Robustness of Epistemic Gossip Protocols Against Data Loss

Yoshikatsu Kobayashi
Department of Computer Science
Univesity of Tsukuba
Tsukuba, Japan
kobayashi@mas.cs.tsukuba.ac.jp

Koji Hasebe
Department of Computer Science
University of Tsukuba
Tsukuba, Japan
hasebe@cs.tsukuba.ac.jp

ABSTRACT

The gossip problem seeks to determine the minimum number of calls required for all agents in a network to share their secrets. To address this problem in a distributed manner, epistemic gossip protocols have been proposed, where each agent decides whom to call based on their knowledge. While extensive research has explored the feasibility of information dissemination under various protocols and environmental conditions, a recent study introduced a model that assumes the presence of unreliable agents. In this model, when an agent fails, it loses both the secrets and telephone numbers obtained from previous calls and returns to its initial state. In this context, the robustness of some existing protocols against data loss due to failures, as well as a sufficient condition for agents to detect failures, has been demonstrated. The objective of this paper is to complement the previous study through a comprehensive analysis and to explore methods for designing robust epistemic gossip protocols. Our contributions are threefold. First, we clarify the necessary and sufficient conditions regarding network structure for existing protocols to succeed (i.e., for all agents to know all secrets) at different levels. Second, we elucidate the necessary and sufficient conditions for failure detection. Finally, we present protocols in which agents, upon detecting their own or others' failures, take actions to recover the lost data and analyze the robustness of these protocols.

KEYWORDS

Gossip Protocols, Epistemic Logic, Failures, Robustness

ACM Reference Format:
Yoshikatsu Kobayashi and Koji Hasebe. 2025. Robustness of Epistemic Gossip Protocols Against Data Loss. In *Proc. of the 24th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2025), Detroit, Michigan, USA, May 19 – 23, 2025*, IFAAMAS, 9 pages.

1 INTRODUCTION

The gossip protocol determines one-to-one communications (called *calls*) among agents in a network to share their *secrets*. The origin of research on gossip can be traced back to the so-called *gossip problem* [5, 12, 16], raised in the early 1970s (cf. also [13] for a survey). In this problem, each agent initially knows only its own secret, and with each call, two agents share their secrets obtained by the previous calls. The gossip problem seeks to determine the minimum number of calls required for all agents to know all the secrets (i.e., for everyone to become an *expert*).

In classical studies, it was assumed that there is a central controller that instructs each agent on whom to call. In contrast, to address the problem in a distributed manner, more recent studies have introduced *epistemic gossip protocols* [1–4, 8, 20] where each agent decides whom to call based on the knowledge obtained from past communications. A common goal among these studies is to determine which protocols and environmental conditions can ensure that all agents eventually become experts. In particular, most attention has been given to the process of (possibly higher-order) knowledge formation for agents during the course of communication.

Recently, a study [10] introduced a model that assumed the presence of unreliable agents based on the propositional S5 epistemic logic (cf. [9]). In this model, when an agent fails, it loses the secrets obtained from previous communications and returns to its initial state. Under this setting, the study demonstrated the necessary and sufficient conditions for protocols called ANY (Any call), and PIG (Possible Information Growth) originally introduced by [21] to be fairly successful (i.e., in a fair sequence where no further call occurs, all agents become expert) and a sufficient condition for agents to detect their own and the other agents' failures. These results provide new insights into the robustness of epistemic gossip protocols against failures and are particularly useful for applications in real-world distributed systems. However, a comprehensive analysis of existing protocols has not yet been achieved. Moreover, as suggested in the previous study, the property about failure detection is valuable for designing protocols where agents detect failures and make calls to recover lost data. However, such protocols have not yet been thoroughly investigated.

The objective of this paper is to complement the robustness analysis of existing protocols under the settings introduced in the study [10] and to show the necessary and sufficient conditions for agents to detect failures. Furthermore, we aim to explore methods for designing robust protocols defining behavior upon detecting a failure.

Our contributions are threefold. First, to expand the scope of our analysis of existing protocols, in addition to ANY, CO, and PIG, we include two additional protocols, HSS (Hear Some Secret) and HMS (Hear My Secret). Also, we consider the four different types of properties that ensure a protocol is *successful* originally introduced by [21]. Based on this, we demonstrate the necessary and sufficient conditions for the protocol to be successful in each of the twenty cases.

Second, we present a sufficient condition for agents to detect failures. In contrast to the previous study [10], which assumes that the number of failures is at most one, our theorem holds even for

Proc. of the 24th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2025), Y. Vorobeychik, S. Das, A. Nowé (eds.), May 19 – 23, 2025, Detroit, Michigan, USA. © 2025 International Foundation for Autonomous Agents and Multiagent Systems (www.ifaamas.org).

multiple failures. Additionally, we provide a necessary condition when the number of failures is at most one.

Finally, to enhance the robustness of protocols against failures, we propose a method to extend a given protocol by adding the following conditional statements: "if an agent detects its own failure, it calls another agent who owns the secret it lost," and "if an agent detects the failure of another agent, it sends its own secret to the failed agent." The first statement is intended to help the failed agent recover its lost data, while the second aims to assist in the recovery of other failed agents. These behaviors can be incorporated into the existing protocols while preserving their original behaviors. Therefore, we introduce ten new protocols by extending the five existing protocols mentioned earlier using these two conditions. Furthermore, using our theorem on failure detection, we demonstrate the robustness of these new protocols. Although these extensions do not improve the robustness of the existing protocols. However, future research may demonstrate the usefulness of this approach by adjusting the behavior of agents in response to failures.

The structure of this paper is as follows. Section 2 presents related work. Section 3 overviews a logical formalization to model the agents who may lose data due to failures, which was originally given by [10]. Section 4 analyzes the robustness of some existing protocols. Section 5 demonstrates the necessary and sufficient conditions for agents to detect failures. Section 6 introduces protocols extended with behaviours added for handling failures. Finally, Section 7 concludes the paper and suggests further research.

2 RELATED WORK

In previous studies on epistemic gossip protocols, various protocols and environmental settings have been considered (cf. [19]).

One of the key distinctions in how information is exchanged during a call is between the *merge-then-learn* and *learn-then-merge* approaches. In the former (e.g., [1]), the information exchanged during a call is merged externally before being delivered to both parties. In the latter, each agent first obtains the information sent by the other party before performing the merge, allowing agents to gain more information than in the former case. Similar to [4, 21], we adopt the learn-then-merge approach.

Studies [21, 22] introduced the concept of *dynamic gossip*, where each agent has its own secret and *telephone number* (or *number*, for short), and calls can only be made to agents whose numbers are known. During a call, agents exchange both their secrets and their numbers. The study [10] assumes dynamic gossip, where agents lose not only the secrets but also the numbers they obtained if they fail. Our model follows this setup.

Although not directly addressed in this study, previous research has analyzed the conditions under which all agents become *super experts* (i.e., agents who know that all agents are experts) and the conditions under which it becomes common knowledge that everyone is an expert [2, 18].

In addition to the study [10], a study [17] also considered unreliable agents and analyzed the feasibility of some existing protocols. In their model, agents may incorrectly convey their secrets.

One of the main contribution of our work is related to failure detection. In the field of distributed systems, a number of studies have investigated the relationship between failure detection capabilities and the achievement of system goals. A pioneering work in this area is the study [6], which proposed a solution to the consensus problem in asynchronous systems assuming crash failures. Since then, many solutions have been proposed for consensus problems, extending the analysis to include Byzantine failures [11, 14, 15]. Recently, the study [7] built on these ideas and proposed an accountable Byzantine consensus algorithm.

3 FAILURE MODEL

In this section, we provide an overview of the model that assumes the presence of unreliable agents, which may lose data previously obtained due to failures. This failure model was orignally introduced in [10], as an extension of the model presented in [21], by incorporating the event of agent failures. In the following, we first define the semantic model (in Section 3.1) and then define the language and syntax of the inference system for this model based on propositional S5 epistemic logic (in Section 3.2). Finally, we introduce the protocols that are the subject of analysis in this paper (in Section 3.3).

3.1 Semantic model

Definition 3.1 (Event). Let A be the set of *agents*. Throughout this paper, we assume that $|A| \geq 3$. A *call* from agent x to agent y is denoted by xy, and a *failure* of agent x is denoted by $[x]$. Collectively, calls and failures are referred to as *events*. The set of calls is denoted by C, the set of failures by F, and the set of events by E, where $E = C \cup F$. For any $xy \in C$ and $z \in A$, z is said to be involved in xy if $z \in \{x, y\}$. Similarly, for any $[x] \in F$ and $z \in A$, z is said to be involved in $[x]$ if $z = x$.

Calls can occur in two modes: asynchronous calls and synchronous calls. In asynchronous calls, agents cannot perceive calls occurring between other agents, whereas in synchronous calls, agents can perceive that a call has occurred, but they cannot know who made the call. In this paper, we focus on asynchronous calls.

Definition 3.2 (Event sequence). Let E^* denote the set of *event sequences*. We introduce the following notations for event sequence, where $\sigma, \sigma', \tau \in E^*$.

- A sequence of events $\mathrm{e}_1, \mathrm{e}_2, \ldots$ is written as $\mathrm{e}_1; \mathrm{e}_2; \cdots$. The semicolon ";" denotes concatenation, so $\sigma; \tau$ denotes the sequence obtained by concatenating τ to the end of σ, and $\sigma; \mathrm{e}$ denotes the sequence obtained by appending $\mathrm{e} \in E$ to the end of σ.
- We write $\sigma \sqsubseteq \tau$ to mean that σ is a prefix of τ (including the case where $\sigma = \tau$). If $\sigma \sqsubseteq \tau$ and $\sigma \neq \tau$, then we write $\sigma \sqsubset \tau$.
- The empty sequence is denoted by ϵ.
- The sequence obtained by removing all calls involving agent x from σ is denoted by σ^{-x}.
- We write $x \in \sigma$ if a call involving agent x is included in σ, and $x \notin \sigma$ if no such call is included.
- The n-th event in σ is denoted by σ_n, and the prefix of length n is denoted by $\sigma|n$, where $\sigma|0 = \epsilon$.

In the model proposed by [10], not only are secrets exchanged, but information regarding the transmission paths of these secrets is

also exchanged. This is represented by a tree structure called a *memory tree*, defined below. In this model, agents exchange information represented by memory trees.

Definition 3.3 (Memory tree). A memory tree is a binary tree defined inductively as follows:

- For any $x \in A$, $\langle x \rangle$ is a memory tree.
- If t and t' are memory trees and $xy \in C$, then $\langle t, xy, t' \rangle$ is a memory tree.

Intuitively, memory tree $\langle t, xy, t' \rangle$ not only represents the merged secrets t and t' held by x and y, but also indicates that these secrets were shared between x and y through the call xy. Thus, for example, when agent z has the memory tree $\langle \langle x, xy, y \rangle, xz \rangle$, z knows not only the secrets of x, y, and z, but also that the secrets of x and y were first gathered by x through the call xy, and then passed to z through the call xz.

In the rest of this paper, $\mathcal{T}$ denotes the set of memory trees. The set of leaves of a memory tree $t \in \mathcal{T}$ is denoted by $leaves(t)$, and the root of t is denoted by $root(t)$. We write $t \subseteq t'$ to indicate that t is a subtree of t'.

Next, we define the *σ-induced memory tree* for an agent $x \in A$ after the execution of a sequence $\sigma \in E^*$.

Definition 3.4 (σ-induced memory tree for x). The σ-induced memory tree for agent x is a memory tree defined by the mapping $mtree : A \times E^* \to \mathcal{T}$, given the sequence σ, as follows:

- If $\sigma = \epsilon$, then $mtree(x, \sigma) = \langle x \rangle$ for any x.
- If $\sigma = \sigma'; yz$, then
$$mtree(x, \sigma) = \begin{cases} \langle mtree(y, \sigma'), yz, mtree(z, \sigma') \rangle & \text{if } x \in \{y, z\}, \\ mtree(x, \sigma') & \text{otherwise.} \end{cases}$$
- If $\sigma = \sigma'; [y]$, then
$$mtree(x, \sigma) = \begin{cases} \langle y \rangle & \text{if } x = y, \\ mtree(x, \sigma') & \text{otherwise.} \end{cases}$$

The memory tree $mtree(x, \sigma)$ represents the set of secrets known by agent x and the transmission paths of those secrets after the sequence σ has been executed. The set of secrets known by agent x corresponds to $leaves(mtree(x, \sigma))$. Additionally, $mtree(x, \sigma)$ contains information about the transmission paths of the secrets known by x.

Next, we define the *gossip graph*. A gossip graph represents the knowledge of agents about (telephone) numbers and secrets at a given point in time, and is defined as a triple consisting of the set of agents A, the relation N regarding numbers, and the set of memory trees $\{t_x\}_{x \in A}$.

Definition 3.5 (Gossip graph). A gossip graph G is a triple $(A, N, \{t_x\}_{x \in A})$, where A is the set of agents, N is a binary relation on A, and $t_x \in \mathcal{T}$ for each $x \in A$.

We define $N^{-1} = \{(x, y) \mid (y, x) \in N\}$. The set of gossip graphs is denoted by $\mathcal{G}$. The gossip graph that represents the agents' knowledge about numbers and secrets at the initial state is called the *initial gossip graph* and is defined as follows.

Definition 3.6 (Initial gossip graph). An initial gossip graph is a gossip graph that satisfies the following conditions.

- N is reflexive.
- $t_x = \langle x \rangle$ for all $x \in A$.

Here, for the initial gossip graph $G = \{A, N, \{t_x\}_{x \in A}\}$, if there exists a path between any x and y in $N \cup N^{-1}$, G is called a *weakly connected graph*. Also, if there exists a path between any x and y in N, G is called a *strongly connected graph*. Furthermore, in G, if an agent a that satisfies the following two conditions:

(1) For any agent $x \neq a$, $(a, x) \notin N$;
(2) For any agent $y \neq a$, there exists an agent $z \neq a, y$ such that $(y, z) \in N$;

it is called an *isolated agent*, and an initial gossip graph that does not contain an isolated agent is called a *non-isolated graph*.

The gossip graph obtained from the initial gossip graph G after the execution of a sequence σ of events is called the σ-induced gossip graph, which represents the global state.

Definition 3.7 (σ-induced gossip graph). Given a gossip graph $G = (A, N, \{t_x\}_{x \in A})$ and a sequence of events σ, the σ-induced gossip graph is denoted by G^σ, where $G^\sigma = (A, N^\sigma, \{mtree(x, \sigma)\}_{x \in A})$. The sets A and $\{mtree(x, \sigma)\}_{x \in A}$ are uniquely determined by σ and $G = (A, N, \{t_x\}_{x \in A})$, so it suffices to define only N^σ as follows:

- $N^\epsilon = N$
- For $\sigma \neq \epsilon$,
  - If $\sigma = \sigma'; xy$, then
$$N^\sigma = N^{\sigma'} \cup \{(m, z) \mid (m = x \wedge (y, z) \in N^{\sigma'}) \vee (m = y \wedge (x, z) \in N^{\sigma'})\}.$$
  - If $\sigma = \sigma'; [x]$, then
$$N^\sigma = N^{\sigma'} \setminus \{(x, z) \mid (x, z) \in N^{\sigma'} \wedge z \neq x\}.$$

A call xy is said to be *valid* in G^σ if $(x, y) \in N^\sigma$ holds in the gossip graph G^σ.

With these assumptions, we define the *asynchronous accessibility relation* as follows.

Definition 3.8 (Asynchronous accessibility relation). The asynchronous accessibility relation $\sim_x$ between gossip graphs is defined as follows:

Let G_1 and G_2 be gossip graphs, and suppose that $G_1^\sigma = (A, N_1^\sigma, \{mtree(x, \sigma)\}_{x \in A})$ and $G_2^\tau = (A, N_2^\tau, \{mtree(x, \tau)\}_{x \in A})$. Then, $G_1^\sigma \sim_x G_2^\tau$ holds if and only if $mtree(x, \sigma) = mtree(x, \tau)$.

The Kripke model consisting of a tuple of a set of gossip graphs and a set of accessibility relations is called the *gossip model* defined as follows.

Definition 3.9 (Gossip model). Given a set of agents A, the asynchronous gossip model $\mathcal{G}^\sim$ is a triple $\mathcal{G}^\sim = (\mathcal{G}, \langle \sim_a \rangle_{a \in A})$, where $\mathcal{G}$ is the set of gossip graphs, and $\langle \sim_a \rangle_{a \in A}$ is the set of accessibility relations for each agent $a \in A$.

3.2 Epistemic Logic

In an epistemic gossip protocol, the condition $\varphi(x, y)$ under which agent x calls agent y is expressed in the language of epistemic logic, and its truth is determined by the gossip model. The language used to describe these conditions is defined as follows:

Definition 3.10 (Language). Given a set of agents A, the language $\mathcal{L}$ used to describe the conditions in epistemic gossip protocols is defined as follows.

$$\varphi ::= \top \mid \mathtt{C}(a,b) \mid \mathtt{F}(a) \mid \mathtt{S}(a,b) \mid \neg\varphi \mid (\varphi \wedge \varphi) \mid K_a\varphi,$$

where $a, b, c \in A$. Logical connectives such as $\rightarrow, \vee, \leftrightarrow$ are defined in the standard way.

In this language, $\mathtt{C}(a,b)$ indicates that a call between a and b has occurred, $\mathtt{F}(a)$ indicates that a has failed at least once, and $\mathtt{S}(a,b)$ indicates that a knows b's secret. Additionally, $K_a\varphi$ expresses that a knows φ. The truth of these formulas is defined as follows.

Definition 3.11 (Truth conditions). Given an asynchronous gossip model $\mathcal{G}^{\sim} = (\mathcal{G}, \langle\sim_a\rangle_{a\in A})$ and any $G^\sigma \in \mathcal{G}$ and any formula φ in $\mathcal{L}$, the truth of φ is defined inductively as follows:

$\mathcal{G}^{\sim}, G^\sigma \models \top$ always

$\mathcal{G}^{\sim}, G^\sigma \models \mathtt{C}(a,b)$ iff $ab \in \sigma$

$\mathcal{G}^{\sim}, G^\sigma \models \mathtt{F}(a)$ iff $[a] \in \sigma$

$\mathcal{G}^{\sim}, G^\sigma \models \mathtt{S}(a,b)$ iff $b \in leaves(mtree(a,\sigma))$

$\mathcal{G}^{\sim}, G^\sigma \models \neg\varphi$ iff $\mathcal{G}^{\sim}, G^\sigma \not\models \varphi$

$\mathcal{G}^{\sim}, G^\sigma \models \varphi_1 \wedge \varphi_2$ iff $\mathcal{G}^{\sim}, G^\sigma \models \varphi_1$ and $\mathcal{G}^{\sim}, G^\sigma \models \varphi_2$

$\mathcal{G}^{\sim}, G^\sigma \models K_a\varphi$ iff for all H^τ such that $G^\sigma \sim_a H^\tau, \mathcal{G}^{\sim}, H^\tau \models \varphi$.

For a formula φ, we define the algorithm of the *epistemic gossip protocol* as follows. In this study, we assume that all agents follow the same protocol.

Definition 3.12 (Gossip protocol for u). A gossip protocol is a nondeterministic algorithm of the following form:

`repeat` forever
  `select` agent $v \in A$ such that condition $\varphi(u,v)$ is satisfied
  `execute` call uv

where $\varphi(u,v)$ is a formula defined in the language introduced in Definition 3.10.

Given a protocol P and an initial gossip graph G, the event sequences that may occur according to the protocol are called P$^{\sim}$-*permitted sequences*. The set of P$^{\sim}$-permitted sequences is called the extension of the protocol.

Definition 3.13 (Permitted sequence). Let P be a protocol defined by the condition $\varphi(x,y)$, and let G be an initial gossip graph.

- A call ab is P$^{\sim}$-permitted in G^σ if $\mathcal{G}^{\sim}, G^\sigma \models \varphi(a,b)$, the call ab is valid in G^σ, and not all agents are experts in G^σ.
- A failure $[c]$ is P$^{\sim}$-permitted in G^σ if there exists a call ab that is P$^{\sim}$-permitted in G^σ.
- A sequence of events σ is P$^{\sim}$-permitted in G if each σ_{n+1} is P$^{\sim}$-permitted in $G^{\sigma|n}$.
- The extension $\mathrm{P}^{\sim}_G$ of protocol P in G is the set of sequences that are P$^{\sim}$-permitted in G.
- A sequence $\sigma \in \mathrm{P}^{\sim}_G$ is P$^{\sim}$-maximal in G if σ is either an infinite sequence or for every event e, $\sigma;\mathrm{e} \notin \mathrm{P}^{\sim}_G$.

For a given initial gossip graph, a protocol is said to be *successful* if it eventually makes all agents experts. This concept can be classified into the following four levels of strength, depending on the rigor of the requirement.

Definition 3.14 (Successful). Let G be an initial gossip graph, and let P be a protocol.

- A sequence $\sigma \in \mathrm{P}^{\sim}_G$ is said to be successful if σ is a finite sequence and all agents are experts in G^σ.
- A sequence $\sigma \in \mathrm{P}^{\sim}_G$ is said to be fair if σ is a finite sequence or, for any call xy, the following holds:
  For every $i \in \mathbb{N}$, there exists a $j \geq i$ such that if the call xy is P$^{\sim}$-permitted in $G^{\sigma|j}$, then $\sigma_j = xy$ for some $j \geq i$.
- A protocol P is said to be strongly successful in G if every maximal sequence $\sigma \in \mathrm{P}^{\sim}_G$ is successful.
- A protocol P is said to be fairly successful in G if there exists at least one maximal and successful sequence $\sigma \in \mathrm{P}^{\sim}_G$, and every fair and maximal sequence $\sigma \in \mathrm{P}^{\sim}_G$ is successful.
- A protocol P is said to be weakly successful in G if there exists at least one maximal and successful sequence $\sigma \in \mathrm{P}^{\sim}_G$.
- A protocol P is said to be unsuccessful in G if there is no successful sequence $\sigma \in \mathrm{P}^{\sim}_G$.

3.3 Protocols

Finally, we list some epistemic gossip protocols introduced in [21] below. These will be the subject of analysis in the next section.

ANY (ANY Call): $\varphi(x,y) := \top$
(While not every agent knows all secrets, randomly select a pair xy such that x knows y's number and let x call y.)

PIG (Possible Information Growth):
$\varphi(x,y) := \neg K_x \neg \bigvee_{z\in A}(\mathtt{S}(x,z) \leftrightarrow \neg\mathtt{S}(y,z))$
(Call xy can be made if x knows y's number and if x considers it possible that there is a secret known by one of x, y but not the other.)

HSS (Hear Some Secret): $\varphi(x,y) := \neg K_x \neg \bigvee_{z\in A}(\mathtt{S}(x,z) \wedge \neg\mathtt{S}(y,z))$
(Call xy can be made if x knows y's number and considers it possible that she knows some secret that y does not know.)

HMS (Hear My Secret): $\varphi(x,y) := \neg K_x \neg\neg\mathtt{S}(y,x)$
(Call xy is possible if x knows y's number and x considers it possible that y does not know x's secret.)

CO′ (Modified Call Me Once): $\varphi(x,y) := \neg K_x(\mathtt{C}(x,y) \vee \mathtt{C}(y,x))$
(Agent x may call agent y if x knows y's number and x does not know there was a prior call between x and y.)

Here, we introduce CO′ instead of the original CO by the following reason. When assuming the existence of failures, protocol CO, defined as $\varphi(x,y) := \neg\mathtt{C}(x,y) \wedge \neg\mathtt{C}(y,x)$, cannot be evaluated by the agent itself. To deal with this issue, the study [10] implicitly assumed the presence of an external agent responsible for determining the truth of the statement. However, in this paper, we modify the protocol so as to make the system fully distributed.

4 ROBUSTNESS IN EXISTING PROTOCOLS

In this section, we analyze the robustness of existing protocols against failures, which has not been clarified in previous research. In [10], necessary and sufficient conditions for the protocols ANY and PIG to be fairly successful were presented. In addition to the two protocols, we also analyze HSS, HMS and CO′. Furthermore,

(A) Existing Protocols

Theorem	Protocol	Strongly	Fairly	Weakly	Unsuccessful
4.2	ANY	N	iff weak [10]	iff weak	N
4.2	PIG	N	iff weak [10]	iff weak	N
4.1, 4.2	HSS	N	iff weak	iff weak	N
4.1, 4.2	HMS	N	iff weak	iff weak	N
4.3, 4.4	CO′	if strong	if strong	iff weak	N

(B) Extended Protocols

Theorem	Protocol	Strongly	Fairly	Weakly	Unsuccessful
6.1, 6.2, 6.4	ANY+	N	if strong	iff weak	N
6.1, 6.2, 6.4	PIG+	N	if strong	iff weak	N
6.1, 6.2, 6.4	HSS+	N	if strong	iff weak	N
6.1, 6.2, 6.4	HMS+	N	if strong	iff weak	N
6.5, 6.6	CO′+	if strong	if strong	iff weak	N
6.1, 6.2, 6.3	ANY#	N	only if non-iso.	iff weak	N
6.1, 6.2, 6.3	PIG#	N	only if non-iso.	iff weak	N
6.1, 6.2, 6.3	HSS#	N	only if non-iso.	iff weak	N
6.1, 6.2, 6.3	HMS#	N	only if non-iso.	iff weak	N
6.5, 6.7	CO′#	only if non-iso.	only if non-iso.	iff weak	N

Table 1: Results of the analysis for (A) existing protocols and (B) extended protocols.

with regard to the feasibility of the protocols, we consider the four levels of *successful* properties: strongly successful, fairly successful, weakly successful, and unsuccessful. Accordingly, we clarify the conditions on the initial gossip graph for which each of the 20 cases holds.

Our results are presented in Table 1, where "strong" ("weak", "non-iso.", respectively) represents the proposition that the initial gossip graph is strongly connected (weakly connected, non-isolated, respectively). Also, "N" represents that there are no conditions under which the property holds.

These results are derived from Theorems 4.1 and 4.2, shown below. Hereafter, we assume that the number of failures is finite.

Theorem 4.1. Protocols HSS and HMS are fairly successful in the initial gossip graph G if and only if G is weakly connected.

Proof. Assuming that the number of failures is finite, there exists a prefix σ' of σ such that no agent gains any new secrets afterward. If σ is not successful, then in $G^{\sigma'}$, there exist agents x and y such that $(x, y) \in N$ and $leaves(mtree(x, \sigma')) \neq leaves(mtree(y, \sigma'))$. However, if σ is fair, the call xy should eventually occur, leading to a contradiction. □

This argument is similar to the proof of Theorem 1 in [10], which established that the necessary and sufficient condition for protocol ANY to be fairly successful is that the initial gossip graph G is weakly connected.

Theorem 4.2. Protocols ANY, PIG, HSS, and HMS are not strongly successful in G.

Proof. Assume that ANY is strongly successful in G. It is clear that G is weakly connected. Since G is weakly connected, there exist agents a and b such that $(a, b) \in N$. In this case, $\sigma = ab; ab; \cdots$ is ANY$^{\sim}$-permitted in G, leading to a contradiction. □

From Theorems 4.1and 4.2, it follows that the protocols are weakly successful if and only if the graph is weakly connected. The reasoning is as follows: First, the necessary and sufficient condition for any of these protocols to be fairly successful is that the graph is weakly connected. Therefore, when the graph is weakly connected, the protocols are weakly successful. Moreover, if the condition on the graph is further relaxed (i.e., if the graph is not connected), there will exist agents who neither know nor are known by anyone else, making it impossible for them to call anyone or be called. Thus, any sequence executed on such a graph will be unsuccessful. Therefore, it is also necessary for the graph to be weakly connected for the protocols to be weakly successful.

Finally, we show the properties of protocol CO′. First, we show the necessary and sufficient conditions for this protocols to be weakly successful. From this theorem, we can see that if G is weakly connected then CO′ is not unsuccessful in G.

Theorem 4.3. For an initial gossip graph G, CO′ is weakly successful in G if and only if G is weakly connected.

Proof. The proof follows similarly to Theorem 15 in [22], which establishes that CO is strongly successful in G if and only if G is weakly connected. □

We show sufficient conditions for CO′ to be strongly successful in G. From this theorem, we can see that if G is strongly connected, then all maximal sequences contained in $\mathrm{CO}'^{\sim}_{G}$ is successful, and thus CO′ is fairly successful in G.

Theorem 4.4. For an initial gossip graph G, if G is strongly connected, then CO′ is strongly successful in G.

Proof. Let σ be a maximal sequence contained in $\mathrm{CO}'^{\sim}_{G}$. The failures contained in σ occur at most finitely many times. Therefore, if we denote the prefix of σ up to its last failure as σ_1, then σ_1 is a finite sequence. Now, we set $\sigma = \sigma_1; \sigma_2$, it follows from the definition of CO′ that σ_2 does not contain the same call more than once. Hence, σ is also a finite sequence.

Since G is strongly connected, there exists a directed path π from any x to y. By induction on the length of π, we can conclude that in G^{σ},$y \in leaves(mtree(x, \sigma))$. □

In closing this section, we compare our results with previous research that does not assume failures. In [21], the necessary conditions for some protocols, including those we analyzed, to be successful were clarified under models without failures. Comparing these results with ours, we find that for ANY and PIG, there was no difference regardless of whether failures were allowed. On the other hand, HSS had the property of being strongly successful when the initial gossip graph was weakly connected in the absence of failures. However, when failures were allowed, it was found that HSS could not be strongly successful under the same conditions. Similarly, for HMS, it was shown that this protocol could be strongly successful in a special graph called a sun*graph in the absence of failures, but under the same conditions with failures, it could not be strongly successful. Finally, in [21], they demonstrated that CO becomes strongly successful when the initial gossip graph is weakly connected. However, [10] showed that this result does not hold when failures are assumed. In contrast, we demonstrate that CO′ becomes strongly successful when the initial gossip graph is strongly connected.

5 CONDITIONS TO DETECT FAILURES

In this section, we analyze the conditions under which agents detect their own and others' failures during a protocol execution. More formally, in the execution (event sequence) σ of a given protocol, agent a is said to detect agent b's (where possibly $a = b$) failure if $\mathcal{G}^{\sim}, G^{\sigma} \models K_a \mathsf{F}(b)$ holds.

In [10], a sufficient condition was provided under the assumption that at most one failure occurs. In this paper, we present sufficient conditions under the assumption that multiple failures may occur. Additionally, under the restriction that at most one failure occurs, we also provide a necessary condition for an agent to detect a failure. Our proof strategy is as follows: we first prove the necessary and sufficient conditions for detecting a failure under the assumption that at most one failure occurs (by Theorem 5.1) and then show the sufficient condition also holds for the case of multiple failures (by Theorem 5.2).

The mechanism by which agent b detects a's failure is as follows. First, a makes a call to c ($\neq a, b$), then a fails, and later a makes a call to d ($\neq a, b, c$). At this point, the memory tree that a sent to d does not contain the record of the call ac. Therefore, when the two memory trees that a sent to c and d eventually reach b via other agents, b can detect a's failure from the inconsistency between these two messages originating from a. By these theorems, we formally prove that this mechanism works.

As shown in the following section, Theorem 5.2 is used to analyze the robustness of a protocol in which an agent who detects a failure repairs it. This raises one question: whether higher-order knowledge regarding the failure holds, specifically, whether agent a can know that agent b knows about c's failure. If this holds, it could be possible to design an advanced protocol in which, for instance, a instructs b to repair the data lost by c using such higher-order knowledge. However, Theorem 5.3 shows that this is not possible. That is, higher-order knowledge about failures does not hold.

Before proving the theorems, we first provide the definitions of several necessary concepts. To prove the theorems presented below, we will need nine lemmas (Lemmas 4.1 to 4.9). However, due to space limitations, their statements and proofs are provided only in the Appendix (Supplementary Material).

Definition 5.1 (Single memory). Let t be a memory tree of height at least 1, and let a be an agent. If one of the children of t is $\langle a \rangle$, then t is said to have a *single memory* of a.

Definition 5.2. Let t be a memory tree of height at least 1, and let a be an agent. The set $sub_a(t)$ is defined as $sub_a(t) = \{t' \mid t' \subseteq t \text{ and } t' \text{ has a single memory of } a\}$.

Definition 5.3. Let a be an agent and t a memory tree. If $sub_a(t)$ contains two or more elements, then t is said to have multiple single memories of a.

We now present the necessary and sufficient conditions for identifying a single failure.

Theorem 5.1. Let σ be a sequence containing at most one failure, and let x, a be agents, with an asynchronous gossip model $\mathcal{G}^{\sim}$ and an initial gossip graph G. Then $\mathcal{G}^{\sim}, G^{\sigma} \models K_x \mathsf{F}(a)$ if and only if $mtree(x, \sigma)$ contains multiple single memories of a.

Proof. ($\Rightarrow$) Assume $\mathcal{G}^{\sim}, G^{\sigma} \models K_x \mathsf{F}(a)$. Write $\sigma = \sigma_1; [a]; \sigma_2$. Suppose $mtree(x, \sigma)$ does not contain multiple single memories of a.

(Case 1) If $|sub_a(mtree(x, \sigma_1; [a]; \sigma_2))| = 0$, and $x \neq a$, then $\langle a \rangle \notin leaves(mtree(x, \sigma_1; [a]; \sigma_2))$. Since x, who does not know a's secret, does not know that a call involving a occurred before a's failure (by Lemma 5.3), $G^{\sigma_1;[a];\sigma_2} \sim_x G^{\sigma_1{}^{-a};[a];\sigma_2}$. Therefore, since no agent can know about the failure of a who has never made a call (by Lemma 5.1), $G^{\sigma_1{}^{-a};[a];\sigma_2} \sim_x G^{\sigma_1{}^{-a};\sigma_2}$, which leads to a contradiction.

(Case 2) If $|sub_a(mtree(x, \sigma_1; [a]; \sigma_2))| = 1$, let t be the memory tree containing a's single memory in $mtree(x, \sigma)$. Since any node that is not a leaf in $mtree(x, \sigma)$ is included in σ (by Lemma 5.6), $root(t) \in \sigma_1; [a]; \sigma_2$.

(Case 2.1) If $root(t) \in \sigma_1$ and $x = a$, then $a \in \sigma_2$. Let the first call involving a be ab, and we set $\sigma_2 = \tau_1; ab; \tau_2$. In this setting, $mtree(a, \sigma_1; [a]; \tau_1; ab)$ contains a single memory of a. Therefore, since $t = mtree(a, \sigma_1; [a]; \tau_1; ab)$, it follows that $ab \in \sigma_1$. Hence, we can write $\sigma_1 = \upsilon_1; ab; \upsilon_2$, and $mtree(b, \upsilon_1; ab)$ contains a memory tree with a's single memory. Therefore, $mtree(a, \sigma_1; [a]; \tau_1; ab)$ contains multiple single memories of a, which is a contradiction. Next, assume $x \neq a$. Here, generally, for $mtree(x, \sigma_1; [a]; \sigma_2)$, if it contains only one memory tree with a's single memory, and the root of that memory tree is included in σ_1, then an agent x does not know that a call involving a has occurred (by Lemma 5.9). Therefore, since no agent can detect failure of the agent who has never made a call before or will never make a call in the future (by Lemma 5.2), $G^{\sigma_1;[a];\sigma_2} \sim_x G^{\sigma_1;[a];\sigma_2{}^{-a}}$, which leads to a contradiction.

(Case 2.2) Assume that $root(t) \notin \sigma_1$. For $mtree(x, \sigma_1; [a]; \sigma_2)$, generally, if it contains only one memory tree with a's single memory, and the root of that memory tree is not included in σ_1, then x does not know that a call involving a has occurred (by Lemma 5.8). So, $G^{\sigma_1;[a];\sigma_2} \sim_x G^{\sigma_1{}^{-a};[a];\sigma_2}$. Therefore, since no agent can know about the failure of a who has never made a call (by Lemma 5.1), $G^{\sigma_1{}^{-a};[a];\sigma_2} \sim_x G^{\sigma_1{}^{-a};\sigma_2}$, which leads to a contradiction.

($\Leftarrow$) Assume $|sub_a(mtree(x, \sigma))| \geq 2$. If $\mathcal{G}^{\sim}, G^{\sim} \models \neg K_x \mathsf{F}(a)$, then there exists some G^{τ} such that $G^{\sigma} \sim_x G^{\tau}$ and $\mathcal{G}^{\sim}, G^{\tau} \models \neg \mathsf{F}(a)$. In $mtree(x, \tau)$, there exist two distinct memory trees, t_1 and t_2, each containing a single memory of a. Here, any memory tree contained in $mtree(x, \tau)$ that includes the single memory of a can be represented using a prefix of τ (by Lemma 5.5). Therefore, for some prefixes τ_1 and τ_2 of τ, it holds that $t_1 = mtree(a, \tau_1)$ and $t_2 = mtree(a, \tau_2)$.

Next, we distinguish cases based on the inclusion relation between τ_1 and τ_2. Here, we show only the case where $\tau_1 \sqsubset \tau_2$. If $\tau_1 \sqsubset \tau_2$, then $mtree(a, \tau_1) \subseteq mtree(a, \tau_2)$. In this case, since $t_2 = mtree(a, \tau_2)$ contains a single memory of a, we must have $a \in \tau_2$. Let the last call in τ_2 involving a be ab, and write $\tau_2 = \upsilon_1; ab; \upsilon_2$. Then, $mtree(a, \tau_2) = \langle \langle a \rangle, ab, mtree(b, \upsilon_1) \rangle$, thus $mtree(a, \tau_1) \subseteq mtree(b, \upsilon_1)$. Thus, there exists some $\tau_3 \subseteq \upsilon_1$ such that $t_1 = mtree(a, \tau_3)$. Therefore, $t_1 = mtree(a, \tau_3) \subseteq mtree(a, \upsilon_1) = \langle a \rangle$, which is a contradiction. □

The proof of Theorem 5.1 ($\Leftarrow$) remains valid even when the number of failures is allowed to be any finite number. Therefore,

this proof provides sufficient conditions for failure identification in more general cases as follows.

Theorem 5.2. Let σ be a sequence, and let x and a be agents, with an asynchronous gossip model $\mathcal{G}^{\sim}$ and an initial gossip graph G. If $mtree(x, \sigma)$ contains multiple single memories of a, then $\mathcal{G}^{\sim}, G^{\sigma} \models K_x \mathsf{F}(a)$.

Proof. In the proof of Theorem 5.1 ($\Leftarrow$), the assumption that the sequence contains only one failure was not used. Thus, this theorem can be proven using the same argument. □

Finally, we show that the higher-order knowledge regarding agent failures does not hold.

Theorem 5.3. Let G be an initial gossip graph, and let σ be any sequence, with agents x, y, and a ($x \neq y$). Then $\mathcal{G}, G^{\sigma} \models \neg K_x K_y \mathsf{F}(a)$.

Proof. Suppose there exists a gossip graph G^{σ} such that $\mathcal{G}, G^{\sigma} \models K_x K_y \mathsf{F}(a)$. Then, for any gossip graph G^{τ} such that $G^{\sigma} \sim_x G^{\tau}$, it must hold that $\mathcal{G}, G^{\tau} \models K_y \mathsf{F}(a)$. Hence, since $G^{\sigma} \sim_x G^{\sigma;[y]}$, we have $\mathcal{G}, G^{\sigma;[y]} \models K_y \mathsf{F}(a)$. In other words, for any gossip graph G^{υ} such that $G^{\sigma;[y]} \sim_y G^{\upsilon}$, it must hold that $\mathcal{G}, G^{\upsilon} \models \mathsf{F}(a)$. However, taking $\upsilon = \epsilon$, we clearly have $\mathcal{G}, G^{\epsilon} \models \neg \mathsf{F}(a)$, which is a contradiction. □

6 FAILURE-AWARE ROBUST PROTOCOL DESIGN

This section presents a method to designing robust protocols against failures. The basic idea is that when an agent notices either its own failure or the failure of another agent, it takes actions to repair the data lost due to the failure.

6.1 Extended protocols

In this study, we consider the following behaviors in two cases: when an agent detects its own failure and when it detects the failure of another agent.

Self recovery: when x notices its own failure, if it does not possess the secret of $y (\neq x)$, it calls y to obtain the missing secret.

External recovery: when x notices the failure of $y (\neq x)$, if x is an expert, it calls y to share its secret with y.

These actions are only taken when a failure is detected, and otherwise, the agent follows a given protocol π. For any protocol π, we can consider the following extended protocols: π+ (self-recovery) and π# (external recovery).

π+: $\varphi(x, y) := (\neg K_x \mathsf{F}(x) \wedge \psi(x, y)) \vee (K_x \mathsf{F}(x) \wedge \neg \mathsf{S}(x, y))$

π#: $\varphi(x, y) := (\neg K_x \mathsf{F}(x) \wedge \psi(x, y)) \vee (K_x \mathsf{F}(y) \wedge \bigwedge_{z \in A} \mathsf{S}(x, z))$

where $\psi(x, y)$ is the definition of π.

6.2 Robustness Analysis

Below, we consider the ten protocols obtained by extending the five existing protocols with both + and #, and analyze their robustness using Theorem 5.2. The results are shown in Table 1 and are derived from the following theorems.

Theorem 6.1. Protocols ANY+, PIG+, HSS+, HMS+, ANY#, PIG#, HSS#, HMS# are not strongly successful in G.

Proof. The proof follows similarly to Theorem 4.2. □

Theorem 6.2. ANY+, ANY#, PIG+, PIG#, HSS+, HSS#, HMS+ and HMS# are weakly successful if and only if G is weakly connected.

Proof. ($\Leftarrow$) We will show the theorem only for PIG+. First, we construct the event sequence σ using the following procedure. For any $a, b \in A$ (with $a \neq b$), since G is weakly connected, there exists an undirected path (called π_{ba}) from b to a. Let the agents on π_{ba} be ordered as $b, b_1, b_2, \ldots, b_m, a$. For b and b_1, either $(b, b_1) \in N$ or $(b_1, b) \in N$ holds. If the former holds, the first event of σ is bb_1; otherwise it is $b_1 b$. Next, as the second event of σ, we append a call $b_1 b_2$ or $b_2 b_1$ following the same procedure. This process is repeated until a, and we refer to the resulting sequence as τ). Furthermore, for any agent $c \neq a, b$, since there also exists an undirected path π_{ca} from c to a, we append a similar sequence of calls for the agents on π_{ca}, ordered as $c, c_1, c_2, \ldots, c_{m'}, a$, to τ.

After repeating this until a becomes an expert, a will call everyone. The sequence constructed in this way is called σ. We will show by induction on n that σ_{n+1} is PIG+$^{\sim}$-permitted on $G^{\sigma|n}$.

(B. C.) Let $\sigma_1 = kl$. Since σ_1 is valid in G and $\mathcal{G}, G^{\sigma|0} \models \mathsf{S}(k, k) \wedge \neg \mathsf{S}(l, k)$, it follows that $\mathcal{G}, G^{\sigma|0} \models \neg K_x \neg \bigvee_{z \in A} (\mathsf{S}(k, z) \leftrightarrow \neg \mathsf{S}(l, z))$. Thus, σ_1 is PIG+$^{\sim}$-permitted in $G^{\sigma|0}$.

(I. S.) Let $\sigma_{n+1} = kl$. In this case, $G^{\sigma|n} \sim_k G^{\sigma|n;[l]}$ holds, and $\mathcal{G}, G^{\sigma|n;[l]} \models \mathsf{S}(k, k) \wedge \neg \mathsf{S}(l, k)$. Thus, since $\mathcal{G}, G^{\sigma|n} \models \neg K_x \neg \bigvee_{z \in A} (\mathsf{S}(k, z) \leftrightarrow \neg \mathsf{S}(l, z))$, it follows that σ_{n+1} is PIG+$^{\sim}$-permitted in $G^{\sigma|n}$.

Therefore, σ is PIG+$^{\sim}$-permitted on G.

($\Rightarrow$) Assume that PIG+ is weakly successful in G. If G is not weakly connected, then for some agent x and any sequence σ contained in PIG+$^{\sim}_{G}$, $x \notin \sigma$ holds. Therefore, PIG+ would be unsuccessful in G, leading to a contradiction. □

Furthermore, from Theorem 6.2, we can see that if G is weakly connected, then PIG+$^{\sim}_{G}$ contains a successful sequence. Therefore, if G is weakly connected, we can demonstrate that PIG+ and similar protocols are not unsuccessful in G.

Next, we present the necessary conditions for protocols with # to be fairly successful.

Theorem 6.3. If PIG#, ANY#, HSS#, and HMS# are fairly successful in G, then G is a non-isolated.

Proof. Here, we consider PIG#. Assume that G is not a non-isolated graph and that PIG# is fairly successful. For an isolated agent a in G, we consider the following sequence. First, for an agent $k_1 \neq a$ and an agent $l_1 \neq a$ such that $(k_1, l_1) \in N$, let the sequence be $\pi_{k_1} = [k_1]; k_1 l_1; [k_1]; k_1 l_1$. Next, for an agent $k_2 \neq a, k_1$ and an agent $l_2 \neq a$ such that $(k_2, l_2) \in N$, let the sequence be $\pi_{k_2} = [k_2]; k_2 l_2; [k_2]; k_2 l_2$. Continuing in this manner, for every agent $k \neq a$, we consider the sequence π_k and concatenate them all into a sequence $\pi_1; \cdots; \pi_n$, which we denote as ρ.

Here, for any n, we demonstrate that ρ_{n+1} is PIG#$^{\sim}$-permitted in $G^{\rho|n}$ as follows.

(B. C.) ρ_0 is a failure. In this case, if any call in $G^{\rho|0}$ is not PIG#$^{\sim}$-permitted, then since $\rho|0$ is fair and maximal, this contradicts the assumption that PIG# is fairly successful.

(I. S.) If ρ_{n+1} is a failure, the argument is almost the same as (B. C.). Let ρ_{n+1} be a call, and denote it as $\rho_{n+1} = kl$. Then, we have $G^{\rho|n} \sim_k G^{\rho|n;[l]}$, from which we can derive $\mathcal{G}, G^{\rho|n} \models \neg K_k \neg \bigvee_{z\in A}(\mathsf{S}(k,z) \leftrightarrow \neg\mathsf{S}(l,z))$. Additionally, since $G^{\rho|n} \sim_k G^{\epsilon}$, we obtain $\mathcal{G}, G^{\rho|n} \models \neg K_k F(k)$.

At this point, for any agent $m \neq a$, there exist an agent o, a sequence σ_1, and a sequence σ_2 that does not contain the failure of m, such that $\sigma_1; [m]; mo; [m]; mo; \sigma_2 = \rho$. Since σ_2 does not contain the failure of m, $mtree(m, \rho)$ contains multiple instances of the single memory of m. Thus, by Theorem 5.2, $\mathcal{G}, G^{\rho} \models K_m \mathsf{F}(m)$. Moreover, since ρ does not include any calls involving a, m is not an expert in G^{ρ}. Thus, $\mathcal{G}, G^{\rho} \models K_m \mathsf{F}(m) \wedge \bigvee_{z\in A} \neg\mathsf{S}(m,z)$. Additionally, in G^{ρ}, a does not know the phone number of any agent, so no call is PIG#$^{\sim}$-permitted. Therefore, ρ is PIG#$^{\sim}$-maximal, fair, and unsuccessful, leading to a contradiction. □

Theorem 6.4. If the initial gossip graph G is strongly connected, then ANY+, PIG+, HSS+, and HMS+ are fairly successful.

Proof. We only show the proof for ANY+. Consider any sequence σ that is ANY+$^{\sim}$-permitted, fair, and maximal in G. Assume σ is an infinite sequence. For some $\tau(\sqsubset \sigma)$, for any $\tau \sqsubseteq \tau' \sqsubset \sigma$, and for any agent x, we have $leaves(mtree(x, \tau)) = leaves(mtree(x, \tau'))$. At this point, for some y, let z be an agent not included in $leaves(mtree(y, \tau))$. Since G is strongly connected, there exists a directed path π from y to z. Let the agents included in π be $y, z_0, z_1, \cdots, z$. If $\mathcal{G}, G^{\tau} \models \neg(y, z_0)$, then for any $v(\sqsubseteq v' \sqsubset \sigma)$, we have $\mathcal{G}, G^{v} \models \neg\mathsf{S}(y, z_0)$. Since σ is fair, τ must include yz, leading to a contradiction. Therefore, $\mathcal{G}, G^{\tau} \models \mathsf{S}(y, z_0)$, and $(y, z_1) \in N^{\tau}$. Continuing in this manner, we can show that $\mathcal{G}, G^{\tau} \models \mathsf{S}(y, z)$, leading to a contradiction.

Thus, σ is a finite sequence. If there exists a v such that some agent w is not included in $leaves(mtree(v, \tau))$, we can derive a similar contradiction. Therefore, σ is successful. □

Next, we demonstrate the properties of protocol CO′ and the derived protocols CO′+ and CO′#. First, as Theorem 4.3, we show the necessary and sufficient conditions for these protocols to be weakly successful. From this theorem, we can see that if G is weakly connected, CO′+ and CO′# are not unsuccessful in G.

Theorem 6.5. For an initial gossip graph G, CO′+ and CO′# are weakly successful in G if and only if G is weakly connected.

Proof. Similar to the proof of Theorem 4.3. □

We show sufficient conditions for CO′+ to be strongly successful in G. Similar to the case of original CO′ (i.e., Theorem 4.4), by the following theorem, we can see that if G is strongly connected, then all maximal sequences contained in CO′+$^{\sim}_G$ is successful, and thus CO′+ is fairly successful in G.

Theorem 6.6. For an initial gossip graph G, if G is strongly connected, then CO′+ is strongly successful in G.

Proof. Similar to the proof of Theorem 4.4. □

Finally, we present the necessary condition for CO′# to be fairly successful. From this theorem, the existence of a maximal and unsuccessful sequence in CO′#$^{\sim}_G$ is established, which also provides the necessary condition for CO′# to be strongly successful in G.

Theorem 6.7. If CO′# is fairly successful in G, then G is non-isolated.

Proof. It suffices to show that the sequence ρ defined in Theorem 6.3 is CO′#$^{\sim}$-permitted. We prove this by the same induction as in Theorem 6.3.

(B. C.) Same as in Theorem 6.3.

(I. S.) If ρ_{n+1} is a failure, the proof follows the same reasoning as in Theorem 6.3. Let ρ_{n+1} be a call and denote it as $\rho_{n+1} = kl$. Since $(k, l) \in N$, the call kl is valid in $G^{\rho|n}$. Moreover, since ρ_n is a failure $[k]$, agent k is not an expert in $G^{\rho|n}$. Thus, $G^{\rho|n} \sim_k G^{\epsilon}$ holds, and $\mathcal{G}, G^{\epsilon} \models \neg\mathsf{C}(k, l) \wedge \neg\mathsf{C}(l, k)$, which implies that $\mathcal{G}, G^{\rho|n} \models \neg K_k(\mathsf{C}(k, l) \vee \mathsf{C}(l, k))$. □

6.3 Discussion

As shown in Table 1, the feasibility of the protocols extended with the recovery behavior has actually worsened compared to the original protocols. The reason for this is that the definitions of + and # are designed to minimize unnecessary calls. More specifically, for +, it may be possible to allow calls not only to agents who do not know the secret but also to more agents when an agent detects its own failure. Similarly, for #, it may be possible to allow calls when an agent detects a failure, even if the agent is not yet an expert. In general, in gossip protocols (not just epistemic gossip), there is a trade-off between the frequency of calls within the system and the robustness or speed of information sharing. Although our current results did not lead to an improvement in the robustness of existing protocols, adjusting + and # to increase the frequency of calls could solve this issue, which we plan to investigate in future research.

7 CONCLUSION

In this paper, we analyzed the robustness of epistemic gossip protocols against failures, with a particular focus on failures that cause data loss, as introduced in the prior work [10].

We clarified the conditions on the initial gossip graph under which each of the five existing protocols can achieve one of the four levels of "success" properies. Additionally, we provided sufficient conditions for agents to detect failures by relaxing assumptions made in previous studies, and we also proved necessary conditions. Furthermore, using these theorems, we proposed a new protocol that exhibits behavior to recover lost data when agents detect failures and analyzed its robustness. While these new protocols did not improve the robustness of the existing ones, we plan to investigate the usefulness of this approach for designing robust protocols in future work.

As future research directions, we are particularly interested in the communication complexity of protocols that assume the presence of unreliable agents. We also aim to further clarify the robustness of both other existing protocols and new protocols based on them. Furthermore, we plan to analyze the robustness against failures other than data loss, such as modifications to the content of secrets.

ACKNOWLEDGMENTS

We are grateful to Professor Yukiyoshi Kameyama for his valuable comments on this paper.

REFERENCES

[1] Krzysztof R. Apt, Davide Grossi, and Wiebe van der Hoek. 2016. Epistemic Protocols for Distributed Gossiping. *Electronic Proceedings in Theoretical Computer Science* 215 (Jun 2016), 51–66.

[2] Krzysztof R Apt and Dominik Wojtczak. 2017. Common knowledge in a logic of gossips. *arXiv preprint arXiv:1707.08734* (2017).

[3] Krzysztof R Apt and Dominik Wojtczak. 2018. Verification of distributed epistemic gossip protocols. *Journal of Artificial Intelligence Research* 62 (2018), 101–132.

[4] Maduka Attamah, Hans Van Ditmarsch, Davide Grossi, and Wiebe van der Hoek. 2014. Knowledge and Gossip. In *ECAI*. 21–26.

[5] Brenda Baker. 1972. Gossips and Telephones. *Discrete Mathematics* 2 (1972), 191–193.

[6] Tushar Deepak Chandra and Sam Toueg. 1996. Unreliable failure detectors for reliable distributed systems. *Journal of the ACM (JACM)* 43, 2 (1996), 225–267.

[7] Pierre Civit, Seth Gilbert, and Vincent Gramoli. 2021. Polygraph: Accountable byzantine agreement. In *2021 IEEE 41st International Conference on Distributed Computing Systems (ICDCS)*. IEEE, 403–413.

[8] Martin C Cooper, Andreas Herzig, Faustine Maffre, Frédéric Maris, and Pierre Régnier. 2019. The epistemic gossip problem. *Discrete Mathematics* 342, 3 (2019), 654–663.

[9] Ronald Fagin, Yoram Moses, Joseph Y. Halpern, and Moshe Y. Vardi. 1995. *Reasoning about knowledge*. MIT press.

[10] Kosei Fujishiro and Koji Hasebe. 2020. Robustness and Failure Detection in Epistemic Gossip Protocols. In *Formal Methods and Software Engineering: 22nd International Conference on Formal Engineering Methods, ICFEM 2020, Singapore, Singapore, March 1–3, 2021, Proceedings 22*. Springer, 20–35.

[11] Andreas Haeberlen, Petr Kouznetsov, and Peter Druschel. 2007. PeerReview: Practical accountability for distributed systems. *ACM SIGOPS operating systems review* 41, 6 (2007), 175–188.

[12] András Hajnal, Eric C Milner, and Endre Szemerédi. 1972. A cure for the telephone disease. *Canad. Math. Bull.* 15, 3 (1972), 447–450.

[13] Sandra M. Hedetniemi, Stephen T. Hedetniemi, and Arthur L. Liestman. 1988. A survey of gossiping and broadcasting in communication networks. *Networks* 18, 4 (1988), 319–349.

[14] Kim Potter Kihlstrom, Louise E Moser, and P Michael Melliar-Smith. 2003. Byzantine fault detectors for solving consensus. *Comput. J.* 46, 1 (2003), 16–35.

[15] Dahlia Malkhi and Michael Reiter. 1997. Unreliable intrusion detection in distributed computations. In *Proceedings 10th Computer Security Foundations Workshop*. IEEE, 116–124.

[16] Robert Tijdeman. 1971. On a telephone problem. *Nieuw Archief voor Wiskunde* 3, 19 (1971), 188–192.

[17] Line van den Berg and Malvin Gattinger. 2020. Dealing with unreliable agents in dynamic gossip. In *International Workshop on Dynamic Logic*. 51–67.

[18] Hans Van Ditmarsch, Malvin Gattinger, and Rahim Ramezanian. 2023. Everyone knows that everyone knows: Gossip protocols for super experts. *Studia Logica* 111, 3 (2023), 453–499.

[19] Hans Van Ditmarsch, Davide Grossi, Andreas Herzig, Wiebe van Der Hoek, and Louwe B Kuijer. 2016. Parameters for epistemic gossip problems. In *LOFT 2016-12th Conference on Logic and the Foundations of Game and Decision Theory*.

[20] Hans van Ditmarsch, Wiebe van Der Hoek, and Louwe B Kuijer. 2020. The logic of gossiping. *Artificial Intelligence* 286 (2020), 103306.

[21] Hans van Ditmarsch, Jan van Eijck, Pere Pardo, Rahim Ramezanian, and François Schwarzentruber. 2017. Epistemic protocols for dynamic gossip. *Journal of Applied Logic* 20 (2017), 1–31.

[22] Hans van Ditmarsch, Jan van Eijck, Pere Pardo, Rahim Ramezanian, and François Schwarzentruber. 2019. Dynamic gossip. *Bulletin of the Iranian Mathematical Society* 45, 3 (2019), 701–728.