Billiard Quorums on the Grid*

Divyakant Agrawal *Ömer Eğecioğlu* *Amr El Abbadi*
Department of Computer Science
University of California
Santa Barbara, CA 93106

Abstract

Maekawa considered a simple but suboptimal grid based quorum generation scheme in which N sites in a network are logically organized in the form of a $\sqrt{N} \times \sqrt{N}$ grid, and the quorum sets are row-column pairs. Even though the quorum size $2\sqrt{N}$ of the grid scheme is twice as large as finite projective plane based optimal size quorums, it has the advantage of being simple and geometrically evident. In this paper we construct grid based quorums which use a modified grid, and paths that resemble billiard ball paths instead of horizontal and vertical line segments of rows and columns in the grid scheme. The size of these quorums is $\sqrt{2}\sqrt{N}$. The construction and its properties are geometrically evident as in the case of Maekawa's grid, and the quorum sets can be generated efficiently.

Keywords: Distributed system, mutual exclusion, coterie, quorum.

1 Introduction

One of the most fundamental synchronization problems in distributed systems is the problem of ensuring mutually exclusive accesses by competing processes to a critical resource. Several solutions have been proposed to solve the mutual exclusion problem in distributed systems [Lam78, RA81, CR83, Mae85, SK85, vdS87, Sin88, HPR88, Ray89]. These solutions can be broadly categorized into two types: permission based and token based solutions [San87, Ray91]. The advantage of permission based mutual exclusion algorithms is that they exhibit excellent fault-tolerance and load-balancing characteristics. The general approach underlying this type of algorithms is that a process (or equivalently a site in the network) wishing to enter the critical section obtains permission from other sites in the network. The group of sites which grant permission is referred to as a *quorum* for the requesting site. Any two quorums have a non-empty intersection, thus guaranteeing mutually exclusive access to a critical resource by competing sites.

The main drawback of permission based mutual exclusion algorithms is that the communication cost to enter critical section is directly proportional to the size of quorums. Much research has been done to optimize communication and minimize the size of quorums [Mae85, GB85, AE91, Kum91, CAA92]. Maekawa [Mae85] has shown that under certain conditions (which are strongly desirable in a distributed system), the optimal quorum size is $\sqrt{N}$, where N is the total number of sites in the network. This optimal size is achievable when the sites in the network can be organized logically as a finite projective plane. Finite projective planes can be constructed by considering

*This work was partially supported by NSF under IRI94-11330 and CCR95-05807 and by NASA under NAGW-3888.

certain subspaces of a vector space over a finite field when N is of the form $N = k^2 + k + 1$ where k is a power of a prime number (called the order of the projective plane). In general however, constructing quorums using arbitrary finite projective planes is rather involved.

In order to avoid the complications of finite projective planes, Maekawa suggested a simple alternative in which the sites in the network are logically organized in the form of a grid of $\sqrt{N} \times \sqrt{N}$ sites. The quorum for a site here would correspond to obtaining permissions from all the sites in the corresponding row and column of the requesting site. The size of such a set is $2\sqrt{N}$ (or more precisely $2\sqrt{N} - 1$). It is clear geometrically that any two quorums constructed in this way have a non-empty intersection.

In this paper we develop a solution for grid based generation of quorums for mutual exclusion protocols which brings down the quorum size from $2\sqrt{N}$ to $\sqrt{2}\sqrt{N}$. This construction results in the same suboptimal quorum size as the Triangular system of Lovász [Lov73]. An advantage of our construction is that it extends Maekawa's grid based approach: once the method for generating quorums is given, it is geometrically obvious that they satisfy the properties required of quorum sets. Thus although suboptimal from the theoretical point of view, the $\sqrt{2}\sqrt{N}$ quorum size is achieved without resorting to any tool other than grids and paths, similar to Maekawa's grid based quorum generation.

In the next section we present a model of a distributed system and present the mutual exclusion problem. In Section 3, motivates our quorum generation mechanism, and Section 4 presents the algorithm for forming what we term *billiard quorums*. In Section 5, we present a number properties and examples of these quorums and give an efficient procedure for generating them. Section 6 concludes the paper.

2 The Problem Statement

A distributed system consists of a set of distinct sites that communicate with each other by sending messages over a communication network. We assume that sites are *fail-stop* [SS82] and they can be logically organized to form a structure such as a grid. To have exclusive access to a resource in the network, a site s_i is required to receive permission from some set of sites S_i in the network. If all sites in S_i grant permission to s_i, then s_i is allowed to access the resource. To ensure mutual exclusion, any S_i must have at least one site in common with any other set. This can be stated formally as follows:

The Intersection Property. For any two sets S_i and S_j, $S_i \cap S_j \neq \phi$.

These concepts have been formalized in terms of the notions of *quorums* [Gif79] and *coteries* [GB85]. Maekawa [Mae85] has shown that within a constant factor, the quorum size q is optimal when $q = \sqrt{N}$. Furthermore, in the case of a finite projective plane of N points, each site is assigned a unique quorum S_i, and the conditions of intersection and optimality are also satisfied. Such a plane is known to exist whenever $q - 1$ is a power of a prime. For other values of q, one way to create the sets S_i is by relaxing some of the conditions imposed on the S_i, or by creating sets for a larger N and then discarding some sets. Using such a construction for the sets S_i, a mutual exclusion protocol is proposed in [Mae85]. In addition to the intersection property, which is necessary for mutual exclusion, Maekawa suggested additional properties for quorums to ensure fully distributed solutions and fairness in the workload of the individual sites. One property is referred to as the *equal effort property* and requires all quorums to be of equal size. Another is the *inclusion property* which requires that the quorum being constructed on behalf of a site must include that site. The final property is referred to as the *equal responsibility property* and requires that number of times a site appears in different quorums is the same for all sites in the network. Not all quorum generation algorithms satisfy all of these properties, however.

Although optimal in terms of size, little is know about finite projective planes for orders other than powers of primes. In fact, there are no finite projective planes of order k if either $k-1$ or $k-2$ is divisible by 4 and if k cannot be expressed as the sum of two integral squares [Mae85, AS68]. Maekawa therefore proposed generating quorums using a simple logical grid structure imposed on the sites [Mae85].

3 Motivation

In order to motivate our approach, it will be useful to consider an idealized representation of a square grid in the continuous domain. In this way, the grid is represented by the unit square and a site in the network is represented by a point in the unit square. Given such a point s, Maekawa's approach of a row and a column including s would correspond to a horizontal line segment H together with a vertical line segment V through s as shown in Figure 1. The "quorum" for the point s in the continuous case would thus consist of all the points on the union of the line segments H and V. Similar to the case of the $\sqrt{N} \times \sqrt{N}$ square grid in the continuous case each point s is assigned a distinct quorum, the size (total length) of the quorums is the same ($2\sqrt{1}$ in this case), and any two quorums intersect.

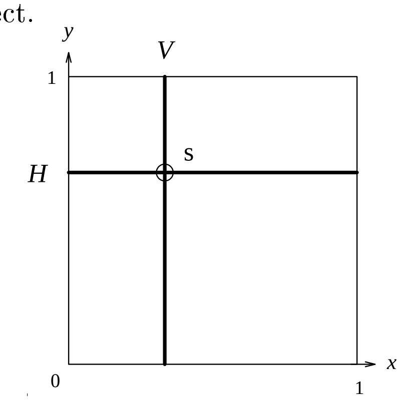

Figure 1: The unit square: an idealized grid

We can try to make other choices for line segments associated to a point that would satisfy the above three properties. For example, instead of choosing horizontal and vertical line segments, we can explore the possibility of using line segments with slopes ± 1.

Given a point s in the unit square, we define the *billiard path* through s as the first $\sqrt{2}$ units of length of the locus of a billiard ball shot directly towards s from the boundary $x = 0$ or $y = 0$ of the unit square along a path parallel to the line $y = x$. Thus a billiard path through s is made up of 2 line segments, one that starts either on the x or the y axis, has slope 1, and which includes s; and a tail end segment of slope -1 which terminates on the boundaries $x = 1$ or $y = 1$ of the unit square. Two such paths are shown in Figure 2 (a) and (b). As a limiting case, the corner points are assigned the appropriate diagonal of the unit square as the billiard path through them. However, two points that lie on a line of slope 1 both define the same billiard path. In order to make the assignment of paths to points unique, we next define a *broken billiard path* through s. This is obtained from the billiard path through s by the rotation of the line segments after s in the billiard path by $90°$ at s. Figure 2 (c) and (d) are broken billiard paths corresponding to the points s and s'. Note that the length of a broken billiard path is also $\sqrt{2}$. Also, note that for each point s above the line $y = x$, the broken billiard path through s has a local vertical peak at s, and for each point s' below the line $y = x$, the path has a local left-to-right peak at s. These peaks serve to distinguish different broken billiard paths.

Our aim is to export the idea of broken billiard paths back to the finite grid to produce quorums, once the sites are logically organized in an appropriate configuration. We can show that any two

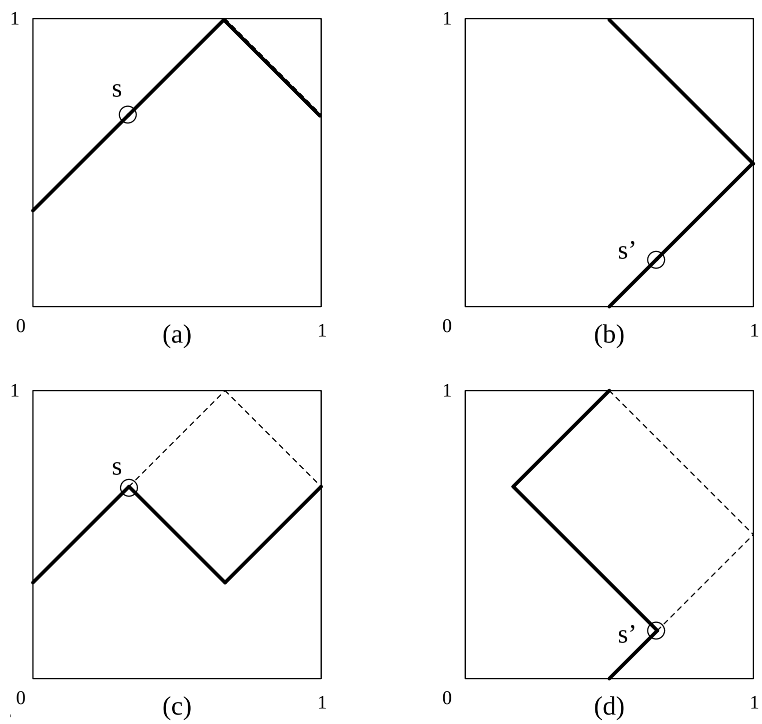

Figure 2: (a), (b): Billiard paths. (c), (d): Broken billiard paths.

distinct broken billiard paths in the continuous case necessarily intersect. However, a number of issues remain to be solved for the discrete case. For example the diagonally opposite corner points $(0,0)$ and $(1,1)$ on the unit square (and the points $(0,1)$ and $(1,0)$) produce the same broken billiard path. Another problem arises when we try to incorporate the notion of broken billiard paths to the case when the unit square is represented by a discretized square grid of $\sqrt{N} \times \sqrt{N}$ cells, each cell corresponding to a site in the network. In particular, now the length of a diagonal is the same as the length of a side, indicating that properties that are evident in the continuous domain may be lost after the discretization process. A more acute problem arises in adapting broken billiard

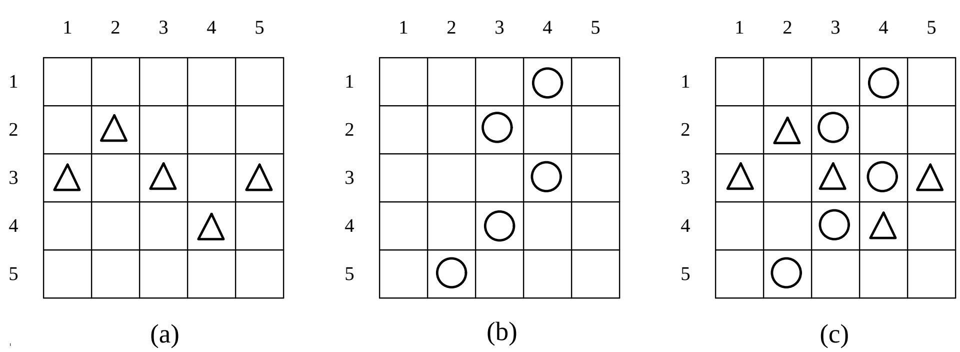

Figure 3: Non-intersecting paths in the discrete case.

paths to a discrete domain. Figures 3(a) and 3(b) show the broken billiard paths of points s and s' in a 5×5 grid. Figure 3(c) illustrates that intersecting paths in the continuous domain become non-intersecting when diagonal adjacencies are allowed in the discrete case.

However, these complications arise only if we insist on the same grid as in Maekawa's construction. A more natural grid structure that supports broken billiard paths is obtained from the ordinary square grid by considering only every other cell in a checkerboard fashion. We describe in detail the construction of quorums of size $\sqrt{2}\sqrt{N}$ on this modified grid next.

4 Broken Billiard Paths on Grids

To eliminate cross-overs of the type shown in Figure 3 in the discrete domain we use only every other cell of the ordinary square grid, leaving the other ones out. This modified grid construction for the $q \times q$ case for $q = 9$ is given in Figure 4. The rows and columns are labeled by $i = 1, 2, \ldots, q$ and $j = 1, 2, \ldots, q$ increasing from top to bottom, and from left to right respectively, as the labeling of entries of a matrix. The assignment of site labels $1, 2, \ldots, N$ to the cells in this configuration is done in row major format. For the example in Figure 4 the number of sites is $N = 40$. They are assigned as follows: in the first row, we discard the first cell, and allocate site 1 to the second

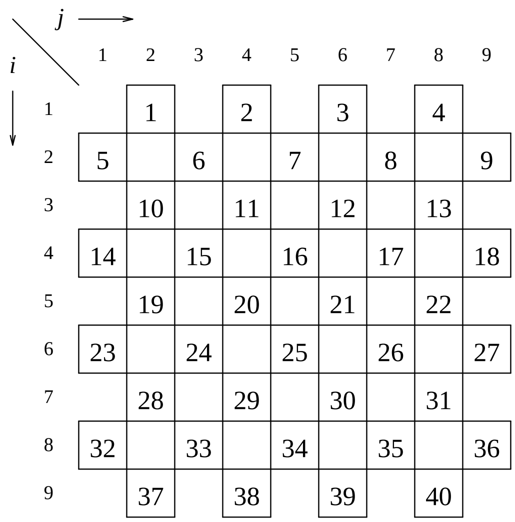

Figure 4: The modified grid in the discrete case for $q = 9$ and $N = 40$.

cell, discard the third, and allocate site 2 to fourth cell, and so on, discarding all odd cells and allocating sites to all even cells of the first row in the grid. In the second row, we allocate a site to every odd cell and leave out the even numbered cells. In general, the cell in position i, j of the ordinary square grid is included in the modified grid if and only if $i + j$ is an odd number.

The quorum construction for a given site in the modified grid follows the rules specified for using line segments that are based on a broken billiard paths of the continuous case. For example, Figure 5(a) shows the quorum set S_{11} for site 11 which occupies position $(i, j) = (3, 4)$. The quorum set produced for this site is $S_{11} = \{11, 15, 16, 18, 19, 21, 22, 23, 26\}$, which is constructed by taking the union $\{23, 19, 15, 11\} \cup \{16, 21, 26\} \cup \{22, 18\}$ corresponding to the sites that lie on three pieces of the broken billiard path at $(3, 4)$. Part (b) of Figure 5 shows the manner of construction of the quorum set S_{34} for site 34, which occupies position $(8, 5)$ on the modified grid. In this case

$$S_{34} = \{38, 34\} \cup \{29, 24, 19\} \cup \{15, 11, 7, 3\} = \{3, 7, 11, 15, 19, 24, 29, 34, 38\} .$$

Note that the quorum size is $9 = \lceil \sqrt{2}\sqrt{40} \rceil$, and the anomalies that appear for the points on the diagonals for the continuous case no longer exist for the modified grid.

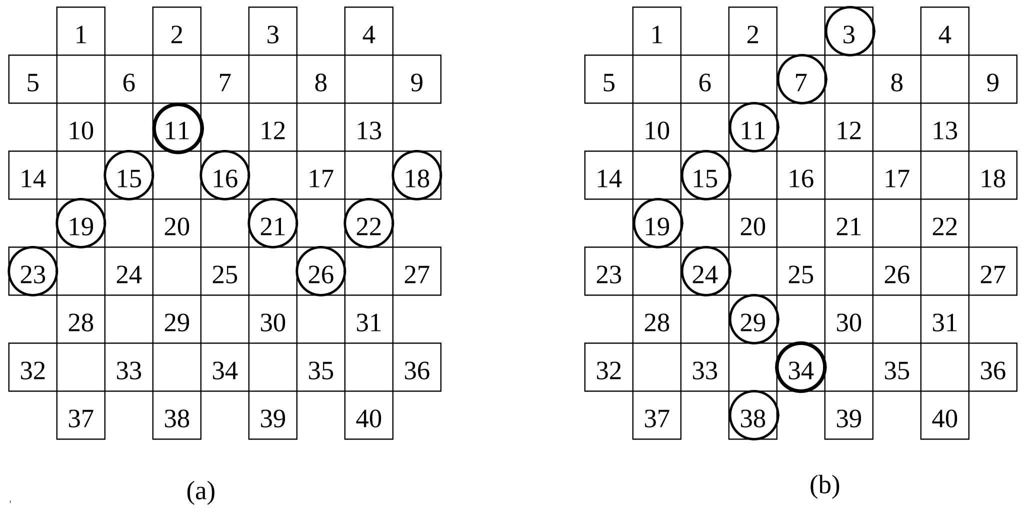

Figure 5: Examples of broken billiard paths on the modified grid: (a) corresponds to $s = 11$, (b) corresponds to $s = 34$.

5 Properties of billiard quorums and examples

Modified grids are obtained from a square grid of size $q \times q$ only when q is an odd integer (q is required to be odd to maintain symmetry). Each broken billiard path in the $q \times q$ modified grid contains q sites. If $q = 2t + 1$, then by counting alternate rows, the corresponding modified grid is seen to contain $N = t(t+1) + (t+1)t = 2t(t+1)$ cells. Therefore the number of sites and the quorum size obtained by using broken billiard paths in the modified grid are related as

$$N = (q^2 - 1)/2,$$

which implies that $q = \lceil \sqrt{2}\sqrt{N}\ \rceil$. The fact that any two of the billiard quorums constructed on the modified grid intersect is geometrically evident. Thus the family of sets generated is mutually non-disjoint.

Proposition 1 *The billiard quorums constructed on the $q \times q$ modified grid satisfy the intersection, uniqueness, equal effort, and the inclusion properties. The quorums are of size $q = \lceil \sqrt{2}\sqrt{N}\ \rceil$.*

Note that the sites near the boundary of the modified grid participate in fewer quorums than sites that are close to the center. In particular billiard quorums do not satisfy the equal responsibility property. Furthermore, as in Maekawa's construction, our development of billiard quorums assigns each site a unique quorum. The resulting coterie is *dominated* [GB85] since one can easily include more quorums to the coterie (e.g., the sites in a row and a column of the modified grid) and still maintain the intersection property. In terms of availability, the billiard quorum approach suffers from the same problem as Maekawa's grid quorums: it can be shown that as the number of sites increases, the probability that a billiard quorum exists asymptotically reaches zero. This will occur in any quorum system unless the number of quorums is at least exponential in their size [PW95]. However, the hierarchical structuring approach proposed by Kumar and Cheung [KC91] for grid quorums can also be adapted to billiard quorums. The hierarchical structuring approach potentially offers asymptotically high availability, while maintaining the quorum sizes.

Next, we consider the assignment of sites to cells in a modified grid, which allows for linear time generation of billiard quorums. When we use the row-major ordering as we have done here (see Figure 4), the following result is immediate

Proposition 2 *Suppose $N = (q^2 - 1)/2$ for some odd integer q. In the row-major assignment of sites $1, 2, \ldots, N$ to the cells of the modified grid as shown in Figure 4,*

1. *The site assigned to cell (i, j) is $n = ((i-1)q + j)/2$ (note that $i + j$ is odd),*

2. *Given site n with $1 \le n \le N$, its location on the modified grid is (i, j) where*

$$j = 2n \pmod{q}\ , \quad \textit{and} \quad i = 1 + (2n - j)/q\ ,$$

where $\pmod{q}$ is the standard modulo function except that it returns q instead of 0.

For example, when $q = 9$ and $N = 40$, the location $(8, 5)$ contains the site $34 = (7 * 9 + 5)/2$. The site $n = 27$ is assigned to cell (i, j) where $j = 54 \pmod{9} = 9$, and $i = 1 + (54 - 9)/9 = 6$.

For efficient generation of billiard quorum sets, we relate the sites assigned to neighboring cells along lines of slope ± 1. Let $R_{i,j}$ denote the site assigned to a generic cell (i, j) of the modified grid. By Proposition 2, $R_{i,j} = ((i-1)q + j)/2$, and we can compute the sites assigned to its 4 immediate neighbors along NE, SE, SW and NW directions. These are found to be

$$\begin{aligned} R_{i-1,j+1} &= R_{i,j} - (q-1)/2, \\ R_{i+1,j+1} &= R_{i,j} + (q+1)/2, \\ R_{i+1,j-1} &= R_{i,j} + (q-1)/2, \\ R_{i-1,j-1} &= R_{i,j} - (q+1)/2. \end{aligned} \tag{1}$$

```
Procedure generate (n):
\* The quorum size q, the number of sites N, and the variables x_1, x_2, ..., x_q are
   external to generate. q is an odd integer and N is of the form N = (q^2 - 1)/2. *\
var i, j, k integer;
  begin
    j := 2 * n mod q;      \* Assume mod returns 1, 2, ..., q *\
    i := 1 + (2 * n - j)/q;
    if i + j < q + 1 then
      begin
        x_1 := [(i + j - 2) * q + 1]/2;      \* The billiard path starts at (i + j - 1, 1) *\
        for k = 1 to j - 1 do x_{k+1} := x_k - (q - 1)/2;
        for k = j to q - i do x_{k+1} := x_k + (q + 1)/2;
        for k = q - i + 1 to q - 1 do x_{k+1} := x_k - (q - 1)/2;
      end;
    if i + j > q + 1 then
      begin
        x_1 := N - [2 * q - i - j - 1]/2;      \* The billiard path starts at (q, i + j - q) *\
        for k = 1 to q - i do x_{k+1} := x_k - (q - 1)/2;
        for k = q - i + 1 to j - 1 do x_{k+1} := x_k - (q + 1)/2;
        for k = j to q - 1 do x_{k+1} := x_k - (q - 1)/2;
      end;
end generate
```

Figure 6: Billiard quorum generation.

The formulas in 1 give us an efficient means of generating billiard quorums. Procedure generate given in Figure 6 takes as input a site n, and returns the quorum set $S_n = \{x_1, x_2, \ldots, x_q\}$. First, the location (i,j) of n on the modified grid is computed so that $R_{i,j} = n$. The site x_1 in the set S_n is the first cell on the left or the bottom side of the modified grid from which the billiard path towards $R_{i,j}$ starts. In particular x_1 is on the leftmost column of the grid whenever $i + j < q + 1$ (i.e. when the cell is above the diagonal $y = x$), and on the bottommost row whenever $i + j > q + 1$ (i.e. when the cell is below the diagonal $y = x$). As examples, $x_1 = 23$ for the broken billiard path in Figure 5 (a), and $x_1 = 38$ for the path in Figure 5 (b). A listing of billiard quorums for small parameters is given in Figure 7.

$q = 3$	$q = 5$	$q = 7$
$S_1 = \{1,2,3\}$	$S_1 = \{1,3,4,7,10\}$	$S_1 = \{1,4,5,9,13,17,21\}$
$S_2 = \{2,3,4\}$	$S_2 = \{2,4,5,6,8\}$	$S_2 = \{2,5,6,8,10,11,14\}$
$S_3 = \{1,3,4\}$	$S_3 = \{3,6,9,10,12\}$	$S_3 = \{3,6,7,9,12,15,18\}$
$S_4 = \{1,2,4\}$	$S_4 = \{4,5,6,7,8\}$	$S_4 = \{4,8,12,16,20,21,24\}$
	$S_5 = \{2,5,7,9,11\}$	$S_5 = \{5,8,9,11,13,14,17\}$
	$S_6 = \{5,6,7,8,9\}$	$S_6 = \{6,7,9,10,12,15,18\}$
	$S_7 = \{2,4,7,9,11\}$	$S_7 = \{3,7,10,13,16,19,22\}$
	$S_8 = \{5,7,8,9,11\}$	$S_8 = \{8,11,12,14,16,17,20\}$
	$S_9 = \{2,4,6,9,11\}$	$S_9 = \{7,9,10,12,13,15,18\}$
	$S_{10} = \{1,4,7,10,12\}$	$S_{10} = \{3,6,10,13,16,19,22\}$
	$S_{11} = \{2,4,6,8,11\}$	$S_{11} = \{11,14,15,17,19,20,23\}$
	$S_{12} = \{1,3,6,9,12\}$	$S_{12} = \{7,10,12,13,15,16,18\}$
		$S_{13} = \{3,6,9,13,16,19,22\}$
		$S_{14} = \{2,6,10,14,17,20,23\}$
		$S_{15} = \{7,10,13,15,16,18,19\}$
		$S_{16} = \{3,6,9,12,16,19,22\}$
		$S_{17} = \{2,5,9,13,17,20,23\}$
		$S_{18} = \{7,10,13,16,18,19,22\}$
		$S_{19} = \{3,6,9,12,15,19,22\}$
		$S_{20} = \{2,5,8,12,16,20,23\}$
		$S_{21} = \{1,5,9,13,17,21,24\}$
		$S_{22} = \{3,6,9,12,15,18,22\}$
		$S_{23} = \{2,5,8,11,15,19,23\}$
		$S_{24} = \{1,4,8,12,16,20,24\}$

Figure 7: Billiard quorums for small N.

6 Discussion

In this paper we presented a simple geometric technique for generating quorums. Our approach is based on the logical organization of sites in the network in the form of a modified grid, and the realization of quorum sets as sites lying on certain paths resembling trajectories of billiard balls. The resulting quorums are of size $\sqrt{2}\sqrt{N}$, as compared to $2\sqrt{N}$ of Meakawa's grid based method. Even though both Meakawa's grid based solution and billiard quorums are suboptimal in terms of quorum size when compared to the optimal projective plane based quorum generation method, grid based solutions are simple to realize and geometrically intuitive.

Acknowledgment

We would like to thank the referee for a detailed analysis of the results in this paper. The referee pointed out several properties of billiard quorums and general availability issues for coteries, and also brought to our attention the Triangular system of Lovász.

References

[AE91] D. Agrawal and A. El Abbadi. An Efficient and Fault-tolerant Solution for Distributed Mutual Exclusion. *ACM Transactions on Computer Systems*, pages 1–20, February 1991.

[AS68] A. A. Albert and R. Sandler. *An Introduction to Finite Projective Planes*. Holt, Rinehart, and Winston, New York, 1968.

[CAA92] S. Y. Cheung, M. H. Ammar, and M. Ahamad. The Grid Protocol: A High Performance Scheme for Maintaining Replicated Data. *IEEE Transactions on Knowledge and Data Engineering*, 4(6):582–592, December 1992.

[CR83] O. Carvalho and G. Roucairol. On Mutual Exclusion in Computer Networks. *Communications of the ACM*, 26:146–147, February 1983.

[GB85] H. Garcia-Molina and D. Barbara. How to assign votes in a distributed system. *Journal of the Association of the Computing Machinery*, 32(4):841–860, October 1985.

[Gif79] D. K. Gifford. Weighted Voting for Replicated Data. In *Proceedings of the Seventh ACM Symposium on Operating Systems Principles*, pages 150–159, December 1979.

[HPR88] J. M. Helary, N. Plouzeau, and M. Raynal. A Distributed Algorithm for Mutual Exclusion in an Arbitrary Network. *The Computer Journal*, 31(4):289–295, 1988.

[KC91] A. Kumar and S. Y. Cheung. A High Availability $\sqrt{N}$ Hierarchical Grid Algorithm for Replicated Data. *Information Processing Letters*, 40:311–316, 1991.

[Kum91] A. Kumar. Hierarchical Quorum Consensus - A New Algorithm for Managing Replicated Data. *IEEE Transactions on Computers*, 39(9):996–1004, September 1991.

[Lam78] L. Lamport. Time, Clocks, and the Ordering of Events in a Distributed System. *Communications of the ACM*, 21(7):558–565, July 1978.

[Lov73] L. Lovász. Coverings and Colorings of Hypergraph. In *Proceedings of the 4th Southeastern Conference on Combinatorics, Graph Theory, and Computing*, pages 3–12, 1973.

[Mae85] M. Maekawa. A $\sqrt{n}$ algorithm for mutual exclusion in decentralized systems. *ACM Transactions on Computer Systems*, 3(2):145–159, May 1985.

[PW95] D. Peleg and A. Wool. The Availability of Quorum Systems. *Information and Computation*, 123(2):210–223, 1995.

[RA81] G. Ricart and A.K. Agrawala. An optimal algorithm for mutual exclusion in computer networks. *Communications of the ACM*, 24:9–17, January 1981.

[Ray89] K. Raymond. A Tree-Based Algorithm for Distributed Mutual Exclusion. *ACM Transactions on Computer Systems*, 7(1):61–77, February 1989.

[Ray91] M. Raynal. A Simple Taxonomy for Distributed Mutual Exclusion Algorithms. *ACM SIGOPS Operating Systems Review*, 25(2):47–50, April 1991.

[San87] B. A. Sanders. The information structure of distributed mutual exclusion algorithms. *ACM Transactions on Computer Systems*, 5:284–299, August 1987.

[Sin88] M. Singhal. A dynamic information-structure mutual exclusion algorithm for distributed systems. Technical report, Department of Computer & Information Science, The Ohio State University, 1988.

[SK85] I. Suzuki and T. Kasami. A Distributed Mutual Exclusion Algorithm. *ACM Transactions on Computer Systems*, 3(4):344–349, November 1985.

[SS82] R. Schlichting and F. B. Schneider. Fail-Stop Processors: An Approach to Designing Fault-Tolerant Computing Systems. *ACM Transactions on Computer Systems*, 1(3):222–238, August 1982.

[vdS87] J. L. van de Snepscheut. Fair mutual exclusion on a graph of processes. *Distributed Computing*, 2:113–115, 1987.