The Journal of Supercomputing (2022) 78:168–199
https://doi.org/10.1007/s11227-021-03884-4

Efficient XML data placement schemes over multiple mobile wireless broadcast channels

Samaneh Fathi Ozonbolagh[1] · Meghdad Mirabi[2]

Accepted: 11 May 2021 / Published online: 24 May 2021

Abstract

Broadcasting is one of the basic ways to access XML data via mobile wireless networks. In these networks, XML data are disseminated over a wireless broadcast channel and mobile clients can connect to this channel via portable devices and receive the XML data they need. To date, various indexing methods have been proposed to broadcast the XML data over mobile wireless broadcast networks, each of which attempts to reduce the latency and energy consumption of mobile devices in processing the different types of XML queries. Most of the proposed indexing methods use only a single wireless broadcast channel for XML data dissemination. Due to the sequential data access over a wireless broadcast channel, the access time to retrieve the desired XML data over a broadcast channel is longer if the length of generated XML data stream is large. This paper proposes two new XML data placement schemes to disseminate the XML data over multiple wireless broadcast channels by partitioning the XML data stream into several partitions and placing them into multiple wireless broadcast channels. The goal of both of which is a further reduction of the access time during the process of XML querying over the broadcast channels. According to the experimental results, the proposed XML data placement schemes have achieved the goal of further reducing the access time, especially when the size of XML document to be disseminated is large.

Keywords Indexing · Mobile wireless networks · Multiple broadcast channels · Path query · XML stream

✉ Meghdad Mirabi
meghdad.mirabi@gmail.com

Samaneh Fathi Ozonbolagh
fathi_samane@yahoo.com

[1] Faculty of Mechatronic, Islamic Azad University, Karaj Branch, Alborz, Iran

[2] Department of Statistical Research and Information Technology, Institute for Research and Planning in Higher Education, Tehran, Iran

1 Introduction

Today, mobile wireless technology is in a deep tangle with everybody's daily life. At the end of 2011, there was 6 billion mobile subscriptions, which is equivalent to 87% of the whole population of the world [1]. In mobile wireless networks, data are sent and received on the air via radio waves and thus it allows mobile clients to receive their required data on the air. One of the efficient data dissemination ways in mobile wireless networks is data broadcasting, which is being highly regarded nowadays [2, 3]. In mobile wireless broadcast networks, a broadcast server sends data over a wireless broadcast channel at a broadcast cycle and mobile clients can connect to that channel in order to receive the data they need [4, 5].

In order to connect to the mobile wireless broadcast channels and receive the required data over these channels, mobile clients use portable devices such as laptops, tablets, and smart phones. The major problems of these portable devices are the limitation of battery energy, the limitation of memory capacity, and the limitation of processing resources. Therefore, it is necessary to devise solutions to speed up data access over the mobile wireless broadcast channels and reduce the energy consumption of portable devices for downloading the required data over these channels [6, 7].

Generally, each portable device has two working modes, namely doze-mode and active-mode. In the doze-mode, the portable device does not perform the task of processing or receiving data from the broadcast channel. On the other hand, in the active-mode, the portable device connects to the broadcast channel and receives the intended data. The energy consumption of the portable devices in the active-mode is higher than that in the doze-mode. Therefore, the new solutions should be designed in such a way that they allows the portable devices to remain in the doze-mode as long as their required data are not sent out over a broadcast channel and switch to the active-mode and receive the required data as soon as the required data are over the broadcast channel [8–10].

Actually, two important criteria to measure the access performance to the data in a mobile wireless broadcast channel are access time and tuning time [2, 3, 5, 9]. Access time refers to the period of time from the moment that the mobile client sends a query over a wireless broadcast channel to the moment that the query results are received by the mobile client. Therefore, it is used to measure the efficient access to data over a mobile wireless broadcast channel. Tuning time represents the sum of the intervals during which the mobile client remains in the active-mode in order to retrieve the desired data over a mobile wireless broadcast channel, thus it is used to estimate the energy consumption in portable devices.

In recent years, different XML data stream structures have been proposed for broadcasting the XML data over a wireless broadcast channel to reduce the access time and tuning time [11–19]. These XML data stream structures utilize the indexing techniques, which put the index information combined with the XML data, to disseminate the XML data over a wireless broadcast channel. Using the index information, the mobile clients can find out the time when the intended data are given on the broadcast channel. This allows the mobile clients to put their portable devices in the doze-mode and, as soon as the intended data appears on the broadcast channel, connect to the channel and receive the data. However, these XML data stream structures are designed to be used only in a single wireless broadcast channel. By assuming that there are multiple disjoint physical broadcast channels, it is possible to place the data over multiple wireless broadcast channels in a way that it reduces the access time of mobile clients in retrieving the required data over the broadcast channels. This assumption has wider applicability since there are many frequencies on which data can be transmitted in a wireless network. In addition, the use of multiple wireless broadcast channels allows better configurability, scalability, and fault tolerance [20–22].

In [23], an efficient XML data placement scheme is proposed to place the XML data over multiple wireless broadcast channels by grouping and partitioning the XML data. In the proposed XML data placement scheme, the XML data stream is divided into several partitions and each partition is placed over a broadcast channel. By dividing the XML data stream into several partitions, the length of the XML data stream in each broadcast channel is shortened. This causes a further reduction in the access time compared to the schemes that use only a single wireless broadcast channel [11–19].

The overall approach in [23] is to group the XML nodes with the same root-to-node path together using the path summary technique [24] and place these groups in multiple wireless broadcast channels. The XML nodes grouped together are placed in multiple broadcast channels based on the depths of root-to-node paths in ascending order. Also, the Pre/Post XML labeling scheme [25] is used in [23] in order to find the exact location of XML nodes in the XML document when the XML data is retrieved from the wireless broadcast channels. Now, assume that the XML document considered to be disseminated over multiple wireless broadcast channels is a file with large size and the number of XML nodes with the same root-to-node path with high depth is large. Therefore, the access time to retrieve the XML nodes with the same root-to-node path with high depth will still have a big value. In this paper, we propose two XML data placement schemes which overcomes the problem explained in [23] once the XML nodes are disseminated over multiple wireless broadcast channels. Hence, the main contributions of this paper are summarized as follows:

- We propose two XML data placement schemes in order to place the XML nodes over multiple wireless broadcast channels by partitioning the XML data stream into several partitions and placing them into multiple wireless broadcast channels. The proposed XML data placement schemes improve the performance of XML querying over multiple wireless broadcast channels in terms of access time compared to the XML data placement scheme proposed in [23].
- We devise an algorithm to generate an XML data stream based on our proposed XML data placement schemes. The generated XML data stream contains indexes to jump forward to the XML query results which are located in a wireless broadcast channel.

- We evaluate the performance of proposed XML data placement schemes over multiple wireless broadcast channels in processing of XML queries in terms of access time and tuning time by performing several experiments using different XML data sets.

The rest of the paper is organized as follows: Sect. 2 includes an explanation of XML data model and XML query language. It also includes a short description of the previous studies for broadcasting the XML data in the mobile wireless broadcast networks. In Sect. 3, the proposed XML data placement schemes are presented and explained by examples. In Sect. 4, the process of XML querying over our proposed XML data placement schemes is described. In Sect. 5, the performance of proposed XML data placement schemes in XML querying over multiple wireless broadcast channels are evaluated using different XML data sets. Finally, in Sect. 6, the paper is concluded with a conclusion and discussion of future works.

2 Research background and related works

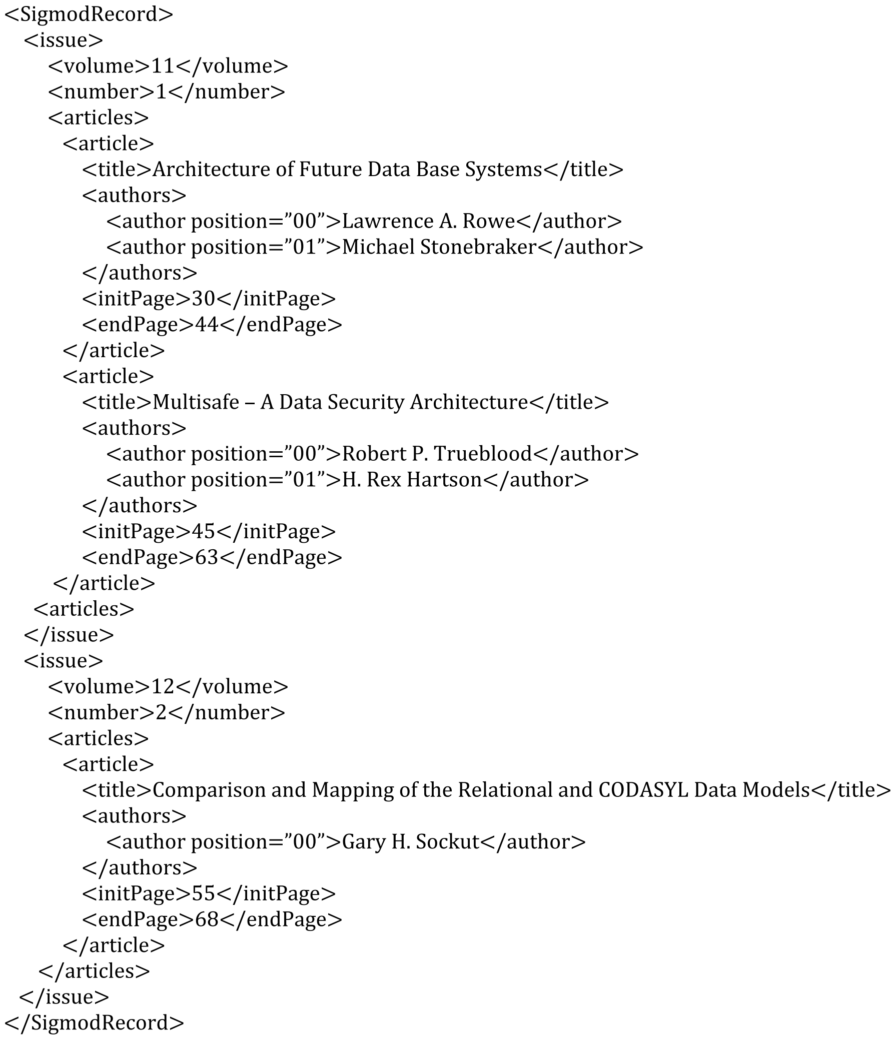
In this section, we first explain the XML data model [26] and the XPath query language [27] in short and then we explain different ways for broadcasting the XML data in mobile wireless broadcast networks.

2.1 XML data model and XPath query language

XML (eXtensible Markup Language) [26] is a markup language that has been introduced by the World Wide Web Consortium (W3C) in 1996. Due to the high capability and flexibility of this language for sending various kinds of data (ranging from different types of text data to audio/video data) on the Internet, XML is used as a defacto standard for presenting and exchanging the data on the internet.

An XML document is a set of elements formed a hierarchical structure. An XML element is a pair of start and end tags and the text data appearing between these two tags. Each XML document has a root element that includes all the elements of the XML document. Notice that the text might be combined with the sub-elements of an element. Also, XML language specifies another concept for the elements, which is called attribute. For example, in Fig. 1, "position" is an attribute of the element "author." An element's attribute is a string that cannot include the markups and appears in the form of "name = value" before the symbol ">" at the start tag. An XML document can be demonstrated by a tree structure in which the elements and the parent–child relationships are represented by the nodes and the edges, respectively. For example, Fig. 2 demonstrates the tree structure of the XML document shown in Fig. 1.

XPath is one of the popular XML query languages [27]. The result set of an XPath query is obtained based on the location path of the XML nodes in the XML tree. A location path includes the nodes' location path steps. The processing of each step of the location path of an XML node identifies a set of XML nodes in the XML



```
<SigmodRecord>
  <issue>
     <volume>11</volume>
     <number>1</number>
     <articles>
       <article>
         <title>Architecture of Future Data Base Systems</title>
         <authors>
            <author position="00">Lawrence A. Rowe</author>
            <author position="01">Michael Stonebraker</author>
         </authors>
         <initPage>30</initPage>
         <endPage>44</endPage>
       </article>
       <article>
         <title>Multisafe – A Data Security Architecture</title>
         <authors>
            <author position="00">Robert P. Trueblood</author>
            <author position="01">H. Rex Hartson</author>
         </authors>
         <initPage>45</initPage>
         <endPage>63</endPage>
      </article>
    <articles>
  </issue>
  <issue>
     <volume>12</volume>
     <number>2</number>
     <articles>
       <article>
         <title>Comparison and Mapping of the Relational and CODASYL Data Models</title>
         <authors>
            <author position="00">Gary H. Sockut</author>
         </authors>
         <initPage>55</initPage>
         <endPage>68</endPage>
       </article>
     </articles>
  </issue>
</SigmodRecord>
```

Fig. 1 An example of XML document

tree. In fact, some parts of the XML document are addressed by interpreting the path expressions of each XML node.

For example, the location path "/SigmodRecord/issue/volume" includes three steps "/SigmodRecord," "/issue," and "/volume," at each of which some XML nodes of the XML document are selected.

Some of the most commonly used symbols for selecting the XML nodes in an XML document in the XPath language are as follows:

- "/": It is used to specify the parent–child relationship between two XML nodes in the XML tree.

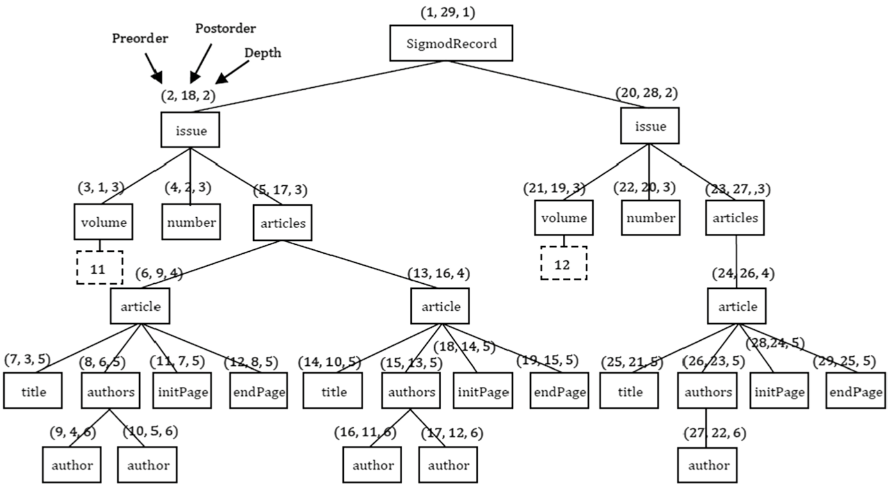


Fig. 2 An example of XML tree

- "//": It is used to specify the ancestor–descendant relationship between two XML nodes in the XML tree.
- "@": It represents an attribute of an XML node in the XML tree.
- "*": It represents any XML node with an arbitrary node name in the XML tree.
- "[]": It is used to specify a predicate condition in the XML query.

Generally, the XPath queries are divided into two groups:

- *Simple Path XML Query* This kind of XPath queries may include the symbols "/," "//," and "*." For example, the simple path XML query Q = "/SigmodRecord//articles/article" finds all of the XML nodes (i.e., elements) "article" that are the children of the node "articles," which is itself a descendant node of the node "SigmodRecord."
- *Twig Pattern XML Query* It is an XML query that may include the symbols "/," "//," "*," and "[]." For example, the twig pattern XML query Q = "/SigmodRecord/issue[volume/ text() = "12″]//author" finds all the nodes "author" that are the descendant node "issue," which is itself the child of the node "SigmodRecord," with the condition that the node "volume" is the child of the node "issue" and has a value of "12."

2.2 Related works

There are two broadcasting modes for data dissemination in terms of connection type and data access speed, namely push-based mode and on-demand mode [6–9]. In the push-based mode, the whole of data are disseminated over a wireless broadcast channel periodically and mobile clients can be connected to that channel in order to receive the required data. On the other hand, in the on-demand mode, mobile clients first send their requests to the broadcast server via an uplink channel and the broadcast server considers all of the requests in order to decide about the content and order of data sending for the next broadcast cycle. It should be noted that this paper proposes two new XML data placement schemes to disseminate the XML data over multiple wireless broadcast channels in the push-based mode.

Several studies have been done to reduce the access time by proposing an efficient scheduling method which determines the content and order of data in a wireless broadcast channel [28–32]. However, these scheduling methods have been proposed for XML data dissemination in the on-demand mode and they cannot be applied for our proposed XML data placement schemes.

Data broadcasting in the push-based mode has various applications in mobile wireless networks due to the unique features such as bandwidth efficiency (the broadcast channel is shared to large number of mobile clients), energy efficiency (connection to the broadcast channels does not require energy consumption for sending the requests to the broadcast server), and scalability (the number of mobile clients that can be connected to a broadcast channel is not limited) [2–9]. In the following, we provide a short review of the existing research studies for broadcastig XML data in the push-based mode.

In [11], a node structure called S-Node for XML nodes in the XML data stream is proposed which can contain three different indexing methods to process different types of simple path XML queries. These three indexing methods are One Sibling Address (OSA), Two Sibling Address (TSA), and Same Path Address (SPA). In the OSA indexing method, each S-Node contains the arrival time of the next sibling node in the XML data stream. This index is used to jump forward to the next sibling S-Node when the current S-Node has no child node while the next query result (i.e., next S-Node) exists or the current S-Node itself is not as a result of submitted simple path XML query. In the TSA indexing method, each S-Node contains two arrival times, the arrival time of nearest sibling node having the same tag name as the current S-Node and the arrival time of nearest sibling node having a tag name different from that of the current S-Node. By exploiting the TSA indexing method, the simple path XML queries can be processed more efficiently than the OSA indexing method. In the case that the current S-Node does not have any child while the next query result (i.e., Next S-Node) exists or the current S-Node itself is not the query answer, mobile client can jump forward to the nearest sibling node having different tag name from the current S-Node. Therefore, a large number of irrelevant S-Nodes is skipped in the XML data stream. In the case that the current S-Node is the query answer, mobile clients can jump forward to the nearest sibling node having the same tag name as the current S-Node. By following the sequence of same tag name addresses, mobile clients can selectively access to all S-Nodes which may be the results of submitted simple path XML query. In the SPA indexing method, each S-Node contains the arrival time of nearest sibling node having the same root-to-node path. By exploiting this indexing method, a chain of S-Nodes having the same root-to-node path is constructed throughout the XML data stream. Therefore, this indexing method can facilitate the process of searching the XML query results over the XML data stream.

An indexing method based on the path summary technique [24] has been proposed by [12]. By exploiting the path summary technique for indexing the XML nodes in the XML data stream, the XML nodes which have the same root-to-node path are grouped together and the redundant tag names are removed from the XML data stream. Therefore, this indexing method can reduce the access time in processing the different types of simple path XML queries over the XML data stream.

In [13], a node structure called DIX has been proposed to implement a fully distributed index for XML data broadcast over a wireless broadcast channel. The DIX node structure contains the field location path index (LPI). It allows mobile clients can start the process of simple path XML queries over the XML data stream without exploiting the XML data stream from the root node. In order to efficient process the simple path XML queries over a wireless broadcast channel, two indices called Clone node Link (CL) and Foreign node Link (FL) are proposed for each DIX node. The CL index is used to access the candidate nodes which may be the results of submitted simple path XML query and the FL index is used to skip from the XML nodes irrelevant to the submitted simple path XML query.

In [14], a novel structure for XML data stream called PS + Pre/Post has been proposed by integrating the path summary (PS) technique [24] and the Pre/Post labeling scheme [25]. By exploiting the path summary structure for accessing the XML nodes over the XML data stream, the redundancy of root-to-node paths of the set of XML nodes with the same root-to-node path grouped together can be removed and therefore, the access time to retrieve the XML nodes over the XML data stream can be reduced. However, the path summary technique does not keep the Parent–Child relationship and the Ancestor–Descendant relationship between the XML nodes over the XML data stream which are required to process twig pattern XML queries. By labeling the XML nodes with the Pre/Post labeling scheme, the structural relationships between the XML nodes can be easily determined and therefore, different twig pattern XML queries can be efficiently processed over the XML data stream.

In [15], a new indexing method has been proposed to process twig pattern XML queries over the XML data stream. The proposed indexing method uses two types of lineage encoding called lineage vertical (V) code and lineage horizontal (H) code. The lineage code which is represented by (V, H) is able to represent the parent–child relationships between XML nodes. Therefore, this indexing method is able to process different types of simple and twig pattern XML queries over the XML data stream in mobile wireless broadcast networks.

In [16], a new XML data distribution scheme is proposed for the XML indexing method in [14] by distributing the partial and relevant parts of both the indexes and the XML data into suitable positions over a wireless broadcast channel. In this distribution scheme, each broadcast cycle is divided into a set of quantums. Each quantum contains a set of paths and XML nodes. By exploiting this distribution scheme, mobile clients can process different types of simple and twig pattern XML queries without exploiting the XML stream at the beginning of the next broadcast cycle.

In [17], a novel XML stream structure is proposed by grouping and summarizing the structural information of XML nodes in the XML tree such as the root-to-node paths, the contents of texts and attributes of XML nodes in order to reduce the size of XML stream. The proposed XML stream structure contains several indexes in order to skip from the irrelevant parts of XML stream during the processing of XML queries over the XML stream. By reducing the size of XML stream and skipping from the irrelevant parts of the XML stream, the proposed XML stream structure is able to efficiently process different types of simple and twig pattern XML queries in terms of access time and tuning time.

In order to secure dissemination of XML data over a wireless broadcast channel, mobile clients should obey a set of access authorizations specified on the original XML document. In such environments, mobile clients can only access authorized parts of encrypted XML stream based on their access authorizations. In [18], the first XML stream structure called SecNode is proposed which supports two security services data confidentiality and access control of XML data over a wireless broadcast channel. Each SecNode contains the indices such as First Node Address, Last Node Address, and Next Candidate to selectively access authorized parts of XML data over the XML stream.

In [19], a new encrypted XML data stream structure is proposed which supports the confidentiality of XML data over the broadcast channel. In the proposed stream structure, the size of encrypted XML data stream is reduced compared to the XML data structure proposed by [18] by grouping the paths, XML nodes, texts, and attributes together. The proposed structure includes several indexes to skip from irrelevant data over the broadcast channel.

The problem of all previous indexing methods [11–19] is that the performance of XML querying in terms of access time degrades as the number of XML data being broadcast increases. It is due to the sequential access of XML data in a wireless broadcast channel. The increasing number of XML data causes mobile clients to wait for a substantial amount of time before receiving the desired XML data. In [23], a new XML data placement scheme to broadcast the XML data over multiple wireless broadcast channels is proposed by optimally partitioning the XML data among multiple wireless broadcast channels in order to reduce the average access time of mobile clients in processing of different types of simple and twig pattern XML queries.

There are several factors to be considered in order to compare the existing indexing methods for XML data broadcast which are listed as follows:

- *Number of broadcast channels used for XML data broadcast* Whether the proposed indexing method uses a single broadcast channel or multiple broadcast channels to disseminate the XML data.
- *Type of improvement* Whether the proposed indexing method can improve the performance of XML querying in terms of access time and tuning time or not.
- *Type of XML queries supported by the proposed indexing method* Whether the proposed indexing method is able to process different types of simple path and twig pattern XML queries or not.

Table 1 summarizes the existing indexing methods for XML data broadcast in the push-based mode.

Table 1 Comparison of different indexing methods for XML data broadcast in the push-based mode

Related works	Single channel/ multiple channels	Improvement		Processing of XML queries	
		Access time	Tuning time	Simple path query	Twig pattern query
[11]	Single channel	✓	✓	✓	–
[12]	Single channel	✓	✓	✓	–
[13]	Single channel	✓	✓	✓	–
[14]	Single channel	✓	✓	✓	✓
[15]	Single channel	✓	✓	✓	✓
[16]	Single channel	✓	–	✓	✓
[17]	Single channel	✓	✓	✓	✓
[18]	Single channel	✓	✓	✓	✓
[19]	Single channel	✓	✓	✓	✓
[23]	Multiple channels	✓	–	✓	✓

3 The proposed data placement schemes for XML data broadcast

In this section, we first explain the structure of XML data stream used in our proposed XML data placement schemes. This XML data stream structure is similar to the XML data stream structure in [23]. Then, we explain our proposed XML data placement schemes for placing the XML data over multiple wireless broadcast channels.

3.1 The structure of XML data stream

Generally, the structure of XML data stream used in our proposed XML data placement schemes is divided into two parts, namely "index segment" and "data segment." In our proposed XML data placement schemes, the first broadcast channel is always used to broadcast the part of "index segment" and the other broadcast channels are used to broadcast the part of "data segment." It means that the part of "index segment" is always placed on the first broadcast channel but the part of "data segment" can be partitioned and placed on more than one broadcast channel.

In the XML data stream structure, the part of "index segment" contains index information about the XML data disseminated over other wireless broadcast channels. Therefore, mobile clients can use this information to find out the channel number and the arrival time of XML data on which they are interested to retrieve over the wireless broadcast channels.

Definition 1 The root-to-node path of XML node *e* refers to a sequence of node names from the root node to the node *e*.

Table 2 An example of path summary

Ordered ID	Root-to-node path
1	/SigmodRecord
2	/SigmodRecord/issue
3	/SigmodRecord/issue/volume
4	/SigmodRecord/issue/number
5	/SigmodRecord/issue/articles
6	/SigmodRecord/issue/articles/article
7	/SigmodRecord/issue/articles/article/title
8	/SigmodRecord/issue/articles/article/authors
9	/SigmodRecord/issue/articles/article/initPage
10	/SigmodRecord/issue/articles/article/endPage
11	/SigmodRecord/issue/articles/article/authors/author

Table 3 Structure of each root-to-node path in the index segment

Field	Description
$\text{RootToNodePathLength}_i$	The length of root-to-node path *i*
RootToNodePath_i	The root-to-node path *i*
$\text{FirstChildAddress}_i$	The distance between the root-to-node paths Path_i and Path_j in the index segment. The root-to-node path Path_j is the first child of root-to-node path Path_i
FirstAddress_i	The arrival time of first XML node with the root-to-node path Path_i in the data segment
LastAddress_i	The arrival time of last XML node with the root-to-node path Path_i in the data segment
ChannelNumber_i	The channel number in which XML nodes with the root-to-node path Path_i are disseminated over it

For example, in Fig. 2, the root-to-node path of XML node “title” is “/SigmodRecord/issue/ articles/article/title.”

By grouping the set of XML nodes having the same root-to-node path, a set of unique root-to-node paths is constructed called path summary (PS).

Definition 2 The path summary of XML tree *T* (PS_T) is an ordered set of unique root-to-node paths in the XML tree *T* where the paths are ordered based on the breadth-first traverse of XML tree *T*.

For example, Table 2 shows the path summary of XML tree shown in Fig. 2. As shown in Table 2, the XML tree shown in Fig. 2 contains 11 unique root-to-node paths.

Definition 3 Assume that $\text{PS}_T = \{\text{Path}_1, \text{Path}_2, \ldots, \text{Path}_n\}$ is the path summary of XML tree *T*. Each root-to-node path of the XML tree *T* in the part of “index segment” contains several fields as shown in Table 3.

For example, Table 4 shows the unit structure of root-to-node path with the ordered ID 6 in the path summary shown in Table 2. The field length of root-to-node path with the ordered ID 6 is 36 since the length of this path is 36 characters. The field root-to-node path is "/SigmodRecord/issue /articles/article" since the 6th root-to-node path in the path summary is that path. The field First Child Address is *X* which is the arrival time of root-to-node path of the first child node of path "/SigmodRecord/issue/articles/article" (i.e., "/SigmodRecord/issue/articles/article/title"). The field First Address is *Y* which is the arrival time of first XML node with the root-to-node path "/SigmodRecord/issue/articles/article" over a wireless broadcast channel. The field Last Address is *Z* which is the arrival time of last XML node with the root-to-node path "/SigmodRecord/issue/ articles/article" over a wireless broadcast channel. The fields First Address and Last Address are used by mobile clients to determine when their mobile devices should be in the active mode to retrieve the desired XML data over a wireless broadcast channel. The field Channel Number is 2 which means the result of simple path XML query with the root-to-node path "/SigmodRecord/issue/articles/article" is disseminated over the broadcast channel with the number 2.

Definition 4 Assume that $\text{Nodes}_\text{T} = \{\text{Node}_1, \text{Node}_2, \ldots, \text{Node}_\text{m}\}$ is the set of XML nodes in the XML tree *T*. Each XML node in the XML tree *T* in the part of "data segment" has several fields as shown in Table 5.

For example, Table 6 shows the unit structure of node "author" with the preorder 9 in the XML tree shown in Fig. 2. The fields Preorder, Postorder, ParentPreorder, and Depth are 9, 4, 8, and 6, respectively, as shown in Fig. 2. The field TextFlag is 1 since node "author" with the preorder 9 has a text. The field TextLength is 16 since the text of node "author" with the preorder 9 has 16 characters. The field TextValue is "Lawrence A. Rows" since the value of text of node "author" has this value. The node "author" with the preorder 9 has an attribute with the name "position" and value "00." Therefore, the field AttributeNumber is 1. Also, the fields AttributeNameLength, AttributeName, AttributeValueLength, and AttributeValue are 8, "position," 2, and "00," respectively.

Table 4 Unit structure of the root-to-node path with the ordered ID 6 in the index segment

Field	Content
$\text{RootToNodePathLength}_\text{i}$	36
$\text{RootToNodePath}_\text{i}$	/SigmodRecord/issue/articles/article
$\text{FirstChildAddress}_\text{i}$	X
$\text{FirstAddress}_\text{i}$	Y
$\text{LastAddress}_\text{i}$	Z
$\text{ChannelNumber}_\text{i}$	2

Table 5 Structure of each XML node in the data segment

Field	Description
$\text{Preorder}_{\text{i}}$	The preorder of XML node *i* in the XML tree *T* when the XML tree *T* is traversed in preorder sequence
$\text{Postorder}_{\text{i}}$	The postorder of XML node *i* in the XML tree *T* when the XML tree *T* is traversed in postorder sequence
$\text{ParentPreorder}_{\text{i}}$	The parent's preorder of XML node *i* in the XML tree *T* when the XML tree *T* is traversed in preorder sequence
Depth_{i}	The depth of XML node *i* in the XML tree *T*
$\text{TextFlag}_{\text{i}}$	It indicates whether the XML node *i* contains a text or not
If the XML node *i* contains a text, the XML node *i* contains the following two fields:	
$\text{TextLength}_{\text{i}}$	The length of text of XML node *i* in the XML tree *T*
$\text{TextValue}_{\text{i}}$	The text value of XML node *i* in the XML tree *T*
$\text{AttributeNumber}_{\text{i}}$	The total number of attributes of XML node *i* in the XML tree *T*
If the XML node *i* contains one or more attributes, the XML node *i* contains the following four fields for each attribute:	
$\text{AttributeNameLength}_{\text{ij}}$	The length of *j*th attribute's name of XML node *i* in the XML tree *T*
$\text{AttributeName}_{\text{ij}}$	The name of *j*th attribute of XML node *i* in the XML tree *T*
$\text{AttributeValueLength}_{\text{ij}}$	The length of *j*th attribute's value of XML node *i* in the XML tree *T*
$\text{Attribute Value}_{\text{ij}}$	The value of *j*th attribute of XML node *i* in the XML tree *T*

Table 6 Unit structure of node “author” with the preorder 9 in the data segment

Field	Content
Preorder	9
Postorder	4
ParentPreorder	8
Depth	6
TextFlag	1
TextLength	16
TextValue	Lawrence A. Rowe
AttributeNumber	1
AttributeNameLength	8
AttributeName	position
AttributeValueLength	2
AttributeValue	00

3.2 XML data placement schemes

In our proposed XML data placement schemes, XML data must be placed over multiple wireless broadcast channels in such a way that the workloads of broadcast channels to be balanced. The following assumptions are used in order to place the XML data over multiple wireless broadcast channels in our proposed XML data placement schemes.

Assumption 1 The minimum number of broadcast channels to disseminate the XML data is 2. The first broadcast channel is always used to broadcast the part of “index segment” and the second broadcast channel is used to broadcast the part of “data segment” of the XML data stream.

Assumption 2 The maximum number of broadcast channels to disseminate the XML data is equal to “the number of unique root-to-node paths in the XML tree T” + 1.

Assumption 3 All of the XML nodes related to a root-to-node path are disseminated over a single broadcast channel and they cannot be partitioned and placed on different broadcast channels.

In the following, our proposed XML data placement schemes to disseminate the XML data over multiple wireless broadcast channels are explained.

The process of XML data placement over multiple wireless broadcast channels in the first proposed XML data placement scheme is shown in Fig. 3. It should be mentioned that the part of “index segment” is disseminated over the first broadcast channel and this process is not explained in Fig. 3.

As shown in Fig. 3, the total size of the set of XML nodes of each root-to-node path in the XML tree T is calculated based on Definition 4. Then, based on the

```
Input:
    n : The total number of root-to-node paths of the XML tree $T$;
    m: The total number of broadcast channels to disseminate the XML data;
Output:
    The set of broadcast channels which includes all of the XML nodes of the XML tree $T$;

1.  The total size of the set of XML nodes of each root-to-node path in the XML tree $T$ is calculated based on
    Definition 4. Based on the total size of XML nodes in each root-to-node path, the set of root-to-node paths in
    the XML tree $T$ is ordered in descending order. Assume that the result of ordering is as follows: $SP_T$={$Path_1$,
    $Path_2$,..., $Path_n$}
2.  channelNumber = 1;
3.  currentPath = 1;
4.  while (true) {
5.      while (channelNumber != m){
6.          If (currentPath == n + 1) break;
7.          channelNumber ++;
8.          Place all of the XML nodes of root-to-node path $Path_{currentPath}$ on the broadcast channel
            $Channel_{channelNumber}$;
9.          currentPath ++;
10.     }
11.     while (channelNumber != 1){
12.         If (currentPath == n + 1) break;
13.         Place all of the XML nodes of root-to-node path $Path_{currentPath}$ on the broadcast channel
            $Channel_{channelNumber}$;
14.         channelNumber --;
15.         currentPath ++;
16.     }
17.     If (currentPath == n + 1) break;
18. }
```

Fig. 3 The first XML data placement scheme

total size of XML nodes in each root-to-node path, the set of root-to-node paths in the XML tree *T* is ordered in descending order (Line 1). The reason is that the set of XML nodes with the same root-to-node path which has large size can be disseminated over the broadcast channels with a higher priority at the first (Lines 5–10). Then, in order to keep the load of broadcast channels balanced, the broadcast channels which have small amounts of XML data are used to disseminate the rest of XML data in the part of "data segment" (Lines 11–16). By changing the priority of broadcast channels to disseminate the XML data, the broadcast channels that have disseminated the XML data with large size can disseminate the XML data with the small size in order to keep the load balancing between all of the wireless broadcast channels.

Example 1 Assume that the XML data of XML tree *T* with 5 unique root-to-node paths must be disseminated over 3 wireless broadcast channels based on the first XML data placement scheme shown in Fig. 3. Table 7 shows the root-to-node paths

Table 7 General information of XML tree *T*

Root-to-Node Paths	/a	/a/b	/a/b/c	/a/b/d	/a/b/e
Total Size of the XML Nodes Based on Definition 4 (in Bytes)	19	46	74	27	27

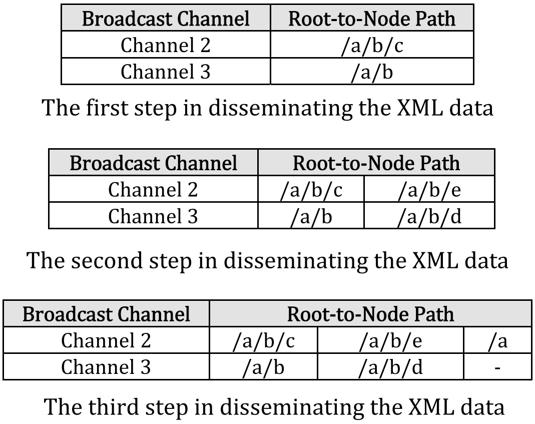
for the XML tree *T* among with the total size of the XML nodes related to each root-to-node path.

Based on the first XML data placement scheme in Fig. 3, the root-to-node paths of XML tree *T* are ordered in descending order based on the total size of their XML nodes. Therefore, the result is the set of root-to-node paths as follows: {"/a/b/c," "/a/b," "/a/b/d," "/a/b/e," "/a"}. The process of XML data placement for this set of root-to-node paths is shown in Fig. 4.

It should be mentioned that the first broadcast channel (i.e., Channel 1) is used to disseminate the index information. Therefore, there will remain two broadcast channels to disseminate the XML data in the part of "data segment." According to the first XML data placement scheme shown in Fig. 3, at the first, the XML nodes with the root-to-node paths "/a/b/c" and "/a/b" are placed on the broadcast channels 2 and 3, respectively, since the total sizes of the XML nodes in these two root-to-node paths are larger than other root-to-node paths. Then, the XML nodes with the root-to-node paths"/a/b/d" and "/a/b/e" are placed on the broadcast channels 3 and 2, respectively, in order to keep the load of broadcast channels balanced. Then, the XML nodes with the root-to-node path "/a" are placed on the broadcast channel 2 based on the first XML data placement scheme shown in Fig. 3.

The process of XML data placement over multiple wireless broadcast channels in the second proposed XML data placement scheme is shown in Fig. 5. It should be mentioned that the part of "index segment" is disseminated over the first broadcast channel and this process is not explained in Fig. 5.

As shown in Fig. 5, similar to the first XML data placement scheme, at the first, the set of root-to-node paths is sorted in descending order based on the size of XML nodes in Definition 4. Now, if the result of division (the total number of root-to-node paths / (the total number of broadcast channels—1)) is a number less than 2, the first XML data placement scheme shown in Fig. 3 is used to disseminate the XML data over multiple wireless broadcast channels (Lines 2–4). In the case that the result of division [the total number of root-to-node paths / (the total number of broadcast channels—1)] is an even integer number, the variable "g" (i.e., [(the total number



Broadcast Channel	Root-to-Node Path
Channel 2	/a/b/c
Channel 3	/a/b

The first step in disseminating the XML data

Broadcast Channel	Root-to-Node Path	
Channel 2	/a/b/c	/a/b/e
Channel 3	/a/b	/a/b/d

The second step in disseminating the XML data

Broadcast Channel	Root-to-Node Path		
Channel 2	/a/b/c	/a/b/e	/a
Channel 3	/a/b	/a/b/d	-

The third step in disseminating the XML data

Fig. 4 An example of XML data placement based on the first XML data placement scheme when the total number of broadcast channels is 3

```
Input:
        n : The total number of root-to-node paths of the XML tree $T$;
        m: The total number of broadcast channels to disseminate the XML data;
Output:
        The set of broadcast channels which includes all of the XML nodes of the XML tree $T$;

1.   The total size of the set of XML nodes of each root-to-node path in the XML tree $T$ is calculated based on
     Definition 4. Based on the total size of XML nodes in each root-to-node path, the set of root-to-node paths
     in the XML tree $T$ is ordered in descending order. Assume that the result of ordering is as follows:
     $SP_T$={Path$_1$, Path$_2$,..., Path$_n$}
2.   If ((n / (m - 1)) < 2){
3.       The first XML data placement scheme shown in Fig 3 is used to disseminate the XML data;
4.   } else{
5.       if ([n / (m - 1)] is an even integer number){
6.           //"g" indicates how the XML nodes with different root-to-node paths should be disseminated
             over broadcast channels.
7.           g = [(n / (m - 1)) / 2];
8.           channelNumber = 1;
9.           while (channelNumber != m){
10.              channelNumber ++;
11.              Select "g" root-to-node paths from the first of the set of ordered root-to-node paths $SP_T$
                 and "g" root-to-node paths from the last of the set of ordered root-to-node paths $SP_T$
                 and then disseminate the XML nodes of these root-to-node paths over the broadcast
                 channel Channel$_{channelNumber}$;
12.          }
13.      } else if ([n / (m - 1)] is an odd integer number){
14.          //"g" indicates how the XML nodes with different root-to-node paths should be disseminated
             over broadcast channels.
15.          g = [((n / (m - 1)) – 1) / 2];
16.          channelNumber = 1;
17.          while (channelNumber != m){
18.              channelNumber ++;
19.              Select "g" root-to-node paths from the first of the set of ordered root-to-node paths $SP_T$
                 and "g" + 1 root-to-node paths from the last of the set of ordered root-to-node Paths
                 $SP_T$ and then disseminate the XML nodes of these root-to-node paths over the broadcast
                 channel Channel$_{channelNumber}$;
20.          }
21.      }
22.      If((n / (m - 1)) is not an interger number){
23.              Select the broadcast channel which the total size of XML data over it is samller than other
                 broadcast channels and then the reminder XML nodes that have not been disseminated
                 over a broadcast channel are disseminated over this broadcast channel;
24.      }
25.  }
```

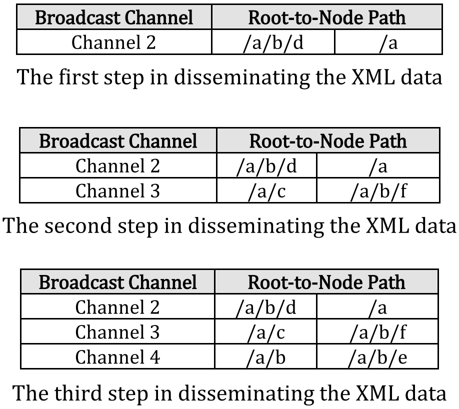
Fig. 5 The second XML data placement scheme

of root-to-node paths/(the total number of broadcast channels—1))/2]) is calculated in order to determine how the XML nodes with different root-to-node paths should be disseminated over multiple broadcast channels (Lines 5–12). In this case, "g" root-to-node paths from the first of the set of ordered root-to-node paths and "g" root-to-node paths from the last of the set of ordered root-to-node paths are selected and disseminated over the second broadcast channel. This process is continued until all of the XML nodes are disseminated over broadcast channels. In the case that the result of division [the total number of root-to-node paths/(the total number of broadcast channels—1)] is an odd integer number, the variable "g" (i.e., [((the total number of root-to-node paths/(the total number of broadcast channels—1))—1)/2]) is calculated in order to determine how the XML nodes with different root-to-node paths should be disseminated over multiple broadcast channels (Lines 13–21). In

Table 8 General Information of XML Tree *T*

Root-to-Node Paths	/a	/a/b	/a/c	/a/b/d	/a/b/e	/a/b/f
Total Size of the XML Nodes Based on Definition 4 (in Bytes)	19	46	52	74	27	27

Fig. 6 An example of XML data placement based on the second XML data placement scheme when the total number of broadcast channels is 4



Broadcast Channel	Root-to-Node Path	
Channel 2	/a/b/d	/a

The first step in disseminating the XML data

Broadcast Channel	Root-to-Node Path	
Channel 2	/a/b/d	/a
Channel 3	/a/c	/a/b/f

The second step in disseminating the XML data

Broadcast Channel	Root-to-Node Path	
Channel 2	/a/b/d	/a
Channel 3	/a/c	/a/b/f
Channel 4	/a/b	/a/b/e

The third step in disseminating the XML data

this case, "g" root-to-node paths from the first of the set of ordered root-to-node paths and "g" + 1 root-to-node paths from the last of the set of ordered root-to-node paths are selected and disseminated over the second broadcast channel. This process is continued until all of the XML nodes are disseminated over broadcast channels. In the case that the result of division (the total number of root-to-node paths / (the total number of broadcast channels—1)) is not an integer number, the broadcast channel which has small amounts of XML data is used to disseminate the rest of XML data in the part of "data segment" in order to keep the load of broadcast channels balanced (Lines 22–24).

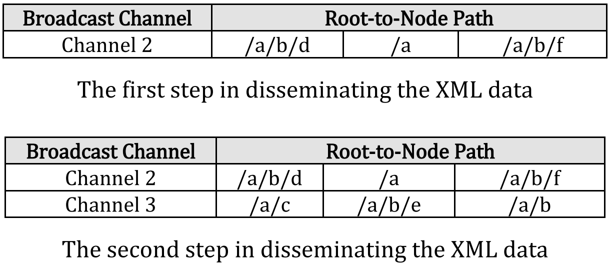
Example 2 Assume that the XML data of XML tree *T* with 6 unique root-to-node paths must be disseminated over 4 wireless broadcast channels based on the second XML data placement scheme shown in Fig. 5. Table 8 shows the root-to-node paths for the XML tree *T* among with the total size of the XML nodes related to each root-to-node path.

Based on the second XML data placement scheme in Fig. 5, the root-to-node paths of XML tree *T* are ordered in descending order based on the total size of XML nodes. Therefore, the result is the set of root-to-node paths as follows: {"/a/b/d," "/a/c," "/a/b," "/a/b/e," "/a/b/f," "/a"}. The process of XML data placement for this set of root-to-node paths is shown in Fig. 6. It should be mentioned that the first broadcast channel (i.e., Channel 1) is used to disseminate the index information. Therefore, there will remain three broadcast channels to disseminate the XML data. Since the result of division [the total number of root-to-node paths / (the total number of broadcast channels—1)] is 2 (i.e., [6/(4–1)] = 2) and the value of "g" is equal to 1 (i.e., [(6/(4–1))/2] = 1), one root-to-node path from the first of the set of ordered root-to-node paths (i.e., "/a/b/d") and one root-to-node path from the last of the set of ordered root-to-node paths (i.e., "/a") are selected and the XML nodes of these two root-to-node paths are disseminated over the second broadcast channel. Again, one root-to-node path from the first of the set of ordered root-to-node paths (i.e., "/a/c") and one root-to-node path from the last of the set of ordered root-to-node paths (i.e., "/a/b/f") are selected and the XML nodes of these two root-to-node paths are disseminated over the third broadcast channel. Again, one root-to-node path from the first of the set of ordered root-to-node paths (i.e., "/a/b") and one root-to-node path from the last of the set of ordered root-to-node paths (i.e., "/a/b/e") are selected and the XML nodes of these two root-to-node paths are disseminated over the forth broadcast channel.

Example 3 Assume that the XML data of XML tree *T* with 6 unique root-to-node paths must be disseminated over 3 wireless broadcast channels based on the second XML data placement scheme shown in Fig. 5. Table 8 shows the root-to-node paths for the XML tree *T* among with the total size of the XML nodes related to each root-to-node path. Based on the second XML data placement scheme in Fig. 5, the root-to-node paths of the XML tree *T* is ordered in descending order based on the total size of XML nodes. Therefore, the result is the set of root-to-node paths as follows: {"/a/b/d," "/a/c," "/a/b," "/a/b/e," "/a/b/f," "/a"}.

The process of XML data placement for this set of root-to-node paths is shown in Fig. 7. It should be mentioned that the first broadcast channel (i.e., Channel 1) is used to disseminate the index information. Therefore, there will remain two broadcast channels to disseminate the XML data.

Since the result of division [the total number of root-to-node paths / (the total number of broadcast channels—1)] is 3 (i.e., [6/(3–1)] = 3) and the value of "g" is equal to 1 (i.e., [((6/(3–1))–1)/2] = 1), one root-to-node path from the first of the set of ordered root-to-node paths (i.e., "/a/b/d") and two root-to-node paths from the last of the set of ordered root-to-node paths (i.e., "/a" and "/a/b/f") are selected and the XML nodes of these three root-to-node paths are disseminated over the second broadcast channel. Again, one root-to-node path from the first of the set of ordered



Broadcast Channel	Root-to-Node Path		
Channel 2	/a/b/d	/a	/a/b/f

The first step in disseminating the XML data

Broadcast Channel	Root-to-Node Path		
Channel 2	/a/b/d	/a	/a/b/f
Channel 3	/a/c	/a/b/e	/a/b

The second step in disseminating the XML data

Fig. 7 An example of XML data placement based on the second XML data placement scheme when the total number of broadcast channels is 3

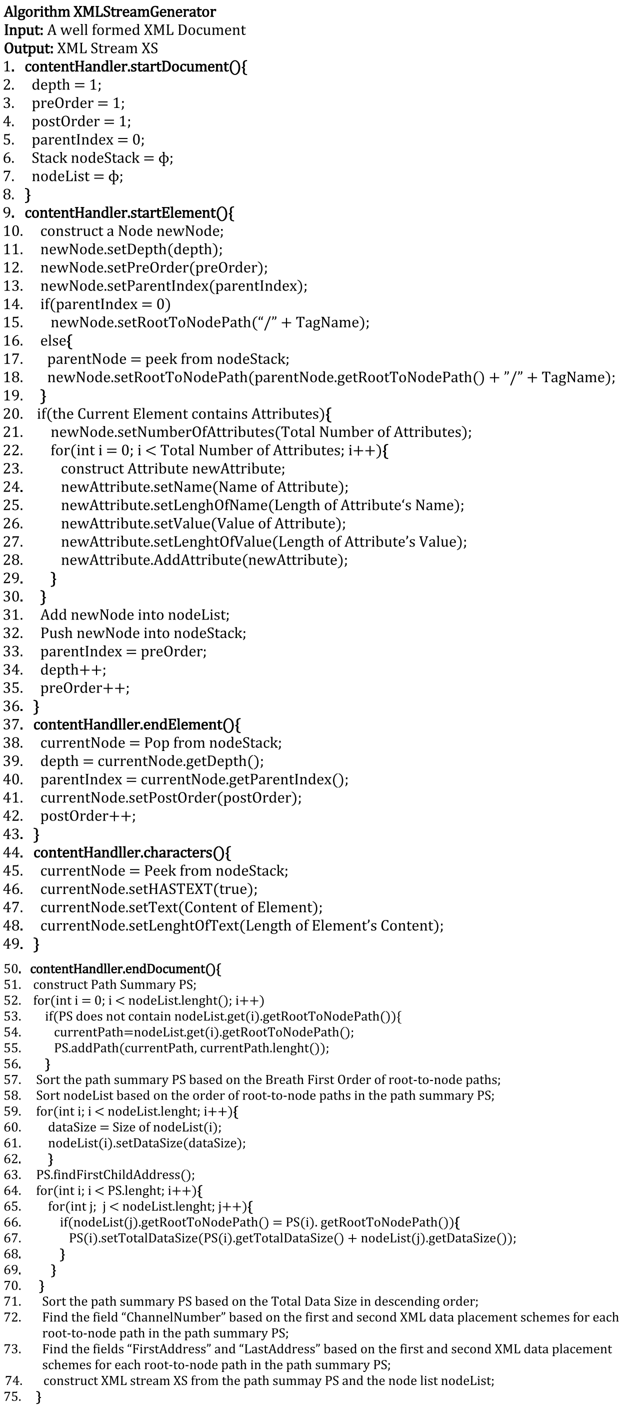
root-to-node paths (i.e., "/a/c") and two root-to-node paths from the last of the set of ordered root-to-node paths (i.e., "/a/b/e" and "/a/b") are selected and the XML nodes of these three root-to-node paths are disseminated over the third broadcast channel.

Figure 8 shows the XMLStreamGenerator algorithm which is devised to generate an XML stream based on our proposed XML data placement schemes. We use the SAX API which is an event-driven API for parsing the XML documents.

In this algorithm, the XMLStreamGenerator first initializes some of the relevant parameters using the startDocument() event handler (Lines 1–8). When a start XML tag is encountered during the parsing of a well-formed XML document, the XMLStreamGenerator creates a new node namely newNode (Line 10) and sets its related parameters (Lines 11–30) by executing the startElement() event handler. It also adds the newNode into the list of nodes (nodeList) (Line 31). Since some parameters of the newNode have not yet set, the XMLStreamGenerator pushes the newNode into a stack (nodeStack) (Line 32). If the current node contains a text content, the XMLStreamGenerator sets its related parameters by executing the characters() event handler (Lines 44–49). When an end XML tag is encountered during the parsing of a well-formed XML document, the XMLStreamGenerator executes the endElement() event handler to set some of the relevant parameters for the newNode such as the postorder (Lines 37–43). When the whole of XML document is parsed, the XMLStreamGenerator executes the endDocument() event handler and constructs the path summary PS for the parsed XML document by exploiting the root-to-node paths of the XML nodes in the XML document (Lines 50–56). It also sorts the root-to-node paths in the path summary PS based on the breadth-first order (Line 57). Then, the XMLStreamGenerator sorts nodes in the nodeList based on the order of root-to-node paths in the path summary PS (Line 58). Afterward, the XMLStreamGenerator obtains the size of each XML node (Lines 59–62). Then, the XMLStreamGenerator finds the address of first child path for each root-to-node path in the path summary PS (Line 63). It also calculates the sum of sizes for the nodes related to each root-to-node path as well as the field totalDataSize for each root-to-node path (Lines 64–70). Next, the XMLStreamGenerator sorts the root-to-node paths in descending order based on their total sizes (line 71). Then, the XMLStreamGenerator sets the fields ChannelNumber, FirstAddress, and LastAddress for each root-to-node path in the path summary PS based on the XML data placement schemes proposed in this paper (Lines 72–73). Finally, the XMLStreamGenerator generates the XML data stream XS (Line 74).

4 XML query processing based on the proposed XML data placement schemes

The process of XML querying in our proposed XML data placement schemes is similar to that in [23]. The algorithm of processing the simple path XML queries over multiple wireless broadcast channels in [23] is shown in Fig. 9. As shown in Fig. 9, when a mobile client sends a query Q over a wireless broadcast network, the mobile client is first connected to the broadcast channel on which the index segment has been sent (i.e., the broadcast channel with the number 1), and



```
Algorithm XMLStreamGenerator
Input: A well formed XML Document
Output: XML Stream XS
contentHandler.startDocument(){
  depth = 1;
  preOrder = 1;
  postOrder = 1;
  parentIndex = 0;
  Stack nodeStack = ϕ;
  nodeList = ϕ;
}
contentHandler.startElement(){
  construct a Node newNode;
  newNode.setDepth(depth);
  newNode.setPreOrder(preOrder);
  newNode.setParentIndex(parentIndex);
  if(parentIndex = 0)
    newNode.setRootToNodePath("/" + TagName);
  else{
    parentNode = peek from nodeStack;
    newNode.setRootToNodePath(parentNode.getRootToNodePath() + "/" + TagName);
  }
  if(the Current Element contains Attributes){
    newNode.setNumberOfAttributes(Total Number of Attributes);
    for(int i = 0; i < Total Number of Attributes; i++){
      construct Attribute newAttribute;
      newAttribute.setName(Name of Attribute);
      newAttribute.setLenghOfName(Length of Attribute's Name);
      newAttribute.setValue(Value of Attribute);
      newAttribute.setLenghtOfValue(Length of Attribute's Value);
      newAttribute.AddAttribute(newAttribute);
    }
  }
  Add newNode into nodeList;
  Push newNode into nodeStack;
  parentIndex = preOrder;
  depth++;
  preOrder++;
}
contentHandller.endElement(){
  currentNode = Pop from nodeStack;
  depth = currentNode.getDepth();
  parentIndex = currentNode.getParentIndex();
  currentNode.setPostOrder(postOrder);
  postOrder++;
}
contentHandller.characters(){
  currentNode = Peek from nodeStack;
  currentNode.setHASTEXT(true);
  currentNode.setText(Content of Element);
  currentNode.setLenghtOfText(Length of Element's Content);
}
contentHandller.endDocument(){
  construct Path Summary PS;
  for(int i = 0; i < nodeList.lenght(); i++)
    if(PS does not contain nodeList.get(i).getRootToNodePath()){
      currentPath=nodeList.get(i).getRootToNodePath();
      PS.addPath(currentPath, currentPath.lenght());
    }
  Sort the path summary PS based on the Breath First Order of root-to-node paths;
  Sort nodeList based on the order of root-to-node paths in the path summary PS;
  for(int i; i < nodeList.lenght; i++){
    dataSize = Size of nodeList(i);
    nodeList(i).setDataSize(dataSize);
  }
  PS.findFirstChildAddress();
  for(int i; i < PS.lenght; i++){
    for(int j;  j < nodeList.lenght; j++){
      if(nodeList(j).getRootToNodePath() = PS(i). getRootToNodePath()){
        PS(i).setTotalDataSize(PS(i).getTotalDataSize() + nodeList(j).getDataSize());
      }
    }
  }
  Sort the path summary PS based on the Total Data Size in descending order;
  Find the field "ChannelNumber" based on the first and second XML data placement schemes for each
  root-to-node path in the path summary PS;
  Find the fields "FirstAddress" and "LastAddress" based on the first and second XML data placement
  schemes for each root-to-node path in the path summary PS;
  construct XML stream XS from the path summay PS and the node list nodeList;
}
```

Fig. 8 XMLStreamGenerator algorithm

```
Algorithm SimplePathQueryProcessor
Input:Path Stream over the broadcast channel with the channel number 1, Data Stream over other broadcast channels (i.e
channel numbers 2, ..., n), Simple Path XML Query Q
Output: XML Query Results Set R
1.   R = ϕ;
2.   While(true){
3.     rootToNodePath = ””;
4.     While(rootToNodePath != XML Query Q){
5.       currentPath = Retrieve the current path from the broadcast channel Channel$_1$;
6.       rootToNodePath = The field RootToNodePath of the currentPath;
7.     }
8.     firstTime = The field FirstAdderess of the currentPath;
9.     lastTime = The field LastAddress of the currentPath;
10.    channelNumber= the field ChannelNumber of the currentPath;
11.    Based on the field ChannelNumber of the currentPath, Connect to that channel and wait in the doze mode until
       the node with the arrival time equal to the variable firstTime arrives on the air;
12.    While(the current time is not equal to the variable lastTime){
13.      R = R U {Retrieve the current node from the broadcast channel Channel$_{channelNumber}$};
14.    }
15.    break;
16.  }
```

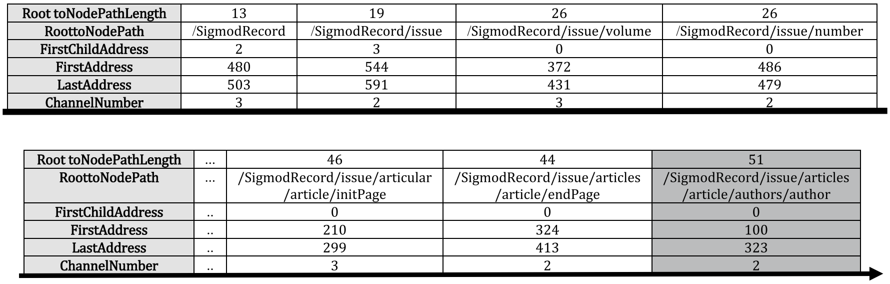
Fig. 9 Algorithm of simple path XML querying over multiple wireless broadcast channels

the root-to-node paths are read one by one until a root-to-node path equal to the simple path XML query Q is found (i.e., the path currentPath) (Lines 2–7). The mobile client can find the fields FirstAddress, LastAddress, and channelNumber from the path currentPath in order to determine the arrival times of query results over the broadcast channel on which the XML data has been sent (Lines 8–10). By knowing this information, the mobile client is connected to the intended channel and switches to the doze-mode and as soon as the intended XML data is appeared over the broadcast channel, the mobile client again switches to the active-mode and receives the XML query results (Lines 11–14).

Example 4 Assume that there are 3 wireless broadcast channels to disseminate the XML data of the XML tree shown in Fig. 2.

The first broadcast channel is used to disseminate the content of index segment and other two broadcast channels are used to disseminate the content of data segment. Now, assume that the mobile client *x* sends the simple path XML query Q = ”/SigmodRecord/issue /articles/article/authors/author” on the air. The mobile client *x* first connects to the broadcast channel on which the content of index segment is sent (i.e., the first broadcast channel). The content of index segment based on its structure defined in Definition 3 is shown in Fig. 10. The mobile client *x* finds out that the results set of submitted query *Q* have been sent on the second broadcast channel with a time distance ranging between 100 and 324. Therefore, the mobile client connects to the second broadcast channel and remains in the active-mode during this time interval and to receive the results set of simple path XML query *Q*.

The processes of other types of simple path XML queries and twig pattern XML queries in our proposed XML data placement schemes are similar to these in [23]. Refer to [23] for more information.



Root toNodePathLength	13	19	26	26
RoottoNodePath	/SigmodRecord	/SigmodRecord/issue	/SigmodRecord/issue/volume	/SigmodRecord/issue/number
FirstChildAddress	2	3	0	0
FirstAddress	480	544	372	486
LastAddress	503	591	431	479
ChannelNumber	3	2	3	2

Root toNodePathLength	...	46	44	51
RoottoNodePath	...	/SigmodRecord/issue/articular /article/initPage	/SigmodRecord/issue/articles /article/endPage	/SigmodRecord/issue/articles /article/authors/author
FirstChildAddress	..	0	0	0
FirstAddress	..	210	324	100
LastAddress	..	299	413	323
ChannelNumber	..	3	2	2

Fig. 10 The content of index segment over the first broadcast channel for the XML tree shown in Fig. 2

5 Performance evaluation

In this section, the performance of our proposed XML data placement schemes to process the simple path XML queries is evaluated by performing several experiments.

All of the experiments were conducted on a computer system with Core™i5 processor, 4 GB memory running on Windows 7 operating system where all of codes were implemented by JAVA.

5.1 Experimental settings

We logically modeled the XML data stream on multiple wireless broadcast channels as binary files where the broadcast server writes byte streams on the files and mobile clients read the files and process the XML queries [23].

In our simulation model, we assumed that all the broadcast channels have the same bandwidth and the broadcast bandwidth was fully utilized for broadcasting the XML data [11–19].

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 of XML query processing at the other mobile clients [11–19].

We assumed that the XML stream is broadcasted and accessed in units with a fixed size and thus we measured the access time and tuning time in processing different types of XML queries by the number of bytes. In the view of assumption that the network speed is fixed, the number of bytes can be converted into time since the elapsed time for reading a unit with a fixed size is computed as the unit size divided by the network speed [11–19].

To measure the performance variation based on the types of XML data sets, we used the Mondial and XMark data sets. Table 9 shows the characteristics of these two XML data sets. These XML data sets were used to perform the different experiments in the previous research studies [11–19].

To measure the performance variation based on the types of XML queries, we used different simple path XML queries. The list of XML queries used in

Table 9 The characteristics of XML Data Sets

Data Set	Size (KB)	Number of elements	Number of attributes	Number of root-to-node paths	Maximum fan out
Mondial	1743	22,423	47,423	33	955
XMark	1155	17,132	3919	421	255

Table 10 XML query sets

Data Set	Query Name	Path Order	Data Order	Simple Path XML Query
Mondial	M1	29	2	/mondial/country/province/city
	M2	31	4	/mondial/country/province/city/population
	M3	11	7	/mondial/country/city
	M4	12	8	/mondial/country/ethnicgroups
	M5	3	11	/mondial/organization
	M6	22	21	/mondial/river/to
XMark	X1	272	7	/site/closed_auctions/closed_auction/annotation/description/parlist/listitem/text
	X2	148	8	/site/open_auctions/open_auction/annotation/description/text
	X3	185	10	/site/regions/namerica/item/mailbox/mail/text
	X4	309	69	/site/regions/namerica/item/description/parlist/listitem/text/emph
	X5	172	252	/site/regions/australia/item/description/parlist/listitem
	X6	411	377	/site/regions/africa/item/incategory

our experiments is shown in Table 10. We selected the simple path XML queries which have different query results in terms of the total size of XML node results. It should be mentioned that the filed "Path Order" in Table 10 indicates the order number of root-to-node paths in descending order for each XML data set and the field "Data Order" indicates the order number of root-to-node paths based on the total size of their XML nodes in descending order for each XML data set. Since the results of the experiments for other simple path XML queries and twig pattern XML queries are the same as the simple path XML queries shown in Table 10, we have ignored to explain this part of experiments.

The performance metrics used in our experiments were the volume of XML data stream, the access time, and the tuning time. The volume of XML data stream is the size of XML data stream in bytes based on the structure of XML data placement scheme used for generating an XML data stream. The access time is the total number of bytes to read from the binary file (i.e., the XML data stream) from the moment an XML query is submitted on the air to the moment the query results are retrieved from the wireless broadcast channel. The tuning time is the total number of bytes to read in binary file (i.e., the XML data stream)

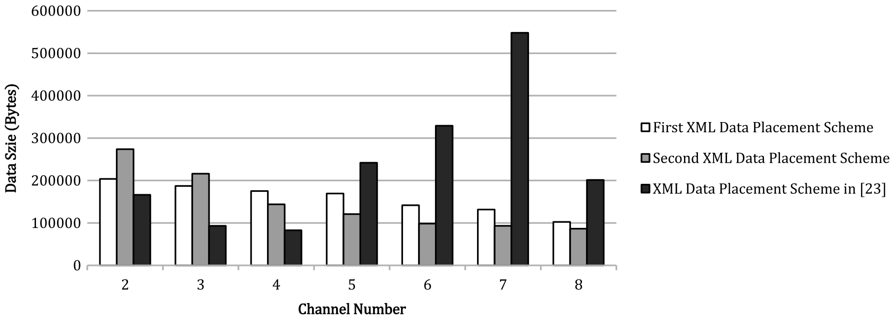


Fig. 11 The volume of XML data in the data segment over multiple wireless broadcast channels on the Mondial Data Set

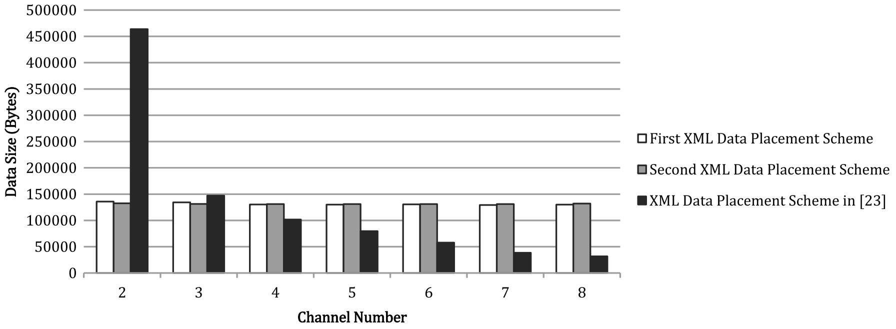


Fig. 12 The volume of XML data in the data segment over multiple wireless broadcast channels on the XMark data set

when the mobile device is in the active mode [11–19]. It should be noted that a mobile device is in the active mode when it listens to a broadcast channel, reads data over the channel and then downloads them. It is in the doze mode when it is waiting for the desired data to be disseminated over the broadcast channel.

5.2 Experimental results on volume of disseminated XML data

Figures 11 and 12 show the volume of XML data disseminated over different broadcast channels in the Mondial and XMark data sets when the total number of wireless broadcast channels is 8, respectively.

In our experiments, the first channel was used to disseminate the part of index segment and it is not shown in Figs. 11 and 12 since the volume of data in this channel for all the XML data placement schemes (i.e., The first and second proposed XML data placement schemes and the XML data placement scheme in [23]) is the same. As shown in Figs. 11 and 12, the volume of XML data across almost all of the broadcast channels is the same in the first and second proposed XML data placement schemes compared to the proposed XML data placement scheme in [23]. It

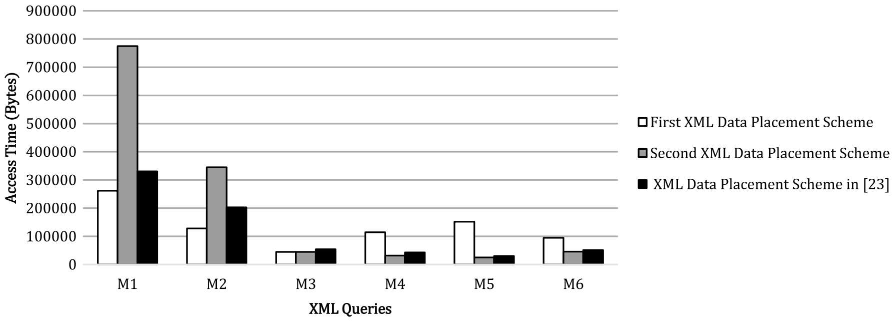


Fig. 13 Access time on the Mondial data set

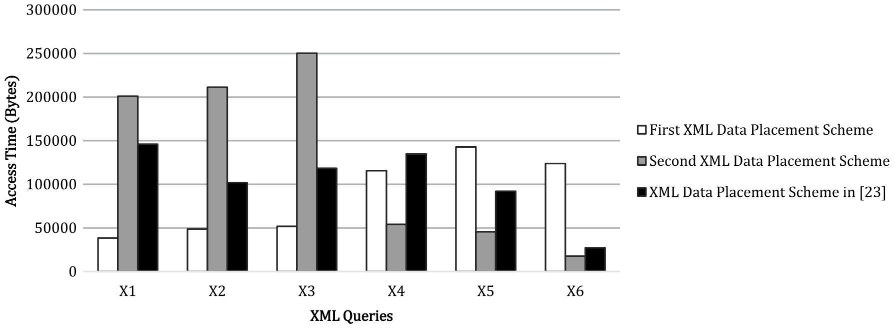


Fig. 14 Access time on the XMark data set

means that our proposed XML data placement schemes can better balance the dissemination of XML data over multiple wireless broadcast channels compared to the proposed XML data placement scheme in [23]. The reason is that the process of assigning a broadcast channel to disseminate all the XML nodes of a specified root-to-node path in [23] is based on the ascending order of paths in the path summary while this process in our proposed XML data placement schemes is based on the total size of the set of XML nodes of each root-to-node path in the path summary.

5.3 Experimental results on access time

Figures 13 and 14 show the access time of mobile clients to retrieve the results of XML queries over the broadcast channels in the Mondial and the XMark data sets, respectively. In all of the experiments, the first broadcast channel was used to disseminate the part of index segment and 7 broadcast channels were used to disseminate the part of data segment of the XML data stream. As shown in Figs. 13 and 14, the first XML data placement scheme seems to be successful in achieving the goal of improving the access time for the XML queries M1, M2, M3 in the Mondial data set and the XML queries X1, X2, and X3 in the XMark data set compared to the XML data placement XML proposed in [23]. The reason is that the XML nodes of root-to-node paths having the huge amounts of XML data have a higher priority to be disseminated over the broadcast channels in the first proposed XML data placement scheme. As shown in Table 10, the order numbers of XML queries M1, M2, M3 based on the size of their related XML nodes are 2, 4, and 7, respectively, and the order numbers of XML queries X1, X2, X3 based on the size of their related XML data are 7, 8, and 10, respectively. Therefore, the access times to retrieve the results of these XML queries are less than others since the XML nodes for these XML queries are disseminated sooner in the first proposed XML data placement scheme. On the other hand, the second XML data placement scheme seems to be successful in achieving the goal of improving the access time for the XML queries M4, M5, M6 in the Mondial data set and the XML queries X4, X5, and X6 in the XMark data set compared to the XML data placement scheme proposed in [23] since the second proposed XML data placement scheme keeps a balance between the access times in processing of XML queries having the root-to-node paths at the first of the path summary with large amounts of XML data and the XML queries having the root-to-node paths at the last of the path summary with small amounts of XML data.

Figures 15, 16, 17, 18 show the access time of mobile clients to retrieve the results of the XML queries M2 and M5 in the Mondial data set and the XML queries X2 and X5 in the XMark data set, respectively, when the number of broadcast channels is varied.

As shown in Figs. 15, 16, 17, 18, the access time of mobile clients to retrieve the desired XML data is reduced as the number of broadcast channels used in our experments increases in all of the XML data placement schemes. It means that the performance of XML querying over mobile wireless broadcast networks in terms of access time can be improved by disseminating the XML data over multiple wireless broadcast channels. However, the access time of mobile clients to retrieve the results of XML queries M2 and X2 in the first XML data placement scheme is less than other XML data placement schemes. The reason is that the XML nodes of root-to-node paths having the huge amounts of XML data (i.e., the results of XML queries M2 and X2) have a higher priority to be disseminated over the broadcast channels

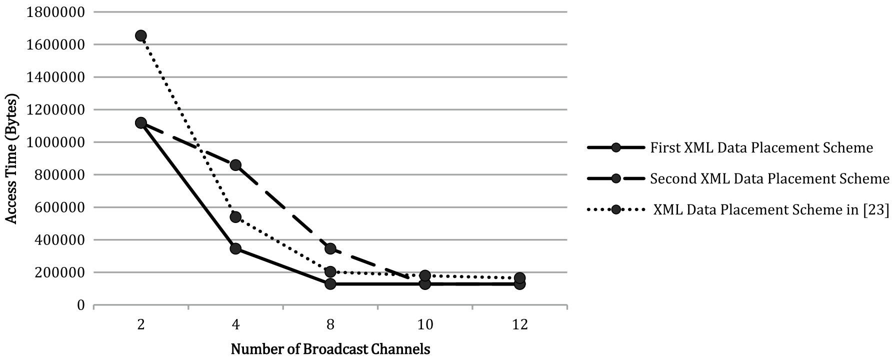


Fig. 15 Access time on the XML query M2

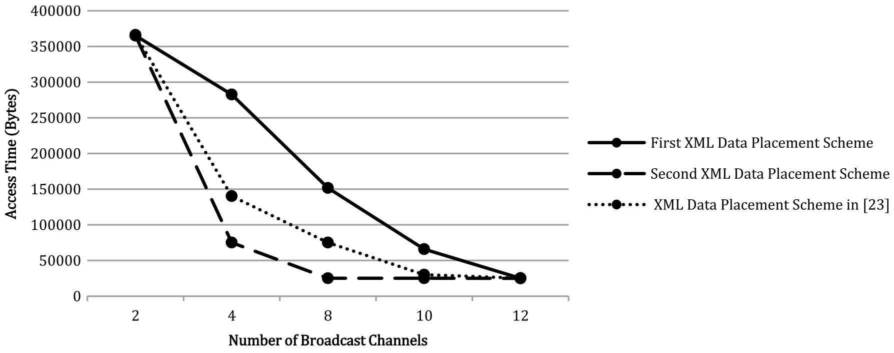


Fig. 16 Access time on the XML query M5

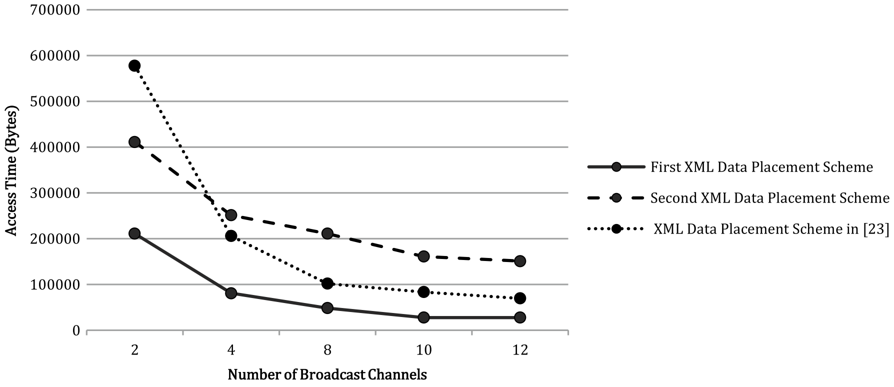


Fig. 17 Access time on the XML query X2

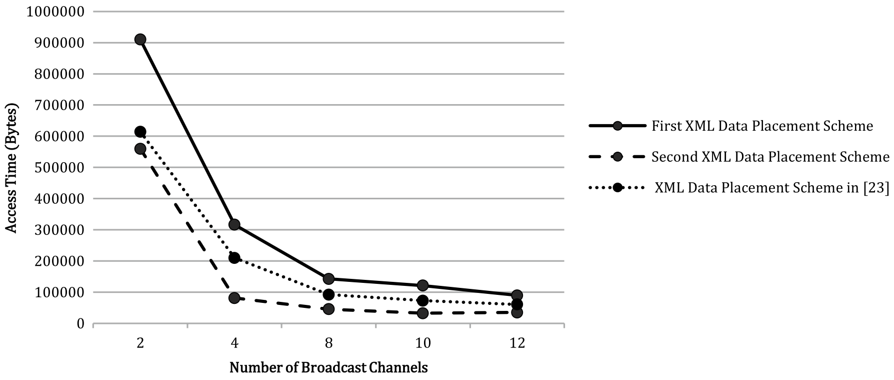


Fig. 18 Access time on the XML Query X5

in the first proposed XML data placement scheme. In addition, the access time of mobile clients to retrieve the results of XML queries M5 and X5 in the second XML data placement scheme is less than other XML data placement schemes. The reason

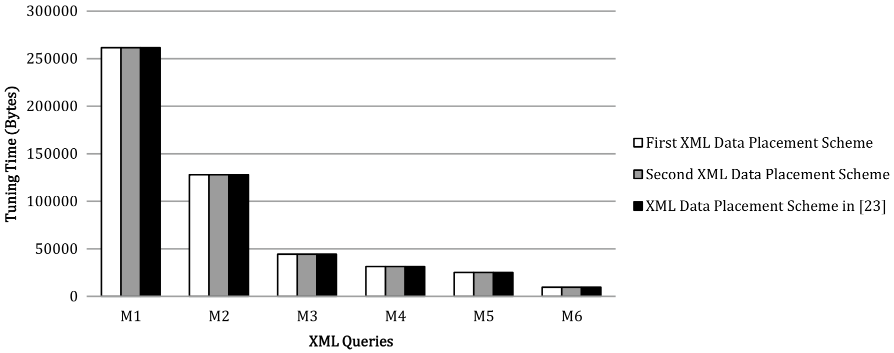


Fig. 19 Tuning time on the Mondial data set

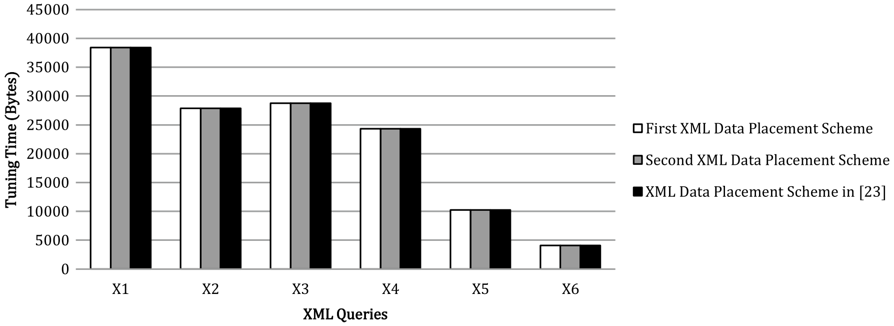


Fig. 20 Tuning time on the XMark data set

is that the second proposed XML data placement scheme keeps a balance between the access times in processing of XML queries having the root-to-node paths at the first of path summary with large amounts of XML data and the XML queries having the root-to-node paths at the last of path summary with small amounts of XML data.

5.4 Experimental results on tuning time

Figures 19 and 20 show the tuning time of mobile clients to retrieve the results of different XML queries over the broadcast channels in the Mondial and XMark data sets, respectively. In all of the experiments, the first broadcast channel was used to disseminate the part of index segment and 7 broadcast channels were used to disseminate the part of data segment. As shown in Figs. 19 and 20, the tuning time of a mobile client in retrieving the desired XML data did not change with increasing the number of wireless broadcast channels. It means that disseminating the XML data over multiple wireless broadcast channels does not degrade the performance of XML querying in terms of tuning time. In addition, the results show that the performance of XML querying in terms of tuning time over multiple wireless broadcast channels for all of the XML data placement schemes is the same since the structure of XML data stream (i.e., the structure of index segment and the structure of data segment) for all of the XML data placement schemes is the same.

6 Conclusion and future works

In this paper, two different XML data placement schemes were proposed to broadcast the XML data over multiple wireless broadcast channels. The aim of the first proposed XML data placement scheme was to improve the performance of XML querying in terms of access time in the case that the total size of XML nodes is large where these XML nodes are the results of XML queries with high depths. By performing different experiments using different XML data sets, it was concluded that the first XML data placement scheme could achieve its goal, and as much as the total size of XML nodes is larger, the goal is achieved better.

Furthermore, the aim of the second proposed XML data placement scheme was to keep a balance between the access times in processing of XML queries which have low depths with large amounts of XML data and the XML queries which have high depths with small amounts of XML data. The experimental results showed that the second XML data placement scheme could achieve its goal in most of the experiments.

In the future, some other subjects ignored in this paper can be considered. For example, some challenging issues like data lost and data confidentiality in broadcasting the XML data in mobile wireless broadcast networks can be considered. Actually, the wireless networks are unstable and thus the broadcast data might be lost. It is necessary to think of appropriate solutions for resolving such a problem. Furthermore, the broadcast data suffers from the lack of security since many clients can access data over the broadcast channels unlimitedly. Therefore, some suitable solutions should be proposed to guarantee the data confidentiality of XML data when they are disseminated over mobile wireless broadcast channels.

References

1. Qin Y, Sheng QZ, Wang H, Falkner NJG (2018) Organizing XML data in a wireless broadcast system by exploiting structural similarity. Wirel Pers Commun 98:1299–1329
2. Imielinski T, Viswanathan S, Badrinath BR (1997) Data on air: organization and access. IEEE Trans Knowl Data Eng 9:353–372
3. Imieĺinski T, Badrinath BR (1993) Data management for mobile computing. SIGMOD Record 22:34–39
4. Imielinski T, Viswanathan S, Badrinath BR (1994) Energy efficient indexing on air. In: Proceedings of the ACM SIGMOD International Conference on Management of Data, Minneapolis, Minnesota (USA). pp. 25–36
5. Acharya S, Alonso R, Franklin M, Zdonik S (1995) Broadcast disks: data management for asymmetric communication environments. In: Proceedings of the ACM SIGMOD International Conference on Management of Data, San Jose, California (USA). pp. 199–210
6. Chung YD, Kim MH (2000) An index replication scheme for wireless data broadcasting. J Syst Softw 51:191–199

7. Chung YD, Kim MH (2001) Effective data placement for wireless broadcast. Distrib Parallel Databases 9:133–150
8. Chung YD, Yoo S, Kim MH (2010) Energy and latency efficient processing of full-text searches on a wireless broadcast stream. IEEE Trans Knowl Data Eng 22:207–218
9. Lee C-C, Leu Y (2005) Efficient data broadcast schemes for mobile computing environments with data missing. Inf Sci 172:335–359
10. Boroujeni AB, Mirabi M (2016) A novel replication strategy for efficient XML data broadcast in wireless mobile networks. J Inf Sci Eng 32:309–327
11. Park C-S, Kim CS, Chung YD (2005) 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, Kunming (China), 2005, pp. 223–235
12. Park S-H, Choi J-H, Lee S (2006) 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 (DEXA 2006), Krakow (Poland), pp. 358–367
13. Park JP, Park C-S, Chung YD (2010) Energy and latency efficient access of wireless XML stream. J Database Manag 21:58–79
14. Mirabi M, Ibrahim H, Fathi L (2013) PS+Pre/Post: a novel structure and access mechanism for wireless XML stream supporting twig pattern queries. Pervasive Mob Comput 15(2014):3–25
15. Park JP, Park CS, Chung YD (2013) Lineage Encoding: An Efficient Wireless XML data Streaming Supporting Twig Pattern Queries. IEEE Trans Knowl Data Eng 24:1559–1573
16. Javani M, Mirabi M (2017) An efficient index and data distribution scheme for XML data broadcast in mobile wireless networks. J Inf Sci Eng 33:159–182
17. Shekarriz M, Babamir SM, Mirabi M (2020) Query processing optimization in broadcasting XML data in mobile communications. J Supercomput. https://doi.org/10.1007/s11227-020-03479-5
18. Safabahar B, Mirabi M (2017) A new structure and access mechanism for secure and efficient XML data broadcast in mobile wireless networks. J Syst Softw 125:119–132
19. Shokri M, Mirabi M (2019) An efficient stream structure for broadcasting the encrypted XML data in mobile wireless broadcast channels. J Supercomput 75:7147–7173
20. Lo S-C, Chen ALP (2000) Optimal index and data allocation in multiple broadcast channels. In: Proceedings of the 16th International Conference on Data Engineering, San Diego, California (USA), 2000, pp. 293–302
21. Yee WG, Navathe SB, Omiecinski E, Jermaine C (2002) Efficient data allocation over multiple channels at broadcast servers. IEEE Trans Comput 51:1231–1236
22. Yee WG, Navathe SB (2003) Efficient data access to multi-channel broadcast programs. In: Proceedings of the 12th International Conference on Information and Knowledge Management (CIKM 2003), New Orleans (USA), 2003, pp. 153–160
23. Mirabi H, Mirabi M, Borjian Boroujeni A (2016) An efficient XML data placement scheme over multiple wireless broadcast channels. J Inf Sci Eng 32:1183–1203
24. Haw SC, Lee CS (2009) Extending path summary and region encoding for efficient structural query processing in native XML databases. J Syst Softw 82:1025–1035
25. Mirabi M, Ibrahim H, Udzir NI, Mamat A (2012) An encoding scheme based on fractional number for querying and updating XML data. J Syst Softw 85:1831–1851
26. Bray T, Paoli J, Sperberg-McQueen CM, Maler E, F. Yergeau F (2008) Extensible markup language (XML) 1.0 (5th Edition) W3C Recommendation," http://www.w3.org/TR/REC-xml/
27. Berglund A, Boag S, Chamberlin D, Fernndez MF, Kay M, Robie J, Siméon J (2010) XML path language (XPath) 2.0 (Second Edition), http://www.w3.org/TR/xpath20/
28. Sun W, Qin Y, Yu P, Zhang Z (2007) On-demand XML data broadcast in wireless computing environments. In: Proceedings of the 3rd International Conference on Wireless Communications, Networking and Mobile Computing, Shanghai (China), pp. 3035–3038
29. Qin Y, Sun W, Zhang Z, Yu P, He Z (2008) Query-grouping based scheduling algorithm for on-demand XML data broadcast. In: Proceedings of the 4th International Conference on Wireless Communications, Networking and Mobile Computing, Dallian (China), pp. 1–4
30. Qin Y, Sun W, Zhang Z, Yu P, He Z, Chen W (2009) A novel air index scheme for twig queries in on-demand XML data broadcast. In: Proceedings of the 20th International Conference on Database and Expert Systems Applications, Linz (Austria), pp. 412–426

31. Qin Y, Wang H, Sun L (2011) Cluster-based scheduling algorithm for periodic XML data broadcast in wireless environments. In: 2011 IEEE Workshops of International Conference on Advanced Information Networking and Applications, Biopolis (Singapore), 2011, pp. 855–860
32. Sun W, Qin Y, Wu J, Zheng B, Zhang Z, Yu P, Liu P, Zhang J (2013) Air indexing for on-demand XML data broadcast. IEEE Trans Parallel Distrib Syst 25:1371–1381

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.