An Efficient Eigenvalue-based P2P XML Routing Framework

Qiang Wang
Univ. of Waterloo, Canada
q6wang@uwaterloo.ca

M. Tamer Özsu
Univ. of Waterloo, Canada
tozsu@uwaterloo.ca

Abstract

Many emerging applications that use XML are distributed, usually over large peer-to-peer (P2P) networks on the Internet. The deployment of an XML query shipping system over P2P networks requires a specialized synopsis to capture XML data in routing tables. In this paper, we propose a novel eigenvalue-based routing synopsis for deployment over unstructured P2P networks. Based on well-established Cauchy's Interlacing Theorem, our approach employs multiple eigenvalues as routing synopsis and facilitates query routing process with computationally-inexpensive range inequality checking. Through extensive experiments, we demonstrate the effectiveness of our approach.

1 Introduction

Nowadays, XML is widely used in massively distributed environments such as Internet and P2P networks. For example, XML is used to define description and exchange languages (*e.g.,* SOAP[1] and WSDL[2]) in Web service applications; distributed native XML data management systems are developed to manage semi-structured scientific data [10]. Recently, XML is also employed in emerging peer-to-peer (P2P) applications (*e.g.,* JXTA[3], Groove Network[4] and others) for data exchange.

Since XML data are deployed at multiple distributed sites, queries are usually shipped to data, and query processing is completed at local sites without additional communications, thus XML-aware query routing becomes crucial. This is especially important for in-network query processing over P2P systems, which primarily depend on content-based query routing mechanisms at application level.

[1]http://www.w3.org/TR/soap
[2]http://www.w3.org/TR/wsdl
[3]http://www.jxta.org
[4]http://www.groove.net

In P2P literature, XML routing problem has been studied from different directions. Distributed inverted lists are built to catalog XML paths (for XML queries or data) over specific distributed hash table (DHT) [5, 3]. However, the approaches are not applicable over general P2P networks (*e.g.,* unstructured P2P networks). Second, *XML routing synopses* have been developed to capture characteristics of XML data and supply correct (*i.e.,* non-false-negative) XML routing in unstructured P2P networks. For instance, [8] develops hierarchical Bloom-filters to encode XML documents for routing purposes. Although Bloom-filter itself provides compact representation over a set of data, the size of the overall encoding of documents tightly depends on the structure (*e.g.,* depth) of the document. Thus this synopsis may potentially become huge, especially when XML documents contain recursive structures. [13] proposes a configurable structure-based routing synopsis based on *Simulation transformation* over original XML documents [2]. Unfortunately, the routing synopsis may degrade to the original XML document in certain cases, such that the synopsis size may still be proportional to the document size.

Eigenvalue-based synopsis has been recently developed [14] for XML indexing. In contrast to previous approaches, this synopsis maps each document to fixed-length eigenvalue features; moreover, it employs inexpensive range inequality checking over eigenvalues to test whether queries can be satisfied by documents. Unfortunately, this approach is not directly applicable in P2P XML routing systems because: (1) in P2P systems, queries are various and some may correspond to multiple eigenvalues, thus the mere use of two eigenvalues (and checking with a single range inequality) may not be sufficiently discriminative for query routing tasks; (2) the approach does not support core XPath axes (*e.g.,* wild cards or ancestor-descendant axes) directly; instead, it relies on an additional structural-join process to handle these axes, which may incur additional computation costs; and (3) since the approach targets at XML indexing problem, it does not touch on specific problems that exist in XML routing systems, such as how to aggregate routing synopses among multiple peers in overlay networks.

In this work, we exploit *Cauchy Interlacing Theorem* (refer to Section 3 for the definition) to enable the employment of multiple eigenvalues as routing synopsis, which tends to be more discriminative regarding routing process, meanwhile remaining compact and efficient.

Briefly, in our approach, the routing synopsis of a specific XML document consists of a constrained number of eigenvalues over the anti-symmetric adjacency matrix of the document. The number of eigenvalues per routing entry (*i.e.,* per peer) is bounded by $min(|V|, \mathcal{O}(|q|)$, where $|V|$ is the number of distinct element names of the document and $|q|$ bounds the maximum number of eigenvalues of queries, which is observable during long-term query processing. Correspondingly, query routing is realized through range inequality checking over eigenvalues. This approach has the following advantages over previous ones: first, in contrast to [8, 13], the routing synopsis is compact in that each peer corresponds to a bounded number of eigenvalue features, irrelevant to structure or size of the original document. Second, query routing process is based on range inequality checking, without employing complicated (and expensive) computation between queries and documents, such as the Simulation-relationship computation [13]. Third, our approach is designed to support wild cards and ancestor-descendant axis without adding significant computation overhead (see Section 3.5), which has not been achieved in previous feature-based approaches [14, 8][5]. Finally, without depending on tree structures, our approach is general to be applicable over complex XML documents with recursive structures, and is easily integrated with potential graph-structured synopses over original XML documents.

Briefly, the contributions of this work include:

- To our knowledge, this work is the first to employ eigenvalue features in building XML routing synopsis for (unstructured) P2P networks. The synopsis is highly compact, consisting of a bounded number of eigenvalues for each routing entry, and is efficient in pruning irrelevant queries for routing process through inexpensive range inequality checking.
- We address the aggregation of eigenvalue features of the data hosted on multiple peers, which enables eigenvalue-based query routing process in hierarchical P2P overlay networks.
- Finally, through experiments, we demonstrate the effectiveness of our approach; moreover, we show that, over hierarchical overlay networks, an enhanced eigenvalue-based synopsis is more effective for XML routing than previous approaches.

[5]Without loss of generality, we regard that [8] employs Bloom-filter "features" of XML data.

The remainder of this paper is organized as follows. We review the mostly related work in Section 2. Section 3 addresses the eigenvalue-based routing synopsis and details the XML routing algorithm in hierarchical overlay networks. We evaluate the system performance in Section 4 and conclude in Section 5.

2 Related Work

XML is widely supported in P2P networks for storage, description, and exchange purposes. For example, ActiveXML [11] is focused on distributed XML data management and query processing. XPeer [12] describes a self-organizing XML P2P data management system that supplies various query processing functionalities. When these systems are deployed in large-scale P2P networks, the performance of XML routing becomes crucial. In this work, we developed compact and efficient routing synopsis, which can be used to facilitate query routing in such systems.

Various approaches have been proposed to support XML query routing. Distributed inverted lists are built to catalog XML paths over specific DHT protocols [5, 3]. However, these work does not fit unstructured P2P networks that are widely used in practice (*e.g.,* Gnutella[6] and KazAa[7]). Instead, our approach targets at unstructured (especially ultra-peer-based or hierarchical) P2P networks, and is sufficiently general to be applied in other XML routing systems.

[8] employs hierarchical Bloom-filters to encode XML documents for routing purposes. Although Bloom-filter itself provides compact representation over a set of data, the size of the overall encoding depends on the structure of the XML document (*e.g.,* the depth of the documents), which may potentially become huge when XML documents are deep or contain recursive structures. [13] proposes a configurable structure-based routing synopsis based on Simulation transformation [2]. Unfortunately, the worst-case size of routing synopsis is proportional to the document size.

Eigenvalue-based XML synopsis is recently proposed in [14]. Briefly, a Hermitian matrix is generated

[6]http://www.gnutella.com
[7]http://www.kazaa.com

over the anti-symmetric adjacency matrix of a specific XML document; then the minimum and maximum eigenvalues of the Hermitian matrix are designated as features of the document in XML indexing. Since only one range inequality (*i.e.,* over two eigenvalues) is employed, the approach may not be sufficiently discriminative for handling queries that correspond to multiple eigenvalues. In this work, we consider Cauchy Interlacing Theorem (abbreviated as *Interlacing Theorem* in the remainder) and employ multiple eigenvalues as routing synopsis to improve its discriminative in P2P query routing.

3 Eigenvalue-based Routing Algorithm

3.1 Background

Given an arbitrary directed graph $G = (V, E)$, we define an anti-symmetric adjacency matrix M over G as follows: each row (or column) of the matrix corresponds to a unique vertex $v \in V$; for each edge $(v_i, v_j) \in E$, we specify matrix entry $M[i][j] = 1$ and $M[j][i] = -1$ (for anti-symmetric purpose); other entries are all set to 0.

It has been proved in [14] that, when M is anti-symmetric, iM is a Hermitian matrix, where $i = \sqrt{-1}$ is the *imaginary unit.* Then we can exploit *Interlacing Theorem* [4] over iM to facilitate query routing process. For completeness, we restate the Theorem as below.

Theorem 1 (Interlacing Theorem)**.** *Let A be a Hermitian matrix with eigenvalues $\lambda_1,...\lambda_n$ and B be one of its principal sub-matrices; let B have eigenvalues $\mu_1,...\mu_m$. Then the inequalities $\lambda_{n-m+i} \leq \mu_i \leq \lambda_i$ ($i = 1,...,m$) hold.*

Briefly, we exploit Theorem 1 to facilitate XML routing as follows. Consider an arbitrary XML document D and an XPath query q that contains branching and parent-child axis (other axes are addressed in Section 3.5). It is feasible to facilitate the test of whether q is potentially satisfied by D by checking whether the range inequalities (imposed by Theorem 1) hold over the corresponding eigenvalues. For clarity, denote by μ_i the eigenvalues of q, and λ_i the counterpart of D, the inequality "interlacing" relationship is illustrated in Figure 1.

For example, consider an XML instance and an XPath query together with their respective anti-symmetric matrices, as shown in Figure 2 (a) and (b). For conciseness, we entitle each row (and column) of matrix with the first character of the corresponding element name (*e.g.,* "C" for "Conference"). It is direct to compute the eigenvalues of the query, which

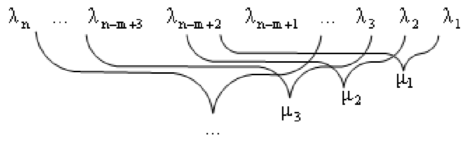

Figure 1. The Interlacing Property

is $\{\mu_1 = \sqrt{2}\}$, and the counterpart for the XML document is $\{\lambda_1 = 2, \lambda_2 = 1\}$. Since the query path is contained in the XML document, it is not surprising that $\lambda_2 \leq \mu_1 \leq \lambda_1$ holds.

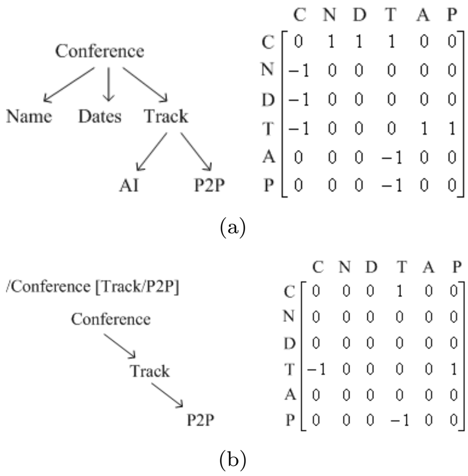

Figure 2. Eigenvalue Features

3.2 Routing Algorithm

Without loss of generality, we assume that all peers agree on a consistent XML schema, which can be obtained through schema integration mechanisms [1, 7]. Then each peer is able to define dimension schemes (*i.e.*, the element names corresponding to rows and columns) consistently and compute (comparable) eigenvalues individually. The time complexity of eigenvalue computation is $\mathcal{O}(n^3)$, where n is the cardinality of the matrix. Since the communication cost is dominant in modern large-scale P2P networks and the computation is over the structures (excluding attribute values) of documents, we regard the computation cost affordable.

Note that, in this work, we generate anti-symmetric adjacency matrix based on distinct element names. However, the matrix may also encode edges so as to capture more structural properties of original XML documents. Since original XML documents may potentially contain a large number of edges, which leads to a large matrix, we can first transform the original document into its Simulation-based synopsis [13], without introducing more false-positive with respect to Branching Path Queries [6][8]. For space constraints, we leave the discussion of edge-based eigenvalue features to future work.

Since each XPath query only contains branching predicates and parent-child axis, we can directly transform a query to a tree (or graph), such that the matrix generation and eigenvalue computation are equivalent to the counterparts over XML documents. We address how to support other axes (*e.g.,* wild cards and ancestor-descendant axis) in Section 3.5.

Now, given eigenvalues of specific XML documents and queries, we employ the inequalities imposed in the Interlacing Theorem to check whether the query structures are potentially contained in the documents. When all corresponding inequalities are satisfied, queries are routed to the peers hosting the documents, and query processing can be completed locally; instead, when any inequality constraint is disobeyed, the query will never be answered by the query according to Interlacing Theorem, thus the query will not be shipped. For clarity, we sketch the query routing algorithm in Algorithm 1. Note that, false positive may exist since the eigenvalue-based routing synopsis is a *lossy* feature of the original document. In contrast, no false negative exists such that the algorithm guarantees the correctness of the query routing.

Algorithm 1 *query_routing*

$\lambda \leftarrow n$ eigenvalues of XML document D;
$\mu \leftarrow m$ eigenvalues of query q;
for each query eigenvalue μ_i **do**
  if $(\mu_i > \lambda_i)$ or $(\mu_i < \lambda_{n-m+i})$ **then**
    terminate without shipping q to D;
  end if
end for
ship q to the corresponding peer that hosts D;

3.3 Query Shipping in Overlay Networks

In P2P networks, query shipping is usually realized through multi-hop routing process. We now discuss how to deploy routing synopsis in hierarchical overlay networks to support effective query routing process.

Many practical P2P systems, such as Gnutella and KazAa, employ hierarchical overlay network architectures, where upper-level peers (called *ultra-peers*) manage the routing synopsis of all peers within the corresponding branch. Although most practical systems employ two-level architectures for simplicity, we consider a general setting that allows multiple (*i.e.,* more than two) levels, which has also been studied in [8].

Specific to our approach, the routing synopsis of a document consists of a (bounded) number of its eigenvalues, then the eigenvalues that correspond to lower-level (ultra-) peers aggregate up the hierarchical overlay network through Algorithm 2, where we consider an arbitrary ultra-peer p, and denote by $\{p_i\}$ the children (ultra-) peers in the overlay network, with corresponding routing synopsis represented as $\{s_i\}$. Since routing synopsis from different peers may be of various sizes, the cardinality of the eigenvalues of the aggregated synopsis is equal to the maximum size among all.

Algorithm 2 *aggregate_synopsis*

$s_{min} \leftarrow p$'s minimum eigenvalue set;
$s_{max} \leftarrow p$'s maximum eigenvalue set;
for each s_i from a child (ultra-) peer p_i **do**
  if $s_{min}.size() > s_i.size()$ **then**
    $\{\lambda_1^{min}, \lambda_2^{min}, ..., \lambda_n^{min}\} \leftarrow s_{min}$;
    $\{\lambda_1^{max}, \lambda_2^{max}, ..., \lambda_n^{max}\} \leftarrow s_{max}$;
    $\{\lambda_1, \lambda_2, ..., \lambda_k\} \leftarrow s_i$;
  else
    $\{\lambda_1^{min}, \lambda_2^{min}, ..., \lambda_k^{min}\} \leftarrow s_{min}$;
    $\{\lambda_1^{max}, \lambda_2^{max}, ..., \lambda_k^{max}\} \leftarrow s_{max}$;
    $\{\lambda_1, \lambda_2, ..., \lambda_n\} \leftarrow s_i$;
  end if
  for each $j \in [1, n]$ **do**
    if $j \leq k$ **then**
      $\lambda_j^{max} \leftarrow max(\lambda_j^{max}, \lambda_j)$;
    else
      $\lambda_j^{max} \leftarrow \lambda_j$;
    end if
    if $j < (n - k + 1)$ **then**
      $\lambda_j^{min} \leftarrow \lambda_j$;
    else
      $\lambda_j^{min} \leftarrow min(\lambda_j, \lambda_j^{min})$;
    end if
  end for
end for

Note that, now each ultra-peer maintains two separate aggregated routing synopsis s_{min} and s_{max}, reflecting the minimum and maximum eigenvalues respectively. This is to guarantee that the Interlacing Theorem inequalities still hold for all queries that satisfy the inequalities over any routing synopsis before aggregation. This ensures that no qualified queries is pruned from potential query processing at lower levels.

[8] Branching Path Queries covers a major part of XPath axes, including *self*, *child*, *descendant*, *descendant-or-self*, *parent*, *ancestor* and *ancestor-or-self* axes.

We claim this correctness guarantee in Theorem 2 and prove it as follows. Intuitively, we relax the aggregated eigenvalue features iteratively so that inequalities that hold over the documents on lower-level (ultra-) peers are still satisfied at upper levels. Initially, the minimum and maximum eigenvalues of ultra-peers are both set as eigenvalues of their own XML data.

Theorem 2. *The aggregated synopsis produced through Algorithm 2 guarantees query routing correctness.*

Proof. Without loss of generality, consider the eigenvalues of two arbitrary XML documents $\lambda = \{\lambda_1, \lambda_2, ..., \lambda_n\}$ and $\lambda' = \{\lambda'_1, \lambda'_2, ..., \lambda'_k\}$, where $k \leq n$; given two queries q and q', such that q satisfies the inequalities over λ, while q' satisfies the inequalities over λ'. We assume that the number of eigenvalues of both queries equals to m, where $m \leq k$ without loss of generality. Denote by μ_i and μ'_i the eigenvalues of the corresponding queries respectively. To demonstrate that Algorithm 2 produces aggregated synopsis that do not impact the correctness of query routing process, it is sufficient to prove that, the inequalities still hold for q and q' over the aggregated synopsis.

Denote by λ^{min} and λ^{max} the aggregated routing synopsis over λ and λ'; Algorithm 2 specifies $\lambda^{min} = \{\lambda_1, ..., \lambda_{n-k}, min(\lambda_{n-k+1}, \lambda'_1), ..., min(\lambda_n, \lambda'_k)\}$, and $\lambda^{max} = \{max(\lambda_1, \lambda'_1), ..., max(\lambda_k, \lambda'_k), \lambda_{k+1}, ..., \lambda_n\}$.

Since $\lambda_{n-m+i} \leq \mu_i \leq \lambda_i$, while $\lambda^{min}_{n-m+i} \leq \lambda_{n-m+1}$ and $\lambda_i \leq \lambda^{max}_i$, it is straightforward that $\lambda^{min}_{n-m+i} \leq \mu_i \leq \lambda^{max}_i$, such that q still satisfies the inequalities over the aggregated routing synopsis. Similarly, $\lambda'_{k-m+i} \leq \mu'_i \leq \lambda'_i$ and $\lambda^{min}_{n-m+i} = min(\lambda_{n-m+i}, \lambda'_{k-m+i})$, thus $\lambda^{min}_{n-m+i} \leq \lambda'_{k-m+i}$, meanwhile it is obvious that $\lambda'_i \leq \lambda^{max}_i$, thus the inequalities over the aggregated routing synopsis still hold for query q'.

Because both q and q' still satisfy the inequalities over the aggregated eigenvalues, the aggregated synopsis do not introduce any false negative, thus the correctness of the routing process is guaranteed in hierarchical overlay network structure. □

Correspondingly, it is necessary to make a one-line change to adapt Algorithm 1 to support separate aggregated routing synopsis: change the condition test of line (4) into ($\mu_i > \lambda^{min}_i$ or $\mu_i < \lambda^{max}_{n-m+i}$).

The routing synopsis on (ultra-) peers consists of $min(4 \cdot |q|, |V|)$ eigenvalues for each routing entry, where $|V|$ is the number of distinct XML element names that exist on peers within the corresponding overlay network branch, and $4 \cdot |q|$ bounds the number of eigenvalues of queries, where the constant factor is due to the use of two synopses (*i.e.,* minimum and maximum synopses) respectively. In practice, queries are short, such that the space complexity of the routing synopsis for each peer is usually bounded by a small integer.

3.4 Routing Synopsis Maintenance

When documents change (*e.g.,* structural changes), the corresponding eigenvalues need to be re-computed from scratch. Correspondingly, the aggregated synopsis at upper-level (ultra-) peers need updated if the changes introduce smaller eigenvalues for minimum aggregated synopsis or larger ones for maximum aggregated synopsis, which can be realized incrementally through Algorithm 2. Similarly, when new documents are added within a branch of an ultra-peer, the aggregated synopsis is maintained in the same manner. However, when documents are removed from the branch, incremental maintenance is infeasible unless the provenance of aggregated synopsis (*e.g.,* the source routing synopsis) is provided. Without enforcing proactive maintenance that requires provenance information, for document removal case, we can re-compute routing synopsis at upper-levels periodically, depending on specific application requirements. Note that, such deferred maintenance does not introduce extra false negative into query routing process.

Finally, when a variety of XML documents with different schema are deployed in P2P networks, we can enhance the effectiveness of routing synopsis with other features. For example, we can deploy fixed-length Bloom filters that encode element names on ultra-peers so as to pre-filter irrelevant peers from specific query routing. As shown in [8], Bloom filters are easy to be aggregated in hierarchical overlay networks and manipulations are computationally inexpensive. We will describe this "enhanced approach" in Section 4 and omit technical details here for space constraints.

3.5 Supporting Other Axes

Besides parent-child axis, wild cards and ancestor-descendant axis are also important since they belong to core XPath components. In this work, we exploit the following property (*i.e.,* Theorem 3) of eigenvalues over multiplication of matrix to handle these axes.

Theorem 3. *Denote by* $\{\lambda_1, \lambda_2, ...\lambda_n\}$ *the eigenvalues of matrix* M*; the corresponding eigenvalues of matrix* M^x *are equal to* $\{\lambda^x_1, \lambda^x_2, ...\lambda^x_n\}$.

Proof. Without loss of generality, consider an arbitrary eigenvalue λ_i of matrix M. Then $\lambda_i \times Vec = M \times$

Vec, where Vec denotes the corresponding eigenvector. According to the *associativity of matrix multiplication*,

$$\begin{aligned} M^x \times Vec &= M^{x-1} \times (M \times Vec) \\ &= M^{x-1} \times \lambda_i \times Vec \\ &= \lambda_i \times M^{x-1} \times Vec \\ &= \ldots \\ &= \lambda_i^x \times Vec \end{aligned}$$

Thus, the eigenvalues of M^x corresponds to those of M with a bijection, equal to $\{\lambda_1^x, \lambda_2^x, \ldots\lambda_n^x\}$. □

In essence, for a specific XML document and its anti-symmetric adjacency matrix M, M^x captures all hyper-edges crossing x edges (*e.g.,* x consecutive wild cards), while $M^* = \bigcup_{i=1}^{|V|} M^i$ covers ancestor-descendant relationship among XML elements. Thus, we can employ Algorithm 1 to check the eigenvalues of XPath queries with x consecutive wild cards (or ancestor-descendant axes) against the counterparts of M^x (or M^* respectively). The eigenvalues of M^x (or M^*) are computed directly based on M's eigenvalues, without involving matrix multiplication (or transitive-closure computation) for queries with wild cards or ancestor-descendant axis. However, since λ^x increases exponentially, which may lead to quick loss of discriminative, it is more appropriate to enforce this approach for queries with a small number of wild cards (or small documents for ancestor-descendant axis). For space constraints, we do not discuss more details in this paper and leave them to future work.

4 Performance Evaluation

4.1 Comparison system

A Bloom-filter approach is proposed in [8]. Briefly, a Depth Bloom filter (DBF) is created for each XML snippet with each different depths (up to the depth of the document) encoding path snippets; a Breadth Bloom filter (BBF) is created to encode all the element names at each XML tree level. Both DBF and BBF constitute the routing synopsis for a specific document. Given an XPath query q containing branching and parent-child axis, its DBF and BBF are easily computed. Then the testing of whether q is contained in a specific document D can be facilitated with bit-operations over Bloom filters. Since the routing synopsis is based on the Bloom filter "feature" of the document, it is closely related to our approach. Due to the use of uniform hashing to generate Bloom filters, the approach does not resolve wild card or ancestor-descendant axes directly.

4.2 Experiment Setting

We synthesize hierarchical overlay networks based on transit-stub hierarchical graphs generated with GT-ITM[9]. Specifically, we construct r transit domains, each with $\frac{N}{r}$ intra-domain peers, which are interconnected based on a spanning tree of all the peers within the domain. Without loss of generality, we choose $r = 7$ and N up to 2000 in this experiment.

We consider commonly used XML benchmarks including DBLP[10], XMark[11], Treebank[12], and XBench series benchmark (*i.e.,* DCSD and TCMD)[13]. The size of the data deployed on each peer is uniformly distributed between 2k and 20k bytes.[14]

In this work, we only consider absolute XPath queries that contain branching and parent-child axis. On the one hand, the comparison system does not support wild card or ancestor-descendant axis, which makes comparison infeasible; and on the other hand, this work is an initial study of using eigenvalue features for XML routing purposes, and we intend to consider more axes in future work. Due to lack of general XPath query generators, we implemented a query generator that can produce XPath queries that are *positive* or *negative* to a specific document, where a positive query can be satisfied by the document, while a negative query can not.

We use the Matrix Template Library[15] to compute eigenvalues, which takes milliseconds over the documents and queries for this experiment.

4.3 False Positive

For each XML benchmark, we generate 250 documents and 10 negative queries for each document respectively. Then we measure the false positive (ratio) of our approach against the Bloom-filter-based comparison approach. We also evaluate the approach that uses two extreme eigenvalues, as employed in [14]. The false positive ratio of this approach is high, close to 1. Realizing that [14] actually decomposes a document into multiple sub-documents and employs all corresponding eigenvalues in indexing, we would like to evaluate this improvement strategy in future work. Note that, we skip the evaluation of false

[9] http://www.cc.gatech.edu/fac/Ellen.Zegura/graphs.html
[10] http://dblp.uni-trier.de/xml/
[11] http://monetdb.cwi.nl/xml/index.html
[12] http://www.cis.upenn.edu/treebank/
[13] http://db.uwaterloo.ca/ ddbms/projects/xbench/
[14] According to [9], the typical XML documents on the Web are small and the average document size is around 4KB.
[15] http://www.osl.iu.edu/research/mtl/

Benchmark	DBLP		Treebank		DCSD		TCMD		XMark	
	mean	std	mean	std	mean	std	mean	std	mean	std
Bloom-filter-based	0.976	0.060	0.685	0.296	0.920	0.079	0.517	0.139	1	0
Eigenvalue-based	0.306	0.170	0.136	0.064	0.120	0.042	0.141	0.076	0.134	0.059

Table 1. False Positive Ratio

negative, which is always zero for both our approach and comparison system.

The *Mean* and *Std* (*i.e.,* standard deviation) of the false positive ratio over various XML benchmarks are shown in Table 1, which demonstrate that our Eigenvalue-based approach is more discriminative to recognize negative queries over specific documents.

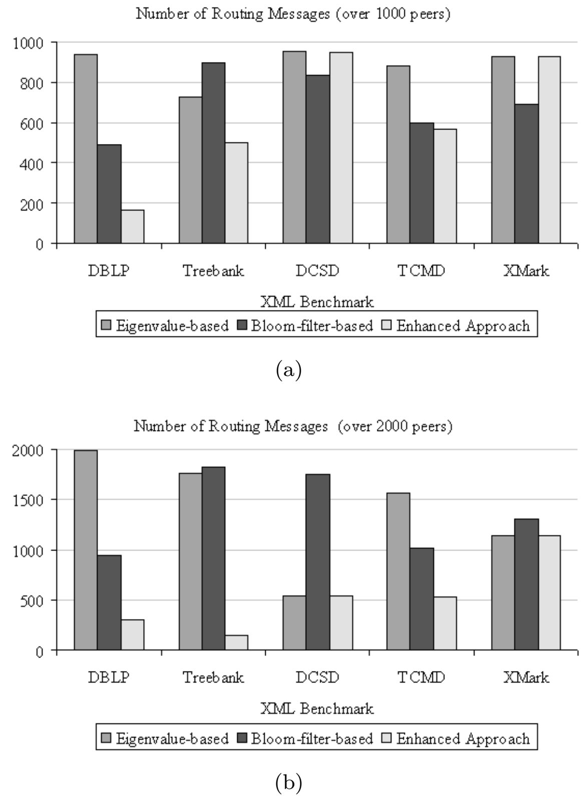

Figure 3. Routing Cost

4.4 Communication Cost

We test the query shipping cost for randomly generated positive and negative XPath queries (only containing branching and parent-child axis). The communication cost is measured with the number of messages consumed to ship the query. We do not measure the query processing latency. Instead, we use the concurrent number of hops as an alternative because it is straightforward to estimate the actual latency when we obtain the average hop-wise latency, which may be quite different for specific environments. For example, the average pair-wise latency over the Internet nowadays can be hundreds of milliseconds, while the latency across Enterprise-level networks can be less than ten milliseconds.

We consider a hierarchical network consisting of 1000 peers. For each XML benchmark, we deploy synthesized documents over peers randomly and then measure the routing cost for shipping 100 randomly generated XML queries (with branching and parent-child axis). All results are averaged over two running with different random seeds. Without loss of generality, we set the length of each BBF and DBF as 12 bytes.

Although our approach is more discriminative to recognize negative queries over specific documents, when we deploy it over a multi-level hierarchical network, generally it does not win over the comparison system, as shown in Figure 3 (a). Intuitively, this is primarily due to two reasons: (1) the aggregated eigenvalue synopses at upper-level ultra-peers are not sufficiently discriminative; (2) since we compute eigenvalues over label-splitting graphs without considering more subtle structures (*e.g.,* common edges at different levels) of the documents, negative queries with respect to one document may be (false) positive to others.

Interestingly, motivated by the comparison system that employs Bloom filter technology, we enhance our approach with a fixed-length Bloom filter to encode label names with a 15-byte Bloom-filter per routing entry. This mechanism considerably benefits the discriminative of aggregated synopsis on ultra-peers. As shown in Figure 3 (a), the *Enhanced Approach* takes much less routing cost than other approaches. Meanwhile, the synopsis size keeps bounded as a small integer, since the Bloom-filter is of fixed-length. We plan to explore this direction in future work. We also increase the size of Bloom-filters for the comparison system, and the results do show an improved discriminative against negative queries, however with a proportional increase of routing synopsis space.

For comparison, we show the counterpart over a hierarchical overlay network that contains 2000 peers in Figure 3 (b). The results show that, the enhanced approach is still preferable than other two approaches. In comparison to that over 1000-peer network (in Figure 3 (a)), the routing cost increases slightly since more peers are assigned data that are positive to specific queries.

We report the space cost per peer on routing synopsis in Figure 4, which demonstrates that our routing synopsis (of the enhanced approach) is more compact. Note that, the space complexity of our routing synopsis may be further reduced once the query load is known (*i.e.,* the maximum number of eigenvalues of queries). In contrast, the comparison system requires to encode over multiple levels and snippet lengths of the documents, thus the routing synopsis may grow significantly when documents grow deep and large. This may constrain its deployment in specific P2P systems with limited storage size, such as sensor networks.

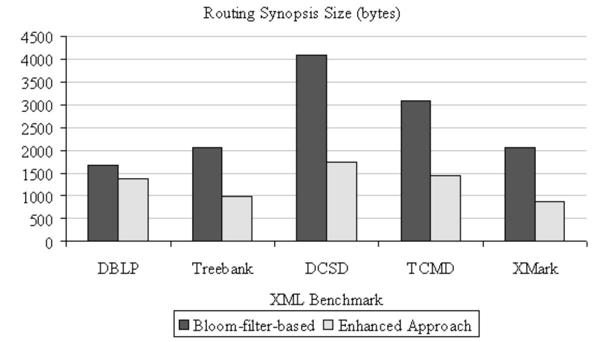

Figure 4. Synopsis Size

5 Conclusion

XML data are distributed over large peer-to-peer (P2P) networks on the Internet, and XML query routing mechanisms are crucial to facilitate XML data management and query processing.

In this work, based on well-established linear algebra property, we developed a compact yet effective eigenvalue-based XML routing synopsis over unstructured P2P networks. It is compact in that the number of eigenvalues maintained on each (ultra-) peer is bounded by a small number; it is also discriminative, as demonstrated by extensive experiments; finally, our approach is sufficiently general to be applied over graph-structure data, which has not been achieved by previous XML routing approaches. We also developed an effective aggregation algorithm to deploy eigenvalue-based routing synopses in hierarchical P2P overlay networks, which ensures the correctness of query routing.

Acknowledgement: We would like to thank Ning Zhang and Justin Wan for the discussion of Interlacing Theorem and related topics.

References

[1] M. Arenas, V. Kantere, A. Kementsietsidis, I. Kiringa, R. J. Miller, and J. Mylopoulos. The Hyperion project: from data integration to data coordination. *ACM SIGMOD Record*, 32(3):53–58, 2003.

[2] B. Bloom and R. Paige. Transformational design and implementation of a new efficient solution to the ready simulation problem. *Science of Computer Programming*, 24:189–220, 1995.

[3] A. Bonifati, U. Matrangolo, A. Cuzzocrea, and M. Jain. XPath lookup queries in P2P networks. In *International Workshop on Web Information and Data Management*, 2004.

[4] D. M. Cvetkovic, M. Doob, and H. Sachs. *Spectra of Graphs: Theory and Application.* Academic Press, 1979.

[5] L. Galanis, Y. Wang, S. R. Jeffery, and D. J. DeWitt. Processing queries in a large peer-to-peer system. In *Proc. of the 15th Int. Conf. on Advanced Information Systems Engineering*, 2003.

[6] R. Kaushik, P. Bohannon, J. Naughton, and H. Korth. Covering indexes for branching path queries. In *Proc. ACM SIGMOD Int. Conf. on Management of Data*, 2002.

[7] A. Kementsietsidis, M. Arenas, and R. J. Miller. Mapping data in peer-to-peer systems: Semantics and algorithmic issues. In *Proc. ACM SIGMOD Int. Conf. on Management of Data*, pages 325–336, 2003.

[8] G. Koloniari and E. Pitoura. Content-based routing of path queries in peer-to-peer systems. In *Advances in Database Technology — EDBT'04*, pages 29–47, 2004.

[9] L. Mignet, D. Barbosa, and P. Veltri. The XML web: a first study. In *Proc. 12th Int. World Wide Web Conference*, pages 500–510, 2003.

[10] M. Nicola and B. van der Linden. Native XML support in DB2 universal database. In *Proc. 31th Int. Conf. on Very Large Data Bases*, pages 1164–1174, 2005.

[11] V. Papadimos, D. Maier, and K. Tufte. Distributed query processing and catalogs for peer-to-peer systems. In *First Biennial Conference on Innovative Data Systems Research*, 2003.

[12] C. Sartiani, P. Manghi, G. Ghelli, and G. Conforti. Xpeer: A self-organizing xml p2p database system. In *Proc. of the First International Workshop on Peer-to-Peer Computing and Databases*, 2003.

[13] Q. Wang, A. K. Jha, and M. T. Özsu. An XML routing synopsis for unstructured P2P networks. In *the First International Workshop on XML, Web, and Internet Contents Technologies*, pages 176–183, 2006.

[14] N. Zhang, M. T. Özsu, I. F. Ilyas, and A. Aboulnaga. FIX: Feature-based indexing techniques for xml documents. In *Proc. 32th Int. Conf. on Very Large Data Bases*, 2006.