Robustness and Failure Detection in Epistemic Gossip Protocols

Kosei Fujishiro and Koji Hasebe

Department of Computer Science, University of Tsukuba
1-1-1, Tennodai, Tsukuba 305-8573, Japan
fujishiro@mas.cs.tsukuba.ac.jp, hasebe@cs.tsukuba.ac.jp

Abstract. Gossip problem is an information dissemination problem in which networked agents (nodes) must share their secrets by the minimum number of calls. In recent years, to solve the problem, various epistemic gossip protocols have been proposed, where the agents decide who to call based on the higher-order knowledge about the possession of secrets. Although most previous studies on the epistemic gossip protocol have restricted their scope to the environments including only reliable agents, from the practical viewpoint, it is worthwhile investigating robust protocols against agent failure. In this paper, we assume the existence of unreliable agents and analyze the robustness of some existing protocols using epistemic logic. In our model, when an agent fails, it loses the secrets and telephone numbers gained by previous calls and returns to its initial state. In addition, during each call, agents share not only their possessing secrets but also the history of the transmission path of each secret. For these settings, we show that the protocols ANY and PIG are successful (i.e., the protocols always lead to the state where every agent knows all secrets). We also show that the protocol CO is not immediately successful under the assumption that agents can fail, but it becomes successful if the protocol execution satisfies some conditions. Furthermore, we clarify sufficient conditions for agents to detect the failure of other agent, which are useful for designing robust protocols.

1 Introduction

A gossip protocol determines a one-to-one communication (call) to share secrets among networked agents (nodes). This network is represented as a directed graph, where the edge from agent a to agent b indicates the relation that a knows the telephone number of b, i.e. a can call b. Initially, each agent only knows its own secret, and during each call, two agents share their secrets gained by previous calls. One of the main challenges with the gossip protocol is to find the shortest sequence of calls that leads to everyone knowing all secrets. The minimum length of sequence depends on the initial graph, and according to the early studies [11, 9], it is $2n - 4$ for the complete graph, and $2n - 3$ for the other connected graphs for agents $n > 3$.

Recently, to solve the problem, various epistemic gossip protocols have been proposed [6, 5]. In these protocols, an agent decides who to call based not only

on the agent's knowledge about its possessing secrets and telephone numbers, but also on higher-order knowledge (i.e., the knowledge that the agent has about the knowledge of other agents). These studies also analyzed the conditions for the protocols to be successful (i.e., the protocols must achieve the state where every agent knows all secrets) using epistemic logic [8]. The results provide useful information for designing epistemic gossip protocols. However, most previous studies on epistemic gossip protocols have been limited to models consisting of only reliable agents, which were not realistic from a practical viewpoint. Therefore, to enhance the reliability of the protocols, it is worthwhile investigating the robustness against agent failure.

In this paper, we assume the existence of unreliable agents and analyze the robustness of some existing protocols in such a model using epistemic logic. Specifically, here we focus on the dynamic gossip protocols [6], where agents exchange not only their possessing secrets but also their telephone numbers when making a call. In our proposed model, when an agent fails, it loses the secrets previously learnt and returns to its initial state. In addition, during each call, agents share not only the secrets but also the history of the transmission path of each secret.

For these settings, we analyzed the protocols ANY (ANY call), PIG (Possible Information Growth), and CO (Call me Once), which were originally introduced by [6]. For the former two, we prove that these are successful even in the environment where agents can fail. On the other hand, for the latter protocol, we prove that it is not immediately successful, but it becomes successful if its execution satisfies some conditions. Furthermore, we clarify sufficient conditions for agents to detect the failure of other agents.

The contribution of this paper is twofold. First, we give a new model to analyze the robustness of epistemic gossip protocols. Second, we demonstrate a logical analysis of some epistemic gossip protocols and prove properties about the robustness and failure detection. Although we do not present a concrete robust protocol, our method is still useful in protocol verification and design.

The structure of this paper is as follows. Section 2 presents related work. Section 3 defines our model and the logic used for the protocol analysis. Section 4 shows some properties about robustness and failure detection in some existing epistemic gossip protocols. Finally, Section 5 concludes the paper and presents possible future directions.

2 Related Work

The earliest studies of the gossip problem date back to the 1970s [11, 9]. (See also [10] for a survey summarizing results on the classic gossip problem and its variants as well as techniques to schedule centralized optimal calls.)

In recent years, as an approach to the gossip problem, there have been a number of studies on the epistemic gossip protocol. Attamah et al. [2] have proposed a method for the autonomous distributed control of calls by epistemic gossip protocol and investigated its various extensions. Van Ditmarsch et al.

[6, 7] have presented some epistemic gossip protocols with dynamic setting, in which agents exchange both their secrets and telephone numbers when making a call. In [6, 7], they introduced some successful protocols named ANY, CO, LNS, and PIG. Our model and analysis presented in this paper are extensions of the method proposed in the studies [6, 7].

Apt et al. [1] provides a framework for formally analyzing the validity and termination of the epistemic gossip protocol of the merge-then-learn method and presents the protocol for complete graphs and ring graphs. On the other hand, in our study and in [6], the learn-then-merge method is adopted in which the information sent by each agent is acquired and then merged.

Cooper et al. [5] have extended the gossip problem by paying attention to epistemic depth. Specifically, they set a state about higher-order knowledge as a goal and investigate the optimal call scheduling for it. Unlike our research, their focus is on centralized scheduling.

Similar to our research, van den Berg [3] has assumed the existence of unreliable agents in the setting of dynamic gossip and investigated their effects and how to identify them. However, as a possible application, unlike our research, the prevention of the spread of fake news is emphasized. Therefore, the modeling of agent "unreliability" is different with ours. In that setting, there is a result that is similar to the one in this research that the existing protocol called LNS will not succeed owing to the existence of unreliable agents. It also describes the counterintuitive and interesting results about the difficulty of identifying unreliable agents.

This paper also analyzes the failure detection in an environment where agents can fail. Failure detection is a central research issue for ensuring the reliability of distributed systems. As an early important study, Chandra and Toueg [4] have argued the importance of failure detectors in distributed systems. This study defines two classes of failure detectors with the notions of completeness and correctness, and shows how they can be used to solve consensus problems in asynchronous systems where crash failures can occur.

3 Basic Concepts

In this section, we define our proposed model including unreliable agents and introduce epistemic logic used in protocol specification and analysis.

3.1 Modeling Agent Failure

Our model is obtained by adding events of agent failure to the model defined by [6]. We first define the set of events as follows.

Definition 1 (Event). Let xy denote the call from agent x to agent y and $[x_1 \dots x_k]$ denote the simultaneous failure of agents $x_1, \dots, x_k$. We write either xy or yx as $\overline{xy}$. A call or an agent's failure is called an event. Let C and F be the sets of calls and failures, respectively. The set of events E is defined as $\mathsf{E} = \mathsf{C} \cup \mathsf{F}$. We write $x \in \mathsf{e}$ to denote that agent x is involved in the event e.

Protocol execution is modeled as a sequence of events defined below.

Definition 2 (Event sequence). The expression $\mathsf{e}_1;\mathsf{e}_2;\ldots;\mathsf{e}_n$ is used to denote the sequence $\mathsf{e}_1,\mathsf{e}_2,\ldots,\mathsf{e}_n$ of n events. E^* is the set of all event sequences. Throughout we also use some notations defined below.

- Empty sequence is denoted by ϵ.
- Semi-colon is also used to concatenate events or event sequences.
- $\sigma \sqsubseteq \tau$ indicates that either σ is a prefix of τ or $\sigma = \tau$.
- σ_n is used to denote the n-th event in the sequence of $\sigma \in \mathsf{E}^*$
- $\sigma|n$ is used to denote the prefix of σ up to the n-th event.
- For each $x \in A$, σ_x is used to denote the subsequence of σ consisting of all σ_n which x is involved in.

In [6], during a call, agents exchange their secrets and telephone numbers gained by previous calls, while in our model, we assume that the history of the transmission path of each secret is also exchanged. This history is represented by a tree structure, called a memory tree, and is defined below.

Definition 3 (Memory tree). A memory tree is a binary tree defined as follows.

- Base Case: $\langle x \rangle$ is a memory tree for any $x \in A$.
- Ind. Step: if T and T' are memory trees, then $\langle T, xy, T' \rangle$ is a memory tree for any call xy.

We denote the root of a memory tree T by $r(T)$ and the set of all leaves by $\mathit{leaves}(T)$. For the memory tree T_x of agent x, $\mathit{leaves}(T_x)$ is the set of secrets owned by x. For $T = \langle T_1, xy, T_2 \rangle$, we define T_L and T_R as $T_L = T_1$ and $T_R = T_2$, respectively. We denote that T is a subtree of T' by $T \subseteq T'$.

Next, we define the gossip graph. The gossip graph is a directed graph indicating the agents' knowledge about their telephone numbers at a certain point in time. Formally, a gossip graph consists of a set A of agents, a binary relation N over A representing the agents' knowledge about telephone numbers, and a class $\{T_x\}_{x \in A}$ each of which represents the set of memory trees stored by an agent. Similar to the definition of [6], $(x, y) \in N$ means that x knows the telephone number of y. Throughout, we also use the notations Nxy and N_x to denote $(x, y) \in N$ and $\{y \in A \mid Nxy\}$, respectively.

Definition 4 (Gossip graph). A gossip graph is a tuple $G = (A, N, \{T_x\}_{x \in A})$, where A is a finite set of agents, N is a subset of $A \times A$ and T_x is a memory tree belonging to x. A gossip graph which satisfies $T_x = \langle x \rangle$ and $(x, x) \in N$ for any x is called an initial gossip graph. A gossip graph is weakly connected if in the graph (A, N) for any $x, y \in A$ there exists a directed path from x to y. A gossip graph is complete if $N = A \times A$. Agent x is called expert if $\mathit{leaves}(T_x) = A$.

A gossip graph that represents the initial state is called the initial gossip graph. Along with a protocol execution, a gossip graph starts with one of the initial gossip graphs and changes every time an event occurs. Intuitively, when a call occurs between agents, all information contained in each other's memory tree is shared, whereas when an agent fails, it loses the stored memory tree and returns to the initial state. These processes are formally defined as follows.

Definition 5 (Event-induced gossip graph). For an initial gossip graph $G = (A, N, \{T_x\}_{x\in A})$ and an event sequence σ, a new gossip graph $G^\sigma = (A, N^\sigma, \{T_x^\sigma\}_{x\in A})$ obtained by executing σ in G is defined as follows.

- Base Case: if $\sigma = \epsilon$, then $N^\epsilon = N$ and $T_x^\epsilon = T_x$ for all x. (Therefore, $G^\epsilon = G$.)
- Ind. Step: if $\sigma = \sigma'; \mathsf{e}$,
  - for the case where e is a call xy,

$$N_z^{\sigma';xy} = \begin{cases} N_x^{\sigma'} \cup N_y^{\sigma'} & (z \in \{x, y\}) \\ N_z^{\sigma'} & (\text{otherwise}) \end{cases}$$

$$T_z^{\sigma';xy} = \begin{cases} \langle T_x^{\sigma'}, xy, T_y^{\sigma'} \rangle & (z \in \{x, y\}) \\ T_z^{\sigma'} & (\text{otherwise}) \end{cases}$$

  - for the case where e is a failure $\mathsf{e} = [x_1 \dots x_k]$,

$$N_z^{\sigma';[x_1\dots x_k]} = \begin{cases} N & (z \in \{x_1, \dots, x_k\}) \\ N_z^{\sigma'} & (\text{otherwise}) \end{cases}$$

$$T_z^{\sigma';[x_1\dots x_k]} = \begin{cases} T_z & (z \in \{x_1, \dots, x_k\}) \\ T_z^{\sigma'} & (\text{otherwise}) \end{cases}$$

A call xy is said to be valid if the gossip graph $G = (A, N, \{T_x\}_{x\in A})$ satisfies Nxy. In addition, we consider the failure $[x_1 \dots x_k]$ to be valid at G for any gossip graph G. Thus, it is assumed that the failure (in some cases, consecutively) occurs at any timing between calls. However, consecutive failures are regarded as one simultaneous failure. For example, $[x_1]; [x_2]; [x_3]$ is regarded to be the same as $[x_1x_2x_3]$. When any element σ_n of the sequence σ of events is valid in $G^{\sigma|n-1}$, σ is said to be valid in G.

Example 1. Consider the sequence $ab; [b]; ca$. Figure 1 shows event-induced graph G^σ for each prefix $\sigma \sqsubseteq ab; [b]; ca$. In the graphs of the leftmost column, each node labeled x represents agent x. Each dashed arrow from x to y represents Nxy, and solid one represents $y \in leaves(T_x^\sigma)$. Arrows to oneself is omitted. Each row corresponds to $G^\sigma = (A, N^\sigma, \{T_x^\sigma\}_{x\in A})$ for the same prefix σ.

The Gossip graph partially represents the knowledge of agents, but not the higher-order knowledge. Therefore, the higher-order knowledge of the agents in the epistemic gossip protocol is represented by the Kripke model (cf. [8]). A state (possible world) in the model is called a gossip state.

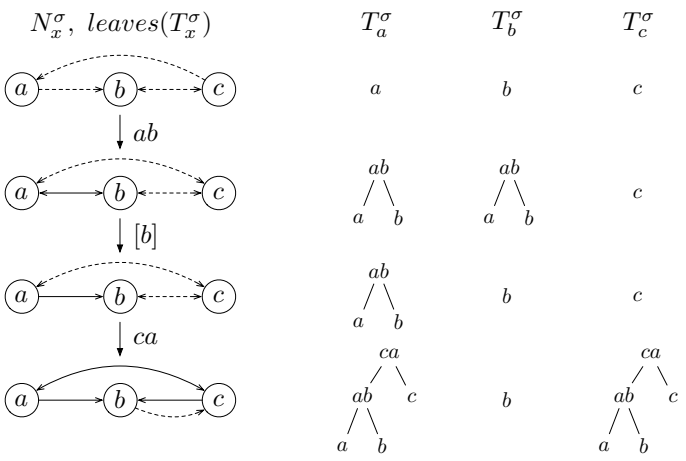

Fig. 1. Event-induced graphs for each prefix $\sigma \sqsubseteq ab; [b]; ca$

Definition 6 (Gossip state). A gossip state is a pair (G, σ) of an initial gossip graph G and a finite event sequence σ that is valid in G.

In our model, each agent assumes the initial gossip graph and the event sequence that realize the current gossip state on the basis of the following common knowledge.

- The set of agents is A.
- The gossip graph changes depending on events, as defined in the Definition 5.
- The graph when no event has occurred yet is one of the initial gossip graphs.
- When no event has occurred yet, each agent does not know the telephone numbers that the other agents know.
- Each agent does not know the protocols followed by the other agents.

The last statement being common knowledge means that the agent considers any valid sequence to be executable, although in reality not all valid sequences may be executed depending on the given protocol.

Based on the assumptions stated above, the notion of accessibility relation is defined below. Here, different relations are given for two types of call modes, asynchronous and synchronous call. Asynchronous call cannot be recognized by agents other than the calling agents. However, synchronous are recognized by all agents, but it is impossible to know who is calling. In either call mode, the failure of the other agents cannot be recognized. Furthermore, if (3) in Definitions 7 and 8 are assumed, the failure of oneself cannot be recognized.

Definition 7 (Asynchronous accessibility relation). Let $G = (A, N, \{T_x\}_{x \in A})$ and $H = (A, O, \{U_x\}_{x \in A})$ be initial gossip graphs, and let σ and τ be valid event sequences in G and H, respectively. The asynchronous accessibility relation $\sim_x$

is the reflexive, symmetric, transitive closure of binary relation $\sim'_x$ on gossip graphs defined as follows.

(1) $(G, \epsilon) \sim'_x (H, \epsilon)$ if $N_x = O_x$.
(2) For any $(\sigma, \tau) \neq (\epsilon, \epsilon)$, we have $(G, \sigma) \sim'_x (H, \tau)$ if any one of the following conditions hold:
   (a) $\sigma = \sigma'; yz$, $x \notin \{y, z\}$, and $(G, \sigma') \sim'_x (H, \tau)$;
   (b) $\sigma = \sigma'; yz$, $\tau = \tau'; yz$, $x \in \{y, z\}$, for each $u \in \{y, z\}$ it is the case that $N_u^{\sigma'} = O_u^{\tau'}$ and $T_u^{\sigma'} = U_u^{\tau'}$, and $(G, \sigma') \sim'_x (H, \tau')$;
   (c) $\sigma = \sigma'; [x_1 \dots x_k]$, $x \notin \{x_1, \dots, x_k\}$, and $(G, \sigma') \sim'_x (H, \tau)$;
   (d) $\sigma = \sigma'; [x_1 \dots x_k]$, $\tau = \tau'; [y_1 \dots y_l]$, $x \in \{x_1, \dots, x_k\} \cap \{y_1, \dots, y_l\}$, and $(G, \sigma') \sim'_x (H, \tau')$.
(3) (Optional) If $\sigma = \sigma'; [x_1 \dots x_k]$ and $x \in \{x_1, \dots, x_k\}$, then $(G, \sigma) \sim'_x (G, \epsilon)$.

Definition 8 (Synchronous accessibility relation). Let $G = (A, N, \{T_x\}_{x \in A})$ and $H = (A, O, \{U_x\}_{x \in A})$ be initial gossip graphs, and let σ and τ be valid event sequences in G and H, respectively. The synchronous accessibility relation $\approx_x$ is the reflexive, symmetric, transitive closure of binary relation $\approx'_x$ on gossip graphs defined as follows.

(1) $(G, \epsilon) \approx'_x (H, \epsilon)$ if $N_x = O_x$.
(2) For any $(\sigma, \tau) \neq (\epsilon, \epsilon)$ we have $(G, \sigma) \approx'_x (H, \tau)$ if any one of the following conditions hold:
   (a) $\sigma = \sigma'; yz$, $\tau = \tau'; uv$, $x \notin \{y, z, u, v\}$, and $(G, \sigma') \approx'_x (H, \tau')$;
   (b) $\sigma = \sigma'; yz$, $\tau = \tau'; yz$, $x \in \{y, z\}$, for each $u \in \{y, z\}$ it is the case that $N_u^{\sigma'} = O_u^{\tau'}$ and $T_u^{\sigma'} = U_u^{\tau'}$, and $(G, \sigma') \approx'_x (H, \tau')$;
   (c) $\sigma = \sigma'; [x_1 \dots x_k]$, $x \notin \{x_1, \dots, x_k\}$, and $(G, \sigma') \approx'_x (H, \tau)$;
   (d) $\sigma = \sigma'; [x_1 \dots x_k]$, $\tau = \tau'; [y_1 \dots y_l]$, $x \in \{x_1, \dots, x_k\} \cap \{y_1, \dots, y_l\}$, and $(G, \sigma') \approx'_x (H, \tau')$.
(3) (Optional) If $\sigma = \sigma'; [x_1 \dots x_k]$ and $x \in \{x_1, \dots, x_k\}$, then $(G, \sigma) \approx'_x (G, \epsilon)$.

The Kripke model, which is based on the accessibility relation and the gossip state defined above, is called the gossip model.

Definition 9 (Gossip model). Given a set of agents A, the asynchronous gossip model and the synchronous gossip model are respectively the tuples

$$\mathcal{G}^{\sim} = (\mathcal{G}, \langle \sim_a \rangle_{a \in A}, \langle \xrightarrow{\mathsf{e}} \rangle_{\mathsf{e} \in \mathsf{E}}) \quad \text{and} \quad \mathcal{G}^{\approx} = (\mathcal{G}, \langle \approx_a \rangle_{a \in A}, \langle \xrightarrow{\mathsf{e}} \rangle_{\mathsf{e} \in \mathsf{E}}),$$

where

- $\mathcal{G}$ is the set of gossip states;
- $\sim_a$ and $\approx_a$ are relations defined in 7 and 8;
- $\xrightarrow{\mathsf{e}}$ is the relation on $\mathcal{G}$ such that for any G where event e is valid and for any σ it is the case that $(G, \sigma) \xrightarrow{\mathsf{e}} (G, \sigma; \mathsf{e})$.

3.2 Epistemic Logic

To specify and analyse protocols, we use epistemic logic defined below.

Definition 10 (Language). Given a set of agents A, the language $\mathcal{L}$ used to specify conditions of protocols is defined by the following BNF:

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

where $a, b, c \in A$. We define connectives $\rightarrow$, $\vee$ and $\leftrightarrow$ in a standard way and denote the dual of K_a by $\hat{K}_a$.

Intuitively, $\mathtt{N}(a,b)$ means that a knows the telephone number of b. $\mathtt{S}(a,b)$ means that a knows the secret of b. $\mathtt{C}(ab,c)$ means that ab is included in the call involving c. $\mathtt{C}(ab,c)$ is false whenever c is neither a nor b, and $\mathtt{C}(ab,a)$ and $\mathtt{C}(ab,b)$ are true at (G,σ) when σ contains ab. $\mathtt{F}(a)$ indicates that a has failed at least once. $K_a\varphi$ means that a knows φ, and $\hat{K}_a\varphi$ means that a considers φ to be possible.

Formally, the truth conditions for the formulas are defined as follows.

Definition 11 (Semantics). Let $\mathcal{G}^{\sim} = (\mathcal{G}, \langle\sim_a\rangle_{a\in A}, \langle\xrightarrow{ab}\rangle_{a,b\in A})$ be an asynchronous gossip model. For any $(G,\sigma) \in \mathcal{G}$ with $G = (A, N, \{T_x\}_{x\in A})$ and for any formula φ in $\mathcal{L}$, we define $\mathcal{G}^{\sim}, (G,\sigma) \models \varphi$ by induction on φ as follows.

$\mathcal{G}^{\sim}, (G,\sigma) \models \top$	iff	always
$\mathcal{G}^{\sim}, (G,\sigma) \models \mathtt{N}(a,b)$	iff	$N^\sigma ab$
$\mathcal{G}^{\sim}, (G,\sigma) \models \mathtt{S}(a,b)$	iff	$b \in \mathit{leaves}(T_a^\sigma)$
$\mathcal{G}^{\sim}, (G,\sigma) \models \mathtt{C}(ab,c)$	iff	$ab \in \sigma_c$
$\mathcal{G}^{\sim}, (G,\sigma) \models \mathtt{F}(a)$	iff	$[x_1 \dots x_k] \in \sigma$ and $a \in \{x_1, \dots, x_k\}$
$\mathcal{G}^{\sim}, (G,\sigma) \models \neg\varphi$	iff	not $\mathcal{G}^{\sim}, (G,\sigma) \models \varphi$
$\mathcal{G}^{\sim}, (G,\sigma) \models (\varphi_1 \wedge \varphi_2)$	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 any (H,τ) with $(G,\sigma) \sim_a (H,\tau)$, we have $\mathcal{G}^{\sim}, (H,\tau) \models \varphi$

We define $\mathcal{G}^{\approx}, (G,\sigma) \models \varphi$ by replacing $\sim_a$ with $\approx_a$ in the above condition on $K_a\varphi$.

The algorithm that specifies the behavior the agent should follow in the epistemic gossip protocol is defined below. In this study, we assume that all agents follow the same protocol.

Definition 12 (Gossip protocol). A gossip protocol is the nondeterministic algorithm of the following form:

> `while` not all agents are experts and there are $u, v \in A$ s.t. $\varphi(u,v)$ is satisfied and call uv is valid;
>
> > `select` $u, v \in A$ s.t. $\varphi(u,v)$ is satisfied and call uv is valid;
> > `execute` call uv;

where $\varphi(u, v)$ is a formula in the language $\mathcal{L}$.

For a given protocol P, all possible sequences of events that can be executed according to that protocol are called the $\mathrm{P}^{\sim}$-permitted sequence. A set of $\mathrm{P}^{\sim}$-permitted sequences is called extensions.

Definition 13 (Permitted sequence). Let P be a protocol given by condition $\varphi(x, y)$ and G be an initial gossip graph.

- A call ab is $\mathrm{P}^{\sim}$-permitted in (G, σ) if $\mathcal{G}^{\sim}, (G, \sigma) \models \varphi(a, b)$, call ab is valid in G^{σ} and not all agents are experts in G^{σ}.
- A failure $[x_1 \dots x_k]$ is $\mathrm{P}^{\sim}$-permitted in (G, σ) if there exists a $\mathrm{P}^{\sim}$-permitted call in (G, σ).
- An event sequence σ is $\mathrm{P}^{\sim}$-permitted in G if each event σ_{n+1} is $\mathrm{P}^{\sim}$-permitted in $(G, \sigma|n)$.
- The extension of P in G is the set of all $\mathrm{P}^{\sim}$-permitted event sequences in G, denoted by $\mathrm{P}^{\sim}_G$.
- A sequence $\sigma \in \mathrm{P}^{\sim}_G$ is $\mathrm{P}^{\sim}$-maximal on G if it is infinite or there is no event e such that $\sigma; \mathsf{e} \in \mathrm{P}^{\sim}_G$.

$\mathrm{P}^{\approx}$-permitted, $\mathrm{P}^{\approx}_G$ and $\mathrm{P}^{\approx}$-maximal are defined similarly. When we discuss both $\mathrm{P}^{\sim}$ and $\mathrm{P}^{\approx}$ together, we simply write P.

Given an initial gossip graph and a protocol, the more number of sequences are included in the extension of the protocol that succeed in spreading the secret, the more the protocol is considered to be successful. Thus, protocols are classified into the following four types, depending on their level of success.

Definition 14 (Successful). Let G be an initial gossip graph and P be a protocol.

- A sequence $\sigma \in \mathrm{P}_G$ is successful if it is finite and in G^{σ} all agents are experts.
- A sequence $\sigma \in \mathrm{P}_G$ is fair if it is finite or for any call xy the following condition holds.

  If for any $i \in \mathbb{N}$ there exists $j \geq i$ such that xy is P-permitted in $G^{\sigma|j}$, then for any $i \in \mathbb{N}$ there exists $j \geq i$ such that $\sigma_j = xy$.
- P is strongly successful on G if all maximal $\sigma \in \mathrm{P}_G$ are successful.
- P is fairly successful on G if all fair and maximal $\sigma \in \mathrm{P}_G$ are successful.
- P is weakly successful on G if there exists $\sigma \in \mathrm{P}_G$ which is maximal and successful.
- P is unsuccessful on G if there is no $\sigma \in \mathrm{P}_G$ which is maximal and successful.

4 Analysis of Robustness and Failure Detection

In this section, we present the results of our protocol analysis.

4.1 Properties on Robustness

We analyze the following three protocols [6].

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) $\varphi(x,y) := \hat{K}_x \bigvee_{z\in A}(\mathtt{S}(x,z) \leftrightarrow \neg\mathtt{S}(y,z))$
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.

CO (Call Me Once) $\varphi(x,y) := \neg\mathtt{C}(xy,x) \wedge \neg\mathtt{C}(yx,x)$
Agent x may call agent y if x knows y's number and there was no prior call between x and y.

Here, we assume that the number of failures is finite. Thus, for protocol P, we restrict P_G to a set of sequences, each of which contains a finite number of failures.

First, we confirm that for the protocols ANY and PIG, the properties shown in [6] also hold even if the agents fail.

Theorem 1. *Protocol ANY is fairly successful on G iff G is weakly connected.*

Proof. We can prove the statement in a way similar to [7]. We prove only $\Leftarrow$ part because the converse is obvious. Let σ be an ANY-permitted and fair sequence. It suffices to show that σ is not infinite. For contradiction, we assume that σ is infinite. Now that we assume that σ contains only a finite number of failures, there is a finite prefix $\tau \sqsubseteq \sigma$ such that for any $\tau'(\tau \sqsubseteq \tau' \sqsubseteq \sigma)$, we have $N^\tau = N^{\tau'}$ and $leaves(T_x^\tau) = leaves(T_x^{\tau'})$ for any $x \in A$. Since σ is not successful, there are $x, y \in A$ such that Nxy and $leaves(T_x^\tau) \neq leaves(T_y^\tau)$ (Otherwise σ is successful because G is weakly connected). However, since σ is fair, in σ the call xy is executed after τ. This is a contradiction. □

Theorem 2. *Protocol PIG$^\sim$ is fairly successful on G iff G is weakly connected.*

Proof. We can prove the statement by similar argument of [7]. We prove only $\Leftarrow$ direction because the converse is obvious. Let σ be a PIG$^\sim$-maximal sequence.

We first show that if σ is infinite, it is not fair. Since we assume that σ contains only a finite number of failures, there is a finite prefix $\tau \sqsubseteq \sigma$ such that for any $\tau'(\tau \sqsubseteq \tau' \sqsubseteq \sigma)$, we have $N^\tau = N^{\tau'}$ and $leaves(T_x^\tau) = leaves(T_x^{\tau'})$ for any $x \in A$. Further, since σ is not successful, there are $x, y \in A$ such that Nxy and $leaves(T_x^\tau) \neq leaves(T_y^\tau)$. This implies that after τ the call xy is always PIG$^\sim$-permitted. However, xy is not executed after τ. Therefore, σ is not fair.

We next show that if σ is finite, it is successful. The sequence σ is finite only if all agents are experts in G^σ or for any $x, y \in A$ it is the case that $\mathcal{G}^\sim, (G,\sigma) \not\models \hat{K}_x \bigvee_{z\in A}(\mathtt{S}(x,z) \leftrightarrow \neg\mathtt{S}(y,z))$. In the former case, σ is successful by definition. In the latter case, by definition of $\hat{K}_x$, we have $\mathcal{G}^\sim, (G,\sigma) \models K_x \neg \bigvee_{z\in A}(\mathtt{S}(x,z) \leftrightarrow \neg\mathtt{S}(y,z))$. This implies that for any $x, y \in A$ it is the case that $leaves(T_x^\sigma) = leaves(T_y^\sigma)$. Therefore, σ is successful.

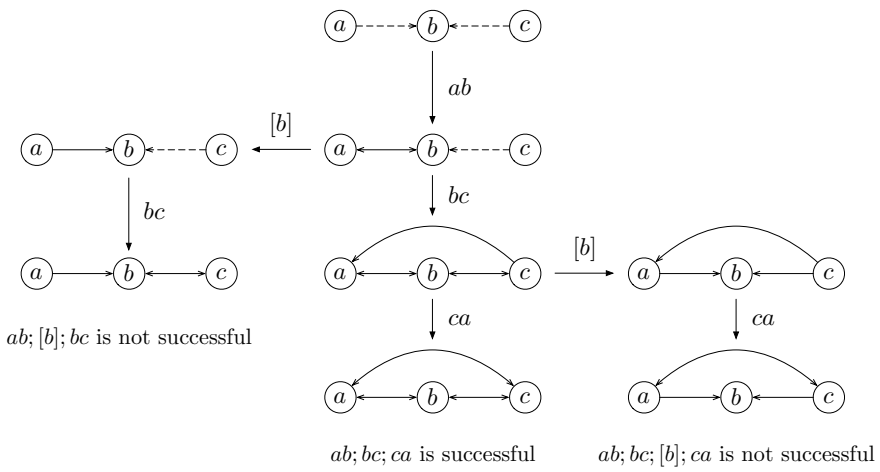

Fig. 2. A counter example of Theorem 13 in [7]

Finally, it remains to show that there exists a successful σ. Let σ be a successful ANY-permitted sequence of minimum length. We show that σ is also PIG$^\sim$-permitted. Clearly, in σ there is no call between two experts. The prefix $\sigma|1$ is PIG$^\sim$-permitted. We need to show that each σ_{n+1} is PIG$^\sim$-permitted in $G^{\sigma|n}$. If σ_{n+1} is a failure, then it is clearly PIG$^\sim$-permitted in $G^{\sigma|n}$. Thus, we assume σ_{n+1} is a call xy. In the case $leaves(T_x^{\sigma|n}) = A$, we have $leaves(T_y^{\sigma|n}) \neq A$ Therefore, xy is PIG$^\sim$-permitted in $G^{\sigma|n}$. In the case $leaves(T_x^{\sigma|n}) \neq A$, let $z \in A \setminus leaves(T_x^{\sigma|n})$. Then we have $(G, \sigma|n) \sim_x (G, \sigma|n; zy)$ and $leaves(T_x^{\sigma|n}) \neq leaves(T_y^{\sigma|n;zy})$. Therefore, xy is PIG$^\sim$-permitted in $G^{\sigma|n}$. □

The reason why these theorems hold is that ANY and PIG are the "careful" protocol for failures. In other words, in these protocols, agents are forced to repeatedly exchange information in case of failure. A careful protocol assumes the worst case and decides who to call, regardless of whether or not the other agent actually fails. Therefore, if fairness is not assumed, redundant calls may be repeated.

Unlike ANY and PIG, the property shown in [6] does not hold for CO. More precisely, CO is successful in a weakly connected graph when there is no agent failure (cf. Theorem 13 in [7]) but it does not when failure may occur. As a counterexample of this theorem in our model, in Figure 2, we show a sequence of events beginning with a weakly connected initial gossip graph but do not achieve a successful state.

The result shown above is obtained because CO, unlike ANY and PIG, is a protocol that reduces too much redundancy and, thus, fails to fully recover failures. A closer look at the cause yields two useful suggestions. The first suggestion is obtained from the counterexample sequence $ab; bc; [b]; ca$, which suggests that agents who may fail should not be experts first. This occurs because the infor-

mation owned by such agents may be lost owing to a failure in the future. The other suggestion is obtained from the counterexample sequence $ab; [b]; bc$, which suggests that information should not be routed through an agent who may fail. This occurs because the transmission of information may fail depending on the timing of the agent failure.

The following theorem shows that a successful sequence can be achieved if we schedule the partial sequence of calls to avoid the undesired steps presented in the counterexamples above.

Theorem 3. *For an initial gossip graph $G = (A, N, \{T_x\}_{x \in A})$, we assume the following.*

- *A single agent a is the only agent who can fail.*
- *There are at least two agents who do not fail.*
- *The restriction of G to $A \setminus \{a\}$ is weakly connected.*
- *There is $x \in A \setminus \{a\}$ such that Nxa.*

Then the sequence $xa; \sigma; ya$ obtained by the following procedure is CO-permitted and successful (even if σ contains a finite number of a's failures).

(1) Execute a call xa.
(2) Execute CO among the agents in $A \setminus \{a\}$ (let us denote the event sequence executed in this step by σ, which may contain some $[a]$).
(3) Select an agent $y \in A \setminus \{a\}$ other than x and then execute a call ya.

Proof. By executing the call xa in (1), x obtains the secret and the telephone number of a. Since CO is strongly successful in a weakly connected graph when there is no agent failure (cf. Theorem 13 in [7]), σ is finite and after the execution of σ, all $z \in A \setminus \{a\}$ are experts. Since there are at least two agents who do not fail, there is an agent $y \in A \setminus \{a\}$ other than x. By the call ya in (3), a lastly becomes an expert. Therefore, $xa; \sigma; ya$ is successful. Moreover, since there is no calls which a is involved in, $xa; \sigma; ya$ is CO-permitted. □

However, in the framework of epistemic gossip protocol, such scheduling cannot be realized directly. This fact suggests to us, as an alternative approach, to design a protocol with the level of carefulness that reconciles the trade-off relationship between the protocol CO and the protocol ANY or PIG. This is an ideal protocol that is able to detect failures and recover lost information when needed. In the next subsection, as a first step in designing such protocol, we investigate sufficient conditions in the sequence of calls that allow agents to detect failures.

4.2 Analysis of Failure Detection

In this subsection, we assume that the following are common knowledge between agents: gossip graph G is complete; only one particular agent (say, a) can fail; failure can occur only once. Formally, assuming these three things to be common knowledge means that we consider $\mathcal{G}$, which consists of only gossip state (G, σ)

for which the three facts stated above hold. Also, $\approx_x$ is used as the reachability relation. Under these assumptions, we use the notation $\sigma \approx_x \tau$ to denote $(G, \sigma) \approx_x (G, \tau)$ and the notation $\sigma \models \phi$ to denote $\mathcal{G}^{\approx}, (G, \sigma) \models \phi$.

For a given protocol P and agents $a, x \in A$, if $\sigma \models K_x F(a)$, agent x is said to detect a's failure in $\sigma \in \mathrm{P}_G$. By the definition of $\models$, this is equivalent to $[a] \in \tau$ for any τ that satisfies $\sigma \approx_x \tau$. A counterexample τ of this condition, namely τ that satisfies $\sigma \approx_x \tau$ and $[a] \notin \tau$, is called an optimistic path. The formal definition is given below.

Definition 15 (Optimistic path). For an event sequence σ, an optimistic path of x for σ is a sequence τ such that $\sigma \approx_x \tau$ and $[a] \notin \tau$. Let $opt_x : \mathsf{E}^* \to 2^{\mathsf{E}^*}$ be the function which maps an event sequence σ to the set of optimistic paths of x for σ. That is, we define opt_x as follows.

$$opt_x(\sigma) := \{\tau \mid \sigma \approx_x \tau \text{ and } [a] \notin \tau\}.$$

In order to show that $\sigma \models K_x F(a)$ holds, it suffices to show that the set $opt_x(\sigma)$ is empty. By the definition of $\approx_x$, the set $opt_x(\sigma)$ can be calculated by induction on σ:

- Base Case: If $\sigma = \epsilon$, then $opt_x(\sigma) = \{\epsilon\}$.
- Ind. Step: If $\sigma = \sigma'; \mathsf{c}$ and $\mathsf{c} \in \mathsf{C}$,
  - For the case $x \in \mathsf{c}$, let $\mathsf{c} = \overline{xy}$. Then it follows that

$$opt_x(\sigma) = \{\tau; \mathsf{c} \mid \tau \in opt_x(\sigma') \text{ and } T_y^{\sigma'} = T_y^{\tau}\}.$$

  - For the case $x \notin \mathsf{c}$, it follows that

$$opt_x(\sigma) = \{\tau; \mathsf{c}' \mid \tau \in opt_x(\sigma') \text{ and } x \notin \mathsf{c}'\}.$$

  If $\sigma = \sigma'; [a]$, then $opt_x(\sigma) = opt_x(\sigma')$.

Here we note that no assumptions about telephone number appear in the calculations presented above, because we assume that G is a complete graph. In addition, optimistic path is usually calculated bottom-up beginning with $\tau = \epsilon$.

Example 2. For $\sigma = ax; [a]; ab; bx$, the set $opt_x(\sigma)$ can be calculated by the following steps:

$$\begin{aligned}
opt_x(\epsilon) &= \{\epsilon\} \\
opt_x(ax) &= \{\epsilon; ax \mid T_a^{\epsilon} = T_a^{\epsilon}\} = \{ax\} \\
opt_x(ax; [a]) &= \{ax\} \\
opt_x(ax; [a]; ab) &= \{ax; \mathsf{c} \mid x \notin \mathsf{c}\} \\
opt_x(ax; [a]; ab; bx) &= \{ax; \mathsf{c}; bx \mid x \notin \mathsf{c} \text{ and } T_b^{ax;[a];ab} = T_b^{ax;\mathsf{c}}\}.
\end{aligned}$$

Since there is no call c such that $x \notin \mathsf{c}$ and $T_b^{ax;\{a\};ab} = T_b^{ax;\mathsf{c}}$, we have $opt_x(\sigma) = \emptyset$. Therefore, after the execution of σ, agent x can detect the failure of a. In a similar way, we can determine whether $\sigma \models K_x F(a)$ is true or not, given $\sigma \in \mathsf{E}^*$.

From now on, we consider a more general pattern of σ that satisfies $\sigma \models K_x F(a)$. The pattern of σ, shown in Theorem 4 below, is a generalization of σ presented in Example 2. The underlying idea is that an agent can detect the failure by comparing information before and after the a's failure. Before proving the theorem, we provide some lemmas. Hereafter, we use $sub_a(T)$ to denote $\{T' \subseteq T \mid a \in r(T')\}$.

Lemma 1. *If a finite event sequence σ satisfies $[a] \notin \sigma$, then for any $x \in A$ and for any $T \in sub_a(T_x^\sigma)$ there is $\tau \sqsubseteq \sigma$ such that $T = T_a^\tau$.*

Proof. We prove this by induction on σ.

- Base Case: If $\sigma = \epsilon$, for any $x \in A$ it is the case that $T_x^\sigma = T_x^\epsilon = \langle x \rangle$. Thus, the statement holds for $\sigma = \epsilon$.
- Ind. Step: Let $\sigma = \sigma'; \mathsf{c}$ and $\mathsf{c} \in \mathsf{C}$. We fist consider the case $a \notin \mathsf{c}$. Let $\mathsf{c} = yz$. Then it follows that

$$sub_a(T_x^\sigma) = \begin{cases} sub_a(T_y^{\sigma'}) \cup sub_a(T_z^{\sigma'}) & \text{if } x \in \{y, z\}, \\ sub_a(T_x^{\sigma'}) & \text{otherwise.} \end{cases}$$

  Therefore, by the induction hypothesis, we have the statement. We then consider the case $a \in \mathsf{c}$. Let $\mathsf{c} = \overline{ay}$. Then it follows that

$$sub_a(T_x^\sigma) = \begin{cases} sub_a(T_a^{\sigma'}) \cup sub_a(T_y^{\sigma'}) \cup \{T_a^\sigma\} & \text{if } x \in \{a, y\}, \\ sub_a(T_x^{\sigma'}) & \text{otherwise.} \end{cases}$$

  We can take $\tau = \sigma$ for T_a^σ. Therefore, together with the induction hypothesis, we have the statement. □

Lemma 2. *If an event sequence σ satisfies $[a] \notin \sigma$, then for any $x \in A$ and for any $T, T' \in sub_a(T_x^\sigma)$ it is the case that $T \subseteq T'$ or $T' \subseteq T$.*

Proof. By Lemma 1, for any $x \in A$ and for any $T, T' \in sub_a(T_x^\sigma)$ there are $\tau, \tau' \sqsubseteq \sigma$ such that $T = T_a^\tau$ and $T' = T_a^{\tau'}$. If $\tau \sqsubseteq \tau'$, then $T_a^\tau \subseteq T_a^{\tau'}$, that is, $T \subseteq T'$. If $\tau' \sqsubseteq \tau$, then $T_a^{\tau'} \subseteq T_a^\tau$, that is, $T' \subseteq T$. □

Lemma 3. *For any event sequence σ and any $x \in A$, if $[a] \notin \sigma$ and $r(T_x^\sigma) = ax$ (xa, resp.), then for any $T \in sub_a(T_{x,R}^\sigma)$ ($sub_a(T_{x,L}^\sigma)$, resp.), it is the case that $T \subseteq T_{x,L}^\sigma$ ($T_{x,R}^\sigma$, resp.).*

Proof. Since we assume $r(T_x^\sigma) = ax$ (xa, resp.), there is a prefix $\tau \sqsubseteq \sigma$ such that $\tau = \tau'; ax$ and $T_x^\sigma = T_x^\tau$. Furthermore, we have $T_{x,L}^\sigma = T_{x,L}^\tau = T_a^{\tau'}$ ($T_x^{\tau'}$, resp.) and $T_{x,R}^\sigma = T_{x,R}^\tau = T_x^{\tau'}$ ($T_a^{\tau'}$, resp.). Since $[a] \notin \sigma$ implies $[a] \notin \tau'$, using Lemma 1, for any $T \in sub_a(T_x^{\tau'})$ there is a prefix $\rho \sqsubseteq \tau'$ such that $T = T_a^\rho$. Moreover, since $\rho \sqsubseteq \tau'$, we have $T_a^\rho \subseteq T_a^{\tau'}$, that is, $T \subseteq T_a^{\tau'}$. Therefore, for any $T \in sub_a(T_{x,R}^\sigma)$ ($sub_a(T_{x,L}^\sigma)$, resp.), we have $T \subseteq T_{x,L}^\sigma$ ($T_{x,R}^\sigma$, resp.). □

Theorem 4. *When agent x obtains agent a's secret distributed before and after a's failure from two paths which do not share any nodes, x can detect a's failure. Formally, if $\{b_1, \ldots, b_k\}, \{c_1, \ldots, c_l\} \subseteq A$ and $\{b_1, \ldots, b_k\} \cap \{c_1, \ldots, c_l\} = \emptyset$ with $k, l \geq 0$, and if σ is a sequence consisting of the following events:*

$$\overline{ab_1}, \overline{b_1b_2}, \ldots, \overline{b_{k-1}b_k}, \overline{b_kx}, \overline{ac_1}, \overline{c_1c_2}, \ldots, \overline{c_{l-1}c_l}, \overline{c_lx}, [a],$$

and if

$$\overline{ab_1} \prec \overline{b_1b_2} \prec \cdots \prec \overline{b_{k-1}b_k} \prec \overline{b_kx} \tag{1}$$

$$\overline{ac_1} \prec \overline{c_1c_2} \prec \cdots \prec \overline{c_{l-1}c_l} \prec \overline{c_lx} \tag{2}$$

$$\overline{ab_1} \prec [a] \prec \overline{ac_1} \tag{3}$$

where $\mathsf{e}_1 \prec \mathsf{e}_2$ means that e_1 is executed earlier than e_2 in σ, then $\sigma \models K_xF(a)$.

Proof. We divide the proof according to whether k and l are equal to 0 or not, respectively.

- For $k = l = 0$, we have $\sigma = \overline{ax}; [a]; \overline{ax}$. Then $opt_x(\sigma) = \{\overline{ax}; \overline{ax} \mid T_a^{\overline{ax};[a]} = T_a^{\overline{ax}}\}$. Since $T_a^{\overline{ax};[a]} \neq T_a^{\overline{ax}}$, we have $opt_x(\sigma) = \emptyset$. Therefore, $\sigma \models K_xF(a)$.
- For $k > 0$ and $l = 0$, we have $\sigma = \sigma'; \overline{ax}$ or $\sigma = \sigma'; \overline{b_kx}$. If $\sigma = \sigma'; \overline{ax}$, then $r(T_x^\sigma) = \overline{ax}$. By the conditions (1), (2) and (3), it is the case that $T_a^{\sigma'} = \langle a \rangle$ and $T_x^{\sigma'}$ has $\langle\langle a \rangle, ab_1, \langle b_1 \rangle\rangle$ or $\langle\langle b_1 \rangle, b_1a, \langle a \rangle\rangle$ as a subtree. Since T_x^σ is $\langle T_a^{\sigma'}, ax, T_x^{\sigma'}\rangle$ or $\langle T_x^{\sigma'}, xa, T_a^{\sigma'}\rangle$, using Lemma 3, we have $\sigma \models K_xF(a)$. If $\sigma = \sigma'; \overline{b_kx}$, by the conditions (1), (2) and (3), the tree $T_{b_k}^{\sigma'}$ has $\langle\langle a \rangle, ab_1, \langle b_1 \rangle\rangle$ or $\langle\langle b_1 \rangle, b_1a, \langle a \rangle\rangle$ as a subtree, and $T_x^{\sigma'}$ is $\langle\langle a \rangle, ax, \langle x \rangle\rangle$ or $\langle\langle x \rangle, xa, \langle a \rangle\rangle$. Since the tree T_x^σ is $\langle T_{b_k}^{\sigma'}, b_kx, T_x^{\sigma'}\rangle$ or $\langle T_x^{\sigma'}, xb_k, T_{b_k}^{\sigma'}\rangle$, using Lemma 2, we have $\sigma \models K_xF(a)$.
- For $k = 0$ and $l > 0$, we have $\sigma = \overline{ax}; [a]; \overline{ac_1}; \overline{c_1c_2}; \ldots; \overline{c_{l-1}c_l}; \overline{c_lx}$. Let $\sigma = \sigma'; \overline{c_lx}$. Then $T_{c_l}^{\sigma'}$ has $\langle\langle a \rangle, ac_1, \langle c_1 \rangle\rangle$ or $\langle\langle c_1 \rangle, c_1a, \langle a \rangle\rangle$ as a subtree, and $T_x^{\sigma'}$ has $\langle\langle a \rangle, ax, \langle x \rangle\rangle$ or $\langle\langle x \rangle, xa, \langle a \rangle\rangle$ as a subtree. Since the tree T_x^σ is $\langle T_{c_l}^{\sigma'}, c_lx, T_x^{\sigma'}\rangle$ or $\langle T_x^{\sigma'}, xc_l, T_{c_l}^{\sigma'}\rangle$, using Lemma 2, we have $\sigma \models K_xF(a)$.
- For $k > 0$ and $l > 0$, we have $\sigma = \sigma'; \overline{b_kx}$ or $\sigma = \sigma'; \overline{c_lx}$. If $\sigma = \sigma'; \overline{b_kx}$, by the assumption that $\{b_1, \ldots, b_k\} \cap \{c_1, \ldots, c_l\} = \emptyset$ and the conditions (1), (2) and (3), it is the case that $T_{b_k}^{\sigma'} = T_{b_k}^{\overline{ab_1};\overline{b_1b_2};\ldots;\overline{b_{k-1}b_k}}$ and $T_x^{\sigma'} = T_x^{\overline{ac_1};\overline{c_1c_2};\ldots;\overline{c_{l-1}c_l};\overline{c_lx}}$. Since $\{b_1, \ldots, b_k\} \cap \{c_1, \ldots, c_l\} = \emptyset$, it follows that $\langle b_1 \rangle \subseteq T_{b_k}^{\sigma'}$ and $\langle b_1 \rangle \not\subseteq T_x^{\sigma'}$. Using Lemma 2, we have $\sigma \models K_xF(a)$. If $\sigma = \sigma'; \overline{c_lx}$, By the assumption that $\{b_1, \ldots, b_k\} \cap \{c_1, \ldots, c_l\} = \emptyset$ and the conditions (1), (2) and (3), it is the case that $T_{c_l}^{\sigma'} = T_{c_l}^{\overline{ac_1};\overline{c_1c_2};\ldots;\overline{c_{l-1}c_l}}$ and $T_x^{\sigma'} = T_x^{\overline{ab_1};\overline{b_1b_2};\ldots;\overline{b_{k-1}b_k};\overline{b_kx}}$. Since $\{b_1, \ldots, b_k\} \cap \{c_1, \ldots, c_l\} = \emptyset$, it follows that $\langle c_1 \rangle \subseteq T_{c_l}^{\sigma'}$ and $\langle c_1 \rangle \not\subseteq T_x^{\sigma'}$. Using Lemma 2, we have $\sigma \models K_xF(a)$. □

5 Conclusions and Future Work

In this paper, in order to increase the reliability of epistemic gossip protocols, we proposed a logical analysis method of robustness against agent failure. In our

model, when agent fails, it loses the secrets and telephone numbers gained by previous calls and returns to the initial state. In addition, during each call, agent share not only the secrets but also the history of the transmission path of each secret.

For this settings, we showed that the protocols ANY and PIG are fairly successful if the graphs were connected, as in the case where no failure is assumed. On the other hand, for the protocol CO, we showed that there exists a sequence of calls that is not successful in a weakly connected graph. These results suggest the need for a failure detection mechanism. Therefore, in this paper, we also showed the sufficient condition of the sequence of calls for an agent to detect the failure of other agents in PIG. Our results provide useful information to make the protocol robust against agent failure.

There are still issues to be addressed as an extension of this study. Although condition that the sequence of calls must satisfy to detect other agent's failure, we have not achieved a concrete protocol that allows the sequence that satisfies this condition. Moreover, currently, we have only obtained a sufficient condition. Thus, there should be a more general form of a sequence of calls where someone can detect a failure. From a more practical point of view, the robustness against various other types of failures such as the Byzantine failure of agents and communication failures has not yet been clarified. We plan to address these research issues by extending the framework given in this study.

References

1. Apt, K.R., Grossi, D., van der Hoek, W.: Epistemic protocols for distributed gossiping. Electronic Proceedings in Theoretical Computer Science **215**, 51–66 (Jun 2016)
2. Attamah, M., Van Ditmarsch, H., Grossi, D., van der Hoek, W.: Knowledge and gossip. In: ECAI. pp. 21–26 (2014)
3. van den Berg, L.: Unreliable Gossip. Master's thesis, Universiteit van Amsterdam (2018)
4. Chandra, T.D., Toueg, S.: Unreliable failure detectors for reliable distributed systems. Journal of the ACM (JACM) **43**(2), 225–267 (1996)
5. Cooper, M.C., Herzig, A., Maffre, F., Maris, F., Régnier, P.: The epistemic gossip problem. Discrete Mathematics **342**(3), 654–663 (2019)
6. van Ditmarsch, H., van Eijck, J., Pardo, P., Ramezanian, R., Schwarzentruber, F.: Epistemic protocols for dynamic gossip. Journal of Applied Logic **20**, 1–31 (2017)
7. van Ditmarsch, H., van Eijck, J., Pardo, P., Ramezanian, R., Schwarzentruber, F.: Dynamic gossip. Bulletin of the Iranian Mathematical Society **45**(3), 701–728 (2019)
8. Fagin, R., Moses, Y., Halpern, J.Y., Vardi, M.Y.: Reasoning about knowledge. MIT press (1995)
9. Hajnal, A., Milner, E.C., Szemerédi, E.: A cure for the telephone disease. Canadian Mathematical Bulletin **15**(3), 447–450 (1972)
10. Hedetniemi, S.M., Hedetniemi, S.T., Liestman, A.L.: A survey of gossiping and broadcasting in communication networks. Networks **18**(4), 319–349 (1988)
11. Tijdeman, R.: On a telephone problem. Nieuw Archief voor Wiskunde **3**(19), 188–192 (1971)