JOURNAL OF INFORMATION SCIENCE AND ENGINEERING 32, 309-327 (2016)

DOI:10.1688/JISE.2016.32.2.4

A Novel Replication Strategy for Efficient XML Data Broadcast in Wireless Mobile Networks

ALI BORJIAN BOROUJENI[1] AND MEGHDAD MIRABI[2]
[1]*School of Computer Engineering*
Iran University of Science and Technology
Tehran, Iran
[2]*Department of Computer Engineering*
Faculty of Engineering
Islamic Azad University
South Tehran Branch, Tehran, Iran
E-mail: {web.ebox; meghdad.mirabi}@gmail.com

Recently, the use of XML for data broadcasting in mobile wireless networks has gained many attentions. In these networks, a stream of XML data is broadcasted via a wireless channel, and mobile clients access the broadcast stream using energy-restricted portable devices. Several indexing methods have been proposed to selectively access XML data over a broadcast stream. Although existing indexing methods improve the performance of XML query processing in terms of access time and tuning time but they do not use a replication strategy to replicate the indexes in the broadcast XML stream. In this paper, we propose a novel replication strategy for XML data broadcast called Triangle-based Replication (TR) strategy which replicates the partial and relevant parts of indexes into suitable positions in the broadcast XML stream. Experimental results show that our proposed XML replication strategy has better performance compared to existing XML replication strategies.

***Keywords*:** indexing, replication, wireless broadcast channel, XML query processing, XML stream

1. INTRODUCTION

Today, the use of wireless networks with simplicity and the many benefits that they provide to their users is rapidly expanding. Make communication between portable devices without the need of using cables, the possibility of moving devices, low cost, easy implementation, and simple installation, are some of the benefits of wireless networks [1-4].

In the broadcast data access, data is first divided into small packets which form a broadcast stream. Then, this broadcast stream is gradually placed on a wireless broadcast channel. Generally, each data packet has a related index (or address) in the broadcast stream. This index (or address) is used to find the exact location of data in the stream [4-9].

Typically, broadcast data access can be classified into two modes. They are on-demand mode and push-based mode [1-3]. In the on-demand broadcasting mode, mobile clients send their queries to the broadcast server via the uplink channel and the broadcast server considers all the pending queries and decides the content and order of data for the next broadcast cycle in the downlink channel. In this broadcasting mode, the cost of

Received November 14, 2014; revised February 23 & April 19, 2015; accepted May 23, 2015.
Communicated by Ce-Kuen Shieh.

sending a query by a mobile client is much more than the cost of receiving the query results from the broadcast channel. Therefore, mobile clients consume a lot of energy to send their queries to the broadcast server. In the push-based method, mobile clients do not send their queries to the broadcast server. They constantly listen to the broadcast channel and as soon as the indexes (or addresses) of desired data reach, these indexes (or addresses) have been considered to access the desired data in the broadcast stream. Therefore, in this broadcasting mode, there is no extra cost for sending queries from mobile clients to the broadcast server.

In order to calculate the total amount of consumed energy of a mobile client to access the desired data in a broadcast, tuning time as a performance metric is defined. It is the total time that a mobile client remains in the active mode to get the desired data from the broadcast stream on the air. As much as a mobile client remains in the active mode to get the desired data, it consumes more amounts of energy. In order to calculate the total amount of latency for receiving the desired data over the broadcast channel, access time as a performance metric is defined. It is the interval between the times that a mobile client submits its query on the air to the time that the mobile client receives the query results from the broadcast channel [1-3].

As XML (eXtensible Markup Language) [10] is emerging as a standard for data dissemination over the Internet, the use of XML for data broadcasting in wireless networks is rapidly increasing. There are many applications which use XML for data broadcasting in wireless environments such as traffic and travel information systems and weather information systems [11].

Several indexing methods have been proposed to selectively access XML data over a broadcast stream in the push-based method such as the methods proposed by [11-16]. Although these indexing methods improve the performance of XML query processing in terms of access time and tuning time but they do not use a replication strategy to replicate the indexes in the XML stream in order to further reduce the access time in processing of XML queries.

Chung and Lee [17] proposed an indexing method called (1, X) indexing method to selectively access XML data over a broadcast stream in mobile wireless networks. The main idea of the (1, X) indexing method is the same as the (1, M) indexing method proposed by [1]. In the (1, M) indexing method, a global index is constructed for the whole flat data and then it is placed in every $1/M$ fraction of the broadcast data. The index of the (1, X) indexing method is the same as the (1, M) indexing method except the identifier of data records in flat data is replaced with the path information in the XML document and the address of data is replaced by the pair of start and end addresses of elements in the XML document. Therefore, the index of the (1, X) method consists of a triple of <path expression, start address, end address> which is replicated X times in the broadcast. Note that the index of the (1, X) indexing method is replicated many times (X times) while the number of indexes used by a mobile client in each broadcast cycle is exactly one. It means that the duplicated indexes are not useful to the mobile client. By replicating the global index X times, the total size of broadcast stream in the wireless channel is increased which increases the access time. In order to reduce the access time, Chung and Lee [17] replicated the partial and relevant parts of the global index into suitable positions in the broadcast stream. They first constructed two different XML trees: an XML data tree and an XML index tree. Then, they proposed three different replication strate-

gies called PP (Path of index, Path of data), TT (Tree of index, Tree of data), and TP (Tree of index, Path of data) based on these two XML trees. In the PP strategy, the partial indexes (*i.e.* the paths of index nodes in the XML index tree) and the partial data (*i.e.* the paths of data nodes in the XML data tree) are replicated when the broadcast stream is generated. In the TT strategy, the global indexes (*i.e.* a sub-tree with the height h of the XML index tree) and the global data (*i.e.* a sub-tree with the height h of the XML data tree) are replicated. In the TP strategy, the global indexes and the partial data are replicated.

In this paper, we propose a replication strategy for XML data broadcast in the push-based mode called Triangle-based Replication (TR) strategy which replicates the partial and relevant parts of the indexes into suitable positions in the broadcast stream. In this strategy, the total number of replicated nodes is determined based on the importance of XML nodes in the related XML tree. Since the total number of requests for each XML node in a broadcast stream corresponds to the total number of root-to-node paths which contains that XML node, the TR strategy adjusts the replication rate for each XML node based on this factor that we called it, Visit Rate Factor. Hence, the main contributions of this paper are summarized as follows:

- We define a new replication strategy for XML data broadcast called Triangle-based Replication (TR) strategy which replicates the partial and relevant parts of indexes into suitable positions in the broadcast XML stream.
- We propose an algorithm to generate a broadcast XML stream based on the TR strategy.
- We introduce different parameters and performance metrics which must be taken into account when evaluating the efficiency of different replication strategies.
- We compare the efficiency of TR strategy with the existing replication strategies (*i.e.*, PP, TT, and TP) by performing several experiments using different XML data sets.

The rest of this paper is organized as follows: in Section 2, the (1, X) indexing method is explained in details since our proposed replication strategy can be applied to the (1, X) indexing method. In Section 3, the proposed replication strategy (TR strategy) for XML data broadcast over a wireless channel is explained. In Section 4, we introduce different parameters and performance metrics for evaluating the efficiency of different replication strategies for XML data broadcast in wireless mobile networks. In Section 5, we compare the efficiency of our proposed replication strategy (TR strategy) with the existing replication strategies (*i.e.* PP, TT, and TP) using different XML data sets. Finally, in Section 6, we conclude the paper with a conclusion and some directions on future works.

2. (1, X) INDEXING METHOD

Generally, an XML document can be modeled by a tree structure. In this tree structure, elements are represented by nodes and Parent-Child (P-C) relationships between the elements are represented by edges.

To show that how the (1, X) indexing method works, we use a sample of the XML tree shown in Fig. 1. For simplicity of explanation, we suppose that this XML tree is a fully balanced tree with a fixed fan out (n=3) and height (n=3).

Definition 1: The root-to-node path of a node α in an XML tree T is a sequence of node names (or tag names) from the root node r to the node α which are separated by "/".

For example, the root-to-node path of the node c_3 in the XML tree illustrated in Fig. 1 is the path. "/*Root*/a_1/b_2/c_3."

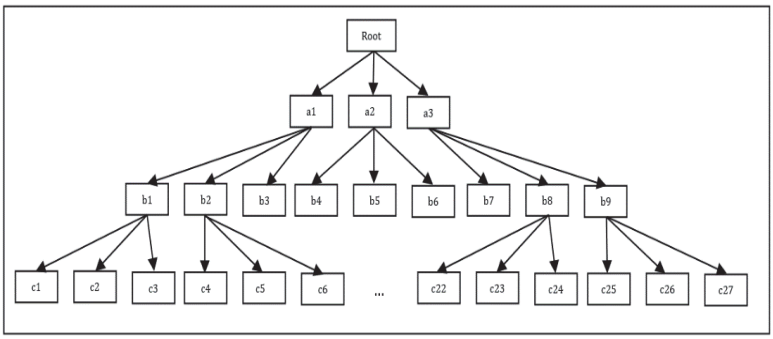

Fig. 1. A sample of the XML tree.

In the (1, X) indexing method, two XML trees are constructed (*i.e.*, an XML index tree and an XML data tree) and then the broadcast XML stream is constructed based on these two XML trees.

In the XML data tree, the structure of each XML node contains the following three fields:

- Element: It is a pair of the start and end tags with their attributes and/or its text.
- Child-Link: It is the address of its first child node (*i.e.* left-most child node) in the XML data tree. It will be equal to null in the case of a leaf node.
- Sibling-Link: It is the address of its next sibling node (*i.e.* right sibling node) in the XML data tree. It will be equal to null in the case that there is no next sibling node.

By exploiting this structure for each XML node in the XML tree, we can access the children of an XML node by following its Child-Link field that points to the address of the first child node, and then continue the path with the next sibling node that its address is stored in the Sibling-Link field. To retrieve the original XML document, we just need to traverse the XML data tree with a Depth First Search (DFS) algorithm and insert the Element fields of the child nodes between the start and end tags of their parents.

In the XML index tree, the structure of each XML node contains the following five fields:

- Path-Specification: It is a path expression that demonstrates an XML element.
- Data-Link: It is the address of the XML node in the XML data tree which is determined by the Path-Specification field.
- Child-Link: It is the address of its first child node (*i.e.*, left-most child node) in the XML index tree. It will be equal to null in the case of a leaf node.
- Sibling-Link: It is the address of its next sibling node (*i.e.*, right sibling node) in the XML index tree. It will be equal to null in the case that there is no next sibling node.

- Homonym-Link: It is the address of its next homonym node. The homonym nodes in an XML document are the XML nodes that have the same tag name but they are in different positions in the XML tree. This field will be equal to null in the case that there is no homonym node.

The XML data tree for the XML tree shown in Fig. 1 is illustrated in Fig. 2 and its XML index tree is illustrated in Fig. 3.

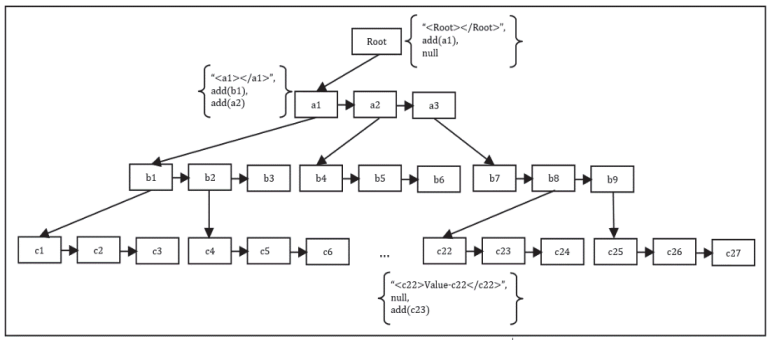

Fig. 2. A sample of the XML data tree.

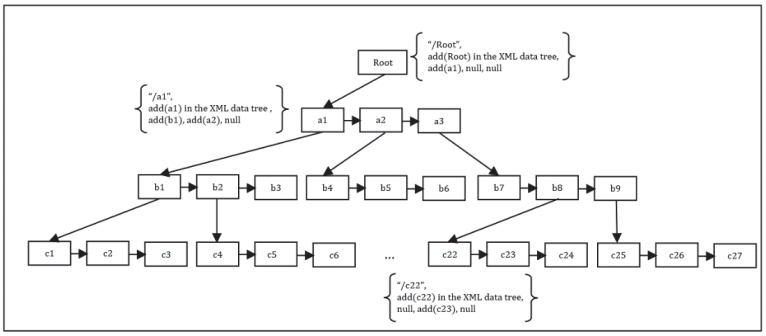

Fig. 3. A sample of the XML index tree.

The process of generating a broadcast XML stream is based on these two XML trees. This process is dependent on the algorithm that the replication strategies (*i.e.* PP, TT, and TP) use. In the following, we only explain the (1, *X*) indexing method and do not explain the different replication strategies (*i.e.* PP, TT, and TP) in order to save the length of the paper. Refer to [17] for more information.

In the (1, *X*) indexing method, a global index for the XML tree is first constructed and then it is placed in every 1/*X* fraction of the broadcast XML stream, where the index contains the addresses of all the XML nodes in the XML tree. In this indexing method, the XML data tree is divided into *X* fractions. For example, if we consider the (1, *X*) indexing method with $X = 4$ then there are a global index which should be replicated four times (*i.e.* $I1$, $I2$, $I3$, and $I4$) in the broadcast XML stream and each of them is placed in front of every 1/4 fraction of the data (*i.e.*, $D1$, $D2$, $D3$, and $D4$). Note that all the index-

es are the same which means $I1=I2=I3=I4$. It should be mentioned that the total XML data can be retrieved from $D1+D2+D3+D4$ since we divided the XML data tree into four fractions. Therefore, the broadcast XML stream generated by the (1, X) indexing method is as follows: ($I1$, $D1$, $I2$, $D2$, $I3$, $D3$, $I4$).

Example 1: For the XML tree shown in Fig. 1, the broadcast XML stream with the (1, X) indexing method with X=4 is shown in Fig. 4. In Fig. 4, the white rectangles represent the index nodes while the gray rectangles represent the data nodes.

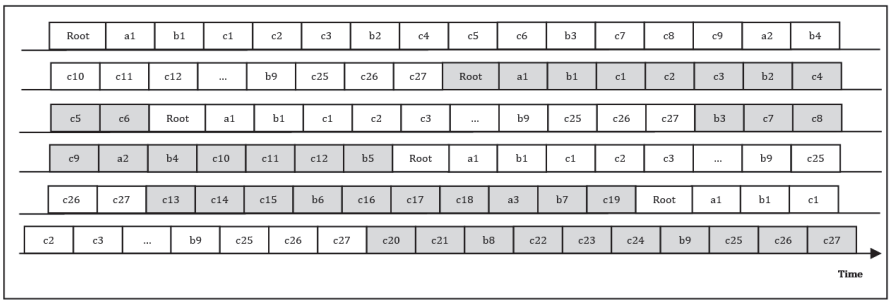

Fig. 4. A sample of the broadcast XML stream using the (1, X) indexing method.

3. THE TRIANGLE-BASED REPLICATION STRATEGY

In the Triangle-based Replication strategy (TR strategy), the XML tree is first traversed based on the Breadth First Search (BFS) algorithm which traverses the XML tree from top to down and level by level. In each level, the XML nodes in that level are considered and if the total number of XML nodes in that level is greater than or equal to the level number, the partial paths from the root node to the specified nodes are constructed. By aggregating the XML nodes in these paths, a sub-tree from the original XML tree is constructed which forms a triangular shape (triangle name in the TR strategy comes from here). Then, the constructed sub-tree is traversed based on the Depth First Search (DFS) algorithm which traverses the sub-tree node by node. The process of generating a broadcast XML stream with the TR strategy is as follows: for each triangle, the index of each XML node with its relevant data is placed in the broadcast XML stream.

Example 2: For the XML tree shown in Fig. 1, the broadcast XML stream with the TR strategy contains three triangles as follows:

Triangle1: Root, a1, a2, a3
Triangle2: Root, a1, b1, b2, b3, a2, b4, b5, b6, a3, b7, b8, b9
Triangle3: Root, a1, b1, c1, c2, c3, b2, c4, c5, c6, b3, c7, c8, c9, a2, b4, c10, c11, c12, b5, c13, c14, c15, b6, c16, c17,c18, a3, b7, c19, c20, c21, b8, c22, c23, c24, b9, c25, c26, c27

The generated broadcast XML stream with the TR strategy for the XML tree shown in Fig. 1 is shown in Fig. 5. In Fig. 5, the white rectangles represent the index nodes while the gray rectangles represent the data nodes.

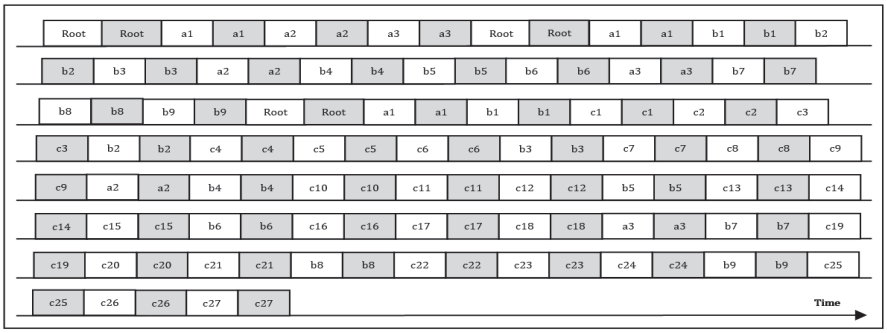

Fig. 5. A sample of the broadcast XML stream using the TR replication strategy.

The process of generating a broadcast XML stream with the TR strategy for the unbalanced XML tree shown in Figs. 6 (a)-(e) is explained below.

Consider the XML tree illustrated in Fig. 6 (a). This XML tree contains 18 XML nodes, distributed in 5 levels, from level 1 to level 5. The root level is considered to be in level 1. The TR strategy starts its work from level 2. There are 3 XML nodes in level 2 (*i.e.*, a1, a2, and a3). The gray nodes in Fig. 6 (b) show the triangle constructed in this step since the total number of XML nodes in level 2 is greater than the level number (3 > 2). By traversing the constructed triangle based on the DFS algorithm, a part of the broadcast XML stream is generated as shown in the Part A of Fig. 7. There are 3 XML nodes in level 3 (*i.e.* *b*1, *b*2, and *b*3) and the total number of XML nodes in this level is equal to the level number (3 = 3). The gray nodes in Fig. 6 (c) show the triangle constructed in this step. It should be noted that the TR strategy just considers the XML nodes and their ancestors in each level. Therefore, in this level, the XML node *a*2 is omitted in constructing the second triangle. By traversing the constructed triangle based on the DFS algorithm, a part of the broadcast XML stream is generated as shown in the Part B of Fig. 7. There are 3 XML nodes in level 4 (*i.e.*, c1, c2, and c3) as shown in Fig. 6 (d). The TR strategy does not construct a new triangle and it just considers the XML nodes in this level since the total number of XML nodes in this level is not greater than or equals to the level number (3 < 4). By considering the XML nodes in this level, a part of the broadcast XML stream is generated as shown in the Part C of Fig. 7. Finally, there are 8 XML nodes in level 5 (*i.e.*, d1, d2, d3, d4, d5, d6, d7, and d8). The TR strategy constructs the third triangle as shown in Fig. 6 (e). The TR strategy does not consider the XML nodes *a*2 and *b*3 since they are not the ancestors for any of the XML nodes in level 5. By traversing the constructed triangle based on the DFS algorithm, a part of the broadcast XML stream is generated as shown in the Part D of Fig. 7.

Fig. 8 shows the TriangleStreamGenerator algorithm which is devised to generate an XML stream based on our proposed replication strategy (TR strategy). In this algorithm, we use the XML Reader from the Microsoft .NET Framework that provides fast, non-cached access to XML data. In this algorithm, the input is a well-formed XML document (*i.e.*, XI) and the output is a broadcast stream (*i.e.* XS). In the Triangle-Stream-Generator algorithm, the depth of the XML tree is determined from the XML document XI (Line 1). To keep the XML nodes inside of each triangle, a temporary XML Document structure is used called tempXMLDocument (Line 2). In the Tri-

angleStreamGenerator algorithm, the related XML tree is traversed by the BFS algorithm from level 2. It should be noted that the root node is not considered as a separated triangle in the TR strategy. Therefore, the process of finding the triangles in the related XML tree is started from level 2 (Line 3).

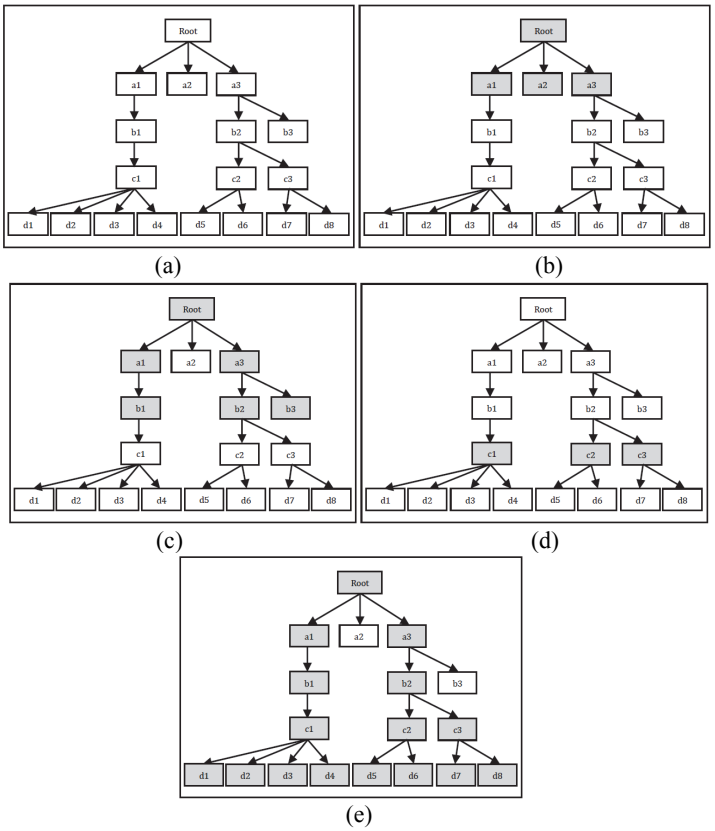

Fig. 6. A sample of the unbalanced XML tree.

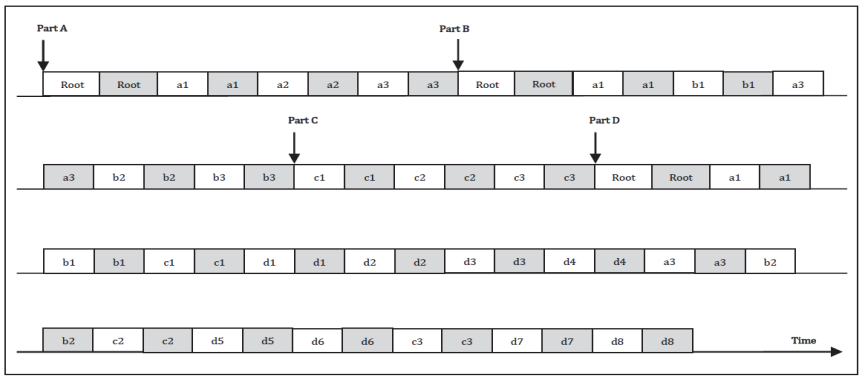

Fig. 7. A sample of the broadcast XML stream using the TR replication strategy.

At the beginning of traversing each level, the tempXMLDocument structure is cleared since this structure is used to store the XML nodes for a new triangle (Line 4). A triangle is constructed just when the total number of XML nodes in a level is greater than or equal to the level number. Therefore, the algorithm examines this condition and if the condition is satisfied (Line 5), a new triangle is constructed from the XML nodes in this level and their ancestors. Otherwise, the algorithm just considers the XML nodes in this level and omits their ancestors (Lines 13-17). In the TriangleStreamGenerator algorithm, all the XML nodes in each level as well as their ancestors are found and added into the tempXMLDocument structure in the case that they are not already exist in the temp-XMLDocument structure (Lines 6-10). Then, the XML nodes in the triangle are ordered based on the DFS algorithm (Line 11) and the broadcast stream XS is generated by adding the index of each XML node in the triangle followed by its data in the broadcast stream (lines 18-21). Finally, the broadcast stream XS is returned (Line 23).

```
Algorithm TriangleStreamGenerator
Input: A Well-Formed XML Document (XI)
Output: XML Stream (XS)
1. depth = setDepth (XI);
2. tempXMLDocument = create a new empty stream to store the XML nodes of each triangle;
3. for (int i = 2; i <= depth; i++){
4.   clearContent (tempXMLDocument);
5.   If ((the total number of XML nodes in the XML document XI Where
              Depth(XML nodes) equals to i) >=   i) {
6.       foreach (node C in the XML document XI Where Depth (node C) equals to i)
7.            foreach (node B in AncestorsOrSelf of node C)
8.                 if (tempXMLDocument does not contain node B) {
9.                      tempXMLDocument.Add (node B);
10.                     }
11.      Order tempXMLDocument by the DFS algorithm;
12. }
13. else {
14.      foreach (node C in XI Where Depth (node C) equals to i){
15.           tempXMLDocument.Add (node C);
16.      }
17. }
18. foreach (node C in tempXMLDocument) {
19.           add the index of node C to the XML stream XS;
20.           add the data of node C to the XML stream XS;
21. }
22. }
23. Return the XML stream XS;
```

Fig. 8. TriangleStreamGenerator Algorithm.

4. ANALYSIS

In this section, we define different parameters and performance metrics which must be taken into account when evaluating the efficiency of different replication strategies for the (1, *X*) indexing method.

4.1 Visit Rate Factor

In the different replication strategies (*i.e.*, PP, TT, TP, and TR), a broadcast XML stream is generated by replicating the index and data nodes through the stream. This is important to know what nodes should be replicated, in what places, how many times, and

in what orders. In this paper, we introduce "Visit Rate Factor" as a parameter which affects the performance of different replication strategies. The Visit Rate Factor is used to find the priority of each XML node in an XML document. The Visit Rate Factor for the XML node α in the XML tree T is defined by the total number of root-to-node paths that this XML node exists in. It is defined as follows:

Definition 2: The Visit Rate Factor of the XML node α in the XML tree T is defined as follows:

$$\text{Visit Rate Factor } (\alpha) = \text{The Total Number of Child and Descendant Nodes of Node } (\alpha) + 1. \quad (1)$$

Based on the Definition 2, the minimum Visit Rate Factor in the XML tree T is for the leaf nodes (*i.e.* equal to 1). It is clear since there is just one root-to-node path for each leaf node in the XML tree T. The maximum Visit Rate Factor is for the root node since the root node appears in all the root-to-node paths. By exploiting the Visit Rate Factor, we can measure the priority of each XML node in the XML tree T.

4.2 Tuning Time

The tuning time is the sum of times that a mobile client elapses in the active mode to retrieve the desired data (*i.e.* index information and XML data). This performance metric estimates the energy consumption of mobile clients in processing the XML queries.

In order to make the formula for the tuning time, we need to define the Bcast as follows:

Definition3: The Bcast is a set of nodes that contains at least one index node and at least one data node for each element in the XML dataset.

Definition 4: The tuning time of a mobile client to retrieve the XML node α in the XML tree T is defined as follows:

$$TurningTime(\alpha) = \frac{\sum_{i=2}^{m}(A_i - A_{i-1} - 1)(A_i - A_{i-1})}{2n} + \frac{(n - A_m + A_1 - 1)(n - A_m - A_1)}{2n}. \quad (2)$$

- A_i: The position of the ith index (replication number i) of the XML node α in the Bcast;
- n: The length of the Bcast;
- m: The total number of index replications for the XML node α in the Bcast;

It should be noted that we only calculate the tuning time of a mobile client as the time elapsed to reach the index information of XML node α since the total time elapsed to download the index of XML node α and its data in all the replication strategies (*i.e.*, TT, PP, TP, and TR) is the same since all the replication strategies use the structure of (1, X) indexing method for the index and XML nodes in the XML tree.

***Proof*:** To prove the accuracy of Eq. (2), consider a broadcast XML stream like the stream shown in Fig. 9.

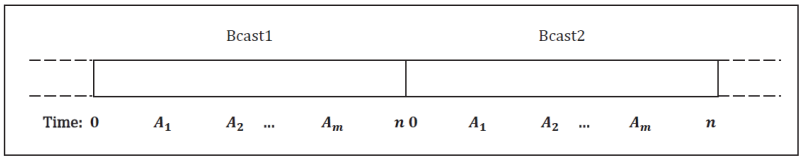

Fig. 9. A sample of the broadcast XML stream.

Assume that the total number of replicated indexes for the XML node α in a Bcast is m and these indexes arrives at the times A_1, A_2,…, A_m. Now, suppose that the mobile client c_1 sends its request to retrieve the XML node α on the air at the time that the first packet from the Bcast is over the wireless channel. Therefore, the mobile client c_1 stays in the active mode until the arrival time of index A_1. Now, suppose that the mobile client c_1 sends its request to retrieve the XML node α when the second packet from the Bcast is over the wireless channel. Therefore, the waiting time of the mobile client c_1 is equal to A_1-1. This will continue until the mobile client c_1 sends its request to retrieve the XML nodeα when the index A_1 is over the wireless channel. At this case, the waiting time of the mobile client c_1 is equal to zero because the mobile client c_1 receives its desired address from the index A_1 at the time of submitting its request. To calculate the tuning time, these series of the numbers are added together and then the result is divided to the length of the Bcast (*i.e.* n). Therefore, we have:

$$\textit{Tining Time}(\alpha)=$$

$$\frac{\left[(A_2-A_1-1)+(A_2-A_1-2)+\ldots+(2)+(1)+(0)\right]+\left[(A_3-A_2-1)+(A_3-A_2-2)+\ldots+(2)+(1)+(0)\right]+\ldots}{n}$$

$$\frac{\left[(A_m-A_{m-1}-1)+(A_m-A_{m-1}-2)+\ldots+(2)+(1)+(0)\right]+\left[(n-A_m-A_1-2)+(A_3-A_2-2)+\ldots+(2)+(1)+(0)\right]}{n}$$

$$=\frac{\left[\frac{(A_2-A_1-1)(A_2-A_1)}{2}\right]+\left[\frac{(A_3-A_2-1)(A_3-A2)}{2}\right]+\ldots+\left[\frac{(A_m-A_{m-1}-1)(A_m-A_{m-1})}{2}\right]+\left[\frac{(n-A_m-A_1-1)(n-A_m-A_1)}{2}\right]}{n}$$

$$\Rightarrow \textit{Tining Time}(\alpha)=\frac{\sum_{i=2}^{m}(A_i-A_{i-1}-1)(A_i-A_{i-1})}{2n}+\frac{(n-A_m-A_1-1)(n-A_m-A_1)}{2n}. \qquad \square$$

Based on Eq. (2), it is deduced that the tuning time of a mobile client to retrieve the XML node α depends on the distance between the tandem replicated indexes of the XML node α (*i.e.*, $A_i-A_{i-1}-1$). It also depends on the length of the Bcast (*i.e.*, n).

Definition 5: Assume that the total number of XML nodes in the XML tree T is t. The Arithmetic Average Tuning Time for the Bcast b is defined as follows:

$$Arithmetic\ Average\ Tuning\ Time(b)=\frac{\sum_{i=1}^{t} Tuning\ \ Time(i)}{t}. \tag{3}$$

It is the arithmetic average of the sum of tuning times of all the XML nodes in the XML tree *T*.

As explained in Section 4.1, the XML nodes have different priorities based on their Visit Rate Factors. In order to measure the average tuning time based on this criterion, the Weighted Average Tuning Time is defined as follows:

Definition 6: Assume that the total number of XML nodes in the XML tree *T* is *t*. The Weighted Average Tuning Time for the Bcast *b* is defined as follows:

$$Weighted\ Average\ Tuning\ Time(b)=\frac{\sum_{i=1}^{t} Tuning\ \ Time(i)\times Visit\ \ Rate\ \ Factor\ \ (i)}{t\sum_{i=1}^{t} Visit\ \ Rate\ \ Factor\ \ (i)}. \tag{4}$$

The weighted average tuning time is a more accurate performance metric to calculate the average tuning time compared to the arithmetic average tuning time since it considers the weight of each XML node in the calculation.

4.3 Distance Time

Once a mobile client sends its request on the air, it stays in the active mode until the arrival time of the related index. Since the index indicates the position of XML data in the broadcast XML stream, the mobile client switches to the doze mode until the arrival time of the related XML data. When the related XML data is arrived, the mobile client switches to the active mode to download the desired XML data. The interval between the times that the mobile client reaches to the index until the arrival time of the related XML data is called distance time. It is defined as follows:

Definition 7: The Distance Time is the interval between the times that a mobile client reaches to the index until the arrival time of the related XML data in the broadcast XML stream. The Distance Time of a mobile client to retrieve the XML node α in the XML tree *T* is defined as follows:

$$DistanceTime(b)=\frac{\sum_{i=1,j=1}^{m,k}(A_i-A_{i-1})}{n} \text{ where } A_0=0. \tag{5}$$

A_i: The position of the *i*th index (replication number *i*) of the XML node α in the Bcast;
D_i: The position of the *i*th XML data (replication number *i*) of the XML node α in the Bcast;
n : The length of the Bcast;
m: The total number of index replications for the XML node α;
k : The total number of data replications for the XML node α;

***Proof*:** To prove the accuracy of Eq. (5), consider a broadcast XML stream like the stream shown in Fig. 10.

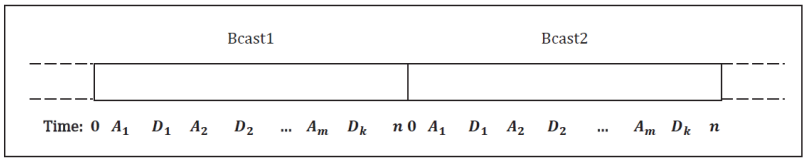

Fig. 10. A sample of the broadcast XML stream.

Assume that the total number of replicated indexes for the XML node α in a Bcast is m and these indexes arrive at the times A_1, A_2, …, A_m. Also, assume that the total number of replicated data for the XML node α in a Bcast is k and these data arrives at the times D_1, D_2, …, D_k. Now, suppose that the mobile client c_1 sends its request to retrieve the XML node α on the air between the time (A_1–0). Therefore, the distance time of the mobile client c_1 for this period of time is equal to (D_1–A_1). Now suppose that the mobile client c_1 sends its request to retrieve the XML node α on the air between the time (D_1–A_1). In this case, the mobile client c_1 does not have the index of the XML node α. Therefore, it has to stay in the active mode until the time A_2 that is the next nearest replicated index of the XML node α. Thus, the distance time of the mobile client c_1 for this period of time is equal to (D_2–A_2). In the other hand, the distance time of the mobile client c_1 for the time (A_2–D_1) is equal to (D_2–A_2). Therefore, it is clear that the distance time of the mobile client c_1 for the requests between the tandem replicated indexes A_1 and A_2 is equal to (D_2–A_2). Thus, for two tandems replicated indexes A_{i-1} and A_i, the distance time of the mobile client c_1 is equal to (D_j–A_i) where D_j is the first replicated data of the XML node α after the index A_i in the broadcast XML stream. For the broadcast XML stream shown in Fig. 15, the distance time of the mobile client c_1 to retrieve the XML node α is as follows:

$$
\begin{aligned}
&Distance\,Time(\alpha) \\
&= \frac{(A_1 - 0)\times(D_1 - A_1) + (A_2 - A_1)\times(D_2 - A_2) + \ldots + (A_m - A_{m-1})\times(D_k - A_m)}{n} \\
&\Rightarrow \frac{\sum_{i=1,j=1}^{m,k}(A_i - A_{i-1})(D_j - A_i)}{n} \text{ where } A_0 = 0.
\end{aligned}
$$

❑

4.4 Access Time

The access time is the period of time which a mobile client elapses from the moment of submitting a request to the moment that it retrieves the desired data over the wireless channel. This performance metric estimates the access efficiency of mobile clients in processing the XML queries.

Definition 8: The access time of a mobile client to retrieve the XML node α in the XML tree T is defined as follows:

$$Access\ Time\ (\alpha) = Turning\ Time\ (\alpha) + Distance\ Time(\alpha). \quad (6)$$

It should be noted that we only calculate the access time of a mobile client as the time elapsed to reach the index information of XML node α as well as the time elapsed to reach the XML node α since the total time elapsed to download the index of XML node α and its data in all the replication strategies (*i.e.*, TT, PP, TP, and TR) is the same since all the replication strategies use the structure of (1, *X*) indexing method for the index and XML nodes in the XML tree.

***Proof*:** To prove the accuracy of Eq. (6), consider a broadcast XML stream like the stream shown in Fig. 11. Now, suppose that the mobile client c_1 sends its request on the air at the time q. The mobile client c_1 has to stay in the active mode until the arrival time of the nearest replicated index of the XML node α (*i.e.*, A_i). This interval is the tuning time. Once the mobile client c_1 reaches to the index, it switches to the doze mode until the arrival time of the related XML data (*i.e.*, D_j).This interval is equal to (D_j–A_i) and we called it as the distance time. Therefore, it is obvious that the sum of these times (*i.e.turning time+distance time*) is equal to the interval between the time that the mobile client c_1 sends its request on the air and the time that the mobile client c_1 reaches to its desired data. This interval is the defined as the access time. ❑

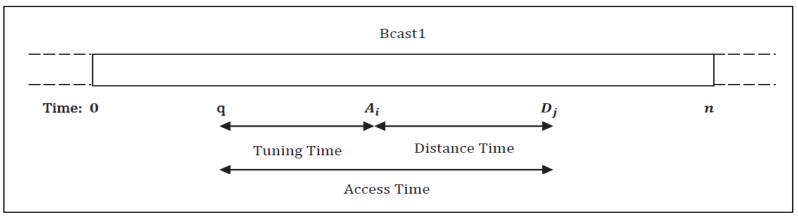

Fig. 11. A sample of the broadcast XML stream.

Definition 9: Assume that the total number of XML nodes in the XML tree T is t. The Arithmetic Average Access Time for the Bcast b is defined as follows:

$$Aritnmetic\ Average\ Access\ Time(b) = \frac{\sum_{i=1}^{t} Access\ Time(i)}{t}. \quad (7)$$

It is the arithmetic average of the sum of access times of all the XML nodes in the XML tree T.

Definition 10: Assume that the total number of XML nodes in the XML tree T is t. The Weighted Average Access Time for the Bcast b is defined as follows:

$$Weighted\ Average\ Acess\ Time(b) = \frac{\sum_{i=1}^{t} Access\ Time(i) \times Visit\ Rate\ Factor(i)}{t\sum_{i=1}^{t} Visit\ Rate\ Factor(i)}. \quad (8)$$

The weighted average access time is a more accurate performance metric to calcu-

late the average access time compared to the arithmetic average access time since it considers the weight of each XML node in the calculation.

5. PERFORMANCE EVALUATION

In this section, we compare the efficiency of different replication strategies (*i.e.*, TT, TP, PP, and TR) by performing several experiments using different XML data sets. It should be noted that our proposed replication strategy (*i.e.* TR) like the existing replication strategies (*i.e.*, TT, TP, and PP) only can be applied to the (1, *X*) indexing method since the structures of XML index tree and XML data tree in all the replication strategies are similar to the (1, *X*) indexing method. It is the reason that why we used the (1, *X*) indexing method in our experiment for all the replication strategies.

All the experiments were conducted on a system with the Intel 3.0 GHz processor and 4GB RAM running on Windows 7 Ultimate 64-bits where all the codes were implemented in C# with the Microsoft .NET Framework 4.0.

5.1 Experimental Setting

To simulate the wireless broadcast channel, we logically modeled the wireless XML stream as a binary file, where the broadcast server writes a byte stream on the file and the mobile clients read the file as a broadcast XML stream.

In our simulation model, we assumed that the broadcast bandwidth is fully utilized for XML data broadcast. To measure the access time and tuning time, we considered only the activity of a mobile client since the activity of a mobile client does not affect the performance at the other mobile clients.

To measure the performance variation based on the types of XML data sets, we used several XML data sets. All the XML data sets were collected from the XML data repository at the University of Washington[1]. Table 1 shows the characteristics of the XML data sets used in our experiments.

Table 1. XML data sets.

Data Set	Size (KB)	Number of Elements	Number of Attributes	Max Depth
321gone	25	311	0	5
Yahoo	26	342	0	5
Ebay	36	156	0	5
University Courses	278	10,546	0	4
SigmodRecord	467	11,526	3,737	6
Shakespeare	1,061	25,339	0	7
XMark	1,155	17,132	3,919	12

We generated the broadcast XML stream for each of the XML data sets based on the different replication strategies which are TT, TP, PP, and TR and then calculated the XML stream size, the arithmetic/weighted average access time, and the arithmetic/weighted average tuning time for each replication strategy.

[1] http://www.cs.washington.edu/research/

5.2 Experimental Results on XML Stream Size

Fig. 12 shows the size of XML stream for the different replication strategies in the 321gone, Yahoo, and Ebay data sets, respectively. As shown in Fig .12, the size of XML stream of the TR replication strategy is less than the other replication strategies on all the XML data sets. It means that the TR replication strategy occupies less space to transmit XML index and XML data.

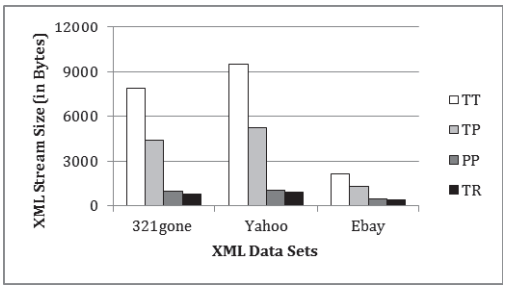

Fig. 12. The size of XML stream.

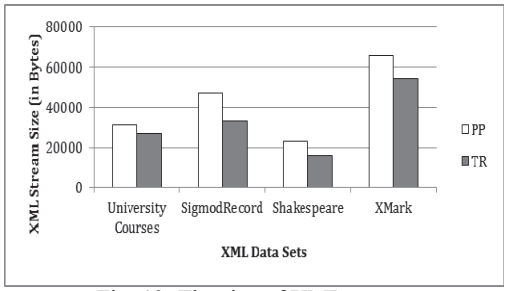

Fig. 13. The size of XML stream.

Fig. 13 shows the size of XML stream for the PP and TR replication strategies in the four XML data sets (*i.e.*, University Courses, SigmodRecord, Shakespeare, and XMark). As shown in Fig. 13, the size of XML stream of the TR replication strategy is less than the PP replication strategy on all the XML data sets. It means that the TR replication strategy occupies less space to transmit XML index and XML data.

5.3 Experimental Results on Access Time

Figs. 14 and 15 show the arithmetic average access time and the weighted average access time for each XML node in the 321gone, Yahoo, and Ebay data sets, respectively.

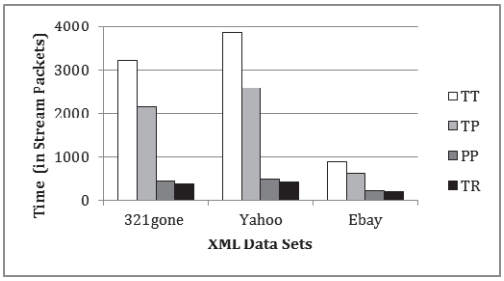

Fig. 14. The arithmetic average access time.

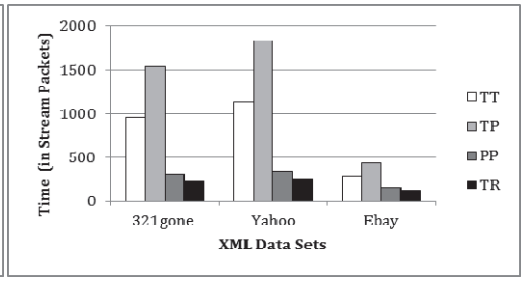

Fig. 15. The weighted average access time.

It is clear that the arithmetic average access time and the weighted average access time in the TT and TP replication strategies are greater than the arithmetic average access time and the weighted average access time in the PP and TR replication strategies. It is because of the following reasons:

- The distances between tandem index and data nodes in the TT and TP replication strategies are large which they increase the arithmetic average access time and the weighted average access time.
- The distances between tandem index replications in the TT and TP replication strategies are large which they increase the arithmetic average access time and the weighted average access time.

Figs. 16 and 17 show the arithmetic average access time and the weighted average access time in the PP and TR replication strategies for each XML node in the four XML data sets (*i.e.* University Courses, SigmodRecord, Shakespeare, and XMark).

As shown in Figs. 16 and 17, the arithmetic average access time and the weighted average access time in the TR replication strategy are smaller than the arithmetic average access time and the weighted average access time in the PP replication strategy. It means that the TR strategy has the best performance in terms of access time.

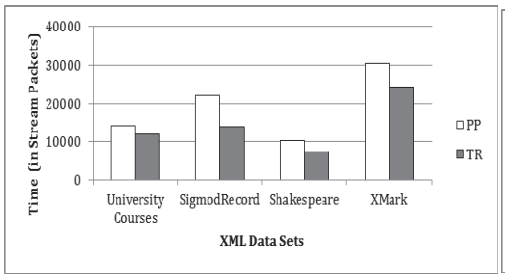

Fig. 16. The arithmetic average access time.

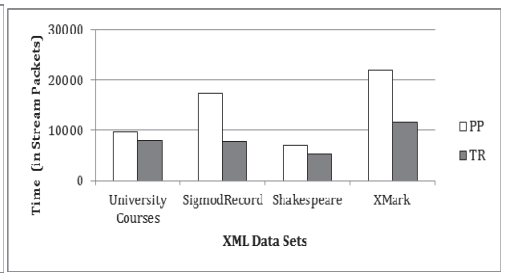

Fig. 17. The weighted average access time.

5.4 Experimental Results on Tuning Time

Figs. 18 and 19 show the arithmetic average tuning time and the weighted average tuning time for each XML node in the 321gone, Yahoo, and Ebay data sets, respectively. It is clear that the arithmetic average tuning time and the weighted average tuning time in the TT and TP replication strategies are not efficient compared with the arithmetic average tuning time and the weighted average tuning time in the PP and the TR replication strategies. It is because the distances between tandem index replications are large in the TT and TP replication strategies which they increase the arithmetic average tuning time and the weighted average tuning time.

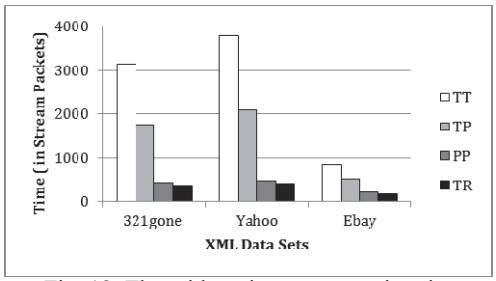

Fig. 18. The arithmetic average tuning time.

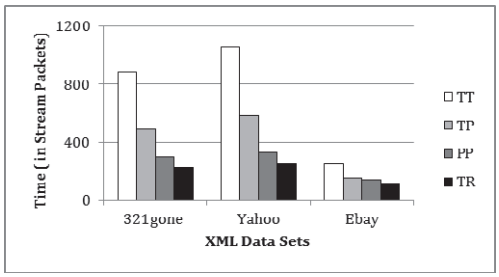

Fig. 19. The weighted average tuning time.

Figs. 20 and 21 show the arithmetic average tuning time and the weighted average tuning time in the PP and TR replication strategies for each XML node in the four XML data sets (*i.e.*, University Courses, SigmodRecord, Shakespeare, and XMark). As shown in Figs. 20 and 21, the arithmetic average tuning time and the weighted average tuning time in the TR replication strategy are smaller than the arithmetic average tuning time and the weighted average tuning time in the PP replication strategy. It means that the TR strategy has the best performance in terms of tuning time.

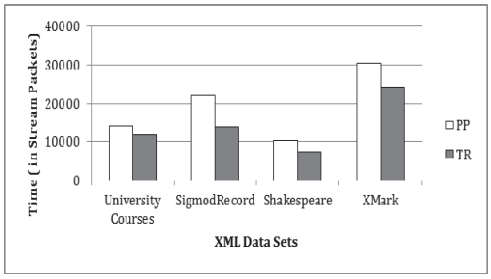

Fig. 20. The arithmetic average tuning time.

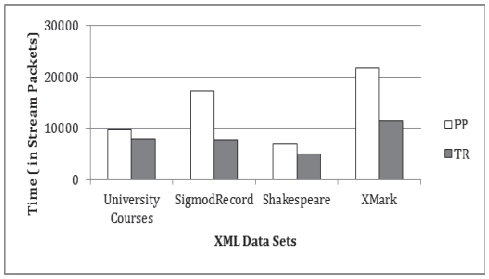

Fig. 21. The weighted average tuning time.

6. CONCLUSION

In this paper, we proposed a new replication strategy for XML data broadcast in wireless mobile networks called TR strategy. We also introduced "Visit Rate Factor" as a parameter which affects the performance of different replication strategies in terms of access time and tuning time. Based on this parameter, we defined the weighted average access time and the weighted average tuning time. By performing several experiments on the different XML data sets, we demonstrated that the TR strategy has the best performance in terms of access time and tuning time among the other replication strategies.

REFERENCES

1. T. Imielinski, S. Viswanathan, and B. R. Badrinath, "Data on air: Organization and access," *IEEE Transactions on Knowledge and Data Engineering*, Vol. 9, 1997, pp. 353-372.
2. S. Acharya, R. Alonso, M. Franklin, and S. Zdonik, "Broadcast disks: Data management for asymmetric communication environments," in *Proceedings of ACM SIGMOD International Conference on Management of Data*, 1995, pp. 199-210.
3. T. Imieĺinski and B. R. Badrinath, "Data management for mobile computing," *SIGMOD Record*, Vol. 22, 1993, pp. 34-39.
4. T. Imielinski, S. Viswanathan, and B. R. Badrinath, "Energy efficient indexing on air," in *Proceedings of ACM SIGMOD International Conference on Management of Data Minneapolis*, 1994, pp. 25-36.
5. Y. D. Chung and M. H. Kim, "An index replication scheme for wireless data broadcasting," *Journal of Systems and Software*, Vol. 51, 2000, pp. 191-199.
6. Y. D. Chung and M. H. Kim, "Effective data placement for wireless broadcast," *Distributed and Parallel Databases*, Vol. 9, 2001, pp. 133-150.

7. M.-S. Chen, K.-L. Wu, and P. S. Yu, "Optimizing index allocation for sequential data broadcasting in wireless mobile computing," *IEEE Transactions on Knowledge and Data Engineering*, Vol. 15, 2003, pp. 161-173.
8. C.-C. Lee and Y. Leu, "Efficient data broadcast schemes for mobile computing environments with data missing," *Information Sciences*, Vol. 172, 2005, pp. 335-359.
9. Y. D. Chung, S. Yoo, and M. H. Kim, "Energy and latency efficient processing of full-text searches on a wireless broadcast stream," *IEEE Transactions on Knowledge and Data Engineering*, Vol. 22, 2010, pp. 207-218.
10. T. Bray, J. Paoli, C. M. Sperberg-McQueen, E. Maler, and F. Yergeau, *Extensible Markup Language (XML) 1.0*, 5th ed., W3C Recommendation, 2008, http://www.w3.org/TR/REC-xml/.
11. J. P. Park, C.-S. Park, and Y. D. Chung, "Energy and latency efficient access of wireless XML stream," *Journal of Database Management*, Vol. 21, 2010, pp. 58-79.
12. C.-S. Park, C. S. Kim, and Y. D. Chung, "Efficient stream organization for wireless broadcasting of XML data," in *Proceedings of the 10th Asian Computing Science Conference on Advances in Computer Science*: *Data Management on the Web*, 2005, pp. 223-235.
13. S.-H. Park, J.-H. Choi, and S. Lee, "An effective, efficient XML data broadcasting method in a mobile wireless network," in *Proceedings of the 17th International Conference on Database and Expert Systems Applications*, 2006, pp. 358-367.
14. J. P. Park, C.-S. Park, M. K. Sung, and Y. D. Chung, "Attribute summarization: A technique for wireless XML streaming," in *Proceedings of the 2nd International Conference on Interaction Sciences: Information Technology*, 2009, pp. 492-496.
15. J. P. Park, C.-S. Park, and Y. D. Chung, "Lineage encoding: An efficient wireless XML streaming supporting twig pattern queries," *IEEE Transactions on Knowledge and Data Engineering*, Vol. 25, 2012, pp. 1559-1573.
16. M. Mirabi, H. Ibrahim, and L. Fathi, "PS+Pre/Post: A novel structure and access mechanism for wireless XML stream supporting twig pattern queries," *Pervasive and Mobile Computing*, Vol. 15, 2013, pp. 3-25.
17. Y. D. Chung and J. Y. Lee, "An indexing method for wireless broadcast XML data," *Information Sciences*, Vol. 177, 2007, pp. 1931-1953.

Ali Bojian Boroujeni obtained his Master in Computer Engineering from the Iran University of Science and Technology in 2015. His research interests include XML data broadcast, privacy preserving, big data analysis, and data mining.

Meghdad Mirabi obtained his Master and Ph.D. degrees in Computer Science from the Universiti Putra Malaysia (UPM), Malaysia in 2009 and 2013, respectively. His research interests include XML data management, wireless information retrieval, and information security.