Computing and Informatics, Vol. 34, 2015, 1289–1308

ENERGY EFFICIENT XPATH QUERY PROCESSING ON WIRELESS XML STREAMING DATA

P. Prabhavathy, S. Bose

DCSE, Anna University
Chennai, TN, India
e-mail: pprabhavathy@gmail.com, sbs@cs.annauniv.edu

A. Kannan

DIST, Anna University
Chennai, TN, India
e-mail: kannan@annauniv.edu

Abstract. An energy efficient way of disseminating XML data to several mobile clients is broadcast. Information such as alert on emergencies, election results and sporting event results can be of interest to large number of mobile clients. Since eXtensible Markup Language (XML) is widely used for information exchange, wireless information services require an energy efficient XML data dissemination. XML Path (XPath) represents selective data required by mobile clients. XPath query processing involves two performance metrics, namely tune-in time and access time. In this paper, we propose a novel structure for streaming XML data called Path Stream Group Level (PSGL) node by exploiting the tree structure of XML document. It possesses various small indices such as level, child, sibling, attribute, text for selective download of XML data by mobile clients. It organizes data based on the level of XML document tree and groups XML elements with same XML path prefix to conserve battery power at mobile clients. Experimental results show that proposed method has reduced tune-in time when compared with existing approaches. Hence PSGL approach enhances performance with energy conservation for processing various types of XPath queries.

Keywords: Wireless XML streaming, energy conserving XPath query processing, mobile devices

Mathematics Subject Classification 2010: 68N01, 68U01, 68M14, 68M20

1 INTRODUCTION

XML is designed to describe data using arbitrary tags. It has emerged as most relevant standardization effort in document representation by mark-up languages in web based information systems. The main features of XML are related to use of tags for defining nested document structures. There are several classes of applications for XML, including publishing, specification and data exchange. For example, web services in Service Oriented Architecture (SOA) uses XML standard for Simple Object Access Protocol (SOAP) messages as well as Web Service Description Language (WSDL) documents.

Nowadays, it is essential to exploit XML data characteristics for achieving good performance in emerging applications such as wireless information services. The great scientist Albert Einstein quotes "Everything should be made as simple as possible, but not simpler", which aptly suites for the XML standard representation of data.

XML document has a semi structured data model. Hence XML data can be visualized as a data tree involving elements as nodes and nested tags as edges for parent-child relationship. Each XML tag can have zero or many attributes as well as sub-element tags. Zero or more text characters can be enclosed within tags. Many XML elements can have a same tag name but different location path in the Document Object Model (DOM) tree structure.

The main contributions of this paper are summarized as follows:

- We propose a novel structure PSGL node comprising many air indices for selective download of relevant information by mobile clients for energy conservation.
- An algorithm for generating PSGL node for each unique element from XML document and broadcasting stream is provided.
- Various types of XPath queries such as simple path, path with predicates and twig pattern are considered for experimentation with our proposed work.
- Performance evaluation of our proposed work is done with existing streaming techniques using various XML data sets.

The remainder of this paper is organized as follows: in Section 2, the background and problem statement related to this study are described. In Section 3, our novel PSGL node structure for wireless XML streaming is presented. In Section 4, experimental results in processing different types of XPath queries from two different XML datasets are presented. In Section 5, existing methods in XML streaming over wireless networks are explained. Finally, in Section 6, this paper is concluded with a discussion for future works.

2 BACKGROUND AND PROBLEM STATEMENT

Let us consider an example XML document and its DOM tree structure mentioned in Figure 1 for demonstration purpose of the proposed work in conserving energy

```
<?xml version="1.0"?>
<catalog>Book details
  <book type="technical">B101
    <author gender="Male" native="thirupattur">
                    <firstname>Mathew</firstname>
                    <lastname>Gambardella</lastname>
                     <address kind="residence">Woodlands gardens
                                    <town>Chetpet</town>
                                    <city>Chennai</city>
                                    <state>TamilNadu</state>
                     </address>
                     <address kind="office">Brindavan gardens
                                    <town>Kottur</town>
                                    <city>Chennai</city>
                                    <state>TamilNadu</state>
                     </address>
    </author>
    <title>XML Developer's Guide</title>
    <genre mode="General">Computer</genre>
    <price>
                    <member>240</member>
                    <foreigner>294</foreigner>
    </price>
<publication type="local">
    <publish_date>2000\10\01</publish_date>
<publisher>Tata McGraw Hill</publisher>
<address>
<state>delhi </state>
</address>
</publication>
  </book>
  <book type="commercial">B102
    <author gender="Female" native="mannargudi">
                    <firstname>Kim</firstname>
                    <lastname>Ralls</lastname>
                    <address kind="residence">Casablanca buildings
```


at wireless devices. Actually the DOM tree structure of Catalog XML document contains 42 nodes where only few nodes are shown in Figure 1 for the purpose of clarity.

Definition 1 (Tune-in Time)**.** The time spent by a mobile device staying active in order to acquire the requested data is termed as the tune-in time. During tune-in time, the mobile device will be in active mode. Any wireless device in active mode will consume high energy when compared with its doze (sleep) mode. Hence increase in tune-in time increases energy consumption and drains battery power quickly leading to reduced lifetime of a mobile device. Hence tune-in time should be minimized by selective downloading of XML data for energy conservation.

Definition 2 (Access Time)**.** The average time elapsed from the moment a mobile device makes a query to the moment when all requested data frames are received by

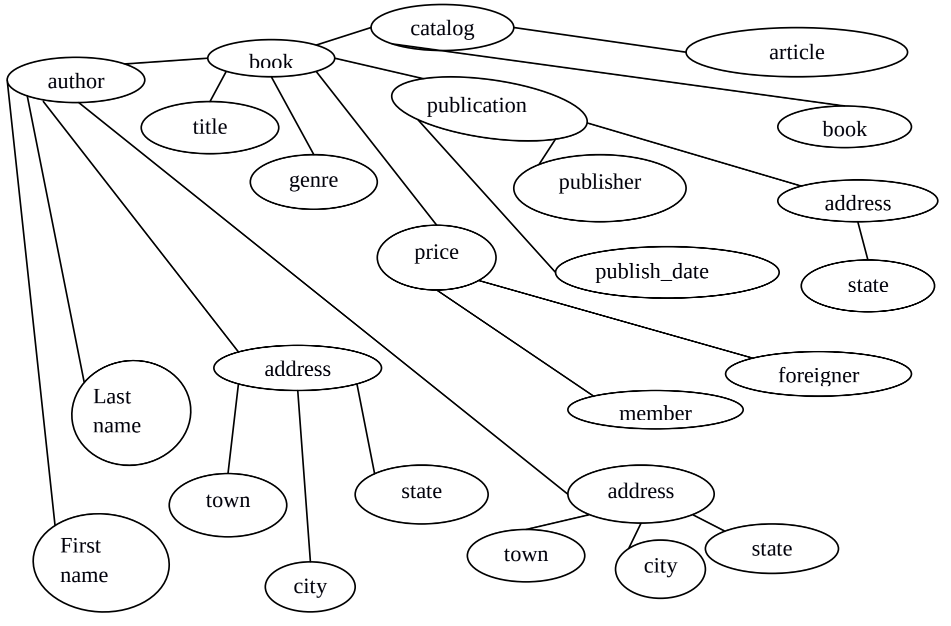


Figure 1. An example XML document and its DOM tree structure

it is termed as the access time. So it is the sum of tune-in time and waiting time for retrieving desired data frames in the broadcast channel. Hence access time should be considered as an optimal value.

Definition 3 (Indexing Efficiency)**.** The tune-in time saved per unit of access time overhead due to indexing is termed as the indexing efficiency. The ultimate aim of power conservative indexing is to reduce tune-in time while keeping an allowable access time overhead. This is measured by indexing efficiency.

Definition 4 (Simple XPath query)**.** A simple XPath query is a path location comprising intermixed with child axis (/), predicate (@) and/or wildcards (*). Each location step is an element name of the corresponding XML document.For example:

Q1 = */catalog/article*
Q2 = */catalog/book/**
Q3 = */catalog/book/author[@gender="Male"]*

The *Q1* fetches Technical Magazine as a result. *Q2* fetches all child elements of a book element namely the author, title, genre, price and publication whereas *Q3* retrieves only male authors found in Catalog XML document.

Definition 5 (Twig pattern XPath query)**.** Twig pattern XPath query is branched path location having location steps intermixed with child axis (/), branch predicate (@) and/or wildcards (*). Branch predicate is a fragment of location path in the corresponding XML document. For example:

$$\textbf{Q4} = /catalog/book/price[member/text() = "240"]/foreigner$$
$$\textbf{Q5} = /catalog/book/genre[mode/text() = "General"]/price/member$$
$$\textbf{Q6} = /catalog/book[title/text() = "MidnightRain"]/publication/address/state$$

The *Q4* fetches 294 as a foreigner book price where the corresponding member price is 240. *Q5* fetches 240 as a member book price, whose book genre is General, in Catalog XML document. *Q6* got California as a publisher state where the book title is the Midnight Rain.

Let us consider the XPath queries *Q3* and *Q6* to depict the problem statement. XPath query processing is done by evaluating all location steps in the XPath expression. During processing of each location step, many XML nodes are selected from the XML DOM tree that satisfies the axis of the query. The query *Q3* consists of 3 location steps whereas *Q6* has 6 location steps.

In the existing methods, more number of XML nodes are processed until the predicate condition becomes true to fetch the query result. More time and energy is consumed by these methods to provide the query result. Hence this paper aims at providing the energy efficient XPath query processing for mobile devices by providing novel Path Stream Group Level (PSGL) structure for streaming XML data in a Wireless broadcast channel.

In PSGL, XML nodes are binded together based on the location path information which is unique for each level in XML DOM tree. Each PSGL node structure has element name along with air index information for its child elements, sibling elements, attributes and text values. Thereby a very few PSGL nodes are downloaded for query processing to fetch the exact query result. Thus, it saves time and energy. The structural relationships such as parent-child and sibling among the XML DOM nodes are coded using selection bit strings with count value and location path information. Hence, the exact query answer is retrieved in an energy efficient manner using PSGL node structure over the wireless XML streaming broadcast channel.

3 PROPOSED WORK: PSGL STRUCTURED NODE

3.1 Wireless XML Broadcasting Stream Using PSGL Structured Approach

The PSGL structured XML nodes are flushed into the wireless broadcast channel for wireless streaming of XML document. The wireless XML stream structure for existing approaches and PSGL approach is shown along with DOM tree for PSGL

Stream structure with index followed by data

Ind_1	Ind_2	Ind_3	………	Ind_m	Dat_1	Dat_2	Dat_3	………	Dat_n

---→Timeline

Interleaved stream structure with index and data

Ind_1	Dat_1	Ind_2	Dat_2	Ind_3	Dat_3	………	………	Ind_m	Dat_n

---→Timeline

Partial index replication with interleaved stream structure

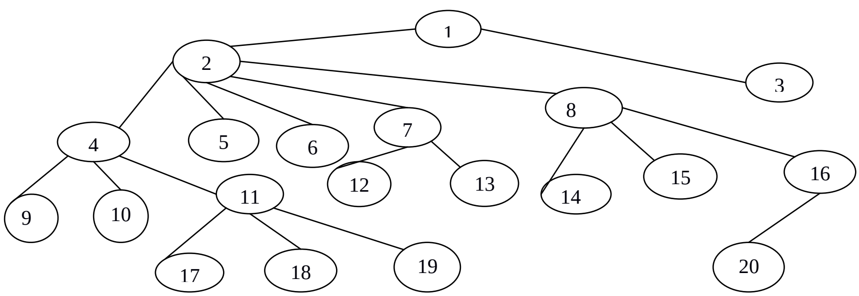
Ind	Ind_1	Dat_1	Ind_2	Dat_2	Ind	Ind_3	Dat_3	………	Ind	Ind_m	Dat_n

---→Timeline

Path stream group level node approach based stream structure

Loc Path Id	LvlInd	NdEleName	Att Ind	Txt Ind	Sib Ind	Chd Ind	SBS SBSct	Att Val	Txt Val	-----	Dat_n

---→Timeline



Location path is kept as node identifier. For figure clarity, numbers are provided for location path identifier

1: /catalog
2: /catalog/book
3: /catalog/article
4: /catalog/book/author
5: /catalog/book/title
6: /catalog/book/genre
7: /catalog/book/price
8: /catalog/book/publication
9: /catalog/book/author/firstname
10: /catalog/book/author/lastname
11: /catalog/book/author/address
12: /catalog/book/price/member
13: /catalog/book/price/foreigner
14: /catalog/book/publication/publish_date
15: /catalog/book/publication/publisher
16: /catalog/book/publication/address
17: /catalog/book/author/address/town
18: /catalog/book/author/address/city
19: /catalog/book/author**/address/state**
20: /catalog/book/publication/**address/state**

Note: 19 & 20 has same element name but different location paths. Existing XML streaming methods fail to differentiate it, leading to irrelevant data download which obviously increases power consumption at mobile devices

Figure 2. Wireless XML stream structures with DOM tree for PSGL structured nodes

structured nodes in Figure 2. The description for various fields found in the PSGL structured node stream is given below:

Loc Path Id: Location Path Identifier of particular level in XML DOM tree structure

LvlInd: Current level with arrival time of next level in XML broadcast stream

NdEleName: XML node element name

Att Ind: XML attribute name with arrival time of its value

Txt Ind: Arrival time of particular instance in list of text values

Chd Ind: Child node arrival time in broadcast stream

Sib Ind: Sibling node arrival time in broadcast stream

SBS: Selection Bit String representing parent child relationship in XML DOM tree structure

SBSct: SBS count represents number of siblings in particular XML DOM tree level

Att Val: Values of Attributes in that PSGL structured XML node

Txt Val: Values of Text in that PSGL structured XML node

Consider that the XML document shown in Figure 1 has been transformed into PSGL node structured for wireless streaming. PSGL node for the XML element "book" has the following information: ($t0, t1, t2, \ldots, t8$ represents air index)

Loc Path Id: /catalog/book

LvlInd: 1, t0

NdEleName: book

Att Ind: [type, t1]

Txt Ind: t2

Chd Ind: [author, t3], [title, t4], [genre, t5], [price, t6], [publication, t7]

Sib Ind: article, t8

SBS: 11

SBSct: 5, 5

Att Val: type = technical, commercial

Txt Val: B101, B102

The XML DOM tree for PSGL structured nodes is shown in Figure 2. The XML DOM tree shown in Figure 1 contains 42 nodes whereas DOM tree based on the PSGL structured node has only 20 nodes which avoided redundancy by grouping nodes based on path location with DOM tree level. A XML node element named "address" is found as child in both "publication" and "author" XML node elements. A query for fetching author's address requires clarity in XML node structure to fetch the exact result. So XML node element name cannot be kept as identifier for grouped node.

In PSGL structured node, path information along with DOM tree level is kept as an identifier to resolve this issue. This saves energy as well as fetches the exact query result during XPath query processing. This is clearly shown in Figure 2. The level index included in PSGL structured node approach enhances indexing efficiency measure and thereby it conserves battery power during XPath query processing. Sibling index is used to find all XML node elements found in the same tree level to match with a location step in XPath query. In the existing air indexing techniques, these features are not available. This novel PSGL structured node approach conserves energy by reducing tune-in time.

3.2 Algorithms Used for Implementation

Three algorithms are used in this work for implementation. Algorithm 1 used for PSGL structured stream generation for wireless XML broadcasting. The XML document to be broadcast is parsed using a Simple API for XML (SAX) parser at wireless data broadcasting server. It generates an energy efficient wireless XML broadcasting stream using PSGL structured node approach.

Input: XML document from repository
Output: PSGL structured nodes broadcasting stream
Fetch XML document from repository for SAX parsing
Initialise the data structures as well as variables as null
At start tag of XML element, do the following:

1. Increase the values of depth, nodeId and level of tree
2. Check stack data structure PS for current XML element name E
3. If PS does not have E, then frame new PSGL node named E and do the following:
   (a) Assign values for list of attribute and text found in PSGL node structure
   (b) Push E onto the top of PS
4. If PS has E, then add attribute, text values to list of PSGL structure named E
5. At end tag of XML element E, add PSGL node E into queue data structure Q

Repeat step 3 till all elements in the XML document gets exhausted
All PSGL nodes are checked with each other in Q for framing air indices and SBS
Completed PSGL nodes are placed in Q queue and flushed for broadcasting

Algorithm 1. PSGL structured stream generation for broadcasting

Algorithm 2 explains the working of a simple XPath query processing over the PSGL structured wireless XML stream. It constructs XPath query tree (T) and SBS for selectively downloading required PSGL nodes. It calculates the selection

Input: Simple XPath query tree T
Output: Final result text for query tree T
Initialize PSGL root node of query tree (T) with SBS as "1" and SBSct list as [1]
Repeat until query tree has nodes that have not been traversed
If (next node does not have predicate condition) Increment Level
Else Set flag *// presence of predicate indication*
Download PSGL node indicated by next node
If (flag is zero) Download PSGL node indicated by Level Index (LI)
Store values of Attribute List (AL) and Text List (TL) into current node in T
Select a set R of values in current node using SBS and return R
Else if (node has predicate conditions (PC) on Attribute values or Text values)
Download relevant values using Attribute Index (AI) and text Index (TI)
Frame selection bit predicate (SBpred) using Attribute or text value
Wait till next child in T arrives in broadcast stream by using childIndex of current node
//mobile client in doze mode to avoid energy consumption
Download the child PSGL node from the wireless XML stream (XS)
SBS = GetChildResultPart (current PSGL node, SBSpred)
SBS = (SBS of child PSGL node) Bitwise-AND (SBpred)
Return final result text of query tree T

Algorithm 2. Simple XPath query processing over PSGL structured XML wireless broadcasting stream

bit string based on the predicate (SBpred) condition used in the query. A bitwise AND operation between SBS and SBpred is used to arrive at the result SBS that identifies the required element. This process is repeated for all nodes till the final query result is arrived. Simple XPath query processing uses a method called GetChildResultPart(child PSGL node, SBpred of parent PSGL node) which is used to find the exact child XML element of corresponding parent XML element.

Algorithm 3 explains the processing of a twig patterned XPath query over the PSGL structured wireless XML stream. It has branches or twigs with a predicate condition, which has to be processed. The highest branching PSGL node which has the twig is downloaded and its SBpred is calculated using GetSBSof() function in this algorithm.

4 PERFORMANCE COMPARISON BY EXPERIMENTATION

We have implemented all the above mentioned algorithms in Java using Oracle's jdk 1.7.0 Update 17. We evaluate the performance of our PSGL structured wireless XML stream in processing different types of XPath queries. Comparison of S-node [16], DIX [29], PS + Pre/Post [31] approaches with our PSGL approach is done by using three datasets namely University Courses (U), SigmodRecord (S) and Mondial (M) from the XML repository [1]. All the experiments were conducted on a system with an Intel® Core™ i3 3.07 GHz processor and 2 GB main memory running Windows 7 operating system.

Input: Twig pattern XPath query tree T
Output: Final result R
Initialize PSGL root node of query tree (T) with SBS as "1" and SBSct list as [1]
Repeat until query tree has nodes that have not been traversed
If (next node does not have predicate condition) Increment Level
Else Set flag *// presence of predicate indication*
Download PSGL node indicated by next node
If (flag is zero) Download PSGL node indicated by Level Index (LI)
Store values of Attribute List (AL) and Text List (TL) into current node in T
Select a set R of values in current node using SBS and return R
Else if (current node has predicate conditions (PC) on Attribute values or Text values)
Download relevant values using Attribute Index (AI) and text Index (TI)
Frame selection bit predicate (SBpred) using Attribute or text value
Wait till next child in T arrives in broadcast stream by using childIndex of current node
//mobile client in doze mode to avoid energy consumption
Download the child PSGL node from the wireless XML stream (XS)
Let n be the required PSGL node with twig branching in T
SBSn = GetSBSof(n)
Let SP be the sequence path of nodes in tree T starting from n
PSGL node Pk = n and SBpred = SBSn
Repeat until all nodes in T are traversed
Let CPk be the child PSGL node of Pk
SBSc = GetChildResultPart(CPk, SBpred)
Pk = CPk and SBpred = SBSc
Select a result set R of values in leaf PSGL node CPk using SBSc and return R

Algorithm 3. Twig pattern XPath query processing over PSGL structured XML wireless broadcasting stream

An assumption is made that the bandwidth of broadcast channel is fully utilized for wireless XML data broadcasting. Activity of mobile device is considered for measuring access and tune-in times. Access time and tune-in time in processing different types of XPath queries is measured using the number of buckets. Bucket is the smallest logical unit in wireless broadcast channel. Based on the assumption of fixed network speed, number of buckets can be converted into time because time spent for reading a bucket is computed as bucket size divided by network speed [6]. Since the experimental results are independent on the bucket size used in experimentation, we have used the size as 128 bytes in our experiments. Table 1 shows the description of XML data sets used in our experiments.

Six XPath queries are considered from each XML dataset for experimentation purpose that is depicted in Table 2. XPath expressions are named based on its corresponding XML dataset. Xpath expressions 1 and 2 are from Simple XPath query, 3 and 4 are from predicated simple XPath query whereas 5 and 6 are from twig patterned XPath query.

XML Dataset Name	Size (KB)	Maximum Depth
University Courses	278	4
SigmodRecord	467	6
Mondial	1 743	5

Table 1. XML datasets used for experimentation

Expression Name	XPath Expression
U1	/root/course/subj
U2	/root/course/place/room
U3	/root/course/place/room[text()="120"]
U4	/root/course/time/end-time[@CourseId="VOLLUM-120"]
U5	/root/course[subj/text()="BIOL"]/time/start-time
U6	/root/course[subj/text()="BIOL"]/place/building
S1	/SigmodRecord/issue/volume
S2	/SigmodRecord/issue/articles/article
S3	/SigmodRecord/issue/articles/article/endPage[text()¡"150"]
S4	/SigmodRecord/issue/articles/article/authors/author[@position="03"]
S5	/SigmodRecord/issue[number/text()="5"]/articles/article/title
S6	/SigmodRecord/issue[volume/text()="10"]/articles/article/authors/author
M1	/mondial/country/name
M2	/mondial/country/languages
M3	/mondial/country/name[text()="Belgium"]
M4	/mondial/country/province[@city="Mariehamn"]
M5	/mondial/country[name/text()="Belgium"]/province/located-at
M6	/mondial/country/province[city/text()="Mariehamn"]/located-at

Table 2. XPath expressions used for experimentation

Performance metrics used for the comparison of PSGL approach with existing approaches are: stream size, access time ratio and tune-in time ratio of wireless XML stream(XS) relative to original XML document(D). These are defined as follows:

$$\text{Size ratio} = \frac{\text{Size of XS}}{\text{Size of D}} \times 100 \tag{1}$$

$$\text{Time ratio} = \frac{\text{Number of buckets to read XS}}{\text{Number of buckets to read D}} \times 100 \tag{2}$$

$$\text{Access time ratio} = \frac{\text{Number of buckets read in access time from XS}}{\text{Number of buckets read from D for query result}} \times 100 \tag{3}$$

$$\text{Tune-in time ratio} = \frac{\text{Number of buckets read during tune-in time from XS}}{\text{Number of buckets read from D in active mode}} \times 100 \tag{4}$$

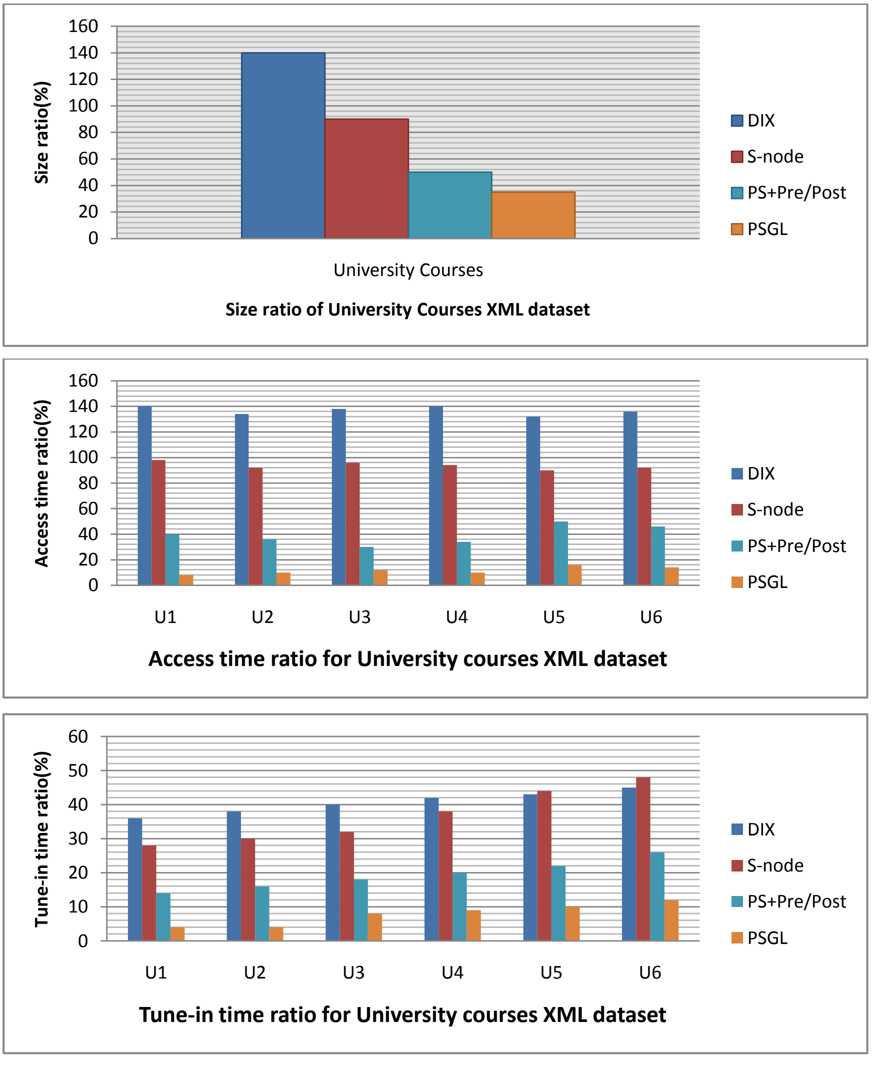


Figure 3. Performance comparison using University Courses XML dataset

The size ratio of PSGL approach is less than PS + Pre/Post, DIX and S-node approaches which is depicted in Figure 3. Major factors affecting the performance of access time ratio in Equation (3) are XML stream size and XML query processing finish time. XML stream size is measured using size ratio as mentioned in Equation (1). From the Figure 3, it is obvious that access time ratio of PSGL approach is less than the PS+Pre/Post, DIX and S-node approaches. The reason behind is that the PSGL structured XML stream has a minimum size by binding together XML

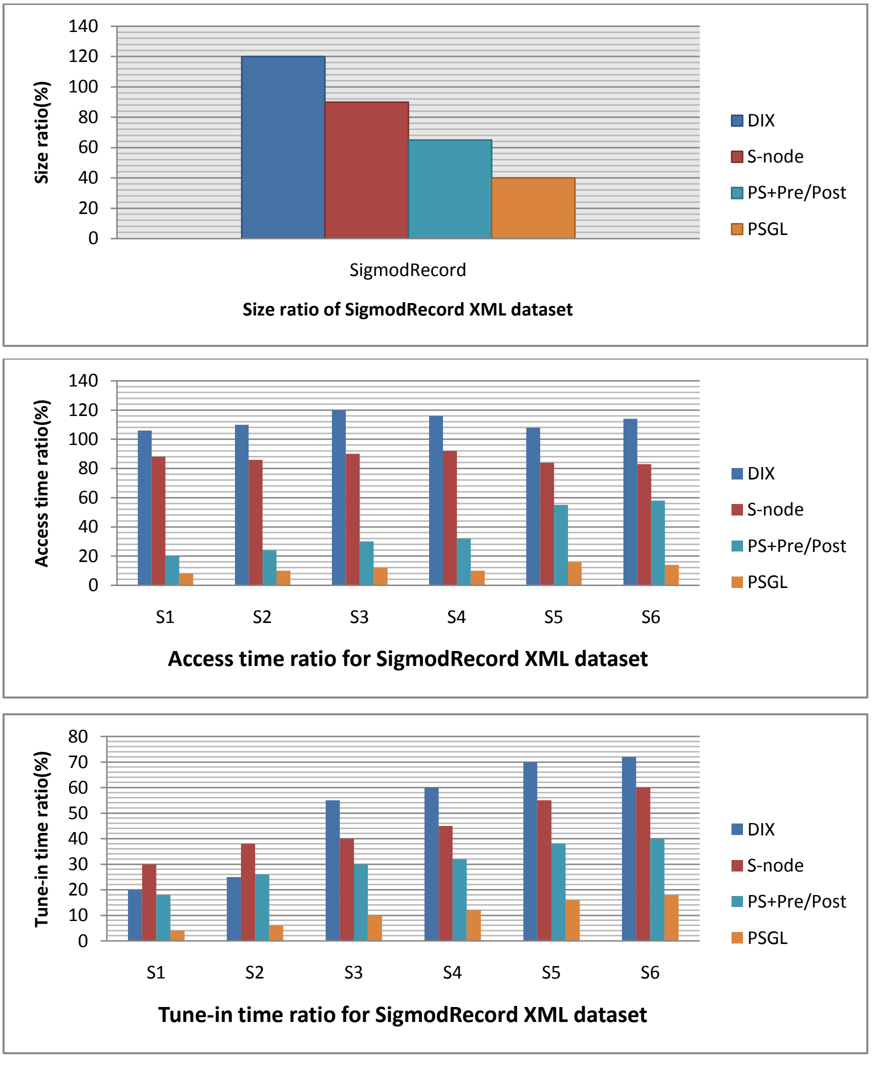


Figure 4. Performance comparison using SigmodRecord XML dataset

tags residing at the same level of tree. It completes the XPath query processing immediately using the various air indices provided in the PSGL structured XML stream. Though S-node, DIX and PS + Pre/Post approaches fastly process XPath queries over wireless stream, their large XML stream size affects the access time ratio.

The tune-in time of the PSGL approach is less than other XML streaming methods for all types of XPath queries from different XML data sets, since it selectively

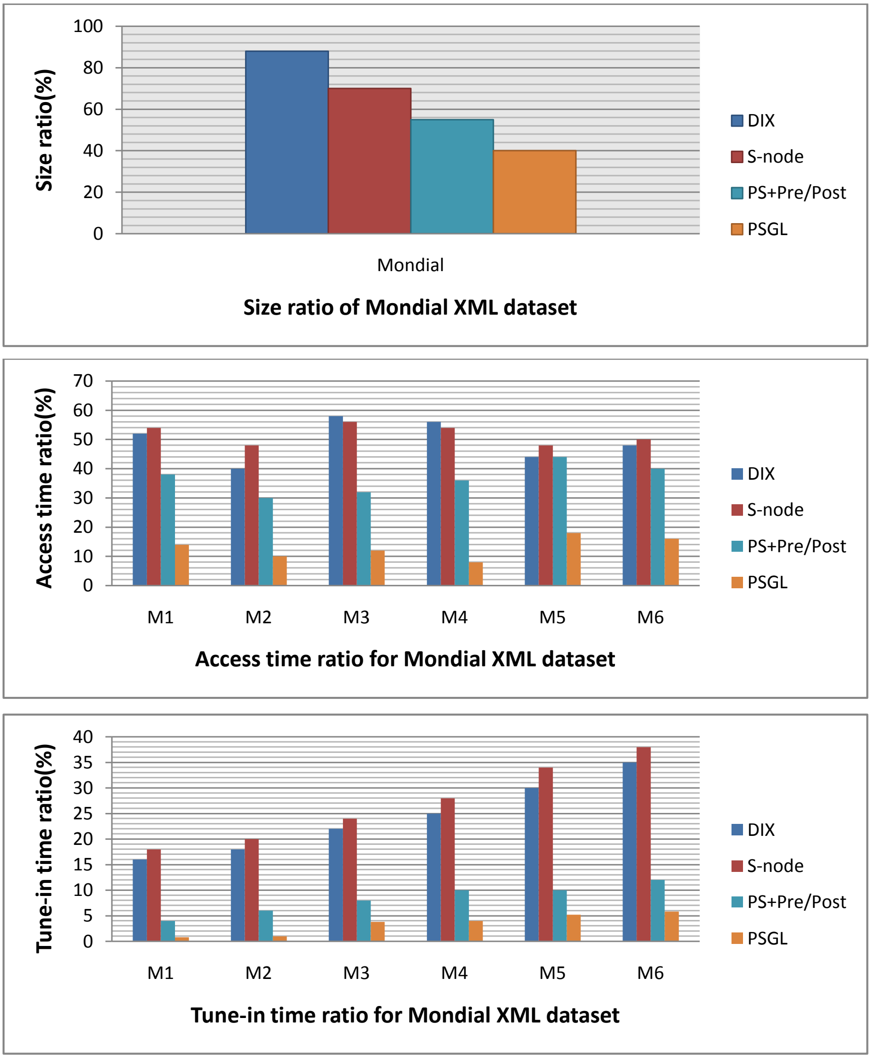


Figure 5. Performance comparison using Mondial XML dataset

tunes in to download relevant information from the wireless stream. The tune-in time ratio of S-node approach is greater than DIX and PS + Pre/Post approaches for many XPath queries on different XML datasets, because it cannot skip irrelevant parts of XML stream. DIX and PS+Pre/Post approaches has less tune-in time ratio when compared with S-node approach, since they have many indices to jump forward in wireless stream in order to skip irrelevant parts of the XML stream. PSGL approach consumes less tune-in time when compared with all other approaches.

This is clearly depicted in Figures 3, 4 and 5 for various XML datasets. Hence, the PSGL approach efficiently processes all types of XPath queries by conserving battery power in mobile devices.

5 RELATED WORKS

Due to the rapid development of mobile devices and wireless technologies, various services are delivered in wireless broadcasting environments. These services may include traffic conditions, digital broadcasting, weather reports and financial information [6].

But the traditional approach is designed for flat data items with key values, which cannot be directly applied to XML data because data stored in XML are hierarchically structured and self-describing. Since XML is widely used as a standard for information exchange among various applications, there has been a growing need to transfer and access XML data through wireless broadcast channel. Thus, the research area of XML is currently expanding into wireless computing environments [2, 3, 6]. Wireless XML stream structure based on path summary technique is very useful in wireless XML broadcasting [5].

A mobile client can predict the arrival time of its desired data using air index to stay in the power saving mode. Hence, it tunes into the broadcast channel only when the requested data arrive [2]. Wireless data dissemination is done in broadcasting mode for the efficient bandwidth utilization. In general, it has two modes of broadcasting, namely: pull based and push based. Pull based broadcasting is done based on requests gathered from group of clients whereas push based broadcasts hot data are without explicit client requests [21].

5.1 Data Broadcasting

The data broadcast technology stands to play a major role in the dissemination based applications for two reasons. First, data are distributed from a source to large number of clients that have overlapping interests. Second, communication technology has enabled a large scale dissemination supporting broadcast. The data broadcast has an asymmetric nature [3].

Asymmetry imposes constraints on the behaviour of clients and servers in a networked application. For example, in a system with a high client-to-server ratio, clients must limit their interactions with the server to a level which the server and backchannel can handle. WWW and FTP servers deal with this problem by limiting the number of connections they are willing to accept at a given time. Such a limitation can result in delays when accessing popular sites, such as a site containing a new release of a popular software package or up-to-date information on elections, sporting events, stocks, etc. [4].

[26] proposed periodic dissemination in context of mobile systems. It gathers data with a same access frequency to form records for the broadcast disk structure.

Fixed inter-arrival time of subsequent copies of data items in a broadcast cycle and maximum utilization of available bandwidth are considered as good features of a broadcast program. They formulated prefetch algorithms, integrated pull and push techniques [19].

[4] organized broadcast disk for efficient delivery of correct data. The authors proposed an adaptive information dispersal algorithm. Temporal matching performed by clock circuitry in a client device that consumes less power. But, a high degree of clock synchronization is an issue. [2] proposed randomized algorithm. Tune-in time for accessing data depends on number of multicast addresses available in broadcast program. If multicast addresses available are greater than the number of data items to broadcast then tune-in time is considered as one unit. Otherwise tune-in time depends on distribution of data among those available multicast addresses. Continuous access of broadcast data in wireless mobile computing is considered in [13] using artificial pointers.

5.2 XML Streaming

Most of the XML research had focused on the wired environment where XML data and its index are stored in the memory or disk [10]. But, in wireless broadcasting environments, the index information is on air. Hence, previous works on XML in wired environments are not directly applied to wireless broadcasting environments because these works do not need to consider the energy efficiency.

XML streaming algorithms broadly fall into three categories: the automaton based approach, the array based approach and the stack based approach. Automata are widely used for pattern retrieval against the XML stream. Algorithm XSQ supports a larger fragment of XPath than the earlier automaton based algorithms [8]. XSQ uses pushdown transducers as the basic building block for its system design. The evaluation problem is more complicated for a query with predicates. The translates of the given query to finite state automata requires states computation with corresponding transition tables.

TurboXPath [11] is a representative of the array-based approach. It first builds a parse tree for a given query and then finds matches of the parse tree nodes on the input XML stream. The algorithm works by trying to match each incoming (open or close) event with all the entries in the Work Array. Drawback of TurboXPath is that it works for tree-pattern queries on non-recursive XML documents.

Algorithm PathStack [18] was designed to evaluate path queries on preprocessed XML data. A number of more recent algorithms [25] exploit its ideas in order to evaluate Tree Pattern Queries on streaming XML data. These stack-based algorithms extend PathStack to compactly encode query pattern matches in a chain of stacks. This technique avoids the enumeration and explicit storage of the query pattern matches during the evaluation. The stack-based algorithms evaluate Tree Pattern Queries on XML streams in polynomial time and space, and this is a significant improvement over the automaton-based and the array-based approaches.

Large XML documents are fragmented for efficient query processing over XML streams. In [27] memory efficient query processing over XML fragment stream is considered, whereas in [28] time efficient query processing is done.From the above survey, it is obvious that energy conservation is of no primary concern in previous push based wireless XML streaming for XPath query processing. Our previous work related to PSGL structure is explained in [30]. To the best of our knowledge, PSGL structure in push based wireless XML broadcasting is the only method that has primary concern to energy conservation for XPath query processing in mobile devices.

From the above survey, it is obvious that energy conservation is not a primary concern in the previous wireless XML streaming for XPath query processing. But, PSGL structure based wireless XML streaming has as the primary concern the energy conservation for XPath query processing in mobile devices.

6 CONCLUSIONS

XPath expressions has location steps and predicate conditions with twig pattern for XML queries. Mobile devices have aimed at applications which can utilize their restricted battery power efficiently for increasing the battery lifetime. The existing approaches on wireless XML streaming only addressed processing of simple path queries and twig pattern queries in DFS traversal on XML query tree, which increases access time for deeper XPath queries. They have large wireless XML stream size due to huge indices information in the broadcasting channel.

Our PSGL approach generates relatively small wireless XML stream by binding XML nodes based on path and level grouping of nodes. Hence wireless XML stream generated by PSGL approach has lesser size than the existing approaches, and a relatively reduced access time for all XPath queries. PSGL node XML streaming approach has the features of both Depth First Traversal in SAX parsing and Breadth First Traversal in grouping parsed PSGL nodes having same location path to achieve optimal efficiency for deeper XPath expression. It means that it provides a reduced tune-in which makes it an energy efficient wireless XML streaming approach for XPath query processing in mobile devices.

Security mechanism for ensuring data integrity and increasing data availability by overcoming a loss of information incurred by wireless transmission are yet to be addressed for the efficient wireless XML data dissemination in mobile computing.

REFERENCES

[1] University of Washington XML Repository web site. Available on: `http://www.cs.washington.edu/research/xmldatasets/www/repository.html`.

[2] Imielinski, T.—Viswanathan, S.—Badrinath, B.: Energy Efficient Indexing on Air. Proceedings of ACM SIGMOD Conference on Management of Data, New York, USA, June 1994, pp. 25–36.

[3] Imielinski, T.—Viswanathan, S.—Badrinath, B.: Power Efficient Filtering of Data on Air. Proceedings of 4th International Conference of Extending Database Technology, March 1994, pp. 245–258.

[4] Acharya, S.—Alonso, R.—Franklin, M.—Zdonik, S.: Broadcast Disks: Data Management for Asymmetric Communication Environments. Proceedings of ACM SIGMOD Conference on Management of Data, San Jose, USA, May 1995, pp. 199–210.

[5] Goldman, R.—Widom, J.: DataGuides: Enable Query Formulation and Optimization in Semistructured Databases. Proceedings of International Conference on Very Large Data Bases (VLDB), San Francisco, USA, 1997, pp. 436–445.

[6] Imielinski, T.—Viswanathan, S.—Badrinath, B.: Data on Air: Organization and Access. IEEE Transactions on Knowledge and Data Engineering, Vol. 9, 1997, No. 3, pp. 353–372.

[7] Altinel, M.—Franklin, M.: Efficient Filtering of XML Documents for Selective Dissemination of Information. Proceedings of International Conference on Very Large Data Bases (VLDB), Cairo, Egypt, 2000, pp. 53–64.

[8] Ludascher, B.—Mukhopadhyay, P.—Papakonstantinou, Y.: A Transducer Based XML Query Processor. Proceedings of International Conference on Very Large Data Bases (VLDB), 2002, pp. 227–238.

[9] Diao, Y.—Franklin, M. J.: High Performance XML Filtering: An Overview of YFilter. IEEE Data Engineering Bulletin, Vol. 26, 2003, No. 1, pp. 41–48.

[10] Peng, S.—Chawathe, S.: XPath Queries on Streaming Data. Proceedings of International Conference on ACM SIGMOD, June 2003, pp. 431–442.

[11] Diao, Y. M.—Altinel, M.—Franklin, M.—Zhang, H.—Fischer, P. M.: Path Sharing and Predicate Evaluation for High-Performance XML Filtering. ACM Transactions on Database Systems, Vol. 28, 2003, No. 4, pp. 467–516.

[12] Gupta, A.—Suciu, D.: Stream Processing of XPath Queries with Predicates. Proceedings of International Conference on ACM SIGMOD, 2003, pp. 419–430.

[13] Saxena, P. C.—Arora, I. J.: Continuous Access of Broadcast Data Using Artificial Pointers Under a Wireless Mobile Computing. Computing and Informatics, Vol. 22, 2003, No. 5, pp. 473–510.

[14] Green, T. J—Gupta, A.—Miklau, G.—Onizuka, M.—Suciu, D.: Processing XML Streams with Deterministic Automata and Stream Index. ACM Transaction on Database Systems, Vol. 29, 2004, No. 4, pp. 752–788.

[15] Ramanan, P.: Evaluating an XPath Query on a Streaming XML Document. Proceedings of International Conference on Management of Data (COMAD 2005), Hyderabad, India, December 20–22, 2005.

[16] Park, C. S.—Kim, C. S.—Chung, Y. D.: Efficient Stream Organization for Wireless Broadcasting of XML Data. Proceedings of International Conference on Asian Computing Science, 2005, pp. 223–235.

[17] Park, S. H—Choi, J. H—Lee, S.: An Effective, Efficient XML Data Broadcasting Method in Mobile Wireless Network. Proceedings of 17th International DEXA Conference, Poland, September 2006, pp. 358–367.

[18] Chen, Y.—Davidson, S. B.—Zheng, Y.: An Efficient XPath Query Processor for XML Streams. Proceedings of the 22nd International Conference on Data Engineering (ICDE '06), 2006.

[19] Gou, G.—Chirkova, R.: Efficient Algorithms for Evaluating XPath over Streams. Proceedings of International Conference on ACM SIGMOD, 2007, pp. 49–280.

[20] Min, J. K.—Park, M. J.—Chung, C. W.: XTREAM: An Efficient Multi-Query Evaluation on Streaming XML Data. Information Sciences, Vol. 177, 2007, No. 17, pp. 3519–3538.

[21] Chung, Y. D.: An Indexing Scheme for Energy-Efficient Processing of Content-Based Retrieval Queries on a Wireless Data Stream. Information Sciences, Vol. 177, 2007, No. 2, pp. 525–542.

[22] Nizar, A.—Kumar, P. S.: Efficient Evaluation of Forward XPath Axes over XML Streams. Proceedings of International Conference on Management of Data (COMAD), Mumbai, India, December 17–19, 2008, pp. 222–233.

[23] Nizar, A.—Kumar, P. S.: Ordered Backward XPath Axis Processing Against XML Streams. Proceedings of 6th International XML Database Symposium (XSym), Lyon, France, August 24, 2009, pp. 1–16.

[24] Park, J. P.—Park, C. S.—Chung, Y. D.: Attribute Summarization: A Technique for Wireless XML Streaming. Proceedings of International Conference on Interaction Sciences: Information Technology, Culture and Human, December 2009, pp. 492–496.

[25] Wu, X.—Theodoratos, D.—Zuzarte, C.: Efficient Evaluation of Generalized Tree-Pattern Queries on XML Streams. The VLDB Journal, Vol. 19, 2010, No. 5, pp. 661–686.

[26] Chung, Y. D.—Yoo, S.—Kim, M. H.: Energy- and Latency-Efficient Processing of Full-Text Searches on a Wireless Broadcast Stream. IEEE Transactions on Knowledge and Data Engineering, Vol. 22, 2010, No. 2, pp. 207–218.

[27] Lee, S.—Kim, J.—Kang, H.: Memory-Efficient Query Processing over XML Fragment Stream with Fragment Labeling. Computing and Informatics, Vol. 29, 2010, No. 5, pp. 757–782.

[28] Kim, J.—Kang, H.: A Method of XML Document Fragmentation for Reducing Time of XML Fragment Stream Query Processing. Computing and Informatics, Vol. 31, 2012, No. 3, pp. 639–664.

[29] Park, J. P.—Park, C. S.—Chung, Y. D.: Energy and Latency Efficient Access of Wireless XML Stream. Journal of Database Management, Vol. 21, 2010, No. 1, pp. 58–79.

[30] Prabhavathy, P.—Bose, S.: Path Stream Group Level Encoding: Efficient Wireless XML Streaming. Proceedings of International Conference on Recent Trends in Information Technology (ICRTIT), Chennai, India, July 25, 2013, pp. 582–589.

[31] Mirabi, M.—Ibrahim, H.—Fathi, L.: PS + Pre/Post: A Novel Structure and Access Mechanism for Wireless XML Stream Supporting Twig Pattern Queries. Pervasive and Mobile Computing, Vol. 15, 2014, pp. 3–25.

P. Prabhavathy is currently working as teaching fellow in the Department of Information Science and Technology at Anna University. She has handled subjects on multimedia systems, programming data structures, software testing and management of information system for undergraduate engineering students. She handled computer organization and design, open technologies and mobile application development handled for postgraduate students. Academic subjects handling shows her knowledge in a versatile area. She published paper "Smart Phone User Assistance for Android" in Journal on Engineering Science and Technology. She is currently pursuing her Ph.D. in the area of wireless XML broadcasting at Anna University. She has delivered lectures on service oriented architecture and attended many workshops. Her research interest is also extended in the areas of XPath query processing and mobile computing.

S. Bose is one of the leading cryptographer scientists in India and has written a book titled "Cryptography and Network Security". He received his Ph.D. degree in information security from Anna University in 2007. He is currently working as Associate Professor in the Department of Computer Science and Engineering, Anna University and also as Dean-in-Charge in the University College of Engineering, Arni. He is a renowned expert and mass stunning speaker who have inspired a lot of students on ethical hacking through his hands on experience sessions. His publications in highly indexed journals talk about the vast knowledge and skill set in the area of cryptography. His research interest is also extended in the areas of network analysis using social networking, 4G communication, group key management in multicasting, information security and wireless XML broadcasting.

A. Kannan is currently working as Professor and Head in the Department of Information Science and Technology, College of Engineering, Guindy, Anna University, India. He has 25 years of teaching experience. He received his Ph.D. degree in computer science and engineering from Anna University in 2001. His research areas include software engineering, database management systems, and artificial intelligence.