Scheduling Analysis of Broadcasting Real-Time Data Based on Prediction Scheme over Wireless Multi-Channels

Ding-Jung Chiang
Department of Digital Multimedia Design
Taipei Chengshih University of Science and Technology
Taipei, Taiwan, R.O.C
djchiang@tpcu.edu.tw

Ching-Sheng Wang
Department of Computer Science and Information Engineering
Aletheia University
New Taipei, Taiwan, R.O.C
cswang@mail.au.edu.tw

Chien-Liang Chen
Department of Computer Science and Information Engineering
Aletheia University
New Taipei, Taiwan, R.O.C
clchen@au.edu.tw

Wen-Jay Lo
Exploration and Development Research Institute
CPC Corporation
Miaoli, Taiwan, R.O.C
155276@cpc.com.tw

***Abstract*—With the rapid progress of wireless technology, mobile users can retrieve multiple real-time data with portable devices from mobile service centers. Providing deadline guarantees for queries over mobile environments is a challenging problem due to real-time data arrival rates and time-varying data contents. In this paper, we propose a prediction-based scheme for periodic continuous queries over wireless multi-channels. It is an important issue to effectively disseminate various materials in mobile environments. This paper highlights important problems that have mobile real-time systems from improving the system performance it could be for periodic continuous queries. While current systems aim to foster significant improvements in access latency, this paper argues that most systems are still limited to just being online material without performance concerns. Mobile real-time infrastructure has become a topic for research. A performance-driven model to mobile real-time delivery is proposed in this paper. A novel methodology for deploying periodic continuous queries based on prediction mechanism in mobile environments is presented. We focus on describing the dynamic processing in terms of performance, rather than the details of its implementation.**

***Keywords*—Wireless Multi-Channel, Kalman Filter, Broadcast Real-Time Data,Periodic Continuous Queries**

I. Introduction

Over the last decade the rapid advance of cloud computing and mobile network technologies has radically changed the potential for periodic continuous queries described in [1], [2] and [3]. Wireless communication and computing technology can facilitate important issues and provide access to a wide variety of quality real-time resources. Two of the key technological points are real-time data and wireless network. These technologies are making more and more resources available for periodic continuous queries, but their effectiveness is often debated. Further, even if real-time databases can help achieve periodic continuous outcomes, technologies by themselves rarely have substantial impact on periodic continuous queries. As the mobile environments in which we use technology become more complex and more diverse, we need to extend and expand our notion of usability to include a broad spectrum of data with time constraint and periodic continuous queries described in [4] and [5]. We take as an example the case of dynamic deployment for periodic continuous queries in mobile environments described in [6] and [7]. While system performance is the subject matter for periodic continuous queries, the queries themselves present a challenging case study in mobile systems. Broadcast delivery described in [8] has been proposed and proven to be an efficient way of disseminating data to the mobile client population. With broadcasting, the server can satisfy all pending requests on a data item simultaneously, thus, eliminating the potentially very large overload of data requests, and saving both the wireless bandwidth and a mobile client's battery energy. Another feature is that it greatly increases the scalability of the broadcast system by keeping the server from being swamped with data requests described in [9]. For periodic continuous queries in mobile environments, the client population is the critical factor of data delivery to system performance. To uplift periodic continuous queries, real-time data has become new trend in mobile databases. There are three key features such as frequent disconnection, limit of bandwidth allocation and user mobility to influence system performance in mobile environments. When mobile clients increase in a mobile cell, the bottleneck is becoming more urgent for mobile system. Prediction mechanism proved that applications enhance their ability efficiently according to experimental research. In our approach, we provide an efficient method based on Kalman filter theory described in [10] to decrease system overload so that real-time data resources satisfy sufficiently various requests of periodic continuous queries. From the observation of experimental results, our approach outperforms the traditional strategies of deploying periodic continuous queries in mobile environments. The rest of the paper is organized as follows. In section 2 we discuss related work for periodic continuous query strategies and

QSHINE 2015, August 19-20, Taipei, Taiwan

DOI 10.4108/eai.19-8-2015.2261092

mobile data dissemination. In section 3 we describe our model and propose the prediction approach. In section 4 we present the experimental results and simulation environment. Finally, we conclude the paper in section 5.

II. Preliminary

There are several scheduling algorithms for multichannel systems in mobile environments described in [4]. However, we demonstrate that traditional well-known algorithms do not always perform the best in a mobile environment, such as greedy and dynamic programming, when they are applied with time constraints over the on-demand wireless multichannel in a mobile environment. Scheduling of transactions for real-time databases in a non-mobile environment is studied extensively in [9].

A real-time client/server model is considered in which the server assigns priorities to transactions based on several strategies. We describe the features of traditional real-time algorithms as the followings:

- Earliest Deadline First(EDF)[11] Algorithm: as its name implies, for EDF the transaction with the earliest deadline is given the highest priority.
- Least Slack (LS) first[12] Algorithm: the slack time is defined as: $d-(t+E-P)$, where d is the deadline, t is the current time, E is the execution time and P is the processor time used thus far. If the slack time is ≥ 0, it means that the transaction can meet its deadline if it executes without interference. The slack time indicates how long a transaction can be delayed and still meet its deadline.
- Longest Wait First (LWF)[13] Algorithm: the sum of the total time that all pending requests for a data item have been waiting is calculated, and the data item with the largest total wait time is chosen to broadcast next.
- Requests Times Wait (RxW)[14] Algorithm: RxW makes scheduling decisions based on the current state of a queue (instead of access probabilities).

The Least Slack LS differs from EDF because the priority of a transaction depends on the service time it has received. If a transaction is restarted, its priority will change. Simulation results show that the EDF is the best overall policy for real-time database systems in a non-mobile environment. However, when system loads are high, the LS and EDF strategies lose their advantage, even over FCFS, as most transactions are likely to miss their deadlines. LWF algorithm has been shown to outperform all other strategies at minimizing wait time. In LWF, However, LWF has been recognized as expensive to implement described in [15]. The RxW algorithm provides an estimate of the LWF algorithm by multiplying the number of pending requests for a data item times the longest request wait time. In general, the performance of the approximate algorithms has been shown to be close to LWF.

There has been some research work to consider broadcasting for mobile real-time systems described in [16]. A push-based protocol for organizing broadcast disks for real-time applications, called Adaptive Information Dispersal Algorithm (AIDA) described in [17]. In this work, the data must be broadcast periodically to satisfy the timing constraints. The AIDA protocol considers fault-tolerance and the data items are allocated to the broadcast disks to minimize the impact of intermittent failures by utilizing redundancy. AIDA guarantees a lower bound on the probability of meeting timing constraints. Similar work addressing fault-tolerant real-time broadcast disks appears described in [4]. In this work, the authors show that designing strategies for real-time broadcast disks is related to pinwheel scheduling described in [18]. The authors derived a pinwheel algebra, which utilizes rules that can be used to construct fault-tolerant real-time broadcast disks. Their work differs from our work because we assume that we schedule all data items with time constraints using adaptive algorithms under limited bandwidth to minimize miss rate.

In the on-demand wireless multichannel broadcast model, the server periodically repeats a computed broadcast program, based on user access patterns. A broadcast cycle is defined as one transmission of the periodic broadcast program. Deadline constraints have been integrated into the broadcast model described in [16]. In order to minimize the total number of deadlines missed by making the most effective use of the available bandwidth, scheduling approach has to focus on critical factors such as access frequency, time constraint, and bandwidth requirements. In [19], scheduling mechanisms for broadcasting data that are to minimize the delay incurred by insufficient channels, but it is reasonable that all clients are satisfied with an expected time to optimize average access time.

III. Problem Formulation and Proposed Method

Multiple real-time data is defined as a real-time, continuous required (implicitly by arrival time or explicitly by time stamps) sequence of data items. A service center is a system especially constructed to process continuous queries on dynamic real-time data. Real-time database systems are different from traditional database management systems in that traditional database management systems expect the data to be required in the system and the queries to be dynamic whereas service centers expect dynamic real-time data and continuous queries. Due to the high volume of required data, it is often assumed that it is not possible to store a broadcast program in its storage, nor is it feasible to query the whole program history. Typically, the queries are executed on a group of data. A group data is a segment of broadcast program that is considered for the current query. Emerging applications, such as sensor networks, wireless traffic network, and financial stock market, have brought research related to real-time data in focus. These applications inherently generate real-time data and real-time database systems are well suited for such applications.

We now describe a framework to support the on-demand broadcast scheduling problem with time constraints. In this section, the real-time scheduling problem, system architecture and solving mechanism are introduced. We now describe a framework to support periodic continuous queries with moving ability over mobile environments. In this section, the prediction mechanism, problem formation and system architecture shown as figure 1 and solving mechanism are introduced. The generalization of the wireless networks combined with the increase of the extra-light devices transforms in-depth the data-processing

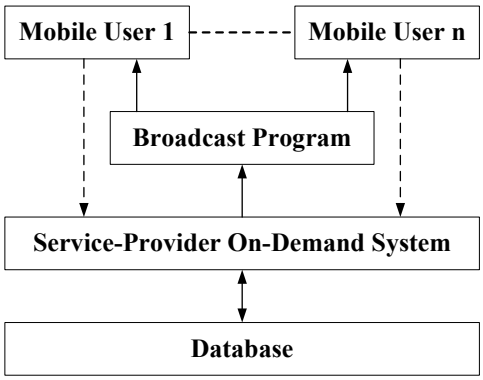

Fig. 1. Wireless Broadcasting System Based On On-demand Mode

applications, as well in their design as in their use. The hardware and software infra-structure is already available or about to be. First, the wireless Internet access is now operational, by the 802.11 standard (wireless local area network) or by cellular network. In addition, the extra-light devices increase in number and become connected to the network: tablets, smart phones, laptops, and the other portable devices, etc. All offer calculation capacities, storage or interaction in constant progress according to a lot of research. The mobile users can reach data and carry out treatments anywhere, any time and starting from any device. Our objective is thus to accompany this evolution in the information technology field by proposing techniques to optimize information accesses.

A. Prediction Mechanism Based on Kalman Filter

The Kalman filter is a very powerful tool when it comes to controlling prediction systems. It addresses the general problem of trying to estimate the state of a discrete-time controlled process that is governed by the linear stochastic difference equation. The five kernel definitions of time discrete Kalman filter equations and the predict-update equations are as below:

Definition 1. Kernel Equations of Kalman Filter

Predict Equations:

$$x_t = A_t x_{t-1} + B_t u_t \tag{1}$$
$$P_t = A_t P_{t-1} A_t^T + Q_t \tag{2}$$

Update Equations:

$$x_t = x_{t-1} + K_t \left(y_t - H_t x_{t-1}\right) \tag{3}$$
$$K_t = P_{t-1} H_t^T \left(H_t P_{t-1} H_t^T + R_t\right)^{-1} \tag{4}$$
$$P_t = P_{t-1} H_t^T \left(I - K_t H_t\right) P_{t-1} \tag{5}$$

where

- x: Estimated state.
- A: State transition matrix (i.e., transition between states).
- u: Control variables.
- B: Control matrix (i.e., mapping control to state variables).
- P: State variance matrix (i.e., estimated error).
- Q: Process variance matrix (i.e., error due to process).
- y: Measurement variables.
- H: Measurement matrix (i.e., mapping measurements onto state).
- K: Kalman gain.
- R: Measurement variance matrix (i.e., error from measurements).

Subscripts are as follows: t current time period, and $t-1$ previous time period. The random variable matrices Q and R represent the process and measurement noise (respectively). They are assumed to be independent (of each other), white, and with normal probability distributions. In practice, the process noise covariance Q and measurement noise covariance R matrices might change with each time step or measurement, however here we assume they are constant. The matrix A in the difference equation (1) and (2) relate the state at the previous time step $t-1$ to the state at the current step t, in the absence of either a driving function or process noise. Note that in practice A might change with each time step, but here we assume it is constant. The matrix B relates the optional control input u to the state x. The matrix H in the update equation (3), (4) and (5) relate the state to the update x_t. In practice H might change with each time step or measurement, but here we assume it is constant.

B. Problem Formation and Solution Scheme

The Kalman filter predicts state by assuming a predefined model of a system. Therefore, the Kalman filter model must be reasonable. Its operations should be defined as follows:

1) Analyze the problem: Look at the problem. Break it down to the mathematical steps.
2) Model the state process: Start with a basic process. It may not work effectively at the beginning, but this can be refined later.
3) Model the measurement process: Analyze how the approach to measure the process. The measurement space may not be in the same space as the state.
4) Model the noise: This needs to be done for both the prediction and update process. The base Kalman filter assumes Gaussian (white) noise, so make the variance and covariance (error) meaningful.
5) Test the model: The result trend shows the relationship between exact data and error distribution. If the filter needs to be corrected, the parameters will be adjusted.

We consider a simple situation showing a way to measure the number denoted as N of periodic continuous queries with portable devices in a wireless cell. This is shown in the figure 1. We are trying to estimate the number of mobile clients in the hot spot, which is unknown. The measurements obtained are from the known number. This could be an initial number, or an assumed number. The number could be:

- Maximizing, emptying or static (i.e., the number of mobile clients is increasing, decreasing or not changing).

- The relative measuring number to the average number of mobile clients is changing over time, or is static.

The first is the most basic model, the number is static (i.e., the number is constant $N = c$). Using the equations above kernel definition, the state variable can be reduced to a scalar (i.e., $x_t = x$ where x is the estimate of N). We are assuming a constant model, therefore $x_{t+1} = x_t$, so $A_t = 1$, for any $t \geq 0$. Control variables B and u are not used (i.e. $both = 0$). In our model, we have the known number. This is represented by $y = y$. The value we are measuring could be a scaled measurement. For simplicity, we will assume that the measurement is the exact same scale as our state estimate (i.e. $H = 1$). For this model, we are going to assume that there is noise from the measurement (i.e. $R = r$). The process is a scalar, therefore $P = p$. And as the process is not well defined, we will adjust the noise (i.e. $Q = q$). We will now demonstrate the effects of changing these noise parameters. The filter can be simplified as follows:

Definition 2. Our Approach Model

Predict Equations:

$$x_t = x_{t-1} \tag{6}$$
$$p_t = p_{t-1} + q_t \tag{7}$$

Update Equations:

$$x_t = x_{t-1} + K_t\left(y_t - x_{t-1}\right) \tag{8}$$
$$K_t = p_{t-1}(p_{t-1} + r)^{-1} \tag{9}$$
$$p_t = \left(1 - K_t\right) p_{t-1} \tag{10}$$

We formulate our problem to make it a resolvable problem as follows. Given a number of data items N to be broadcast in multiple wireless channels K. Each data item is associated with an access probability. Every access of a client is only one data item. Expected delay, w_i, is the expected number of ticks a client must wait for the broadcast of data item d_i. Average expected delay is the number of ticks a client must wait for an average request and is computed as the sum of all expected delays, multiplied by their access probabilities, where w_i is expected delay and p_i is access probability for data item d_i respectively. With miss rate of real-time data, a request for data item d_i has exceeded its maximum waiting when timing fault (expected delay for data item d_i exceeds its waiting time $t_i < W$) occurred at some time slot. The miss rate of all real-time data items is defined as follows (where i indicates identification number of data item):

$$MissRate = \sum_{j=1}^{K} \sum p_i \tag{11}$$

Our goal is to predict the number of mobile clients in the hot spot that minimizes the overload of system. We provide an algorithm referred as algorithm prediction of system overload described in [20] to predict the periodic continuous queries with portable devices so as to balance the system efficiency over mobile environments. If mobile client number minimized, let system efficiency gets optimization.

We formulate our problem and design our model to make it a resolvable example as follows. Given the three wireless broadcast channels, consider a set of 10 data items,

Algorithm 1 Predict-Update

{Predict-Update process estimates the state of system}
Input: The number of mobile clients at the previous time step;
Output: The number of mobile clients at the current time step;
Predict the current state based on the previous state;
Adjust state variance matrix by process variance matrix;
Update the current state using Kalman gain and measurement variables;
Adjust Kalman gain by measurement variance at the current time step;
Adjust state variance matrix by Kalman gain at the current time step;

$\{d_1, d_2, d_3, d_4, d_5, d_6, d_7, d_8, d_9, d_{10}\}$, with the following predicted states and measured states shown as table I. For the first prediction and measure, we assume the true level of the system state is $N = 1$. We initialize the state with an arbitrary number, with an extremely high variance as it is completely unknown: $x_0 = 0$ and $p_0 = 1000$. If you initialize with a more reasonable variable, you will get faster convergence. The system noise (i.e., system measurement error) we will choose will be $q = 0.0001$, as we think we have an accurate model. Let this process begin with the operations.

Predict Process:

$$x_0 = 0$$
$$p_0 = 1000 + 0.0001$$

The hypothetical measurement we get is $y_1 = 0.9$ (due to noise). We assume a measurement noise of $r = 0.1$.

Update Process:

$$k_1 = 1000.0001(1000.0001 + 0.1)^{-1} = 0.9999$$
$$x_1 = 0 + 0.9999\,(0.9 - 0) = 0.8999$$
$$p_1 = (1 - 0.9999)\,1000.0001 = 0.1000$$

To observe the step 1, the initialization of 0, has been brought close to the true value of the system. Also, the variance (i.e. error) has been brought down to a reasonable value. To continue the process at step 2:

Predict Process:

$$x_1 = 0.8999$$
$$p_1 = 0.1000 + 0.0001 = 0.1001$$

The hypothetical measurement we get this time is $y_2 = 0.8$ (due to noise).

Update Process:

$$k_2 = 0.1001(0.1001 + 0.1)^{-1} = 0.5002$$
$$x_2 = 0.8999 + 0.5002\,(0.8 - 0.8999) = 0.8499$$
$$p_2 = (1 - 0.5002)\,0.1001 = 0.0500$$

To observe the variance is reducing each time. If we continue (with hypothetical $y_t - values$) this we get the following results. On the table I, the first row represents predicted states and the second row represents measured states. To observe the serious outcome, our approach model shows the prediction accuracy.

TABLE I. PREDICTION AND MEASUREMENT OF EXAMPLE DATA

d_1	d_2	d_3	d_4	d_5
0.0000	0.8999	0.8499	0.9334	0.9501
0.8999	0.8499	0.9334	0.9501	0.9501
d_6	d_7	d_8	d_9	d_{10}
0.9501	0.9669	1.0006	0.9878	0.9722
0.9669	1.0006	0.9878	0.9722	0.9905

IV. SIMULATION AND EXPERIMENTAL RESULTS

To begin with, we randomly chose a scalar constant. We then simulated 50 distinct measurements that had error normally distributed around zero with a standard deviation of 0.001. We could have generated the individual measurements within the filter loop, but pre-generating the set of 50 measurements allowed me to run several simulations with the same exact measurements (i.e. same measurement noise) so that comparisons between simulations with different parameters would be more meaningful. During the simulation process, we fixed the measurement variance at R. Because this is the true measurement error variance, we would expect the best performance in terms of balancing responsiveness and estimate variance. This will become more evident in the following simulations. Figure 2 depicts the result distribution which represents the relationship of prediction and measurement. In figure 3, both prediction and measurement are close in some estimated values in term of accurate prediction from our model. In except of the initial state which is extreme assumption, the experimental result shows the better performance and verifies the fact that our approach model is an efficient prediction mechanism.

In our mobile simulation model, bandwidth is not explicitly modeled. Instead, similar to previous work described in [21], we use broadcast ticks as a measure of time. The greatest advantage of this approach is that the results are not limited to any particular bandwidth and/or data item size. Rather, it aims to capture the fundamental characteristics of the systems described in [22]. The model simulates one hop wireless network shown as [23]. All data items are stored in a data server in a fixed location. Mobile clients need to send requests to the server via an uplink back-channel before the requested page can be broadcast in [24]. The arrival of requests generated by mobile clients follows a $Poisson$ process and the inter-arrival time is exponential with mean λ. Each request has a request id, and arrival time. For each page, a queue is maintained to store the information about requests on the object. We assume the results produced after the maximum waiting time are useless, so all requests that have missed their waiting time are discarded. Mobile clients are responsible for re-sending requests when link errors occur. We also assume a time fault can capture the mobility of clients who are no longer able to receive the broadcast. In our model, since newly generated data requests are sent to the server immediately, the request generating time is equal to the time the server receives it (assuming network delay is ignored). We also ignore the overloads of request processing at the server, because the main purpose of the model is to support the mobile environments.

We assume requests generated by mobile clients are read only, and no update request is allowed. Concurrency control issues are not our main concern, and thus, not considered. At each tick of the simulation clock, the following occurs. A simulated request generator generates requests with exponential inter-arrival time. The information about each request id and arrival time is recorded. The request is then inserted to the corresponding queue. The server checks the deadlines of all the arrived requests, and discards those requests that have missed their time fault. Then the server selects a page to broadcast by applying a scheduling strategy and starts to broadcast the selected page. All requests requesting the page are satisfied when the broadcast is finished. A client can request multiple pages and a page can be requested by multiple mobile clients at a time. We assume that data demand probabilities p_i follow the $Zipf$ distribution described in [25], where p_i represents the $i'th$ most popular page. The $Zipf$ distribution allows the pages requested to be skewed.

$$p_i = \frac{\left(1/i\right)^{\theta}}{\sum_{i=1}^{M}\left(1/i\right)^{\theta}}, (i = 1, 2, 3, ..., M) \tag{12}$$

We only choose the online heuristic algorithms to simulate the environments since we believe these online heuristic algorithms better adapt to the dynamic changes of the intensity and distribution of system workloads. The push-based (access probabilities, broadcast histories, etc.) off-line algorithms are not considered due to the fact that they are mainly for fairly stable systems. We implement the simulation model described in the previous section using visual $C++$. In each experiment, we run the simulation for 1000 time units, and we use an average of 50 runs of each simulation as the final result. The default total number of data pages stored in the server, referred to as DBSIZE, is 100 pages. Client requests reach the system with exponential inter-arrival time with mean λ, and λ is varied in our simulation from 2-50. It is assumed that each data request requires 1 broadcast tick to broadcast. An open system model is used to simulate the system for extremely large, highly dynamic populations. Data access follows a skewed $Zipf$ distribution with parameter θ to control the skewness. The minimum slack time is 10, with the maximum slack time ranging from 20 to 100. This variation in maximum slack time allows us to vary the tightness in the deadlines. In addition to a uniform distribution of time fault, an exponential distribution is utilized with lambda ranging from 10 to 300.

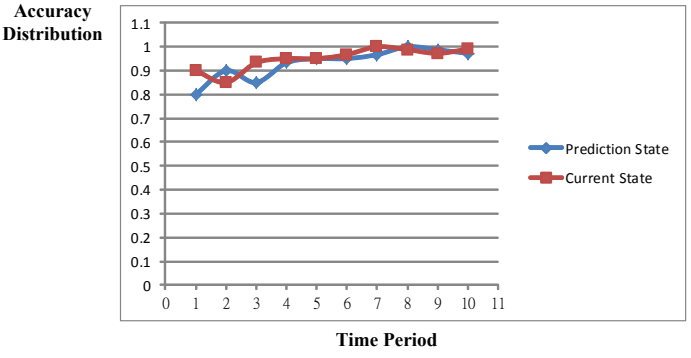

Fig. 2. Accuracy Distribution between Prediction and Measurement States

V. CONCLUSION

In this paper, we present a prediction model based on Kalman filter for scheduling real-time data to balance system

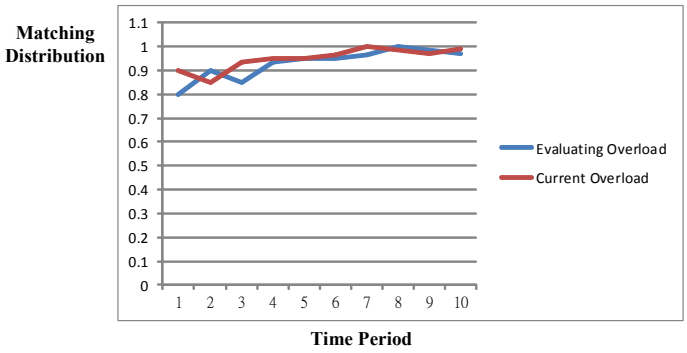

Fig. 3. Matching Distribution between Prediction and Measurement on Correction-Update

overload over on-demand wireless broadcasting environments. As this paper demonstrates, the traditional well-proven strategies do not balance efficiently system overload in a mobile environment. We propose an efficient prediction mechanism based on online heuristic algorithm, which is designed for timely balancing mobile clients in mobile environments to minimize deadline miss rate and reduce system overload. We simulate our model over on-demand wireless broadcasting environments based on heuristic scheduling and adjustment. The proposed approach is shown to generally outperform the existing strategies with different accuracy distributions. In the future, we plan to adjust our system parameters by reducing its time complexity. Other concentrations include estimated state of variance variables and matrices that can handle measurement errors, update process, control variables and matrices.

ACKNOWLEDGMENT

We are grateful to the support of Ministry of Science and Technology, Taiwan, R.O.C. under contract number MOST 103-2221-E-149-008.

REFERENCES

[1] S. Babu and J. Widom, "Continuous queries over data streams," *SIGMOD Record*, vol. 30, no. 3, pp. 109–120, September 2001.

[2] D. Carney, U. etintemel, M. Cherniack, and C. Convey, "Monitoring streams a new class of data management applications," in *Proceedings of the 28th VLDB Conference*, Hong Kong, China, August 2002, pp. 215–226.

[3] H. Wang, Y. Xiao, and L. Shu, "Scheduling periodic continuous queries in real-time data broadcast environments," *IEEE TRANSACTIONS ON COMPUTERS*, vol. 61, no. 9, pp. 1325–1340, SEPTEMBER 2012.

[4] F. Adelstein, S. K. Gupta, G. G. R. III, and L. Schwiebert, *Fundamentals of Mobile and Pervasive Computing*. McGraw-Hill, 2005.

[5] J. Zhifeng and L. V. C. M., "End-to-end quality of service provisioning for nternet access via third generation wireless networks," *Journal of Internet Technology*, vol. 6, no. 4, pp. 367–374, October 2005, object-Oriented Technology and Applications.

[6] T. Aktas, A. O. zgur Ylmaz, and E. Aktas, "Practical methods for wireless network coding with multiple unicast transmissions," *IEEE TRANSACTIONS ON COMMUNICATIONS*, vol. 61, no. 3, pp. 1123–1133, March 2013.

[7] Y. Wei, V. Prasad, S. H. Son, and J. A. Stankovic, "Prediction-based qos management for real-time data streams," in *Proceedings of the 27th IEEE International Real-Time Systems Symposium (RTSS'06)*, Rio de Janeiro, Brazil, December 2006, pp. 344 – 358.

[8] S. Acharya, R. Alonso, M. Franklin, and S. Zdonik., "Broadcast disks: data management for asymmetric communication environments." in *Proceeding of ACM SIGMOD*, March 1995, pp. 199–210.

[9] S. Bodas, S. Shakkottai, L. Ying, and R. Srikant, "Scheduling in multi-channel wireless networks: Rate function optimality in the small-buffer regime," *IEEE TRANSACTIONS ON INFORMATION THEORY*, vol. 60, no. 2, pp. 1101–1124, February 2014.

[10] R. E. KALMAN, "A new approach to linear filtering and prediction problems," *Transactions of the ASMEJournal of Basic Engineering*, vol. D, no. 82, pp. 35–45, 1960.

[11] C. Liu and J. Layland., "Scheduling algorithms for multiprogramming in hard real-time traffic environments." *Journal of the Association for Computing Machinery*, vol. 20, no. 1, pp. 179–194, 1973.

[12] R. M. Sivasankaran, J. A. Stankovic, D. Towsley, B. Purimetla, and K. Ramamritham, "Priority assignment in real-time active databases," *The International Journal on Very Large Data Bases*, vol. 5, no. 1, pp. 019–034, January 1996.

[13] J. W. Wong, "Broadcast delivery," *Proceedings of the IEEE*, vol. 76, no. 12, pp. 1566–1577, Dec. 1988.

[14] D. Aksoy and M. Franklin., "Rxw: a scheduling approach for large-scale on-demand data broadcast," *IEEE/ACM Transactions on Networking*, vol. 7, no. 6, pp. 846–860, 1999.

[15] A. Boukerche and H. O. II, "Media synchronization and qos packet scheduling algorithms for wireless systems," *ACM Mobile Networks and Applications*, vol. 10, no. 1-2, pp. 233–249, February 2005.

[16] D.-J. Chiang, T. K. Shih, and C.-L. Chen, "Disseminating data with time constraint based on multichannel over ubiquitous computing environments," *WORLD WIDE WEB-INTERNET AND WEB INFORMATION SYSTEMS*, vol. 14, pp. 223–241, 2011.

[17] A. Bestavros., "Aida-based real-time fault-tolerant broadcast disks." in *Proceedings of Real-Time Technology and Applications Symposium.*, 1996, pp. 49–58.

[18] S. Baruah and A. Bestavros., "Pinwheel scheduling for fault-tolerant broadcast disks in real-time database systems." in *Proceedings of the 13th International Conference on Data Engineering.*, April 1997, pp. 543–551.

[19] Y. Dhungana and C. Tellambura, "Uniform approximations for wireless performance in fading channels," *IEEE TRANSACTIONS ON COMMUNICATIONS*, vol. 61, no. 11, pp. 4768–4779, November 2013.

[20] T. H. Cormen, C. E. Leiserson, and R. L. Rivest., *Introduction to Algorithms*. The MIT, 1992.

[21] G. K. Karagiannidis, "Performance bounds of multihop wireless communications with blind relays over generalized fading channels," *IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS*, vol. 5, no. 3, pp. 498–503, March 2006.

[22] K. Kathiravan, V. Divya, and S. T. Selvi, "A hybrid probabilistic counter-based broadcast approach protocol for mobile ad hoc networks," *International Journal of Ad Hoc and Ubiquitous Computing*, vol. 4, no. 2, pp. 108–114, 2009.

[23] A. Keshavarz-Haddad and R. H. Riedi, "Bounds on the benefit of network coding for wireless multicast and unicast," *IEEE TRANSACTIONS ON MOBILE COMPUTING*, vol. 13, no. 1, pp. 102–115, January 2014.

[24] H. Y. Lateef, D. McLernon, and M. Ghogho, "Performance analysis of multi-user, multi-hop cooperative relay networks over nakagami-m fading channels," *IEEE COMMUNICATIONS LETTERS*, vol. 15, no. 7, pp. 776–778, July 2011.

[25] G.K.Zipf., *Human Behaviour and the Principle of the Least Effort.* Addison-Wesley, 1949.