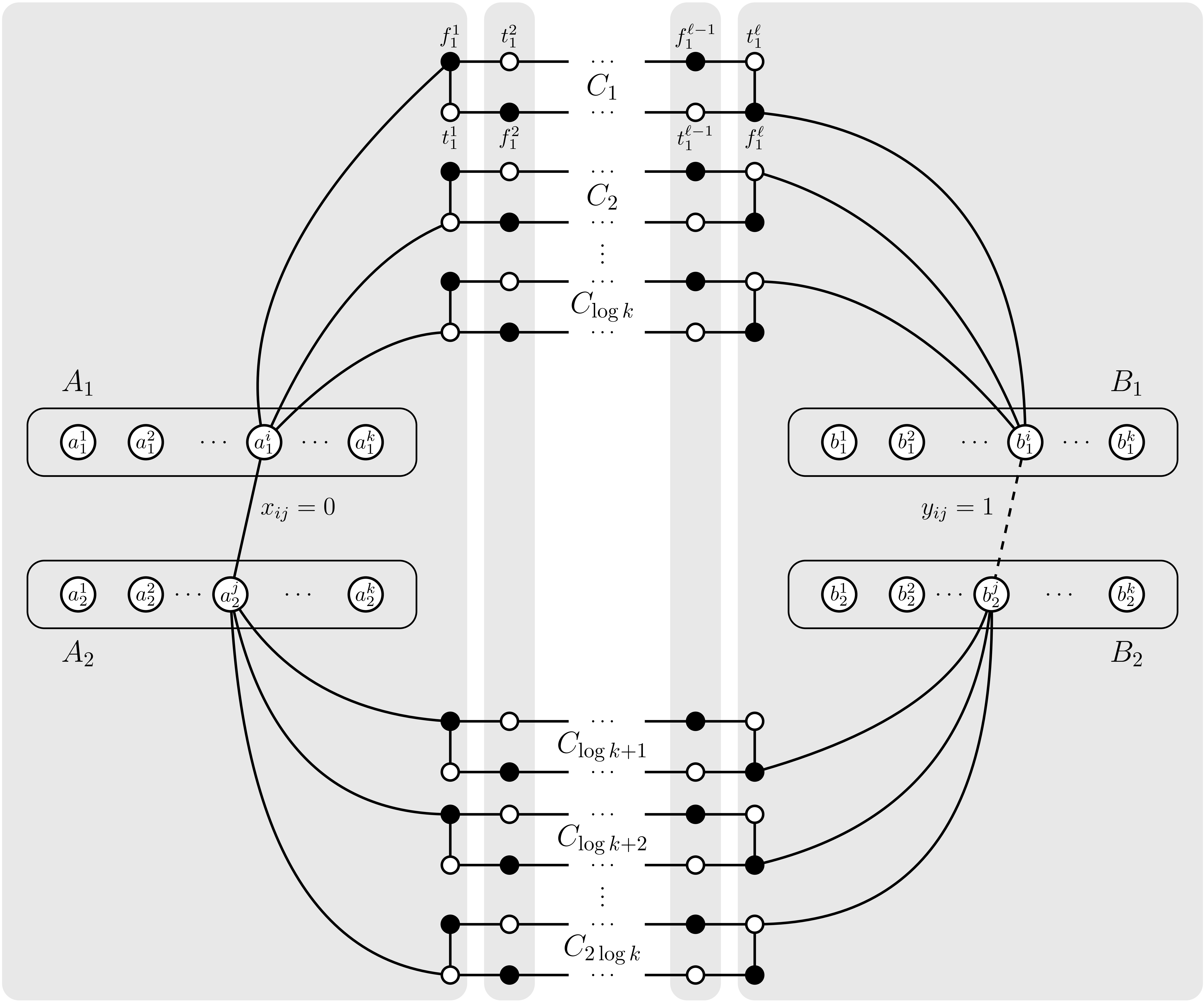
- Picking one at random gives constant error probability!





[DPP+24] Dufoulon, Pai, Pandurangan, Pemmaraju, Robinson. ITCS 2024

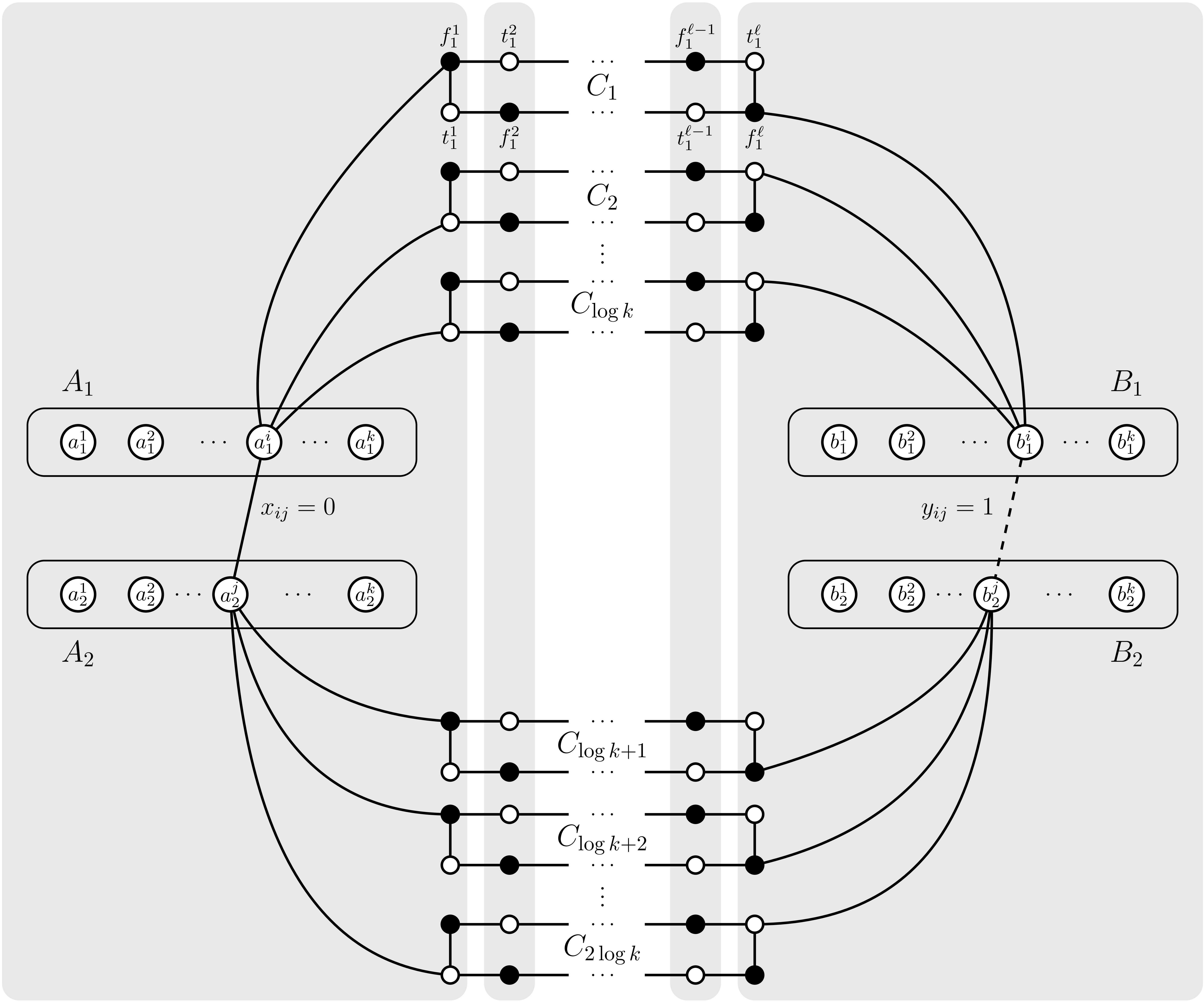
$\mathrm{SD}(x, y) =$ True iff $G_{x,y}$ has an MVC of size at least $C(k, \ell)$.



[DPP+24] Dufoulon, Pai, Pandurangan, Pemmaraju, Robinson. ITCS 2024

SD(x, y) = True iff $G_{x,y}$ has an MVC of size at least $C(k, \ell)$.

Computing MVC of $G_{x,y}$ in r rounds requires

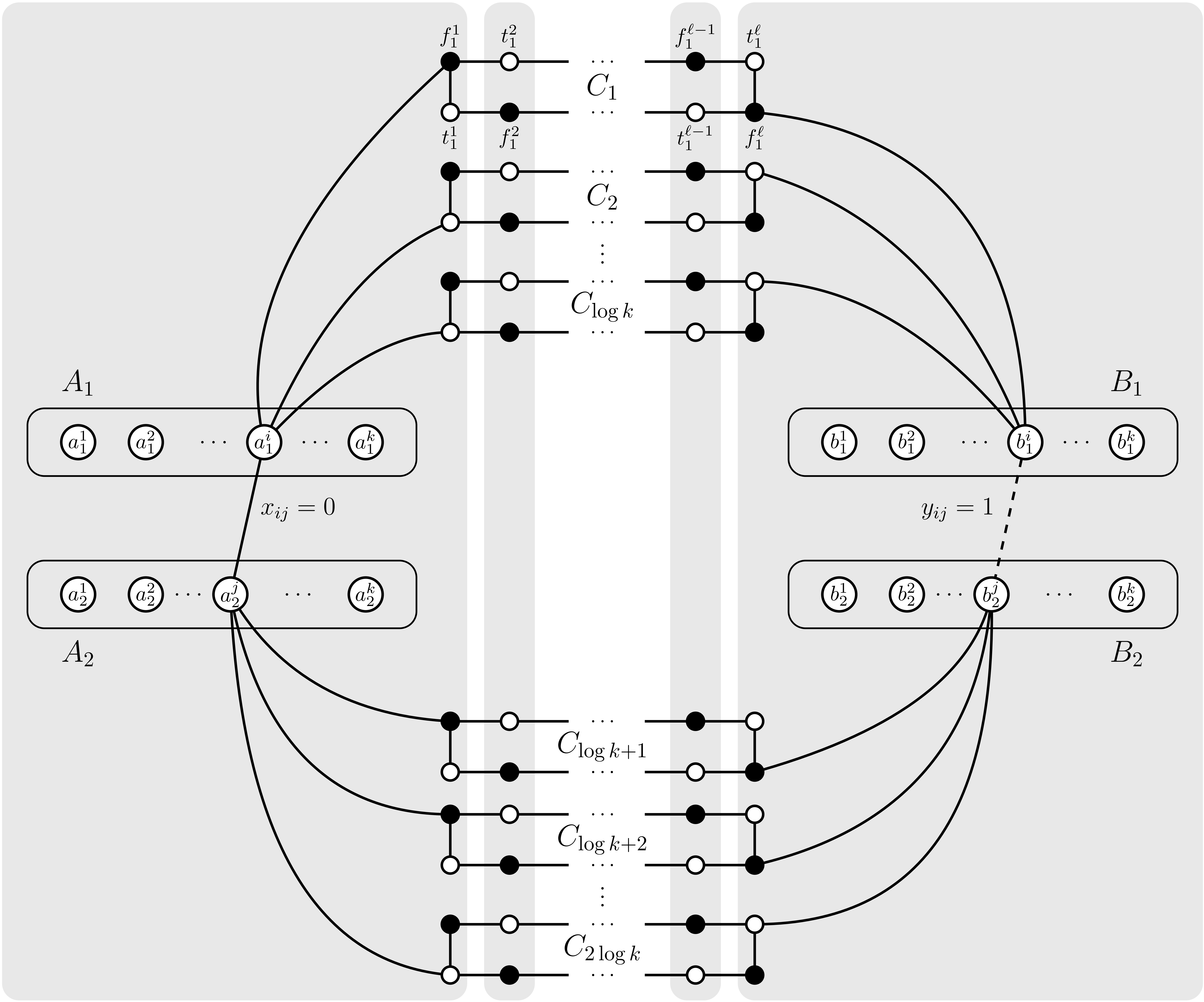


[DPP+24] Dufoulon, Pai, Pandurangan, Pemmaraju, Robinson. ITCS 2024

SD(x, y) = True iff $G_{x,y}$ has an MVC of size at least $C(k, \ell)$.

Computing MVC of $G_{x,y}$ in r rounds requires

$$\tilde{\Omega}\left(\frac{\ell \cdot k^2}{1 + \log r}\right)$$



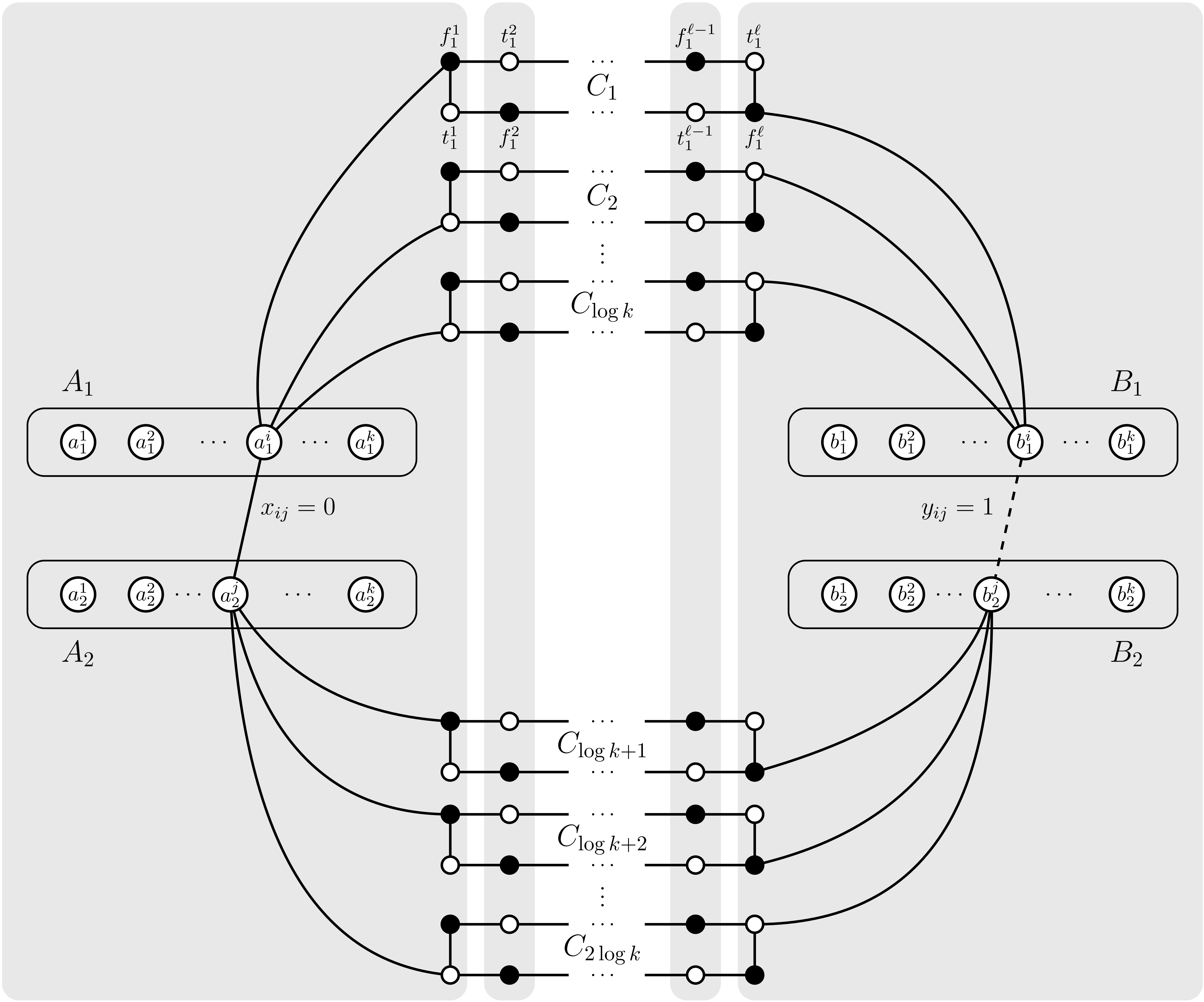
[DPP+24] Dufoulon, Pai, Pandurangan, Pemmaraju, Robinson. ITCS 2024

$\mathrm{SD}(x, y) =$ True iff $G_{x,y}$ has an MVC of size at least $C(k, \ell)$.

Computing MVC of $G_{x,y}$ in r rounds requires

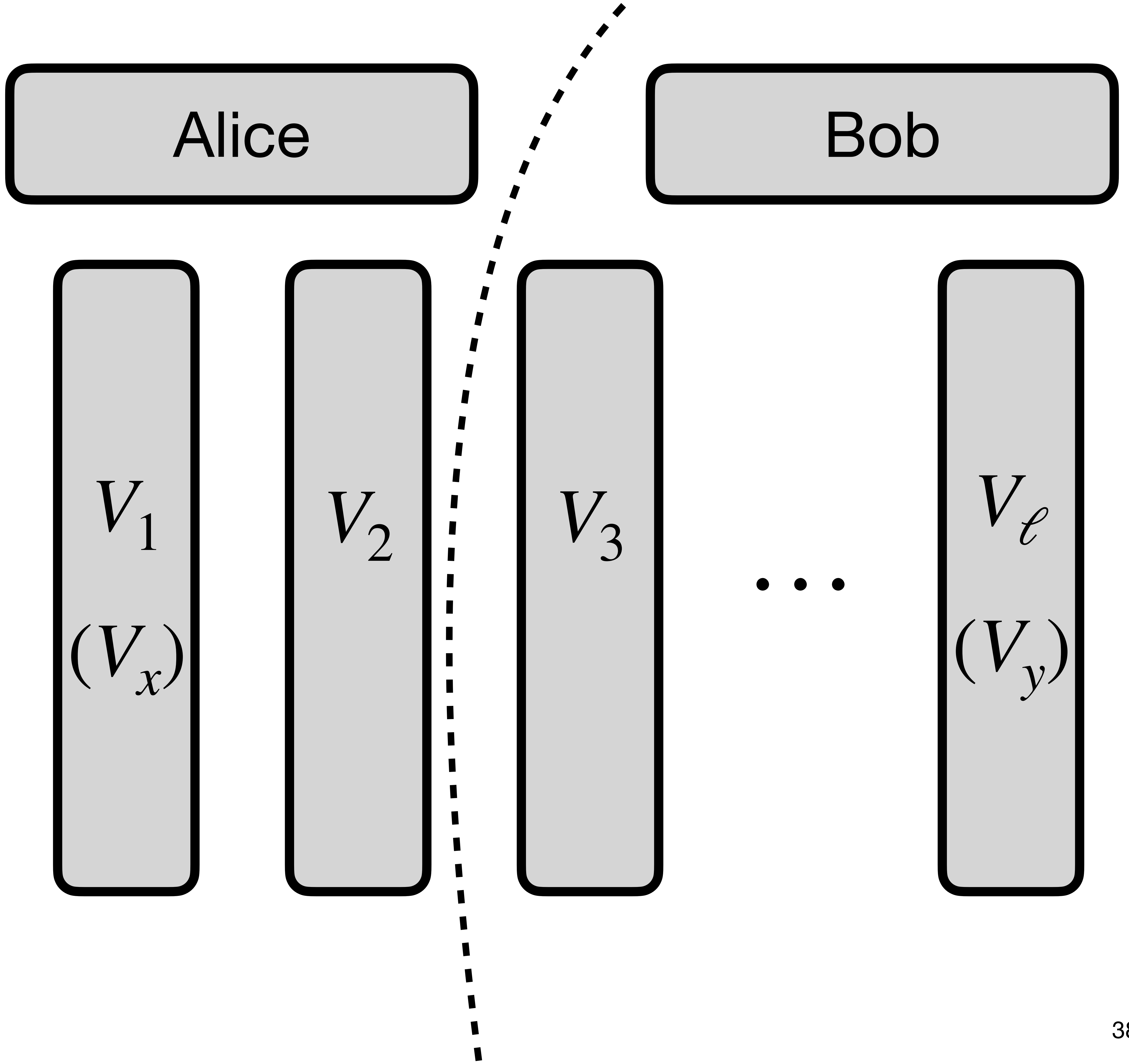
$$\tilde{\Omega}\left(\frac{\ell \cdot k^2}{1 + \log r}\right)$$

messages.



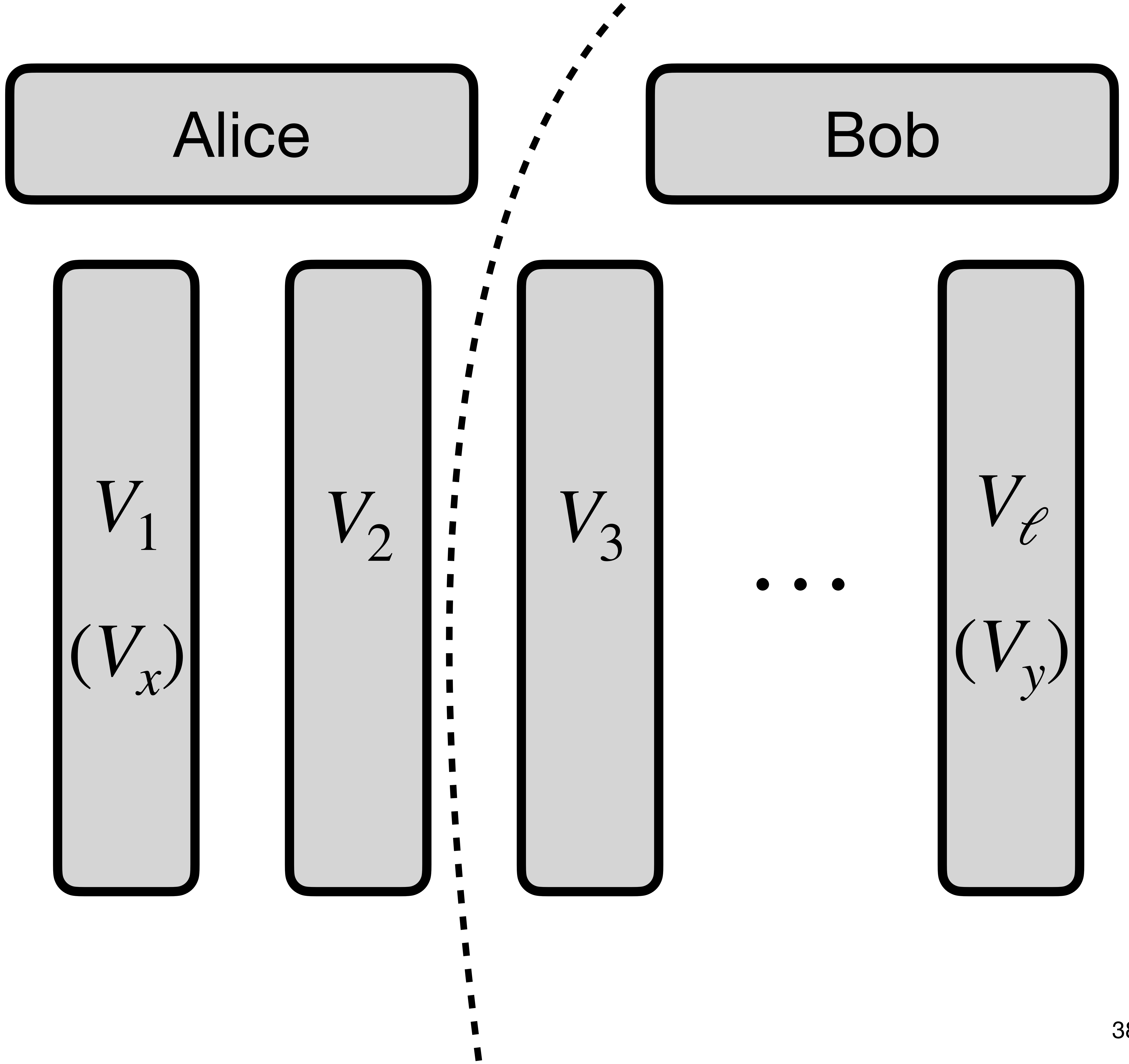
[DPP+24] Dufoulon, Pai, Pandurangan, Pemmaraju, Robinson. ITCS 2024

Message Complexity of MVC



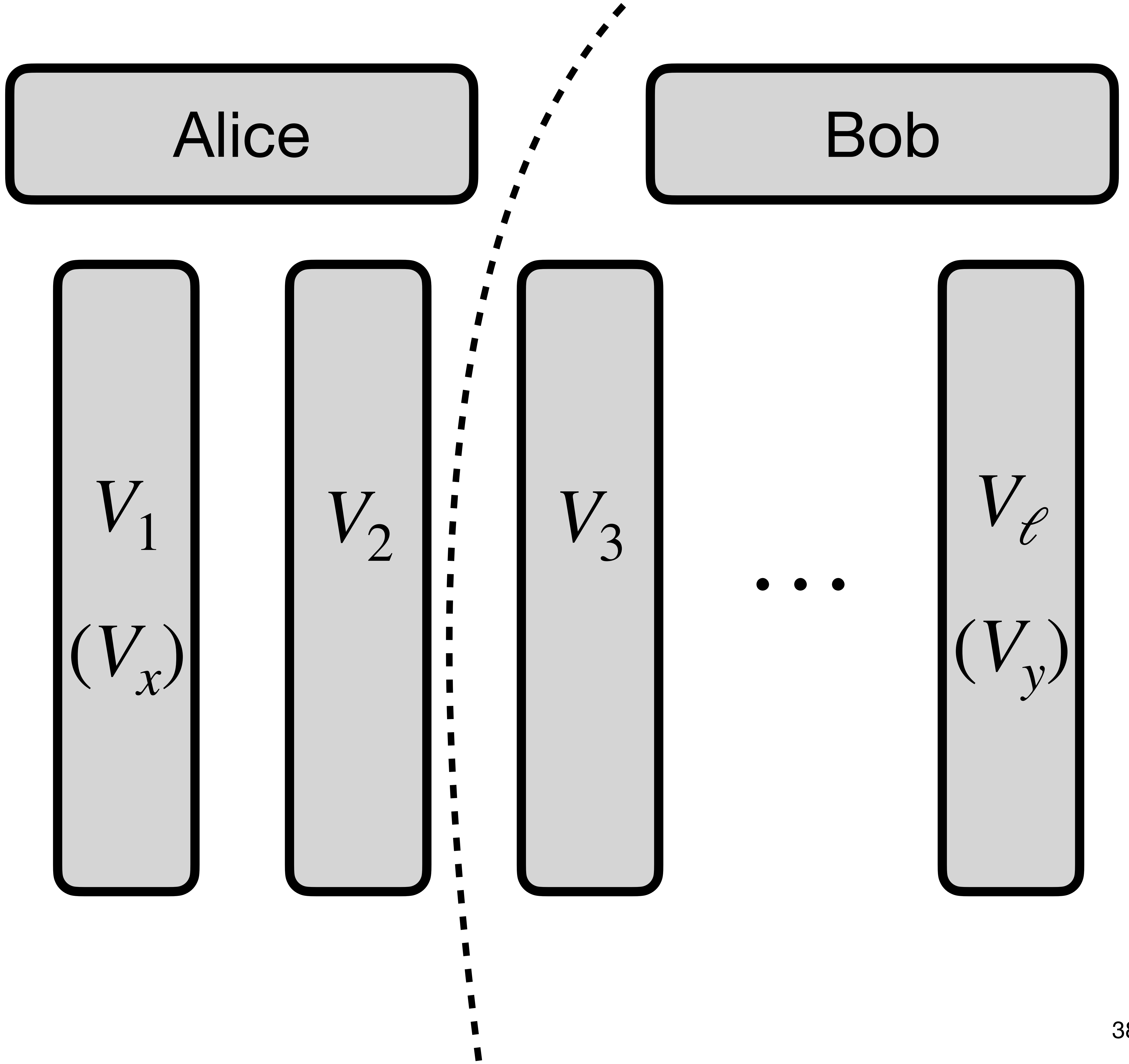
Message Complexity of MVC

- Alice and Bob first construct $G_{x,y}$



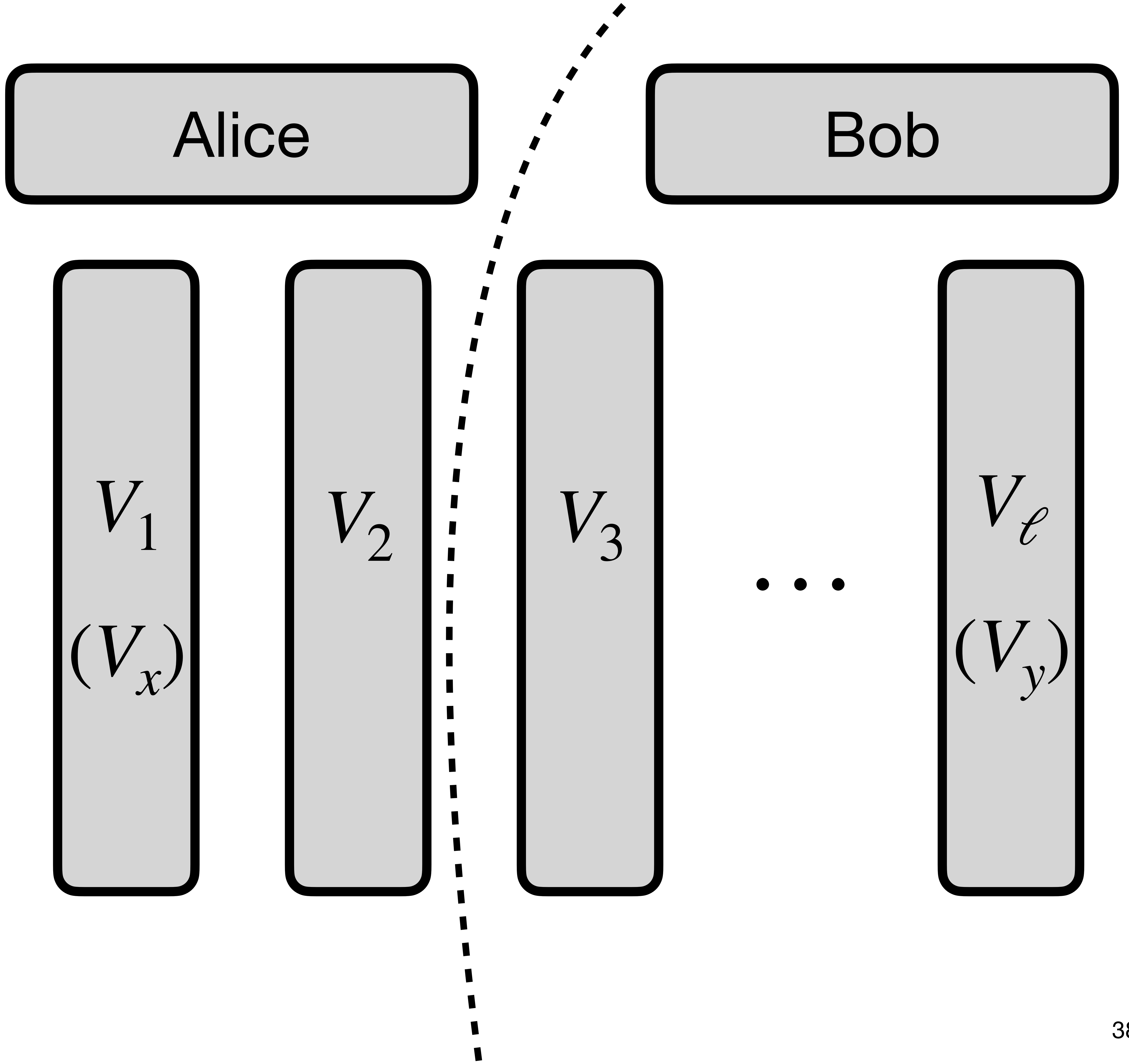
Message Complexity of MVC

- Alice and Bob first construct $G_{x,y}$
- Then they pick an $i \in \{1,\ldots,\ell-1\}$ uniformly at random.



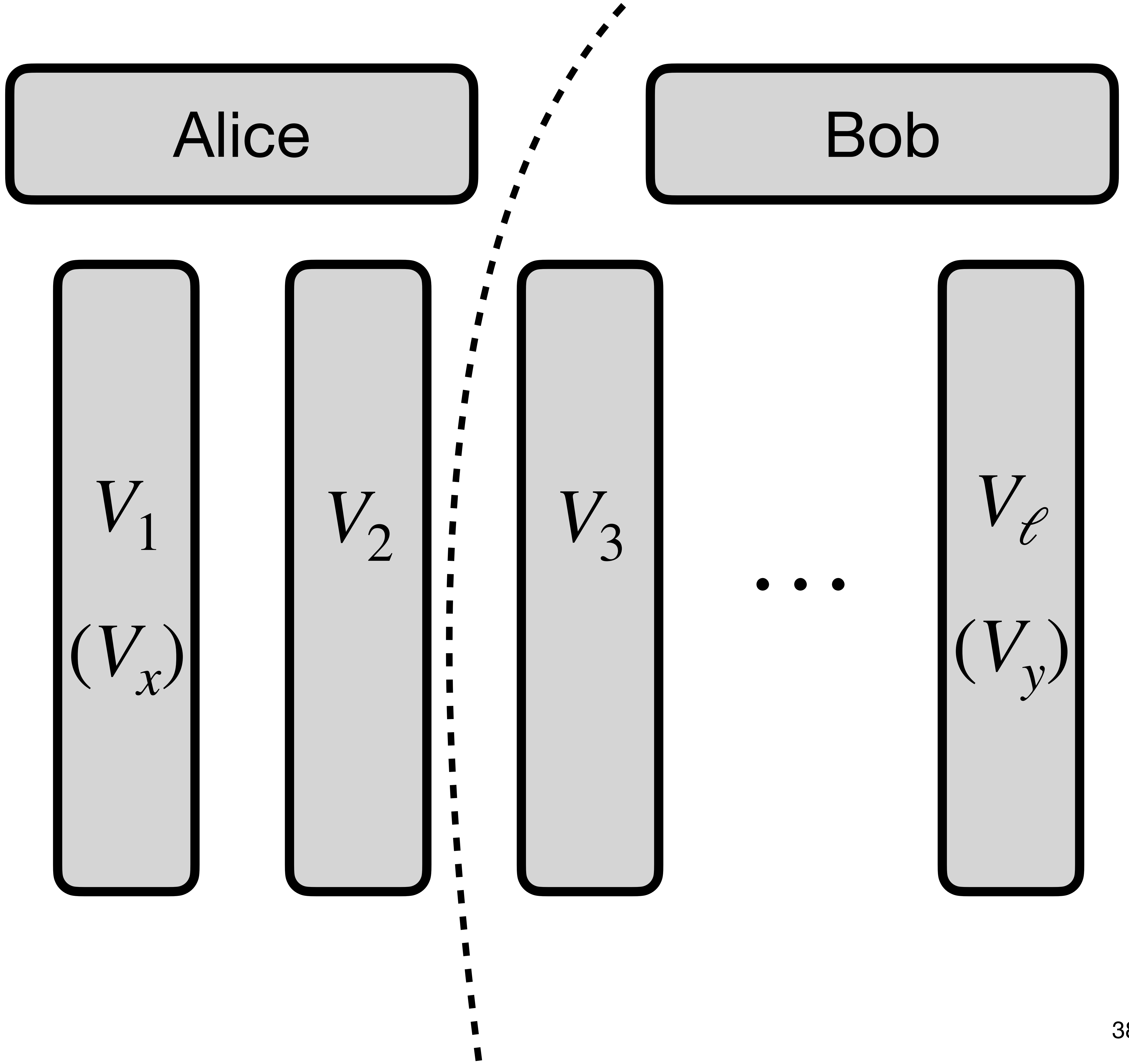
Message Complexity of MVC

- Alice and Bob first construct $G_{x,y}$
- Then they pick an $i \in \{1,\ldots,\ell - 1\}$ uniformly at random.
- Simulate r-round CONGEST algorithm with cut between V_i and V_{i+1}.

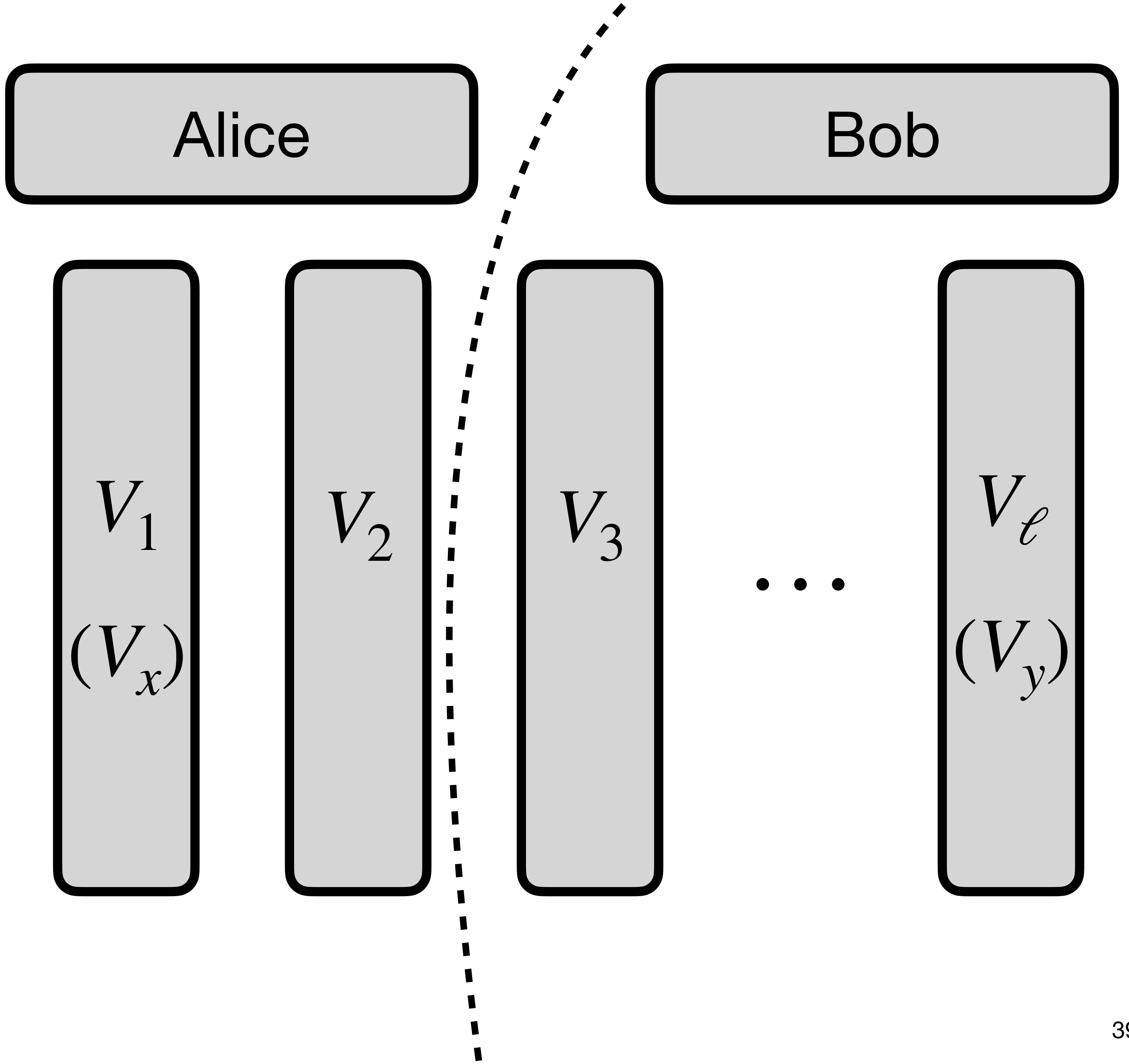


Message Complexity of MVC

- Alice and Bob first construct $G_{x,y}$
- Then they pick an $i \in \{1,\ldots,\ell-1\}$ uniformly at random.
- Simulate r-round CONGEST algorithm with cut between V_i and V_{i+1}.
    - Use synchronous clock for efficient simulation.

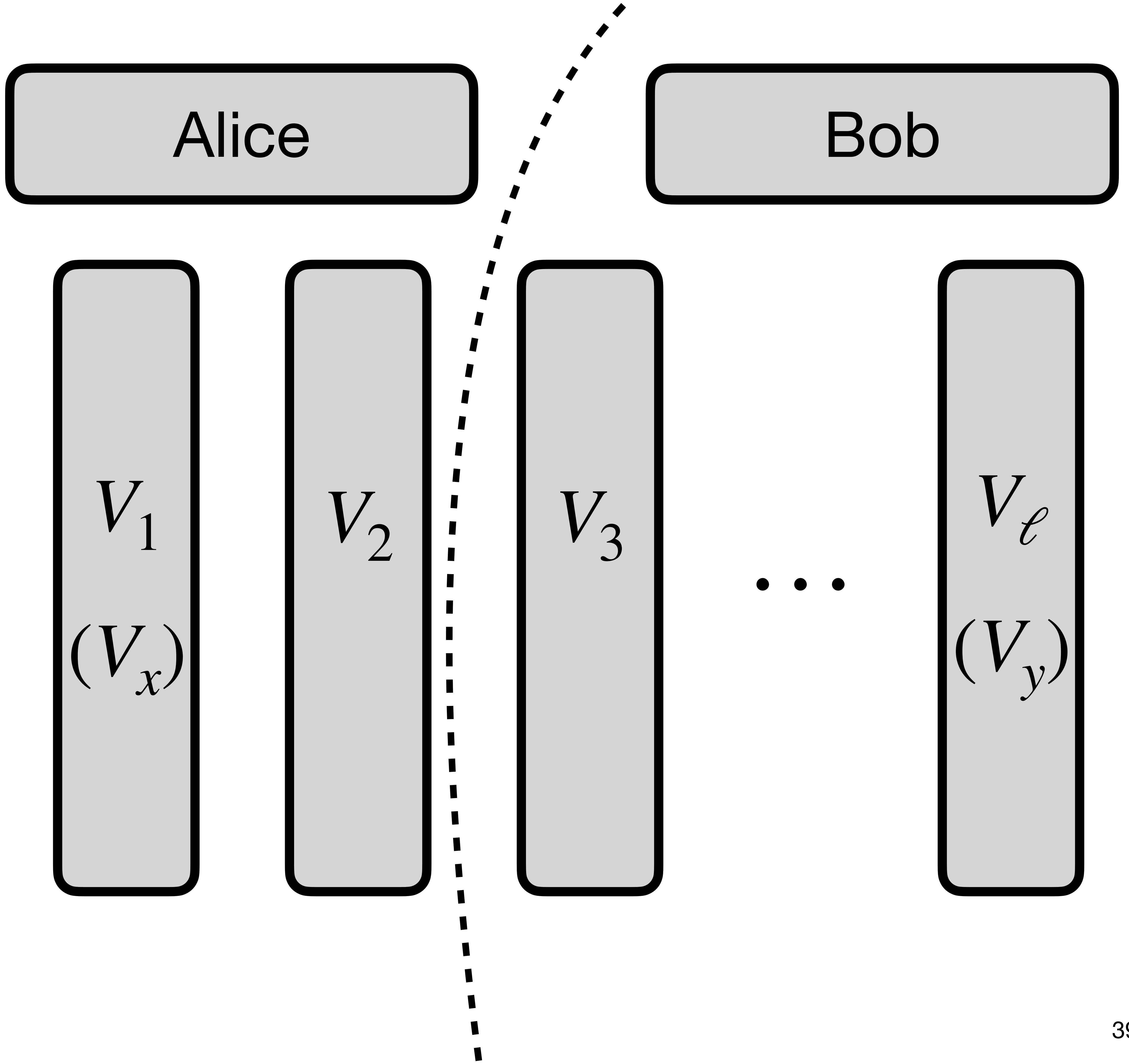


Message Complexity of MVC



Message Complexity of MVC

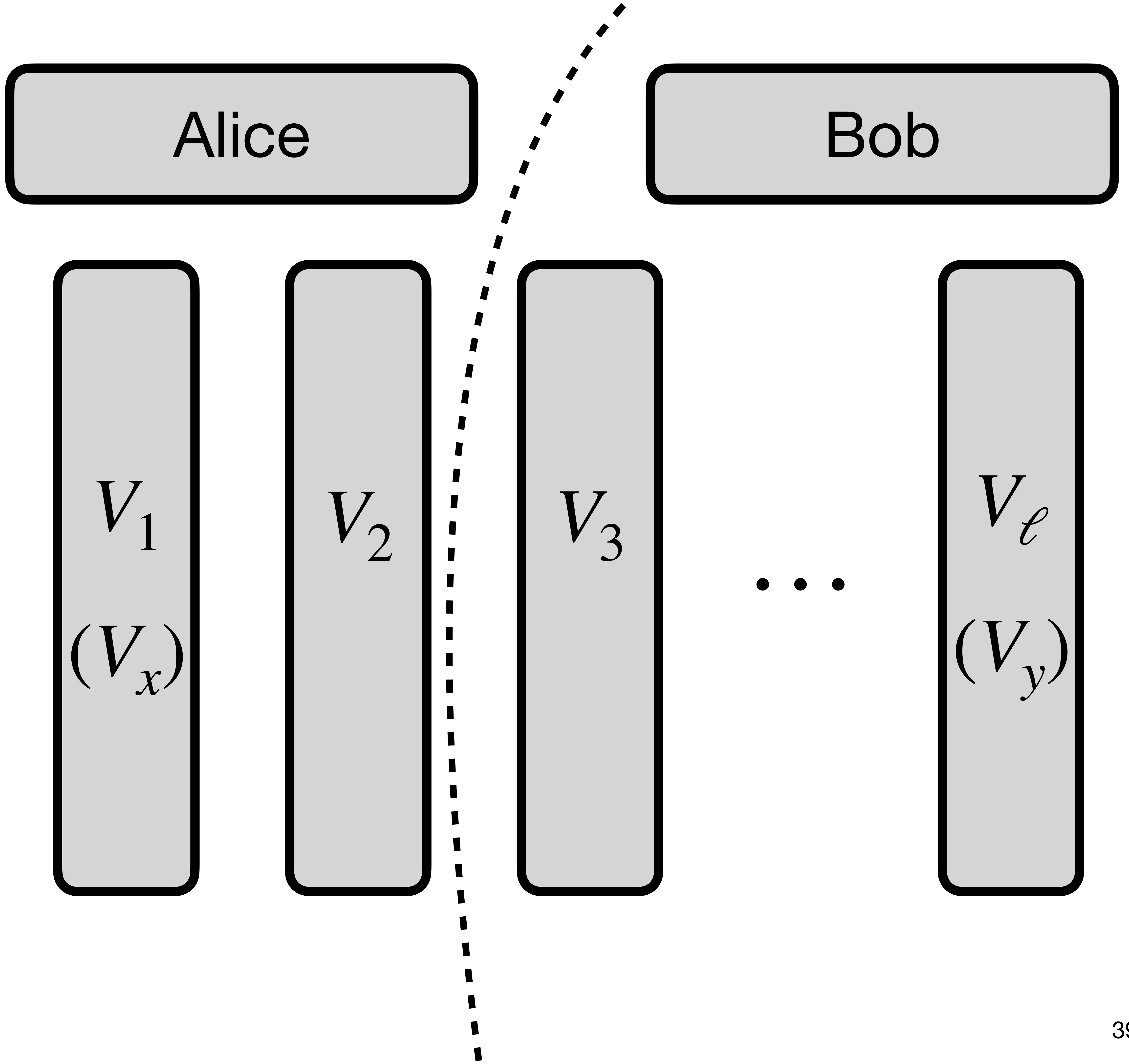
- By the randomized SD lower bound Alice and Bob must exchange



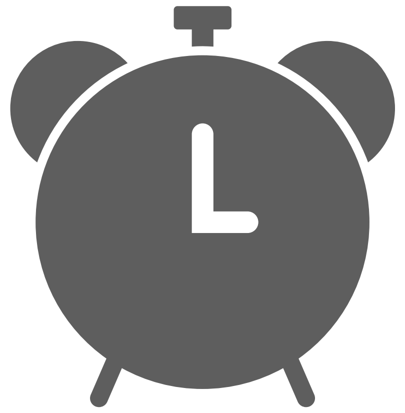
Message Complexity of MVC

- By the randomized SD lower bound Alice and Bob must exchange

$$\tilde{\Omega}\left(\frac{k^2}{1+\log r}\right)$$ bits across the cut.



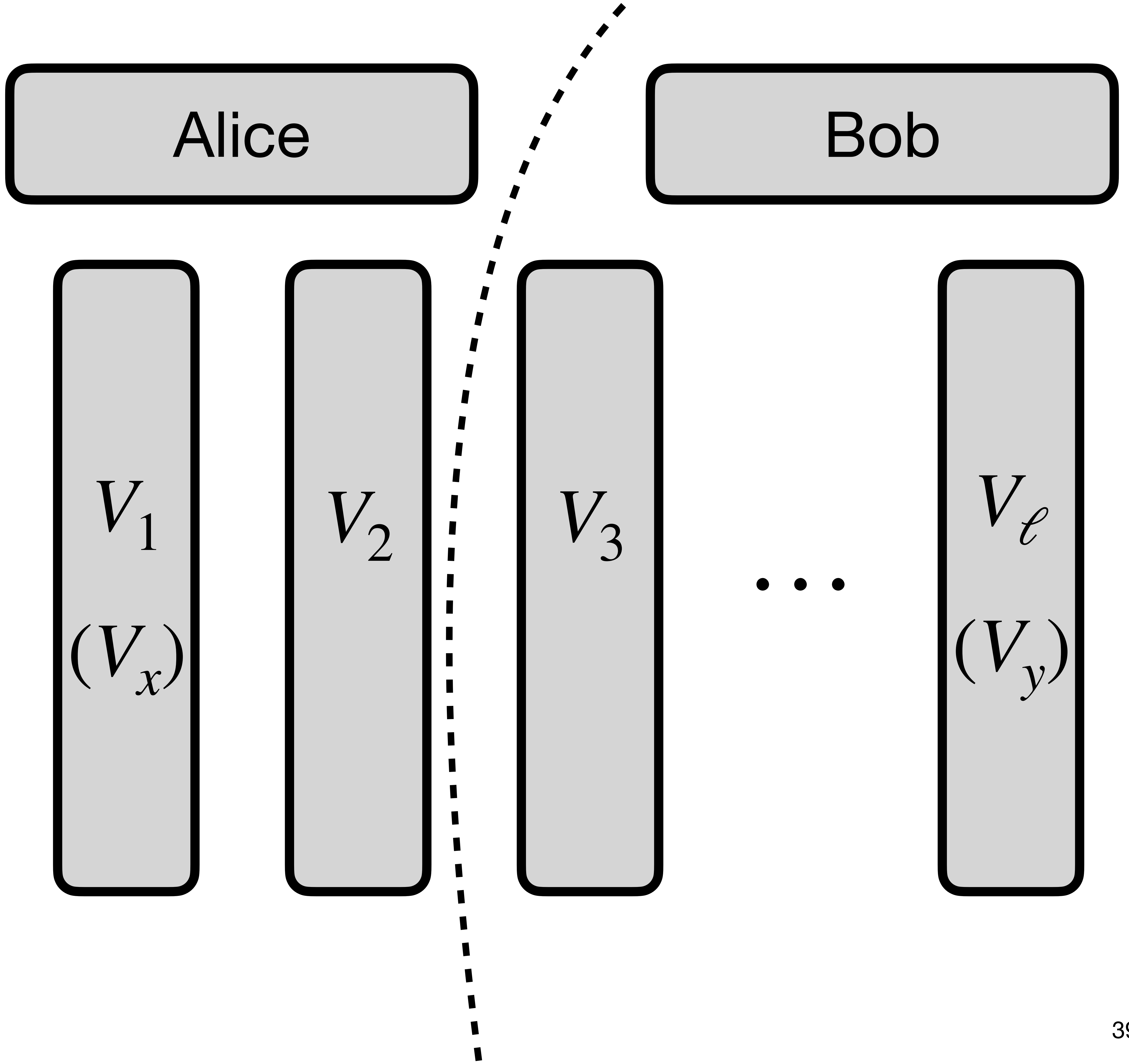
Message Complexity of MVC



- By the randomized SD lower bound Alice and Bob must exchange

$$\tilde{\Omega}\left(\frac{k^2}{1+\log r}\right) \text{ bits across the cut.}$$

- This must be true for a constant fraction of the ℓ cuts.



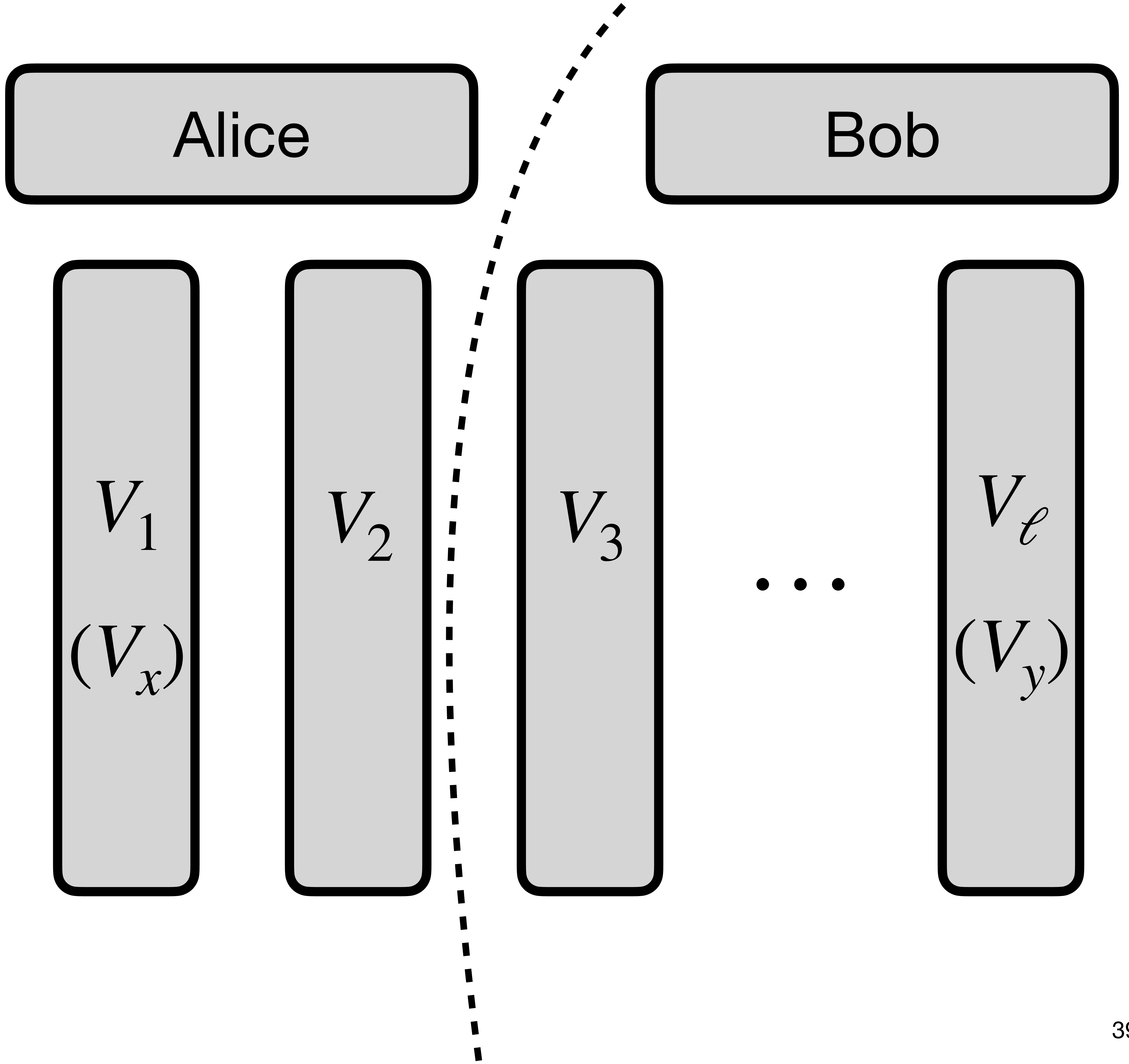
Message Complexity of MVC

- By the randomized SD lower bound Alice and Bob must exchange

  $$\tilde{\Omega}\left(\frac{k^2}{1+\log r}\right)$$ bits across the cut.

- This must be true for a constant fraction of the ℓ cuts.

$$\text{Message complexity} = \tilde{\Omega}\left(\frac{\ell \cdot k^2}{1+\log r}\right)$$



Larger Initial Information

Larger Initial Information

- The argument goes through even if the nodes have more initial information.

Larger Initial Information

- The argument goes through even if the nodes have more initial information.
- KT-ρ model where nodes have topology information up to ρ hops.

Larger Initial Information

- The argument goes through even if the nodes have more initial information.
- KT-ρ model where nodes have topology information up to ρ hops.
- Don't sample first and last ρ cuts.

Larger Initial Information

- The argument goes through even if the nodes have more initial information.
- KT-ρ model where nodes have topology information up to ρ hops.
- Don't sample first and last ρ cuts.
- Lower bound holds for KT-ρ with message complexity

Larger Initial Information

- The argument goes through even if the nodes have more initial information.
- KT-ρ model where nodes have topology information up to ρ hops.
- Don't sample first and last ρ cuts.
- Lower bound holds for KT-ρ with message complexity

$$\tilde{\Omega}\left(\frac{(\ell - 2\rho + 1) \cdot k^2}{1 + \log r}\right)$$

Larger Initial Information

- The argument goes through even if the nodes have more initial information.
- KT-ρ model where nodes have topology information up to ρ hops.
- Don't sample first and last ρ cuts.
- Lower bound holds for KT-ρ with message complexity

$$\tilde{\Omega}\left(\frac{(\ell - 2\rho + 1) \cdot k^2}{1 + \log r}\right)$$

- This will give a cubic lower bound for ρ as large as $O(n/\log n)$.

Open Questions

Open Questions

- Can we prove such time budgeted lower bounds for symmetry breaking problems? MIS, $(\Delta + 1)$-coloring, Maximal Matching…

Open Questions

- Can we prove such time budgeted lower bounds for symmetry breaking problems? MIS, $(\Delta + 1)$-coloring, Maximal Matching…
- What other problems require $\Omega(n^3)$ messages?

Open Questions

- Can we prove such time budgeted lower bounds for symmetry breaking problems? MIS, $(\Delta + 1)$-coloring, Maximal Matching…
- What other problems require $\Omega(n^3)$ messages?
- What about Maximum Matching?

Open Questions

- Can we prove such time budgeted lower bounds for symmetry breaking problems? MIS, $(\Delta + 1)$-coloring, Maximal Matching…
- What other problems require $\Omega(n^3)$ messages?
- What about Maximum Matching?
    - Can we get an $O(n^2)$ message algorithm with reasonable round complexity?

Thank you!

Thank you!

Time for Questions