Packet Dropping Algorithms for Energy Savings

Wanshi Chen, Urbashi Mitra and Michael J. Neely
Dept. of Electrical Engineering
University of Southern California
Los Angeles, USA
Email: {wanshich, ubli, mjneely}@usc.edu

***Abstract*— This paper investigates proactive packet dropping to achieve transmission energy savings. Such a scheme can be employed for applications which can tolerate a small fraction of packet losses. For a group of packets subject to a single transmission deadline, the optimal dropping scheme (vis-à-vis total transmission energy) is derived. For packets subject to individual delay constraints, the optimal scheme depends on the energy function and packet sizes. Thus, asymptotically optimal dropping schemes, $i.e.$, when packet size grows large, are pursued. The asymptotically optimal dropping scheme for a single dropped packet is obtained. For dropping more than one packet, two suboptimal, recursive schemes are proposed. These schemes achieve performance very close to the asymptotically optimal schemes as determined by an exhaustive search. Additionally, two performance bounds are derived. It is observed via simulations that significant energy savings are possible via intelligent packet dropping schemes.**

I. Introduction

The fundamental trade-off between energy and delay in wireless networks was first characterized for a single queue in [1] and later extended to a multi-user network in [2]. The minimization of transmission energy in some cases can be directly translated into network life maximization [3] or system throughput maximization [4]. Note that application delay consists of not only physical layer transmission and processing delay, but also upper layer delays such as buffering delay. As a result, cross-layer design becomes necessary [1][5][6].

In [3], the optimal energy-efficient transmission algorithm was developed for a group of packets subject to a common single transmission deadline. Under a Poisson packet arrival model, it was shown in [7] that the average transmission delay associated with this scheme grows monotonically and at a rate close to $\sqrt{M}$, where M is the total number of data packets. Energy-efficient transmission with individual packet delay constraints was studied in [7][8][9].

Packet transmissions over wireless networks are subject to packet losses. In addition to physical layer transmission errors, packet losses may also come from upper layers operations such as buffer overflow, delay constraint violations, etc. We underscore that these upper layer packet losses are unintentional. We propose to consider intentional packet dropping to improve energy efficiency.

Some delay-sensitive applications such as voice and video streaming typically can tolerate a small fraction of packet losses. This motivates the following questions: should we proactively drop packets? How much transmission energy savings can be achieved via proactive packet dropping? In this paper, we initiate the study of these questions. A related work appears in [10] which discussed asymptotic transmission energy and packet delay tradeoff under intelligent packet dropping. The effects of source fidelity on communication networks were investigated in [11]. In [12], the optimal transmission policy in minimizing the packet losses subject to average packet delay and transmission power was investigated. A related issue of minimizing average power subject to average delay and packet loss constraints was studied in [13].

We start with the single transmission deadline model [3] and show that dropping the last set of K packets results in maximum transmission energy savings. For the individual delay constraints model, optimal packet dropping is much more complicated and seems intractable. Furthermore, we see that with the individual delay constraints, an optimal dropping scheme is energy function and packet-size dependent. As a result, we study asymptotically optimal dropping schemes when the packet size is sufficiently large. We derive the asymptotically optimal and sub-optimal packet dropping schemes, along with two upper bounds by exploiting packet arrival and departure properties. Significant transmission energy savings are possible via intelligent packet dropping as evidenced by the numerical results.

This paper is organized as follows. In Section II, the system model is described. Section III presents the optimal dropping algorithm for the single deadline model and its impact. The intelligent packet dropping scheme for the individual delay constraint model is discussed in Section IV. Numerical results are given in Section V. Finally, some concluding remarks are drawn in Section VI.

II. System Model

Suppose there are M packets to be transmitted, with packet arrival times $t_i, i = 1, \cdots, M$, through an additive white Gaussian noise (AWGN) channel. The packet arrivals are assumed to be random, but known to the optimal scheduler. The packet inter-arrival times are denoted by $d_i = t_{i+1} - t_i$. The packet sizes of the M packets are assumed to be the same and equal to B. The design principles discussed in the sequel can also be extended to unequal packet size scenarios. Each packet is delivered over a channel with certain transmission duration, denoted by τ_i. Let $w(\tau)$ be the energy required to transmit a packet with transmission duration τ, and this energy function is assumed to be non-negative, monotonically decreasing, and strictly convex, as in [3][7].

In the single transmission deadline model [3], the goal is to deliver all packets by the time deadline T while minimizing the total packet transmission energy. The optimal scheduling algorithm, without packet dropping, is explicitly specified in [3]. One feature of the optimal solution in this model is that transmission durations are non-increasing, $i.e.$, $\tau_i \geq \tau_{i+1}$ for $i \in [1, \cdots, M-1]$.

In the individual packet delay constraint model [7], each packet has its delay constraint $T_i, i \in [1, \cdots, M]$, at or before which the packet has to be successfully delivered since its arrival. In this paper, we will focus the case when the individual delay constraints are equal, $i.e.$, $T_i = T, \forall i$. The optimal scheduling algorithm was derived in [7][9]. Note that the property of $\tau_i \geq \tau_{i+1}$ may no longer hold [7][9].

In the following, we will investigate optimal schemes of dropping $K \geq 1$ packets from a total of M packets for both the single transmission deadline model and the individual delay constraint model. The optimal criterion herein is defined as minimizing the average transmission energy of the remaining $N = M - K$ packets. Note that once a set of K packets to drop is identified, the optimal scheduling strategies of [3][7][9] can be employed to minimize energy for the remaining packets. The challenge is to identify the best K packets to drop.

III. Optimal Packet Dropping for the Single Transmission Deadline Model

We assume the common transmission deadline T remains unchanged after packet dropping. It is certainly not optimal to drop the first packet as this will shrink the total overall time resource for scheduling. We have the following proposition:

Proposition 3.1: Dropping the last set of K packets from the M packets and then applying the optimal scheduler from [3] on the remaining packets results in the minimum transmission energy and hence is an optimal packet dropping scheme for the single transmission deadline model.

Proof: If we drop the last K packets, the arrival times of the remaining N packets are the same as the original first N packets. Consider a new scheme of dropping K packets in which at least one dropped packet $\in [2, \cdots, N]$. This new set of packet arrival times can always be emulated by the original first N packets by buffering these packets separately and releasing them to the actual queueing system exactly at the emulated arrival times. Thus, dropping the last K packets can emulate any other dropping scheme and is optimal. ■

Note that given a vector of inter-arrival times and a delay constraint, the optimal *scheduling* algorithm is unique [9]. However, the optimal *packet dropping* scheme may not be unique (the optimal scheduling for the remaining N packets is still unique). Indeed, we can exhibit examples where dropping a different set of K packets yields exactly the same energy savings as dropping the last K packets. However, we have:

Proposition 3.2: Suppose we have an optimal scheme for dropping K packets from M packets; denote the indices of the dropped K packets as $[m_1, \cdots, m_K]$ with $m_i < m_j, \forall i < j$. Then, dropping any combination of K packets within $[m_1, \cdots, M]$ such that $m_i^{'} \geq m_i, 1 \leq i \leq K$, will result in the same minimum transmission energy and is thus also energy optimal.

In such cases where optimality is non-unique, it is desirable from a delay perspective to drop packets as early as possible.

IV. Optimal Packet Dropping for the Individual Delay Constraint Model

Note that for the single transmission deadline model, the optimal packet dropping algorithm is simple and its optimality is regardless of packet sizes. However, as we will show, the optimal packet dropping algorithm for the individual delay constraint model is much more complicated and seems intractable. The optimal choice of packets to drop is no longer packet size independent.

Generally, it is not optimal to drop a packet right before or right after a packet with an inter-arrival time equal to or larger than T, as it will reduce the total allowable time resources, compared with dropping any other packet. We introduce the following definitions (see also [7]). All definitions below apply to a given sequence of packet arrival times for which the optimal scheduler (without dropping) are applied.

Definition: A *scheduling separation interval* is an interval during which all packets, except possibly the last one, have $d_i < T$.

Note that Packets belong to different scheduling separation intervals can be scheduled independently since an inter-arrival time of more than T requires all packets before that inter-arrival time must be delivered due to the T deadline.

Definition: A *group* is a collection of consecutive packets in a given scheduling separation interval such that, if the optimal scheduling algorithm is applied to the entire set of packets of the scheduling separation interval, the first packet of the group would begin its transmission with an empty buffer, and the last packet of the group would also end with an empty buffer.

Definition: A *delay-critical packet* is a packet with a packet delay equal to T under the optimal scheduling (without packet dropping) algorithm.

Definition: A *subgroup* is a collection of consecutive packets, within a group, having the same transmission duration.

Definition: A *type-1 group* is a group containing no delay-critical packet, except possibly the last packet in the group. When the last packet is a delay-critical packet, this group is the last group of a given scheduling separation interval.

Definition: A *type-2 group* is a group containing at least one delay-critical packet, in addition to possibly the last packet in the group.

Note that it is easy to identify the different groups associated with an inter-arrival sequence simply by performing one sweep of the optimal scheduling algorithm to the packets. Under the optimal scheduling, a subgroup may end with either an empty buffer or a delay-critical packet [7]. A type-1 group only has one subgroup. A type-2 group has two or more subgroups and the transmission durations of these subgroups within a given type-2 group are monotonically increasing [7].

Definition: A *minimum group* is a group which contains the minimum transmission duration. If two or more groups

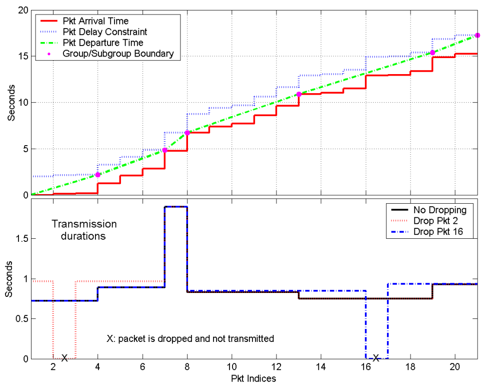

Fig. 1. An example run of the optimal transmission durations, M = 20, T= 2, and $\lambda = 1$ packets/second.

have the same minimum transmission duration, the minimum subgroup refers to one of the type-2 groups (if it exists), or the last type-1 group in a given scheduling separation interval.

Definition: A *minimum subgroup* is the subgroup which has the minimum transmission duration in the minimum group. For a type-1 group, the minimum subgroup is the minimum group itself. For a type-2 group, the first subgroup is referred as the minimum subgroup.

Correspondingly, we can also call a subgroup a type-1 (or type-2) subgroup if it belongs to a type-1 (or type-2) group. Figure 1 shows an example run of the optimal transmission durations of 20 packets. It can be seen that the minimum group (a type-2 group) consists of packets 1 to 7, with packets 1 to 3 belonging to subgroup 1. Packet 3 is a delay-critical packet. Packets 8 to 12 belong to a type-1 group.

Claim 4.1: If the minimum group is a type-1 group, it must be the last group of a scheduling separation interval.

Proof: By contradiction. From the definition of the minimum group, and the non-increasing properties of transmission durations between adjacent groups [7][9], if a type-1 group minimum group is followed by one or more groups, its transmission duration has to be less than that of the immediate subsequent group. This causes a contradiction. ■

Claim 4.2: When dropping a single packet, the transmission duration of the minimum subgroup can be increased only if the dropped packet is part of the minimum subgroup.

Proof: We provide a sketch of the proof. Note that all preceding (sub)groups have transmission durations no less than that of the minimum subgroup and a minimum subgroup always starts with an empty buffer, the minimum subgroup can not benefit from the released time resource by dropping any preceding packets. On the other hand, since the last packet of the minimum subgroup is a delay-critical packet, the minimum subgroup cannot benefit from dropping a packet after the minimum subgroup. ■

Now it may seem that dropping one packet from the minimum subgroup would yield less total transmission energy compared dropping one packet from any other subgroups. This is not necessarily the case. In Figure 1, the multiplicity (i.e., the number of packets) of the minimum subgroup is 3 (packets 1 to 3). Note that the subgroup containing packets 13 to 18 has the second shortest transmission duration with multiplicity of 6. Due to the multiplicity difference and minimum subgroup switchover after packet dropping, dropping packet 2 may not necessarily be advantageous over dropping packet 16 for a given packet size. To summarize:

Claim 4.3: When dropping a packet, it is not always optimal to drop it from the minimum subgroup.

That is, in most cases it is impossible to design an optimal packet dropping scheme based on transmission durations alone, as the full energy function $w(\tau, B)$ is required. This is very different from the single deadline case, in which the optimal transmission durations do not depend on the packet size or energy function.

It is often of interest to limit the maximum transmit power, which is determined by the minimum transmission duration of the minimum subgroup. Moreover, in scenarios when the minimum subgroup is outweighed in energy contribution by another group (or subgroup), the difference in contribution by these two groups typically is not significant. Thus, we introduce the following definition for a class of packet-size independent dropping schemes:

Definition: An *asymptotically* optimal dropping scheme is a packet dropping scheme which results in the minimum average transmission energy as the packet size approaches infinity.

In order to have the minimum subgroup contribute the most asymptotically to the total transmission energy, regardless of its multiplicity, the energy function is limited to have some desirable properties. Denote τ_{min} as the minimum transmission duration with a multiplicity of L_{min} and another transmission duration τ_{other} with a multiplicity L_{other}. As long as $w(\tau_{min}, B)/w(\tau_{other}, B)$ is still a monotonically increasing function of B, there always exists a B_0, such that when $B > B_0$, $L_{min}w(\tau_{min}, B) > L_{other}w(\tau_{other}, B)$, or the minimum subgroup is the leading contributor in the total transmission energy. This property is typical for energy functions, and is satisfied, e.g., by the energy function $w(\tau, B) = \tau(2^{2B/\tau} - 1)$ from [3]. From Claim 4.2, when the energy function has these desirable properties, we have:

Proposition 4.1: When dropping one packet, the packet has to be dropped from the minimum subgroup in order to achieve an asymptotically optimal total transmission energy. If there are multiple minimum subgroups, choose the one with the largest multiplicity. The multiplicity here is defined as the number of consecutive packets around the minimum subgroup (which may include previous and/or subsequent subgroups) having the same minimum optimal transmission duration.

Now, what if we need to drop multiple packets? Consider a recursive packet dropping scheme in which we will drop one packet at a time and repeat for K packets. For each step, the packet is dropped based on Proposition 4.1. For convenience, we denote this scheme as the '*recursive min subgroup*' approach. However, is this recursive approach asymptotically optimal when dropping $K > 1$ packets? We have the following claim:

Claim 4.4: The 'recursive min subgroup' approach described above is not necessarily asymptotically optimal when $K > 1$ packets are dropped.

Proof: We provide a counter example of 5 packets with inter-arrival times $[1, 1, 1, 1, T]$ with T = 2. Suppose we need to drop two packets. An asymptotically optimal dropping scheme (also an optimal scheme in this case) is to drop packets 2 and 4, such that the new inter-arrival times are given by [2, 2, 2]. By the recursive approach, we would drop 3 first, followed by either packet 2 or 4, resulting in a suboptimal set of inter-arrival times of [3, 1, 2] or [1, 3, 2], respectively. ■

The same result in Claim 4.4 also holds for optimal packet dropping schemes given a specific energy function and a packet size. In fact, the example in the proof of Claim 4.4 demonstrates that it is not optimal (vs. asymptotically optimal) to sequentially drop the best single packet. Thus, optimally dropping K packets does not seem to admit an algorithm with tractable complexity.

Note that when two or more packets arrive very close in time, these packets are likely to become delay-constrained. Thus, dropping a packet with the minimum inter-arrival time in the minimum subgroup is expected to significantly alleviate the delay-constrained issue. If $K > 1$, this can be repeated K times, similar to the 'recursive min subgroup' approach. We denote this recursive approach as the '*recursive min inter-arrival*' approach. One may of course further simplify the dropping scheme by just dropping the K packets with the minimum inter-arrival times among the M packets, which is denoted as '*simple min inter-arrival*' approach. It is worth mentioning that the K dropped packets by the 'recursive min inter-arrival' scheme do not necessarily have the minimum inter-arrival times among the M packets.

Due to the difficulty in obtaining explicit (asymptotically) optimal packet dropping schemes, we introduce two performance bounds for asymptotically optimal packet dropping schemes in the sequel. The extension to optimal packet dropping schemes is straightforward.

Consider dropping a packet from the scheduling separation interval containing the minimum subgroup. However, instead of dropping it from the minimum subgroup, we drop the second last packet of the given scheduling separation interval and schedule the packets in this scheduling separation interval using the single transmission deadline model. This can be repeated K times. Note that those scheduling separation intervals with no packets dropped are still scheduled based on the individual delay constraint model. We denote this dropping and scheduling approach as the '*recursive single deadline*' approach. We have the following proposition:

Proposition 4.2: The total transmission energy resulted from the asymptotically optimal packet dropping schemes for the individual delay constraint model is asymptotically lower bounded by that achieved by the 'recursive single deadline' approach described above.

We can improve the performance bound by observing the following. Regardless of which packet is dropped in the s-th scheduling separation interval (note that the first packet and the last packet, M_s, are not dropped), the resulting arrival times are always lower bounded by $[t_1, \cdots, t_{M_s-2}, t_{M_s}]$ and the resulting departure time constraints are always upper bounded by $[t_1, t_3, \cdots, t_{M_s-1}, t_{M_s}] + T$, where t_j is the arrival time of packet j in the scheduling separation interval. Similarly, when $K_s > 1$ packets need to be dropped, the energy performance is lower bounded by assuming arrival times of $[t_1, \cdots, t_{M_s-K_s-1}, t_{M_s}]$ and departure time constraints of $[t_1, t_{K_s+2}, \cdots, t_{M_s-1}, t_{M_s}] + T$ (equivalent to drop packets $M_s - K_s$ to $M_s - 1$, or the last set of K_s packets except packet M_s, with a new delay constraint vector $[0, \sum_{j=1}^{K_s} d_{j+1}, \sum_{j=1}^{K_s} d_{j+2}, \cdots, \sum_{j=1}^{K_s} d_{j+M_s-K_s-2}, 0] + T$). A recursive procedure, similar to that in the 'recursive single deadline' approach, can be derived based on the above dropping and scheduling scheme. Such an approach is denoted as the '*recursive arrival departure*' approach.

Proposition 4.3: The total transmission energy resulting from the asymptotically optimal packet dropping schemes for the individual delay constraint model is asymptotically lower bounded by that achieved by the 'recursive arrival departure' approach described above.

Corollary 4.3.1: E{Asymptotically Optimal} $\geq$ E{Recursive Arrival Departure} $\geq$ E{Recursive Single Deadline}, where E{.} denotes the asymptotical total transmission energy of a given scheme.

V. Numerical Results

In this section, we present numerical results for intelligent packet dropping schemes. We assume a Poisson arrival rate of $\lambda = 1$ packet/second. An energy function of $w(\tau) = \tau(2^{2B/\tau} - 1)$ [3] is used.

For the single transmission deadline model, Figure 2 shows the normalized transmission energy [1] for the no packet-dropping case, and the optimal dropping (dropping the last K packets) schemes. The transmission deadline T is set to $1/\lambda$ seconds after the last packet arrives. The results were averaged over 1000 independent runs. It can be observed that compared with no packet dropping, the optimal dropping scheduler yields significant energy savings even when $K = 1$ and it achieves transmission energy close to the lower bound of $w(1/\lambda)$. The energy savings due to the optimal dropping increases as the normalized packet size increases.

For the individual delay constraint model, Figure 3 compares the performance difference between an asymptotically optimal packet dropping scheme via brute-force search and the 'recursive min subgroup' scheme. The transmission energy is normalized by that of the no-dropping case with a reduced packet arrival rate of $\lambda(1 - K/M)$. The results are averaged over 10 independent runs. The two performance bounds are also shown for comparison. As can be seen, the 'recursive min subgroup' packet dropping scheme yields a transmission

[1] We first multiply by $(1 - K/M)$ in order to make a fair comparison between the schedulers with and without packet dropping, and then normalize with respect to the ideal case when packets have fixed inter-arrival times of $1/\lambda$.

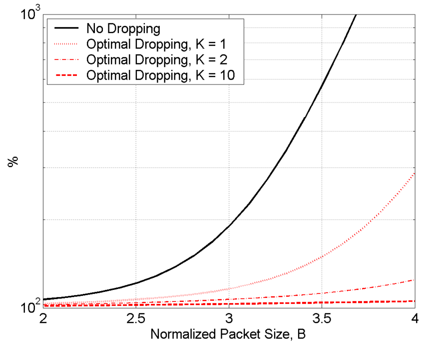

Fig. 2. Normalized transmission energy (in %) vs. B for the single deadline model, $M = 1000$.

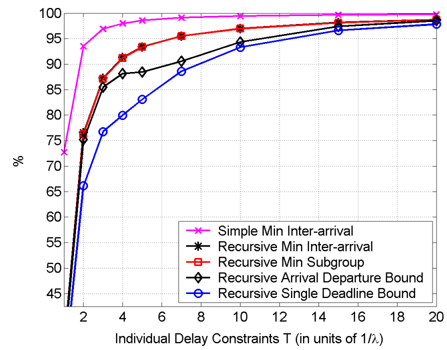

Fig. 4. Normalized transmission energy vs. T for the individual delay constraint model, $M = 1000$, $K = 10$, and $B = 2$.

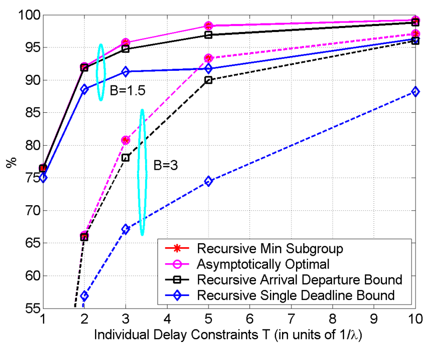

Fig. 3. Normalized transmission energy vs. T for the individual delay constraint model, $M = 200$, $K = 2$.

energy which is indistinguishable from that of the asymptotically optimal scheme achieved via brute-force search. The performance bound by the 'recursive arrival departure' scheme is fairly tight, especially when the individual delay constraint is relatively small or large. The performance bound by the 'recursive single deadline', on the other hand, is rather loose.

Figure 4 shows similar comparisons with $M = 1000$, and $K = 10$. As can be seen, the energy savings due to packet dropping decrease with T, which shows that the dropping scheme is more effective for more demanding delay constraints. The 'recursive min inter-arrival', based on the observation of the minimum inter-arrival time in the minimum subgroup in each iteration, achieves almost the same energy savings as the more complicated 'recursive min subgroup' scheme. The simplest scheme, 'simple min inter-arrival', which blindly drops the packets with minimum inter-arrival times, yields some energy savings, but noticeably less than other suboptimal dropping schemes.

VI. Conclusions

This paper investigated proactive packet dropping to achieve transmission energy savings. The optimal packet dropping schemes for the single transmission deadline model was derived. For packets subject to individual delay constraints, the optimal scheme appears to be intractable. Thus, asymptotically optimal dropping schemes were pursued, along with suboptimal packet dropping schemes and performance bounds. Simulations demonstrated significant energy savings are possible via intelligent packet dropping schemes.

Acknowledgment

This research has been funded in part by one or more of the following grants or organizations: NSF Special Projects ANI-0137091, USC WISE Program, NSF ANI-0087761, and NSF Grant OCE 0520324.

References

[1] R. Berry and R. G. Gallager, "Communication over fading channels with delay constraints," *IEEE Trans. Info. Theory*, vol. 48, no. 5, pp. 1135–1149, May 2002.

[2] M. J. Neely, "Optimal energy and delay tradeoffs for multi-user wireless downlinks," *IEEE Infocom*, 2006.

[3] E. Uysal-Biyikoglu, B. Prabhakar, and A. El Gamal, "Energy-efficient packet transmission over a wireless link," *IEEE/ACM Trans. Networking*, vol. 10, no. 4, pp. 487–499, Aug 2002.

[4] TIA-856-A, "cdma2000 high rate packet data air interface specification (2004)," April 2004.

[5] R. A. Berry and E. M. Yeh, "Cross-layer wireless resource allocation," *IEEE Signal Processing Magazine*, vol. 21, no. 5, pp. 56–68, Sept 2004.

[6] E. Telatar and R. G. Gallager, "Combining queuing theory with information theory for multiaccess," *IEEE J. Selected. Areas Commun.*, vol. 13, pp. 963–969, Aug 1995.

[7] W. Chen and U. Mitra, "Energy efficient scheduling with individual packet delay constraints," *IEEE Infocom*, 2006.

[8] D. Rajan, A. Sabharwal, and B. Aazhang, "Delay-bounded packet scheduling of bursty traffic over wireless channels," *IEEE Trans. Info. Theory*, vol. 50, no. 1, pp. 125–144, Jan 2004.

[9] M. A. Zafer and E. Modiano, "A calculus approach to minimum energy transmission policies with quality of service gurantees," *IEEE Infocom*, March 2005.

[10] M. J. Neely, "Intelligent packet dropping for optimal energy-delay tradeoffs in wireless downlinks," *Proc. of the 4th Int. Symposium on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks (WiOpt)*, 2006.

[11] D. N. C. Tse, "Variable-rate lossy compression and its effects on communication networks," *Ph.D. Thesis, Mass. Inst. Tech.*, Feb 1995.

[12] D. Rajan, A. Sabharwal, and B. Aazhang, "Transmission policies for bursty traffic sources on wireless channels," *35th Annual Conf. on Info. Sciences and Systems*, March 2001.

[13] H. Wang and N. B. Mandayam, "A simple packet transmission scheme for wireless data over fading channels," *IEEE Trans. Commun.*, vol. 52, pp. 1055–1059, July 2004.