



# Signed Quorum Systems

Haifeng Yu

IRP-TR-04-04

**Research at Intel**

INFORMATION IN THIS DOCUMENT IS PROVIDED IN CONNECTION WITH INTEL® PRODUCTS. NO LICENSE, EXPRESS OR IMPLIED, BY ESTOPPEL OR OTHERWISE, TO ANY INTELLECTUAL PROPERTY RIGHTS IS GRANTED BY THIS DOCUMENT. EXCEPT AS PROVIDED IN INTEL'S TERMS AND CONDITIONS OF SALE FOR SUCH PRODUCTS, INTEL ASSUMES NO LIABILITY WHATSOEVER, AND INTEL DISCLAIMS ANY EXPRESS OR IMPLIED WARRANTY, RELATING TO SALE AND/OR USE OF INTEL PRODUCTS INCLUDING LIABILITY OR WARRANTIES RELATING TO FITNESS FOR A PARTICULAR PURPOSE, MERCHANTABILITY, OR INFRINGEMENT OF ANY PATENT, COPYRIGHT OR OTHER INTELLECTUAL PROPERTY RIGHT. Intel products are not intended for use in medical, life saving, life sustaining applications.

Intel may make changes to specifications and product descriptions at any time, without notice.

Copyright © Intel Corporation 2004

\* Other names and brands may be claimed as the property of others.

# Signed Quorum Systems

Haifeng Yu  
Intel Research Pittsburgh / Carnegie Mellon University  
yhf@cs.cmu.edu

## ABSTRACT

With  $n$  servers that independently fail with probability of  $p < 0.5$ , it is well known that the majority quorum system achieves the best availability among all quorum systems. However, even this optimal construction requires  $(n + 1)/2$  functioning servers out of  $n$ . Furthermore, the number of probes needed to acquire a quorum is also lower bounded by  $(n + 1)/2$ .

Motivated by the need for a highly available and low probe complexity quorum system in the Internet, this paper proposes *signed quorum systems* (SQS) that can be available as long as any  $O(1)$  servers are available, and simultaneously have  $O(1)$  probe complexity. SQS provides probabilistic intersection guarantees and exploits the property of *independent mismatches* in today's Internet. Such key property has been validated previously under multiple Internet measurement traces. This paper then extensively studies the availability, probe complexity, and load of SQS, derives lower bounds for all three metrics, and constructs matching upper bounds. We show that in addition to the qualitatively superior availability and probe complexity, SQS also decouples availability from load and probe complexity, so that optimal availability can be achieved under most probe complexity and load values.

## Categories and Subject Descriptors

C.2.4 [Computer-Communication Networks]: Distributed Systems – *distributed applications*

## General Terms

Algorithms, Design, Performance, Reliability

## Keywords

Quorum Systems, Availability, Probe Complexity, Load, Tradeoff

## 1. INTRODUCTION

Quorum systems are well known techniques to achieve mutual exclusion, preserve consistency on replicated data, improve availability, and share load. A *quorum system* is a set of *quorums*, each

of which itself is a subset of *servers* from a universe of servers. It is guaranteed that any two quorums intersect. To perform an action potentially conflicting with other actions, a *client* coordinates the action with a quorum. Availability is improved because the system is available whenever a single quorum is available, while per-server load is reduced because a client no longer needs to contact all servers.

For traditional quorum systems, it is known [2] that when servers fail independently (in fail-stop fashion) with probability of  $p < 0.5$ , the majority quorum system [15] (where a quorum is any majority subset of all servers) achieves the best availability. However, even in this optimal quorum system, the system needs to have  $(n + 1)/2$  functioning servers to be available. Furthermore, the number of messages (probes) needed for a client to acquire a quorum is also lower bounded by  $(n + 1)/2$ . By guaranteeing intersection only with high probability, probabilistic quorum systems (PQS) [9] overcome such fundamental limitation on availability and probe complexity. However, the PQS construction in [9] still needs  $\theta(\sqrt{n})$  functioning servers and similar probe complexity.

Traditional quorum systems also have fundamental tradeoffs [11] among availability, probe complexity and load (the probability of accessing the busiest server):

$$1 - \text{Availability} \geq p^{n \times \text{Load}} \quad (1)$$

$$1 - \text{Availability} \geq p^{\text{Probe Complexity}} \quad (2)$$

$$\text{Load} \geq 1/\text{Probe Complexity} \quad (3)$$

Such fundamental tradeoffs prevent us from obtaining the best of multiple measures simultaneously.

This paper is motivated by the need for a highly-available and low probe complexity quorum system in the Internet. For distributed systems in the wide-area environment, management is typically decentralized and nodes are less well maintained, resulting in larger  $p$  values. Yet it is still important to achieve good availability with a small number of servers. The cost of wide-area communication in the Internet is also much more expensive than local processing, making it crucial to bound probe complexity.

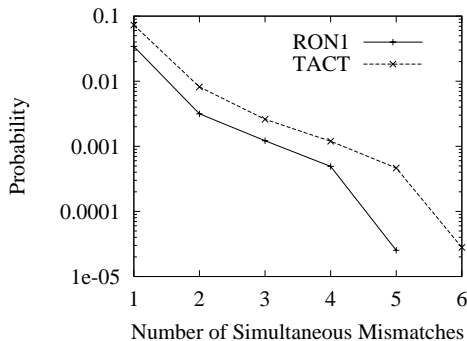
## 1.1 Our Results

To qualitatively improve the availability and probe complexity of previous quorum techniques, this paper proposes *signed quorum systems* (SQS) that can be available as long as any  $O(1)$  servers are available, and simultaneously have  $O(1)$  probe complexity. Furthermore, SQS breaks the tradeoff between availability and load/probe complexity (Inequality 1 and 2) – optimal availability can now be achieved under most load and probe complexity values.

In signed quorum systems, a quorum may contain negative ele-

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

PODC'04 July 25–28, 2004, St. Johns, Newfoundland, Canada.  
Copyright 2004 ACM 1-58113-802-4/04/0007 ...\$5.00.



**Figure 1: Sample results from [17] to validate the assumption of independent mismatches. Both curves are near-linear, indicating independence mismatches.**

ments. For example, over a three server universe  $\{1, 2, 3\}$ , a signed quorum system can be  $\{\{-1, 3\}, \{1, -2, -3\}\}$ . In the quorum of  $\{-1, 3\}$ , the element “3” is interpreted same as before. Namely, the client needs to obtain a reply from server 3. The element “-1” means that the client believes that server 1 has failed, according to some inaccurate failure detection mechanism. Since failure detection can be inaccurate, it is possible that another client can still obtain a reply from server 1. Such an event is called a *mismatch*.

Following the arguments of PQS, SQS also provides probabilistic guarantee on intersection. However, SQS does not achieve such guarantee via an explicit access strategy [9]. Rather, SQS is designed for servers randomly distributed across the Internet, and we make a key assumption\* on the independence of mismatches *on different servers*. If mismatches happen independently for different servers with probability of  $\varepsilon$ , then the probability of one client acquiring  $\{-1, 3\}$  and another client acquiring  $\{1, -2, -3\}$  is at most  $\varepsilon^2$ . This is true because two mismatches are needed on server 1 and 3. We define a *dual pair* to be a pair in the form of  $\{i, -i\}$ , and the number of dual pairs between two quorums is called their *dual overlap*. The previous two quorums thus have a dual overlap of two (from the dual pairs of  $\{-1, 1\}$  and  $\{3, -3\}$ ). In SQS, it is required that any two quorums either intersect over some positive element, or their dual overlap is at least  $2\alpha$ , where  $\alpha$  is a positive constant. As a result, if quorums  $Q$  and  $Q'$  do not intersect on any positive element, then the probability that they can be both acquired is as small as  $\varepsilon^{2\alpha}$ .

To intuitively understand the validity of our key assumption, notice that when mismatches are strongly correlated for servers randomly distributed in the Internet, they tend to indicate the existence of a “hard” partition<sup>†</sup>. However, recent network measurements and evaluation [1, 14, 18] have shown that “hard” partitions where a significant fraction of Internet nodes are unable to communicate with the rest of the network, are rare in today’s Internet.

Validating this key assumption of independent mismatches was one focus of our previous work [17] on the witness model, an implicit (non-optimal) SQS construction. Our results based on the RON traces [1] from MIT and the TACT [18] trace from Duke show that the average correlation among mismatches is below 5%. Figure 1 plots some sample results.

\*Section 2.2 will show that in fact, PQS also needs to make similar assumptions in asynchronous systems.

<sup>†</sup>A client with a lost network connection can also observe correlated mismatches, but we have shown [17] that a simple filtering step can be quite effective to prevent those clients from acquiring any quorum.

Continuing from our earlier efforts, this paper proposes and formalizes the concept of SQS, and then extensively studies the availability, probe complexity, and load of SQS. Specifically, we make the following contributions:

- We propose and formalize the concept of SQS, which generalizes traditional quorum systems. SQS provides probabilistic intersection guarantees and the probability of non-intersection is exponentially small with  $2\alpha$ .
- We study the availability of SQS and prove that our  $\mathcal{OPT}_a$  construction has the optimal availability, where the system is available as long as any  $\alpha$  servers are available. Such availability is not possible under traditional quorum systems or PQS (with non-zero intersection guarantee).
- For the set of SQS with optimal availability, we derive lower bound for probe complexity. Matching upper bound is achieved by our  $\mathcal{OPT}_d$  construction that has the same availability as  $\mathcal{OPT}_a$ . Especially, the expected probe complexity of  $\mathcal{OPT}_d$  is smaller than  $2\alpha/(1-p) = O(1)$  under arbitrary  $n$  values. Such probe complexity is again, not possible under traditional quorum systems or PQS (with non-zero intersection guarantee).
- We design a powerful *composition* technique that allows us to compose certain traditional quorum system  $\mathcal{Q}$  over  $k$  servers with  $\mathcal{OPT}_a$  over  $n$  servers ( $k \leq n$ ). We prove that the composition result is an SQS with  $\mathcal{OPT}_a$ ’s availability, and  $\mathcal{Q}$ ’s probe complexity and load. This shows that SQS breaks the fundamental tradeoff between availability and load/probe complexity (Inequality 1 and 2). Now optimal availability can be achieved all the time, as long as probe complexity is  $\Omega(\alpha)$ .
- We show that a similar load and probe complexity tradeoff (Inequality 3) exists for SQS. For any probe complexity value  $x$ ,  $\Omega(\alpha) < x < O(\sqrt{n})$ , we show that composition allows us to construct an SQS with load of  $O(1/x)$ <sup>‡</sup> and optimal availability.

The next section discusses related work, and we introduce the formal definitions of SQS, availability, load, and probe complexity in Section 3. Section 4 formalizes the notion of mismatches and rigorously argues why our SQS definition is able to bound the probability of non-intersection. We construct SQS with optimal availability, optimal probe complexity and optimal load in Section 5, 6, and 7. Finally, Section 8 draws our conclusions.

## 2. RELATED WORK

This work is a continuation from our earlier work [17] on the *witness model*, which is an implicit non-optimal SQS construction. There we focus on validating the assumption on independent mismatches and the application of the witness model to distributed consensus. The general concept of SQS, their availability, probe complexity and load properties are not studied in [17].

### 2.1 Strict Quorum Systems

Most traditional quorum systems are *strict* in the sense that they always guarantee quorum intersection. Within such context, there have been extensive research efforts [2, 4, 5, 6, 7, 11, 13, 12, 15] on quorum systems, and we only focus on those closely related to this paper.

<sup>‡</sup>Load itself has a lower bound of  $\Omega(1/\sqrt{n})$ .

Most previous work [2, 12] on the availability of quorum systems uses the same availability definition as ours, namely, the probability that at least one live quorum exists given that servers fail independently. The probe complexity of quorum systems is first studied in [13], where only worst-case probe complexity for deterministic probe algorithms is considered. Later, Hassin et.al [6] study the average probe complexity for deterministic algorithms and the worst-case probe complexity for randomized algorithms. Another measure for probe complexity is *the cost of failure* [3], defined as the normalized number of extra probes needed due to failures.

Load balancing in quorum systems is first studied in [7], where the metric considered is the load ratio between the busiest and least-loaded server. Naor et.al [11] define and study load as the probability of accessing the busiest server. Naor et.al further prove a tradeoff between availability and load (Inequality 1). The other two tradeoffs (Inequality 2 and 3) also directly follow from their proof.

Quorum systems have also been used to mask Byzantine failures [8]. To mask such failures, it is required that any two quorums intersect on sufficient number of servers so that the client will not be misled by malicious servers. In comparison, SQS only tolerates fail-stop failures and dual overlap is for the number of dual pairs, instead of direct intersection.

## 2.2 Probabilistic Quorum Systems

SQS follows PQS's spirit of providing probabilistic guarantees on quorum intersection. In PQS, such guarantee is achieved by enforcing an access strategy on quorums. For example, in one PQS construction [9], quorums are all the sets of size  $l\sqrt{n}$  and each quorum is accessed with equal probability. This guarantees an intersection probability of at least  $1 - e^{-l^2}$ .

Compared to PQS, it may appear that SQS makes an additional strong assumption of independent mismatches. In asynchronous systems, however, we argue that implementing PQS needs to make similar assumptions as well. The fundamental reason is that with an asynchronous scheduler, we do not have full control over the access strategy and the intersection guarantee of PQS can be disrupted by the scheduler.

Following consider a concrete example with two servers (1 and 2), and two clients ( $x$  and  $y$ ). Construct a PQS  $\mathcal{Q} = \{\{1\}, \{2\}, \{1, 2\}\}$  and an access strategy that simply chooses each quorum in  $\mathcal{Q}$  with probability of  $1/3$ . A simple calculation can show that intersection happens with probability of  $7/9$ . A straightforward implementation of the previous PQS would be simply to have every client try to acquire each quorum with probability of  $1/3$ . However, suppose that the asynchronous scheduler delays all messages from  $x$  to server 2. Since  $x$  cannot wait forever, it is forced to use the quorum of  $\{1\}$  all the time. Similarly, the scheduler can force  $y$  to (ultimately) always choose and use the quorum of  $\{2\}$ . At this point, the actual access strategy is already different from what we intend, and the intersection probability is actually zero. A closer look reveals that the problem is exactly caused by mismatches on both 1 and 2. Thus to implement PQS, some assumption needs to be made on mismatches to limit the power of the asynchronous scheduler.

In SQS, such assumption on mismatches is explicitly made and validated. Given that the access strategy can be disturbed by the scheduler, it is no longer a convenient way to define SQS. Instead, we directly impose a requirement on dual overlap. Ultimately, there is still a distribution (access strategy) on the quorums that are used in a given SQS. However, such distribution is determined by the scheduler and failures in the system. The dual overlap requirement ensures probabilistic intersection under any of the possible result-

ing distributions.

It is interesting to see that after clearly stating our assumption of independent mismatches and defining SQS based on dual overlap, we are able to achieve an availability and probe complexity not possible under PQS. In summary, it seems to us that implementing PQS in asynchronous systems needs an implicit assumption on mismatches, but since the assumption is implicit, PQS has not fully exploited its power.

## 3. PRELIMINARIES AND DEFINITIONS

### 3.1 Unsigned and Signed Quorum Systems

We consider a *universe*  $U = \{1, 2, \dots, n\}$  of servers. An *unsigned set system* over the universe  $U$  is a set of subsets of  $U$ . For simplicity, we will drop the phrase "over the universe  $U$ " in the following discussion.

**DEFINITION 3.1.** An (unsigned) quorum system (UQS)  $\mathcal{Q} = \{Q_1, Q_2, \dots, Q_m\}$  is an unsigned set system where  $Q_i \cap Q_j \neq \emptyset$  ( $1 \leq i, j \leq m$ ).  $Q_i$  ( $1 \leq i \leq m$ ) is called a quorum.

For any element  $i \in U$ , its *dual* is  $-i$ , and the dual of  $-i$  is  $i$ . For any set  $S \subseteq U$ , define its *dual* (denoted as  $Dual(S)$ ) to be  $\{Dual(i) | i \in S\}$ . For any set  $S \subseteq (U \cup Dual(U))$ , define its *positive part* (denoted as  $S^+$ ) to be  $S \cap U$ , and its *negative part* (denoted as  $S^-$ ) to be  $S \cap Dual(U)$ .

**DEFINITION 3.2.** A signed set system over the universe  $U$  is a set of subsets of  $U \cup Dual(U)$ , where for any set  $S$  in the signed set system,  $S \cap Dual(S) = \emptyset$ .

**DEFINITION 3.3.** For a given positive integer  $\alpha$ , a signed quorum system (SQS)  $\mathcal{Q} = \{Q_1, Q_2, \dots, Q_m\}$  is a signed set system where for any  $Q_i$  and  $Q_j$ ,  $1 \leq i, j \leq m$ , at least one of the following two conditions is satisfied:

**Intersection**  $Q_i^+ \cap Q_j^+ \neq \emptyset$

**Dual Overlap**  $|Q_i \cap Dual(Q_j)| \geq 2\alpha$

By definition, any UQS is also an SQS. When  $n < 2\alpha$ , it is impossible to satisfy dual overlap. Thus we assume  $n \geq 2\alpha$  for the remainder of the paper. Directly follows from the definition, it can be proven that any quorum in SQS must have at least one positive element.

### 3.2 Availability

Next, we define three measures of quality for quorum systems: *availability*, *load*, and *probe complexity*. Throughout this paper, we assume that each server fails independently with probability of  $p$ ,  $p < 0.5$ . Availability (informally) is the probability that the system has at least one live quorum. Following we formalize such definition using set operations.

**DEFINITION 3.4.** A configuration of the system is a set  $C$ , such that  $C \subseteq (U \cup Dual(U))$ ,  $|C| = n$  and  $C \cap Dual(C) = \emptyset$ .

Each configuration captures a unique state of all the servers. The element  $i$  belongs to the configuration if server  $i$  is available, and  $-i$  belongs to the configuration otherwise. Defining a configuration to be a set instead of a binary vector allows the use of set operations for all our reasoning.

Let  $\mathcal{C}$  denote the set of all  $2^n$  configurations. For convenience, define  $\mathcal{C}_i = \{C | C \in \mathcal{C} \text{ and } |C^+| = i\}$ ,  $0 \leq i \leq n$ .

DEFINITION 3.5. For any quorum  $Q_i$  in an SQS  $\mathcal{Q} = \{Q_1, Q_2, \dots, Q_m\}$ , its acceptance set (denoted as  $As(Q_i)$ ) is  $\{C | C \in \mathcal{C} \text{ and } Q_i \subseteq C\}$ . The acceptance set of  $\mathcal{Q}$  (denoted as  $As(\mathcal{Q})$ ) is  $\cup_{1 \leq i \leq m} As(Q_i)$ . A signed set system is called an acceptance set if it is the acceptance set of some SQS.

Clearly, if  $Q_i \subseteq C$ , it means that the quorum  $Q_i$  can be acquired under  $C$ . Each configuration  $C$  has a certain probability of occurring:  $Prob[C] = p^{|C^-|}(1-p)^{|C^+|}$ . Availability is then the probability of all configurations under which some quorum can be acquired.

DEFINITION 3.6. For any SQS  $\mathcal{Q}$ , its availability (denoted as  $Avail(\mathcal{Q})$ ) is defined as  $\sum_{C \in As(\mathcal{Q})} Prob[C]$ .

### 3.3 Probe Complexity

Informally, probe complexity is the number of probes a client needs to make in order to acquire a quorum, or to realize that no live quorum exists. Notice that to acquire a quorum in SQS, the client also needs to probe (and fail to obtain a reply) from those servers corresponding to the negative elements in the quorum.

To formalize, we first define the probe strategy used by the client to determine which server to probe next:

DEFINITION 3.7. *Probe Strategy* [13]: A probe strategy is a binary tree. Each non-leaf node of the tree is labeled with a server, and two outgoing edges are marked by the two possible outcomes of the probe, namely, success or failure. Each leaf of tree denotes an outcome of the probe algorithm, which can either be that a quorum has been acquired or that no live quorums exist.

A probe strategy is *non-adaptive* if for any node in the probe tree, all its children are labeled with the same server. In other words, a non-adaptive probe strategy does not adjust the probe sequence based on the results of earlier probes.

Let  $\mathcal{Q}$  be an SQS and let  $\Psi$  be the set of all probe strategies for  $\mathcal{Q}$ . For any configuration  $C \in \mathcal{C}$  and any probe strategy  $\psi \in \Psi$ , define  $path(\psi, C)$  to be  $\psi$ 's branch corresponding to  $C$ , and define  $depth(\psi, C)$  to be the length of  $path(\psi, C)$ . The *expected probe complexity for deterministic probe algorithms* (denoted as  $PC_e(\mathcal{Q})$ ) [6] is then:

$$PC_e(\mathcal{Q}, \psi) = \sum_{C \in \mathcal{C}} depth(\psi, C) \cdot Prob[C]$$

$$PC_e(\mathcal{Q}) = \min_{\psi \in \Psi} \{PC_e(\mathcal{Q}, \psi)\}$$

The *worst-case probe complexity for deterministic probe algorithms* (denoted as  $PC_w(\mathcal{Q})$ ) [13] is defined as:

$$PC_w(\mathcal{Q}, \psi) = \max_{C \in \mathcal{C}} \{depth(\psi, C)\}$$

$$PC_w(\mathcal{Q}) = \min_{\psi \in \Psi} \{PC_w(\mathcal{Q}, \psi)\}$$

Randomized probe strategies are necessary to achieve any non-trivial load. A randomized probe algorithm has a distribution  $\mu$  over  $\Psi$ . Let  $PC_e^*(\mathcal{Q})$  denote the *expected probe complexity for randomized probe algorithms*:

$$PC_e^*(\mathcal{Q}, \mu) = \sum_{C \in \mathcal{C}} E_\mu[depth(\psi, C), \psi \in \Psi] \cdot Prob[C]$$

$$PC_e^*(\mathcal{Q}) = \min_\mu \{PC_e^*(\mathcal{Q}, \mu)\}$$

It can be easily shown that  $PC_e(\mathcal{Q}) = PC_e^*(\mathcal{Q})$  for any  $\mathcal{Q}$ , and thus we will focus on  $PC_e^*(\mathcal{Q})$  only. The *worse-case probe complexity for randomized probe algorithms* (denoted as  $PC_w^*(\mathcal{Q})$ ) [6]

is:

$$PC_w^*(\mathcal{Q}, \mu) = \max_{C \in \mathcal{C}} \{E_\mu[depth(\psi, C), \psi \in \Psi]\}$$

$$PC_w^*(\mathcal{Q}) = \min_\mu \{PC_w^*(\mathcal{Q}, \mu)\}$$

It has been shown [6] that in certain cases,  $PC_w^*(\mathcal{Q}) < PC_w(\mathcal{Q})$ .

### 3.4 Load

In most previous work [9, 11], load is defined based on the *access strategy* of a quorum system. The access strategy is a distribution over all quorums describing the probability that each individual quorum is used. Load is then defined as the probability of accessing the busiest server (as determined by the access strategy).

However, to acquire a quorum  $Q$ , we may potentially probe and induce load on more servers than those in  $Q$ , which makes the previous definition inaccurate. Thus this paper uses a more practical definition directly based on probe strategy. Such definition is part of our contribution.

For any node  $a$  in the probe strategy tree  $\psi$ , define its *load* to be:

$$\sum_{C \in \mathcal{C} \text{ and } path(\psi, C) \text{ contains } a} Prob[C]$$

Server  $i$ 's *load* is the sum of the load of those tree nodes labeled  $i$ . With a deterministic probe algorithm, the root of the probe tree always has a load of 1.0. Thus, we only define load for randomized probe algorithms. Let  $\mathcal{Q}$  be an SQS,  $\Psi$  be the set of all probe strategies for  $\mathcal{Q}$ , and  $\mu$  be a distribution over  $\Psi$ , we define  $\mathcal{Q}$ 's load (denoted as  $Load(\mathcal{Q})$ ) to be:

$$\min_\mu \{ \max_{1 \leq i \leq n} \{ E_\mu[\text{server } i\text{'s load under } \psi, \psi \in \Psi] \} \}$$

With our new definition, the load of a quorum system is at least as high as the load under previous definitions [9, 11]. All lower bounds in the paper are applicable to the traditional optimistic definition, while all upper bounds hold under our new pessimistic definition. This makes our results as strong as possible.

Also notice that according to our definition,  $Load(\mathcal{Q})$  and  $PC_e^*(\mathcal{Q})$  may or may not be achieved under the same probe algorithm. But for all our probe complexity and load upper bounds, we actually use the same probe algorithm, which again makes our results strong.

## 4. RATIONALE OF SQS DEFINITION

This section proves that our simple SQS definition is sufficient to control the probability of non-intersection. We first formalize our notion of mismatch and the independent mismatch assumption. We say that a client *reaches* a server if and only if it obtains a reply from the server. A server probed by two clients can be in one of the following four states: i)  $(-, -)$ : Neither client reaches the server; ii)  $(+, -)$ : The first client reaches the server, but not the second client; iii)  $(-, +)$ : The first client does not reach the server, but the second client does; iv)  $(+, +)$ : Both clients reach the server. The states of  $(-, +)$  and  $(+, -)$  are called *mismatches*.

We assume that mismatch on one server is independent of other servers' states. Further, we assume that  $Prob[\text{mismatch} | \text{state is not } (-, -)] \leq \varepsilon$  for any server. This intuitively means that given that one client reaches a server, the probability that the other client cannot reach the server is at most  $\varepsilon$ . Notice that just assuming  $Prob[\text{mismatch}] \leq \varepsilon$  is not sufficient. Otherwise we can let  $Prob[(-, +)] = Prob[(+, -)] = \varepsilon/2$ ,  $Prob[(+, +)] = 0$ , and  $Prob[(-, -)] = 1 - \varepsilon$ , then intersection can never happen.

We now use an example to show that it is not trivial that SQS definition controls non-intersection probability. Dual overlap between two non-intersecting quorums  $Q_1$  and  $Q_2$  does bound the

probability that they are both acquired. However, there can be many other quorums that do not intersect with  $Q_1$  either, and the probability of obtaining *one* of those quorums may not be low. Suppose  $(n - 1) = (m - 1) \times 2\alpha$  and we consider the SQS  $\mathcal{Q} = \{Q_1, Q_2, \dots, Q_m\}$ , where:

$$\begin{aligned} Q_1 &= \{1, 2, \dots, (n - 1)\} \\ Q_2 &= \{-1, -2, \dots, -2\alpha, n\} \\ Q_3 &= \{-(2\alpha + 1), -(2\alpha + 2), \dots, -4\alpha, n\} \\ &\dots \\ Q_m &= \{-(n - 2\alpha), -(n - 2\alpha + 1), \dots, -(n - 1), n\} \end{aligned}$$

Now assume that the first client acquires  $Q_1$  and the second client reaches the last server. The second client may or may not reach the first  $n - 1$  servers, and there is a mismatch whenever it cannot reach a server. The independent mismatch assumption controls the probability that mismatches may occur for a *given* set of  $2\alpha$  servers. However, when  $n$  grows we have many such sets, and with high probability, the second client can actually find a quorum that does not intersect with  $Q_1$ .

Fortunately, we will show that by restricting the probe strategy and by imposing some natural requirement on the clients, the probability of non-intersection in an SQS is truly exponentially small with  $2\alpha$ . This allows us to use the simple definition of SQS and not to be concerned with the details of such probability. All the lower bounds in this paper are derived without the restriction on probe strategy and client behavior, while all upper bounds satisfy such restrictions.

We first explain the requirement on client behavior. When a client acquires a quorum  $Q$  in an SQS using a given probe strategy, define its *probed servers* to be the set  $S$  where  $i$  belongs to  $S$  if the client reaches server  $i$ , and  $-i$  belongs to  $S$  if the client does not reach server  $i$ . Clearly,  $S$  is a superset of  $Q$ .  $S$  may contain elements not in  $Q$  because the client may probe servers not belonging to the quorum ultimately acquired (i.e., wasted probes). In a traditional quorum system, a client only needs to coordinate with (e.g., read from) the servers in  $Q^+$ . In SQS, we instead require that the client coordinate with all servers in  $S^+$ . Namely, the client should coordinate with any server that it reaches during the probing process. Since the servers in  $S^+$  are already reached by the client, such a requirement does not result in material difference from a practical perspective. In particular, it does not affect the availability, probe complexity or load of the SQS. Most protocols using quorum systems already implicitly meet this requirement. One (contrived) counter-example is when we use SQS to implement a shared register, the reader can violate this requirement by explicitly discarding the responses from servers not belonging to the acquired quorum.

With the above requirement on client, we define intersection as following:

**DEFINITION 4.1.** *Consider two clients who each acquire  $Q_1$  and  $Q_2$  in an SQS, respectively. Let  $S_1$  and  $S_2$  be the probed servers of the two clients, respectively. We say that the two clients intersect if and only if  $S_1^+ \cap S_2^+ \neq \emptyset$ .*

We now prove that with the above intersection definition and a deterministic, non-adaptive probe strategy, SQS controls the probability of non-intersection.

**THEOREM 4.2.** *Consider two clients using the same deterministic, non-adaptive probe strategy, and let “non-intersection” denote the event that both clients acquire some quorum, but they do not intersect. Then  $\text{Prob}[\text{non-intersection}] \leq \varepsilon^{2\alpha}$ .*

**Proof:** Let  $Q_1$  be the random variable denoting the quorum acquired by the first client and  $S_1$  be the probed servers. Similarly

define  $Q_2$  and  $S_2$ . Let  $D$  be the random variable denoting the set of servers that are in the  $(-, -)$  state. It suffices to prove that  $\text{Prob}[(S_1^+ \cap S_2^+ = \emptyset) \mid D] \leq \varepsilon^{2\alpha}$  under any  $D$ .

Since we are considering a deterministic, non-adaptive probe strategy, the clients always probe the servers in the same order. Let such order be  $T$  and next delete all servers in  $D$  from  $T$ . We argue that if  $S_1^+ \cap S_2^+ = \emptyset$  (which implies  $Q_1^+ \cap Q_2^+ = \emptyset$ ), then both clients must have probed the first  $2\alpha$  servers in  $T$ . The reason is that when  $Q_1^+ \cap Q_2^+ = \emptyset$ ,  $Q_1$  and  $Q_2$  must satisfy dual overlap. It is impossible for  $D$  to contain any server in  $Q_1 \cap \text{Dual}(Q_2)$ , since the servers in  $D$  are all in state  $(-, -)$ . Thus every server in  $Q_1 \cap \text{Dual}(Q_2)$  must be in  $T$ , and each client probes at least  $2\alpha$  servers in  $T$ . Finally, since  $T$  is the only probe sequence allowed, both client must have probed at least the *first*  $2\alpha$  servers in  $T$ .

Now consider these first  $2\alpha$  servers in  $T$ . For  $S_1^+$  and  $S_2^+$  not to intersect, there must be a mismatch on each of these  $2\alpha$  servers. So we have  $\text{Prob}[(S_1^+ \cap S_2^+ = \emptyset) \mid D] \leq \text{Prob}[\text{mismatches on first } 2\alpha \text{ servers in } T \mid D] \leq \varepsilon^{2\alpha}$ .  $\square$

To understand the previous theorem, consider the example at the beginning of this section. With a deterministic and non-adaptive probe strategy, the two clients will probe the servers in exactly the same order. Further because of our extended definition of intersection, a client cannot intentionally skip any server causing intersection. This means that even though the second client will still likely acquire some  $Q_i$  ( $2 \leq i \leq m$ ) that does not intersect with  $Q_1$ , the probed servers of the second client will intersect with  $Q_1$  with high probability.

The extended definition of intersection seems necessary for Theorem 4.2 to hold. The requirement of “deterministic and non-adaptive probe strategy” is a sufficient but not necessary condition. We conjecture that any probe strategy that is independent of mismatches suffices. However, we are already able to construct all our upper bounds using deterministic and non-adaptive probe strategies (except for the SQS in Section 7.2, where we will present a customized proof for its randomized and adaptive probe strategy). Thus we leave the exact requirements on probe strategy to future work.

## 5. SQS WITH OPTIMAL AVAILABILITY

Starting from this section, we search for the “best” SQS in terms of availability, probe complexity, and load. We derive lower bounds for all three metrics, and achieve matching upper bounds (within constants). Our results will clearly determine the benefits we can obtain in exchange for a small probability of non-intersection.

### 5.1 Sufficient and Necessary Conditions for Optimal Availability

SQS relaxes from UQS by allowing dual overlap to replace intersection. This improves availability because quorums may now contain only a small number of positive elements and rely on the negative elements to satisfy the dual overlap condition. In order to find the SQS with the highest availability, we first restrict ourselves to acceptance sets. It is easy to show that acceptance sets themselves are SQS also. Furthermore, an SQS’s acceptance set has the same availability as the SQS. This means that there must exist an acceptance set that can achieve optimal availability.

**THEOREM 5.1.** *If  $\mathcal{Q}$  is an SQS, then: i)  $As(\mathcal{Q})$  is an SQS; ii)  $As(As(\mathcal{Q})) = As(\mathcal{Q})$ ; iii)  $Avail(\mathcal{Q}) = Avail(As(\mathcal{Q}))$ .*

**Proof:** It is obvious that  $As(\mathcal{Q})$  is an SQS. Since all elements of  $As(\mathcal{Q})$  have size of  $n$  and are actually configurations, it is also obvious that  $As(\mathcal{Q}) \subseteq As(As(\mathcal{Q}))$ . Now consider any  $S \in As(As(\mathcal{Q}))$ . There must exist  $S' \in As(\mathcal{Q})$  and  $S'' \in \mathcal{Q}$ , such

size = $n$	$\text{pos} \geq \alpha$
When $n = 3$ and $\alpha = 1$ , $\mathcal{OPT}_a =$	
$\{1, -2, -3\}$	$\{1, 2, -3\}$
$\{-1, 2, -3\}$	$\{-1, 2, 3\}$
$\{-1, -2, 3\}$	$\{1, -2, 3\}$

**Figure 2:**  $\mathcal{OPT}_a$ . All quorums have size of  $n$  and at least  $\alpha$  positive elements.

that  $S'' \subseteq S' \subseteq S$ . Since  $S''$  belongs to  $\mathcal{Q}$  and  $S$  is a configuration, we know that  $S \in \text{As}(\mathcal{Q})$  and  $\text{As}(\text{As}(\mathcal{Q})) \subseteq \text{As}(\mathcal{Q})$ . This means  $\text{As}(\text{As}(\mathcal{Q})) = \text{As}(\mathcal{Q})$ . Finally, directly from the definition of availability, we have  $\text{Avail}(\mathcal{Q}) = \text{Avail}(\text{As}(\mathcal{Q}))$ .  $\square$

An acceptance set has the nice property that its quorums are actually configurations. As a result, the availability of an acceptance set  $\mathcal{Q}$  is simply  $\sum_{Q \in \mathcal{Q}} \text{Prob}[Q]$ .

Since the size of any quorum in an acceptance set is  $n$ , as long as all quorums have at least  $\alpha$  positive elements, dual overlap or intersection must be satisfied. Thus, we construct the following SQS  $\mathcal{OPT}_a = \cup_{i=\alpha}^n \mathcal{C}_i$ .  $\mathcal{OPT}_a$  contains all configurations that have at least  $\alpha$  available servers (Figure 2).

It is easy to see that  $\mathcal{OPT}_a$  is at least “locally optimal”. Namely, we cannot add another configuration into  $\mathcal{OPT}_a$  while still keeping it an SQS. We now intend to prove that  $\mathcal{OPT}_a$  is also “globally optimal”. The challenge is to show that availability cannot be improved by replacing some existing quorums in  $\mathcal{OPT}_a$  with some new ones. We prove this by carefully grouping the new quorums to correspond to the sets of  $\mathcal{C}_\alpha, \dots, \mathcal{C}_{2\alpha-1}$ . Using a bipartite graph and based on the definition of SQS, we show that the size of each group cannot be larger than the number of deleted old quorums from the corresponding  $\mathcal{C}_i$ . Finally, it is easy to show that with  $p < 0.5$ , any new quorum contributes less to availability than any old quorum in the corresponding  $\mathcal{C}_i$ . Following we formalize our arguments:

**THEOREM 5.2.**  $\mathcal{OPT}_a$  is an SQS.

**LEMMA 5.3.** For any acceptance set  $\mathcal{Q}$ , if  $\mathcal{C}_i \cap \mathcal{Q} \neq \emptyset$  for some  $0 \leq i \leq \alpha - 1$ , then  $\text{Avail}(\mathcal{Q}) < \text{Avail}(\mathcal{OPT}_a)$ .

**Proof:** Let  $\mathcal{T}_i = \mathcal{C}_i \cap \mathcal{Q}$ , for  $0 \leq i \leq n$ . Since  $\mathcal{T}_i$  and  $\mathcal{T}_j$  must be disjoint for different  $i$  and  $j$ , we know that the availability of  $\mathcal{Q}$  is  $\sum_{i=0}^n \sum_{C \in \mathcal{T}_i} \text{Prob}[C]$ . Similarly, the availability of  $\mathcal{OPT}_a$  is  $\sum_{i=\alpha}^n \sum_{C \in \mathcal{C}_i} \text{Prob}[C]$ .

Next, we show that  $|\mathcal{T}_i| + |\mathcal{T}_{2\alpha-i-1}| \leq |\mathcal{C}_{2\alpha-i-1}|$  for  $0 \leq i \leq \alpha - 1$ . We construct the following bipartite graph, which contains all configurations in  $\mathcal{T}_i$  (left side of the graph) and  $\mathcal{C}_{2\alpha-i-1}$  (right side of the graph) as nodes. An edge is added between two vertices  $C \in \mathcal{T}_i$  and  $C' \in \mathcal{C}_{2\alpha-i-1}$  in the bipartite graph if and only if  $C^+ \cap C'^+ = \emptyset$ . Because  $|C^+| = i$  and  $|C'^+| = 2\alpha - i - 1$ ,  $C$  and  $C'$  can never satisfy the dual overlap condition. Thus adding an edge between  $C$  and  $C'$  means that they cannot both appear in  $\mathcal{Q}$ . Each vertex in the left part of the bipartite graph has a degree of exactly  $\binom{n-i}{2\alpha-i-1}$ . On the other hand, the degree of the vertices in the right part of the graph is at most  $\binom{n-2\alpha+i+1}{i}$ . We want to show that  $\binom{n-i}{2\alpha-i-1} > \binom{n-2\alpha+i+1}{i}$ :

$$\begin{aligned} & \binom{n-i}{x} > \binom{n-x}{i} \quad (\text{where } x = 2\alpha - i - 1 > i) \\ \Leftrightarrow & (n-i)!i! > x!(n-x)! \\ \Leftrightarrow & (n-i)(n-i-1)\dots(n-x+1) \\ & > x(x-1)\dots(i+1) \end{aligned}$$

Both sides in the last inequality have  $x - i$  terms. Since  $n - i > x$  (given that  $n \geq 2\alpha$ ), the last inequality holds, as well as all inequalities above.

With  $|\mathcal{T}_i|$  vertices at the left side of the graph, the graph has altogether  $|\mathcal{T}_i| \times \binom{n-i}{2\alpha-i-1}$  edges. Since each vertex at the right side of the graph has at most a degree of  $\binom{n-2\alpha+i+1}{i}$ , and also because  $\binom{n-i}{2\alpha-i-1} > \binom{n-2\alpha+i+1}{i}$ , there must be at least  $|\mathcal{T}_i|$  (or  $|\mathcal{T}_i| + 1$  when  $\mathcal{T}_i$  is not empty) vertices with non-zero degree at the right side of the graph. None of these configurations can appear in  $\mathcal{Q}$ . As a result, we know that  $|\mathcal{T}_{2\alpha-i-1}| \leq |\mathcal{C}_{2\alpha-i-1}| - |\mathcal{T}_i|$ . The “ $\leq$ ” sign becomes “ $<$ ” if  $\mathcal{T}_i$  is nonempty.

For any configuration  $C \in \mathcal{C}_i$  and  $C' \in \mathcal{C}_{2\alpha-i-1}$  ( $0 \leq i \leq \alpha - 1$ ), it must be true that  $\text{Prob}[C] \leq \text{Prob}[C']$  (because  $p \leq 0.5$  and also  $2\alpha - i - 1 > i$ ). As a result, we have  $\sum_{C \in \mathcal{T}_i} \text{Prob}[C] + \sum_{C \in \mathcal{T}_{2\alpha-i-1}} \text{Prob}[C] \leq \sum_{C \in \mathcal{C}_{2\alpha-i-1}} \text{Prob}[C]$ , where “ $\leq$ ” becomes “ $<$ ” if  $\mathcal{T}_i$  is nonempty. Now we have:

$$\begin{aligned} \text{Avail}(\mathcal{Q}) &= \sum_{i=0}^n \left( \sum_{C \in \mathcal{T}_i} \text{Prob}[C] \right) \\ &= \sum_{i=0}^{\alpha-1} \left( \sum_{C \in \mathcal{T}_i} \text{Prob}[C] + \sum_{C \in \mathcal{T}_{2\alpha-i-1}} \text{Prob}[C] \right) \\ &\quad + \sum_{i=2\alpha}^n \sum_{C \in \mathcal{T}_i} \text{Prob}[C] \\ &< \sum_{i=\alpha}^{2\alpha-1} \sum_{C \in \mathcal{C}_i} \text{Prob}[C] + \sum_{i=2\alpha}^n \sum_{C \in \mathcal{C}_i} \text{Prob}[C] \\ &= \sum_{i=\alpha}^n \sum_{C \in \mathcal{C}_i} \text{Prob}[C] = \text{Avail}(\mathcal{OPT}_a) \end{aligned}$$

$\square$

**THEOREM 5.4.** For any SQS  $\mathcal{Q}$ ,  $\text{Avail}(\mathcal{OPT}_a) \geq \text{Avail}(\mathcal{Q})$ .

**Proof:** From Theorem 5.1, we know that  $\text{As}(\mathcal{Q})$  has the same availability as  $\mathcal{Q}$ . If  $\mathcal{C}_i \cap \text{As}(\mathcal{Q}) \neq \emptyset$  for same  $i$ ,  $0 \leq i \leq \alpha - 1$ , Lemma 5.3 tells us that  $\text{As}(\mathcal{Q})$  has lower availability than  $\mathcal{OPT}_a$ . On the other hand, if  $\mathcal{C}_i \cap \text{As}(\mathcal{Q}) = \emptyset$  for all  $i$ ,  $0 \leq i \leq \alpha - 1$ , then  $\text{As}(\mathcal{Q})$  can have availability no higher than  $\mathcal{OPT}_a$ .  $\square$

**COROLLARY 5.5.** For any SQS  $\mathcal{Q}$ , if  $\mathcal{OPT}_a \subseteq \text{As}(\mathcal{Q})$ , then  $\text{Avail}(\mathcal{Q}) = \text{Avail}(\mathcal{OPT}_a)$ .

From Lemma 5.3, we can prove a stronger result, which says that we always get  $\mathcal{OPT}_a$  if we “expand” any SQS with optimal availability:

**COROLLARY 5.6.** For any SQS  $\mathcal{Q}$ ,  $\text{Avail}(\mathcal{Q}) = \text{Avail}(\mathcal{OPT}_a)$  if and only if  $\text{As}(\mathcal{Q}) = \mathcal{OPT}_a$ .

## 5.2 Non-existence of “Global Minimum”

Even though  $\mathcal{OPT}_a$  achieves optimal availability, it has some undesirable properties. For example, all its quorums are of size  $n$ , which means  $n$  probes are needed to acquire any quorum. Are there any SQS with better properties but still preserving optimal availability? Such definition of “better” is usually defined as *domination* [4]:

**DEFINITION 5.7.** Two SQS  $\mathcal{Q}$  and  $\mathcal{Q}'$ ,  $\mathcal{Q}$  dominates  $\mathcal{Q}'$  (denoted as  $\mathcal{Q} \prec \mathcal{Q}'$ ) if for  $\forall Q' \in \mathcal{Q}'$ ,  $\exists Q \in \mathcal{Q}$ , such that  $Q \subseteq Q'$ .

If  $\mathcal{Q}$  dominates  $\mathcal{Q}'$ , it usually means that  $\mathcal{Q}$  has both better probe complexity and load.

For UQS, it can be easily shown that the majority quorum system dominates any UQS that has optimal availability, which means that the majority system is the “globally minimum”. If we can find such a “global minimum” for SQS, then that “minimum” will likely give us the best probe complexity and load. Interestingly however, we will show that such a “global minimum” does not exist for SQS.

To prove this result, we first study the common properties of SQS with optimal availability (illustrated in Figure 3):

**THEOREM 5.8.** *Suppose  $n \geq 3\alpha - 1$ . For any SQS  $\mathcal{Q}$  where  $Avail(\mathcal{Q}) = Avail(OPT_a)$ , it must be true that:*

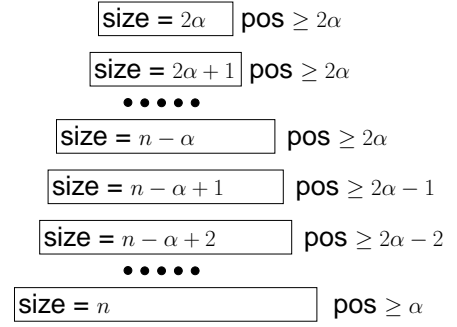
1.  $\forall Q \in \mathcal{Q}, |Q^+| \geq \alpha$ .
2.  $\mathcal{C}_\alpha \subseteq \mathcal{Q}$ .
3.  $\forall Q \in \mathcal{Q}$ , if  $\alpha \leq |Q^+| \leq 2\alpha - 1$ , then  $|Q| \geq n + \alpha - |Q^+|$ .
4.  $\forall Q \in \mathcal{Q}, |Q| \geq 2\alpha$ .

**Proof:**

1. Prove by contradiction and suppose  $\exists Q \in \mathcal{Q}$ , such that  $|Q^+| < \alpha$ . Then there must exist  $C \in \mathcal{C}_{|Q^+|}$  such that  $Q \subseteq C$  and  $C \in As(\mathcal{Q})$ . However, from Lemma 5.3, we know that  $As(\mathcal{Q})$  (and in turn,  $\mathcal{Q}$ ) cannot have optimal availability. Contradiction.
2. Prove by contradiction and assume  $\exists C \in \mathcal{C}_\alpha$ , such that  $C \notin \mathcal{Q}$ . Without loss of generality, suppose  $C = \{1, 2, \dots, \alpha, -(\alpha+1), \dots, -n\}$ . Since  $\mathcal{Q}$  has optimal availability, Corollary 5.6 tells us that  $C \in As(\mathcal{Q})$ . Thus there exists  $Q \in \mathcal{Q}$  such that  $Q \subset C$ . Since  $|Q^+| \geq \alpha$  (from 1), again without loss of generality, assume  $Q = \{1, 2, \dots, \alpha, -(\alpha+1), \dots, -x\}$ , where  $x \leq n - 1$ .  
Next we will find a quorum  $Q' \in \mathcal{Q}$  such that  $Q$  and  $Q'$  cannot satisfy intersection or dual overlap. Because  $\mathcal{Q}$  has optimal availability and  $As(\mathcal{Q}) = OPT_a$ , we know that  $\exists T \in As(\mathcal{Q})$ , where  $T = \{-1, -2, \dots, -(n-\alpha), n-\alpha+1, \dots, n\}$ . Since  $n \geq 3\alpha - 1$ , we have  $n - \alpha + 1 > \alpha$  and  $T^+ \cap Q^+ = \emptyset$ . Furthermore, it is clear that  $|T \cap Dual(Q)| = 2\alpha + x - n \leq 2\alpha - 1$ . Since  $T \in As(\mathcal{Q})$ , there must exist  $Q' \in \mathcal{Q}$  such that  $Q' \subseteq T$ . Based on the property between  $T$  and  $Q$ , it is easy to see that  $Q^+ \cap Q'^+ = \emptyset$  and  $|Q \cap Dual(Q')| \leq 2\alpha - 1$ . This means that  $\mathcal{Q}$  cannot be an SQS. Contradiction.
3. Prove by contradiction and assume there exists  $Q \in \mathcal{Q}$ , such that  $|Q^+| = x$  and  $|Q| = y$ , where  $\alpha \leq x \leq 2\alpha - 1$  and  $x \leq y \leq n + \alpha - x - 1$ . Without loss of generality, suppose  $Q = \{1, 2, \dots, x, -(x+1), \dots, -y\}$ . Since  $\mathcal{C}_\alpha \subseteq \mathcal{Q}$ , we have  $Q' \in \mathcal{Q}$ , where  $Q' = \{-1, -2, \dots, -(n-\alpha), n-\alpha+1, \dots, n\}$ . Since  $n \geq 3\alpha - 1$  and  $\alpha \leq x \leq 2\alpha - 1$ , it must be true that  $n - \alpha + 1 > x$  and  $Q^+ \cap Q'^+ = \emptyset$ . Furthermore, because  $y \leq n + \alpha - x - 1$ , we have  $|Q \cap Dual(Q')| = x + (y - (n - \alpha)) \leq 2\alpha - 1$ . As a result,  $Q$  and  $Q'$  satisfy neither intersection nor dual overlap.  $\mathcal{Q}$  is not an SQS. Contradiction.
4. From 1, we already know that  $\forall Q \in \mathcal{Q}, |Q^+| \geq \alpha$ . If  $|Q^+| \geq 2\alpha$ , then trivially  $|Q| \geq 2\alpha$ . If  $\alpha \leq |Q^+| \leq 2\alpha - 1$ , then from 3, we know that  $|Q| \geq n + \alpha - |Q^+| \geq n - \alpha + 1 \geq 2\alpha$ .

□

To make our result on “global minimum” strong, we define a weaker form of domination to eliminate the effects of permutation.



**Figure 3:** Any quorum in  $\mathcal{Q}$ , where  $Avail(\mathcal{Q}) = Avail(OPT_a)$ , must be one of the forms above.

**DEFINITION 5.9.** *Consider a permutation  $X = (x_1, x_2, \dots, x_n)$  of  $(1, 2, \dots, n)$ . For any set  $S \subseteq U \cup Dual(U)$ ,  $S$ 's permutation according to  $X$  (denoted as  $Perm_X(S)$ ) is  $\{i|x_i \in S\} \cup \{-i|(-x_i) \in S\}$ . An SQS  $\mathcal{Q}$ 's permutation according to  $X$  (denoted as  $Perm_X(\mathcal{Q})$ ) is  $\{Perm_X(Q)|Q \in \mathcal{Q}\}$ . Two SQS  $\mathcal{Q}$  and  $\mathcal{Q}'$ ,  $\mathcal{Q}$  dominates  $\mathcal{Q}'$  after permutation (denoted as  $\mathcal{Q} \prec_{\exists} \mathcal{Q}'$ ) if  $\exists X$ , such that  $\mathcal{Q} \prec Perm_X(\mathcal{Q}')$ .*

The crux of proving the non-existence of “global minimum” is the following two SQS constructions, where no SQS can dominate both of them.

$$\begin{aligned} OPT_b &= \{\{1, 2, \dots, 2\alpha\} \cup OPT_a\} \\ HOL\mathcal{E} &= \{S|S \subseteq (U \cup Dual(U)) \text{ and } |S| = n - 1 \\ &\quad \text{and } |S^+| = \alpha + 1\} \\ OPT_c &= HOL\mathcal{E} \cup OPT_a \end{aligned}$$

We can easily prove that both  $OPT_b$  and  $OPT_c$  are SQS with optimal availability:

**THEOREM 5.10.**  *$OPT_b$  is an SQS and  $Avail(OPT_b) = Avail(OPT_a)$ .*

**THEOREM 5.11.**  *$OPT_c$  is an SQS and  $Avail(OPT_c) = Avail(OPT_a)$ .*

**Proof:** Obviously,  $OPT_c$  is a signed set system. Consider any two sets  $S$  and  $T$  in  $OPT_c$ . If both of them belong to  $OPT_a$ , then clearly they satisfy either intersection or dual overlap. If none of them belong to  $OPT_a$ , and  $S^+ \cap T^+ = \emptyset$ , then it must be true that  $|S^+ \cap Dual(T^-)| \geq \alpha$ . Similarly, we have  $|T^+ \cap Dual(S^-)| \geq \alpha$ . Thus,  $S$  and  $T$  must satisfy the dual overlap condition. Finally, if  $S \in HOL\mathcal{E}$  and  $T \in OPT_a$ , suppose  $T \in \mathcal{C}_j$  ( $j \geq \alpha$ ). If  $S^+ \cap T^+ = \emptyset$ , it must be true that  $|S^+ \cap Dual(T^-)| \geq \alpha + 1$  and  $|T^+ \cap Dual(S^-)| \geq \alpha - 1$ . Then again,  $S$  and  $T$  satisfy the dual overlap condition. Thus we have shown that  $OPT_c$  is an SQS. Finally, since  $OPT_a \subseteq As(OPT_c)$ , we know from Corollary 5.5 that  $Avail(OPT_c) = Avail(OPT_a)$ . □

Now notice that  $OPT_b$  contains the quorum of  $\{1, 2, \dots, 2\alpha\}$ , while  $OPT_c$  contains the quorum of  $\{-2, -3, \dots, -(n - \alpha - 1), (n - \alpha), \dots, n\}$ . When  $n \geq 3\alpha + 1$ , these two quorums do not satisfy either intersection or dual overlap. From this observation, we can prove that no SQS can dominate both  $OPT_b$  and  $OPT_c$ . This is true even after permutation since  $OPT_c$  remains unchanged after any permutation.

**THEOREM 5.12.** *Suppose  $n \geq 3\alpha + 1$ . There does not exist an SQS  $\mathcal{Q}$ , such that for  $\forall \mathcal{Q}'$  where  $Avail(\mathcal{Q}') = Avail(OPT_a)$ ,  $\mathcal{Q} \prec_{\exists} \mathcal{Q}'$ .*

**Proof:** We will show that there does not exist an SQS  $\mathcal{Q}$  such that  $\mathcal{Q} \prec_{\exists} OPT_b$  and  $\mathcal{Q} \prec_{\exists} OPT_c$ . Prove by contradiction and suppose such  $\mathcal{Q}$  exists. This means there exist two permutations  $X$  and  $Y$  such that  $\mathcal{Q} \prec Perm_X(OPT_b)$  and  $\mathcal{Q} \prec Perm_Y(OPT_c)$ . Let  $\bar{X}$  be  $X$ 's reverse permutation, i.e.,  $Perm_{\bar{X}}(Perm_X(S)) = S$  for  $\forall S$ . Obviously, such  $\bar{X}$  must exist.

Now we have:

$$\begin{aligned} \mathcal{Q} &\prec Perm_X(OPT_b) \\ \Rightarrow Perm_{\bar{X}}(\mathcal{Q}) &\prec Perm_{\bar{X}}(Perm_X(OPT_b)) \\ \Rightarrow Perm_{\bar{X}}(\mathcal{Q}) &\prec OPT_b \end{aligned}$$

and:

$$\begin{aligned} \mathcal{Q} &\prec Perm_Y(OPT_c) \\ \Rightarrow Perm_{\bar{X}}(\mathcal{Q}) &\prec Perm_{\bar{X}}(Perm_Y(OPT_c)) \\ \Rightarrow Perm_{\bar{X}}(\mathcal{Q}) &\prec OPT_c \\ &\text{(because } OPT_c \text{ remains unchanged} \\ &\text{after any permutation)} \end{aligned}$$

Since  $\mathcal{Q}$  is an SQS,  $Perm_{\bar{X}}(\mathcal{Q})$  must also be an SQS. Let  $Q'_1 = \{1, 2, \dots, 2\alpha\} \in OPT_b$ . There must exist  $Q_1 \in Perm_{\bar{X}}(\mathcal{Q})$ , such that  $Q_1 \subseteq Q'_1$ . Let  $Q'_2 = \{-2, -3, \dots, -(n - \alpha - 1), (n - \alpha), \dots, n\} \in OPT_c$ . There must exist  $Q_2 \in Perm_{\bar{X}}(\mathcal{Q})$ , such that  $Q_2 \subseteq Q'_2$ .

Now we will show that with both  $Q_1$  and  $Q_2$ ,  $Perm_{\bar{X}}(\mathcal{Q})$  cannot be an SQS. Since  $n \geq 3\alpha + 1$ , we know that  $Q_1^+ \cap Q_2^+ = \emptyset$ , which implies  $Q_1^+ \cap Q_2^+ = \emptyset$ . Furthermore,  $|Q_1 \cap Dual(Q_2)| \leq |Q'_1 \cap Dual(Q'_2)| = 2\alpha - 1$ . Thus,  $Q_1$  and  $Q_2$  satisfy neither intersection nor dual overlap, and  $Perm_{\bar{X}}(\mathcal{Q})$  is not an SQS. Contradiction.  $\square$

## 6. SQS WITH OPTIMAL PROBE COMPLEXITY

When we only optimize for probe complexity, the trivial SQS  $\{\{1\}\}$  gives the best probe complexity of 1, which is not interesting. From a practical perspective, we usually want an SQS with good availability and good probe complexity. It is reasonable to argue that ‘‘good’’ availability at least means  $Avail(\mathcal{Q}) \rightarrow 1.0$  when  $n \rightarrow \infty$ .<sup>§</sup> It is easy to show that simply to satisfy such a weak requirement would entail at least  $2\alpha$  probes:

**THEOREM 6.1.** *For any SQS  $\mathcal{Q}$  and any probe strategy  $\psi$  for  $\mathcal{Q}$ , if  $depth(\psi, C) \leq 2\alpha - 1$  for any configuration  $C$ , then  $\lim_{n \rightarrow \infty} Avail(\mathcal{Q}) \not\rightarrow 1.0$ .*

**Proof:** We consider two cases:

1. The path corresponding to  $C$  ultimately acquires a quorum  $Q$ . Since  $depth(\psi, C) \leq 2\alpha - 1$ , it must be true that  $|Q| \leq 2\alpha - 1$ .  $Q$  thus can never satisfy the dual overlap condition with another quorum, which means that  $Q$  intersects with every quorum in  $\mathcal{Q}$ . If all servers in  $Q$  fail, then no quorum can be acquired. The probability that all servers in  $Q$  fail is  $p^{|Q|} \geq p^{2\alpha-1}$ . Thus  $\lim_{n \rightarrow \infty} Avail(\mathcal{Q}) \leq 1 - p^{2\alpha-1} \not\rightarrow 1.0$ .
2. The path corresponding to  $C$  ultimately claims that no quorum can be acquired under  $C$ . Suppose the path contains successful probes from the set  $S_1$  of servers, and failed probes from the set  $S_2$  of servers. We know that  $|S_1| + |S_2| \leq 2\alpha - 1$ . Obviously, with probability of at least  $(1 - p)^{|S_1|}$ .

<sup>§</sup>Such property is formally called *Condorcet* in [12].

$p^{|S_2|} \geq (p - p^2)^{2\alpha-1}$ , no quorum from  $\mathcal{Q}$  can be acquired. As a result,  $\lim_{n \rightarrow \infty} Avail(\mathcal{Q}) \leq 1 - (p - p^2)^{2\alpha-1} \not\rightarrow 1.0$ .

$\square$

On the other hand, interestingly, we will show that even if we insist on optimal availability, there exists an SQS whose probe complexity is smaller than  $2\alpha/(1 - p)$ .

### 6.1 Probe Complexity Lower Bounds for SQS with Optimal Availability

The main challenge in deriving the lower bounds comes from the fact that the probe algorithm can be adaptive based on the results of previous probes. We first discuss how to obtain a lower bound for  $PC_e^*$ . In the same client-server context for SQS, we define a *ServerProbe* problem:

**DEFINITION 6.2.** *ServerProbe Problem: A client probes the  $n$  servers one by one according to a particular strategy, and stops after the  $i$ th probe if any of the following conditions are met (let  $pos$  be the number of successful probes and  $neg$  be the number of failed probes, where  $pos + neg = i$ ):*

1.  $pos \geq 2\alpha$ .
2.  $pos \geq n + \alpha - i$ .
3.  $neg \geq n + 1 - \alpha$ .

*Under a given configuration, the number of probes before stopping is called the total probes. The expected total probes under all configurations is called the complexity of the ServerProbe problem.*

It is important to see that even though the definition of the problem has the concept of a ‘‘strategy’’, the complexity of ServerProbe is not affected by the strategy. This is true because all servers are identical and none of the termination conditions distinguish between different servers. We will later show that the expected probe complexity of an SQS with optimal availability is at least as large as the complexity of the ServerProbe problem.

Following we compute the complexity of the ServerProbe problem. Let  $f(i) = Prob[\text{total probes} \leq i]$  and  $a(x, y) = \binom{x}{y} p^{x-y} (1 - p)^y$ . Suppose  $n \geq 3\alpha - 1$  and we consider the following cases:

1.  $0 \leq i \leq 2\alpha - 1$ . None of the three conditions can possibly be met. Thus  $f(i) = 0$ .
2.  $2\alpha \leq i \leq n - \alpha$ . It is impossible to meet the last condition of  $neg \geq n - \alpha + 1$ . The first two conditions can be merged into a single condition of  $pos \geq 2\alpha$ . Thus we have  $f(i) = \sum_{j=2\alpha}^i a(i, j)$ .
3.  $n - \alpha + 1 \leq i \leq n$ . The three conditions can be merged into the following two:  $pos \geq n + \alpha - i$  or  $pos \leq i + \alpha - (n + 1)$ . Thus we have  $f(i) = \sum_{j=0}^{i+\alpha-(n+1)} a(i, j) + \sum_{j=n+\alpha-i}^i a(i, j)$ .

The complexity of ServerProbe is then  $g(n) = \sum_{i=1}^n i \times (f(i) - f(i - 1))$ .

**LEMMA 6.3.** *Suppose  $n \geq 3\alpha - 1$ . Let  $\mathcal{Q}$  be any SQS where  $Avail(\mathcal{Q}) = Avail(OPT_a)$ . Let  $\psi$  be  $\mathcal{Q}$ 's probe strategy. Consider any configuration  $C$  and  $\psi$ 's corresponding branch of path( $\psi, C$ ):*

- *If the leaf of path( $\psi, C$ ) claims that no quorum can possibly be acquired under  $C$ , then path( $\psi, C$ ) contains at least  $n + 1 - \alpha$  failed probes.*

- If the leaf of  $\text{path}(\psi, C)$  claims that  $Q \in \mathcal{Q}$  has been acquired, then  $\text{path}(\psi, C)$  contains at least  $2\alpha$  or  $n + \alpha - \text{depth}(\psi, C)$  successful probes.

**Proof:**

- Prove by contradiction. Since  $\mathcal{Q}$  has optimal availability, from Theorem 5.8 we know that  $\mathcal{C}_\alpha \subseteq \mathcal{Q}$ . If  $\text{path}(\psi, C)$  contains  $n - \alpha$  or fewer failed probes, then it is still possible that some quorum in  $\mathcal{C}_\alpha$  is available. Thus the leaf cannot claim that no quorum can be acquired. Contradiction.
- If the leaf claims that  $Q \in \mathcal{Q}$  has been acquired, from Theorem 5.8, we know that  $|Q^+| \geq \alpha$ . If  $|Q^+| \geq 2\alpha$ , then clearly  $\text{path}(\psi, C)$  has at least  $2\alpha$  successful probes. If  $\alpha \leq |Q^+| \leq 2\alpha - 1$ , from Theorem 5.8, we know that  $\text{depth}(\psi, C) \geq |Q| \geq n + \alpha - |Q^+|$ .

□

LEMMA 6.4. *Suppose  $n \geq 3\alpha - 1$ . For any SQS  $\mathcal{Q}$  where  $\text{Avail}(\mathcal{Q}) = \text{Avail}(\text{OPT}_a)$ , we have  $PC_e^*(\mathcal{Q}) \geq g(n)$ .*

**Proof:** For any probe strategy  $\psi$  for  $\mathcal{Q}$ , we consider a ServerProbe problem using the same strategy. Under any configuration  $C$  where  $\text{depth}(\psi, C) = i$ , the total probes of the ServerProbe problem must be equal to or smaller than  $i$  (Lemma 6.3). This means that  $PC_e^*(\mathcal{Q})$  is at least the complexity of the ServerProbe problem, which is  $g(n)$ . □

We now move on to worst-case probe complexity. For  $PC_w$ , it is easy to show that the lower bound is  $n$ . This is true because from Theorem 5.8,  $\mathcal{C}_\alpha \subseteq \mathcal{Q}$  and in the worst case,  $n$  probes are already necessary to determine whether some quorum in  $\mathcal{C}_\alpha$  is alive:

LEMMA 6.5. *For any SQS  $\mathcal{Q}$  where  $\text{Avail}(\mathcal{Q}) = \text{Avail}(\text{OPT}_a)$ ,  $PC_w(\mathcal{Q}) = n$ .*

**Proof:** Construct an adversary which crashes  $n - \alpha$  servers among the first  $n - 1$  servers probed by the client. We show that the client cannot stop within or immediately after these  $n - 1$  probes. Prove by contradiction. Suppose the client returns and claims that a quorum has been acquired. Since the client does not know the state of the last server, it must be true that  $\mathcal{Q}$  contains a quorum  $Q$  accepted by some  $C \in \mathcal{C}_{\alpha-1}$ . However, this is impossible because otherwise  $\mathcal{Q}$  will not have optimal availability (Lemma 5.3). On the other hand, if the client stops and claims that no quorum can be acquired, then consider the configuration composed of the state of the first  $n - 1$  servers as seen by the client, and also an available last server. Clearly, this configuration belongs to  $\mathcal{C}_\alpha$ , and thus also belongs to  $\text{OPT}_a$  and  $As(\mathcal{Q})$  (since  $\mathcal{Q}$  has optimal availability). As a result, the configuration would accept a quorum in  $\mathcal{Q}$ . Contradiction. □

To derive a lower bound on  $PC_w^*$ , as in [6], we use Yao's theorem [16]. The theorem says that the expected time of a randomized algorithm  $A_1$  with any input distribution  $D_1$  is always bounded from below by the expected time of the best deterministic algorithm  $A_2$  on inputs coming from any distribution  $D_2$ . Thus all we need to do is to construct a difficult distribution  $D_2$  such that any  $A_2$  will perform poorly on average. Following we first cite a lemma from [6] and then prove our lower bound.

LEMMA 6.6. [6] *Consider an urn containing  $n$  elements of which  $w$  are white and  $b$  are black, and suppose elements are taken out one by one without replacement. Then the expected number of trials until obtaining the  $i$ th white element is  $\frac{i(n+1)}{w+1}$ .*

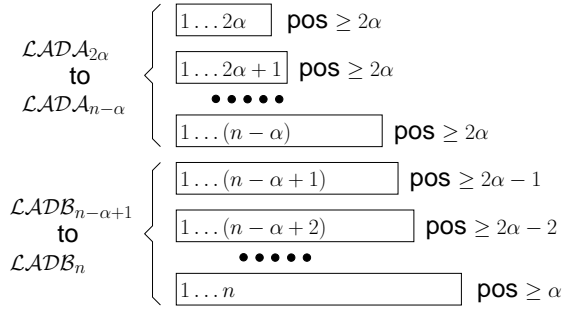


Figure 4:  $\text{OPT}_d$ .

LEMMA 6.7. *Suppose  $n \geq 3\alpha - 1$ . For any SQS  $\mathcal{Q}$  where  $\text{Avail}(\mathcal{Q}) = \text{Avail}(\text{OPT}_a)$ ,  $PC_w^*(\mathcal{Q}) \geq \frac{(n-\alpha+1)(n+1)}{n-\alpha+2} = \Omega(n)$ .*

**Proof:** Consider the distribution of all configurations in  $\mathcal{C}_{\alpha-1}$ , where each element is chosen with a probability of  $1/|\mathcal{C}_{\alpha-1}|$ . Lemma 5.3 tells us that no quorum in  $\mathcal{Q}$  can be accepted under any configuration in  $\mathcal{C}_{\alpha-1}$ . By Lemma 6.3, we know that the probe algorithm needs to observe at least  $n + 1 - \alpha$  failed probes. After each probe, the remaining servers are totally symmetric in the sense that their probability of being available is equal. Thus it does not matter which server is probed first. The problem now is exactly as in Lemma 6.6 and the expected number of probes needed is  $\frac{(n-\alpha+1)(n+1)}{n-\alpha+2}$ . □

THEOREM 6.8. *Suppose  $n \geq 3\alpha - 1$ . For any SQS  $\mathcal{Q}$ , where  $\text{Avail}(\mathcal{Q}) = \text{Avail}(\text{OPT}_a)$ , we have:*

$$\begin{aligned} PC_w(\mathcal{Q}) &= n \\ PC_e^*(\mathcal{Q}) = PC_e(\mathcal{Q}) &\geq g(n) \\ PC_w^*(\mathcal{Q}) &\geq \Omega(n) \end{aligned}$$

**Proof:** From Lemma 6.4, Lemma 6.5 and Lemma 6.7. □

## 6.2 Probe Complexity Upper Bounds for SQS with Optimal Availability

We now try to construct an optimal-availability SQS that can reach the lower bound on  $PC_e^*$ . (The lower bounds on  $PC_w$  and  $PC_w^*$  are trivially met.) The only way we can match this lower bound is to stop immediately after the conditions in Lemma 6.3 are satisfied. Just to meet these requirements, some of the quorums are already determined. It is then important to ensure that these quorums can actually constitute an SQS, and further, an SQS with optimal availability. We construct the following (Figure 4):

$$\begin{aligned} \mathcal{LAD}_i &= \{S | S \subseteq \{1, 2, \dots, i, -1, -2, \dots, -i\} \text{ and } \\ &\quad S \cap \text{Dual}(S) = \emptyset \text{ and } |S| = i\} \\ \mathcal{LADA}_i &= \{S | S \in \mathcal{LAD}_i \text{ and } |S^+| \geq 2\alpha\} \\ &\quad \text{for } 2\alpha \leq i \leq n - \alpha \\ \mathcal{LADB}_i &= \{S | S \in \mathcal{LAD}_i \text{ and } |S^+| \geq n + \alpha - i\} \\ &\quad \text{for } n - \alpha + 1 \leq i \leq n \\ \text{OPT}_d &= (\cup_{i=2\alpha}^{n-\alpha} \mathcal{LADA}_i) \cup (\cup_{i=n-\alpha+1}^n \mathcal{LADB}_i) \end{aligned}$$

It is easy to show that  $\text{OPT}_d$  is an SQS with optimal availability:

LEMMA 6.9.  $\forall S \in \mathcal{LAD}_i$  and  $\forall T \in \mathcal{LAD}_j$ ,  $i \leq j$ :

1. If  $|S^+| + |T^+| - (j - i) \geq 2\alpha$ , then either  $S^+ \cap T^+ \neq \emptyset$  or  $|S \cap \text{Dual}(T)| \geq 2\alpha$ .

2. If  $|S^+| \geq 2\alpha$ , then either  $S^+ \cap T^+ \neq \emptyset$  or  $|S \cap \text{Dual}(T)| \geq 2\alpha$ .

**Proof:**

1. If  $S^+ \cap T^+ \neq \emptyset$ , then we are done. Otherwise let  $T' = \{x \mid -i \leq x \leq i \text{ and } x \in T\}$ . Since  $|T| - |T'| = j - i$ , we know that  $|T'^+| \geq |T^+| - (j - i)$ . Given that  $S^+ \cap T'^+ = \emptyset$ , it is obvious that  $|S \cap \text{Dual}(T)| = |S \cap \text{Dual}(T')| = |S^+| + |T'^+| \geq |S^+| + |T^+| - (j - i) \geq 2\alpha$ .
2. If  $S^+ \cap T^+ \neq \emptyset$ , then we are done. Otherwise we have  $|S \cap \text{Dual}(T)| \geq |S^+| \geq 2\alpha$ .

□

**THEOREM 6.10.**  $\text{OPT}_d$  is an SQS and  $\text{Avail}(\text{OPT}_d) = \text{Avail}(\text{OPT}_a)$ .

**Proof:** First, it is clear that  $\text{OPT}_d$  is a signed set system. For any two sets  $S$  and  $T$  in  $\text{OPT}_d$ , consider the following three cases:

1.  $S \in \mathcal{LADA}_i$  and  $T \in \mathcal{LADA}_j$  for some  $i$  and  $j$ ,  $2\alpha \leq i \leq j \leq n - \alpha$ . Because  $|S^+| \geq 2\alpha$ , we know from Lemma 6.9 that either  $S^+ \cap T^+ \neq \emptyset$  or  $|S \cap \text{Dual}(T)| \geq 2\alpha$ .
2.  $S \in \mathcal{LADA}_i$  and  $T \in \mathcal{LADB}_j$  for some  $i$  and  $j$ ,  $2\alpha \leq i \leq n - \alpha$  and  $n - \alpha + 1 \leq j \leq n$ . Because  $|S^+| \geq 2\alpha$ , we know from Lemma 6.9 that either  $S^+ \cap T^+ \neq \emptyset$  or  $|S \cap \text{Dual}(T)| \geq 2\alpha$ .
3.  $S \in \mathcal{LADB}_i$  and  $T \in \mathcal{LADB}_j$  for some  $i$  and  $j$ ,  $n - \alpha + 1 \leq i \leq j \leq n$ . We have  $|S^+| + |T^+| - (j - i) \geq (n + \alpha - i) + (n + \alpha - j) - (j - i) = 2n + 2\alpha - 2j \geq 2n + 2\alpha - 2n = 2\alpha$ . We now know from Lemma 6.9 that either  $S^+ \cap T^+ \neq \emptyset$  or  $|S \cap \text{Dual}(T)| \geq 2\alpha$ .

Thus,  $\text{OPT}_d$  is an SQS. Finally, it is obvious that  $\text{OPT}_a = \mathcal{LADB}_n \subseteq \text{As}(\text{OPT}_d)$ , and from Corollary 5.5, we have  $\text{Avail}(\text{OPT}_d) = \text{Avail}(\text{OPT}_a)$ . □

Our probe strategy for  $\text{OPT}_d$  simply probes the servers one by one according to increasing indexes. It is easy to see that the achieved expected probe complexity is exactly  $g(n)$ :

**THEOREM 6.11.** If  $n \geq 3\alpha - 1$ , then:

$$\begin{aligned} \text{Avail}(\text{OPT}_d) &= \text{Avail}(\text{OPT}_a) \\ \text{PC}_e^*(\text{OPT}_d) &\leq g(n) < 2\alpha/(1-p) \end{aligned}$$

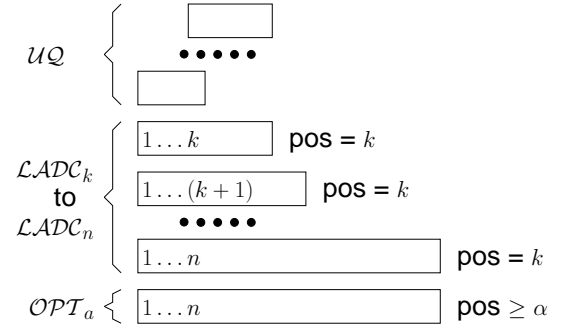
## 7. SQS WITH OPTIMAL LOAD

### 7.1 Lower Bounds

Different from previous load definitions, our more practical definition captures the load induced by wasted probes. Such definition introduces extra challenge because we can no longer compute load based on distributions on the quorums. To address such challenge, since one major goal/benefit of SQS is high availability, we focus on SQS whose availability is larger than 0.5. We first obtain lower bound given that some quorum is acquired. The overall load is then lower bounded within a 0.5 factor. We use the following notations:

$$\begin{aligned} \text{LoadA}(\mathcal{Q}) &= \text{Load}(\mathcal{Q}) \text{ given that a quorum is acquired} \\ \text{LoadF}(\mathcal{Q}) &= \text{Load}(\mathcal{Q}) \text{ given that no quorum can be} \\ &\quad \text{acquired} \\ \text{Load}(\mathcal{Q}) &= \text{LoadA}(\mathcal{Q}) \times \text{Avail}(\mathcal{Q}) + \\ &\quad \text{LoadF}(\mathcal{Q}) \times (1 - \text{Avail}(\mathcal{Q})) \end{aligned}$$

Similarly, we define  $\text{PCA}_e^*(\mathcal{Q})$  and  $\text{PCF}_e^*(\mathcal{Q})$  for probe complexity.



**Figure 5: Composition of  $\mathcal{UQ}$  and  $\text{OPT}_a$ .**

**LEMMA 7.1.** When  $\text{Avail}(\mathcal{Q}) \geq 0.5$ , we have  $\text{Load}(\mathcal{Q}) \geq 0.5\text{LoadA}(\mathcal{Q})$  and  $\text{PC}_e^*(\mathcal{Q}) \geq 0.5\text{PCA}_e^*(\mathcal{Q})$ .

**LEMMA 7.2.** For any SQS  $\mathcal{Q}$ , suppose the smallest quorum size in  $\mathcal{Q}$  is  $x$ , then  $\text{PCA}_e^*(\mathcal{Q}) \geq x$ .

Our lower bound on load is exactly the same as for UQS [11], with a similar proof. This means that if we are only concerned with load, then SQS does not provide any benefits:

**THEOREM 7.3.** For any SQS  $\mathcal{Q}$ , suppose the smallest quorum size in  $\mathcal{Q}$  is  $x$ , then  $\text{LoadA}(\mathcal{Q}) \geq \max(x/n, 1/x)$ .

**Proof:** Suppose  $\mathcal{Q} = \{Q_1, Q_2, \dots, Q_m\}$ , and without loss of generality, suppose  $|Q_1| = x$ . Let  $w_i$  be the probability that  $Q_i$  is acquired, given the fact that  $\mathcal{Q}$  is available. Thus we have  $\sum_{i=1}^m w_i = 1$  (since some quorum is acquired). From the definition of load, we have  $n \cdot \text{LoadA}(\mathcal{Q}) \geq \sum_{i=1}^m w_i |Q_i| \geq |Q_1| = x$ , which means  $\text{LoadA}(\mathcal{Q}) \geq x/n$ . Now consider the sum of the load on the servers in  $Q_1$ . We have:

$$\begin{aligned} |Q_1| \cdot \text{LoadA}(\mathcal{Q}) &\geq \sum \text{load on all servers in } Q_1 \\ &\geq \sum_{Q_i \cap Q_1 \neq \emptyset} w_i + \sum_{Q_i \cap Q_1 = \emptyset} 2\alpha w_i \geq 1 \end{aligned}$$

□

**COROLLARY 7.4.** For any SQS  $\mathcal{Q}$  where  $\text{Avail}(\mathcal{Q}) \geq 0.5$ , we have  $\text{Load}(\mathcal{Q}) \geq 0.5/\sqrt{n}$  and  $\text{Load}(\mathcal{Q}) \geq 0.25/\text{PC}_e^*(\mathcal{Q})$ .

This shows that SQS has similar tradeoff between load and probe complexity as UQS.

### 7.2 Composition of UQS and $\text{OPT}_a$

To approach the lower bound on load, we compose certain UQS with  $\text{OPT}_a$  to obtain new SQS. The nice property of composition is that the resulting SQS has the availability of  $\text{OPT}_a$ , and the load and probe complexity of the UQS.

**DEFINITION 7.5.** Consider any UQS  $\mathcal{UQ}$  over the universe of  $\{1, 2, \dots, k\}$  ( $k \leq n$ ), where the size of any quorum in  $\mathcal{UQ}$  is at least  $2\alpha$ . Define the composition of  $\mathcal{UQ}$  and  $\text{OPT}_a$  (denoted as  $\mathcal{UQ} + \text{OPT}_a$ ) to be the signed set system  $\mathcal{Q}$  where:

$$\begin{aligned} \mathcal{LADC}_i &= \{S \mid S \in \mathcal{LAD}_i \text{ and } |S^+| = k\} \text{ for } k \leq i \leq n \\ \mathcal{Q} &= \mathcal{UQ} \cup (\cup_{i=k}^n \mathcal{LADC}_i) \cup \text{OPT}_a \end{aligned}$$

Figure 5 illustrates the composition technique. From now on, when we use the notation  $\mathcal{UQ} + \text{OPT}_a$ , we imply that  $\mathcal{UQ}$  satisfies the conditions in the above definition.

$$\begin{aligned}
& \text{Prob}[\text{non-intersection}] \\
&= \text{Prob}[\text{non-intersection and } Q_1 \in \mathcal{U}\mathcal{Q} \text{ and } Q_2 \in \mathcal{U}\mathcal{Q}] + \text{Prob}[\text{non-intersection and } Q_1 \in \mathcal{S}\mathcal{Q} \text{ and } Q_2 \in \mathcal{U}\mathcal{Q}] + \\
&\quad \text{Prob}[\text{non-intersection and } Q_1 \in \mathcal{U}\mathcal{Q} \text{ and } Q_2 \in \mathcal{S}\mathcal{Q}] + \text{Prob}[\text{non-intersection and } Q_1 \in \mathcal{S}\mathcal{Q} \text{ and } Q_2 \in \mathcal{S}\mathcal{Q}] \\
&\leq \text{Prob}[\text{non-intersection} \mid (Q_1 \in \mathcal{U}\mathcal{Q} \text{ and } Q_2 \in \mathcal{S}\mathcal{Q})] \cdot \text{Prob}[Q_1 \in \mathcal{U}\mathcal{Q} \text{ and } Q_2 \in \mathcal{S}\mathcal{Q}] + \\
&\quad \text{Prob}[\text{non-intersection} \mid (Q_1 \in \mathcal{S}\mathcal{Q} \text{ and } Q_2 \in \mathcal{U}\mathcal{Q})] \cdot \text{Prob}[Q_1 \in \mathcal{S}\mathcal{Q} \text{ and } Q_2 \in \mathcal{U}\mathcal{Q}] + \delta \\
&= \varepsilon^{2\alpha} \cdot (\text{Prob}[Q_1 \in \mathcal{U}\mathcal{Q} \text{ and } Q_2 \in \mathcal{S}\mathcal{Q}] + \text{Prob}[Q_1 \in \mathcal{S}\mathcal{Q} \text{ and } Q_2 \in \mathcal{U}\mathcal{Q}]) + \delta \\
&\leq 2\varepsilon^{2\alpha}
\end{aligned}$$

**Figure 6: Putting the four cases of non-intersection together.**

**THEOREM 7.6.** *For any  $\mathcal{U}\mathcal{Q}$ ,  $\mathcal{U}\mathcal{Q} + \mathcal{OPT}_a$  is an SQS.*

In order to preserve the load and probe complexity of  $\mathcal{U}\mathcal{Q}$  in  $\mathcal{U}\mathcal{Q} + \mathcal{OPT}_a$ , our probe algorithm first probes the quorums in  $\mathcal{U}\mathcal{Q}$ , and then moves on to other quorums in  $\mathcal{U}\mathcal{Q} + \mathcal{OPT}_a$ . If the availability of  $\mathcal{U}\mathcal{Q}$  is reasonably high (with respect to the value of  $k$ ), the possibility of probing other quorums is low. Thus the resulting probe complexity and load will be dominated by the probe complexity and load of  $\mathcal{U}\mathcal{Q}$ . On the other hand, as long as we try all quorums in  $\mathcal{U}\mathcal{Q} + \mathcal{OPT}_a$ , the availability of  $\mathcal{OPT}_a$  is preserved.

**THEOREM 7.7.** *For any  $\mathcal{U}\mathcal{Q}$  and  $\mathcal{Q} = \mathcal{U}\mathcal{Q} + \mathcal{OPT}_a$ :*

$$\begin{aligned}
\text{Load}(\mathcal{Q}) &\leq \text{Load}(\mathcal{U}\mathcal{Q}) + (1 - \text{Avail}(\mathcal{U}\mathcal{Q})) \\
\text{PC}_e^*(\mathcal{Q}) &\leq \text{PC}_e^*(\mathcal{U}\mathcal{Q}) + k/(1-p) \cdot (1 - \text{Avail}(\mathcal{U}\mathcal{Q})) \\
\text{Avail}(\mathcal{Q}) &= \text{Avail}(\mathcal{OPT}_a)
\end{aligned}$$

**Proof:** By definition, there exists a probe algorithm  $A$  for  $\mathcal{U}\mathcal{Q}$  that can achieve load of  $\text{Load}(\mathcal{U}\mathcal{Q})$  and probe complexity of  $\text{PC}_e^*(\mathcal{U}\mathcal{Q})$ . We construct a probe algorithm for  $\mathcal{Q}$  as following:

1. Use  $A$  on the servers  $\{1, 2, \dots, k\}$ . If a quorum in  $\mathcal{U}\mathcal{Q}$  is acquired, return.
2. Probe the servers one by one from 1 to  $n$ . If a quorum in  $(\cup_{i=k}^n \mathcal{LADC}_k)$  is acquired, return.
3. At this point, all servers have been probed. If some quorum in  $\mathcal{OPT}_a$  has been acquired, return. Otherwise claim that no quorum can be acquired.

If the client returns in the first step, then the load on any server must be no larger than  $\text{Load}(\mathcal{U}\mathcal{Q})$ . If the client continues to the second step (which happens with probability of  $1 - \text{Avail}(\mathcal{U}\mathcal{Q})$ ), then the load on any node can be at most 1. Thus we have  $\text{Load}(\mathcal{Q}) \leq \text{Load}(\mathcal{U}\mathcal{Q}) + (1 - \text{Avail}(\mathcal{U}\mathcal{Q}))$ . For probe complexity, if the client returns in the first step, the expected probe complexity is  $\text{PC}_e^*(\mathcal{U}\mathcal{Q})$ . If the client continues to the second step, then it will stop as soon as having  $k$  successful probes (or having probed all servers). A simple calculation can show that the expected number of probes is upper bounded by  $k/(1-p)$  for arbitrary  $n$ . Finally, it is obvious that  $\mathcal{OPT}_a \subseteq \text{As}(\mathcal{Q})$  and Corollary 5.5 tells us that  $\text{Avail}(\mathcal{Q}) = \text{Avail}(\mathcal{OPT}_a)$ .  $\square$

**COROLLARY 7.8.** *For any  $\mathcal{U}\mathcal{Q}$ , if  $1 - \text{Avail}(\mathcal{U}\mathcal{Q}) = O(1/k)$ , then:*

$$\begin{aligned}
\text{Load}(\mathcal{U}\mathcal{Q} + \mathcal{OPT}_a) &= O(\text{Load}(\mathcal{U}\mathcal{Q})) \\
\text{PC}_e^*(\mathcal{U}\mathcal{Q} + \mathcal{OPT}_a) &= O(\text{PC}_e^*(\mathcal{U}\mathcal{Q})) \\
\text{Avail}(\mathcal{U}\mathcal{Q} + \mathcal{OPT}_a) &= \text{Avail}(\mathcal{OPT}_a)
\end{aligned}$$

**Proof:** Probe complexity and availability are trivial from Theorem 7.7. For load, notice that the lower bound on  $\text{Load}(\mathcal{U}\mathcal{Q})$  is  $\Omega(1/\sqrt{k})$ , which already dominates  $O(1/k)$ .  $\square$

Different from our previous SQS constructions,  $\mathcal{U}\mathcal{Q} + \mathcal{OPT}_a$  uses a potentially randomized and adaptive probe strategy (because of the probe strategy on  $\mathcal{U}\mathcal{Q}$ ), and Theorem 4.2 does not apply. Following we prove that the probability of non-intersection is still properly bounded under such probe strategy:

**THEOREM 7.9.** *For any  $\mathcal{U}\mathcal{Q}$  and  $\mathcal{Q} = \mathcal{U}\mathcal{Q} + \mathcal{OPT}_a$ , consider two clients using our previous probe strategy on  $\mathcal{Q}$ , and let “non-intersection” denote the event that both clients acquire some quorum, but they do not intersect. Then  $\text{Prob}[\text{non-intersection}] \leq 2\varepsilon^{2\alpha}$ .*

**Proof:** Let  $Q_1$  be the random variable denoting the quorum acquired by the first client and  $S_1$  be the probed servers. Similarly define  $Q_2$  and  $S_2$ . Let  $\mathcal{S}\mathcal{Q} = \mathcal{Q} - \mathcal{U}\mathcal{Q}$ . We trivially have  $\text{Prob}[\text{non-intersection and } Q_1 \in \mathcal{U}\mathcal{Q} \text{ and } Q_2 \in \mathcal{U}\mathcal{Q}] = 0$ . Now consider the case where  $Q_1 \in \mathcal{U}\mathcal{Q}$  and  $Q_2 \in \mathcal{S}\mathcal{Q}$ . Because the second client must have probed all of the first  $k$  servers, it must have probed all servers in  $Q_1$ . In order for the two clients not to intersect, there must be mismatches on all servers in  $Q_1$ . Since  $|Q_1| \geq 2\alpha$ , we have  $\text{Prob}[\text{non-intersection} \mid (Q_1 \in \mathcal{U}\mathcal{Q} \text{ and } Q_2 \in \mathcal{S}\mathcal{Q})] \leq \varepsilon^{2\alpha}$ . Similarly, we can prove that  $\text{Prob}[\text{non-intersection} \mid (Q_1 \in \mathcal{S}\mathcal{Q} \text{ and } Q_2 \in \mathcal{U}\mathcal{Q})] \leq \varepsilon^{2\alpha}$ .

For the case where both  $Q_1$  and  $Q_2$  belong to  $\mathcal{S}\mathcal{Q}$ , notice that  $\mathcal{S}\mathcal{Q}$  is an SQS itself. We construct a probe strategy for  $\mathcal{S}\mathcal{Q}$  using the second and the third step of the probe strategy for  $\mathcal{Q}$ . Clearly, this new probe strategy is deterministic and non-adaptive. From Theorem 4.2, we know that the probability (denoted as  $\delta$ ) that two clients of  $\mathcal{S}\mathcal{Q}$  acquire quorums in  $\mathcal{S}\mathcal{Q}$  but do not intersect is upper bounded by  $\varepsilon^{2\alpha}$ . Define the *global state* to be the vector describing the exact state (i.e.,  $(+, +)$ ,  $(+, -)$ ,  $(-, +)$  or  $(-, -)$ ) of all servers when probed by two clients<sup>¶</sup>. Consider any global state where i) two clients of  $\mathcal{Q}$  acquire quorums  $Q_1$  and  $Q_2$ , and ii)  $Q_1 \in \mathcal{S}\mathcal{Q}$  and  $Q_2 \in \mathcal{S}\mathcal{Q}$ , and iii) the two clients do not intersect. Then under the same global state, the two clients of  $\mathcal{S}\mathcal{Q}$  will also acquire  $Q_1$  and  $Q_2$ , and they will not intersect either. So we have  $\text{Prob}[\text{non-intersection and } Q_1 \in \mathcal{S}\mathcal{Q} \text{ and } Q_2 \in \mathcal{S}\mathcal{Q}] \leq \delta$ . Finally, Figure 6 puts the above four cases together and proves that  $\text{Prob}[\text{non-intersection}] \leq 2\varepsilon^{2\alpha}$ .  $\square$

### 7.3 Composition with the Paths UQS

Now we compose the Paths [10, 11] UQS with  $\mathcal{OPT}_a$ .

**THEOREM 7.10.** **[10, 11]**<sup>||</sup> *Let  $\mathcal{PH}(l)$  denote the Paths quorum*

<sup>¶</sup>We cannot use the notion of configuration here because configuration only describes the server states as observed by a single client.

<sup>||</sup>A trivial alteration of the probe algorithm in [10] is needed to make the results hold under our new load definition.

system with  $k = 2l^2 + 2l + 1$  servers. Then:

$$\begin{aligned} \text{Load}(\mathcal{PH}(l)) &= O(1/l) \\ PC_e^*(\mathcal{PH}(l)) &= O(l) \\ 1 - \text{Avail}(\mathcal{PH}(l)) &= O(e^{-l}) \end{aligned}$$

It is trivial to show that the smallest quorum size in  $\mathcal{PH}(l)$  is  $l$ .

**COROLLARY 7.11.** *Let  $2\alpha \leq l \leq (\sqrt{2n-1} - 1)/2$ , then:*

$$\begin{aligned} \text{Load}(\mathcal{PH}(l) + \mathcal{OPT}_\alpha) &= O(1/l) \\ PC_e^*(\mathcal{PH}(l) + \mathcal{OPT}_\alpha) &= O(l) \\ \text{Avail}(\mathcal{PH}(l) + \mathcal{OPT}_\alpha) &= \text{Avail}(\mathcal{OPT}_\alpha) \end{aligned}$$

Setting different values for  $l$  yields SQS constructions that reach the optimal tradeoff between load and probe complexity while preserving optimal availability.

## 8. CONCLUSIONS

Motivated by the need for highly available and low probe complexity quorum systems in the Internet, this paper proposes *signed quorum systems* (SQS). SQS provides probabilistic intersection guarantee and utilizes the *independent mismatch* property in today's Internet. We show that our optimal SQS construction  $\mathcal{OPT}_\alpha$  is available as long as any  $\alpha$  servers are available, and simultaneously has a probe complexity of at most  $2\alpha/(1-p)$ . These properties qualitatively improve upon traditional quorum systems, where even the optimal construction requires  $(n+1)/2$  available servers. Our composition technique further shows that SQS can decouple availability from load and probe complexity, and optimal availability can now be achieved under most load and probe complexity values.

## 9. ACKNOWLEDGMENTS

I would like to thank Phillip Gibbons, Dahlia Malkhi, David Peleg, Adrian Perrig, Dazhi Wang, Avishai Wool, and Peng Yin for helpful discussion on various issues related to this paper. I also thank the anonymous reviewers for their detailed feedbacks, which significantly improved this paper.

## 10. REFERENCES

- [1] D. Andersen, H. Balakrishnan, F. Kaashoek, and R. Morris. Resilient Overlay Networks. In *Proceedings of the 18th Symposium on Operating Systems Principles (SOSP)*, October 2001.
- [2] D. Barbara and H. Garcia-Molina. The Reliability of Voting Mechanisms. *IEEE Trans. Comput.*, pages 1197–1208, October 1987.
- [3] R. Bazzi. Planar quorums. *Theoretical Computer Science*, 243:243–268, 2000.
- [4] H. Garcia-Molina and D. Barbara. How to Assign Votes in a Distributed System. *Journal of the ACM*, October 1985.
- [5] D. K. Gifford. Weighted Voting for Replicated Data. In *Proceedings of the 7th SOSP*, 1979.
- [6] Y. Hassin and D. Peleg. Average probe complexity in quorum systems. In *Proceedings of the ACM Symposium of Principles of Distributed Computing*, 2001.
- [7] R. Holzman, Y. Marcus, and D. Peleg. Load balancing in quorum systems. *SIAM Journal on Discrete Mathematics*, 10(2):223–245, 1997.
- [8] D. Malkhi and M. Reiter. Byzantine Quorum Systems. In *Proceedings of the 29th ACM Symposium on Theory of Computing*, pages 569–578, May 1997.
- [9] D. Malkhi, M. Reiter, A. Wool, and R. Wright. Probabilistic Quorum Systems. *The Information and Computation Journal*, 170(2), November 2001.
- [10] M. Naor and U. Wieder. Scalable and Dynamic Quorum Systems. In *Proceedings of the ACM Symposium of Principles of Distributed Computing*, 2003.
- [11] M. Naor and A. Wool. The Load, Capacity, and Availability of Quorum Systems. *SIAM Journal on Computing*, 27(2):423–447, 1998.
- [12] D. Peleg and A. Wool. The Availability of Quorum Systems. *Information and Computation*, pages 210–223, 1995.
- [13] D. Peleg and A. Wool. How to Be an Efficient Snooper, or the Probe Complexity of Quorum Systems. In *Proceedings of the ACM Symposium of Principles of Distributed Computing*, 1996.
- [14] S. Savage, T. Anderson, A. Aggarwal, D. Becker, N. Cardwell, A. Collins, E. Hoffman, J. Snell, A. Vahdat, G. Voelker, and J. Zahorjan. Detour: A Case for Informed Internet Routing and Transport. *IEEE Micro*, 19(1), January 1999.
- [15] R. H. Thomas. A Majority Consensus Approach to Concurrency Control for Multiple Copy Databases. *ACM Transactions on Database Systems*, 4:180–209, 1979.
- [16] A. Yao. Probabilistic Computations: Towards a Unified Measure of Complexity. In *Proceedings of the 17th Annual Symposium on Foundations of Computer Science*, pages 222–227, 1977.
- [17] H. Yu. Overcoming the Majority Barrier in Large-Scale Systems. In *Proceedings of the 17th International Symposium on Distributed Computing (DISC)*, October 2003.
- [18] H. Yu and A. Vahdat. The Costs and Limits of Availability for Replicated Services. In *Proceedings of the 18th ACM Symposium on Operating Systems Principles (SOSP)*, October 2001.