ISSN 2229-5518

WIRELESS XML STREAMING USING LINEAGE ENCODING

Sathiya G

Abstract - With the rapid development of wireless network technologies, wireless mobile computing has become popular. Here, defined a streaming unit of a wireless XML stream called G-node to make wireless broadcasting more efficient. G-node enables mobile client to skip downloading of irrelevant data. For efficient XML twig pattern query processing, proposed two techniques such as Structure Indexing and Attribute Summarization. Structure Indexing directly captures the structural information of XML elements and Attribute Summarization reduces the size of the wireless XML stream. Here proposed Lineage Encoding to represent parent-child relationships between elements in G-nodes. Lineage Encoding makes use of light-weight and efficient bit string representation also provides algorithms for generating wireless XML stream. This scheme is the efficient wireless XML streaming approach where scalability, energy-efficiency, latency-efficiency and effective utilization of bandwidth are achieved.

Index Terms - Attribute Summarization, Data Mining, Lineage Encoding, Structure Indexing, Twig Patters, Wireless Broadcasting, XML Dissemination.

1 INTRODUCTION

As there is a huge development in wireless network technologies, wireless mobile computing has become very popular. Mobile devices such as smart phones and laptops are being used by the users for communication while they are moving.

Wireless broadcasting is an effective way of disseminating information in the wireless mobile environment. Benefits of using this approach: 1)the mobile clients need not send any request message that consumes more energy.2)More number of clients can be supported by the server without any additional cost(i.e., scalability),and 3)many clients can share the same broadcast channel(i.e., effective utilization of bandwidth).In wireless XML broadcasting, the xml data is retrieved by the broadcast server from the XML repository. Then, the parsed and generated wireless XML stream is continuously disseminated via a broadcast channel. The mobile client selectively downloads the XML data for query processing by tuning into the broadcast channel. Since we use mobile devices with limited battery power, energy conservation becomes a major consideration (i.e., energy efficiency). The tuning time and access time are used to measure energy efficiency and latency efficiency in wireless broadcasting. The tuning time is the sum elapsed time spent by a mobile Client to download the required data. When a mobile client downloads the data it consumes more energy than when it waits for data.

- *G Sathiya received her B.TECH degree in Information Technology from P.S.N.A. College of Engineering, Dindigul and received her M.E. degree in Computer Science & Engineering from J.J. College of Engineering, Trichirapalli. She is currently working as Assistant Professor in the Department of Information Technology, Tagore Engineering College Chennai, India. E-mail: sathiyame205@@gmail.com*

In this paper, proposed a novel, energy and latency efficient wireless XML streaming scheme supporting twig pattern queries in the wireless mobile environment.

The overview of the work is summarized as follows:

- Define a streaming unit of wireless XML stream, called G-node, The G-node structure eliminates structural overheads of XML documents.
- propose a light-weight encoding scheme, called Lineage Encoding, to represent parent-child
- Relationships among XML elements in G-nodes. Also define relevant operators that exploit bit-wise operations on the lineage codes. To the best of my knowledge, this scheme is the first wireless XML streaming approach that completely supports twig pattern query processing in the wireless broadcast environment.
- Provide algorithms for generating wireless XML stream consisting of G-nodes and query processing over the wireless XML stream.
- Evaluate the performance of this scheme by conducting extensive experiments using synthetic dataset.

The remainder of this paper is organized as follows: In section 2 presents related work on conventional XML query processing and wireless XML streaming approaches. Backgrounds and problem statement are described in section 3. The proposed system is explained in section 4. In section 5, draw conclusions and suggest future work.

2 RELATED WORKS

Several approaches have been proposed to efficiently process XML twig pattern queries. Multipredicate merge join algorithm uses a merge join algorithm to provide higher cache utilization and superior performance than a standard RDBMS algorithm. Al-Khalifa et al. investigated descendant nodes located at higher positions than their ancestor nodes in a stack. Based on this observation, they

propose efficient stack-based join algorithms. Twig Stack reduces the amount of the intermediate results and computational cost for merging the intermediate results using a chain of linked stacks that represent partial results to root-to-query path. XR Twig demonstrates superior performance because it skips elements that do not tally with given twig patterns, using an index-based algorithm. For indexed XML data, to enhance performance for matching twig pattern queries with the OR-predicate. Kaushik et al. proposed an XML path query processing algorithm integrating inverted list and structure indices. This technique reduces computation cost omitting join operation. A number of researches have been proposed to efficiently evaluate a large number of XML queries on a stream of XML documents. XFilter defines a Finite State Machine representation. YFilter proposes a filtering approach based on Nondeterministic Finite Auto-mata. It improves matching performance by processing common query path only once. Lazy DFA constructs a Deterministic Finite Automation in a lazy manner, thus, the number of states are reduced. XPush focuses on efficient evaluation of predicates. It avoids evaluation of redundant predicates by constructing a single deterministic push down automation in a lazy manner. GFilter addresses processing of the more complex Generalised-Tree-Pattern queries. It achieves polynomial time and space complexity by avoiding redundant predicate evaluation.

Conventional XML query processing methods mainly address the problem of efficiency and scalability. None of these approaches focuses on the energy-efficiency issue.

Path summary and Attribute Summarization exploits the benefits of the structure indexing to improve the performance of XML query processing in terms of both access and tuning time. Hover, they does not support twig pattern queries as described in the paper.

3 Background

3.1 XML data model and xpath expression

An XML document can be represented as a rooted, ordered, and labeled tree. Elements, attributes, and texts are represented by nodes, and the parent-child relationships are represented by edges in the XML tree.

In this paper, used Xpath as a query language. The results of an Xpath query are selected by a location path. A location path consists of location steps. Processing each location step selects a set of nodes in the document tree that satisfy axis, node test and predicates .For example, a query that finds cities located in Belgium can be represented by the following Xpath expression:

Q1://country[@name="Belgium"]/province/city

The above query consists of four location steps://country[@name="Belgium"], /province and /city. The first location step is to find a "country" node among the descendent elements of the document root, where its "name" attribute equals "Belgium". The second location step is to find "province" child nodes of the "country" nodes. The final step is to find the "city" nodes.

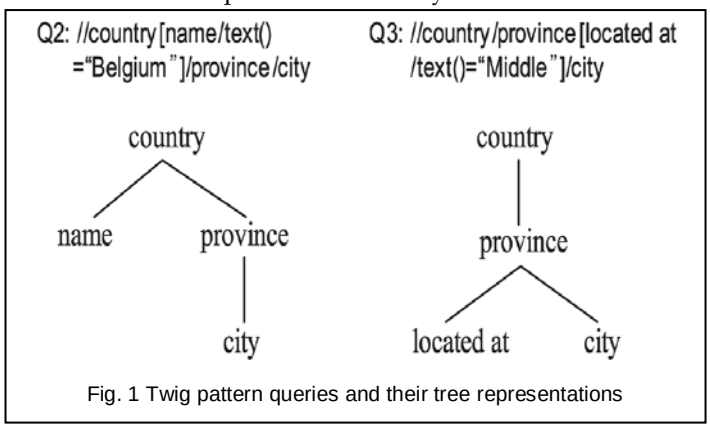

Fig. 1 Twig pattern queries and their tree representations

3.2 Twig Pattern Query

A twig pattern query consists of two or more path expressions, thus it involves element selections satisfying complex patterns in tree-structured XML data. The twig pattern query is a core operation in XML query processing and popularly used as it can represent complex search conditions.

3.3 Structural Indexing & Ambiguity Problem

Many techniques using a structure index have been proposed for efficient XML query processing. The structure information of XML documents and is used for XML query processing.

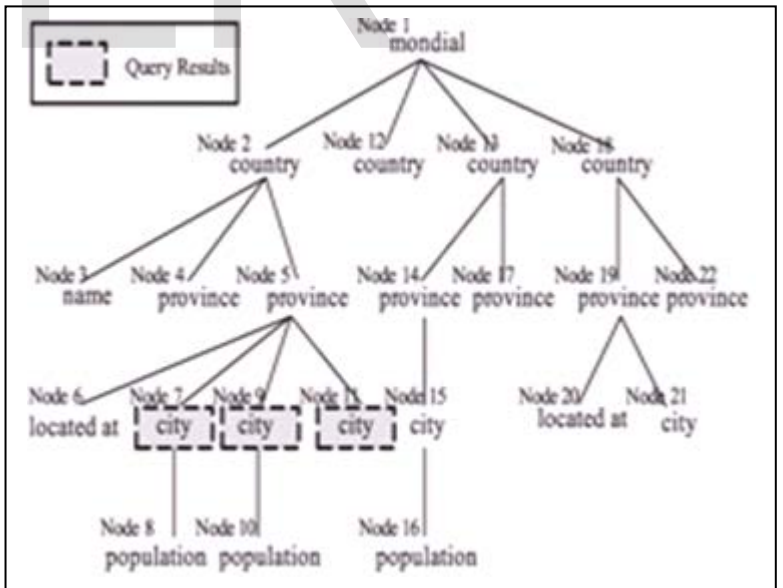

Fig. 2 Query processing results

Conventional wireless XML streaming methods using a structure index exhibit good performance for simple path query processing benefitting from the size reduction. These approaches integrate multiple elements of the same path into one node, thus, the size of data stream can be reduced

by eliminating redundant tag names. Hover, they do not support twig pattern queries because they do not preserve all parent-child relationships.

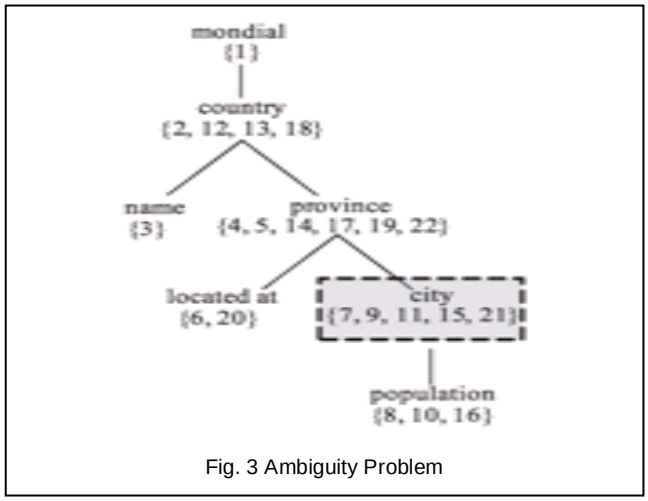

Fig. 3 Ambiguity Problem

Fig. 2 shows results of processing the query Q2 1) over the example XML document and 2) over its structure index. Query processing over the XML document exactly retrieves the results satisfying the given twig pattern as shown. Specifically, only three child nodes of Node 5(i.e. Nodes 7,9,11) are the exact answers for the given twig pattern query. The previous wireless XML streaming approaches using a structure indices, hover, cannot find the exact elements satisfying the given twig pattern because each structure index does not preserve the parent-child relationships among elements. For example, Nodes 7,9,11 cannot be differentiated from the other city nodes in Figure.

4 Proposed Method

In this section, proposed a new stream organization for XML data. Section 4.1 presents the unit structure of the stream called G-node. Section 4.2 explains the attribute summarization used in G-node. Section 4.3 presents XML stream generation algorithm.

Lineage codes in G-node for supporting xml twig pattern queries will be described later.

4.1 G-Node

It generates a wireless XML stream for integrating information of elements of the same path. That is, the XML data stream consists of the sequence of integrated (group) nodes, called G-node.

Definition: (G-Node)

The G-node denoted by Gp=(GDp , AVLp, TLp) is a data structure containing information of all the elements ep whose location path is p, where GDp is a group descriptor of Gp , AV Lp is a list containing all attribute values of e_p *, and T Lp is a list containing all text contents of* e_p

The Fig. 4 illustrates the structure of a G-node that integrates the elements of the path "/mondial/country/province" in the example XML document given.

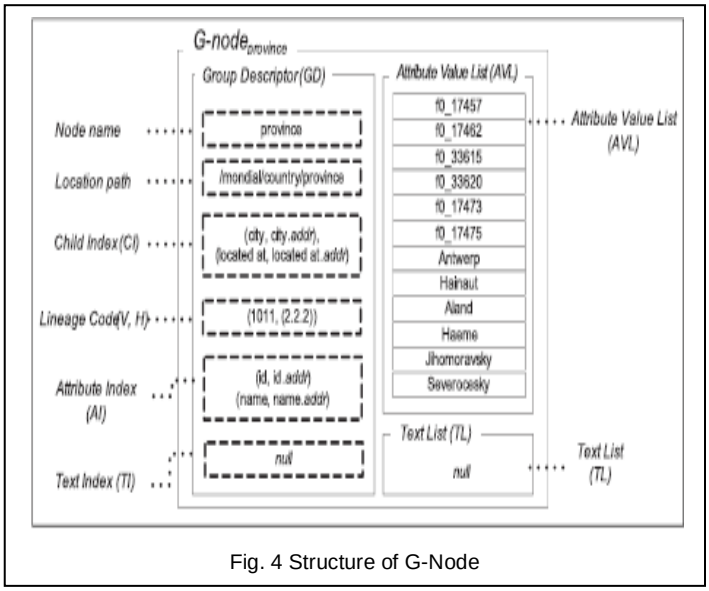

Fig. 4 Structure of G-Node

The group descriptor is a collection of indices for selective access of a wireless XML stream. Node name is the tag name of integrated elements, and location path is an XPath expression of integrated elements from the root node to the element node in the document tree. Child Index (CI) is a set of addresses that point to the starting positions of child G-nodes in the wireless XML string. Attribute Index (AI) contains the pair of attribute name and address to the starting position of the values of the attribute that are stored contiguously in Attribute Value List.

TextIndex (TI) is an address pointing to a starting position of Text List. In this scheme, an address means a point in time when the relevant data is broadcast on the air. The components of the group descriptor are used to process XML queries in the mobile client efficiently. Specifically, Node name, Location path are used to identify G-nodes. Indices relating to time information such as CI, AI and TI are used to selectively download the nextG-nodes, attribute values ant text. Finally, a Lineage Code(V,H) is used to handle axis and predicate conditions.

AttributeValueList(AVL) and Text List(TL) store attribute values and text contents of the elements represented by the G-node, respectively. Attribute values and text contents are stored in document order of elements.

4.2 Attribute Summarization

In this scheme, exploit the benefits of attribute summarization technique to reduce the size of a wireless XML stream. In XML, an element may have multiple attributes, each of which consists of a name and a value pair. In addition, there is a structural characteristic that elements with the same tag name and location path contain the attributes of the same name. For example, all "province" elements appearing the XML document in figure commonly contain "id" and "name" "attributes". Based on this observation, attribute summarization

eliminates repetitive attribute names in a set of elements when generating a stream of G-nodes.

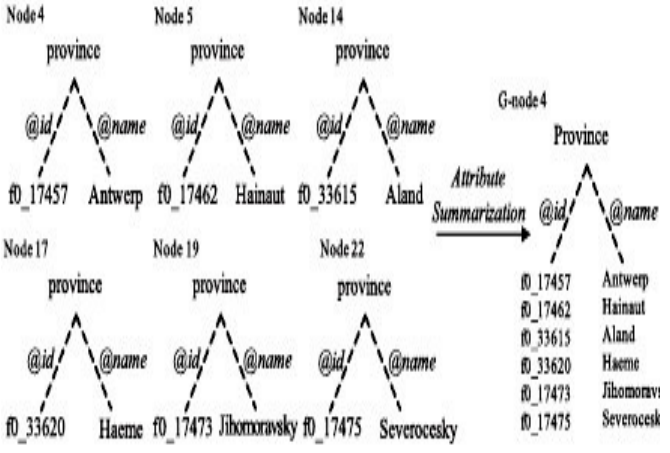

Fig. 5Attribute Summarization

Fig. 5.Shows the concept of attribute summarization. Six "province" nodes in the XML document presented in example XML document commonly contain two types of attributes whose names are "id" and "name". In this scheme redundant attribute names (i.e., five "id" and "name" strings) are eliminated, thus, the size of XML stream can be significantly reduced.

4.3 Wireless XML Stream Generation

In this section, explain how to generate wireless XML stream. A server retrieves an XML document to be disseminated from the XML repository and it generates wireless XML stream by using SAX (Simple API for XML) which is an event- driven API. SAX invokes content handlers during the parsing of an XML document. After streaming of XML data, streamed XML data are disseminated via a broadcast channel.

Algorithm 1 shows the wireless XML stream generation algorithm. First, the stream generator initializes related parameters at the start of document parsing (Lines 2-4). Whenever the stream generator encounters a start tag of an element, it decides if the relevant G-node is already constructed,. If the relevant G-node does not exist in the G-node queue, the stream generator constructs a new G-node. Otherwise, it inserts attributes of the current element into the existing G-node (Lines 6-17).

If the encountered element contains text, the stream generator invokes ContentHandler.characters() and inserts text into TL of the relevant G-node (Lines 21-23). The stream generator generates Child Index, Attribute Index, text index, and Lineage Codes (V, H) at the end of document parsing (Lines25-27). Finally, it flushes all G-nodes in the G-node queue into a wireless XML stream (Line 38).

4.4 Lineage Encoding

Lineage Encoding, to support queries involving predicates and twig pattern matching. Two kinds of lineage codes, i.e., vertical code denoted by Lineage Code (V) and horizontal code denoted by Lineage Code (H), are used to represent parent-child relationships among XML elements in two G-nodes are used. Here proposed a light-weight encoding scheme, called Lineage Encoding, to represent parent-child relationships among XML elements in the G-nodes. Also defined relevant operators and functions that exploit bit-wise operations on the lineage codes. To the best of my knowledge, this scheme is the first wireless XML streaming approach that completely supports twig pattern query processing in the wireless broadcast environment. The Attribute Value List (AVL) generated in Attribute Summarization with lineage encoded data is the key to process the Twig Pattern Queries in Selective tuning approach in the mobile end.

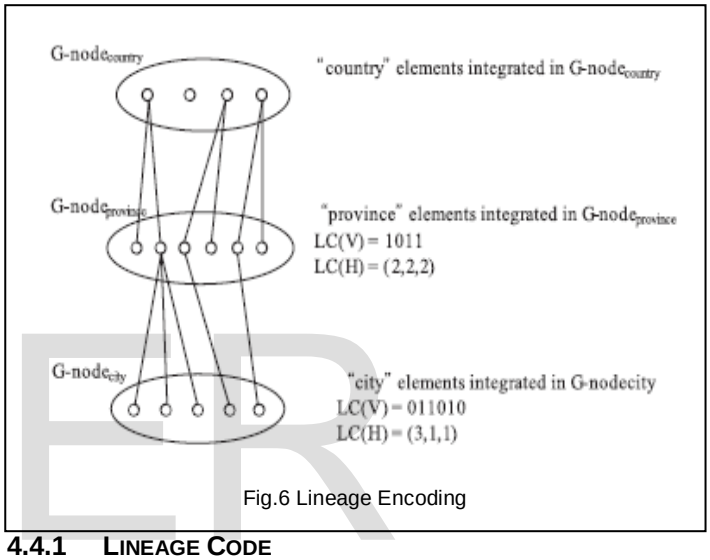

Fig.6 Lineage Encoding

4.4.1 Lineage Code

A novel encoding scheme is proposed, called Lineage Encoding, to support queries involving predicates and twig pattern matching. In the proposed scheme, two kinds of lineage codes, i.e., vertical code denoted by Lineage Code(H), are used to represent parent-child relationships among XML elements in two G-nodes.

4.4.2 Selective Functions

To process the query:

Q"//country[name/text()="Belgium"]/province/city"

should find the subset of "name" element satisfying the given predicate condition, select their parent "country" elements and then identify "province" elements which are children of those "country" elements.

Function 1. SelectChildren(C, SBp)

Input:

a G-node C, a selection bit string SBp for the parent G-node of C

Output:

a selection bit string SBc for C

Let the Lineage Code of C = (V,H)

Vm =Shrink&Mask(V, SBp)

SBc = Unpack(Vm,H)
Return SBc ;

Function 2. SelectParents(C,SBc)
Input:
a G-node C, a selection bit string SBc for C
Output:
the selection bit string SBp for the parent G-node P of C.
Let the Lineage Code of C = (V, H).
Vm = Pack(SBc,H);
SBp = Expand&Mask(V,Vm);
Return SBp;

In Function 1 and 2, it has 2 operators such as Shrink& Mask, Unpack. Similarly, in Function 3 & 4, it has Pack, Expand& Mask Operators. Both are very useful for Lineage Encoding, which is used to fine the lineage code.

Function 3. GetSelectionBitStringOf(N)
Input:
a G-node N in the query tree T
Output:
a selection bit string SBn for N
Let SBn be a bit string of length l, where l equals the number of elements in N and all bits are 1;
IF(N has a predicate on its attributes or text) {
Evaluate the predicate to obtain a selection bit string SBp identifying the ansr elements to the predicate in N;
SBn=SBn Bitwise-AND SBp;
}
FOR-EACH(child G-node C of N)
IF(all bits in SBn equal to 0) Return SBn;
}
FOR-EACH(child G-node C of N)
IF(all bits in SBn equal to 0) Return SBn;
SBn = GetSelectionBitStringOf©; //recursive call
SBn = SelectParents(C,SBc);
SBn = SBn Bitwise-AND SBp;}
Return SBn;

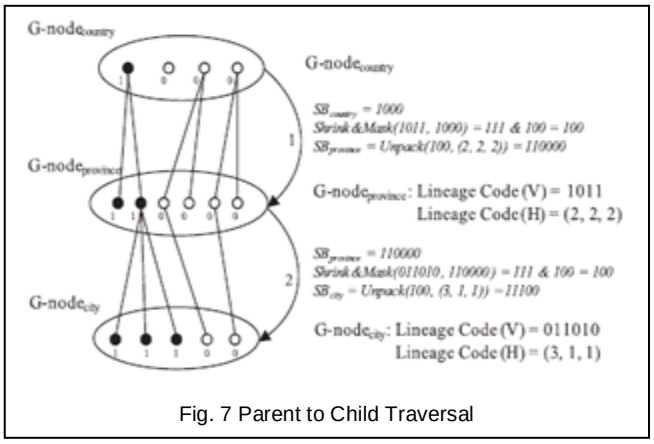

Fig. 7 Parent to Child Traversal

4.4.3 Query Processing Over Wireless XML Stream

In this section, described how a mobile client can retrieve the data of its interests. Assuming that there is no descendant axis in the user query, query processing algorithms for a simple path query and a twig pattern query are presented in Sections 4.4.3.1 and 4.4.3.2, respectively.

4.4.3.1 Simple Path Code Processing

Algorithm 1 shows the simple path query processing over the wireless XML stream. Given a query, the mobile client constructs a query tree (Line 6). Then, it starts to find relevant G-nodes over the wireless XML stream (Lines 7-22). The mobile client downloads a group descriptor of the G-node which corresponds to the query node (Line 8). If the current node is the leaf node, the mobile client download all the attribute values and texts in AVL and TL (Line10).Otherwise, the mobile client selectively downloads only the attributes values and/or texts involved in the predicate using AI and TI (Lines 12-14). Then, the mobile client continues to find the next child G-node using CI (Line 15).At each query node, the mobile client computes the selection bit string using the SelectChildren() function (Line 17). If the query node contains predicate conditions, the mobile client computes the selection bit string satisfying given predicates (Lines 18-21).After traversing all relevant nodes, the mobile client selects the ansr elements by the final selection bit string (Lines 23-24).

Algorithm 1 Simple Path Query Processing

Input: Wireless XML Stream DS, a simple path query Q
Output: Result set R satisfying Q
begin
01: result set R = □;
02: Initialize the selection bit string SB as 1;
03: Initialize Lineage Code of the root G-node as (1,((1));
04: Initialize nextNode as the address of the root G-node in DS;
05: Construct a query tree T for Q;
06: REPEAT {
07: Tune a group descriptor GD of the G-node indicated by nextNode;
08: IF (current node CN is the leaf node in T)
THEN
09: Store AVL and TL into the node in T ;
10: ELSE
11: IF (CN contains predicate conditions P)
THEN
12: Tune the relevant attribute values and/or text using AI and/or TI;
13: END IF
14: Assign the address of the next node in CI to nextNode;
15: END IF
16: SB = SelectChildren(CN, SB);
17: IF (CN contains predicate conditions P) THEN

18: Compute the selection bit string SBpred satisfying P ;
19: SB =SB Bitwise-AND SBpred;
20: END IF
21: } UNTIL (all nodes in T are completely traversed)
22: Select a set R of elements in CN using the selection bit string SB;
23: Return R;
End

4.4.3.2 TWIG PATTERN QUERY PROCESSING

Twig pattern query processing consists of three phases: Tree traversal phase, Sub paths traversal phase, and Main path traversal phase.

The main path denotes a path from the root node to a leaf node which represents the target element of the query, while the sub paths denote branch paths excluding the main path in the query tree.

Algorithm 2:Twig pattern query processing

Input: Wireless XML stream DS, twig pattern query Q
Output: Result set R satisfying Q
Begin
01: result set R = 5; // initialization
02: Initialize the selection bit string SB as 1;
03: Initialize Lineage Code of the root G-node as (1, (1));
04: Initialize nextNode as the address of the root G-node in DS;
05:
06: // Tree traversal phase
07: Construct a query tree T for Q;
08: REPEAT{
09: Tune a group descriptor GD of the G-node indicated by nextNode;
10: IF (current node CN is the leaf node in T)THEN
11: Store AVL and TL the node in T ;
12: ELSE
13: IF (CN contains predicate conditions P) THEN
14: Tune the relevant attribute values and/or text using AI and/or TI;
15: Store the relevant attribute values and/or text into the node in T ;
16: END IF
17: Assigning the address of the next node in CI to nextNode;
18: END IF
19: } UNTIL (all nodes in T are completely traversed)
20:
21: // subpaths traversal phase
22: Let N be the highest branching node in T ;
23: SBN = GetSelectionBitstringOf(N);
24:
25: // Main path traversal phase
26: Let MP be the main path in T starting from N ;
27: P =N ;
28: SBp =SBN ;
29: REPEAT {
30: Let C be the child node of P in MP;
31: SBc = SelectChildren(C, SBp);
32: P = C ;
33: SBp = SBc ;
34: } UNTIL (C is the leaf node)
35: Select a set R of elements in C using the selection bit stringSBc ;
36: Return R;
End

5 EXPERIMENTAL RESULTS

The results for each step explained in the proposed method are discussed in this section. Here, used a real XML data set and synthetic data set for the correctness of experiments. The results for XML Dissemination, Linage Encoding for Real XML Data Set database are given in Figure 8,9,10.

Fig. 8 XML Dissemination

Fig. 9 Lineage Encoding

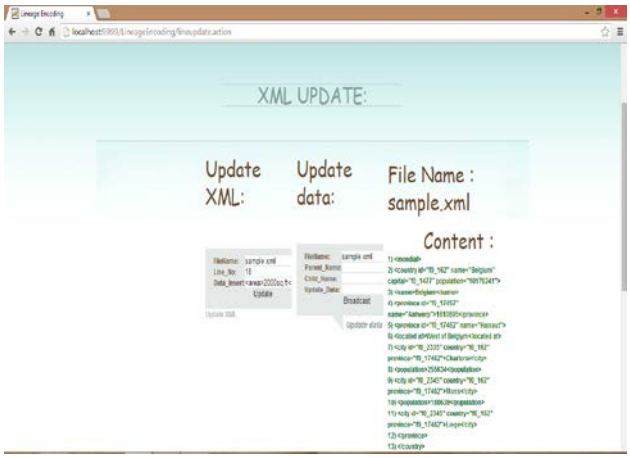

Fig.10 XML Updation

6 Performance Evaluation

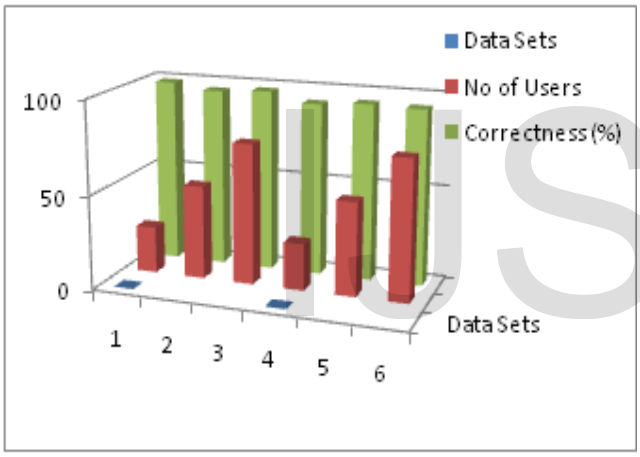

Fig.11 Correctness of real XML Data Set & Synthetic Data Set

Here, evaluated the performance of this scheme in the experiments, compared not only to the previous wireless XML streaming methods but to conventional XML query processing methods supporting twig pattern matching. Here, used a real XML data set and a synthetic data set for the correctness of experiments. This scheme is effective and efficient in terms of the access time and tuning time. Also showed conventional XML query processing methods are inefficient in the wireless mobile environment due to their huge indices.

7 Conclusion and Future Work

Twig pattern queries containing complex conditions are popular and critical in XML query processing. This paper presents an efficient wireless XML streaming method supporting twig pattern queries. The previous work on wireless XML streaming only addressed processing of simple path queries. Thus, they are inefficient for twig pattern queries. In contrast, it provides an energy and latency efficient way to evaluate predicates and twig pattern matching. Specifically, it reduces the size of the XML stream, exploiting the benefits of the structure indexing and attributes summarization. In addition, it reduces the tuning time as it provides an effective way for selective access of XML elements as ll as their attribute values and texts.

It describes Lineage Encoding to support queries involving predicates and twig pattern matching. also defined the relevant operators and functions to efficiently process twig pattern matching. The mobile client can retrieve the required data satisfying the given twig pattern by performing bit-wise operations on the Lineage Codes in the relevant G-nodes. Thus, it can support twig pattern query processing while providing both energy and latency efficiencies

In the future, plan to analyze the issues that were not fully addressed in this paper. First, depth-first traversal of elements increases the access time for specific queries. Second, as communication is not stable in wireless broadcasting environment, the indexing mechanism should consider network failures such as tail drops and packet losses.

Acknowledgment

The author thanks all those who have helped to make this paper possible and who contributed to the content of this paper. The author thanks the revisers who helped in improving quality of paper.

References

1. S. Acharya, R. Alonso, M. Franklin, and S, Zdonik, "Broadcast Disks: Data Management for Asymmetric Communication Environments, " Proc. ACM SIGMOD Int'l Conf. Management of Data Conf., pp. 199-210, Mar. 1995.

2. S,Chen, H. Li, J. Tatemura, W. Hsiung, D. Agrawal, and K.S. Candan, "Scalable Filtering of Multiple Generalized-Tree-Pattern Queries over XML Streams," IEEE Trans. Knowledge and Data Eng., vol. 20, no. 12, pp. 1627-1640, Dec. 2008.

3. Y.D. Chung, S. Yoo, and M.H. Kim, "Energy- and Latency-Efficient Processing of Full-Text Searches on a Wireless Broadcast Stream," IEEE Trans. Knowledge and Data Eng., vol. 22, no. 2, pp. 207-218, Feb. 2010.

4. M. Francechet, "XPathMark: An XPath Benchmark for XMark Generated Data," Proc. Third Int'l Conf. Database and XML Technologies (XSYM), 2005.

5. T. Imielinski, S. Viswanathan, and B. Badrinath, "Energy Efficient Indexing on Air," Proc. ACM SIGMOD Int'l Conf. Management of Data Conf., pp. 25-36, 1994.

6. H. Jiang, W, Wang, H. Lu, and J. Yu, "Holistic Twig Joins on Indexed XML Documents," Proc. Int'l Conf. Very Large Data Bases (VLDB), pp. 273-284, 2003.

7. J.P. Park, C.S. Park, and Y.D. Chung, "Attribute Summarization: A Technique for Wireless XML Streaming," Proc. Int'l Conf. Interaction Sciences, pp. 492-496, Dec. 2009.

8. J.P. Park, C.S. Park and Y.D. Chung, :Energy and Latency Efficient Access of Wireless XML Stream," J. Database Management, vol. 21, no. 1, pp. 58-79, 2010.

9. W. Wang, H.Wang, H. Lu, X. Lin, and J. Li, "Efficient Processing of XML Path Queries Using the Disk-Based F&B Index," Proc. Int'l Conf. Very Large Data Bases (VLDB), pp. 145-156, 2005.

10. For real XML Data Set, http://www.cs.washington.edu/research/xmldatasets/

11. For Synthetic Data Set, https://www.epimodels.org/midas/pubsyntdata1.do, http://ndssl.vbi.vt.edu/synthetic-data/, www.cs.ubc.ca/labs/db/heptox/exp_xmark.htm