Characterizing Asynchronous Message-Passing Models Through Rounds

Adam Shimi
IRIT – Université de Toulouse, 2 rue Camichel, F-31000 Toulouse, France
http://www.irit.fr
adam.shimi@irit.fr

Aurélie Hurault
IRIT – Université de Toulouse, 2 rue Camichel, F-31000 Toulouse, France
http://www.irit.fr
aurelie.hurault@irit.fr

Philippe Quéinnec
IRIT – Université de Toulouse, 2 rue Camichel, F-31000 Toulouse, France
http://www.irit.fr
philippe.queinnec@irit.fr

Abstract

Message-passing models of distributed computing vary along numerous dimensions: degree of synchrony, kind of faults, number of faults... Unfortunately, the sheer number of models and their subtle distinctions hinder our ability to design a general theory of message-passing models. One way out of this conundrum restricts communication to proceed by round. A great variety of message-passing models can then be captured in the Heard-Of model, through predicates on the messages sent in a round and received during or before this round. Then, the issue is to find the most accurate Heard-Of predicate to capture a given model. This is straightforward in synchronous models, because waiting for the upper bound on communication delay ensures that all available messages are received, while not waiting forever. On the other hand, asynchrony allows unbounded message delays. Is there nonetheless a meaningful characterization of asynchronous models by a Heard-Of predicate?

We formalize this characterization by introducing Delivered collections: the collections of all messages delivered at each round, whether late or not. Predicates on Delivered collections capture message-passing models. The question is to determine which Heard-Of predicates can be generated by a given Delivered predicate. We answer this by formalizing strategies for when to change round. Thanks to a partial order on these strategies, we also find the "best" strategy for multiple models, where "best" intuitively means it waits for as many messages as possible while not waiting forever. Finally, a strategy for changing round that never blocks a process forever implements a Heard-Of predicate. This allows us to translate the order on strategies into an order on Heard-Of predicates. The characterizing predicate for a model is then the greatest element for that order, if it exists.

2012 ACM Subject Classification Theory of computation → Distributed computing models

Keywords and phrases Message-passing, Asynchronous Rounds, Dominant Strategies, Failures

Digital Object Identifier 10.4230/LIPIcs.OPODIS.2018.0

Funding This work was supported by project PARDI ANR-16-CE25-0006.

22st International Conference on Principles of Distributed Systems (OPODIS 2018).
Editors: Jiannong Cao, Faith Ellen, Luis Rodrigues, and Bernardo Ferreira; Article No. 0; pp. 0:1–0:20

Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

1 Introduction

1.1 Motivation

Even when restricted to message-passing, distributed computing spawns a plethora of models: with various degrees of synchrony, with different kinds of faults, with different failure detectors... Although some parameters are quantitative, such as the number of faults, the majority are qualitative instead, for example the kinds of faults. Moreover, message-passing models are usually defined by a mix of mathematical formalism and textual description, with crucial details nested deep inside the latter. This is why these models resist unification into a theory of distributed computing, and why results in the field are notoriously hard to organize, use and extend.

One solution requires constraining communication to proceed by round: each process repeatedly broadcasts a message with its current round number, waits for as many messages as possible bearing this round number, and changes round by computing both its next state and next message. The variations between models are then captured by the dynamic graph specifying, for each round, from which processes each process received a message with this round number before the end of its round; this fits the concept of dynamic network from Kuhn and Oshman [11]. Nonetheless, we will privilege the perspective of Charron-Bost and Schiper Heard-Of model [4], which places itself more at the level of processes. Here, the Heard-Of collection of an execution contains, for each round r and each process j, the set of processes from which j received a message sent in round r before going to round $r + 1$. Then, a predicate on Heard-Of collections characterizes a message-passing model.

Yet rounds don't remove the complexities and subtleties of message-passing models – they just shift them to the characterization of a given model by a Heard-Of predicate. This characterization depends on how rounds can be implemented in the underlying model. In the synchronous case, processes progress in lock-step, and every message that will ever be received is received during its corresponding round. Hence, the Heard-Of predicate characterizing a synchronous model simply specifies which messages can be lost. In asynchronous models on the other hand, messages can be late, and thus the distance between the round numbers of processes is unbounded. The combination of these uncertainties implies that processes do not know which messages will be delivered and when. Thus, there is a risk of not waiting for useful messages that will eventually arrive, and to wait forever for messages that never will.

To the best of our knowledge, there is no systematic study of the Heard-Of predicates generated by various asynchronous message-passing models. Because it is a crucial step in unifying distributed computing's menagerie of models through rounds, we also believe this topic to be of importance.

1.2 Approach and Overview

As hinted above, the difficulty lies in the potential discrepancy between messages delivered on time – captured by Heard-Of collections – and messages delivered at all. We want to determine the former, but it is considerably easier to specify the latter for an operational model: list the messages that will eventually arrive. We therefore center our formalization around Delivered collections, infinite sequences of communication graphs capturing, for each round, all messages sent at this round and eventually delivered for a given operational model. The question is thus to characterize by a Heard-Of predicate which messages can be waited for when the deliveries are those from the Delivered predicate.

From these Delivered collections, we build runs representing the different scheduling

of deliveries and changes of round. Some of these runs, called valid, define a Heard-Of collection; invalid runs have processes blocked forever at some round. We filter the latter thanks to strategies: sets of local states for which a process is allowed to change round. Runs for a strategy must also satisfy a fairness condition ensuring that if a process can change round continuously, it does. Strategies with only valid runs for a Delivered predicate, that is strategy implementing a Heard-Of predicate, are called valid.

The next question is how to choose a valid strategy and the corresponding predicate, as characterizing a Delivered predicate and its underlying model? We answer by taking the strategy generating the Heard-Of predicate that is the smallest overapproximation of the Delivered predicate. From this intuition, we define a partial order on valid strategies called domination; a characterizing strategy is a greatest element for this order, and the characterizing predicate is the one it generates.

The results obtained with this approach are threefold:

- The formalization itself, with a complete example: the asynchronous message-passing model with reliable communication and at most F permanent crashes.
- The study of carefree strategies, the ones depending only on messages from the current round. This restricted class is both well-behaved enough to always have a unique dominating strategy, and expressive enough to capture interesting Delivered predicates.
- The study of reactionary strategies, the ones depending only on messages from past and current rounds. Here too we show well-behavior of this class as well as an example where reactionary is needed for domination, and another one where it is insufficient.

Along these results, we also formally prove the characterization of asynchronous models by Heard-Of predicates given by Charron-Bost and Schiper [4].

We begin by the formalization in **Section 2**, while **Section 3** introduces a fully developed example: the asynchronous message-passing model with reliable communication and at most F permanent crashes. **Section 4** explores carefree strategies in terms of well-behavior and expressivity, closing with the example with at most B failed broadcasts per round, a Delivered predicate dominated by a carefree strategy. We follow by studying reactionary strategies in **Section 5**. Here again, well-behavior and expressivity are examined, followed by the example of at most F permanent initial crashes. **Section 6** and **Section 7** then conclude the paper with a discussion of related works, the value of our results and some perspectives.

2 Formalization

All our abstractions revolve around infinite sequences of graphs, called collections. A Delivered collection maps each round r and process j to the set of processes from which j receive a message sent at r. A Heard-Of collection maps each round r and process j to the set of processes from which j received, before going to round $r+1$, the message sent at r. The difference lies in considering all deliveries for Delivered collections, but only the deliveries before the end of the round of the receiver for Heard-Of collections.

▶ **Definition 1** (Collections and Predicates)**.** Let Π a set of processes. $Col : (\mathbb{N}^* \times \Pi) \mapsto \mathcal{P}(\Pi)$ is either a **Delivered collection** or a **Heard-Of collection** for Π, depending on the context.

In the same way, $Pred : \mathcal{P}((\mathbb{N}^* \times \Pi) \mapsto \mathcal{P}(\Pi))$ is either a **Delivered predicate** or a **Heard-Of predicate** for Π.

For a given Col, the kernel of round r are the processes from which everyone receives a message for this round $K_{Col}(r) \triangleq \bigcap_{j \in \Pi} Col(r, j)$.

2.1 Runs and Strategies

The behavior of processes is classically specified by runs, sequences of both states and transitions satisfying some restricting conditions: messages cannot be delivered before the round they are sent and are delivered only once. Given a Delivered collection, we additionally require that the delivered messages are exactly the ones in the Delivered collection.

As we only care about which messages can be waited for, ignoring the content of messages or the underlying computation, we limit the state to the received messages and the round.

▶ **Definition 2** (Run). Let Π be a set of n processes. Let $Q = (\mathbb{N} \times \mathcal{P}(\mathbb{N}^* \times \Pi))$ the set of process states. The first element is the round of a process (written $q.round$ for $q \in Q$) and the second is the set of pairs $\langle$round it was sent, sender$\rangle$ for each delivered message (written $q.received$ for $q \in Q$). Let the set of transitions $T = \{next_j \mid j \in \Pi\} \cup \{deliver(r,k,j) \mid r \in \mathbb{N}^* \wedge k, j \in \Pi\} \cup \{end\}$. $next_j$ is the transition for j changing round, $deliver(r,k,j)$ is the transition for the delivery to j of the message sent by k in round r, and end is the transition to end a finite run. For $qt \in Q^n \times T$ and $j \in \Pi$, we write $qt.state$ for the state, $qt.state.j$ for the local state of j and $qt.transition$ for the transition. Finally, let $(Q^n \times T)^\infty$ be the set of finite and infinite words on the set $(Q^n \times T)$. Then, $t \in (Q^n \times T)^\infty$ is a **run** $\triangleq$

- **(Initial state)** $t[0].state = \langle 1, \emptyset \rangle^n$
- **(Transitions)** $\forall i \in [0, size(t)) :$

$$\left(\begin{array}{l} \exists r \in \mathbb{N}^*, \exists k, j \in \Pi : t[i].transition = deliver(r,k,j) \implies \\ \quad t[i+1].state = t[i].state \\ \qquad\qquad Except\ t[i+1].state.j.received = t[i].state.j.received \cup \{(r,k)\} \\ \wedge\ \exists j \in \Pi : t[i].transition = next_j \implies \\ \quad t[i+1].state = t[i].state\ Except\ t[i+1].state.j.round = t[i].state.j.round + 1 \\ \wedge\ (t[i].transition = end) \implies (i = size(t) - 1) \end{array} \right)$$

- **(Delivery after sending)** $\forall i \in [0, size(t)) :$ $t[i].transition = deliver(r,k,j) \implies t[i].state.k.round \geq r$
- **(Unique delivery)** $\forall \langle r,k,j \rangle \in (\mathbb{N}^* \times \Pi \times \Pi) : (\exists i \in [0, size(t)) : t[i].transition = deliver(r,k,j)) \implies (\forall i' \in [0, size(t)) \setminus \{i\} : t[i'].transition \neq deliver(r,k,j))$

Let $CDel$ be a Delivered collection. Then, $runs(CDel)$, the **runs of** $CDel \triangleq$

$$\left\{ t\ a\ run \;\middle|\; \begin{array}{l} \forall \langle r,k,j \rangle \in \mathbb{N}^* \times \Pi \times \Pi : \\ \quad (k \in CDel(r,j) \wedge \exists i \in [0, size(t)) : t[i].state.j.round \geq r) \\ \quad \iff \\ \quad (\exists i \in [0, size(t)) : t[i].transition = deliver(r,k,j)) \end{array} \right\}$$

For $PDel$ a Delivered predicate, we write $runs(PDel) = \{runs(CDel) \mid CDel \in PDel\}$.

Our definition of runs does not force processes to change rounds. This contradicts our intuition about a system using rounds: processes should keep on "forever", or at least as long as necessary. In a valid run, processes change round an infinite number of times.

▶ **Definition 3** (Validity). A run t is **valid** $\triangleq \forall j \in \Pi : |\{i \in \mathbb{N} \mid t[i].transition = next_j\}| = \aleph_0$.

Valid run are necessarily infinite. Yet the definition above allows finite runs thanks to the *end* transition. This is used in proofs by contradiction which imply the manipulation of invalid runs and thus potentially finite ones.

Next, we define the other building block of our approach: strategies. They are simply sets of local states, representing the states where processes can change round.

▶ **Definition 4** (Strategy). $f : \mathcal{P}(Q)$ is a **strategy**.

Combining a Delivered predicate and a strategy results in runs capturing the behavior of processes for the corresponding Delivered collections when following the strategy. In these runs, processes can change round only when allowed by the strategy, and must also do so if the strategy allows it continuously.

▶ **Definition 5** (Runs Generated by a Strategy)**.** Let f be a strategy and t a run. t is a **run generated by** $f \triangleq t$ satisfies the following:

- **(Next only if allowed)** $\forall i \in [0, size(t)), \forall j \in \Pi : (t[i].transition = next_j \implies t[i].state.j \in f)$
- **(Infinite fairness of next)** If t is infinite, then $\forall j \in \Pi : |\{i \in \mathbb{N} \mid t[i].transition = next_j\}| < \aleph_0 \implies |\{i \in \mathbb{N} \mid t[i].state.j \notin f\}| = \aleph_0$
- **(Finite fairness of next)** If t is finite, then $\forall j \in \Pi : t[size(t) - 1].state.j \notin f$.

For a Delivered predicate $PDel$, we note $runs_f(PDel) = \{t \text{ a run} \mid t \text{ generated by } f \wedge t \in runs(PDel)\}$.

From that point, it is clear that a well-behaved strategy is such that all its runs are valid.

▶ **Definition 6** (Valid Strategy)**.** Let $PDel$ a Delivered predicate and f a strategy. f is a **valid strategy** for $PDel \triangleq \forall t \in runs_f(PDel) : t$ is a valid run.

Validity guarantees an infinite number of complete rounds for every run of the strategy. This ensures that a run defines a Heard-Of collection, as we see next.

2.2 From Delivered Collections to Heard-Of Collections

Recall that the difference between a Heard-Of and a Delivered collection is that the latter takes into account all delivered messages, while the former only considers messages from a round if they were received before or during the corresponding round of the receiver.

If a run is valid, then all processes have infinitely many rounds, and thus it defines a Heard-Of collection through its behavior.

▶ **Definition 7** (Heard-Of Collection of a Valid Run)**.** Let t a valid run. Then, CHO_t is the **Heard-Of collection of** $t \triangleq$

$$\forall r \in \mathbb{N}^*, \forall j \in \Pi : CHO_t(r, j) = \left\{ k \in \Pi \;\middle|\; \exists i \in \mathbb{N} : \begin{pmatrix} & t[i].state.j.round = r \\ \wedge & t[i+1].j.state.round = r + 1 \\ \wedge & \langle r, k \rangle \in t[i].state.j.received \end{pmatrix} \right\}$$

It is useful to go the other way, and extract from a Heard-Of collection some canonical valid run generating it. Our choice is a run where processes change round in lockstep, every message from a Heard-Of set is delivered in the round where it was sent, and every late message is delivered in the round following the one where it was sent.

▶ **Definition 8** (Standard Run of a Heard-Of collection)**.** Let cho be a Heard-Of collection. For $r > 0$, let $onTimeMes_r$ be a permutation of $\{deliver(r, k, j) \mid k, j \in \Pi \wedge k \in cho(r, j)\}$, $lateMes_r$ a permutation of $\{deliver(r-1, k, j) \mid k, j \in \Pi \wedge k \notin cho(r-1, j)\}$, and $nexts$ be a permutation of $\{next_j \mid j \in \Pi\}$. Then, the run starting at the initial state and with transitions defined by the word $\prod_{r>0} (lateMes_r.onTimeMes_r.nexts)$ is a **standard run** of cho.

This canonical run is a run of any Delivered predicate containing the collection where every message is delivered. This collection captures the case where no failure occurs: every process always broadcasts and no message is lost. Having this collection in a Delivered predicate ensures that although faults might happen, they are not forced to do so.

▶ **Lemma 9** (Standard Run is a Run for Total Collection). *Let cho be a Heard-Of collection and let PDel be a Delivered predicate containing the total collection* $CDel_{total}$ *defined by* $\forall r > 0, \forall j \in \Pi : CDel_{total}(r, j) = \Pi$. *Then, a standard run* t *of cho is a run of PDel.*

Proof. First, t is a run since it satisfies the four constraints defining a run:

- The first state is the initial state by definition.
- We only gave the transitions, and the states changes as mandated by the transition function.
- Every message from round r is delivered after either r or $r+1$ *next* transitions for the sender, which ensures it is delivered after being sent.
- Every sent message is delivered either during the round it was sent or during the next one, and thus delivered only once.

Furthermore, one message from each process is eventually delivered to everyone for each round in t, which means that t is a run of the total Delivered collection, and thus a run of *PDel*. ◀

▶ **Definition 10** (Heard-Of Predicate Generated by Strategy). If f is a valid strategy for *PDel*, we write $PHO_f(PDel)$ for the Heard-Of collections of the runs generated by f for *PDel*: $PHO_f(PDel) \triangleq \{CHO_t \mid t \in runs_f(PDel)\}$.

Every valid strategy generates a Heard-Of predicate from the Delivered predicate. We now have a way to go from a Delivered predicate to a Heard-Of one: design a valid strategy for the former that generates the latter. But we still have not answered the original question: among all the Heard-Of predicates one can generate from a given Delivered predicate, which one should we consider as the characterization of the Delivered predicate?

First, remark that all Delivered collections can be generated as Heard-Of collections by a valid strategy: simply deliver all messages from a round before changing the round of a process – the change of round must eventually happen by validity of the strategy. Thus, every Heard-Of predicate generated from a Delivered one is an overapproximation of the latter: for any strategy f, $PDel \subseteq PHO_f(PDel)$. But we want to receive as many messages as possible on time, that is to be as close as possible to the original Delivered predicate. The characterizing Heard-Of predicate is thus the smallest such overapproximation, if it exists.

We formalize this intuition by defining a partial order on valid strategies for a Delivered predicate capturing the implication of the generated Heard-Of predicates. One strategy dominates another if the Heard-Of set it generates is included in the one generated by the other. Dominating strategies are then the greatest elements for this order. By definition of domination, all dominating strategies generate the same dominating Heard-Of predicate, which characterizes the Delivered predicate.

▶ **Definition 11** (Domination Order, Dominating Strategy and Dominating Predicate). Let *PDel* be a Delivered predicate and let f and f' be two valid strategies for *PDel*. Then, f **dominates** f' for *PDel*, written $f' \prec_{PDel} f \triangleq PHO_{f'}(PDel) \supseteq PHO_f(PDel)$.

A greatest element for $\prec_{PDel}$ is called a **dominating strategy** for *PDel*. Given such a strategy f, the **dominating predicate** for *PDel* is then $PHO_f(PDel)$.

3 A Complete Example: At Most F Crashes

To provide a more concrete intuition, we turn to an example: the message-passing model with asynchronous and reliable communication, and at most F permanent crashes. To find the corresponding Delivered predicate, we characterize which messages are delivered in a round

execution of this model: all the messages sent by a process before it crashes are delivered; it sends no message after; at the round where it crashes, there may be an incomplete broadcast if the process crashes in the middle of it. Additionally, at most F processes can crash.

▶ **Definition 12** ($PDel^F$)**.** The Delivered predicate $PDel^F$ for the asynchronous model with reliable communication and at most F permanent crashes $\triangleq$
$$\left\{ CDel \in (\mathbb{N}^* \times \Pi) \mapsto \mathcal{P}(\Pi) \;\middle|\; \forall r > 0, \forall j \in \Pi : \begin{array}{ll} & |CDel(r,j)| \geq n - F \\ \wedge & CDel(r+1,j) \subseteq K_{CDel}(r) \end{array} \right\}.$$

The folklore strategy for this model is to wait for at least $n - F$ messages before allowing the change of round.

▶ **Definition 13** (waiting for $n - F$ messages)**.** The strategy to wait for $n - F$ messages is: $f_{n-F} \triangleq \{q \in Q \mid |\{k \in \Pi \mid \langle q.round, k\rangle \in q.received\}| \geq n - F\}$

To see why this strategy is used in the literature, simply remark that at least $n - F$ messages must be delivered to each process at each round. Thus, waiting for that many messages ensures that no process is ever blocked. Rephrased with the concepts introduced above, f_{n-F} is a valid strategy for $PDel^F$.

▶ **Lemma 14** (Validity of f_{n-F})**.** *f_{n-F} is valid for $PDel^F$.*

Proof. We proceed by contradiction: **Assume** f_{n-F} is invalid for $PDel^F$. Thus, there exists $t \in runs_{f_{n-F}}(PDel^F)$ invalid. Because t is infinite, the problem is either the infinite fairness of *next* or a *next* done when f_{n-F} does not allow it. Each *next* transition being played an infinite number of times, we conclude that some *next* is played while f_{n-F} does not allow it. Let r be the smallest round where it happens and j be a process blocked at r in t. Let also $CDel_t$ be a Delivered collection of $PDel^F$ such that $t \in runs(CDel_t)$.

We know by definition of $PDel^F$ that $|CDel_t(r,j)| \geq n - F$. The minimality of r and the fact that $t \in runs(CDel)$ then ensure that all messages in this Delivered set are delivered at some point in t. By definition of f_{n-F}, the transition $next_j$ is then available from this point on. This **contradicts** the fact that j cannot change round at this point in t. ◀

The Heard-Of predicate generated by f_{n-F} was first given by Charron-Bost and Schiper [4] as a characterization the asynchronous model with reliable communication and at most F crashes. The intuition behind it is that even in the absence of crashes, we can make all processes change round by delivering any set of at least $n - F$ messages to them.

▶ **Theorem 15** (Heard-Of Characterization of f_{n-F})**.**
$PHO_{f_{n-F}}(PDel^F) = \{cho \in (\mathbb{N}^* \times \Pi) \mapsto \mathcal{P}(\Pi) \mid \forall r \in \mathbb{N}^*, \forall j \in \Pi : |cho(r,j)| \geq n - F\}$.

Proof. First, we show $\subseteq$. Let $cho \in PHO_{f_{n-F}}(PDel^F)$ and $t \in runs_{f_{n-F}}(PDel^F)$ a run of f_{n-F} generating cho. By definition of the runs of f_{n-F}, processes change round only when they received at least $n - F$ messages from the current round, which implies that $\forall r \in \mathbb{N}^*, \forall j \in \Pi : |cho(r,j)| \geq n - F$.

Then, we show $\supseteq$. Let cho a Heard-Of collection over Π such that $\forall r \in \mathbb{N}, \forall j \in \Pi : |cho(r,j)| \geq n - F$. Let t be a standard run of cho; since $PDel^F$ contains the total collection, t is a run of $PDel^F$ by Lemma 9. To prove this is also a run of f_{n-F}, we proceed by contradiction: **Assume** it is not: because t is infinite, the problem is either the infinite fairness of *next* or a *next* done when f_{n-F} does not allow it. Each *next* transition being played an infinite number of times in a standard run, the only possibility left is the second one: some *next* transition in t is done while f_{n-F} does not allow the corresponding process

to change round. Let r be the smallest round where this happens, and j one of the concerned processes at round r. By definition of t as a standard run, j received all messages from $cho(r,j)$ before the problematic *next*. And $|cho(r,j)| \geq n-F$ by hypothesis. By definition of f_{n-F}, the transition $next_j$ is then available from this point on. This **contradicts** the fact that j cannot change round at this point. We conclude that $cho \in PHO_{f_{n-F}}(PDel^F)$. ◀

Finally, we want to vindicate the folklore intuition about this strategy: that it is optimal in some sense. Intuitively, waiting for more than $n-F$ messages per round means risking waiting forever, and waiting for less is wasteful. Our domination order captures this concept of optimality: we show that f_{n-F} is indeed a dominating strategy for $PDel^F$. Therefore, $PHO_{f_{n-F}}(PDel^F)$ is the dominating predicate for $PDel^F$.

To do so, we introduce another canonical run, this time for the combination of a Delivered collection and a strategy. This run consists in successive iterations where all messages sent and not yet delivered are delivered, and then all processes which are allowed to change round by the strategy do. This captures the run where every message is delivered as early as possible after being sent.

▶ **Definition 16** (Earliest Run of Strategy for Delivered Collection)**.** Let $CDel$ be a Delivered collection and f be a strategy. We now define for $r>0$ the sets $dels_r$ and $nexts_r$, as well as the states $qDels_r$ and $qNexts_r$. The two sets are respectively the set of deliveries at iteration r and the set of *nexts* at iteration r. As for the states, they respectively capture the state of the system just before the first delivery of the r-ith iteration and just before the first *next* of this iteration. We define them all by the following recurrence:

- **State at iteration r after nexts (of r)** We have $qDels_1 = \langle 1, \emptyset \rangle^n$ and
$$\forall r>1 : \left(\begin{array}{ll} \forall j \in \Pi : & \\ \quad \textit{if } next_j \in nexts_r & \left(\begin{array}{l} qDels_r.j = qNexts_{r-1}.j \ \textit{Except} \\ \quad qDels_r.j.round = qNexts_{r-1}.j.round + 1 \end{array} \right) \\ \quad \textit{otherwise} & qDels_r.j = qNexts_{r-1}.j \end{array} \right)$$
- **Delivers at iteration r**
We have $dels_1 = \{deliver(1,k,j) \mid k \in CDel(1,j)\}$ and $\forall r>1 : dels_r = \{deliver(r',k,j) \mid r' = qDels_r.j.round \wedge r' \neq qNexts_{r-1}.j.round \wedge k \in CDel(r',j)\}$.
- **State at iteration r after deliveries**
$$\forall r>0 : \left(\begin{array}{l} \forall j \in \Pi : qNexts_r.j = qDels_r.j \ \textit{Except} \\ \quad qNexts_r.j.received = qDels_r.j.received \cup \{\langle r,k \rangle \mid \langle r,k,j \rangle \in dels_r\} \end{array} \right)$$
- **Nexts allowed at iteration r**
$\forall r>0 : nexts_r = \{next_j \mid qNexts_r.j \in f\}$.

For all $r>0$, let $Wdels_r$ be a permutation of $dels_r$, and $Wnexts_r$ a permutation of $nexts_r$. Then, the run starting at the initial state and with the following transitions: $\sum_{r>0} (Wdels_r.Wnexts_r)$ – with an *end* at the end if it is a finite run – is an **earliest run** of f for $CDel$.

▶ Lemma 16b (Earliest Run is a Run). Let $CDel$ be a Delivered collection and f be a strategy. Then, an earliest run of f for $CDel$ is a run of f.

Proof. First, t is a run since it satisfies the four constraints defining a run:

- The first state is the initial state by definition.
- Transitions change state as required by definition.
- Every message from round r is delivered while the emitter is in round r, which ensures it is delivered after being sent.
- Every sent message is delivered during the round it was sent, and thus delivered only once.

It is also a run of f because it satisfies the three constraints:

- By definition, processes only change round when allowed by f.
- If t is infinite, then a process which has only finitely many changes of round is blocked at some round forever; this means that there are infinitely many iterations where f does not continuously enable the change of round.
- If t is finite on the other hand, this means that for all iterations from a point on, no process is allowed to change round, and thus that in their last state all processes are forbidden to change round by f.

We therefore deduce t is a run of f. ◀

By combining standard and earliest runs, we show that any valid strategy for $PDel^F$ is dominated by f_{n-F} and thus that $PHO_{f_{n-F}}(PDel^F)$ is the dominating predicate for $PDel^F$.

▶ **Theorem 17** (f_{n-F} Dominates $PDel^F$)**.** *f_{n-F} dominates $PDel^F$.*

Proof. Let f be a valid strategy for $PDel^F$; we now prove that $f \prec_{PDel^F} f_{n-F}$, that is $PHO_{f_{n-F}}(PDel^F) \subseteq PHO_f(PDel^F)$. Let $cho \in PHO_{f_{n-F}}(PDel^F)$, and let t be a standard run of cho. Since $PDel^F$ contains the total collection, t is a run of $PDel^F$ by Lemma 9. We only need to prove that it is also a run of f to conclude.

We do so by contradiction. **Assume** t is not a run of f: because t is infinite, the problem is either the infinite fairness of *next* or a *next* done by a process j when f does not allow it. Each *next* transition being played an infinite number of times in a standard run, the only possibility left is the second one. At the point of the forbidden *next*, by definition of a standard run, j has received every message from previous rounds, and all messages from $cho(r, j)$. By application of Theorem 15 and cho being in $PHO_{f_{n-F}}(PDel^F)$, $cho(r, j)$ contains at least $n - F$ processes.

Let $CDel_{block}$ be the Delivered collection where all processes from which j did not receive a message at the problematic *next* in t stop sending messages from this round on:

$$\forall r' > 0, \forall k \in \Pi : CDel_{block}(r', k) = \begin{cases} \Pi & \text{if } r' < r \\ cho(r, j) & \text{otherwise} \end{cases}$$

This is a Delivered collection of $PDel^F$: processes that stop sending messages never do again, and at most F processes do so because $cho(r, j)$ contains at least $n - F$ processes.

Let t_{block} be an earliest run of f for $CDel_{block}$. This is a run of f, by Lemma 16b. We then have two possibilities.

- During one of the first $r-1$ iterations of t_{block}, there is some process which cannot change round. Let r' be the smallest iteration where it happens, and k be a process unable to change round at this iteration. By minimality of r', all processes arrive at round r', and by symmetry of $CDel_{block}$ they all receive the same messages as k. Thus, all processes are blocked at round r', there are no more next or deliveries, and t_{block} is therefore invalid.
- For the first $r - 1$ iterations, all processes change round. Thus, every one arrives at round r. By definition of an earliest run, all messages from the round are delivered before any *next*. The symmetry of $CDel_{block}$ also ensures that every process received the same messages, that is all messages from round $< r$ and all messages from $cho(r, j)$. These are exactly the messages received by j in t at round r. But by hypothesis, j is blocked in this state in t. We thus deduce that all processes are blocked at round r in t_{block}, and thus that it is an invalid run.

Either way, we deduce that f is invalid, which is a **contradiction**. ◀

This means that when confronted with a model captured by $PDel^F$, there is no point in remembering messages from past rounds – and messages from future rounds are simply

buffered. Intuitively, messages from past rounds are of no use in detecting crashes in the current round. As for messages from future rounds, they could serve to detect that a process has not crashed when sending its messages from the current round. This does not alter the Heard-Of predicate because nothing forces messages from future rounds to be delivered early, and thus there is no way to systematically use the information from future rounds.

4 Carefree Strategies

We now turn to more general results about Delivered predicates and strategies. We focus first on a restricted form of strategies, the carefree ones: they depend only on the received messages from the current round. For example, f_{n-F} is a carefree strategy. These are quite simple strategies, yet they can be dominating, as shown for f_{n-F} and $PDel^F$.

4.1 Definition and Expressiveness Results

▶ **Definition 18** (Carefree Strategy)**.** Let f be a strategy and, $\forall q \in Q$, let $cfree(q) = \{k \in \Pi \mid \langle q.round, k\rangle \in q.received\}$. f is a **carefree strategy** $\triangleq \forall q, q' \in Q : cfree(q) = cfree(q') \implies (q \in f \iff q' \in f)$.

For f a carefree strategy, let $Nexts_f \triangleq \{cfree(q) \mid q \in f\}$. It uniquely defines f.

Thus a carefree strategy can be defined by a set of sets of processes: receiving a message from all processes in any of those set makes the strategy authorize the change of round. This gives us a simple necessary condition on such a strategy to be valid: its *Nexts* set must contains all Delivered set from the corresponding Delivered predicate. If it does not, then an earliest run of any collection containing a Delivered set not in the *Nexts* would be invalid.

This simple necessary condition also proves sufficient.

▶ **Lemma 19** (Validity of Carefree)**.** *Let PDel be a Delivered predicate and f a carefree strategy. Then, f is valid for PDel $\iff \forall CDel \in PDel,\ \forall r > 0, \forall j \in \Pi : CDel(r, j) \in Nexts_f$.*

Proof. ($\Rightarrow$) Let f be valid for *PDel*. We show by contradiction that it satisfies the right-hand side of the above equivalence. **Assume** there is $CDel \in PDel, r > 0$ and $j \in \Pi$ such that $CDel(r, j) \notin Nexts_f$. Then, let t be an earliest run of f for *CDel*. This is a run of f by Lemma 16b.

The sought contradiction is reached by proving that t is invalid. To do so, we split according to two cases.

- During one of the first $r-1$ iterations of t, there is some process which cannot change round. Let r' be the smallest iteration where it happens, and k be a process unable to change round at the r'-ith iteration.
  By minimality of r', all processes arrive at round r' in t; by definition of an earliest run, all messages for k from round r' are delivered before the *next* for the iteration. Let q the local state of k at the first *next* in the r'-ith iteration, and let q' be any local state of k afterward. The above tells us that as long as $q'.round = q.round$, we have $cfree(q) = cfree(q')$ and thus $q' \notin f$. Therefore, k can never change round while at round r'.
  We conclude that t is invalid.
- For the first $r-1$ iterations, all processes change round. Thus every one arrives at round r in the $r-1$-ith iteration. By definition of an earliest run, all messages from the round are delivered before any *next*. By hypothesis, j cannot change round because its Delivered set is not in $Nexts_f$. Let q the local state of j at the first *next* in the r-ith

iteration, and let q' be any local state of j afterward. The above tells us that as long as $q'.round = q.round$, we have $cfree(q) = cfree(q') = CDel(q.round, j)$ and thus $q' \notin f$. Therefore, j can never change round while at round r.
Here too, t is invalid.

Either way, we reach a **contradiction** with the validity of f.
($\Leftarrow$) Let $PDel$ and f such that $\forall CDel \in PDel, \forall r > 0, \forall j \in \Pi : CDel(r, j) \in Nexts_f$. We show by contradiction that f is valid.

Assume the contrary: there is some $t \in runs_f(PDel)$ which is invalid. Thus, there are some process blocked at a round forever in t. Let r be the smallest such round, and j be a process blocked at round r in t. By minimality of r, all processes arrive at round r. By definition of a run of $PDel$, there is a $CDel \in PDel$ such that t is a run of $CDel$. Thus, eventually all messages from $CDel(r, j)$ are delivered.

From this point on, f allows j to change round by definition, and the fairness of *next* imposes that j does at some point. We conclude that j is not blocked at round r in t, which **contradicts** the hypothesis. ◀

Carefree strategies are elegant, but they also have some drawbacks: mainly that the Heard-Of predicates they can implement are quite basic. Precisely, when the Delivered predicate contains the total collection, they implement predicates where the Heard-Of collections are all possible combinations of Delivered sets from the original Delivered predicate.

▶ **Theorem 20** (Heard-Of Predicates of Carefree Strategy). *Let $PDel$ be a Delivered predicate containing the total collection, and let f be a valid carefree strategy for $PDel$. Then, $\forall cho$ a Heard-Of collection for $\Pi : cho \in CHO_f(PDel) \iff \forall r > 0, \forall j \in \Pi : cho(r, j) \in Nexts_f$.*

Proof. ($\Rightarrow$) This direction follows from the definition of a carefree strategy: it allows changing round only when the messages received from the current round form a set in its *Nexts*.

($\Leftarrow$) Let cho be a Heard-Of collection for Π such that $\forall r > 0, \forall j \in \Pi : cho(r, j) \in Nexts_f$. Let t be a standard run of cho. It is a run by Lemma 9. It is also a run of f because at each round, processes receive messages from a set in $Nexts_f$ and are thus allowed by f to change round. We conclude that $cho \in PHO_f(PDel)$. ◀

Finally, carefree strategies always have a dominating element for a given Delivered predicate. This is because the strategy waiting for exactly the Delivered sets of the predicate waits for more messages than any other valid carefree strategy, by Lemma 19.

▶ **Theorem 21** (Always a Dominating Carefree Strategy). *Let $PDel$ be a Delivered predicate and let f_{cfDom} be the carefree strategy with $Nexts_f = \{CDel(r, j) \mid CDel \in PDel \wedge r > 0 \wedge j \in \Pi\}$. f_{cfDom} dominates all carefree strategies for $PDel$.*

Proof. First, f_{cfDom} is valid for $PDel$ by application of Lemma 19.

As for domination, we also deduce from Lemma 19 that $\forall f$ a valid carefree strategy for $PDel$, $Nexts_{f_{cfDom}} \subseteq Nexts_f$ and thus $f_{cfDom} \subseteq f$. Therefore, $\forall q \in Q : q \in f_{cfDom} \implies q \in f$. This gives us $runs_{f_{cfDom}}(PDel) \subseteq runs_f(PDel)$, and we conclude $PHO_{f_{cfDom}}(PDel) \subseteq PHO_f(PDel)$. Therefore, f_{cfDom} dominates all valid carefree strategies for $PDel$. ◀

In the case where $PDel$ allows the delivery of all messages, that is contains the total collection, there is only one carefree strategy which dominates all carefree strategies.

▶ **Theorem 22** (With Total Collection, Unique Dominating Carefree Strategy). *Let $PDel$ be a Delivered predicate containing the total collection. Let f_{cfDom} be the carefree strategy with*

$Nexts_f = \{CDel(r,j) \mid CDel \in PDel \wedge r > 0 \wedge j \in \Pi\}$. f_{cfDom} is the unique carefree strategy which dominates all carefree strategies for PDel.

Proof. First, f_{cfDom} dominates all carefree strategies for $PDel$ by Theorem 21. To show uniqueness, let f be a valid carefree strategy different from f_{cfDom}; thus $Nexts_f \neq Nexts_{f_{cfDom}}$. By Lemma 19, we also have $Nexts_{f_{cfDom}} \subseteq Nexts_f$. Therefore, $Nexts_{f_{cfDom}} \subsetneq Nexts_f$.

Let $D \in Nexts_f \setminus Nexts_{f_{cfDom}}$ and let $cho \in (\mathbb{N}^* \times \Pi) \mapsto \mathcal{P}(\Pi)$ the Heard-Of collection such that $\forall r > 0, \forall j \in \Pi : cho(r,j) = D$. By application of Theorem 20, this is a Heard-Of collection generated by f but not by f_{cfDom}. Therefore, $PHO_f(PDel) \not\subset PHO_{f_{cfDom}}(PDel)$, and f' does not dominate f_{cfDom}. ◀

4.2 When Carefree is Enough

Finally, the value of carefree strategy depends on which Delivered predicates have such a dominating strategy. We already know that $PDel^F$ does; we now extend this result to a class of Delivered predicates called Round-symmetric. This condition captures the fact that given any Delivered set D, one can build, for any $r > 0$, a Delivered collection where processes receive all messages up to round r, and then they share D as their Delivered set in round r. As a limit case, the predicate also contains the total collection.

▶ **Definition 23** (Round-Symmetric Delivered Predicate)**.** Let $PDel$ be a Delivered Predicate. $PDel$ is **round-symmetric** $\triangleq$

- **(Total collection)** $PDel$ contains the total collection: $CDel_{total} \in PDel$ where $CDel_{total}$ is defined by $\forall r > 0, \forall j \in \Pi : CDel_{total}(r,j) = \Pi$.
- **(Symmetry up to a round)** $\forall D \in \{CDel(r,j) \mid CDel \in PDel \wedge r > 0 \wedge j \in \Pi\}$, $\forall r > 0, \exists CDel \in PDel, \forall j \in \Pi : (\forall r' < r : CDel(r',j) = \Pi \wedge CDel(r,j) = D)$

What round-symmetry captures is what makes $PDel^F$ be dominated by a carefree strategy: the inherent symmetry of these Delivered collections allows us to block processes with exactly the same received messages. This allows us to show that any valid strategy should allow changing round at this point, which is fundamental to any proof of domination.

▶ **Theorem 24** (Sufficient Condition of Carefree Domination)**.** *Let PDel be a Round-symmetric Delivered predicate. Then, there is a carefree strategy which dominates PDel.*

Proof. Let f be a carefree strategy dominating all carefree strategies for $PDel$ – it exists by Theorem 21 – and let f' be a valid strategy for $PDel$. We now prove that $f' \prec_{PDel^F} f$, that is $PHO_f(PDel) \subseteq PHO_{f'}(PDel)$. Let also $cho \in PHO_f(PDel)$ and t be a standard run of cho.

By Lemma 9, t is a run; we now prove by contradiction it is also a run of f'. **Assume** it is not: because t is infinite, the problem is either the infinite fairness of *next* or a *next* done when f' does not allow it. Each *next* transition being played an infinite number of times in a standard run, the only possibility left is the second one: some *next* transition in t is done while f' does not allow the corresponding process to change round. There thus exists a smallest r such that some process j is not allowed by f' to change round when $next_j$ is played at round r in t. Because $PDel$ contains the total collection, there is only one carefree strategy dominating all carefree strategies for $PDel$ by Theorem 22. This is the strategy from Theorem 21. Thus, the application of Theorem 20 yields that $cho(r,j) \in Nexts_f = \{CDel(r,j) \mid CDel \in PDel \wedge r > 0 \wedge j \in \Pi\}$.

Next, the round-symmetry of $PDel$ allows us to build $CDel_{block} \in PDel$ such that $\forall r' < r, \forall k \in \Pi : CDel_{block}(r',k) = \Pi$ and $\forall k \in \Pi : CDel_{block}(r,k) = cho(r,j)$.

Finally, we build t_{block}, an earliest run of f' for $CDel_{block}$. By Lemma 16b, we know t_{block} is a run of f'. We then have two possibilities.

- During one of the first $r-1$ iterations of t_{block}, there is some process which cannot change round. Let r' be the smallest iteration where it happens, and k be a process unable to change round at the r'-ith iteration.
  By minimality of r', all processes arrive at round r', and by symmetry of $CDel_{block}$ they all receive the same messages as k. Thus, all processes are blocked at round r', there are no more next or deliveries, and t_{block} is therefore invalid.
- For the first $r-1$ iterations, all processes change round. Thus, every one arrives at round r. By definition of an earliest run, all messages from the round are delivered before any *next*. The symmetry of $CDel_{block}$ also ensures that every process received the same messages, that is all messages from round $< r$ and all messages from $cho(r, j)$. These are exactly the messages received by j in t at round r. But by hypothesis, j is blocked in this state in t. We thus deduce that all processes are blocked at round r in t_{block}, and thus that it is an invalid run.

Either way, we deduce that f' is invalid, which is a **contradiction**. ◀

As another example of a Delivered predicate satisfying this condition, we study the model where at most B broadcasts per round can fail: either all processes receive the message sent by a process at a round or none does; there are at most B processes per round that can be in the latter case, where no one receives their message. The Delivered predicate states that each process receives the kernel of the round, and this kernel contains at least $n - B$ processes.

▶ **Definition 25** ($PDel^B$)**.** The Delivered predicate $PDel^B$ corresponding to at most B full broadcast failures is:
$\{CDel \in (\mathbb{N}^* \times \Pi) \mapsto \mathcal{P}(\Pi) \mid \forall r > 0, \forall j \in \Pi : CDel(r, j) = K_{CDel}(r) \wedge |K_{CDel}(r)| \geq n - B\}$.

The astute reader might have noticed that the carefree strategy dominating all carefree for this Delivered predicate is f_{n-B} (which is f_{n-F} with F instantiated to B).

▶ **Lemma 26** (f_{n-B} Carefree Dominates $PDel^B$)**.** *f_{n-B} dominates carefree strategies for $PDel^B$.*

Proof. The definition of $PDel^B$ gives us $\{CDel(r, j) \mid CDel \in PDel \wedge r > 0 \wedge j \in \Pi\} = \{ S \in \mathcal{P}(\Pi) \mid |S| \geq n - B\} = Nexts_{f_{n-B}}$, because the only constrained on the Delivered sets are that they must be the same for all processes at a round and that they must have at least $n - B$ processes. We therefore conclude from Lemma 19 that f_{n-B} is valid for $PDel^B$ and from Theorem 21 that it dominates carefree strategy for the same predicate. ◀

▶ **Theorem 27** (f_{n-B} Dominates $PDel^B$)**.** *f_{n-B} dominates $PDel^B$.*

Proof. We only need to prove that $PDel^B$ is round-symmetric; the application of Theorem 24 will then yield that $PDel^B$ is dominated by a carefree strategy, thus dominated by the carefree strategy that dominates all carefree strategies.

- $PDel^B$ contains the total collection, because it is the collection where every round kernel is Π.
- Let $D \in \{CDel(r, j) \mid CDel \in PDel^B \wedge r > 0 \wedge j \in \Pi\}$. By definition of $PDel^B$, D contains at least $n - B$ processes. Let $r > 0$.
  We then build $CDel : (\mathbb{N}^* \times \Pi) \mapsto \mathcal{P}(\Pi)$ such that $\forall j \in \Pi : \begin{pmatrix} \forall r' < r : CDel(r', j) = \Pi \\ \wedge\ \forall r' \geq r : CDel(r, j) = D \end{pmatrix}$.
  At each round, the Delivered sets contains at least $n - B$ processes because D does, and

all processes share the same Delivered set. We conclude that $CDel \in PDel^B$, and thus that the symmetry up to a round is satisfied.

◀

▶ **Theorem 28** (Heard-Of Characterization of $PDel^B$). *The Heard-Of predicate implemented by f_{n-B} in $PDel^B$ is $CHO_{f_{n-B}}(PDel^B) = \{cho \in (\mathbb{N}^* \times \Pi) \mapsto \mathcal{P}(\Pi) \mid \forall r > 0, \forall j \in \Pi : |cho(r,j)| \geq n - B\}$.*

Proof. We know that $Nexts_{f_{n-B}} = \{cfree(q) \mid q \in Q \wedge |\{k \in \Pi \mid \langle q.round, k\rangle \in q.received\}| \geq n - B\} = \{S \in \mathcal{P}(\Pi) \mid |S| \geq n - B\}$. We conclude by application of Theorem 20. ◀

5 Beyond Carefree Strategies: Reactionary Strategies

Sometimes, carefree strategies are not enough to capture the subtleties of a Delivered predicate. Take the one corresponding to at most F initial crashes for example: to make the most of this predicate, a strategy should remember from which processes it received a message, since it knows this process did not crash. A class of strategies which allows this is the class of reactionary strategies: they depend on messages from current and past rounds, as well as the round number. The only part of the local state these strategies cannot take into account is the set of messages received from "future" rounds, a possibility due to asynchrony.

5.1 Definition and Expressiveness Results

▶ **Definition 29** (Reactionary Strategy). Let f be a strategy, and $\forall q \in Q$, let $reac(q) \triangleq \langle q.round, \{\langle r,k\rangle \in q.received \mid r \leq q.round\}\rangle$. f is a **reactionary strategy** $\triangleq \forall q, q' \in Q : reac(q) = reac(q') \implies (q \in f \iff q' \in f)$. We write $Nexts^R_f \triangleq \{reac(q) \mid q \in f\}$ for the set of reactionary states in f. This uniquely defines f.

Even if reactionary strategies are more complex, we can still prove the same kind of results as for carefree ones, namely about validity and the existence of a reactionary strategy dominating all reactionary strategies.

▶ **Lemma 30** (Validity of Reactionary). *Let $PDel$ be a Delivered predicate and f a reactionary strategy. Then, f is valid for $PDel \iff \forall CDel \in PDel,\ \forall r > 0, \forall j \in \Pi : \langle r, \{\langle r', k\rangle \mid r' \leq r \wedge k \in CDel(r', j)\}\rangle \in Nexts^R_f$.*

Proof. ($\Rightarrow$) Let f be valid for $PDel$. We show by contradiction that it satisfies the right-hand side of the above equivalence. **Assume** there is $CDel \in PDel, r > 0$ and $j \in \Pi$ such that $\langle r, \{\langle r', k\rangle \mid r' \leq r \wedge k \in CDel(r', j)\}\rangle \notin Nexts^R_f$. Let t be an earliest run of f for $CDel$. This is a run of f by Lemma 16b.

The sought contradiction is reached by proving that t is invalid. To do so, we split according to two cases.

- During one of the first $r-1$ iterations of the t, there is some process which cannot change round. Let r' be the smallest iteration where it happens, and k be a process unable to change round at the r'-ith iteration.
  By minimality of r', all processes arrive at round r' in t; by definition of an earliest run, all messages for k from all rounds up to r' are delivered before the *next* for the iteration. Let q the local state of k at the first *next* in the r'-ith iteration, and let q' be any local state of k afterward. The above tells us that as long as $q'.round = q.round$, we have $reac(q) = reac(q')$ and thus $q' \notin f$. Therefore, k can never change round while at round r'.
  We conclude that t is invalid.

- For the first $r-1$ iterations, all processes change round. Thus, every one arrives at round r in the $r-1$-ith iteration. By definition of an earliest run, all messages from rounds up to r are delivered before any *next* at round r. By hypothesis, j cannot change round because its reactionary state – the combination of its rounds and the messages it received from past and current rounds – is not in $Nexts^R_f$. Let q be the local state of j at the first *next* in the r-ith iteration, and let q' be any local state of j afterward. The above tells us that as long as $q'.round = q.round$, we have $reac(q) = reac(q')$ and thus $q' \notin f$. Therefore, j can never change round while at round r.
  Here too, t is invalid.

Either way, we reach a **contradiction** with the validity of f.

($\Leftarrow$) Let $PDel$ and f such that $\forall CDel \in PDel, \langle r, \{\langle r', k\rangle \mid r' \leq r \wedge k \in CDel(r', j)\}\rangle \in Nexts^R_f$. We show by contradiction that f is valid.

Assume the contrary: there is some $t \in runs_f(PDel)$ which is invalid. Thus, there are some process blocked at a round forever in t. Let r be the smallest such round, and j be a process blocked at round r in t. By minimality of r, all processes arrive at round r. By definition of a run of $PDel$, there is a $CDel \in PDel$ such that t is a run of $CDel$. Thus, eventually all messages from all Delivered sets of j up to round r are delivered.

From this point on, f allows j to change round by definition, and the fairness of *next* imposes that j does at some point. We conclude that j is not blocked at round r in t, which **contradicts** the hypothesis. ◀

There is also always a reactionary strategy dominating all reactionary strategies for a given Delivered predicate. Analogously to the carefree case, this stems from the fact that the strategy which only waits for the prefixes of Delivered collections in the predicate waits for more messages than any other valid reactionary strategy, by Lemma 30.

▶ **Theorem 31** (Always a Dominating Reactionary Strategy)**.** *Let $PDel$ a Delivered predicate and let f_{rcDom} be the reactionary strategy defined by*
$Nexts^R_{rcDom} = \{\langle r, \{\langle r', k\rangle \mid r' \leq r \wedge k \in CDel(r', j)\}\ \rangle \mid r > 0 \wedge CDel \in PDel\}$.
f_{rcDom} dominates all reactionary strategies for $PDel$.

Proof. First, f_{rcDom} is valid for $PDel$ by application of Lemma 30.

As for domination, we also deduce from Lemma 30 that $\forall f$ a valid reactionary strategy for $PDel$, $Nexts^R_{f_{rcDom}} \subseteq Nexts^R_f$ and thus $f_{rcDom} \subseteq f$. Therefore, $\forall q \in Q : q \in f_{rcDom} \implies q \in f$. This gives us $runs_{f_{rcDom}}(PDel) \subseteq runs_f(PDel)$, and we conclude $PHO_{f_{rcDom}}(PDel) \subseteq PHO_f(PDel)$. We conclude that f_{rcDom} dominates all valid carefree strategies for $PDel$. ◀

5.2 Example Dominated by Reactionary Strategy

To show the usefulness of reactionary strategies, we study the Delivered predicate corresponding to reliable communication and at most F initial crashes: either processes crash initially and no message of theirs is ever delivered, or they do not and all their messages will be delivered eventually.

▶ **Definition 32** ($PDel^F_{ini}$)**.** The Delivered predicate $PDel^F_{ini}$ for at most F initial crashes is:
$\{CDel \in (\mathbb{N}^* \times \Pi) \mapsto \mathcal{P}(\Pi) \mid \exists \Sigma \subseteq \Pi : |\Sigma| \geq n - F \wedge \forall r > 0, \forall j \in \Pi : CDel(r, j) = \Sigma\}$.

As we mentioned above, it is possible to take advantage of the past by waiting in the current round for messages from processes which sent a message in a past round.

▶ **Definition 33** (Past complete strategy)**.** The **past-complete strategy** f_{pc} is defined by $Nexts^R_{f_{pc}} = \{\langle r, [1,r] \times \Sigma\rangle \mid r > 0 \wedge \Sigma \subseteq \Pi \wedge |\Sigma| \geq n - F\}$.

▶ **Lemma 34** (f_{pc} Reactionary Dominates $PDel^F_{ini}$)**.** *f_{pc} dominates all reactionary strategies for $PDel^F_{ini}$.*

Proof. It follows from Theorem 31, because $Nexts^R_{f_{pc}} = \{\langle r, \{\ \langle r', k\rangle \mid r' \leq r \wedge k \in \Sigma\}\rangle \mid r > 0 \wedge \Sigma \subseteq \Pi \wedge |\Sigma| \geq n - F\} = \{\langle r, \{\langle r', k\rangle \mid r' \leq r \wedge k \in CDel(r', j)\}\rangle \mid r > 0 \wedge CDel \in PDel^F_{ini}\}$. The last equality follows from the fact that Delivered sets are always the same for all Delivered collection in $PDel^F_{ini}$. ◀

Reactionary strategies can generate more complex and involved Heard-Of predicates than carefree ones. The one generated by f_{pc} for $PDel^F_{ini}$ is a good example: it ensures that all Heard-Of sets contains at least $n - F$ processes, and it also forces Heard-Of sets for a process to be non-decreasing, and for all rounds to eventually converge to the same Heard-Of set. This follows from the fact that a process can detect an absence of crash: if one message is received from a process, it will always be safe to wait for messages from this process as it did not crash and never will. Since there is no loss of message for non crashed processes, every one eventually receives a message from every process sending messages, and thus the Heard-Of sets converge.

▶ **Theorem 35** (Heard-Of Characterization of $PDel^F_{ini}$)**.** $PHO_{f_{pc}}(PDel^F_{ini}) =$

$$\left\{ cho \in (\mathbb{N}^* \times \Pi) \mapsto \mathcal{P}(\Pi) \;\middle|\; \left(\begin{array}{rl} & \forall r > 0, \forall j \in \Pi : \left(\begin{array}{rl} & |cho(r,j)| \geq n - F \\ \wedge & cho(r,j) \subseteq cho(r+1,j) \end{array} \right) \\ \wedge & \exists \Sigma_0 \subseteq \Pi, \exists r_0 > 0, \forall r \geq r_0, \forall j \in \Pi : cho(r,j) = \Sigma_0 \end{array} \right) \right\}$$

Proof. First, we show $\subseteq$. Let $cho \in PHO_{f_{pc}}(PDel^F_{ini})$ and $t \in runs_{f_{pc}}(PDel^F_{ini})$ a run of f_{pc} implementing *cho*. Then, by definition of f_{pc}, processes change round only when they have received at least $n - F$ messages from the current round and they have completed their past, which implies that $\forall r \in \mathbb{N}^*, \forall j \in \Pi : \left(\begin{array}{rl} & |cho(r,j)| \geq n - F \\ \wedge & cho(r,j) \subseteq cho(r+1,j) \end{array} \right)$.

As for the last part of the Heard-Of predicate, notice that by definition of runs, there is a point in t where all messages from round 1 have been delivered. From this point on, all processes wait for all messages of processes which send them. Thus, there is a round – the maximal round of a process when the last message from round 1 is delivered in t – from which the rounds are space-time uniform.

Then, we show $\supseteq$. Let *cho* a Heard-Of collection over Π in the set on the right. We now build t, a special run of *cho* in iterations: at each iteration, we deliver messages from Heard-Of sets and messages to complete the past of these sets then change round for all processes. That is, we deliver all messages from past rounds sent by the processes in the Heard-Of sets which had not yet been delivered. The initial state is also the usual one.

It is a run because it satisfies all four constraints:

- t has the right initial state by definition.
- We only gave the transitions and initial state, so the states satisfy the transition function.
- Every message is delivered in a round greater or equal to the one it was sent in.
- Messages are delivered only once.

It is also a run of f_{pc} because each Heard-Of set contains at least $n - F$ processes by hypothesis and we complete the past of each process before playing any *next*. We still have to show that it is a run of a Delivered collection in $PDel^F_{ini}$.

By hypothesis, there is a round r_0 from which the Heard-Of collection is space-time uniform – all processes share the same Heard-Of set forever. Let Σ_0 this Heard-Of set. We

then build $CDel$ such that $\forall r > 0, \forall j \in \Pi : CDel(r, j) = \Sigma_0$. Because $\forall r > 0, \forall j \in \Pi : cho(r+1, j) \supseteq cho(r, j)$, and because we only ever deliver messages from processes in some Heard-Of sets, we know every message delivered in t is in $CDel$. On the other hand, all messages from $CDel$ are delivered eventually:

- If a message is in a Heard-Of set, it is delivered in the corresponding round.
- If it is not in a Heard-Of set, it is delivered at most in the first round after its sending round where the Heard-Of set contains this process. Such a round must exist by hypothesis, because only processes from Σ_0 get their messages delivered, and eventually all processes have Σ_0 as their Heard-Of set.

We conclude that $t \in runs_{f_{pc}}(PDel^F_{ini})$, and thus that $cho \in PHO_{f_{pc}}(PDel^F_{ini})$. ◀

As for f_{pc} dominating $PDel^F_{ini}$, the argument is quite similar to the one for f_{n-F} dominating $PDel^F$, with a more subtle manipulation of Delivered collection because we need to take the past into account.

▶ **Theorem 36** (f_{pc} Dominates $PDel^F_{ini}$)**.** *f_{pc} dominates $PDel^F_{ini}$.*

Proof. Let f be a valid strategy for $PDel^F_{ini}$; we now prove that $f \prec_{PDel^F_{ini}} f_{pc}$, that is $PHO_{f_{pc}}(PDel^F_{ini}) \subseteq PHO_f(PDel^F_{ini})$. Thus, let $cho \in PHO_{f_{pc}}(PDel^F_{ini})$. We now build t, a special run of cho: at each round, we deliver messages from Heard-Of sets and messages to complete the past of these sets, then all processes change round. That is, we deliver all messages from past rounds sent by processes in the current Heard-Of sets which had not yet been delivered. The initial state is also the usual one.

It is a run because it satisfies all four constraints:

- t has the right initial state by definition.
- We only gave the transitions and initial state, so the states satisfy the transition function.
- Every message is delivered in a round greater or equal to the one it was sent in.
- Messages are delivered only once.

We want to show that t is a run of f; we do so by contradiction. **Assume** it is not: because t is infinite, the problem is either the infinite fairness of *next* or a *next* done when f does not allow it. Each *next* transition being played an infinite number of times, the only possibility left is the second one: some *next* transition in t is done while f does not allow the corresponding process to change round. There thus exists a smallest r such that some process j is not allowed by f to change round when $next_j$ is played at round r in t. At this point, we know that the past of j is complete and that it received all messages from $cho(r, j)$. By Theorem 35, we know that $\forall r' < r : cho(r', j) \subseteq cho(r, j)$. Thus, when we complete the past of j, we deliver all messages at each round $\leq r$ from the processes in $cho(r, j)$.

We now build a Delivered collection $CDel_{block}$ such that $\forall r' > 0, \forall k \in \Pi : CDel_{block}(r', k) = cho(r, j)$. This is in $PDel^F_{ini}$ because $cho(r, j)$ contains at least $n - F$ processes by Theorem 35. Finally, we build t_{block}, an earliest run of f for $CDel_{block}$. By Lemma 16b, we know t_{block} is a run of f. We then have two possibilities.

- During one of the first $r - 1$ iterations of t_{block}, there is some process which cannot change round. Let r' be the smallest iteration where it happens, and k be a process unable to change round at the r'-ith iteration.

  By minimality of r', all processes arrive at round r', and by symmetry of $CDel_{block}$ they all receive the same messages as k. Thus, all processes are blocked at round r', there are no more next or deliveries, and t_{block} is therefore invalid.

- For the first $r-1$ iterations, all processes change round. Thus, every one arrives at round r at the r-ith iteration. By definition of an earliest run, all messages from the round are delivered before any *next*. The symmetry of $CDel_{block}$ also ensures that every process received the same messages, that is all messages from round $< r$ and all messages from $cho(r, j)$. These are exactly the messages received by j in t at round r. But by hypothesis, j is blocked in this state in t. We thus deduce that all processes are blocked at round r in t_{block}, and thus that it is an invalid run.

  Either way, we deduce that f is invalid, which is a **contradiction**. ◀

5.3 When The Future Serves

In the above, we considered cases where the dominating strategy is at most reactionary: only the past and present rounds are useful for generating Heard-Of collections. But messages from future rounds serve in some cases. We give an example, presenting only the intuition.

▶ **Definition 37** ($PDel^1_{lost}$)**.** The Delivered predicate $PDel^1_{lost}$ corresponding to at most 1 message lost is: $\{CDel \in (\mathbb{N}^* \times \Pi) \mapsto \mathcal{P}(\Pi) \mid \sum_{r>0, j\in\Pi} |\Pi \setminus CDel(r, j)| \leq 1\}$.

Application of our results on carefree strategy shows that the carefree strategies dominating all carefrees for this predicate is f_{n-1}. Similarly, looking at the past only allows processes to wait for $n-1$ messages because one can always deliver all messages from the past, and then the loss might be a message from the current round. If we look at the messages from the next round, on the other hand, we can ensure that at each round, at most one message among all processes is not delivered on time.

▶ **Definition 38** (Asymmetric Strategy)**.** Let $after : Q \mapsto \mathcal{P}(\Pi)$ such that $\forall q \in Q : after(q) = \{k \in \Pi \mid \langle q.round + 1, k\rangle \in q.received\}$, and $cfree$ as in Definition 18.

Define $f_{asym} \triangleq \left\{ q \in Q \;\middle|\; \begin{array}{ll} & cfree(q) = \Pi \\ \vee & (|after(q)| = n-1 \wedge |cfree(q)| = n-1\} \end{array} \right\}$.

Intuitively, this strategy is valid because at each round and for each process, only two cases exist: either no message for this process at this round is lost, and it receives a message from each process; or one message for this process is lost at this round, and it only receives $n-1$ messages. But all other processes receive n messages, thus change round and send their message from the next round. Since the one loss already happened, all these messages are delivered, and the original process eventually receives $n-1$ messages from the next round.

This strategy also ensures that at most one process per round receives only $n-1$ messages on time – the others must receive all messages. This vindicates the value of messages from future rounds for some Delivered predicates, such as the ones with asymmetry in them.

6 Related Works

Rounds Everywhere Rounds in message-passing algorithms date at least back to their use by Arjomandi et al. [1] as a synchronous abstraction of time complexity. Since then, they are omnipresent in the literature. First, the number of rounds taken by a distributed computation is a measure of its complexity. Such round complexity was even developed into a full-fledged analogous of classical complexity theory by Fraigniaud et al. [7]. Rounds also serve as stable intervals in the dynamic network model championed by Kuhn and Osham [11]: each round corresponds to a fixed communication graph, the dynamicity following from

possible changes in the graph from round to round. Finally, many fault-tolerant algorithms are structured in rounds, both synchronous [6] and asynchronous ones [3].

Although we only study message-passing models in this article, one cannot make justice to the place of rounds in distributed computing without mentioning its even more domineering place in shared-memory models. A classic example is the structure of executions underlying the algebraic topology approach pioneered by Herlihy and Shavit [9], Saks and Zaharoglou [13], and Borowsky and Gafni [2].

Abstracting the Round Gafni [8] was the first to attempt the unification of all versions of rounds. He introduced the Round-by-Round Fault Detector abstraction, a distributed module analogous to a failure detector which outputs a set of suspected processes. In a system using RRFD, the end condition of rounds is the reception of a message from every process not suspected by the local RRFD module; communication properties are then defined as predicates on the output of RRFDs. Unfortunately, this approach does not suit our needs: RRFDs do not ensure termination of rounds, while we require it.

Next, Charron-Bost and Schiper [4] took a dual approach to Gafni's with the Heard-Of Model. Instead of specifying communication by predicates on a set of suspected processes, they used Heard-Of predicates: predicates on a collection of Heard-Of sets, one for each round r and each process j, containing every process from which j received the message sent in round r before the end of this same round. This conceptual shift brings two advantages: a purely abstract characterization of message-passing models and the assumption of infinitely many rounds, thus of round termination.

But determining which model implements a given Heard-Of predicate is an open question. As mentioned in Marić [12], the only known works addressing it, one by Hutle and Schiper [10] and the other by Drăgoi et al. [5], both limit themselves to very specific predicates and partially synchronous system models.

7 Conclusion and Perspectives

We propose a formalization for characterizing Heard-Of predicate of an asynchronous message-passing model through Delivered predicates and strategies. We also show its relevance, expressivity and power: it allows us to prove the characterizations from Charron-Bost and Schiper [4], to show the existence of characterizing predicates for large classes of strategies as well as the form of these predicates. Yet there are two aspects of this research left to discuss: applications and perspectives.

First, what can we do with this characterizing Heard-Of predicate? As mentioned above, it gives the algorithm designer a concise logical formulation of the properties on rounds a given model can generate. Therefore, it allows the design of algorithms at a higher level of abstraction, implementable on any model which can generate the corresponding Heard-Of predicate. The characterizing predicate is also crucial to verification: it bridges the gap between the intuitive operational model and its formal counterpart. To verify a round-based algorithm for a given message-passing model, one needs only to check if its correctness for the characterizing predicate. If it is the case, we have a correct implementation by combining the algorithm with a dominating strategy; if it is not, then the algorithm will be incorrect for all predicates generated by the model.

Finally, it would be beneficial to prove more results about the existence of a dominating strategy, as well as more conditions for the dominating strategy to be in a given class. There is also space for exploring different Delivered predicates. For example, some of our results

suppose that the predicate contains the total Delivered collection; what can we do without this assumption? Removing it means that faults are certain to occur, which is rarely assumed. Nonetheless, it might be interesting to study this case, both as a way to strengthen our results, and because forcing failures might be relevant when modelling highly unreliable environments such as the cloud or natural settings. Another viable direction would be to add oracles to the processes, giving them additional information about the Delivered collection, and see which Heard-Of predicates can be generated. These oracles might for example capture the intuition behind failure detectors.

References

1 Eshrat Arjomandi, Michael J. Fischer, and Nancy A. Lynch. A difference in efficiency between synchronous and asynchronous systems. In *Thirteenth Annual ACM Symposium on Theory of Computing*, STOC '81, pages 128–132, 1981. doi:10.1145/800076.802466.

2 Elizabeth Borowsky and Eli Gafni. Generalized FLP impossibility result for T-resilient asynchronous computations. In *Twenty-fifth Annual ACM Symposium on Theory of Computing*, STOC '93, pages 91–100. ACM, 1993. doi:10.1145/167088.167119.

3 Tushar Deepak Chandra, Vassos Hadzilacos, and Sam Toueg. The weakest failure detector for solving consensus. *J. ACM*, 43(4):685–722, July 1996. doi:10.1145/234533.234549.

4 Bernadette Charron-Bost and André Schiper. The heard-of model: computing in distributed systems with benign faults. *Distributed Computing*, 22(1):49–71, April 2009. doi:10.1007/s00446-009-0084-6.

5 Cezara Drăgoi, Thomas A. Henzinger, and Damien Zufferey. Psync: A partially synchronous language for fault-tolerant distributed algorithms. In *43rd Symposium on Principles of Programming Languages*, pages 400–415, 2016. doi:10.1145/2837614.2837650.

6 Michael J. Fischer and Nancy A. Lynch. A lower bound for the time to assure interactive consistency. *Information Processing Letters*, 14(4):183–186, 1982. doi:10.1016/0020-0190(82)90033-3.

7 Pierre Fraigniaud, Amos Korman, and David Peleg. Towards a complexity theory for local distributed computing. *J. ACM*, 60(5):35:1–35:26, October 2013. doi:10.1145/2499228.

8 Eli Gafni. Round-by-round fault detectors (extended abstract): Unifying synchrony and asynchrony. In *Seventeenth ACM Symposium on Principles of Distributed Computing*, PODC '98, pages 143–152. ACM, 1998. doi:10.1145/277697.277724.

9 Maurice Herlihy and Nir Shavit. The topological structure of asynchronous computability. *J. ACM*, 46(6):858–923, November 1999. doi:10.1145/331524.331529.

10 M. Hutle and A. Schiper. Communication predicates: A high-level abstraction for coping with transient and dynamic faults. In *37th International Conference on Dependable Systems and Networks (DSN'07)*, pages 92–101, June 2007. doi:10.1109/DSN.2007.25.

11 Fabian Kuhn and Rotem Oshman. Dynamic networks: Models and algorithms. *SIGACT News*, 42(1):82–96, March 2011. doi:10.1145/1959045.1959064.

12 Ognjen Marić, Christoph Sprenger, and David Basin. Cutoff bounds for consensus algorithms. In Rupak Majumdar and Viktor Kunčak, editors, *Computer Aided Verification*, pages 217–237. Springer International Publishing, 2017.

13 Michael Saks and Fotios Zaharoglou. Wait-free k-set agreement is impossible: The topology of public knowledge. *SIAM J. Comput.*, 29(5):1449–1483, March 2000. doi:10.1137/S0097539796307698.