

**UNIVERSIDADE DE LISBOA
INSTITUTO SUPERIOR TÉCNICO**

**Sustaining Cooperation in Dependable Systems: A
Game Theoretical Approach**

Xavier Araújo Morgado Vilaça

Supervisor: Doctor Luís Eduardo Teixeira Rodrigues

**Thesis approved in public session to obtain the PhD Degree in
Information Systems and Computer Engineering
Jury final classification: Pass with Distinction**

Jury

Chairperson: Chairman of the IST Scientific Board

Members of the Committee:

Doctor Luís Eduardo Teixeira Rodrigues

Doctor Chryssis Georgiou

Doctor Francisco João Duarte Cordeiro Correia dos Santos

Doctor José Orlando Roque Nascimento Pereira

UNIVERSIDADE DE LISBOA
INSTITUTO SUPERIOR TÉCNICO

**Sustaining Cooperation in Dependable Systems: A
Game Theoretical Approach**

Xavier Araújo Morgado Vilaça

Supervisor: Doctor Luís Eduardo Teixeira Rodrigues

**Thesis approved in public session to obtain the PhD Degree in
Information Systems and Computer Engineering
Jury final classification: Pass with Distinction**

Jury

Chairperson: Chairman of the IST Scientific Board

Members of the Committee:

Doctor Luís Eduardo Teixeira Rodrigues, Professor Catedrático do Instituto Superior Técnico da Universidade de Lisboa

Doctor Chryssis Georgiou, Associate Professor, University of Cyprus, Cyprus

Doctor Francisco João Duarte Cordeiro Correia dos Santos, Professor Associado do Instituto Superior Técnico da Universidade de Lisboa

Doctor José Orlando Roque Nascimento Pereira, Professor Auxiliar da Escola de Engenharia da Universidade do Minho

Funding Institutions

Fundação para a Ciência e Tecnologia

2016

Acknowledgements

This thesis could not have been concluded without the support of many people. A special mention goes to my advisor Luís Rodrigues, Professor Joe Halpern, my family, and the colleagues with whom I regularly had lunch/coffee or shared an office for five long years, namely, Nuno Machado, Nuno Diegues, Diego Didona, Oksana Denysyuk, Pedro Ruivo, Hugo Rodrigues, Pedro Mota, Daniel Andrade, and Beatriz Ferreira.

This work was partially supported by Fundação para a Ciência e Tecnologia (FCT) via the individual Doctoral grant SFRH/BD/79822/2011, via the INESC-ID multi-annual funding through the PIDDAC Program fund grant, under project PEst-OE/ EEI/ LA0021/ 2013, via the project PEPITA (PTDC/EEI-SCR/2776/2012), via the project DependableCloud ERC-2012-StG-307732, and via the project Abyss (PTDC/ EEI-SCR/ 1741/ 2014).

Lisboa, 2016

Xavier Araújo Morgado Vilaça

To the dawn breaker.

Resumo

Um sistema distribuído é constituído por vários processos que executam um protocolo distribuído que providencia um serviço confiável. Tipicamente, considera-se que todos os processos cooperam, executando o protocolo de acordo com a especificação, exceto quando ocorrem falhas; se os processos não cooperarem, o serviço que o sistema deveria fornecer pode ser comprometido. Infelizmente, a assunção de que os processos cooperam pode não ser válida em sistemas abertos, em que cada processo é gerido por uma entidade distinta. Na realidade, se as entidades forem egoístas e beneficiarem de desvios do protocolo, estas podem alterar o protocolo executado pelos processos. De forma a evitar este problema, protocolos distribuídos devem fomentar cooperação, isto é, devem providenciar incentivos que neguem qualquer benefício a entidades responsáveis por desvios.

Uma forma de modelar comportamento egoísta consiste em adotar a abordagem de Teoria de Jogos. Nesta abordagem, assume-se que cada processo é controlado por um agente racional que visa maximizar uma função de utilidade pessoal, interações são modeladas como jogos e protocolos correspondem a estratégias que definem as ações durante o jogo. O objetivo principal de Teoria de Jogos consiste em desenvolver equilíbrios, isto é, protocolos em que nenhum agente racional obtém uma utilidade superior se o processo controlado por si se desviar do protocolo. Estes protocolos fomentam cooperação, sendo, portanto, extremamente relevantes para o desenvolvimento de sistemas distribuídos confiáveis.

Neste trabalho, aplicam-se conceitos de Teoria de Jogos na identificação e análise de protocolos que fomentam cooperação em três problemas distribuídos fundamentais: (i) o problema de disseminação epidémica, (ii) o problema das trocas de mensagens par-a-par em redes dinâmicas, e (iii) o problema de consenso entre processos que podem falhar por paragem. Como principais resultados, definem-se condições necessárias e suficientes para o desenvolvimento de protocolos que resolvem os problemas em questão e que são equilíbrios. Deste modo, identificam-se os requisitos necessários e suficientes para a construção de sistemas distribuídos confiáveis robustos a comportamento egoísta.

Abstract

A dependable distributed system is composed of different processes that execute a distributed protocol to provide some reliable distributed service. Typically, one assumes that all processes cooperate by executing the specified protocol, unless faults occur; if the processes do not cooperate, then the service that the system is intended to provide may be compromised. Unfortunately, the assumption that processes do not deviate from the protocol may not hold in open systems, where each process is under the control of a different entity. In fact, if the entities are selfish and they benefit from deviations, then they may change the protocol run by the processes. To avoid this problem, protocols must sustain cooperation, i.e., they must provide incentives that deny any benefits to the entities responsible for deviations.

One way of modelling selfish behaviour is to adopt the approach of Game Theory. In this approach, processes are seen as being under the control of rational agents that seek to maximize individual utility functions, interactions are modelled as games, and protocols correspond to strategies of the game that specify the actions taken at each point in time. The main goal is to devise equilibria protocols, i.e., protocols such that no agent increases its utility by causing a deviation. Equilibria protocols sustain cooperation, thus being extremely relevant to the development of dependable distributed systems.

In this work, we apply Game Theory to identify and analyse protocols that sustain cooperation in three fundamental distributed problems: (i) the problem gossip dissemination, (ii) the problem of pairwise exchanges of messages over links of a dynamic network, and (iii) the problem of consensus with crash failures. Our main results identify necessary and sufficient conditions for devising equilibria protocols that solve the aforementioned problems. These results unveil the necessary and sufficient requirements for the construction of dependable distributed systems robust to selfish behaviour.

Palavras Chave

Keywords

Palavras Chave

Comportamento Racional

Teoria de Jogos

Disseminação Epidémica

Trocas Par-a-par

Consenso

Keywords

Rational Behaviour

Game Theory

Gossip Dissemination

Pairwise Exchanges

Consensus

Acronyms

NE Nash Equilibrium

SPE Subgame Perfect Equilibrium

SE Sequential Equilibrium

\mathcal{G} -OAPE \mathcal{G} -Oblivious Adversary Perfect Equilibrium

WTP Weak Timely Punishments

STP Strong Timely Punishments

ED Eventual Distinguishability

CKD Connectivity with Knowledge of Degree

f -NE f -Nash Equilibrium

π -NE π -Nash Equilibrium

π -SE π -Sequential Equilibrium

Table of Contents

1	Introduction	1
1.1	Problem Statement	1
1.2	Contributions	3
1.2.1	Gossip Dissemination	3
1.2.2	Pairwise Exchanges in Dynamic Networks	4
1.2.3	Fair Consensus with Crashes	4
1.3	Publications	5
1.4	Document Outline	5
2	Background in Game Theory	7
2.1	Game Structure	7
2.1.1	Game Tree	8
2.1.2	Available Information	9
2.1.2.1	Information Completeness	9
2.1.2.2	Information Perfectness	9
2.1.2.3	Information Recall	10
2.1.3	Strategies	10
2.1.4	Utilities	10
2.2	Notions of Equilibrium	12
2.2.1	Nash Equilibrium	13

2.2.2	Subgame Perfect Equilibrium	14
2.2.3	Sequential Equilibrium	16
2.3	Existence and Multiplicity of Equilibria	17
2.3.1	Finite Games	18
2.3.2	Folk Theorems in Infinitely Repeated Games	18
2.3.2.1	Information	20
2.3.2.2	Communication	21
2.3.2.3	Monitoring Non-deterministic Behaviour	21
3	Related Work	23
3.1	Proofs of Folk Theorems	23
3.1.1	Perfect Public Monitoring.	23
3.1.2	Imperfect Public Monitoring	24
3.1.3	Perfect Private Monitoring	24
3.1.4	Imperfect Private Monitoring	25
3.1.5	Discussion	25
3.2	Rational Behaviour in Gossip Dissemination	28
3.3	Rational Behaviour in Pairwise Exchanges	30
3.3.1	Game Theoretical Approaches to Distributed Pairwise Exchanges	30
3.3.2	Game Theoretical Approaches to Dynamic Networks	30
3.4	Rational Behaviour in Consensus	31
3.5	Game Theoretical Approaches to Other Problems	32
4	Model	33
4.1	General Aspects	33
4.1.1	Communication	33

4.1.2	Actions	34
4.1.3	Information	34
4.1.4	Histories and Runs	35
4.1.5	Strategies and Protocols	36
4.2	Gossip Dissemination	36
4.2.1	Problem of Infinitely Repeated Gossip Dissemination	36
4.2.2	Utility	37
4.2.3	Approximate Folk Theorem	38
4.3	Pairwise Exchanges in Dynamic Networks	39
4.3.1	Problem of Infinitely Repeated Pairwise Exchanges In Dynamic Networks	39
4.3.2	Utility	39
4.3.3	Notion of Equilibrium	40
4.4	Fair Consensus with Crashes	41
4.4.1	Fair Consensus Problem	41
4.4.2	Utility	43
4.4.3	Notions of Equilibrium	43
	4.4.3.1 f -Nash equilibrium	43
	4.4.3.2 π -Nash Equilibrium	44
	4.4.3.3 π -Sequential Equilibrium	44
5	Gossip Dissemination	47
5.1	Dissemination Protocol	48
5.1.1	Algorithm	52
5.1.2	Parametrising the Protocol	58
5.2	Proof of Main Result	60

5.2.1	Cryptographic Assumptions	60
5.2.2	Sequential Equilibrium Proof	62
5.2.3	Average Utility	79
5.3	Fully Distributed Protocol	80
6	Pairwise Exchanges in Dynamic Networks	85
6.1	Key Concepts	88
6.2	Sustaining Cooperation with Strongest Adversary	91
6.2.1	Need for Timely Punishments	91
6.2.2	A Protocol for Valuable Pairwise Exchanges	95
6.2.3	Relaxing the Assumptions about the Utility	101
6.3	Sustaining Cooperation in General Pairwise Exchanges	102
6.3.1	Problems with Nonsymmetric Protocols	103
6.3.1.1	Problem of Omissions as Punishments	104
6.3.1.2	Problem of Punishments with Large Upload	106
6.3.2	Need for Eventual Distinguishability	107
6.3.3	A Protocol for General One-shot Pairwise Exchanges	113
6.3.4	Avoiding Prior Knowledge of Degree	119
6.3.5	Complexity	121
7	Fair Consensus with Crashes	125
7.1	An Impossibility Result	126
7.2	Obtaining a π -Nash equilibrium	127
7.2.1	A Naive Protocol	128
7.2.2	A π -Nash equilibrium	130
7.2.3	Analysis	133

7.3 A π -Sequential Equilibrium for Fair Consensus	152
8 Conclusions	157
Bibliography	164
A One-shot Deviation Property for \mathcal{G}^*-OAPE	165

List of Figures

2.1	Game tree of the file transfer game.	15
6.1	Results for Valuable Pairwise Exchanges (Section 6.2).	87
6.2	Results for General One-shot Pairwise Exchanges (Section 6.3).	88
6.3	Ambiguous punishment.	106
6.4	Indistinguishable stage.	109

List of Tables

2.1	Prisoners' Dilemma game.	13
2.2	Strategies of the file transfer game.	15
3.1	Comparison of folk theorems proofs.	27
5.1	Notation - gossip dissemination.	84
6.1	Notation - pairwise exchanges in dynamic networks.	89
7.1	Notation - consensus with crashes.	156

1 Introduction

A dependable distributed system is composed of different processes that execute a distributed protocol to provide some reliable distributed service. Typically, one assumes that all processes cooperate by executing the protocol, unless faults occur; if the processes do not cooperate, then the service that the system is intended to provide may be compromised. Unfortunately, the assumption that processes do not deviate from the protocol may not hold in open systems, where each process is under the control of a different entity. In fact, if the entities are selfish and they benefit from deviations, then they may change the protocol run by the processes (Cohen 2003; Hughes et al. 2005; Piatek et al. 2007). For instance, in file-sharing, where uploading large files is costly, selfish entities benefit from receiving files while not uploading data in return¹. Another example is when processes have to collectively decide whether to commit or abort a distributed transaction; the entities may manipulate the protocol to reach their most preferred decision. To address the problem of selfish behaviour, protocols must *sustain cooperation*, i.e., they must provide incentives to deny any benefit to entities that cause deviations.

1.1 Problem Statement

This thesis takes a game theoretical approach (Osborne & Rubinstein 1994) to gain further insight into how to sustain cooperation in dependable systems. In this approach, processes are seen as being under the control of rational agents that strive to maximize individual utility functions (henceforth, we use the designation agent to denote both the rational entities and the processes (computing entities) controlled by the agents). Game theory models interactions between agents as games: the messages that agents send correspond to actions in the game and protocols correspond to strategies of the game that specify the actions taken at each point in time. The aim is to devise *equilibria* protocols, i.e., protocols where no agent gains by deviating

¹This type of behaviour is known as free-riding.

(increases its utility) given that others do not deviate. These protocols are guaranteed to sustain cooperation, since no agent has incentives to deviate. Hence, they are extremely relevant to the development of dependable distributed systems.

We apply game theory to three fundamental distributed problems: (1) gossip dissemination, (2) pairwise exchanges in dynamic networks, and (3) consensus with crashes. We briefly introduce each of these problems below:

- Gossip dissemination is a method of disseminating data from a source to a set of agents. This method achieves a good tradeoff between reliability of data delivery to all agents and redundancy of data sent to each agent. In gossip dissemination, data is forwarded in an epidemic fashion: starting from the source, each agent forwards the data once after its first reception to a set of f^{fan} randomly chosen agents, where f^{fan} is a parameter known as *fanout*. It has been shown that, for values of f^{fan} of the order of $\log(n)$ (where n is the number of agents), the probability of data delivery to all agents is close to 1 while redundancy is minimized (Kermarrec et al. 2003). For this reason, gossip protocols are widely used for disseminating large amounts of data, including in database replication (Birman et al. 1999) and live streaming (Li et al. 2006; Li et al. 2008; Guerraoui et al. 2010). In this work, we address the problem of sustaining cooperation in gossip dissemination in synchronous systems. We assume that agents are rational, care about the disseminated data but avoid communication costs, and do not fail. It is well known that it is not possible to sustain cooperation if agents only interact a finite number of times (Osborne & Rubinstein 1994). However, if agents interact infinitely often, then results known as folk theorems show that we can sustain cooperation by holding agents accountable for their actions in the present with punishments in the future (Mailath & Samuelson 2007). In this work, we aim at proving a Folk Theorem for infinitely repeated gossip dissemination.
- A variety of protocols requires pairs of neighbouring agents of a dynamic network to repeatedly interact in pairwise exchanges of messages that are of mutual interest to both parties. Well known examples include file-sharing (Cohen 2003) and dissemination protocols such as (Li et al. 2006; Li et al. 2008). These protocols can operate in very diverse settings, e.g., wireless ad-hoc or peer-to-peer overlay networks. These settings possess three important characteristics. First, networks are inherently dynamic, whether due to uncontrolled mobility or maintenance of the overlay. Second, bandwidth is often scarce which implies

that communication is costly, and the amount of storage of each agent is limited. Third, information about the network topology is incomplete. Pairwise exchanges are similar to gossip dissemination in that we cannot sustain cooperation if agents interact only once. We determine necessary and sufficient conditions for sustaining cooperation in infinitely repeated pairwise exchanges over links of dynamic networks, assuming a synchronous system, that agents do not fail, and that the utilities are functions of the messages that the agents send and receive.

- In the consensus problem, each agent proposes a value and then outputs some proposed value as the consensus decision; agents must reach consensus by deciding on the same value. Consensus is a fundamental problem in distributed computing; it plays a key role in state machine replication, transaction commitment, and many other tasks where agreement among agents is required. In this work, we are also interested in the property of *fairness*, which is the property that the value of each agent is chosen as the decision with equal probability. Fairness seems critical in applications where the value proposed by each agent reflects its preference for the final decision and where we do not want agents to be able to influence the outcome of consensus unduly. We address the problem of sustaining cooperation in fair consensus in synchronous systems. We assume that agents may fail by crashing and care only about the decision of consensus, i.e., (a) an agent's utility depends only on the consensus value achieved (and not, for example, on the number of messages the agent sends) and (b) agents strictly prefer reaching consensus to not reaching consensus.

1.2 Contributions

The thesis provides a better understanding on how to address each of the aforementioned problems in distributed systems in the presence of rational agents. We enumerate the main contributions for each of the problems addressed in the thesis.

1.2.1 Gossip Dissemination

- We prove a slightly weaker version of a Folk Theorem in infinitely repeated gossip dissemination that holds for the notion of sequential equilibrium, assuming the existence of

a trusted *mediator* and that agents are computationally bounded. We also show that the role of the mediator can be distributed.

- To prove this result, we devise a protocol that disseminates data in a gossip fashion with a wide range of fanouts f^{fan} , and show that the protocol is a sequential equilibrium. At the core of this protocol, there is a monitoring mechanism providing incentives for agents to cooperate.

1.2.2 Pairwise Exchanges in Dynamic Networks

- We provide a new game theoretical model of repeated interactions in dynamic networks, where agents have incomplete information of the topology.
- We define a new notion of equilibrium for this model that refines sequential equilibrium.
- We identify conditions that are necessary and sufficient to devise equilibria solutions that sustain cooperation in the aforementioned model and require bounded memory.

1.2.3 Fair Consensus with Crashes

- We define a new notion of *ex post Nash equilibrium* appropriate for crash failures. We show that even in synchronous systems, there is no fair consensus protocol that is an ex post Nash equilibrium if there can be even one crash failure. This shows that we cannot sustain cooperation with every possible beliefs that agents may have about crashes.
- To get around our impossibility result, we assume that there is some distribution π on the failure pattern (i.e., a description of which agents fails and how they fail in terms of the messages they send to other agents). We show that under some minimal assumptions about π , if agents care only about consensus, then there is a Nash equilibrium that tolerates up to f failures, as long as $f + 1 < n$, where n is the total number of agents and f is the upper bound on the number of failures.
- We generalize sequential equilibrium to our setting, where there might be failures, and show that the strategy that gives a Nash equilibrium can be slightly extended to give a sequential equilibrium that tolerates up to f failures.

1.3 Publications

Some of the results presented in this thesis have been published as follows:

- X. Vilaça and L. Rodrigues. On the Effectiveness of Punishments in a Repeated Epidemic Dissemination Game. In *The 15th International Symposium on Stabilization, Safety, and Security of Distributed Systems (SSS 2013)*, Osaka, Japan.
- X. Vilaça and L. Rodrigues. On the Range of Equilibria Utilities of a Repeated Epidemic Dissemination Game with a Mediator. In *Proceedings of the 16th International Conference on Distributed Computing and Networking (ICDCN 2015)*, Goa, India.
- J. Halpern and X. Vilaça. Rational Consensus: *Proceedings of the 35th ACM Symposium on Principles of Distributed Computing (PODC 2016)*, Chicago, Illinois, USA.
- X. Vilaça and L. Rodrigues. Accountability in Dynamic Networks *Proceedings of the 18th International Conference on Distributed Computing and Networking (ICDCN 2017)*, Hyderabad, India.

1.4 Document Outline

Chapter 2 provides some necessary background in Game Theory. Chapter 3 presents the related work. Chapter 4 describes the system model. Chapters 5, 6, and 7 describe our contributions regarding the problems of gossip dissemination, pairwise exchanges in dynamic networks, and fair consensus with crashes, respectively. Finally, Chapter 8 concludes the thesis.

Background in Game Theory



Game Theory provides a set of mathematical tools for studying strategic interactions among agents. The core notion of Game Theory is that of a *game*, which describes the rules of interaction and the information available to agents. Agents are the players of the game, that is, they follow different strategies specifying which actions they take at each point in time. Agents are assumed to be rational, in the sense that they have individual utilities and they follow the strategies that provide them the highest possible utility. The aim is to devise equilibria strategies which, roughly speaking, are strategies such that no agent can increase its utility by deviating.

In this section we introduce the key game theoretical concepts used in our work. We divide these concepts into those related to the game structure and those related to notions of equilibria strategies. We also provide a discussion about the existence and multiplicity of equilibria strategies. A more thorough discussion of the subject is provided for instance by Fudenberg & Tirole (1991), Osborne & Rubinstein (1994), and Nisan et al (2007).

2.1 Game Structure

In a game Γ , agents take one or more actions. A strategy for the game specifies the actions that an agent takes at each point in time. Agents simultaneously choose strategies that maximize their *utilities* at the beginning of the game, i.e., they choose a strategy before being informed of the choices of other agents. The set of available strategies is determined by the *game tree* and by the *information* available to agents at each point in time.

We now describe the concepts of game tree, available information, strategy, and utility in more detail.

2.1.1 Game Tree

We consider games in the extensive form (Osborne & Rubinstein 1994). A game in the extensive form is characterized by a game tree. At every node of the tree, each agent follows one of multiple available actions. The actions followed by all agents determine the transition to the next node. Therefore, a node n is completely determined by the *history* of actions that lead to n . In our discussion, sometimes we distinguish between *one-shot* games, where agents take actions at only one node, and *sequential* games, where agents take actions at multiple nodes. Unless stated otherwise, we restrict the discussion to games where all agents simultaneously take actions at every node (before being informed of the choices of other agents) and the number of available actions is finite. Many games of interest that capture distributed interactions fall into this category, including the games analysed in this thesis.

In some cases, the same game Γ is played repeatedly over time (Mailath & Samuelson 2007). For instance, this is the case of repeated streaming sessions of sporting events. In the repeated version of Γ , time is divided into stages. In each stage, the game Γ is played once; we call Γ the stage game. Both models of *finitely* and *infinitely* repeated games have been considered in the literature. While in most cases interactions are only finitely repeated, it has been argued that a model of infinitely repeated games is appropriate for modelling games where the end-horizon of interactions is unknown, that is, where agents are always uncertain of when the game will end (Mailath & Samuelson 2007).

We now introduce some useful terminology of repeated games. We focus on stages games Γ that can be characterized by a finite game tree. In these games, an *outcome* of Γ is a leaf node of the game tree. The game tree of the repeated version of Γ consists in the repetition of the game tree of Γ for each stage t and outcomes of the game trees from stages $t' < t$. A node n of the game tree of the repeated version of Γ is said to be from stage t if the actions that agents take at n are taken in stage t . An *outcome of stage t* is a node from stage $t + 1$ that corresponds to an outcome of Γ . Finally, an *outcome o* of the repeated game is a function that maps each stage t to an outcome $o(t)$ of stage t .

2.1.2 Available Information

A key notion in any game is that of *common knowledge*. Intuitively, a fact is common knowledge if each agent (i) knows that fact, (ii) knows that every agent knows that fact, (iii) knows that every agent knows every agent knows that fact, and so on (see (Osborne & Rubinstein 1994) for a formal definition). Prior to the start of the game or at any node in the game tree, agents may acquire only partial information about the game structure and past actions of other agents, thus some of the facts about the game are not common knowledge. Considering the facts that are common knowledge, information available to agents can be modelled according to its *completeness, perfectness, and recall*.

2.1.2.1 Information Completeness

Completeness refers to information regarding the rationality of agents, available actions, and utilities. If the fact that all agents are rational, their utilities, and the available actions are common knowledge, then the game is of *complete information*, otherwise the game is of *incomplete information*. In this thesis, we consider both types of games, namely, we consider games of complete information, games where the utility of agents is not known, and games where the game tree is not known. In particular, we consider a well known class of games of incomplete information called Bayesian games. In a Bayesian game, each agent i is of one of multiple possible types. The type of the agent i completely determines its utility, and is not known by other agents. We also consider classes of games where the game tree is not known. In these games, an agent may not know the actions available to other agents at every node of the game tree (note that agents must always know their available actions).

2.1.2.2 Information Perfectness

Perfectness concerns the information available to agents at every node n in the game tree regarding the past actions of agents, i.e., the actions taken at nodes that precede n in the game tree. Games can be of *perfect* or *imperfect* information, depending on whether agents are informed of all the actions followed in the past or have only partial information about those actions, respectively. All the results of this thesis are for games of imperfect information.

In a game of perfect information, at every point in time, agents may always know the node n

at which they are currently taking actions, by using their knowledge about past actions to trace the path in the game tree from the root node to n . In a game of imperfect information, agents cannot always be sure at which node they are taking actions. That is, there are sets of nodes that provide the same information to i regarding the past actions of other agents. Formally, an information set I_i of agent i is a set of histories corresponding to nodes that provide the same information to i regarding past actions. Note that the set of actions available to an agent i is the same for all histories h from the same information set I_i . In a repeated game, if all histories from I_i are from the same stage t , then we say that I_i is from stage t .

2.1.2.3 Information Recall

Recall refers to the ability of agents to remember the actions they followed in the past. Specifically, a game is of *perfect recall* if every agent i can distinguish between nodes n^1 and n^2 from the game tree whenever n^1 precedes n^2 , such that i always recalls the actions it took at n^1 when taking actions at n^2 ; otherwise, the game is of *imperfect recall*. In this thesis, we only consider games of perfect recall. In games of perfect recall, if history h^1 precedes history h^2 , then i can distinguish h^2 from h^1 , hence h^1 and h^2 belong to separate information sets. Consequently, in repeated game, all histories from any information set I_i are from the same stage t and thus I_i is also from stage t .

2.1.3 Strategies

A strategy σ_i for agent i is a function mapping every information set I_i of i to a probability distribution on actions available to i at I_i . That is, σ_i specifies a complete plan of actions for i . In Bayesian games, the strategy is also a function of the type of each agent. A strategy profile $\vec{\sigma}$ specifies the strategies followed by all agents.

2.1.4 Utilities

We define the utility for non-repeated games and then generalize the definition to repeated games. In a non-repeated game, the utility is a function of the outcome of the game. Specifically, every agent i obtains a utility $U_i(o)$ when the outcome o is reached. This captures the preferences of i regarding the different outcomes. When selecting a strategy σ_i prior to the beginning of

the game, i must compute the *expected utility* of following σ_i . (Notice that strategies can be non-deterministic.) The definition of expected utility varies depending on whether agents have complete information:

- In a game of complete information, the strategy profile $\vec{\sigma}$ defines a probability distribution $P_{\vec{\sigma}}$ on outcomes of the game: the probability $P_{\vec{\sigma}}(o)$ of each outcome o is the probability attributed by $\vec{\sigma}$ to agents taking the actions specified in o at each information set. The expected utility $u_i(\vec{\sigma})$ of i given that agents follow $\vec{\sigma}$ is the expected value of $U_i(o)$, where here o is a random variable representing the possible outcomes distributed according to $P_{\vec{\sigma}}$.
- In a game of incomplete information, the strategy profile $\vec{\sigma}$ does not suffice to compute the expected utility; agents must also form expectations regarding the missing information. In particular, in Bayesian games, agents must form an expectation regarding the types of other agents. In this thesis, we explore two different approaches for modelling this expectation, namely the *ex post* and *ex ante* approaches. In the *ex post* approach, we fix the types of agents, so that the agents essentially know what the types are when computing their expected utility. Given the strategy profile $\vec{\sigma}$ and any vector of types $\vec{\theta}$ specifying the type θ_j of each agent j , there is a probability distribution $P_{\vec{\sigma}, \vec{\theta}}$ on outcomes. The expected utility of i when agents follow $\vec{\sigma}$ conditioned on the vector of types being $\vec{\theta}$, denoted by $u_i(\vec{\sigma} \mid \vec{\theta})$, is the expected value of $U_i(o)$, where now o is distributed according to $P_{\vec{\sigma}, \vec{\theta}}$. In the *ex ante* approach, we assume that there is a probability distribution π on vectors of types. Given π and $\vec{\sigma}$, there is a probability distribution $\pi_{\vec{\sigma}}$ on outcomes. The expected utility $u_i(\vec{\sigma})$ of agent i when agents follow $\vec{\sigma}$ is the expected value of $U_i(o)$, where now o is distributed according to $\pi_{\vec{\sigma}}$. In part of our work, we also consider games where the game tree is not known. We can also adopt the *ex post* (*ex ante*) approach to compute the expected utility in these games, except that, instead of fixing the vector of types (resp., instead of assuming a distribution π on types), we fix the game tree (resp., we assume a probability distribution on game trees).

More generally, when the game Γ is repeated in $T \geq 1$ stages (where T may be infinite), the expected utility is a function of the utility obtained in each stage. In this thesis, we consider the standard definition of discounted utilities. In this definition, utilities obtained in future stages

are less valuable than present ones. That is, the utility obtained in a stage t as computed in the present decays exponentially as t increases; the expected utility of i is then the sum of the expected utilities obtained in all stages as computed in the present.

More precisely, we consider a discount factor $\delta \in (0, 1)$ which discounts future utilities to the present. Given an outcome o of the repeated game, the utility $U_i(o)$ of agent i is $\sum_{1 \leq t \leq T} \delta^{t-1} U_i(o(t))$. In a game of complete information, a strategy profile $\vec{\sigma}$ defines a probability distribution $P_{\vec{\sigma}}$ on outcomes of the repeated game. The expected utility $u_i(\vec{\sigma})$ of i is the expected value of $U_i(o)$, where o is a random variable distributed according to $P_{\vec{\sigma}}$. In Bayesian games, the expected utility is defined in the same way, except the probability distributions on outcomes are $P_{\vec{\sigma}, \vec{\theta}}$ and $\pi_{\vec{\sigma}}$ when using the ex post and ex ante approaches, respectively, where $\vec{\theta}$ is a vector of types and π is a probability distribution on vectors of types. Again, we can extend these definitions to games where agents have incomplete information of the game tree.

In infinitely repeated games, we can see the discount factor δ as the probability of the game ending in any given stage: agents interact repeatedly and are never sure of when the interactions will end; interactions end in stage t with independent probability δ . The possibility of interactions ending at any stage is what causes agents to value future utilities less than present ones: agents may not be around in the future to receive those utilities.

2.2 Notions of Equilibrium

The main goal of Game Theory is to predict the strategies followed by agents. The usual assumption is that agents follow equilibria strategy profiles¹. Multiple notions of equilibrium have been proposed in the literature, which establish different criteria for identifying equilibria strategy profiles. We now define the most relevant notions to our work, which are *Nash equilibrium* and *sequential equilibrium*. For the definition of sequential equilibrium, it is useful to first define the related notion of *subgame perfect equilibrium*, which we also include here.

¹When the game has multiple equilibria strategy profiles, agents must agree on some equilibria strategy profile. In distributed systems, the usual assumption is that agents implicitly agree to follow the pre-defined protocol.

	C	D
C	1, 1	-1, 2
D	2, -1	0, 0

Table 2.1: Prisoners' Dilemma game.

2.2.1 Nash Equilibrium

We provide definitions of Nash equilibrium for the classes of games of complete and incomplete information considered in this thesis. In games of complete information, a Nash equilibrium (Nash 1950; Nash 1951) is a strategy profile $\vec{\sigma}$ such that, for every agent i and strategy σ'_i , we have $u_i(\vec{\sigma}) \geq u_i((\sigma'_i, \vec{\sigma}_{-i}))$, where $(\sigma'_i, \vec{\sigma}_{-i})$ is the strategy profile where only i deviates from $\vec{\sigma}$ by following σ'_i . Intuitively, $\vec{\sigma}$ is a Nash equilibrium if no agent i gains by deviating from the strategy σ_i given that other agents do not deviate.

In Bayesian games, the definition of Nash equilibrium depends on the approach used to model the expectation of agents regarding types. Namely, we define the notions of *ex post Nash* equilibrium and *ex ante Nash* equilibrium, corresponding to the ex post and ex ante approaches of modelling expectations about types, respectively. An ex post Nash equilibrium is a strategy profile $\vec{\sigma}$ such that, for all vectors $\vec{\theta}$ of types, $\vec{\sigma}$ is a Nash equilibrium that results from agents computing their expected utility using the probability distribution $P_{\vec{\sigma}, \vec{\theta}}$ defined on outcomes. That is, no agent has incentives to deviate even if it knows what the types of other agents are. An ex ante Nash equilibrium is a Nash equilibrium $\vec{\sigma}$ that results from agents using a probability distribution π on vectors of types to compute their expected utility, where expectation is taken relative to the probability distribution $\pi_{\vec{\sigma}}$ on outcomes. The definitions of ex post (ex ante) Nash equilibrium for games where the game tree is not known is identical, except we fix (resp., consider distributions on) game trees instead of vectors of types.

The canonical example of a game that has a Nash equilibrium is the Prisoners' Dilemma with utilities depicted in Table 2.1. In this game, two agents 1 and 2 simultaneously decide whether to cooperate (C) or defect (D). The utility of agents is a function of the benefits and costs of cooperation. Specifically, if agent i cooperates, then agent i incurs a cost of 1 and agent $1 - i$ obtains a benefit 2; if agent i defects, then agent i incurs no cost and agent $1 - i$ obtains no benefit. In this game, the only Nash equilibrium is the strategy profile (D,D), where both agents defect. Since both agents would do better if they cooperated, this is considered to be a social dilemma. We discuss this problem in more detail and possible solutions in Section 2.3.

2.2.2 Subgame Perfect Equilibrium

The notion of subgame perfect equilibrium (Selten 1965) (SPE) refines Nash equilibrium for games of complete and perfect information; it addresses the problem of Nash equilibria strategy profiles that rely on empty threats of punishment. Specifically, a strategy profile may rely on empty threats if, for instance, whenever an agent i deviates and some agent j detects the deviation, j punishes i by taking actions that decrease the utilities of both i and j ; the threat of punishment by j is empty, because j is never willing to carry such costly punishment. The notion of SPE does not admit equilibria strategy profiles that rely on empty threats of punishment.

More precisely, the problem with the definition of Nash equilibrium is that it does not consider deviations at nodes *off the equilibrium path*. That is, an agent i decides whether it will deviate from the equilibrium strategy profile $\vec{\sigma}$ at the beginning of the game, while expecting other agents to follow $\vec{\sigma}_{-i}$; i does not expect to take actions at a node in the game tree that can only be reached if someone else deviates from $\vec{\sigma}_{-i}$. Therefore, even if $\vec{\sigma}$ is a Nash equilibrium, agent i may gain by deviating once some other agent deviates.

To better understand the issue, consider an example of a file transfer game with two agents p_1 and p_2 and a trusted third party. The game tree is depicted in Fig. 2.1. In this game, agent p_1 is the first to move at node 1 by deciding whether to send (S) or not (N) a file to agent p_2 . Then, agent p_2 can complain (C) to a trusted third party about agent p_2 not sending the file or can say nothing (N). The utilities of all possible combinations of actions are included in the pairs near the outcomes of the game tree (Fig. 2.1), where a pair (u_1, u_2) represents the utility u_1 of agent p_1 and the utility u_2 of agent p_2 : if agent p_1 sends the file, then p_1 incurs a cost 1 and p_2 obtains a benefit 1, otherwise, p_1 incurs no cost and p_2 obtains no benefit; if p_2 complains, then the third party punishes both agents, causing a utility loss of 1. (Agent p_2 may be lying, so it is safer to punish both agents.)

The strategies and respective utilities are summarized in Table 2.2, where the lines represent the strategies of agent p_1 , and the columns represent the strategies of agent p_2 . Specifically, agent p_1 may send (S) or not (N) the file. The strategy of agent p_2 specifies whether p_2 complains (C) or not (N) at both the nodes 2 and 3 where agent p_2 takes an action; this strategy takes the form a_1a_2 , where a_1 and a_2 are the actions taken by agent p_2 after agent p_1 follows S and N , respectively. For instance, NC is the strategy where agent 2 complains iff agent p_1 does not

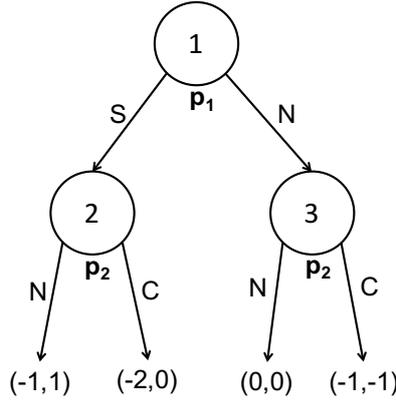


Figure 2.1: Game tree of the file transfer game.

	NN	NC	CN	CC
S	-1, 1	-1, 1	-2, 0	-2, 0
N	0, 0	-1, -1	0, 0	-1, -1

Table 2.2: Strategies of the file transfer game.

send the file.

This game admits exactly one Nash equilibrium where agent p_1 sends the file to agent p_2 , which is (S, NC) , i.e, agent p_1 sends the file and agent p_2 complains iff agent p_1 does not send the file. This equilibrium relies on an empty threat of punishment: the only incentive for agent p_1 to send the file is because it expects agent p_2 to complain in case agent p_1 does not send the file; however, when agent p_1 does not send the file, agent p_2 gains by deviating and not complaining. Hence, agent p_2 may not follow the strategy NC .

The definition of SPE avoids this problem by requiring the strategy profile to be a Nash equilibrium even when considering deviations off the equilibrium path. Specifically, for every node n of the game tree, there is a subgame induced by n : the game defined by the subtree starting at node n . A strategy profile $\vec{\sigma}$ is a SPE if, for every node n from the game tree, $\vec{\sigma}$ is a Nash equilibrium of the subgame induced by n (Selten 1965). Under this definition, the strategy profile (S, NC) is not an equilibrium in our example, since it is not a Nash equilibrium of the subgame induced by node 3

We now provide the formal definition of SPE for repeated games, which can be easily instantiated to non-repeated games. Given a history h from stage t and a strategy profile $\vec{\sigma}$, let $u_i(\vec{\sigma} | h)$ denote the expected utility of i conditioned on h being realized and agents following $\vec{\sigma}$ at and after h , which we define as follows. Let $\mathcal{O}(h)$ be the set of outcomes o of the repeated game compatible with h (i.e., h corresponds to a node n that precedes the outcome $o(t)$ in the

game tree for some stage t). Note that $\vec{\sigma}$ defines a probability distribution $P_{\vec{\sigma},h}$ on outcomes in $\mathcal{O}(h)$. As in the definition of $u_i(\vec{\sigma})$ for repeated games, the expected utility $u_i(\vec{\sigma} | h)$ is the expected value of $U_i(o)$, except now the expectation is taken relative to $P_{\vec{\sigma},h}$. A strategy profile $\vec{\sigma}$ is a SPE iff, for every history h , agent i , and strategy σ'_i , we have $u_i(\vec{\sigma} | h) \geq u_i((\sigma'_i, \vec{\sigma}_{-i}) | h)$. Note that, even though this definition is appropriate only for games of both perfect and complete information, we can directly extend it to games of incomplete information using both ex post and ex ante approaches.

2.2.3 Sequential Equilibrium

The notion of sequential equilibrium (SE) is a refinement of SPE for games of imperfect information, where agents are not perfectly informed of the actions of other agents (Kreps & Wilson 1982). The main difference lies in the way agents compute their expected utility. Recall that, in a game of imperfect information, agents take actions at information sets, where an information set I_i of agent i is a set of histories that provide the same information to i about past actions of other agents. Roughly speaking, a strategy profile $\vec{\sigma}$ is a SE if, for every agent i and information set I_i , agent i does not gain by deviating from σ_i at and after I_i , given that other agents also do not deviate. Hence, $\vec{\sigma}$ is essentially a SPE, except agents compute the expected utility conditioning on information sets instead of histories.

Specifically, let $u_i(\vec{\sigma} | I_i)$ denote the expected utility of agent i conditioned on information set I_i . To define $u_i(\vec{\sigma} | I_i)$, one needs a probability distribution $P_{\vec{\sigma},I_i}$ on outcomes in $\mathcal{O}(I_i)$, where $\mathcal{O}(I_i)$ is the set of outcomes of the repeated game such that the information set of i is I_i (i.e., $\mathcal{O}(I_i)$ is the union of $\mathcal{O}(h)$ for all $h \in I_i$). The obvious way to define $P_{\vec{\sigma},I_i}$ is to condition on agents following $\vec{\sigma}$ from the beginning and on the outcome being in $\mathcal{O}(I_i)$. Unfortunately, if I_i is inconsistent with $\vec{\sigma}$ in that the probability $P_{\vec{\sigma}}(o)$ of outcome o is 0 for every outcome $o \in \mathcal{O}(I_i)$, then $P_{\vec{\sigma},I_i}$ is not well defined. To address this problem, in the definition of SE, Kreps & Wilson (1982) assume that agents form beliefs about the past behaviour of other agents, which are captured by a belief system μ : for every information set I_i , μ defines a probability distribution μ_{I_i} on histories in I_i ; thus, $\mu_{I_i}(h)$ represents the belief by agent i that agents followed the actions specified in h . A belief system μ and a strategy profile $\vec{\sigma}$ define a probability distribution $\mu_{\vec{\sigma},I_i}$ on outcomes in $\mathcal{O}(I_i)$ in the obvious way. Given this, the expected utility $u_i(\vec{\sigma} | I_i)$ of agent i conditioned on I_i is the expected value of $U_i(o)$, where the expectation is taken relative to $\mu_{\vec{\sigma},I_i}$.

According to the definition of SE by Kreps & Wilson, the belief system μ cannot be arbitrary. More precisely, the authors require μ to be consistent with the equilibrium strategy profile, where a belief system μ is said to be *consistent* with $\vec{\sigma}$ iff there exists a sequence $\vec{\sigma}^1, \vec{\sigma}^2, \dots$ of completely mixed strategy profiles (attribute positive probability to agents following all possible actions at every information set) that converges to $\vec{\sigma}$ such that, for every history $h \in I_i$,

$$\mu_{I_i}(h) = \lim_{M \rightarrow \infty} \frac{P_{\vec{\sigma}^M}(h)}{\sum_{h' \in I_i} P_{\vec{\sigma}^M}(h')},$$

where $P_{\vec{\sigma}'}(h'')$ is the probability of history h'' being realized given that agents follow strategy profile $\vec{\sigma}'$. Intuitively, the definition of consistent belief states that, if the information set I_i of agent i is consistent with all agents following $\vec{\sigma}$ in that there is an outcome $o \in \mathcal{O}(I_i)$ that results from agents using $\vec{\sigma}$ (i.e., $P_{\vec{\sigma}}(o) > 0$), then i should believe that no agent deviated from $\vec{\sigma}$. In this case, the probability distribution $\mu_{\vec{\sigma}, I_i}$ on outcomes in $\mathcal{O}(I_i)$ is defined by conditioning on agents following $\vec{\sigma}$ from the beginning and on the outcome being in $\mathcal{O}(I_i)$. However, if I_i is inconsistent with $\vec{\sigma}$, then i has to believe that agents followed some alternative strategy profile $\vec{\sigma}'$. In this case, $\mu_{\vec{\sigma}, I_i}$ is defined by conditioning on agents following $\vec{\sigma}'$ from the beginning and the outcome being in $\mathcal{O}(I_i)$.

Having defined expected utilities conditioned on information sets and consistent beliefs, we can now provide the formal definition of SE. A strategy profile $\vec{\sigma}$ is a SE iff there exists a belief system μ consistent with $\vec{\sigma}$ such that, for every agent i , information set I_i , and strategy σ'_i , we have $u_i(\vec{\sigma} | I_i) \geq u_i((\sigma'_i, \vec{\sigma}_{-i}) | I_i)$. Intuitively, a strategy profile is an SE if there is a consistent belief system μ such that no agent i gains by deviating at any information set I_i , given that i 's beliefs at I_i about past behaviour of other agents are captured by μ .

2.3 Existence and Multiplicity of Equilibria

The most important results in game theory pertain the number of equilibria strategy profiles that a game admits. We discuss the most relevant results in finite and infinitely repeated games.

2.3.1 Finite Games

Nash (1950, 1951) proved that every finite one-shot game admits at least one Nash equilibrium. Using backwards induction, it follows that every finite game in the extensive form admits a sequential equilibrium (and in particular admits a subgame perfect equilibrium) (Osborne & Rubinstein 1994): starting at nodes of the game tree that immediately precede only outcomes, agents follow Nash equilibria strategy profiles for the subgames defined by those nodes (which exist according to Nash's result); the same reasoning is applied backwards to define a strategy profile that is a Nash equilibrium at every subgame and thus is a sequential equilibrium.

The backwards induction result implies that every finitely repeated game has a sequential equilibrium, where agents follow a Nash equilibrium strategy profile in every stage. Unfortunately, this is the only type of sequential equilibria that exists (Osborne & Rubinstein 1994). In some games, such equilibria result in undesirable outcomes. For instance, in the finitely repeated Prisoners' Dilemma game (Table 2.1), the only subgame perfect equilibrium consists in agents always defecting. Since agents would obtain a higher utility if both agents cooperated, this situation is seen as a social dilemma. In fact, many distributed interactions such as pairwise exchanges have the structure of a Prisoners' Dilemma game. In those interactions, the social dilemma is an obstacle to sustaining cooperation. Solutions to this dilemma for finitely repeated games include the assumption that agents can sometimes be irrational by not always following the strategy that maximizes their utility (Fudenberg & Maskin 1986). As discussed in the next section, in infinitely repeated games the aforementioned dilemma does not arise, so we can sustain cooperation in these games even if all agents are rational.

2.3.2 Folk Theorems in Infinitely Repeated Games

In infinitely repeated games, results known as *folk theorems* show that the number of sequential equilibria strategy profiles of many infinitely repeated games is infinite (Mailath & Samuelson 2007). In particular, we can attain desirable outcomes in equilibria strategy profiles, e.g., we may persuade agents to always cooperate in the infinitely repeated Prisoners' Dilemma game. Roughly speaking, a folk theorem states that, for every possible utility u that an agent i may obtain in a single stage of a game Γ , there is an equilibrium strategy profile $\vec{\sigma}$ for the infinitely repeated version of Γ such that if agents use $\vec{\sigma}$, then u is the *average* utility of i per

stage in the infinitely repeated version of Γ .

More precisely, a folk theorem is defined in terms of *feasible* and *individually rational* vectors of utilities, notions that we now define. Let $\vec{\eta}$ denote a vector of utilities, which specifies the utility η_i of every agent i . A vector $\vec{\eta}$ is said to be *feasible* if there exists a strategy profile $\vec{\tau}$ for Γ such that η_i is the expected utility of i in Γ when agents follow $\vec{\tau}$. A vector $\vec{\eta}$ is said to be *individually rational* if $\eta_i \geq u_i^*$ for all agents i , where u_i^* is the minimax utility of i , defined as $u_i^* = \min_{\tau_{-i}} \max_{\tau_i} u_i(\vec{\tau})$. Intuitively, u_i^* is the utility of i when other agents apply the worst possible punishment to i . The average utility of agent i when agents follow a strategy profile $\vec{\sigma}$ in the infinitely repeated version of Γ , which we denote by $\bar{u}_i(\vec{\sigma})$, is given by $\bar{u}_i(\vec{\sigma}) = (1 - \delta)u_i(\vec{\sigma})$, where $u_i(\vec{\sigma})$ is the expected utility of i in the infinitely repeated game. An *approximate* folk theorem for the infinitely repeated game Γ states that, for all constants $\epsilon > 0$ and all feasible and individually rational vectors $\vec{\eta}$ of utilities of Γ , there exists $\delta^* \in (0, 1)$ such that for all $\delta \in (\delta^*, 1)$ there is a strategy profile $\vec{\sigma}$ for the infinitely repeated game such that $\vec{\sigma}$ is an equilibrium and $|\bar{u}_i(\vec{\sigma}) - \eta_i| \leq \epsilon$ for all agents i ; an (exact) folk theorem states that the above holds for $\epsilon = 0$ (Fudenberg & Maskin 1986; Fudenberg & Maskin 1991).

A vast literature proves folk theorems in a variety of models (see (Mailath & Samuelson 2007) for a complete survey). Usually, the proof of a folk theorem is by construction: for each feasible and individually rational vector of utilities $\vec{\eta}$, we define a strategy profile $\vec{\sigma}^*$ and show that if δ is sufficiently close to 1, then $\vec{\sigma}^*$ is an equilibrium and the average utility of every agent i is exactly η_i (or arbitrarily close to η_i in an approximate folk theorem). Normally, the structure of $\vec{\sigma}^*$ is as follows. Agents monitor the behaviour of every agent i to see if i is following $\vec{\sigma}^*$. If agents detect that some agent i deviates, then they trigger a punishment of i , where they take actions in multiple stages that decrease the utility of i to the minimax value. Note that for all feasible and individually rational utilities η_i the value η_i is larger than the minimax value, hence the utility of agent i decreases during a punishment. Proofs of folk theorems show that, if i deviates at stage t , then the deviation is detected with high probability. It follows that, if δ is sufficiently close to 1, then the (expected) utility loss of i for being punished in stages $t' > t$ is greater than the immediate gain of deviating at stage t , so i does not gain by deviating at stage t .

Existing proofs of folk theorems differ in how they model the information that agents acquire about other agents' actions, the model of communication between agents, and the approach used

to monitor non-deterministic behaviour. We now present a taxonomy of the different types of models and approaches considered in existing work. In Chapter 3, we discuss existing proofs of folk theorems in light of this taxonomy.

2.3.2.1 Information

Information about the agents' actions is captured by a *monitoring infrastructure*, which provides a signal ω_i to every agent i every time a node in the game tree is reached. The signal ω_i gives (at least partial) information to i about the actions of other agents. For instance, if i can perfectly observe the actions of other agents such that i always knows the node at which agents are taking actions, then ω_i represents the current node in the game tree. In some models, the monitoring infrastructure is restricted by an underlying network, such that an agent i obtains information about the actions of agent j only if j is a neighbour of i in the network. In these models, the signal ω_i consists in a vector of independent signals ω_{ij} for each neighbour j of i .

Given a node x of the game tree, a monitoring infrastructure is characterized by a probability distribution P_x on vectors $\vec{\omega}$ of signals, where $P_x(\vec{\omega})$ is the probability that the monitoring infrastructure provides the signals given by $\vec{\omega}$ when the node x is reached. A monitoring infrastructure can be classified according to the *accuracy* and *symmetry* of the signals it provides to the agents:

- Signal accuracy: monitoring is said to be *globally perfect* if every agent i learns the actions of other agents at every node x , i.e., $P_x(\vec{\omega}^*) = 1$, where ω_i^* corresponds to x for all agents i ; monitoring is said to be *locally perfect* if the monitoring infrastructure is restricted by a network, and i can perfectly observe the actions of every neighbour j ; otherwise, monitoring is said to be *imperfect*.
- Signal symmetry: monitoring is said to be *public* if $\omega_i = \omega_j$ for all agents i and j and vectors of signals $\vec{\omega}$ that the monitoring infrastructure provides with positive probability; otherwise monitoring is said to be *private*.

Henceforth, we say that monitoring is perfect if it is either globally or locally perfect.

2.3.2.2 Communication

Existing work consider three main types of models of communication, namely, models where there is no communication, models where agents can communicate via *cheap talk*, and models where communication is *costly*:

- No communication: there is no communication between agents, so all the information that agents have about other agents' actions comes from the signals that the monitoring structure provides to agents.
- Cheap talk: in addition to taking actions, agents can send messages containing information about the signals they receive from the monitoring infrastructure; agents do not incur costs for sending messages.
- Costly communication: agents can send messages and they incur costs for doing so.

2.3.2.3 Monitoring Non-deterministic Behaviour

In most stage games Γ , the set of feasible and individually rational utilities contains utilities u_i for agent i that correspond to non-deterministic strategy profiles of Γ . For instance, in the Prisoners' Dilemma, the pair of feasible and individually rational utilities $(1/2, 1/2)$ corresponds to the strategy where agents cooperate with probability $1/2$ (note that the minimax utility of both agents is 0 in this game). In these games, to prove a folk theorem by construction one must define a strategy profile where the average utility of agent i is u_i even if u_i corresponds to a non-deterministic strategy $\vec{\tau}$ profile for the stage game. The obvious way to do this is to require agents to follow $\vec{\tau}$ in every stage. However, if the only information that agents have about other agents' actions are the signals that the monitoring infrastructure provides, then an agent i cannot tell whether some other agent j is following $\vec{\tau}$ in every stage: agent i may know that j took action a_j at information set I_j , but i does not know whether j took a_j with probability $\tau_j(a_j | I_j)$. Therefore, agents cannot persuade agent j to follow $\vec{\tau}$ in every stage. Different proofs of folk theorems address this difficulty using one of three main approaches, namely, they assume *public randomization*, they replace non-deterministic actions with *deterministic streams of actions*, or they make agents *indifferent between actions* at every information set:

- Public randomization: at every information set I_i , a public randomization device outputs random numbers that determine the action a_i that i should follow with probability $\tau_i(a_i | I_i)$; since the device is public, all agents can consult the device to check if i is following τ_i , thus they can detect a deviation and trigger a punishment of i if i deviates from τ_i in some stage.
- Deterministic stream of actions: in this approach, the strategy σ_i^* for the repeated game defines a deterministic stream of actions for agent i that yields an average utility to i that is at least arbitrarily close to the utility of i when agents follow $\vec{\tau}$ in a stage; for instance, if τ_i requires that i takes action a_i^1 or a_i^2 with equal probability $1/2$, then σ_i^* requires i to take a_i^1 in half of the stages and to take action a_i^2 in the remaining stages.
- Indifference between actions: in this approach, at every information set I_i , agent i obtains the same expected utility for taking each action a_i in the support of τ_i at I_i (i.e., $\tau_i(a_i | I_i) > 0$), so i does not gain from not selecting those actions at random according to the distribution defined by τ_i , and other agents do not have to punish i for taking any of those actions.

Summary

In this chapter we have presented the key game theoretical concepts used in our work. In the next chapter, we will survey some of the most relevant proofs of folk theorems and applications of game theory to dependable distributed systems.

3

Related Work

In this chapter, we discuss the most relevant literature that proves folk theorems and that addresses rational behaviour in the problems of gossip dissemination, pairwise exchanges in dynamic networks, and consensus. We conclude the chapter with a brief mention of relevant work that applies game theory to other distributed problems.

3.1 Proofs of Folk Theorems

We start with a discussion of the proofs of folk theorems that are more relevant to our work. We characterize these proofs according to the models of information, models of communication, and the type of games that they consider, and according to the approach that these proofs use to monitor non-deterministic behaviour. We discuss proofs that consider the four main types of monitoring in turn, namely, perfect public, perfect private, imperfect public, and imperfect private monitoring.

3.1.1 Perfect Public Monitoring.

Fudenberg & Maskin (1986, 1991) and Abreu et al. (1994) were the first to prove a folk theorem in one-shot games with n agents for the notion of subgame perfect equilibrium, assuming that monitoring is perfect and public and that agents do not communicate. Fudenberg & Maskin first showed (1986) that a restriction on the utilities named *full dimensionality* is sufficient to prove a folk theorem assuming public randomization; later (1991), they proved an identical result except in the equilibrium strategy profile agents follow deterministic streams of actions that yield any feasible and individually rational utility. Abreu et al. (1994) showed that in one-shot games with n agents the condition of full dimensionality is not necessary and derived a weaker condition known as NEU that is still sufficient to prove a folk theorem; the authors also use the approach of generating deterministic streams of actions to yield any feasible and individually rational

average utility to agents.

Other relevant proofs were provided in the work of Dolev et al. (2011), Rubinstein & Wolinsky (1995), and Mailath & Samuelson (2007). Dolev et al. approached the problem of devising equilibria strategy profiles that correspond to self-stabilizing protocols (Dolev 2000) with optimal complexity in systems that may be temporarily controlled by malicious agents. The authors prove a folk theorem for the notion of subgame perfect equilibrium assuming that monitoring is public and perfect. Rubinstein & Wolinsky and Mailath & Samuelson discussed how to extend the result by Fudenberg & Maskin (1991) to sequential games under the assumption that agents can apply punishments that discriminate agents, i.e., it is possible to punish any given agent i by following a Nash equilibrium of the stage game.

3.1.2 Imperfect Public Monitoring

The main result for models of imperfect public monitoring is due to Fudenberg et al. (1994). The authors prove folk theorems in one-shot games with n agents for the notion of subgame perfect equilibrium, assuming that there is no communication between agents. In their proof, they define a strategy profile $\vec{\sigma}^*$ such that agents are indifferent between actions in the support of $\vec{\sigma}^*$ at every information set.

3.1.3 Perfect Private Monitoring

Existing work proves folk theorems with perfect private monitoring assuming that the observations of agents are conditioned by an underlying network. Kinatader (2008) proves a folk theorem for sequential equilibrium in one-shot games with n agents, in a model where agents cannot communicate and the network restricts the observations of agents but not the actions. Their equilibrium strategy profile generates deterministic streams of actions that yields any feasible and individually rational utility. Laclau (2012) proves a similar result in a model where agents can communicate via cheap talk and the network restricts both the actions and observations of agents. Laclau also defines a strategy profile where agents take deterministic streams of actions.

3.1.4 Imperfect Private Monitoring

In models of imperfect private monitoring, existing work proved folk theorems for the notion of sequential equilibrium for the three models of communication, i.e., models with no communication, models with cheap talk, and models with costly communication. To the best of our knowledge, no existing proofs consider restrictions imposed by the network.

No Communication. Sekiguchi (1997), Piccione (2002), Ely & Välimäki (2002), and Matsushima (2004) prove folk theorems in one-shot games with two agents. The authors devise equilibria strategy profiles $\vec{\sigma}^*$ where agents are indifferent between actions in the support of $\vec{\sigma}^*$ at all information sets. Bhaskar & Obara (2002) prove a folk theorem in one-shot games with two agents, assuming that agents have access to a public randomization device.

Cheap Talk. Proofs by Compte (1998), Kandori et al. (1998), Fudenberg & Levine (2007), and Obara (2009) show that folk theorems hold in one-shot games with n agents. The authors devise strategy profiles where agents take deterministic streams of actions.

Costly Communication. Sugaya (2011) proves a folk theorem in one-shot games with two agents assuming that agents can send costly messages. In the equilibrium strategy profile that they define, agents are indifferent between actions at every information set.

3.1.5 Discussion

Table 3.1 summarizes the above discussion. As discussed in Chapter 4, we consider a model where monitoring is locally perfect, agents can communicate via costly messages, and both the actions and observations of agents are restricted by a network: the action of agent i consists in the messages that i sends to each neighbour j of i in the network and the message that i sends to j is observed by both i and j and only by these two agents. In addition, we consider infinitely repeated sequential games with $n \geq 2$ agents, and we do not assume that there is a public randomization device. In our results about pairwise exchanges, we also consider a dynamic network where agents have incomplete information of the network topology. Therefore, the models of existing proofs of folk theorems differ from our model in the considered game type,

communication model, and network restrictions.

In more detail, as we can see from Table 3.1, very few works consider sequential games. Rubinstein & Wolinski (1995) and Mailath & Samuelson (2007) discuss how to prove a folk theorem in games with perfect public monitoring. Sorin (1995) analyses infinitely repeated sequential games with imperfect monitoring, but he does not prove a folk theorem and assumes that signals are public, so he does not consider network restrictions.

Even in one-shot games, most proofs of folk theorems do not model network restrictions. The exceptions are the proofs by Kinaterder (2008) and Laclau (2012), which consider a model of perfect private monitoring where observations are restricted by a network. Since these results are for one-shot games, they do not apply to gossip dissemination. Moreover, Kinaterder and Laclau assume that the networks are static and that agents have complete information of the network topology, hence their results do not apply to pairwise exchanges in dynamic networks either. Finally, only Laclau models communication between agents, but he assumes that agents can communicate via cheap talk. As shown by our work, sustaining cooperation with costly communication is not trivial. Sugaya (2011) proves a folk theorem with costly communication in a model stronger than ours in that he considers imperfect monitoring. However, he does not consider network restrictions, and his result is restricted to one-shot games with two agents.

Monitoring Type	References	Game type	N. agents	Non-determ. Monitoring	Communication	Network Restrictions
Public Perfect	(Fudenberg & Maskin 1986)	One-shot	n	Public Randomization	None	None
	(Fudenberg & Maskin 1991; Abreu et al. 1994; Dolev et al. 2011)	One-shot	n	Stream	None	None
	(Rubinstein & Wolinsky 1995; Mailath & Samuelson 2007)	Sequential	n	Stream	None	None
Public Imperfect	(Fudenberg et al. 1994)	One-shot	n	Indifference	None	None
Private Perfect	(Kinaterder 2008)	One-shot	n	Stream	Cheap-talk	Communication
	(Laclau 2012)	One-shot	n	Stream	Cheap-talk	Communication & Actions
Private Imperfect	(Sekiguchi 1997; Piccione 2002; Ely & Välimäki 2002; Matsushima 2004)	One-shot	2	Indifference	None	None
	(Compte 1998; Kandori & Matsushima 1998; Fudenberg & Levine 2007; Obara 2009)	One-shot	n	Stream	Cheap-talk	None
	(Sugaya 2011)	One-shot	2	Indifference	Costly	None

Table 3.1: Comparison of folk theorems proofs.

3.2 Rational Behaviour in Gossip Dissemination

We now discuss work that addresses rational behaviour in gossip dissemination. Existing literature considers two main approaches for gossip dissemination, namely, symmetric (or balanced) exchanges and pushed-based gossip dissemination (Vilaca & Rodrigues 2013; Li et al. 2006; Li et al. 2008; Guerraoui et al. 2010; Mokhtar et al. 2014). In both approaches, a source disseminates a stream of blocks of data. For each block, the source sends the block to a set of f^{fan} randomly selected agents, where f^{fan} is the fanout. The two models differ in how agents then disseminate the blocks among themselves after receiving the blocks from the source. In symmetric exchanges, pairs of agents periodically exchange an equivalent number of missing blocks. In push-based gossip dissemination, upon receiving each block for the first time, each agent forwards the block to f^{fan} randomly selected agents. Existing work adopted both game theoretical and practical approaches to deal with rational behaviour in the two considered models.

Vilaca & Rodrigues (2013) perform a game theoretical analysis of infinitely repeated push-based gossip dissemination. The authors show that incentives based on direct reciprocity such as tit-for-tat are not effective at deterring deviations in gossip dissemination if the reliability is too high. As a result, in a proof by construction of a folk theorem in gossip dissemination, we must devise a strategy profile where agents send messages containing information about their private observations. Even though Vilaca & Rodrigues do not prove a folk theorem, their result justifies our choice of devising a distributed monitoring mechanism to prove a variant of a folk theorem in gossip dissemination.

Guerraoui et al. (2010) and Mokhtar et al. (2014) propose LifTinG and AcTinG, respectively, which are push-based gossip dissemination systems. Both systems use monitoring to persuade agents to disseminate blocks of data; they differ in the techniques used to monitor the behaviour of agents. In LifTinG, agents perform direct verifications of the messages that other agents send to see if they forward every received block of data to f^{fan} agents. Moreover, agents periodically cross-check the histories of blocks that they received to check if some agent did not forward blocks at random. If a deviation is detected, then the deviating agent is evicted from the system as a punishment. In AcTinG, agents send in each message an unforgeable log of operations. Each operation registers a message that an agent sends (receives) and the respective destination (source). Whenever an agent sends or receives a message, he must append the corresponding

operation to the log and send it to other agents, or else a deviation is detected. The logs included in messages allow agents to detect, with high probability, deviations in which an agent does not forward an event according to the protocol. As in *LifTing*, if a deviation is detected, then the deviating agent is evicted from the system. Neither Guerraoui et al. nor Mokhtar et al. used game theory to analyse gossip dissemination games.

Li et al. proposed *BAR Gossip* (2006) and *FlightPath* (2008), which are gossip dissemination protocols that enforce symmetric exchanges between rational agents. Specifically, agents are organized into an overlay that is deterministically defined by pseudo-random number generators. Periodically, every two neighbouring agents engage in a symmetric exchange by taking the following steps: (1) they exchange a list of blocks that they received, (2) they send a list of the identifiers of the blocks they miss, and (3) they exchange an equivalent number of missing blocks. Li et al. show that this strategy is a Nash equilibrium strategy for pairwise exchanges of blocks. However, their analysis is limited to one-shot pairwise interactions, so the authors do not prove a folk theorem in infinitely repeated gossip dissemination. Their experiments show that agents also do not gain by deviating in repeated interactions. This shows that the approach of Li et al. avoids the problem identified by Vilaça & Rodrigues (2013). However, this also comes at a cost: if two agents do not have enough blocks to exchange, then some agents may fall behind in terms of the number of blocks they receive. In the worst case, some agents never have enough blocks to exchange with other agents, and they end up never receiving a large number of blocks. *BAR Gossip* addresses this problem by requiring agents to always exchange a minimum number of blocks in every interaction; if an agent does not have enough new blocks to share with its neighbour, then the agent must send garbage blocks to fill the minimum required number of blocks. The overhead of this approach makes it impossible to prove an approximate folk theorem. This is because the communication cost imposed by such overhead decreases the average utility of every agent by a constant factor, so the average utility cannot be arbitrarily close to any feasible and individually rational utility. *FlightPath* does not require agents to exchange garbage. Instead, exchanges can be slightly asymmetric. However, the strategy profile defined by *FlightPath* for pairwise exchanges is only an approximate Nash equilibrium, whereas we devise strategy profiles that are exact sequential equilibria.

3.3 Rational Behaviour in Pairwise Exchanges

In the literature, there is work that adopted a game theoretical approach to pairwise exchanges in distributed systems and work that used game theory to model dynamic networks.

3.3.1 Game Theoretical Approaches to Distributed Pairwise Exchanges

Some authors used game theory to analyse pairwise exchanges in file sharing and gossip dissemination protocols. In file sharing, the focus has been on the analysis of the incentives used by BitTorrent (Feldman et al. 2004; Jun & Ahamad 2005; Rahman et al. 2011). Specifically, the authors identify vulnerabilities with these incentives and propose more robust incentives. In gossip dissemination, Li et al. (2006, 2008) propose a Nash equilibrium protocol that enforces balanced exchanges between agents in a gossip dissemination protocol. Unlike these works, we do not limit our analysis to one-shot pairwise exchanges.

3.3.2 Game Theoretical Approaches to Dynamic Networks

Work in network formation games and dynamic games has proposed game theoretical models akin to our model of dynamic networks.

In network formation games, the network topology is the outcome of the agents' actions (Fabrikant et al. 2003; Moscibroda et al. 2006a). A model of network formation games is appropriate for modelling overlays where the topology is built and maintained by the agents (e.g., BitTorrent (Cohen 2003) and HyParView (Leitão et al. 2007)). This model is inappropriate for modelling the type of dynamic networks that we consider in this thesis. Specifically, we consider that the changes in the topology are not controlled by the agents but instead are caused by exogenous factors such as uncontrolled mobility. For instance, this is true in wireless-ad hoc networks (Srinivasan et al. 2003), and distributed overlays such as (Li et al. 2006; Li et al. 2008), among other dynamic networks.

In dynamic games, the structure of the game being repeated varies in each repetition according to a known probability distribution (Mailath & Samuelson 2007). This model captures stochastic variations in the network topology over time. We believe that this model is unrealistic in our setting, since in a distributed system agents may not know an exact probability

distribution on topologies; the only information agents may have is that the topologies satisfy minimum properties. For instance, in BAR Gossip (Li et al. 2006) agents know that the overlay is always connected, but the way in which the overlay is built does not guarantee that agents know the probability distribution on topologies at every point in time¹. We argue that a more appropriate model is to adopt an ex post approach: we restrict the set of possible networks to include only networks that satisfy some minimum properties, and require the protocol to be an equilibrium for all networks that satisfy those properties.

3.4 Rational Behaviour in Consensus

Recently, there has been some interest in solving problems related to consensus in a model where agents are rational (Bei et al. 2012; Abraham et al. 2013; Afek et al. 2014).

Abraham et al. (2013) address rational behaviour in the problem of leader election in asynchronous systems, under the assumption that agents do not fail. The authors propose a leader election protocol that is fair in the sense that each agent has equal chance of being elected the leader, and they show that the protocol is a sequential equilibrium even if agents can collude. This protocol can be used to solve fair consensus in the absence of failures: agents elect a leader and let its input dictate the final decision. Unfortunately, the protocol is not an equilibrium if agents may crash, because it is not resilient to deviations masked by crash-fault behaviour (e.g., if an agent pretends to crash, then its behaviour is indistinguishable from that of a faulty agent). Our work shows that these are the hardest type of deviations to deal with in a setting where agents are rational and may crash.

Bei et al. (2012) and Afek et al. (2014) propose consensus protocols resilient to rational behaviour and crashes. However, these protocols work only under strong assumptions about agents' utilities. Specifically, Afek et al. assume that every agent has a strict preference for outcomes where it learns the input of other agents, while Bei et al. require that their protocol be robust to deviations (that is, it achieves agreement even if rational agents deviate), a requirement that we view as unreasonably strong, since it implies that agents never pretend to crash. Neither of these protocols satisfy the fairness requirement. Moreover, the protocol proposed by Afek

¹The overlay is built with pseudo-random number generators seeded by information signed with a private key. Although the network topology of such overlay is likely to be approximately random, agents do not know an exact probability distribution on network topologies.

et al. is not even an equilibrium if some agent knows the input of other agents. Finally, these protocols are not sequential equilibria. In contrast, our protocol solves fair consensus with crashes assuming only that agents care about consensus, and is a sequential equilibrium even if agents know the inputs of other agents.

3.5 Game Theoretical Approaches to Other Problems

We now mention other relevant work that applies game theory to distributed problems. Halpern & Teague (2004) were perhaps the first to apply game theory to distributed systems. In this work, the authors address rational behaviour in the problem of secret-sharing and multiparty computation considering that agents may collude. This work was later extended by Abraham et al. (Abraham et al. 2006) to deal with both malicious and rational behaviour. Aiyer et al (2005) proposed the Byzantine-Altruistic-Rational (BAR) model, and they used game theory to implement and analyse a protocol for state machine replication in the BAR model. Moscibroda et al. (2006b) analysed rational and malicious behaviour in a virus inoculation game. Dolev et al. (2010) proposed the abstraction of a game authority and a corresponding self-stabilizing implementation as an approach to ensure that rational agents follow equilibria strategy profiles in systems where some agents may be malicious, even if those strategy profiles were devised for models where all agents are rational. Wong et al. (2011) showed how the presence of altruistic and Byzantine agents can help solving the social dilemma of the finitely repeated Prisoners' Dilemma game. Finally, Wong et al (2013) proposed a new approach to devise equilibria strategy profiles resilient to collusion.

Summary

We have discussed the most relevant work that proves folk theorems and addresses rational behaviour in the problems of gossip dissemination, pairwise exchanges in dynamic networks, consensus, and other related problems. In the next chapter, we describe our model.

4 Model

This section describes the model considered in the thesis. First, we introduce the general aspects of this model that are common to all considered problems. Then, we describe the aspects specific to the problems of gossip dissemination, pairwise exchanges in dynamic networks, and fair consensus with crashes.

4.1 General Aspects

We consider a synchronous message-passing system with n agents. Time is divided into synchronous stages. Stages are further divided into a sequence of τ synchronous rounds. Each round is divided into a *send phase*, where agents send messages to other agents, a *receive phase*, where agents receive messages sent by other agents in the send phase of that round, and an *update phase*, where agents perform a final update of the value of local variables based on what they have sent and received. We denote by \mathcal{N} the set of agents and assume that they have commonly-known identifiers in $\{0, \dots, n - 1\}$.

In each stage, some stage game Γ is played by the agents. The game Γ represents the problem being addressed. We now define the general aspects of Γ related to communication, the actions and information available to agents, strategies, and protocols. Other relevant aspects like utility and notions of equilibrium are specific to each problem, and we discuss them in the sections relative to each problem.

4.1.1 Communication

In each stage t , an undirected graph G^t restricts communication between agents, such that agent i can send messages in stage t to agent j iff j is a neighbour of i in G^t . We assume reliable and authenticated communication channels between agents. In some of our results, we consider that the communication graph may vary between stages. Formally, an evolving graph G is a

sequence G^1, G^2, \dots of graphs with one graph per stage, where G^t is the communication graph of stage t . Let \mathcal{G} be the set of all evolving graphs. We consider that an *adversary* selects an evolving graph G drawn from a subset $\mathcal{G}^* \subseteq \mathcal{G}^1$. The subset \mathcal{G}^* represents restrictions on the behaviour of the adversary that are common knowledge. For instance, if the network is formed by an overlay designed for data dissemination, then \mathcal{G}^* contains only evolving graphs where the communication graphs are connected.

We assume that the adversary is *oblivious* to the actions of agents, that is, the adversary selects some evolving graph $G \in \mathcal{G}^*$ at the start of the game, without being informed of the strategies followed by agents. An oblivious adversary is an appropriate abstraction for modelling networks where the changes in the network topology are caused by factors exogenous to the game theoretical model (Kuhn et al. 2010), e.g., physical topological changes in wireless ad-hoc networks (Srinivasan et al. 2003), deterministic changes in overlays resilient to rational behaviour such as (Li et al. 2006; Li et al. 2008), among others.

4.1.2 Actions

In every stage t , agent i has an input value v_i^t . In each round m of stage t , i sends messages to other agents and outputs values to the application. Specifically, fix an evolving graph $G \in \mathcal{G}^*$ selected by the adversary. A round- m action a_i of agent i is a pair (t_i^m, d_i^m) , where t_i^m is a function that maps every agent j to the message $t_i^m(j)$ that i sends to j (if i omits a message or j is not a neighbour of i in G^t , then $t_i^m(j)$ takes the null value \perp), and d_i^m is the value that i outputs to the application (again, if i does not output a value, then d_i^m takes the null value \perp). We assume that all messages sent are received in the round in which they are sent.

4.1.3 Information

We consider that agents have perfect recall, have imperfect information, and may or may not have incomplete information.

Agents have imperfect information regarding the actions of other agents. Specifically, if agent i follows round- m action $a_i^m = (t_i^m, d_i^m)$, then an agent j observes the message $t_i^m(j)$ that

¹This model is based on the model proposed by Kuhn & Ohsman (2010). The only difference is that they assume that the graph may change in every round

i sends to j , and this is the only information about i 's action that j obtains until the end of round m . Therefore, if j is not a neighbour of i in G^t , then j acquires no information about the round- m action of i .

Agents have incomplete information regarding the inputs. In the problem of consensus with crashes, agents also have incomplete information about crashes. Since the inputs determine the utilities of agents and crashes restrict the behaviour of agents, consensus corresponds to a Bayesian game where the types of agents are defined by their inputs and the way they crash (i.e., the messages they send before crashing).

Agents may also have incomplete information about the evolving graph $G \in \mathcal{G}^*$ selected by the adversary. Formally, at the beginning of every stage t , agent i acquires information regarding the communication graph G^t (we assume that agents always know their neighbours). We represent this information as a set $\mathcal{G}_i^t(G^t)$ of graphs, such that $G^t \in \mathcal{G}_i^t(G^t)$ and, in every graph \bar{G} in $\mathcal{G}_i^t(G^t)$, agent i obtains the same information about the communication graph in \bar{G}^t and G^t . For instance, if the only information agents have about G^t is the identity of their neighbours, then $\mathcal{G}_i^t(G^t)$ is the set of graphs where i has the same set of neighbours as in G^t .

4.1.4 Histories and Runs

Fix $G \in \mathcal{G}^*$ selected by the adversary. We take a *round- m history* h_i from stage t for agent i to be a sequence of tuples of the form $(\mathcal{G}_i^{t'}(G^{t'}), v_i^{t'}, s_1^{t'}, \dots, s_{m'}^{t'})$ for each stage $t' \leq t$, where $\mathcal{G}_i^{t'}(G^{t'})$ represents information about $G^{t'}$ as described before, $m^{t'} = \tau$ if $t' < t$, $m^t = m - 1$, $v_i^{t'}$ is agent i 's initial input in stage t' , and $s_{m'}^{t'}$ is a tuple of the form $(t_i^{m'}, d_i^{m'}, r_i^{m'})$, where $t_i^{m'}$ specifies the messages that i sends in round m' , $d_i^{m'}$ is i 's output in round m' , and $r_i^{m'}$ is a function specifying the round- m message $r_i^{m'}(j)$ sent by agent j to i (if j is not a neighbour of i , then $r_i^{m'}(j) = \perp$). Let $\mathcal{H}_i(G)$ be the set of histories for i compatible with G (i.e., the information that h_i provides to i regarding the communication graph of stage t is given by $\mathcal{G}_i(G^t)$ for all stages t). A *global (round- m) history* from stage t has the form (G, h_1, \dots, h_n) where $h_i \in \mathcal{H}_i(G)$ is a round- m history from stage t for agent i . Let $\mathcal{H}(G)$ be the set of histories with evolving graph G . A *run* r is a function from a stage number t and a round number m to global histories such that (a) $r(t, m)$ is a global round- m history from stage t and (b) if $t = t'$ and $m < m'$ or $t < t'$, then for each agent i , i 's history in $r(t, m)$ is a prefix of i 's history in $r(t', m')$.

As discussed in Section 2, a game Γ is defined by a game tree, where an outcome of Γ is a leaf node of the game tree and an outcome o of the repeated version of Γ is a function mapping each stage t to an outcome $o(t)$ of stage t . In our distributed setting, the nodes in the game tree correspond to global histories and an outcome of the repeated game corresponds to a run. Since we consider games of imperfect information, agents take actions at information sets, where now a round- m information set I_i from stage t for agent i is a set of round- m (global) histories from stage t that provide the same information to i about the actions of other agents and the communication graphs. We denote by $\mathcal{A}_i(I_i)$ the set of actions available to i at I_i . An agent i 's information set I_i at a global history is determined by i 's history in that global history. Thus, we identify a round- m information set I_i from stage t for agent i with a history h_i for agent i . If I_i is the information set associated with history h_i , we denote by $\mathcal{R}(I_i)$ the set of runs r where i has history h_i in $r(t, m)$.

4.1.5 Strategies and Protocols

In game theory, a *strategy* for agent i is a function that associates with each information set I_i for agent i a distribution over the actions that i can take at I_i . In distributed computing, a protocol for agent i is a function that associates with each history h_i for agent i a distribution over the actions that i can take at h_i . Since we are identifying histories for agent i with information sets, it is clear that a protocol for agent i can be identified with a strategy for agent i . Henceforth, we use the designations protocol and strategy interchangeably.

4.2 Gossip Dissemination

We consider the problem of infinitely repeated gossip dissemination.

4.2.1 Problem of Infinitely Repeated Gossip Dissemination

In every stage t , a *source* inputs a value v^t to be disseminated across all agents. The source is not part of the set of agents and can be trusted to always follows the protocol. In Section 5.3, we discuss how to drop this assumption. We assume that agents do not crash and that the

communication graph is always complete. We also assume that v^t is random, such that agents cannot guess v^t beforehand,

We focus on protocols that disseminate v in an epidemic fashion using an eager-push approach (Birman et al. 1999). This approach achieves a good tradeoff between communication overhead, reliability of data delivery, and simplicity of the analysis². Specifically, a stage t is divided into τ^D rounds of dissemination. The source splits the disseminated value v^t into a number ν of blocks of fixed size named *events*. Let e_y^t denote the y^{th} event from stage t . For each y between 1 and ν , the source sends the tuple $(y, e_y^t, [e_y^t])$ to all agents from a random set of f^{fan} agents, where f^{fan} is a parameter known as *fanout* and $[e_y^t]$ is a signature of e_y^t . This procedure is repeated by every agent that receives e_y^t , i.e., upon receiving $(y, e_y^t, [e_y^t])$ for the first time (either from the source or from some other agent), each agent i outputs the tuple to the application and forwards it to f^{fan} agents chosen at random. We consider that $\tau^D \geq \nu + n$, such that agents never receive an event for the first time in round τ , and hence always have time to forward every event after its first reception. We also assume that it is computationally hard for agents to replicate the signature of the source.

4.2.2 Utility

The utility U_i of agent i in a stage t is the difference between the benefits of i receiving events and the costs of sending messages during stage t . Specifically, agent i obtains a benefit β_i for receiving the y^{th} event in stage t if and only if agent i outputs the tuple $(y, e_y^{zt}, [e_y^{zt}])$ to the application, and i incurs a communication cost α_i per bit sent in a message. Therefore, given a global history h that corresponds to an outcome of stage t , the total utility $U_i(h)$ of i is then given by

$$U_i(h) = \sum_{1 \leq y \leq \nu} q_i(h, y) \beta_i - \sum_{\mathbf{m} \in \mathcal{M}_i(h)} \alpha_i |\mathbf{m}|,$$

where $q_i(h, y)$ is 1 if i outputs $(y, e_y^{zt}, [e_y^{zt}])$ or 0 otherwise, and $\mathcal{M}_i(h)$ is the set of messages that i sends in stage t . For the sake of simplicity, we assume that α_i and β_i are normalized so that the cost of sending a tuple is 1, that the costs of receiving messages are negligible, and that ν and β_i are constant for all stages. Our results can be easily generalized to drop these assumptions.

²Other more efficient approaches that use epidemic dissemination could be considered (e.g., lazy-push approach (Guerraoui et al. 2010)). However, the analysis of such approaches would be more complex, while still providing the same insight regarding how to sustain cooperation in push-based gossip dissemination.

The definition of expected utility of the repeated game is similar to the definitions included in Chapter 2 for games of imperfect and complete information, except we define the utility in terms of runs instead of outcomes of the repeated game. We assume that a discount factor $\delta \in (0, 1)$ discounts future utilities to the present. Given a run r , stage t and agent i , let $U_i(r | t)$ denote the utility of i as computed in stage t when the run is r , which is given by the sum $\sum_{t' \geq 1} \delta^{t'-t} U_i(r(t'))$. Given an information set I_i from stage t , a strategy profile $\vec{\sigma}$, and a belief system μ consistent with $\vec{\sigma}$, μ and $\vec{\sigma}$ define a probability distribution $\mu_{\vec{\sigma}, I_i}$ on runs in $\mathcal{R}(I_i)$. The expected utility $u_i(\vec{\sigma} | I_i)$ of i conditioned on the run being in $\mathcal{R}(I_i)$ is the expected value of $U_i(r | t)$, where the expectation is taken relative to $\mu_{\vec{\sigma}, I_i}$. We denote by $u_i(\vec{\sigma})$ the expected utility of i conditioning on the initial information set.

4.2.3 Approximate Folk Theorem

We aim to prove a slightly weaker version of an approximate Folk Theorem for the notion of sequential equilibrium. We focus on protocols that disseminate data using an eager-push approach. In these protocols, the set of feasible and individually rational vectors of utilities is determined by the value of the fanout f^{fan} . Specifically, if agents disseminate events in an eager-push fashion with fanout f^{fan} and every message only include tuples containing events, then the expected utility of agent i in a single stage is $x_i(f^{fan}) = \nu q(f^{fan})(\beta_i - f^{fan})$, where $q(f^{fan})$ is the probability of i receiving an event. Intuitively, agent i receives each event with probability $q(f^{fan})$; if i receives the event, then i obtains a benefit β_i and then forwards the event to f^{fan} agents, thus incurring a cost f^{fan} for sending those messages. Say that a fanout f^{fan} is individually rational if $x_i(f^{fan}) \geq 0$, which is true iff $\beta_i > f^{fan}$. Since the minimax utility of any agent in gossip dissemination is 0, the set feasible and individually rational utilities is the set of vectors $(x_1(f^{fan}), x_2(f^{fan}), \dots, x_n(f^{fan}))$ for all individually rational fanouts f^{fan} . Hence, an approximate folk theorem for eager-push gossip dissemination states that for all individually rational fanouts, there exists a strategy profile $\vec{\sigma}$ such that, if agents are sufficiently patient (i.e., δ is sufficiently close to 1), then $\vec{\sigma}$ is a sequential equilibrium and the average utility of every agent i is close to $x_i(f^{fan})$. In this thesis, we prove a slightly weaker result: we show that the above holds if β_i is sufficiently larger than f^{fan} .

Formally, given a constant c , say that a fanout f^{fan} is c -individually rational if $\beta_i > c f^{fan}$ for all agents i . Theorem 1 formalizes the main result proved in this part of the thesis.

Theorem 1. *There is a constant $c > 0$ such that, for every constant $\epsilon > 0$ and c -individually rational fanout f^{fan} , there exists a protocol $\vec{\sigma}$ and $\delta^* \in (0, 1)$, such that, for all $\delta \in (\delta^*, 1)$, $\vec{\sigma}$ is a sequential equilibrium and $|\bar{u}_i(\vec{\sigma}) - x_i(f^{fan})| \leq \epsilon$ for every agent i .*

Note that, if $\beta_i > nc$ for every agent i , then our result implies that an approximate Folk Theorem holds, since in this case every fanout f^{fan} is c -individually rational³.

4.3 Pairwise Exchanges in Dynamic Networks

We consider the problem of infinitely repeated pairwise exchanges over links of a dynamic network.

4.3.1 Problem of Infinitely Repeated Pairwise Exchanges In Dynamic Networks

In every stage t , every two agents i and j that are neighbours in the communication graph G^t exchange their inputs v_i and v_j , respectively, and output the received values to the application. Unlike gossip dissemination, we consider that the set \mathcal{G}^* (from which the adversary selects the evolving graph) may contain arbitrary evolving graphs. However, we show that \mathcal{G}^* must necessarily be restricted in order to sustain cooperation in pairwise exchanges.

4.3.2 Utility

We associate to every agent a utility U_i that maps each outcome of a given stage to the difference between the benefits of i receiving values from its neighbours and the costs of i sending and receiving messages. This utility is identical to the utility of gossip dissemination, except agents incur costs for receiving messages; these costs are now relevant to our results. Specifically, we assume that i obtains a benefit β_i for receiving a value from some neighbour, incurs a cost α_i per bit sent in a message, and incurs a cost γ_i per bit received in a message. The utility of i

³ In the proof of the theorem, we show that the result holds for all $c \geq e$, where here e is the base of the natural logarithm.

for an outcome of stage t that corresponds to global history h is thus

$$U_i(h) = \beta_i |R_i(h)| - \sum_{\mathbf{m} \in \mathcal{M}_i^{\text{sent}}(h)} \alpha_i |\mathbf{m}| - \sum_{\mathbf{m} \in \mathcal{M}_i^{\text{rec}}(h)} \gamma_i |\mathbf{m}|,$$

where $R_i(h)$ is the set of values received by i in h , and $\mathcal{M}_i^{\text{sent}}(h)$ and $\mathcal{M}_i^{\text{rec}}(h)$ are the set of messages sent and received by i in h , respectively.

We now define the expected utility for the repeated game. Agents have incomplete information of the game tree, hence we need to model agents' expectations regarding the evolving graph selected by the adversary. We adopt an ex post approach, i.e., we fix the evolving graph $G \in \mathcal{G}^*$ and assume that agents know that the adversary selected G when computing their expected utility. Specifically, given $G \in \mathcal{G}^*$ and information set $I_i \in \mathcal{I}_i(G)$, let $\mathcal{R}(G, I_i)$ be the set of runs in $\mathcal{R}(I_i)$ where the evolving graph is G . In this context, a belief system μ defines for each $G \in \mathcal{G}^*$ and $I_i \in \mathcal{I}_i(G)$, a probability distribution μ_{G, I_i} over global histories in I_i . As in gossip dissemination, a belief system μ and a strategy profile $\vec{\sigma}$ define a probability distribution $\mu_{\vec{\sigma}, G, I_i}$ on runs in $\mathcal{R}(G, I_i)$; the expected utility $u_i(\vec{\sigma} \mid G, I_i)$ of agent i conditioned on G and information set I_i is the expected value of $U_i(r)$, where now the expectation is taken relative to $\mu_{\vec{\sigma}, G, I_i}$. The expected utility $u_i(\vec{\sigma} \mid G)$ of i conditioned on evolving graph G is the expected utility of i conditioned on G and the initial information set.

4.3.3 Notion of Equilibrium

We define a new notion of equilibrium for our model that we call \mathcal{G}^* -Oblivious Adversary Perfect Equilibrium (\mathcal{G}^* -OAPE). This notion refines the notions of consistent belief and sequential equilibrium by taking an ex post approach to modelling incomplete information about the evolving graph.

Formally, a belief system μ is said to be consistent with the strategy profile $\vec{\sigma}^*$ and set \mathcal{G}^* iff there is a sequence $\vec{\sigma}^1, \vec{\sigma}^2, \dots$ of completely mixed protocols converging to $\vec{\sigma}^*$ such that for every $G \in \mathcal{G}^*$, agent i , information set $I_i \in \mathcal{I}_i(G)$, and global history $h \in I_i$,

$$\mu_{G, I_i}(h) = \lim_{M \rightarrow \infty} \frac{P_{G, \vec{\sigma}^M}(h)}{\sum_{h' \in I_i} P_{G, \vec{\sigma}^M}(h')},$$

where $P_{G, \vec{\sigma}^M}(h'')$ is the probability of history h'' being realized given that agents follow strategy

profile $\vec{\sigma}''$ and the evolving graph is G .

A strategy profile $\vec{\sigma}$ is a \mathcal{G}^* -OAPE iff there exists a belief system μ consistent with $\vec{\sigma}$ and \mathcal{G}^* such that, for all evolving graphs $G \in \mathcal{G}^*$, agents i , information sets $I_i \in \mathcal{I}_i(G)$, and strategies σ'_i , $u_i(\vec{\sigma} \mid G, I_i) \geq u_i((\sigma'_i, \vec{\sigma}_{-i}) \mid G, I_i)$. Intuitively, no agent gains by deviating at any information set, given that other agents do not deviate, even if agents know the evolving graph generated by the adversary. If \mathcal{G}^* is a singleton set, then a \mathcal{G}^* -OAPE strategy profile is a sequential equilibrium.

4.4 Fair Consensus with Crashes

We analyse a single stage of the fair consensus problem. Throughout this section and in the chapter that presents our main results relative to consensus, we simplify the notation wherever we can by omitting the stage number.

4.4.1 Fair Consensus Problem

In the fair consensus problem, every agent i proposes its input v_i to other agents. Agents must then agree on some proposed value v , which they output to the application. We call v the *decided value*. Agents must agree on v despite some agents *crashing*. In addition, the agreement must be *fair*, in the sense that the probability of agreeing on each value v is proportional to the number of agents with input v .

Formally, we assume a complete communication graph. Agents are either correct or faulty in a run. An agent fails only by crashing. If it crashes in round m of run r , then it may send a message to some subset of agents in round m , but from then on, it sends no further messages. Thus, we take a *failure* \mathbf{f} of agent i to be a tuple (i, m, A) , where m is a round number (intuitively, the round at which i crashes) and A is a set of agents (intuitively, the set of agents j to whom i can send a message before it fails). We assume that if $m > 1$, then A is non-empty, so that i sends a message to at least one agent in round m if i fails in round m . (Intuitively, if $m > 1$, we are identifying the failure pattern where i crashes in round m and sends no message with the failure pattern where i crashes in round $m - 1$ and sends messages to all the agents.) A *failure pattern* F is a set of failures of distinct agents i . A run r has *context* (F, \vec{v}) if (a) \vec{v} describes the

initial inputs of the agents in r , (b) if $(i, m, A) \in F$, then i sends all messages according to its protocol in each round $m' < m$, sends no messages in each round $m' > m$, and sends messages according to its protocol only to the agents in A in round m , and (c) all messages sent in r are received in the round that they are sent. Let $\mathcal{R}(F, \vec{v})$ consist of all runs r that have context (F, \vec{v}) . Let $\mathcal{R}(F)$ consist of all runs that have F as the set of failures.

A protocol achieves consensus if it satisfies the following properties (Fischer et al. 1985):

- **Agreement:** No two correct agents decide different values.
- **Termination:** Every correct agent eventually decides.
- **Integrity:** All agents decide at most once.
- **Validity:** If an agent decides v , then v was the initial input of some agent.

We are interested in one other property: fairness. Note that, once we fix a context, a protocol for the agents generates a probability on *runs*, and hence on outcomes, in the obvious way. Fairness just says that each agent has probability at least $1/n$ of having its value be the consensus decision, no matter what the context. More precisely, we have the following condition:

- **Fairness:** For each context (F, \vec{v}) , if c of the nonfaulty agents in F have initial preference v , then the probability of v being the consensus decision conditional on $\mathcal{R}(F, \vec{v})$ is at least c/n .

Intuitively, in the absence of failures, fairness is the requirement that every input is decided with equal probability, i.e., with probability $1/n$. If some agents fail, then fairness requires that failures do not negatively impact the probability of deciding on the input v_i of any nonfaulty agent i , so that v_i is decided with probability at least as high as the probability of deciding on v_i when there are no failures (i.e., at least $1/n$).

For ease of exposition, we take the set of possible inputs to be $\{0, 1\}$. (Our results can easily be extended to deal with larger sets of possible values.) We also assume that there is a special value Ψ that an agent can decide on. By deciding on Ψ , an agent guarantees that there is no consensus (by violating Validity). If we assume that all agents prefer to reach consensus on some value to not reaching consensus at all, in the language of Ben Porath (2003), this means that each agent has a *punishment strategy*.

4.4.2 Utility

We focus on the case where agents care only about consensus, since this type of utility function seems to capture many situations of interest. In this type of utility, the utility U_i of every agent i captures the preferences of i for the decided value (and does not depend for instance on the messages that i sends), and the input v_i is i 's most preferred value. Specifically, we consider that i 's utility is either (1) β_{0i} if consensus is reached on i 's initial input, (2) β_{1i} if there is consensus but not on i 's initial input, or (3) β_{2i} if there is no consensus. The assumption that agents care only about consensus means that, for all i , $\beta_{0i} > \beta_{1i} > \beta_{2i}$.

4.4.3 Notions of Equilibrium

We are interested in equilibria protocols that solve fair consensus. Since existing notions of equilibrium do not take into account crashes, we have to define new notions of equilibrium appropriate for our setting.

Specifically, we can see the consensus problem as a Bayesian game, where the type of an agent is defined by its input and failure, so the types of all agents are defined by the context. Agents have incomplete information about the context. To model expectations regarding this information, we adopt both ex post and ex ante approaches. First, we define a new notion named f -Nash equilibrium that refines the notion of ex post Nash equilibrium, where f is the upper bound on the number of failures. Then, we define the notions of π -Nash equilibrium and π -sequential equilibrium that refine the notions of ex ante Nash equilibrium and sequential equilibrium, respectively, where π is a probability distribution on contexts.

4.4.3.1 f -Nash equilibrium

A strategy profile $\vec{\sigma}$ is an f -Nash equilibrium if, for each fixed context (F, \vec{v}) where there are at most f faulty agents in F , and all agents i , there is no strategy σ'_i for agent i such that i can increase its expected utility by following σ'_i . Formally, if $u_i(\vec{\sigma}' | F, \vec{v})$ denotes i 's expected utility if strategy profile $\vec{\sigma}'$ is played, conditional on the run being in $\mathcal{R}(F, \vec{v})$, we require that for all strategies σ'_i for i , $u_i((\sigma'_i, \vec{\sigma}_{-i}) | F, \vec{v}) \leq u_i(\vec{\sigma} | F, \vec{v})$. The notion of f -Nash equilibrium

extends the notion of ex post Nash equilibrium by allowing up to f faulty agents⁴; a 0-Nash equilibrium is an ex post Nash equilibrium. Note that the definition of u_i admits the possibility of agent i crashing (this assumption has no impact on our results).

4.4.3.2 π -Nash Equilibrium

Given a distribution π on contexts and a strategy profile $\vec{\sigma}$, π and $\vec{\sigma}$ determine a probability on runs denoted $\pi_{\vec{\sigma}}$ in the obvious way. We say that $\vec{\sigma}$ is a π -Nash equilibrium if, for all agents i and all strategies σ'_i for i , we have $u_i(\sigma'_i, \vec{\sigma}_{-i}) \leq u_i(\vec{\sigma})$, where now the expectation is taken with respect to the probability $\pi_{\vec{\sigma}}$. If π puts probability 1 on there being no failures, then we get the standard notion of ex ante Nash equilibrium.

4.4.3.3 π -Sequential Equilibrium

According to the original definition of sequential equilibrium (Kreps & Wilson 1982), a strategy profile $\vec{\sigma}$ is a sequential equilibrium if there is a belief system μ consistent with $\vec{\sigma}$ such that no agent has incentives to deviate from $\vec{\sigma}$. To incorporate crashes in this definition, we have to first redefine the notion of consistent belief system.

Say that a belief system μ is *consistent with $\vec{\sigma}$ and π* if there exists a sequence of *completely mixed* strategy profiles $\vec{\sigma}^1, \vec{\sigma}^2, \dots$ converging to $\vec{\sigma}$ such that

$$\mu_{I_i}(h) = \lim_{M \rightarrow \infty} \frac{\pi_{\vec{\sigma}^M}(h)}{\sum_{h' \in I_i} \pi_{\vec{\sigma}^M}(h)},$$

where $\pi_{\vec{\sigma}''}(h'')$ is the probability of global history h'' being realized, given that agents follow the strategy profile $\vec{\sigma}''$ and the probability distribution over contexts is π . Note that μ_{I_i} , π , and $\vec{\sigma}$ together define a probability distribution over runs in $\mathcal{R}(I_i)$. Let $\mu_{I_i, \pi, \vec{\sigma}}$ denote this probability distribution, and let $u_i(\vec{\sigma} | I_i)$ be the expected utility of i when agents follow $\vec{\sigma}$ conditioned on the run being in $\mathcal{R}(I_i)$, where the expectation is taken relative to $\mu_{I_i, \pi, \vec{\sigma}}$.

A strategy profile $\vec{\sigma}$ is a π -*sequential equilibrium* if there exists a belief system μ consistent with $\vec{\sigma}$ and π such that, for every agent i , information set I_i , and strategy σ'_i , $u_i((\sigma'_i, \vec{\sigma}_{-i}) | I_i) \geq$

⁴This definition is in the spirit of the notion of (k, t) -robustness as defined by Abraham et al. (2006), where coalitions of size k are allowed in addition to t “faulty” agents, but here we restrict the behaviour of the faulty agents to crash failures rather than allowing the faulty agents to follow an arbitrary protocol, and take $k = 1$.

$$u_i((\sigma'_i, \vec{\sigma}_{-i}) \mid I_i).$$

Summary

We described the general aspects of our model and the aspects specific to the problems of gossip dissemination, pairwise exchanges in dynamic networks, and fair consensus with crashes. In the next chapter, we present the results obtained in the problem of gossip dissemination.

5 Gossip Dissemination

In this section, we prove Theorem 1, i.e., we prove a variant of an approximate Folk Theorem for eager-push gossip dissemination protocols. We reinstate the theorem here.

Theorem 1. *There is a constant $c > 0$ such that, for every constant $\epsilon > 0$ and c -individually rational fanout f^{fan} , there exists a protocol $\vec{\sigma}$ and $\delta^* \in (0, 1)$, such that, for all $\delta \in (\delta^*, 1)$, $\vec{\sigma}$ is a sequential equilibrium and $|\bar{u}_i(\vec{\sigma}) - x_i(f^{fan})| \leq \epsilon$ for every agent i .*

To prove Theorem 1, we show that, for every fanout f^{fan} , there is a protocol $\vec{\sigma}^{f^{fan}}$ where agents use fanout f^{fan} to disseminate data, which satisfies the following two properties if agents are sufficiently patient (i.e., if δ is sufficiently close to 1) and the benefit/cost ratio is sufficiently high: (1) $\vec{\sigma}^{f^{fan}}$ is a sequential equilibrium and (2) the average utility of every agent i can be as close to $x_i(f^{fan})$ as we want.

We assume that the source can perform the role of a *trusted mediator*. In the protocol $\vec{\sigma}^{f^{fan}}$, agents disseminate events using fanout f^{fan} . Periodically, they also exchange monitoring information with the source regarding the events that they sent to and received from other agents. The source uses this information to detect deviations; if the source detects a deviation of agent i (e.g., if i did not forward an event to f^{fan} agents), it triggers a punishment of i . In a punishment of agent i , other agents do not forward events to i .

We show that $\vec{\sigma}^{f^{fan}}$ is a sequential equilibrium if agents are sufficiently patient and f^{fan} is c -individually rational for some constant c . In the proof, we apply the one-shot deviation property (Hendon et al. 1996): we show that for all agents i , stages t , and round- m information set I_i from t , i does not gain by performing a one-shot deviation at I_i , where a one-shot deviation of i consists in i taking an action a_i at I_i such that $\sigma_i^{f^{fan}}(a_i | I_i) = 0$ and then following the protocol at every other information set. We show that if agent i performs a one-shot deviation at I_i , then either (i) the utility of i does not increase or (ii) i gains by not sending some messages in stage t , but the source detects the deviation and triggers a punishment of i in a stage $t' > t$,

which causes the utility of i in stage t' to decrease. If agents are sufficiently patient, then the future utility loss in stage $t' > t$ outweighs the immediate gain in stage t , so i does not gain from performing a one-shot deviation at I_i . It follows by the one-shot deviation property that $\vec{\sigma}^{fan}$ is a sequential equilibrium.

We also show that the average utility $\bar{u}_i(\vec{\sigma}^{fan})$ of agent i when all agents follow the protocol $\vec{\sigma}^{fan}$ is the difference between $x_i(f^{fan})$ and the average cost per stage of sending monitoring information. We prove that the average cost of monitoring can be arbitrarily small if agents are sufficiently patient. It follows that $\bar{u}_i(\vec{\sigma}^{fan})$ can be arbitrarily close to $x_i(f^{fan})$.

We now discuss the protocol $\vec{\sigma}^{fan}$ and prove the main result. Table 5.1 summarizes the notation used in this chapter.

5.1 Dissemination Protocol

We start by describing a simpler naive protocol, which helps us identifying the main challenges addressed in the definition of $\vec{\sigma}^{fan}$, before describing the algorithm of $\vec{\sigma}^{fan}$ in detail.

Consider the following naive protocol. In every stage t , agents send messages in τ rounds, where the first τ^M rounds are used for exchanging monitoring information with the source and the last τ^D rounds are used for disseminating events using an eager-push approach. In monitoring rounds, every agent i sends information to the mediator regarding the behaviour of each agent $j \neq i$ in stage $t-1$ (if $t = 1$, then agents do not send monitoring information). Specifically, i first checks whether j sent only correctly formatted messages. If j sent an incorrect message, then i sends an *accusation* against j to the source. Otherwise, for each event e_y^{t-1} disseminated in stage $t-1$, i reports to the source whether j sent to or received from i the tuple $(y, e_y^{t-1}, [e_y^{t-1}])$. The source collects accusations and reports from all agents and then gives a final verdict on whether each agent j should be punished in stage t . Namely, the source triggers a punishment of j if the source receives an accusation against j , does not receive a monitoring message from j , or the source detects that j received event e_y^{t-1} but did not forward it to f^{fan} agents. Agents punish j by not forwarding events to j during stage t .

It is easy to see that, if an agent i sends fewer messages than required in stage t by forwarding an event to fewer than f^{fan} agents or by not sending monitoring messages to the source, then the deviation of i is detected and i is punished in stage $t+1$. In this punishment, i loses the

benefit $\delta\beta_i\nu$ of receiving the events disseminated in stage $t + 1$. If $\beta_i\nu$ is larger than the costs of sending messages in stage t and i is sufficiently patient, then the loss outweighs the gain of not sending messages in stage t , so i has no incentives to deviate by sending fewer messages.

Unfortunately, the naive protocol fails to meet the necessary requirements to prove Theorem 1. First, it is not a sequential equilibrium, because rational agents can increase their expected utility by lying about monitoring information and by not selecting the sets of agents to which they forward events at random. Second, the overhead incurred by agent i for reporting on the events sent and received by other agents is too high, such that the average utility of i is lower than $x_i(f^{an})$ by a constant factor.

More precisely, in the definition of $\vec{\sigma}^{fan}$, we address the following problems of the naive protocol:

1. **Problem:** *Agents gain by lying about monitoring information.* If agent i is punished in stage t , then agents do not forward events through i , so the probability of an agent receiving events in stage t decreases. Hence, agents increase their expected benefits of receiving events by lying about i 's behaviour in previous stages and avoiding the punishment of i .

Solution: We enforce the following two properties. First, each agent i is only allowed to send monitoring information relative to other agents; information sent by i has no effect on the probability of i being punished. Second, the probability of each event reaching an agent i that is not punished is *constant*, regardless of the punishments applied to other agents. To achieve this, we use *commutative symmetric ciphers*, which are symmetric ciphers that satisfy the property of commutativity of the cipher operation. Specifically, let $(e)_\kappa$ denote the cipher of event e with key κ . Commutativity of ciphers is the property that for all events e and keys κ and κ' , we have $((e)_\kappa)_{\kappa'} = ((e)_{\kappa'})_\kappa$ ¹. In symmetric commutative ciphers, the cipher operation is identical to the decipher operation, so $((e)_\kappa)_\kappa = e$.

Agents punish other agents that deviated in previous stages using commutative ciphers instead of omitting events. More precisely, the mediator generates a unique random key κ_i for each agent i that is revealed to all agents but i prior to dissemination. All agents forward events in an eager-push fashion regardless of punishments. However, when for-

¹This property is satisfied by any symmetric cipher algorithm that uses block ciphering algorithm with the CTR mode of operation (Bellare et al. 1997; ISO/IEC 2006). In these algorithms, the cipher operation is performed by xoring data with a stream of bits generated from κ .

warding an event e to an agent i that is being punished, agents cipher e with κ_i so that i cannot decipher the event (thus, they effectively deny i the benefit β_i of receiving e). Similarly, after receiving an event e from agent i that is being punished, agents decipher the event with κ_i . The commutativity property guarantees that agents obtain each event disseminated by the source if (and only if) they are not being punished. Therefore, unpunished agents receive each event with the same probability as they would if no agent were being punished (so they are not affected by the punishment of other agents)², whereas agents being punished are denied the benefits of receiving events.

2. **Problem:** *Agents gain by not selecting f^{fan} agents at random when forwarding an event.*

Consider a history h_i for agent i from stage t where i has just received a (possibly ciphered) event e in h_i from some agent j and is about to forward the event to f^{fan} agents. Note that, when i forwards e , there is the possibility that the dissemination of e loops back to i . This probability may depend on to which agents i forwards the event. So, there may be a set S_y^t of f^{fan} agents such that, if i forwards e to agents in S_y^t , then the probability of e looping back is minimal (e.g., i may have already received e from those agents in h_i and thus knows that they will not forward the event again). As it will become clearer later, agent i always has to forward event e upon its first reception to f^{fan} agents, even if e does not correspond to the event sent by the source; if i forwards e to the agents in S_y^t , then the probability of e looping back to i is minimal and thus the probability of other agents sending to i the event actually disseminated by the source is maximal. Therefore, i gains by deterministically sending e to the agents in S_y^t .

Solution: The set of agents S_y^t to which agent i forwards e_y^t is defined deterministically by a pseudo-random number generator (PRNG) that is seeded by a random seed sd_i . Given sd_i , the source can verify whether i forwards events according to the protocol. For reasons that will become clearer in the proof of Theorem 1, other agents must not be able to predict to which agents i will forward events throughout stage t . Therefore, we trust the source to generate sd_i at random and to reveal it only to i , prior to dissemination.

3. **Problem:** *The overhead of monitoring is too large.* The average utility of agent i is $x_i(f^{fan}) - c$, where c is the average cost of monitoring per stage. If agents report on every

² This is true because we assume that agents do not incur costs for computing the ciphers. We believe that this is a reasonable assumption, since symmetric ciphers are computationally cheap.

event sent to or received from other agents with probability 1, then c is constant and the average utility cannot be as close as we want to $x_i(f^{fan})$. Hence, agents can only report on a subset of events. However, if agents report on too few events and an agent i deviates by forwarding an event to fewer than f^{fan} events, then the probability of detecting the deviation is low, such that i gains by deviating.

Solution. In $\vec{\sigma}^{fan}$, time is divided into *epochs*, where an epoch is a sequence of n^{spe} stages. Agents only send monitoring information to the source in the monitoring rounds of the first stage of each epoch. Punishments are performed in an epoch basis: if agent i deviates in a stage from epoch z , then i is punished in all stages from epoch $z + 1$; if agent i deviates again during epoch $z + 1$, then i is punished in epoch $z + 2$, and so on. In stage 1 of every epoch $z > 1$, agents send accusations against other agents that deviated in a stage from epoch $z - 1$, and they report on a subset of events sent to or received from each agent i in epoch $z - 1$. Specifically, we split the sequence of events disseminated during epoch $z - 1$ into n^{seq} subsequences of equal size. We denote by $\mathcal{E}(z', s)$ the set of events from the s^{th} subsequence from epoch z' , and we denote by $sub(t, y)$ the subsequence that contains the y^{th} event of stage t . For every epoch $z > 1$ and subsequence $1 \leq s \leq n^{seq}$, the source requests with independent probability p^{seq} a report from all agents i saying, for all events $e \in \mathcal{E}(z, s)$, the stage and round numbers when i first received and sent e to other agents; i sends those reports relative to each subsequence if and only if the source requests them. The source then uses accusations and reports to determine whether each agent should be punished in epoch z . In Section 5.1.2, we discuss how to carefully select the parameters p^{seq} and n^{seq} to achieve a good tradeoff between the overhead of monitoring and the probability of detecting deviations. We identify values for those parameters such that (1) the probability of detecting deviations is high, such that agents do not gain by deviating, and (2) the average overhead of monitoring per stage decreases with the number n^{spe} of stages per epoch, such that we can arbitrarily minimize the average overhead by increasing n^{spe} .

Therefore, the protocol runs in an infinite number of epochs, where each epoch is divided into n^{spe} stages. Every stage is further divided into $\tau = 2n^{seq} + 2 + \tau^D$ rounds. In the first $2n^{seq} + 1$ rounds of stage 1 of every epoch $z > 1$, the source collects reports of events and accusations regarding epoch $z - 1$; agents send no messages in the first $2n^{seq} + 1$ rounds of every other stage.

In round $2n^{\text{seq}} + 2$ of stage 1 of every epoch z , the source sends the seeds, keys, and verdicts on whether each agent should be punished during epoch z . In the last τ^{D} rounds of every stage, agents disseminate events using the PRNG.

5.1.1 Algorithm

We now describe the algorithm of $\vec{\sigma}^{\text{fan}}$ in more detail. Let sd_i^z and κ_i^z be the seed and key of agent i in epoch z , respectively. Given an epoch number $z \geq 1$, a stage number $1 \leq t \leq n^{\text{spe}}$, and a number $1 \leq y \leq \nu$, let e_y^{zt} denote the y^{th} event disseminated in stage t from epoch z . For every agent i , there is a set generator SG_i that, given sd_i^z , $1 \leq t \leq n^{\text{spe}}$, and $1 \leq y \leq \nu$, returns a set $SG_i(sd_i^z, t, y)$ of f^{fan} agents to which i must forward the y^{th} event from stage t . We assume that the sequence of sets that SG_i generates is pseudo-random. We discuss this and other cryptographic assumptions in Section 5.2.1.

In every epoch z , agent i uses three variables that keep track of the behaviour of every agent j , namely, (i) a vector vs^z representing the validity of the messages sent by j , where $vs^z(j) = \text{True}$ iff j sent all messages requested by the source and sent only valid (i.e., correctly formatted) messages during epoch z , (ii) a vector of reports re^z , where, for all events e_y^{zt} disseminated in epoch z , $re^z(i, j, t, y)$ is either \perp if i did not receive e_y^{zt} from j or is the first round when i received e_y^{zt} from j , and (iii) a vector of reports se^z defined identically to re^z except it represents the rounds when i first sent each event to j or takes the value \perp if i did not forward the event to j . At the end of every epoch z , the source decides, with independent probability p^{seq} for each subsequence of events numbered $1 \leq s \leq n^{\text{seq}}$, whether to request reports on events from s . We use a variable res^z to represent this decision, where $res^z(s) = \text{True}$ if and only if the source decides to request reports from s . In every odd round m in $\{1, 3, \dots, 2n^{\text{seq}} - 1\}$, where $m = 2s + 1$ with $0 \leq s \leq n^{\text{seq}} - 1$, the source sends a message containing $res^z(s)$; in round $m + 1$, every agent i sends a reply iff $res^z(s) = \text{True}$; this reply contains all the entries in the vectors re^{z-1} and se^{z-1} regarding the rounds when i first sent an received events from $\mathcal{E}(z, s)$ during epoch $z - 1$, respectively. In round $2n^{\text{seq}} + 1$, agents also send the vectors vs^{z-1} , where $vs^{z-1}(j) = \text{False}$ represents an accusation against j saying that j did not send only correct messages. At this point, the source generates the keys κ_i^z and the seeds sd_i^z for all agents i . The source also generates a vector bp^z of verdicts such that $bp^z(i) = \text{True}$ iff agent i is to be punished in epoch z , where agent i is to be punished in z if (i) the source receives an accusation

against i or (ii) the source detects that i first received an event e_y^{zt} in round m' but did not forward e_y^{zt} in round $m' + 1$ to exactly the agents in $SG_i(sd_i^{z-1}, t, y)$. In round $2n^{\text{seq}} + 2$, the source sends to agent i the seed sd_i^z , the vector bp^z , and the keys κ_j^z for every agent $j \neq i$. In all rounds $m > 2n^{\text{seq}} + 2$ of every stage t from epoch z , the source and the agents disseminate events. More precisely, for each round $m = 2n^{\text{seq}} + 2 + y$ with $1 \leq y \leq \nu$, the source selects a set S of f^{fan} agents at random, and sends to every agent $j \in S$ the tuple $(y, e, [e_y^{zt}])$, where e is either e_y^{zt} if i is not being punished ($bp^z(i) = \text{False}$) or the ciphered event $(e_y^{zt})_{\kappa_i^z}$ if i is being punished ($bp^z(i) = \text{True}$). Upon receiving a tuple (y, e, s) for the first time in round m , agent i first decipheres e with κ_j^z if i receives the pair from an agent j that is being punished; let e^* be either $(e)_{\kappa_j^z}$ if $bp^z(j) = \text{True}$ or e otherwise; i then forwards (y, e', s) to every agent $l \in SG_i(sd_i^z, t, y)$, where again e' is either $(e^*)_{\kappa_l^z}$ if l is being punished or e^* otherwise.

If all agents follow the protocol and agent i receives $(y, e, [e_y^{zt}])$, then, by the commutativity of ciphers, $e = (e_y^{zt})_{\kappa_i^z}$ if i is being punished or $e = e_y^{zt}$ otherwise. Therefore, if some agent deviates and i is being punished, then there is no way for i to determine whether e corresponds to e_y^{zt} ciphered with κ_i^z . For this reason, we do not require agents to always forward valid events, i.e., events that always correspond to either e_y^{zt} or to e_y^{zt} ciphered with some key. Nevertheless, we show that agents do not gain from not forwarding events correctly.

Every agent i has incentives to send valid messages in epoch z due to the threat of punishment in epoch $z + 1$: if i sends an invalid message or omits a message containing reports or accusations to the mediator, then the mediator sets $vs^z(i) = \text{False}$ and i is punished in epoch $z + 1$; similarly, if i sends an invalid message to an agent j , then j sets $vs^z(i) = \text{False}$, and sends an accusation against i to the source in epoch $z + 1$, triggering a punishment; if i sends all requested monitoring messages, then i is indifferent regarding the content of each report and accusation, since these do not affect the probability of i being punished in epoch $z + 1$; finally, if i does not forward every event e_y^{zt} according to the pseudo-random set generator and the seed of i , then with high probability (i.e., p^{seq}) the source requests reports on the events from $sub(t, y)$ and detects the deviation of i , again triggering a punishment in epoch $z + 1$. Once i sends an invalid message in epoch z , i knows that it will be punished in epoch $z + 1$, so i has no incentives to continue sending messages. Conversely, once i does not forward an event e_y^{zt} according to the protocol, i knows that, if the source requests reports relative to the subsequence $sub(t, y)$, then i will be punished in epoch $z + 1$, so i has no incentives to keep forwarding events from

$\mathcal{E}(z, \text{sub}(t, y))$. In other words, there are information sets I_i for i such that the fact that i omits certain messages at I_i has no impact on the probability of i being punished in epoch $z + 1$. Therefore, the best response strategy for i is to not send those messages. To implement such strategy, agent i uses a set ME_i^z to keep track of the subsequences s that contain events that were not correctly forwarded by i , such that i only forwards an event e_y^{zt} if the subsequence $\text{sub}(t, y)$ is not in ME_i^z ; in addition, i uses the variable $vs^z(i)$ to keep track of whether i sent all requested messages and did not send invalid messages, such that i only sends messages if $vs^z(i) = \text{True}$.

The pseudo-code of the algorithm executed by agent i is depicted in Alg. 1. We denote by v_i the value of variable v in agent i . Agent i starts by initializing the variables used in the algorithm (lines 1-12). In addition to the variables described above, i uses a set Out_i^z to keep track of the events that i already output to the application, a set $EO_i^z(t, y)$ for each stage t and event number y that contains all the tuples with the form (y, e, s) that were sent to i , and a set $EF_i^z(j, t, m)$ for each agent j , stage t , and round m , which contains the events that i is expected to forward to j in round m . To simplify the exposition, we also define these variables for the source, so for instance, $vs_i^z(\text{source})$ is *True* iff the source only sent valid messages, which is always the case, since the source always follows the protocol.

In the send phase of round m , agent i sends messages only if $vs_i^z(i) = \text{True}$ (lines 14-26). In the first n^{seq} even rounds of stage 1 of each epoch z , i sends the reports of events requested by the source. Specifically, for every round $m = 2s$ with $1 \leq s \leq n^{\text{seq}}$, if $res_i^z(s) = \text{True}$, then i sends vectors Re_i^s and Se_i^s that contain the values in re_i^z and se_i^z relative to the events from $\mathcal{E}(z, s)$ (lines 17-20). In round $2n^{\text{seq}} + 1$ of stage 1 of every epoch $z > 1$, i sends the vector vs_i^z to the source, which contains accusations against other agents (line 22). Finally, in the dissemination rounds $m > 2n^{\text{seq}} + 2$ of every stage t , for every agent $j \neq i$, i sends to j the set $EF_i^z(j, t, m)$ of events that i received in round $m - 1$ (lines 24-25). To simplify the exposition, we consider that an omission of i in a dissemination round is equivalent to i sending an empty set of events.

In the receive phase of round m , agent i processes every round- m message that i received from other agents or from the source (lines 27-58). If m is one of the first n^{seq} odd rounds of stage 1, then i processes the pair $(s, res^z(s))$ that the source sends to i and updates $res_i^z(s)$ correspondingly (line 29). If either (i) m is round $2n^{\text{seq}} + 1$, $t = 1$, and i omitted the accusations to the source or (ii) m is one of the first even rounds from stage 1 from epoch z , the source

requested reports from i , and i did not send those reports, then i sets $vs_i^z(i) = False$ (line 31). If $m = 2n^{\text{seq}} + 2$ and $t = 1$, then i stores the seed sd_i^z sent by the source and, for every agent $j \neq i$, i saves the key κ_j^z and updates the verdict $bp_i^z(j)$ (line 33-36). Finally, if m is a dissemination round (i.e., $m > 2n^{\text{seq}} + 2$), then i processes all the messages containing events that i received and that i sent to other agents (lines 38-57):

- For every entity j different from i (where j may be the source or another agent), i updates the variables according to the message that j sends to i . If j sends an invalid message to i , then i sets $vs_i^z(j) = False$. Otherwise, if j sends a valid set S of events, then i registers those events in re_i^z (lines 39-49): for every $(y, e, s) \in S$, if j did not send the y^{th} event to i before and $0 \leq \text{age}(y, m) < n$, then i sets $re_i^z(i, j, t, y)$ to $\text{age}(y, m)$, otherwise, i sets $vs_j^z(j) = False$, where $\text{age}(y, m) = m - 2n^{\text{seq}} - 1 - y$ (line 43). Agent i registers $\text{age}(y, m)$ instead of m to decrease the average cost of monitoring per stage. After processing the events received from j , i prepares to forward some of those events. Given agent l and event e' , let $\varphi_l(e')$ be a function that returns $(e')_{\kappa_l^z}$ if $bp_i^z(l) = True$ or returns e' otherwise. For each tuple (y, e, s) received from j , i computes $e^* = \varphi_j(e)$ (lines 44), adds (y, e^*, s) to the set $EO_i^z(t, y)$ (line 45), and if i did not forward the y^{th} event before, i adds $(y, \varphi_l(e'), s)$ to $EF_i^z(l, m + 1)$ for all agents $l \in SG_i(sd_i^z, t, y)$ (lines 47-48).
- For every agent $j \neq i$, i updates the variables basing on the dissemination message that i sent to j (lines 50-57). If i sends an invalid message to j , then i sets $vs_i^z(i) = False$ (50), otherwise, if i sent a set S of events to j , then i updates the variables se_i^z and ME_i^z (lines 51-57). More precisely, for every tuple $(y, e, s) \in S$ that i sent to j , if $0 \leq \text{age}(y, m) < n$ and i did not send the event before to j , then i sets $se_i^z(i, j, t, y)$ to $\text{age}(y, m)$; otherwise, i sets $vs_i^z(i) = False$. Moreover, i adds to ME_i^z the subsequence number $sub(t, y)$ of every event e_y^{zt} not correctly forwarded by i to j , where event e_y^{zt} is not correctly forwarded if (i) $(y, e, s) \in S$ and $(y, e', s') \notin EF_i^z(j, t, m)$ for all e' and s' (i should not have forwarded the event to j) or (ii) $(y, e, s) \notin S$ for all e and s but there exist e' and s' such that $(y, e', s') \in EF_i^z(j, t, m)$ (so, i should have forwarded a tuple (y, e', s') to j but did not).

We assume that i processes these messages sent by other agents j following the order of the agents' identifiers. This assumption will play a role in the proof that the protocol is a sequential equilibrium.

In the update phase of every dissemination round m , i removes all tuples (y, e, s) such that $sub(t, y) \in ME_i^z$ from the set $EF_i^z(j, t, m + 1)$ for all agents j , so that i sends no further events from those subsequences (lines 61-62). In addition, i outputs all tuples (y, e, s) such that i did not output (y, e, s) before and i believes that (y, e, s) was sent by the source (lines 63-67). More precisely, we consider that there is a function $comp$ such that $comp(e, s)$ returns the event e_y^{zt} if $s = [e_y^{zt}]$ and i can perform a finite computation on e to retrieve e_y^{zt} , or $comp(e, s)$ returns \perp otherwise. Intuitively, e may correspond to the event e_y^{zt} ciphered with keys known to i , so i may try deciphering e with multiple combinations of keys to retrieve e_y^{zt} . (Obviously, if $e = e_y^{zt}$ and $s = [e]$, then $comp(e, s)$ returns e .) For all pairs (t, y) such that $(t, y) \notin Out_i^z$ and i received a tuple (y, e, s) such that $comp(e, s) \neq \perp$, i outputs $(y, comp(e, s), s)$.

The pseudo-code of the algorithm run by the source is depicted in Alg. 2. We omit the subscript i from each variable. The source first initializes the variables used in the algorithm in an identical fashion to Alg. 1 (lines 1-9). The only difference is that the source does not use the sets Out_i^z and EO_i^z and initializes $res^1(s)$ to *False* for all subsequences s (recall that the source never requests reports in the first epoch).

In the send phase, the source only sends messages in the first n^{seq} odd rounds of stage 1 of each epoch, in round $2n^{seq} + 2$ of stage 1 of every epoch, and in the dissemination rounds of every stage (lines 11-25). Specifically, for every round $m = 2s + 1$ from stage 1 with $0 \leq s \leq 2n^{seq} - 1$, the source sends the pair $(s, res^z(s))$ to all agents i (line 14). In round $m = 2n^{seq} + 2$ of stage 1, the source sends to every agent i the seed sd_i^z and a set K_i^z containing the key κ_j^z and verdict $bp^z(j)$ of every agent $j \neq i$ (lines 17-20). In a dissemination round $m = 2n^{seq} + 2 + y$ with $1 \leq y \leq \nu$, the source sends $(y, e, [e_y^{zt}])$ to every agent j from a random set S of f^{fan} agents, where e either corresponds to $(e_y^{zt})_{\kappa_j^z}$ if j is being punished ($bp^z(j) = True$) or is e_y^{zt} otherwise (lines 22-24).

In the receive phase, the source processes all the messages that contain monitoring information (lines 26-41). In the first n^{seq} even rounds $m = 2s$ of stage 1 of every epoch z such that $res^z(s) = True$, the source processes the reports that every agent i sends relative to the subsequence s . If i does not send valid reports, then the source sets $vs^z(i) = False$ (line 30), otherwise the source stores the reports included by i in Re_i^s and Se_i^s in the variables re^z and se^z , respectively (lines 31-34). In round $2n^{seq} + 1$ of stage 1 of every epoch $z > 1$, the source stores the accusations received from every agent i : if agent i does not send a valid vector vs_i^z , then the

Algorithm 1 $\sigma_i^{f^{fan}}$: i 's gossip dissemination protocol with fanout f^{fan}

```

1: for all epochs  $z \geq 1$  do
2:    $Out_i^z, ME_i^z \leftarrow \emptyset$ 
3:    $vs_i^z(i) \leftarrow True$ 
4:    $bp_i^z(i) \leftarrow False$ 
5:   for all agents  $j \in \mathcal{N} \cup \{\text{source}\} \setminus \{i\}$  do
6:      $vs_i^z(j) \leftarrow True$ 
7:      $bp_i^z(j) \leftarrow False$ 
8:     for all stages  $1 \leq t \leq n^{spe}$  and rounds  $1 \leq m \leq \tau$  do
9:        $EO_i^z(j, t, m) \leftarrow \emptyset$ 
10:       $EF_i^z(j, t, m) \leftarrow \emptyset$ 
11:     for all stages  $1 \leq t \leq n^{spe}$  and events  $1 \leq y \leq \nu$  do
12:        $re_i^z(i, j, t, y), se_i^z(i, j, t, y) \leftarrow \perp$ 
13: for all epochs  $z \geq 1$ , stages  $1 \leq t \leq n^{spe}$ , rounds  $1 \leq m \leq \tau$  do
14:   Phase 1: send phase
15:   if  $vs_i^z(i) = True$  then ▷ Sends messages if  $i$  only sent valid messages before
16:     if  $t = 1$  and  $m = 2s$  with  $1 \leq s \leq n^{seq}$  and  $res_i^z(s) = True$  then
17:       for all agents  $j \neq i$  and events  $e_y^{(z-1)t} \in \mathcal{E}(z-1, s)$  do ▷ Source requested reports for  $s^{th}$  sequence
18:          $Re_i^s(i, j, t, y) \leftarrow re_i^{z-1}(i, j, t, y)$ 
19:          $Se_i^s(i, j, t, y) \leftarrow se_i^{z-1}(i, j, t, y)$ 
20:         Send  $\langle Re_i^s, Se_i^s \rangle$  to the source
21:       else if  $z > 1$  and  $t = 1$  and  $m = 2n^{seq} + 1$  then
22:         Send  $vs_i^{z-1}$  to the source ▷ Send accusations
23:       else if  $m > 2n^{spe} + 2$  then ▷ Dissemination round
24:         for all agents  $j \neq i$  do
25:           Send  $EF_i^z(j, t, m)$  to  $j$  ▷ Send events to  $j$ 
26:   EndPhase
27:   Phase 2: receive phase
28:   if  $t = 1$  and  $m = 2s + 1$  with  $0 \leq s \leq n^{seq} - 1$  and source sent  $\langle s, r \rangle$  then
29:      $res_i^z(s) \leftarrow r$  ▷ Source requests reports of  $s^{th}$  sequence if  $r = True$ 
30:   else if  $t = 1$  and  $(m = 2n^{seq} + 1$  or  $\exists_s(m = 2s$  and  $res_i^z(s) = True))$  and  $i$  did not send a valid message then
31:      $vs_i^z \leftarrow False$  ▷  $i$  did not send requested reports or accusations
32:   else if  $t = 1$  and  $m = 2n^{seq} + 2$  and source sent  $\langle s, K_i^z \rangle$  then
33:      $sd_i^z \leftarrow s$  ▷ Seed for epoch  $z$ 
34:     for all  $(j, v, \kappa) \in K_i^z$  do ▷ Stores keys and verdicts
35:        $bp_i^z(j) \leftarrow v$ 
36:        $\kappa_j^z \leftarrow \kappa$ 
37:   else if  $m > 2n^{seq} + 2$  then ▷ Dissemination round
38:     for all  $j \in \mathcal{N} \cup \{\text{source}\} \setminus \{i\}$  do
39:       if  $j$  sent invalid message then  $vs_i^z(j) \leftarrow False$  ▷ Invalidates state
40:       else if  $j$  sent set  $S$  of events to  $i$  then
41:         for all  $(y, e, s) \in S$  do
42:           if  $0 \leq \text{age}(y, m) < n$  and  $re_i^z(i, j, t, y) = \perp$  then ▷  $j$  did not send  $y^{th}$  event to  $i$  before
43:              $re_i^z(i, j, t, y) \leftarrow \text{age}(y, m)$  ▷ Registers round in which  $j$  first sends event to  $i$ 
44:              $e^* \leftarrow \varphi_j(e)$  ▷ If  $j$  is being punished, deciphers event with  $\kappa_j^z$ 
45:              $EO_i^z(t, y) \leftarrow EO_i^z(t, y) \cup \{(y, e^*, s)\}$ 
46:             if  $se_i^z(i, l, t, y) = \perp$  for all  $l$  then ▷  $i$  has not forwarded event before
47:               for all  $l \in SG_i(sd_i^z, t, y)$  do
48:                  $EF_i^z(l, t, m+1) \leftarrow EF_i^z(l, t, m+1) \cup \{(y, \varphi_l(e^*), s)\}$  ▷ Prepares to forward event
49:             else  $vs_i^z(j) \leftarrow False$ 
50:           if  $i$  sent invalid message to  $j$  then  $vs_i^z(i) \leftarrow False$  ▷  $i$  sends no further messages
51:           else if  $i$  sent valid set  $S$  to  $j$  then
52:             for all  $(y, e, s) \in S$  do
53:               if  $0 \leq \text{age}(y, m) < n$  and  $se_i^z(i, j, t, y) = \perp$  then
54:                  $se_i^z(i, j, t, y) \leftarrow \text{age}(y, m)$  ▷ Registers round in which  $i$  first sends event to  $j$ 
55:               else  $vs_i^z(i) \leftarrow False$ 
56:             for all  $(y, e, s)$  not correctly forwarded by  $i$  do
57:                $ME_i^z \leftarrow ME_i^z \cup \{\text{sub}(t, y)\}$  ▷ Some event from  $\text{sub}(t, y)$  was not forwarded correctly
58:   EndPhase
59:   Phase 3: update phase
60:   if  $m > 2n^{seq} + 2$  then ▷ Dissemination round
61:     for all agents  $j \neq i$  and  $(y, e, s) \in EF_j^z(j, t, m+1)$  with  $\text{sub}(t, y) \in ME_i^z$  do
62:        $EF_i^z(j, t, m) \leftarrow EF_j^z(j, t, m) \setminus \{(y, e, s)\}$  ▷ Cleans  $EF_i^z$ 
63:      $O \leftarrow \emptyset$ 
64:     for all  $1 \leq y \leq \nu$  such that  $(t, y) \notin Out_i^z$ ,  $(y, e, s) \in EO_i^z(t, y)$ , and  $\text{comp}(e, s) \neq \perp$  do
65:        $O \leftarrow O \cup \{(y, \text{comp}(e, s), s)\}$  ▷ Retrieves  $e_y^{zt}$  from  $e$ 
66:        $Out_i^z \leftarrow Out_i^z \cup \{(t, y)\}$ 
67:     Output( $O$ )
68:   EndPhase

```

source sets $vs^z(i) = False$ (line 37), otherwise, the source stores every accusation contained in vs_i^z in the variable vs^z (lines 39-40).

In the update phase, the source prepares the requests of reports for the next epoch and prepares the verdicts, keys, and seeds (lines 42-61). In the last round from epoch z , for every subsequence numbered $1 \leq s \leq n^{\text{seq}}$, the source sets $res^{z+1}(s) = True$ with independent probability p^{seq} , otherwise it sets $res^{z+1}(s) = False$ (lines 43-45). In round $2n^{\text{seq}} + 1$ of stage 1 of every epoch z , for every agent i , the source generates the key κ_i^z and seed sd_i^z at random, and, if $z > 1$, the source determines the verdict $bp_i^z(i)$ of i (lines 46-60). Specifically, for every event $e_{y'}^{(z-1)t'}$ disseminated in epoch $z - 1$, the source determines the first round m^* when i received a tuple with the form (y, e, s) , where $m^* = \perp$ if i did not receive any such tuple; the source triggers a punishment of i ($bp^z(i) = True$) if and only if (i) $vs^{z-1}(i) = False$, which holds when some agent sent an accusation against i , the source received an invalid message from i , or i did not send all requested messages, (ii) i received the event ($m^* \neq \perp$) and did not forward it in round $m^* + 1$ to all agents from $SG_i(sd_i^{z-1}, t', y')$, or (iii) i sent the event to some agent not from $SG_i(sd_i^{z-1}, t', y')$ or in some round $m' \neq m^* + 1$.

5.1.2 Parametrising the Protocol

The parameters n^{spe} , p^{seq} , and n^{seq} determine both the average cost of sending monitoring messages and the probability of detecting deviations. We now describe how to define the parameters n^{seq} and p^{seq} as functions of n^{spe} in a way that ensures that the probability of detecting deviations is sufficiently high, such that agents do not gain by deviating, and the average cost of monitoring decreases with n^{spe} , such that we can arbitrarily minimize this cost by increasing n^{spe} .

In the next section, we show that the average overhead of monitoring is sufficiently low if (1) p^{seq} decreases with n^{spe} and (2) $p^{\text{seq}}n^{\text{seq}}$ does not decrease with n^{spe} . Moreover, the probability of detecting deviations is sufficiently high if (1) p^{seq} decreases slower than $1/n^{\text{spe}}$ and (2) $p^{\text{seq}}n^{\text{seq}}$ does not increase with n^{spe} . More precisely, the following three conditions that relate p^{seq} and n^{seq} with n^{spe} are sufficient to prove the main result of this chapter:

C1. $p^{\text{seq}} = \omega(1/n^{\text{spe}})$

C2. $p^{\text{seq}} = o(1)$

Algorithm 2 Source's gossip dissemination protocol with fanout f^{fan}

```

1: for all epochs  $z \geq 1$  do
2:   for all agents  $i \in \mathcal{N} \cup \{\text{source}\}$  do
3:      $vs^z(i) \leftarrow \text{True}$ 
4:      $bp^z(i) \leftarrow \text{False}$ 
5:     for all agents  $j \neq i$ , stages  $1 \leq t \leq n^{\text{spe}}$ , and events  $1 \leq y \leq \nu$  do
6:        $re^z(i, j, t, y) \leftarrow \perp$ 
7:        $se^z(i, j, t, y) \leftarrow \perp$ 
8:   for all subsequences  $1 \leq s \leq n^{\text{seq}}$  do
9:      $res^1(s) \leftarrow \text{False}$ 

10: for all epochs  $z \geq 1$ , stages  $1 \leq t \leq n^{\text{spe}}$ , rounds  $1 \leq m \leq \tau$  do
11:   Phase 1: send phase
12:   if  $t = 1$  and  $m = 2s + 1$  with  $0 \leq s \leq n^{\text{seq}} - 1$  then
13:     for all agents  $i$  do
14:       Send  $\langle s, res^z(s) \rangle$  to  $i$  ▷ Mediator requests reports for sequence  $s$  if  $res^{z-1}(s) = \text{True}$ 
15:   else if  $t = 1$  and  $m = 2n^{\text{spe}} + 1$  then
16:     for all agents  $i$  do
17:        $K \leftarrow \emptyset$ 
18:       for all agents  $j \neq i$  do
19:          $K \leftarrow (j, bp^z(j), \kappa_j^z)$ 
20:       Send  $\langle sd_i^z, K \rangle$  to  $i$  ▷ Sends seeds, verdicts, and keys before dissemination starts
21:   else if  $m = 2n^{\text{spe}} + 2 + y$  with  $1 \leq y \leq \nu$  then ▷ Dissemination round
22:      $S \leftarrow$  random set of  $f^{fan}$  agents
23:     for all  $j \in S$  do
24:       Send  $\{(y, \varphi_j(e_y^{zt}), [e_y^{zt}])\}$  to  $j$  ▷ Punishes  $j$  iff  $bp_j^z(j) = \text{True}$ 
25:   EndPhase
26:   Phase 2: receive phase
27:   for all agents  $i \in \mathcal{N}$  do
28:     if  $t = 1$  and  $m = 2s$  with  $1 \leq s \leq n^{\text{seq}}$  and  $req^z(s) = \text{True}$  then
29:       if  $i$  did not send valid message then
30:          $vs^z(i) \leftarrow \text{False}$  ▷  $i$  did not send requested reports
31:       else if  $i$  sent  $\langle Re_i^s, Se_i^s \rangle$  then
32:         for all agents  $j \neq i$  and events  $e_y^{(z-1)t} \in \mathcal{E}(z-1, s)$  do ▷ Registers reports
33:            $re^{z-1}(i, j, t, y) \leftarrow Re_i^s(i, j, t, y)$ 
34:            $se^{z-1}(i, j, t, y) \leftarrow Se_i^s(i, j, t, y)$ 
35:       else if  $t = 1$  and  $m = 2n^{\text{spe}} + 1$  then
36:         if  $i$  did not send valid message then
37:            $vs^z(i) \leftarrow \text{False}$  ▷  $i$  did not send accusations
38:         else if  $i$  sent  $vs_i^{z-1}$  then
39:           for all agents  $j \neq i$  such that  $vs_i^{z-1}(j) = \text{False}$  do
40:              $vs^{z-1}(j) \leftarrow \text{False}$  ▷ Registers accusation against  $j$ 
41:       EndPhase
42:   Phase 3: update phase
43:   if  $t = n^{\text{spe}}$  and  $m = \tau$  then ▷ End of an epoch
44:     for all subsequences  $1 \leq s \leq n^{\text{seq}}$  do
45:        $res^{z+1}(s) \leftarrow \text{True}$  with prob.  $p^{\text{seq}}$  or  $\text{False}$  otherwise ▷ Requests reports for  $s$  with probability  $p^{\text{seq}}$ 
46:   else if  $t = 1$  and  $m = 2n^{\text{spe}}$  then ▷ Prepares verdicts, keys, and seeds
47:     for all agents  $i$  do
48:        $\kappa_i^z \leftarrow$  unique random key
49:        $sd_i^z \leftarrow$  random seed
50:     if  $z > 1$  then
51:       for all sequences  $1 \leq s \leq n^{\text{seq}}$  and events  $e_{y'}^{(z-1)t'} \in \mathcal{E}(z-1, s)$  such that  $res^{z-1}(s) = \text{True}$  do
52:          $m^* \leftarrow \min_m \exists j \in \mathcal{N} \cup \{\text{source}\} \setminus \{i\} se^{z-1}(j, i, t', y') = m$  ▷ First round when  $i$  receives event
53:         if  $vs^{z-1}(i) = \text{False}$  then
54:            $bp^z(i) \leftarrow \text{True}$  ▷  $i$  sent invalid message in epoch  $z-1$ 
55:         else if  $m^* \neq \perp$  and  $\exists j \in SG_i(sd_i^{z-1}, t', y') re^{z-1}(j, i, t', y') \neq m^* + 1$  then
56:            $bp^z(i) \leftarrow \text{True}$  ▷  $i$  did not send event in  $m^* + 1$  to all agents in  $SG_i(sd_i^{z-1}, t', y')$ 
57:         else if  $\exists j \notin SG_i(sd_i^{z-1}, t', y') re^{z-1}(j, i, t', y') \neq \perp$  then
58:            $bp^z(i) \leftarrow \text{True}$  ▷  $i$  sent event to some agent  $j \notin SG_i(sd_i^{z-1}, t', y')$ 
59:         else
60:            $bp^z(i) \leftarrow \text{False}$  ▷  $i$  is not punished
61:       EndPhase

```

C3. $p^{\text{seq}} n^{\text{seq}} = \theta(1)$

Note that C3 implies that $p^{\text{seq}} = c/n^{\text{seq}}$ for some constant c . For instance, if $p^{\text{seq}} = 1/\sqrt{n^{\text{spe}}}$ and $n^{\text{seq}} = \sqrt{n^{\text{spe}}}$, then all three properties are satisfied. More generally, it suffices that $p^{\text{seq}} = c/g(n^{\text{spe}})$ and $n^{\text{seq}} = g(n^{\text{spe}})$ for some constant c and function g sublinear on n^{spe} .

5.2 Proof of Main Result

We now prove our main result. We first discuss the most important cryptographic assumptions. Then, we prove that, if agents are sufficiently patient and f^{fan} is c -individually rational for some constant c , then $\vec{\sigma}^{\text{fan}}$ is a sequential equilibrium, and the average utility of every agent i with $\vec{\sigma}^{\text{fan}}$ is arbitrarily close to $x_i(f^{\text{fan}})$. Theorem 1 follows immediately from this.

5.2.1 Cryptographic Assumptions

We discuss assumptions about the symmetric ciphers, the signature-scheme used by the source to sign events, and the pseudo-random set generator.

Regarding symmetric ciphers, we make the standard assumption that it is computationally hard for an agent to break the commutative symmetric cipher. Specifically, given an agent i and event e_y^{zt} , we assume that if $e = (e_y^{zt})_{\kappa_i^z}$, then $\text{comp}(e, [e_y^{zt}]) = \perp$. In other words i obtains benefits in an epoch z iff i is not being punished.

Regarding signatures, we assume that it is computationally hard for agents to replicate the signature of the source. More precisely, to simplify the analysis, we assume that the actions of agent i available to I_i are restricted as follows: i can send a tuple (y, e, s) to agent j in round m such that $s = [e_y^{zt}]$ iff i receives (y, e', s) in round $m' < m$ from some agent l ; moreover, if $\text{comp}(\varphi_j(e), s) \neq \perp$, then $\text{comp}(\varphi_l(e'), s) \neq \perp$. That is, we assume that agent i cannot replicate the signature $[e_y^{zt}]$ without previously receiving the signature in a message, and i cannot guess e_y^{zt} before receiving (y, e', s) such that i can compute e_y^{zt} from $\varphi_l(e')$ and s .

Finally, given an agent i and seed sd_i , we assume that the function SG_i generates a sequence of νn^{spe} sets of f^{fan} agents, such that, for every t and y , the set $SG_i(sd_i, t, y)$ (which corresponds to the $(ty)^{\text{th}}$ set of this sequence) is generated approximately at random and independently

of other sets. We now show that if a PRNG function exists, then SG_i can be defined from G such that the above assumption holds. Note that we can define a mapping between sets of f^{fan} agents and numbers in $\{0 \dots \binom{n-1}{f^{fan}} - 1\}$, so every stream of sets of f^{fan} agents has a bitwise representation. Thus, we can define SG_i using any PRNG function that generates approximately random streams of bits. For instance, we can use the PRNG function used by Abraham et al. (2013).

Specifically, a PRNG function G generates a stream of $m(k)$ bits from a seed $s \in \{0, 1\}^k$ such that no probabilistic time machine can distinguish between the outcome of G and a truly random sequences of $m(k)$ bits, where k is a security parameter and m is a polynomial on k such that $m(k) > k$. Formally, a function G is a PRNG if it satisfies the following property:

- For all constants $\xi > 0$, there exists k^* such that, for all $k \geq k^*$, sequence of bits $\vec{b} \in \{0, 1\}^{m(k)}$, and bit position $1 \leq l \leq m(k)$, we have

$$\left| \frac{1}{2^{m(k)}} - \frac{|S|}{|S'|} \right| < \xi,$$

where S is the set of seeds s such that $G(s) = \vec{b}$ and S' is the set of seeds s' such that $G(s') = \vec{b}$ or $G(s') = \vec{b}'$, where \vec{b}' differs from \vec{b} only in the l^{th} bit.

We define SG_i assuming only that a pseudo-random function G exists. Let $b = \binom{n-1}{f^{fan}}$ be the number of different subsets of f^{fan} agents to which i may forward events, and fix a correspondence between numbers in $0 \dots b-1$ and sets. Let $m(k) = \nu n^{\text{spe}} \cdot \phi(k) \cdot \lceil \log(b) \rceil$, where $\phi(k)$ is some polynomial. Intuitively, a stream of $m(k)$ bits represents a stream of νn^{spe} numbers, where each number is represented using $\phi(k) \cdot \lceil \log(b) \rceil$ bits. Let $K = \{0 \dots 2^{\phi(k) \cdot \lceil \log(b) \rceil}\}$ denote the set of all numbers from this stream. Given the stream of νn^{spe} numbers generated by G , a stage number $1 \leq t \leq n^{\text{spe}}$, and event number $1 \leq y \leq \nu$, let $o \in K$ be the $(t + y)^{th}$ number in the stream; we define $SG_i(sd_i, t, y)$ as the set of f^{fan} agents that corresponds to the number $l = o \bmod b$.

We show in Proposition 2 that SG_i generates streams of approximately random and independent sets. Let \vec{S} be a stream of νn^{spe} sets that fixes every set S_i^y to which agent i has to forward event e_y^{zt} . Let $\vec{S}_{-(t,y)}$ be a stream that fixes every set but S_y^t , and let $P_{SG_i}(S_y^t | \vec{S}_{-(y,t)})$ be the probability of SG_i generating set S_y^t conditioning on SG_i generating the stream of sets $\vec{S}_{-(y,t)}$.

Proposition 2. (PRSG1) *If G is a PRNG, then for all $\xi > 0$, there exists k^* such that for all $k \geq k^*$, stage number $1 \leq t \leq n^{spe}$, event number $1 \leq y \leq \nu$, sets S_y^t , and streams $\vec{S}_{-t,y}$, we have*

$$\left| P_{SG_i}(S_y^t \mid \vec{S}_{-(t,y)}) - \frac{1}{b} \right| \leq \xi.$$

Proof. Note that there is a one-to-many mapping between sets and numbers in K . By the properties of the modulo operation, there are values d and $r \in \{0, \dots, b-1\}$ such that r sets are mapped to $d+1$ numbers in K and $b-r$ sets mapped only to d numbers in K , where d is the integer division of $\#K$ by b and $r \in \{1, \dots, b-1\}$ is the remainder of this division. By the assumption that G is a PRNG, the probability of generating each number $o \in K$ as the y^{th} number from the stream of νn^{seq} numbers is arbitrarily close to selecting o independently at random, even if we fix the other numbers of the stream. Therefore, the probability of selecting each set as the y^{th} set is arbitrarily close to a value v in $[d/\#K, \dots, (d+1)/\#K]$, even if we fix the other sets. As $\phi(k)$ increases, the interval converges to the point $1/b$, so, by increasing k and $\phi(k)$, the probability of selecting each set gets arbitrarily close to $1/b$. \square

5.2.2 Sequential Equilibrium Proof

To prove that $\vec{\sigma}^{fan}$ is a sequential equilibrium, we have to show that there is a belief system consistent with $\vec{\sigma}^{fan}$ such that no agent i gains by deviating at any given information set I_i . We start by defining a belief system μ^{fan} consistent with $\vec{\sigma}^{fan}$ and then show that no agent i gains by deviating at any I_i given that i uses μ^{fan} to compute its expected utility.

In the definition of μ^{fan} , we have to model the belief that every agent i has regarding how the events that other agents sent to i were generated, since this belief determines whether i should output a tuple at each information set I_i . We define a belief system such that all the information that i believes to have in I_i about any given event e_y^{zt} is the sequence S of valid tuples with the form (y, e, s) that i receives in I_i . Moreover, if i receives $(y, \varphi_l(e), s)$ from l , i is not being punished, and $comp(e, s) = \perp$, then i believes that e is a random event, such that i cannot compute e_y^{zt} from e , even if i tries to decipher e with keys that i knows.

We now provide a formal definition of μ^{fan} that captures the above intuition. Let $\vec{\sigma}^M$ be a completely mixed strategy profile (i.e., attributes positive probability to every action at all information sets) defined as follows. Recall that there is a finite number of actions a_i available

to an agent i at any given information set I_i ; let K_{I_i} denote this number. For all agents i and round- m information set I_i for i , agent i follows σ_i^{ffan} at I_i with probability $1 - 1/1M$ and distributes the remaining probability $1/M$ across all K_{I_i} actions as follows. If m is a monitoring round, then i selects each action with probability $1/MK_{I_i}$. If m is a dissemination round, i decides on which value to output and which messages to send to each agent j independently. Namely, i outputs each set S of tuples with independent probability $1/Mo$, where o is the number of sets that i may output. For each agent $j \neq i$, i sends an invalid message with probability $1/(2MJ)$ where J is the number of possible invalid messages, and with probability $1/2M$ sends a valid message \vec{m}_i defined as follows. Given y , let $S_{I_i,y}^j$ be the set of tuples that i may send to j at I_i containing y and let S_y^j be the subset of tuples (y, e, s) such that e corresponds to e_y^{zt} ciphered with zero or more keys κ_l^z for $l \neq i$ (i.e., $e = ((e_y^{zt})_{\kappa_{l_1}^z} \dots)_{\kappa_{l_k}^z}$ for some set of k agents l_1, \dots, l_k different from i). For each $1 \leq y \leq \nu$, i sends no tuple with the form (y, e, s) with probability $1/2$ and with probability $1/2$ selects a tuple (y, e, s) defined as follows: with probability $(1 - 1/M)(1/\#(S_{I_i,y}^j \setminus S_y^j))$, i selects a tuple $(y, e, s) \in S_{I_i,y}^j \setminus S_y^j$, and with probability $(1/M)(1/\#(S_y^j))$ i selects $(y, e, s) \in S_y^j$. For all agents i , information sets I_i , and global history $h \in I_i$, we have:

$$\mu^{ffan}(h) = \lim_{M \rightarrow \infty} \frac{P_{\vec{\sigma}^M}(h)}{\sum_{h' \in I_i} P_{\vec{\sigma}^M}(h')}.$$

It is easy to see that $\vec{\sigma}^M$ is completely mixed and converges to $\vec{\sigma}^{ffan}$. Moreover, note that when an agent i deviates in a dissemination round, i decides on which tuple to send to each agent j independently, and the probability of i including in valid messages only random tuples converges to 1. So, with μ^{ffan} , if agent i is not being punished and some l sends (y, e, s) to i such that $comp(e, s) = \perp$, then i believes that e is a random event, so i does not gain from outputting e .

Having defined μ^{ffan} , we now show that no agent i gains by deviating at any given information set I_i , given that the expected utility of i is computed using μ^{ffan} . Note that the number of deviations that an agent i may perform at I_i is infinite. Therefore, it is not tractable to compare the utility of agents using $\vec{\sigma}^{ffan}$ with that of agents using $(\sigma_i, \vec{\sigma}_{-i}^{ffan})$ for all possible deviating strategies σ_i of i . Fortunately, by the one-shot deviation property (Hendon et al. 1996) (OSD), we only need to consider one-shot deviations of i at I_i , i.e., deviations where i does not take actions at I_i according to the protocol but does not deviate at every other information set. By the OSD, if i does not gain by performing a one-shot deviation at any given information set

I_i , then $\vec{\sigma}^{fan}$ is a sequential equilibrium. Proposition 3 formalizes this intuition³. Given an information set I_i for i , an action $a_i \in \mathcal{A}_i(I_i)$, and a strategy profile $\vec{\sigma}$, let $\vec{\sigma}|_{I_i, a_i}$ denote the strategy profile that differs from $\vec{\sigma}$ exactly in that i deterministically takes action a_i at I_i .

Proposition 3. (*One-shot Deviation Property*) *A strategy profile $\vec{\sigma}$ is a sequential equilibrium if and only if there exists a belief system μ consistent with $\vec{\sigma}$ such that for all agents i , information sets I_i for i , and actions $a_i^*, a_i' \in \mathcal{A}_i(I_i)$ such that $\sigma_i^{fan}(a_i^* | I_i) > 0$, $u_i(\vec{\sigma}|_{I_i, a_i^*} | I_i) \geq u_i(\vec{\sigma}|_{I_i, a_i'} | I_i)$.*

We now show that for all agents i and information sets I_i , i does not gain by performing a one-shot deviation at I_i . We start by proving Lemmas 4 and 5, which list properties of $\vec{\sigma}^{fan}$ regarding the probability of i being punished in future epochs and the probability of i sending and receiving events, respectively, when i performs a one-shot deviation at I_i .

Lemma 4 relates the probability of agent i being punished in epoch $z + 1$ with the values of $vs_i^z(i)$ and ME_i^z . Given an information set I_i , let $v_i|_{I_i}$ denote the value of variable v_i in agent i at I_i . Given epoch z , agent i , and information set I_i , say that the source triggers a punishment of i in epoch z' after I_i with probability p if p is the probability $\mu_{\vec{\sigma}^{fan}, I_i}^{fan}(r)$ of every run r in which the source sets $bp^{z'}(i) = True$ in round $2n^{seq} + 1$ from stage 1 from epoch z' . Finally, given a run r and agent j , let $r_j(z, t, m)$ denote j 's information set at round m from stage t from epoch z .

Lemma 4. *For all agents i and round- m information set I_i from stage t from epoch z , the following properties hold:*

- M1. If $vs_i^z(i)|_{I_i} = True$, then the source triggers a punishment of i in epoch $z + 1$ after I_i with probability $P^*(\#ME_i^z|h) = 1 - (1 - p^{seq})^{\#ME_i^z|_{I_i}}$.*
- M2. If $vs_i^z(i)|_{I_i} = False$, then the source triggers a punishment of i in epoch $z + 1$ after I_i with probability 1.*
- M3. For all $z' > z + 1$, the source triggers a punishment of i in epoch z' after I_i with probability 0.*

Proof. We prove each property in turn:

³A proof of the one-shot deviation property is provided in (Hendon et al. 1996).

- M1. If $vs_i^z(i)|_{I_i} = True$, then i sends no invalid messages in epoch z and sends all requested messages to the source, which implies that in round $2n^{\text{seq}} + 1$ of the first stage from epoch $z+1$ no agent sends an accusation against i and the source never sets $vs_i^z(i) = False$. Hence, the source triggers a punishment of i in epoch $z + 1$ iff it detects that i did not forward some event according to SG_i and sd_i^z . Say that the reports relative to i and event e_y^{zt} are inconsistent in a run $r \in \mathcal{R}(I_i)$ iff there exists j such that $se_j^z(j, i, t, y)|_{r_j(z+1,1,1)} = m^* \neq \perp$, m^* is the first round when some agent j sent a valid tuple (y, e, s) to i (according to se_j^z), and there exists $l \neq i$ such that either (i) $l \in SG_i(sd_i^z, t, y)$, and $re_l^z(l, i, t, y)|_{r_l(z+1,1,1)} \neq m^* + 1$ or (ii) $l \notin SG_i(sd_i^z, t, y)$ and $re_l^z(l, i, t, y)|_{r_l(z+1,1,1)} \neq \perp$. Since i forwards all tuples (y', e', s') with $sub(t', y') \notin ME_i^z|_h$ after I_i when using $\bar{\sigma}^{fan}$, (i.e., sends the tuple to the agents in $SG_i(sd_i^z, t', y')$ upon receiving it for the first time), it is easy to see that, for all runs r such that $\mu_{\bar{\sigma}^{fan}, I_i}^{fan}(r) > 0$, the reports relative to i and e_y^{zt} are inconsistent in r iff $sub(t, y) \in ME_i^z|_{I_i}$. Note that in a run r if the source requests the reports relative to subsequence s in the monitoring rounds of stage 1 from epoch $z + 1$, then every agent j sends $se_j^z(j, i, t, y)|_{r_j(z+1,1,1)}$ and $re_j^z(j, i, t, y)|_{r_j(z+1,1,1)}$ to the source for every $(t, y) \in \mathcal{E}(s)$. Therefore, in a run r , the source detects that i did not forward events correctly iff the source requests some subsequence s and there exists (t, y) such that the reports relative to i and e_y^{zt} are inconsistent in r , which is true iff $s \in ME_i^z|_{I_i}$. Such runs happen with probability $P^*(\#ME_i^z(i)|_{I_i})$.
- M2. If $vs_i^z(i)|_{I_i} = False$, then at least one of the following is true: (i) i sent an invalid message to the source or omitted a message when i was supposed to send that message such that the source sets $vs^z(i) = False$, (ii) i sent an invalid message to an agent j , such that j did set $vs_j^z(i) = False$, in which case j sends an accusation against i to the source in stage 1 from epoch $z + 1$. Whether (i) or (ii) are true, the source sets $vs^z(i) = False$ and triggers a punishment of i in epoch $z + 1$.
- M3. Since $z' - 1 > z$, when using $\bar{\sigma}^{fan}$ agent i sends only valid messages and omits no message in epoch $z' - 1$, and forwards all events according to $\bar{\sigma}^{fan}$ and $sd_i^{z'-1}$, so i never sets $vs_i^{z'-1}(i) = False$ and never adds tuples to $ME_i^{z'-1}$. By M1, the source triggers a punishment of i in epoch z' with probability $P^*(\#ME_i^{z'-1}) = P(0) = 0$.

□

We now prove Lemma 5, which compares the probability of i receiving and forwarding each event when i follows $\vec{\sigma}^{fan}$ and when it performs a one-shot deviation. First, we need some additional definitions. Given a round- m information set I_i , event e_y^{zt} is said to be disseminated after I_i if the source disseminates the event in round m or after I_i is observed, otherwise, e_y^{zt} is said to be disseminated before I_i . Let q_i be a function such that for all runs r , epoch numbers z , stage numbers $1 \leq t \leq n^{spe}$, and event numbers $1 \leq y \leq \nu$, $q_i(r, z, t, y) = 1$ if i receives a tuple (y, e, s) for some e and s or $q_i(r, z, t, y) = 0$ otherwise. We denote by $\mathbb{E}^{\vec{\sigma}}[q_i(r, z, t, y) \mid I_i]$ the expected value of $q_i(r, z, t, y)$ when agents use the strategy profile $\vec{\sigma}$ conditioning on the run being $\mathcal{R}(I_i)$, that is,

$$\mathbb{E}^{\vec{\sigma}}[q_i(r, z, t, y) \mid I_i] = \sum_{r \in \mathcal{R}(I_i)} \mu_{\vec{\sigma}^{fan}, I_i}^{fan}(r) q_i(r, z, t, y).$$

Given an information set I_i , strategy profile $\vec{\sigma}$, and numbers t and y , let $P_{\vec{\sigma}, I_i}^{send}(t, y)$ and $P_{\vec{\sigma}, I_i}^{out}(t, y)$ denote the probabilities of i sending and outputting the tuple $(y, e_y^{zt}, [e_y^{zt}])$, respectively. Actions a_i^* and a_i' are said to be equivalent if agent i sends only valid messages in both actions and, for all event numbers y , i sends a tuple containing y to j in a_i^* iff i sends a tuple containing y to j in a_i' ; a_i^* is said to be contained in a_i' if i sends a tuple containing y to j in a_i^* only if i sends a tuple containing y to j in a_i' . Finally, given a run r , agent j , epoch z , and e' , we denote by $\varphi_j^{r,z}(e)$ the event e' such that $e' = (e)_{\kappa_j^z}$ if the source triggers a punishment of j during epoch z in r or $e' = e$ otherwise.

Lemma 5 shows that (i) disseminating events with the pseudo-random set generator is equivalent to sending events using eager-push gossip dissemination (Properties D1 and D2), (ii) an agent receives events disseminated in the future with a probability that is independent from the current action (Property D3), (iii) every agent i receives each event e_y^{zt} disseminated in the future ciphered with κ_i^z iff i is being punished (Property D4), (iv) if a_i^* and a_i' are equivalent, then the probability of i receiving and forwarding each event already sent by the source is independent of whether i takes a_i^* or a_i' (Property D5), and (v) if a_i^* is contained in a_i' , then the probability of i outputting an event already being disseminated does not increase if i takes a_i' instead of a_i^* (Property D6).

Lemma 5. *Fix an agent i , epoch $z \geq 1$, stage $1 \leq t \leq n^{spe}$, round $1 \leq m \leq \tau$, round- m information set I_i for i from stage t from epoch z , and belief system μ consistent with $\vec{\sigma}^{fan}$. Fix two arbitrary actions $a_i^*, a_i' \in \mathcal{A}_i(I_i)$, and let $\vec{\sigma}^* = (\sigma_i^{fan} \mid_{I_i, a_i^*}, \vec{\sigma}_{-i}^{fan})$ and $\vec{\sigma}' = (\sigma_i^{fan} \mid_{I_i, a_i'}, \vec{\sigma}_{-i}^{fan})$.*

The following properties hold:

D1. For all $z' > z$, $1 \leq t' \leq n^{spe}$, and $1 \leq y' \leq \nu$, we have $\mathbb{E}^{\bar{\sigma}^*} [q_i(r, z', t', y') \mid I_i] \approx q(f^{fan})$.

D2. For all events $e_{y'}^{z't'}$ disseminated after I_i , we have $\mathbb{E}^{\bar{\sigma}^*} [q_i(r, z', t', y') \mid I_i] \approx v \leq q(f^{fan})$.

D3. For all events $e_{y'}^{z't'}$ disseminated after I_i , we have $\mathbb{E}^{\bar{\sigma}^*} [q_i(r, z', t', y') \mid I_i^*] = \mathbb{E}^{\bar{\sigma}' } [q_i(r, z', t', y') \mid I_i']$.

D4. For all events $e_{y'}^{z't'}$ disseminated after I_i and runs $r \in \mathcal{R}(I_i)$ such that $\mu_{\bar{\sigma}^*}^{fan}(r) > 0$, if some agent j sends (y', e, s) to i in r in stage t' from epoch z' , then $s = [e_{y'}^{z't'}]$ and $\varphi_j^{r, z'}(e) = \varphi_i^{r, z'}(e_{y'}^{z't'})$.

D5. For all events $e_{y'}^{z't'}$ disseminated before I_i , agent $l \neq i$, and round $m' \geq m$, if a_i^* and a_i' are equivalent, then

$$P_{\bar{\sigma}^*, I_i}^{send}(t', y') = P_{\bar{\sigma}', I_i}^{send}(t', y').$$

D6. For all events $e_{y'}^{z't'}$ disseminated before I_i , if a_i^* is contained in a_i' , then

$$P_{\bar{\sigma}^*, I_i}^{out}(t, y) = P_{\bar{\sigma}', I_i}^{out}(t, y).$$

Proof. We prove properties D1-D6 one at a time:

D1. Fix z' , t' , and y' . Consider a protocol $\bar{\sigma}''$ where the source and the agents disseminate $e_{y'}^{z't'}$ using eager-push gossip dissemination. We show that we can partition the set of runs into sets R of runs of $\bar{\sigma}^*$ and sets R' of runs of $\bar{\sigma}''$ such that the probability of the run being in R with $\bar{\sigma}^*$ is approximately equal to the run being in R' with $\bar{\sigma}''$, and the set of the messages that agents send in R containing $e_{y'}^{z't'}$ is the same as the set of corresponding messages sent in R' . Property D1 follows immediately from this.

Let \vec{S} be a vector that defines a set S_j of f^{fan} agents $l \neq j$ for all entities $j \in \mathcal{N} \cup \{\text{source}\}$. The vector \vec{S} characterizes the dissemination of the event $e_{y'}^{z't'}$ with both $\bar{\sigma}^*$ and $\bar{\sigma}''$: every entity j forwards the event $e_{y'}^{z't'}$ to the f^{fan} agents in S_j ; in $\bar{\sigma}^*$, if j is an agent, then $S_j = SG_j(sd_j^{z'}, t', y')$. Let $\mathcal{R}(\vec{S}) \subseteq \mathcal{R}(I_i)$ denote the set of all runs where agents follow $\bar{\sigma}^{fan}$ or $\bar{\sigma}'$ at and after I_i , and agents disseminate $e_{y'}^{z't'}$ according to \vec{S} . Note that the probability of a run of $\bar{\sigma}^*$ being in $\mathcal{R}(\vec{S})$ depends only on the probability of the source

selecting S_{source} and selecting seeds $sd_j^{z'}$ such that $SG_j(sd_j^{z'}, t', y') = S_j$ for all agents j ; similarly, the probability of a run of $\vec{\sigma}''$ being in $\mathcal{R}(\vec{S})$ depends only on the probability of every entity j selecting S_j , given that S_j is selected independently at random. By PRSG1 and the fact that seeds are selected uniformly at random, the probability of a run of $\vec{\sigma}^*$ being in $\mathcal{R}(\vec{S})$ is arbitrarily close to that of a run of $\vec{\sigma}''$ being in $\mathcal{R}(\vec{S})$, i.e., for an arbitrarily small $\xi > 0$, there exists SG_i such that

$$\left| \sum_{r \in \mathcal{R}(\vec{S})} \mu_{\vec{\sigma}^*, I_i}^{fan}(r) - \sum_{r \in \mathcal{R}(\vec{S})} \mu_{\vec{\sigma}'', I_i}^{fan}(r) \right| \leq \xi. \quad (5.1)$$

Note that \vec{S} determines whether i receives a tuple with the form (y', e', s') (since all agents forward the event according to the protocol), i.e, the value $q_i(r, z', t', y')$ is constant for all runs $r \in \mathcal{R}(\vec{S})$; let $q_i(\vec{S}, z', t', y')$ denote this value. We have

$$q(f^{fan}) = \sum_{\vec{S}: q_i(\vec{S}, z', t', y')=1} \sum_{r \in \mathcal{R}(\vec{S})} \mu'_{\vec{\sigma}'', I_i}(r),$$

and

$$\mathbb{E}^{\vec{\sigma}^*} [q_i(r, z', t', y') \mid I_i] = \sum_{\vec{S}: q_i(\vec{S}, z', t', y')=1} \sum_{r \in \mathcal{R}(\vec{S})} \mu_{\vec{\sigma}^*, I_i}(r).$$

Consequently, the result follows immediately from (5.1).

- D2. The proof is similar to that of Property D1. Fix $e_{y'}^{z't'}$ and $\vec{\sigma}''$ where agents disseminate $e_{y'}^{z't'}$ in stage t' using eager-push gossip dissemination. We show that there is a partition of the set of runs into subsets R of runs of $\vec{\sigma}^*$ and subsets R' of runs of $\vec{\sigma}''$, such that the probability of the run being in R with $\vec{\sigma}^*$ is approximately equal to the probability of the run being in R' with $\vec{\sigma}''$, and the set of the messages that agents send in R containing $e_{y'}^{z't'}$ is a subset of the set of corresponding messages in R' . Property D7 follows immediately from this.

Again, we can describe the dissemination of $e_{y'}^{z't'}$ in both $\vec{\sigma}^*$ and $\vec{\sigma}''$ by a vector \vec{S} of sets of f^{fan} agents. However, now in runs of $\vec{\sigma}^*$ some agents j may not forward tuples containing $e_{y'}^{z't'}$ because it may be that $vs_j^z(j) = \text{False}$ or $sub(t', y') \in ME_j^z$. Let $\mathcal{R}(\vec{S})$ be defined as in the proof of D1, except in runs of $\vec{\sigma}^*$ the set S_j is defined by $SG_j(sd_j^z, t', y')$, regardless of whether j forwards the event. By the definition of μ^{fan} and the fact that

i never learns the seeds of other agents, PRSG1 (5.1) holds (here, we use the fact that in I_i the only information that agent i may have of sd_j for $j \neq i$ is the sequence of sets $S_{y''}^{t'} = SG_j(sd_j^z, t', y'')$ for $y'' < y'$). For all runs $r \in \mathcal{R}(\vec{S})$ of $\vec{\sigma}^*$ and $r' \in \mathcal{R}(\vec{S})$ of $\vec{\sigma}''$, in r' agents send messages containing $e_{y'}^{z't'}$ according to \vec{S} whereas in r agents omit a subset of those messages, so $q_i(r, z, t', y') \leq q_i(r', z, t', y')$. Therefore, the result follows immediately from (5.1).

D3. The probability of i receiving a tuple with the form (y', e, s) in stage t' from epoch z' depends on the seeds $sd_j^{z'}$ and on whether j forwards (y', e, s) upon receiving it for the first time, for all agents $j \neq i$. Specifically, before i receives such tuple, every agent j forwards the tuple to all agents $SG_j(sd_j^{z'}, t', y')$ upon receiving it for the first time iff $vs_j^{z'}(j) = \text{True}$ and $sub(t', y') \notin ME_j^{z'}$. Note that the values of the variables $vs_j^{z'}(j)$ and $ME_j^{z'}$ depend only on what j does before I_i , since j follows $\vec{\sigma}^{fan}$ at and after I_i , so they are independent of i 's action at I_i . Consequently, the probability of i receiving a tuple (y', e, s) is independent of i 's action at I_i . Therefore, D3 holds.

D4. Fix an event $e_{y'}^{z't'}$ disseminated after I_i and run $r \in \mathcal{R}(I_i)$ such that $\mu_{\vec{\sigma}^{fan}, I_i}^{fan}(r) > 0$, and let m' be the round when the source sends $e_{y'}^{z't'}$. We show using induction on the rounds $m' \leq m'' \leq m' + n - 1$ that, if entity j' sends in r the tuple (j, e, s) to agent l , then $e = \varphi_l(\varphi_j^{r, z'}(e_{y'}^{z't'}))$ and $s = [e_{y'}^{z't'}]$. D4 follows immediately by Commutativity.

If $m'' = m'$, then j' is the source; if the source sends (y', e, s) to l , then $e = \varphi_l^{r, z'}(e_{y'}^{z't'})$ and $s = [e_{y'}^{z't'}]$, proving the base case. Now, suppose that $m'' > m'$ and j' receives (y', e', s) from some agent l' . Let $e = \varphi_{l'}^{r, z'}(e')$. By the hypothesis, $e' = \varphi_j^{r, z'}(\varphi_{l'}^{r, z'}(e_{y'}^{z't'}))$ and $s = [e_{y'}^{z't'}]$. By commutativity, we have

$$e = \varphi_l^{r, z'}(e') = \varphi_l^{r, z'}(\varphi_j^{r, z'}(\varphi_{l'}^{r, z'}(e_{y'}^{z't'}))) = \varphi_l^{r, z'}(\varphi_{l'}^{r, z'}(\varphi_j^{r, z'}(e_{y'}^{z't'}))) = \varphi_j^{r, z'}(e_{y'}^{z't'}).$$

So, if j sends (y', e'', s') to l , then $s' = s = [e_{y'}^{z't'}]$ and $e'' = \varphi_l(e) = \varphi_l^{r, z'}(\varphi_j^{r, z'}(e_{y'}^{z't'}))$, as we intended to prove. This concludes the proof of D4.

D5. Suppose that the actions a_i^* and a_i' are equivalent. Fix an event $e_{y'}^{z't'}$ disseminated before I_i . We can assume that $0 \leq \text{age}(y', m) < n - 1$ and that i did not forward the event, since otherwise D5 would follow immediately otherwise. So, we have $z = z'$ and $t' = t$. The dissemination of the event in rounds $m' \geq m$ depends on the seed sd_j^z and the variables

$vs_j^z(j)$, $ME_j^z(j)$, re_j^z , and se_j^z at round m for all agents j . Given a configuration c of the values of these variables at m , there is a correspondence between runs $r \in \mathcal{R}(I_i)$ of $\vec{\sigma}^*$ and runs $r' \in \mathcal{R}(I_i)$ of $\vec{\sigma}'$ such that $\mu_{\vec{\sigma}^*, I_i}^{fan}(r) = \mu_{\vec{\sigma}', I_i}^{fan}(r')$ and the values of the aforementioned variables are given by c . Fix any such corresponding runs r and r' . Note that only the variables re_j^z and se_j^z may change at or after m . We show using induction on the rounds $m' \geq m$ that an agent j sends a round- m' message containing a tuple with the form (y', e', s') to agent l in run r' for some e' and s' iff j sends a round- m' message to l containing a tuple with the form (y', e, s) to l in run r for some e and s . Since this is true for all corresponding runs r and r' , D5 follows immediately. The base case is clearly true for all $j \neq i$, since j sends the same round- m messages in r and r' , and it is true for $j = i$, since a_i^* and a_i' are equivalent. In the inductive step, by the hypothesis, every agent j receives tuples in round m'' containing y' from the same set agents in r and r' , and re_j^z and se_j^z are updated in the same way in r and r' . Hence, j forwards a tuple containing y' to agents in $SG_j^z(sd_j^z, t, y')$ in round $m' + 1$ in r' iff j does the same in r . This proves D5.

- D6. Suppose that a_i^* is contained in a_i' . Fix an event e_y^{zt} disseminated before I_i . We can assume that i did not output the event nor outputs it in a_i^* . As in D5, we fix two equally likely runs r and r' of $\vec{\sigma}$ after i takes a_i^* and a_i' at I_i , respectively, such that the values of the variables that determine how the event is disseminated are the same. Note that the verdicts $bp^z(l)$ and keys κ_l^z are the same in r and r' in every agent j for all agents l ; we denote by $\varphi_l(e)$ the event e' that corresponds to $(e)_{\kappa_l^z}$ iff $bp^l(l) = True$ or corresponds to e otherwise. Given an agent $j \neq i$, round $m' \geq m$, and run r'' , say that j processes tuple (y, e^*, s) at m' in r'' if j receives a tuple (y, e, s) from l in round m' such that $e^* = \varphi_l(e)$ and j forwards the tuple in round $m' + 1$. We now show using induction that for all rounds $m' \geq m$ and agents $j \neq i$, the following two properties hold (i) if some agent l sends a tuple containing y to j at round m' in r but not in r' , then l sends a tuple to j containing y in r' at some round $m'' < m'$, and (ii) if j processes the tuple $(y, \varphi_i(e^*), s^*)$ at m' in r' and $comp(\varphi_j(e^*), s^*) \neq \perp$, then j also processes the tuple $(y, \varphi_i(e^*), s^*)$ at m' in r . Note that i outputs the tuple $(y, e_y^{zt}, [e_y^{zt}])$ at round m' in r' only after i receives (y, e, s) from j in round m' such that $comp(\varphi_j(e), s) \neq \perp$. In this case, j processes the tuple $(y, \varphi_i(e), s)$ at $m' - 1$ in r' . By (ii) of the hypothesis, j also processes $(y, \varphi_i(e), s)$ at $m' - 1$ in r , so i receives $(y, \varphi_j(e), s)$ from j and outputs $(y, e_y^{zt}, [e_y^{zt}])$ in r as well. Since this is true for all runs r and r' , D6 follows immediately.

First, consider that $m' = m$. Since a_i^* is contained in a'_i , then (i) is clearly true. Now, suppose that j processes $(y, \varphi_i(e^*), s^*)$ in round m' such that $\text{comp}(\varphi_j(e^*), s^*) \neq \perp$. Then, (ii) holds if the agent j only receives the tuple (y, e^*, s^*) from agents $l \neq i$, since every such l sends the same round- m messages in r and r' . So, suppose that j receives (y, e^*, s^*) from i . i can only send such tuple if i received $(y, \varphi_l(e'), s')$ from some l in I_i such that $\text{comp}(e', s') \neq \perp$, but in this case i outputs $(y, e_y^{zt}, [e_y^{zt}])$ in I_i or in a_i^* . Since we have assumed that this is not the case, j cannot receive (y, e^*, s^*) from i , so the hypothesis holds for the base case.

Now, suppose that $m' > m$. To prove (i), suppose that some l sends (y, e, s) to j in r at round m' but does not send a tuple containing y to j in r' at round m' . Then, l must have received a tuple containing y for the first time at round $m' - 1$ from some agent l' in r , whereas l' does not send any such tuple at $m' - 1$ to l in r' . By (i) of the hypothesis, l' must have sent such tuple to l at a round $m'' < m' - 1$ in r' , hence l must have sent a tuple containing y to j at round $m''' \leq m'' + 1$ in r' . This proves (i) for the inductive step. Now, suppose that j processes $(y, \varphi_i(e^*), s^*)$ at round m' such that $\text{comp}(\varphi_j(e^*), s^*) \neq \perp$ in r' . Then, some agent l sends $(y, \varphi_i(e'), s')$ to j , where $e' = \varphi_l(e^*)$, so l processes $(y, \varphi_i(e''), s')$ at $m' - 1$ in r' , where $e'' = \varphi_j(e')$. Note that since j processes the tuples that it receives in round m' by the order of the agents' identifiers, l is the agent with the smallest identifier that sends a tuple containing y in round m' . By commutativity, we have

$$\text{comp}(\varphi_l(e''), s^*) = \text{comp}(\varphi_l(\varphi_j(\varphi_l(e^*))), s^*) = \text{comp}(\varphi_j(e^*), s^*) \neq \perp.$$

Thus, by (ii) of the hypothesis, l processes $(y, \varphi_i(e''), s^*)$ at m' and sends $(y, \varphi_i(e'), s)$ to j at $m' - 1$ in r . Since j receives no tuple containing y prior to round m' , by (i) for the inductive step, the set of agents that send a tuple containing y in r is a subset of the corresponding set in r' , so, in r , l is also the agent with the smallest identifier among those that send a tuple containing y to j in round m' . This implies that j also processes the tuple $(y, \varphi_i(e^*), s^*)$ at round m' in r , which proves (ii). This concludes the proof of D6.

□

We have identified properties of $\vec{\sigma}^{fan}$ when agents perform one-shot deviations. Now, we calculate the difference between the expected utilities of agent i following $\vec{\sigma}^{fan}$ and performing

a one-shot deviation; we denote this different by Δ . The value of Δ is a function of three factors, namely, (1) a long-term factor, which consists in the difference in the expected utilities for all epochs $z' > z + 1$, (2) a medium-term factor, which consists in the difference in the expected utilities for epoch $z + 1$, and (3) a short-term factor, which consists in the difference in the expected utility for epoch z . We now analyse each of these factors in turn.

Given a strategy profile $\vec{\sigma}$ and information set I_i , let $u_i^{z'}(\vec{\sigma} | I_i)$ denote the expected utility of i in epoch z' when agents use $\vec{\sigma}$ after I_i . Lemma 6 shows that the long-term factor is 0.

Lemma 6. *For all agents i , round- m information set I_i for i from stage t from epoch z , epoch $z' \geq z + 1$, and actions $a_i^*, a_i' \in \mathcal{A}_i(I_i)$, we have $u_i^{z'+1}(\vec{\sigma}^{f^{an}} |_{I_i, a_i^*} | I_i) - u_i^{z'+1}(\vec{\sigma}^{f^{an}} |_{I_i, a_i'} | I_i) = 0$.*

Proof. Fix i , I_i , z' , and a_i^*, a_i' . If all agents use $\vec{\sigma}^{f^{an}}$ after I_i , then they send messages according to $\vec{\sigma}^{f^{an}}$ in epochs z' and $z' + 1$, and by M1 of Lemma 4, no agent is punished in epoch $z' + 1$. Consequently, the expected utility of i is constant, regardless of the action of i taken at I_i . To see this, note that: (i) i incurs the same expected costs for sending monitoring messages, since these costs depend only on the probability of the source selecting each subsequence of events; (ii) i incurs the same expected costs for sending dissemination messages, since by D3 from Lemma 5, for all events $e_y^{(z'+1)t'}$ disseminated in epoch $z' + 1$, i receives a tuple (y, e, s) from some agent and forwards it to f^{an} agents with the same probability, whether i follows a_i^* or a_i' ; and (iii) by D4 of Lemma 5, if i receives tuple (y, e, s) in a run r'' , then $e = \varphi_i^{r'', (z'+1)}(e_y^{(z'+1)t'})$ and $s = [e_y^{(z'+1)t'}]$; since i is not punished in epoch $z' + 1$, we have $\varphi_i^{r, z'}(e) = \varphi_i^{r', z'}(e) = e$ for all e and runs $r \in \mathcal{R}(I_i)$ and $r' \in \mathcal{R}(I_i)$ of $\vec{\sigma}^{f^{an}} |_{I_i, a_i^*}$ and $\vec{\sigma}^{f^{an}} |_{I_i, a_i'}$, respectively; hence, i obtains the same expected benefit of outputting each event. This proves the result. \square

Lemma 7 analyses the medium-term factor. Given action $a_i \in \mathcal{A}_i(I_i)$ and epoch z' , let $vs_i^{z'}(i) |_{I_i, a_i}$ and $ME_i^{z'} |_{I_i, a_i}$ denote the values of variables $vs_i^{z'}(i)$ and $ME_i^{z'}$ after i follows a_i at I_i . Note that these values depend only on I_i and on what i does at I_i . Let $\beta_i^{n^{spe}} = q(f^{an})n^{spe}\nu\beta_i$ denote the approximation of the expected benefits of i for receiving events in future epochs: there are $n^{spe}\nu$ events disseminated during an epoch, and i receives each of those events with probability approximately $q(f^{an})$.

Lemma 7. *Fix an agent i , round- m information set I_i from stage t from epoch z , and actions $a_i^*, a_i' \in \mathcal{A}_i$ such that $\sigma_i^{f^{an}}(a_i^* | I_i) > 0$; given epoch z' , let $\Delta^{z'} = u_i^{z'}(\vec{\sigma}^{f^{an}} |_{I_i, a_i^*} | I_i) - u_i^{z'}(\vec{\sigma}^f, a_i' | I_i)$. We have:*

1. If $vs_i^z(i)|_{I_i} = \text{False}$, then $\Delta^{z+1} = 0$.
2. If $vs_i^z(i)|_{I_i} = \text{True}$ and $vs_i^z|_{I_i, a'_i} = \text{False}$, then $\Delta^{z+1} \approx \beta_i^{n^{\text{spe}}} (1 - P^*(\#ME_i^z|_{I_i}))$.
3. If $vs_i^z(i)|_{I_i, a'_i} = \text{True}$, then $\Delta^{z+1} \approx \beta_i^{n^{\text{spe}}} (P^*(\#ME_i^z|_{I_i, a'_i}) - P^*(\#ME_i^z|_{I_i}))$.

Proof. If all agents use $\vec{\sigma}^{fan}$ in epoch $z+1$, then expected costs of i sending monitoring messages is constant, since these depend only on the probability of the source requesting reports relative to each subsequence, and i sends all requested monitoring messages; moreover, by D1 from Lemma 5, i receives and forwards each event to f^{fan} agents with probability approximately equal to $q(f^{fan})$; in addition, by D4 of Lemma 5, if the source does not trigger a punishment of i in epoch $z+1$, then i receives and outputs each event $e_{y'}^{(z+1)t'}$ disseminated in epoch $z+1$ with approximate probability $q(f^{fan})$, otherwise i outputs no event. Since in a_i^* agent i sends only valid messages and forwards every event according to $\vec{\sigma}^{fan}$, the probability of the source triggering a punishment of i in $z+1$ does not increase if i takes a_i^* at I_i , i.e., $vs_i^z(i)|_{I_i, a_i^*} = vs_i^z(i)|_{I_i}$ and $ME_i^z(i)|_{I_i, a_i^*} = ME_i^z(i)|_{I_i}$; if i takes a'_i at I_i , then the probability of the source triggering a punishment may only increase, i.e., $ME_i^z(i)|_{I_i} \subseteq ME_i^z(i)|_{I_i, a'_i}$ and we never have $vs_i^z(i)|_{I_i, a'_i} = \text{True}$ and $vs_i^z(i)|_{I_i} = \text{False}$. Therefore, we have $\Delta^{z'} = (p_{a'_i} - p_{I_i})\beta_i^{n^{\text{spe}}}$, where $p_{a'_i}$ and p_{I_i} are the probabilities of the source triggering a punishment of i in epoch $z+1$ after i takes a'_i and a_i^* at I_i , respectively. The result follows immediately, since we have:

1. If $vs_i^z(i)|_{I_i} = \text{False}$, then by M1 from Lemma 4 $p_{a'_i} = p_{I_i} = 1$.
2. If $vs_i^z(i)|_{I_i} = \text{True}$ and $vs_i^z|_{I_i, a'_i} = \text{False}$, then by M1 and M2 from Lemma 4 $p_{a'_i} = 1$ and $p_{I_i} = P^*(\#ME_i^z|_{I_i})$.
3. If $vs_i^z(i)|_{I_i, a'_i} = \text{True}$, then by M2 from Lemma 4 $p_{a'_i} = P^*(\#ME_i^z|_{I_i, a'_i})$ and $p_{I_i} = P^*(\#ME_i^z|_{I_i})$.

This concludes the proof. □

Finally, Lemma 8 analyses the short-term factor. Let $\gamma_i^{n^{\text{spe}}}$ denote the expected costs of i for sending events during an epoch: i receives and forwards each of the νn^{spe} events to f^{fan} agents with probability approximately $q(f^{fan})$, so, $\gamma_i^{n^{\text{spe}}} = q(f^{fan})\nu n^{\text{spe}} f^{fan}$. Let $\rho_i^{n^{\text{spe}}}$ denote the expected cost of sending reports in monitoring messages: i sends reports relative to an expected number of $p^{\text{seq}}\nu n^{\text{spe}}$ events, and each report contains $\log(n)$ bits, so $\rho_i^{n^{\text{spe}}} = p^{\text{seq}}\nu n^{\text{spe}} n \log(n)\alpha_i$.

Finally, let ρ_i denote an upper bound on the cost of sending accusations and sending any given monitoring message: i sends n accusations, where each accusation has a length of one bit, and i sends $\nu(n^{\text{spe}}/n^{\text{seq}})n$ reports in a single monitoring message, where each report has a length of $\log(n)$ bits, so $\rho_i = n\alpha_i + \nu(n^{\text{spe}}/n^{\text{seq}})n \log(n)\alpha_i$.

Lemma 8. *Fix an agent i , round- m information set I_i for i from stage t from epoch z , and actions $a_i^*, a_i' \in \mathcal{A}_i(I_i)$ such that $\sigma_i^{\text{fan}}(a_i^* | I_i) > 0$; let $\Delta^z = u_i^z(\vec{\sigma}^{\text{fan}} |_{I_i, a_i^*} | I_i) - u_i^z(\vec{\sigma}^{\text{fan}} |_{I_i, a_i'} | I_i)$. We have:*

1. *If $vs_i^z(i) |_{I_i} = \text{False}$, then $\Delta^z \geq 0$.*
2. *If $vs_i^z(i) |_{I_i} = \text{True}$ and $vs_i^z(i) |_{I_i, a_i'} = \text{False}$, then the value of Δ^z depends on whether m is monitoring round:*

- *If m is a monitoring round (i.e., $t = 1$ and $m \leq 2n^{\text{seq}} + 2$), then*

$$\Delta^z \geq -(\gamma_i^{n^{\text{spe}}} + \rho_i^{n^{\text{spe}}} + \rho_i).$$

- *If m is not a monitoring round (i.e., $t > 1$ or $m > 2n^{\text{seq}} + 2$), then*

$$\Delta^z \geq -(\gamma_i^{n^{\text{spe}}} + n(f^{\text{fan}} + \beta_i)).$$

3. *If $vs_i^z(i) |_{I_i, a_i'} = \text{True}$, then let $x = \#ME_i^z |_{I_i}$ and $x' = \#ME_i^z |_{I_i, a_i'}$:*

- *If $x' = x$, then $\Delta \geq 0$.*
- *If $x' > x$, then*

$$\Delta^z \geq -n(f^{\text{fan}} + \beta_i) - (x' - x) \frac{n^{\text{spe}} \nu}{n^{\text{seq}}} f^{\text{fan}}.$$

Proof. By D3 of Lemma 5, for all events $e_y^{zt'}$ disseminated after I_i in epoch z , i receives a tuple with the form (y, e, s) with the same probability whether i takes a_i^* or a_i' at I_i , and by D4 of the same lemma, e is $\varphi_i(e_y^{zt'})$ and $s = [e_y^{zt'}]$. Since $\varphi_i^{r,z}(e_y^{zt'})$ is the same event for all runs $r \in \mathcal{R}(I_i)$, i outputs $e_y^{zt'}$ with the same probability whether i takes a_i^* or a_i' at I_i . This implies that the expected benefits of receiving events in epoch z disseminated after I_i are the same whether i takes a_i^* or a_i' at I_i .

Regarding events output by i in round m , it is true that i does not gain from outputting a tuple (y, e, s) in a_i' that i does not output in a_i^* . To see this, note that by the definition of

μ^{fan} , if i outputs a tuple (y, e, s) in a'_i , then i gains a benefit β_i only if s is a valid signature of e . This is true only if $e = e_y^{zt}$ and i receives (y, e', s) from l in I_i such that $comp(\varphi_l(e'), s) = e$, but in this case i outputs (y, e, s) in I_i or in a_i^* . It is also easy to see that if i outputs an event in a_i^* , then i does not gain by not outputting the event in a'_i , since i outputs a tuple in a_i^* iff i gains the benefit β_i by doing so. Therefore, i never gains by not outputting in a'_i the exact same events that i outputs in a_i^* .

Consequently, Δ^z is a function of two factors:

1. i may avoid the cost of sending some messages in epoch z by taking a'_i at I_i . Specifically, i may avoid the cost of sending some of the messages sent in a_i^* . In addition, i may avoid the cost of sending some messages that i sends in later rounds when i takes a_i^* at I_i : if i omits a monitoring message or sends an invalid message and $vs_i^z(i)|_{I_i} = True$, then i sends no additional messages; if i does not send invalid messages but omits some events from subsequence s and $vs_i^z(i)|_{I_i} = True$, then i does not forward any more events from s .
2. i may increase the benefit of receiving events disseminated before I_i . In particular, if i sends tuples in a_i^* that contain garbage, then by not sending those tuples i may increase the chances of later receiving the corresponding events disseminated by the source in plain.

Since i sends messages according to $\vec{\sigma}^{fan}$ in a_i^* , we have $vs_i^z(i)|_{I_i, a_i^*} = vs_i^z|_{I_i}$ and $ME_i^z|_{I_i} = ME_i^z|_{I_i, a_i^*} \subseteq ME_i^z|_{I_i, a'_i}$, and we never have $vs_i^z(i)|_{I_i} = False$ while $vs_i^z(i)|_{I_i, a'_i} = True$. So, Δ^z is a function of the variables $vs_i^z(i)|_{I_i}$, $vs_i^z(i)|_{I_i, a'_i}$, $ME_i^z|_{I_i}$, and $ME_i^z(i)|_{I_i}$. We now calculate Δ^z for all possible values of these variables in turn:

1. If $vs_i^z(i)|_{I_i} = False$, then i sends no messages in a_i^* nor after taking a_i^* , so i avoids no costs by taking a'_i . Since i sends no events in a_i^* , a_i^* is contained in a'_i , so by D6 from Lemma 5, i also does not output events already disseminated with higher probability by taking a'_i instead of a_i^* at I_i . Therefore, $\Delta^z \geq 0$.
2. If $vs_i^z(i)|_{I_i} = True$ and $vs_i^z(i)|_{I_i, a'_i} = False$, then the difference in the utility depends on whether m is a monitoring or dissemination round:
 - If m is a monitoring round ($t = 1$ and $m \leq 2n^{seq} + 2$), then i avoids at most the cost ρ_i of sending a monitoring message to the source in a_i^* and accusations in round

$2n^{\text{seq}} + 2$, avoids the maximum cost $\rho_i^{n^{\text{spe}}}$ of sending reports in monitoring rounds $m' > m$, and the maximum cost $\gamma_i^{n^{\text{spe}}}$ of forwarding events in all stages from epoch z (note that by D2 of Lemma 5 i receives and forwards each event with approximate probability $q(f^{\text{fan}})$), so we have

$$\Delta^z \geq -(\gamma_i^{n^{\text{spe}}} + \rho_i^{n^{\text{spe}}} + \rho_i).$$

- If m is not a dissemination round ($t > 1$ or $m > 2n^{\text{seq}} + 2$), then again i avoids the cost $c \leq \gamma_i^{n^{\text{spe}}}$ of sending events in all rounds that follow m in epoch z , avoids the cost $c' \leq n f^{\text{fan}}$ of forwarding n events disseminated before I_i (recall that there are at most n events being disseminated at every point in time), and receives each of the n events disseminated before I_i still being disseminated with a probability that increases by $p \leq 1$, so we have

$$\Delta^z \geq -(\gamma_i^{n^{\text{spe}}} + n(f^{\text{fan}} + \beta_i)).$$

3. If $vs_i^z(i)|_{I_i, a'_i} = \text{True}$, then let $x' = ME_i^z|_{I_i, a'_i}$ and $x = ME_i^z|_{I_i}$:

- If $x' = x$, then a'_i and a_i^* are equivalent, so by D5 from Lemma 5, i avoids no costs of sending events disseminated before I_i . In addition, a_i^* is contained in a'_i , so by D6 from Lemma 5 the probability of i outputting events disseminated before I_i does not increase. Therefore, $\Delta \geq 0$.
- If $x' > x$, then again i gains at most $(\beta_i + f^{\text{fan}})n$, and avoids at most the cost $(x' - x)q(f^{\text{fan}})f^{\text{fan}}\nu n^{\text{spe}}/n^{\text{seq}}$ of not forwarding $\nu n^{\text{spe}}/n^{\text{seq}}$ events from $x' - x$ subsequences of events disseminated after I_i , so we have

$$\Delta^z \geq -(x' - x)\frac{n^{\text{spe}}\nu}{n^{\text{seq}}}f^{\text{fan}} - n(f^{\text{fan}} + \beta_i).$$

This proves the result. □

We can now prove Theorem 9, which shows that $\vec{\sigma}^{f^{\text{fan}}}$ is a sequential equilibrium if agents are sufficiently patient and f^{fan} is c -individually rational for some constant c .

Theorem 9. *There is a constant $c \geq 1$ such that, for all c -individually rational fanouts f^{fan} , there exists $\delta^* \in (0, 1)$ such that for all $\delta \in (\delta^*, 1)$, $\vec{\sigma}^{f^{\text{fan}}}$ is a sequential equilibrium.*

Proof. Fix an individually rational f^{fan} , agent i , round- m information set I_i for i from stage t from epoch z , and actions $a_i^*, a_i' \in \mathcal{A}_i(I_i)$ such that $\sigma_i^{f^{fan}}(a_i^* | I_i) > 0$. Let $\Delta = u_i(\vec{\sigma}^{f^{fan}} |_{I_i, a_i^*} | I_i) - u_i(\vec{\sigma}^{f^{fan}} |_{I_i, a_i'} | I_i)$. We show that there is a constant $c > 0$ such that if f^{fan} is c -individually rational and agents are sufficiently patient, then $\Delta \geq 0$. The result follows immediately by the one-shot deviation property.

By Lemmas 6, 7, and 8, the value of Δ is a function of the differences of the expected utility in epochs z and $z + 1$; let Δ^z and Δ^{z+1} denote these differences. As Lemmas 7 and 8 show, the factors Δ^z and Δ^{z+1} depend only on the values of the variables $vs_i^z(i)$ and ME_i^z at I_i and after i takes a_i' . We now show that $\Delta \geq 0$ for all possible values of these variables if δ is sufficiently close to 1, n^{spe} is sufficiently large, and β_i is sufficiently larger than f^{fan} . By Lemmas 7 and 8, we have:

1. If $vs_i^z(i)|_{I_i} = \text{False}$, then $\Delta \geq 0$.
2. If $vs_i^z(i)|_{I_i} = \text{True}$ and $vs_i^z(i)|_{I_i, a_i'} = \text{False}$, then we need to consider the cases where m is a monitoring round and when it is not separately:
 - If m is a monitoring round ($t = 1$ and $m \leq 2n^{\text{seq}} + 2$), then there is a value v such that we have:

$$\Delta \approx v \geq (\delta^{n^{\text{spe}}} \beta_i^{n^{\text{spe}}} - \gamma_i^{n^{\text{spe}}}) - (\rho_i^{n^{\text{spe}}} + \rho_i).$$

Let $\epsilon = \beta_i - f^{fan}$; since $\beta_i > f^{fan}$, we have $\epsilon > 0$. Let $\epsilon(\delta) = \delta^{n^{\text{spe}}} \beta_i^{n^{\text{spe}}} - \gamma_i^{n^{\text{spe}}}$. We have

$$\delta^{n^{\text{spe}}} \beta_i^{n^{\text{spe}}} = \delta^{n^{\text{spe}}} \frac{1 - \delta^{n^{\text{spe}}}}{1 - \delta} \nu q(f^{fan}) \beta_i.$$

As δ approaches 1, $\delta^{n^{\text{spe}}} \beta_i^{n^{\text{spe}}}$ converges to $n^{\text{spe}} \nu q(f^{fan}) \beta_i$; conversely, $\gamma_i^{n^{\text{spe}}}$ converges to $n^{\text{spe}} \nu q(f^{fan}) f^{fan}$; consequently, we have

$$\lim_{\delta \rightarrow 1} \epsilon(\delta) = n^{\text{spe}} q(f^{fan}) \nu \epsilon > 0. \quad (5.2)$$

Moreover, we have

$$\rho_i^{n^{\text{spe}}} + \rho_i = p^{\text{seq}} n^{\text{spe}} \nu n \log(n) \alpha_i + \frac{n^{\text{spe}} \nu}{n^{\text{seq}}} n \log(n) \alpha_i + n \alpha_i. \quad (5.3)$$

In (5.3), the only factors that depend on n^{spe} are $p^{\text{seq}} n^{\text{spe}}$ and $n^{\text{spe}} \nu / n^{\text{seq}}$: by C2,

$p^{\text{seq}}n^{\text{spe}} = o(n^{\text{spe}})$ (i.e., grows slower than n^{spe} as n^{spe} increases); by C1 and C3, $n^{\text{seq}} = \omega(1)$, so $n^{\text{spe}}/n^{\text{seq}} = o(n^{\text{spe}})$; hence, (5.3) is a sublinear function on n^{spe} . By (5.2) and (5.3, we have)

$$\lim_{\delta \rightarrow 1, n^{\text{spe}} \rightarrow \infty} \Delta \geq \lim_{\delta \rightarrow 1, n^{\text{spe}} \rightarrow \infty} (\epsilon(\delta) - (\rho_i^{n^{\text{spe}}} + \rho_i)) = \lim_{n^{\text{spe}} \rightarrow \infty} (\theta(n^{\text{spe}}) - o(n^{\text{spe}})) = \infty.$$

Therefore, for values of δ sufficiently close to 1 and n^{spe} sufficiently large, we have $\Delta \geq 0$.

- If m is not a monitoring round ($t > 1$ or $m > 2n^{\text{seq}} + 2$), then we have:

$$\Delta \approx v = \delta^{n^{\text{spe}}} \beta_i^{n^{\text{spe}}} (1 - p) - \gamma_i^{n^{\text{spe}}} - n(f^{fan} + \beta_i), \quad (5.4)$$

where $p = P^*(\#ME_i^z | I_i) \leq 1 - (1 - p^{\text{seq}})^{n^{\text{seq}}}$. As in the previous case, as δ approaches 1, (5.4) converges to

$$n^{\text{spe}} \nu q(f^{fan})(\beta_i(1 - p) - f^{fan}) - n(f^{fan} + \beta_i). \quad (5.5)$$

By C1-C3, as n^{spe} grows, $1 - p \geq (1 - p^{\text{seq}})^{n^{\text{seq}}}$ converges to some constant c . To see this, note that $p^{\text{seq}} = c'/g(n^{\text{spe}})$ and $n^{\text{seq}} = g(n^{\text{spe}})$ for some constant c' and function $g(n^{\text{spe}})$ that grows with n^{spe} , so

$$\lim_{n^{\text{spe}} \rightarrow \infty} (1 - p^{\text{seq}})^{n^{\text{seq}}} = \lim_{n^{\text{spe}} \rightarrow \infty} (1 - c'/g(n^{\text{spe}}))^{g(n^{\text{spe}})} = e^{-c'}.$$

If $\beta_i > e^{c'} f^{fan}$, then (5.5) goes to ∞ as n^{spe} increases. This implies that, for a sufficiently large n^{spe} and a δ sufficiently close to 1, we have $\Delta \geq 0$.

3. Let $x = \#ME_i^z | I_i$, $x' = \#ME_i^z | I_i, a'_i$, and $d = x' - x$. If $vs_i^z(i) | I_i, a'_i = \text{True}$ and $d = 0$, then $\Delta \geq 0$. If $vs_i^z(i) | I_i, a'_i = \text{True}$ and $d > 0$, then we have

$$\Delta \approx v \geq \delta^{n^{\text{spe}}} \beta_i^{n^{\text{spe}}} (p' - p) - dq(f^{fan})(\nu n^{\text{spe}}/n^{\text{seq}}) f^{fan} - n(f^{fan} + \beta_i),$$

where $p = 1 - (1 - p^{\text{seq}})^x$ and $p' = 1 - (1 - p^{\text{seq}})^{x'}$. Let $L = \lim_{\delta \rightarrow 1} \Delta$. Using the same reasoning as before, we have:

$$L \geq n^{\text{spe}} \nu q(f^{fan}) \beta_i (p' - p) - dq(f^{fan})(\nu n^{\text{spe}}/n^{\text{seq}}) f^{fan} - n(f^{fan} + \beta_i).$$

Let $c = \beta_i / f^{fan}$ and let

$$L' = \frac{L}{q(f^{fan})n^{spe}\nu\beta_i(p' - p)} = 1 - \frac{d}{cn^{seq}(p' - p)} - \frac{n(c + 1)}{cq(f^{fan})\nu n^{spe}(p' - p)}.$$

Recall that d is the number of different subsequences containing events omitted in a'_i and events not yet disseminated. Note also that by C1 and C3 we have $n^{seq} = o(n^{spe})$, so the number of events per subsequence $n^{spe}\nu/n^{seq}$ grows with n^{spe} . In particular, if n^{spe} is sufficiently large, then this number is larger than n . Suppose that this is the case. Since i can omit at most n events in a'_i , i omits events from at most one subsequence that contains events not yet disseminated, so $d = 1$. Therefore, we have

$$p' - p = (1 - p^{seq})^x - (1 - p^{seq})^{x'} = (1 - p^{seq})^x(1 - (1 - p^{seq})^d) \geq (1 - p^{seq})^{n^{seq}} p^{seq},$$

and

$$L' \geq 1 - \frac{1}{cn^{seq}(1 - p^{seq})^{n^{seq}} p^{seq}} - \frac{n(c + 1)}{c\nu n^{spe}(1 - p^{seq})^{n^{seq}} p^{seq}}.$$

By C1-C3, there exists a value c'' that is constant on n^{spe} such that $cn^{seq}(1 - p^{seq})^{n^{seq}} p^{seq}$ converges to c'' as n^{spe} increases (again, by C3, $(1 - p^{seq})^{n^{seq}}$ converges to a constant and $n^{seq} p^{seq}$ is constant by definition), whereas $c\nu n^{spe}(1 - p^{seq})^{n^{seq}} p^{seq}$ goes to ∞ (by C1, $n^{spe} p^{seq}$ goes to infinity). Consequently, we have

$$\lim_{n^{spe} \rightarrow \infty} L' = 1 - \frac{1}{cc''}.$$

So, if $c > 1/c''$, then the limit is positive, which implies that $\lim_{n^{spe} \rightarrow \infty} L = \infty$. Therefore, there exists a constant c such that, if $\beta_i > cf^{fan}$, then, for a sufficiently large n^{spe} and δ sufficiently close to 1, $\Delta \geq 0$. In particular, if $n^{seq} = \sqrt{n^{spe}}$ and $p^{seq} = 1/\sqrt{n^{spe}}$, then the above holds for all $c \geq e$. This concludes the proof.

□

5.2.3 Average Utility

We now prove Theorem 10, which shows that the average utility of agent i when agents use $\vec{\sigma}^{fan}$ can be arbitrarily close to $x_i(f^{fan})$ for sufficiently patient agents.

Theorem 10. *For all arbitrarily small values $\epsilon > 0$, there exists $\delta^* \in (0, 1)$ such that for all $\delta \in (\delta^*, 1)$ and agents i , we have $|\bar{u}_i(\vec{\sigma}^{fan}) - x_i(f^{fan})| < \epsilon$.*

Proof. Fix agent i . The average utility of i is the difference between the average benefits and costs per stage of receiving and forward events and the average costs per stage of sending monitoring messages. If all agents use $\vec{\sigma}^{fan}$, then by D1 from Lemma 5 i receives tuples corresponding to each disseminated event with approximate probability $q(f^{fan})$ and forwards it to f^{fan} agents. Moreover, since by M1 and M2 from Lemma 4 i is never punished, by D4 from Lemma 5 every tuple that i receives contains the corresponding event disseminated by the source in plain. Hence, the expected utility of i regarding dissemination only is arbitrarily close to $x_i(f^{fan})$. In addition to this, once every epoch, i send messages containing reports and accusations. Specifically, for each $1 \leq s \leq n^{seq}$, i sends a message containing $nn^{spe}\nu/n^{seq}$ reports with independent probability p^{seq} (i.e., one report per agent and event in s); each report consists in $\log(n)$ bits that represent a round number $1 \leq m \leq n$. Thus, the expected cost per epoch of sending reports is $\alpha_i n^{spe} \nu p^{seq} n \log(n)$. In addition, i sends one message containing n accusations. Hence, the total cost per epoch is $\alpha^* = \alpha_i(n + nn^{spe} p^{seq} \nu \log(n))$. The average cost per stage is $\alpha^* + \delta^{n^{spe}} \alpha^* + \delta^{2n^{spe}} \alpha^* + \dots$, i.e., it is given by the sum

$$\sum_{t=0}^{\infty} \delta^{tn^{spe}} \alpha^* = \alpha^* \frac{1 - \delta}{1 - \delta^{n^{spe}}}.$$

As δ approaches 1, this sum converges to

$$\frac{\alpha^*}{n^{spe}} = \alpha_i \left(\frac{n}{n^{spe}} + n \log(n) \nu p^{seq} \right). \quad (5.6)$$

By C2, (5.6) converges to 0 as n^{spe} increases. Therefore, for all arbitrarily small constants $\epsilon > 0$, if δ is sufficiently close to 1 and n^{spe} is sufficiently large, then $|\bar{u}_i(\vec{\sigma}^{fan}) - x_i(f^{fan})| < \epsilon$. This proves the result. \square

5.3 Fully Distributed Protocol

In streaming services executed among multiple administrative domains such as peer-to-peer networks, there is usually a user that volunteers to serve as a source of the stream, so unless the volunteer colludes with some other user it is safe to assume that the source is trusted (Li et al.

2006; Li et al. 2008). However, in some applications users may only be willing to serve as the source of the stream if they expect to receive another stream in return, such that the source cannot be trusted. Moreover, the source is a single point of failure regarding its role as the mediator. We now discuss how we can extend our protocol to distribute the role of the source and the mediator in a way that the main result of this chapter still holds in a fully distributed setting where the source is also a rational agent.

To obtain a folk theorem in a fully distributed setting, we need to address three additional challenges, namely, (1) the source gains by not sending the stream, (2) the role of the mediator cannot be performed by a single agent, and (3) the mediator may not perform its role correctly.

We can address challenge (1) if agents cooperate to disseminate two or more streams, say s_0 and s_1 , with different sources. The idea is that, for $i \in \{0, 1\}$, agents that are not the source of stream s_i report to the source of s_{1-i} about the events that the source of s_i sends in a similar fashion to $\vec{\sigma}^{fan}$ such that if the source of s_{i-1} detects a deviation in epoch z , then the source of s_i is punished in epoch $z + 1$ by not receiving the stream s_{i-1} .

Regarding challenge (2), we do not need the source to perform the role of the mediator. In fact, we can distribute the role of the mediator in a similar fashion to LiFTing (Guerraoui et al. 2010): we assign to each agent i a set S_i of agents that must perform the role of mediator of i ; this set may contain any agent other than i . At the beginning of every epoch, agents in S_i execute the same protocol used to collect accusations against i and information about the events that i sent and received in the previous epoch, with the following difference: in the centralized protocol, the mediator has to decide whether to collect information about the events from each subsequence with independent probability; in the distributed version, agents in S_i have to execute some distributed coin mechanism such as the one proposed in (Abraham et al. 2013) to correlate their decisions. That is, for each subsequence, agents in S_i execute the coin mechanism to decide with probability p^{seq} whether to request reports regarding events from the subsequence; if so, then they send a request to all agents other than i , which then broadcast the corresponding reports to all agents in S_i . (The fact that agents only send a request with probability p^{seq} ensures that the average cost of sending those requests is also sublinear on n^{spe} .) In the next step, all agents other than i broadcast their accusations (if they have any) to the agents in S_i . At the end of the monitoring rounds, the agents in S_i use a distributed coin to generate a unique key κ_i , which they send to all agents other than i . To generate the seed sd_i ,

agents have to use a different approach, for our mechanism cannot allow the agents in S_i to learn the seed; instead, each agent generates a separate seed and sends it only to i ; i then computes a final seed sd_i from the received seeds (e.g., using the same modulo technique proposed in the problem of fair consensus); at the beginning of the next epoch, all agents in S_i reveal their seeds to each other, allowing them to learn the final seed that i was supposed to use. To dissuade agents in S_i from not performing the role of the mediator correctly, we can punish agents in S_i for omitting messages or for disagreeing about the verdict or key of i .

Regarding challenge (3), although the solution to challenge (2) ensures that agents from the set S_i of mediators of any given agent i do not gain from not performing the role of the mediator correctly, we cannot assume that there are never disagreements among the agents in S_i , which is a problem to proving that the protocol is a sequential equilibrium. The solution is for agents to reach a consensus on the data that the agents in S_i send regarding the key and verdict of i . Specifically, agents broadcast the information they receive during n rounds. If an agent j omits a message, sends an invalid message, or sends two different messages to two different agents, then j is marked as "crashed" by the other agents and is punished in the next epoch (agents can exchange additional monitoring information for this purpose); once j is marked as crashed, it sends no further messages until the end of the epoch. By the end of the n rounds, either (i) all agents have been marked as crashed or (ii) all agents that have not been marked as crashed agree on the information that the mediators sent to them or if the mediators did not send the same information to all those agents, then they agree on some pre-defined values. This guarantees that all the agents that disseminate events during every epoch use the same keys and verdicts. Given this guarantee, the proof that $\vec{\sigma}^{fan}$ is a sequential equilibrium follows without change.

Summary

In this section, we have proved a slightly weaker version of an approximate Folk Theorem for eager-push gossip dissemination protocols. The most important consequence of this result is that we can sustain cooperation with protocols that use almost any fanout f^{fan} , while minimizing the overhead of the incentives provided for rational agents to follow the protocol. In the proof of this result, we defined a protocol that disseminates events using any fanout f^{fan} , where the source monitors the behaviour of agents and triggers punishments after agents deviate. We proved that agents did not gain from deviating provided that they could not break the cryptographic

primitives used in the protocol, and we showed that the average cost of monitoring per stage can be arbitrarily minimized. We have also discussed how to distribute the role of the source.

We believe that the protocol defined in this chapter also sustains cooperation if agents may crash or communication channels are unreliable, provided that the probabilities of a single agent crashing and messages being lost are sufficiently small. Moreover, the protocol is almost robust to rational behaviour if the system is partially asynchronous, in the sense that the time it takes for messages sent in a stage t to be delivered is arbitrarily large, but messages sent in t are still received in that stage, i.e., there is asynchrony within a stage and synchrony between stages. In this setting, agents still do not gain by not forwarding events correctly and not sending monitoring information. However, an agent can gain by delaying the dissemination of an event, and no agent can detect and punish such deviation. This issue could be addressed if we could assume that agents communicate through FIFO channels and that every agent fears that, by delaying a message in stage t , this message is not be delivered in t with high probability, in which case delaying a message is equivalent to an omission.

For future work, it would be interesting to actually prove an approximate theorem for computationally unbounded agents, so that we would not have to assume a sufficiently high benefit/cost ratio and we would not have to use cryptography to detect and punish deviations. However, we stress that we cannot prove an exact Folk theorem without the support of an exogenous monitoring infrastructure, since agents must share their private observations; the communication costs that agents incur for sharing private information always introduce a non-negligible overhead.

In the next chapter, we discuss our results related to pairwise exchanges in dynamic networks.

Type	Notation	Description
Agents	\mathcal{N}	Set of agents.
	n	Number of agents.
Actions and histories	a_i^m	Round- m action of agent i .
	h	Global history.
	I_i	Information set for agent i .
	$\mathcal{R}(I_i)$	Set of runs compatible with information set I_i .
	$\mathcal{R}(h)$	Set of runs compatible with global history h .
	$\mathcal{A}_i(I_i)$	Set of actions available to i at I_i .
Time	τ	Total number of rounds per stage.
	τ^D	Number of dissemination rounds per stage.
	τ^M	Number of monitoring rounds per stage.
Utilities	δ	Discount factor.
	μ	Belief system.
	β_i	Benefit that i obtains when it outputs an event.
	α_i	Cost incurred by i per bit sent in a message.
	$u_i^z(\vec{\sigma} I_i)$	Expected utility of i in epoch z when agents use $\vec{\sigma}$ conditioned on I_i .
	$u_i(\vec{\sigma} I_i)$	Expected utility of i when agents use $\vec{\sigma}$ conditioned on I_i .
	$u_i(\vec{\sigma})$	Expected utility of i when agents use $\vec{\sigma}$.
	$x_i(f^{fan})$	Expected utility of i in one stage of eager-push gossip dissemination.
Events	e_y^{zt}	y^{th} event from stage t from epoch z .
	ν	Number of events per stage.
	\mathcal{E}^{zt}	Set of events from stage t from epoch z .
	$\mathcal{E}(z, s)$	Set of events from s^{th} subsequence from epoch z .
	$sub(t, y)$	Sequence number of y^{th} event from stage t .
	Cryptography	$[e]$
$SG_i(sd_i^z, t, y)$		Pseudo-random set of f^{fan} agents for y^{th} event from t seeded by sd_i^z .
$(e)_\kappa$		Cipher of event e with key κ .
Algorithm	f^{fan}	Fanout.
	re_i^z, se_i^z	Received and sent events in epoch z .
	ME_i^z	Subsequences containing incorrectly forwarded events in epoch z .
	Out_i^z	Events already output in epoch z .
	EF_i^z	Events to be forwarded in each round from epoch z .
	vs_i^z	Validity of sent and received messages.
	$age(t, y, m)$	Age of y^{th} event from stage t at round m .
	p^{seq}	Request probability of each subsequence.
	n^{spe}	Number of stages per epoch.
	$comp(e, s)$	Function that retrieves disseminated event from e and s .
	$\varphi_i(e)$	Ciphers/deciphers e with κ_i iff i is being punished.
	bp_i^z	Verdicts for epoch z .
EO_i^z	Events to be output at each round.	
Strategies	σ_i	Strategy for agent i .
	$\vec{\sigma}$	Strategy profile.
	$\vec{\sigma}^{fan}$	Gossip dissemination protocol with fanout f^{fan} .
	μ^{fan}	Belief system consistent with $\vec{\sigma}^{fan}$.
	$\vec{\sigma} _{I_i, a_i}$	One-shot deviation where i takes a_i at I_i .

Table 5.1: Notation - gossip dissemination.

Pairwise Exchanges in Dynamic Networks

In this chapter, we address the problem of sustaining cooperation in infinitely repeated pairwise exchanges over links of a dynamic network. Our goal is to identify necessary and sufficient restrictions on the set \mathcal{G}^* of evolving graphs that the adversary may generate. We focus on protocols that satisfy three properties, namely, (1) the protocols are \mathcal{G}^* -OAPE, (2) if no agent deviates, then agents always exchange their values in every interaction, and (3) protocols are bounded, i.e., they are self-stabilizing in bounded time (Dolev 2000). A protocol that satisfies the first two properties is said to *sustain cooperation* in pairwise exchanges. The third property is an additional requirement that is important both in theory and in practice, since bounded protocols can recover from transient failures and require bounded memory. Bounded memory is a crucial property in dynamic networks where agents have a limited memory capacity, which is often the case of mobile networks and wireless ad-hoc networks. Our results show that the existence of such protocols depends not only on the restrictions on \mathcal{G}^* but also on the structure and utility of pairwise exchanges and the type of incentives used by the protocols.

We start by determining the weakest restrictions on \mathcal{G}^* necessary to sustain cooperation. Specifically, we show that \mathcal{G}^* must admit *weak timely punishments*, even if the protocols are not bounded. Intuitively, this means that for every pairwise exchange between agents i and j , at least one of the two agents $i' \in \{i, j\}$ is able to communicate a deviation of the other agent j' to some third agent l that is capable of punishing j' in a later stage. We show that a consequence of this result is that \mathcal{G}^* must admit *strong timely punishments* to sustain cooperation in one-shot pairwise exchanges, which means that in every interaction between i and j in stage t , both i and j must be capable of communicating deviations in stage t . These restrictions are not met by some networks such as file-sharing overlays (e.g., Bittorrent (Cohen 2003)), where users with similar interests interact frequently with each other but only rarely with users with different interests; in these cases, an exchange between agents with different interests may not admit a timely punishment.

Our next result provides a bounded protocol $\vec{\sigma}^{\text{val}}$ that sustains cooperation in pairwise exchanges, assuming that \mathcal{G}^* admits strong timely punishments. We also assume that each stage contains at least three rounds and pairwise exchanges are *valuable* in the sense that agents have a high benefit/cost ratio of receiving/sending messages and neglect download costs. Valuable exchanges occur, for instance, when agents share small but highly valuable secrets such as private keys (Halpern & Teague 2004; Abraham et al. 2006) and the bandwidth is asymmetric.

In many cases, the assumption that pairwise exchanges are valuable is too restrictive. For instance, in file-sharing, we may expect agents to be interested in exchanging large files, but the cost of downloading such files is certainly non-negligible, and the benefit/cost ratio of receiving/sending files may be small. Moreover, if the network is too dynamic, then agents may not be able to exchange more than one message in each interaction. Our next results identify necessary and sufficient restrictions on \mathcal{G}^* to sustain cooperation with bounded protocols in general one-shot pairwise exchanges.

We first identify problems with protocols that use certain types of punishments as an incentive to sustain cooperation. Specifically, in a one-shot pairwise exchange between agents i and j , agent i can punish j in three different ways: (1) i may not send its value to j , this way denying the benefit β_j to j , (2) i may require j to send larger messages, this way forcing j to incur an additional cost for sending messages, and (3) i may send larger messages to j , such that j incurs a larger cost for receiving messages. We show that if agents omit messages as punishments of type (1) or uses punishments of types (2) or (3), then the protocol is not an equilibrium if \mathcal{G}^* is only restricted by weak timely punishments. This shows that, to sustain cooperation in general one-shot pairwise exchanges, either we must place additional restrictions on \mathcal{G}^* or we must not use any of the above types of punishments. Indeed, Li et al. (2006, 2008) propose protocols that do not use the above punishments and are effective at sustaining cooperation. In these protocols, agents always exchange messages of fixed size, and they punish their neighbours by sending garbage data instead of their value; we call such protocols *symmetric*. Symmetric protocols not only avoid the problems with other types of punishments but also are simpler to develop and analyse. Our two final results identify a necessary and a sufficient restriction on \mathcal{G}^* to sustain cooperation with symmetric and bounded protocols.

We show that if there is a symmetric and bounded protocol that sustains cooperation in general one-shot pairwise exchanges, then \mathcal{G}^* must be restricted, in addition to strong timely



Figure 6.1: Results for Valuable Pairwise Exchanges (Section 6.2).

punishments, by a restriction that we call *eventual distinguishability*. Roughly speaking, this restriction states that whenever two (or more) agents may punish i for omitting messages in the past (towards some other agent j), they have the information necessary to coordinate their actions, so that the total number of additional punishments for a single deviation of i is neither too large nor too low.

Whether \mathcal{G}^* is restricted by eventual distinguishability depends both on the properties of graphs in \mathcal{G}^* and on the information available to agents about the topology. Our last result shows that if agents can learn the degree of their neighbours and all evolving graphs in \mathcal{G}^* satisfy a connectivity property similar to the property defined by Kuhn and Oshman (2010), then there is a symmetric and bounded protocol $\vec{\sigma}^{\text{gen}}$ that sustains cooperation in general one-shot pairwise exchanges. This result is a generalization of the experimental results of Li et al. (2006, 2008), which show that symmetric and bounded protocols sustain cooperation in overlays where the degree is constant for all agents (and thus known) and the topology is always connected.

Our main results are summarized in Figs. 6.1 and 6.2. In valuable pairwise exchanges (Fig. 6.1), we show that if \mathcal{G}^* does not admit weak timely punishments (WTP), then no protocol sustains cooperation (grey area), but if \mathcal{G}^* is restricted by strong timely punishments (STP) and there are at least three rounds per stage, then there is a bounded protocol $\vec{\sigma}^{\text{val}}$ that sustains cooperation. In general one-shot pairwise exchanges (Fig. 6.2), we show that if \mathcal{G}^* is not restricted by strong timely punishments (STP), then no protocol sustains cooperation (grey area), if \mathcal{G}^* is not restricted by eventual distinguishability (ED), then no symmetric and bounded protocol sustains cooperation (dotted area), and if \mathcal{G}^* is restricted by connectivity with known degrees (CKD), then there is a symmetric and bounded protocol $\vec{\sigma}^{\text{gen}}$ that sustains cooperation.

We now prove each of these results in turn. In the proofs of these results, we use multiple key

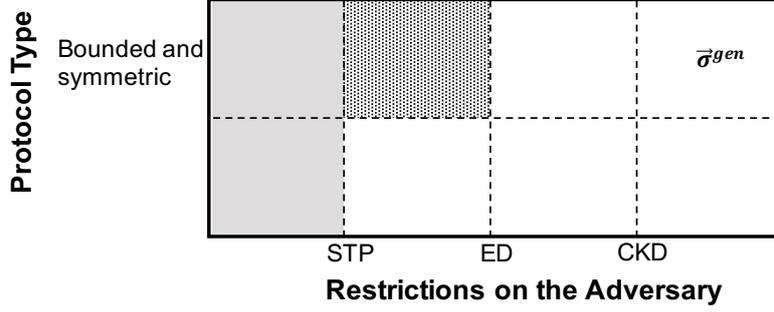


Figure 6.2: Results for General One-shot Pairwise Exchanges (Section 6.3).

concepts related to properties of the protocols and evolving graphs. We define these concepts first, before proving the results. Table 6.1 summarizes the most important notation used in this chapter.

6.1 Key Concepts

We define the concepts of punishment opportunities and indistinguishable evolving graphs, which are related to properties of the evolving graphs, we introduce the notion of evasive protocols, and we define symmetric and bounded protocols.

Punishment Opportunities. Roughly speaking, a punishment opportunity (PO) for an interaction of agent i with j is a later interaction between an agent l and i where l may have been informed of a deviation of i towards j and thus has the opportunity to punish i . This requires the existence of a temporal path (a sequence of causally influenced interactions) from j to l in the evolving graph such that i cannot interfere with information forwarded from j to l along this path. Formally, given $G \in \mathcal{G}^*$, agent i , and stage t , an i -edge is a pair (j, t) such that (i, j) is an edge in G^t . Given round m from stage t and round m' from stage t' , we say that j causally influences l in G between (t, m) and (t', m') , denoted $(j, t, m) \rightsquigarrow^G (l, t', m')$, if $(t, m) < (t', m')$ (i.e., $t < t'$ or $t = t'$ and $m < m'$) and either $j = l$ or there is a j -edge (o, t'') in G and a round m'' from stage t'' such that $(o, t'', m'') \rightsquigarrow^G (l, t', m')$. We say that j causally influences l in G without interference from i between (t, m) and (t', m') , denoted as $(j, t, m) \rightsquigarrow_i^G (l, t', m')$, if the above holds for $o \neq i$. A PO of i for (j, t, m) in G is a tuple (l, t', m') such that (l, t') is an i -edge in G and $(j, t, m) \rightsquigarrow_i^G (l, t', m')$.

Type	Notation	Description
Agents	\mathcal{N}	Set of agents.
	n	Number of agents.
	v_i^t	Value of i in stage t .
Network	\mathcal{G}^*	Set of evolving graphs generated by the adversary.
	G^t	Communication graph from stage t .
	$\mathcal{G}_i^t(G)$	Set of graphs that provide same information to i as G^t .
	$(j, t, m) \rightsquigarrow^G (l, t', m')$	Causal influence in G .
	$(j, t, m) \rightsquigarrow_i^G (l, t', m')$	Causal influence without interference from i in G .
	(j, t)	i -edge.
Actions and histories	a_i^m	Round- m action of agent i .
	h	Global history.
	I_i	Information set for agent i .
	$\mathcal{I}_i(G)$	Set of information sets compatible with G .
	$\mathcal{R}(G, I_i)$	Set of runs compatible with information set I_i and evolving graph G .
	$\mathcal{R}(h)$	Set of runs compatible with global history h .
Time	τ	Total number of rounds per stage.
Utilities	δ	Discount factor.
	μ	Belief system.
	β_i	Benefit that i obtains when it outputs a value.
	α_i	Cost that i incurs per bit sent in a message.
	γ_i	Cost that i incurs per bit received in a message.
	π_i	Cost that i incurs per penance sent in a message.
	$u_i^z(\bar{\sigma} G, I_i)$	Expected utility of i in epoch z when agents use $\bar{\sigma}$ conditioned on G and I_i .
	$u_i(\bar{\sigma} G, I_i)$	Expected utility of i when agents use $\bar{\sigma}$ conditioned on G and I_i .
	$u_i(\bar{\sigma} G, h)$	Expected utility of i when agents use $\bar{\sigma}$ conditioned on G and h .
	$u_i(\bar{\sigma} G)$	Expected utility of i when agents use $\bar{\sigma}$ conditioned on G .
Algorithm	acc_i^t	Accusations against other agents.
	vs_i^t	Validity of messages exchanged between i and neighbours.
	$p_i^t(j)$	Number of penances that i owes to j in stage t .
	ρ	Maximum delay of dissemination of monitoring information.
	pnd_i^t	Number of pending punishments.
	deg_i^t	Degree of i in stage t .
Strategies	σ_i	Strategy for agent i .
	$\bar{\sigma}$	Strategy profile.
	$\bar{\sigma}^{\text{val}}$	Protocol for valuable pairwise exchanges.
	$\bar{\sigma}^{\text{gen}}$	Protocol for general one-shot pairwise exchanges.
	$\bar{\sigma} _{I_i, a_i}$	One-shot deviation where i takes a_i at I_i .

Table 6.1: Notation - pairwise exchanges in dynamic networks.

Indistinguishable Evolving Graphs. We say that an evolving graph G is indistinguishable from evolving graph G' to agent i at stage t if i acquires the same information about G and G' , regardless of the protocol followed by agents. Specifically, given $G \in \mathcal{G}^*$, stage t , and agent

i , let $\mathcal{G}_i^t(G)$ be the set of graphs that provide the same the information to i about the topology in stage t as G^t , and let $C_i^t(G)$ be the set of agents l such that $(j, t', 1) \rightsquigarrow^G (i, t, \tau)$ for some $t' \leq t$ (recall that τ is the number of rounds per stage). That is, $C_i^t(G)$ is the set of agents j that can causally influence i between the beginning of stage t' and the end of stage t . For all agents i , evolving graphs $G, G' \in \mathcal{G}^*$, and stages t , G is indistinguishable from G' to i at t iff (i) $C_i^t(G) = C_i^t(G')$ and (ii) for all stages $t' \leq t$ and agents $j \in C_i^t(G)$, $\mathcal{G}_j^{t'}(G) = \mathcal{G}_j^{t'}(G')$.

Evasive Protocols. Given a strategy profile $\vec{\sigma}^*$, an evasive protocol for agent i is a strategy σ'_i where i deviates from σ_i^* and then hides this deviation from as many agents as possible, for as long as possible. In our impossibility results, we formally define specific types of evasive protocols.

Bounded Protocols. A protocol $\vec{\sigma}$ is said to be bounded if the duration of punishments is bounded. Specifically, every protocol can be represented as a state machine (Osborne & Rubinstein 1994): each global history h corresponds to a state s^h from a set $S_{\vec{\sigma}}$ and, if agents follow $\vec{\sigma}$ at h and the resulting global history is h' , then the corresponding state $s^{h'}$ is also in $S_{\vec{\sigma}}$. We consider that agents are in one of two types of states, namely, either they punish some agents or they send messages of fixed size l containing their values. Say that a state $s \in S_{\vec{\sigma}}$ is a *cooperation state* if for all stages t , global histories h from t that correspond to s , runs $r \in \mathcal{R}(h)$ of $\vec{\sigma}$, and agents i , i sends its value v_i^t in a message of size l to every neighbour in G^t in stage t . A protocol $\vec{\sigma}$ is said to be *bounded* if it is self-stabilizing in bounded time (Dolev 2000): (1) the set of possible states $S_{\vec{\sigma}}$ is finite, (2) there is a nonempty subset $S^* \subseteq S_{\vec{\sigma}}$ of states that are cooperation states, (3) if agents follow $\vec{\sigma}$ at a state $s^* \in S^*$, then the resulting state is also in S^* , and (4) there is a constant $\rho > 0$ such that for all stages t , global histories h from stage t , and runs $r \in \mathcal{R}(h)$ of $\vec{\sigma}$, $r(t + \rho)$ corresponds to a state in S^* . A bounded protocol $\vec{\sigma}$ *sustains cooperation* if it is a \mathcal{G}^* -OAPE and the initial state is a cooperation state.

Symmetric Protocols. A protocol $\vec{\sigma}$ is said to be *symmetric* if for all agents i , σ_i requires agent i to send messages with a fixed size to all neighbours at all information sets. Otherwise, the protocol is said to be nonsymmetric.

6.2 Sustaining Cooperation with Strongest Adversary

We now prove our first two results. We start by showing that weak timely punishments are a necessary restriction on \mathcal{G}^* in order to sustain cooperation in pairwise exchanges, and that strong timely punishments are necessary to sustain cooperation in one-shot pairwise exchanges. Then, we define a protocol for pairwise exchanges that is a \mathcal{G}^* -OAPE if \mathcal{G}^* is restricted by strong timely punishments and pairwise exchanges are valuable.

6.2.1 Need for Timely Punishments

We say that the *adversary is restricted by weak timely punishments* iff there exists some bound $\rho > 0$ such that for all evolving graphs $G \in \mathcal{G}^*$, stages t , and edges (i, j) in G^t , there exists a stage $t' > t$, round m' from stage t' , and agent j' such that $t' < t + \rho$ and, for some round m from stage t , either (j', t', m') is a PO of i for (j, t, m) in G or (j', t', m') is a PO of j for (i, t, m) in G . We call this restriction *weak timely punishments*. Intuitively, if i and j interact and send messages in some round m from stage t , then at least one of them must be able to report on a deviation of the other to another agent j' that is able to punish the deviating agent in a later stage.

We say that the adversary is *restricted by strong timely punishments* iff there exists some bound $\rho > 0$ such that for all evolving graphs $G \in \mathcal{G}^*$, stages t , agents i , and i -edges (j, t) in G , there exist a stage $t' > t$ and agent j' such that $t' < t + \rho$ and $(j', t', 1)$ is a PO of i for (j, t, τ) in G . We call this restriction *strong timely punishments*. This restriction is stronger than weak timely punishments in two aspects namely, (1) both i and j can be punished after a bounded number of stages for deviating in stage t , and (2) agents can be punished in the first round of some stage $t' > t$ for deviations in the last round of stage t .

Theorem 12 shows that weak timely punishments are necessary to sustain cooperation. The proofs use the notion of single-omission evasive strategy, which we now define.

Given a strategy profile $\vec{\sigma}$, evolving graph $G \in \mathcal{G}^*$, agents i and j , and round m from stage t , a single-omission strategy σ'_i relative to the tuple $(\vec{\sigma}, G, j, t, m)$ is a strategy where i deviates from σ_i by omitting messages to j at all round- m information sets and then behaves as if i had not deviated, such that all agents l not causally influenced by j never learn of the deviation of i . Formally, σ'_i is defined as follows. σ'_i is identical to σ_i at every round- m' information set

I_i from stage t' such that $(t', m') < (t, m)$ or $I_i \notin \mathcal{I}_i(G)$. At every remaining information set I_i , σ'_i specifies a probability distribution over a pair (a_i, I'_i) containing an action a_i that i takes at I_i and information set I'_i that i uses to later pretend that it did not deviate from σ_i at I_i . More precisely, we define σ'_i recursively for pairs $(t', m') \geq (t, m)$ as follows. At every round- m information set $I_i \in \mathcal{I}_i(G)$ from stage t , with probability $\sigma_i^*(a_i^* | G^1)$, i follows a_i identical to a_i^* except i omits messages to j . After observing an information set I'_i consistent with i following a_i , i selects I''_i identical to I'_i except in I''_i agent i follows a_i^* at I_i . This concludes the definition for round m . Now, given the definition for $(t', m') \geq (t, m)$, and given round- m' information set $I_i \in \mathcal{I}_i(G)$ and corresponding information set $I'_i \in \mathcal{I}_i(G)$ selected by i , i follows a_i with probability $\sigma_i(a_i | I'_i)$. After taking action a'_i and observing an information set I''_i , agent i selects I'''_i equivalent to I'_i for pairs $(t'', m'') \leq (t', m')$ and compatible with I''_i in the round- m' actions (i.e., i makes the same observations regarding round- m' actions in I'''_i and I''_i).

We now show that, if for some pair (t', m') of a stage t' and round m' , there is no PO (j, t'', m'') of i for (j, t, m) in G for every pair $(t'', m'') \leq (t', m')$ and i follows a single-omission evasive strategy σ'_i relative to $(\vec{\sigma}, G, j, t, m)$, then every neighbour of i in stages $t'' \leq t'$ makes the same observations with the same probability at every such round m'' from stage t'' , whether i follows σ'_i or σ_i . Let $\vec{\sigma}' = (\sigma'_i, \vec{\sigma}_{-i})$. For the sake of simplicity, we consider that every run r of $\vec{\sigma}'$ also specifies the choices of information sets made by i prior to every round. Given a pair (t'', m'') , let $S_{t'', m''}$ be the set of agents l not causally influenced by j between (t, m) and (t'', m'') , i.e., $(j, t, m) \rightsquigarrow^G (l, t'', m'')$ is false. Given an information set $I_i \in \mathcal{I}_i(G)$, global history $h \in I_i$, pair $(t'', m'') \geq (t, m)$, set $S_{t'', m''}$, and round- m'' information set $I_{S_{t'', m''} \cup \{i\}}^* = \bigcap_{l \in S_{t'', m''} \cup \{i\}} I_l^*$ from stage t'' that fixes the observations of agents in $S_{t'', m''} \cup \{i\}$, let $Q^{\vec{\sigma}'}(I_{S_{t'', m''} \cup \{i\}}^* | G, h)$ be the probability of the run r of $\vec{\sigma}'$ being such that the observations of agents in $S_{t'', m''}$ are given by $I_{S_{t'', m''}}^*$ and i selects I_i^* in r . Lemma 11 shows that $Q^{\vec{\sigma}'}(I_{S_{t'', m''} \cup \{i\}}^* | G, h)$ is the same as the probability of agents in $S_{t'', m''} \cup \{i\}$ observing $I_{S_{t'', m''} \cup \{i\}}$ when they follow $\vec{\sigma}$. Since there is no PO (l, t'', m'') of i for (j, t, m) in G with $(t'', m'') \leq (t', m')$, every neighbour l of i in stages $t'' \leq t'$ is in $S_{t'', m''}$ for all pairs $(t'', m'') \leq (t', m')$, so every such neighbour l observes each information set at every such round m'' with the same probability, whether i follows σ'_i or σ_i .

Lemma 11. *Given strategy profile $\vec{\sigma}$, agents i and j , stages t and t' , rounds m from t and m' from t' , and evolving graph $G \in \mathcal{G}^*$, if there is no PO (l, t'', m'') of i in G for (j, t, m) for all $(t'', m'') \leq (t', m')$, then for all round- m information sets $I_i \in \mathcal{I}_i(G)$ from stage t , global history*

$h \in I_i$, pairs $(t'', m'') \leq (t, m)$, and round- m'' information sets $I_{S_{t'', m''} \cup \{i\}}$ from t'' , we have

$$Q^{\vec{\sigma}'}(I_{S_{t'', m''} \cup \{i\}} | G, h) = \sum_{r \in \mathcal{R}(I_{S_{t'', m''} \cup \{i\}})} P_{\vec{\sigma}, G}(r | h), \quad (6.1)$$

where $\vec{\sigma}' = (\sigma'_i, \vec{\sigma}_{-i})$, σ'_i is a single-omission evasive strategy relative to $(\vec{\sigma}, G, j, t, m)$, and $P_{\vec{\sigma}, G}(r | h)$ is the probability of run r conditioned on h .

Proof. Suppose without loss of generality that m is not the last round from stage t (the analysis is exactly the same for the last round of stage t , except the pair that follows (t, m) is $(t+1, 1)$). Consider round $m'' = m+1$ and fix h . Clearly, every agent different from i follows each round- m action with the same probability, whether i follows σ'_i or σ_i . In addition, when following σ'_i , i selects a_i^* with probability $\sigma_i(a_i^* | I_i)$, and sends messages according to a_i^* except to j . Since $S_{t, m+1}$ includes all agents but j and i , then every $l \in S_{t, m+1}$ makes private observations with the same probability whether i follows σ_i or σ'_i . Moreover, i selects each information set I_i^* with a probability that depends on the actions of other agents and a_i^* only, so i selects I_i^* with the same probability that i observes this information set when agents use $\vec{\sigma}$. This proves the result for every pair (t'', m'') that immediately follows (t, m) .

Now, suppose that the hypothesis holds for stage t'' and round m'' from stage t'' with $(t, m) < (t'', m'') < (t', m')$, and suppose again that m'' is not the last round from stage t'' . Fix round- m'' information set $I_{S_{t'', m''} \cup \{i\}}^1$ from stage t'' . Since i has no PO (l, t'', m'') for (j, t, m) , for every agent $l \in S_{t'', m''+1} \cup \{i\}$, $l \in S_{t'', m''} \cup \{i\}$, so l follows each action a_i^* with probability $\sigma_l(a_i^* | G, I_l^1)$. Moreover, every neighbour of l in round m'' is in $S_{t'', m''} \cup \{i\}$, hence, given $I_{S_{t'', m''} \cup \{i\}}$, l makes observations in round m'' and i selects each round- $m''+1$ information set with the same probability, whether i follows σ_i or σ'_i . The result follows directly from the hypothesis. \square

We can now prove Theorem 12.

Theorem 12. *If the adversary is not restricted by weak timely punishments, then there is no protocol that sustains cooperation in pairwise exchanges.*

Proof. The proof is by contradiction. Suppose that $\vec{\sigma}^*$ is a \mathcal{G}^* -OAPE and that agents always exchange their values while using $\vec{\sigma}^*$. Suppose also that the adversary is not restricted by weak

timely punishments. For every $\rho > 0$, there is $G \in \mathcal{G}^*$, stage t , and edge (i, j) in G^t such that, for all rounds m from stage t and every tuple (l, t', m') with $t' > t$ that is either a PO of i for (j, t, m) or a PO of j for (i, t, m) in G , we have $t' - t \geq \rho$. This implies that, for every stage $t' \geq t$ with $t' < t + \rho$, every i -edge (l, t') with $t' > t$, and rounds m from stage t and m' from stage t' , $(j, t, m) \rightsquigarrow^G (l, t', m')$ is false; similarly, the same holds for j . Let ρ be such that $y\delta^\rho/(1-\delta) < c$, where y is the (bounded) maximum range of the utility of a single agent in a stage, and $c > 0$ is the minimum cost of sending a message in a stage, and fix corresponding G, i, j , and t . Since both i and j send their values in stage t , there are rounds m_i and m_j from stage t , round- m_i information set $I_i \in \mathcal{I}_i(G)$, and round- m_j information set $I_j \in \mathcal{I}_j(G)$ consistent with $\vec{\sigma}^*$ such that i sends a message to j at I_i , j sends a message to i at I_j , and both i and j receive each others' values after sending those messages. Suppose without loss of generality that i is the last agent to send a message at an information set I_i consistent with $\vec{\sigma}^*$, and let m be the last round when i sends a message. Let σ'_i be a single-omission evasive strategy relative to $(\vec{\sigma}^*, G, j, t, m)$, and let $\vec{\sigma}' = (\sigma'_i, \vec{\sigma}^*_{-i})$. By Lemma 11, for all stages t' with $t < t' < t + \rho$ and rounds m' from stage t' , the round- m' neighbours of i observe each information set after round m' with the same probability, whether i follows σ'_i or σ_i^* . Moreover, i also takes the same actions at every such round m' . It follows that the expected utility of i in round m' is the same, whether agents follow $\vec{\sigma}'$ or $\vec{\sigma}^*$. In stage t , i avoids a cost $c > 0$ of omitting a message to j in round m . Moreover, the maximum utility loss in every stage $t' \geq t + \rho$ is yn . So, if j does not punish i in stage t , then we have

$$u_i(\vec{\sigma}^* \mid G, I_i) - u_i(\vec{\sigma}' \mid G, I_i) < -c + \delta^\rho yn/(1-\delta) < 0.$$

This contradicts the assumption that $\vec{\sigma}^*$ is a \mathcal{G}^* -OAPE. So, suppose instead that j punishes i in stage t . Since i has already received j 's value, the only way that j can punish i is to send messages to i or to require i to send further messages. By the same arguments as above, either i or j gain by omitting those additional messages, so $\vec{\sigma}^*$ can never be a \mathcal{G}^* -OAPE, which is a contradiction. This concludes the proof. \square

In one-shot pairwise exchanges between pairs of agents i and j , both i and j must send messages in every round of each stage t , so in particular i and j send messages in the last round of every stage t . By the same arguments of the proof of Theorem 12, if the adversary is not restricted by strong timely punishments, then there is an agent i that can follow a single-omission evasive strategy such that i omits messages in stage t , no agent punishes i in stage t ,

and agents that punish i in stages $t' > t$ only do it when it is too late, i.e., when the loss of those punishments discounted to stage t is lower than the immediate gain of i omitting messages in stage t . Therefore, Corollary 13 follows immediately from the proof of Theorem 12.

Corollary 13. *If the adversary is not restricted by strong timely punishments, then no protocol sustains cooperation in one-shot pairwise exchanges.*

Although the proofs of these results are relatively straightforward, the results are central for the remainder of the chapter, as they show that it must be common knowledge that \mathcal{G}^* satisfies some minimum properties. Moreover, the results are general, since they apply to all types of protocols, not just bounded and symmetric.

6.2.2 A Protocol for Valuable Pairwise Exchanges

We now define a bounded protocol $\vec{\sigma}^{\text{val}}$ that sustains cooperation in a setting of valuable pairwise exchanges (defined below), assuming that the adversary is restricted by strong timely punishments, and that the number of rounds per stage τ is at least three. We first discuss the protocol $\vec{\sigma}^{\text{val}}$, before providing a formal definition of valuable pairwise exchanges and proving that $\vec{\sigma}^{\text{val}}$ sustains cooperation.

Let ρ be the constant of the definition of strong timely punishments. The main incentives of $\vec{\sigma}^{\text{val}}$ are based on proportional punishments: if at stage t agent j knows that i deviated in c stages from $\{t - \rho \dots t - 1\}$, then j forces i to pay a cost proportional to c by sending c penance values that we call penances. Agents detect deviations and inform other agents as follows: if an agent i omits a message to neighbour j in stage t , then j emits an accusation; this accusation is disseminated across the network; when an agent l later interacts with i in a stage $t' < t + \rho$, l forces i to send a number of penances proportional to the number of accusations that l has against i for stages in $\{t' - \rho \dots t' - 1\}$.

In more detail, for every stage t , agent i keeps a vector acc_i^t of accusations, where, for all agents j and stages $t' \in \{t - \rho \dots t - 1\}$, $acc_i^t(t', j)$ is a boolean that (if equal to *True*) represents an accusation against i emitted by j in stage t' . The protocol requires three rounds per stage: in every round, every agent i sends acc_i^t to all neighbours j , and updates this vector at the end of the round by registering new accusations; in round 2, i also sends a number of penance values to j proportional to the number of accusations against i that j sends to i in round 1; finally, in

round 3, i sends its value to j . All messages have a fixed size, except round 2 messages, which have a variable size that depends on the number of penances that an agent sends. Agents i and j exchange messages until one of them does not send a valid (correctly formatted) message or omits penances. To keep track of this, agent i uses a vector vs_i^t of booleans, where, for all agents $j \neq i$, $vs_i^t(j) = True$ iff both i and j sent only correctly formatted messages and did not omit any requested penances to each other. This implies that agent i sends its value to agent j only if j sends the required messages in rounds 1 and 2 to i (and in particular j sends the required number of penances to i), so j pays the "cost" of not receiving i 's value in round 3 if j does not send the required messages in rounds 1 and 2. If at the beginning of round 3 $vs_i^t(j) = True$ and j does not send its value to i in round 3, then i sets $acc_i^t(t, j) = True$, i.e., i emits an accusation against j in stage t . This accusation is forwarded to other agents in stages $t' \in \{t+1 \dots t+\rho-1\}$. We show that, if the adversary is restricted by strong timely punishments, then j always expects to send an additional penance in the future if j omits a round-3 message to i .

The pseudo-code for the algorithm run by agent i is depicted in Alg. 4. Initially, i initializes its variables (lines 1-5). In addition to the aforementioned variables, agent i uses variables $p_i^t(j)$ and $p_j^t(i)$, which indicate the number of penances that i has to send to j and receive from j in stage t , respectively. The values of variables $p_i^t(j)$ and $p_j^t(i)$ are computed from the accusations that j sends to i and that i sends to j in round 1 of stage t , respectively. Specifically, for each stage $t' \in \{t-\rho \dots t-1\}$ such that j has an accusation to i for stage t' , i must send n penances. This guarantees that i incurs a cost at least as high as the cost that i avoids by omitting round-3 messages to all neighbours in stage t' . Conversely, j must also send n penances for each different accusation that i sends to j in round 1. In the send phase of every round m from stage t , i sends a tuple $\langle acc_i^t, S \rangle$ to every neighbour j such that $vs_i^t(j) = True$, where S is either (i) \perp if $m = 1$, (ii) a set of $p_i^t(j)$ penances if $m = 2$, or (iii) i 's input v_i^t in stage t if $m = 3$ (lines 7-16). In the receive phase, i updates its variables according to the messages that each neighbour j sends to i (lines 19-32). If j sends an invalid message or omits a message to i and $vs_i^t(j) = True$, then i sets $vs_i^t(j)$ to *False*; moreover, if this happens in round 3, then i emits an accusation against j for stage t , i.e., i sets $acc_i^t(t, j) = True$ (lines 19-22). Agent i also sets $vs_i^t(j)$ to *False* if j omits some penance in round 2 (line 25). If instead j sends the requested message, then i registers all new accusations received from j (lines 27 -28); in round 1, i calculates the number of penances $p_i^t(j)$ that i must send in round 2 (line 30); and, in round 3, i saves j 's input (line 32). Agent i performs a similar update to vs_i^t for each message that i sends or omits to neighbour j

(lines 33-39). Namely, i sets $vs_i^t(j)$ to *False* if i sends an invalid message, omits a message, or omits penances. In round 2, i also calculates the number of penances $p_j^t(i)$ that j must send to i in round 2. Finally, in the update phase of round 3, i outputs every received value (lines 41-47).

Algorithm 3 σ_i^{val} : i 's protocol for valuable pairwise exchanges

```

1: for all stages  $t$  and agents  $j \neq i$  do
2:    $vs_i^t(j) \leftarrow \text{True}$ 
3:    $v_j^t \leftarrow \perp$ 
4:   for all stages  $t' \in \{t - \rho \dots t - 1\}$  do
5:      $acc_i^t(t', j) \leftarrow \text{False}$ 
6: for all stages  $t \geq 1$  and rounds  $1 \leq m \leq 3$  do
7:   Phase 1: send phase
8:     for all neighbours  $j \neq i$  such that  $vs_i^t(j) = \text{True}$  do  $\triangleright$  Send messages only while relationship is in a good state
9:       if  $m = 1$  then
10:        Send  $\langle acc_i^t, \perp \rangle$  to  $j$   $\triangleright$  Sends only accusations
11:       else if  $m = 2$  then
12:         $S \leftarrow$  set of  $p_i^t(j)$  penances  $\triangleright$  Number of accusations that  $j$  has against  $i$ 
13:        Send  $\langle acc_i^t, S \rangle$  to  $j$   $\triangleright$  Sends  $p_i^t(j)$  penances
14:       else if  $m = 3$  then
15:        Send  $\langle acc_i^t, v_i^t \rangle$  to  $j$   $\triangleright$  Sends value to  $j$ 
16:   EndPhase
17:   Phase 2: receive phase
18:     for all neighbours  $j \neq i$  do
19:       if  $vs_i^t(j) = \text{True}$  and  $j$  omits or sends invalid message to  $i$  then
20:          $vs_i^t(j) \leftarrow \text{False}$ 
21:       if  $m = 3$  then
22:          $acc_i^t(t, j) \leftarrow \text{True}$ 
23:       else if  $vs_i^t(j) = \text{True}$  and  $j$  sent  $\langle acc_j^t, S \rangle$  to  $i$  then
24:         if  $m = 2$  and  $\#S < p_j^t(i)$  then
25:            $vs_i^t(j) \leftarrow \text{False}$ 
26:         else
27:           for all agents  $l \neq j, i$  and stages  $t' \in \{t - \rho \dots t - 1\}$  such that  $acc_j^t(t', l) = \text{True}$  do
28:              $acc_i^t(t', l) \leftarrow \text{True}$   $\triangleright$  Registers all new accusations
29:           if  $m = 1$  then
30:              $p_i^t(j) \leftarrow n \times \#\{t' \mid t - \rho \leq t' \leq t - 1 \wedge acc_j^t(t', i) = \text{True}\}$   $\triangleright$  Number of accusations against  $i$ 
31:           else if  $m = 3$  then
32:              $v_j^t \leftarrow S$   $\triangleright$  Prepares to output value
33:         if  $vs_i^t(j) = \text{True}$  and  $i$  omits or sends invalid message to  $j$  then
34:            $vs_i^t(j) \leftarrow \text{False}$ 
35:         else if  $vs_i^t(j) = \text{True}$  and  $i$  sent  $\langle acc^t, S \rangle$  to  $j$  then
36:           if  $m = 2$  and  $\#S < p_j^t(i)$  then
37:              $vs_i^t(j) \leftarrow \text{False}$   $\triangleright$   $i$  omitted penances
38:           else if  $m = 1$  then
39:              $p_j^t(i) \leftarrow n \times \#\{t' \mid t - \rho \leq t' \leq t - 1 \wedge acc^t(t', j) = \text{True}\}$   $\triangleright$  Number of accusations against  $j$ 
40:   EndPhase
41:   Phase 3: update phase
42:     if  $m = 3$  then
43:        $O \leftarrow \emptyset$ 
44:       for all agents  $j \neq i$  such that  $v_j^t \neq \perp$  do
45:          $O \leftarrow O \cup \{v_j^t\}$ 
46:       Output( $O$ )  $\triangleright$  Outputs all received values
47:   EndPhase

```

It is easy to see that $\bar{\sigma}^{\text{val}}$ is bounded. We now show that $\bar{\sigma}^{\text{val}}$ sustains cooperation under two main assumptions, namely, that the adversary is restricted by strong timely punishments and that pairwise exchanges are valuable. Specifically, we assume that utilities are normalized,

such that, for all agents i , the total cost that i incurs for sending the value v_i^t in round 3 and for sending the vector acc_i^t in all three rounds is 1; let π_i denote the cost of i sending a penance. We say that pairwise exchanges are valuable if $\pi_i > 0$, $\beta_i > 1 + n\rho\pi_i$, and $\gamma_i = 0$ for all agents i . Intuitively, in a valuable pairwise exchange, agents neglect the costs of receiving messages and have a high benefit/cost ratio, such that they always prefer to send any required number of penances (note that $n\rho$ is the maximum number of penances that any agent is required to send) and receive a value in return than to not send or receive any messages. In Section 6.2.3, we show that we can relax some of these assumptions with cryptography.

We now prove Theorem 14, which shows that $\vec{\sigma}^{\text{val}}$ sustains cooperation if the adversary is restricted by strong timely punishments, pairwise exchanges are valuable, and δ is sufficiently close to 1, which is a standard assumption in results about infinitely repeated games where future utilities are discounted by δ . The proof uses the one-shot deviation property for the notion of \mathcal{G}^* -OAPE, which we prove in Appendix A. The key ideas of the proof are the following. If an agent i performs a one-shot deviation at rounds 1 or 2 of stage t , then i avoids the cost of sending additional messages in stage t , but loses the benefit of receiving values in round 3; in valuable pairwise exchanges, the loss in round 3 always outweighs the gain. If agent i performs a one-shot deviation at round 3, then i may avoid the cost of sending round-3 messages, but i has to send more penances in some stage $t' > t$; if δ is sufficiently close to 1, then the cost of sending these penances outweighs the gain in stage t .

Theorem 14. *If the adversary is restricted by strong timely punishments, then there exists $\delta^* \in (0, 1)$ such that for all $\delta \in (\delta^*, 1)$, $\vec{\sigma}^{\text{val}}$ sustains cooperation in valuable pairwise exchanges.*

Proof. Fix $G \in \mathcal{G}^*$, agent i , stage t , round- m , round- m information sets $I_i \in \mathcal{I}_i(G)$ from stage t , $h \in I_i$, and actions a_i^*, a_i' such that $\sigma_i^{\text{val}}(a_i^* | I_i) > 0$. Let $\Delta = u_i(\vec{\sigma}^1 | G, h) - u_i(\vec{\sigma}^2 | G, h)$, where $\vec{\sigma}^1 = \vec{\sigma}^{\text{val}}|_{I_i, a_i^*}$, $\vec{\sigma}^2 = \vec{\sigma}^{\text{val}}|_{I_i, a_i'}$, and $u_i(\vec{\sigma} | h)$ is i 's expected utility conditioned on the run being in $\mathcal{R}(h)$. We show that, if the adversary is restricted by strong timely punishments, pairwise exchanges are valuable, and δ is sufficiently close to 1, then $\Delta \geq 0$. By the one-shot deviation property, it follows that $\vec{\sigma}^{\text{val}}$ is a \mathcal{G}^* -OAPE. Since in $\vec{\sigma}^{\text{val}}$ agents always exchange their values, it follows immediately that $\vec{\sigma}^{\text{val}}$ sustains cooperation in valuable pairwise exchanges.

Given a stage t' , let $\Delta^{t'} = u_i^{t'}(\vec{\sigma}^1 | h) - u_i^{t'}(\vec{\sigma}^2 | h)$, where $u_i^{t'}(\vec{\sigma} | h)$ denotes i 's expected utility in stage t' when agents use $\vec{\sigma}$ conditioned on the run being in $\mathcal{R}(h)$. Fix a stage $t' \geq t$

and agent j . As in gossip dissemination, we denote by $v_j|_{I_j}$ the value of variable v_j held by agent j at an information set I_j . We first prove four facts about $\Delta^{t'}$, where we say that a fact is true when agents use strategy profile $\vec{\sigma}$ if the fact holds in every run $r \in \mathcal{R}(h)$ of $\vec{\sigma}$:

- *Fact 1:* For all agents l , if agents use $\vec{\sigma}^2$, then at the end of stage t' we have $acc_j^{t'}(t, l, i) = True$ iff (1) l is a neighbour of i in G^t , (2) $m = 3$, (3) $vs_i^t(j)|_{I_i} = True$ and i does not send a valid round-3 message to l in a'_i , and (4) $(l, t, 3) \rightsquigarrow_i^G (j, t', 1)$.

Proof. Given $t'' > t$, let $S^{t''}$ be the set of agents o such that $(l, t, 3) \rightsquigarrow_i^G (o, t'', 1)$. We show using induction on t'' that the fact holds. At the end of stage t , we have $acc_i^t(t, l, i) = True$ iff l is a neighbour of i , i does not send a valid round-3 message to l , and $vs_i^t|_{I_i} = True$. Since $S^{t+1} = \{l\}$, the base case follows immediately. Continuing inductively, at the end of every subsequent stage $t'' + 1$, an agent o has an accusation if o already had that accusation at the end of stage t'' or received it from $l \neq i$ in some round from stage $t'' + 1$. In the former case, by the hypothesis, $o \in S^{t''} \subseteq S^{t''+1}$. In the latter case, by the definition of causal influence without interference, $o \in S^{t''+1}$. Either way, the induction hypothesis holds, which concludes the proof of Fact 1. \square

- *Fact 2:* For all agents l and stages $t'' \neq t$, we have $acc_j^{t'}(t'', l, i) = True$ at the end of stage t' when agents use $\vec{\sigma}^1$ iff the same is true when agents use $\vec{\sigma}^2$.

Proof. Note that $acc_j^{t'}(t'', l, i)$ is only defined if $t'' \leq t'$. If $t' = t$ and $t'' < t$, then j has an accusation against i at the end of stage t for t'' iff it already had one such accusation according to h or receives it in stage t from some agent $l \neq i$. Either way, the actions of i have no impact on whether j has an accusation against i for t'' at the end of stage t . Since agents never make up accusations, the fact holds if $t' > t$ and $t'' < t'$ (this is easily shown using induction on $t' > t$). Now suppose that $t'' = t' > t$. In this case, i follows σ_i^{val} in stage t' whether agents are using $\vec{\sigma}^1$ or $\vec{\sigma}^2$. So, i sends all requested messages to every neighbour l , such that $acc_i^{t'}(t', l, i) = False$ at the end of stage t' for all agents l . In particular, $acc_j^{t'}(t', j, i) = False$. Therefore, by the same arguments of the case where $t'' < t$ and $t' > t$, Fact 2 is also true if $t < t'' < t'$. This concludes the proof. \square

- *Fact 3:* If $t' > t$, then $\Delta^{t'} \geq 0$.

Proof. Whether agents use $\vec{\sigma}^1$ or $\vec{\sigma}^2$ at and after h , i follows σ_i^{val} in stage t' , hence i sends $acc_i^{t'}$ in every round to every neighbour j , sends all requested penances in round 2, and sends its value in round 3. By Facts 1 and 2, j never sends more accusations against i in round 1 when agents use $\vec{\sigma}^1$ than when agents use $\vec{\sigma}^2$: this is clearly true by Fact 2 for accusations relative to stages $t'' \neq t$, whereas, by Fact 1, j never sends an accusation relative to t when agents use $\vec{\sigma}^1$ because i sends the requested messages in round 3 from stage t (note that a'_i can be any action, including a_i^*). This implies that i receives the value from every neighbour j , and the costs that i incurs for sending messages when agents use $\vec{\sigma}^2$ are at least as high as when agents use $\vec{\sigma}^1$. Since i incurs no costs for receiving messages, it follows immediately that $\Delta^{t'} \geq 0$. \square

- *Fact 4:* If $m = 3$, $vs_i^t|_{I_i}(j) = \text{True}$, j is a neighbour of i in G^t , and i does not send a valid message to j in a'_i , then there exists a stage $t' \in \{t + 1 \dots t + \rho - 1\}$ such that $\Delta^{t'} = n\pi_i$.

Proof. Suppose that $m = 3$, i does not send a valid message to j in a'_i , and $vs_i^t|_{I_i}(j) = \text{True}$. By Fact 1, for all stages $t' > t$ and agents l such that $(j, t, 3) \rightsquigarrow_i^G (l, t', 1)$, $acc_l^{t'}(t, i) = \text{True}$ if agents use $\vec{\sigma}^2$, and $acc_l^{t'}(t, i) = \text{False}$ if agents use $\vec{\sigma}^1$. By the assumption that the adversary is restricted by strong timely punishments, there exist $t' > t$ and l such that the above holds, $t' < t + \rho$, and l is a neighbour of i in G^t . By Fact 2, l sends one more accusation against i when agents use $\vec{\sigma}^2$ than when they use $\vec{\sigma}^1$, so i sends n additional penances to l when agents use $\vec{\sigma}^2$. By the same arguments of Fact 3, i incurs the same costs for sending accusations and values in stage t' , whether agents use $\vec{\sigma}^1$ or $\vec{\sigma}^2$; i also receives values from all its neighbours in stage t' and sends no more penances relative to stages $t'' < t$ to each neighbour o when using $\vec{\sigma}^1$ than when using $\vec{\sigma}^2$. It follows immediately that $\Delta^{t'} \geq n\pi_i$, as we intended to prove. \square

We conclude by showing that $\Delta \geq 0$. Note that $\Delta = \sum_{t' \geq t} \delta^{t'-t} \Delta^{t'}$. First, suppose that $m \leq 2$ and that i does not send valid messages or omits penances to c neighbours in a'_i . Then, by taking a'_i agent i avoids at most the cost c for sending accusations and values in stage t to those c agents, and avoids at most the cost $cn\rho\pi_i$ of sending penances messages in round 2 to each of the c neighbours (note that i still sends all requested messages to the remaining neighbours). However, i also loses the benefit $c\beta_i$ of receiving the value from those c neighbours. Therefore, $\Delta^t \geq c(\beta_i - n\rho\pi_i - 1)$. By the assumption that pairwise exchanges are valuable, $\Delta^t \geq 0$. By

Fact 3, $\Delta \geq 0$. Now, consider that $m = 3$. If $vs_i^t|_{I_i}(j) = \text{False}$, then i sends no messages in a_i^* , so i does not gain by taking a_i' , and $\Delta \geq 0$. So, suppose instead that $vs_i^t|_{I_i}(j) = \text{True}$ and that i does not send valid messages to c agents j in a_i' . If $c = 0$, then i avoids no costs for sending messages, hence $\Delta^t \geq 0$. If $c > 0$, then i avoids at most the cost c for sending accusations and reports to the c neighbours, hence $\Delta^t \geq -c \geq -n$; however, by Fact 4, there exists a stage $t' \in \{t + 1 \dots t + \rho - 1\}$ such that $\Delta^{t'} \geq n\pi_i$. By Fact 3, we have $\Delta \geq n(\delta\pi_i - 1)$. Since $\pi_i > 1$, if δ is sufficiently close to 1, then $\Delta \geq 0$. This concludes the proof. \square

6.2.3 Relaxing the Assumptions about the Utility

The assumption that pairwise exchanges are valuable can be restrictive in practice if n or ρ are large, since we need $\beta_i = \Omega(n\rho)$ for all agents i . Moreover, agents must know the exact values of β_i and α_i to adjust the size of the penances, which is also restrictive. We now describe a protocol $\vec{\sigma}^*$ that uses a technique proposed by Li et al. (2006) to avoid the assumptions that β_i is large and that agents know β_i and α_i . The idea of $\vec{\sigma}^*$ is that, instead of sending the value in round 3 and sending accusations in every round, agent i sends accusations only in round 1, sends the value to a neighbour j also in round 1 ciphered with a private key κ_j , sends penances in round 2, and then reveals κ_j to j in round 3, but only if j sent all the required messages in rounds 1 and 2. As in $\vec{\sigma}^{\text{val}}$, agents exchange messages until one of them omits penances or does not send a valid message, and agent i emits an accusation against j iff j is expected to send κ_i to i in round 3 but fails to do so; if i emits the accusation, then j is also required to send additional penances in some later stage. The advantage of this approach is that the cost of a penance only has to be higher than the cost of sending a key in round 3. That is, it suffices that $\pi_i > \alpha_i^\kappa$ and $\beta_i > 1 + n\rho\pi_i$ for all agents i , where α_i^κ is the cost of i sending a key. If $\alpha_i^\kappa < \epsilon/(n\rho\pi_i)$ for some constant ϵ , then it suffices that $\beta_i > 1 + \epsilon$. If the key is small comparatively to n , then ϵ is also small and the minimum required benefit/cost ratio is close to the optimal ratio of 1. Of course, this approach only works if agents are computationally bounded, such that they cannot break the ciphering algorithm, and \mathcal{G}^* is sufficiently restricted, such that accusations are still effectively disseminated across the network.

Specifically, given $G \in \mathcal{G}^*$, agent i , and i -edges (j, t) and (l, t') in G , we say that l is stage-reachable from j between t and t' , denoted as $(j, t) \rightarrow_i^G (l, t')$, if $t < t'$ and either $j = l$ or there exists agent $o \neq i$ and stage t'' such that (i, o) is an edge in $G^{t''}$ and $(o, t'') \rightarrow_i^G (o, t'')$. We assume

that there is a bound $\rho > 0$ such that, for all evolving graphs $G \in \mathcal{G}^*$, agents i , and i -edges (j, t) in G , there is an i -edge (l, t') in G such that $(j, t) \rightarrow_i^G (l, t')$ and $t' < t + \rho$. Intuitively, this assumption guarantees that if agent i omits a round-3 message to j and agents use $\vec{\sigma}^*$, then j emits an accusation against i and the accusation reaches an agent l that interacts with and punishes i after a bounded number of stages, even though agents only forward accusations once every stage.

The arguments that $\vec{\sigma}^*$ is a \mathcal{G}^* -OAPE differ from those used in the proof of Theorem 14 in two cases. First, if an agent i receives a value v from agent j in round 1 from stage t , now i is not sure of whether v is v_j^t ciphered with some key known to j , so i is not sure if it is worth it to send the penances in round 2; however, by the consistency property of belief systems, i believes that v is indeed v_j^t ciphered with a key κ_i known by j , so, by the same arguments as before, i does not gain from not sending the penances. Second, in round 3, agent i now avoids at most the cost $n\alpha_i^k$ for not sending valid messages to all neighbours; however, by our assumptions about \mathcal{G}^* , there exists an agent l that is stage-reachable from some neighbour of i and interacts with i in stage $t' < t + \rho$; this agent requests n additional penances from i , so i loses $\delta^\rho n\pi_i$; if $\pi_i > \alpha_i^k$ and δ is sufficiently close to 1, then i does not gain by deviating. This shows that there exists a belief system μ consistent with $\vec{\sigma}^*$ and \mathcal{G}^* such that if δ is sufficiently close to 1, then no agent gains by performing a one-shot deviation from $\vec{\sigma}^*$ at every information set. By the one-shot deviation property, it follows that the protocol $\vec{\sigma}^*$ sustains cooperation.

6.3 Sustaining Cooperation in General Pairwise Exchanges

We have defined a protocol that sustains cooperation assuming that the adversary is restricted by strong timely punishments, that pairwise exchanges are valuable, and that $\tau \geq 3$. We now address the problem of sustaining cooperation in pairwise exchanges where there is only one round per stage, agents may not neglect download costs, and the benefit/cost ratio can be small. Specifically, we assume that the benefit β_i of an agent i receiving a value from the neighbour j is larger than the costs that i incurs for engaging in an exchange with j at a cooperation state. We also consider that utilities are normalized such that the cost of an exchange (of sending and receiving values) at a cooperation state is 1. Therefore, the only assumption that we make about the utilities is that $\beta_i > 1$. This is the minimum restriction on the utilities of agents necessary to sustain cooperation: if the benefit β_i were not higher than the communi-

cation costs of an exchange, then i would prefer not to participate in the system. Finally, we do not assume that agents have exact knowledge of the utilities of other agents (which is usually the case in practice), so protocols must be independent from the utilities. Given this, we say that a bounded protocol $\vec{\sigma}$ sustains cooperation in general one-shot pairwise exchanges if the protocol requires only one round per stage, the initial state of $\vec{\sigma}$ is a cooperation state, and $\vec{\sigma}$ is a \mathcal{G}^* -OAPE for all utilities such that $\beta_i > 1$ for all agents i .

In the remainder of the section, we identify problems with protocols that are not symmetric, we show that \mathcal{G}^* must be further restricted by eventual distinguishability to sustain cooperation with symmetric and bounded protocols, and we present a symmetric and bounded protocol that sustains cooperation in general one-shot pairwise exchanges if the adversary is restricted by connectivity with knowledge of the neighbours' degrees. Because we consider only one round per stage, we never refer to rounds; we drop the round number from the notation, so, e.g., $(j, t) \rightsquigarrow_i^G (l, t)$ denotes that j causally influences l between $(t, 1)$ and $(t', 1)$ in G without interference from i , and a PO (l, t') of i in G for (j, t) is an i -edge in G such that $(j, t) \rightsquigarrow_i^G (l, t)$.

6.3.1 Problems with Nonsymmetric Protocols

Nonsymmetric protocols differ from symmetric protocols in the type of punishments that agents use as an incentive. Specifically, consider the three types of punishments that can be applied by an agent i to its neighbour j in a one-shot pairwise exchange: (1) i may not send its value to j , thus denying the benefit β_j to j , (2) i may require j to send larger messages, and (3) i may send larger messages to j . In symmetric protocols, agents always use the first type of punishment, where they punish their neighbours by sending garbage data instead of their value, and they always send messages of fixed size. In nonsymmetric protocols, agents may also omit bits as a punishment of type 1 or use punishments of type (2) or (3). Note that in general pairwise exchanges agents may still neglect download costs, so the third type of punishments is not always effective. We now show that the other types of punishments used by nonsymmetric protocols are also not effective in general. Specifically, we identify one type of evolving graphs for which protocols where agents omit bits as a punishment are not equilibria. Then, we show that if punished agents are required to send larger messages each time they deviate, then the protocol is not an equilibrium.

6.3.1.1 Problem of Omissions as Punishments

We start by describing a problem inherent to the type of protocols defined in existing proofs of Folk Theorems for games with the structure of pairwise exchanges (Mailath & Samuelson 2007) (e.g., the Prisoners' dilemma game). In these protocols, if an agent i deviates in some stage t , then in a later interaction between i and some agent j in stage $t' > t$, agents i and j do not exchange messages, thus denying i the benefit of receiving j 's value. In dynamic networks, such protocols suffer from the following problem. Note that if i deviates in stage t , then different agents may have to punish i for deviating in different evolving graphs. Suppose that j is the agent that punishes i in a stage t' when the evolving graph is G , and let G' be another evolving graph where j cannot know that i deviates at stage t (e.g., because there is no temporal path from a neighbour of i in stage t to j). If i cannot distinguish G from G' at stage t' , then i is not sure of whether j knows about the deviation. If j knows about the deviation, then i is better off by omitting a message to j . If j does not know about the deviation, then i may fear that by omitting a message to j agent j interprets this as a deviation and triggers additional punishments of i , so that i prefers to send a message to j . Therefore, i would like to condition its action at stage t' on the evolving graph. However, since G and G' are indistinguishable to i , i must take the same action at stage t' , whether the evolving graph is G or G' , so the protocol cannot be an equilibrium. In this case, we say that the punishment of j is ambiguous and that the protocol has ambiguous punishments. To better understand the problem, we now describe a concrete scenario. This scenario shows that protocols that have ambiguous punishments are not equilibria if \mathcal{G}^* is only restricted by strong timely punishments. However, we stress that similar problems arise if agents only omit some bits instead of entire messages or if only the punisher agent omits messages.

Consider the scenario depicted in Fig. 6.3. There are five agents numbered 1 to 5, and two evolving graphs G and G' from \mathcal{G}^* , with the following structure: in stage 1, agent 1 interacts with agent 2 in both evolving graphs; in stage 2, agent 2 interacts with agent 4 in G and interacts with agent 3 in G' ; finally, in stage 3 agent 1 interacts with agents 3 and 4 in both evolving graphs. Suppose that $\vec{\sigma}^*$ is a protocol that sustains cooperation, such that, if agent 1 omits a message to 2 and the adversary selects G , then 1 and 4 do not exchange messages in their interaction in stage 3 as a punishment for 1 deviating towards agent 2. Note that agent 1 cannot distinguish G from G' at stage 3, so if the agent omits a message to 2 in stage 1, then 1 omits

a message to agent 4 in stage 3, whether the evolving graph is G or G' . Now, suppose that the adversary selects G' and that the communication graphs of G' in stages $t \geq 4$ form lines as depicted in Fig. 6.3: 1 is always in between the agents from the set $\mathcal{N}_1 = \{2, 3\}$ and the agents from the set $\mathcal{N}_2 = \{4, 5\}$, and the neighbours of 1 in each stage alternate between 2 or 3 on the left side and between 4 and 5 on the right side. It is easy to see that G' is restricted by strong timely punishments. Consider the following strategy σ_1 for agent 1: (a) 1 does not deviate until stage 3; (b) in stage 3, 1 omits messages to agent 4, (c) in every stage $t > 3$, 1 sends messages towards agents in \mathcal{N}_2 as if 1 omitted messages to agent 2 in stage 1, and (d) in every stage $t > 3$, 1 sends messages towards agents in \mathcal{N}_1 as if 1 were following the protocol $\bar{\sigma}^*$ from the beginning. Note that agent 1 controls the information that flows after stage 3 between agents in \mathcal{N}_1 and \mathcal{N}_2 . If agent 1 uses σ_1 , then agent 1 avoids the costs of sending messages to agent 4 in stage 3 and is not punished by the agents in \mathcal{N}_1 , for those agents never learn that 1 deviates. If agents in \mathcal{N}_2 also do not punish 1, then 1 gains by following σ_1 instead of σ_1^* , so σ_1^* is not a \mathcal{G}^* -OAPE. Instead, suppose that agents in \mathcal{N}_2 punish 1 if 1 omits a message to agent 4. Consider the following strategy σ'_1 for 1: (a) 1 omits messages to agent 2 in stage 1, (b) 1 sends a message to agent 4 in stage 3 as if 1 did not deviate, (c) in every stage $t > 3$, 1 sends messages to agents in \mathcal{N}_1 as if 1 did not deviate from σ_1^* and omitted a message in stage 3 to agent 4, and (d) in every stage $t > 3$, sends messages to agents in \mathcal{N}_2 as if 1 never deviated from σ_1^* . Now, if agent 1 follows σ'_1 , then agents in \mathcal{N}_2 do not punish 1 for omitting messages to agent 4 in stage 3, whereas agents in \mathcal{N}_1 do not punish 1 for not following σ_1^* at stage 3. This implies that agent 1 gains by following σ'_1 at stage 3, so again $\bar{\sigma}^*$ cannot be a \mathcal{G}^* -OAPE, which is a contradiction.

In summary, if the adversary selects G , then σ_1^* requires agent 1 to send a message to agent 4 at stage 3 if 1 does not deviate in stage 1 (because $\bar{\sigma}^*$ sustains cooperation), and requires agent 1 to omit messages to agent 4 if agent 1 omits messages, as a punishment for agent 1 deviating in stage 1. Since agent 1 cannot distinguish G from G' , if the adversary generates G' , then σ_1^* also requires agent 1 to send messages to agent 4 in stage 3 if and only if agent 1 does not deviate in stage 1. However, on one hand, if agents in \mathcal{N}_2 punish 1 for omitting messages to agent 4 in stage 3, then there is an information set I_1 from stage 3 such that (i) 1 did not deviate in I_1 before stage 3 and (ii) 1 gains by deviating from σ_1^* at I_1 , that is, 1 omits messages to agent 4 and then hides the deviation from all agents; on the other hand, if agents in \mathcal{N}_2 punish 1 for omitting messages to agent 4 in stage 3, then there is an information set I'_1 from stage 3 such that (i) 1 did omit a message to agent 2 in stage 1 and (ii) 1 gains by deviating from σ_1^* at

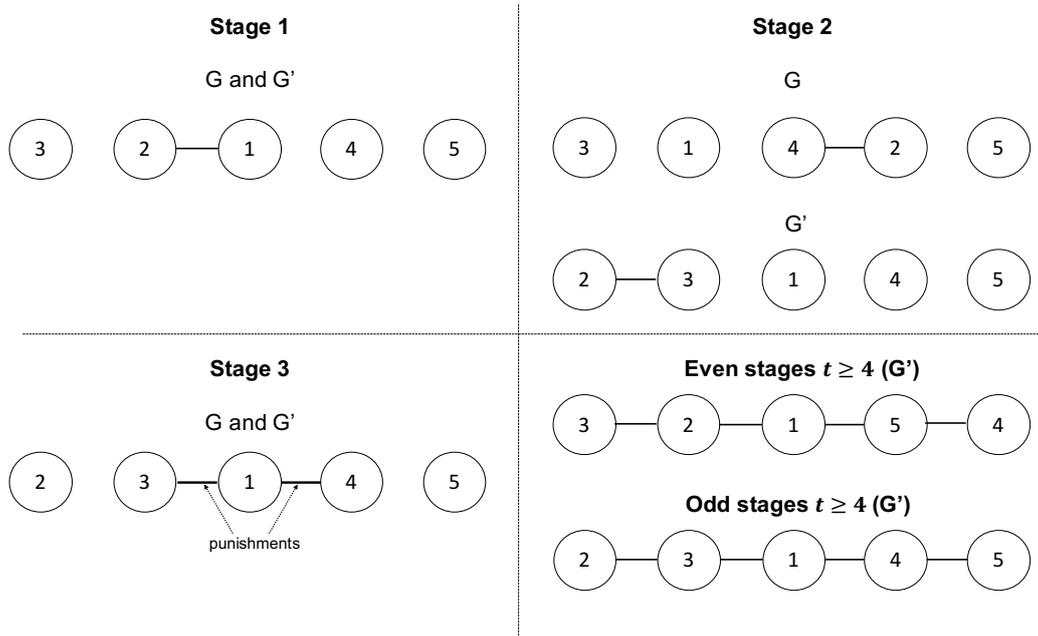


Figure 6.3: Ambiguous punishment.

I_1 , i.e., 1 sends a message to agent 4 and then hides the deviation from all agents. Either way, agent 1 gains by deviating from σ_1^* at some information set if the evolving graph is G' , so $\vec{\sigma}^*$ cannot be a \mathcal{G}^* -OAPE.

6.3.1.2 Problem of Punishments with Large Upload

Consider that the protocol requires every agent i to always send messages, and if i omits messages at stage t , then i is required to send larger messages to some neighbour j at a later stage $t' > t$. Now, suppose that i keeps defecting its neighbours. For a given evolving graph $G \in \mathcal{G}^*$, there is an infinite sequence of agents j_1, j_2, j_3, \dots such that i omits a message of size x_1 to j_1 , then j_2 requires i to send a message of size $x_2 > x_1$ as a punishment but i avoids this punishment by omitting the message, then j_3 requires i to send a message of size $x_3 > x_2$ but i omits this message, and so on. Since the upload capacity is limited, there is a point in time where i can no longer be forced to send larger messages. Hence, if i always omits messages, then i is never punished for deviating, and thus the protocol is not an equilibrium.

6.3.2 Need for Eventual Distinguishability

We now restrict the analysis to symmetric and bounded protocols. We show that \mathcal{G}^* must satisfy the restriction of eventual distinguishability in order to sustain cooperation with symmetric and bounded protocols in general one-shot pairwise exchanges. Roughly speaking, this restriction says that for every evolving graph $G \in \mathcal{G}^*$, there is a stage t^* such that for all stages $t \geq t^*$ and agents i , if i deviates and omits messages to some neighbour j in stage t , then some of the agents that can punish i for deviating (i.e., agents l that interact with i in stages $t' > t$ such that $(j, t) \rightsquigarrow_i^G (l, t')$) can coordinate their punishments such that i gets punished *exactly* one additional time for deviating towards j . The need for this restriction stems from the facts that (1) in symmetric protocols, i must be punished at least once for each deviation, or else i gains by deviating, and (2) if i always omits messages to its neighbours in every stage t and i is punished more than once for each of those deviations, then the number of punishments that i expects to receive after each stage t grows without bound with t , so the delay of punishments can be arbitrarily large and the protocol cannot be bounded.

More precisely, in a symmetric protocol $\vec{\sigma}$, agent i either punishes its neighbours by sending a message containing garbage or sends a message of the same size containing i 's value. Therefore, given an evolving graph G and information set I_i , we can define the expected number of punishments $p_{\vec{\sigma}, G, I_i}$ of i at I_i as the expected number of interactions after I_i in which i is punished, given that agents follow $\vec{\sigma}$ at and after I_i and the evolving graph is G . In a punishment, agent i loses the benefit β_i of receiving a value. In general pairwise exchanges, we only assume that $\beta_i > 1$, so β_i can be arbitrarily close to the cost 1 of i sending and receiving messages. This implies that, for each deviation where i omits a message to some neighbour (and thus avoids the cost 1), i must expect to be punished in at least one additional interaction, or else i gains by omitting messages. Therefore, if at information set I_i from stage t agent i omits messages to y neighbours, then i must be punished y additional times after stage t . Moreover, if i interacts with x agents in stage t and I'_i is the resulting information set of agents taking actions in stage t , then we must have $p_{\vec{\sigma}, G, I'_i} \geq p_{\vec{\sigma}, G, I_i} + y - x$: since i expects to be punished $p_{\vec{\sigma}, G, I_i}$ times at I_i and i can be punished by at most x agents in stage t , the expected number of punishments after stage t must be at least $p_{\vec{\sigma}, G, I_i} - x$ added to the number y of omissions of i in stage t . In other words, the expected number of punishments of i varies over time. If i always omits messages to all neighbours, then the expected number of punishments grows after the first time agent i

omits messages, and then never decreases ever since. This is because at every stage t we have $y = x$ and $p_{\bar{\sigma}, G, I'_i} \geq p_{\bar{\sigma}, G, I_i} + y - x \geq p_{\bar{\sigma}, G, I_i}$. Now, suppose that for an infinite number of stages t , we have $p_{\bar{\sigma}, G, I'_i} > p_{\bar{\sigma}, G, I_i}$. Then, the expected number of punishments grows without bound, and so the protocol cannot be bounded. This happens if for every such stage t , i expects to be punished more than once for omitting messages to some neighbour j in stage t . It turns out that if \mathcal{G}^* is not sufficiently restricted, then there are evolving graphs G such that, for some stages t , the expected number of punishments of i must increase by more than x if i interacts with and omits messages to x neighbours in stage t , no matter what protocol the agents use. We call these stages indistinguishable stages. The restriction of eventual distinguishability requires that, in every evolving graph G , eventually no stage is indistinguishable.

To better understand the definition of indistinguishable stage, we provide an example, which is depicted in Fig. 6.4. There are three agents numbered 1 to 3. The adversary may generate three alternative evolving graphs G^1 , G^2 , and G^3 . In stage 1, agent 1 interacts with agent 2. In stage 2, agent 2 interacts with agents 3. In stage 3, the interactions depend on the evolving graph: (G^1) i interacts only with 2, (G^2) i interacts only with 3, and (G^3) i interacts with both 2 and 3. If the protocol is symmetric and sustains cooperation, then 1 must send a message to 2 in stage 1. If 1 omits messages to 2, then 1 must be punished at least once by either 2 or 3. Suppose that the maximum delay of punishments is two stages, such that stage 3 is the only opportunity to punish i for omitting messages to 2 in stage 1. Then, agent 2 must punish agent 1 if the evolving graph is G^1 and agent 3 must punish 1 if the evolving graph is G^2 . Since 3 does not know about the deviation of 1, agent 2 must tell 3 in stage 2 that 1 deviated if the evolving graph is G^2 . Suppose that the only information that agents have about the topology is the identity of their neighbours. Then, agent 2 cannot distinguish G^1 from G^3 at stage 2, so 2 must also tell agent 3 about the deviation if the evolving graph is G^3 . Moreover, neither agent 2 can distinguish G^1 from G^3 nor agent 3 can distinguish G^2 from G^3 at stage 3. Consequently, if the evolving graph is G^3 and agent 1 omits messages to 2 in stage 1, then both agents 2 and 3 learn of the deviation of 1 and punish 1 in stage 3. Therefore, after stage 1, agent 1 expects to be punished by two neighbours, even though 1 only omits messages to one neighbour.

This example shows that there is an information set $I_1 \in \mathcal{I}_1(G^3)$ for agent 1 from stage 2 such that, according to the information in I_1 available to 1, the expected number of punishments of 1 conditional on G^3 and I_1 is 2, whereas i only omitted messages to one agent in stage 1. If

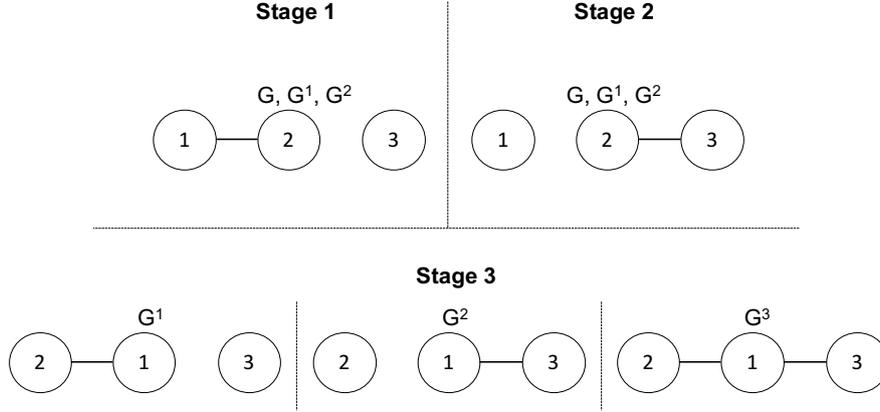


Figure 6.4: Indistinguishable stage.

this type of interactions keeps occurring in G^3 for ever, then, for an arbitrarily large number c , there is an information set $I_1 \in \mathcal{I}_1(G^3)$ such that, basing on the information available at I_1 , agent 1 expects to be punished by at least c neighbours after I_1 , so the protocol cannot be bounded.

We now provide a formal definition of indistinguishable stages and eventual distinguishability. Given an evolving graph G , agent i , and constant ρ , let $\Phi_\rho^G(t)$ be the set of PO's (l, t') of i in G for i -edges (j, t'') such that $t'' \geq t$ and $t' < t + \rho$. We say that stage t is (G, i, ρ) -indistinguishable if there are two evolving graphs $G^1, G^2 \in \mathcal{G}^*$ such that (1) for every $G' \in \{G^1, G^2\}$ and $(j, t') \in \Phi_\rho^{G'}(t)$, G' is indistinguishable from G to j at t' , (2) $|\Phi_\rho^{G^1}(t) \cap \Phi_\rho^{G^2}(t)| < k$, where k is the number of i -edges in stage t , and (3) $\Phi_\rho^{G^1}(t) \cup \Phi_\rho^{G^2}(t) = \Phi_\rho^G(t)$; otherwise, we say that t is (G, i, ρ) -distinguishable. We say that the adversary is restricted by *eventual distinguishability* iff there is a constant $\rho > 0$ and a stage t^* such that for all $G \in \mathcal{G}^*$ and agent i , every stage $t > t^*$ is (G, i, ρ) -distinguishable.

This definition has two parts: (1) the definition of (G, i, ρ) -indistinguishable stage and (2) the requirement that, eventually, no stage is (G, i, ρ) -indistinguishable. Part (1) is a generalization of the scenario depicted in Fig. 6.4. Specifically, the problem identified in the example happens when, for some agent i , stage t , and bound ρ on the delay of punishments, (1) there are sets S_1 and S_2 of agents responsible for punishing i in G^1 and G^2 , respectively, in the $\rho - 1$ rounds after i omits messages to some (or all) neighbours in stage t (in our example, $S_1 = \{2\}$ and $S_2 = \{3\}$), (2) S_1 and S_2 do not intersect, and (3) for $l = 1, 2$, no agent in S_l can distinguish G^3 from G^l . We generalize this intuition by showing that the problem arises even if S_1 and S_2

intersect, but do not intersect in more than k agents, where k is the degree of i in stage t . We also show that, if S_1 and S_2 contain all the agents that can punish i no later than ρ stages after t for i omitting messages in stages $t \dots t + \rho - 1$, then no protocol can avoid the aforementioned problem. This is exactly the case if $S_1 = \Phi_\rho^{G^1}(t)$ and $S_2 = \Phi_\rho^{G^2}(t)$.

Theorem 17 proves the need for eventual distinguishability. The proof is divided into three parts. First, we show in Lemma 15 that if agent i defects k neighbours in a stage t , then the expected number of punishments in the $\rho - 1$ rounds following stage t must increase at least by k , where ρ is the maximum time it takes for punishments to end in a bounded protocol. Then, Lemma 16 shows that, if i constantly defects all neighbours and the adversary is not restricted by eventual distinguishability, then the expected number of punishments of i grows without bound. Finally, we prove Theorem 17, where we identify a contradiction between Lemma 16 and the requirement that the protocol must be bounded.

We start with the proof of Lemma 15. Fix a symmetric and bounded protocol $\vec{\sigma}^*$ that sustains cooperation in general one-shot pairwise exchanges, fix a belief system μ^* consistent with $\vec{\sigma}^*$ and \mathcal{G}^* , and let ρ be the maximum time it takes for $\vec{\sigma}^*$ to converge to a cooperation state. Given an agent i , $G \in \mathcal{G}^*$, information set $I_i \in \mathcal{I}_i(G)$ from stage t , and i -edge (j, t) with $t' > t$, $\vec{\sigma}^*$ defines a probability $P_{\vec{\sigma}^*, G, I_i}(j, t')$ of j punishing i at stage t' . Let $p_{\vec{\sigma}^*, G, I_i}^\rho$ be the expected number of punishments of i in the $\rho - 1$ stages following t , defined as

$$p_{\vec{\sigma}^*, G, I_i}^\rho = \sum_{i\text{-edge } (j, t'): t < t' < t + \rho} P_{\vec{\sigma}^*, G, I_i}(j, t').$$

Lemma 15. *For all evolving graphs $G \in \mathcal{G}^*$, agents i , stages t , information sets $I_i \in \mathcal{I}_i(G)$ from stage t , and protocol $\vec{\sigma}' = (\sigma'_i, \vec{\sigma}'_{-i})$ where, in σ'_i , i omits messages to k neighbours at I_i and follows σ_i^* afterwards, we have*

$$p_{\vec{\sigma}', G, I_i}^\rho \geq p_{\vec{\sigma}^*, G, I_i}^\rho + k.$$

Proof. The proof is by contradiction. Suppose that there is G , i , I_i , and $\vec{\sigma}'$ where i omits messages to k neighbours at I_i and

$$p_{\vec{\sigma}', G, I_i}^\rho < p_{\vec{\sigma}^*, G, I_i}^\rho + k. \tag{6.2}$$

At I_i , i avoids the cost k of sending messages by following σ'_i instead of σ_i^* , whereas the expected benefits and costs of receiving messages are the same. In every stage $t' > t$ and interaction with neighbour j from stage t' , both i and j exchange messages of fixed size. Thus, i incurs the fixed cost 1 of sending and receiving messages, and the expected utility loss of following σ'_i instead of σ_i^* in an interaction with j is determined by the increase in the probability of j punishing i times the loss of β_i . Let S be the largest set of i -edges such that $\Delta(j, t') > 0$ for every $(j, t') \in S$, where $\Delta(j, t') = P_{\vec{\sigma}^t, G, I_i}(j, t') - P_{\vec{\sigma}^*, G, I_i}(j, t')$. Since $\vec{\sigma}^*$ is self-stabilizing in at most ρ rounds, for every $(j, t') \in S$, we have $t' - t < \rho$, and

$$\begin{aligned} u_i(\vec{\sigma}^t \mid G, I_i) - u_i(\vec{\sigma}^* \mid G, I_i) &\geq \\ &\geq k - \sum_{(j', t') \in S} \delta^{t'-t} \Delta(j', t') \beta_i \geq \\ &\geq k - \sum_{(j', t') \in S} \Delta(j', t') \beta_i. \end{aligned} \tag{6.3}$$

By (6.2) and the fact that $\vec{\sigma}^*$ is independent of the utilities, there is a value of $\beta_i > 1$ sufficiently close to 1 such that, for some G and I_i , the utility difference between i following σ'_i and σ_i^* is strictly positive, regardless of the value of δ , so i gains by deviating from σ_i^* at I_i . This is a contradiction to $\vec{\sigma}^*$ being a \mathcal{G}^* -OAPE for every $\beta_i > 1$, thus proving the result. \square

We now prove Lemma 16.

Lemma 16. *If the adversary is not restricted by eventual distinguishability, then there exist $G \in \mathcal{G}^*$ and agent i such that for every $c > 0$, there is $I_i \in \mathcal{I}_i(G)$ such that $p_{\vec{\sigma}^*, G, I_i}^\rho \geq c$.*

Proof. Suppose that the adversary is not restricted by eventual distinguishability. There is $G \in \mathcal{G}^*$, i , and an infinite sequence of stages t_1, t_2, \dots such, that for every k , t_k is (G, i, ρ) -indistinguishable. Given stage t , let $\vec{\sigma}^t = (\sigma_i^t, \vec{\sigma}_{-i}^*)$ be the protocol where i always omits messages up to and including stage t , and follows σ_i^* afterwards. By Lemma 15, for every $G' \in \mathcal{G}^*$ and information set $I_i \in \mathcal{I}_i(G')$ from stage t ,

$$p_{\vec{\sigma}^t, G, I_i}^\rho \geq p_{\vec{\sigma}^*, G, I_i}^\rho + k, \tag{6.4}$$

where k is the number of neighbours of i in stage t . It is easy to show that only the agents in

$\Phi_\rho^G(m)$ can punish i for omitting messages in stage t , hence we have

$$p_{\bar{\sigma}^t, G, I_i}^\rho = p_{\bar{\sigma}^*, G, I_i}^\rho + \sum_{(j, t') \in \Phi_\rho^{G'}(t)} \Delta(j, t').$$

In particular, this is true for $t = t_k$ and $G' \in \{G^1, G^2\}$ such that (1) for every $(j, t) \in \Phi_\rho^{G'}(m)$, G' is indistinguishable from G to j at t , (2) $|\Phi_\rho^{G^1}(t) \cap \Phi_\rho^{G^2}(t)| < k$, and (3) $\Phi_\rho^{G^1}(t) \cup \Phi_\rho^{G^2}(t) = \Phi_\rho^G(t)$. Since G and G' are indistinguishable to every such j at stage t_k , j must punish i with the same probability both in G and G' , so by (6.4) we can write

$$\begin{aligned} & p_{\bar{\sigma}^t, G, I_i}^\rho &= \\ &= p_{\bar{\sigma}^*, G, I_i}^\rho + \sum_{(j, t') \in \Phi_\rho^{G^1}(t)} \Delta(j, t') + \\ & \quad + \sum_{(j, t') \in \Phi_\rho^{G^2}(t)} \Delta(j, t') - \\ & \quad - \sum_{(j, t') \in \Phi_\rho^{G^1}(t) \cap \Phi_\rho^{G^2}(t)} \Delta(j, t') &\geq \\ &\geq p_{\bar{\sigma}^*, G, I_i}^\rho + 2k - (k - 1) &\geq \\ &\geq p_{\bar{\sigma}^*, G, I_i}^\rho + k + 1. \end{aligned}$$

Since beliefs are consistent with $\bar{\sigma}^*$ and G , there is an information set $I'_i \in \mathcal{I}_i(G)$ from stage $t + 1$ such that

$$p_{\bar{\sigma}^t, G, I'_i}^\rho \geq p_{\bar{\sigma}^*, G, I_i}^\rho + k + 1 - k \geq p_{\bar{\sigma}^*, G, I_i}^\rho + 1.$$

This implies that, after every stage t_k , there is an information set such that the expected number of punishments of i increases by at least 1 if i follows σ'_i , while it does not decrease after every other stage. It is easy to show using induction that, for every stage t , there is an information set $I_i \in \mathcal{I}_i(G)$ from stage t such that $p_{\bar{\sigma}^*, G, I_i}^\rho \geq k^*$, where k^* is the largest k with $t_k < t$. Since k^* can be arbitrarily large, the result follows immediately. \square

Finally, we prove Theorem 17.

Theorem 17. *If the adversary is not restricted by eventual distinguishability, then there is no symmetric and bounded protocol that sustains cooperation in general one-shot pairwise exchanges.*

Proof. The proof is by contradiction. Suppose that the adversary is not restricted by eventual distinguishability. By Lemma 16, there is G and i such that for every $c > 0$, there is a stage t and information set $I_i \in \mathcal{I}_i(G)$ from stage t such that $p_{\bar{\sigma}^*, G, I_i}^\rho \geq c$. This is true for t large enough

such that c is larger than the number of i -edges between t and $t + \rho$. This is a contradiction to $\vec{\sigma}^*$ converging to a cooperation state in ρ stages, proving the result. \square

6.3.3 A Protocol for General One-shot Pairwise Exchanges

We now describe a restriction of connectivity on \mathcal{G}^* that ensures that the adversary is restricted by eventual distinguishability. We also show that under this restriction there is a symmetric and bounded a protocol $\vec{\sigma}^{\text{gen}}$ that sustains cooperation in general one-shot pairwise exchanges.

Recall that, in the scenario of Fig. 6.4, the problem is that neither agent 2 can distinguish G^1 from G^3 at stage 3 nor agent 3 can distinguish G^2 from G^3 at stage 3, and thus agents 2 and 3 cannot coordinate their actions to punish 1 only once. This problem does not arise if agents 2 and 3 can learn the degree of agent 1 before sending messages in stage 3. However, the knowledge of the degree is not sufficient to avoid the aforementioned problem in general. To see this, consider a similar scenario where agent 3 only interacts with agent 1 in stage 4. In this alternative scenario, agents 2 and 3 would not be able to coordinate their punishments of agent 1, so agent 1 would still be punished twice for a single deviation. The problem would not arise if 2 also interacted with agent 3 in stage 3, since in this case, 2 could tell 3 not to punish 1 in stage 4.

More generally, the adversary is restricted by eventual distinguishability if the following condition holds: for every $G \in \mathcal{G}^*$, agent i , and stage t , agent i knows the degree of its neighbours in stage t and there is a constant ρ such that the neighbours of i in stage t causally influence every neighbour of i in stage $t + \rho$ without interference from i . This condition is exactly met for $\rho = n$ if \mathcal{G}^* is restricted by a condition similar to 1-connectivity from (Kuhn, Lynch, & Oshman 2010). Specifically, we say that the adversary is *restricted by connectivity with knowledge of degrees* iff for every $G \in \mathcal{G}^*$, agent i , and stage t , i knows the degree of every neighbour in G^t before sending messages in stage t and the graph obtained from G^t by removing the edges to i is connected. We now describe a symmetric and bounded protocol $\vec{\sigma}^{\text{gen}}$ that sustains cooperation in general one-shot pairwise exchanges if the adversary is restricted by connectivity.

The protocol $\vec{\sigma}^{\text{gen}}$ only requires one round per stage. At every stage t , agent i sends a message to each neighbour j containing monitoring information and a value v that is either garbage or

the value v_i^t ; i sends garbage data as a punishment with a probability proportional to the number of past deviations of j . Specifically, the protocol matches each deviation of j to a punishment: agent i keeps count of the number of pending punishments to be applied to j in future stages, which corresponds to the number of past deviations of j ; for each new deviation of j , i increments the number of pending punishments, and decrements it after knowing that j interacted with some agent l that could have applied a pending punishment (i.e., when l interacted with j , l could have been informed of a corresponding deviation). To detect deviations, in every stage t , agents report on the interactions that they have with their neighbours in t ; these reports indicate for each pair of agents (i, j) whether i interacts with j in stage t and, if so, whether j sent a valid message to i . Agents disseminate reports and number of pending punishments across the network to ensure that all agents that punish a single agent j synchronize their punishments in a way that ensures that expected number of punishments of j per deviation is 1.

In more detail, given an evolving graph $G \in \mathcal{G}^*$ and stage t , let \deg_j^t denote the degree of agent j in G^t . Punishments are applied in cycles with period n : if agent i omits messages to k neighbours in stage t , then the neighbours of i in stage $t + n$ punish i with probability proportional to k ; if x neighbours punish i in stage t and $x < k$, then the neighbours of i in stage $t + 2n$ punish i with a probability proportional to $k - x + y$, where y is the number of new deviations of i in stage $t + n$; this procedure is repeated for all stages $t + cn$ for some constant $c > 0$. Agent i uses variables pnd_i^t and acc_i^t to represent the numbers of pending punishments and the reports that i has at stage t , respectively. Specifically, for all agents $j \neq i$ and stages $t' \in \{t - n \dots t\}$, $pnd_i^t(t', j)$ is the number of pending punishments to be applied to j in stages $t' + 1, t' + n + 1 \dots$; for all pairs of agents (j, l) such that $l \neq j$ and stages $t' \in \{t - n \dots t\}$, $acc_i^t(t', j, l)$ is the report of the interaction between j and l in stage t' , where $acc_i^t(t', j, l) = \perp$ if j and l did not interact, $acc_i^t(t', j, l) = True$ if l did not send a valid message to j (i.e., j emitted an accusation against l), and $acc_i^t(t', j, l) = False$ otherwise. Before interacting with agent j in stage t , agent i determines the number of pending punishments $pnd_i^t(t - 1, j)$ of j for stages $t, t + n, \dots$: if $t < n$, then $pnd_i^t(t - 1, j) = 0$, otherwise, i computes the degree \deg_j^{t-n} of j in stage $t - n$, basing on the reports of interactions that other agents had with j in stage t , and sets $pnd_i^t(t - 1, j) = \max(0, pnd_i^t(t - n, j) - \deg_j^{t-n})$, while ensuring that the number of pending punishments is a number in $\{0 \dots n - 1\}$.

In every stage t , i sends a message to j containing acc_i^t , pnd_i^t , and a value v , where v is either

garbage data with probability $\min(1, pnd_i^t(t, j)/deg_j^t)$, or is v_i^t with the remaining probability. Intuitively, i punishes j in stage t with a probability proportional to the number of pending punishments of j and inversely proportional to the degree of j ; we make sure that all neighbours of j punish j in stage t' , so that the expected number of punishments of j is $\min(deg_j^t, pnd_i^t(t, j))$. This way, every agent j is never punished in expectation more than once for each deviation. Moreover, we show that, if the adversary is restricted by connectivity, then every agent j expects to be punished for a deviation in stage t by some neighbour in a stage in $t, t+n, t+2n \dots t+n^2$, so the maximum delay of a punishment is n^2 and the protocol is bounded. In Section 6.3.5, we discuss how to decrease the delay of punishments and the complexity.

Alg. 4 presents the pseudo-code of the strategy σ_i^{gen} for agent i . Agent i first initializes the numbers of pending punishments and the reports for stage 1 with the default values 0 and \perp , respectively (lines 1-5). In the send phase of stage t , i decides whether to punish neighbour j with a probability p^* that is proportional to the number of pending punishments of j that i knows of at the end of stage $t-1$ and inversely proportional to the degree of j in stage t (line 9); i sends a message containing pnd_i^t , acc_i^t , and a value v that is either garbage if i decides to punish j or is v_i^t otherwise. In the receive phase of stage t , i processes the message received from each agent j (lines 13-27). If j sends a valid message, then i sets the report $acc_i^t(t, i, j) = False$ to indicate that j interacted with i and sent a valid message (line 16); if j is a neighbour and did not send a valid message, then i sets $acc_i^t(t, i, j) = True$, which represents an accusation of i against j (line 24); if j is not a neighbour, then i sets $acc_i^t(t, i, j) = \perp$ to indicate that j did not interact with i in stage t . If j sends a valid message, then i also stores the value v that j sends to be output at the end of the stage (line 17). (We assume that i loses nothing for outputting garbage, so that i may output the value v even if $v \neq v_j^t$.) Finally, i updates the number of pending punishments of every agent $l \neq j$ for stages $t' \in \{t-n \dots t-1\}$ to the maximum between the value previously held by i and the value that j sends (line 20), and i updates every new report that j sends to i (line 22). In the update phase of stage t , i first adds every received value to a set O of values to be output and updates the number of pending punishments $pnd_i^t(t, j)$ of every agent j (lines 30-36). Specifically, i first computes the degree $deg_j^{t'}$ of j in stage $t' = t+1-n$ and the number y of deviations of j in stage t' . Note that $deg_j^{t'}$ is the number of agents that could have punished j in stage t' . So, i sets $pnd_j^t(t, j)$ to the result of deducting $deg_j^{t'}$ from $pnd_j^t(t', j)$ and adding y , so that j expects to be punished exactly once for each deviation. In the next step, i initializes the variables for the next stage (lines 37-41). In the

last step, i outputs the set O , which contains all the values that i receives in stage t (line 42).

Algorithm 4 σ_i^{gen} : i 's protocol for general one-shot pairwise exchanges

```

1: for all agents  $j$  and stages  $t' \in \{1 - n \dots 0\}$  do
2:   for all agents  $l \neq j$  do
3:      $acc_i^1(t', j, l) \leftarrow \perp$ 
4:   if  $j \neq i$  then
5:      $pnd_i^t(t', j) \leftarrow 0$ 
6:   for all stages  $t \geq 1$  do
7:     Phase 1: send phase
8:     for all neighbours  $j \neq i$  do
9:        $p^* \leftarrow \min(1, \frac{pnd_i^t(t-1, j)}{\deg_j^t})$  ▷ Probability of punishment
10:       $v \leftarrow$  garbage with prob.  $p^*$  or  $v_j^t$  otherwise ▷ Punishes by sending garbage
11:      Send  $\langle acc_i^t, pnd_i^t, v \rangle$ 
12:    EndPhase
13:    Phase 2: receive phase
14:    for all agents  $j \neq i$  do
15:      if  $j$  sent valid  $\langle acc_j^t, pnd_j^t, v_j \rangle$  then
16:         $acc_i^t(t, i, j) \leftarrow False$  ▷  $j$  behaved according to protocol
17:         $v_j^t \leftarrow v_j$  ▷ Prepares to output value
18:      for all agents  $l$  and  $t' \in \{t - n \dots t - 1\}$  do
19:        if  $l \neq j$  then
20:           $pnd_i^t(t', l) \leftarrow \min(\max(pnd_i^t(t', l), pnd_j^t(t', l)), n - 1)$  ▷ Pending punishments bounded by  $n - 1$ 
21:        for all  $j' \neq j, l$  such that  $acc_j^t(t', l, j') \neq \perp$  and  $acc_i^t(t', l, j') = \perp$  do
22:           $acc_i^t(t', l, j') \leftarrow acc_j^t(t', l, j')$  ▷ Registers all new reports
23:        else if  $j$  is a neighbour of  $i$  then
24:           $acc_i^t(t, i, j) \leftarrow True$  ▷ Emits accusation
25:        else
26:           $acc_i^t(t, i, j) \leftarrow \perp$  ▷  $i$  did not interact with  $j$ 
27:      EndPhase
28:    Phase 3: update phase
29:     $O \leftarrow \emptyset$ 
30:    for all agents  $j \neq i$  do
31:      if  $acc_i^t(t, j) = False$  then
32:         $O \leftarrow O \cup \{v_j^t\}$ 
33:       $t' \leftarrow t + 1 - n$ 
34:       $\deg_j^{t'} \leftarrow \#\{l \neq j \mid acc_i^t(t', l, j) \neq \perp\}$  ▷ Degree of  $j$  in stage  $t + 1 - n$ 
35:       $y \leftarrow \#\{l \neq j \mid acc_i^t(t', l, j) = True\}$ 
36:       $pnd_i^t(t, j) \leftarrow \max(0, pnd_i^t(t' - 1, j) - \deg_j^{t'}) + y$  ▷ Updates number of pending punishments
37:    for all agents  $j$  and stages  $t' \in \{t - n + 1 \dots t\}$  do
38:      if  $j \neq i$  then
39:         $pnd_i^{t+1}(t', j) \leftarrow pnd_i^t(t', j)$ 
40:      for all agents  $l \neq j$  do
41:         $acc_i^{t+1}(t', j, l) \leftarrow acc_i^t(t', j, l)$ 
42:    Output( $O$ )
43:  EndPhase

```

The protocol $\vec{\sigma}^{\text{gen}}$ is bounded and symmetric. Theorem 18 shows that $\vec{\sigma}^{\text{gen}}$ sustains cooperation in general one-shot pairwise exchanges if δ is sufficiently close to 1 and the adversary is restricted by connectivity with known degrees.

Theorem 18. *If the adversary is restricted by connectivity with known degrees, then there exists $\delta^* \in (0, 1)$ such that, for all $\delta \in (\delta^*, 1)$, $\vec{\sigma}^{\text{gen}}$ sustains cooperation in general one-shot pairwise exchanges.*

Proof. $\vec{\sigma}^{\text{gen}}$ only requires one round per stage and agents always exchange their values if no agent deviates. So, we only have to show that $\vec{\sigma}^{\text{gen}}$ is a \mathcal{G}^* -OAPE provided that the adversary is restricted by connectivity and δ is sufficiently close to 1. Fix $G \in \mathcal{G}^*$, agent i , stage t , information set $I_i \in \mathcal{I}_i(G)$ from stage t , $h \in I_i$, and actions a_i^*, a_i' such that $\sigma_i^{\text{gen}}(a_i^* | I_i) > 0$. Let $\Delta = u_i(\vec{\sigma}^1 | h) - u_i(\vec{\sigma}^2 | h)$, where $\vec{\sigma}^1 = \vec{\sigma}^{\text{gen}}|_{I_i, a_i^*}$ and $\vec{\sigma}^2 = \vec{\sigma}^{\text{gen}}|_{I_i, a_i'}$. We show that $\Delta \geq 0$. By the one-shot deviation property (Appendix A), it follows that $\vec{\sigma}^{\text{gen}}$ is a \mathcal{G}^* -OAPE.

Given stage t' , let $\Delta^{t'} = u_i^{t'}(\vec{\sigma}^1 | h) - u_i^{t'}(\vec{\sigma}^2 | h)$, where $u_i^{t'}$ is as in the proof of Theorem 14. We prove six facts first, which hold for every run of $\vec{\sigma}^1$ and $\vec{\sigma}^2$ in $\mathcal{R}(h)$. Fix agents j and l different from i , and stages t', t'' :

- *Fact 1:* If i follows a_i' and $t' \geq t$, then $\text{acc}_i^{t'}(t, j, i)$ is an accurate report of the interaction between i and j in stage t : (1) if i and j do not interact or it is false that $(j, t) \rightsquigarrow_i^G(l, t'+1)$, then the report says that they did not interact; (2) if i and j interact and $(j, t) \rightsquigarrow_i^G(l, t'+1)$, then $\text{acc}_i^{t'}(t, j, i) = \text{True}$ iff i does not send a valid message to j in a_i' .

Proof. At the end of stage t , the report of l says that i and j interacted iff $l = j$ and i and j are neighbours. This is exactly the case when $(j, t) \rightsquigarrow_i^G(l, t+1)$. Moreover, (2) follows directly by construction of $\vec{\sigma}^{\text{gen}}$. In every subsequent stage t' , agent l updates $\text{acc}_i^{t'}(t, j, i)$ according to the reports different from \perp that it receives from other agents. Whether $\text{acc}_i^{t'}(t, j, i) = \perp$, if l updates the report to a value $v \neq \perp$, then l receives v from l' such that $\text{acc}_{l'}^{t'-1}(t, j, i) = v$. By the hypothesis, v is accurate, and $(j, t) \rightsquigarrow_i^G(l', t')$, which implies that $(j, t) \rightsquigarrow_i^G(l, t'+1)$ and $\text{acc}_i^{t'+1}(t, j, i)$ is accurate, as we intended to prove. \square

- *Fact 2:* If $t' = t + n - 1$ and agents use $\vec{\sigma}^2$, then $\text{pnd}_i^{t'}(t', i) = y + \max(x - \text{deg}_i^t, 0)$ after the update phase, where x is the maximum of $\text{pnd}_o^t(t-1, i)$ for all agents $o \neq i$, and y is the number of neighbours to which i omits a message in stage t .

Proof. It is easy to show using induction that for every stage $t'' \in \{t \dots t'\}$, $\text{pnd}_i^{t''}(t, i) = x^{t''}$ at the end of stage t'' , where $x^{t''}$ is the maximum of $\text{pnd}_o^t(t-1, i)$ for all agents $o \neq i$ such that $(o, t) \rightsquigarrow_i^G(l, t''+1)$. This is clearly true for $t'' = t$, since only l is causally influenced by l between t and $t+1$. In every other stage t'' , l sets $\text{pnd}_i^{t''}(t, i)$ to the maximum of the values that other agents send to l in stage t'' , which by the induction hypothesis is $x^{t''}$. Since the adversary is restricted by connectivity with knowledge of degrees, l is

causally influenced by every agent without interference from i between t and $t + n - 1$, so $pnd_i^{t'}(t, i) = x$ after the receive phase of stage t' . In the update phase, l computes the value \deg_i^t and the number y of deviations of i in stage t (by Fact 1, this information is accurate), and sets $pnd_i^{t'}(t', i) = \min(0, x - \deg_i^t) + y$. This concludes the proof. \square

- *Fact 3:* If $t'' \neq t$ and $t' \geq t$, then the values of $pnd_i^{t'}(t'' - 1, i)$ and $acc_i^{t'}(t'', j, i)$ are constant whether agents use $\vec{\sigma}^1$ or $\vec{\sigma}^2$.

Proof. If $t' = t$ and $t'' < t$, then those values depend only on the values received by l from agents different from i in stage t , which are determined by the history h . Continuing inductively, for every stage $t' > t$, l updates $pnd_i^{t'}(t'' - 1, i)$ and $acc_i^{t'}(t'', j, i)$ according to the values that agents $l' \neq i$ send to j in stage t' , which are the same by the hypothesis, so the values of $pnd_i^{t'}(t'' - 1, i)$ and $acc_i^{t'}(t'', j, i)$ are constant, whether agents use $\vec{\sigma}^1$ or $\vec{\sigma}^2$. If $t'' > t$, then l does not update $pnd_i^{t'}(t'' - 1, i)$ and $acc_i^{t'}(t'', j, i)$ if $t' < t'' + 1$. If $t' = t'' + 1$, l updates $pnd_i^{t'}(t'' - 1, i)$ basing on the values $pnd_i^{t'}(t' - n, i)$ and $acc_i^{t'}(t' - n + 1, j, i)$, which are constant by the same arguments of the proof of the case $t'' < t$. If $t' = t''$, then l updates $acc_i^{t'}(t'', j, i)$ iff l interacts with j in stage t'' and updates it to *False*, since i follows σ_i^{gen} after t , whether agents use $\vec{\sigma}^1$ or $\vec{\sigma}^2$. Finally, if $t' > t''$, the result follows directly by the same arguments used in the case where $t'' < t$ and $t' > t$. \square

- *Fact 4:* If $t' > t$, then the value $pnd_i^{t'}(t', i)$ at the end of the update phase of t' is at least as high when agents use $\vec{\sigma}^1$ as when agents use $\vec{\sigma}^2$.

Proof. Let $t'' = t' - n + 1$. The value $pnd_i^{t'}(t', i)$ is computed from $pnd_i^{t'}(t'' - 1, i)$ and $acc_i^{t'}(t'', l, i)$ for all $l \neq i$. If $t'' \neq t$, then, by Fact 3, the values are constant, whether agents use $\vec{\sigma}^1$ or $\vec{\sigma}^2$. If $t'' = t$, then by Fact 2 and the fact that i sends all messages required by σ_i^{gen} in a_i^* , the value $pnd_i^{t'}(t - 1, i)$ at the end of the update phase of stage t' is higher by y if agents use $\vec{\sigma}^2$ than if they use $\vec{\sigma}^1$, where y is the number of neighbours of i to which i does not send a valid message in a_i' , so the fact holds for $t' = t + n - 1$. Since all agents follow $\vec{\sigma}^{\text{gen}}$ in stages $t'' > t$, Fact 4 follows immediately by Fact 2 for all stages $t' = t + cn - 1$ with $c > 0$. This concludes the proof. \square

- *Fact 5:* If $t' > t$, then $\Delta^{t'} \geq 0$.

Proof. In stage t' , both i and its neighbours send messages of fixed size, so the costs of sending and receiving messages are constant for i whether agents use $\bar{\sigma}^1$ or $\bar{\sigma}^2$. By Fact 4, i is never punished with higher probability if agents use $\bar{\sigma}^1$ than if they use $\bar{\sigma}^2$, so the expected benefits are at least as high when agents use $\bar{\sigma}^1$. This implies that $\Delta^{t'} \geq 0$. \square

- *Fact 6:* If i does not send valid messages to exactly y neighbours in a'_i , then $\sum_{t < t^* \leq t+n^2} \Delta^{t^*} \geq \beta_i y$.

Proof. Given a constant $c \geq 0$, let $t^c = t + cn$, and let $pn d_l^{\bar{\sigma}^1}(t', i)$ and $pn d_l^{\bar{\sigma}^2}(t', i)$ denote the values of the variable $pn d_l^{t'-1}(t' - 1, i)$ at the end of the update phase of stage $t' - 1$ when agents use $\bar{\sigma}^1$ and $\bar{\sigma}^2$, respectively. By Fact 2, if i does not send valid messages to y neighbours in a'_i , we have $pn d_l^{\bar{\sigma}^2}(t^1, i) - pn d_l^{\bar{\sigma}^1}(t^1, i) = y$, and l deducts $\deg_i^{t^c}$ from $pn d_l^{t^c+n-1}(t^c, i)$ in the update phase of stage $t^c + n - 1$ for all $c \geq 1$, until $pn d_l^{t^c+n-1}(t^c, i)$ gets the value 0. It is easy to see that if c^* is the smallest constant such that $pn d_l^{\bar{\sigma}^1}(t^{c^*}, i) < \deg_i^{t^{c^*}}$, then $pn d_l^{\bar{\sigma}^1}(t^{c^*+1}, i) = 0$ and we still have $pn d_l^{\bar{\sigma}^2}(t^{c^*}, i) - pn d_l^{\bar{\sigma}^1}(t^{c^*}, i) = y$, so i expects to be punished an additional y times in stages $t^{c'}$ for $c' \geq c^*$ if agents use $\bar{\sigma}^2$. Since $pn d_l^{\bar{\sigma}^2}(t^1, i) < n$ and i has at least one neighbour in every stage, we must have $pn d_l^{\bar{\sigma}^2}(t^{n+1}, i) = 0$, so the additional punishments happen in stages $t + n, t + 2n, \dots, t + n^2$. By the same arguments of the proof of Fact 5, it follows that the expected utility of i decreases at least by $y\beta_i$ in those stages if agents use $\bar{\sigma}^2$ instead of $\bar{\sigma}^1$, whereas the utility does not increase in every other stage. The result follows immediately. \square

We can now show that $\Delta \geq 0$. Suppose that i does not send valid messages to exactly y neighbours in a'_i . If $y = 0$, then $\Delta^t \geq 0$ and, by Fact 5, $\Delta \geq 0$. If $y > 0$, then i saves at most the cost y of sending messages in a'_i , but by Fact 6 agent i loses at least $\delta^{n^2} y\beta_i$ in future stages. Hence, we have $\Delta \geq y(\delta^{n^2} \beta_i - 1)$. Since $\beta_i > 1$ in general pairwise exchanges, if δ is sufficiently close to 1, then $\delta^{n^2} \beta_i \geq 1$ and $\Delta \geq 0$. This concludes the proof. \square

6.3.4 Avoiding Prior Knowledge of Degree

The knowledge of degree can be a restrictive assumption in practice. We now discuss an extension of $\bar{\sigma}^{\text{gen}}$ that does not require this assumption, at the expense of requiring three rounds per stage. The idea is that agents reveal their degrees first and then exchange values. It may

seem that two rounds are sufficient for this. Unfortunately, this is not the case, because agents can lie about their degrees. In particular, an agent i may declare a higher degree to a neighbour j to decrease the probability of being punished by j (recall that j punishes i with a probability inversely proportional to the degree of i). We can address this problem by including in the report information about the declared degrees, so that an agent that lies about its degree is caught and punished. This is still not sufficient though, because of the following scenario. Suppose that i only has one neighbour j in stage t and only one pending punishment. If i does not lie, then j punishes i with probability 1. If i declares a degree of $n - 1$ instead, then the probability of being punished by j is only $1/(n - 1)$. This yields an increase in the expected benefits of stage t from 0 to $\beta_i(n - 2)/(n - 1)$. In addition, suppose that i defects j by omitting its value. If i omits its value to j in addition to lying about its degree, then the total gain in stage t can be close to $1 + \beta(n - 2)/(n - 1)$, whereas the future loss is β_i . Since we only assume that $\beta_i > 1$, the loss may not outweigh the gain. The problem is that i reveals the degree before incurring the cost of sending the value. We can avoid this problem in the same way we relaxed the assumptions about valuable pairwise exchanges in Section 6.2.3, i.e., by using the technique employed in (Li et al. 2006).

In more detail, we need three communication rounds per stage. In round 1, agents exchange monitoring information and the values ciphered with random private keys. In round 2, they reveal the degrees. Finally, in round 3, they decide whether to cooperate by sending the private keys or to punish by sending arbitrary keys, with the same probability as in $\bar{\sigma}^{\text{gen}}$. Agents i and j only exchange messages in round m if they both send valid messages in all rounds $m' < m$; if j sends an invalid message to i or omits a message, then i emits an accusation against j (note that this is true for all rounds $1 \leq m \leq 3$). Let α_i^κ denote the cost that i incurs for sending a key. Note that i can lie about its degree only if i sends valid messages in rounds 1 and 2. The arguments that this extended protocol still sustains cooperation in general pairwise exchanges are the same as those of the proof Theorem 18 for the cases where i does not lie about its degree but may send invalid messages or omit messages to y neighbours: i gains at most y in stage t but loses $\delta^{n^2}y\beta_i$ due to future punishments, so if $\beta_i > 1$ and δ is sufficiently close to 1, then i does not gain. If i lies about its degree to y neighbours, then i gains at most $y(\beta(n - 2)/(n - 1) + \alpha_i^\kappa)$ (i may omit the keys in round 3), whereas the future loss is at least $\delta^{n^2}y\beta_i$. If $\alpha^\kappa < \beta/(n - 1)$, then the gain is less than $y\beta_i$, so again the future loss outweighs the gain of stage t if δ is sufficiently close to 1.

6.3.5 Complexity

We now discuss how to extend $\vec{\sigma}^{\text{gen}}$ to improve the complexity by assuming further restrictions on \mathcal{G}^* and that agents are computationally bounded. The bit complexity of $\vec{\sigma}^{\text{gen}}$ regarding each message sent in every stage t is $O(n^3)$: each message contains n^3 reports, with one report per pair of agents and stage $t' \in \{t - n \dots t - 1\}$, and contains n^2 numbers of pending punishments. The maximum delay of punishments is $O(n^2)$. We can improve both the complexity and the delay due to the following observation: the factors n^3 and n^2 are a result of (1) the fact that agents have to wait n stages before punishing an agent for a deviation and (2) the fact that agents have to forward reports and numbers of pending punishments relative to every other agent. Therefore, we can improve complexity by decreasing the waiting time for agents to punish any agent and by decreasing the number of agents relative to which agents forward reports and numbers of pending punishments. However, we need further restrictions on \mathcal{G}^* to do this.

Specifically, we can improve both the waiting time and the quantity of information included in each message if \mathcal{G}^* is restricted as follows: there exist constants c, ρ such that for all $G \in \mathcal{G}^*$, stage t , and agent i , (a) every neighbour of i in G^t causally influences every neighbour l of i in $G^{t+\rho}$, (b) if S_i^t is the set of agents j such that $(j, t - \rho) \rightsquigarrow^G (i, t)$, then $\#S_i^t \leq c$, and (c) every agent i always has at least one neighbour. These restrictions are characteristic of networks that have a high clustering coefficient, e.g., small world networks such as social networks (Holland & Leinhardt 1971; Watts & Strogatz 1998).

We now discuss a modified version $\vec{\sigma}^*$ of the protocol $\vec{\sigma}^{\text{gen}}$ that has a lower complexity and lower delay of punishments. We assume that agents have unforgeable signatures. Protocol $\vec{\sigma}^*$ is identical to $\vec{\sigma}^{\text{gen}}$ in that agent i still sends messages in every stage t containing reports, numbers of pending punishments, and values, and punishes each neighbour with the same probability as in $\vec{\sigma}^{\text{gen}}$. Protocol $\vec{\sigma}^*$ differs from $\vec{\sigma}^{\text{gen}}$ in that (i) every report $acc_i^t(t', j, l)$ that i sends must be signed by agent j , (ii) every number of pending punishments $pnd_i^t(t', j)$ must be signed by some agent $l \neq j$, and (iii) punishments are applied in cycles with period ρ , so agent i calculates the number of pending punishments $pnd_i^t(t, j)$ of agent j basing on the reports $acc_i^t(t - \rho + 1, l, j)$ for every $l \neq j$ and the number of pending punishments $pnd_i^t(t - \rho, j)$.

By the aforementioned restrictions on \mathcal{G}^* , every agent i has at most c neighbours, thus for each stage t' , agents may only send numbers of pending punishments $pnd_i^t(t', j)$ and reports

$acc_i^t(t', i, j)$ signed by i for at most c different agents j . Since every agent j is causally influenced by at most c agents in ρ rounds, if agents cannot forge signatures, then j only receives c^2 reports and numbers of pending punishments signed by different agents in stage t relative to each stage $t' \in \{t - \rho \dots t - 1\}$. Therefore, agents only need to send ρc^2 reports and numbers of pending punishments in each message, so the bit complexity of $\vec{\sigma}^*$ is $O(\rho c^2)$. In addition, note that if an agent i omits messages to a neighbour j in a stage t , then all neighbours of i in stages $t + \rho, t + 2\rho, \dots$ know of this deviation and punish i one additional time; since i has always at least one neighbour and at most c neighbours in each stage, by the same arguments of the proof of Theorem 18, i is punished one additional time for deviating towards j no later than stage $t + c\rho$. Hence, the delay of punishments in $\vec{\sigma}^*$ is $O(\rho c)$. If ρ and c are both sublinear on n , then both the bit complexity and the delay of punishments are also sublinear on n , so $\vec{\sigma}^*$ is significantly more scalable than $\vec{\sigma}^{\text{gen}}$. The same arguments of the proof of Theorem 14 also show that $\vec{\sigma}^*$ sustains cooperation under the aforementioned restrictions on \mathcal{G}^* : agents never gain by lying about monitoring information, and if agent i omits messages in stage t to some agent j , then i is punished at least once no later than stage $t + c\rho$, so i also does not gain by omitting messages to j if δ is sufficiently close to 1.

Summary

We have introduced a new game theoretical model of repeated games played on dynamic networks and a novel notion of equilibrium for this model. Then, we used this model to analyse infinitely repeated pairwise exchanges. In this analysis, we have identified necessary and sufficient restrictions on the set of dynamic networks to sustain cooperation. Specifically, we identified a tradeoff between restrictions on the network, restrictions on the utility, knowledge that agents have about the topology, and types of punishments used by the protocols.

First, we have shown that it is not possible to sustain cooperation if the set \mathcal{G}^* of evolving graphs that the adversary may generate is not restricted by weakly timely punishments; we also showed that no protocol sustains cooperation in one-shot pairwise exchanges if the set \mathcal{G}^* is not restricted by strong timely punishments. Our second result provided a protocol that sustains cooperation in valuable pairwise exchanges if agents can send more than one message per stage, and \mathcal{G}^* is restricted by strong timely punishments. This shows that the restriction of strong timely punishments is almost tight to sustain cooperation in many interactions of interest

that can be modelled as valuable pairwise exchanges, e.g., secret-sharing of small but valuable secrets (Halpern & Teague 2004; Abraham et al. 2006). Another consequence of these results is that it is not possible to sustain cooperation in certain networks formed in file-sharing systems such as BitTorrent (Cohen 2003), where agents with different interests interact only rarely.

Next, we considered the problem of sustaining cooperation in general one-shot pairwise exchanges. We have identified multiple problems with protocols that are not symmetric, and we have identified a restriction on dynamic networks named eventual distinguishability that is necessary to sustain cooperation with symmetric and bounded protocols. Finally, we have shown that it is possible to sustain cooperation with symmetric and bounded protocols provided that the network is always connected and agents know the degree of their neighbours. These results provide a confirmation of the experimental results by Li et al. (2006, 2008).

For future work, it would be interesting to prove stronger impossibility results that would close the gaps between weak and strong timely punishments in pairwise exchanges with multiple rounds and between eventual distinguishability and connectivity with knowledge of degree in general one-shot pairwise exchanges. Another open issue is collusion. We believe that both the necessary and sufficient restrictions presented in this chapter could be strengthened by generalizing the notion of causal influence without interference from individual agents to the absence of interference from members of a coalition. Given these restrictions, the protocols presented in this chapter are resilient to collusion.

In the next chapter, we present our results in the problem of consensus with rational behaviour and crashes.

Fair Consensus with Crashes

We now address the problem of fair consensus with rational behaviour and crashes. We show that (1) there is no fair consensus protocol that is an f -Nash equilibrium if $f \geq 1$, (2) there is a fair consensus protocol that is a π -Nash equilibrium if $n > f + 1$ and π satisfies minimum properties, and (3) there is a fair consensus protocol that is a π -sequential equilibrium under the same assumptions about n and π .

In the impossibility proof that there is no f -Nash equilibrium, we show that if the context (F, \vec{v}) has a specific form, then some agent i can take advantage of knowing this to increase the probability of obtaining consensus on its preferred value. More precisely, we show that there is always a context (F, \vec{v}) , a round m , and an agent j that crashes in round m , such that j sends a round- m message to i , but if i deviates by pretending that it did not receive that message from j , then the probability of deciding on i 's input increases, and so the expected utility of i also increases.

We define a fair consensus protocol $\vec{\sigma}^{nec}$ that is a π -Nash equilibrium if $n > f + 1$ and π satisfies minimum properties. The protocol runs in $f + 1$ rounds. In each round, agents exchange their inputs, reports of crashes, and random numbers. Agents use information about crashes to agree on a set S of nonfaulty agents and their inputs, and they use the random numbers to elect a leader at random among agents in S whose input is selected as the final decision. $\vec{\sigma}^{nec}$ provides two types of incentives for agents to not deviate:

1. *Threat of punishment to agents that send messages inconsistent with the protocol:* every agent i checks for messages inconsistent with all agents following the protocol or crashing; if an inconsistency is detected, then i ensures that there is no consensus by deciding Ψ , thus punishing the agent that deviated (recall that we assume that agents prefer to reach consensus than not to).
2. *Threat of punishment to agents that pretend to crash:* the inputs of faulty agents are

excluded from the decision; therefore, agents that pretend to crash are punished by having their inputs decided with lower probability.

We prove that $\vec{\sigma}^{nec}$ satisfies the five properties of fair consensus assuming that at most f agents fail and that $n > f + 1$, but we make no assumptions about π . Then, we prove that $\vec{\sigma}^{nec}$ is a π -Nash equilibrium assuming that agents care only about consensus, that $n > f + 1$, and that π satisfies two properties, namely, π *supports reachability* and π *is uniform*. Roughly speaking, we say that π supports reachability if it attributes small probability to particular failure patterns that prevent information from one agent reaching an agent that has not crashed by the end of the protocol; we say that π is uniform if it attributes equal probability to equivalent failures of different agents. We believe that these assumptions apply in many practical systems; we discuss this further in Section 7.3. To prove that $\vec{\sigma}^{nec}$ is a π -Nash equilibrium, we show that if an agent i deviates from $\vec{\sigma}^{nec}$ by following an alternative strategy σ_i , then the expected utility of i does not increase, for all possible deviations of i in σ_i . In the proof, we enumerate all the ways in which i can deviate.

Finally, we prove that we can obtain a π -sequential equilibrium with minimal changes on $\vec{\sigma}^{nec}$. To prove this result, we define a belief system μ^{sec} that is consistent with $\vec{\sigma}^{nec}$ and satisfies the following property: if an agent i detects an inconsistency, then i believes with probability 1 that consensus will not be reached, such that agent i has no incentives to deviate by not deciding Ψ . Therefore, the threats of punishment used in $\vec{\sigma}^{nec}$ are credible with μ^{sec} .

We now prove each of the results in turn. Table 7.1 summarizes the most important notation used in this chapter. Since we only consider one stage of consensus, we drop all references to stages in the notation.

7.1 An Impossibility Result

We now prove our main impossibility result that there is no f -Nash equilibrium protocol that solves fair consensus with crashes.

Theorem 19. *If $\vec{\sigma}$ solves fair consensus, agents care only about consensus, and $f \geq 1$, then $\vec{\sigma}$ is not an f -Nash equilibrium*

Proof. Consider the initial configuration \vec{v} where all agents but i have initial preference 0 and i has initial preference 1. If F^1 is the failure pattern where no agent fails, by Fairness, the agents must decide 1 with positive probability in context (F^1, \vec{v}) . It follows that there must be a failure pattern F^2 where only agent i fails but the agents decide 1 with positive probability in context (F^2, \vec{v}) . (In F^2 , i fails only after a decision has been made in F^1 .) If F^0 is the failure pattern where only i fails, and i fails immediately, before sending any messages, then it is clear that no agents can distinguish this context from one where all agents have initial preference 0, so all agents must decide 0, by the Validity requirement.

Put a partial order \leq on failure patterns where only i crashes by taking $F \leq F'$ if either i crashes in an earlier round in F than in F' , or i crashes in the same round m in both F and F' , but the set of agents to whom i sends a message in F is a subset of the set of agents to whom i sends a message in F' . Clearly $F^0 < F^2$. Thus, there exists a minimal failure pattern F^* such that $F^0 < F^* \leq F^2$, only i fails in F^* , the consensus is on 1 with positive probability in context (F^*, \vec{v}) , the consensus is 0 with probability 1 in all contexts (F, \vec{v}) where only agent i fails in F and $F < F^*$. We can assume without loss of generality that i sends a message to some agent j in the round m in which i fails. To see this, note that if i crashes in the first round then i must send a message to some agent (otherwise $F^* = F^0$ and the decision is 0 with probability 1). And if i crashes in round $m > 1$, we have assumed that i sends at least one message before crashing (recall that we identify an agent crashing at round $m > 1$ and sending no messages with the agent crashing at round $m - 1$ and sending to all agents).

Now suppose that an agent j that receives a message from i in round m pretends not to receive that message. This makes the situation indistinguishable from the context (F, \vec{v}) where F is just like F^* except that i does not send a message to j in round m . Since $F^0 \leq F < F^*$, the decision must be 0 with probability 1 in context (F, \vec{v}) . Since j has initial preference 0 in \vec{v} , j can increase its expected utility by this pretence, so $\vec{\sigma}$ is not an f -Nash equilibrium. \square

7.2 Obtaining a π -Nash equilibrium

We now prove a positive result. If we are willing to assume that there is a distribution π on contexts with some reasonable properties, then we can get a fair π -Nash equilibrium. But, as we show below, there are some subtle problems in doing this.

Before discussing these problems, it is useful to recall some results from social choice theory. Consider a setting with n agents where each has a preference order (i.e., a total order) over some set O of outcomes. A *social-choice function* is a (possibly randomized) function that maps a profile of preference orders to an outcome. For example, we can consider agents trying to elect a leader, where each agent has a preference order over the candidates; the social-choice function chooses a leader as a function of the expressed preferences. A social-choice function is *incentive compatible* if no agent can do better by lying about its preferences. The well-known *Gibbard-Satterthwaite* theorem (Gibbard 1973; Satterthwaite 1975) says that if there are at least three possible outcomes, then the only incentive-compatible deterministic social-choice function f is a *dictatorship*; i.e., the function f just chooses a player i and takes the outcome to be i 's most-preferred candidate, ignoring all other agents' preferences. Gibbard (1977) extends this result to show that if there are at least three outcomes, then the only randomized incentive-compatible social-choice function is a *random dictatorship*, which essentially amounts to choosing some player i according to some probability distribution and then choosing i 's value.

Bei et al. (2012) point out that a strategy profile that solves consensus can be viewed as a social-choice function: agents have preferences over three outcomes, 0, 1, and Ψ , and the consensus value (or Ψ , if there is no consensus) can be viewed as the outcome chosen by the function. A strategy profile that is a Nash equilibrium is clearly incentive-compatible; no agent has an incentive to lie about its preferences. Thus, it follows from Gibbard's (1977) result that a solution to rational consensus must be a randomized dictatorship. And, indeed, our protocols can be viewed as implementing a randomized dictatorship: one agent is chosen at random, and its value becomes the consensus value. However, implementing such a randomized dictatorship in our setting is nontrivial because of the possibility of failures¹.

7.2.1 A Naive Protocol

We start with a protocol that, while not solving the problem, has many of the essential features of our solution, and also helps to point out the subtleties. Consider the following slight variant of one of the early protocols for consensus (Dolev & Strong 1982). In round 1,

¹We remark that Theorem 1 of Bei et al. (2012) claims that, given a fixed failure pattern, a strategy profile for consensus that is a Nash equilibrium must implement a dictatorship, rather than randomized dictatorship. While this is true if we restrict to deterministic strategies, neither we nor Bei et al. do so. We have not checked carefully whether results of Bei et al. that depend on their Theorem 1 continue to hold once we allow for randomized dictatorships.

each agent i broadcasts a tuple $(i, v_i, x_{i0}, \dots, x_{if})$, where v_i is i 's initial preference, and x_{it} is a random element in $\{0, \dots, n - t\}$. For round $2, \dots, f + 1$, each agent i broadcasts all the tuples (j, v_j, \vec{x}_j) that i received and did not already forward in earlier rounds. At the end of round $f + 1$, each agent checks for consistency; specifically, it checks that it has received tuples from at least $n - f$ agents and that it has not received distinct tuples claimed to have been sent by some agent j . If i detects an inconsistency, then i decides Ψ . Otherwise, suppose that i received tuples from $n - t$ agents. Then i computes the sum mod $n - t$ of the values x_{jt} for each agent j from which it received a tuple. If the sum is S , then i decides on the value of the agent with the $(S + 1)$ st highest id among the $n - t$ agents from which it received tuples. (Here is where we are implementing the random dictatorship.) Note that the random value x_{jt} is used by i in computing the consensus value if exactly t faulty agents are discovered; the remaining random values sent by agent j in the first round are discarded.

It is straightforward to check that if all nonfaulty agents follow this protocol, then they will all agree on the set of tuples received (see the proof of Theorem 20 for an argument similar in spirit), and so will choose the same decision value, and each agent whose value is considered has an equal chance of having their value determine the outcome. But this will not be in general a π -Nash equilibrium if π allows up to f failures, that is, π puts probability 0 on all failure patterns that have more than f failures and $f \geq 2$.

Consider a distribution π that puts positive probability on all contexts with at most f failures, and an initial configuration where agent 1 prefers 1, but all other agents prefer 0. Agent 1 follows the protocol in the first round, and receives a message from all the other agents. We claim that agent 1 may have an incentive to pretend to fail (without sending any messages) at this point. Agent 1 can gain by doing this if one of the other agents, say agent 2, crashed in the first round and sent a message only to agent 1. In this case, if 1 pretends to crash, no other agent will learn 2's initial preference, so 1's initial preference will have a somewhat higher probability (at least $\frac{1}{n-1} - \frac{1}{n}$) of becoming the consensus decision. Of course, there is a risk in pretending to crash: if f agents really do crash, then an inconsistency will be detected, and the decision will be Ψ . Let $\alpha_{<f}$ be the probability of there being fewer than f failures and at least one agent crashing in the first round who does not send to any agent other than 1 (this is the probability that 1 gains some utility by its action); let $\alpha_{=f}$ be the probability of there being f crashes other than 1 (this is an upper bound on the probability that 1 loses utility by its action).

Then 1's expected gain by deviating is at least

$$(\beta_{0i} - \beta_{1i}) \left(\frac{1}{n-1} - \frac{1}{n} \right) \alpha_{<f} - (\beta_{0i} - \beta_{2i}) \alpha_{=f}.$$

This is a small quantity. However, if f is reasonably large and failures are unlikely, we would expect $\alpha_{=f}$ to be much smaller than $\alpha_{<f}$, so as the number f of failures that the protocol is designed to handle increases, deviating becomes more and more likely to produce a (small) gain.

7.2.2 A π -Nash equilibrium

There are three problems with the preceding protocol. The first is that, even if 1 pretends to fail, 1's value will be considered a potential consensus value, since everyone received the value before 1 failed. This means that there is little downside in pretending to fail. Roughly speaking, we deal with this problem by taking into consideration only the values of nonfaulty agents when deciding on a consensus value. The second problem is that since agents learn the random values (x_{i0}, \dots, x_{if}) that will be used in determining the consensus value in round 1, they may be able to guess with high probability the value that will be decided on at a point when they can still influence the outcome. To address this problem, agents do not send these random values in the first round; instead, they use secret sharing (Shamir 1979), so as to allow the nonfaulty agents to reconstruct these random values when they need to decide on the consensus value. This prevents agents from being able to guess with high probability what the decision will be too early. The third problem is that in some cases agents can safely lie about the messages sent by other agents (e.g., i can pretend that another agent did not crash). We could solve this by assuming that messages can be signed using unforgeable signatures. We do not need this or any other cryptographic assumption. Instead, we use some randomization to ensure that if an agent lies about a message that was sent, it will be caught with high probability.

Thus, in our algorithm, an agent i generates random numbers for two reasons. The first is that it generates $f+1$ random numbers (x_{i0}, \dots, x_{if}) , where x_{it} is used in choosing the consensus value if there are exactly t faulty agents discovered, and then, as we suggested above, shares them using secret sharing, so that the numbers can be reconstructed at the appropriate time (see below). The second is that it generates $n-1$ additional random numbers, denoted $z_{ij}^m[i]$, one for each agent $j \neq i$, in each round m , and sends them to j in round m . Then if agent j

claims that it got a message in round m from i , it will have to also provide $z_{ij}^m[i]$ as proof.

In more detail, we proceed as follows. Initially, each agent i generates a random tuple (x_{i0}, \dots, x_{if}) , where x_{it} is in $\{0, \dots, n - t\}$. It then computes $f + 1$ random polynomials q_{i0}, \dots, q_{if} , each of degree 1, such that $q_{it}(0) = x_{it}$. It then sends $(q_{i0}(j), \dots, q_{if}(j))$ to agent j . The upshot of this is that no agent will be able to compute x_{it} given this information (since one point on a degree-1 polynomial q_{it} gives no information regarding $q_{it}(0)$). In addition, in round 1, each agent i sends v_i to each agent j , just as in the naive algorithm; it also generates the random number $z_{ij}^1[i]$ and a special random number z , and sends each agent the vector $z_{ij}^1 = (z_{ij}^1[1], \dots, z_{ij}^1[n])$, where $z_{ij}^1[j'] = 0$ for $j' \neq i, j$ and $z_{ij}^1[j] = z$. (As we said, these random numbers form a “signature”; their role will become clearer in the proof.) Finally, in round 1, agent i sends a status report SR_i^1 ; we discuss this in more detail below. In the receive phase of round 1, agent i adds all the values received from other agents to the set ST_i .

In round m with $2 \leq m \leq f$, i again sends a status report SR_i^m and a vector z_{ij}^m . For each agent j , $SR_i^m[j]$ is a tuple of the form (m, x) , where m is the first round that i knows that j crashed ($m = \infty$ if i believes that j has not yet crashed), and x is either the vector z_{ji}^{m-1} of random values sent by j in $m - 1$ (if i believes that j has not yet crashed) or an agent that told i that j crashed in round m . The tuple z_{ij}^m is computed by setting $z_{ij}^m[l]$ for $l \neq i, j$ to be $z_{li}^{m-1}[l]$, the random number sent by l in the previous round (this will be used to prove that i really got a message from l in the previous round—it is our replacement for unforgeable signatures); again, $z_{ij}^m[i]$ is a random value generated by i . In round $f + 1$, i also sends j the secret shares y_{ij}^t it received in round 1 from each agent l (i.e., the value $q_l^t(i)$ that it received from l , assuming that l did not lie). This enables j to compute the polynomials q_{it} , and hence the secret $q_{it}(0) = x_{it}$ for $0 \leq t \leq f$.

If i detects an inconsistency in round $m \leq f + 1$, then i decides Ψ , where i detects an inconsistency in round m if the messages received by i are inconsistent with all agents following the protocol except that up to f agents may crash. This can happen if

1. j sends incorrectly formatted messages;
2. $m = 2$ and agents j' and $j'' \neq i$ disagree about the random values $z_{jj'}^1[j']$ and $z_{jj''}^1[j'']$ sent by j in round 1;
3. $m > 2$ and some agent $j' \neq j$ reports that j sent a value $z_{jj'}^{m-1}[i]$ in round $m - 1$ different

- from the value $z_{ij}^{m-2}[i]$ sent by i to j in round $m - 2$;
4. $m = f + 1$ and it is not possible to interpolate a polynomial q_j^t through the shares y_{ji}^t received by i from j in round 1 and the values y_{jl}^t received from $l \neq j$ in round $f + 1$.
 5. i learns that j crashed in some round m' (i.e., either j omits a message to i in round m' or some agent j' sends i a status report in round m that says that j crashed in m') and sent a message in round $m'' > m'$ (i.e., either i receives a message from j in round $m'' > m'$ or some agent j'' sends i a status report saying that it received a message from j in a round $m'' > m'$);
 6. for some agents j, j' , and j'' , j sends i a status report in round m that says that j'' crashed in round m' and that j' reported this, but j' sends i a status report in round m that says that j'' did not crash before round $m'' > m'$;
 7. for some agents j, j' , and j'' , j sends i a status report in round m that says that j' did not crash by round $m - 1$ and j'' crashed in some round $m' < m$, while j' sends i a status report in round $m - 1$ saying that j'' crashed in round $m'' < m'$ (so either j ignored the report about j'' sent by j' or j' lied to j);
 8. more than f crashes are detected by i by round m (i.e., f or more agents have not sent messages to i or were reported to crash in some round up to and including m).

If agent i does not detect an inconsistency at some round $m \leq f + 1$, i proceeds as follows in round $f + 1$. For each round $1 \leq m \leq f + 1$ in a run r , agent i computes $NC_m(r)$, the set of agents that it believes did not crash up to and including round m . Take $NC_0(r) = \mathcal{N}$ (the set of all agents). Say that round m in run r *seems clean* if $NC_{m-1}(r) = NC_m(r)$. As we show (Theorem 20), if no inconsistency is detected in run r , then there must be a round in r that seems clean. Moreover, we show that if m^* is the first round in r that seems clean to a nonfaulty agent i , then all the nonfaulty agents agree that m^* is the first round that seems clean in r , and they agree on the initial preference of all agents in $NC_{m^*}(r)$, and the random numbers sent by these agents in round 1 messages in run r . The agents then use these random numbers to choose an agent j among the agents in $NC_{m^*}(r)$ and take v_j to be the consensus value.

The pseudocode for the algorithm of $\bar{\sigma}^{nec}$ that implements this idea is given in Alg. 5. Lines 1–14 initialize the values of ST and $SR^1[j]$, as well as the random numbers required in

round 1; that is, i generates $x_i[t]$ and the corresponding polynomial q_i^t used for secret sharing for $0 \leq t \leq f$, and random vectors $(z_{ij}^1[1], \dots, z_{ij}^1[n])$ for $j \neq i$, where $z_{ij}^1[l] \in \{0, \dots, n-1\}$. In the send phase of round m , i sends SR_i^m and z_{ij}^m . If $m = 1$, then i also sends v_i and $(y_{ij}^0, \dots, y_{ij}^f)$ to j , where $y_{ij}^t = q_i^t(j)$; that is, y_{ij}^t is j 's share of the secret x_i^t . Finally, if $m = f + 1$, instead of sending z_{ij}^m to j , i sends all the shares y_{li}^t it has received from other agents, so that all agents can compute the secret (lines 16-21). In phase 2 (the “receive” phase) of round m , i processes all the messages received and keeps track of all agents who have crashed (lines 22-38). If i receives a round m message from j , then i adds (j, v_j) to ST_i if $m = 1$, includes in $SR_i^m[j]$ the vector z sent by j to i , and updates the status report $SR_i^m[l]$ of each agent l . Specifically, if j reports that j' crashed in a round m' and i earlier considered it possible that j' was still nonfaulty at round m' , then i includes in $SR_i^m[l]$ the fact that j' crashed and that j is an agent that reported this fact (lines 29-33); if i does not receive a round m message from j and i believed that j did not crash before, then i marks j as crashed (line 35). In phase 3 (the “update” phase) of round $m \leq f$, i generates the random value $z_{ij}^{m+1}[i]$ for the next round. If i detects an inconsistency, then i decides Ψ (line 41); if no inconsistency is detected by the end of round $f + 1$, then i decides on a value (lines 46-56) by computing the set $NC_{m'}$ for every round m' , determining the earliest round m^* that seems clean ($NC_{m^*} = NC_{m^*-1}$), computing a random number $S \in \{0, \dots, n - t - 1\}$, where t is the number of crashes that occurred before m^* , by summing the random numbers $x_j[t]$ of $j \in NC_{m^*}$ (computed by interpolating the polynomials), and deciding on the value of the agent in NC_{m^*} with the $(S + 1)$ st highest id.

7.2.3 Analysis

We now prove that $\vec{\sigma}^{nec}$ gives a π -Nash equilibrium, under reasonable assumptions about π . We first prove that the protocol satisfies all the properties of fair consensus without making any assumptions about π .

Theorem 20. *$\vec{\sigma}^{nec}$ solves fair consensus if at most f agents crash, $f + 1 < n$, and all the remaining agents follow the protocol.*

Proof. Consider a run r where all agents follow $\vec{\sigma}^{nec}$ and at most f agents crash. It is easy to see that no inconsistency is detected in r . Since an agent crashes in at most one round and there are at most f faulty agents, there must exist a round $1 \leq m \leq f + 1$ when no agent crashes. Let m^*

Algorithm 5 $\sigma_i^{nec}(v_i)$: i 's consensus protocol with initial value v_i

```

1: decided  $\leftarrow False$ 
2:  $ST_i \leftarrow \{(i, v_i)\}$ 
3:  $z \leftarrow \text{random in } \{0 \dots n - 1\}$ 
4: for all  $j \neq i$  do
5:    $SR_i^1[j] \leftarrow (\infty, \perp)$  ▷ All agents are initially active
6:   for all  $l \neq j, i$  do
7:      $z_{ij}^1[l] \leftarrow 0$ 
8:    $z_{ij}^1[i] \leftarrow \text{random in } \{0 \dots n - 1\}$  ▷ Random number to be used by  $j$  in round 2
9:    $z_{ij}^1[j] \leftarrow z$  ▷ Proves that  $i$  sends round 1 message to  $j$ 
10: for all  $0 \leq t \leq f$  do
11:    $x_i[t] \leftarrow \text{random in } \{0, \dots, n - t - 1\}$  ▷ A random number for each possible value of  $t$ 
12:    $q_i^t \leftarrow \text{random polynomial of degree 1 with } q_i^t(0) = x_i[t]$ 
13:   for all  $j \neq i$  do
14:      $y_{ij}[t] \leftarrow q_i^t(j)$ 

15: for all round  $1 \leq m \leq f + 1$  such that  $\neg \text{decided}$  do
16:   Phase 1: send phase
17:   for all  $j \neq i$  do
18:     if  $m = 1$  then Send  $\langle v_i, SR_i^m, (y_{ij}^0, \dots, y_{ij}^f), z_{ij}^m \rangle$  to  $j$ 
19:     if  $2 \leq m \leq f$  then Send  $\langle SR_i^m, z_{ij}^m \rangle$  to  $j$ 
20:     if  $m = f + 1$  then Send  $\langle SR_i^m, (y_{ii}^0, \dots, y_{ii}^f)_{l \neq j} \rangle$  to  $j$ 
21:   EndPhase

22:   Phase 2: receive phase
23:    $SR_i^{m+1} \leftarrow SR_i^m$ 
24:   for all  $j \neq i$  do
25:     if receive valid message from  $j$  then
26:       if  $m = 1$  then  $ST_i \leftarrow ST_i \cup \{(j, v_j)\}$  ▷  $ST_i$  contains all the values that  $i$  has seen
27:        $SR_i^{m+1}[j] \leftarrow (\infty, z_{ij}^m)$  ▷ Note that  $j$  is still active
28:       for all  $l \neq i, j$  do
29:         if  $SR_j^m[l] = (m', j')$  and  $SR_i^m[l] = (m'', j'')$  and  $m' < m''$  then
30:            $SR_i^{m+1}[l] \leftarrow (m', j)$  ▷  $l$  crashed earlier than previously thought
31:            $z_{il}^{m+1}[j] \leftarrow \perp$ 
32:         else if  $SR_j^m[l] = (\infty, z_j^m)$  then
33:            $z_{il}^{m+1}[j] = z_{ji}^m[j]$ 
34:         else if  $SR_i^{m+1}[j] = (\infty, z')$  for some  $z'$  then
35:            $SR_i^{m+1}[j] \leftarrow (m, i)$  ▷  $i$  detects a crash of  $j$ 
36:         for all  $l \neq i$  do
37:            $z_{il}^{m+1}[j] \leftarrow \perp$ 
38:   EndPhase

39:   Phase 3: update phase
40:   if an inconsistency is detected then
41:     Decide( $\Psi$ ) ▷ Punishment
42:     decided  $\leftarrow True$ 
43:   else if  $m \leq f$  then
44:     for all  $j \neq i$  do
45:        $z_{ij}^{m+1}[i] \leftarrow \text{random in } \{0, \dots, n - 1\}$ 
46:   else if decided =  $False$  then
47:      $NC_0 = \mathcal{N}$ 
48:     for all  $1 \leq m' \leq f + 1$  do
49:        $NC_{m'} \leftarrow \{j \in \mathcal{N} \setminus \{i\} \mid \forall m'' \leq m', l, SR_i^{m''+2}[j] \neq (m'', l)\} \cup \{i\}$  ▷ Agents that did not crash up to round  $m'$ 
50:      $m^* \leftarrow \text{first round } m' \text{ such that } NC_{m'} = NC_{m'-1}$  ▷ First round that seems clean
51:      $t \leftarrow n - |NC_{m^*}|$  ▷ Number of crashes prior to  $m^*$ 
52:     for all  $j \in NC_{m^*}$  do
53:        $q_j^t \leftarrow \text{unique polynomial interpolating the values } y_{ji}^t \text{ received}$  ▷ otherwise, an inconsistency was detected
54:        $x_j[t] \leftarrow q_j^t(0)$ 
55:      $S \leftarrow \sum_{j \in NC_{m^*}} x_j[t] \bmod (n - t)$  ▷ Calculate a random number in  $0, \dots, n - t - 1$ 
56:     Decide( $v_j$ ), where  $j$  is the  $(S + 1)$ st highest id in  $NC_{m^*}$ 
57:   EndPhase

```

be the first such round. We prove that for all nonfaulty agents i and j , $NC_m^i(r) = NC_m^j(r)$ for all $m \leq m^*$ (where $NC_m^i(r)$ denotes i 's version of $NC_m(r)$ in run r , and similarly for j). To see this, fix two nonfaulty agents i and j . Agent i adds agent l to $NC_m^i(r)$ iff i receives a message from l in every round $m' \leq m$ of run r , and i receives no status report indicating that l crashed in some round $m' \leq m$. If $m < m^*$, then it must be the case that j also received a message from l in every round $m' < m$ of r and neither received nor sent a status report indicating that l crashed in a round $m' \leq m$; otherwise j would have learned about this crash by round m and would have told i by round $f + 1$ that l was faulty (since j is nonfaulty). Thus, $l \in NC_m^j(r)$. If $m = m^*$, then l sends a round m' message to all agents for all $m' < m^*$; and since no agents fail in round m^* , by assumption, we again have $l \in NC_m^j(r)$. Thus, $NC_m^i(r) \subseteq NC_m^j(r)$; similar arguments give the opposite inclusion.

Note that since no agent crashes in round m^* , it is easy to see that we must have $NC_{m^*}^i(r) = NC_{m^*-1}^i(r)$ for all nonfaulty agents i , so round m^* seems clean. With these observations, we can now prove that $\bar{\sigma}^{nec}$ satisfies each requirement of Fair Consensus in r .

Validity: Since no inconsistency is detected, every agent i decides a value different from Ψ in r . Agent i always finds some round m^* that seems clean, computes a nonempty set $NC_{m^*}(r)$, which includes at least i , and knows the random numbers sent by these agents in round m^* . Since ST_i contains only initial preferences, i decides the initial preference of some agent in $NC_{m^*}(r)$.

Termination and Integrity: Every agent either crashes before deciding or decides exactly once at the end of round $f + 1$.

Agreement: We have shown that all nonfaulty agents i and j agree on $NC_m(r)$ for all $m \leq m^*$. We thus omit the superscripts i and j on $NC_m(r)$ from here on in. Given this, they agree on whether each round $m \leq m^*$ seems clean and thus agree that some $\bar{m} \leq m^*$ is the first round that seems clean in r . Moreover, i and j receive identical round 1 messages from the agents in $NC_{\bar{m}}(r)$. It follows that i adds a tuple (l, v_l) to ST_i for $l \in NC_{\bar{m}}(r)$ iff j adds that tuple to ST_j . Suppose that $|NC_{\bar{m}}(r)| = n - t$. Since $NC_{\bar{m}}(r)$ must include all the nonfaulty agents, we must have $t \leq f$. Clearly, if $l \in NC_{\bar{m}}(r)$, then i and j must receive the values y_{li}^t and y_{lj}^t in round 1 messages sent by l . Agents i and j also receive $y_{l'l'}^t$ from each nonfaulty agent l' . Since there are at least $n - f \geq 2$ nonfaulty agents, and l follows σ_l^{nec} , i and j will be able to interpolate the polynomial q_l^t , and compute $x_l[t] = q_l^t(0)$. Consequently, i and j agree on the

information relevant to the consensus decision, so must decide on the same value.

Fairness: The probability of the initial preference of each agent in $NC_{\bar{m}}(r)$ being decided is $1/|NC_{\bar{m}}(r)|$. Since $|NC_{\bar{m}}(r)| \leq n$, if c nonfaulty agents in $NC_{\bar{m}}$ initially have preference v , then the probability of v being decided is at least $c/|NC_{\bar{m}}(r)| \geq c/n$. Since $NC_{\bar{m}}(r)$ contains all the nonfaulty agents, Fairness holds. \square

It remains to show that $\bar{\sigma}^{nec}$ is a π -Nash equilibrium. We show that $\bar{\sigma}^{nec}$ is a π -Nash equilibrium under appropriate assumptions about π . Specifically, we assume that π *supports reachability* and is *uniform*, notions that we now define.

The reachability assumption has three parts. The first two parts consider how likely it is that some information that an agent j has will reach an agent that will decide on a value; the third part is quite similar, and considers how likely it is that a nonfaulty agent becomes aware that an agent j failed in round m . Of course, the answer to these questions depends in part on whether agents are supposed to send messages in every round (as is the case with $\bar{\sigma}^{nec}$). In the formal definition, we implicitly assume that this is the case. (So, effectively, the reachability assumption is appropriate only for protocols where agents send messages in every round.) Given agents i and $j \neq i$, a round- m information set I_i for i , a failure pattern F compatible with I_i , in that $\mathcal{R}(F) \cap \mathcal{R}(I_i) \neq \emptyset$, and $m' \geq m$, say that a nonfaulty agent $l \neq i$ is *reachable from j without i between rounds m' and $f + 1$ given F* if there is a sequence $j_{m'}, \dots, j_{f+1}$ of agents different from i such that $j = j_{m'}$, for $m'' = \{m', \dots, f\}$, $j_{m''}$ has not failed prior to round m'' according to F , and either does not fail in round m'' or, if $m'' < f + 1$, $j_{m''}$ fails in round m'' but sends a message to $j_{m''+1}$ before failing (i.e., if $(j_{m''}, m'', A) \in F$, then $j_{m''+1} \in A$), and $l = j_{f+1}$.

Note that if j is nonfaulty according to F , then a nonfaulty agent is certainly reachable from j without i between rounds m' and $f + 1$; just take $j_{m'} = \dots = j_{f+1} = j$. But even if j fails in round m' according to F , as long j can send a message to a nonfaulty agent other than i , or there is an appropriate chain of agents, then a nonfaulty agent is reachable from j without i by round $f + 1$. The probability of there being a failure pattern for which a nonfaulty agent is reachable from j without i depends in part on how many agents are known to have failed in I_i ; the more agents are known not to have failed, the more likely we would expect a nonfaulty agent to be reachable from j without i .

We also want this condition to hold even conditional on a set of failure patterns, provided

that the set of failure patterns does not favor particular agents failing. To make this precise, we need a few more definitions. Say that an agent j is *known to be faulty in I_i* if j is faulty in all runs in $\mathcal{R}(I_i)$; thus, j is known to be faulty in I_i if j did not send a message to i at round $m - 1$ according to I_i . Say that a set \mathcal{F} of failure patterns *satisfies the permutation assumption with respect to a set F of failures and an information set I_i* if, for all permutations g of the agents that keep fixed the agents that fail in F or are known to be faulty in I_i , if $F' \in \mathcal{F}$, then so is $g(F')$, where $g(F')$ is the failure pattern that results by replacing each triple $(j, m'', A) \in F'$ by $(g(j), m'', g(A))$. \mathcal{F} *satisfies the permutation assumption with respect to I_i* if \mathcal{F} satisfies it with respect to the empty set of failures and I_i . Let $\mathcal{R}(\mathcal{F}) = \cup_{F \in \mathcal{F}} \mathcal{R}(F)$.

We say that π *supports reachability* if for all agents i , all time- m information sets I_i such that M agents are not known to be faulty in I_i , failure pattern F , and all sets \mathcal{F} of failure patterns that satisfy the permutation assumption with respect to F and I_i , we have that

1. if $j \neq i$ is not known to be faulty in I_i and is not in F , then

$$\begin{aligned} &\pi(\text{no nonfaulty agent } l \neq i \text{ is reachable from } j \text{ without } i \\ &\quad \text{between rounds } m \text{ and } f + 1 \mid \mathcal{R}(I_i) \cap \mathcal{R}(\mathcal{F}) \cap \mathcal{R}(F)) \leq \frac{1}{2M}; \end{aligned}$$

2. if $j \neq i$ is not known to be faulty in I_i and is not in F , then

$$\begin{aligned} &\pi(\text{no nonfaulty agent } l \neq i \text{ is reachable from } j \text{ without } i \\ &\quad \text{between rounds } m - 1 \text{ and } f + 1 \mid \mathcal{R}(I_i) \cap \mathcal{R}(\mathcal{F}) \cap \mathcal{R}(F)) \leq \frac{1}{2M}; \end{aligned}$$

3. if a message from some agent j not in F was received up to and including round $m - 2$ but not in round $m - 1$, then

$$\begin{aligned} &\pi(\text{no nonfaulty agent } l \neq i \text{ is reachable from an agent } j' \neq i \\ &\quad \text{that did not receive a message from } j \text{ in round } m - 1 \text{ without } i \\ &\quad \text{between rounds } m \text{ and } f + 1 \mid \mathcal{R}(I_i) \cap \mathcal{R}(\mathcal{F}) \cap \mathcal{R}(F)) \leq \frac{1}{2M}. \end{aligned}$$

The first two requirements essentially say that if i hears from j in round $m - 1$, then it is likely that other agents will hear from j as well in a way that affects the decision, even if i does not forward j 's information. That is, it is unlikely that j will fail right away, and do so in a way that prevents its information from having an effect. Similarly, the third requirement says that

if i does not hear from j in round $m - 1$ (as reflected in I_i), then it is likely that other agents will hear that j crashed at or before round $m - 1$ even if i does not report this fact.

We next define the notion of uniformity. Given two failure patterns F^1 and F^2 , we say that F^1 and F^2 are *equivalent* if there is a permutation g of the agents such that $F^2 = g(F^1)$. We say that π is *uniform* if, for all equivalent failure patterns F^1 and F^2 and vectors \vec{v} of initial preferences, we have $\pi(F^1, \vec{v}) = \pi(F^2, \vec{v})$. Intuitively, if π is uniform, then the probability of each failure pattern depends only on the number of messages omitted by each agent in each round; it does not depend on the identity of faulty agents.

The following lemma will prove useful in the argument, and shows where the uniformity assumption comes into play. Roughly speaking, the lemma says that if the agents run $\vec{\sigma}^{nec}$, then each agent i 's expected value of its initial preference being the consensus value is just its current knowledge about the fraction of nonfaulty agents that have its initial preference. The lemma's claim is somewhat stronger, because it allows for expectations conditional on certain sets of agents failing.

Before stating the lemma, we need some definitions. Let $\mathcal{R}(D_{\geq m})$ consist of all runs where a decision is made and the first round that seems clean is $m' \geq m$. A set \mathcal{F} of failure patterns, a failure pattern F , a round- m information set I_i for i , and $m' \geq m$ are *compatible* if (a) all the failures in F happen before round m' , (b) $m' \leq f + 1$, and (c) \mathcal{F} satisfies the permutation assumption with respect to I_i and F . Given an agent i and a run r where consensus is reached, let $nc(r)$ be the number of agents who apparently have not crashed in the first round of r that seems clean (i.e., if m is the first clean round in r , then $nc(r) = |NC_m(r)|$), and let $ac(r)$ be the number of these agents in r that have initial preference 1. Given an information set $I_i \in \mathcal{I}_i$ and a failure pattern F , let A_F be the set of agents who are faulty in F ; let A consist of the agents known to be faulty in I_i ; let $n(I_i, F) = n - |A \cup A_F|$; and let $a(I_i, F)$ be the agents not in $A \cup A_F$ that have initial preference 1. Note that nc and ac are random variables on runs (i.e., functions from runs to numbers); technically, $a(I_i, F)$ and $n(I_i, F)$ are also random variables on runs, but $n(I_i, F)$ is constant on runs in $\mathcal{R}(I_i)$, while $a(I_i, F)$ is constant on runs in $\mathcal{R}(I_i)$ if $m \geq 2$, since then I_i contains the initial values of nonfaulty agents.

Lemma 21. *If i is an agent who is nonfaulty at the beginning of round $m \leq f + 1$ and has information set I_i (so that I_i is a round- m information set), F is a failure pattern, $m' \geq m$, \mathcal{F} is a set of failure patterns such that \mathcal{F} , F , I_i , and m' are compatible, π is a distribution that*

supports reachability and is uniform, and $\pi_{\vec{\sigma}^{nec}}(\mathcal{R}(I_i) \cap \mathcal{R}(F) \cap \mathcal{R}(\mathcal{F}) \cap \mathcal{R}(D_{\geq m'})) > 0$, then

$$E[ac/nc \mid \mathcal{R}(I_i) \cap \mathcal{R}(F) \cap \mathcal{R}(\mathcal{F}) \cap \mathcal{R}(D_{\geq m'})] = E[a(I_i, F)/n(I_i, F) \mid \mathcal{R}(I_i)], \quad (7.1)$$

where the expectation is taken with respect to $\pi_{\vec{\sigma}^{nec}}$.

Proof. Let $f' = |A \cup A_F| = n - n(I_i, F)$. For all f'' with $f' \leq f'' \leq f$, let $\mathcal{R}_{f''}$ consists of all runs r where agents are using $\vec{\sigma}^{nec}$ such that exactly f'' agents are viewed as faulty in the first round that seems clean. We claim that, for all f'' , we have

$$E[ac/nc \mid \mathcal{R}_{f''} \cap \mathcal{R}(I_i) \cap \mathcal{R}(F) \cap \mathcal{R}(\mathcal{F}) \cap \mathcal{R}(D_{\geq m'})] = E[a(I_i, F)/n(I_i, F) \mid \mathcal{R}(I_i)].$$

Clearly, (7.1) follows immediately from this claim.

We can calculate the relevant expectations using algebra, but there is an easier way to see that the claim holds. First suppose that $m' > 1$ (so that $a(I_i, F)$ and $n(I_i, F)$ are constants on $\mathcal{R}(I_i)$). If the first clean round occurs at or after m' , then it is easy to see that all the agents in $A \cup A_F$ will be viewed as faulty in that round (by all nonfaulty agents), since all these agents fail before round m' . Note that the set of agents viewed as faulty in the first clean round of run r is completely determined by the failure pattern in r . Moreover, it easily follows from the uniformity assumption, the fact that $\vec{\sigma}^{nec}$ treats agents uniformly, and the fact that \mathcal{F} satisfies the permutation assumption that each set B of cardinality f'' that includes $A \cup A_F$ is equally likely to be the set of agents viewed as faulty in the first clean round of a run in $\mathcal{R}_{f''} \cap \mathcal{R}(I_i) \cap \mathcal{R}(F) \cap \mathcal{R}(\mathcal{F}) \cap \mathcal{R}(D_{\geq m})$.

Consider the following experiment: choose a set B of f'' agents containing $A \cup A_F$ uniformly at random, and then choose one more agent $j \notin B$ at random. Assign a pair (B, j) value 1 if the agent j chosen has initial preference 1 in all runs of I_i ; otherwise, assign it value 0. It is easy to see that the expected value of a pair is precisely $E[ac/nc \mid \mathcal{R}_{f''} \cap \mathcal{R}(I_i) \cap \mathcal{R}(F) \cap \mathcal{R}(\mathcal{F}) \cap \mathcal{R}(D_{\geq m})]$. The f'' agents in B constitute the set of faulty agents. The fact that B is chosen uniformly at random (among sets of cardinality f'' containing $A \cup A_F$) corresponds to the assumption that all choices of B are equally likely. The last agent chosen determines the consensus value; as long as there is at least one nonfaulty agent, the procedure used in runs of $\vec{\sigma}^{nec}$ guarantees that all choices of j are equally likely.

Now switch the order that the choices are made: we first choose a nonfaulty agent not in $A \cup A_F$ uniformly at random and then choose $f'' - |A \cup A_F|$ other agents not in $A \cup A_F$ who will fail uniformly at random. It is clear that there is a one-to-one correspondence between the choices in the first experiment and the second experiment: in corresponding choices, the same set of $f'' - f'$ agents fail and the same other agent is chosen to determine the consensus value. Moreover, corresponding choices are equally likely. With the second experiment, it is immediate that the expected value is $a(I_i, F)/n(I_i, F)$.

If $m' \leq 1$, then the argument is the same, except that the value of (B, j) is chosen according to the distribution of initial preferences of agents $j \notin B$ in runs where the faulty agents are exactly the ones in B . This concludes the proof. \square

Theorem 22 shows that $\vec{\sigma}^{nec}$ is a π -Nash equilibrium, as long as $f + 1 < n$ and π supports reachability and is uniform.

Theorem 22. *If $f + 1 < n$, π is a distribution that supports reachability, is uniform, and allows up to f failures, and agents care only about consensus, then $\vec{\sigma}^{nec}$ is a π -Nash equilibrium.*

Proof. Fix an agent i and a strategy σ_i . We must show that we have

$$u_i(\vec{\sigma}^{nec}) \geq u_i((\sigma_i, \vec{\sigma}_{-i}^{nec})). \quad (7.2)$$

Suppose, by way of contradiction, that (7.2) does not hold. Then i must deviate from σ_i^{nec} at some round m . Consider all the ways that i can deviate in round m that can affect the outcome (we discuss what it means to affect the outcome shortly):

1. i pretends to crash; it does not send messages to some subset of agents in round m (and then does not send messages from then on).
2. $m = 1$ and i sends $(i, 1 - v_i)$ to some agent $j \neq i$ (i.e., i lies about its initial preference to at least one agent).
3. i sends an incorrectly formatted message to $j \neq i$ (i.e., i sends a message that is different in format from that required by $\vec{\sigma}^{nec}$).

4. $m = 1$ and i sends values y_{ij}^t to an agent $j \neq i$ such that there is no polynomial q_i^t of degree 1 that interpolates them all or does not choose the polynomials q_i^t at random.
5. i does not choose some z_{ij}^m appropriately (as specified by $\vec{\sigma}^{nec}$).
6. $m < f + 1$ and i decides on a value in $\{0, 1\}$ in round m or $m = f + 1$ and i decides on an incorrect value on the equilibrium path.
7. $m = f + 1$ and i sends a value $y_{j'i}^t$ to $j' \neq i$ different from the value $y_{j'i}^t$ that i received from j in round 1.
8. i does not send a round $m' < m$ message to some agent j that i does not know at round m to have been faulty in round m' , and sends a round m message to $j' \neq i$.
9. i lies about j 's initial preference to an agent $j' \neq i$; that is, i sends a pair (j, v_j) to j' although there is no pair $(j, v_j) \in ST_i$ or it does not send a pair (j, v_j) to j' although there is such a pair in ST_i .
10. i lies about j 's status to $j' \neq i$; that is, i sends j' a status report \overline{SR}_i^m such that $\overline{SR}_i^m[j] \neq SR_i^m[j]$.

Note that in a deviation of type 8, we did not consider the case where i deviates by not sending a message to j in round m' and then sending a message to j' if i knows that j failed in round m' . In this case, i 's deviation is undetectable, and will not affect the outcome. Clearly if i performs only such undetectable deviations, then σ_i is equivalent to σ_i^{nec} , so we do not need to worry about these deviations.

We consider these deviations one by one, and show that none of them makes i better off. More precisely, we show that if σ_i involves only deviations 1– d on the list above for appropriate choices of d , then (7.2) holds. But even this “brute force” argument requires some care, using a somewhat delicate induction on the number of deviations that i is better off not deviating.

We now prove (7.2). We start with the first type of deviation; that is, suppose that σ_i involves only i pretending to crash and that if I_i^* is a time- m^* information set for i , \mathcal{F} is a set of failure patterns that satisfies the permutation assumption relative to I_i^* , $\pi_{\vec{\sigma}^{nec}}(\mathcal{R}(I_i^*) \cap \mathcal{R}(\mathcal{F})) > 0$, and either there are no deviations in runs in $\mathcal{R}(I_i^*)$ or the first deviation in a run in $\mathcal{R}(I_i^*)$ occurs at or after information set I_i^* , then

$$u_i(\vec{\sigma}^{nec} \mid \mathcal{R}(I_i^*) \cap \mathcal{R}(\mathcal{F})) \geq u_i((\sigma_i, \vec{\sigma}_{-i}^{nec}) \mid \mathcal{R}(I_i^*) \cap \mathcal{R}(\mathcal{F})). \quad (7.3)$$

(7.2) clearly follows from (7.3) by taking I_i^* to be the initial information set and letting \mathcal{F} be the set of all failure patterns compatible with I_i^* .

Given a strategy profile $\vec{\sigma}$, let $\mathcal{R}(\vec{\sigma})$ denote the possible runs of $\vec{\sigma}$. If there are no runs in $\mathcal{R}(\sigma_i, \vec{\sigma}_{-i}^{nec}) \cap \mathcal{R}(I_i^*)$ in which i pretends to fail, then conditional on $\mathcal{R}(I_i^*)$, σ_i and σ_i^{osc} agree, so (7.3) holds. If there are runs in $\mathcal{R}(\sigma_i, \vec{\sigma}_{-i}^{nec}) \cap \mathcal{R}(I_i^*)$ in which i pretends to fail, then we proceed by induction on the number of information sets I_i at or after I_i^* at which i first pretends to crash such that $\pi_{(\sigma_i, \vec{\sigma}_{-i}^{nec})}(\mathcal{R}(I_i^*) \cap \mathcal{R}(\mathcal{F}) \cap \mathcal{R}(I_i)) > 0$. Suppose that i first pretends to crash at some information set I_i that comes at or after I_i^* and $\pi_{(\sigma_i, \vec{\sigma}_{-i}^{nec})}(\mathcal{R}(I_i^*) \cap \mathcal{R}(\mathcal{F}) \cap \mathcal{R}(I_i)) > 0$. Thus, there are no runs in $\mathcal{R}(I_i)$ in which i pretends to crash prior to information set I_i . Let σ'_i be identical to σ_i except that i does not pretend to fail at or after I_i . By (7.3),

$$u_i(\vec{\sigma}^{nec} \mid \mathcal{R}(I_i^*) \cap \mathcal{R}(\mathcal{F})) \geq u_i((\sigma'_i, \vec{\sigma}_{-i}^{nec}) \mid \mathcal{R}(I_i^*) \cap \mathcal{R}(\mathcal{F})).$$

We now show that

$$u_i((\sigma'_i, \vec{\sigma}_{-i}^{nec}) \mid \mathcal{R}(I_i^*) \cap \mathcal{R}(\mathcal{F})) \geq u_i((\sigma_i, \vec{\sigma}_{-i}^{nec}) \mid \mathcal{R}(I_i^*) \cap \mathcal{R}(\mathcal{F})). \quad (7.4)$$

(7.3) follows immediately.

To prove (7.4), since $\mathcal{R}(I_i^*)$ is the union of all the time- m information sets for i that follow I_i^* , it suffices to prove that for all time- m information sets I'_i for i that follow I_i^* , we have

$$u_i((\sigma'_i, \vec{\sigma}_{-i}^{nec}) \mid \mathcal{R}(I'_i) \cap \mathcal{R}(\mathcal{F})) \geq u_i((\sigma_i, \vec{\sigma}_{-i}^{nec}) \mid \mathcal{R}(I'_i) \cap \mathcal{R}(\mathcal{F})) \quad (7.5)$$

(provided, of course, that $\pi_{\vec{\sigma}^{nec}}(\mathcal{R}(I'_i) \cap \mathcal{R}(\mathcal{F})) > 0$; in the future, we take it for granted that the relevant results apply only if we are conditioning on a set with positive measure). (7.4) clearly follows from (7.5), since the time- m information sets for i partition $\mathcal{R}(I_i^*) \cap \mathcal{R}(\mathcal{F})$.

If $I'_i \neq I_i$, then (7.5) holds trivially, since in that case σ'_i agrees with σ_i at I'_i and all subsequent information sets. Thus, it suffices to prove (7.5) in the case that $I'_i = I_i$. We can assume without loss of generality that i 's actions at and after I_i are deterministic. If i is better

off by pretending to fail at I_i with some probability, then i is better off by pretending to fail at I_i with probability 1. Note that (a) whether or not there is a seemingly clean round, (b) which is the first seemingly clean round if there is one, and (c) which agents are considered nonfaulty at that round are completely determined by the failure pattern. Specifically, a particular failure pattern $F' \in \mathcal{F}$ determines the first seemingly clean round m^* . We partition the set \mathcal{F} into four sets, $\mathcal{F}_1, \dots, \mathcal{F}_4$, and show that conditional on $\mathcal{R}(I_i) \cap \mathcal{R}(\mathcal{F}_j)$, agent i does at least as well by using σ'_i as it does by using σ_i , for $j = 1, \dots, 4$.

\mathcal{F}_1 deals with a trivial case; the remaining elements of the partition consider the first seemingly clean round of $(\sigma'_i, \vec{\sigma}_{-i}^{nec})$ and $(\sigma_i, \vec{\sigma}_{-i}^{nec})$. (7.5) in the case that $I'_i = I_i$ clearly follows from this.

- (a) \mathcal{F}_1 consists of the failure patterns in \mathcal{F} where with $(\sigma_i, \vec{\sigma}_{-i}^{nec})$ an inconsistency is detected (because $f + 1$ agents seem to fail). Clearly, conditional on $\mathcal{R}(I_i) \cap \mathcal{R}(\mathcal{F}_1)$, i 's utility is at least as high with $(\sigma'_i, \vec{\sigma}_{-i}^{nec})$ as with $(\sigma_i, \vec{\sigma}_{-i}^{nec})$. It may be that with some failure patterns in \mathcal{F}_1 , no inconsistency is detected if i uses $(\sigma'_i, \vec{\sigma}_{-i}^{nec})$. But if the failure pattern is such that an inconsistency is detected with σ'_i , then an inconsistency is certainly detected with σ_i . Thus, in all the remaining runs, we consider no inconsistency is detected with either σ_i or σ'_i .
- (b) \mathcal{F}_2 consists of the failure patterns $F' \in \mathcal{F} - \mathcal{F}_1$ such that in all runs r' in $\mathcal{R}((\sigma'_i, \vec{\sigma}_{-i}^{nec})) \cap \mathcal{R}(I_i) \cap \mathcal{R}(F')$, the first clean round occurs at some round $m_1 < m$. It is easy to check that in a run r of $\mathcal{R}((\sigma_i, \vec{\sigma}_{-i}^{nec}))$ corresponding to r' , the first clean round also occurs at m_1 , so that all agents get the same utility at r and r' . Thus, conditional on $\mathcal{R}(I_i) \cap \mathcal{R}(\mathcal{F}_2)$, i 's utility is the same with $(\sigma'_i, \vec{\sigma}_{-i}^{nec})$ and $(\sigma_i, \vec{\sigma}_{-i}^{nec})$.
- (c) \mathcal{F}_3 consists of the failure patterns in $\mathcal{F} - \mathcal{F}_1$ that result in m being the first seemingly clean round with both $(\sigma_i, \vec{\sigma}_{-i}^{nec})$ and $(\sigma'_i, \vec{\sigma}_{-i}^{nec})$. This can happen in runs in $\mathcal{R}(\sigma_i, \vec{\sigma}_{-i}^{nec})$ only if the fact that i started pretending to fail at I_i with σ_i is not detected by any agent that does not crash (i.e., if no agent that decides is reachable from an agent that does not hear from i in round m). Conditional on $\mathcal{R}(I_i) \cap \mathcal{R}(\mathcal{F}_3)$, i 's utility is the same with $(\sigma'_i, \vec{\sigma}_{-i}^{nec})$ and $(\sigma_i, \vec{\sigma}_{-i}^{nec})$.
- (d) \mathcal{F}_4 consists of the failure patterns in $\mathcal{F} - \mathcal{F}_1$ where the first seemingly clean round with $(\sigma'_i, \vec{\sigma}_{-i}^{nec})$ comes at or after m while with $(\sigma_i, \vec{\sigma}_{-i}^{nec})$, the first clean round m^* comes strictly

before m or strictly after m . Let M be the number of agents that are not known to be faulty in I_i , and let a be the number of these that share i 's initial preference. It is straightforward to check that \mathcal{F}_4 satisfies the permutation assumption with respect to I_i , so by Lemma 21, conditional on $\mathcal{R}(I_i) \cap \mathcal{R}(\mathcal{F}_4)$, i 's expected utility with $(\sigma'_i, \vec{\sigma}_{-i}^{nec})$ is $\frac{a\beta_{0i}}{M} + \frac{(M-a)\beta_{1i}}{M}$.

To compute i 's expected utility with $(\sigma_i, \vec{\sigma}_{-i}^{nec})$, we must first consider how we could have m^* (the first seemingly clean round) occur before round m . This can happen if (and only if) i first learns in round $m'' - 1 \geq m - 1$ that some agent j^* crashed in round $m' \leq m - 1$, no agent nonfaulty agent j' (other than i) will learn that j^* crashed in round m' if i pretends to crash, and, as a result, round m' will seem clean to j' . This, in turn can happen if (and only if) either (i) $m' = m - 1$, and i does not hear from j^* for the first time in round $m - 1$, or (ii) $m' < m$, i did not hear from j^* for the first time in round $m' + 1$, and there is a chain $j_1, \dots, j_{m'' - m'}$ of agents that "hides" the fact that j^* actually crashed in round m' from i (and all other nonfaulty agents) until round m'' : j_1 does not hear from j^* in round m' ; for $h < m'' - m$, i does not hear from j_h in round $m' + h$; but j_{h+1} hears from j_h in round $m' + 1$ (thus, j_2 hears that j^* crashed in round m' from j_1 in round $m' + 1$, j_3 hears about this from j_2 in round $m' + 2$, and so on), i hears from $j_{m'' - m'}$ in round m'' (and so hears in round m'' that j^* crashed in round m'); and there is no shorter chain like this from j^* to i . Note that i can tell by looking at its history at time m whether it is possible that (i) or (ii) occurred. Specifically, (i) can occur only if there is an agent j^* that i does not hear from for the first time in round m , and (ii) can occur only if there is a chain $j_1, \dots, j_{m - m'}$ such that, for $h' < m - m'$, i does not hear from $j_{h'}$ for the first time in round $m' + h'$, either does not hear from $j_{m - m'}$ in round m or hears from $j_{m - m'}$ that j^* crashed in round m' , and i does not hear that j^* crashed in round m' before round m . Also note that in case (ii), i 's history must be such that none of the rounds between $m' + 1$ and $m'' - 1$ (inclusive) can seem clean to i (or the other nonfaulty agents).

Agent i 's expected utility with $(\sigma_i, \vec{\sigma}_{-i}^{nec})$ conditional on $\mathcal{R}(I_i) \cap \mathcal{R}(\mathcal{F}_4)$ depends on whether i 's history (and hence I_i) is such that (i) or (ii) could have occurred. If (i) or (ii) could not have occurred, then we must have $m^* > m$. To compute i 's expected utility with $(\sigma_i, \vec{\sigma}_{-i}^{nec})$, we can apply Lemma 21, but now we must include i among the faulty agents (since in the first seemingly clean round in runs of $\mathcal{R}(\sigma_i, \vec{\sigma}_{-i}^{nec}) \cap \mathcal{R}(I_i) \cap \mathcal{R}(\mathcal{F}_4)$, i will be viewed as faulty by the nonfaulty agents). Let F be the failure pattern $\{(i, m, A)\}$, where A is

the set of agents to which i sends a message in round m according to σ_i . Since $m^* > m$, we have $\mathcal{R}(I_i) \cap \mathcal{R}(\mathcal{F}_4) \cap \mathcal{R}(F) = \mathcal{R}(I_i) \cap \mathcal{R}(\mathcal{F}_4) \cap \mathcal{R}(F) \cap \mathcal{R}(D_{\geq m+1})$. Since I_i , \mathcal{F}_4 , F , and $m+1$ are compatible, by Lemma 21, i 's expected utility with $(\sigma_i, \vec{\sigma}_{-i}^{nec})$ conditional on $\mathcal{R}(I_i) \cap \mathcal{R}(F_4)$ is $\frac{(a-1)\beta_{0i}}{M-1} + \frac{(M-a)\beta_{1i}}{M-1}$. Since $\beta_{1i} > \beta_{2i}$, conditional on $\mathcal{R}(I_i) \cap \mathcal{R}(\mathcal{F}_4)$, i 's utility is higher with $(\sigma'_i, \vec{\sigma}_{-i}^{nec})$ than with $(\sigma_i, \vec{\sigma}_{-i}^{nec})$.

Now if I_i is such that (i) or (ii) could happen, we use the reachability assumption to provide upper bounds on the probability that $m^* < m$. Note that if (i) holds, $m^* < m$ only if no nonfaulty agent other than i hears that j^* crashed in round m' . By part 3 of the reachability assumption, this happens with probability at most $1/2M$. If (ii) holds, $m^* < m$ only if there is an appropriate chain. If $m'' = m$, then agent $j_{m-m'}$ in the chain is not known to be faulty in I_i , so by part 1 of the reachability assumption, the probability that no nonfaulty agent other than i hears from $j_{m-m'}$ that j^* crashed in round m' , conditional on $\mathcal{R}(I_i) \cap \mathcal{R}(\mathcal{F}_4)$ is again at most $1/2M$. Similarly, if $m'' > m$, then $j_{m-m'+1}$ is not known to be faulty in I_i , so by part 2 of the reachability assumption, the probability that no nonfaulty agent other than i hears from $j_{m-m'+1}$ that j^* crashed in round m' , conditional on $\mathcal{R}(I_i) \cap \mathcal{R}(\mathcal{F}_4)$ is again at most $1/2M$. Thus, the probability that $m^* < m$ conditional on $\mathcal{R}(I_i) \cap \mathcal{R}(\mathcal{F}_4)$ is at most $1/M$, even if both (i) and (ii) can occur. In the runs of $\mathcal{R}(\sigma_i, \vec{\sigma}_{-i}^{nec}) \cap \mathcal{R}(I_i) \cap \mathcal{R}(\mathcal{F}_4) \cap \mathcal{R}(F)$ where the first seemingly clean round is $m^* < m$, i 's utility is at most β_{0i} . If (i) or (ii) could happen and the first clean round is not before m , then it must occur strictly after m , as noted above. If it does occur after time m , then by the argument above, i 's expected utility is $\frac{(a-1)\beta_{0i}}{M-1} + \frac{(M-a)\beta_{1i}}{M-1}$. Thus, if I_i is such that (i) or (ii) could happen, then i 's expected utility conditional on $\mathcal{R}(I_i) \cap \mathcal{R}(\mathcal{F}_4)$ is at most

$$\left(\frac{1}{M} + \frac{M-1}{M} \cdot \frac{a-1}{M-1} \right) \beta_{0i} + \frac{M-1}{M} \cdot \frac{M-a}{M-1} \beta_{1i} = \frac{a}{M} \beta_{0i} + \frac{M-a}{M} \beta_{1i}.$$

In either case, conditional on $\mathcal{R}(I_i) \cap \mathcal{R}(\mathcal{F}_4)$, i 's utility is at least as high with $(\sigma'_i, \vec{\sigma}_{-i}^{nec})$ as with $(\sigma_i, \vec{\sigma}_{-i}^{nec})$.

Now, consider a deviation of type 2. If σ_i is a strategy with deviations of only types 1 and 2, let σ'_i be the strategy identical to σ_i except that i does not lie about its initial value and behaves as if it had not deviated from σ_i afterwards. There is a bijection between runs of $(\sigma_i, \vec{\sigma}_{-i}^{nec})$ and runs of $(\sigma'_i, \vec{\sigma}_{-i}^{nec})$, so that two corresponding runs r and r' are identical except that in run r

agent i may lie about its initial value and in r' agent i does not. (So, among other things, the random choices made in r and r' are the same.) Again, the lie does not affect which round (if any) will be considered clean nor which agents will be viewed as nonfaulty in that round. If i is not one of the agents considered nonfaulty in the clean round, or if i is considered nonfaulty but i is not the agent whose preference is chosen, then the outcome is the same in r and r' . If i is the agent whose value is chosen, then i is worse off if it lies than if it doesn't. Thus, i does not gain if it lies about its initial value. Again, (7.5) holds. Thus, (7.3) holds for deviations of types 1 and 2.

Finally, we show that (7.3) holds if we allow deviations of types 3–10. To deal with these, we proceed by induction on the number of deviations of types 3–10 in σ_i , removing deviations starting from the earliest deviation. That is, we consider the information set I_i where the first deviation of type 3–9 occurs, so that the only deviations prior to I_i are of type 1 or 2, and show that we can do better by removing the deviation at I_i . Before getting into the details, we need to state carefully what counts as a deviation of type 1 or 2 prior to I_i . We try to “explain” as much as possible by i pretending to fail, so as to delay the first deviation not of types 1 or 2 as late possible. Thus, if i pretends to fail at information set I'_i (i.e., sends message according to σ_i^{nec} up to I'_i , sends messages, again according to σ_i^{nec} , to some agents at I'_i and does not send messages to some agents it does not know to be faulty), and then sends a message to some agent at some information set I''_i after I'_i , then we say that the first deviation not of types 1 and 2 occurs at the information set that immediately precedes I''_i (it is a deviation of type 8).

In the base case, σ_i contains no deviations of type 3–10; we have already shown that (7.3) holds in this case. For the inductive step, let I_i be an information set at which σ_i has a deviation of type 3–10 and there are no deviations of type 3–10 prior to I_i . We consider each deviation of type 3–10 in turn.

3. If i sends an incorrectly formatted message to j , then either j receives this message and decides Ψ or j crashes before sending any messages to an agent $j' \neq i$ (or before deciding, if $m = f$). Let σ'_i be the strategy that is identical to σ_i except i sends a correctly formatted message to j . In all cases, i does at least as well if i uses the strategy σ'_i as it does using σ_i . Thus, (7.3) follows from the induction hypothesis.
4. If $m = 1$ and i sends values y_{ij}^t to an agent j such that there is no polynomial q_i^t of degree 1 that interpolates them then either an inconsistency is detected or i would have

done at least as well by choosing these values according to some polynomial. (Here and in the remainder of the proof, when we say “an inconsistency is detected”, we mean “an inconsistency is detected by a nonfaulty agent different from i ”.) If i does not choose q_i^t at random, since $f + 1 < n$, there exists a nonfaulty agent $j \neq i$ that sends values based on truly random polynomials. Thus, the agent whose preference determines the consensus value is chosen at random, even if q_i^t is not chosen at random. So choosing q_i^t at random does not affect the expected outcome. Again, (7.3) follows from the induction hypothesis.

5. Suppose that i does not choose z_{ij}^m according to protocol. From the perspective of an agent $j' \neq i$ following the protocol $\sigma_{j'}^{nec}$, it does not affect the outcome if these values are not chosen randomly. So, yet again, i does just as well if i chooses the numbers randomly, and (7.3) holds.
6. Clearly there is no benefit to i deciding on a value other than Ψ early (it can decide the same value at round $f + 1$) and no benefit in deciding an incorrect value (since this guarantees that there is no consensus). Thus, yet again, (7.3) holds.
7. Suppose that $m = f + 1$ and i lies about y_{ji}^t to some $l \neq i$ for $j \neq i$. If it turns out that there are not $n - t$ agents that seem to be nonfaulty in the first clean round, then the value of y_{ji}^t is irrelevant; it is not used in the calculation. If there are $n - t$ seemingly nonfaulty agents in the clean round, then either an inconsistency is detected due to the lie (if y_{ji}^t is sent to some nonfaulty agent, who then cannot interpolate a polynomial through it and the other values received), in which case i is clearly worse off, or the sum S computed will be a random element of $\{0, \dots, n - t - 1\}$, so the initial preference of each of the seeming nonfaulty agents is equally likely to be chosen whether or not i lies. Thus, i does not gain by lying about y_{ji}^t , so (7.3) holds.
8. Suppose that i does not send a message in round $m' < m$ to an agent j that i does not know (at round m) to have been faulty at round m' and then i sends a message to $j' \neq i$ in round m . Let I_i be the round- $m - 1$ information set and I'_i the round- m information set that immediately precedes I_i . If $m' < m - 1$, since m is the first round that a deviation of types 3–10 occurs, and since i does not know at any round $m'' < m$ that j was faulty at round m' (since i does not know it at round m), i does not send messages between rounds m' and m . Thus, sending a round m message to j' either leads to an inconsistency being detected or does not affect the outcome (which can be the case if j fails before deciding

Ψ). This means that i does at least as well if i does not send a message to j' at round m , so (7.3) holds. So we can assume without loss of generality that $m' = m - 1$, and that m' is the first round that i did not send a message to an agent j . Similarly, we can assume that i gets a message from j' in round $m - 1$; otherwise we can consider the strategy σ'_i where i does send a message to j' in round $m - 1$, and otherwise agrees with σ_i , and again the result follows from the induction hypothesis.

The rest of the proof proceeds much in the spirit of the proof for deviations of type 1. We partition \mathcal{F} into subsets $\mathcal{F}_1, \dots, \mathcal{F}_4$, and show that, for $j = 1, \dots, 4$, i does at least as well with $\vec{\sigma}^{nec}$ as with $(\sigma_i, \vec{\sigma}_{-i}^{nec})$ conditional on $\mathcal{R}(I_i) \cap \mathcal{R}(F_j)$; (7.3) then follows. As in the case of type 1 failures, \mathcal{F}_1 consists of the failure patterns in \mathcal{F} where, with $(\sigma_i, \vec{\sigma}_{-i}^{nec})$, $f + 1$ failures are detected. Clearly, conditional on $\mathcal{R}(I_i) \cap \mathcal{R}(\mathcal{F}_1)$, i 's utility is higher with $\vec{\sigma}^{nec}$ than with $(\sigma_i, \vec{\sigma}_{-i}^{nec})$.

Let \mathcal{F}_2 be the set of failure patterns in $\mathcal{F} - \mathcal{F}_1$ such that in runs from $\mathcal{R}(I'_i) \cap \mathcal{R}(\mathcal{F}_2)$, the agents that decide do not hear about i 's round m message to j' . Let σ'_i be identical to σ_i except that at I'_i agent i does not send a message to j' . It is not hard to check that \mathcal{F}_2 satisfies the permutation assumption with respect to I_i . Clearly, i gets the same utility with σ_i as with σ'_i conditional on $\mathcal{R}(I_i) \cap \mathcal{R}(\mathcal{F}_2)$. Since, with σ'_i , i has fewer deviations of types 3–9 than with σ_i , by the induction hypothesis, (7.3) holds conditional on $\mathcal{R}(I_i) \cap \mathcal{R}(\mathcal{F}_2)$.

Now let \mathcal{F}_3 consist of all failure patterns in $\mathcal{F} - \mathcal{F}_1$ such that, with σ_i , the agents that decide hear both that i sent a message to j' in round m and that i did not send a message to some agents in round $m - 1$. Thus, with σ_i , an inconsistency will be detected, so i does at least as well with σ'_i as with σ_i conditional on $\mathcal{R}(I_i) \cap \mathcal{R}(\mathcal{F}_3)$. \mathcal{F}_3 also satisfies the permutation assumption with respect to I_i , so (7.3) holds conditional on $\mathcal{R}(I_i) \cap \mathcal{R}(\mathcal{F}_3)$ by the induction hypothesis.

Finally, let \mathcal{F}_4 be the remaining failure patterns in $\mathcal{F} - \mathcal{F}_1$, the ones where agents that decide hear about the message sent by i to j' but not about the omissions of i in round $m - 1$. Let σ''_i be a strategy identical to σ_i , except that at I_i i does not deviate from σ_i^{nec} . Conditional on $\mathcal{R}(I_i) \cap \mathcal{R}(\mathcal{F}_4)$, i clearly gets the same utility with $(\sigma_i, \vec{\sigma}_{-i}^{nec})$ as with $(\sigma''_i, \vec{\sigma}_{-i}^{nec})$. It is not hard to show that \mathcal{F}_4 also satisfies the permutation assumption with respect to I_i . With σ''_i , i has fewer deviations of types 3–9 than with σ_i . Thus, by the induction hypothesis, (7.3) holds conditional on $\mathcal{R}(I_i) \cap \mathcal{R}(\mathcal{F}_4)$. This completes the

argument for deviations of type 8.

10. Suppose that i lies about j 's status to an agent $j' \neq i$. That is, either (a) i says that j did not crash before round m' although i knows that j did crash in round $m' - 1$; (b) i says that j crashed at or before round m' although i received a message from j in round m' and either $m' = m - 1$ or $m' < m$ and i did not receive a message from any agent saying that j crashed in round m' ; or (c) i lies about the numbers z_{ji}^{m-1} sent by j or about which agent reported that j crashed. Again we consider each of these cases in turn. We can assume without loss of generality that i did not pretend to crash in I_i , since otherwise the arguments for deviations of type 8 would apply.

- (a) Suppose that i lies by saying that j did not crash before m' even though i knows that j did in fact crash earlier. This means that i is claiming to have received a message from j in round m' . Clearly, it cannot be the case that i knows that j crashed before $m' - 1$, because then i would know that no agent would get a message from j in round $m' - 1$, and an inconsistency would be detected by j' if the deviation had any impact on the outcome. Thus, we can assume that j in fact crashed in round $m' - 1$. Since we are assuming that i first deviates in round m , i must have learned in round $m - 1$ about j 's crash in round $m' - 1$. That means that either (i) $m' = m$ and i did not receive a message from j in round $m - 1$ or (ii) $m' < m$ and i must have received a message from some agent j'' with this information in round $m - 1$. We can assume without loss of generality that i gets a message from j' in round $m - 1$, for otherwise i would do at least as well by not lying to j , and (7.3) would hold by the induction hypothesis.

Consider case (i). If $m = 2$, then i pretending that j did not crash in round 1 can help only if this leads to round 1 being viewed as clean. But this is the case only if j' received a message from j in round 1 (although i did not). According to σ_i^{nec} , i 's round- m message includes the status report SR_i^m . Agent i must send such a status report even with σ_i , otherwise an inconsistency is detected and clearly i is worse off. Since i claims to have received a message from j in round 1, $SR_i^m[j]$ has the form (∞, z_{ji}^1) , where $z_{ji}^{m-1}[i]$ is the random number sent in round 1 to all agents. Given that we have assumed that j also sent a round 1 message to j' , j' also received $z_{ji}^1[i] = z_{j'j}^1[j']$. Thus, j' will detect an inconsistency and decide Ψ unless i correctly

guesses $z_{j_i}^1[i]$. The probability of i guessing $z_{j_i}^1[i]$ correctly is at most $\frac{1}{n}$.

We now partition \mathcal{F} into three sets of failure patterns \mathcal{F}_1 , \mathcal{F}_2 , and \mathcal{F}_3 , and show that for $j = 1, 2, 3$, i does at least as well with $\vec{\sigma}^{nec}$ as with $(\sigma_i, \vec{\sigma}_{-i}^{nec})$. Again, \mathcal{F}_1 consists of the failure patterns in \mathcal{F} where with $(\sigma_i, \vec{\sigma}_{-i}^{nec})$, $f + 1$ failures are detected. Clearly the claim holds in this case. \mathcal{F}_2 consists of the failure patterns F' in $\mathcal{F} - \mathcal{F}_1$ where the message that i sent in I_i has no impact on the outcome; that is, either i crashes before sending the message to j' or no nonfaulty agent is reachable from j' without i between round $m + 1$ and $f + 1$. Let σ'_i be identical to σ_i except that, at I_i , i replaces the reports relative to j with SR_i (the correct report) in messages sent to j' , while sending the same messages to other agents. Thus, i has fewer deviations with σ'_i than with σ_i . Clearly, conditional on $\mathcal{R}(I_i) \cap \mathcal{R}(\mathcal{F}_2)$, i gets the same expected utility with $(\sigma_i, \vec{\sigma}_{-i}^{nec})$ as with $(\sigma'_i, \vec{\sigma}_{-i}^{nec})$. It is easy to check that \mathcal{F}_2 satisfies the permutation assumption with respect to I_i , so by the induction hypothesis, (7.3) holds conditional on $\mathcal{R}(I_i) \cap \mathcal{R}(\mathcal{F}_2)$.

Let \mathcal{F}_3 consist of the remaining failure patterns in \mathcal{F} . In runs of $\mathcal{R}(\sigma_i, \vec{\sigma}_{-i}^{nec}) \cap \mathcal{R}(I_i) \cap \mathcal{R}(\mathcal{F}_3)$, j' detects an inconsistency and decides Ψ unless i guesses the random number correctly. Again, it is not hard to check that \mathcal{F}_3 satisfies the permutation assumption with respect to I_i . Since the largest utility that i can get if no inconsistency is detected is β_{0i} ,

$$u_i((\sigma_i, \vec{\sigma}_{-i}^{nec}) \mid \mathcal{R}(I_i) \cap \mathcal{R}(\mathcal{F}_3)) \leq \frac{1}{n}\beta_{0i} + \frac{n-1}{n}\beta_{2i}.$$

On the other hand, by Lemma 21,

$$u_i(\vec{\sigma}^{nec} \mid \mathcal{R}(I_i) \cap \mathcal{R}(\mathcal{F}_3)) \geq \frac{1}{n}\beta_{0i} + \frac{n-1}{n}\beta_{1i}.$$

Since $\beta_{1i} > \beta_{2i}$, we have

$$u_i(\vec{\sigma}^{nec} \mid \mathcal{R}(I_i) \cap \mathcal{R}(\mathcal{F}_3)) \geq u_i((\sigma_i, \vec{\sigma}_{-i}^{nec}) \mid \mathcal{R}(I_i) \cap \mathcal{R}(\mathcal{F}_3)).$$

Therefore, (7.3) holds if $m = 2$.

Continuing with case (i), suppose that $m > 2$. Now it is possible that i pretending that j did not crash can help even if j did not send a message to j' . Nevertheless, essentially the same argument will work. This is because now SR_i would have to

include z_{ji}^{m-1} . Moreover, $z_{ji}^{m-1}[j'] = z_{j'j}^{m-2}[j']$, the random number in $\{0, \dots, n-1\}$ sent by j' to j in round $m-2$. Clearly, j' knows this number, so i would have to guess it correctly. The argument now proceeds as above.

Now consider case (ii). There are two ways in which i can ignore the information that j'' sent about j in round $m-1$. The first is to pretend that j'' crashed in round $m-1$; the second is for i to lie about the message that it received from j'' (but to say that it did get a message from j''). In the first case, as with deviations of type 8, we can assume without loss of generality that i does not know that j'' is faulty at the beginning round m . We partition \mathcal{F} into three sets much as in the argument for case (i): \mathcal{F}_1 , the failure patterns in which more than $f+1$ failures are detected with σ_i ; \mathcal{F}_2 , the failure patterns where i 's lie has no impact on the outcome; and \mathcal{F}_3 , the remaining failure patterns. Again, it is easy to see that (7.3) holds conditional on $\mathcal{R}(I_i) \cap \mathcal{R}(\mathcal{F}_1)$ and $\mathcal{R}(I_i) \cap \mathcal{R}(\mathcal{F}_2)$. To see that (7.3) holds conditional on $\mathcal{R}(I_i) \cap \mathcal{R}(\mathcal{F}_3)$, we use the reachability assumption, much as we did for as in (d) of the argument for deviations of type 1. By part 1 of the reachability assumption, if i pretends that j'' crashed in round $m-1$, an inconsistency will be detected with probability at least $(2M-1)/2M$. Thus, the same argument as that used in part (e) of the argument for deviations of type 1 shows that (7.3) holds conditional on $\mathcal{R}(I_i) \cap \mathcal{R}(\mathcal{F}_3)$.

The analysis is essentially the same if i lies about the message it received from j'' , except that, conditional on $\mathcal{R}(I_i) \cap \mathcal{R}(\mathcal{F}_3)$, by the reachability assumption, j' receives the round $m-1$ message from j'' with probability at least $(2M-1)/2M$, so j' receives inconsistent reports about j 's status in round $m-1$, and decides Ψ .

- (b) Suppose that i lies to some j' in round m by saying that j crashed at or before round m' although i received a message from j in round m' and either $m' = m-1$ or $m' < m-1$ and i did not receive a message from any agent saying that j crashed in round m' . If $m' = m-1$, then we can proceed as in part (a). Specifically, we can use the reachability assumption to show that i is better off if i does not lie.

The analysis is similar if i pretends to have received a message in round $m-1$ from some agent j'' saying that j crashed in an earlier round. If i did not receive a message from j'' in round $m-1$ saying that j crashed before m' but is claiming to have done so, then we can again use the same arguments as in part (a) where either i must guess the random number $z_{j'j''}^{m-2}[j']$ known by j' (if j'' did not send a round $m-1$ message

to i) or i has to lie about the round $m - 1$ report of j'' .

- (c) It is easy to see that i does not gain if i lies about which agent told him that j crashed or about the values z_{ji}^{m-1} sent by j to i in round $m - 1$ (and may be worse off, if an inconsistency is detected).

This completes the proof of the inductive step and, with it, the proof of the theorem. \square

7.3 A π -Sequential Equilibrium for Fair Consensus

We now show that the protocol $\bar{\sigma}^{nec}$ can be extended to a π -sequential equilibrium with minimal changes. In the proof of Theorem 22, we showed that i could not gain by deviating at an information set I_i where there were no deviations of type 1–9 prior to I_i . We did not show that i does not gain from deviating at I_i if an inconsistency is detected at I_i , so that i is expected to decide Ψ . In fact, if i believes that the inconsistency may go unnoticed by other agents due to crashes and consensus may still be reached on some value in $\{0, 1\}$, then i always gains by not deciding Ψ . However, suppose that μ^{sec} is a belief system such that at an information set I_i for i that is off the equilibrium path due to a deviation (or multiple deviations) from $\bar{\sigma}^{nec}$ by agents other than i , i believes that these agents decided Ψ when they deviated. (Intuitively, i believes that if the agents were crazy enough to deviate in the first place, then they were also crazy enough to decide Ψ .) In that case, deciding Ψ is also a best response for i .

The belief system μ^{sec} is not enough to deal with information sets I_i off the equilibrium path due to i himself having deviated. Agent i cannot believe that it played Ψ when it in fact did not. To get a sequential equilibrium, we modify σ_i^{nec} at information sets off the equilibrium path that are reached due only to agent i 's deviations. Define the strategy σ_i^{sec} so that it agrees with σ_i^{nec} at every information set I_i where agent i has not deviated in the past. Thus, in particular, i decides Ψ with σ_i^{sec} if i detects an inconsistency at one of these information sets. More generally, say that an information set I_i is *unsalvageable* if i knows at I_i that another agent j deviated or detected an inconsistency at a point when j had not crashed, and thus decided Ψ . I_i is certainly unsalvageable if reaching I_i requires deviations by agents other than i (for then the agent that performed that deviation decided Ψ). But even if i is the only agent who deviates at I_i , I_i may be unsalvageable. For example, i does not send a message to j in round m_1 , i sends a message to j in round $m_2 > m_1$, and then j sent a message to i in round $m_2 + 1$,

the round- $(m_2 + 2)$ information set where i receives j 's message is also unsalvageable. If I_i is unsalvageable, i decides Ψ . Finally, if I_i is salvageable, then at I_i agent i acts in a way that is most likely to have the other agents think that there has been no inconsistency. In general, there may be more than one failure pattern that will prevent a nonfaulty agent from realizing that there is an inconsistency. For example, if $f = 1$, $n = 3$, and agent 1 did not send a message to agent 2 in round m , but did send a message to agent 3, then i can either not send a message to any agent in round $m + 1$, or it can send a message to agent 3. If it is more likely that neither 2 nor 3 failed in round m than agent 2 failed before telling agent 3 that it did not hear from 1, then it would be better for i not to send a message to 2 or 3 in round $m + 1$. If there is more than one best response, then i chooses a fixed one according to some ordering on actions. (Note that this means that, unlike $\bar{\sigma}^{nec}$, the behavior of $\bar{\sigma}^{sec}$ may depend on π .)

Having defined $\bar{\sigma}^{sec}$, we can now define μ^{sec} formally. We assume that there are only finitely many actions that i can play at each of its information sets I_i : it can send one of K_{I_i} possible messages and/or decide one of Ψ , 0, or 1 if it has not yet made a decision, or do nothing. Given an integer $M > 0$, let $\bar{\sigma}^M$ be the strategy profile where at each information set I_i , agent i plays $\sigma_i^{sec}(I_i)$ with probability $1 - 1/M$, and divides the remaining probability $1/M$ over all the actions that can be played at I_i as follows: if i has already decided before, then i sends each of the K_{I_i} possible messages with equal probability $\frac{1}{M(K_{I_i}+1)}$ and does nothing with probability $\frac{1}{M(K_{I_i}+1)}$; if i has not yet decided at I_i , then for each of the K_{I_i} messages \mathbf{m} that it can send, it decides Ψ and sends \mathbf{m} with probability $\frac{1}{M(K_{I_i}+1)} - \frac{1}{M^2(K_{I_i}+1)}$, decides Ψ and sends no message with probability $\frac{1}{M(K_{I_i}+1)} - \frac{1}{M^2(K_{I_i}+1)}$, and performs each of the remaining $3(K_{I_i} + 1)$ possible actions with equal probability $\frac{1}{3M^2(K_{I_i}+1)}$. Clearly $\bar{\sigma}^M$ is completely mixed and the sequence $\bar{\sigma}^M$ converges to $\bar{\sigma}^{sec}$. Given a round- m information set I_i and global history $h \in I_i$, let

$$\mu_{I_i}^{sec}(h) = \lim_{M \rightarrow \infty} \frac{\pi_{\bar{\sigma}^M}(h)}{\pi_{\bar{\sigma}^M}(I_i)}.$$

The effect of this definition of $\mu_{I_i}^{sec}$ beliefs is that if I_i is off the equilibrium path as a result of some other agent j 's deviation, then i believes that j played Ψ . Moreover, i believes that other agents j have similar beliefs.

Theorem 23 shows that $\bar{\sigma}^{sec}$ is a π -sequential equilibrium for a reasonable and uniform π .

Theorem 23. *If $f + 1 < n$, π is a distribution that supports reachability, is uniform, and allows*

up to f failures, and agents care only about consensus, then $\vec{\sigma}^{sec}$ is a π -sequential equilibrium.

Proof. Fix an agent i , a round- m information set I_i , and strategy σ_i . It is easy to see that μ^{sec} is consistent. Thus, it suffices to show that

$$u_i((\sigma_i^{sec}, \vec{\sigma}_{-i}^{sec}) \mid \mathcal{R}(I_i)) \geq u_i((\sigma_i, \vec{\sigma}_{-i}^{sec}) \mid \mathcal{R}(I_i)), \quad (7.6)$$

where the expected utilities are taken relative to $\mu_{\vec{\sigma}^{sec}, I_i}^{sec}$.

We need to consider the cases where (a) I_i is consistent with $\vec{\sigma}^{sec}$, (b) I_i is inconsistent with $\vec{\sigma}^{sec}$ and unsalvageable, and (c) I_i is inconsistent with $\vec{\sigma}^{sec}$ and salvageable. In case (a), σ_i^{sec} agrees with σ_i^{nec} ; the argument of the proof of Theorem 22 shows that it is a best response. In case (b), the definition of μ^{sec} guarantees that i ascribes probability 1 to whichever agent has deviated or detected a deviation playing Ψ , so it is a best response for i to play Ψ . Finally, in case (c), for failure patterns where some other agent j detects i 's deviation, i ascribes probability 1 to j playing Ψ , so it does not matter what i does. On the other hand, for failure patterns where all the nonfaulty agents will consider it possible that there are no deviations, the proof of Theorem 22 shows that i should continue to play in a way consistent with σ_i^{nec} . If there are several choices of how to play that might be consistent with σ_i^{nec} , then i should clearly play one that is best. \square

Summary

We have shown that there is no f -Nash equilibrium protocol that solves fair consensus if $f \geq 1$. We have also provided a strategy for fair consensus that is a π -Nash equilibrium and can be extended to a π -sequential equilibrium, where π is a distribution on contexts that allows up to f failures and satisfies minimal conditions, as long as $n > f + 1$. Although our argument is surprisingly complicated, we have considered only the simplest possible case: synchronous systems, crash failures, and only one player deviating (i.e., no coalitions). A small variant of our strategy also gives a Nash and sequential equilibrium even if coalitions are allowed, but proving this seems significantly more complicated. Of course, things will get even worse once we allow more general types of failures, such as omission failures and Byzantine failures. But such failure types, combined with rational agents, are certainly of interest if we want to apply consensus in,

for example, financial settings of the type considered by Mazières (2015). Consensus is known to be impossible in an asynchronous setting, even with just one failure (Fischer et al. 1985), but algorithms that attain consensus with high probability are well known (e.g., (Aspnes 2003)). We may thus hope to get an ϵ - π -Nash equilibrium in the asynchronous setting if we also allow rational agents. We believe that the techniques developed in this paper will be applicable to these more difficult problems.

It is also worth examining our assumptions regarding distributions in more detail. The uniformity assumption implies that no agent is more likely to fail than any other. If all agents can be identified with identical computers, then this seems quite reasonable. But if one agent can be identified with a computer that is known to be more prone to failure, then the uniformity assumption no longer holds. Note that the uniformity assumption does allow for correlated failures, just as long as the permutation of a correlated failure is just as likely as the unpermuted version.

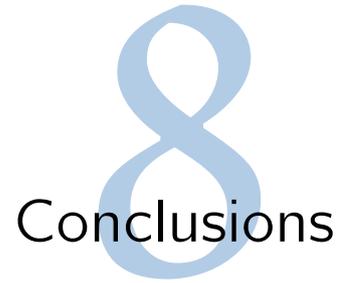
Now consider the assumption that π supports reachability. If we are considering Nash equilibrium (where there is only one deviating agent), the assumption says that the probability, conditional on an information set I_i (and some assumptions about failures), that some information (about a message sent by an agent that crashes or about the fact that an agent crashed in a particular round) is quite high, where “quite high” is a function of the number of agents M that are nonfaulty according to I_i . Since the more nonfaulty agents there are, the more likely it is that an agent $l \neq i$ is reachable from j without i .

Our final comment concerns the fairness assumption. While this assumption distinguishes our work from some of the other related work (e.g., (Afek et al. 2014; Bei et al. 2012)), since, as we observed above, a consensus protocol must essentially implement a randomized dictatorship, achieving fairness once we get consensus in the presence of rational and faulty agents is not that difficult; we must simply ensure that the rational agents cannot affect the probability of a particular agent being selected as dictator. We enforce this using appropriate randomization in our protocol. The requirement by Bei et al. (2012) that consensus must be achieved no matter what the deviating agents do turns out to have far more impact on the technical results than the fairness requirement.

In the next chapter, we conclude the thesis with a final discussion.

Type	Notation	Description
Agents	\mathcal{N}	Set of agents.
	n	Number of agents.
	v_i	i 's input for consensus.
Failures	f	Upper bound on number of failures.
	\vec{f}	Failure of an agent.
	F	Failure pattern (set of failures \vec{f}).
	\mathcal{F}	Set of failure patterns.
	\vec{v}	Configuration of values.
	(F, \vec{v})	Context.
	π	Probability distribution on contexts.
Actions and histories	a_i^m	Round- m action of agent i .
	Ψ	Action that aborts consensus.
	h	Global history.
	I_i	Information set for agent i .
	$\mathcal{R}(I_i)$	Set of runs compatible with information set I_i .
	$\mathcal{R}(h)$	Set of runs compatible with global history h .
	$\mathcal{R}(F, \vec{v})$	Set of runs with context (F, \vec{v}) .
	$\mathcal{R}(F)$	Set of runs with failure pattern F .
	$\mathcal{R}(\mathcal{F})$	Set of runs compatible with set \mathcal{F} of failure patterns.
	$\mathcal{A}_i(I_i)$	Set of actions available to i at I_i .
Utilities	μ	Belief system.
	β_{0i}	Benefit of i when agents reach consensus on i 's value.
	β_{1i}	Benefit of i when agents reach consensus on another value.
	β_{2i}	Benefit of i when agents do not reach consensus.
	$u_i(\vec{\sigma} \mid I_i)$	Expected utility of i when agents use $\vec{\sigma}$ conditioned on I_i .
	$u_i(\vec{\sigma})$	Expected utility of i when agents use $\vec{\sigma}$.
	$u_i(\vec{\sigma} \mid (F, \vec{v}))$	Expected utility of i conditioned on context (F, \vec{v}) .
	$u_i(\vec{\sigma} \mid F)$	Expected utility of i conditioned on failure pattern F .
	$u_i(\vec{\sigma} \mid R)$	Expected utility of i conditioned on the run being in R .
Algorithm	ST_i	Set of tuples (j, v_j) that i receives in round 1.
	SR_i^m	Status reports of i at round m .
	$x_i[t]$	Secret random number of i for each number $t \in \{0, \dots, f\}$ of failures.
	q_i^t	Polynomial encoding $x_i[t]$.
	y_{ij}^t	j 's share of the secret x_i^t .
	$z_{ij}^m[l]$	Random number that proves to l that i sent a message to j .
	NC_m	Set of agents that did not crash before round $m + 1$.
	m^*	First round that seems clean.
Strategies	σ_i	Strategy for agent i .
	$\vec{\sigma}$	Strategy profile.
	$\vec{\sigma}^{nec}$	π -NE protocol for fair consensus.
	$\vec{\sigma}^{sec}$	π -SE protocol for fair consensus.
	μ^{sec}	Belief system consistent with $\vec{\sigma}^{nec}$.

Table 7.1: Notation - consensus with crashes.



Conclusions

To summarize the work in this thesis, we addressed the problem of rational behaviour in infinitely repeated gossip dissemination, infinitely repeated pairwise exchanges in dynamic networks, and fair consensus with crashes. Our main contributions were as follows. Regarding the problem of gossip dissemination, we proved a slightly weaker version of a Folk Theorem for the notion of sequential equilibrium. Regarding the problem of pairwise exchanges, we proposed a new game theoretical model of repeated games, in dynamic networks, we defined a new notion of equilibrium for this model that refines sequential equilibrium, and we identified multiple necessary and sufficient restrictions on the network, structure of pairwise exchanges, and protocols to sustain cooperation according to our new notion of equilibrium. Regarding the problem fair consensus with crashes, we proved that there is no Nash equilibrium solution if agents can know what the failures are, but if there is a probability distribution on failures that satisfies minimal assumptions, then there is a Nash equilibrium protocol that solves fair consensus with crashes; a slightly modified version of this protocol is also a sequential equilibrium.

These results provide new insights towards the goal of devising dependable distributed systems robust to rational behaviour: the impossibility results establish conditions under which we cannot sustain cooperation, and the possibility results show that we can sustain cooperation under mild assumptions. In particular, the results for consensus and pairwise exchanges in dynamic networks show that no protocol is robust to rational behaviour if agents can have any beliefs about the information they lack, i.e., information about crashes in consensus and information about the evolution of the network in pairwise exchanges. Consequently, agents must form some beliefs about the missing information in the form of probability distributions. However, our possibility results show that agents do not need to know exact probability distributions on crashes or network topologies; they only need to know that such distributions exist and that they satisfy minimum properties, namely, the distribution on failures must satisfy reachability and uniformity, and the distribution on network topologies must satisfy strong timely punishments or connectivity with known degrees. In the problem of gossip dissemination, we do not need

to make any assumptions about beliefs other than that agents believe that every other agent wants to participate in the system and that events are always disseminated independently of each other, which are perfectly reasonable assumptions in practice. For all three problems, we defined protocols that implement novel techniques used by the agents to detect deviations and promptly punish the deviating agents. Since all protocols are sequential equilibria, the threats of punishments are credible.

We believe that these results can be easily generalized to deal with crashes and message loss in all three problems. We also believe that we can extend the results of fair consensus and pairwise exchanges in dynamic networks to cope with collusion with minimal changes in the protocols. Beyond these extensions, the main direction for future work would be to understand how to sustain cooperation in asynchronous systems in the presence of Byzantine behaviour.

Bibliography

- Abraham, I., D. Dolev, R. Gonen, & J. Halpern (2006). Distributed computing meets game theory: robust mechanisms for rational secret sharing and multiparty computation. In *Proceedings of the 25th Annual ACM SIGACT-SIGOPS Symposium on Principles of Distributed Computing*, PODC'06, Denver, CO, USA, pp. 53–62. ACM.
- Abraham, I., D. Dolev, & J. Halpern (2013). Distributed protocols for leader election: A game-theoretic perspective. In *27th International Symposium on Distributed Computing*, DISC'13, pp. 61–75. Springer.
- Abreu, D., P. Dutta, & L. Smith (1994). The folk theorem for repeated games: A NEU condition. *Econometrica* 62(4), 939–948.
- Afek, Y., Y. Ginzberg, S. Landau Feibish, & M. Sulamy (2014). Distributed computing building blocks for rational agents. In *Proceedings of the 33th ACM Symposium on Principles of Distributed Computing*, PODC'14, Paris, France, pp. 406–415. ACM.
- Aiyer, S., L. Alvisi, A. Clement, M. Dahlin, J. Martin, & C. Porth (2005). BAR fault tolerance for cooperative services. In *Proceedings of the 20th ACM Symposium on Operating Systems Principles*, SOSP'05, Brighton, United Kingdom, pp. 45–58. ACM.
- Aspnes, J. (2003). Randomized protocols for distributed consensus. *Distributed Computing* 16(2–3), 165–176.
- Bei, X., W. Chen, & J. Zhang (2012). Distributed consensus resilient to both crash failures and strategic manipulations. Available at <http://arxiv.org/abs/1203.4324>; version 3.
- Bellare, M., A. Desai, E. Jorjani, & P. Rogaway (1997). A concrete security treatment of symmetric encryption. In *38th Annual Symposium on Foundations of Computer Science*, FOCS'97, pp. 394–. IEEE.
- Ben-Porath, E. (2003). Cheap talk in games with incomplete information. *Journal of Economic Theory* 108(1), 45–71.

- Bhaskar, V. & I. Obara (2002). Belief-Based Equilibria in the Repeated Prisoners' Dilemma with Private Monitoring. *Journal of Economic Theory* 102(1), 40–69.
- Birman, K. P., M. Hayden, O. Ozkasap, Z. Xiao, M. Budiu, & Y. Minsky (1999). Bimodal multicast. *ACM Trans. Comput. Syst.* 17(2), 41–88.
- Cohen, B. (2003). Incentives build robustness in bittorrent. In *Proceedings of the 1st Workshop on Economics of Peer-to-Peer Systems, P2PEcon'03*, Berkeley, CA, USA.
- Compte, O. (1998). Communication in repeated games with imperfect private monitoring. *Econometrica* 66(3), pp. 597–626.
- Dolev, D. & H. R. Strong (1982). Polynomial algorithms for multiple processor agreement. In *14th Annual ACM Symposium on Theory of Computing, STOC'82*, pp. 401–407.
- Dolev, S. (2000). *Self-Stabilization*. MIT Press.
- Dolev, S., E. Schiller, P. Spirakis, & P. Tsigas (2010). Game authority for robust and scalable distributed selfish-computer systems. *Theor. Comput. Sci.* 411(26-28), 2459–2466.
- Dolev, S., E. M. Schiller, P. Spirakis, & P. Philippas (2011). Strategies for repeated games with subsystem takeovers implementable by deterministic and self-stabilising automata. *Int. J. Auton. Adapt. Commun. Syst.* 4(1), 4–38.
- Ely, J. & J. Välimäki (2002). A robust folk theorem for the prisoner's dilemma. *Journal of Economic Theory* 102(1), 84–105.
- Fabrikant, A., A. Luthra, E. Maneva, C. H. Papadimitriou, & S. Shenker (2003). On a network creation game. In *Proceedings of the 22nd ACM Symposium on Principles of Distributed Computing, PODC'03*, Boston, MA, USA, pp. 347–351. ACM.
- Feldman, M., K. Lai, I. Stoica, & J. Chuang (2004). Robust incentive techniques for peer-to-peer networks. In *Proceedings of the 5th ACM conference on Electronic commerce, EC'04*, New York, NY, USA, pp. 102–111.
- Fischer, M. J., N. A. Lynch, & M. S. Paterson (1985). Impossibility of distributed consensus with one faulty processor. *Journal of ACM* 32(2), 374–382.
- Fudenberg, D. & D. Levine (2007). The nash-threats folk theorem with communication and approximate common knowledge in two player games. *Journal of Economic Theory* 132(1), 461 – 473.

- Fudenberg, D., D. Levine, & E. Maskin (1994). The folk theorem with imperfect public information. *Econometrica* 62(5), pp. 997–1039.
- Fudenberg, D. & E. Maskin (1986). The folk theorem in repeated games with discounting or with incomplete information. *Econometrica* 54(3), 533–554.
- Fudenberg, D. & E. Maskin (1991). On the dispensability of public randomization in discounted repeated games. *Journal of Economic Theory* 53(2), 428 – 438.
- Fudenberg, D. & J. Tirole (1991). *Game Theory*. MIT Press.
- Gibbard, A. (1973). Manipulation of voting schemes. *Econometrica* 41, 587–602.
- Gibbard, A. (1977). Manipulation of schemes that mix voting with chance. *Econometrica* 45(3), 665–681.
- Guerraoui, R., K. Huguenin, A. Kermarrec, M. Monod, & S. Prusty (2010). Lifting: lightweight freerider-tracking in gossip. In *Proceedings of the ACM/IFIP/USENIX 11th International Conference on Middleware, Middleware'10*, Bangalore, India, pp. 313–333.
- Halpern, J. & V. Teague (2004). Rational secret sharing and multiparty computation: Extended abstract. In *Proceedings of the Thirty-sixth Annual ACM Symposium on Theory of Computing, STOC '04*, Chicago, IL, USA, pp. 623–632. ACM.
- Hendon, E., H. Jacobsen, & B. Sloth (1996). The one-shot-deviation principle for sequential rationality. *Games and Economic Behavior* 12(2), 274–282.
- Holland, P. W. & S. Leinhardt (1971). Transitivity in structural models of small groups. *Small Group Research* 2(2), 107–124.
- Hughes, D., G. Coulson, & J. Walkerdine (2005). Free riding on gnutella revisited: The bell tolls? *IEEE Distributed Systems Online* 6(6), 1–.
- ISO/IEC (2006). Information technology – Security techniques – Modes of operation for an n-bit block cipher. ISO ISO/IEC 10116:2006, International Organization for Standardization, Geneva, Switzerland.
- Jun, S. & M. Ahamad (2005). Incentives in bittorrent induce free riding. In *Proceedings of the 2005 ACM SIGCOMM Workshop on Economics of Peer-to-peer Systems, P2PECON '05*, Philadelphia, Pennsylvania, USA, pp. 116–121. ACM.
- Kandori, M. & H. Matsushima (1998). Private observation, communication and collusion. *Econometrica* 66(3), pp. 627–652.

- Kermarrec, A., L. Massoulié, & A. J. Ganesh (2003). Probabilistic reliable dissemination in large-scale systems. *IEEE Trans. Parallel Distrib. Syst.* 14(3), 248–258.
- Kinatered, M. (2008). Repeated Games Played in a Network. Working papers, Fondazione Eni Enrico Mattei.
- Kreps, D. & R. Wilson (1982). Sequential equilibria. *Econometrica* 50(4), 863–894.
- Kuhn, F., N. Lynch, & R. Oshman (2010). Distributed computation in dynamic networks. In *Proceedings of the 42Nd ACM Symposium on Theory of Computing, STOC '10*, Cambridge, MA, USA, pp. 513–522. ACM.
- Laclau, M. (2012). A folk theorem for repeated games played on a network. *Games and Economic Behavior* 76(2), 711–737.
- Leitão, J., J. Pereira, & L. Rodrigues (2007). Hyparview: A membership protocol for reliable gossip-based broadcast. In *IEEE/IFIP International Conference on Dependable Systems and Networks, DSN'07*, pp. 419–428. IEEE.
- Li, H. C., A. Clement, M. Marchetti, M. Kapritsos, L. Robison, L. Alvisi, & M. Dahlin (2008). Flightpath: obedience vs. choice in cooperative services. In *Proceedings of the 8th USENIX symposium on Operating Systems Design and Implementation, OSDI'08*, San Diego, CA, USA, pp. 355–368.
- Li, H. C., A. Clement, E. L. Wong, J. Napper, I. Roy, L. Alvisi, & M. Dahlin (2006). BAR gossip. In *Proceedings of the 7th symposium on Operating Systems Design and Implementation, OSDI'06*, Seattle, WA, USA, pp. 191–204.
- Mailath, G. & L. Samuelson (2007). *Repeated Games and Reputations*. Oxford University Press.
- Matsushima, H. (2004). Repeated games with private monitoring: Two players. *Econometrica* 72(3), 823–852.
- Mazières, D. (2015). The Stellar consensus protocol: a federated model for internet-level consensus. Available at www.stellar.org/papers/stellar-consensus-protocol.pdf.
- Mokhtar, S. B., J. Decouchant, & V. Quéma (2014). AcTinG: Accurate freerider tracking in gossip. In *Proceedings of the IEEE 33rd International Symposium on Reliable Distributed Systems, SRDS'14*, pp. 291–300. IEEE.

- Moscibroda, T., S. Schmid, & R. Wattenhofer (2006a). On the topologies formed by selfish peers. In *Proceedings of the 25th Annual ACM SIGACT-SIGOPS Symposium on Principles of Distributed Computing*, PODC'06, Denver, CO, USA, pp. 133–142. ACM.
- Moscibroda, T., S. Schmid, & R. Wattenhofer (2006b). When selfish meets evil: Byzantine players in a virus inoculation game. In *Proceedings of the Twenty-fifth Annual ACM Symposium on Principles of Distributed Computing*, PODC'06, Denver, CO, USA, pp. 35–44. ACM.
- Nash, J. (1950). Equilibrium points in n-person games. *Proceedings of the National Academy of Sciences of the United States of America* 36(1), 48–49.
- Nash, J. F. (1951). Non-cooperative games. *Annals of Mathematics* 54(2), 286–295.
- Nisan, N., T. Roughgarden, E. Tardos, & V. V. Vazirani (2007). *Algorithmic Game Theory*. New York, NY, USA: Cambridge University Press.
- Obara, I. (2009). Folk theorem with communication. *Journal of Economic Theory* 144(1), 120 – 134.
- Osborne, M. & A. Rubinstein (1994). *A course in game theory*. The MIT Press.
- Piatek, M., T. Isdal, T. Anderson, A. Krishnamurthy, & A. Venkataramani (2007). Do incentives build robustness in bit torrent? In *Proceedings of the 4th USENIX conference on Networked systems design and implementation*, NSDI'07, Cambridge, MA, USA, pp. 1–1. USENIX Association.
- Piccione, M. (2002). The repeated prisoner's dilemma with imperfect private monitoring. *Journal of Economic Theory* 102(1), 70–83.
- Rahman, R., T. Vinkó, D. Hales, J. Pouwelse, & H. Sips (2011). Design space analysis for modeling incentives in distributed systems. *SIGCOMM Comput. Commun. Rev.* 41(4), 182–193.
- Rubinstein, A. & A. Wolinsky (1995). Remarks on infinitely repeated extensive-form games. *Games and Economic Behavior* 9(1), 110 – 115.
- Satterthwaite, M. (1975). Strategy-proofness and Arrow's conditions: existence and correspondence theorems for voting procedures and social welfare functions. *Journal of Economic Theory* 10, 187–217.

- Sekiguchi, T. (1997). Efficiency in repeated prisoner's dilemma with private monitoring. *Journal of Economic Theory* 76(2), 345–361.
- Selten, R. (1965). Spieltheoretische behandlung eines oligopolmodells mit nachfragerträgeit. *Zeitschrift für Gesamte Staatswissenschaft* 121, 301–324 and 667–689.
- Shamir, A. (1979). How to share a secret. *Communications of the ACM* 22, 612–613.
- Sorin, S. (1995). A note on repeated extensive games. *Games and Economic Behavior* 9(1), 116 – 123.
- Srinivasan, V., P. Nuggehalli, C. Chiasserini, & R. Rao (2003). Cooperation in wireless ad hoc networks. In *Proceedings of the 22nd Annual Joint Conference of the IEEE Computer and Communications*, INFOCOM'03, San Francisco, CA, USA, pp. 808 – 817. IEEE.
- Sugaya, T. (2011). Folk theorem in repeated games with private monitoring. Economic Theory Center Working Paper No. 011-2011, Stanford University (submitted to Journal of Economical Literature).
- Vilaca, X. & L. Rodrigues (2013). On the effectiveness of punishments in a repeated epidemic dissemination game. In *Proceedings of the 15th International Symposium on Stabilization, Safety, and Security of Distributed Systems*, SSS'13, Osaka, Japan, pp. 206–220. Springer-Verlag.
- Watts, D. J. & S. H. Strogatz (1998). Collective dynamics of small-world networks. *Nature* 393(6684), 440–442.
- Wong, E. & L. Alvisi (2013). What's a little collusion between friends? In *Proceedings of the 2013 ACM symposium on Principles of distributed computing*, PODC '13, Montreal, Quebec, Canada, pp. 240–249. ACM.
- Wong, E., I. Levy, L. Alvisi, A. Clement, & M. Dahlin (2011). Regret freedom isn't free. In *Proceedings of the 15th international conference on Principles of Distributed Systems*, OPODIS'11, Toulouse, France, pp. 80–95. Springer-Verlag.

One-shot Deviation Property for \mathcal{G}^* -OAPE

We now show that the One-Shot-Deviation Principle (Hendon, Jacobsen, & Sloth 1996) also holds for the notion of \mathcal{G}^* -OAPE. The proof is almost identical to (Hendon, Jacobsen, & Sloth 1996), so we only include a sketch. We use the same notation of Chapter 5, i.e., $\vec{\sigma}^*|_{I_i, a_i}$ denotes the strategy profile identical to $\vec{\sigma}^*$ at every information set, except i deterministically follows a_i at I_i .

Proposition 24. One-Shot-Deviation Principle. *A protocol $\vec{\sigma}^* \in \Sigma$ is a \mathcal{G}^* -OAPE if and only if there exists μ^* consistent with $\vec{\sigma}^*$ and \mathcal{G}^* such that, for every $G \in \mathcal{G}^*$, round m , agent i , round- m $I_i \in \mathcal{I}_i(G)$, and actions $a_i^*, a_i \in \mathcal{A}_i(G^m)$, we have $u_i(\vec{\sigma}^*|_{I_i, a_i^*} | G, I_i) \geq u_i(\vec{\sigma}^*|_{I_i, a_i} | G, I_i) | G, I_i$.*

Proof. (Sketch) We proceed as in (Hendon, Jacobsen, & Sloth 1996), except that we fix $G \in \mathcal{G}^*$. The implication is clear: since $\vec{\sigma}^*$ is a \mathcal{G}^* -OAPE, agent i cannot increase its expected utility by following $\vec{\sigma}^*|_{I_i, a_i'}$ instead of $\vec{\sigma}^*$, and $u_i(\vec{\sigma}^*|_{I_i, a_i^*} | G, I_i) = u_i(\vec{\sigma}^* | G, I_i)$. As for the reverse implication, fix round- m I_i . Suppose that the right-hand side of the proposition holds and that there is σ_i such that for some $\epsilon > 0$

$$u_i((\sigma_i, \vec{\sigma}_{-i}^*) | G, I_i) - u_i(\vec{\sigma}^* | G, I_i) = 2\epsilon. \quad (\text{A.1})$$

Define $m' > m$ such that $\delta^{m'-m} \gamma n / (1 - \delta) < \epsilon$. Let σ_i' be identical to σ_i at every round- m'' information set for $m'' \leq m'$, but is identical to σ_i^* at every round- m'' information set for $m'' > m'$. It can be shown using the right-hand side and backwards induction that

$$u_i(\vec{\sigma}^* | G, I_i) \geq u_i((\sigma_i', \vec{\sigma}_{-i}^*) | G, I_i) \geq u_i((\sigma_i, \vec{\sigma}_{-i}^*) | G, I_i) - \epsilon.$$

This contradicts (A.1), proving the reverse implication. This concludes the proof. □