Gathering for mobile agents with a strong team in weakly Byzantine environments

○廣瀬 慈恩[1]　土田 将司[1]　中村 純哉[2]　大下 福仁[1]　井上 美智子[1]

[1]奈良先端科学技術大学院大学

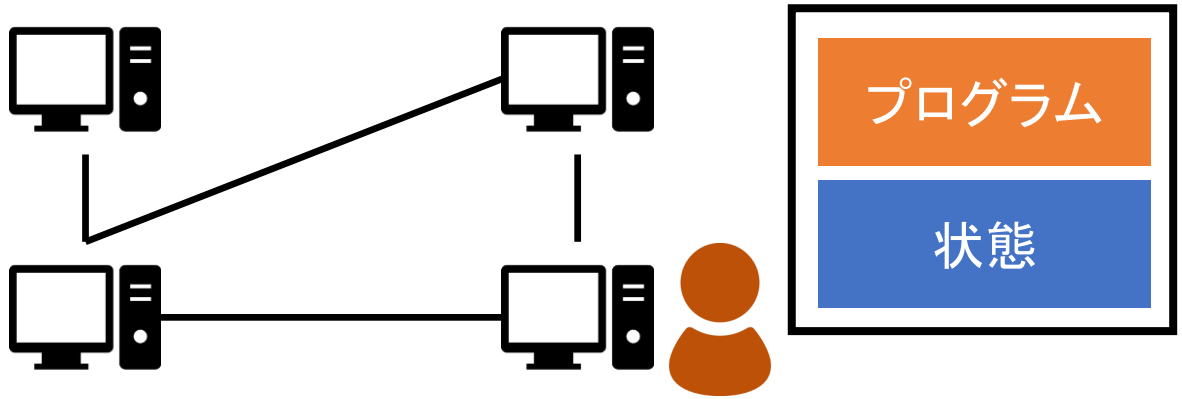
[2]豊橋技術科学大学

目次

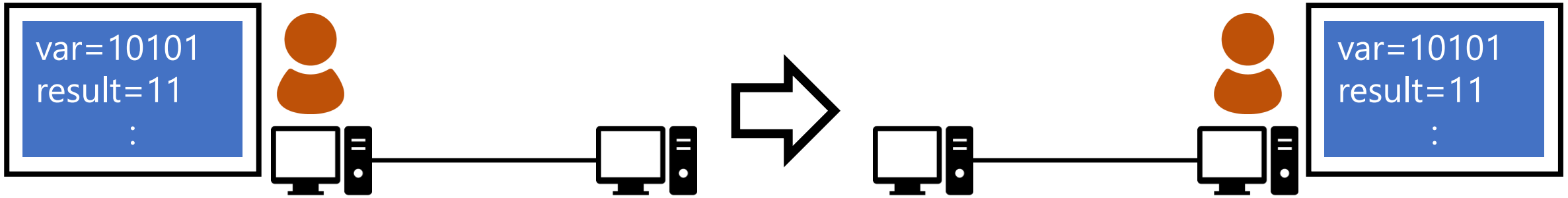
- 背景
- 成果
- モデル・タスク
- 提案アルゴリズム
  - 基本方針
  - 詳細
- まとめ

モバイルエージェント

- 自律的に計算機間を移動するソフトウェア
  - プログラムと状態を持つ
- 分散システムの設計手法

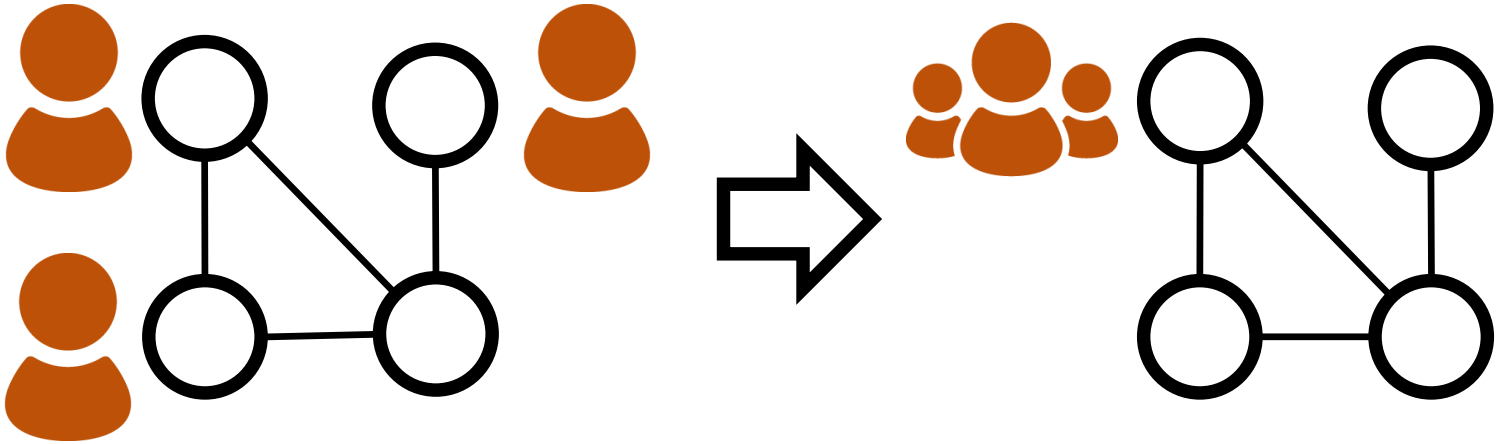


- 各エージェントは移動時にメモリの状態を保持
  - エージェントの移動で，変数の状態などは変わらない



集合

- 目的：全エージェントを1つのノードに集める
- 利点：情報共有の効率化
- 以下の仮定を組み合わせた様々なモデルの下で研究されている
  - ノードのメモリの有無
  - エージェントのIDの有無
  - エージェントの移動の同期性　など



ビザンチン環境

- ビザンチンエージェントがネットワークに存在する
  - アルゴリズムに従わず、任意の動作をするエージェント
  - クラッキングなどによるエージェントの改ざんを考慮

行動例

アルゴリズムに従わない

虚偽の情報伝達

ビザンチン環境下で動作するアルゴリズム

あらゆるエージェントの故障への耐性を保証

ビザンチン環境における同期エージェントの集合

- 既存アルゴリズムの時間複雑度は小さくない

	前提知識	故障の種類	故障数の条件	集合に要する時間
[1]	n	弱ビザンチン	$f+1 \leq k$(最適)	$O\left(n^4 \cdot \lvert\Lambda_{good}\rvert \cdot X(n)\right)$ --- ①
[1]	f	弱ビザンチン	$2f+2 \leq k$(最適)	①より大きいnと$\lvert\Lambda_{good}\rvert$の多項式
[2]	n, f	強ビザンチン	$2f+1 \leq k$(最適)	nと$\lvert\Lambda_{good}\rvert$の指数
[2]	f	強ビザンチン	$2f+2 \leq k$(最適)	nと$\lvert\Lambda_{good}\rvert$の指数
[3]	$\lceil \log \log n \rceil$	強ビザンチン	$4f^2+9f+4 \leq k$	①より大きいnと$\lvert\Lambda_{good}\rvert$の多項式

n:ノード数、k:システム全体のエージェント数、f:故障エージェント数、$f < k \leq n$、
$\lvert\Lambda_{good}\rvert$:正常エージェントの最大IDの長さ
$X(n)$:nノード全ての訪問(グラフ探索)にかかる時間(一般グラフ:$O(n^5 \log n)$、木やリング:$O(n)$))

[1]Y. Dieudonné, et al., Gathering despite mischief. ACM Trans on Algorithms 11, 1-28 (2014)
[2]S. Bouchard, et al., Byzantine gathering in networks. Distributed Computing 29(6), 435-457 (2016)
[3]S. Bouchard, et al., Byzantine gathering in polynomial time. In:ICALP. pp.502-514 (2020)

本研究の成果

- 故障数の条件の緩和により，
  弱ビザンチン環境における集合に要する時間を短縮
  - ほとんどのエージェントが故障するとは考えにくい

	前提知識	同時起動	故障数の条件	同時終了	集合に要する時間
[1]	n	なし	$f+1 \leq k$	あり	$O\left(n^4 \cdot \lvert\Lambda_{good}\rvert \cdot X(n)\right)$
提案1	N	あり	$4f^2+9f+4 \leq k$	なし	$O\left(\left(f+\lvert\Lambda_{good}\rvert\right) \cdot X(N)\right)$
提案2	N	あり	$4f^2+9f+4 \leq k$	あり	$O\left(\left(f+\lvert\Lambda_{all}\rvert\right) \cdot X(N)\right)$

$f < n$

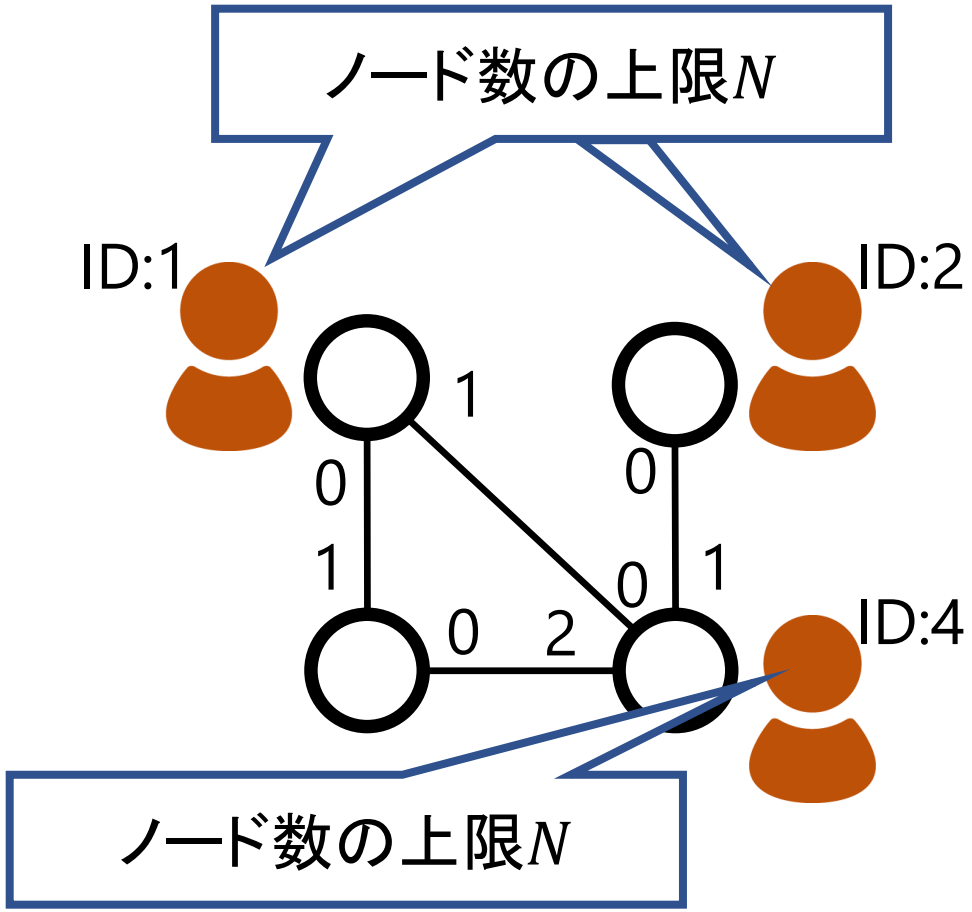
n：ノード数、N：ノード数の上限、k：システム全体のエージェント数、
f：故障エージェント数、$f < k \leq n \leq N$、
$\lvert\Lambda_{good}\rvert$：正常エージェントの最大IDの長さ、$\lvert\Lambda_{all}\rvert$：エージェントの最大IDの長さ、$\lvert\Lambda_{good}\rvert \leq \lvert\Lambda_{all}\rvert$、
$X(n)$：nノード全ての訪問（グラフ探索）にかかる時間(一般グラフ：$O(n^5 \log n)$、木やリング：$O(n)$))

目次

- 背景
- 成果
- モデル・タスク
- 提案アルゴリズム
  - 基本方針
  - 詳細
- まとめ

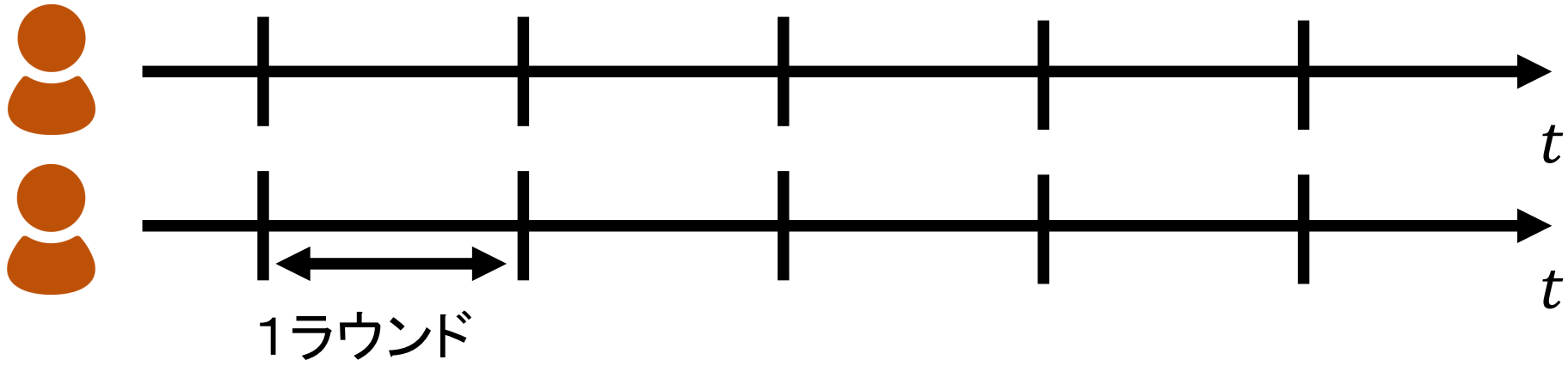
モデル

- 分散システムはグラフで表現
  - ノードはIDやメモリを持たない
  - 各ノードに繋がる辺はポート番号により識別可能
    - ポート番号：0～（次数－1）の範囲の固定番号
- エージェント
  - 任意の場所から同時にアルゴリズムを開始
  - 固有のIDを所持
  - ノード数の上限Nのみを知る
  - 無限容量のメモリを持つ
  - 同じノードならば情報共有可能

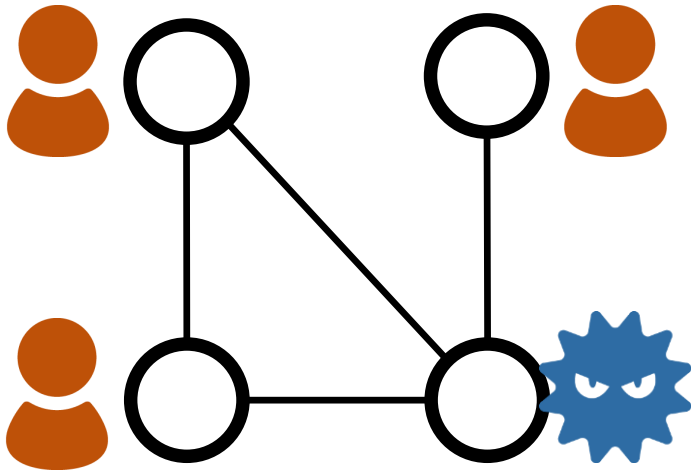


モデル

- 同期モデル
  - 1単位時間(ラウンド)で隣接ノードに移動可能

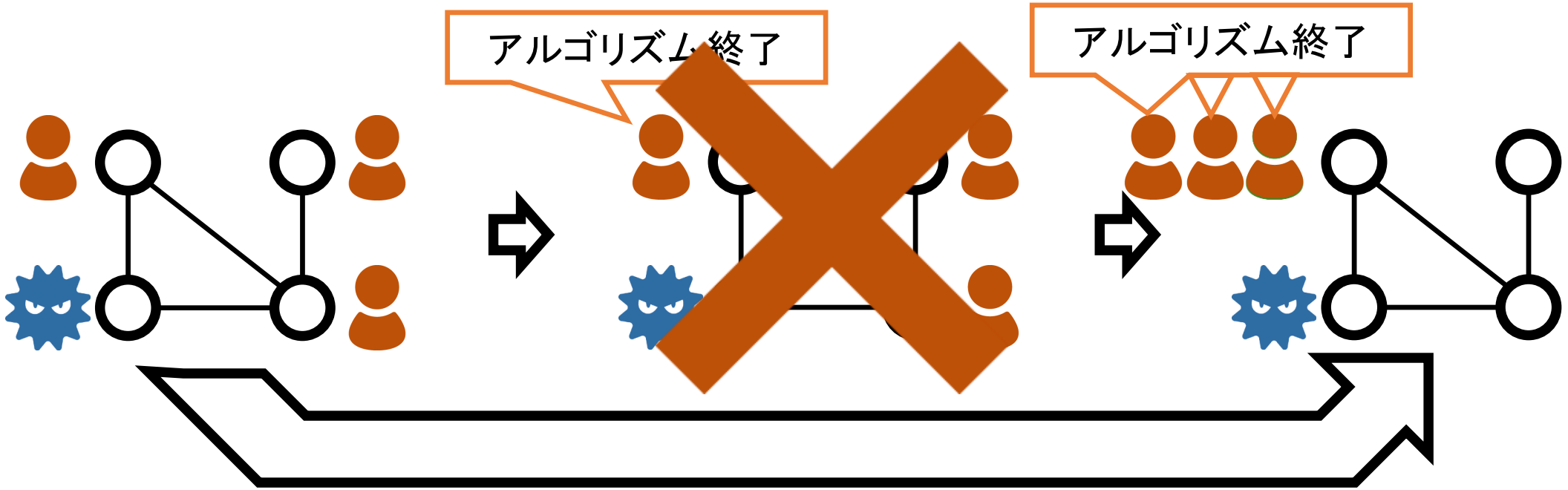


- 弱ビザンチン環境
  - ビザンチンエージェントがf体存在
    - アルゴリズムに従わず，任意の行動をとる
    - 自身のIDを偽装できない
  - 正常エージェントは少なくとも$4f^2 + 8f + 4$体



タスクと提案アルゴリズム

- ビザンチン環境における集合
  - ビザンチン環境下で全ての正常エージェントを1つのノードに集める
    - ビザンチンエージェントは集めなくて良い
- 提案アルゴリズム
  - 同時終了なし：エージェント毎にアルゴリズムを終了
  - 同時終了あり：全エージェントが同時にアルゴリズムを終了



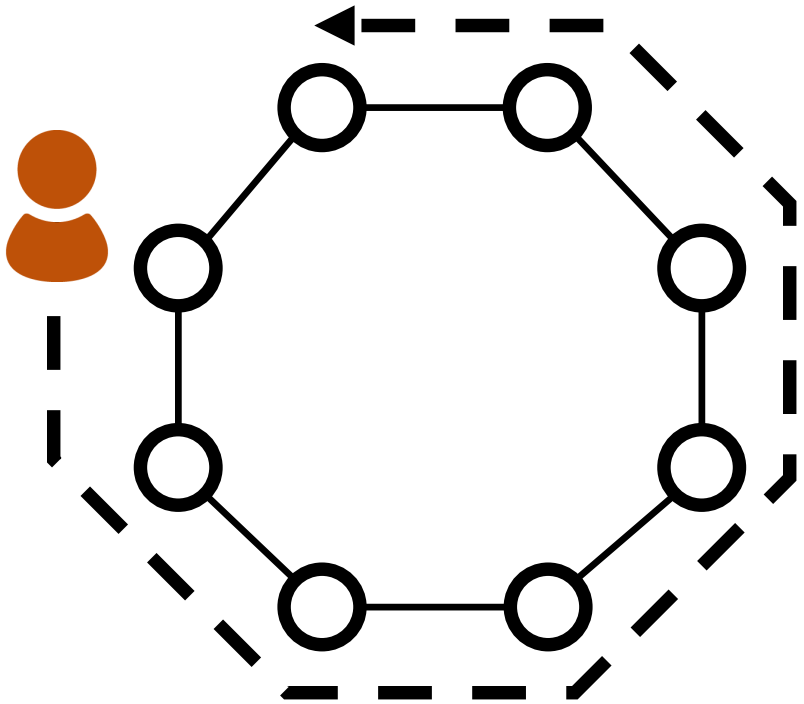
目次

- 背景
- 成果
- モデル・タスク
- 提案アルゴリズム
  - 基本方針
  - 詳細
- まとめ

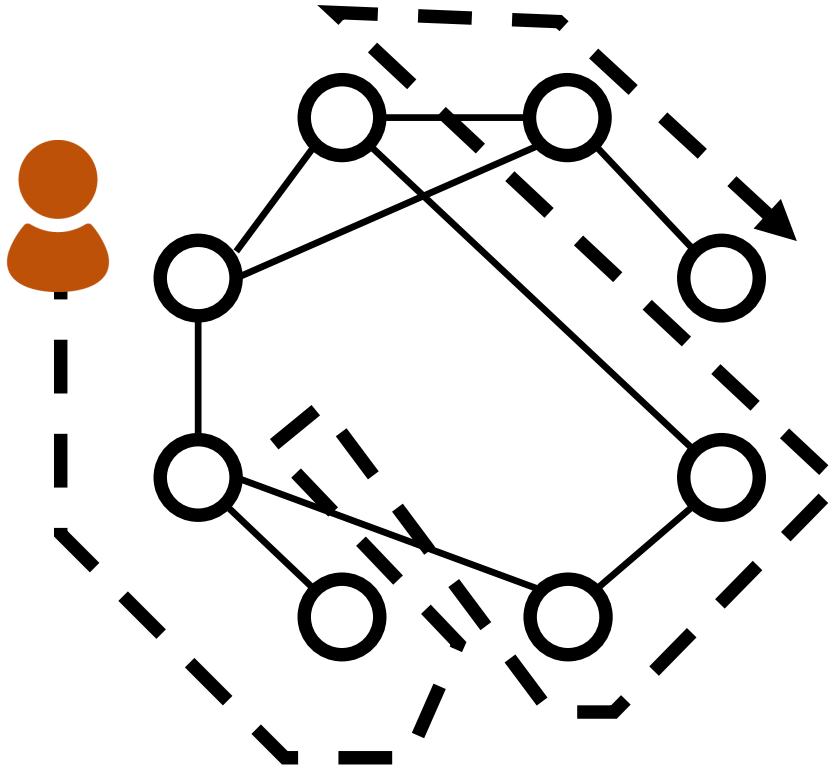
準備：探索アルゴリズム

- 提案アルゴリズムは既存の探索アルゴリズムを使用[4]
  - 目的：グラフ内の全てのノードの訪問
  - 前提知識nに対する時間複雑度：$X(n)$

木やリング：$X(n) = O(n)$



一般グラフ：$X(n) = O(n^5 \log n)$

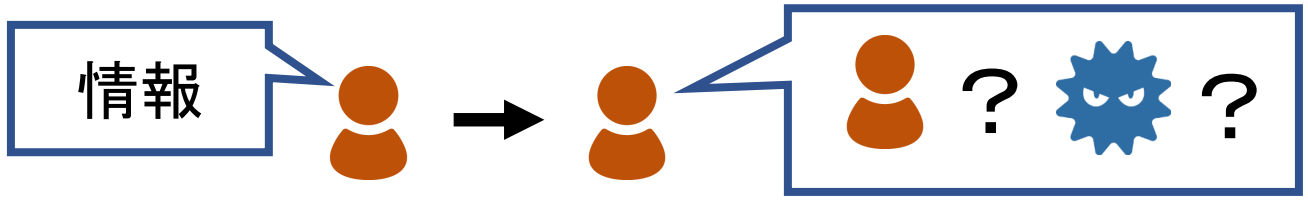


[4] Ta-Shma, A., et al., Deterministic rendezvous, treasure hunts and strongly universal exploration sequences. In: SODA. pp. 599–608 (20

難しさとその解決方法

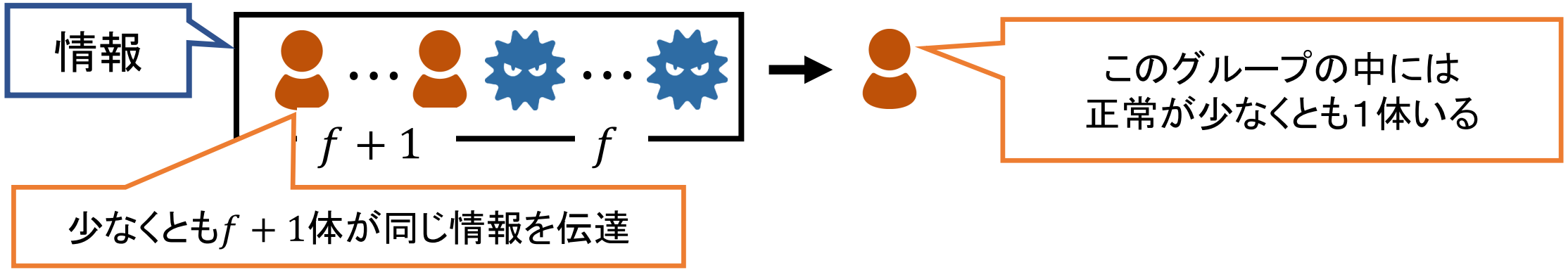
難しさ

- 他のエージェントの情報は信用できない



解決方法

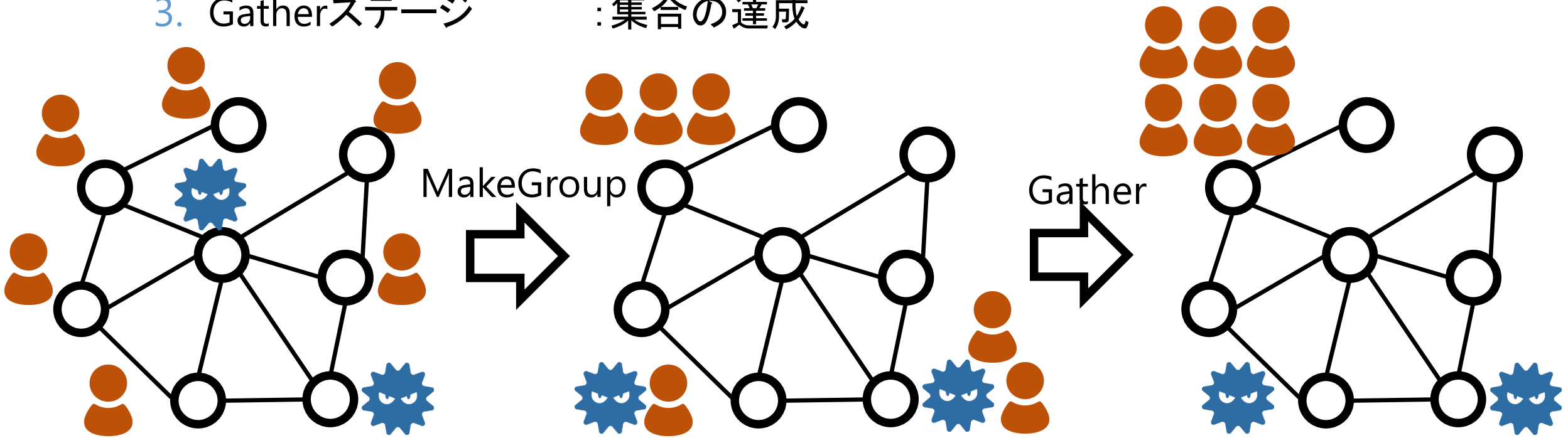
- 少なくとも$2f+1$体のエージェントからできるグループの作成
  - 大量の正常エージェントが存在するため取れる戦略



グループの持つ情報は信頼できる

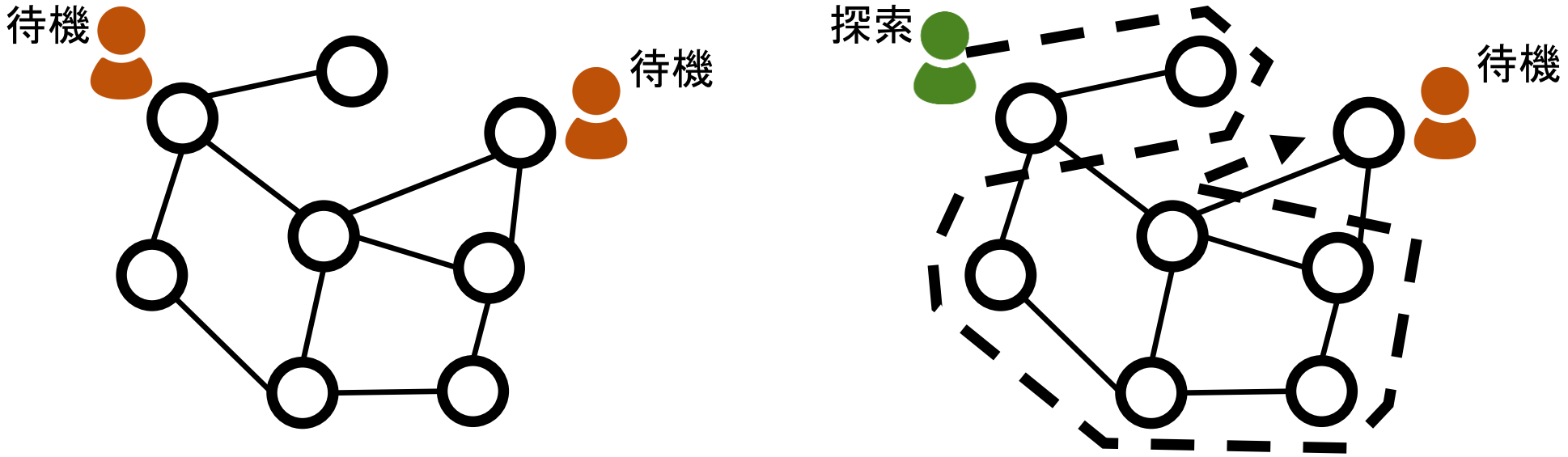
提案アルゴリズムの基本方針

- エージェントグループを作る過程を挟むことで
  ビザンチンエージェントの影響を弱めた
- 3つのステージによって集合を達成
  1. CollectIDステージ　　：全ての正常エージェントのIDの収集
  2. MakeGroupステージ　：エージェントグループの作成
  3. Gatherステージ　　：集合の達成



CollectIDステージ

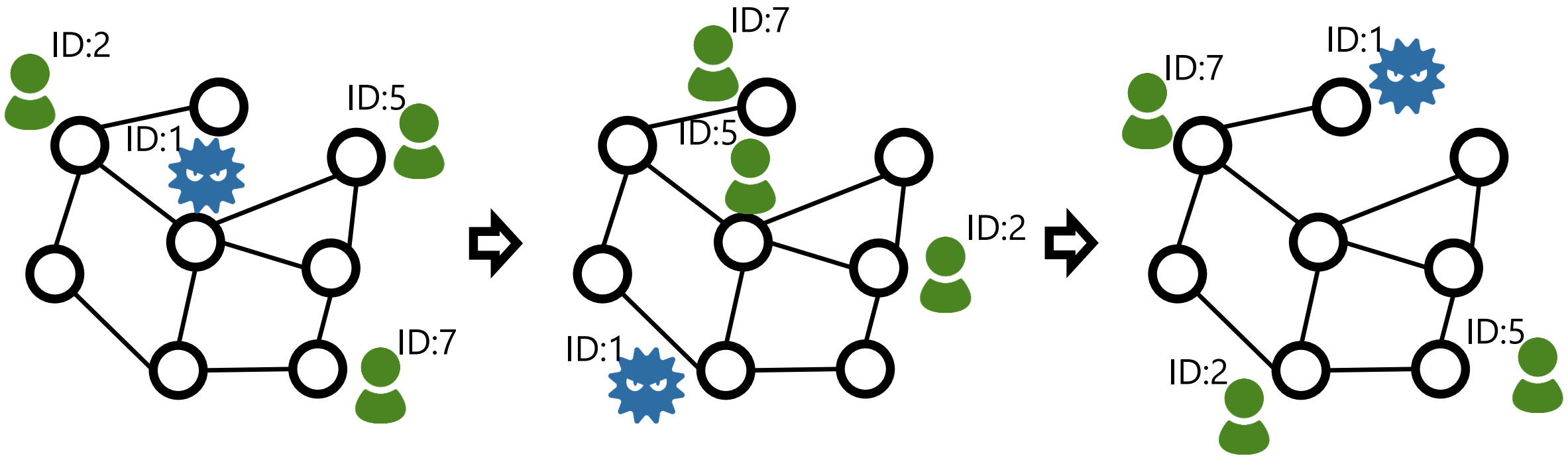
- 目的：全ての正常エージェントのIDの収集
- 動作：2体のエージェントが出会う既存アルゴリズム[5]を使用
  - 各エージェントが$O(\log ID)$回探索もしくは待機を繰り返す
    - 探索：探索アルゴリズムの使用
    - 待機：ノードで$X(N)$ラウンド待機
- CollectIDステージの時間複雑度：$O\left(\log ID \cdot X(N)\right)$



[5] Dessmark, A., et al., Deterministic rendezvous in graphs. Algorithmica 46(1), 69–96

MakeGroupステージ：アイディア

- 目的：信頼できるグループの作成
  - 十分に多くのエージェントで構成されたグループ
- アイディア：最小IDの元に集まる
  - 最小IDは待機，それ以外は最小IDを探索アルゴリズムで捜索
  - ビザンチンが最小IDならば，信頼できるグループが作れない可能性

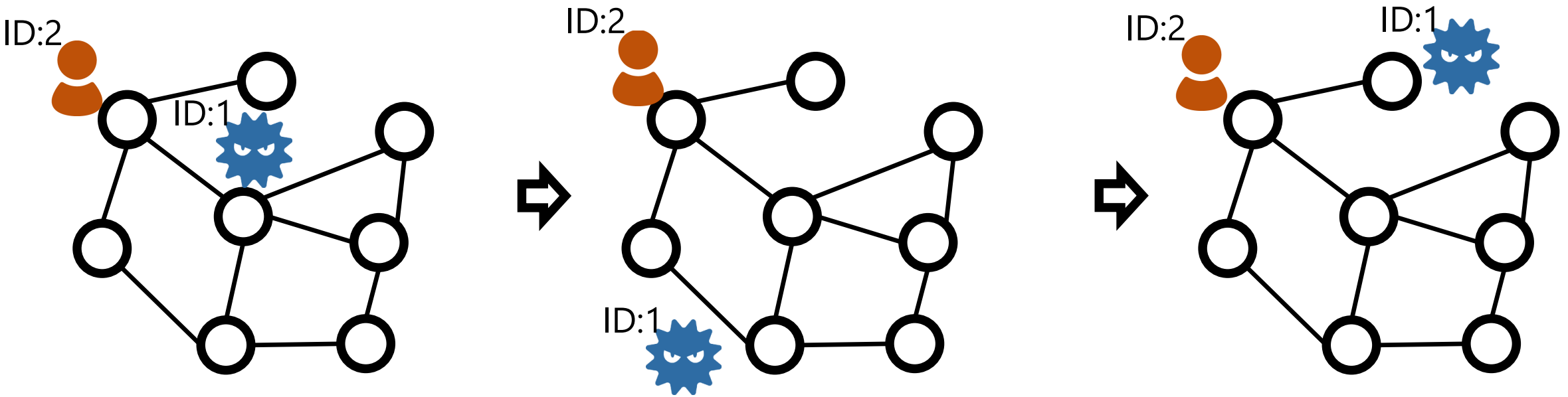


MakeGroupステージ：アイディアの改善

- 集めたIDで小さい方から$f+1$以内
  - 信頼できるグループが出来るまで待機
- それ以外
  - 集めたIDで最小IDを探索アルゴリズムで探索
  - 見つからないならば，次に小さいIDを探索

エージェントはfを知らない
→ 見積もり値を使用

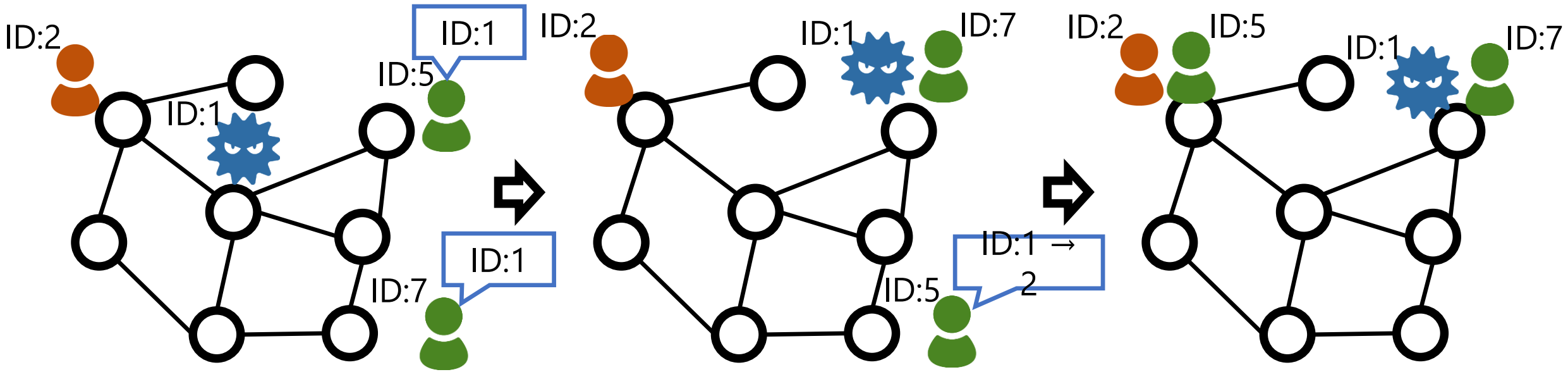
少なくとも1体の正常エージェントは待機



MakeGroupステージ：アイディアの改善

- 集めたIDで小さい方から$f+1$以内
  - 信頼できるグループが出来るまで待機
- それ以外
  - 集めたIDで最小IDを探索アルゴリズムで探索
  - 見つからないならば，次に小さいIDを探索

正常エージェントは高々$f+1$グループに分かれる

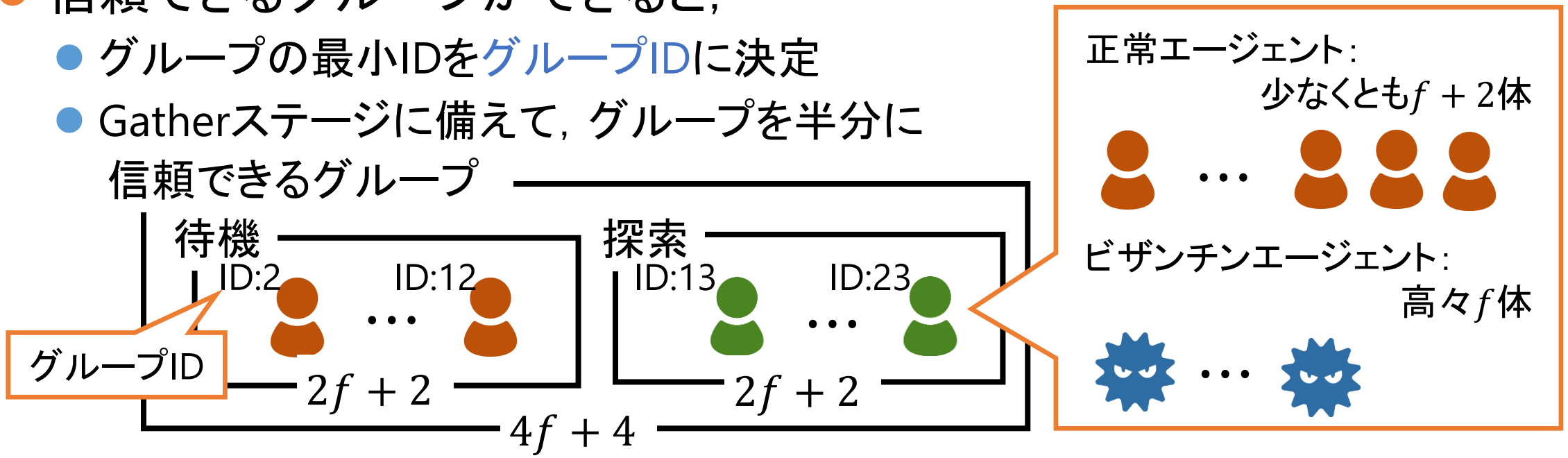


MakeGroupステージ：信頼できるグループ

- 高々$f+1$グループに分かれる
  - fの条件式：$4f^2+9f+4=(4f+4)(f+1) \leq$ 全エージェント数

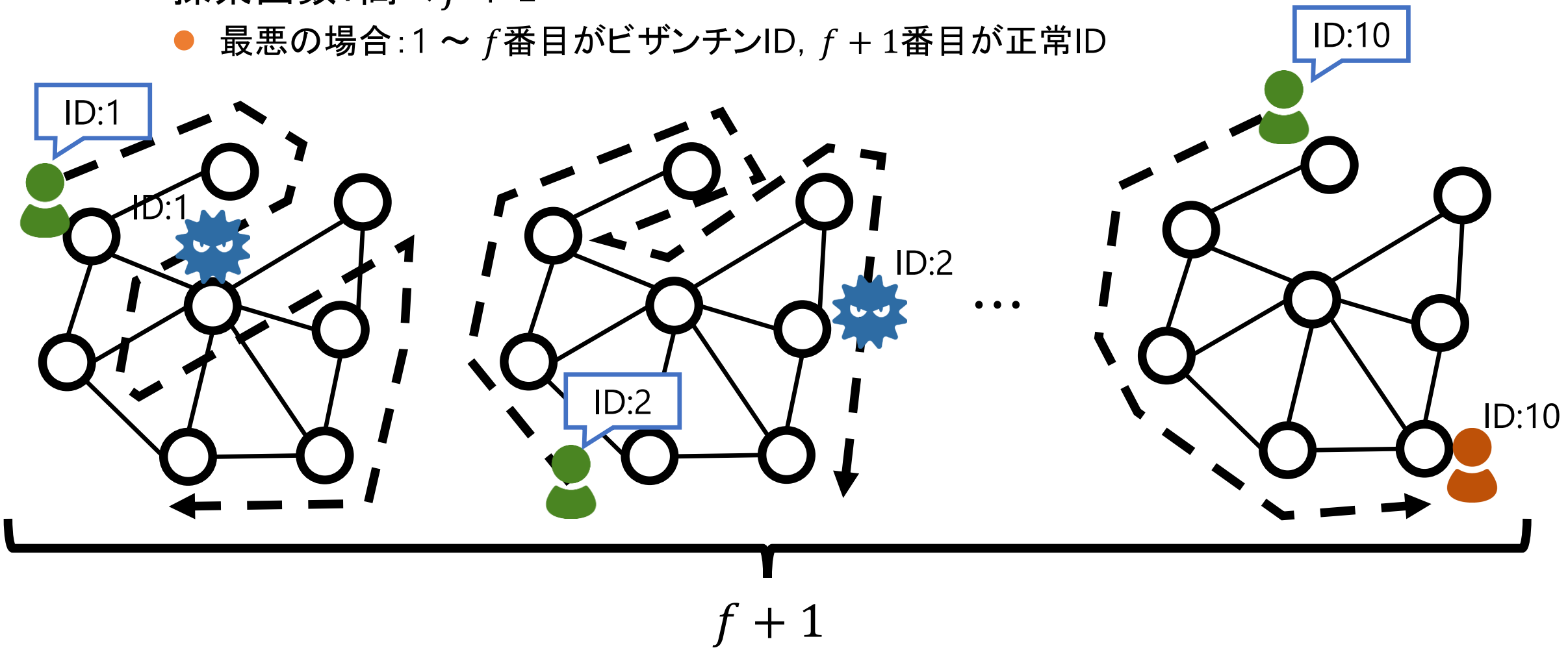
少なくとも1グループには少なくとも$4f+4$体のエージェント
→ 少なくとも1グループは信頼できるグループ

- 信頼できるグループができると，
  - グループの最小IDをグループIDに決定
  - Gatherステージに備えて，グループを半分に



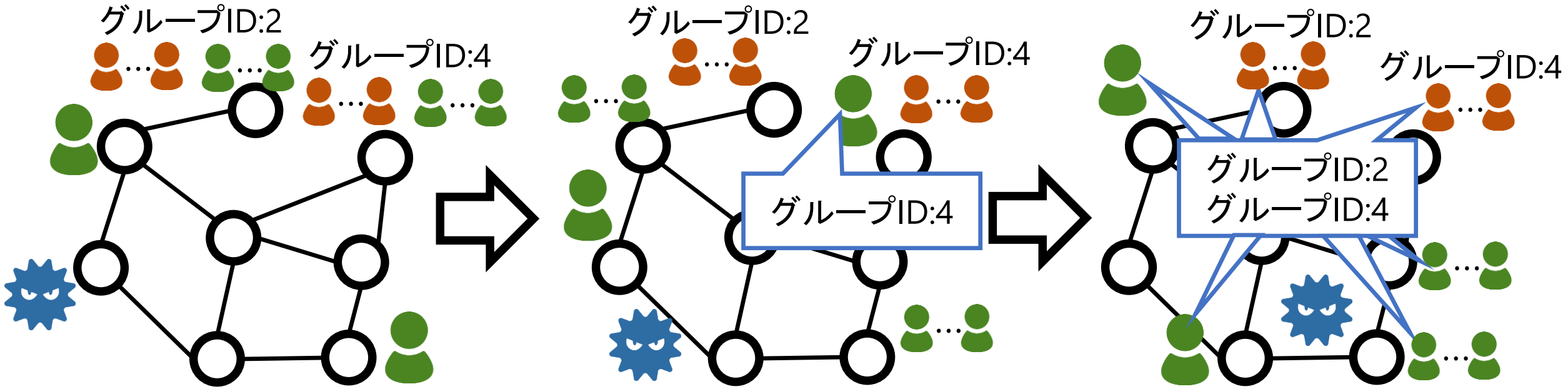
MakeGroupステージ：時間複雑度

- MakeGroupステージの時間複雑度：$O\left(f \cdot X(N)\right)$
  - 探索回数：高々$f+1$
    - 最悪の場合：1 ～ f番目がビザンチンID，$f+1$番目が正常ID



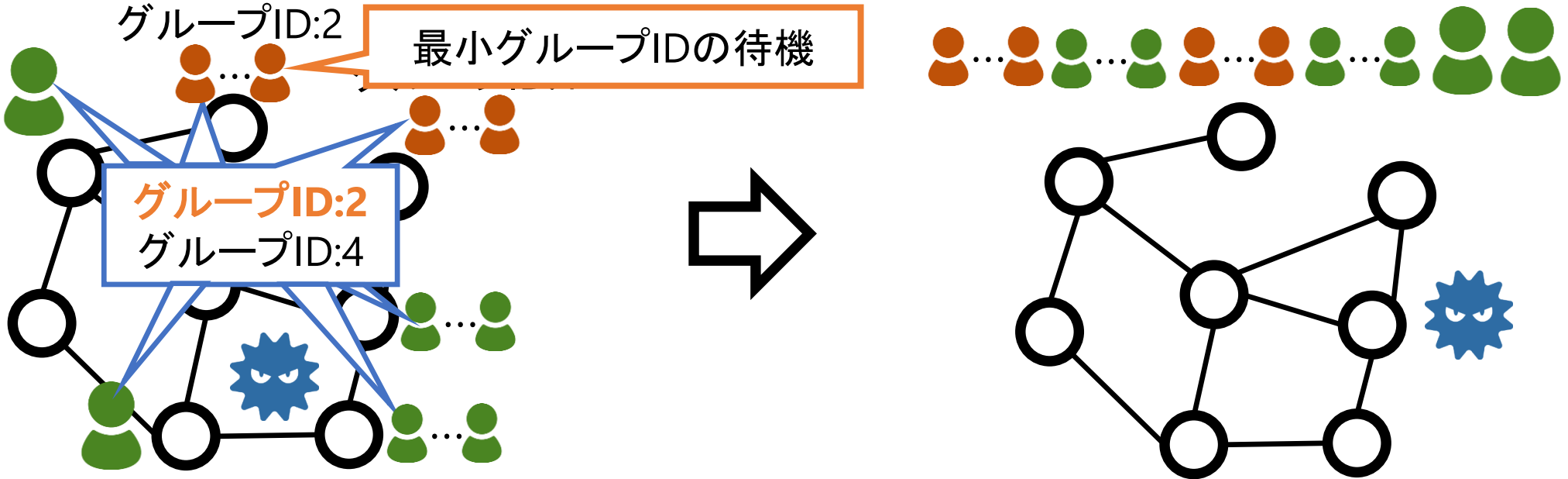
Gatherステージ：動作の前半部分

- 目的：グループIDの共有
- …：信頼できるグループ
  - …（待機）：$X(N)$ラウンド待機，グループIDの伝達と収集
  - …（探索）：探索アルゴリズムを使用，グループIDの伝達と収集
- …：それ以外
  - 探索アルゴリズムを使用，グループID収集



Gatherステージ：動作の後半部分

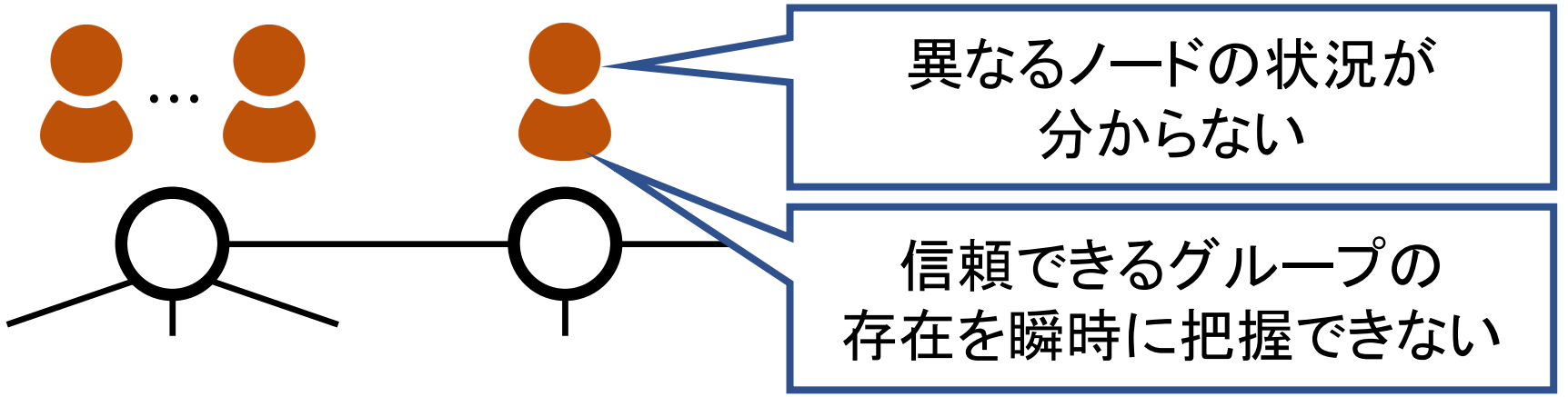
- 目的：集合の達成
- 最小グループIDの待機
  - $X(N)$ラウンド待機後，アルゴリズム終了
- それ以外
  - 最小グループIDの待機グループを探索アルゴリズムで捜索
  - 発見したならば，アルゴリズム終了



Gatherステージ：課題とその解決方法

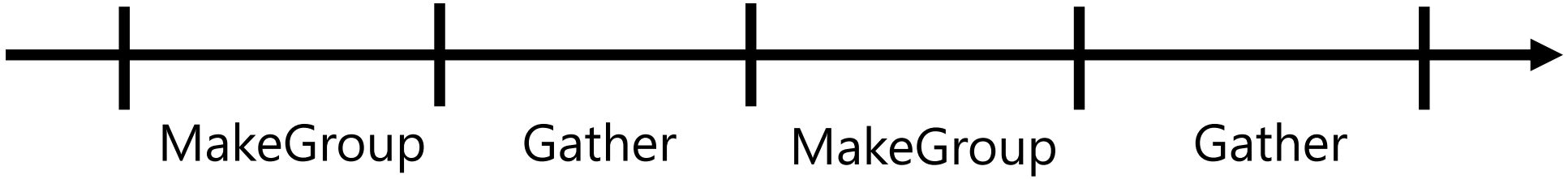
課題

- 同じタイミングでGatherステージに入れない



解決方法

- MakeGroupステージの$X(N)$ラウンド毎にGatherステージを挿入
  - 交互に実行しても漸近的な時間複雑度は変化しない
  - Gatherステージ：信頼できるグループがいなければ，影響なし



同時終了なしアルゴリズムの時間複雑度

- CollectIDステージ　　：全ての正常エージェントのIDの収集
  - 時間複雑度：$O\left(\left|\Lambda_{good}\right| \cdot X(N)\right)$
    - $\left|\Lambda_{good}\right| (= \log \max(ID_{good}))$：正常エージェントの最大IDの長さ
- MakeGroupステージ　：信頼できるグループの作成
  - 時間複雑度：$O(f \cdot X(N))$
- Gatherステージ：集合の達成
  - 時間複雑度：$O(X(N))$
    - Gatherステージの前半：$X(N)$
    - Gatherステージの後半：$X(N)$

同時終了なしアルゴリズムの時間複雑度

$$O\left(\left(f + \left|\Lambda_{good}\right|\right) \cdot X(N)\right)$$

まとめ

成果

- 故障数の条件緩和により，既存研究より高速なアルゴリズムを提案

今後の課題

- 同時起動なしの場合におけるアルゴリズムの設計

	前提知識	同時起動	故障数の条件	同時終了	集合に要する時間
[1]	n	なし	$f+1 \leq k$	あり	$O\left(n^4 \cdot \lvert\Lambda_{good}\rvert \cdot X(n)\right)$
提案1	N	あり	$4f^2+9f+4 \leq k$	なし	$O\left(\left(f+\lvert\Lambda_{good}\rvert\right) \cdot X(N)\right)$
提案2	N	あり	$4f^2+9f+4 \leq k$	あり	$O\left(\left(f+\lvert\Lambda_{all}\rvert\right) \cdot X(N)\right)$

n：ノード数、N：ノード数の上限、k：システム全体のエージェント数、f：故障エージェント数、
$f < k \leq n \leq N$、
$\lvert\Lambda_{good}\rvert$：正常エージェントの最大IDの長さ、$\lvert\Lambda_{all}\rvert$：エージェントの最大IDの長さ、$\lvert\Lambda_{good}\rvert \leq \lvert\Lambda_{all}\rvert$、
$X(n)$：nノード全ての訪問（グラフ探索）にかかる時間(一般グラフ：$O(n^5 \log n)$、木やリング：$O(n)$))