

Game Theoretic Power Aware Wireless Data Access

Mark Kai Ho Yeung and Yu-Kwong Kwok

Department of Electrical and Electronic Engineering
The University of Hong Kong, Pokfulam, Hong Kong
Email: {khyeung,ykwok}@eee.hku.hk

Abstract—This paper examines the following wireless data access scenario: a number of clients are interested in a set of data items kept at the server. A client sends a query request to inform the server of its desired data item. The server replies in the common broadcast channel. We first define a utility function that considers client’s power consumption in transmit, receive and idle modes. Specifically, utility is expressed as the number of queries that can be completed given a fixed energy source. Based on the utility function, we formulate our power aware wireless data access scheme as a non-cooperative game, called the WDA game. From our theoretical analysis, we show that clients are *not always* necessary to send query requests to the server. Instead, each client determines the request probability without *any explicit* communication with one another. Furthermore, we design and evaluate the server and client algorithms for the WDA game. Simulation results confirm that our proposed scheme, comparing with a simple always-request one, increases the utility and lifetime of *every client* while reducing the number of requests sent, at the cost of a slightly larger average query delay.

Index Terms—Wireless Data Access, Game theory, Economics, Simulations.

I. INTRODUCTION

With the popularity of personal area networks (PANs), such as 802.11x and Bluetooth, there has been increasingly large amount of information being delivered over the wireless medium. Furthermore, the world-wide deployment of third generation (3G) cellular systems is going to complement the coverage limitations in PANs. 3G, together with PANs, makes ubiquitous information access a reality [13]. Recently, network operators have started offering real-time video services in 3G [9]. In addition to voice and video traffic sources, it is expected to have more and more data applications being deployed in the near future. One fundamental support to many interesting applications is to provide efficient *on-demand information access*, which enables mobile clients to selectively “pull” the desired data from the server.

On-demand information access has recently attracted much attention. In particular, researchers have proposed different techniques to conserve clients’ energy. For example, the wireless broadcast channel has been widely used in data dissemination [1], [5], [14], [17]. The amount of broadcast traffic does not depend on client population size. Datta *et al.* [7] presented a comprehensive energy consumption model to study a set of

Corresponding Author: Yu-Kwong Kwok. This research was supported by a grant from the Research Grants Council of the HKSAR under project number HKU 7157/04E.

broadcast protocols for data retrieval. We have also done some work in this area [14], [15]. However, most previous work focus on the design of different cache invalidation schemes. To maintain cache consistency, each client is required to retrieve periodic invalidation information indefinitely. This may not be desirable for clients with low request rates or data items with frequent updates.

To avoid the above drawback, we utilize the broadcast channel from another perspective. Specifically, we consider the following scenario: a number of clients are interested in a set of data items kept at the server. Each client sends a request to inform the server of its desired data item. The server replies in the common broadcast channel periodically. There is no need for clients to maintain consistency. To quantify power consumption in the query process, we consider that clients dissipate energy in transmit, receive and idle modes. We formulate the distributed client power conservation problem with game theory, which has become an increasingly popular tool in modeling different networking problems [4].

The remainder of the paper is organized as follows. Section II describes the details of system model; energy consumption model; and the formal problem statement. Furthermore, we analyze the performance of our proposed power aware data access scheme. Simulation results are presented and discussed in Section III. Finally, we draw some concluding remarks in Section IV.

II. PROPOSED POWER AWARE WIRELESS DATA ACCESS GAME

A. Problem Formulation

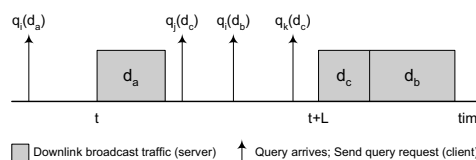


Fig. 1. System model for wireless data access.

Figure 1 depicts the system model for wireless data access. It consists of a server and a set of clients, N . The clients are interested in a common set of data items, D , which are kept at the server. To request a specific data item, d_a , client i is required to inform the server by sending an uplink request, represented by $q_i(d_a)$. The server replies with the content of the requested

data item, d_a , in the common broadcast channel. This allows the broadcast data item to be shared among different clients. As illustrated in Figure 1, both clients j and k request the same data item, d_c , in the second interval. However, server is required to broadcast the content of d_c only once in the next broadcast period. In this paper, we investigate the energy consumption characteristics of the above data access model. Specifically, we are interested in conserving the energy consumed among the clients.

To successfully complete a query, a client expends its energy in two different parts: (1) informing the server of the desired data item, E_{UL} ; and (2) downloading the content of the data item from the common broadcast channel, E_{DL} . The energy cost of sending a request to the server is represented by E_s . Note that E_{UL} does not necessarily equal E_s (see Section II-A). It is assumed that E_{DL} is proportional to the size of a data item. In practice, E_s is a function of various quantities [12], including spatial separation, speed, instantaneous channel quality, bit-error rate requirement, etc. For simplicity, however, E_s is also assumed to be a fixed quantity, except in Section III-B.5.

Define E_{total} as the amount of energy available to a client. We use the number of queries that can be completed to quantify the performance of a wireless data access scheme. Mathematically, client i 's utility is given by,

$$U_i = \frac{E_{total}}{E_{UL}^i + E_{DL}^i} \quad (1)$$

The objective is to reduce the amount of energy consumed in the query process such that every client's utility (Equation (1)) is increased, but with bounded query delay.

In a simple data access scheme, whenever a new query arrives, the client is required to send an uplink request to inform the server, i.e., $E_{UL} = E_s$. Thus, we have,

$$U_i = \frac{E_{total}}{E_s + E_{DL}} \quad (2)$$

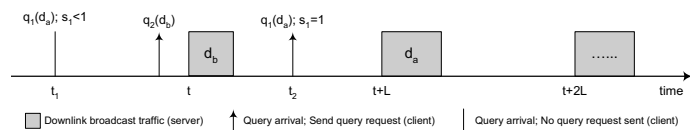


Fig. 2. Client 1 chooses not to request at t_1 but cannot find d_a in the 1st server broadcast. Then, it requests with probability 1 at t_2 .

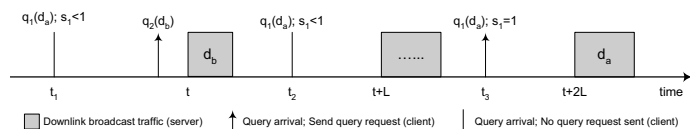


Fig. 3. Client 1 chooses not to request at both t_1 and t_2 but it cannot find d_a in the 1st and 2nd server broadcast. Then, it requests with probability 1 at t_3 .

Due to data locality, some data items are more popular than the others [3]. It is possible that more than one client independently requests for the same “hot” data item. However, these duplicated requests are a wasteful of (1) battery energy; and (2) uplink bandwidth. In fact, these requesting clients, but not all,

are more advantageous to “back-off” and let some others send the request to the server on behalf of themselves. Ideally, only one request is needed to trigger the server to broadcast the data item. However, this would require *explicit* coordination among clients. Such extra communication overheads may not justify the energy conserved from uplink requests. More importantly, the client, who actually sends the request, expends its own battery energy and possibly bears monetary cost for the goodness of others. Without appropriate incentives, there is obviously *no* client willing to take up the requesting role.

To analyze the above conflicting situation, we model the data access problem as a static non-cooperative game—**WDA game**—each *player* (client) maximizes its own *utility* (Equation (1)) with *no* explicit communication with one another.

Denote $N = \{1, \dots, n\}$ as the set of players (clients). Each player determines its *request probability*—probability of sending an uplink request to the server. The strategy space of player i is given by, $S_i = \{s_i | 0 \leq s_i \leq 1\} \subset \mathbb{R}^1$. The strategy combination is denoted as, $s = (s_1, \dots, s_n) \in S$, where $S = \times_{j \in N} S_j \subset \mathbb{R}^n$ is the Cartesian product of the n players' strategy spaces. Furthermore, define $s_{-i} = (s_1, \dots, s_{i-1}, s_{i+1}, \dots, s_n) \in S_{-i}$, where $S_{-i} = \times_{j \in N \setminus \{i\}} S_j \subset \mathbb{R}^{n-1}$, as the strategy combination of all the players, except i . $U_i(s) \in \mathbb{R}^1$ represents the utility of player i when the strategy combination is s . A strategy combination, s^* is said to achieve *Nash equilibrium* when:

$$U_i(s^*) \geq U_i(s_{-i}^*, s_i) \quad \forall s_i \in S_i, i \in N \quad (3)$$

Notice that the utility function, U_i , depends on each player's strategy, s_i , which in turn affects the strategies of all the other players, s_{-i} . In other words, a strategy combination is said to achieve the state of *Nash equilibrium* when *no* player can improve its utility by *unilaterally* deviating from its own strategy. A salient feature is that there is no coordination among the set of players. In general, a game may have multiple equilibria or even none at all.

B. WDA Game—2-person version

To study the WDA game, we start with analyzing the 2-person version, i.e., $N = \{1, 2\}$. There are two clients in the system, each of which determines its request probability, s_i , independently. Consider that case that client 1 chooses *not* to request. If client 2 sends a request for the same data item, the server will make the data item available in the next broadcast period. As such, the uplink cost, E_{UL} , for client 1 is zero. Otherwise, the desired data will not appear in the immediate server's broadcast. If client 1 is not patient enough to wait for another period, its strategy is to send the request with probability 1, as illustrated in Figure 2. The uplink cost for clients 1 and 2 are, respectively,

$$E_{UL}^1 = s_1 E_s + (1 - s_1)(1 - s_2)(E_s + E_w) \quad (4)$$

$$E_{UL}^2 = s_2 E_s + (1 - s_1)(1 - s_2)(E_s + E_w) \quad (5)$$

where E_w is defined as the amount of energy required to wait for a broadcast period, i.e., E_w represents the “cost of waiting”. It is observed, from Equations (4) and (5), that the utility

functions are symmetrical. This motivates us to search for symmetric equilibrium strategies.

Let $E_w = \alpha E_s$ and differentiate U_1 with respect to s_1 gives,

$$\frac{\partial U_1}{\partial s_1} = -\frac{E_{total}E_s}{E_{UL}^1 + E_{DL}} \{s_2(1 + \alpha) - \alpha\} \quad (6)$$

Depending on the values of s_2 , $\frac{\partial U_1}{\partial s_1}$ takes on different values:

- 1) $s_2 < \frac{\alpha}{1+\alpha} \Rightarrow \frac{\partial U_1}{\partial s_1} > 0$
Client 1's best-reply strategy is, $s_1^* = 1$, which reduces to the original simple access scheme.
- 2) $s_2 > \frac{\alpha}{1+\alpha} \Rightarrow \frac{\partial U_1}{\partial s_1} < 0$
Client 1's best-reply strategy is, $s_1^* = 0$, i.e., client 2's request probability is large such that client 1 is always advantageous to wait for a broadcast period.
- 3) $s_2 = \frac{\alpha}{1+\alpha} \Rightarrow \frac{\partial U_1}{\partial s_1} = 0$
The best-reply for client 1 is any feasible strategy, i.e., player 1 is indifferent between request and wait. In particular, the strategy combination, $(s_1^*, s_2^*) = (\frac{\alpha}{1+\alpha}, \frac{\alpha}{1+\alpha})$, achieves a *weak* Nash equilibrium.

To avoid the weak equilibrium and the associated degenerated solutions, consider the case that client 1 is patient enough to wait for one extra broadcast period. In other words, client 1 waits for a maximum of two server broadcasts before forcing to request with probability one (see Figure 3). Thus, the uplink energy cost becomes:

$$E_{UL} = E_s \{s_1 + (1 - s_1)(1 - s_2)(s_1 + \alpha) + (1 - s_1)^2(1 - s_2)^2(1 + 2\alpha)\} \quad (7)$$

The best-reply strategy for client 1 is shown to be,

$$s_1 = \frac{2(1 + 2\alpha)(1 - s_2)^2 - (1 - \alpha)(1 - s_2) - 1}{2(1 + 2\alpha)(1 - s_2)^2 - 2(1 - s_2)} \quad (8)$$

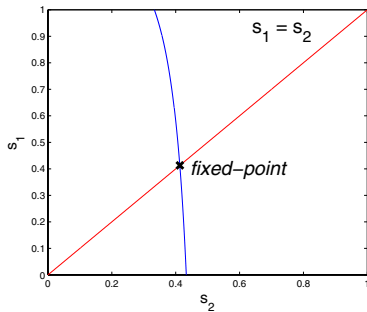


Fig. 4. s_1 versus s_2 .

Figure 4 shows a plot of Equation (8) when $\alpha = 0.5$. It is observed that there is only one *fixed-point* solution within the feasible strategy space, i.e., $s_1^* = s_2^* = 0.4131$, which is the unique equilibrium strategy of the game.

C. WDA Game— n -person version

Based on the analysis in Section II-B, the 2-person WDA game is transformed to the n -person version as follows:

If client i has a query pending for the data item, d_x , the equilibrium strategy, s_i^* , is used. Otherwise, $s_i = 0$ for that data item, since client i is not interested in d_x . Comparing with Equation (7), client i 's uplink energy cost for d_x is: $E_{UL}^i = E_s \{s_i + (s_i + \alpha) \prod_{j \in M} (1 - s_j) + (1 + 2\alpha) \prod_{j \in M} (1 - s_j)^2\}$, where $M \subseteq N$, is the set of clients interested in d_x . Thus, the symmetric equilibrium strategy is given by,

$$s^* = \frac{2(1 + 2\alpha)\chi^2 - (1 - \alpha)\chi - 1}{2(1 + 2\alpha)\chi^2 - 2\chi} \quad (9)$$

where $\chi = (1 - s^*)^{m-1}$ with $m = |M|$.

Theorem 1: There exists an equilibrium strategy for the n -person WDA game.

Proof: First, each player's strategy space, $S_i \in \mathfrak{R}^1$, is non-empty, convex and compact. Second, the utility function, U_i , are continuous on S , $\forall i \in N$. Furthermore, the best-reply mapping is single-valued. By Brouwer's fixed point theorem [2], there exists a fixed point solution in the best-reply mapping. Thus, we can conclude that there exists an equilibrium strategy for the n -person WDA game (Section 2.2 pp. 14–15, [11]).

Although Theorem 1 does not rule out the possibility of more than one equilibrium, we observe, from simulations, that the equilibrium strategy (Equation (9)) appears to be unique.

D. The Protocol

To achieve the stated equilibrium strategy, we design algorithms for server and clients, which are formalized as follows:

Each client executes Algorithm 1 to determine the equilibrium strategy. Clients do not solve Equation (8) every time. Instead, the solutions are calculated and stored for table lookup. Each client determines its request probability independently based on the m value announced by the server (see below). It should be emphasized that there is no coordination among clients. If there is more than one query pending, the client plays with the joint request probability.

Using Algorithm 2, the server keeps track of the set of m and announces the values to each client via periodic broadcast. For $m = 1$, the request probability is 0.4132 (see Section II-B). If the server intentionally announces $m = 0.4668$, each client would play 0.6567, which corresponds to the optimal strategy. Thus, clients play with the optimal request probability as if it is an equilibrium strategy. This requires the server to perform the actual-to-optimal m mapping. To improve efficiency, the mapping is also done via table lookup. It is assumed that the server knows (1) the mean query generation time of a typical client, e.g., from previous usage statistics; and (2) the number of alive clients, e.g., from the network layer, from which the values of m can be estimated.

Server runs Algorithm 3 to reply clients' queries. In addition to the data content, the server also broadcasts an index list, similar to the one used in [5]. Using the index list, clients can tune to their desired data item(s) and ignore the others. This further conserves clients' energy in monitoring the broadcast channel, which is also costly [8].

Algorithm 1 Client—Query

```
1: for each new query,  $q_i(d_j)$  do
2:   Use  $m_j$  to determine  $s^*$  from Equation (9);
3:    $s = 1 - (1 - s)(1 - s^*)$ ;
4:    $Q = Q \cup d_j$ ;
5:   Send( $s$ );
6: end for
7: for each server broadcast period,  $L$  do
8:   if  $Q \neq \emptyset$  then
9:     Check the index list,  $I$ ;
10:    Download the desired data item(s);
11:   end if
12:   if  $Q \neq \emptyset$  then
13:     if a query has missed two server broadcast then
14:        $s = 1$ ;
15:     end if
16:     Send( $s$ );
17:   end if
18: end for
19: Function Send( $s$ )
20: if  $\text{rand} < s$  then
21:   Send a request;
22:    $Q = \emptyset$ ;  $s = 0$ ;
23: end if
```

Algorithm 2 Server— m -values

```
1: for  $j = 1$  to  $|D|$  do
2:    $m'_j = p_j(\theta) \sum_{i \in N'} \frac{L}{T_q^i}$ ;
3:    $m'_j \rightarrow m_j$ , using actual-to-optimal  $m$  table;
4: end for
```

III. PERFORMANCE EVALUATION

We evaluate our proposed game theoretic power aware wireless data access scheme using MATLAB simulation.

A. Simulation Model and Parameters

Table I shows the values of major simulation parameters. Most of the parameter values are the same as in [5], [17]. Each client generates a stream of exponentially distributed queries with mean query arrival time, T_q , drawn uniformly between T_q^{\min} and T_q^{\max} , which models different levels of interest on

¹For illustration purposes, the energy available, E_{total} , is set to 100J. In fact, different values would only change the scale of the performance graphs (see also Section III-B.6).

Algorithm 3 Server—Reply

```
1: for each received query  $q_i(d_x)$  do
2:    $I = I \cup d_x$ ;
3: end for
4: for each server broadcast period,  $L$  do
5:   Broadcast  $I$ ;
6:   Broadcast the content of data items in  $I$ ;
7: end for
```

TABLE I
SIMULATION PARAMETERS.

Parameter	Value
Database size ($ D $)	1,000
Data item size	1KB–10KB
Broadcast interval (L)	20s
Client population size ($ N $)	200
Transmit power	0.5W
Receive power	0.1W
Relative cost of waiting (α)	0.5
Bandwidth	144Kbps
Energy available ¹ (E_{total})	100J
Minimum query arrival rate (T_q^{\min})	5s
Maximum query arrival rate (T_q^{\max})	50s
Uplink request size	1KB

the database among the set of clients. We use the *Zipf-like* distribution [8], with $\theta = 0.9$, to model the non-uniform access pattern. Each client queries the j^{th} data item with probability given by: $p_j(\theta) = 1/\{j^\theta \sum_{k=1}^{|D|} \frac{1}{k^\theta}\}$, where $0 \leq \theta \leq 1$. The value of θ determines the “skewness” of the access pattern. $\theta = 1$ gives the strict *Zipf* distribution while $\theta = 0$ results in the uniform distribution. To focus on the data access process, the effect of client caching is not considered. Thus, clients do *not* need to expend energy to obtain and process any consistency information. The server replies clients’ queries every L seconds and announces m -values whenever the set of *alive* client changes. It is assumed that both the uplink and downlink bandwidth are used exclusively for data access.

B. Simulation Results

We compare the performance of the WDA game with that of a simple access scheme in terms of (1) *utility*; (2) *lifetime*; (3) *amount of uplink traffic*; and (4) *average query delay*. We also study the effect of client population size, $|N|$, on the above four metrics.

1) *Utility (number of completed queries)*: Figure 5(a) shows the number of queries completed (*utility*) versus the i^{th} client. Notice that client indices are in ascending order of their mean query arrival time, i.e., the 1st client has the highest query rate. First, we observe that *every client’s* utility is increased in the WDA game compared to the simple access scheme. This is because each client individually and independently optimizes its utility function (Equation (1)) to determine its equilibrium request probability. If a client chooses not to request, the client expends energy to wait for server’s reply. Although the reply may not appear, the analysis shows that the decision would be advantageous in terms of expectation. This is confirmed by the simulation results. Second, clients with higher utility values show larger improvements, since they benefit more due to their frequent queries. On the other hand, low-utility clients dissipate the majority of their energy in idle state and do not show significant improvement in utility.

2) *Lifetime*: The lifetime of *each client* is shown in Figure 5(b). The lifetime of a client is defined as the time taken to

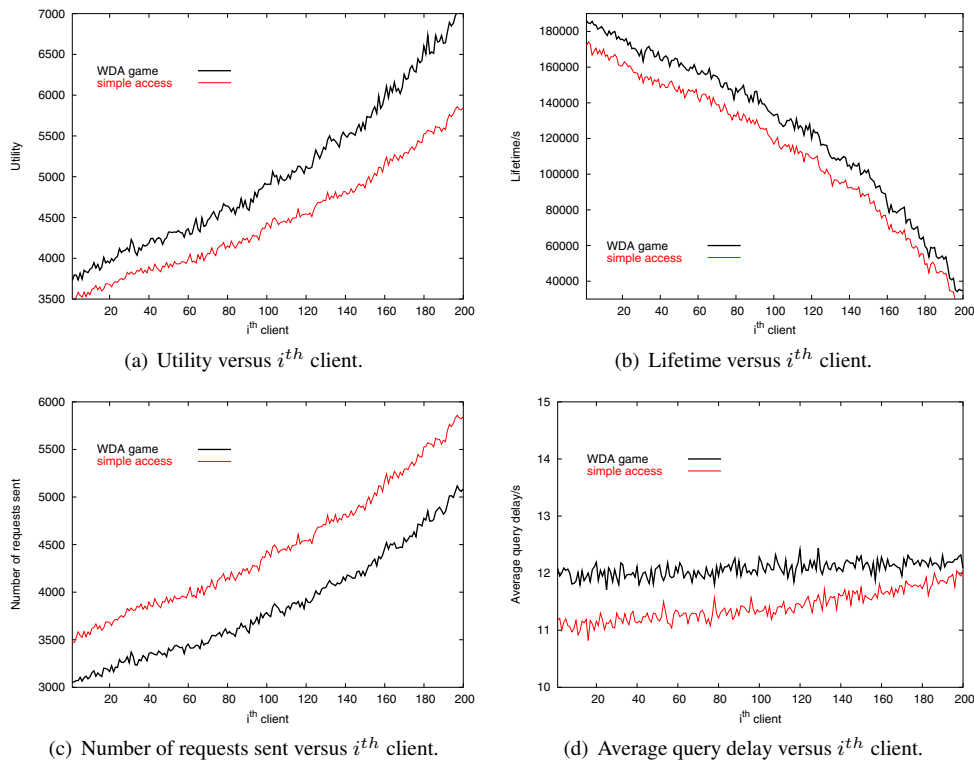


Fig. 5. Performance of the WDA game in different metrics.

deplete its available energy, E_{total} . First, the lifetime of *every client* is increased in the WDA game compared to the simple access scheme. Second, we observe that high-utility clients have relatively shorter lifetimes. This is because of their high query arrival rates: energy dissipates more rapidly in sending requests to the server. As such, they deplete their energy source earlier. Third, low-utility clients achieve larger improvements in lifetime compared to the high-utility ones. It is because the energy conserved from uplink requests has a more significant effect in the low-utility group, which is mainly in the idle state with low power consumption.

3) *Uplink Traffic (number of uplink requests sent)*: Figure 5(c) shows the number of requests sent for the i^{th} client. We observe a similar result: *each client* sends fewer requests to the server. This provides an explanation for the increase in lifetime of *every client*. Unlike the previous two metrics, each client shows a very similar reduction in the number of requests sent. This is due to the common access pattern, i.e., the *Zipf-like* distribution, used in the simulation model. The result also suggests that similar amount of energy is conserved among the clients. As such, all clients benefit from playing the game. Besides the energy conservation effect, fewer requests sent also reduces: (1) uplink bandwidth requirement; (2) collision probability; and (3) interference to other users.

4) *Query Delay*: The performance improvements come with a cost: increase in average query delay, which is defined as the time between arrival of a query and complete reception of the reply from the server. Figure 5(d) shows the clients' average query delay in the WDA game and a simple access scheme. As expected, clients playing the WDA game experience slightly longer average query delay. This is inherently in the design of

the game: wait for some one to make the request. However, the delay is bounded by the additional requirement: if the desired data item does not appear in the immediately *two* server's broadcast, the client is forced to send a request with probability one. Indeed, if a client can tolerate a larger worst-case delay or energy conservation becomes very significant (e.g., in low-energy state), it is possible to revise the requirement—say, check *three* server's broadcast before forcing to request.

5) *Effect of Channel Fluctuations*: We have assumed that the energy cost of sending a query request, E_s , is fixed. In this section, we would like to investigate the effect of channel fluctuations, i.e., variable E_s , on various performance metrics.

We adopted a simple two-state first-order Markov chain described in [6] to model the channel fluctuations. Specifically, the channel state is either *good* or *bad*. The transition probability matrix, M_c , is given by [6]: $M_c = \begin{bmatrix} p & 1-p \\ 1-q & q \end{bmatrix}$, where p and $1-q$ are the probabilities that the channel state is good in the current timeslot, given that the channel state in the previous timeslot was good or bad, respectively. Both p and q depend on (1) fading margin, F ; (2) timeslot duration, T ; and (3) Doppler bandwidth $f_D = V/\lambda$, where V is the maximum mobile speed and λ is the carrier wavelength.

Consider a network model [14], [15] with carrier frequency, $f_c = 1800\text{MHz}$; $V = 50\text{km/hr}$; and $T = 2.5\text{ms}$, the transition probabilities are given by, $(p, q) = (0.9100, 0.1446)$ for $F = 10\text{dB}$; $(p, q) = (0.7665, 0.3722)$ for $F = 5\text{dB}$. The transmit power is set to 0.5W and 1.0W when the channel is good or bad, respectively.

Figure 6 shows the performance of the WDA game in different channel conditions. $F = \infty$ corresponds to the results

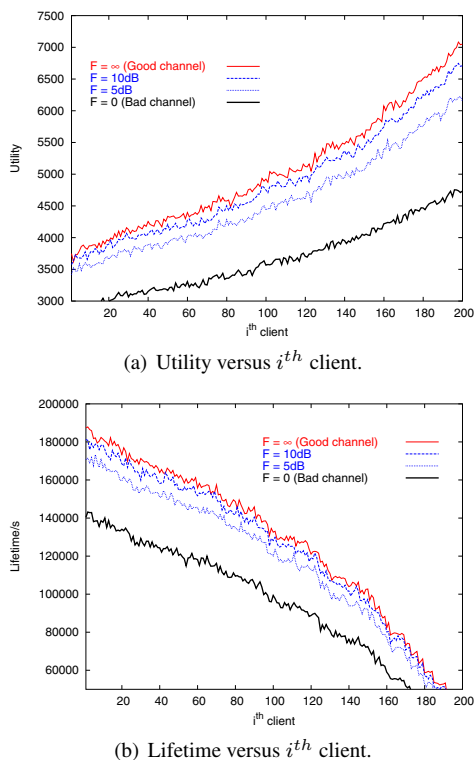


Fig. 6. Effect of channel fluctuations on the WDA game ($F = \infty, 10\text{dB}, 5\text{dB}, 0$ (from top to bottom)).

in previous sections, i.e. the channel is always good. On the other hand, $F = 0$ represents the worst-case scenario, i.e., the channel is always bad.

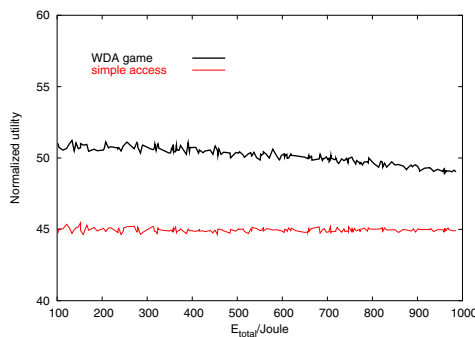


Fig. 7. Effect of initial energy, E_{total} .

6) *Effect of Initial Energy Distribution:* Figure 7 shows the effect of initial energy distribution in terms of normalized utility. There are 200 clients with $T_q = 25\text{s}$ and E_{total} evenly distributed between 100J and 1,000J. The performance is measured in terms of normalized utility, which is defined as utility per Joule of energy. From the figure, it is observed that clients with a smaller E_{total} achieved a slightly larger normalized utility. This suggests that the WDA game is more efficient for low-energy clients, which is a desirable property. Similar results were observed in terms of lifetime (not shown [16]).

IV. CONCLUDING REMARKS

We studied the following wireless data access scenario: a number of clients send query requests to a server to query for a set of data items; the server replies with the requested content via the common broadcast channel. We designed a utility function (Equation (1)), which takes power consumption in transmit, receive and idle modes, into consideration. Based on this power aware utility function, we formulated our power aware wireless data access scheme as a non-cooperative game—**WDA game**. We have shown that clients are *not always* necessary to send query requests to the server. Instead, each client determines its equilibrium request probability without *any explicit* communication with one another. We also formalized and evaluated the algorithms for the server and clients in the WDA game. Simulation results confirmed that our proposed power aware wireless data access scheme, comparing with a simple always-request one, increases the utility and lifetime of *every client* while reducing the number of requests sent, at the cost of a slightly larger average query delay.

REFERENCES

- [1] D. Barbara and T. Imielinski, "Sleepers and Workaholics: Caching Strategies in Mobile Environments," *Proc. ACM SIGMOD Int'l Conf. Management of Data*, vol. 23, no. 2, pp. 1–12, May 1994.
- [2] K. C. Border, *Fixed Point Theorems with Applications to Economics and Game Theory*, Cambridge University Press, 144 pages, ISBN: 0521388082, July 1989.
- [3] L. Breslau, P. Cao, L. Fan, G. Phillips and S. Shenker, "Web Caching and Zipf-like Distributions: Evidence and Implications," *Proc. INFOCOM 1999* vol. 1, pp. 126–134, Mar. 1999.
- [4] C. Buragohain, D. Agrawal and S. Suri, "A Game Theoretic Framework for Incentives in P2P Systems," *Proc. Peer-to-Peer Computing* vol. 3, pp. 48–56, Sept. 2003.
- [5] G. Cao, "A Scalable Low-Latency Cache Invalidation Strategy for Mobile Environments," *IEEE Trans. Knowledge and Data Engineering*, vol. 15, no. 5, pp. 1–15, May 2003.
- [6] A. Chockalingam, M. Zorzi, L. B. Milstein and P. Venkataram, "Performance of a Wireless Access Protocol on Correlated Rayleigh-Fading Channels with Capture," *IEEE Trans. Communications*, vol. 46, no. 5, pp. 644–655, May 1998.
- [7] A. Datta, D. E. VanderMeer, A. Celik and V. Kumar, "Broadcast Protocols to Support Efficient Retrieval from Databases by Mobile Users," *ACM Trans. Database Systems* vol. 24, no. 1, pp. 1–79, Mar. 1999.
- [8] L. M. Feeney and M. Nilsson, "Investigating the Energy Consumption of a Wireless Network Interface in an Ad Hoc Networking Environment," *Proc. INFOCOM 2001*, vol. 3, pp. 1548–1557, Apr. 2001.
- [9] Hutchison Whampoa, <http://www.three.com>.
- [10] R. Krishnan, M.D. Smith, Z. Tang and R. Telang, "The impact of free-riding on peer-to-peer networks," *Proc. 37th Int'l Conf. System Sciences*, pp. 199–208, Jan. 2004.
- [11] M. J. Osborne and A. Rubinstein, *A Course in Game Theory*, MIT Press, 368 pages, ISBN: 0262650401, July 1994.
- [12] J. D. Parsons, *The Mobile Radio Propagation Channel*, Second Edition, John Wiley & Sons, 436 pages, ISBN: 047198857X, Nov. 2000.
- [13] H. Schulzrinne, X. Wu, S. Sidiroglou and S. Berger, "Ubiquitous Computing in Home Networks," *IEEE Communications Magazine*, vol. 41, no. 11, pp. 128–135, Nov. 2003.
- [14] M. K. H. Yeung and Y.-K. Kwok, "Wireless Cache Invalidation Schemes with Link Adaptation and Downlink Traffic," *IEEE Trans. Mobile Computing*, vol. 4, no. 1, pp. 68–83, Jan.-Feb. 2005.
- [15] M. K. H. Yeung and Y.-K. Kwok, "Design and Analysis of Channel Adaptive Wireless Cache Invalidation Strategies with Downlink Traffic," *Proc. IEEE Int'l Symp. Parallel Architectures, Algorithms, and Networks*, pp. 48–53, May 2004.
- [16] M. K. H. Yeung and Y.-K. Kwok, "A Game Theoretic Approach to Power Aware Wireless Data Access," Technical Report HKU-EE-TMC-0306-1104, Nov. 12, 2004.
- [17] L. Yin and G. Cao, "Adaptive Power-Aware Prefetch in Wireless Networks," *IEEE Trans. Wireless Communications*, vol. 3, no. 5, pp. 1648–1658, Sept. 2004.