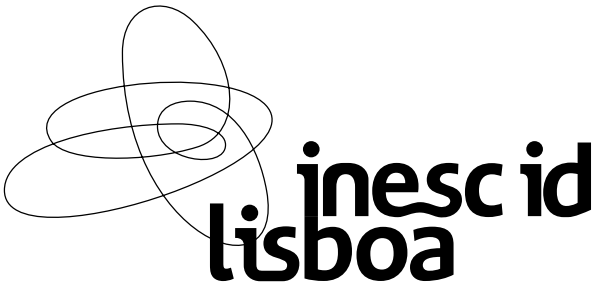
INESC-ID Technical Report 10/2007

The Availability and Performance of Epidemic Quorum Algorithms

João Barreto and Paulo Ferreira
INESC-ID/IST
Rua Alves Redol N°9, 1000-029 Lisboa, Portugal
{joao.barreto,paulo.ferreira}@inesc-id.pt

February 2007



Abstract

Epidemic quorum systems enable highly available agreement even when a quorum is not simultaneously connected, making them suitable for weakly connected environments. Although recent work has proposed epidemic quorum algorithms, their availability and performance trade-offs are not well studied.

This paper formally defines generic epidemic quorum systems. The formalism unifies proposed epidemic quorum systems and is a framework for devising and studying epidemic coteries. We prove the safety of the resulting systems and analytically characterize their availability and performance. In particular, we identify previously undocumented trade-offs between both aspects that do not exist in the classical counterpart. Furthermore, we present analytical results comparing the availability and performance of relevant epidemic quorum systems, relating them to classical quorum systems.

1 Introduction

Quorum systems are a basic tool for reliable agreement in distributed systems [PW95]. Their applicability is wide, ranging from data replication protocols, distributed mutual exclusion, name servers, selective dissemination of data, and distributed access control and signatures [AW96].

Classical quorum systems require agreement on a value to be accepted by a *quorum* of live processes that are simultaneously connected in the same network partition. This is not adequate in weakly connected networks, e.g. mobile or sensor networks, where connected quorums are improbable.

Epidemic quorum systems eliminate such shortcoming, allowing unconnected quorums. An epidemic quorum algorithm tries to ensure agreement by running a finite number of *elections*. Intuitively, on each election, each process may vote for one proposed value. By epidemic propagation of votes, eventually each process should be able to determine, from its local state, whether the system has agreed on a given value w, or the current election is inconclusive and, hence, a new one starts.

Recent work [Kel99, HSAA03] has proposed epidemic quorum algorithms. However, to the best of our knowledge, neither are epidemic quorum systems well defined, nor are their availability or performance well studied. In addition, the extensive studies on classical quorum systems are not valid for epidemic quorum systems, as their failure conditions and quorum definitions are radically distinct.

This paper studies epidemic quorum systems and algorithms, introducing novel results that help understand such an approach. Our contribution is twofold:

1. We formally define epidemic coteries and a generic epidemic quorum algorithm, and prove its safety. The formalism unifies proposed epidemic quorum systems and is a framework for devising and studying epidemic coteries.

2. Based on the formalism, we formally characterize the liveness of epidemic quorum systems. Such characterization enables a novel, analytical comparison of the availability and performance of epidemic quorum systems. Namely, we study two relevant epidemic quorum constructions, majority and linear plurality. Further, we present conditions where epidemic quorum systems decide faster than classical ones, and identify trade-offs that do not exist in the classic approach.

2 Epidemic Quorum Systems

Consider a set of distributed processes, U, where $|U| = y$. It runs in an asynchronous system, without any access to a global state or global clock. Processes may fail permanently; we assume non-byzantine failures. Transient network partitions may also occur, restricting communication to processes inside the same partition.

The distributed processes wish to agree on a single value, taken from a set of values, Val, proposed during the agreement process; $z = |Val|$ denotes proposal contention. We address such a problem using epidemic quorum algorithms. Similarly to classical quorum algorithms [PW95], epidemic quorum algorithms ensure agreement on a single value by having coteries of processes vote for the proposed values. However, the epidemic and classical notions of coteries are distinct, as well as the corresponding algorithms.

This section formally defines epidemic quorum systems and algorithms, presenting some fundamental properties. In Section 2.1 we describe epidemic quorum systems. We then address epidemic quorum algorithms in Section 2.3 and their liveness properties in Section 3.

We have proven all the results presented in the paper. When absent in the paper, the proof of each theorem/lemma/proposition may be found in the complementary appendix.

2.1 Epidemic Coteries

We start with some definitions and lemmas that will allow us to construct and reason about epidemic quorum systems in the remainder of the paper.

Definition 1. *Vote set, vote configuration and q-vote configuration*

A vote set s is a set of processes, $s \subseteq U$.

A vote configuration c is a set of non-empty vote sets, $c = \{s_1, s_2, .., s_n\}$, such that $\forall s_i, s_j \in c, s_i \cap s_j = \emptyset$.

A q-vote configuration qc is a pair, $qc = \langle Q_{qc}, \{A^1_{qc}, A^2_{qc}, .., A^{n_{qc}}_{qc}\}\rangle$, where $\{Q_c, A^1_{qc}, .., A^{n_{qc}}_{qc}\}$ is a vote configuration.

A value-vote configuration vc is a set of pairs, $vc = \{\langle\, a^1, v^{a^1}_{vc}\rangle, .., \langle\, a^{n_{vc}}, v^{a^{n_{vc}}}_{vc}\rangle\}$, where $\{v^1_{vc}, .., v^{n_{vc}}_{vc}\}$ is a vote configuration, each $a^1, .., a^{n_{vc}} \in Val$ and, for all i, j, $a^i \neq a^j$. We denote vote configuration $\{v^1_{vc}, .., v^{n_{vc}}_{vc}\}$ by $VCfg(vc)$.

We use vote, q-vote and value-vote configurations to reason about the votes that p is aware of in a given moment. Deciding election outcomes will require speculating about the votes that some value may potentially receive in the best case; i.e. if all processes whose vote is still ignored by p happen to vote for that value.

Definition 2. *Potential vote set*

Let s be a vote set. The potential vote set of s in a vote configuration c, denoted $[s]_c$, is defined as $[s]_c = s \cup (U \backslash (\bigcup s : s \in c)\}$. In particular, $[\emptyset]_c = (U \backslash (\bigcup s : s \in c)\}$.

Further, the potential vote set of s in a q-vote configuration qc, denoted $[s]_{qc}$, is defined as $[s]_{qc} = s \cup (U \backslash (Q_{qc} \cup A^1_{qc} \cup .. \cup A^{n_{qc}}_{qc}))$.

We are now able to define the notions of coverage and potential coverage of a q-vote configuration by another one, the building blocks for a definition of epidemic coterie. We say that a larger q-vote configuration, qc, covers a smaller one, qd, when each vote set in the latter is a subset of a distinct vote set of the former, where Q_{qc} must be the superset of Q_{qd}. Similarly, potential coverage means that a q-vote configuration may still grow (with the unknown votes) to cover another one.

Definition 3. *Coverage and potential coverage between q-vote configurations*
Let qc and qd be q-vote configurations.
We say qc covers *qd, or $qc \succ qd$, if and only if:*

1. $Q_{qc} \supseteq Q_{qd}$, *and*
2. *There exists an injective function $f : \{1, .., n_{qd}\} \rightarrow \{1, .., n_{qc}\}$ such that $\forall_{1 \leq k \leq n_{qd}} : A_{qc}^{f(k)} \supseteq A_{qd}^{k}$.*

We say qc may cover *qd, or $qc \rhd qd$, if and only if:*

1. $[Q_c]_{qc} \supseteq Q_{qd}$, *and*
2. *There exists an injective function $f : \{1, .., n_{qd}\} \rightarrow \{1, .., n_{qc}\}$ such that $\forall_{1 \leq k \leq n_{qd}} : [A_{qc}^{f(k)}]_{qc} \supseteq A_{qd}^{k}$ or $[\emptyset]_{qc} \supseteq A_{qd}^{k}$.*

Proposition 1. *Let c and d be q-vote configurations such that $d \succ c$. For any A_d^i, either there exists A_c^j such that $[A_c^j]_c \supseteq [A_d^i]_d$, or $[\emptyset]_c \supseteq [A_d^i]_d$.*

We may finally define epidemic coteries (EC).

Definition 4. *Epidemic Coterie*
Let $\mathcal{E}$ be a non-empty set of q-vote configurations. $\mathcal{E}$ is an Epidemic Coterie (EC) *if, $\forall c, d \in \mathcal{E} : c \neq d$:*

1. $\forall j : \langle A_c^j, \{Q_c\} \cup \{A_c^i : i \neq j\}\rangle \not\rhd d$, *and*
2. $\langle \emptyset, \{Q_c\} \cup \{A_c^1, .., A_c^{n_c}\}\rangle \not\rhd d$, *and*
3. $c \not\succ d$.

We denote by $QSet(\mathcal{E})$ the set of every $Q_c : c \in \mathcal{E}$, and by $AQSet(\mathcal{E})$ the set of every $A_c^j : c \in \mathcal{E}$, for all j. We call each Q_c the quorum *of c. Also, we call each A_c^j an* anti-quorum *of c.*

Intuitively, the above definition ensures that, from a given q-vote configuration in the EC, no other *conflicting* q-vote configuration (in the sense of having a quorum formed exclusively by voters not in the former's quorum) may be reached.

2.2 Examples: Majority and Linear Plurality ECs

This section defines and characterizes two example ECs, $\mathcal{E}_{Maj}$ and $\mathcal{LP}$, in a system with five processes, $U = \{p_1, p_2, p_3, p_4, p_5\}$. $\mathcal{E}_{Maj}$ and $\mathcal{LP}$ are instances of majority and linear plurality constructions, respectively; such constructions are subjacent to the systems proposed and empirically studied by Holliday et al. [HSAA03] and Keleher [Kel99].

We start by defining the ECs. $\mathcal{E}_{Maj}$ is the set of q-vote configurations such that $QSet(\mathcal{E}_{Maj})$ is the collection of all sets of 3 processes (i.e. $(y+1)/2$), and $AQSet(\mathcal{E}_{Maj})$ is the empty set.

Quorum	Anti-Quorums
p_1, p_2, p_3	
p_1, p_2, p_4	
p_1, p_2, p_5	
p_1, p_3, p_4	
p_1, p_3, p_5	
p_1, p_4, p_5	
p_2, p_3, p_4	
p_2, p_3, p_5	
p_2, p_4, p_5	
p_3, p_4, p_5	

Quorum	Anti-Quorums
p_1, p_2	$\{p_3\}, \{p_4\}$
p_1, p_2	$\{p_3\}, \{p_5\}$
p_1, p_2	$\{p_4\}, \{p_5\}$
p_1, p_3	$\{p_2\}, \{p_4\}$
p_1, p_3	$\{p_2\}, \{p_5\}$
p_1, p_3	$\{p_4\}, \{p_5\}$
p_1, p_4	$\{p_2\}, \{p_3\}$
p_1, p_4	$\{p_2\}, \{p_5\}$
p_1, p_4	$\{p_3\}, \{p_5\}$
p_1, p_5	$\{p_2\}, \{p_3\}$
p_1, p_5	$\{p_2\}, \{p_4\}$
p_1, p_5	$\{p_3\}, \{p_4\}$

Quorum	Anti-Quorums
p_2, p_3	$\{p_1\}, \{p_4, p_5\}$
p_2, p_3	$\{p_1\}, \{p_4\}, \{, p_5\}$
p_2, p_4	$\{p_1\}, \{p_3, p_5\}$
p_2, p_4	$\{p_1\}, \{p_3\}, \{, p_5\}$
p_2, p_5	$\{p_1\}, \{p_3, p_4\}$
p_2, p_5	$\{p_1\}, \{p_3\}, \{, p_4\}$
p_3, p_4	$\{p_1\}, \{p_2\}, \{, p_5\}$
p_3, p_5	$\{p_1\}, \{p_2\}, \{, p_4\}$
p_4, p_5	$\{p_1\}, \{p_2\}, \{, p_3\}$
p_1	$\{p_2\}, \{p_3\}, \{, p_4\}, \{, p_5\}$

Figure 1: Q-vote configurations comprising $\mathcal{LP}$, as obtained from Definition 5. $\mathcal{E}_{Maj}$ is a subset of $\mathcal{LP}$, and corresponds to the left-most table only.

$\mathcal{LP}$ is a larger coterie. Plurality requires that, in order to decide a value v, the number of voters for v (the quorum) must be higher than the size of the potential vote set of voters for every all other value (the anti-quorums). Linear plurality assumes a total order among processes, and, in addition to the case above, it also allows the quorum to be equally sized as some anti-quorums in the same q-vote configuration, as long as the lowest processes of the potential vote sets of the former are higher than that of the latter. More precisely:

Definition 5. *Majority and linear plurality ECs*

$$\mathcal{E}_{Maj} = \{\langle q, \emptyset\rangle : q \subseteq U \wedge \ |q| = 3\}$$

$$\mathcal{LP} = \left\{ qc : \left\{ \begin{array}{l} \forall k, |Q_{qc}| > |[A^k_{qc}]_{qc}| \ \textit{or} \ (|Q_{qc}| = |[A^k_{qc}]_{qc}| \wedge \ \exists_{i \in \ Q_{qc}} : \forall_{l \in \ [A^k_{qc}]_{qc}}, i < l) \\ \textit{and} \\ \forall qd \in \mathcal{LP}, qc \not\succ \ qd \end{array} \right. \right\}$$

Figure 1 presents the q-vote configurations that respectively comprise $\mathcal{E}_{Maj}$ and $\mathcal{LP}$. Naturally, $\mathcal{E}_{Maj}$ is a subset of $\mathcal{LP}$. As the reader may confirm, both $\mathcal{E}_{Maj}$ and $\mathcal{LP}$ are ECs.

2.3 Epidemic Quorum Algorithm

This section presents an algorithm that relies on a given EC, $\mathcal{E}$, for achieving distributed consensus, and proves its safety. We start by defining the local state the algorithm maintains at each process. Let $p \in U$ be a process. The local state of p includes:

1. A *current election identifier*, $e_p \in \ N$;
2. A value-vote configuration, $V_p = \{\langle \ a^1, v_p^{a^1}\rangle, .., \langle \ a^{n_p}, v_p^{a^{n_p}}\rangle\}$.

We call each $v_p^{a^i}$ the *local vote set* at p for value a^i. We designate the vote configuration $c_p = \{v_p^1, .., v_p^n\}$ as the *local vote configuration* at p. For

simplicity of presentation, when there does not exist any value-vote pair for some value a^x in V_p, $v_p^{a^x}$ denotes the empty set, $\emptyset$.

The state of each process p evolves as p proposes a value, receives voting information from other processes, and determines the outcome of each election. Algorithms 2, 3 and 4 describe each of such steps, respectively.

Algorithm 1 merge(V_r)

```
1: for all ⟨ a^i, v_r^{a^i} ⟩ ∈ V_r do
2:    if ∃j : a^j = a^i then
3:       v_p^j ← v_p^j ∪ v_r^i
4:    else
5:       V_p ← V_p ∪ ⟨ a^i, v_r^{a^i} ⟩
6:    end if
7: end for
```

Algorithm 2 propose(v)

```
1: if ∀i : p ∉ v_p^i then
2:    merge({⟨ v, {p}⟩})
3: else
4:    merge({⟨ v, ∅⟩})
5: end if
```

When a value v is proposed at process p (Algorithm 2), p votes for it in case it has not yet voted for another value in the current election; this means adding p to the vote set corresponding to v in V_p (line 2). For fairness, the algorithm ensures that, even if p has already voted in the current election, V_p includes an entry for v, possibly with an empty vote set (line 4); this way, processes other than p may still vote for v.

Algorithm 3 receiveVotesFrom(r)

```
1: if e_p < e_r then
2:    e_p ← e_r
3:    for all ⟨ a^i, v_p^{a^i} ⟩ ∈ V_p do
4:       v_p^j ← ∅
5:    end for
6: end if
7: if e_p = e_r then
8:    merge(V_r)
9:    if ∀i : p ∉ v_p^i then
10:      choose some j : 1 ≤ j ≤ |V_p|
11:      propose(a^p)
12:   end if
13: end if
```

When p is accessible from some process r, p may receive voting information from r, as Algorithm 3 describes. If r already holds information about a later

election, then p discards its obsolete vote information (in V_p) and sets e_p to the more recent election, e_r(lines 2 to 5).

Finally, if both processes end up in the same election (this does not happen if $e_r < e_p$), then p complements V_p with new votes learned from r (line 8). Further, in case p has not yet voted in its current election, it may cast a vote for one of the newly received values (lines 10 and 11).

Algorithm 4 checkOutcome()

```
1: if repeat-condition then
2:     V_p ← ∅
3:     e_p ← e_p + 1
4: end if
5: if decide-condition(w) then
6:     decide w
7: end if
```

Based exclusively on its local state, each process determines if an election is guaranteed to produce the same outcome at every other process. Two possible outcomes are possible: the election may either decide one of the proposed values (*decide*), or not decide any value (*repeat*). In the latter case, a new election starts; this is repeated until an election decides, as Algorithm 4 describes.

The conditions for each outcome are based on the EC the algorithm uses. They are the following.

Definition 6. decide-condition(w) *and* repeat-condition
Let $p \in U$ be a process in an EC $\mathcal{E}$.

- decide-condition(w) *at p is defined as:*
  $\exists w : \exists d \in \mathcal{E} : \langle v_p^w, \{v_p^j : j \neq w\}\rangle \succ d.$
- repeat-condition *at p is defined as:*
  $\forall w : \nexists d \in \mathcal{E} : \langle v_p^w, \{v_p^j : j \neq w\}\rangle \rhd d.$

Intuitively, each local vote set at p, v_i^p, grows monotonically as votes are cast and propagated. Ideally, the local vote sets should grow so that eventually they can form a q-vote configuration that covers one q-vote configuration in the EC. The set v_w^p that corresponds to the quorum in the former q-vote configuration determines the decision of value w. Hence, while *decide-condition(w)* is not verified yet, one may determine if the local vote sets may still form a q-vote configuration that may cover some q-vote configuration in the EC. If that does not hold, then the current election round will never decide any value, and a new one may then safely start; in this case, each v_i^p is emptied and its monotonic growth restarts.

As an example, recall the ECs $\mathcal{E}_{Maj}$ and $\mathcal{LP}$, introduced in Section 2.1. Consider that p has the following local vote sets: $v_p^a = \{p_2, p_4\}$, $v_p^b = \{p_3\}$ and $v_p^c = \{p_5\}$. At this moment, neither *decide-condition*(a), nor *decide-condition*(b), nor *decide-condition*(b), nor *repeat-condition* are verified (neither for $\mathcal{E}_{Maj}$, nor for $\mathcal{LP}$). Hence, p waits for further votes. Consider

that p later learns that p_1 has voted for c (i.e. $v_p^c = \{p_1, p_5\}$). If we consider $\mathcal{E}_{Maj}$, then clearly the local vote-sets of p_1 can cover no q-vote configuration in $\mathcal{E}_{Maj}$; hence, *repeat-condition* holds and a new election starts. If, otherwise, we consider $\mathcal{LP}$, p decides c, as there exists a q-vote configuration $d = \langle\{p_1, p_5\}, \{\{p_2\}, \{p_3\}\}\rangle$ in $\mathcal{LP}$, such that $\langle v_p^c, \{v_p^a, v_p^b\}\rangle$, i.e. $\langle\{p_1, p_5\}, \{\{p_2, p_4\}, \{p_3\}\}\rangle$, covers d.

The definition of EC ensures the safety of agreed values, according to the requirements of classical consensus [Lam05].

Lemma 1. *Let p and r be two processes where $e_p = e_r$. Then, $v_p^x \subseteq [v_r^x]_{c_r}$ for all $x \in Val$.*

Theorem 1. *The epidemic quorum algorithm satisfies the following requirements:*

1. *Any value decided must have been proposed at some process (nontriviality).*
2. *A process can only decide a single value (stability).*
3. *Two different processes cannot decide different values (consistency).*

Proof. Clearly, the algorithm ensures nontriviality and stability. To prove consistency, assume, for purposes of contradiction, that two processes, p and r, decide different values a and b (resp.), at elections e_p and e_r (resp.).

Let us first assume that $e_p = e_r$. By Def.6, when p decides, $\exists x \in \mathcal{E} : \langle v_p^a, \{v_p^j : j \neq a\}\rangle \succ x$. So, by Prop.1, for any $a^i \in Val$, either $\exists A_x^j$ such that $[A_x^j]_c \supseteq [v_p^{a^i}]_{c_p}$, or $[\emptyset]_c \supseteq [v_p^{a^i}]_{c_p}$. Since $\mathcal{E}$ is an EC, we know that, for any $y \in \mathcal{E}$ $(y \neq x)$, $\forall j : \langle A_x^j, \{Q_x\} \cup \{A_x^i : i \neq j\}\rangle \not\succ y$, and $\langle \emptyset, \{Q_x\} \cup \{A_x^1, .., A_x^{n_x}\}\rangle \not\succ y$; this implies $\langle v_p^b, \{v_p^j : j \neq b\}\rangle \not\succ y$ (1). However, since the outcome of the same election at r was *decide* for value b, then, at the time of such decision, $\exists y \in \mathcal{E} : \langle v_r^b, \{v_r^j : j \neq b\}\rangle \succ y$. Then, by Lemma 1, for any a^i, $v_r^{a^i} \subseteq [v_p^{a^i}]_{c_p}$. Hence, $\langle v_p^b, \{v_p^j : j \neq b\}\rangle \rhd \langle v_r^b, \{v_r^j : j \neq b\}\rangle$, and finally $\langle v_p^b, \{v_p^j : j \neq b\}\rangle \succ y$, which contradicts (1).

Let us now assume that $e_p < e_r$ (or vice-versa). Clearly, at least one process t (possibly $r = t$) must have determined the outcome of election e_p as *repeat*; otherwise, r would not have reached e_r. In such a moment, by Def. 6, $\nexists x \in \mathcal{E} : \langle [v_t^b]_{c_t}, \{[v_t^j]_{c_t} : j \neq b\}\rangle \succ x$. However, since the outcome of the same election at p was *decide*, $\exists y \in \mathcal{E} : \langle v_p^b, \{v_p^j : j \neq b\}\rangle \succ y$. It follows from Lemma 1 that $\forall i, v_p^i \subseteq [v_t^b]_{c_t}$. Hence, $\langle [v_t^b]_{c_t}, \{[v_t^j]_{c_t} : j \neq b\}\rangle \succ \langle v_p^b, \{v_p^j : j \neq b\}\rangle$, and thus $\langle [v_t^b]_{c_t}, \{[v_t^j]_{c_t} : j \neq b\}\rangle \succ y$, which is a contradiction. □

3 Liveness of Epidemic Quorum Systems

We characterize the liveness of an EC by its availability and the time to decide. Since we deal with a distributed system, measures are obtained at a particular, arbitrary process p; e.g. *availability* means the *availability seen by arbitrary*

process p. Also, we start measuring the time p takes to decide at the moment when p first voted for a value.

We divide time into *communication rounds* (or simply *rounds*). We assume a round to be an upper bound on the time needed for p to send and, subsequently, receive vote information to/from every other correct (i.e. not faulty) process in its current partition. Further, we assume communication is live between processes simultaneously in the same partition and partition changes occur only at the end of each communication round.

We use a simple probabilistic failure model where the probability of permanent, non-byzantine failures is constant and uniform; we denote it by p_f. For simplicity, we assume the probability that a correct process votes in a value in Val to be uniform, $\frac{1}{z}$. Finally, we assume eventual vote propagation between correct processes; i.e., in spite of transient partitioning, p is able to eventually send and receive vote information to/from any other correct process.

Given a particular EC $\mathcal{E}$, we characterize it by two probability values, $dec_{\mathcal{E}}(n)$ and $rep_{\mathcal{E}}(n)$. $dec_{\mathcal{E}}(n)$ (resp. $rep_{\mathcal{E}}(n)$) denotes the probability that p, having collected n votes in a given election, evaluates *decide-condition(w)* for some w (resp. *repeat-condition*) as true. More precisely:

Definition 7. *$dec_{\mathcal{E}}$ and $rep_{\mathcal{E}}$*

Let $T_{all}(n)$ be the collection of value-vote configurations such that, for each $vc_i \in T_{all}(n)$, $\sum_{s \in VCfg(vc_i)} |s| = n$. Given an EC $\mathcal{E}$, let $T_d^{\mathcal{E}}(n)$ and $T_r^{\mathcal{E}}(n)$ be:

$$T_d^{\mathcal{E}}(n) = \{c : c \in T_{all}(n) \text{ and } \exists s \in VCfg(c) : \exists d \in \mathcal{E} : \langle s, \{t : t \in (VCfg(c) \backslash \{s\})\}\rangle \succ d\}$$
$$T_r^{\mathcal{E}}(n) = \{c : c \in T_{all}(n) \text{ and } \forall s \in VCfg(c) : \nexists d \in \mathcal{E} : \langle s, \{t : t \in (VCfg(c) \backslash \{s\})\}\rangle \rhd d\}$$

We define $dec_{\mathcal{E}}$ and $rep_{\mathcal{E}}$ as: $dec_{\mathcal{E}}(n) = |T_d^{\mathcal{E}}(n)|/z^n$ and $rep_{\mathcal{E}}(n) = |T_r^{\mathcal{E}}(n)|/z^n$.

Clearly, $rep_{\mathcal{E}}(n)$ and $dec_{\mathcal{E}}(n)$ correspond to exclusive events; hence, $rep_{\mathcal{E}}(n) + dec_{\mathcal{E}}(n) \leq 1$ for all n. Hereafter, we assume that $\forall n : rep_{\mathcal{E}}(n) < 1$. Otherwise, $dec_{\mathcal{E}}(n)$ would be zero for all n, which would mean a useless EC that decided in no execution.

$dec_{\mathcal{E}}(n)$ and $rep_{\mathcal{E}}(n)$ allow us to abstract our study from each particular EC. We are now able to characterize availability and number of rounds to decide in any generic EC.

Hereafter, we write $\binom{x}{y}$ to denote the binomial coefficient of x and y.

3.1 Availability

First, we define availability as the probability that the system will eventually agree on some value. Since we assume eventual vote propagation between correct processes, transient partitioning is irrelevant when quantifying eventual decision; thus, we look at the system of correct processes as co-existing in the same (imaginary) partition.

Agreement is achieved by a possibly null, finite sequence of elections with a *repeat* outcome, followed by one election with a *decide* outcome.

Theorem 2. *The availability of EC $\mathcal{E}$ is given by:*

$$\sum_{n=0}^{y} \frac{\left(\begin{array}{c} y \\ n \end{array} \right) (1-p_f)^n \; p_f^{(y-n)} dec_{\mathcal{E}}(n)}{1 - rep_{\mathcal{E}}(n)} \tag{1}$$

Proof. Assuming n correct processes, the probability of eventual decision is the probability of any possible sequence of consecutive *repeats*, followed by one *decide*. Since such sequences are exclusive events, their probability is the following sum of a geometric series: $dec_{\mathcal{E}}(n) + dec_{\mathcal{E}}(n) rep_{\mathcal{E}}(n) + dec_{\mathcal{E}}(n) rep_{\mathcal{E}}(n)^2 + .. = dec_{\mathcal{E}}(n) \sum_{i=0}^{\infty} rep_{\mathcal{E}}(n)^{\;i} = \frac{dec_{\mathcal{E}}(n)}{1-rep_{\mathcal{E}}(n)}$. The probability of exactly n correct processes out of y is $\binom{y}{n}(1-p_f)^n \; p_f^{(y-n)}$. Again, the events of exactly n (with $0 \leq \; n \leq y$) processes are exclusive; hence, it follows directly that the probability for any n and any decision sequence is given by (1). □

Due to [FLP85], availability is less than one if at least one process may be faulty. Looking at the above expression, this implies $\forall n, dec_{\mathcal{E}}(n) + rep_{\mathcal{E}}(n) < 1$, which means that an election may block (i.e. never return any outcome).

3.2 Communication rounds to decide

We are also interested in the probability of decision within a given number of rounds, r. We denote such probability distribution as $P_d^{\mathcal{E}}(r)$. It characterizes system performance when agreement is possible.

In order to derive $P_d^{\mathcal{E}}(r)$, we need to obtain the probability that p collects new votes at each round. $p_v(v_i \rightarrow v_r; r)$ denotes the probability that, starting from v_i votes, p collects v_f votes after r rounds. p_v depends on how one models network inaccessibility due to transient partition changes. However, independently of such model, we know that $p_v(v_i \rightarrow v_r; r) = 0$ if $v_f < v_i$; it is trivial to show that the algorithm ensures such property.

One possible model of network inaccessibility is the following: at the end of a given communication round, the partition p belongs to does not include a given *correct* process with a constant and uniform probability, h. We designate such model as $\mathcal{M}_p$. The definition of p_v when $\mathcal{M}_p$ is assumed is given and proved in Appendix. Without loss of generality, we use $\mathcal{M}_p$ when needed for examples in the remainder of the paper.

We may finally obtain the probability of decision within a given number of rounds.

Theorem 3. *Probability of decision within r rounds*

The probability that an arbitrary process, p, starting with $V_p = \emptyset$, decides any value within r or less rounds, $P_d^{\mathcal{E}}(r)$, is given by:

$$P_d^{\mathcal{E}}(r) = \left\{ \begin{array}{ll} 0, & \textit{if} \quad r = 0 \\ \sum_{v_r=0}^{y} p_v(0 \rightarrow v_r; r) dec_{\mathcal{E}}(v_r) + f(0,r), & \textit{if} \quad r > 0 \end{array} \right. \tag{2}$$

where f is defined as:

$$f(v_i, r) = \sum_{v_x=v_i}^{y} p_v(v_i \rightarrow v_x; 1)[(rep_{\mathcal{E}}(v_x) - rep_{\mathcal{E}}(v_i)) P_d^{\mathcal{E}}(r-1) + f(v_x, r-1)] \tag{3}$$

Naturally, $P_d^{\mathcal{E}}(0)$ is zero. With $r > 0$, one may distinguish two situations that fulfill *decide-condition*, (a) and (b) ($P_d^{\mathcal{E}}(r > 0) = P(a) + P(b)$). Figure 2 depicts such situations. (a) corresponds to the cases where, after r or less rounds, p collected enough votes, v_r, to decide in a single election. The cases where at least one *repeat-condition* is verified, and followed by a *decide*, correspond to (b). A detailed proof is provided in Appendix.

Corollary 1. *Availability grows as $dec_{\mathcal{E}}$ and $rep_{\mathcal{E}}$ grow.*

Corollary 2. *$P_d^{\mathcal{E}}$ grows as $dec_{\mathcal{E}}$ grows, and decreases as $rep_{\mathcal{E}}$ grows.*

Proof. By standard manipulation of the expressions in Theorems 2 and 3. □

The above corollaries unveil an interesting trade-off between availability and performance that is intrinsic to ECs. It is not present in the classical counterpart. Its effective outcome on the availability and performance of an epidemic quorum system is not evident and depends on each specific EC. The next section illustrates the trade-off with the liveness study of the majority ($\mathcal{E}_{Maj}$) and linear plurality ($\mathcal{LP}$) ECs, defined in Section 2.2.

3.3 Liveness Study of Majority and Linear Plurality ECs

From $\mathcal{E}_{Maj}$ and $\mathcal{LP}$(see Figure 1), and by standard combinatorics, one may obtain the corresponding $dec_{\mathcal{E}_{Maj}}$ and $rep_{\mathcal{E}_{Maj}}$, and $dec_{\mathcal{LP}}$ and $rep_{\mathcal{LP}}$, respectively. Figures 3 and 4 depict such values, for different z and n values. As expected, $rep_{\mathcal{LP}}$ is always zero, as $\mathcal{LP}$ may always decide, provided enough votes are collected. Further, $dec_{\mathcal{E}_{Maj}}$ and $dec_{\mathcal{LP}}$ are similar when z is either 1 or 2; in this case, the q-vote configurations of $\mathcal{LP}$ with more than one anti-quorum are of no use, and thus, in practice, $\mathcal{LP}$ induces a similar behavior as $\mathcal{E}_{Maj}$.

Most importantly, when $z > 2$, $dec_{\mathcal{LP}}$ is higher than $dec_{\mathcal{E}_{Maj}}$, while $rep_{\mathcal{LP}}$ is lower than $rep_{\mathcal{E}_{Maj}}$. Intuitively, this suggests the following trade-off. On one hand, the higher $dec_{\mathcal{LP}}$ means that p will more easily satisfy *decide-condition(w)*, (i.e. with less votes), thus increasing availability and performance.

On the other hand, when one considers *repeat-condition*, the effect of larger ECs is inverse. The higher $dec_{\mathcal{E}_{Maj}}$ allows p to more easily fulfill *repeat-condition*. A *repeat* outcome is useful in situations where the local vote set has reached a state where *decide-condition(w)* depends on the votes of a number of inaccessible processes (e.g. permanently faulty processes). In such situation, a *repeat* outcome triggers a new election, which acts as a new opportunity for decision; in spite of the inaccessible processes, the new election may eventually lead to a different local vote configuration that finally fulfills *repeat-condition*. If, instead, *repeat-condition* is not met, the election outcome is halted until new votes arrive, which may never happen. Therefore, the higher $rep_{\mathcal{E}_{Maj}}$ increases availability, at the cost of more elections (hence, more communication rounds).

We analytically deduce the effective outcome of the trade-off using Theorems 2 and 3. Figures 5, 6 and 7 present the availability and performance results, assuming $p_f = 0.1$ and the $\mathcal{M}_p$ with $h = 0.1$. We can see that, in

this case, the effect of a higher $dec_{\mathcal{LP}}$ always dominates the effect of a higher rep_{MajEp}, as $\mathcal{LP}$always achieves higher availability and performance. Though not presented herein, the advantage of $\mathcal{LP}$ is verified for any value of p_f and h.

4 Comparison with Classical Quorum Systems and Discussion

The availability of classical quorum systems is extensively studied [JM90, CW96, IK01, PW95, JM05]. However, such results are not valid for epidemic quorum systems, as both approaches differ radically in the way processes obtain quorums. In the classical approach, a coordinator (typically the process proposing the value) runs a synchronous atomic commit (e.g. two-phase commit) to obtain a quorum to atomically vote for a value. The epidemic approach collects votes in an asynchronous coordinator-free way, by means of pair-wise epidemic propagation of vote information. Furthermore, in the classical approach, the coordinator decides whether to proceed with the agreement (if it has obtained a quorum of voters), or to withdraw the votes (due to unavailable quorum). The epidemic approach does not allow vote withdrawal before agreement, and agreement is a local decision of each process, based on the locally collected vote information.

In consequence, ECs, in contrast to classical coteries, may take into account, not only the set of voters for a given value (the *quorum*), but also the set of voters for the competing values (the *anti-quorums*); plurality-based weighted voting [Kel99] is an example of a coterie that the classical approach does not allow.

One may compare classical and epidemic quorum systems by trying to answer two questions: (a) can one build an epidemic quorum system from a classical one (and vice-versa)?; and (b) given a classical and an epidemic quorum system, which one is better?

The first question has already been partially answered by Holliday et al. [HSAA03]. They propose epidemic quorum systems constructed from classical coteries [PW95]. A classical coterie is a set of sets of processes, $Q_1, ..Q_m$, such that, for all Q_i and Q_j, $Q_i \cap Q_j \neq \emptyset$ (intersection property) and $\forall Q_j \in QSet(\mathcal{E}_{cl}), Q_i \nsubseteq Q_j$ (minimality property). We formalize their idea and prove it safe as follows.

Definition 8. *EC equivalent of classical coterie*

Let $\mathcal{C}$ be a classical coterie. Its EC equivalent, denoted $\mathcal{E}_{\mathcal{C}}$, is a set of q-vote configurations such that $QSet(\mathcal{E}_{\mathcal{C}}) = \mathcal{C}$ and $AQSet(\mathcal{E}_{\mathcal{C}}) = \emptyset$.

Proposition 2. *Let $\mathcal{C}$ be a classical coterie. Its EC equivalent, $\mathcal{E}_{\mathcal{C}}$, is an EC.*

The $\mathcal{E}_{Maj}$ EC introduced in Section 2.2 is an example of an EC equivalent of a classical coterie.

Since the universe of ECs is larger than the universe of EC equivalents, one cannot always build a classical quorum system from an epidemic one. An example is the $\mathcal{LP}$ EC.

Concerning the second question, we may try to compare our results with the results available for classical quorum systems. Consider an EC, $\mathcal{E}$, and a classical coterie, $\mathcal{C}$. Further, assume $\mathcal{M}_p$.

The probability of a quorum of $\mathcal{C}$ being accessible to p, $q_{\mathcal{C}}$, is given by $q_{\mathcal{C}} = 1 - \sum_{i=0}^{y} a_i^{\mathcal{C}} f^i (1-f)^{y-i}$; where $(a_0^{\mathcal{C}}, .., a_y^{\mathcal{C}})$ is the availability profile of $\mathcal{C}$ [PW95], and f denotes the probability of each process either being correct or not in the same partition as p. Using our model, f is given by $p_f + (1-p_f)h$.

Hence, the probability of a quorum being accessible to p within r rounds, $q_{\mathcal{C}}^r$ is given by the sum of a geometric series:

$$q_{\mathcal{C}}^r = \sum_{x=0}^{r-1} (1-q_{\mathcal{C}})^x q_{\mathcal{C}} = \frac{q_{\mathcal{C}}(1-(1-q_{\mathcal{C}})^r)}{q_{\mathcal{C}}}$$

$q_{\mathcal{C}}$ is an upper bound on the effective availability of $\mathcal{C}$, as it ignores contention and faults that may occur during the quorum obtention process.[1] Further, $q_{\mathcal{C}}^r$ is an upper bound on $P_d^{\mathcal{C}}$. Still, it enables us to determine a sufficient (not necessary) condition for $\mathcal{E}$ to outperform $\mathcal{C}$:

$$P_d^{\mathcal{E}}(r) > q_{\mathcal{C}}^r, \text{ for all } r > 0.$$

The above condition is a partial answer to the second question. A more precise definition of $q_{\mathcal{C}}$, which takes into account the effects of contention and faults during quorum obtention, would produce a more satisfactory comparison of classical and epidemic quorum systems. This is a goal for future work.

5 Related Work

Epidemic algorithms have recently become popular due to their robustness and scalability [EGKM04]. However, to the best of our knowledge, little work has been devoted to the formal study epidemic quorum systems.

Holliday et al. [HSAA03] have proposed epidemic quorum systems built from classical coteries, introducing the notion of anti-quorums. Their goal is to ensure global serialization of updates in a replicated system. Their algorithm may only decide in the first election, and aborts in the case of a *repeat*. Our study can be made applicable to such algorithm by artificially defining *repeat-condition* as always $false$ and $rep_{\mathcal{E}}(n) = 0$ for all $\mathcal{E}$ and n.

Keleher [Kel99] has proposed a weighted voting system based on linear plurality, for the purpose of data replication. Its linear plurality system differs slightly from ours since ties are broken by comparison of the proposers of the contending values, rather than by comparison of their voters; our formalism may be enhanced so as to support such EC. Being based on linear plurality, Keleher's algorithm only takes one election to decide a value, as *repeat* is an impossible outcome in linear plurality.

VVWV [BF05] is an epidemic commitment protocol that optimizes Keleher's algorithm to be able to decide sequences of multiple happens-before-related of updates in a single election. Sutra et al. [SSB06] generalize VVWV to the case where rich semantics are available to the protocol.

[1] As Ingols and Keidar note and experimentally study in [IK01].

The above systems have been evaluated both in simulations ([BF05, HSAA03, Kel99]) and empirically ([CKBF03]). However, comparative results (either formal or experimental) between different ECs (e.g. majority vs. linear plurality) are nonexistent. Further, neither is the liveness of the proposed epidemic quorum systems formally studied, nor has it been compared to the classical counterpart. We formally study such problems.

The Paxos algorithm [Lam98] allows multiple election instances in an asynchronous system in order to deal with elections where majorities are not achievable (as an epidemic quorum algorithm, as we define it, does). Nevertheless, Paxos is fundamentally different from our algorithm, requiring two rounds of messages to decide, whereas ours may decide in one round. As future work, we plan to analytically compare our algorithm with Paxos. A further point of future work is to generalize Paxos to work with ECs.

6 Concluding Remarks

Although recent work has proposed epidemic quorum algorithms, neither are epidemic quorum systems well defined nor is their liveness well studied. We formalize epidemic quorum systems and provide a generic characterization of their availability and performance. Our contribution highlights previously undocumented trade-offs that arise in epidemic quorum systems. Further, to the best of our knowledge, it provides the first analytical comparative results between proposed epidemic and classical quorum systems. Finally, the formalism serves as a framework for the definition of novel epidemic quorum systems, and for subsequent obtention of new results concerning these systems.

References

[AW96] Yair Amir and Avishai Wool. Evaluating quorum systems over the internet. In *Symposium on Fault-Tolerant Computing*, pages 26–35, 1996.

[BF05] João Barreto and Paulo Ferreira. An efficient and fault-tolerant update commitment protocol for weakly connected replicas. In *Euro-Par*, pages 1059–1068, 2005.

[CKBF03] U. Cetintemel, P. J. Keleher, B. Bhattacharjee, and M. J. Franklin. Deno: A decentralized, peer-to-peer object replication system for mobile and weakly-connected environments. *IEEE Transactions on Computer Systems (TOCS)*, 52, July 2003.

[CW96] Ing-Ray Chen and Ding-Chau Wang. Analyzing dynamic voting using petri nets. In *SRDS '96: Proceedings of the 15th Symposium on Reliable Distributed Systems (SRDS '96)*, page 44, Washington, DC, USA, 1996. IEEE Computer Society.

[EGKM04] P. Eugster, R. Guerraoui, A. M. Kermarrec, and L. Massoulie. From epidemics to distributed computing. *IEEE Computer*, 37(5):60–67, May 2004.

[FLP85] Michael J. Fischer, Nancy A. Lynch, and Michael S. Paterson. Impossibility of distributed consensus with one faulty process. *J. ACM*, 32(2):374–382, April 1985.

[HSAA03] JoAnne Holliday, Robert Steinke, Divyakant Agrawal, and Amr El Abbadi. Epidemic algorithms for replicated databases. *IEEE Transactions on Knowledge and Data Engineering*, 15(5):1218–1238, 2003.

[IK01] K. Ingols and I. Keidar. Availability study of dynamic voting algorithms. In *ICDCS '01: Proceedings of the The 21st International Conference on Distributed Computing Systems*, page 247, Washington, DC, USA, 2001. IEEE Computer Society.

[JM90] S. Jajodia and David Mutchler. Dynamic voting algorithms for maintaining the consistency of a replicated database. *ACM Trans. Database Syst.*, 15(2):230–280, 1990.

[JM05] Flavio Paiva Junqueira and Keith Marzullo. Coterie availability in sites. In *DISC*, pages 3–17, 2005.

[Kel99] P. Keleher. Decentralized replicated-object protocols. In *Proc. of the 18th Annual ACM Symp. on Principles of Distributed Computing (PODC'99)*, 1999.

[Lam98] Leslie Lamport. The part-time parliament. *ACM Transactions on Computer Systems*, 16(2):133–169, May 1998. `http://doi.acm.org/10.1145/279227.279229`.

[Lam05] Leslie Lamport. Generalized consensus and Paxos. Technical Report MSR-TR-2005-33, Microsoft Research, March 2005. `ftp://ftp.research.microsoft.com/pub/tr/TR-2005-33.pdf`.

[PW95] David Peleg and Avishai Wool. The availability of quorum systems. *Information and Computation*, 123(2):210–223, 1995.

[SSB06] Pierre Sutra, Marc Shapiro, and Joao Barreto. An asynchronous, decentralised commitment protocol for semantic optimistic replication. Research Report 6069, INRIA, 12 2006.

Complementary Figures

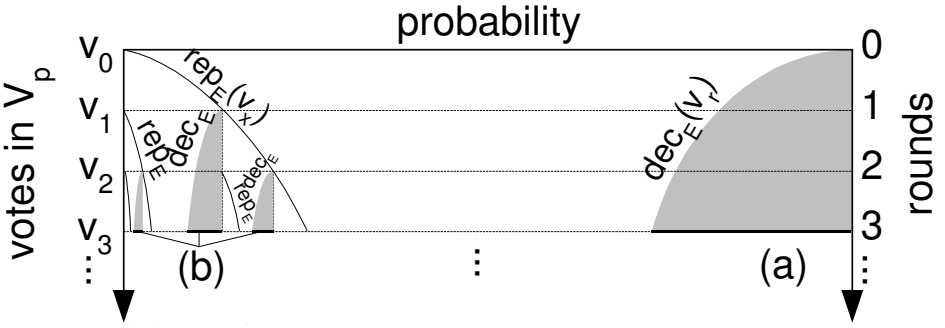


Figure 2: Illustration of the probabilities that comprise $P_d^{\mathcal{E}}(r)$, up to 3 rounds. Decision cases are greyed out, and correspond to components (a) and (b).

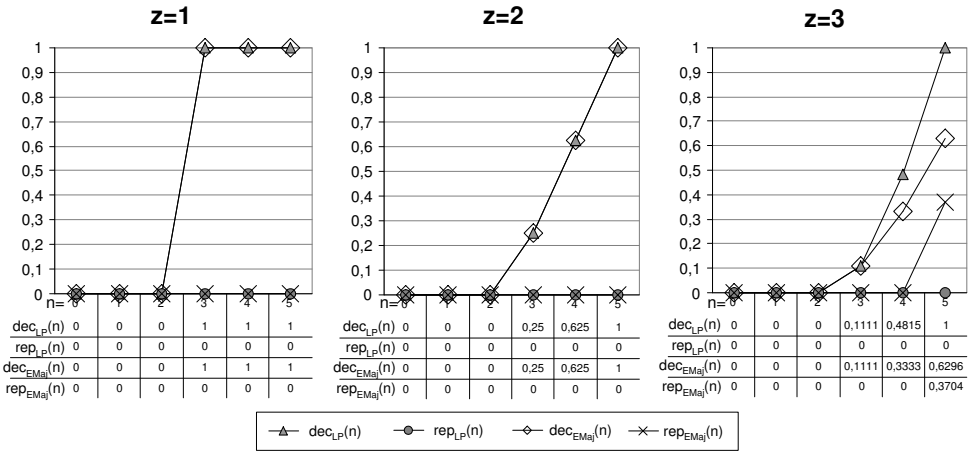


Figure 3: $dec_{\mathcal{E}_{Maj}}$ and $rep_{\mathcal{E}_{Maj}}$, and $dec_{\mathcal{LP}}$ and $rep_{\mathcal{LP}}$, in a system with five processes (1 to 3 concurrent values proposed).

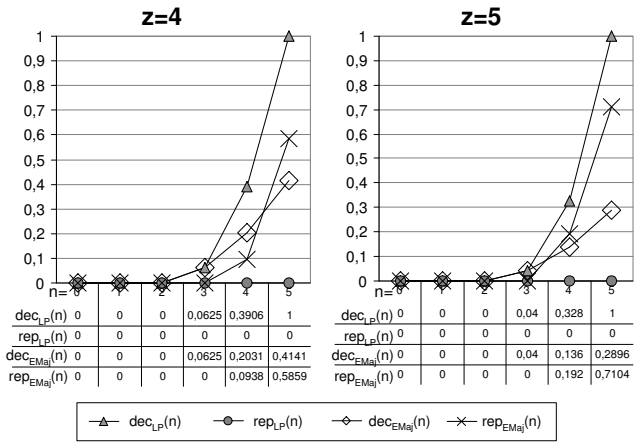


Figure 4: $dec_{\mathcal{E}_{Maj}}$ and $rep_{\mathcal{E}_{Maj}}$, and $dec_{\mathcal{LP}}$ and $rep_{\mathcal{LP}}$, in a system with five processes (4 and 5 concurrent values proposed).

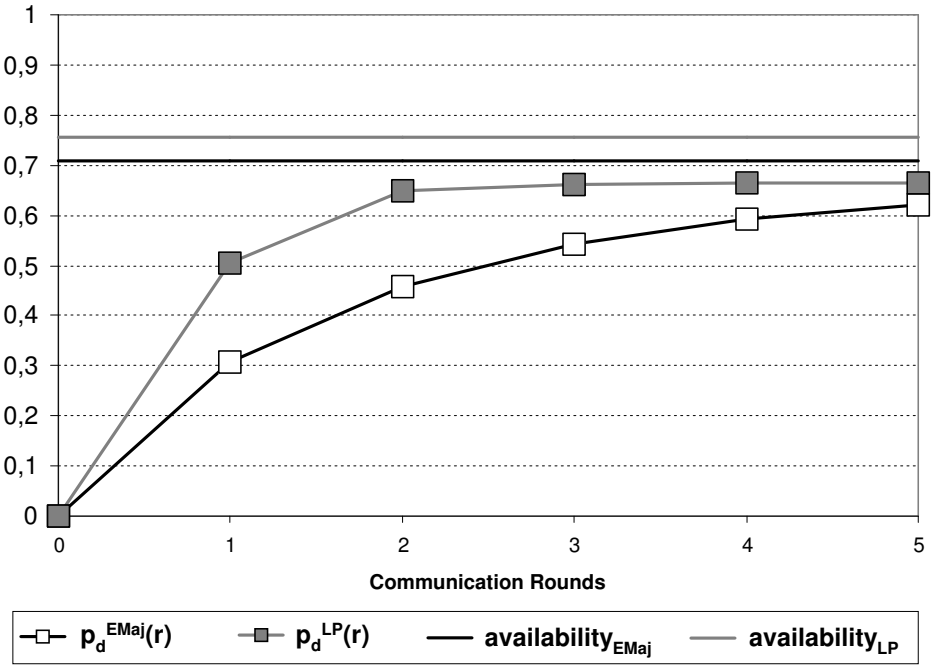


Figure 5: Decision probabilities of within r rounds of $\mathcal{LP}$and $\mathcal{E}_{Maj}$ ECs, assuming $p_f = 0.1$, $h = 0.1$, and 3 concurrent values proposed ($z = 3$).

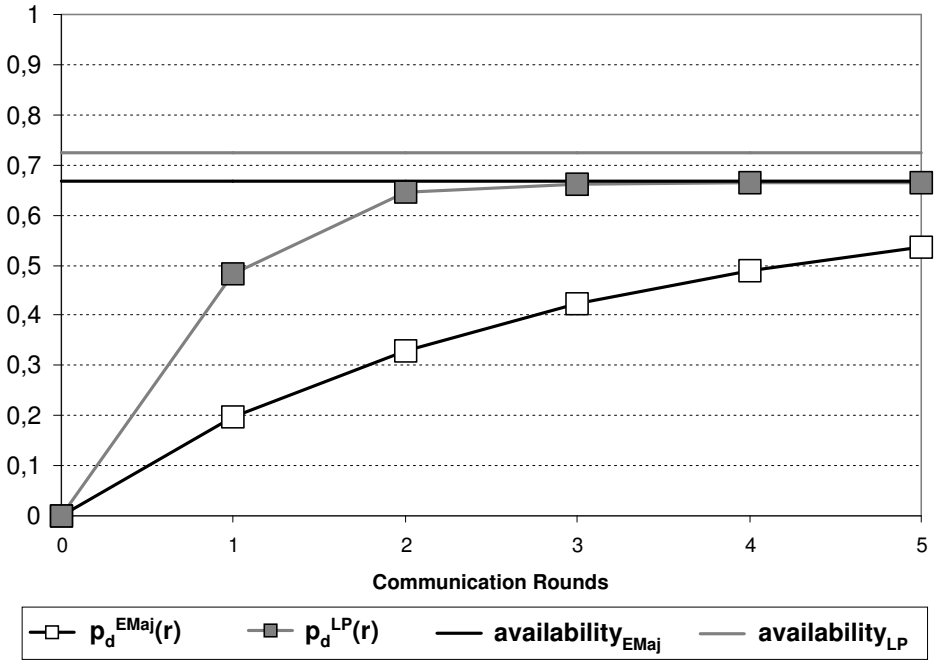


Figure 6: Decision probabilities of within r rounds of $\mathcal{LP}$and $\mathcal{E}_{Maj}$ ECs, assuming $p_f = 0.1$, $h = 0.1$, and 4 concurrent values proposed ($z = 4$).

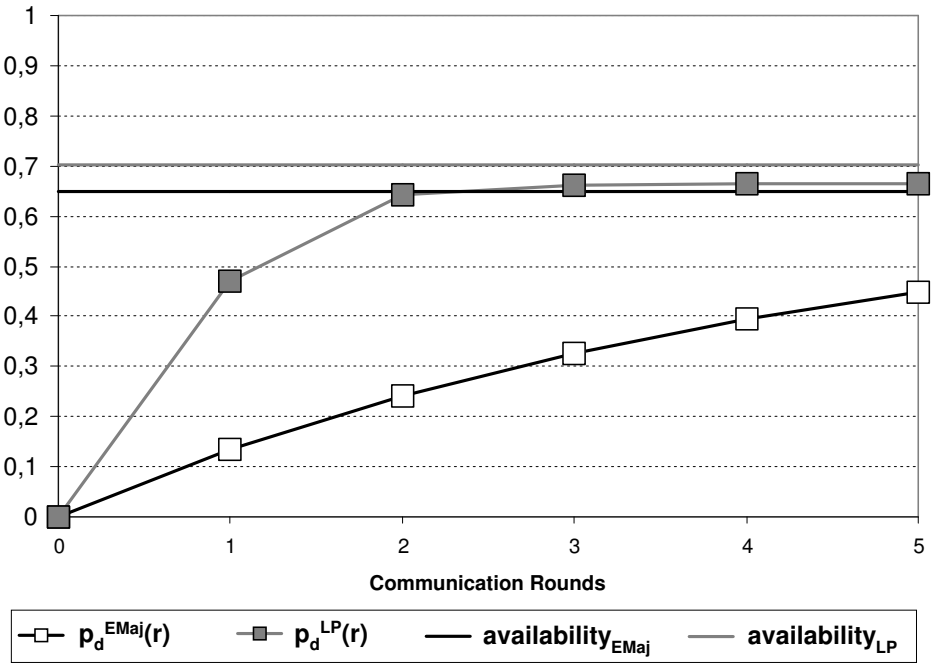


Figure 7: Decision probabilities of within r rounds of $\mathcal{LP}$and $\mathcal{E}_{Maj}$ ECs, assuming $p_f = 0.1$, $h = 0.1$, and 5 concurrent values proposed ($z = 5$).

A Complementary Proofs of Section 2.1

Proposition 1. Let c and d be q-vote configurations such that $d \succ c$. For any A_d^i, either there exists A_c^j such that $[A_c^j]_c \supseteq [A_d^i]_d$, or $[\emptyset]_c \supseteq [A_d^i]_d$.

Proof. Clearly, $\bigcup_i A_i^c \subseteq \bigcup_j A_j^d$, which implies $[\emptyset]_c \supseteq [\emptyset]_d$ (1). Let f be the injective function such that $\forall_{1 \leq k \leq n_c} : A_c^k \subseteq A_d^{f(k)}$. For each $i \in \{1, .., n_c\}$, $A_c^i \subseteq A_d^j$, thus $A_d^j = A_c^i \cup (A_d^j \backslash A_c^i)$, since $c \subseteq d$. Since d is a q-vote configuration, $A_d^j \cap A_d^x = \emptyset, \forall x \neq j$; which implies $(A_d^j \backslash A_c^i) \cap A_d^x = \emptyset$ (2). Since $\forall A_c^y : y \neq i, \exists A_d^j : A_d^j \supseteq A_c^y$, then it follows from (2) that $(A_d^j \backslash A_c^i) \cap A_c^y = \emptyset$. Thus, $(A_d^j \backslash A_c^i) \subseteq [\emptyset]_c$. Finally, $[A_d^j]_d = A_d^j \cup [\emptyset]_d = A_c^i \cup (A_d^j \backslash A_c^i) \cup [\emptyset]_d$, which implies, by (1) and (2), $[A_d^j]_d \subseteq A_c^i \cup [\emptyset]_c$.

The proof for the case of each A_d^j such that $\nexists i \in \{1, .., n_c\} : f(i) = j$ is direct. Clearly, $A_d^j \cap A_c^x = \emptyset, \forall x \neq j$; so, $A_d^j \subseteq [\emptyset]_c$. Therefore, $[A_d^j]_d = A_d^j \cup [\emptyset]_d \subseteq [\emptyset]_c$. □

B Complementary Proofs of Section 2.3

Lemma 1. Let p and r be two processes where $e_p = e_r$. Then, $v_p^x \subseteq [v_r^x]_{c_r}$ for all $x \in Val$.

Proof. Clearly, the algorithm ensures that only the vote of a given process may only be cast by the process itself, and only once each election. Hence, $v_p^x \cap v_r^y = \emptyset$ for any processes p, r and values x, y. Therefore, $v_p^x \backslash v_r^y \subseteq [\emptyset]_{c_r}$. By standard set manipulation, we get $[v_p^x]_{c_r} = v_p^x \cup [\emptyset]_{c_r} \supseteq (v_p^x \cap v_r^y) \cup [\emptyset]_{c_r} \supseteq (v_p^x \cap v_r^y) \cup (v_p^x \backslash v_r^y) = v_p^x$. □

C Complementary Proofs of Section 3

Lemma 2. *In $\mathcal{M}_p$, p_v is given by:*

$$p_v(v_i \rightarrow v_f; r) = \begin{cases} \sum_{n=0}^{v_i} \binom{v_i}{n} \binom{y - v_i}{y - v_f} h^{(y - v_f + n)} (1-h)^{(v_f - n)}, & \text{if } r = 1 \\ \sum_{v_x = v_i}^{v_f} p_v(v_x \rightarrow v_f; 1) p_v(v_i \rightarrow v_x; r-1), & \text{if } r > 1 \end{cases} \quad (4)$$

Proof. Starting with v_i votes, at the end of one round, the number of votes (v_f) is the number of correct processes in the partition of p whose vote was not yet known by p, plus v_i (where n of the voters that have contributed to v_i are inaccessible). As Figure 8 depicts, the number of inaccessible processes, f, is given by $y - v_f + n$.

$p_v(v_i \rightarrow v_f; 1)$ is given by the summing the probabilities of collecting v_f votes for each n, ranging from 0 to v_i. For each value n, p collects v_f votes at the end of the round if: (i) n out of the v_i votes is inaccessible, and (ii) out of the processes that started the round without vote ($y - v_i$), $y - v_f$ remain in that condition. By simple combinatorial reasoning, the number of such

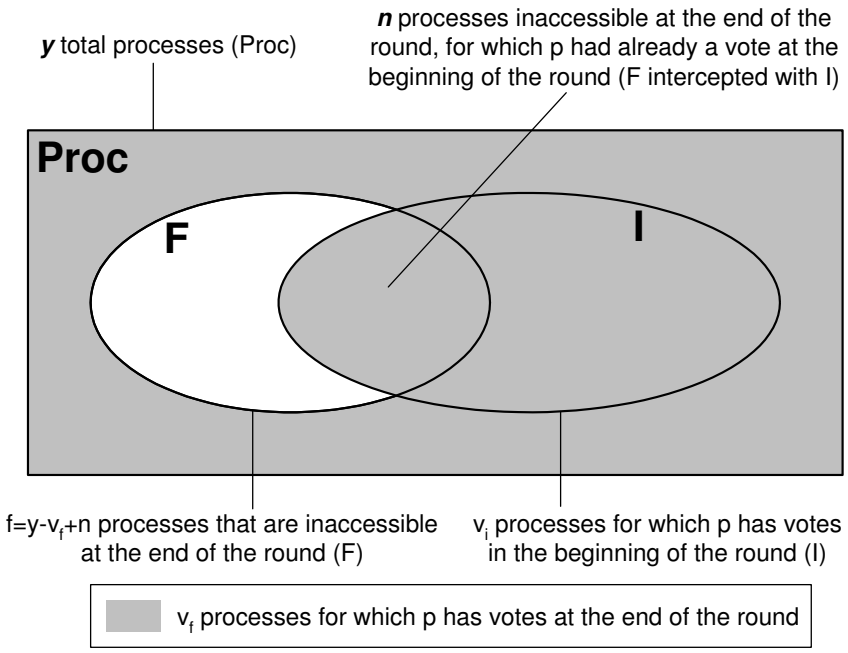


Figure 8: Event sets when considering a round where p, starting with v_i votes, ends up collecting v_r votes.

events is given by $\binom{v_i}{n}\binom{y-v_i}{y-v_f}$. Further, the probability of each such event is the probability of exactly f processes being inaccessible at the end of the round; i.e. $h^{(y-v_f+n)}(1-h)^{(v_f-n)}$.

The case of $r > 1$ is simpler. At a first round, p may collect v_x votes (where $v_i \leq v_x \leq v_f$) (with probability $p_v(v_x \rightarrow v_f; 1)$) and collect the votes still missing in the remaining rounds (with probability $p_v(v_i \rightarrow v_x; r-1)$). Clearly, $p_v(v_i \rightarrow v_f; r)$ is given by the sum of the probabilities for each v_x. □

Theorem 3. Probability of decision within r rounds

The probability that an arbitrary process, p, starting with $V_p = \emptyset$, decides any value within r or less rounds, $P_d^{\mathcal{E}}(r)$, is given by:

$$P_d^{\mathcal{E}}(r) = \begin{cases} 0, & \text{if} \quad r = 0 \\ \sum_{v_r=0}^{y} p_v(0 \rightarrow v_r; r) dec_{\mathcal{E}}(v_r) + f(0, r), & \text{if} \quad r > 0 \end{cases} \tag{5}$$

where f is defined as:

$$f(v_i, r) = \sum_{v_x=v_i}^{y} p_v(v_i \rightarrow v_x; 1)[(rep_{\mathcal{E}}(v_x) - rep_{\mathcal{E}}(v_i)) P_d^{\mathcal{E}}(r-1) + f(v_x, r-1)] \tag{6}$$

Proof. $P_d^{\mathcal{E}}(0)$ is zero since *decide-condition* is impossible with an empty V_p. With $r > 0$, one may distinguish two situations that fulfill *decide-condition*,

(a) and (b); Figure 2 depicts such situations. (a) corresponds to the cases where, after r or less rounds, p collected enough votes, v_r, to decide in a single election. The probability of (a) is the sum, for each v_r, of the probability of collecting exactly v_r votes *and* deciding with such an amount of votes; i.e., $\sum_{v_r=0}^{y} p_v(0 \rightarrow v_r; r) dec_{\mathcal{E}}(v_r)$.

The cases where at least one *repeat-condition* is verified, and followed by a *decide*, correspond to (b). We will show that, at the start of a given round, having v_i initial votes, the probability of deciding, including at least one *repeat*, within the next r or less rounds, is given by $f(v_i, r)$. In such round, at the end of which $v_x \geq v_i$ votes are collected, $f(v_i, r)$ is given by the sum of the probabilities of two exclusive events (multiplied by the probability of collecting v_x votes, $p_v(v_i \rightarrow v_x; 1)$, for each $v_r \geq v_i$): (i) either v_x produce the *repeat* outcome, and so a new election starts; or (ii) no such outcome occurs. Hence, (i) happens if, at the end of the round, p determines *repeat without previously having obtained such outcome with* v_i *votes* (probability of $rep_{\mathcal{E}}(v_x) - rep_{\mathcal{E}}(v_i)$) and decides in the remaining elections ($P_d^{\mathcal{E}}(r-1)$). Otherwise (ii), a *decide* after at least one *repeat* may still occur, starting with the new v_r votes, and within the remaining rounds $(r-1)$ (with probability $f(v_x, r-1)$).

Clearly, (b) is obtained by $f(0, r)$. $(a) + (b)$ gives the probability $P_d^{\mathcal{E}}(r)$ when $r > 0$. □

D Complementary Proofs of Section 4

Theorem 2. Let $\mathcal{C}$ be a classical coterie. Its EC equivalent, $\mathcal{E}_{\mathcal{C}}$, is an EC.

Proof. By Def. 4, and since has empty anti-quorums, $\mathcal{E}_{\mathcal{C}}$ is an EC if, $\forall qc, qd \in \mathcal{E}_{\mathcal{C}}$, (1) $\forall j : [A_{qc}^j]_{qc} \not\supseteq Q_{qd}$, (2) $[\emptyset]_{qc} \not\supseteq Q_{qd}$, and (3)$qc \not\subseteq qd$ and $qd \not\subseteq qc$. (1) is vacuously true. (3) is ensured by the minimality property. To prove (2), assume, for the purpose of contradiction, that there exists $qd \in \mathcal{E}_{\mathcal{C}}$ such that $[\emptyset]_{qc} \supseteq Q_{qd}$. By the intersection property, $\exists X \neq \emptyset : Q_{qc} \cap Q_{qd} = X$. Therefore, $X \subseteq Q_{qd} \subseteq [\emptyset]_{qc}$ (hence, $X \subseteq [\emptyset]_{qc}$), and $X \subseteq Q_{qc}$. So, $X \subseteq Q_{qc} \cap [\emptyset]_{qc}$. Thus $Q_{qc} \cap [\emptyset]_{qc} \neq \emptyset$; however, it follows directly from Def. 2 that $Q_{qc} \cap [\emptyset]_{qc} = \emptyset$, which is a contradiction. □