THE LOAD, CAPACITY AND AVAILABILITY OF QUORUM SYSTEMS*

MONI NAOR† AND AVISHAI WOOL‡

Abstract.
A *quorum system* is a collection of sets (quorums) every two of which intersect. Quorum systems have been used for many applications in the area of distributed systems, including mutual exclusion, data replication and dissemination of information

Given a strategy to pick quorums, the load $\mathcal{L}(\mathcal{S})$ is the minimal access probability of the busiest element, minimizing over the strategies. The capacity $Cap(\mathrm{S})$ is the highest quorum accesses rate that $\mathcal{S}$ can handle, so $Cap(\mathcal{S}) = 1/\mathcal{L}(\mathcal{S})$.

The availability of a quorum system $\mathcal{S}$ is the probability that at least one quorum survives, assuming that each element fails independently with probability p. A tradeoff between $\mathcal{L}(\mathcal{S})$ and the availability of $\mathcal{S}$ is shown.

We present four novel constructions of quorum system, all featuring optimal or near optimal load, and high availability. The best construction, based on paths in a grid, has a load of $O(1/\sqrt{n})$, and a failure probability of $\exp(-\Omega(\sqrt{n}))$ when the elements fail with probability $p < \frac{1}{2}$. Moreover, even in the presence of faults, with exponentially high probability the load of this system is still $O(1/\sqrt{n})$. The analysis of this scheme is based on Percolation Theory.

Key words. quorum systems, load, fault tolerance, distributed computing, percolation theory, linear programming.

AMS subject classifications. 60K35, 62N05, 68M10, 68Q22, 68R05, 90A28, 90C05.

1. Introduction.

1.1. Motivation. *Quorum systems* serve as a basic tool providing a uniform and reliable way to achieve coordination between processors in a distributed system. Quorum systems are defined as follows. A *set system* is a collection of sets $\mathcal{S} = \{S_1, \ldots, S_m\}$ over an underlying universe $U = \{u_1, \ldots, u_n\}$. A set system is said to satisfy the *intersection property,* if every two sets $S, R \in \mathcal{S}$ have a nonempty intersection. Set systems with the intersection property are known as *quorum systems*, and the sets in such a system are called quorums.

Quorum systems have been used in the study of problems such as *mutual exclusion* (cf. [39]), *data replication protocols* (cf. [7, 18]), *name servers* (cf. [32]), *selective dissemination of information* (cf. [46]), and *distributed access control and signatures* (cf. [34]).

A protocol template based on quorum systems works as follows. In order to perform some action (e.g., update the database, enter a critical section), the user selects a quorum and *accesses all its elements.* The intersection property then guarantees that the user will have a consistent view of the current state of the system. For example, if all the members of a certain quorum give the user permission to enter the critical section, then any other user trying to enter the critical section before the first user

* A preliminary version of this paper appeared in the 35'th IEEE Symp. Found. of Comp. Science, 1994.

† Dept. Applied Mathematics and Computer Science, The Weizmann Institute, Rehovot 76100, Israel. Incumbent of the Morris and Rose Goldman Career Development Chair, supported by an Alon fellowship and by a grant from the Israel Science Foundation administered by the Israeli Academy of Sciences and Humanities. E-mail: naor@wisdom.weizmann.ac.il.

‡ Bell Laboratories, Lucent Technologies, 700 Mountain Avenue, Murray Hill, NJ 07974. Work done while at the Department of Applied Mathematics and Computer Science, The Weizmann Institute, Rehovot 76100, Israel. E-mail: yash@research.bell-labs.com.

has exited (and released the permission-granting quorum from its lock) will be refused permission by at least one member of any quorum it chooses to access.

In this work we consider three criteria of measuring how good a quorum system is:

1. `Load` - A strategy is a rule giving each quorum an access frequency (so that the frequencies sum up to 1). A strategy induces a load on each element, which is the sum of the frequencies of all quorums it belongs to. This represents the fraction of the time an element is used. For a given quorum system $\mathcal{S}$, the load $\mathcal{L}(\mathcal{S})$ is the minimal load on the busiest element, minimizing over the strategies. The load measures the quality of a quorum system in the following sense. If the load is low, then each element is accessed rarely, thus it is free to perform other unrelated tasks.

2. `Capacity` - We would like the system to handle as many requests as possible. For this purpose we define $a(\mathcal{S}, k)$, the number of quorum accesses that $\mathcal{S}$ can handle during a period of k time units. This is the maximal t for which there exists a way to schedule t quorum accesses, to quorums $S^1, \dots, S^t$ (allowing repetitions), such that no element is accessed more than k times. The capacity $Cap(\mathcal{S})$ is then the limit as $k \to \infty$ of $a(\mathcal{S}, k)$ normalized by k.

3. `Availability` - assuming that each element fails with probability p, what is the probability, F_p, that the surviving elements do not contain any quorum? This failure probability measures how resilient the system is, and we would like F_p to be as small as possible.

Our goal is to investigate these criteria and find quorum systems that perform well according to all three of them.

1.2. Related Work. The first distributed control protocols using quorum systems [42, 14] use *voting* to define the quorums. Each processor has a number of votes, and a quorum is any set of processors with a combined number of votes exceeding half of the system's total number of votes. The simple majority system is the most obvious voting system.

The availability of voting systems is studied in [4]. It is shown that in terms of availability, the majority is the best quorum system when $p < \frac{1}{2}$. In [35] the failure probability function F_p is characterized, and among other things it is shown that the singleton has the best availability when $p > \frac{1}{2}$. The case when the elements fail with different probabilities p_i, all less than $\frac{1}{2}$, is addressed in [41].

The first paper to explicitly consider mutual exclusion protocols in the context of intersecting set systems is [13]. In this work the term *coterie* and the concept of *domination* are introduced. Several basic properties of dominated and non-dominated coteries are proved.

Alternative protocols based on quorum systems (rather than on voting) appear in [28] (using finite projective planes), [1] (the Tree system), [5, 25] (using a grid) and [23, 24, 38] (hierarchical systems). The triangular system is due to [26, 9]. The Wheel system appears in [29]. The CWlog system appears in [37, 36]. The motivation for several of these alternative systems was to find constructions with high availability and low load (which is referred to in most of these papers as quorum systems with small quorums).

In [19], the question of how evenly balanced the work load can be is studied. Tradeoffs between the potential load balancing of a system and its average load are obtained, and it is shown that in some quorum systems it is impossible to have a perfect load balance, in which all the elements have an equal load.

A concept of capacity in voting systems is defined in [21] and some voting systems are compared. The analysis does not distinguish between properties of the quorum system and properties of the strategy that chooses which quorum to access.

A good reference to percolation theory is [15]. Two successful applications of percolation to questions of computer science are [30] and [8].

While the majority quorum system is the best in terms of availability, and the finite projective planes construction have excellent load, they fail miserably according to the other criteria: the load of majority is 1/2 and the failure probability of the projective planes (FPP) goes to 1 (quickly) as the number of elements grows. In fact, all of the existing constructions are less successful than ours in the simultaneous achievement of high availability and low load.

1.3. New Results. We start by defining the concepts of load and capacity, and showing that they can be formulated as linear or integer linear programs. Then using results of hypergraph theory we show that $Cap(\mathcal{S}) = 1/\mathcal{L}(\mathcal{S})$. Therefore all the information regarding the capacity is captured by $\mathcal{L}(\mathcal{S})$.

We obtain several lower bounds on the load $\mathcal{L}(\mathcal{S})$. We show that if the minimal quorum size is $c(\mathcal{S})$ then $\mathcal{L}(\mathcal{S}) \geq \max\{1/c(\mathcal{S}), c(\mathcal{S})/n\}$, hence $\mathcal{L}(\mathcal{S}) \geq 1/\sqrt{n}$. We also obtain a tradeoff between the load and failure probability, i.e., $F_p \geq p^{n\mathcal{L}(\mathcal{S})}$. In some cases the linear program formulation of load also allows us to efficiently compute the load of a given quorum system, even if the quorums are not represented explicitly, using the Ellipsoid algorithm adaptation of [16]. The behavior of the load when the elements may fail is also studied. We assume the common model that the elements fail independently with probability p. The load then becomes a random variable $\mathcal{L}_p(\mathcal{S})$.

Next we show some conditions that prove that a given strategy w induces the optimal load. This enables us to find optimal strategies and to calculate $\mathcal{L}(\mathcal{S})$ of some quorum systems, without actually solving the linear program.

The major contributions of this work are four novel quorum system construction, all of which have optimal or near optimal load, *and* high availability i.e., a failure probability that tends to 0 exponentially fast when $p < \frac{1}{2}$, or at least when $p < \beta < \frac{1}{2}$. Our best construction is the Paths system, which is based on a percolation grid. It has a load of $O(1/\sqrt{n})$, and a failure probability of $\exp(-\Omega(\sqrt{n}))$ when the elements fail with probability $p < \frac{1}{2}$. Moreover, even in the presence of faults, with exponentially high probability the load of this system is still $O(1/\sqrt{n})$. Two other constructions resemble the Grid construction, but enhanced so their failure probability tends to 0. The B-Grid system has $\mathcal{L}(\text{B-Grid}) = O(1/\sqrt{n})$ and if $p < \frac{1}{3}$ then $F_p(\text{B-Grid}) = O(\exp(-n^{1/4}/2))$. The SC-Grid system has $\mathcal{L}(\text{SC-Grid}) = O(\sqrt{(\ln n)/n})$, and if $p < \frac{1}{2} - \delta$ for some $\delta > 0$ then $F_p(\text{SC-Grid}) \leq \exp(-\Omega(\sqrt{n \ln n}))$. The AndOr system uses the AND/OR trees of [43]. It has $\mathcal{L}(\text{AndOr}) = O(1/\sqrt{n})$, $F_p(\text{AndOr}) \leq \exp(-\Omega(\sqrt{n}))$ when $p < \frac{1}{4}$, and $F_p \leq \exp(-\Omega(n^{0.19}))$ if $p \leq 0.38 - \Omega(n^{-0.19})$. The three latter constructions also enjoy the property that their quorums are all of size $O(\sqrt{n})$.

Finally, we analyze the load of some known quorum system constructions. We show that all voting systems have a load of at least $\frac{1}{2}$, which is very high. We also show that non-dominated coteries have lower load than dominated ones.

The paper is organized in as follows. In Section 2 we present some basic definitions. In Section 3 we define the load and the capacity, their linear programs, and the relationship between them. In Section 4 we prove the basic properties of the load. In Section 5 we present the new constructions. In Section 6 we analyze the load of some quorum systems.

2. Preliminaries.

2.1. Definitions and Notation.

Definition 2.1. *A* Set System $\mathcal{S} = \{S_1, \ldots, S_m\}$ *is a collection of subsets* $S_i \subseteq U$ *of a finite universe* U*. A* Quorum System *is a set system* $\mathcal{S}$ *that has the* Intersection property: $S \cap R \neq \varnothing$ *for all* $S, R \in \mathcal{S}$*.*

Alternatively, quorum systems are known as *intersecting set systems* or as *intersecting hypergraphs*. The sets of the system are called *quorums*. The number of elements in the underlying universe is denoted by $n = |U|$. The number of quorums in the system is denoted by m. The cardinality of the smallest quorum in $\mathcal{S}$ is denoted by $c(\mathcal{S}) = \min\{|S| : S \in \mathcal{S}\}$.

The *degree* of an element $i \in U$ in a quorum system $\mathcal{S}$ is the number of quorums that contain i: $\deg(i) = |\{S \in \mathcal{S} : i \in S\}|$.

Definition 2.2. *Let* $\mathcal{S}$ *be a quorum system.* $\mathcal{S}$ *is* s-uniform *if* $|S| = s$ *for all* $S \in \mathcal{S}$*.*

Definition 2.3. *A quorum system* $\mathcal{S}$ *is* (s, d)-fair *if it is* s*-uniform and* $\deg(i) = d$ *for all* $i \in U$*.* $\mathcal{S}$ *is called* s*-fair if it is* (s, d)*-fair for some* d*.*

Definition 2.4. *A* Coterie *is a quorum system* $\mathcal{S}$ *that has the* Minimality property: *there are no* $S, R \in \mathcal{S}$*,* $S \subset R$*.*

Definition 2.5. *Let* $\mathcal{R}, \mathcal{S}$ *be coteries (over the same universe* U*). Then* $\mathcal{R}$ dominates $\mathcal{S}$*, denoted* $\mathcal{R} \succ \mathcal{S}$*, if* $\mathcal{R} \neq \mathcal{S}$ *and for each* $S \in \mathcal{S}$ *there is* $R \in \mathcal{R}$ *such that* $R \subseteq S$*. A coterie* $\mathcal{S}$ *is called* dominated *if there exists a coterie* $\mathcal{R}$ *such that* $\mathcal{R} \succ \mathcal{S}$*. If no such coterie exists then* $\mathcal{S}$ *is* non-dominated (ND)*. Let* NDC *denote the class of all* ND *coteries.*

2.2. The Probabilistic Failure Model.

We use a simple probabilistic model of the failures in the system. We assume that the elements (processors) fail independently with probabilities p_i. We assume that the failures are *transient*. We assume also that the failures are *crash* failures, and that they are *detectable*.

Definition 2.6. *A* configuration *is a vector* $\mathbf{x} \in \{0,1\}^n$ *in which* $x_i = 1$ *iff the element* $i \in U$ *has* failed*.*

Notation: For a configuration $\mathbf{x}$ let $dead(\mathbf{x}) = \{i \in U : x_i = 1\}$ denote the set of failed elements, and let $live(\mathbf{x}) = \{i \in U : x_i = 0\}$ denote the set of functioning elements.

Notation: We use $q_i = 1 - p_i$ to denote the probability of survival of element i.

Definition 2.7. *For every quorum* $S \in \mathcal{S}$ *let* $\mathcal{E}_S$ *be the event that* S *is* hit*, i.e., at least one element* $i \in S$ *has failed (or,* $x_i = 1$ *for some* $i \in S$*). Let* $fail(\mathcal{S})$ *be the event that all the quorums* $S \in \mathcal{S}$ *are hit, i.e.,* $fail(\mathcal{S}) = \bigcap_{S \in \mathcal{S}} \mathcal{E}_S$*.*

When the failure probabilities are equal, i.e., $\mathbf{p} = (p, \ldots, p)$, we use the definition of [35] of the global system failure probability of a quorum system $\mathcal{S}$, as follows.

Definition 2.8. $F_p(\mathcal{S}) = \mathbb{P}_p(fail(\mathcal{S})) = \mathbb{P}_p\left(\bigcap_{S \in \mathcal{S}} \mathcal{E}_S\right)$.

When we consider the asymptotic behavior of $F_p(\mathcal{S}_n)$ for a sequence $\mathcal{S}_n$ of quorum system over a universe with an increasing size n, we find that for many constructions it is similar to the behavior described by the Condorcet Jury Theorem [6]. Hence, the following definition of [35].

Definition 2.9. *A parameterized family of functions* $g_p(n) : \mathbb{N} \to [0,1]$*, for* $p \in [0,1]$*, is said to be* Condorcet *iff* $\lim_{n \to \infty} g_p(n) = \begin{cases} 0, & p < \frac{1}{2}, \\ 1, & p > \frac{1}{2}, \end{cases}$ *and* $g_{1/2}(n) = \frac{1}{2}$ *for all* n*. If* $g_p(n)$ *has this behavior for* $p \neq \frac{1}{2}$ *but* $g_{1/2}(n) \neq \frac{1}{2}$*, then it is said to be* almost Condorcet*.*

3. Load and Capacity.

3.1. Strategies and Load. A protocol using a quorum system (for mutual exclusion, say) occasionally needs to access quorums during its run. A strategy is a probabilistic rule that governs which quorum is chosen each time. In other words, a strategy gives the frequency of picking each quorum S_j.

DEFINITION 3.1. *Let a quorum system $\mathcal{S} = (S_1, \ldots, S_m)$ be given over a universe U. Then $w \in [0,1]^m$ is a* strategy *for $\mathcal{S}$ if it is a probability distribution over the quorums $S_j \in \mathcal{S}$, i.e., $\sum_{j=1}^{m} w_j = 1$.*

For every element $i \in U$, a strategy w of picking quorums induces the frequency of accessing element i, which we call the *load* on i. The *system load*, $\mathcal{L}(\mathcal{S})$, is the load on the *busiest* element induced by the *best* possible strategy. Formally,

DEFINITION 3.2. *Let a strategy w be given for a quorum system $\mathcal{S} = (S_1, \ldots, S_m)$ over a universe U. For an element $i \in U$, the* load *induced by w on i is $\ell_w(i) = \sum_{S_j \ni i} w_j$. The* load *induced by a strategy w on a quorum system $\mathcal{S}$ is*

$$\mathcal{L}_w(\mathcal{S}) = \max_{i \in U} \ell_w(i).$$

The system load *on a quorum system $\mathcal{S}$ is*

$$\mathcal{L}(\mathcal{S}) = \min_{w}\{\mathcal{L}_w(\mathcal{S})\},$$

where the minimum is taken over all strategies w.

REMARKS:

(i) The load $\mathcal{L}(\mathcal{S})$ should be viewed as a "best case" definition. A load of $\mathcal{L}(\mathcal{S})$ is achieved only if the quorums are chosen according to an optimal strategy. However a protocol using the quorum system may use some other strategy (for instance if computing an optimal strategy is too costly), hence the actual load might be higher than $\mathcal{L}(\mathcal{S})$. This also means that $\mathcal{L}(\mathcal{S})$ is a property inherent to the combinatorial structure of the *quorum system*, and not to the protocol using the system.

(ii) In the definition of $\mathcal{L}(\mathcal{S})$ we are assuming that all the elements of the universe are functioning, so all the quorums of the system are usable. In the sequel the definition is extended to the case where the elements may fail.

3.2. A Linear Programming Formulation of the Load. An alternative way to define the load is via a Linear Programming formulation. This formulation shows that the load $\mathcal{L}(\mathcal{S})$ can be computed in polynomial time using Linear Programming algorithms (cf. [40]) if $\mathcal{S}$ is given explicitly.

DEFINITION 3.3. *Let a quorum system $\mathcal{S} = (S_1, \ldots, S_m)$ be given over a universe U of size n. Define a variable w_j for each quorum $S_j \in \mathcal{S}$ and an additional variable L. Then the* system load *$\mathcal{L}(\mathcal{S})$ is defined by the following linear program.*

$$LP : \mathcal{L}(\mathcal{S}) = \min L, \text{ s.t. } \begin{cases} \sum_{j=1}^{m} w_j = 1, & & \text{(i)} \\ \sum_{S_j \ni i} w_j \le L, & \text{for all } i \in U, & \text{(ii)} \\ w_j \ge 0, \ L \ge 0. & & \text{(iii)} \end{cases}$$

NOTATION: We use $(w; L)$ to denote a tuple of a strategy and a possible load, that together constitute a point in the problem domain $[0,1]^{m+1}$.

REMARK: The program LP is always feasible, since for any quorum system $\mathcal{S}$ and strategy w, the point $(w; 1)$ is trivially feasible. Clearly, LP is also a bounded linear program, so $\mathcal{L}(\mathcal{S})$ is always finite.

The next definition and lemma show that the load $\mathcal{L}(\mathcal{S})$ is closely related to the well known *fractional matching number* of a hypergraph (cf. [12], p. 149).

DEFINITION 3.4. *The fractional matching number of a quorum system, denoted by* ν^**, is*

$$FM : \nu^*(\mathcal{S}) = \max \sum_{j=1}^{m} f_j , \text{ s.t. } \begin{cases} \sum_{S_j \ni i} f_j \le 1, & \text{for all } i \in U, \\ f_j \ge 0. \end{cases}$$

LEMMA 3.5. $\mathcal{L}(\mathcal{S}) = 1/\nu^*(\mathcal{S})$ *for any quorum system* $\mathcal{S}$

Proof. Let w be an optimal strategy for program LP, attaining $\mathcal{L}(\mathcal{S})$. Then f defined by $f_j = w_j/\mathcal{L}(\mathcal{S})$ is feasible in program FM. Since FM is maximizing it follows that $\nu^*(\mathcal{S}) \ge \sum_j f_j = 1/\mathcal{L}(\mathcal{S})$.

On the other hand, consider f which optimizes program FM, with $\sum_j f_j = \nu^*(\mathcal{S})$. Then w defined by $w_j = f_j/\nu^*$ is a strategy (since $\sum_j w_j = 1$), and $(w; 1/\nu^*)$ is feasible for program LP. Since $\mathcal{L}(\mathcal{S})$ is minimal it follows that $\mathcal{L}(\mathcal{S}) \le 1/\nu^*(\mathcal{S})$. □

NOTATION: For a vector $\mathbf{y} \in [0,1]^n$ and a set $S \subseteq U$, let $\mathbf{y}(S) = \sum_{i \in S} y_i$.

FACT 3.6. *Let* $\mathcal{S}$ *be a quorum system as in Definition 3.3. Define a variable* y_i *for each element* $i \in U$*, and an additional variable* T*. The dual of program* LP *is*

$$DLP : t(\mathcal{S}) = \max T, \text{ s.t. } \begin{cases} \sum_{i=1}^{n} y_i \le 1, & & \text{(iv)} \\ \mathbf{y}(S_j) \ge T, & \text{for all } S_j \in \mathcal{S}, & \text{(v)} \\ y_i \ge 0, & & \text{(vi)} \\ T \lesseqgtr 0. & & \text{(vii)} \end{cases}$$

REMARKS:

(i) Formally the variable T is unconstrained (vii). However at the optimum $t(\mathcal{S}) = T$ is positive, since $T = 0$ is feasible for any vector $\mathbf{y} \in [0,1]^n$ and DLP is a maximization problem.

(ii) The value of $t(\mathcal{S})$ does not change if we require equality in (iv), since we can increase the y_i values without violating any inequality in (v) and without changing T.

Using the dual program DLP allows us in some cases to compute $\mathcal{L}(\mathcal{S})$ even when $\mathcal{S}$ is given implicitly, using the Ellipsoid algorithm of [16, 27] (see Section 4.3).

3.3. The Capacity of a Quorum System. Each time that a distributed protocol generates an access to a quorum $S \in \mathcal{S}$, it causes work to be done by the elements of S. During the time that the elements of S are busy with one quorum access, they cannot handle another. However other elements may be used in the next quorum access, making use of the parallelism in the system. We want to find what is the maximal rate of quorum access that the system allows.

Assume that it takes one unit of time for an element to complete the work required for a single quorum access. Now consider a period of k time units, and some schedule of quorum accesses that need to take place during this period. Let the integers r_j count the number of times that each quorum $S_j \in \mathcal{S}$ is accessed, with the total number of accesses being $a = \sum_{S_j \in \mathcal{S}} r_j$. Ignoring the order in which the quorum accesses are scheduled, a necessary condition for the system to handle all a accesses within this period of k time units is that every element $i \in U$ be accessed at most k times. The following definition formalizes this condition using an integer linear program.

DEFINITION 3.7. *The maximum number of quorum accesses which a quorum* *system $\mathcal{S}$ can handle within k units of time is*

$$IP : a(\mathcal{S}, k) = \max \sum_{j=1}^{m} r_j, \text{ s.t. } \begin{cases} \sum_{S_j \ni i} r_j \le k, & \text{for all } i \in U, \\ r_j \ge 0, \\ r_j \in \mathbb{N}. \end{cases}$$

The capacity of the system $\mathcal{S}$ is defined to be the maximal rate at which the system handles quorum accesses. In other words, the capacity is the number of accesses $a(\mathcal{S}, k)$ that the system can handle, normalized by k. Since we are interested in the behavior over long time periods, we let the period k tend to infinity.

DEFINITION 3.8. *The* capacity *of a quorum system $\mathcal{S}$ is*

$$Cap(\mathcal{S}) = \lim_{k \to \infty} \frac{a(\mathcal{S}, k)}{k}.$$

In hypergraph theory the quantity $a(\mathcal{S}, k)$ is known as the *k-matching number* of $\mathcal{S}$, and is usually denoted by ν_k (cf. [12] p. 154). Furthermore, Proposition 5.12 of [12] shows that $\lim_{k \to \infty} \nu_k / k = \nu^*$, hence by the definition of the capacity and Lemma 3.5 we obtain:

COROLLARY 3.9. *Cap(S)= 1/$\mathcal{L}(\mathcal{S})$.* Therefore all the information regarding the capacity is captured by $\mathcal{L}(\mathcal{S})$. In [33] we gave a direct proof of Corollary 3.9 (without using the hypergraph machinery), which indicates how to schedule the quorum accesses so the capacity tends to $1/\mathcal{L}(\mathcal{S})$: select the quorums independently at random using a strategy w which optimizes the load.

3.4. The Load with Failures. In this section we extend our definition of the load to the case where the elements may fail. We use the simple probabilistic failure model of Section 2.2, namely that the elements fail independently with probabilities $\mathbf{p} = (p_1, \ldots, p_n)$.

DEFINITION 3.10. *Let $\mathbf{x} \in \{0,1\}^n$ be the current configuration. Then $\mathcal{S}_{\mathbf{x}}$ is the subcollection of functioning quorums, $\mathcal{S}_{\mathbf{x}} = \{S \in \mathcal{S} : S \subseteq live(\mathbf{x})\}$.*

DEFINITION 3.11. *The load of a quorum system $\mathcal{S}$ over a configuration $\mathbf{x} \in \{0,1\}^n$ is defined as follows. If $\mathcal{S}_{\mathbf{x}} = \varnothing$ then $\mathcal{L}(\mathcal{S}_{\mathbf{x}}) = 1$. If there are functioning quorums, i.e., $\mathcal{S}_{\mathbf{x}} \ne \varnothing$, then*

$$\mathcal{L}(\mathcal{S}_{\mathbf{x}}) = \min L, \text{ s.t. } \begin{cases} \sum_{S_j \in \mathcal{S}_{\mathbf{x}}} w_j = 1, \\ \sum_{\mathcal{S}_{\mathbf{x}} \ni S_j \ni i} w_j \le L, & \text{for all } i \in live(\mathbf{x}), \\ w_j \ge 0, \ L \ge 0. \end{cases}$$

REMARK: When there are no functioning quorums in the current configuration, there is no natural concept of load. We choose to define $\mathcal{L}(\mathcal{S}_{\mathbf{x}}) = 1$ for such a configuration to capture the intuition of a monotonic load; as more elements fail, the load increases. The intuition behind this definition is justified in Proposition 3.16.

DEFINITION 3.12. *Let the elements fail with probabilities $\mathbf{p} = (p_1, \ldots, p_n)$. Then the load is a random variable $\mathcal{L}_{\mathbf{p}}(\mathcal{S})$ defined by*

$$\mathbb{P}\Big(\mathcal{L}_{\mathbf{p}}(\mathcal{S}) = L\Big) = \sum_{\substack{\mathbf{x} \\ \mathcal{L}(\mathcal{S}_{\mathbf{x}}) = L}} \prod_{i \in dead(\mathbf{x})} p_i \prod_{i \in live(\mathbf{x})} q_i.$$

If the probabilities $\mathbf{p} = (p, \ldots, p)$ *are all equal, we denote the random load by* $\mathcal{L}_p(\mathcal{S})$. *Let* $E\mathcal{L}_p(\mathcal{S})$ *denote the expectation of* $\mathcal{L}_p(\mathcal{S})$.

FACT 3.13. *For any quorum system* $\mathcal{S}$, *if the elements never fail then* $E\mathcal{L}_0(\mathcal{S}) = \mathcal{L}(\mathcal{S})$ *and if the elements fail with probability 1 then* $E\mathcal{L}_1(\mathcal{S}) = 1$.

LEMMA 3.14. *Let* $\mathcal{S}$ *be a quorum system. Then* $E\mathcal{L}_p(\mathcal{S}) \geq F_p(\mathcal{S})$ *for any* $p \in [0,1]$.

Proof. By Definition 3.11, in a configuration $\mathbf{x}$ that causes a system failure (i.e., all the quorums are hit) the load is 1. Since $F_p(\mathcal{S})$ is the probability of a system failure, we get

$$E\mathcal{L}_p(\mathcal{S}) = [1 - F_p(\mathcal{S})] \cdot g(\mathcal{S}, p) + F_p(\mathcal{S}) \cdot 1$$

for some $g(\mathcal{S}, p) \geq 0$, and we are done. □

The following examples show that although the FPP quorum system and the Grid system have optimal or near optimal load of $O(1/\sqrt{n})$ when all the elements are functioning (see Example 4.11), this load is not *stable*.

EXAMPLE 3.15. In [35] it is shown that $F_p(\mathrm{FPP}) \underset{n\to\infty}{\longrightarrow} 1$ and $F_p(\mathrm{Grid}) \underset{n\to\infty}{\longrightarrow} 1$ for any $p > 0$. Therefore Lemma 3.14 gives that $E\mathcal{L}_p(\mathcal{S}) \underset{n\to\infty}{\longrightarrow} 1$ for both systems. □

The next proposition shows the correctness of the intuition that if the elements are more fail prone then the load is higher. For the proof we need some notation and two lemmas.

PROPOSITION 3.16. $E\mathcal{L}_p(\mathcal{S})$ *is a monotone non-decreasing function of* $p \in [0,1]$ *for any* $\mathcal{S}$.

NOTATION: For configurations $\mathbf{x}$ and $\mathbf{z}$, denote $\mathbf{x} \geq \mathbf{z}$ if $x_i \geq z_i$ for all $i \in U$.

NOTATION: For a vector $\mathbf{z} = (z_1, \ldots, z_n)$, let $(1_i; \mathbf{z})$ denote the vector $\mathbf{z}$ with a 1 plugged into the i'th coordinate: $(1_i; \mathbf{z}) = (z_1, \ldots, z_{i-1}, 1, z_{i+1}, \ldots, z_n)$, and similarly for $(0_i; \mathbf{z})$.

LEMMA 3.17. *Consider the function* $L(\mathbf{x}) : \{0,1\}^n \mapsto [0,1]$ *defined by* $L(\mathbf{x}) = \mathcal{L}(\mathcal{S}_{\mathbf{x}})$ *for some quorum system* $\mathcal{S}$. *If* $\mathbf{x} \geq \mathbf{z}$ *then* $L(\mathbf{x}) \geq L(\mathbf{z})$.

Proof. If $\mathbf{x} \geq \mathbf{z}$ then every element that is functioning in configuration $\mathbf{x}$ (with $x_i = 0$) is also functioning in $\mathbf{z}$. Therefore $\mathcal{S}_{\mathbf{x}} \subseteq \mathcal{S}_{\mathbf{z}}$. If $\mathcal{S}_{\mathbf{x}} = \varnothing$ then by Definition 3.11 $L(\mathbf{x}) = 1$ and we are done. Otherwise, any strategy w that only uses quorums of $\mathcal{S}_{\mathbf{x}}$ is a valid strategy for $\mathcal{S}_{\mathbf{z}}$ as well, and by the minimality of $\mathcal{L}(\mathcal{S}_{\mathbf{z}})$ the claim follows. □

LEMMA 3.18. *Let* $\mathcal{S}$ *be a quorum system, let the elements fail with probabilities* $\mathbf{p} = (p_1, \ldots, p_n)$ *and let* $L(\mathbf{x}) = \mathcal{L}(\mathcal{S}_{\mathbf{x}})$ *be the load over configuration* $\mathbf{x}$. *Consider the multi-linear function* $h(\mathbf{p}) : [0,1]^n \mapsto [0,1]$ *defined by*

$$h(\mathbf{p}) = \sum_{\mathbf{x} \in \{0,1\}^n} L(\mathbf{x}) \prod_{x_k = 1} p_k \prod_{x_k = 0} q_k = \mathbb{E}[L(\mathbf{x})].$$

Then $\frac{\partial h}{\partial p_i} = h(1_i; \mathbf{p}) - h(0_i; \mathbf{p}) = \mathbb{E}[L(1_i; \mathbf{x}) - L(0_i; \mathbf{x})]$.

Proof. Sum $h(\mathbf{p})$ separately for configurations in which element i is failed ($x_i = 1$) or is functioning ($x_i = 0$).

$$\begin{aligned} h(\mathbf{p}) &= p_i \sum_{\mathbf{x}: x_i = 1} L(\mathbf{x}) \prod_{\substack{x_k = 1 \\ k \neq i}} p_k \prod_{x_k = 0} q_k + q_i \sum_{\mathbf{x}: x_i = 0} L(\mathbf{x}) \prod_{x_k = 1} p_k \prod_{\substack{x_k = 0 \\ k \neq i}} q_k \\ &= p_i h(p_1, \ldots, p_{i-1}, 1, p_{i+1}, \ldots, p_n) + q_i h(p_1, \ldots, p_{i-1}, 0, p_{i+1}, \ldots, p_n) \\ &= p_i h(1_i; \mathbf{p}) + q_i h(0_i; \mathbf{p}). \end{aligned}$$

Taking partial derivatives we get $\frac{\partial h}{\partial p_i} = h(1_i; \mathbf{p}) - h(0_i; \mathbf{p})$. Having element i fail with probability 1 is the same as having a constant 1 in the random configuration $\mathbf{x}$, so $h(1_i; \mathbf{p}) = \mathbb{E}[L(1_i; \mathbf{x})]$. Linearity of the expectation completes the lemma. □

PROOF OF PROPOSITION 3.16: Consider the case where the elements fail with probabilities $\mathbf{p} = (p_1, \ldots, p_n)$, and let $L(\mathbf{x})$ and $h(\mathbf{p})$ be as before. By Lemma 3.17 $L(\mathbf{x})$ is non-decreasing, so $L(1_i; \mathbf{x}) \geq L(0_i; \mathbf{x})$ for every i. Therefore by Lemma 3.18, $\frac{\partial h}{\partial p_i} \geq 0$ as an expectation of non-negative terms, so $h(\mathbf{p})$ is non-decreasing in every coordinate. Plugging $\mathbf{p} = (p, \ldots, p)$ shows that $E\mathcal{L}_p(\mathcal{S}) = h(p, \ldots, p)$ is a non-decreasing function. □

3.5. Other Measures of Load. In order to measure an intuitive notion of "load" of a quorum system, our definition of $\mathcal{L}(\mathcal{S})$ (Definitions 3.2 and 3.3) is not the only one that comes to mind. Here we discuss the shortcomings of some alternatives.

Several authors (e.g., [28, 1] have emphasized the criterion of having small quorums. This is an important parameter since it captures the message complexity of a protocol using the quorum system. However it does not tell us *how to use* the quorums so each element is used as infrequently as possible. Moreover, our lower bounds (Propositions 4.1 and 4.2) show that if the quorum size is small (i.e., $c(\mathcal{S}) < \sqrt{n}$) then decreasing it any further actually increases the load. We therefore argue that when analyzing a quorum system, one should consider *both* its quorum size and load (and of course its availability) since each measures a different aspect of the system's quality. Having a small quorum size does not give us the whole picture.

Looking for systems with small average quorum size can also be misleading. For instance, the average quorum size in the Wheel system [29] is very small (≈ 3) but the load is high: $\mathcal{L}(\text{Wheel}) \approx 1/2$.

Another tempting definition is that of an average load, rather than the maximum, i.e., $AvL(\mathcal{S}) = \min_w \frac{1}{n} \sum_{i \in U} \sum_{S_j \ni i} w_j$, minimizing over strategies w. An equivalent notion is that of the total load, which is the same as the average except for the scaling factor of $1/n$. However by changing the summation order it follows that $AvL(\mathcal{S}) = \min_w \frac{1}{n} \sum_{1 \leq j \leq m} w_j |S_j|$. A strategy that minimizes this expression is the trivial strategy that always uses the smallest quorum $S_{\min}$ (with probability 1), so AvL is an uninteresting measure.

4. Properties of the Load.

4.1. Lower Bounds and a Tradeoff of the Load. In this section we present three lower bounds on the load $\mathcal{L}(\mathcal{S})$, in terms of the smallest quorum cardinality $c(\mathcal{S})$ and the universe size n. Two of these can be found in the hypergraph literature as upper bounds for the fractional matching number ν^*, and we present them here using our terminology. We also show a tradeoff between the availability of a quorum system, quantified by the failure probability F_p, and the load.

PROPOSITION 4.1. ([12], p. 150): $\mathcal{L}(\mathcal{S}) \geq \frac{c(\mathcal{S})}{n}$ *for any quorum system* $\mathcal{S}$. □

PROPOSITION 4.2. $\mathcal{L}(\mathcal{S}) \geq \frac{1}{c(\mathcal{S})}$ *for any quorum system* $\mathcal{S}$.

Proof. Let $S_{\min} \in \mathcal{S}$ be a quorum such that $|S_{\min}| = c(\mathcal{S})$ and let $\mathbf{y}$ be defined by $y_i = \frac{1}{c(\mathcal{S})}$ for $i \in S_{\min}$ and $y_i = 0$ otherwise. Then $(\mathbf{y}; 1/c(\mathcal{S}))$ is feasible for program DLP so the claim follows by the weak duality of linear programming. □

PROPOSITION 4.3. ([2], cf. [12] p. 170): *Let* $m(\mathcal{S})$ *be the number of quorums in* $\mathcal{S}$. *Then*

$$\mathcal{L}(\mathcal{S}) \geq \frac{1}{\sqrt{n}} \sqrt{1 + \frac{c(\mathcal{S}) - 1}{m(\mathcal{S})}} \geq \frac{1}{\sqrt{n}}. \quad \square$$

EXAMPLE 4.4. The following examples show that both Propositions 4.1 and 4.2 give meaningful lower bounds on the load of some quorum systems.

(i) Over an odd-sized universe, all the quorums of the simple majority system Maj are of size $(n+1)/2$, therefore by Proposition 4.1, $\mathcal{L}(\text{Maj}) \geq (n+1)/2n > \frac{1}{2}$.

(ii) In the Tree system [1], the smallest quorums have cardinality $\log(n+1)$. Therefore by Proposition 4.2, $\mathcal{L}(\text{Tree}) \geq 1/\log(n+1)$. □

The following proposition shows a tradeoff between the failure probability and the load.

PROPOSITION 4.5. *$F_p(\mathcal{S}) \geq p^{n\mathcal{L}(\mathcal{S})}$ for any quorum system $\mathcal{S}$ and any $p \in [0,1]$.*

Proof. Consider a quorum $S_{\min}$ with $|S_{\min}| = c(\mathcal{S})$. If all the elements of $S_{\min}$ fail then by the Intersection property the system fails, therefore $F_p(\mathcal{S}) \geq p^{c(\mathcal{S})}$. The claim follows since $c(\mathcal{S}) \leq n\mathcal{L}(\mathcal{S})$ by Proposition 4.1. □

DEFINITION 4.6. *An infinite family of quorum systems $\mathcal{S}_n$ over universes of increasing size n is said to have a* tight tradeoff *if*

$$\mathcal{L}(\mathcal{S}) \leq C \cdot \frac{-\log F_p(\mathcal{S}_\text{n})}{n}$$

for some constant $C = C(p) > 0$ that depends only on $0 < p < \frac{1}{2}$.

REMARK: It is pointless to consider values of $p \geq \frac{1}{2}$ since in [35] it is proved that $F_p(\mathcal{S}) \geq \frac{1}{2}$ for such p and any quorum system $\mathcal{S}$, so Proposition 4.5 is meaningless asymptotically in this case.

4.2. Conditions for Optimality of the Load. In this section we present several conditions that guarantee the optimality of a strategy w. The first condition, which can be applied to any system $\mathcal{S}$, is an immediate consequence of Linear Programming duality.

PROPOSITION 4.7. *Let a quorum system $\mathcal{S}$ be given, and let w be a strategy for $\mathcal{S}$ with an induced load of $\mathcal{L}_w(\mathcal{S}) = L$. Then L is the optimal load iff there exists $\mathbf{y} \in [0,1]^n$ such that $\mathbf{y}(U) = 1$ and $\mathbf{y}(S) \geq L$ for all $S \in \mathcal{S}$.*

Proof. By the premise, $(w; L)$ is a feasible point of LP, with an objective function value of L. Therefore by duality, L is the optimum iff there exists a feasible point of the dual problem DLP with an objective function value of L as well. By the definition of DLP, this implies that L is optimal iff there exists $\mathbf{y}$ such that $(\mathbf{y}; L)$ is dual-feasible, which is guaranteed by the conditions on $\mathbf{y}$. □

One way to search for a good strategy w is to try to find a *balancing strategy*. We can try to do this by constructing a feasible point $(w; L)$ for the following *balanced* load Linear Program, in which the inequalities (ii) of LP are replaced by equalities (ix).

$$BLP: \begin{cases} \sum_{j=1}^{m} w_j = 1, & & \text{(viii)} \\ \sum_{S_j \ni i} w_j = L, & \text{for all } i \in U, & \text{(ix)} \\ w_j \geq 0,\ L \geq 0. & & \text{(x)} \end{cases}$$

The program BLP is not always feasible, since finding a solution would imply that $\mathcal{S}$ can be *perfectly balanced*, which cannot be done for all quorum systems [19]. Nevertheless, one could hope that such a balancing strategy (if found) would induce the optimal load. The next proposition shows that this is true for a certain subclass of quorum systems.

PROPOSITION 4.8. *Let $\mathcal{S}$ be an s-uniform quorum system. Let w be a strategy and $L \geq 0$ be such that $(w; L)$ is a feasible point to program BLP. Then the optimal load is $\mathcal{L}(\mathcal{S}) = L = s/n$.*

Proof. First let us show that $L = s/n$. Using the equalities (ix) we get

$$\sum_{i \in U} \sum_{S_j \ni i} w_j = nL. \tag{1}$$

By switching the summation order and using the s-uniformity of $\mathcal{S}$ and equality (viii) we get

$$\sum_{i \in U} \sum_{S_j \ni i} w_j = \sum_{j=1}^{m} w_j \sum_{i \in S_j} 1 = s \sum_{j=1}^{m} w_j = s. \tag{2}$$

By equating (1) and (2) we conclude that $L = s/n$.

Now let $\mathbf{y} = (1/n, \ldots, 1/n)$ be a weight vector for the elements. Clearly $\mathbf{y}(U) = 1$, and $\mathbf{y}(S) = |S|/n = s/n = L$ for any quorum $S \in \mathcal{S}$, since $\mathcal{S}$ is s-uniform. Therefore $(\mathbf{y}; L)$ is dual-feasible, so by Proposition 4.7, $\mathcal{L}(\mathcal{S}) = L$. □

REMARK: The proof does not use the fact that $\mathcal{S}$ is a *quorum system* in any way, and holds for non intersecting set systems as well.

The condition that Proposition 4.8 places on a strategy w is a very weak one. It suffices to show that w is a feasible balancing strategy for it to induce the unique optimal load, if $\mathcal{S}$ is uniform. The following example shows that the uniformity is crucial; non-uniform quorum systems can have several balancing strategies, with different induced loads.

EXAMPLE 4.9. Consider the quorum system

$$\begin{aligned} \mathcal{S} = \{ & \{1,4,6\}, \{2,4,7\}, \{3,5,6,7\}, \{1,2,3,5\}, \\ & \{1,2,3,4\}, \{2,3,4,5\}, \{3,4,5,6\}, \{4,5,6,7\}, \{5,6,7,1\}, \{6,7,1,2\}, \{7,1,2,3\} \}. \end{aligned}$$

The strategy $w = (0,0,0,0,\frac{1}{7},\frac{1}{7},\frac{1}{7},\frac{1}{7},\frac{1}{7},\frac{1}{7},\frac{1}{7})$ is balancing with a load of $\mathcal{L}_w(\mathcal{S}) = 4/7$. However the strategy $w' = (\frac{1}{4},\frac{1}{4},\frac{1}{4},\frac{1}{4},0,0,0,0,0,0,0)$ is also balancing, with a load of $\mathcal{L}_{w'}(\mathcal{S}) = 1/2$. □

If $\mathcal{S}$ is a fair system, then the next proposition shows that we can compute the load and optimal strategy immediately. This is a restatement of Proposition 5.1 of [12] using the fact that $\mathcal{L}(\mathcal{S}) = 1/\nu^*$.

PROPOSITION 4.10. *Let $\mathcal{S}$ be an (s,d)-fair quorum system. Then $\mathcal{L}(\mathcal{S}) = s/n = d/m$.*

EXAMPLE 4.11. The following examples demonstrate the use of Proposition 4.10. The first shows that the lower bound of Example 4.4 is tight, and the other two show that the optimal load of Proposition 4.3, $1/\sqrt{n}$, can be attained.

(i) Over an odd-sized universe, Maj is an $\frac{n+1}{2}$-fair quorum system, so $\mathcal{L}(\mathrm{Maj}) = \frac{n+1}{2n} \approx \frac{1}{2}$.

(ii) The FPP system [28] is a $(t+1)$-fair quorum system over $n = t^2+t+1$ elements, so $\mathcal{L}(\mathrm{FPP}) = \frac{t+1}{t^2+t+1} \approx \frac{1}{\sqrt{n}}$. In fact equality holds in the tighter lower bound of Proposition 4.3 for this system.

(iii) The Grid system [5] is a $(2h-1)$-fair system over $n = h^2$ elements, so $\mathcal{L}(\mathrm{Grid}) = \frac{2h-1}{h^2} \approx \frac{2}{\sqrt{n}}$. □

4.3. Effective Calculation of the Load. If a quorum system $\mathcal{S}$ is given explicitly, as a list of all m quorums, then program LP of Definition 3.3 can be solved in $poly(n,m)$ time using Linear Programming (cf. [40]). However often $\mathcal{S}$ is given implicitly, say, via some data structure containing the elements coupled with a polynomial-time procedure to produce a quorum on demand. In such a case just writing program

```
Input a point (y; T).
The rows are U_1, ..., U_d.
Q ← ∅; s ← 0
for i = d to 1                        // bottom to top
    r ← Σ_{j∈U_i} y_j
    if r + s < T then
        return U_i ∪ Q                // y(U_i ∪ Q) < T
    else
        j ← argmin_{k∈U_i} {y_k}      // min weight in row i
        s ← s + y_j
        Q ← Q ∪ {j}
end-for
return TRUE                           // (y; T) is dual-feasible
```

FIG. 1. *An oracle for a Crumbling Wall quorum system.*

LP could be an exponential task since typically $m = 2^{\Omega(n)}$. Calculating the load quickly is especially important when failures may occur, since the computation needs to be done repeatedly after each configuration change.

Instead we make use of the adaptation of the Ellipsoid algorithm of [16]. Let d denote the dimension of the problem at hand. The Ellipsoid algorithm uses an oracle, which receives a point $x \in \mathbb{R}^d$ and performs the following action.

(i) If x is a feasible point, then return TRUE.

(ii) Otherwise, return a hyperplane separating x from the feasible region (i.e., return a violated constraint).

Given such an oracle, that works in time τ, the algorithm solves the linear program in time $poly(\tau, d)$.

We achieve nothing by applying this algorithm to problem LP since its dimension is $m + 1$. However we can apply this algorithm to the dual problem DLP, whose dimension is $n + 1$. Translated to our terminology, we need to provide an oracle whose input is a point $(\mathbf{y}; T)$. If $(\mathbf{y}; T)$ is feasible in DLP then the oracle returns TRUE, otherwise it returns a quorum $S \in \mathcal{S}$ such that $\mathbf{y}(S) < T$. If this oracle works in $poly(n)$ time then the algorithm calculates the load in $poly(n)$ time.

REMARK: Solving problem DLP gives us the optimal value of the load, but does *not* find a strategy that induces this load. Just writing down a strategy would cause a time complexity of $\Omega(m)$.

EXAMPLE 4.12. In the systems of the Crumbling Wall class [37] the elements are arranged in rows of different widths, and a quorum is the union of a full row and a representative from each row below the full row. The procedure in Figure 1 is an oracle of the required kind, with a time complexity of $O(n)$. Therefore we can compute the load of any Crumbling Wall using the Ellipsoid algorithm outlined above. □

5. Optimal Load, High Availability Quorum Systems.

5.1. The Paths System. In this system, the elements constitute a type of square grid, and a quorum is the union of two paths, one connecting the left and right sides and one connecting the top and bottom sides. Our analysis shows that $\mathcal{L}(\text{Paths}) = O(\frac{1}{\sqrt{n}})$ and that $F_p(\text{Paths}) \le e^{-\Omega(\sqrt{n})}$ for $p < \frac{1}{2}$, so the tradeoff between the load and failure probability is tight. Moreover, we show that even in the presence of faults, with exponentially high probability the load is still $\mathcal{L}_p(\text{Paths}) = O(\frac{1}{\sqrt{n}})$ for all $p < \frac{1}{2}$, which is the best we can hope for. We also give a simple and efficient algorithm for computing a strategy which induces an almost optimal load when some elements are faulty. The proofs are based on theorems of Percolation Theory (see Appendix A).

DEFINITION 5.1. *Let $G(d)$ be the subgrid of $\mathbb{Z}^2$ with vertex set $\{v \in \mathbb{Z}^2 : 0 \leq v_1 \leq d+1, 0 \leq v_2 \leq d\}$ and edge set consisting of all edges joining neighboring vertices* except *those joining vertices u, v with either $u_1 = v_1 = 0$ or $u_1 = v_1 = d+1$.*

DEFINITION 5.2. *Let $G^*(d)$, the dual of $G(d)$ be the subgrid with vertex set $\{v + (\frac{1}{2}, \frac{1}{2}) : 0 \leq v_1 \leq d, -1 \leq v_2 \leq d\}$ and edge set consisting of all edges joining neighboring vertices* except *those joining vertices u, v with either $u_2 = v_2 = -\frac{1}{2}$ or $u_2 = v_2 = d + \frac{1}{2}$.*

See Figure 2 for a drawing of $G(d)$ and $G^*(d)$. Note that every edge $e \in G(d)$ has a dual edge $e^* \in G^*(d)$ which *crosses* it. We call such e and e^* a *dual pair of edges.* Note also that $G(d)$ and $G^*(d)$ are isomorphic; $G^*(d)$ may be obtained by rotating $G(d)$ at a right angle around the origin and relocating the vertex labeled $(0,0)$ to the point $(d + \frac{1}{2}, -\frac{1}{2})$. Both $G(d)$ and $G^*(d)$ contain $d^2 + (d+1)^2 = 2d^2 + 2d + 1$ edges.

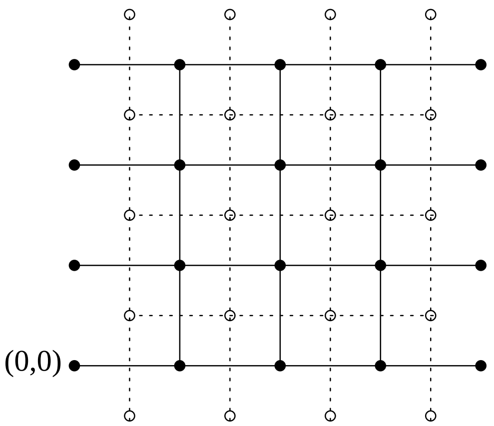

FIG. 2. *The grids $G(3)$ (full lines) and $G^*(3)$ (dotted lines).*

DEFINITION 5.3. *The* Paths *quorum system of order d has $n = 2d^2 + 2d + 1$ elements, and we identify an element in U with a dual pair of edges $e \in G(d)$ and $e^* \in G^*(d)$. A quorum in the system is the union of (elements identified with) the edges of a left-right path in $G(d)$ and the edges of a top-bottom path in $G^*(d)$.*

PROPOSITION 5.4. $\frac{\sqrt{2}}{\sqrt{n}} \lessapprox \mathcal{L}(\text{Paths}) \lessapprox \frac{2\sqrt{2}}{\sqrt{n}}$.

Proof. For the lower bound, note that the smallest quorum has size $c(\text{Paths}) = 2d+1$, and we can apply Proposition 4.1 to get $\mathcal{L}(\text{Paths}) \geq \frac{2d+1}{2d^2+2d+1}$. For the upper bound, consider the quorums of the type $S_j = \{\text{edges joining } u, v \in G(d) : u_2 = v_2 = j\} \cup \{\text{edges joining } u, v \in G^*(d) : u_1 = v_1 = j + \frac{1}{2}\}$, for $j = 0, \ldots, d$. Each element corresponding to a horizontal edge in $G(d)$ appears in two such quorums, except elements on the diagonal that appear only once. A strategy choosing only these quorums with probability $\frac{1}{d+1}$ induces a load of $\frac{2}{d+1}$. □

We now wish to calculate the failure probability of the Paths system. We assume that the elements fail with probability p. A failed element corresponds to *two* closed percolation edges: an edge $e \in G(d)$ and its dual edge $e^* \in G^*(d)$. We say that a *path* *in $G(d)$ is closed* if *all* its edges are closed. Define the following events:

(i) $LR =$ "there exists an open left-right path in $G(d)$",

(ii) $LRC =$ "there exists a closed left-right path in $G(d)$",

(iii) $TB =$ "there exists an open top-bottom path in $G^*(d)$",

(iv) $TBC =$ "there exists an closed top-bottom path in $G^*(d)$".

LEMMA 5.5. *If $p > \frac{1}{2}$ then there exists a positive function φ such that $\mathbb{P}_p(LR) \leq e^{-\varphi(p)d}$.*

Proof. Consider the grid $G(d)$, and let $\lambda = \{v \in \mathbb{Z}^2 : v_1 = d+1\}$ be the set of $\mathbb{Z}^2$ vertices on the infinite vertical line on the right side of $G(d)$. Let R denote the vertices on the right side of $G(d)$. Then summing along the possible starting points on the left side,

$$\mathbb{P}_p(LR) \leq \sum_{k=0}^{d} \mathbb{P}_p((0,k) \leftrightarrow R) \leq \sum_{k=0}^{d} \mathbb{P}_p((0,k) \leftrightarrow \lambda) = (d+1)\mathbb{P}_p(0 \leftrightarrow \lambda).$$

A path from the origin to λ must exit the ball $B(d)$, so we can apply Theorem A.1 to get

$$\leq (d+1)\mathbb{P}_p(0 \leftrightarrow \partial B(d)) \leq (d+1)e^{-\psi(p)d} \leq e^{-\varphi(p)d}. \qquad \square$$

COROLLARY 5.6. *If $q > \frac{1}{2}$ ($p < \frac{1}{2}$) then there exists a positive function φ such that $\mathbb{P}_p(LRC) \leq e^{-\varphi(q)d}$.*

Proof. Exchanging the roles of p and q we get that $\mathbb{P}_p(LRC) = \mathbb{P}_q(LR)$, so we can apply Lemma 5.5. $\square$

PROPOSITION 5.7. *There exists a positive function φ such that $F_p(\text{Paths})$ obeys*

$$\begin{cases} F_p(\text{Paths}) \leq 2e^{-\varphi(q)d}, & p < \frac{1}{2}, \\ F_p(\text{Paths}) \geq 1 - e^{-\varphi(p)d}, & p > \frac{1}{2}, \\ \frac{1}{2} < F_p(\text{Paths}) \leq \frac{3}{4}, & p = \frac{1}{2}, \end{cases}$$

so $F_p(\text{Paths})$ is almost Condorcet.

Proof. By definition, the event "there is a live quorum" is $LR \cap TB$. A moment's reflection shows that an open left-right path exists in $G(d)$ iff no closed top-bottom path exists in $G^*(d)$, since a dual pair of edges e and e^* have the same state (see discussion in [15], pp. 198–199). Therefore the events LR and TBC are complementary. Since $G(d)$ and $G^*(d)$ are isomorphic, then TB and LRC are also complementary events. Therefore the system failure event is

$$\mathit{fail} = \overline{LR \cap TB} = TBC \cup LRC.$$

Additionally, the isomorphism between $G(d)$ and $G^*(d)$ implies that $\mathbb{P}_p(LR) = \mathbb{P}_p(TB)$ and $\mathbb{P}_p(LRC) = \mathbb{P}_p(TBC)$. Now we consider the three cases, as follows.

(i) Let $p < \frac{1}{2}$. Then $F_p = \mathbb{P}_p(\mathit{fail}) = \mathbb{P}_p(LRC \cup TBC) \leq 2\mathbb{P}_p(LRC)$ and so $F_p(\text{Paths}) \leq 2e^{-\varphi(q)d}$ by Corollary 5.6.

(ii) Let $p > \frac{1}{2}$. Then $1 - F_p = \mathbb{P}_p(LR \cap TB) \leq \mathbb{P}_p(LR) \leq e^{-\varphi(p)d}$ by Lemma 5.5.

(iii) Let $p = \frac{1}{2}$. From the above discussion and the proof of Corollary 5.6 it follows that $\mathbb{P}_{1/2}(LR) = \mathbb{P}_{1/2}(TB) = \frac{1}{2}$. For the upper bound, note that both LR and TB are increasing events, so we can use the FKG inequality [10]. Therefore

$$F_{1/2} = 1 - \mathbb{P}_{1/2}(LR \cap TB) \leq 1 - \mathbb{P}_{1/2}(LR)\mathbb{P}_{1/2}(TB) = \frac{3}{4}.$$

For the lower bound, note that Paths is a *dominated* quorum system. Therefore $F_{1/2}(\text{Paths}) > \frac{1}{2}$ by a result of [35]. $\square$

Finally, we show that w.h.p. the load of the Paths system is $O(\frac{1}{\sqrt{n}})$ in the presence of failures, for any failure probability $p < \frac{1}{2}$. In other words, the load has essentially the same asymptotic behavior, as long as there is a good probability that at least one functioning quorum exists.

PROPOSITION 5.8. *For any $0 \le p < \frac{1}{2}$ there exists $\gamma > 0$ such that $\mathcal{L}_p(\text{Paths}) = O(\frac{1}{\sqrt{n}})$ with probability $\ge 1 - e^{-\gamma d}$.*

Proof. Let LR_r be the event "there exist at least $r+1$ edge disjoint left-right paths in $G(d)$". Fix some $\frac{1}{2} > p' > p$. Then by Theorem A.3

$$1 - \mathbb{P}_p(LR_r) \le \left(\frac{q}{q-q'}\right)^r [1 - \mathbb{P}_{p'}(LR)].$$

Now for $p' < \frac{1}{2}$,

$$\mathbb{P}_{p'}(LR) = 1 - \mathbb{P}_{p'}(TBC) = 1 - \mathbb{P}_{p'}(LRC) \ge 1 - e^{-\varphi(q')d}$$

by Corollary 5.6, so

$$\mathbb{P}_p(LR_r) \ge 1 - \left(\frac{q}{q-q'}\right)^r e^{-\varphi(q')d}.$$

Fix $0 < \gamma < \varphi$, let $0 < \beta = \frac{\varphi - \gamma}{\ln[q/(q-q')]}$, and let $r = \beta d$. Then $\mathbb{P}_p(LR_r) \ge 1 - e^{-\gamma d}$. In other words, w.h.p. there exist $\beta d + 1$ edge disjoint left-right paths in $G(d)$. The same also happens for top-bottom paths in $G^*(d)$. Therefore we can find $\beta d + 1$ quorums such that any element appears in at most two of them (once as an edge $e \in G(d)$ and once as the dual edge). We conclude that when such quorums are found, $\mathcal{L}_p(\text{Paths}) \le \frac{2}{\beta d + 1} = O(\frac{1}{\sqrt{n}})$. □

REMARK: This is the strongest possible result regarding load with failures, since if $p \ge \frac{1}{2}$ then by Lemma 3.14 and a result of [35], $E\mathcal{L}_p(\mathcal{S}) \ge F_p(\mathcal{S}) \ge \frac{1}{2}$ for any quorum system $\mathcal{S}$.

Proposition 5.8 guarantees that w.h.p. a good strategy (that induces a load of $O(1/\sqrt{n})$) exists. We now describe an efficient algorithm that finds a nearly optimal strategy w for any given configuration $\mathbf{x}$; the load induced by w is at most twice the optimal load under configuration $\mathbf{x}$, $\mathcal{L}(\text{Paths}_{\mathbf{x}})$.

The algorithm mimics the structure of the existence proof. As a preprocessing step, that needs to be performed after each configuration change, the algorithm finds a maximum collection of disjoint left-right paths, say k_{LR} such paths, and similarly finds k_{TB} disjoint top-bottom paths. This can be done by connecting a source vertex s to all the vertices on the left side and a sink t to the vertices on the right, assigning a capacity of 1 to all the edges, and finding the maximum (s,t) flow (and repeating for TB paths). Since the network is planar we can find the flow in time $O(n \log n)$ using the algorithm of [20], or in time $O(n\sqrt{\log n})$ by [17] using the methods of [11].

Given these path collections, the strategy w is the following: If either $k_{LR} = 0$ or $k_{TB} = 0$ then no live quorums exist in configuration $\mathbf{x}$. Otherwise, whenever a quorum is needed, pick a LR path with uniform probability $1/k_{LR}$ and a TB path with uniform probability $1/k_{TB}$, and use their union. Since the paths are disjoint, each element can appear at most once in a LR path and once in a TB path, so

$$\mathcal{L}_w(\text{Paths}_{\mathbf{x}}) \le 1/k_{LR} + 1/k_{TB}.$$

However, if the maximum flow is k_{LR} then the Max-Flow Min-Cut theorem implies the existence of a k_{LR}-size cut. Therefore any LR path that is open in configuration $\mathbf{x}$ must cross this cut via an edge, so some edge in this cut must have a load of at least $1/k_{LR}$ under any strategy. This implies a lower bound on the load,

$$\mathcal{L}(\text{Paths}_{\mathbf{x}}) \geq \max\{1/k_{LR}, 1/k_{TB}\},$$

hence $\mathcal{L}_w(\text{Paths}_{\mathbf{x}})$ is at most twice the best possible.

REMARK: A related construction, using paths on a triangular lattice with elements corresponding to the nodes, was suggested in [45] (see [44]). They show that their construction has asymptotically high availability ($F_p \to 0$ when $p < \frac{1}{2}$ in our notation). The rate of convergence is not analyzed, and neither is the load (with or without failures). Nevertheless it seems that an analysis similar to ours would show that the characteristics of their system are comparable to those of our Paths system, with a load of $O(1/\sqrt{n})$ and $F_p \leq e^{-\Omega(\sqrt{n})}$ when $p < \frac{1}{2}$.

5.2. The B-Grid **System.** Arrange the elements in a rectangular grid of width d. Split the grid logically into h *bands* of r rows each (so there are $n = dhr$ elements). Call r elements in a column that are all contained in a single band a *mini-column*. Then a quorum consists of one mini-column in every band, and a representative element in each mini-column of one band (see Figure 3).

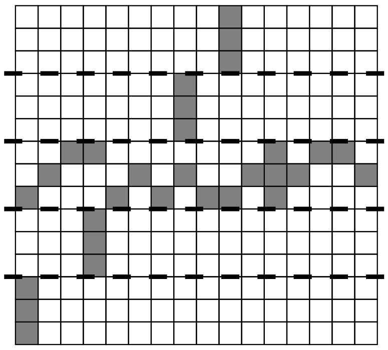

FIG. 3. *The* B-Grid *system over $n = 240$ elements with width $d = 16$, $h = 5$ bands and $r = 3$ rows per band. One quorum is shaded.*

LEMMA 5.9. $\mathcal{L}(\text{B-Grid}) = \frac{d+hr-1}{dhr}$.

Proof. Clearly B-Grid is a fair quorum system, with a quorum size of $d + hr - 1$, and the lemma follows from Proposition 4.10. ☐

LEMMA 5.10. $F_p(\text{B-Grid}) \leq \left(dp^r\right)^h + h\left(1 - q^r\right)^d$.

Proof. Define $\mathcal{E}_1$ to be the event "in every band there exists a mini-column whose elements all failed", and $\mathcal{E}_2$ to be the event "there exists a band in which every mini-column contains a failed element". Clearly the system failure event is $fail = \mathcal{E}_1 \cup \mathcal{E}_2$, so $F_p(\text{B-Grid}) \leq \mathbb{P}(\mathcal{E}_1) + \mathbb{P}(\mathcal{E}_2)$. We get the result since $\mathbb{P}(\mathcal{E}_1) \leq \left(dp^r\right)^h$ and $\mathbb{P}(\mathcal{E}_2) \leq h\left(1 - q^r\right)^d$. ☐

In the next lemma we give a condition on r under which F_p decays exponentially fast in a large range of p values.

LEMMA 5.11. *If $0 \leq p \leq \frac{1}{3}$ and $r = \lfloor \ln d \rfloor$ then $F_p(\text{B-Grid}) \leq e^{-h} + e^{-\frac{1}{2}\sqrt{d}}$ for large values of d such that $\ln h < \frac{1}{2}\sqrt{d}$.*

Proof. To get $\mathbb{P}(\mathcal{E}_1) \le \big(dp^r\big)^h \le e^{-h}$ we require the condition

$$r > \frac{\ln d + 1}{\ln 1/p} \ . \tag{3}$$

To get $\mathbb{P}(\mathcal{E}_2) \le h\big(1 - q^r\big)^d \le e^{-\frac{1}{2}\sqrt{d}}$ we require the condition

$$r < \frac{\ln d - \ln(\ln h + \frac{1}{2}\sqrt{d})}{\ln 1/q} \ . \tag{4}$$

If we consider only $p \le \frac{1}{3}$, then $\frac{1}{\ln 1/p} \le 0.91$ and $\frac{1}{\ln 1/q} \ge 2.466$, and a simple check shows that $r = \lfloor \ln d \rfloor$ fills both conditions (3) and (4) for sufficiently large d if $\ln h < \frac{1}{2}\sqrt{d}$. $\square$

The next propositions are proved by plugging the parameters into Lemma 5.11 and Lemma 5.9. In Proposition 5.12 the failure probability is minimal for the B-Grid system (up to a logarithmic factor in the exponent). In Proposition 5.13 the load is minimal.

PROPOSITION 5.12. *If $d = n^{2/3}$, $r = \lfloor \ln d \rfloor$ and $h = n/(rd)$ then $\mathcal{L}(\text{B-Grid}) = O(n^{-1/3})$ and $F_p(\text{B-Grid}) = O(\exp(-\frac{n^{1/3}}{\ln n}))$ in the range $0 \le p \le \frac{1}{3}$.*

PROPOSITION 5.13. *If $d = \sqrt{n}$, $r = \lfloor \ln d \rfloor$ and $h = n/(rd)$ then $\mathcal{L}(\text{B-Grid}) = O(1/\sqrt{n})$ and $F_p(\text{B-Grid}) = O(\exp(-\frac{n^{1/4}}{2}))$ in the range $0 \le p \le \frac{1}{3}$.*

REMARK: Taking either $d > n^{2/3}$ or $d < \sqrt{n}$ makes both the load and the availability worse. Note that in any case, the tradeoff between the load and failure probability is *not* tight. By Proposition 4.5 we could hope for a failure probability of $O(\exp(-n^{2/3}))$ when the load is $\approx n^{-1/3}$.

5.3. The SC-Grid **System.** Consider a grid made of h rows of cells with width d. In a universe of size $n = dh$, allocate d different elements to each row. Assume that row j is allocated elements $\{1, \dots, d\}$. Then for a parameter $r < d$, place the elements into cells in shifted cyclic order: $\{1, \dots, r\}$ in cell $(1, j)$, $\{2, \dots, r+1\}$ in cell $(2, j)$ and so forth. Every element appears in r cells in the same row. A quorum in the system is a set of elements that are a majority in one cell of every row and a majority in every cell of one row (see Figure 4). This system is somewhat similar to that of [38], in which each grid cell contains a *distinct* set of elements. For simplicity assume that both d and r are odd.

AbC	bCD	CDE	DEA	EAb
fGh	GhI	hIj	Ijf	jfG
Klm	lmN	mNo	NoK	oKl

FIG. 4. *The* SC-Grid *system over $n = 15$ elements with width $d = 5$, $h = 3$ rows and $r = 3$ elements per cell. The elements of one quorum are marked by capitalized letters, and the cells where a majority is achieved are shaded.*

LEMMA 5.14. *Let r be odd and let $d > r$. Consider a cycle of d elements, and the d subsets $C_1, \dots, C_d$ of r consecutive elements along the cycle. Color G of the* *elements in green, and let g_j count the number of green elements in C_j. If $g_j \geq \frac{r+1}{2}$ for all j then $G \geq \lceil d \cdot \frac{r+1}{2r} \rceil$. If $d \mid r$ then the bound can be achieved.*

Proof. Sum the number of green elements in each C_j. Then $\sum_{j=1}^{d} g_j = rG$ since every green element is counted precisely r times. Since $g_j \geq \frac{r+1}{2}$ then $rG \geq d \cdot \frac{r+1}{2}$ and we are done.

If $d = rx$ for some integer x then consider the x disjoint sets $C_\ell = \{(\ell-1)r + 1, \ldots, \ell r\}$ for $1 \leq \ell \leq x$. In each set color the first $\frac{r+1}{2}$ elements in green. Then *every* set C_j contains $\frac{r+1}{2}$ green elements and $G = x \cdot \frac{r+1}{2}$, so the lower bound is achieved. □

LEMMA 5.15. $\frac{rh+d}{2n} \lessapprox \mathcal{L}(\text{SC-Grid}) \lessapprox \frac{r}{d}$.

Proof. By Lemma 5.14 the smallest quorum size is $c(\text{SC-Grid}) \geq \frac{r+1}{2}(h-1) + \frac{d+1}{2}$, so the lower bound follows from Proposition 4.1. For the upper bound, consider the quorums $S_1, \ldots, S_{k+1}$, where S_j contains all the d elements of row j, and $\frac{r+1}{2}$ elements of every row $i \neq j$. Consider a specific row j. As long as $k\frac{r+1}{2} \leq d$ we can use a different set of $\frac{r+1}{2}$ elements from row j in quorum S_i for $i \neq j$, so every element appears in at most 2 quorums. Therefore we can take $k = \lfloor \frac{2d}{r+1} \rfloor$. A strategy that chooses one of these quorums with equal probability induces a load of $\frac{2}{k+1} \approx \frac{r}{d}$. □

NOTATION: Let f_x be the probability that at least $\frac{x+1}{2}$ elements fail out of x when each element fails independently with probability p.

LEMMA 5.16. *$F_p(\text{SC-Grid}) \leq (df_r)^h + hf_d$.*

Proof. Call a cell *failed* if a majority of its elements fail. Let $\mathcal{E}_1$ be the event "all the rows contain at least one failed cell", and let $\mathcal{E}_2$ be the event "there exists a row in which all the cells failed". Then $F_p(\text{SC-Grid}) \leq \mathbb{P}(\mathcal{E}_1) + \mathbb{P}(\mathcal{E}_2)$. Clearly $\mathbb{P}(\mathcal{E}_1) \leq (df_r)^h$. By Lemma 5.14, if all the cells in row j have failed then at least $\frac{d+1}{2}$ of the elements in row j have failed, so $\mathbb{P}(\mathcal{E}_2) \leq hf_d$. □

LEMMA 5.17. *For every $\delta < \frac{1}{2}$ there exists $\varepsilon > 0$ such that when $0 \leq p \leq \frac{1}{2} - \delta$ and $r \geq \frac{2}{\varepsilon} \ln d$ then*

$$F_p(\text{SC-Grid}) \leq d^{-h} + he^{-\varepsilon d}.$$

Proof. By a Chernoff inequality, there exists $\varepsilon > 0$ such that $f_x \leq e^{-\varepsilon x}$ for all x when $0 \leq p \leq \frac{1}{2} - \delta$. For this ε, if $r \geq \frac{2}{\varepsilon} \ln d$ then $f_r \leq 1/d^2$. Plugging this into Lemma 5.16 finishes the lemma. □

By plugging the parameter values into Lemma 5.15 and 5.17 we obtain the following result.

PROPOSITION 5.18. *For every $\delta < \frac{1}{2}$ there exists $\varepsilon > 0$ such that if $0 \leq p \leq \frac{1}{2} - \delta$, then taking $r = \lceil \frac{2}{\varepsilon} \ln d \rceil$, $d = \sqrt{n \ln n}$ and $h = n/d$ gives $F_p(\text{SC-Grid}) = \exp(-\Omega(\sqrt{n \ln n}))$ and $\mathcal{L}(\text{SC-Grid}) = O(\sqrt{\frac{\ln n}{n}})$.*

REMARK: The parameters were chosen to minimize the failure probability. The tradeoff between the load and failure probability is tight for this construction.

5.4. The AndOr System. Consider a complete rooted binary tree of height h, rooted at *root*, and identify the $n = 2^h$ leaves of the tree with the system elements. We define two recursive procedures, that operate on a subtree rooted at v and return a set of elements.

(i) For a leaf v, $ANDset(v) = ORset(v) = \{v\}$.
(ii) $ANDset(v) = ORset(v.left) \cup ORset(v.right)$.
(iii) $ORset(v)$ has a choice; it can be either $ANDset(v.left)$ or $ANDset(v.right)$.

A quorum in the AndOr system is any set $Q = S \cup R$ where S is an $ANDset(root)$ and R is an $ORset(root)$.

It is easy to think of the AndOr system as the conjunction of two boolean functions, corresponding to the top level activations of *ANDset* and *ORset*. Each function is defined by a complete tree of alternating AND and OR gates, over the same inputs, but one function has an AND gate at the root while the other has an OR gate there.

LEMMA 5.19. *If $S = ANDset(root)$ and $R = ORset(root)$, then $|S \cap R| = 1$ for any choices made by the activations of the ORset procedure. Hence* AndOr *is a quorum system.*

Proof. By induction on the tree height h. The case $h = 0$ is obvious. For $h \geq 1$, assume w.l.g. that the *ORset* procedure uses the left subtree. Then any element in the right subtree is not in the intersection, and by the induction hypothesis the intersection in the left subtree has size 1. ☐

LEMMA 5.20. *The* AndOr *system is a fair system, with*

$$c(\text{AndOr}) = \begin{cases} 2\sqrt{n} - 1, & h \text{ even}, \\ 3\sqrt{n/2} - 1, & h \text{ odd}. \end{cases}$$

Proof. The fairness is obvious from symmetry. Let $ANDsize(h) = |ANDset(root)|$ denote the size of the output of the *ANDset* procedure on a tree with height h, and similarly $ORsize(h) = |ORset(root)|$. Then by definition, $ANDsize(0) = ORsize(0) = 1$, and

$$\begin{aligned} ANDsize(h) &= 2\,ORsize(h-1), \\ ORsize(h) &= ANDsize(h-1). \end{aligned}$$

It is easy to show by induction on h that $ORsize(h) = 2^{\lfloor \frac{h}{2} \rfloor}$ and $ANDsize(h) = 2^{\lfloor \frac{h+1}{2} \rfloor}$. Combining with Lemma 5.19 finishes the proof. ☐

PROPOSITION 5.21. $\mathcal{L}(\text{AndOr}) = O(1/\sqrt{n})$.

Proof. Apply Proposition 4.10 using Lemma 5.20. ☐

The following proposition shows the high availability of the AndOr system. The proof is an adaptation of the proof in [43]. We include it here for completeness, omitting some of the technical details.

PROPOSITION 5.22. *Let $\alpha = \frac{3-\sqrt{5}}{2} \approx 0.38$. $F_p(\text{AndOr}) \leq \exp(-\Omega(\sqrt{n}))$ when $p < \frac{1}{4}$ and $F_p \leq \exp(-\Omega(n^{0.19}))$ when $p \leq \alpha - \Omega(n^{-0.19})$.*

Proof. Let $f_A(h)$ denote the probability that all the possible outputs sets of the *ANDset* procedure are hit, and similarly f_O for the *ORset* procedure, on a tree with height h. Clearly $F_p(\text{AndOr}) \leq f_A(h) + f_O(h)$. By the definitions,

$$\begin{aligned} f_A(h) &= 2f_A^2(h-2) - f_A^4(h-2), \\ f_O(h) &= 4f_O^2(h-2) - 4f_O^3(h-2) + f_O^4(h-2), \end{aligned}$$

and $f_A(0) = f_O(0) = p$. Obviously $f_A(h) < 2f_A^2(h-2)$, and also $f_O(h) = f_O^2(h-2)\left(2 - f_O(h-2)\right)^2 < 4f_O^2(h-2)$. Therefore by induction, when h is even,

$$f_A(h) < 2^{2^{\frac{h}{2}}-1} p^{2^{\frac{h}{2}}} < (2p)^{\sqrt{n}}$$

and similarly $f_O(h) < (4p)^{\sqrt{n}}$. So it follows that $F_p(\text{AndOr}) \leq \exp(-\Omega(\sqrt{n}))$ when $p < \frac{1}{4}$. When h is odd the bound is similar.

Now f_O has a stable point at $p = \alpha$ and f_A has a stable point at $p = 1 - \alpha$. As shown by [43], if there are $n = O(d^{5.3})$ leaves in the tree and $p < \alpha - \Omega(d^{-1}) < 1 - \alpha - \Omega(d^{-1})$, then $f_A(h) < 2^{-d-1}$ and the same for f_O. Setting $d = O(n^{1/5.3}) = O(n^{0.19})$ finishes the claim. □

We now describe how to use the AndOr system when some elements have failed. We show an algorithm that finds a nearly optimal strategy w for any given configuration $\mathbf{x}$; the load induced by w is at most twice the optimal load under configuration $\mathbf{x}$, $\mathcal{L}(\text{AndOr}_{\mathbf{x}})$. The description is of an activation at the top level of *ANDset*(*root*), say. The description of the *ORset* activation is identical.

The algorithm is a preprocessing step which needs to be done after each configuration change. It begins by recursively marking the internal nodes in the tree as "alive" or "dead" in the obvious way; an AND node is alive if both its children are alive, and an OR node is alive if at least one child is alive.

Consider a live node v. If it is either an AND node, or an OR node with a single live child, then any strategy that chooses to use (elements in the tree rooted at) v is forced to use all v's live children. Therefore to complete the description of our strategy w need to show what happens at OR nodes with two live children. For this, during the preprocessing each such node v is given a probability $\beta(v)$. If the strategy w decides to use v's tree then it uses its left subtree with probability $\beta(v)$ and its right with probability $1 - \beta(v)$.

To compute the $\beta(v)$ values for live OR nodes v with two live children, the algorithm recursively computes the optimal loads ℓ_L and ℓ_R in the left and right subtrees, respectively. To achieve an optimal load for v's tree, $\beta(v)$ must satisfy $\beta\ell_L = (1-\beta)\ell_R$. Therefore $\beta(v) = \ell_R/(\ell_L + \ell_R)$, and the load induced on v's tree is $\ell_L\ell_R/(\ell_L + \ell_R)$.

The above computation is performed twice, once starting with *ANDset*(*root*) and once with *ORset*(*root*). Note that a node may be marked "alive" w.r.t. the *ANDset*(*root*) activation and "dead" w.r.t. the *ORset*, or vice versa. However every v is assigned a single $\beta(v)$ value since it is an OR node only w.r.t one top level activation.

This w would clearly induce an optimal load for any configuration $\mathbf{x}$ if we were interested in a single top level activation. However since we must activate both *ANDset* and *ORset* at the top level, a moment's reflection shows $\mathcal{L}_w(\text{AndOr}_{\mathbf{x}}) \leq 2\mathcal{L}(\text{AndOr}_{\mathrm{x}})$.

REMARKS:

(i) A quorum system can be constructed from any monotone read-once boolean function in a similar way. This is achieved by taking some AND/OR formula F implementing the function, and making a dual copy if it F^d (in which every AND gate is replaced by an OR gate and vice versa). A quorum is defined to be a union of two sets of elements, one satisfying F and the other satisfying F^d. The proof of Lemma 5.19 would still hold for such a system. However the load and failure probability would depend on the specific structure of the function used.

(ii) The AndOr system is isomorphic to the hierarchical grid construction of [24], when the grids at all the levels are 2×2 grids. The read-quorum and write-quorum procedures of [24] correspond to our top level activations of the *ANDset* and *ORset* procedures, respectively. However ours is a much stronger analysis; we calculate the load, and analyze the rate of decay of F_p and the critical probability α.

6. Load Analyses of some Quorum Systems.

6.1. Non Dominated Coteries Have Lower Load. The following proposition shows that non-dominated coteries (see Definition 2.5) have the lowest loads. This gives further support to the intuitive view that NDC's are preferable to dominated coteries for practical applications.

PROPOSITION 6.1. *Let $\mathcal{S}$, $\mathcal{R}$ be quorum systems over the same universe U such that $\mathcal{R} \succ \mathcal{S}$. Then $\mathcal{L}(\mathcal{R}) \le \mathcal{L}(\mathcal{S})$.*

Proof. Assume that $\mathcal{S} = \{S_1, \ldots, S_m\}$ and $\mathcal{R} = \{R_1, \ldots, R_{m'}\}$. Define a mapping $\varphi : \mathcal{S} \mapsto \mathcal{R}$ as follows. For every set $S_k \in \mathcal{S}$ consider the minimal j such that $R_j \subseteq S_k$, and let $\varphi(S_k) = R_j$. By Definition 2.5 there exists such an R_j for every S_k, so φ is well defined. Let w be an optimal strategy for $\mathcal{S}$. Define w' for $\mathcal{R}$ by

$$w'_j = \begin{cases} \sum_{\varphi(S_k)=R_j} w_k, & \text{if } \exists k : \varphi(S_k) = R_j, \\ 0, & \text{otherwise.} \end{cases}$$

Clearly w' is a strategy for $\mathcal{R}$. The load induced by strategy w' on an element i is

$$\ell_{w'}(i) = \sum_{R_j \ni i} w'_j = \sum_{R_j \ni i} \left(\sum_{\varphi(S_k)=R_j} w_k \right) \le \sum_{S_k \ni i} w_k = \ell_w(i).$$

Applied to the load on the busiest element i this implies that

$$\mathcal{L}_{w'}(\mathcal{R}) \le \mathcal{L}_w(\mathcal{S}) = \mathcal{L}(\mathcal{S}),$$

and by the minimality of $\mathcal{L}(\mathcal{R})$ the result follows. □

REMARK: Proposition 6.1 does not imply that *dominated* quorum systems necessarily have a *high* load. In fact, all our constructions of Section 5 are dominated, and have optimal or near optimal load. By Proposition 6.1 there exist NDC's with loads which are as good or better—but these are more cumbersome to describe explicitly.

6.2. Voting Systems Have High Load. A popular and simple way to construct a quorum system is by *weighted voting* [14, 13, 41, 29]. In this section we show that $\mathcal{L}(\mathcal{S}) > \frac{1}{2}$ for any voting system $\mathcal{S}$, i.e., any voting system is at least as bad as the Maj system in terms of load.

DEFINITION 6.2. *For each $i \in U$ let the integer $v_i \ge 0$ denote the* weight *of i. Let $V = \sum_i v_i$ be the* total weight. *The* voting system *defined by the weights v_i is*

$$\text{Vote} = \{S \subseteq U : \sum_{i \in S} v_i > \frac{V}{2}\}.$$

PROPOSITION 6.3. $\mathcal{L}(\text{Vote}) > \frac{1}{2}$.

Proof. Consider the vector $\mathbf{y}$ defined by $y_i = v_i/V$ for all $i \in U$. Clearly $\mathbf{y}(U) = 1$. By definition 6.2,

$$\mathbf{y}(S) = \frac{1}{V} \sum_{i \in S} v_i > \frac{1}{2},$$

for any quorum $S \in$ Vote. Therefore $(\mathbf{y}; \frac{1}{2})$ is a feasible point to program DLP, so $\mathcal{L}(\text{Vote}) > \frac{1}{2}$ by the weak duality of linear programming. □

6.3. The Tree System. We have shown in Example 4.4 that the load of the Tree quorum system [1] is $\mathcal{L}(\text{Tree}) \ge \frac{1}{\log(n+1)}$. In this section we show that the bound is almost tight; the precise load is $\mathcal{L}(\text{Tree}) = \frac{2}{\log(n+1)+1}$. We first show an upper bound by balancing the load on the elements, and then show a matching lower bound. We use h to denote the height of the tree ($n = 2^{h+1} - 1$).

CLAIM 6.4. $\mathcal{L}(\text{Tree}) \le \frac{2}{h+2}$.

Proof. Denote a tree rooted at node i by $T(i)$, and denote its left and right subtrees by $T_L(i)$ and $T_R(i)$. We build a probabilistic recursive strategy *Pick* to pick a quorum, using values β_h, to be defined later, as follows.

$$Pick(T(i)) = \begin{cases} \{i\} \cup Pick(T_L(i)), & \text{with probability } \beta_h, \\ \{i\} \cup Pick(T_R(i)), & \text{with probability } \beta_h, \\ Pick(T_L(i)) \cup Pick(T_R(i)), & \text{with probability } 1 - 2\beta_h. \end{cases}$$

Let $L(h)$ denote the load induced by strategy *Pick* in a tree of height h. The load is determined either by the load on the root i, or by the most heavily loaded element in one of the subtrees. Therefore $L(h) = \max\{2\beta_h, (1-\beta_h)L(h-1)\}$. Choosing $\beta_h = \frac{L(h-1)}{L(h-1)+2}$ balances the load, so with this choice the load obeys the recurrence

$$L(h) = \frac{2L(h-1)}{L(h-1)+2} ,$$

and $L(0) = 1$. A simple check shows that $L(h) = \frac{2}{h+2}$ solves this recurrence, and then $\beta_h = \frac{1}{h+2}$ for $h \geq 1$. □

CLAIM 6.5. $\mathcal{L}(\text{Tree}) \geq \frac{2}{h+2}$.

Proof. Let $0 \leq t_i \leq h$ denote the distance from node i to the root. To show a matching lower bound we build a dual-feasible weights vector $\mathbf{y}$, defined by

$$y_i = \begin{cases} \frac{1}{h+2}\left(\frac{1}{2}\right)^{t_i}, & 0 \leq t_i < h, \\ \frac{1}{h+2}\left(\frac{1}{2}\right)^{h-1}, & t_i = h. \end{cases}$$

It is easy to see that $\mathbf{y}$ is a valid weight vector. We need to show that $\mathbf{y}(S) \geq \frac{2}{h+2}$ for every quorum $S \in$ Tree.

By induction from the leaves towards the root, one can show that

$$\mathbf{y}(S \cap T(i)) = \begin{cases} \frac{2}{h+2}\left(\frac{1}{2}\right)^{t_i}, & S \cap T(i) \neq \varnothing, \\ 0, & \text{otherwise}, \end{cases} \tag{5}$$

for every $i \in U$ and $S \in$ Tree. Plugging the root of the tree we obtain $\mathbf{y}(S) = \frac{2}{h+2}\left(\frac{1}{2}\right)^0 = \frac{2}{h+2}$. Therefore $(\mathbf{y}; \frac{2}{h+2})$ is feasible for program DLP so the claim follows from the weak duality of linear programming. □

6.4. The Hierarchical Quorum System. In this section we analyze the load and availability of the hierarchical system of [23]. In this system the elements are the leaves of a complete ternary tree. The internal nodes are 2-of-3 majority gates. We show that $F_p(\text{HQS}) \leq \exp(-\Omega(n^{0.63}))$ when $p < \frac{1}{3}$ and $F_p(\text{HQS}) \leq n^{-\alpha(p)}$ when $p < \frac{1}{2}$, and that $\mathcal{L}(\text{HQS}) = n^{-0.37}$.

The analysis is similar in nature to that of the AndOr system. However, HQS is a non-dominated system, so the analysis is good up to $\frac{1}{2}$ rather than up to the 0.38 of the AndOr system. On the other hand the load of HQS is worse than the $O(1/\sqrt{n})$ of the AndOr system.

We use h to denote the height of the tree ($n = 3^h$).

PROPOSITION 6.6. $\mathcal{L}(\text{HQS}) = n^{-0.37}$.

Proof. By symmetry it follows that HQS is a fair system, with $c(\text{HQS}) = n^{\log_3 2} = n^{0.63}$. Therefore by Proposition 4.10, $\mathcal{L}(\text{HQS}) = n^{0.63}/n = n^{-0.37}$. □

PROPOSITION 6.7. *$F_p(\text{HQS}) \leq \exp(-\Omega(n^{0.63}))$ when $p < \frac{1}{3}$ and $F_p(\text{HQS}) \leq n^{-\alpha(p)}$ when $p < \frac{1}{2}$.*

Proof. Let $f(h)$ denote $F_p(\mathrm{HQS})$ on a tree with height h. Then $f(h)$ obeys the recurrence

$$f(h) = 3f^2(h-1) - 2f^3(h-1),$$

and $f(0) = p$. We observe that $p = \frac{1}{2}$ is a stable point, so by a result of [35] it follows that HQS is non-dominated. Now certainly $f(h) < 3f^2(h-1)$, so by induction on h we show that $f(h) < 3^{2^h-1}p^{2^h} < (3p)^{n^{\log_3 2}}$, which proves the case when $p < \frac{1}{3}$.

For larger values of p, we prove by induction on h that $f(h)$ is decreasing when $p < \frac{1}{2}$, and then that

$$f(h) \leq p \cdot (3p - 2p^2)^h.$$

If $p = \frac{1}{2} - \varepsilon$ then $3p - 2p^2 < 1 - \varepsilon$ so $f(h) < \frac{1}{2}(1-\varepsilon)^{\log_3 n} < n^{-\alpha}$ for some $\alpha(p)$. □

REMARK: The HQS system has a tight tradeoff between its availability and load when $p < \frac{1}{3}$.

Acknowledgments. We are grateful to David Peleg for his encouragement, and to Danny Raz for his careful reading of our manuscript. We thank the anonymous referees, whose remarks have helped us improve our presentation, and for bringing [12] and [45] to our attention.

REFERENCES

[1] D. AGRAWAL AND A. EL-ABBADI, *An efficient and fault-tolerant solution for distributed mutual exclusion*, ACM Trans. Comp. Sys., 9 (1991), pp. 1–20.

[2] R. AHARONI, P. ERDŐS, AND N. LINIAL, *Dual integer linear programs and the relationship between their optima*, in Proc. 17th ACM Symp. Theory of Computing, 1985, pp. 476–483.

[3] M. AIZENMAN, J. T. CHAYES, L. CHAYES, J. FRÖHLICH, AND L. RUSSO, *On a sharp transition from area law to perimeter law in a system of random surfaces*, Comm. Mathematical Physics, 92 (1983), pp. 19–69.

[4] D. BARBARA AND H. GARCIA-MOLINA, *The reliability of vote mechanisms*, IEEE Trans. Comput., C-36 (1987), pp. 1197–1208.

[5] S. Y. CHEUNG, M. H. AMMAR, AND M. AHAMAD, *The grid protocol: A high performance scheme for maintaining replicated data*, in Proc. 6th IEEE Int. Conf. Data Engineering, 1990, pp. 438–445.

[6] N. CONDORCET, *Essai sur l'application de l'analyse à la probabilité des decisions rendues à la pluralite des voix.* Paris, 1785.

[7] S. B. DAVIDSON, H. GARCIA-MOLINA, AND D. SKEEN, *Consistency in partitioned networks*, ACM Computing Surveys, 17 (1985), pp. 341–370.

[8] M. DUBINER AND U. ZWICK, *Amplification and percolation*, in Proc. 33rd IEEE Symp. Foundations of Comp. Sci. (FOCS), 1992, pp. 258–267.

[9] P. ERDŐS AND L. LOVÁSZ, *Problems and results on 3-chromatic hypergraphs and some related questions*, in Infinite and Finite Sets, Colloq. Math. Soc. János Bolyai 10, 1975, pp. 609–627.

[10] C. M. FORTUIN, P. W. KASTELEYN, AND J. GINIBRE, *Correlation inequalities on some partially ordered sets*, Comm. Mathematical Physics, 22 (1971), pp. 89–103.

[11] G. N. FREDERICKSON, *Fast algorithms for shortest paths in planar graphs, with applications*, SIAM J. Computing, 16 (1987), pp. 1004–1022.

[12] Z. FÜREDI, *Matchings and covers in hypergraphs*, Graphs and Combinatorics, 4 (1988), pp. 115–206.

[13] H. GARCIA-MOLINA AND D. BARBARA, *How to assign votes in a distributed system*, J. ACM, 32 (1985), pp. 841–860.

[14] D. K. GIFFORD, *Weighted voting for replicated data*, in Proc. 7th Symp. Oper. Sys. Princip., 1979, pp. 150–159.

[15] G. R. GRIMMETT, *Percolation*, Springer-Verlag, 1989.

[16] M. GRÖTSCHEL, L. LOVÁSZ, AND A. SCHRIJVER, *The ellipsoid method and its consequences in combinatorial optimization*, Combinatorica, 1 (1981), pp. 169–197.

[17] R. Hassin, *Maximum flow in (s,t) planar networks*, Inf. Proc. Letters, 13 (1981), p. 107.

[18] M. P. Herlihy, *Replication Methods for Abstract Data Types*, PhD thesis, Massachusetts Institute of Technology, MIT/LCS/TR-319, 1984.

[19] R. Holzman, Y. Marcus, and D. Peleg, *Load balancing in quorum systems*, in Proc. 4th Workshop on Algorithms and Data Structures, Kingston, Ont., Canada, 1995, pp. 38–49. To appear in SIAM J. Discrete Math.

[20] A. Itai and Y. Shiloach, *Maximum flow in planar networks*, SIAM J. Computing, 8 (1979), pp. 135–150.

[21] P. Jalote, S. Rangarajan, and S. K. Tripathi, *Capacity of voting systems*, IEEE Trans. Software Eng., 19 (1993), pp. 698–706.

[22] H. Kesten, *The critical probability of bond percolation on the square lattice equals $\frac{1}{2}$.*, Comm. Mathematical Physics, 71 (1980), pp. 41–59.

[23] A. Kumar, *Hierarchical quorum consensus: A new algorithm for managing replicated data*, IEEE Trans. Comput., 40 (1991), pp. 996–1004.

[24] A. Kumar and S. Y. Cheung, *A high availability $\sqrt{n}$ hierarchical grid algorithm for replicated data*, Inf. Proc. Letters, 40 (1991), pp. 311–316.

[25] A. Kumar, M. Rabinovich, and R. K. Sinha, *A performance study of general grid structures for replicated data*, in Proc. 13th Inter. Conf. Dist. Comp. Sys., 1993, pp. 178–185.

[26] L. Lovász, *Coverings and colorings of hypergraphs*, in Proc. 4th Southeastern Conf. Combinatorics, Graph Theory and Computing, 1973, pp. 3–12.

[27] ———, *An Algorithmic Theory of Numbers, Graphs and Convexity*, SIAM, Philadelphia, 1986.

[28] M. Maekawa, *A $\sqrt{n}$ algorithm for mutual exclusion in decentralized systems*, ACM Trans. Comp. Sys., 3 (1985), pp. 145–159.

[29] Y. Marcus and D. Peleg, *Construction methods for quorum systems*, Tech. Report CS92–33, The Weizmann Institute of Science, Rehovot, Israel, 1992.

[30] T. R. Mathies, *Percolation theory and computing with faulty arrays of processors*, in Proc. 3rd ACM-SIAM Symp. on Discrete Alg., 1992, pp. 100–103.

[31] M. V. Menshikov, *Coincidence of critical points in percolation problems*, Soviet Mathematics Doklady, 33 (1986), pp. 856–859.

[32] S. J. Mullender and P. M. B. Vitányi, *Distributed match-making*, Algorithmica, 3 (1988), pp. 367–391.

[33] M. Naor and A. Wool, *The load, capacity and availability of quorum systems*, in Proc. 35th IEEE Symp. Foundations of Comp. Sci. (FOCS), 1994, pp. 214–225.

[34] ———, *Access control and signatures via quorum secret sharing*, in Proc. 3rd ACM Conf. Comp. and Comm. Security, New Delhi, India, 1996, pp. 157–168. Also available as Theory of Cryptography Library record 96-08, `http://theory.lcs.mit.edu/~tcryptol/1996.html`.

[35] D. Peleg and A. Wool, *The availability of quorum systems*, Information and Computation, 123 (1995), pp. 210–223.

[36] ———, *The availability of crumbling wall quorum systems*, Discrete Applied Math., 74 (1997), pp. 69–83.

[37] ———, *Crumbling walls: A class of practical and efficient quorum systems*, Distributed Computing, 10 (1997), pp. 87–98.

[38] S. Rangarajan and S. K. Tripathi, *A robust distributed mutual exclusion algorithm*, in Proc. 5th Inter. Workshop on Dist. Algorithms (WDAG), LNCS 579, Springer-Verlag, 1991, pp. 295–308.

[39] M. Raynal, *Algorithms for Mutual Exclusion*, MIT press, 1986.

[40] A. Schrijver, *Theory of Linear and Integer Programming*, John Wiley & Sons, Chichester, 1986.

[41] M. Spasojevic and P. Berman, *Voting as the optimal static pessimistic scheme for managing replicated data*, IEEE Trans. Par. Dist. Sys., 5 (1994), pp. 64–73.

[42] R. H. Thomas, *A majority consensus approach to concurrency control for multiple copy databases*, ACM Trans. Database Sys., 4 (1979), pp. 180–209.

[43] L. G. Valiant, *Short monotone formulae for the majority function*, J. Algorithms, 5 (1984), pp. 363–366.

[44] C. Wu, *Replica control protocols that guarantee high availability and low access cost*, Tech. Report 1817, Dept. Computer Science, U. Illinois Urbana-Champaign, Urbana, Illinois, 1993.

[45] C. Wu and G. G. Belford, *The triangular lattice protocol: A highly fault tolerant protocol for replicated data*, in Proc. 11th IEEE Symp. on Reliable Dist. Sys., 1992, pp. 66–73.

[46] T. W. Yan and H. Garcia-Molina, *Distributed selective dissemination of information*, in Proc. 3rd Inter. Conf. Par. Dist. Info. Sys., 1994, pp. 89–98.

A. Results of Percolation Theory. In this section we list the definitions and results that are used in our analysis of the Paths system, following [15].

The percolation model we are interested in is as follows. Let $\mathbb{Z}^2$ be the graph of the square lattice in the plane. Assume that an edge between neighboring vertices in $\mathbb{Z}^2$ is *closed* with probability p and *open* with probability $q = 1 - p$, independently of other edges. This model is known as bond percolation on the square lattice. Another natural model, which plays a minor role in our work, is the site percolation model. In it the *vertices* are closed with probability p. Unless otherwise stated, we always use the bond percolation model.

NOTATION: For an event $\mathcal{E}$ defined in the percolation model (either on $\mathbb{Z}^2$ or on some finite subgraph of $\mathbb{Z}^2$), we denote the probability of $\mathcal{E}$ by $\mathbb{P}_p(\mathcal{E})$.

A key idea in percolation theory is that there exists a *critical probability*, p_c, such that graphs with $p < p_c$ exhibit qualitatively different properties than graphs with $p > p_c$. For example, $\mathbb{Z}^2$ with $p < p_c$ has a single connected (open) component of infinite size. When $p > p_c$ there is no such component (see [15], p. 110). For bond percolation in the plane $p_c = \frac{1}{2}$ [22].

NOTATION: Let $B(d)$ be the ball of radius d with center at the origin; $B(d) = \{v \in \mathbb{Z}^2 : |v_1| + |v_2| \le d\}$. Let $\partial B(d)$ be the surface of $B(d)$, $\partial B(d) = \{v \in \mathbb{Z}^2 : |v_1| + |v_2| = d\}$. For a vertex v and a set of vertices A, let $v \leftrightarrow A$ denote the event that there exists an open path between v and some vertex in A.

The following theorem shows that when the probability p for a closed edge is above the critical probability, the probability of having long open paths decays exponentially fast.

THEOREM A.1. [31] *If $p > \frac{1}{2}$ then there exists $\psi(p) > 0$ such that*

$$\mathbb{P}_p(0 \leftrightarrow \partial B(d)) < e^{-\psi(p)d} \qquad \text{for all } d.$$

DEFINITION A.2. *Let $\mathcal{E}$ be an event defined in the percolation model. Then the* interior *of $\mathcal{E}$ with depth r, denoted $I_r(\mathcal{E})$, is the set of all configurations in $\mathcal{E}$ which are still in $\mathcal{E}$ even if we perturb the states of up to r edges.*

We may think of $I_r(\mathcal{E})$ as the event that $\mathcal{E}$ occurs and is 'stable' with respect to changes in the states of r or fewer edges. The definition is useful to us in the following situation. If LR is the event "there exists an open left-right path in a rectangle D", then by flow considerations it follows that $I_r(LR)$ is the event "there are at least $r+1$ edge disjoint open left-right paths in D".

THEOREM A.3. [3] *Let $\mathcal{E}$ be an increasing event and let r be a positive integer. Then*

$$1 - \mathbb{P}_p(I_r(\mathcal{E})) \le \left(\frac{q}{q - q'}\right)^r [1 - \mathbb{P}_{p'}(\mathcal{E})]$$

whenever $0 \le p < p' \le 1$.

The theorem amounts to the assertion that if $\mathcal{E}$ is likely to occur when the edge failure probability is p', then $I_r(\mathcal{E})$ is likely to occur when the failure probability is smaller than p'.