Net-$\mathcal{X}$: Unified Data-Centric Internet Services

Praveen Rao Justin Cappos Varun Khare Bongki Moon Beichuan Zhang
Department of Computer Science
University of Arizona, Tucson, AZ 85721
{`rpraveen`,`justin`,`vkhare`,`bkmoon`,`bzhang`}`@cs.arizona.edu`

***Abstract*— Databases and networks currently have different service models. Database services are *data-centric* in that users typically describe the content of data and the system finds and returns matching data. However, traditional Internet services are *server-centric* in that users have to know the location of data (*e.g.*, a URL) in order to retrieve it. We envision a future in which Internet services are data-centric. Users specify their interests and publishers describe their data. Based on the matching between user interests and data contents, users can pull data from publishers, and publishers can push data to interested users.**

We propose a unified system design called *Net-$\mathcal{X}$* to support data-centric Internet services seamlessly under a common framework. In *Net-$\mathcal{X}$*, user interests and data contents are characterized by polynomial signatures. These signatures are stored in a distributed hash table, on which interest matching is performed. Users can download matching data from publishers and publishers can push data to many interested users via per-document data-driven dissemination trees. By leveraging a wide range of database and networking techniques, *Net-$\mathcal{X}$* provides a scalable, flexible, and secure infrastructure for data-centric Internet services.

I. Introduction

Database services are *data-centric* in that users typically describe the content of data they want (by posing queries) and the system finds and returns matching data (by executing queries against databases). However, traditional Internet services are *server-centric* in that users have to know the location of data (*e.g.*, a URL) in order to retrieve it. Server-centric services facilitate inter-machine communication. Data-centric services, on the other hand, facilitate human/computer interaction, because they allow users to focus on the content of data rather than its location. This is demonstrated by the surging success of search engines such as Google, which offload the burden of locating data from users. Nevertheless, search engines are still not truly data-centric services due to two major limitations. First, they lack a powerful means for users to accurately describe their interests or for publishers to accurately describe their contents. Second, publishers can only passively wait for users to locate them. There is no way for publishers to find interested users or actively push data to users, which is important for many applications.

	Content-Based Matching	**Data Delivery**
Push Data	Doc ⇒ User Profile Index (query to find users)	Doc ⇒ Interested users (disseminate data to users)
Pull Data	User query ⇒ Doc Index (query to locate docs)	User ⇐ Relevant docs (fetch data from sources)

TABLE I
Dimensions of the Envisioned Services

We envision a future in which Internet services are data-centric: users specify their interests; publishers describe their data; Internet services match user interests and data contents to deliver data to users. The services should be *unified* in the sense that they should support both interest matching and data delivery, and support both pulling data by users and pushing data by publishers, as illustrated in Table I.

The *diversity* and *large scale* of Internet users, publishers, data types, and applications pose daunting design challenges. To address these challenges, we propose a system design called *Net-$\mathcal{X}$*, which leverages recent advances in database and networking technologies such as XML query processing and P2P/Overlay networks. *Net-$\mathcal{X}$* uses the XML data model to represent content and associated metadata, and user interests are specified in XPath [1] or XQuery language [2]. Suppose a user wants to find the song titled `Rain` by `Beatles` expressed in XPath as `/music[band=‘‘Beatles’’][song=‘‘Rain’’]`. *Net-$\mathcal{X}$* returns a list of publishers of that song to the user. The user can then download the song from a publisher. Suppose a user wants to receive news articles regarding `Windows Vista` expressed in XPath as `/news/subject=‘‘Windows Vista’’`. When a publisher has such an article, *Net-$\mathcal{X}$* enables the publisher to locate interested users, and disseminate the content to them. By design, *Net-$\mathcal{X}$* provides a *unified platform* that supports content-based interest matching, and allows users to pull data and publishers to push data.

The salient features of *Net-$\mathcal{X}$*'s design can be summarized as follows. *Net-$\mathcal{X}$* adopts a novel XML indexing scheme on top of a distributed hash table (DHT) to implement the functionality of content-based interest matching. *Net-$\mathcal{X}$* also uses novel per-document data-driven overlays for publishers to push data to users. In *Net-$\mathcal{X}$*, both user profiles and document contents are indexed by capturing the characteristics of their structures and values in XML format (or XQuery language) into a form of signatures. These signatures are stored in a distributed hash table called the *Rendezvous DHT (RDHT)*, which is the place where user interests and document contents meet each other. To pull data from publishers, a user sends a query signature to the RDHT to locate the relevant documents. The RDHT matches the query signature against the document signatures and returns corresponding publisher information for users to download data. To push data to interested users a publisher sends a document signature to the RDHT, finds all matching user profiles, organizes the interested users into an overlay dissemination tree, and disseminates data to them. With the unified services provided by *Net-$\mathcal{X}$* to both users and publishers, data-centric applications can be developed on top of *Net-$\mathcal{X}$*.

II. Background and Motivation

Recent advances in database and networking research have resulted in promising solutions to sub-problems depicted in Table I, but none has attempted a unified system design for data-centric Internet services that enable users to pull data and publishers to push data based on interest matching.

P2P/Overlay networks provide application-layer routing. They are scalable, robust, and can be deployed without changes to underlying router-level networks. Therefore, P2P/Overlay networks provide an ideal platform for distributed search and data dissemination. However, their services are usually oblivious to data content, which means applications or users have to carry out the task of content-based interest matching.

Recently, several XML querying schemes have been built on top of P2P/Overlay networks (*e.g.,* [3][4][5][6][7]), but they focus only on locating data sources for users and do not address the problem of locating interested users (by publishers) or delivering data to users.

Several publish/subscribe systems have been proposed in the past. Two core tasks in such systems include processing of subscriptions and delivering data to interested users. The popularity of XML has spurred the development of scalable XML filtering systems. In such a system, user profiles of interest (or subscriptions) are expressed in XPath [1] or XQuery language [2], and each incoming XML document is examined against user profiles. Many centralized XML filtering schemes have been proposed to process a large number of user profiles (*e.g.,* XFilter [8], YFilter [9], FiST [10]).

The prior research most related to *Net-$\mathcal{X}$* is Content-Based Networks (CBN) [11] and its variants [12][13][14][15]. In CBNs, packets are forwarded based on their data contents rather than IP address. In the original CBN design [11], users and publishers form an overlay network onto which users broadcast their profiles in the form of attribute-value predicates. Based on the user profiles, per-source (*i.e.,* per-publisher) dissemination trees are built on top of the overlay for data delivery. Overlay nodes are required to filter incoming packets against user profiles. This per-packet *in-network filtering* can be expensive, especially with diverse user interests, data contents, and large data volume. As was pointed out in [13], if the data is compressed or encrypted in-network filtering will require decompression/compression or decryption/encryption for each packet. Recent designs such as XTreeNet [12] and SemCast [13] propose to eliminate in-network filtering by trading off the accuracy of interest matching. These schemes forward data based on semantic topics rather than matching user profiles, and thus in-network filtering is not needed. However, unwanted data may be delivered to users. For example, a user interested in documents containing both topic X and topic Y will receive all documents containing X and find only those containing topic Y useful.

A common design choice in the previous work is to perform content-based interest matching and data delivery by building a "double-overlay" (Figure 1). Double-overlays consist of a "substrate layer" that connects all users and publishers and a "content layer" that is built *on top of* the substrate layer based on user profiles or data contents. Since the substrate layer is oblivious to user interest or data content, the content layer, which is restricted by the substrate, cannot be built fully reflecting user interest or data content. Therefore the *dissemination trees will include nodes that are not interested in the documents being disseminated*, causing several major problems as follows.

- *In-network filtering*: Costly per-packet processing.
- *Extra network delay*: Packets take detour through nodes not interested in the documents.
- *Unwanted data*: Due to reduced accuracy of interest matching in an attempt to eliminate in-network filtering.
- *Disincentive*: In a P2P environment, if nodes have to spend much resources to serve documents not of their interest, they will lose the incentive to participate.
- *Difficulty in security and privacy*: Since the documents are sent to users who are not supposed to or not interested in receiving them, it will be difficult to enforce security or privacy polices.

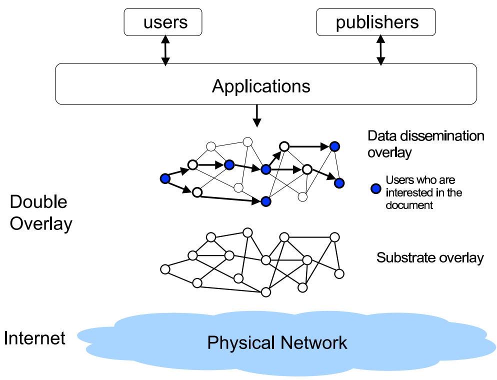


Fig. 1. Double-overlay adopted in previous work

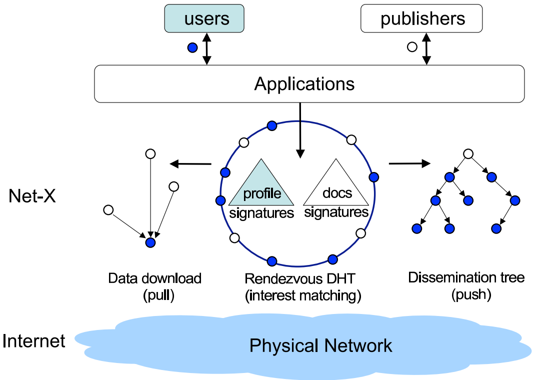


Fig. 2. Overview of *Net-𝒳*

III. *Net-𝒳* Architecture

Net-𝒳 adopts the XML data model, and provides applications with a unified service interface, the functionality of content-based interest matching, pulling data by users, and pushing data by publishers. As illustrated in Figure 2, *Net-𝒳* implements content-based interest matching via the RDHT, and efficient data delivery via per-document dissemination trees. This is a fundamentally different approach than a double-overlay and eliminates its aforementioned major problems. The RDHT (Section III-A) stores signatures of user profiles and signatures of data contents. User profiles contain not only the user's interest in data, but also their network information such as IP addresses, network coordinates [16], bandwidth constraints and so on. Similarly, along with the description of data contents, network information of a publisher such as IP addresses of servers, caches, and mirrors can be provided. This network information will be exploited to facilitate efficient data delivery. Peer nodes in the RDHT can be users, publishers, or any third-party service providers. Users can query the RDHT for interested documents, then download from publishers based on the returned information. Publishers can query the RDHT for interested users then build a dissemination tree comprised of only these users and send data over this tree (Section III-B). Security and privacy are an integral part of *Net-𝒳*. Some important issues are outlined in Section III-C.

A. Bridging Data and Users via Content-Based Interest Matching

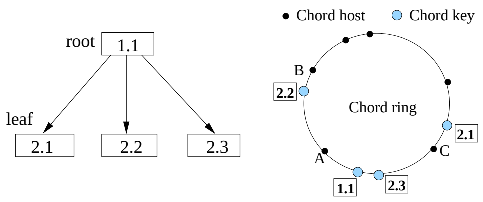
Scalable and fast content-based interest matching is a key aspect of *Net-𝒳*. Realizing it for the scale and diversity of the Internet is a daunting task. One of the main challenges is to index both XML data and user profiles of interest (expressed in XPath or XQuery) in a distributed way. Locating relevant data sources and identifying interested users should be done by contacting as a small number of hosts as possible in the network. Another main challenge comes from the dynamics of the Internet, where users or their computing hosts can join and leave the network at any time.

Our goal is to have efficient content-based interest matching for both pulling and pushing data in a scalable and distributed fashion with the minimum network delay and processing overhead. Our proposed RDHT adopts (1) new structural summarization methods that map both documents and user profiles into a common signature space and (2) new indexing strategies that can be distributed over the Internet and can be deployed independently of data delivery mechanisms. In the interest of space, we describe only the key ideas of our signature scheme and indexing strategies, and provide details in the technical report [17].

1) Polynomial Signatures: It is critical to match documents and user interest *holistically* in the Internet environment. Given an XML twig pattern that denotes a user's interest, if the matching process is performed by first finding *partial* matches (*e.g.*, by breaking a twig pattern into multiple linear patterns) and then taking intersections of the partial matches, the network traffic and local processing cost for lookup will increase considerably.

To enable holistic matching, both documents and user profiles are mapped into a common space so that their structural characteristics are captured and indexed in the same space in a unified fashion. We have designed and tested a new signature scheme where each document or a user profile can be represented by a polynomial with binary coefficients (*i.e.*, Galois Field of modulo two). This polynomial is a product of irreducible polynomials which are determined by factors such as the edges of distinct tag pairs and the structural characteristics of the documents.

There are several advantages in the polynomial signature approach. First, for these polynomials algebraic operations such as addition, multiplication, least common multiples (LCM), and greatest common divisors



(a) Signature index (b) Chord ring

Fig. 3. Storing the signature index nodes on a Chord ring

(GCD) can be efficiently computed by integer or bitwise operations [18]. Second, since for a given polynomial the presence of a particular irreducible polynomial can be tested by a polynomial division, the occurrence of a particular pattern in a document can be checked by a polynomial division operation performed on the polynomial signatures. Third, the similarity between signatures can be measured by using operations such as GCD. Similar signatures can be stored together when organized into a hierarchical index. Our preliminary results show that this type of polynomial signature can be used to provide a necessary condition for an exact match between data contents and user profiles. Our scheme allows no false negatives, but false positives may occur which can be eliminated in post-processing. In our preliminary experiments, the precision of matches was higher than 93% in all test cases.

2) Distributed Signature Index: The cost of finding matching signatures can be reduced significantly by organizing them in a hierarchical index because the search space for a given signature can be pruned quickly. The signatures of the documents (or user profiles) are organized such that the signature of an entry in a non-leaf node of the index is the LCM of the polynomial signatures in its child nodes. As a result, given a query signature the matching document signatures can be found by traversing the index nodes. The search space is pruned by performing a polynomial division of signatures of index entries with the query signature. To find interested users, a signature index that stores the signatures of user profiles is searched similarly. While inserting a signature into the index, a node entry whose signature has the highest similarity to the input signature is chosen. Consequently, similar document signatures (or signatures of user profiles) are expected to be placed close to each other in the signature index. In the Internet environment, this clustering property of the signature index plays a vital role, since similar signatures end up being stored by a few peers and thereby minimizing the time required to locate relevant documents or to identify interested users.

The signature index can be stored in a distributed way using a distributed hash table (DHT) framework (*e.g.*, Chord [19]) by treating the *identifier of an index node* as a key, and the *entire content of the index node* as a value. Figure 3 illustrates how a signature index in tree form is stored on the Chord ring. In Figure 3(a), each node is given a unique id. Figure 3(b) shows the Chord identifier ring where the dark dots denote the identifiers of the peers and the light dots denote the identifiers of the signature index nodes. A key is stored on the first peer that has an identifier equal to or follows the identifier of the key in the Chord ring.

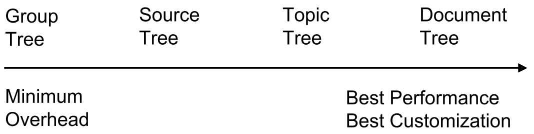


Fig. 4. Types of Dissemination Trees

B. Delivering Data to Interested Users

Once matching is done between user profiles and document contents, the documents need to be delivered to interested users. Depending on who initiates the communication, data delivery can be classified as either “pulling” data by users or “pushing” data by publishers. Potential performance problems that arise when many users are downloading the same document simultaneously (*i.e.*, flash crowd) can be overcome by using existing techniques and systems such as web caching.

The main challenge arises when publishers initiate the communication to push documents to a potentially large number of users. Conceptually, this can be done by overlay multicast. However, existing overlay multicast systems build dissemination trees either per group or per source which do not work well in our target environment given its diversity and large scale. We propose a novel approach of *data-driven*, *per-document dissemination trees* for better performance and flexibility.

1) Dissemination Trees: To disseminate a document to a group of users, we need to build an overlay that is rooted at the publisher and spans all and only the interested users. Existing multicast protocols use either “a group tree” or “source trees” for data dissemination and assume that *all* users in the system are interested in the disseminated data. In the group tree approach, all users in the system are connected by a *single* spanning tree which is used by different sources for data delivery. In the source tree approach, for each data source or publisher there is a spanning tree rooted at the source and data from different sources is forwarded along different source trees. Neither of these approaches is well suited for our target environment. For instance, using the group tree to send everyone documents which only a small number of users are interested in is certainly undesirable. For documents from the same source/publisher, the group of interested users may vary significantly from document to document. Therefore, using a source tree is not a good choice either.

Although recent work such as XTreeNet [12] and SemCast [13] build a dissemination tree for each "topic," document contents and users interests within each topic can still vary a lot. Furthermore, given the same publisher and the same set of users, a dissemination tree may be good for one type of document, but bad for another (*e.g.*, small text files vs. large streaming media).

Net-$\mathcal{X}$ solves the problem by building "per-document trees." For each document, there is an overlay dissemination tree rooted at the publisher and connecting only the users who are interested in this particular document. The major advantage is that the per-document tree can be optimized for the combination of a specific publisher, a specific document, and a specific group of users. For example, completely different trees can be used for small RSS files vs. large MPEG7 movies, personal blog sites vs. national content providers, and small chat-room participants vs. global audiences. This is critical to our goal of accommodating the diversity of publishers, users, and applications, which was not possible in existing systems.

Since the number of dissemination trees is the same as the number of documents, the control overhead of constructing and maintaining these trees could be very high. Figure 4 illustrates the trade-offs between different types of trees. To address this issue, *Net-$\mathcal{X}$* takes a "data-driven" approach. The publisher builds a dissemination tree when it has a document to send and abolishes the tree when the transmission is done. Therefore, at any moment a user only needs to participate in those trees that are currently sending documents of his or her interest. This is in contrast to existing overlay multicast systems which are mainly "membership-driven," meaning that dissemination trees are created and maintained as long as there are users even if there is no active data flow. In *Net-$\mathcal{X}$*, since publishers own the documents they push, they can decide when to keep the existing tree for future use (*e.g.*, if there are incoming documents very similar to the current one) and when to terminate it to reduce control overhead. During data dissemination, the tree is maintained distributedly by users. By combining the *per-document* and *data-driven* approaches, we strive for customized trees with good performance and small overhead for data dissemination.

2) Tree Construction: Our task can be formulated as the problem of building a spanning tree that optimizes dissemination performance under certain constraints such as a bound on each node's out-degree due to its bandwidth limit. Similar problem formulations and heuristic solutions have been proposed, especially in overlay multicast [20][21]. In our target environment, the publisher is the only data source, and the publisher has acquired profiles of all interested users. The profiles contain information like users' IP addresses, bandwidth limit, and Internet coordinates [16], which can be exploited by the publisher to *locally* compute a dissemination tree that is the best for the particular combination of the publisher, the document, and the group of users.

Computing a dissemination tree locally by a publisher eliminates the overhead of distributedly constructing the tree by users (as in many overlay multicast protocols) and may result tree structures with better quality. However, the main advantage is not a particular algorithm, but rather the publisher's *flexibility* to choose the algorithm that suits the dissemination environment best. This flexibility is critical for *Net-$\mathcal{X}$* to accommodate the diversity of publishers, users, documents, and applications, and it is the direct result of using per-document trees.

3) Tree Maintenance: Once the publisher has computed the tree locally, it can start sending data to its direct child nodes. The first packet will contain information of the tree structure so that receivers know where to forward packets down the tree. This process establishes the dissemination tree among user nodes and the tree information is no longer needed in later data packets. When data dissemination is done, the publisher can send a control packet to users to abolish the tree.

The dissemination tree is maintained distributedly by users. During data dissemination, users may join and leave, computers may crash and recover at any moment, and network conditions (e.g., delay, congestion) may also vary. In *Net-$\mathcal{X}$*, users continuously monitor the quality of the dissemination tree and refine or restructure the tree during the dissemination period. Distributed protocols such as NICE [22] and HMTP [23] can be used for this purpose.

C. Security and Privacy

Given the large scale and diversity in publishers, users, and their interests, malicious acts are inevitable. In our vision of unified data-centric Internet services, applications running on top of *Net-$\mathcal{X}$* will implement end-to-end security measures as needed and *Net-$\mathcal{X}$* will provide the basic building blocks to facilitate such implementations. In *Net-$\mathcal{X}$*, the actual data dissemination does not happen until the interest matching is done, which gives both publishers and users an opportunity to enforce security or privacy policies. For example, a publisher, after collecting profiles of all interested users, can disregard users who do not have the permission to access the document. On the other hand, a user can also check a publisher's identity before starting the actual data download. However, in the double-overlay approach, nodes who are not interested in the documents may still receive the documents and have to perform content-based matching, which makes it very difficult for security and privacy. In *Net-$\mathcal{X}$*, on the other hand, the introduction of the RDHT raises security and privacy concerns for the documents and user profiles. Here we discuss some major issues and suggest techniques to remedy these problems.

1) Integrity: Data stored in the RDHT need to be protected from data corruption and malicious modifications. Users and publishers identify themselves by public keys within the RDHT and sign their data items. This can prevent malicious or accidental addition and modification of the data items. To prevent unauthorized removal of items from RDHT, schemes such as [24] can reduce the ability of malicious nodes to remove items.

2) Privacy: Another major challenge is how to protect the privacy of users under the *Net-$\mathcal{X}$* system. Since user profiles are stored in the RDHT, malicious parties may harvest user information by querying the RDHT and spam users with unsolicited contents. If the users only want documents from specific publishers, they can encrypt profiles using the publishers' public keys. The polynomial signatures of user profiles should be readable by everyone for interest matching, but the actual user profiles (containing user IP address etc.) can be encrypted and only the intended publishers are able to decrypt and read the profiles. To avoid spam, we plan to experiment with a publisher reputation system to help users distinguish good publishers from bad ones. If anonymity is a concern, schemes such as Onion Routing [25] can be applied to protect the users and publishers during data dissemination as well as when interacting with the RDHT.

D. Future Direction

There are several research challenges in *Net-$\mathcal{X}$* that we intend to investigate further. For example, with the growing use of wireless devices and increasing availability of multimedia content, content customization becomes important. However, the scale and diversity of the users make it challenging. We plan to develop ways to incorporate content customization during dissemination tree construction in *Net-$\mathcal{X}$*. We plan to investigate new schemes for data delivery when data is being pulled by users. Furthermore, privacy of users and security issues in *Net-$\mathcal{X}$* require rigorous treatment.

IV. Conclusions

Net-$\mathcal{X}$ provides applications with unified service interfaces, the functionality of content-based interest matching, data delivery, pulling data by users, and pushing data by publishers. It adopts the XML data model, and employs novel database and networking techniques such as polynomial signatures and per-document data-driven dissemination trees. Although there are still many research challenges ahead, we believe that *Net-$\mathcal{X}$* is promising in realizing scalable and secure data-centric Internet services.

References

[1] A. Berglund, S. Boag, D. Chamberlin, M. F. Fernandez, M. Kay, J. Robie, and J. Simon, "XML path language (XPath) 2.0 W3C working draft 16," Tech. Rep. WD-xpath20-20020816, World Wide Web Consortium, Aug. 2002.

[2] S. Boag, D. Chamberlin, M. F. Fernandez, D. Florescu, J. Robie, and J. Simon, "XQuery 1.0: An XML Query Language W3C working draft 16," Tech. Rep. WD-xquery-20020816, World Wide Web Consortium, Aug. 2002.

[3] L. Galanis, Y. Wang, S. R. Jeffery, and D. J. DeWitt, "Locating Data Sources in Large Distributed Systems," in *Proceedings of the 29th VLDB Conference*, 2003.

[4] A. Bonifati, U. Matrangolo, A. Cuzzocrea, and M. Jain, "XPath Lookup Queries in P2P Networks," in *the 6th annual ACM International Workshop on Web Information and Data Management (WIDM'04)*, pp. 48–55, Nov. 2004.

[5] C. Sartiani, P. Manghi, G. Ghelli, and G. Conforti, "XPeer: A Self-Organizing XML P2P Database System," in *Inter. Workshop on Peer-to-Peer Computing and Databases*, 2004.

[6] L. Garces-Erice, P. Felber, E. Biersack, G. Urvoy-Keller, and K. Ross, "Data Indexing in Peer-to-Peer DHT Networks," in *Proceedings of the 24th International Conference on Distributed Computing Systems*, Mar. 2004.

[7] G. Koloniari and E. Pitoura, "Content-Based Routing of Path Queries in Peer-to-Peer Systems," in *Proc. of the International Conference on Extending Database Technology*, Mar. 2004.

[8] M. Altinel and M. J. Franklin, "Efficient filtering of XML documents for selective dissemination of information," in *Proceedings of the 26th VLDB Conference*, pp. 53–64, Sept. 2000.

[9] Yanlei Diao, Mehmet Altinel, Michael J. Franklin, Hao Zhang and Peter Fischer, "Path sharing and predicate evaluation for high-performance XML filtering," *ACM Trans. Database Syst.*, vol. 28, no. 4, pp. 467–516, 2003.

[10] J. Kwon, P. Rao, B. Moon, and S. Lee, "FiST: Scalable XML Document Filtering by Sequencing Twig Patterns," in *Proceedings of the 31st VLDB Conference*, pp. 217–228, Aug. 2005.

[11] A. Carzaniga, M. J. Rutherford, and A. L. Wolf, "A routing scheme for content-based networking," in *Proceedings of IEEE INFOCOM 2004*, Mar. 2004.

[12] W. Fenner, M. Rabinovich, K. Ramakrishnan, D. Srivastava, and Y. Zhang, "XTreeNet: Scalable Overlay Networks for XML Content Dissemination and Querying (Synopsis)," in *International Workshop on Web Content Caching and Distribution (WCW'05)*, Sept. 2005.

[13] O. Papaemmanouil and U. Cetintemel, "SemCast: Semantic Multicast for Content-Based Data Dissemination," in *Proceedings of the 21st Inter. Conference on Data Engineering*, Apr. 2005.

[14] Y. Diao, S. Rizvi, and M. Franklin, "Towards an Internet-Scale XML Dissemination Service," in *Proceedings of the 30th VLDB Conference*, Jan. 2004.

[15] B. Chandramouli, J. Xie, and J. Yang, "On the database/network interface in large-scale publish/subscribe systems," in *Proceedings of the 2006 ACM-SIGMOD Conference*, June 2006.

[16] T. E. Ng and H. Zhang, "Predicting Internet Network Distance with Coordinate-Based Approaches," in *IEEE Infocom*, (New York, USA), pp. 170–179, June 2002.

[17] P. Rao and B. Moon, "psiX: Hierarchical distributed index for efficiently locating xml data in peer-to-peer networks," Tech. Rep. 05-10, University of Arizona, Computer Science, 2005.

[18] E. Bach and J. Shallit, *Algorithmic Number Theory (Volume 1: Efficient Algorithms)*. MIT Press, 1996.

[19] I. Stoica, R. Morris, D. Karger, M. F. Kaashoek, and H. Balakrishnan, "Chord: A Scalable Peer-to-peer Lookup Service for Internet Applications," in *Proc. of ACM SIGCOMM*, Aug. 2001.

[20] S. Shi and J. Turner, "Routing in overlay multicast networks," in *Proc. of IEEE INFOCOM*, June 2002.

[21] S. Banerjee, C. Kommareddy, K. Kar, B. Bhattacharjee, and S. Khuller, "Construction of an efficient overlay multicast infrastructure for real-time applications," in *Proc. of IEEE INFOCOM*, Apr. 2003.

[22] S. Banerjee, B. Bhattacharjee, and C. Kommareddy, "Scalable application layer multicast," in *Proc. of ACM SIGCOMM*, Aug. 2002.

[23] B. Zhang, S. Jamin, and L. Zhang, "Host Multicast: a framework for delivering multicast to end users," in *Proc. of IEEE INFOCOM*, June 2002.

[24] M. Castro, P. Druschel, A. J. Ganesh, A. I. T. Rowstron, and D. S. Wallach, "Secure routing for structured peer-to-peer overlay networks.," in *OSDI*, 2002.

[25] D. M. Goldschlag, M. G. Reed, and P. F. Syverson, "Onion routing.," *Commun. ACM*, vol. 42, no. 2, pp. 39–41, 1999.