A Random Projection Approach to Subscription Covering Detection in Publish/Subsribe Systems

Duc A. Tran
Department of Computer Science
University of Massachusetts, Boston
Boston, MA 02125
Email: duc@cs.umb.edu

Thinh Nguyen
Department of Electrical and Computer Engineering
Oregon State University
Corvallis, OR 97331
Email: thinhq@eecs.oregonstate.edu

***Abstract*— Subscription covering detection is useful to improving the performance of any publish/subscribe system. However, an exact solution to querying coverings among a large set of subscriptions in high dimension is computationally too expensive to be practicable. Therefore, we are interested in an approximate approach. We focus on spherical subscriptions and propose a solution based on random projections. Our complexities are substantially better than that of the exact approach. The proposed solution can potentially find exact coverings with a success probability 100% asymptotically approachable.**

I. Introduction

E-commerce is a popular marketplace involving more and more producers and consumers everyday. An effective mechanism for them to find each other is desirable. Of importance is a publish/subscribe system (a.k.a. pub/sub) that enables consumers to subscribe to their interests and producers to publish their products, asynchronously without their knowing each other, so that subscription-product matching can be made quickly.

The simplest design for a pub/sub is to index every subscription and product in one central place, where matchings can be determined locally (e.g., [1]). This centralized approach obviously does not scale with the system size.

For scalability, a pub/sub should be designed as a distributed network involving broker nodes, who share the load of storing, distributing, and matchmaking subscriptions with products [2]–[6]. Underlying a distributed pub/sub are routing mechanisms for subscription propagation and product propagation. Typically, during the propagation of a subscription, each receiving broker records the subscription coupled with the sending broker into a routing table. During a product's propagation, each visited broker searches its routing table to find the matching subscriptions and forwards the product information to the corresponding brokers in the routing table, to find the way to matching consumers.

The routing table at any broker may grow quickly as new subscriptions continue to enter the network. Consequently, searching a routing table for subscriptions matching a product may be a time-consuming process. To keep the routing table small, an effective way is to let a broker forward a subscription during its propagation only if it is not already covered by an existing subscription locally stored [2], [5], [7]. For example, consider a broker A that already stores a subscription $S(X) \doteq (company = *, stock = [\$400, \$500])$ for a consumer X who is interested in companies with stock values between $\$400$ and $\$500$. Suppose that broker A later receives a new subscription $S(Y) \doteq (company = *, stock = [\$420, \$440])$ for a consumer Y who is interested in companies with stock values between $\$420$ and $\$440$. Broker A would not need to forward the new subscription to other brokers because A is already waiting for products that match $S(X)$, which will automatically contain the results for $S(Y)$. In this example, there are two advantages of stop forwarding $S(Y)$: (1) the traffic due to forwarding $S(Y)$ is avoided, and (2) the routing tables at other brokers are not getting larger.

Although potential for improving a pub/sub's performance, the detection of subscription coverings at each broker, if not done efficiently, may turn into a burdening process, especially when there are already many subscriptions, each having many attributes. First, it may take long time to find a covering subscription for every new subscription. Second, when an existing subscription is removed, because the products matching it may no longer arrive, all subscriptions covered by this subscription need to be found and subsequently be forwarded further. To illustrate this case, we revisit the previous example. If

$S(X)$ is removed, the products matching $S(X)$ may never be available. Therefore, broker A needs to find $S(Y)$ and then advertise it further so that the products matching $S(Y)$ can find A. The process of finding all covered subscriptions of a given one may also be time-consuming.

Thus, for a pub/sub network of many subscriptions with many attributes, we need an efficient data structure for organizing the subscriptions so that fast algorithms for detection of subscription coverings can be derived. The challenge is that, despite a few attempts [7]–[10], no *exact* solution to the subscription covering problem can remain efficient if the subscription dimensionality is high [10]–[12]. By "exact", we mean that the covering detection algorithm always finds coverings exactly. Therefore, if a new subscription is found to be covered by an existing subscription, this covering is always correct and the new subscription is never forwarded further.

Our research in this paper is to seek an *approximate* solution. By "approximate", we mean that the covering detection algorithm may return false coverings. Thus, the broker may redundantly forward a subscription even when it is covered by an existing subscription. This redundant forwarding creates some traffic but does not affect the correctness of the system. The advantage of an approximate solution is that it may lower both time and place complexities compared to the exact approach. The accuracy of an approximate solution is the capability to avoid redundant forwarding, which should be maximized.

Recently, approximate solutions to the subscription covering problem have been proposed for rectangular subscriptions [9], [10]. In this paper, we assume spherical subscriptions. First, we show that it is not trivial to approximate a spherical subscription with a rectangular one. We then propose a novel approximation approach based on random projections, in which redundant forwarding occurs with a probability exponentially approaching zero as we increase the number of projections. We propose a simple implementation based on layered range tree to index the subscriptions. This implementation, for n subscriptions in d dimensions, in the worst case results in $O(\log^{2k-1} n)$ query time and $O(n \log^{2k-1} n)$ storage, where k is any integer less than d. These bounds are much better than that for the exact approach.

The remainder of this paper is structured as follows. In Section II, we formalize the problem and present some possible solutions as well as our motivation for using random projections. The proposed data structures and algorithms are described in Section III. Related work is discussed in Section V. We conclude our paper with pointers to our future work in Section VI.

II. Preliminaries

When a product is first generated, we call that an event and represent it by a point in $\mathbf{R}^d$ where d is the number of attributes associated with the event. There are various ways to represent a subscription. Most work models a subscription as a d-dimension rectangle. In our work, we represent each subscription by a d-dimension sphere (s, r) centered at point $s \in \mathbf{R}^d$ with radius $r \in \mathbf{R}^+$. For example, consider a video surveillance sensor network where sensor cameras are deployed in many places to detect criminals. In this application, a subscription specified by a sample photo of a wanted criminal is submitted to the network with the purpose that similar pictures are detected and their locations reported. For image retrieval, a spherical query is used more often than a rectangular query to define the limit of image results.

An event x and a subscription (s, r) is said to *match* each other, denoted by $x \in (s, r)$ iff $\parallel x - s \parallel \leq r$. Also, a subscription ($s$, r) is said to *cover* another subscription (s', r'), denoted by (s, r) $\supseteq$ (s', r'), if the sphere (s, r) contains the sphere (s', r'). Conversely, the latter is said to be *covered* by the former, which is denoted by (s', r') $\subseteq$ (s, r). It is easy to prove the following equivalence:

Proposition 2.1: $(s, r) \supseteq (s', r') \Leftrightarrow \parallel s - s' \parallel \leq r - r'$.

We focus on a single broker that stores a set of n subscriptions $\{(s_1, r_1), (s_2, r_2), ..., (s_n, r_n)\}$. Our problem is to devise an *efficient* method that allows for *fast* detection of covering relationships in this subscription set. Specifically, we need a dynamic data structure to organize the subscriptions so that:

1) The cost, time and space, to construct the data structure and update it upon subscription insertions or deletions is low
2) Finding at least a subscription that covers a given subscription is fast
3) Finding all the subscriptions covered by a given subscription is fast

A. Contributions

When a new subscription enters the broker, the simplest way to find a subscription covering it is to scan the subscription set sequentially and check the covering condition on each visited subscription. With a time complexity of $O(nd)$, which is linear in n, this brute-force

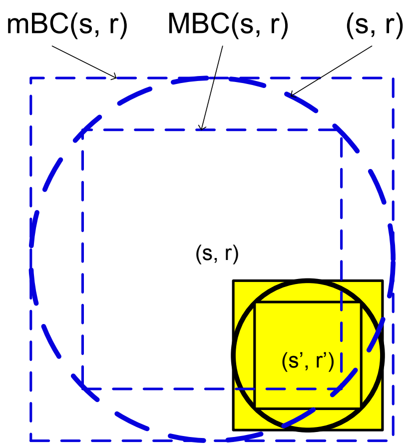

Fig. 1. Approximating spheres with minimum bounding and maximum bounded cubes

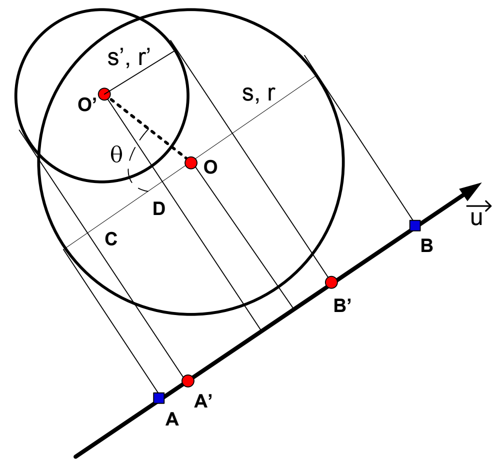

Fig. 2. Projections of two spheres onto a random unit vector

approach is too expensive for a large publish/suscribe system where a broker may contain millions of subscriptions. A complexity sub-linear of n is more desirable.

Another solution is to approximate a spherical subscription by a rectangular subscription because solutions for the latter are widely available. If we denote by $mBC(s,r)$ the minimal bounding d-dimension axis-parallel cube and $MBC(s,r)$ the maximal bounded d-dimension axis-parallel cube of the subscription (s,r), we have the proposition below:

Proposition 2.2: The following causalities are true:

1) $mBC(s, r) \not\supseteq mBC(s', r') \Rightarrow (s, r) \not\supseteq (s', r')$
2) $MBC(s, r) \supseteq MBC(s', r') \Rightarrow (s, r) \supseteq (s', r')$

Therefore, we can conclude on the covering relationship between two spherical subscriptions based on the covering conditions on the minimal bounding cubes and maximal bounded cubes as in Proposition 2.2. The only case that no accurate conclusion about the covering relationship can be made is when $mBC(s, r) \supseteq mBC(s', r')$ and $MBC(s, r) \not\supseteq MBC(s', r')$ (see Figure 1).

This ambiguous case occurs if at least one vertex of $MBC(s',r')$ is outside $MBC(s,r)$. Assuming uniform distribution for the subscriptions (for both centers and radii), the probability for the ambiguous case can approximately be

$$1-\left(\frac{volume(MBC(s,r))}{volume(mBC(s,r))}\right)^d = 1-\frac{(\sqrt{2}r)^d}{(2r)^d} = 1-\left(\frac{1}{\sqrt{2}}\right)^d$$

which is highly likely if d is a large number.

We propose a novel solution that approximates a spherical subscription by its projection on a set of random uni-dimension vectors rather than by its bounding and bounded cubes. We show that not only finding subscription coverings in the new projection space is more efficient, but the approximation accuracy can approach 100% asymptotically. Our motivation for using random projections is explained next.

B. Why Random Projections?

Suppose that u is a random unit vector in R^d. The projection of a subscription (s,r) on this vector is the interval $u(s,r) = [\langle u,s\rangle - r, \langle u,s\rangle + r]$ (where $\langle .,. \rangle$ is the inner product). For example, in Figure 2, the projection of the sphere (s,r) is the interval $AB = [\langle u,s\rangle - r, \langle u,s\rangle + r]$.

It is always true that, if subscription (s,r) covers subscription (s',r'), we also have $u(s,r) \supseteq u(s',r')$. To assess the capability of u in detecting covering relationships, we are interested in the conditional probability that $u(s,r) \supseteq u(s',r')$ given $(s,r) \not\supseteq (s',r')$.

This case is illustrated in Figure 2. Without loss of generality, suppose that $r \geq r'$. Firstly, we have

$$\begin{aligned} & AB = u(s,r) \supseteq u(s',r') = A'B' \\ \Leftrightarrow\; & \overline{OC} \leq r \\ \Leftrightarrow\; & \overline{OD} + r' \leq r \\ \Leftrightarrow\; & \| \langle u, s-s' \rangle \| \leq r - r' \\ \Leftrightarrow\; & \| s-s' \| | \cos\theta | \leq r - r' \\ \Leftrightarrow\; & \frac{r'-r}{\| s-s' \|} \leq \cos\theta \leq \frac{r-r'}{\| s-s' \|} \end{aligned}$$

Assuming that θ is uniformly distributed between 0 and 2π, the conditional probability that the last inequality occurs is

$$Pr\left\{\frac{r'-r}{\parallel s-s'\parallel} \le \cos\theta \le \frac{r-r'}{\parallel s-s'\parallel}\right\} \quad (1)$$
$$= 1-\frac{2}{\pi}\arccos\left(\frac{r-r'}{\parallel s-s'\parallel}\right) \quad (2)$$

According to Maclaurin series,

$$\arccos z \;=\; \frac{\pi}{2}-\left(z^1+\left(\frac{1}{2}\right)\frac{z^3}{3}+\left(\frac{1\cdot 3}{2\cdot 4}\right)\frac{z^5}{5}+...\right)$$

Let $z=\frac{r-r'}{\|s-s'\|}$. Because $(s,r) \not\supseteq (s',r')$, combining with Proposition 2.1, we must have $0 \le z < 1$. Therefore,

$$\arccos z \simeq \frac{\pi}{2}-z$$

This is a tight approximation because, even when z is as large as 0.9, the next terms after term z of the Maclaurin series quickly become very small. For example,

$$\left(\frac{1}{2}\right)\frac{z^3}{3}=0.1215$$
$$\left(\frac{1\cdot 3}{2\cdot 4}\right)\frac{z^5}{5}=0.0442$$

The probability (2) can be approximated as follows:

$$1-\frac{2}{\pi}\arccos z \;\simeq\; 1-\frac{2}{\pi}\left(\frac{\pi}{2}-z\right)=\left(\frac{2}{\pi}\right)\frac{r-r'}{\parallel s-s'\parallel}$$

We note that this probability is conditional on $(s,r) \not\supseteq (s',r')$. Hence, we obtain the following proposition.

Proposition 2.3: Consider the projection of the subscription/event space onto a random unit vector. Assume that the subscriptions follow a uniform distribution. Given two subscriptions (s,r) and (s',r') that do not have a covering relationship, the probability that this projection results in a covering relationship is close to

$$\left(\frac{2}{\pi}\right)\frac{\mid r-r'\mid}{\parallel s-s'\parallel}$$

Proposition 2.3 suggests that we can detect subscription coverings by finding coverings among the subscriptions' projections on a random unit vector, which is less demanding in both time and space. In terms of accuracy using this approach, the probability that a found covering is false nears zero if the original subscriptions have similar radii ($r \approx r'$) or inter-distant centers (large $\parallel s-s'\parallel$).

This probability, however, may be as large as $\frac{2}{\pi} \approx 0.64$ in the worst case, which is rather high. We, therefore, propose to use more than one random projection. The following proposition provides a bound on the accuracy of covering detection based on multiple random projections.

Proposition 2.4: Consider i.i.d. projections of the subscription/event space onto k random unit vectors $\{u_1,u_2,...,u_k\}$. Assume that the subscriptions follow a uniform distribution. Given two subscriptions (s,r) and (s',r'):

- *If at least one projection u_i finds no covering relationship between $u_i(s,r)$ and $u_i(s',r')$, it must be true that no covering relationship exists between (s,r) and (s',r')*
- *If every projection u_i finds that $u_i(s,r) \supseteq u_i(s',r')$, the probability that $(s,r) \supseteq (s',r')$ is closely at least*

$$1-\left(\frac{2}{\pi}\right)^k$$

Proof: The first conclusion is correct because if $(s,r) \supseteq (s',r')$, it is always true that $u_i(s,r) \supseteq u_i(s',r')$ for any i. The second conclusion is a consequence of Proposition 2.3. ■

This proposition implies that it is highly effective to detect subscription coverings by projecting the subscription space onto multiple random uni-dimensions. The error probability approaches zero exponentially as the number of projections increases. We present our data structures and algorithms in the next section.

III. Data Structures and Algorithms

In this section, we propose how to organize the subscriptions so that covering detection based on random projections can be implemented.

Firstly, in the preprocessing phase when the broker first starts, it generates the following matrix

$$\begin{pmatrix} u_1 \\ u_2 \\ ... \\ u_k \end{pmatrix} := \begin{pmatrix} u_1^1 & u_1^2 & ... & u_1^d \\ u_2^1 & u_2^2 & ... & u_2^d \\ ... & ... & ... & ... \\ u_k^1 & u_k^2 & ... & u_k^d \end{pmatrix} \quad (3)$$

where each row vector u_i is a d-dimension unit vector i.i.d. randomly generated. To maximize their mutual independence, if these vectors are not orthogonal, we can use a Gram-Schmidt process [13] to orthogonalize them. For each subscription (s,r), we compute a set of

k uni-dimension intervals, each being a projection of the subscription on a vector u_i:

$$\begin{aligned} u_1(s,r) &= [\langle u_1, s\rangle - r, \langle u_1, s\rangle + r] \\ u_2(s,r) &= [\langle u_2, s\rangle - r, \langle u_2, s\rangle + r] \\ &... \\ u_k(s,r) &= [\langle u_k, s\rangle - r, \langle u_k, s\rangle + r] \end{aligned}$$

These intervals form a k-dimension rectangle in the $(u_1, u_2, ..., u_k)$-coordinate space:

$$RECT(s,r) = u_1(s,r) \times u_2(s,r) \times ... \times u_k(s,r)$$

We refer to this rectangle by a "k-projection rectangle", or simply "projection rectangle" when the dimension is obvious.

Proposition 2.4 implies the following:

1) If $RECT(s,r) \not\supseteq RECT(s',r')$, then $(s,r) \not\supseteq (s',r')$
2) If $RECT(s,r) \supseteq RECT(s',r')$, then $(s,r) \supseteq (s',r')$ with a probability, roughly, at least $1 - (2/\pi)^k$

Therefore, approximating the covering relationship between two subscriptions by that between their projection rectangles is highly accurate. In addition, according to [14], a rectangle in k dimensions can be considered a point in $2k$ dimensions: $R = [a_1, b_1] \times [a_2, b_2] \times ... \times [a_k, b_k] \equiv p_R = (a_1, -b_1, a_2, b_2, ..., a_k, -b_k) \in \mathbf{R}^{2k}$ so that

- To find all rectangles that cover R is equivalent to the orthogonal range query that finds all the points in the range $[a_1, \infty] \times [-b_1, \infty] \times [a_2, \infty] \times [-b_2, \infty] \times ... \times [a_k, \infty] \times [-b_k, \infty]$ of the $2k$-dimension space. These points are called the *dominating* points of the point p_R
- To find all rectangles that are covered by R is equivalent to the orthogonal range query that finds all the points in the range $[-\infty, a_1] \times [-\infty, -b_1] \times [-\infty, a_2] \times [-\infty, -b_2] \times ... \times [-\infty, a_k] \times [-\infty, -b_k]$ of the $2k$-dimension space. These points are called the *dominated* points of the point p_R

Therefore, we propose to index each subscription (s,r) by a $2k$-dimension point:

$$idx(s,r) = \begin{pmatrix} \langle u_1, s\rangle - r \\ -\langle u_1, s\rangle - r \\ \langle u_2, s\rangle - r \\ -\langle u_2, s\rangle - r \\ ... \\ \langle u_k, s\rangle - r \\ -\langle u_k, s\rangle - r \end{pmatrix}$$

We then store these indices using a data structure that supports orthogonal range searching in high dimension. We use a $2k$-dimension layered range tree for simple indexing implementation (see [15], chapter 5). Therefore, we obtain the following properties:

- Building time for n subscriptions: $O(n \log^{2k-1} n)$
- Update time to insert a new subscription or delete an existing subscription: $O(\log^{2k-1} n)$
- Time to query coverings: $O(\log^{2k-1} n + m)$ where m is the number of coverings reported
- Space complexity: $O(n \log^{2k-1} n)$

Using this data structure, to decide whether to stop forwarding a new subscription (s,r) when it arrives at a broker, we follow the algorithm below:

1) Compute $idx(s,r)$ and insert it to the index tree
2) Search the index tree to find *one* subscription (s',r') such that $idx(s',r')$ dominates $idx(s,r)$
   a) IF no such (s',r') is found, forward (s,r)
   b) ELSE check the original covering condition
      i) IF $\| s - s' \| \leq r - r'$, stop forwarding (s,r)
      ii) ELSE forward (s,r)
3) END

This algorithm uses the index tree to quickly find a subscription covering the new subscription in the index space. This subscription (s',r'), if any, could be a true covering subscription (case 2b(i)) or a false covering subscription (case 2b(ii)). In either case, we use the original covering condition $\| s - s' \| \leq r - r'$ (meaning $(s',r') \supseteq (s,r)$) to verify. Therefore, we never withhold a subscription if it is not covered by any subscription.

We may sometimes forward a new subscription even when it is covered by some existing subscription. This case occurs if the subscription (s',r') returned in Step (2) does not actually cover (s,r) (case 2b(ii)) but another subscription not returned by Step (2) does. However, this case is rare with the probability less than $(2/\pi)^k$. If it occurs, it just creates some redundant traffic but does not affect the correctness of the pub/sub system. The time complexity to process a new subscription is

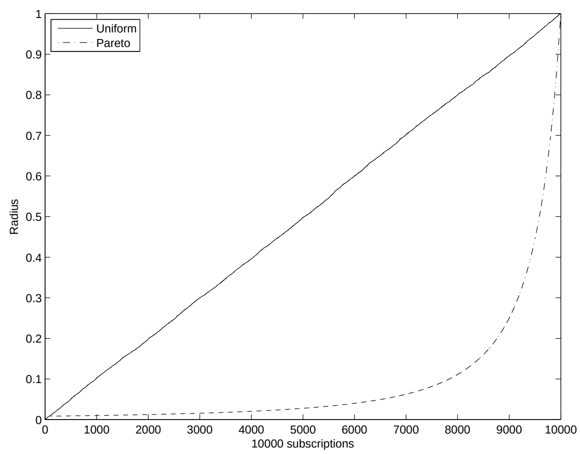

Fig. 3. Subscription radii: The uniform model and Pareto model

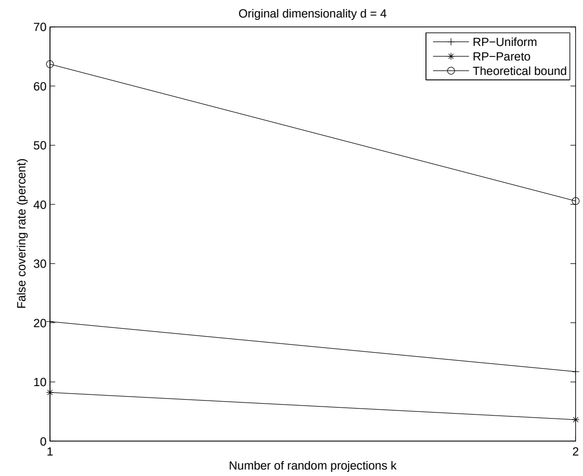

Fig. 4. $d = 4$, $k = \{1,\, 2\}$

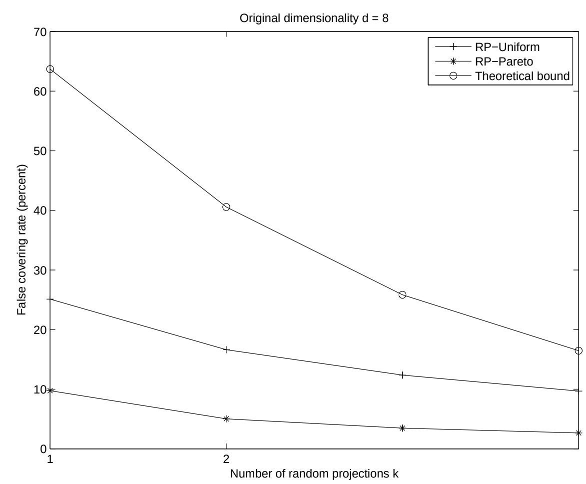

Fig. 5. $d = 8$, k = {1, 2, 3, 4}

$O(\log^{2k-1} n)$ to find (s', r') plus $O(d)$ to check the original covering condition; hence, $O(\log^{2k-1} n + d)$.

Cancellation of an existing subscription (s, r) is simple. First, we compute $idx(s, r)$ and remove it from the index tree. Second, we search the index tree for all subscriptions (s', r') such that $idx(s', r')$ is dominated by $idx(s, r)$ and advertise (s', r') forward based on an underlying subscription routing protocol (which we assume to exist). The time complexity to cancel a subscription is therefore $O(\log^{2k-1} n + m)$, where m is the number of dominated points found.

IV. Simulation Study

We conducted a simulation-based study to evaluate performance of the random projection approach. Because the time and space complexity of this approach can be obtained theoretically, our performance study was focused on its effectiveness; i.e., the probability of error in covering detection. In Section III, we have obtained theoretically an approximate upper bound of this error for the case that subscriptions follow a uniform distribution. In this section, we present the actual results obtained from our simulation.

We generated 10,000 spherical subscriptions. The centers of these subscriptions were generated uniformly in random as points in the d-dimension unit cube. The radii were chosen in the range (0, 1) according to two distribution models: the uniform distribution and the Pareto distribution. The latter one, also known as the 80-20 rule, represents the case that most subscriptions are specific (i.e., small radii), only a few being expansive (i.e., large radii). The subscriptions radii for all 10,000 subscriptions in both models are drawn in Figure 3.

For each given subscription, assumed to be one of the generated subscriptions, our technique was used to find its coverings among the other 9999 subscriptions. Because our theoretical work guarantees that no false non-covering is possible, the metric evaluated was the frequency of false coverings; i.e., that of the case that the a covering detected is not true covering.

We studied the proposed technique for various dimensionalities $d \in \{4, 8, 12, 16, 20\}$, for each case under various numbers of random projections $k \in \{1, 2, ..., d/2\}$. The results are plotted in Figures 4,5,6,7,8, where we also include the approximate upper-bound $(2/\pi)^k$ on the probability of covering error mentioned in Proposition 2.4. These figures demonstrate the following:

- In general, the frequency of error is low (e.g., less than 20% for $k \geq 2$). Furthermore, the error is less for the Pareto model than the uniform model

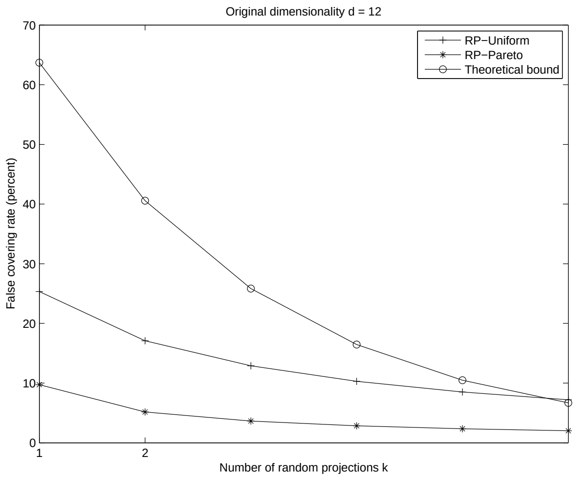

Fig. 6. $d = 12$, $k = \{1, 2, ..., 6\}$

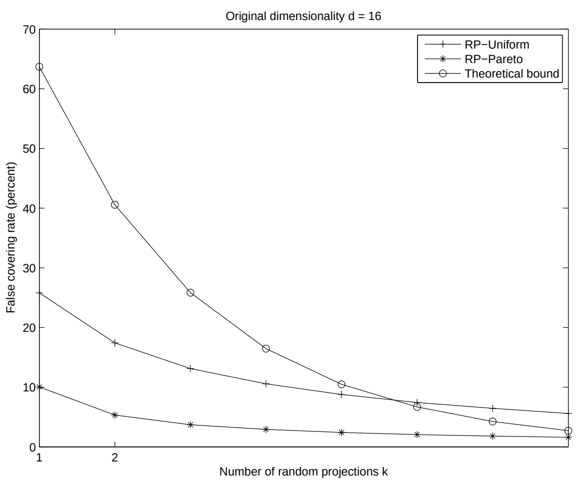

Fig. 7. $d = 16$, $k = \{1, 2, ..., 8\}$

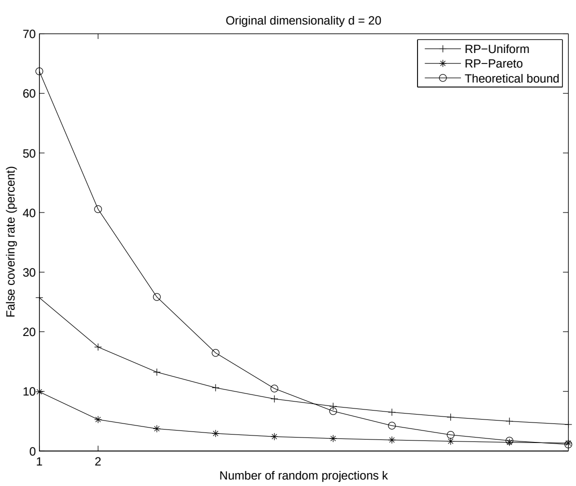

Fig. 8. $d = 20$, $k = \{1, 2, ..., 10\}$

(more than twice less). In other words, if the sizes of the subscriptions are skewed, the technique is more accurate. This result is encouraging because in practice the 80-20 rule should represent the user preferences better than the uniform model.

- The actual frequency of error is significantly less than the theoretical bound for the case of low dimensionality (small d) or few random projections (small k). The theoretical bound is used to prove the asymptotic property of our technique. This simulation result shows that in practice the technique can actually perform better than that theoretically analyzed.
- When d is high (16 or 20) and k also high ($k \geq 6$), the technique performs worse for the uniform case than the $(2/\pi)^k$ upper-bound. This is possible because our theoretical analysis is approximate. However, when this happens, the error frequency is already small. For instance, when $d = 20$ and $k = 6$, the error for the uniform case is less than 10% and for the Pareto case less than 3%).

In summary, the simulation study substantiates the high approximation accuracy of the random projection approach.

V. Related Work

The design of a distributed publish/subscribe network typically involves two main tasks. The first task is to design a communication architecture for efficiently disseminating subscriptions and events over the network. The second task is to design mechanisms for efficient storage of subscriptions and fast matching of subscriptions with events.

The simplest communication architecture is the broadcast approach, in which a subscription traverses a broadcast tree to reach all broker nodes. This approach however has its disadvantages. First, not all the broker nodes receive events satisfying a given subscription, thus it is redudant to store a subscription at every broker node. Second, broadcasting incurs an extremly high communication cost, which is not desirable for a large publish/subscribe network or a network with limited resources such as a sensor network.

A much better option for the communication architecture is to replicate a subscription in a set of select nodes where satisfying events may likely be sent to. Most techniques of this option employ a Distributed Hash Table (DHT) [16]–[18]. A DHT is used to send a subscription or event to a node that is the result of the hash function. The goal is that the node storing a subscription and that receiving a satisfactory event are either identical or within a proximixity of each other. Scribe [19] uses Pastry [18] to map a subscription to a node based on topic hashing, thus those subscriptions and events with the same topic are mapped to the same node. Meghdoot [20] transforms each subscription into a multidimensional point and employs the CAN DHT structure [16] to hash this point to get the node that will store this subscription. Rather than CAN and Pastry, [21] uses the Chord DHT [17] instead. A technique that can be used atop any such DHT structure was proposed in [22].

The aforementioned communication architectures are built on top of an existing DHT overlay. [23] proposes a decentralized architecture based on an unstructured overlay. This technique, called Sub-2-Sub, uses an epidemic-based algorithm [24] to automatically cluster together subscriptions for similar events. Therefore, an event is delivered to only nodes that have relevant subscriptions.

In addition to the communication task, it is important to have a mechanism that allows for efficient storage of subscriptions and fast matching algorithms. Several structures already exist, including the Matching Tree [25], [26], Binary Decision Diagram [27], and SIFT [1], [28]. These early designs are focused mainly on the matching aspect (i.e., matching an event against the subscriptions).

When the size (number and dimension) of the subscriptions is large, the task of maintaining a structure for the subscriptions is not trivial. We need not only fast matching algorithms, but also convenient ways to add to or remove subscriptions from the storage. The set of subscriptions can be simplified by merging overlapping subscriptions or finding covering relationships among them. Instead of disseminating subscriptions separately, similar ones can be merged to reduce the number of subscriptions and thus the resultant traffic [7], [29], [30]. Subscription covering [7]–[10], [31] is our paper's topic. By not forwarding subscriptions that are already covered by an earlier forwarded subscriptions, we can also reduce the size of the subscriptions as well as the traffic involved. Most techniques [7], [8], [31] attempt to find subscription covering exactly, thus inefficient for a large number of subscriptions in high dimension. The works in [9], [10] are similar to our work in the aspect that they also aim at finding coverings approximately without affecting the overall correctness of the system. [9] uses a Monte Carlo Sampling approach to check the covering condition quickly. [10] maps the covering condition between the subscriptions in high dimension to the dominance condition between points on a uni-dimension Space Filling Curve, so that coverings can be found faster. Compared to these two techniques, the uniqueness of our solution is two-fold. First, our approach based on Random Projections is unique. Second, while [9], [10] assume rectangular subscriptions, we address spherical subscriptions and have shown that directly approximating them with rectangular ones does not lead to good accuracy.

VI. Conclusions

Subscription covering is potentially very useful for improving the performance of any pub-sub system. It helps reduce not only the size of any broker's routing table, but also the network traffic due to subscription/event propagation. Overusing it, however, creates additional burden that may adversely slow down the entire system.

The best way to utilize subscription coverings is to use it only when it remains efficient. The current solutions are aimed at finding the exact coverings, which are inefficient for large pub-sub networks. We have proposed a novel solution that finds the coverings approximately but with a high accuracy. We project the subscription/event space onto a few random unit vectors, where covering detection in the projection space is much more efficient. As an approximation of the exact approach, a broker may sometimes waste bandwidth to forward a new subscription even when it is covered by an existing subscription. However, a desirable property of our approximate approach is that this case occurs rarely with a probability exponentially approaching zero as more random projections are used. Our research is unique also because it is the first to address spherical subscriptions.

Our future work includes extending the random-projection framework to the case of rectangular subscriptions and implementing a system based on the proposed technique.

Acknowledgment

This work is funded in part by the NSF Grant CNS-0615055. The authors would like to thank the NSF for this support.

References

[1] F. Fabret, H. A. Jacobsen, F. Llirbat, J. Pereira, K. A. Ross, and D. Shasha, "Filtering algorithms and implementation for very fast publish/subscribe systems," in *SIGMOD '01: Proceedings of the 2001 ACM SIGMOD international conference on Management of data*. New York, NY, USA: ACM Press, 2001, pp. 115–126.

[2] G. Cugola, E. D. Nitto, and A. Fuggetta, "The jedi event-based infrastructure and its application to the development of the opss wfms," *IEEE Trans. Softw. Eng.*, vol. 27, no. 9, pp. 827–850, 2001.

[3] G. Muhl, "Large-scale content-based publish/subscribe systems," Ph.D. dissertation, Darmstadt University of Technology, 2002.

[4] Y. Zhao, D. Sturman, and S. Bhola, "Subscription propagation in highly-available publish/subscribe middleware," in *Middleware '04: Proceedings of the 5th ACM/IFIP/USENIX international conference on Middleware*. New York, NY, USA: Springer-Verlag New York, Inc., 2004, pp. 274–293.

[5] A. Carzaniga, M. J. Rutherford, and A. L. Wolf, "A routing scheme for content-based networking," in *Proceedings of IEEE INFOCOM 2004*, Hong Kong, China, March 2004.

[6] R. Chand and P. Felber, "Xnet: A reliable content based publish subscribe system," in *IEEE Symposium on Reliable Distributed Systems*, Florianopolis, Brazil, October 2004, pp. 264–273.

[7] G. Li, S. Hou, and H.-A. Jacobsen, "A unified approach to routing, covering and merging in publish/subscribe systems based on modified binary decision diagrams," in *ICDCS '05: Proceedings of the 25th IEEE International Conference on Distributed Computing Systems (ICDCS'05)*. Washington, DC, USA: IEEE Computer Society, 2005, pp. 447–457.

[8] P. Triantafillou and A. Economides, "Subscription summarization: A new paradigm for efficient publish/subscribe systems," in *ICDCS '04: Proceedings of the 24th International Conference on Distributed Computing Systems (ICDCS'04)*. Washington, DC, USA: IEEE Computer Society, 2004, pp. 562–571.

[9] A. M. Ouksel, O. Jurca, I. Podnar, and K. Aberer, "Efficient probabilistic subsumption checking for content-based publish/subscribe systems." in *ACM/IFIP/USENIX 7th International Middleware Conference*, 2006, pp. 121–140.

[10] Z. Shen and S. Tirthapura, "Approximate covering detection among content-based subscriptions using space filling curves," in *IEEE International Conference on Distributed Computing Systems*, Toronto, Canada, June 2007.

[11] B. Chazelle, "Lower bounds for orthogonal range searching: I. the reporting case," *J. ACM*, vol. 37, no. 2, pp. 200–212, 1990.

[12] J. JáJá, C. W. Mortensen, and Q. Shi, "Space-efficient and fast algorithms for multidimensional dominance reporting and counting." in *15th International Symposium on Algorithms and Computation*, 2004, pp. 558–568.

[13] G. H. Golub and C. F. V. Loan, *Matrix computations (3rd ed.)*. Baltimore, MD, USA: Johns Hopkins University Press, 1996.

[14] H. Edelsbrunner and M. H. Overmars, "On the equivalence of some rectangle problems." *Inf. Process. Lett.*, vol. 14, no. 3, pp. 124–127, 1982.

[15] M. de Berg, M. van Kreveld, M. Overmars, and O. Schwarzkopf, *Computational Geometry: Algorithms and Applications*, 2nd ed. Springer-Verlag, 2000.

[16] S. Ratnasamy, P. Francis, M. Handley, R. Karp, and S. Shenker, "A scalable content addressable network," in *ACM SIGCOMM*, San Diego, CA, August 2001, pp. 161–172.

[17] I. Stoica, R. Morris, D. Karger, M. Kaashock, and H. Balakrishman, "Chord: A scalable peer-to-peer lookup protocol for internet applications," in *ACM SIGCOMM*, San Diego, CA, August 2001, pp. 149–160.

[18] A. Rowstron and P. Druschel, "Pastry: Scalable, distributed object location and routing for large-scale peer-to-peer systems," in *IFIP/ACM International Conference on Distributed Systems Platforms (Middleware)*, Heidelberg, Germany, November 2001, pp. 329–350.

[19] M. Castro, P. Druschel, A. Kermarrec, and A. Rowstron, "SCRIBE: A large-scale and decentralized application-level multicast infrastructure," *IEEE Journal on Selected Areas in communications (JSAC)*, vol. 20, no. 8, pp. 1489–1499, 2002.

[20] A. Gupta, O. D. Sahin, D. Agrawal, and A. E. Abbadi, "Meghdoot: content-based publish/subscribe over p2p networks," in *Middleware '04: Proceedings of the 5th ACM/IFIP/USENIX international conference on Middleware*. New York, NY, USA: Springer-Verlag New York, Inc., 2004, pp. 254–273.

[21] W. W. Terpstra, S. Behnel, L. Fiege, A. Zeidler, and A. P. Buchmann, "A peer-to-peer approach to content-based publish/subscribe," in *DEBS '03: Proceedings of the 2nd international workshop on Distributed event-based systems*. New York, NY, USA: ACM Press, 2003, pp. 1–8.

[22] I. Aekaterinidis and P. Triantafillou, "Internet scale string attribute publish/subscribe data networks," in *CIKM '05: Proceedings of the 14th ACM international conference on Information and knowledge management*. New York, NY, USA: ACM Press, 2005, pp. 44–51.

[23] S. Voulgaris, E. Rivire, A.-M. Kermarrec, and M. van Steen, "Sub-2-sub: Self-organizing content-based publish subscribe for dynamic large scale collaborative networks," in *5th Int'l Workshop on Peer-to-Peer Systems (IPTPS 2006)*, 2006.

[24] S. Voulgaris, D. Gavidia, and M. van Steen, "Cyclon: Inexpensive membership management for unstructured p2p overlays," *Journal of Network and Systems Management*, vol. 13, no. 2, June 2005.

[25] J. Gough and G. Smith, "Efficient recognition of events in a distributed system," in *In Proceedings of the 18th Australasian Computer Science Conference*, Adelaide, Australia, Feb 1995.

[26] M. K. Aguilera, R. E. Strom, D. C. Sturman, M. Astley, and T. D. Chandra, "Matching events in a content-based subscription system," in *PODC '99: Proceedings of the eighteenth annual ACM symposium on Principles of distributed computing*. New York, NY, USA: ACM Press, 1999, pp. 53–61.

[27] A. Campailla, S. Chaki, E. M. Clarke, S. Jha, and H. Veith, "Efficient filtering in publish-subscribe systems using binary decision," in *International Conference on Software Engineering*, 2001, pp. 443–452.

[28] T. W. Yan and H. Garcia-Molina, "The sift information dissemination system," *ACM Trans. Database Syst.*, vol. 24, no. 4, pp. 529–565, 1999.

[29] G. Muhl, L. Fiege, and A. P. Buchmann, "Filter similarities in content-based publish/subscribe systems," in *ARCS '02: Proceedings of the International Conference on Architecture of Computing Systems*. London, UK: Springer-Verlag, 2002, pp. 224–240.

[30] A. Crespo, O. Buyukkokten, and H. Garcia-Molina, "Query merging: Improving query subscription processing in a multicast environment," *IEEE Transactions on Knowledge and Data Engineering*, vol. 15, no. 1, pp. 174–191, 2003.

[31] Z. Shen, S. Tirthapura, and S. Aluru, "Indexing for subscription covering in publish-subscribe systems," in *ISCA 18th International Conference on Parallel and Distributed Computing Systems (PDCS)*, 2005.