A NOVEL CACHE REPLACEMENT ALGORITHM FOR
COOPERATIVE CACHING IN WIRELESS
MULTIMEDIA SENSOR NETWORKS

VINCENT S. TSAENG\(^1\), MING-HUA HSIEH\(^1\) AND KAWUU W. LIN\(^2,\)*/

\(^1\)Dep. of Computer Science and Information Engineering
National Cheng Kung University
No. 1, University Road, Tainan 701, Taiwan

tsengsm@mail.ncku.edu.tw; mhhsieh@idb.csie.ncku.edu.tw

\(^2\)Dep. of Computer Science and Information Engineering
National Kaohsiung University of Applied Sciences
No. 415, Chien Kung Road, Kaohsiung 807, Taiwan

*Corresponding author: linwc@cc.kuas.edu.tw

Received October 2009; revised February 2010

ABSTRACT. In recent years, integrated applications with multimedia devices and wireless sensor networks promoted the evolution of wireless sensor networks, namely wireless multimedia sensor networks (WMSNs). The applications in WMSNs have to focus on both energy saving and application-level quality of service (QoS). Due to the characteristics in WMSNs, such as resource constraints and variable channel capacity, efficiently achieving the application-level QoS in WMSNs is a challenging task. To overcome this challenge, in this paper, we proposed a new kind of pattern named temporal region requesting pattern (TRRP) and a novel algorithm named TRRP-Mine for mining TRRPs efficiently. We also designed a temporal region requesting cost function of cache replacement, abbreviated as TRRC, for the cooperative caching multimedia content in WMSNs. Empirical evaluations under various simulation conditions showed that the proposed method delivers excellent performance in terms of hit rate and the number of replacements.

KEYWORDS: Multimedia sensor networks, Cache replacement, Temporal region requesting pattern, Data mining

1. Introduction. As wireless technologies progressed rapidly [18] and embedded microsensing MEMS technology facilitated wireless sensor networks (WSNs), the applications of wireless sensor network had attracted extensive attention in the past decade. With the capabilities of widespread surveillance, sensor networks are applied to a lot of applications, such as the environmental data collection [7,13], localization system [9] and pervasive health applications [8]. In recent years, the application that integrates the cheap CMOS cameras with microphones over WSNs is becoming a trend, and this kind of WSNs is named Wireless Multimedia Sensor Networks (WMSNs) [1].

The existing studies of WSNs mainly focus on the energy saving problem [10,14]. Nevertheless, the challenges of developing the applications in WMSNs are not only the energy saving but also the quality of service (QoS) [6] issue in application-level. A general network layer metric for QoS is the network latency. The QoS in multimedia content delivery over Internet can be achieved through DiffServ [6] or Intserv [6]. However, such solutions for providing QoS over Internet face the severe bottlenecks due to the limited power and memory space of sensor nodes. An intuitive way to provide QoS in WMSNs is applying the caching technique to sensor nodes. Assume that each sensor node is equipped with a local storage and capable of caching a small number of requests. When a sensor node
issues a request, the sensor node will check whether it caches the requested item or not. If the requested item is held in the local cache, there is no need to obtain the item from other nodes and the latency is thereby minimized. Based on the caching strategy, a node importance-based cooperative caching method named NICoCa [5] was proposed. By incorporating the node importance of the WMSNs and the residual energy of each sensor node in consideration, NICoCa can prolong the network lifetime and provide QoS in WMSNs. Although the node importance of WMSNs plays a major role in cooperative caching, the attributes of multimedia item are essential factors in cache replacement that should be carefully treated. However, only taking the attributes of multimedia item, such as data size and the timestamp of the latest access into consideration for cache replacement is insufficient. For instance, although a multimedia item is temporally requested by some sensor nodes, it may suffer from a frequent replacement. This is because the multimedia item with larger data size has a higher priority to be selected as the candidate victim of cache replacement. In this way, the network latency would be increased because the sensor node must issue broadcast for the requested multimedia item. For efficiently decreasing the network latency to improve QoS in WMSNs, the selection of adaptive candidate of cache replacement is extremely important. Therefore, we will also focus on the problem of precisely determining a multimedia item that is temporally requested. In addition, the main goal of this work is to propose a pattern-based cost function for the design of cache replacement.

In order to provide QoS in WMSNs, in this paper, we further defined a new kind of pattern, named temporal region requesting pattern (TRRP), and proposed an algorithm named TRRP-Mine for mining TRRPs from the temporal requesting information of sensor nodes efficiently. In addition, we also designed a temporal region requesting cost function (TRRC) of the candidate cache victim to assist the cache replacement in reducing replacements and increasing hit rate. Through empirical evaluations, our proposed approach was shown to outperform other existing approaches in terms of hit rate and the number of replacements. The rest of this paper was structured as follows. Preliminaries were stated in Section 2. In Section 3, we described the problem definition, proposed pattern, mining algorithm, the proposed cost function of the candidate cache victim, and an elaborate example. Some simulation results were made in Section 4. A conclusion was given in Section 5.

2. Preliminaries. In the past decades, a lot of studies had probed into the caching techniques on many applications such as databases [12], web applications and wireless networks. Existing researches in caching on Web could be divided into two main architectures: 1) cooperative and 2) non-cooperative [15]. Besides, there are a great number of caching approaches for wireless cellular network [11]. Some cooperative caching protocols have been proposed for mobile ad hoc networks (MANETs) [2,4,17]. By utilizing geographical proximity or well-known clustering algorithms, a region concept has been employed for the cooperative caching approaches [2,4,17]. In [19], the authors took both data and node locality into considerations and proposed three approaches, which were Cache-Data, CachePath and HybridCache respectively. However, the aforementioned caching techniques do not suit WMSNs due to the severe resource constraints like memory space and energy consumption. Most of existing studies on caching technique of wireless sensor networks focus on the placement of caches [16]. In [5], nevertheless, Dimokas et al. first proposed an approach of cooperative caching in WMSNs. A node importance-based cooperative caching (NICoCa) in [5] created a precedent of cooperative caching on WMSNs.

The plain assumption of NICoCa is that each sensor node has a moderate local storage capacity associated with it, i.e., a flash memory. NICoCa exploits the importance of
sensors relative to the network topology in terms of their positions in the network and/or residual energy. A concept of mediator node [5], which plays a significant part of position in the network, is proposed to aid cooperative caching. Integrating the sensor’s position and residual energy into the caching strategy facilitates the prolongation of network lifetime and short latency of multimedia data retrieval. In our network model, we adopt the concept of NICoCa for the cache discovery component protocol [5] as shown in Figure 1. Four cases of the requested item may occur: 1) Local Hit (LH), 2) Proximity Hit (PH), 3) Remote Hit (RH) and 4) Global Hit (GH). When a sensor node issues a request of a
multimedia item, it first searches its local cache. A Local Hit represents that the multimedia item is found on the local cache and a Proximity Hit denotes that the requested multimedia item is cached by a node in the 2-hop neighborhood of the original sensor node. If the multimedia item is not cached on the original sensor node, the original sensor node broadcasts the request to its 2-hop neighbors and mediator nodes for Proximity Hit. If a proximity miss occurs, the request’s route toward the Data Center proceeds. During the routing, if the requested item is cached by a node having at least one mediator node residing along the routing path, it is called Remote Hit. A Global Hit means that the requested item is acquired from the Data Center. As far as we know, the more the number of LH, PH and RH is, the better the QoS is. Considering the restrictions on hardware, most caching techniques require a cache replacement mechanism. An adaptive cache replacement algorithm benefits the improvement of LH, PH and RH. Figure 2 illustrates the flow chart of cache replacement in NICoCa. A purging procedure of the cached items proceeds when the available space is smaller than the required space. For the selection of the candidate cache victim, Dimokas et al. proposed a cost function integrating many factors, such as data size, concept of Least Recently Used (LRU), Time-to-Live and the number of the multimedia item requested by the sensor node.

The concept of LRU and the number of the multimedia item requested by the sensor node may be insufficient to reveal the requesting information such as items are temporally requested by some sensor nodes. Considering the requesting information, a temporal region requesting pattern (TRRP) was proposed in our work. A TRRP represents that the multimedia item is temporally requested by a sensor region and reveals the relation between the multimedia items’s requesting information and the sensor node. In next section, the problem definition, the definition of TRRP, mining methodology, the proposed cost function and an elaborative example were described.

3. Problem Definition and Mining Approach. In our work, we adopted the network model as proposed in NICoCa [5]. In this model, it is assumed that the elemental source of multimedia data is a Data Center and sensor nodes are capable of caching the datum which have requested. In addition, it is supposed that each sensor node is aware of its 2-hop neighborhood. An ordinary sensor routing protocol accompanies the sending of requests. Each sensor node stores the related metadata of a cached multimedia item (datum) [5]. A request item is in the form as (dataID, timestamp) and represents that a sensor node launches a request for a multimedia item, dataID, at the timestamp. A request transaction of a sensor node consists of a series of request items. Therefore, a request transaction table from NICoCa recording the request logs of each sensor node [5] can be obtained as Table 1 shown. Without loss of generality, some multimedia items are requested temporally by some nodes which are 2-hop neighbors of each other. For the existing cache replacement in WMSNs, to determine whether a multimedia item is purged from the cache of the sensor node or not mainly depends on the size of the multimedia item and the last access timestamp. However, some multimedia items with large data size were temporally requested by the mediator nodes locating a specific region. Such multimedia items may suffer from a frequent replacement due to their large data size. To solve the above problem, one of our purposes was to discover temporal region requesting patterns. A temporal region requesting pattern represents that the multimedia item is temporally requested by the sensor nodes locating a 2-hop region. In our work, we named the temporal region requesting patterns as TRRPs. A mining problem of TRRPs was explained in next subsection.
Table 1. Request transaction table

<table>
<thead>
<tr>
<th>Sensor Node ID</th>
<th>Request Transaction</th>
</tr>
</thead>
<tbody>
<tr>
<td>V₁</td>
<td>(M₁, 21), (M₁, 24), (M₂, 56), (M₃, 75), (M₂, 78)</td>
</tr>
<tr>
<td>V₂</td>
<td>(M₃, 37), (M₄, 44), (M₄, 48), (M₂, 90), (M₃, 102)</td>
</tr>
<tr>
<td>V₃</td>
<td>(M₃, 3), (M₄, 37)</td>
</tr>
<tr>
<td>V₄</td>
<td>(M₄, 50), (M₁, 57)</td>
</tr>
</tbody>
</table>

3.1. Formulation of data mining problem. In this section, we defined some mining terms and formulated our data mining problem. A request transaction of a sensor node, \( v \), is in the form as \( \{(R₁, T₁), (R₂, T₂), \ldots, (Rₙ, Tₙ)\} \), where \( Rᵢ \) represents the requested multimedia item and \( Tᵢ \) denotes the timestamp when \( v \) launches the request and the number of requests issued by \( v \) is denoted as \( RT(v) \). Let \( D_{RT} = \{RT₁, RT₂, \ldots, RTₖ\} \) be a request transaction database that contains \( k \) request transactions.

**Definition 3.1.** Given some sensor nodes, \( S_n = \{v₁, v₂, \ldots, vₖ\} \), where \( vᵢ \) represents the sensor node, compose a sensor region if every sensor node in \( S_n \) is the 2-hop node of each other.

**Definition 3.2.** Given a sensor region, \( S \), and a multimedia item, \( R \), \( R \) is called a region-requesting multimedia item of \( S \) if some sensor nodes in \( S \) request \( R \).

**Definition 3.3.** Given a sensor region, \( S = \{v₁, v₂, \ldots, vₙ\} \), and a region-requesting multimedia item of \( S \), \( R \), where \( vᵢ \) is the sensor node in \( S \), the number of \( vᵢ \) requested \( R \) is defined as \( \text{num}(vᵢ, R) \).

**Definition 3.4.** The time sequence of \( S \) requested \( R \) is defined as: \( TS(S, R) = (T₁, T₂, \ldots, Tₘ) \), where the ascending order of elements in a sequence is decided by using the time as the key.

**Definition 3.5.** Given a threshold of time window, \( δ \), and any two successive requesting timestamps of \( TS(S, R) \), namely \( Tᵢ \) and \( Tᵢ₋₁ \), if \( Tᵢ - Tᵢ₋₁ \leq δ \), it is called that \( Tᵢ \) is \( δ \)-temporal to \( Tᵢ₋₁ \).

**Definition 3.6.** The \( δ \)-temporal continuity of \( TS(S, R) \) represents the number of \( Tᵢ \) \( δ \)-temporal to \( Tᵢ₋₁ \), where \( Tᵢ \) and \( Tᵢ₋₁ \) are two successive requesting timestamps, and is defined as:

\[
Δ(TS(S, R)) = \text{num}(Tᵢ)
\]

where \( Tᵢ - Tᵢ₋₁ \leq δ \), \( Tᵢ \) and \( Tᵢ₋₁ \) are timestamps of \( TS(S, R) \).

**Definition 3.7.** The purity of \( δ \)-temporal continuity of \( TS(S, R) \) is defined as:

\[
P(TS(S, R)) = \frac{Δ(TS(S, R))}{m}
\]

**Definition 3.8.** Given a purity threshold, \( ε \), \( R \) is called a temporal region requesting pattern (TRRP) of \( S_v \) if the following property holds.

\[
P(TS(S, R)) \geq ε \quad \text{and} \quad S_v = \{vᵢ | vᵢ \in S, \text{num}(vᵢ, R) \geq \frac{Δ(TS(S, R))}{|S|} \} \neq \phi
\]
Definition 3.9. A temporal reference rate (TR) of R requested by vi represents the degree of that vi temporally requests R and is derived by the following equation.

\[
TR(v_i, R) = \begin{cases} 
\frac{\sum_{j=1}^{\text{num}(v_i, R)} \text{num}(v_i, R)}{\text{Max}(RT(v_i))} & \text{if } R \text{ is a TRRP of } S_v, \ S_v = \{v_1, v_2, \ldots, v_i\}, \ S_v \subseteq S \text{ and } \text{num}(v_i, R) \geq \frac{\Delta(TS(S, R))}{|S_v|} \\
0 & \text{otherwise}
\end{cases}
\]

In Equation (4), the Max(\(RT(v_i)\)) means the largest number of the requests issued by the sensor node among the sensor region, Sv. TR\(v_i, R\) is calculated by Equation (4) when R is a TRRP and the number of R requested by vi is larger than the average number of R requested by Sv. Otherwise, TR\(v_i, R\) is set as 0. The concept of TR can quantify the degree of R temporally requested by vi.

Definition 3.10. A future discarding rate (FDR) of R requested by vi denotes the degree of that R would be chosen to discard by vi in the future and is derived by the following formulas. Given: \(v_i \in S\), \(TS(S, R) = (T_1, T_2, \ldots, T_m)\)

\[
T_j - T_{j-1} = \begin{cases} 
T_j - T_{j-1} & \text{if } T_j - T_{j-1} > \delta \\
0 & \text{otherwise}
\end{cases}
\]

(5)

\[
Dis(v_i, R) = \begin{cases} 
1 + \frac{m}{\text{Max}(m, \delta)} \sum_{j=2}^{m} \frac{T_j - T_{j-1}}{\text{Max}(m, \delta)} & \text{if } \text{num}(v_i, R) > 0 \\
1 + \frac{m}{\text{Min}(m, \delta)} \sum_{j=2}^{m} \frac{T_j - T_{j-1}}{\text{Min}(m, \delta)} & \text{otherwise}
\end{cases}
\]

(6)

\[
FDR(v_i, R) = \frac{\text{Dis}(v_i, R)}{m}
\]

(7)

Dis\(v_i, R\) is used to calculate the future discarding rate of R corresponding to vi. If R had been requested by vi, Dis\(v_i, R\) would be less than Dis\(v_j, R\) where \(v_j\) never requested R. The concept of FDR is evolved from the Belady’s algorithm. If the time gap between two successive time stamps of R requested by S is larger, it implies that the probability of R requested by S would be smaller in the near future.

Hence, the mining problem is defined as follows. Given a request transaction database, \(D_{RT}\), containing the request transactions of sensor nodes, two specific thresholds, namely temporal window size, \(\delta\) and purity threshold, \(\varepsilon\), the problem is to discover the temporal region requesting patterns (TRRPs). Based on the above definitions, a mining algorithm to discover TRRPs and its procedure were described in next subsection.

3.2. TRRP-Mine methodology. Our method was named Temporal Region Requesting Pattern Mining (TRRP-Mine). The algorithm was illustrated with Figure 3. First, all the sensors were clustered into the 2-hop region set (line 1). Then, the each 2-hop region was retrieved from the 2-hop region set one by one (lines 2 and 3). The time sequence table of the 2-hop region requesting all multimedia items was initialized (line 4). The time sequences of each sensor node in the 2-hop region requesting all items was added to time sequence table (from lines 5 to 8). Afterwards, the time sequences of each multimedia item requested by the 2-hop region were retrieved from the time sequence table and taken as an input for temporal pattern mining, abbreviated as TP-Mine (from lines 9 to 12). Considering the computation ability of sensor node, TP-Mine only required one scanning of the time sequences to discover the TRRPs. In Figure 4, the inputs of TP-Mine were the requesting time sequences of the multimedia item, A, the 2-hop region, NH, time window threshold, \(\delta\) and purity threshold, \(\varepsilon\). In the beginning, a table storing temporal region requesting patterns was initialized (line 1). The time gap between any two successive requesting timestamps, namely \(T_i\) and \(T_{i-1}\), was checked whether \(T_i\) is \(\delta\)-temporal
to $T_{i-1}$. If $T_i$ was $\delta$-temporal to $T_{i-1}$, the $\delta$-temporal continuity of $TS(NH, A)$ would be with a increment of one (from line 2 to line 4). Otherwise, the future discarding rate of A requested by the members of NH would be increased (from lines 6 to 19). Afterwards, based on the given parameters, it was called that A is a TRRP of NH if the purity of $\delta$-temporal continuity of $TS(NH, A)$ was not less than $\varepsilon$ (from lines 20 to 32). The temporal reference rate of A requested by the sensor node in NH was calculated based on Equation (4). The spirit of TRRP-Mine was two-folded, 1) discover the temporal region requesting pattern, and 2) derive the temporal reference rate and the future discarding rate of each multimedia item requested by each sensor node.

In the next subsection, a novel cost function of the candidate cache victim integrating the temporal reference rate and future discarding rate for cache replacement was proposed.

### 3.3. Cost function of cache replacement

In this subsection, we described our proposed cost function of the candidate cache victim, namely Temporal Region Requesting Cost Function, abbreviated as TRRC. Different from the existing cache algorithm, such as MRU, LRU and the cache replacement policy of NICoCa, TRRC employed the concepts of temporal reference and future discarding rate. Given a multimedia item, $R$, and a sensor node, $V$, the formula of TRRC is listed as follows:

$$TRRC(R,V) = \frac{DataSize(R)}{TTL(R)} \times \frac{1}{num(V,R)} \times FDR(V,R) \times (1 - TR(V,R))$$  \hspace{1cm} (8)

The candidate cache victim of the cached items is the one with the largest cost calculated based on TRRC. In Equation (8), the $num(V,R)$ denotes the number of $R$ requested by $V$; the $FDR(V,R)$ represents the future discarding rate of $R$ requested by $V$; the $TR(V,R)$ means the temporal reference rate of $V$ requesting $R$. If $R$ was requested rarely by $V$ and the time gap between two successive requesting stamps of $V$ requesting $R$ was too large, the $FDR(V,R)$ was magnified in the procedure of TRRP-Mine. In this way, the probability of that $R$ was chosen as a candidate cache victim of $V$ increased correspondingly. On the contrast, if $R$ was a TRRP of a sensor region containing $V$, then the $TR(V,R)$ was bigger than the other cached items of $V$ and the corresponding TRRC($R,V$) was smaller than that of the other cached items. Therefore, as the
FDR(V, R) was bigger than that of the other cached items and TR(V, R) was smaller than that of those ones, it implied that the probability of that R would be requested by V in the future was smaller than the other items.

Input: requesting time sequence, SEQ, 2-hop member, NH, time_window, $\delta$, and continuous purity, $\epsilon$.
Method: TP-Mine
Output: TRRP
1. TRRPSET $\leftarrow \phi$
2. FOR i FROM 2 TO SEQ.size
3. IF $T_i - T_{i-1} < \delta$
4. continuity $\leftarrow$ continuity + 1
5. ELSE
6. FOR j FROM 1 TO NH.size
7. $S_n$ $\leftarrow$ NH.getSensor(j)
8. IF $S_n$ had requested A
9. $\text{Dis}[A][S_n] \leftarrow \text{Dis}[A][S_n] + (T_i - T_{i-1})/\text{Math.max}(\text{SEQ.size}, \delta)$
10. ELSE
11. $\text{Dis}[A][S_n] \leftarrow \text{Dis}[A][S_n] + (T_i - T_{i-1})/\text{Math.min}(\text{SEQ.size}, \delta)$
12. ENDIF
13. ENDIF
14. END FOR
15. ENDFOR
16. FOR j FROM 1 TO NH.size
17. $S_n$ $\leftarrow$ NH.getSensor(j)
18. $\text{FDR}[A][S_n] \leftarrow \text{Dis}[A][S_n] / \text{SEQ.size}$
19. ENDFOR
20. IF continuity > SEQ.size $\times \epsilon$
21. R $\leftarrow \phi$
22. FOR j FROM 1 TO NH.size
23. $S_n$ $\leftarrow$ NH.getSensor(j)
24. IF occurrence[A][S_n] $\times$ NH.size > continuity
25. add($S_n$, R);
26. ENDFOR
27. ScoreTmpRefRate($S_n$, A)
28. ENDFOR
29. IF R not equals $\phi$
30. AddToTRRPSET(A,R)
31. ENDFOR
32. ENDFOR

FIGURE 4. TP-Mine algorithm
Taking both the attributes of multimedia item and requesting information of the item cached by the sensor node into consideration, TRRC can avail the cache replacement based on the requested probability built by temporal requesting information.

3.4. An elaborative example. We illustrated the process of discovering TRRPs by an elaborative example. As Table 1 shown, we supposed that sensor nodes composed a sensor region, S. In addition, $M_1$, $M_2$, $M_3$ and $M_4$ are region-requesting multimedia item of S. The time sequences of S requested $M_1$, $M_2$, $M_3$ and $M_4$ are listed as $\langle 21, 24, 57 \rangle, \langle 36, 78, 90 \rangle, \langle 3, 37, 75, 102 \rangle$ and $\langle 37, 44, 48, 50 \rangle$ respectively. Given temporal window size, $\delta = 10$, and purity threshold, $\varepsilon = 0.5$, and take the TS(S, $M_1$) as an example. The $\delta$-temporal continuity of TS(S, $M_1$) is 1, and the purity of $\delta$-temporal continuity of TS(S, $M_1$) is 0.334 less than $\varepsilon$. Hence, $M_1$ is not a TRRP and the temporal reference rate of $M_1$ requested by the sensor nodes in S is 0. The future discarding rate of $M_1$ requested by $V_1$ could be calculated as $\frac{1+(57-24)}{10}/3$, which is 1.43333. Since $V_1$ requests $M_1$, the future discarding rate of $M_1$ requested by $V_1$ is the same as the one of $M_1$ requested by $V_1$. On the contrast, the future discarding rate of $M_1$ requested by $V_2$ could be calculated as $\frac{1+(57-24)/3}{3}$, which is 4. Although $V_1$ and $V_2$ have the same temporal reference rate and future discarding rate of $M_1$, the TRRC($M_1$, $V_1$) is smaller than TRRC($M_1$, $V_2$). This is because the number of $V_1$ requested $M_1$ is larger than the one of $V_2$ requested $M_1$. A TRRP could be discovered from the TS(S, $M_1$). The $\delta$-temporal continuity of TS(S, $M_1$) is 3, and the purity of $\delta$-temporal continuity of TS(S, $M_1$) is 0.75 larger than $\varepsilon$. The temporal reference rate of $M_4$ requested by $V_1$ is 0 since $V_1$ does not request $M_4$. The temporal reference rates of $M_4$ requested by $V_2$, $V_3$ and $V_4$ are the same (4/5 = 0.8). Due to that $V_2$ requests $M_4$ two times, the TRRC($M_4$, $V_2$) is smaller than TRRC($M_4$, $V_3$) and TRRC($M_4$, $V_4$). The above example shows that the temporal continuity gives an impact on the candidate of cache replacement. In the next section, a number of experiments were conducted to show the performance of TRRC.

4. Experimental Results. In this section, we stated the descriptions of the data set generated by our data simulator and the metrics of our evaluations. Then, a number of experiments were conducted through comparisons with different cache algorithms.

4.1. Experimental settings and metrics. All experiments were conducted on a P4-2.0 G Hz machine with 1 GB main memory. The proposed approach and data generator were implemented by using JAVA. We referred to the simulation model [5], and defined the primary parameters for the simulation model listed in Table 2. We collected the requesting logs of three days for mining TRRP and took the requesting logs of one day as the testing data sets. The most popular metrics of cache algorithm are hit rate and number of replacement. The hit rate is defined as:

$$
Hit\ Rate = \frac{\text{Num(LocalHit)} + \text{Num(ProximityHit)} + \text{Num(RemoteHit)}}{\text{Num(Request)}}
$$

Obviously, a higher hit rate and a smaller number of replacements represent a better performance of the cache replacement algorithm. A better cache replacement algorithm can benefit the decrease in the communication cost and energy consumption. Prolongation of network lifetime and short latency in multimedia data retrieval are the aimed goals of this work.

4.2. Comparisons of various cache replacement algorithms. Four major experiments were conducted by varying the following parameters, 1) ItemNum (number of multimedia item), 2) CacheSize (cache memory size of sensor node), 3) AvgItemSize (average data size of multimedia item) and 4) AvgEventProb (The event probability of the
requested items). The event probability of the requested items means the probability of the sensor node issuing the requests based on the historical requesting logs.

Table 2. Primary parameters of the simulation model

<table>
<thead>
<tr>
<th>Parameter Name</th>
<th>Default Value</th>
<th>Range</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$N_s$</td>
<td>30x30</td>
<td></td>
<td>Number of sensor node</td>
</tr>
<tr>
<td>ItemNum</td>
<td>3000</td>
<td></td>
<td>Number of multimedia item</td>
</tr>
<tr>
<td>CacheSize</td>
<td>100 (mb)</td>
<td></td>
<td>Cache memory Size of sensor node</td>
</tr>
<tr>
<td>AvgItemSize</td>
<td>5 (mb)</td>
<td>1-10</td>
<td>Average data size of multimedia item</td>
</tr>
<tr>
<td>AvgRequestNum</td>
<td>2000 (per day)</td>
<td>1000-3000</td>
<td>Average number of requests issued</td>
</tr>
<tr>
<td>AvgEventProb</td>
<td>0.2</td>
<td>0.1-0.3</td>
<td>Event probability of requested</td>
</tr>
<tr>
<td>AvgEventLength</td>
<td>6</td>
<td>3-10</td>
<td>Event length of requested</td>
</tr>
</tbody>
</table>

We investigated the hit rate and number of replacements of five cache replacement algorithms, including LRU, MRU, NICoCa, Belady’s Algorithm+NICoCa and TRRC. The difference of the above algorithms is how to choose the candidate cache victim. LRU and MRU are well-known cache algorithms, which take the timestamp of the request as the discarding priority. LRU discards the least recently used item first. In contrast to LRU, the most recently used item is discarded first by MRU. The cache replacement policy of NICoCa considers four factors, data size of cached item, Time-to-Live, requested timestamp, and the occurrence of requested item. Belady’s Algorithm+NICoCa integrates the cost function of NICoCa into Belady’s Algorithm. The Belady’s algorithm is the ideal cache algorithm and most used to compare the effectiveness. In our experiments, Belady’s Algorithm+NICoCa was supposed that it could predict the next 70 requests in the future. If more than one cached item were required to purge, the candidate cache victim would be selected based on the cost function of NICoCa. Through the experiment of adjusting parameters, we found that when the temporal window size, $\delta$, was set as 60 time units and purity threshold, $\varepsilon$, was set as 0.5, TRRC could have a higher hit rate and fewer replacements. For the simplicity, therefore, the parameter settings of TRRR-Mine in the four major experiments were to set temporal window size as 60 time units and purity threshold as 0.5.

Figures 5(a) and 5(b) showed that the results of varying the number of multimedia item. In Figures 5(a) and 5(b), we observed that the hit rate of the five methods decreased and the number of replacement of them increased as the increase in the number of multimedia item. This was because the number of LH, PH and RH decreased with the increase in the number of multimedia item which the sensor node chose to issue. Among the five methods, LRU and MRU only took the requesting timestamp to calculate the cost of the candidate cache victim. This was the reason that both LRU and MRU had an obviously lower hit rate and more replacements than the others. In terms of hit rate and replacements, TRRC was the most near-optimal method. In addition, the hit rate of TRRC outperformed NICoCa by 7% and the number of replacements of TRRC was smaller than NICoCa by 20%. The results of varying the memory size of cache were shown as Figures 6(a) and 6(b). With the extension of the memory, the number of multimedia item held by the cache increased. In this way, the number of replacement and the hit rate of the five methods had progressive improvement. It was also observed that the hit rate and the number of replacements of LRU, NICoCa and TRRC were gradually near to them of Belady’s Algorithm+NICoCa. This could be explained by that the larger the memory size of cache was, the number of invoking the replacement mechanism was
smaller. In this experiment, TRRC still outperformed the other methods except the Belady’s Algorithm+NICoCa.

We also investigated the hit rate and the number of replacements with the variation in the average data size of the multimedia item. As shown in Figures 7(a) and 7(b), the number of replacement increased progressively and the hit rate of the five methods had obvious degeneracy. This was because the number of multimedia item held by that the cache decreased with the increase in the average data size of multimedia item. A special phenomenon was that as the average data size of multimedia item exceeded 7.5mb, TRRC had a higher hit rate than Belady’s Algorithm+NICoCa. This phenomenon could be explained by the following conditions. As the average data size of multimedia item increased it implied that the replacement mechanism was invoked frequently and the number of multimedia item held by the cache decreased. If the number of multimedia item held by the cache was small and the number of predicted requests in the future (defaulted as 70) was large, all the cached items might occur in the predicted requests.
In this way, the replacement mechanism had to calculate the cost of all the cached items to decide the candidate cache victim. Nevertheless, the cached item with the largest cost might be the one requested in the next K requests, where K was smaller than 70.

The experiments of adjusting the event probability were shown as Figures 8(a) and 8(b). A higher event probability represents that the sensor node more likely issues the requests which have been issued in the past. Therefore, with the increase in the event probability, the hit rate of the five methods had a gradual raise. In the meanwhile, the number of replacements of the five methods had correspondingly decreased. The cost functions of NICoCa, Belady’s Algorithm+NICoCa, and TRRC consider the occurrence of the requested items as well. As the event probability increased, therefore, the gap of the hit rate between the LRU and NICoCa enlarged.

5. Conclusions. This paper proposed a novel pattern named temporal region requesting pattern (TRRP) and a one-scanning mining approach, namely TRRP-Mine, to discover the TRRPs. In addition, by utilizing the discovered TRRPs, a pattern-based cost function, named as Temporal Region Requesting Cost Function (TRRC), was designed for cache replacement of WMSNs. In TRRC, two novel factors, namely temporal reference rate and future discarding rate, were proposed to calculate the score of candidate cache victim based on the requested probability built by temporal requesting information. To our best knowledge, this was the first work applying the pattern-based cost function to explore cache replacement of WMSNs. In the simulations, we compared TRRC with some well-known cache algorithms such as LRU, MRU, NICoCa and Belady’s Algorithm+NICoCa in terms of hit rate and the number of replacement.

In the meantime, the performance was evaluated by adjusting four key parameters, which are ItemNum, CacheSize, AvgItemSize and AvgEventProb respectively. Through empirical evaluations of different parameter settings, the hit rate of TRRC was higher than that of the other methods by 7% and the replacements of TRRC were fewer than those of the others by 25% at least. This result indicated that TRRC achieved efficiently the short latency in multimedia data retrieval. Although the proposed pattern-based cost function (TRRC) requires a period of collection of requesting information to train temporal reference rate and future discarding rate, for the future work, we will apply the stream mining technique for achieving the real-time application.

Acknowledgment. This research was partially supported by National Science Council, Taiwan, under Grant No. 97-2218-E-151-003-MY2.

REFERENCES

Figure 7. (a) Hit rate for LRU, MRU, NICoCa, Belady’s algorithm+NICoCa and TRRC with the average data size of multimedia item varied; (b) Number of replacement for LRU, MRU, NICoCa, Belady’s algorithm+NICoCa and TRRC with the average data size of multimedia item varied.

Figure 8. (a) Hit rate for LRU, MRU, NICoCa, Belady’s algorithm+NICoCa and TRRC with the event probability varied; (b) Number of replacement for LRU, MRU, NICoCa, Belady’s algorithm+NICoCa and TRRC with the event probability varied.


