University of Pennsylvania
ScholarlyCommons

Real-Time and Embedded Systems Lab (mLAB) School of Engineering and Applied Science

1-1-2008

MEERA: Cross-Layer Methodology for Energy Efficient Resource Allocation in Wireless Networks

Sofie Pollin
IMEC, K.U. Leuven

Rahul Mangharam
University of Pennsylvania, rahulm@seas.upenn.edu

Bruno Bougard
IMEC, K.U. Leuven

Liesbet Van der Perre
IMEC

Ingrid Moerman
Ghent University

See next page for additional authors

Suggested Citation:
Pollin, S.; Mangharam, R.; Bougard, B.; Van der Perre, L.; Moerman, I.; Rajkumar, R.; Catthoor, F.; , "MEERA: Cross-Layer Methodology for Energy Efficient Resource Allocation in Wireless Networks," Wireless Communications, IEEE Transactions on , vol.7, no.1, pp.98-109, Jan. 2008.

©2008 IEEE. Personal use of this material is permitted. However, permission to reprint/republish this material for advertising or promotional purposes or for creating new collective works for resale or redistribution to servers or lists, or to reuse any copyrighted component of this work in other works must be obtained from the IEEE.

This paper is posted at ScholarlyCommons. http://repository.upenn.edu/mlab_papers/8
For more information, please contact repository@pobox.upenn.edu.

MEERA: Cross-Layer Methodology for Energy Efficient Resource Allocation in Wireless Networks

Abstract
In many portable devices, wireless network interfaces consume upwards of 30% of scarce system energy. Reducing the transceiver's power consumption to extend the system lifetime has therefore become a design goal. Our work is targated at this goal and is based on the following two observations. First, conventional energy management approaches have focused independently on minimizing the fixed energy cost (by shutdown) and on scalable energy costs (by leveraging, for example, the modulation, code-rate and transmission power). These two energy management approaches present a tradeoff. For example, lower modulation rates and transmission power minimize the variable energy component, but this shortens the sleep duration thereby increasing fixed energy consumption. Second, in order to meet the Quality of Service (QoS) timeliness requirements for multiple users, we need to determine to what extent each system in the network may sleep and scale. Therefore, we propose a two-phase methodology that resolves the sleep-scaling tradeoff across the physical, communications and link layers at design time and schedules nodes at runtime with near optimal energy-efficient configurations in the solution space. As a result, we are able to achieve very low run-time overheads. Our methodology is applied to a case study on delivering a guaranteed QoS for multiple users with MPEG-4 video over a slow-fading channel. By exploiting runtime controllable parameters of actual RF components and a modified 802.11 Medium Access Controller, system lifetime is increased by a factor of 3-to-10 in comparison with conventional techniques.

Keywords
Energy-Efficient Wireless Systems

Disciplines
Electrical and Computer Engineering | Engineering

Comments
Suggested Citation:
Pollin, S.; Mangharam, R.; Bougard, B.; Van der Perre, L.; Moerman, I.; Rajkumar, R.; Catthoor, F.; , "MEERA: Cross-Layer Methodology for Energy Efficient Resource Allocation in Wireless Networks," Wireless Communications, IEEE Transactions on , vol.7, no.1, pp.98-109, Jan. 2008.

©2008 IEEE. Personal use of this material is permitted. However, permission to reprint/republish this material for advertising or promotional purposes or for creating new collective works for resale or redistribution to servers or lists, or to reuse any copyrighted component of this work in other works must be obtained from the IEEE.

Author(s)
Sofie Pollin, Rahul Mangharam, Bruno Bougard, Liesbet Van der Perre, Ingrid Moerman, Ragunathan Rajkumar, and Francky Catthoor

This journal article is available at ScholarlyCommons: http://repository.upenn.edu/mlab_papers/8

MEERA: Cross-Layer Methodology for Energy Efficient Resource Allocation in Wireless Networks

Sofie Pollin, Rahul Mangharam, *Student Member, IEEE*,
Bruno Bougard, Liesbet Van der Perre, Ingrid Moerman, Ragunathan Rajkumar *Member, IEEE*,
Francky Catthoor, *Fellow, IEEE*

***Abstract*— In many portable devices, wireless network interfaces consume upwards of 30% of scarce system energy. Reducing the transceiver's power consumption to extend the system lifetime has therefore become a design goal. Our work is targated at this goal and is based on the following two observations. First, conventional energy management approaches have focused independently on minimizing the fixed energy cost (by shutdown) and on scalable energy costs (by leveraging, for example, the modulation, code-rate and transmission power). These two energy management approaches present a tradeoff. For example, lower modulation rates and transmission power minimize the variable energy component, but this shortens the sleep duration thereby increasing fixed energy consumption. Second, in order to meet the Quality of Service (QoS) timeliness requirements for multiple users, we need to determine to what extent each system in the network may sleep and scale. Therefore, we propose a two-phase methodology that resolves the sleep-scaling tradeoff across the physical, communications and link layers at design time and schedules nodes at runtime with near optimal energy-efficient configurations in the solution space. As a result, we are able to achieve very low run-time overheads. Our methodology is applied to a case study on delivering a guaranteed QoS for multiple users with MPEG-4 video over a slow-fading channel. By exploiting runtime controllable parameters of actual RF components and a modified 802.11 Medium Access Controller, system lifetime is increased by a factor of 3-to-10 in comparison with conventional techniques.**

***Index Terms*— TODO**

I. Introduction

Over the past two decades, processor power consumption has increased by over 200% every four years, while battery energy density has increased by a modest 25% [1]. As the demand for higher data-rate wireless systems is growing [2], the increasing energy disparity poses a challenge to system designers in terms of system lifetime and cooling cost [3]. In particular, the power consumption of 802.11 transceivers [4] has been increasing with the creation of each new standard (Table I). Studies show that users favor handhelds weighing less than 340 grams and devices requiring less frequent recharging [5]. Lithium ion batteries currently provide the highest capacity of approximately 90Whr/Kg [6]. If we target a battery weight less than 50% of the handheld's weight, we get 15Whr of battery energy. Now consider an average user's daily power consumption of 2 hours in receive mode, $1/2$ hour in transmit mode and 4 hours in idle mode [7]. An 802.11a transceiver with the given usage profile expends approximately 7.5Whr or 50% of the handheld's battery capacity. On average, the wireless interface consumes upwards of 30% of a laptop's energy, thus motivating the need to decrease it's energy consumption [8].

TABLE I
WIRELESS TRANSCEIVER POWER CONSUMPTION.

Mode	802.11b	802.11a	802.11g
Sleep	$132mW$	$132mW$	$132mW$
Idle	$544mW$	$990mW$	$990mW$
Receive	$726mW$	$1320mW$	$1320mW$
Transmit	$1089mW$	$1815mW$	$1980mW$

While the highest power is drawn during active modes, the extended duration of the idle mode accounts for a significant budget of the overall system energy. An energy-efficient design must therefore jointly optimize the active and fixed energy consumption. By throttling, for example, the transmit power, modulation rate and code rate, the active energy consumption per bit can be reduced for the same target receive signal-to-noise ratio (SNR) [9]. The trade-off between transmission time and transmission energy is convex - a fundamental property for wireless communication bounded by Shannon's channel capacity [10]. In the remainder of this paper, we classify this active energy saving technique as *scaling*. On the other hand, the static energy consumed during the idle mode may be minimized or eliminated by maximizing the shutdown time, otherwise known as *sleeping*.

The challenge in energy management for dynamic wireless systems is to determine the extent to which the system should sleep and scale while meeting per-packet QoS timing requirements. The key assumption underlying the method is that the systems will operate in dynamic environments where a single energy management solution is not sufficient. The variation present in the system motivates the use of flexible cross-layer

Manuscript received April 15, 2005. The material in this paper was presented in part at IEEE Conference on Computer Communications (INFOCOM), Miami, FL, USA, March 2005.
Sofie Pollin is with IMEC and K.U.Leuven, Kapeldreef 75, 3000 Leuven, Belgium (phone: +3216288750; fax: +3216281515; e-mail: pollins@imec.be).
Rahul Mangharam is with Carnegie Mellon University, Pittsburgh, PA, 15213 USA (e-mail: rahulm@ece.cmu.edu).
Bruno Bougard is with IMEC and K.U.Leuven, Belgium (e-mail: bougardb@imec.be).
Ragunathan Rajkumar is with Carnegie Mellon University, Pittsburgh, PA, 15213 USA (e-mail:raj@ece.cmu.edu).
Liesbet Van der Perre is with IMEC, Belgium (e-mail: vdperre@imec.be).
Francky Catthoor is with IMEC and K.U.Leuven, Belgium (e-mail: catthoor@imec.be).
Ingrid Moerman is with UGent, Belgium (e-mail: ingrid.moerman@intec.ugent.be).
Research supported by .

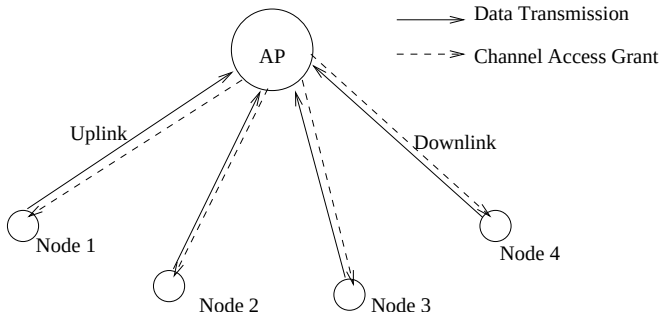


Fig. 1. Centrally controlled point-to-multipoint LAN/PAN topology with uplink and downlink communication.

solutions that adapt to the current instantiation while saving maximally. The main system state variables affecting each user are the present channel state and the instantaneous data rate demanded. Furthermore, in a network of such systems, an efficient energy management algorithm should exploit the variations across users to minimize the overall network energy consumption.

Therefore the problem explored here could be stated as follows: *How does one decide what system configurations to assign to each user at runtime to minimize the overall energy consumption while providing a sufficient level of QoS? This must be achieved for a network of users with bursty delay-sensitive data and over a slow fading channel.*

A. Cross-Layer Resource Allocation

While a cross-layer approach, by definition, violates the concept of strict network layering, we will show that by jointly leveraging the runtime controllable parameters or *control dimensions* across multiple layers we can achieve significant energy savings. Our focus is on wireless networks where all users are in the same collision domain with an access point (AP) to arbitrate exclusive channel access (Fig. 1). Our solution exploits control dimensions from (a) radio frequency integrated circuit (RFIC) system models, (b) communication theoretic tradeoffs, and (c) link-layer scheduling.

The following three observations show the need to integrate energy management approaches across layers. First, state-of-the-art wireless systems such as 802.11a devices are built to function at a fixed set of operating points assuming worst-case conditions. Irrespective of the link utilization, the highest feasible transmission rate is used and the power amplifier operates at maximum output power [8]. For non-scalable systems, the highest rate results in the smallest duty cycle and hence the lowest energy consumption. On the other hand, for scalable systems, this strategy results in excessive energy consumption for average channel conditions and link utilizations. Recent energy-efficient wireless system designs focus on VLSI implementations and adaptive physical layer algorithms where a lower transmission rate results in energy savings per bit [9], [11]. For these schemes to be practical, they should be aware of the hardware energy characteristics at various operating points.

Second, to realize sizable energy savings, systems need to shutdown the components when inactive. This is achieved only by tightly coupling the MAC communication to the power management strategy in order to communicate traffic requirements of each user for scheduling shutdown intervals. Finally, intricate tradeoffs exist between sleeping and scaling while satisfying the timeliness requirements across multiple users. As the channel is shared, lowering the rate of one user reduces the time left for the other delay-sensitive users. This forces them to increase their rate, at the cost of energy consumption or bit errors.

Our methodology for energy-efficient resource allocation, MEERA, couples these three layers in a systematic manner to determine the optimal system-wide power management at runtime.

B. Contributions of this paper

In this paper we propose a two-phase solution to efficiently solve the sleep-scaling tradeoff across the physical, communications and link layers for multiple users. At design time the problem is resolved by selecting a set of appropriate operating points in the solution space. Starting from this configuration space, we schedule the nodes at runtime to achieve near-optimal energy consumption with low overheads. In Section II we present our two-phase energy management methodology. In Section III, we apply this general methodology to a specific case study employing physical layer models based on actual RFICs, an energy-aware MAC, MPEG-4 movies and a realistic indoor channel model [12]. Section IV presents simulation results for multiple users to validate the methodology. Concluding remarks are in Section V.

C. Related Work

There have been several initiatives to design energy-efficient processors [13], [14] primarily employing dynamic voltage scaling and low-power VLSI implementations. In [15], dynamic voltage scaling is further exploited by adding a smart buffer scheme. These methods, however, do not extend well for wireless transceivers, as the performance of analog circuits, which dominate the energy consumption, does not scale as monotonically with lower voltages as digital circuits. Wireless communications present non-linear and discrete energy-performance tradeoffs between coding and transmit power [16], between modulation and active circuit energy consumption [17] and between transmission rates and shutting off the system [11].

To address this, researchers have approached the problem either from an information-theoretic perspective [9, 17] or from an implementation-specific viewpoint [13], [19]. In [17], modulation strategies are derived for delay-bounded traffic. It is shown that when the transmit power and circuitry power are comparable, the transmission energy decreases with the product of bandwidth and transmit duration. They however only consider an idealized network restricted to a single flow with no medium access controller (MAC) or link layer retransmissions, and with ideal constellation sizes. In [9], scaling is framed as a convex optimization problem where

users lower their transmission rate to minimize energy. They do not consider the fixed circuit energy consumed during idle and receive intervals. On the other hand, [16] explores the practical cross-layer trade-off between transmission power control and physical layer (PHY) rate for a centrally controlled MAC with retransmissions. Their solutions are specific and applicable to the 802.11a PHY [2]. They derive performance based on simple AWGN channel models. They also consider only a single flow with no delay constraints or system sleep modes.

A more general framework to exploit the energy scalability of transceivers is provided in [11]. They derive the operating regions when a transceiver may sleep or use scaling. The analysis is based on simplified physical layer energy models and only point-to-point file transfer is considered. Approaches to trade-off energy and rate, taking into account implementation aspects and operating conditions are proposed in [13], [19]. An energy-performance trade-off is presented for a single link at design-time, as function of the system implementation.

Offline hardware energy optimizations for energy-scalable systems are proposed in [11], [17], [18] and in [19] for cellular systems. They express the need for a practical runtime scheme to determine the configurations for one or more users, taking into account environmental and application constraints where appropriate. In order to derive optimal or near-optimal operating points, a framework is needed to consider the impact of the various control dimensions, the trade-offs between them and the overall benefit to the user. In [20] a self-tuning power management scheme is proposed which adapts its sleeping behavior to user access patterns, application context, characteristics of the network interface and energy usage of the platform. They foresee a simple interface that allows applications to give hints about their intent, and show that self-tuning improves both performance and energy conservation compared to static approaches. Similarly, in [21] they abstract the power management to a higher level to exploit application-specific information to balance power savings and data performance.

A middleware scheme is proposed in [22] which exploits global information (e.g. background traffic) and static device-specific characteristics (e.g. hardware and offered video formats) to dynamically adapt the packet schedule for sleep maximization. In [23], the authors present a useful approach to maximize the utility for multimedia applications through scheduling of multiple resources along multiple control dimensions. Our approach to minimize energy consumption has a similar basis and is extended for use in dynamic wireless systems by incorporating communications constraints. Traffic scheduling taking into account both the application and the sleep profile of current hardware, is shown to outperform straightforward transmission schemes significantly [24]. We extend the technique taking into account other hardware energy management techniques. The use of various scheduling schemes to optimize system utility in cellular systems is investigated extensively [25], but to the best of our knowledge no scheduling techniques exist that specifically consider the underlying hardware.

II. Energy-Efficient Design Methodology

The design of low-power wireless systems needs to encompass RF components, adaptive physical layer algorithms, and the MAC protocol while taking into account environment and application constraints. In order to extract significant energy savings from the system resource allocation algorithms must work harmoniously with implementations. This requires a sound methodology that can scale with the combinatorial explosion of the number of configurations and with the non-linear and implementation-specific set of control dimensions.

Our methodology, outlined in Fig. 2, consists of two-phases. In the design or calibration-time phase, we determine the tradeoff between sleeping and scaling across all system states. This is done by mapping a *cost profile* (e.g., energy consumed for a successful transmission), *resource profile* (e.g., time to complete that successful transmission) and *quality profile* (e.g., probability to complete transmission successfully) for each node, for every possible system configuration and state. For even a modest number (e.g., 8) of control dimensions, this can lead to a large number of possible system configurations (on the order of 10^6 for the considered system). However, this system calibration is a one-time investment that allows for fast and efficient run-time system adaptation. To decrease run-time complexity, the Cost-Resource-Quality profiles obtained are then pruned by taking the convex minorant across all three dimensions [26]. Following this, in the runtime phase, each node's pruned Cost-Resource-Quality profile is traversed to determine the operation point for given system states (e.g., channel state) and QoS requirements of all users. The runtime phase is executed in a greedy manner, selecting the operation point for the node with maximal cost savings (i.e., energy) for every additional unit of resource (i.e., time) consumed while meeting the quality constraint. MEERA is implemented in the AP and delivers a bounded sub-optimal performance. We describe the methodology in more detail by formally defining the MEERA resource management model and stating the resource allocation as a convex optimization problem.

A. MEERA Resource Management Model

Consider a wireless network as in Fig. 1 where multiple nodes are centrally controlled by an AP. Each node (such as a handheld video camera) desires to transmit or receive frames at real-time and it is the AP's responsibility to assign channel-access grants. The resource allocation scheme within the AP specifies each user's system configuration settings for the next transmission based on the feedback of the system state from the current transmission. It must ensure that the nodes meet their performance constraints by delivering their data in a timely manner while consuming minimal energy. The problem is now stated formally and a case-specific mapping is provided in the following section.

1) MEERA Definitions: The network consists of n flows $\{F_1, F_2, \ldots, F_n\}$ with periodic delay-sensitive frames or jobs. For notational simplicity, we assume a one-to-one mapping of flows to nodes, but our design methodology is applicable to more flows per node. Each flow $i, 1 \leq i \leq n$, is described by the following properties:

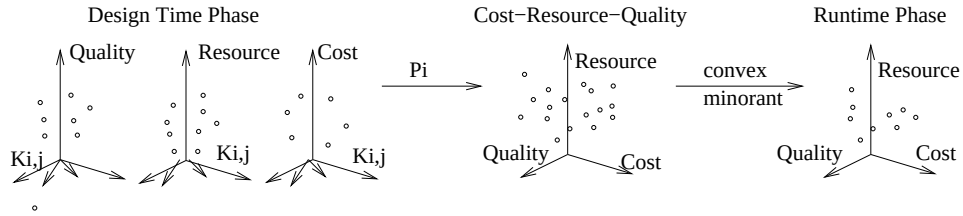


Fig. 2. At design time, a Cost, Resource and Quality profile is determined for each set of control dimensions based on the system state. The optimal Cost-Resource-Quality tradeoff is derived from this mapping to give operating points used at runtime.

1) Cost Function (C_i): This is the optimization objective, e.g. to minimize the total energy consumption of all users in terms of Joules/Job. For example, in a video context, a job is the timely delivery of the current frame of the video application.
2) QoS Constraint (Q_i): The optimization has take into account a minimum performance or QoS in order to satisfy the user. As delivery of real-time traffic is of interest, we describe the QoS in terms of the job failure rate (JFR) [27]. JFR is defined as the ratio of the number of frames not successfully delivered before the deadline to the number of frames issued by the application over the lifetime of the flow. The QoS constraint is specified at run-time by user i as a target JFR_i^*.
3) Shared Resource ($R_{i,l}, 1 \leq i \leq n, 1 \leq l \leq r$): Multiple resource dimensions, r, could be used to schedule flows or tasks in the network (e.g., time, frequency or space). In this paper, we consider the particular case where access to the channel is only divided in time. Therefore, time is the single shared resource and the total is denoted by R. The fraction of resource consumed by node i is denoted by R_i.
4) Control Dimensions ($K_{i,j}, 1 \leq i \leq n, 1 \leq j \leq k$): For a given wireless LAN architecture, there are k control knobs or dimensions, such as modulation, code rate, output power, etc. that control the system Cost, QoS and Resource given the current system state. The control dimension settings are discrete, inter-dependent and together have a non-linear impact. We define a setting of all knobs j for node i to be configuration point $\mathbf{K}_i$. We will define a relationship between $\mathbf{K}_i$ to Q_i, C_i and R_i in the next section.
5) System state ($S_{i,m}, 1 \leq i \leq n, 1 \leq m \leq s$): As we are operating in a very dynamic environment, the system behavior will vary over time. There are s environmental or application factors independent of the user or system's control represented by the system state variable, $S_{i,m}$. The system cost, resources required and quality offered all depend on the system state, as determined during design or calibration-time of the system. In a wireless environment, with e.g. Variable Bit Rate (VBR) video traffic, the system state is the current channel state and application frame size. Application frame size is translated to Queue Size at the MAC level to allow for application aware scheduling without explicit communication with the higher layers. After classification of the various system states, the calibration step can easily be performed to take into account implementation-specific aspects of the device. As the application frame size is translated to a lower layer metric, the calibration is specific to the PHY and MAC only, and can hence be included with the PHY/MAC hardware drivers of practical systems. This device-specific calibration step may be executed at design time or during device initialization as is outlined in Section III. The scheduling algorithm within the AP is executed with a period based on the time during which the system state remains stable, i.e. channel epoch and video frame rate.

To summarize, each flow F_i is associated with a set of possible system states $S_{i,m}$, which determines the mapping of the control dimensions $\mathbf{K}_i$ to the Cost ($\mathbf{K}_i, S_{i,m} \rightarrow C_i$), Resource ($\mathbf{K}_i, S_{i,m} \rightarrow R_i$) and Quality ($\mathbf{K}_i, S_{i,m} \rightarrow Q_i$). It is essential to note that for each user, depending on the current state, the relative energy gains possible by rate scaling or sleeping are different and should hence be exploited differently. Each user experiences different channel and application dynamics, resulting in different system states over time, which may or may not be correlated with other users. This is a very important characteristic which makes it possible to exploit multi-user diversity for energy efficiency.

2) MEERA Model Properties: The key aspects of MEERA are the mapping of the control dimensions to cost, resource and quality profiles respectively, and the generality of this mapping. A resource (respectively cost, quality) profile describes a list of potential resource (respectively cost, quality) allocation schemes resulting from each configuration point. These profiles are then combined to give a Cost-Resource-Quality trade-off, which is essential for solving the resource allocation problem (Fig. 2). The Cost-Resource-Quality trade-off function represents the behavior of a specific system for one user in a given system state.

Cost profile properties

- The finite set of discrete control dimension configurations is ordered by their increasing costs.
- The overall system cost, C, is defined as the weighted sum of costs of all flows, where each flow can be assigned a certain weight depending on its relative importance or to improve fairness [27]. Users may be assigned higher weights for example when their battery capacity is low or when they downscale their transmission rate by decreasing the video quality and get rewarded for reducing the network congestion. Users with a higher

weight will typically be allowed to save more energy compared to other users: $C = \sum_{i=1}^{n} \omega_i C_i$.

Resource profile properties

- The finite set of discrete control settings is ordered by their minimal resource requirement.
- The system resource requirement, R, is defined as the sum of the per flow requirements: $R = \sum_{i=1}^{N} R_i$.

Quality profile properties

- The finite set of discrete control dimension configurations is ordered with quality.
- The system quality, Q, is met when each individual user's constraint is met: $JFR_i \leq JFR_i^*, 1 \leq i \leq n$.

B. MEERA Resource Allocation Problem

We recall that our goal is to assign transmission grants via the AP, resulting in an optimal setting of the control dimensions to each node such that the per-flow QoS constraint for multiple users are optimally met with minimal energy consumption. For a given set of resources, control dimensions and QoS constraints, the scheduling objective can be formally stated as:

$$\min_{C} \sum_{i=1}^{n} \omega_i C_i, \quad 1 \leq m \leq s$$

s.t.

$$\begin{aligned}
JFR_i \leq JFR_i^*, &\quad 1 \leq i \leq n \quad (QoS\ Constraints) \\
\sum_{i=1}^{n} R_{i,l} \leq R_l^{max}, &\quad 1 \leq l \leq r \quad (Resource\ Constraints) \\
\mathbf{K}_i, S_{i,m} \rightarrow R_{i,l}, &\quad 1 \leq j \leq k \quad (Resource\ Profiles) \\
\mathbf{K}_i, S_{i,m} \rightarrow C_i &\quad 1 \leq m \leq s \quad (Cost\ Profiles) \\
\mathbf{K}_i, S_{i,m} \rightarrow Q_i & \quad (Quality\ Profiles)
\end{aligned}$$

The solution of the optimization problem yields a set of feasible operating points, $\mathbf{K} = \{\mathbf{K}_i, 1 \leq i \leq n\}$, for the network which fulfill the QoS target, maintains the shared resource constraint and minimizes the system cost. While the profile mapping and pruning is done during a one-time calibration step, we now describe how, using those profiles, the optimal configuration $\mathbf{K}$ is determined efficiently at runtime.

C. Two-phase Solution Approach

When considering energy-scalable systems, the number of control dimensions is large and leads to a combinatorial explosion of possible configurations (e.g., $O(10^6)$ for the case-study in Section III). At design or calibration time, a pruned set of configurations is determined. This set is represented in a table-driven structure that allows for a fast handling. At runtime, based on the system state, the best operating point is then extracted efficiently by a simple memory access.

1) Design-Time Phase: A property of our model is that the finite set of discrete control dimensions can be ordered, describing a range of possible costs, resources and quality for the system in each system state. For each additional unit of resource allocated, we only need to consider the configuration that satisfies the quality constraint and achieves the minimal cost for that resource unit. For each system state (e.g., channel and application loads), a subset of points is determined by pruning the Cost-Resource-Quality curves to yield only the minimum cost configurations, which will be denoted by $C_i(R_i, Q_i)$.

We define a calibration function , that is computed for every state $S_{i,m}$

$$\begin{aligned}
p_i(R_i, Q_i) = \min\{C_i | (\mathbf{K}_i, S_{i,m} \rightarrow R_i) \wedge (\mathbf{K}_i, S_{i,m} \rightarrow Q_i) \\
\wedge (\mathbf{K}_i, S_{i,m} \rightarrow C_i) \wedge (\mathbf{K}_i) \in \{\mathbf{K}_i\}\}
\end{aligned}$$

and defines a mapping between the Resource, Cost and the Quality of a node in a system state, $S_{i,m}$, as shown in Fig. 2. $\{\mathbf{K}_i\}$ is the set of configuration vectors for node i. Given the points after calibration in the Cost-Resource-Quality space, we are only interested in the ones that represent optimally the trade-off between energy, resource and quality for our system. Although the discrete settings and non-linear interactions in real systems do not lead to a convex trade-off, it can be well approximated as follows.

We calculate the *convex minorant* [26] of these pruned curves along the Cost, Resource and Quality dimensions, and consider the intersection of the results. We call this set the optimal Cost-Resource-Quality trade-off in the remainder (Fig.3). We show that the maximum segment size of the convex minorant determines the solution's deviation from the optimum[1]. Configurations on segments that are small compared to the largest segment size can be pruned away without affecting the bounds of the solution. As a result, we can typically expect less than 30 configurations per state. At run-time, the resource allocation scheme in the AP adapts to system state by fetching the correct configurations from memory. This operation is cheap compared to the cost of calibration that only has to be carried out once. In the next subsection, we detail how this information can be combined efficiently into a global curve representing the network Cost-Resource-Quality tradeoff. Profiling each user separately and combining the information at runtime is optimal for independent users or when the correlation is unknown. When correlation is present, the number of system states to calibrate and the runtime combination could be further reduced.

2) Run-Time Phase: As the current system state of all the users is only known at runtime, a light-weight scheme is necessary to assign the best system configurations for each user, while meeting the QoS requirements. We therefore employ a greedy algorithm to determine the per-flow resource usage R_i for each user to minimize the total cost C while meeting system constraints. The algorithm first constructs the optimal local Cost-Resource trade-off curve $C_i(R_i)$ by taking the optimal points in both dimensions that meet the run-time average quality constraint JFR^*. Next, the scheduler traverses all flows' two-dimensional Cost-Resource curves and at every step consumes resources corresponding to the maximum negative slope across all flows (taking into account user preference

[1]To achieve an optimum, it is necessary to retain the set of points that are Pareto-optimal or dominant in the Cost-Resource-Quality dimensions. A complex optimization problem with backtracking has to be solved at run-time to achieve the optimum based on the Pareto-points.

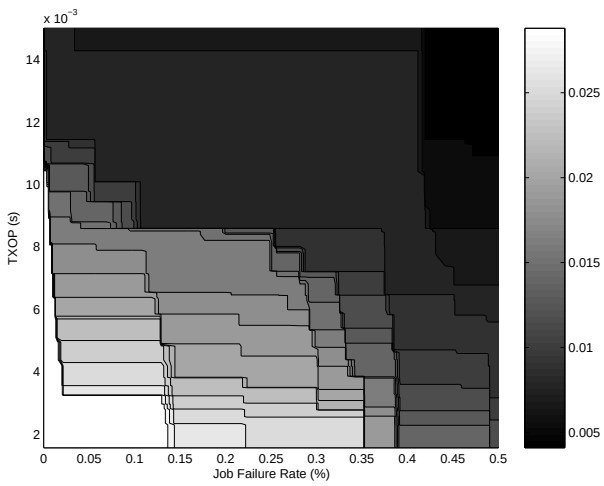


Fig. 3. Design space for job-failure-rate (JFR) vs. TXOP vs. energy consumption trade-off (contour plot). Energy per scheduling period can be reduced by scaling down (increase of TXOP) or by relaxing the performance constraint (JFR).

or weight if appropriate). This ensures that for every additional unit of resources consumed, the additional cost saving is the maximum across all flows taking into account the agreements made at admission time. We assume that the current channel states and application demands are known by each node and periodically communicated to the AP. When the node's state changes, the allocation is recomputed by the AP during the next scheduling instance. The exchange of state information and operating points between nodes and the AP is obtained by coupling the MAC protocol with the resource manager as explained in the next section. Our algorithm is based on Kuhn-Tucker [26, 28] and is described in detail in [29]. In our implementation, we sort the configuration points at design-time in the decreasing order of the negative slope between two adjacent points. The complexity of the runtime algorithm is $O(L.n.log(n))$ for n nodes (~ 20) and L configuration points per curve. In Section III, we demonstrate that for a practical system in each possible system state (i.e., channel and frame size), the number of configuration points to be considered at runtime is relatively small (~ 30). Taking into account that the relation $C_i(R_i)$ is convex, we can now prove that the greedy algorithm leads to the optimal solution for continuous resource allocation. Following that, we then extend the proof for systems with discrete working points to show that the solution is within a bound from the optimal.

Theorem 1: *For a continuous resource allocation to be optimal, a necessary condition is* $\forall i, 1 \leq i \leq n, R_i = 0$, *or for any flows* $\{i, j\}$ *with* $R_i > 0$ *and* $R_j > 0$, *the cost slopes* $C_i'(R_j) = C_j'(R_j)$.

Proof: For a continuous differentiable function, the Kuhn-Tucker [26] theorem proves such a greedy scheme is optimal. Suppose for some $i \neq j$, let the optimal resources be $R_i > 0$, $R_j > 0$, and $|C_i'(R_i)| > |C_j'(R_j)|$. As the savings in cost per resource for F_i is larger, we can subtract an infinitesimal amount of resource from F_j and add it to F_i. Total system cost is reduced, contradicting the assumed optimality. □

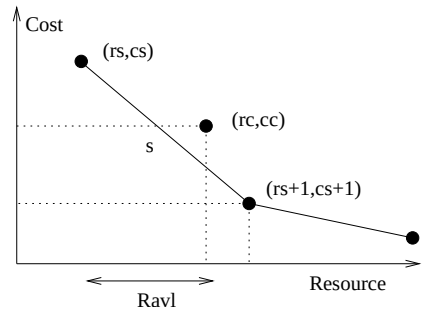


Fig. 4. Bounded deviation from the optimal in discrete Cost-Resource curves.

For a real system, however, the settings for different control dimensions such as modulation or transmit power are discrete. This results in a deviation, Δ, from the optimal resource assignment. We now show the worst-case deviation from the optimal strategy is bounded and small.

Theorem 2: $\exists 0 \leq \Delta < \infty$, *such that* $C_{OPT} \leq C_{MEERA} \leq C_{OPT} + \Delta$, *where* C_{OPT} *is the optimal cost (energy consumed by all users) and* C_{MEERA} *is the cost in the discrete case.*

Proof: For each flow, $\{F_1, F_2, \ldots, F_n\}$, the aggregate system resources consumed are stored in the decreasing order of their negative slope across all per-flow Cost-Resource $C_i(R_i)$ curves. Based on this ordering, the aggregate system $C(R)$ trade-off is constructed, consisting of segments resulting from individual flows. The greedy algorithm traverses the aggregate system $C(R)$ curve, consisting of successive additional resource consumptions (at maximum cost decrease), until the first segment, s, is found that requires more resource than the residual resource R_{avl} (Fig. 4).

Let the two end points of the final segment s be (r_s, c_s) and (r_{s+1}, c_{s+1}) in $C(R)$. Let (r_c, c_c) be the optimal resource allocation in the optimal combined Cost-Resource curve.

$$\begin{aligned} C_{OPT} &\geq C_{MEERA} - (r_c - r_s)(c_{s+1} - c_s)/(r_{s+1} - r_s) \\ &> C_{MEERA} - (r_{s+1} - r_s)(c_{s+1} - c_s)/(r_{s+1} - r_s) \\ &= C_{MEERA} - (c_{s+1} - c_s) \end{aligned}$$

We observe that $c_s - c_{s+1} \leq \Delta$, therefore $C_{MEERA} - C_{OPT} < \Delta$. Moreover, we note that a better approximation can be obtained when more dimensions ($K_{i,j}$) are considered.

III. 802.11 Transceiver Case Study

To demonstrate the usability of the proposed MEERA scheme, in this section, we apply it to control an Orthogonal Frequency Division Multiplexing (OFDM) 802.11a modem [2]. The goal is to show the progressive and substantial energy savings as each set of control dimensions is included and to discuss implementation aspects related to this specific instance of the methodology. The target application is the delivery of delay-sensitive traffic over a slow fading channel with multiple users. We associate the system *Cost* to *energy*, the *Resource* to the *time* over the shared medium and the *Quality* is the *JFR*.

We briefly consider the tradeoffs present across the physical layer circuits, communication settings and link layer in our

system. Increasing the modulation constellation size decreases the transmission time but results in a higher PER for the same channel conditions and PA settings. The energy savings due to decreased transmission time must offset the increased expected cost of re-transmissions. Also, increasing the transmit power increases the signal distortion due to the PA nonlinearity [33]. On the other hand, decreasing the transmission power also decreases the efficiency of the PA. Considering the tradeoff between sleeping and scaling, a longer transmission at a lower and more robust modulation rate needs to compensate for the opportunity cost of not sleeping earlier. Finally, as all users share a common channel, lowering the rate of one user reduces the time left for other delay-sensitive users. This compels other users to increase their rate and consume more energy or experience errors.

A. Top-Down QoS Driven Design

As the system performance requirements are specified at the application layer and the energy consumption is at the lower (hardware) layers, it is essential to: (a) translate the application layer requirements to relevant metrics at each intermediate layer and (b) to define clean interfaces between layers for an energy-performance feedback mechanism. This is to allow for a local calibration of the hardware which makes implementation more feasible, still enabling translation of to application specific quality metrics. For delay-sensitive traffic, the QoS metric of interest is the target JFR^*.

At the link-layer, each application *frame* is fragmented into one or more fixed-sized *fragments*. An application frame size or rate requirements is hence translated into a local Queue Size. Next, as shown in Fig. 5, the JFR^* is translated at the link layer to a PER constraint, which corresponds to a maximum Block Error Rate (BlER) as a function of the physical layer low-level knobs. Based on the performance of the turbo decoder, the BlER is a result of the receive SNR for given constellation order. The performance target of the system, on a frame-by-frame basis, is to ensure a probability that a frame is delivered (which is denoted as S_p^m, with m packets and p retransmissions). This is related to the user's QoS requirement by Eq.(1). This probability S_p^m is a function of PER and control dimension settings.

$$1 - JFR_p^m(K) = \sum_{j=0}^{p} S_j^m(K) \tag{1}$$

$$P = [1 - (1 - BlER)^{L_{frag}/288}] \tag{2}$$

Every target JFR may be satisfied by one or more control dimensions, $\mathbf{K}_i$, each of which is associated with the energy consumed (cost) and time required (resource) to complete the frame transmission in the current system state. The state is defined by a discrete channel state and traffic requirement (i.e. current application frame size), which can easily be monitored as the Queue Size. Channel classification and monitoring is typically a more difficult problem. At runtime, the current system state is periodically reported to the AP as described in detail at the end of this section. Based on a node's current system state and JFR^*, the corresponding Cost-Resource tradeoff curves are fetched from memory at the AP. From each curve, control dimension operating points are then chosen using the fast greedy algorithm to determine the current global Cost-Resource tradeoff such that the total transmission time for all nodes is less than the deadline.

TABLE II

PARAMETER VALUES USED IN OUR EXPERIMENTS

Performance Model	*Energy Model*	*MAC Model*	*Control Dimensions*
$W = 20MHz$	$P_{FE}^T = 200mW$	$L_{frag} = 1024B$	Back-off (dB) $\{6to16\}$
$B = 250kBaud$	$P_{FE}^R = 200mW$	$T_{ACK} = 52\mu s$	$P_{Tx}(dBm)$ $\{0to20\}$
$N = 48$	$P_{DSP}^T = 50mW$	$T_{PLCP} = 20\mu s$	Modulation $\{BPSK, QPSK, 16-QAM, 64-QAM\}$
$T = 198K$	$P_{DSP}^R = 50mW$	$T_{SIFS} = 16\mu s$	Code Rate $\{1/2, 2/3, 3/4\}$
$N_f = 10dB$	$E_{DSP}^R = 8.7nJ/b$	$Block = 288$	$JFR^* = 10^{-3}$

B. Design-Time Energy-Performance Tradeoff Calibration and Mapping

1) Energy Consumption per fragment: Four control dimensions have a significant impact on energy and performance for these OFDM transceivers: the modulation order (N_{Mod}), the code rate (B_c), the power amplifier transmit power (P_{TX}) and its linearity specified by the back-off (b). We consider the eight PHY rates supported by the 802.11a standard based on four modulation and three code rates (Table II). The bit rate (B_{bit}) for each modulation-coding pair with N_c OFDM carriers, N_{Mod} bits per symbol and symbol rate B is given by:

$$B_{bit} = N_c \times N_{Mod} \times B_c \times B \tag{3}$$

We focus on the power amplifier (PA) as it is generally the most power-hungry component in the transmitter consuming upwards of 600mW [32]. The PA efficiency is higher at high output power but due to non-linearity the signal distortion increases too. By throttling the bias current, the PA back-off controls the linearity and directly influences the energy consumption. The relation between the power amplifier back-off and the distortion, $D(b)$, has been characterized empirically for the Microsemi LX5506 [33] 802.11a PA. The PA power (P_{PA}) can be expressed as the ratio of the transmit power (PTx) to the PA efficiency (η_{PA}) that is related to b by an

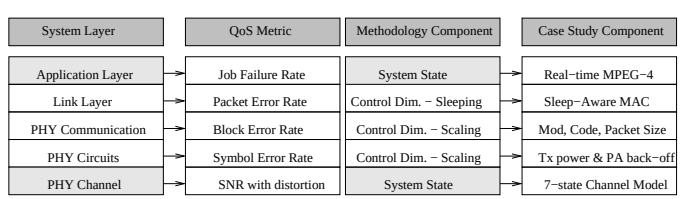


Fig. 5. MEERA cross-layer approach spans multiple layers with corresponding performance metrics. The case study describes a systematic and practical mapping of metrics across layers.

empirical law fitted on measurements:

$$P_{PA} = \frac{P_{Tx}}{\eta_{PA}(b)} \quad (4)$$

We assume the energy consumption of the digital baseband is a linear function of time and number of iterations for the turbo decoding at the receiver [34]. Based on current implementations [32], the frequency synthesizer, ADC, DAC, LNA and filters are assumed to have fixed front-end power consumption P_{FE} (Table II). We obtain the energy needed to send or receive a fragment of length L_{frag}:

$$E_{Tx} = (\frac{P_{PA} + P_{FE}^T + P_{DSP}^T}{B_{bit}}) \times L_{frag} \quad (5)$$

$$E_{Rx} = (\frac{P_{FE}^R + P_{DSP}^R}{B_{bit}} + E_{DSP}^R) \times L_{frag} \quad (6)$$

where P_{DSP}^T and P_{DSP}^R are the baseband processor's power consumption. Obtaining the actual values for energy consumption and PA non-linearity for each of the configurations only depends on the fragment size and not on system state. In practice, this information is obtained very fast by transmitting an L_{frag} packet once (requiring 0.1 to $1.3ms$ depending on the configuration) using each configuration (hence $1.3\times10^{-3}\times 10^6 = 1300s$ or approximately 20 minutes worst case for a complete system profile).

2) Traffic Requirements: We study the impact of both constant bit rate (CBR) and variable bit rate (VBR) traffic. For VBR traffic, we employ MPEG-4 encoded video traces [35] with peak-to-mean frame sizes ranging from 3 to 20. All fragmentation is done at the link layer and we use UDP over IP. As the maximum frame size is assumed to be within the practical limit of 50 fragments, we construct Cost-Resource-Quality curves for $1, 2, 3, 4, 5, 10, 20, 30, 40, 50$ fragments/frame and interpolate for intermediate values. The application layer frame size is translated to lower layers Queue Size and fragment size to facilitate state monitoring and calibration. As a result, no additional measurements are needed to model traffic requirements, which can be fully captured in the mapping.

3) Channel Constraints: We use a 7-state frequency-selective time-varying channel model to compute the PER for all transceiver settings. An indoor channel model based on HIPERLAN/2 [12] was used for a terminal moving uniformly at speeds between 0 to 5.2km/h (walking speed). Based on our turbo decoder design [34], we describe in detail the construction of a 7-state channel Markov-model in [29]. A set of 1000 channel realizations are used to determine the receive SNR with added distortion (SINAD) and extract the channel. The samples should be taken over a period larger than the coherence time of $166ms$ [36]. Assuming that samples should be taken over a large number of coherence times (e.g. 100), this will take $16.6s$ for channel extraction. Given output power P_{Tx} and channel attenuation A, the $SINAD$ is specified as:

$$SINAD = \frac{P_{Tx} \times A}{A \times D_i(b) + kT \times W \times N_f}, \quad (7)$$

where the constants k, T, W and N_f are the Boltzman constant, working temperature, channel bandwidth and noise figure of the receiver respectively. Based on the channel classification, the receive SINAD is mapped to one of 7 discrete channel states. Following this, we us the BlER models detailed in [37] to characterize (at design-time) the digital front-end for a range of $SINAD$. This calibration is a routine hardware characterization in current transceiver design practice. This BlER-$SINAD$ performance information is mapped to the set of channel states. This relation of dynamic channel state (based on hardware implementation) to performance metrics for higher layers is a key enabler for design-time Cost-Resource-Quality profiling.

4) Energy, Time and Quality Profile Mapping: For each combination of feasible control dimensions, $\mathbf{K}_i$ (which we will simplify to $\mathbf{K}$), we compute the total expected energy consumption, total transmission time and resulting JFR while using the properties of a sleep-enabled 802.11e MAC. We assume that during each communication instance, all transmissions use the same configuration to eliminate reconfiguration costs. All transmissions employ the contention-free mode with transmit opportunity (TXOP) grants of 802.11e Hybrid Coordination Function (HCF) [38]. A TXOP is defined as the time interval assigned when a user has exclusive channel access and is specified by a start time and duration. Let E_H, E_{ACK} and T_H, T_{ACK} be the constant energy and time needed to transmit a header and acknowledgement. The energy and time needed for a successful (i.e. good) and a failed[2] transmission is then determined using parameters listed in Table II:

$$E_{good}(\mathbf{K}) = E_K + E_H + (2T_{sifs}P_{idle}) + E_{ACK} \quad (8)$$

$$E_{bad}(\mathbf{K}) = E_K + E_H + (T_{sifs} + T_{ACK})P_{idle} \quad (9)$$

$$T_{good}(\mathbf{K}) = T_K + T_H + (2T_{sifs}) + T_{ACK} \quad (10)$$

$$T_{bad}(\mathbf{K}) = T_{good}(\mathbf{K}) - T_{sifs} \quad (11)$$

We now include the MAC layer retransmissions. Each fragment is transmitted with configuration $\mathbf{K}$, for which we can determine the P_K, based on Eq. (2). The probability that the frame is delivered with exactly $(m+p)$ attempts (including p retransmissions), is given by the recursion:

$$S_p^m(\mathbf{K}) = \sum_{i=1}^{min(m,p)} \binom{m}{i} \times (P_K)^i(1-P_K)^{(m-i)}S_{p-i}^i(\mathbf{K}) \quad (12)$$

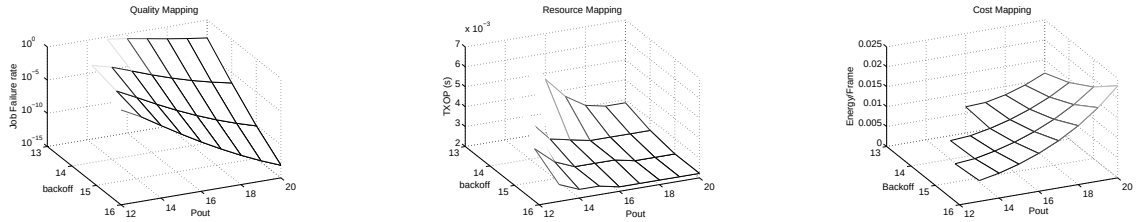
$$S_0^m(\mathbf{K}) = (1-P_K)^m \quad (13)$$

in which $\binom{m}{i}$ denotes the number of combinations to select i fragments out of m. The resulting probability to deliver a frame in terms of JFR is given by Eq.(1). Time and energy required is given by:

$$TXOP_p^m(\mathbf{K}) = [mT_{good}(\mathbf{K})] + [pT_{bad}(\mathbf{K})] \quad (14)$$

$$E_p^m(\mathbf{K}) = \sum_{j=0}^{p} S_j^m(\mathbf{K}) \times \{mE_{good}(\mathbf{K}) + jE_{bad}(\mathbf{K})\} + Z_p^m(\mathbf{K}) + H_p^m(\mathbf{K}) \quad (15)$$

[2]For a failed transmission, we wait the SIFS time and the time needed to decode the ACK. After that time we can be sure the ACK is not received.



(a) Quality Constraint. (b) Shared Resource. (c) Energy Cost.

Fig. 6. The Cost (Energy), Quality (JFR) and Resource (TXOP) mapping for the PA output power and back-off control dimensions for a fixed setting of the modulation and code rate control dimensions. Values shown for transmission of 10 frames in channel state 1.

The expected energy for a given configuration is the sum of the probabilities that the transmission will succeed after m good and j bad transmissions multiplied by the energy needed for good and bad transmissions. A second term $Z_p^m(\mathbf{K})$ should be added to denote the energy consumption for a failed job, hence when there are less than m good transmissions, and $(p+1)$ bad ones:

$$Z_p^m(\mathbf{K}) = JFR_p^m(\mathbf{K}) \times \{E_{bad}(\mathbf{K}) + \sum_{j=0}^{m-1} D_{p+1}^j(\mathbf{K}) \times \Big(jE_{good}(\mathbf{K}) + (P+1)E_{bad}(\mathbf{K})\Big)\}. \quad (16)$$

The third term $H(\mathbf{K})$ denotes the cost that has to be added once every scheduling period. We will show later that this cost corresponds to a wake-up cost only and no reconfiguration cost should be taken into account, where we assume that the cost for each configuration $\mathbf{K}$ is constant.

We determine the Energy-Time-JFR tradeoff as a function of the system state and number of retransmissions for each $\mathbf{K}$. This specifies the full profile for the system, and is determined only once during design or calibration time. The effect of the PA control knobs on the total expected cost, quality and resource is illustrated in Fig. 6. Next, the per-flow Energy-TXOP curves are determined from the $3D$ profiles of the system. Configuration points that do not meet the target JFR^* are pruned. Next, the convex minorant is computed in both Energy and TXOP dimensions. Segments in the intersection are kept and sorted for executing the run-time phase. Although this computation is relatively complex, it runs fast since the number of points is low (~ 30). It has to be executed once for the lifetime of the application, which is typically orders of magnitude larger than the scheduling period. This cost, resource and quality profile information is stored in each node's device driver.

C. Run-Time State Communication

Based on the Energy and TXOP curves for each node, the scheduler in the AP can efficiently derive a near-optimal resource allocation at run-time using the greedy scheme described in Section II. The scheduler requires feedback on the state of each user and then communicates the decisions to the users.

In order to instruct a node to sleep for a particular duration, the AP needs to know when to schedule the next packet. Waking a node earlier than the schedule instance will waste energy. Buffering just two frames informs the AP of the current and also the next traffic demand, allowing a timely scheduling and communication of the next period TXOP. In the ACK, the AP instructs the node to sleep until the time of the next TXOP and also the required configuration The AP now communicates with each node only at scheduling instances (Fig. 7). As the real-time packets are periodic, we eliminate all idle enegy consumption by sleeping between transmission instances.

When a node joins a network, it sends its cost, resource and quality curves (stored in its driver) to the AP during the association phase. The AP then stores this and refers to it during each scheduling instance. By adding just three bytes in the MAC header for the current channel state and the two buffered frame sizes, each node updates the AP of its current requirement in every transmission. Protocols such as 802.11e provide support for this communication and therefore require only minor modifications.

A software-based QoS Module within the AP's network management layer maintains the Cost-Resource curves of all associated nodes and processes the current state of each node. At the beginning of each period, it executes the run-time phase of MEERA and determines the configuration for each node during that period. The QoS Module fetches, from memory, the appropriate cost-resource curve corresponding to each node's current state. In the case study, this corresponds to less than 3000 bits per state. The scheduling period requirement is determined by the rate at which the system state varies. Channel measurements show coherence times of 166ms for stationary objects and moving scatterers [12]. Given a video frame rate of $30ms$, it is clear that this requires a scheduling period

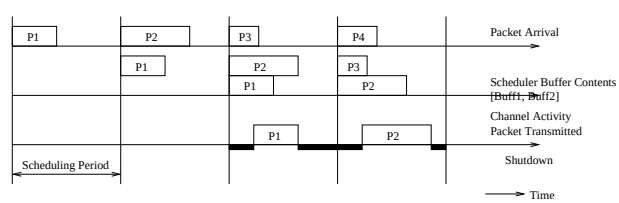


Fig. 7. MAC with two-frame buffering to remove data dependencies and maximize sleep durations. By the third period of the single flow shown, frames 1 and 2 are buffered and frame 1 begins service. As the transmission duration of frame 2 is known at this time, the sleep duration between completion of frame 1 until the start of service of frame 2 is appended in the MAC header.

less than $30ms$. Since this timing requirement is rather low, a QoS Module implemented in software suffices. Alternatively, the QoS Module can be integrated in a light-weight RTOS present in most embedded devices. We expect the performance of MEERA to be lower for mobile networks with faster state dynamics, when it is difficult to feedback the system state timely. Faster adaptation schemes are needed that integrating the adaptation module in hardware close to the physical layer [30].

IV. Numerical Results

We now illustrate the Energy-Performance tradeoff across a range of practical scenarios. The focus is on real-time streaming media applications with a reasonable target JFR^* set to 10^{-3}. In order to evaluate the relative performance of MEERA, we consider four comparative transmission strategies:

1) **Cross-layer:** This is the optimal scheme considering the energy tradeoff between sleep and scaling, exploiting multi-user diversity. The node configurations are based on the profiles described in Section III.
2) **PHY-layer:** This scheme considers only physical layer scaling knobs. The Energy-TXOP profiles are set to scale maximally as no sleep mode is available.
3) **MAC-layer:** In this scheme only sleeping is possible by the energy-aware MAC-layer. The physical layer is fixed to the largest constellation and code rate, with maximum transmit power. This approach is used by commercial 802.11 devices [8]. However, the 2-frame buffering makes the proposed implementation more efficient as it eliminates all idle energy consumption between transmissions.
4) **Fixed:** A basic scheme with no energy management and hence no dynamic range is exploited here. This is similar to the MAC-layer scheme but the transceiver remains in idle after transmission.

The Energy-TXOP profiles are computed for each scheme and used by the scheduling scheme implemented in the Network Simulator ns-2 [39]. This simulator has been extended with transceiver energy and performance models, and a slow fading channel model. Our simulation model implements the functions of the 802.11e with beaconing, polling, TXOP assignment, uplink, and downlink frame exchange, fragmentation, retransmission and variable super-frame sizing. In all results, the total energy consumed by a node is over a long duration to statistically capture the dynamics present in the scenario.

A. Impact of system state variations

Consider the scenario where a single user has to deliver a fixed size frame every scheduling period. In Fig. **??**, the energy consumption (normalized by the maximum energy consumed by Fixed) is plotted for the four schemes over different channel states. MEERA outperforms the other techniques in each state, since it takes advantage of the energy that can be saved by both sleeping and scaling. The energy needed to transmit the frame increases from best to worst channel state due to a combination of (a) the lower constellation needed to meet the PER (hence smaller sleep duration), (b) a higher required output power to account for the channel and (c) the increased cost of retransmissions. We observe, for example, that in the best channel state, the energy consumption is low for both the Fixed and MEERA approaches, and energy gains primarily result from sleeping. However, the transmission energy is more important in the bad channel stateand scaling becomes more effective. The ratio of fixed to scalable energy varies as transceivers are designed differently. MEERA takes this into account during the calibration of system profiles. For a given platform, as users demand different levels of QoS, MEERA jointly leverages the MAC and PHY to maximize the energy saved.

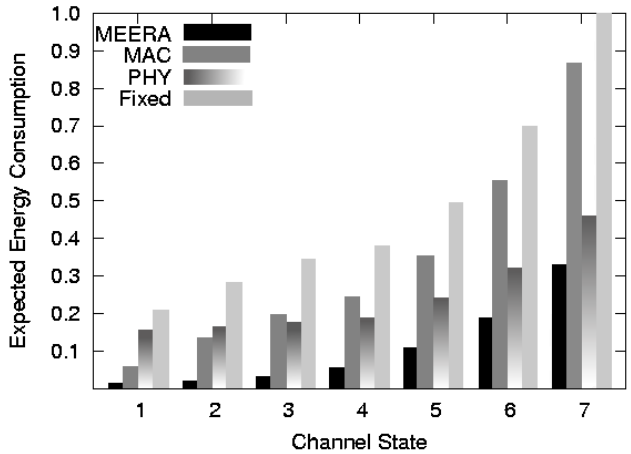


Fig. 8. Energy consumption across different channel states for 1 fragment.

B. Impact of multi-user diversity

We now consider a multiple-user scenario where the TXOP assignments are based on the user's application data-rate requirements and the constraints enforced by other users sharing the same link. In Fig. 9, we present simulation results

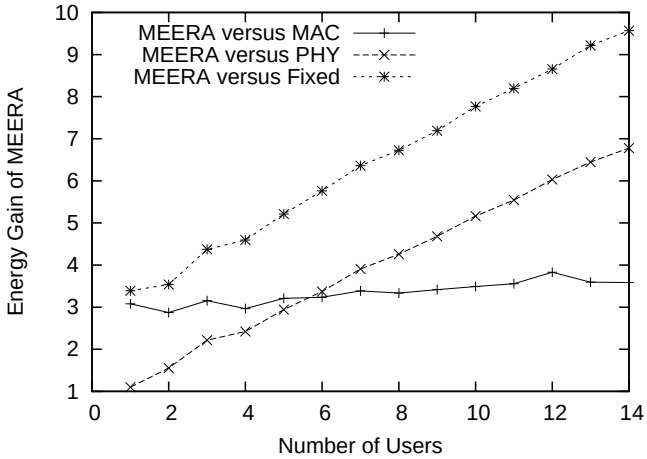


Fig. 9. Energy consumption per flow as a function of the aggregate system load for MPEG-4 traffic.

for MPEG-4 traffic with a peak-to-average frame size ratio of 7 over a time-varying channel. We study the impact of increasing the number of users (with an average aggregate load of 4Mbps) on the energy consumption. MEERA outperforms the MAC-based scheme by 300% as scaling is efficiently exploited due to multi-user diversity and the fading nature of the channel. Compared to the PHY-based scheme, MEERA gains by shutting down users between transmissions. For the considered case, MEERA increases system lifetime by a factor of 3-to-10 compared to traditional approaches.

V. Conclusion

In this paper, we presented a methodology for energy-efficient resource allocation, MEERA, to minimize energy consumption of a wireless transceiver while meeting the timeliness needs of multiple users. MEERA is a cross-layer optimization scheme that fully exploits the energy-performance tradeoffs between RF components, adaptive physical-layer schemes and a sleep-aware medium access control protocol. We have shown that MEERA's system-wide resource allocation consumes 3 to 10 times less energy than current schemes. These savings arise from two complementary contributions. First, we outlined a methodology that is platform-independent and provably near-optimal. We partition the combinatorial-explosive problem space into a well-specified design-time phase and a run-time phase which enables a practical approach where packet-scheduling decisions consider the users' throughput requirements, channel state and level of QoS demanded. The design-time phase derives an energy-performance representation for each user that captures the relevant tradeoffs. At run-time, a fast greedy algorithm selects operating points with a bounded worst-case deviation from the optimum. Secondly, we verified the performance of our scheme over a broad range of scenarios with delay-sensitive constant bit rate and MPEG-4 traffic over a time-varying channel using RF integrated circuit models. MEERA requires minimal modification to the 802.11 protocol while enabling significant energy savings.

References

[1] K. Lahiri, A. Raghunathan, S. Dey, D. Panigrahi *Battery-Driven System Design: A New Frontier in Low Power Design*, VLSI Design, Jan. 2002

[2] IEEE 802.11a, Part 11: *High-speed Physical Layer in the 5 GHz Band*, Supplement to IEEE 802.11 Standard, September 1999.

[3] S. H. Gunther, et al., *Managing the Impact of Increasing Microprocessor Power Consumption.* Intel Technology Journal, First Quarter, 2001

[4] Dell TrueMobile 1400 WLAN Card, 2004. http://support.ap.dell.com/docs/network/p44970/en/specs.htm

[5] Warwick, C.W. *Trends and limits in the "talk times" of personal communicators*, Proc. of the IEEE, 83(4):681-6, 1995.

[6] Robert A. Powers, *Batteries for low power electronics*, Proceedings of the IEEE, Vol. 83, No. 4, April 1995.

[7] Eugene Shih , Paramvir Bahl, Michael J. Sinclair, *Wake on wireless:: an event driven energy saving strategy for battery operated devices*, Conference on Mobile Computing and Networking, 2002.

[8] Atheros White Paper, *Power Consumption and Energy Efficiency Comparisons of WLAN Products*, 2003.

[9] A. El Gamal, et al. *Energy-efficient Scheduling of Packet Transmissions over Wireless Networks*, IEEE INFOCOM, Vol. 3, pp 1773-1783, 2002.

[10] C. Shannon, *A Mathematical Theory of Communication*, Bell System Tech. Journal, Vol. 27, pp. 379-423,623-656. 1948.

[11] Curt Schurgers, *Energy-Aware Communication Systems*, Ph.D. Thesis in Electrical Engineering, Univ. of California, L.A., 2002.

[12] European Telecom Standards Institute, *Channel Models for HIPERLAN/2 in Different Indoor Scenarios*, ETSI 3ERI085B 1998.

[13] A. Chandrakasan, S. Sherig, and R. Brodersen, *Low Power CMOS Digital Design*, IEEE JSSC, vol. 27, pp. 473-483, Apr. 1992.

[14] T. Simunic, L. Benini, P. W. Glynn, G. Micheli, *Dynamic power management for portable systems.* MOBICOM 2000

[15] C. Im, H. Kim, and S. Ha, *Dynamic Voltage Scheduling Technique for Low-Power Multimedia Applications using Buffers*, in Intl. Symp. On Low-Power Electronics and Design, 2001.

[16] D. Qiao, S. Choi, A. Soomro, and K.G. Shin, *Energy-Efficient PCF Operation of IEEE 802.11a Wireless LAN*, Proc. IEEE INFOCOM, 2002.

[17] S. Cui, A. J. Goldsmith, S. Verma, and A. Bahai, *Energy-constrained Modulation Optimization for Uncoded and Coded Systems*, ICC, 2003.

[18] B. Bougard, et al, *A New Approach for Dynamic Trade-off of Performance and Energy in Wireless Communication Systems.* SIPS, 2003.

[19] A.M. Eltawil , B. Daneshrad, *A low-power DS-CDMA RAKE receiver utilizing resource allocation techniques*, Solid-State Circuits, IEEE Journal of Volume 39, Issue 8, pp. 1321-1220, Aug. 2004.

[20] M. Anand, E. Nightingale, and J. Flinn, *Self-Tunning Wireless Network Power Management*, in ACM mobicom, 2003.

[21] R. Kravets and P. Krishnan, *Application-Driven Power Management for Mobile Communication*, WINET 2000.

[22] S. Chandra and A. Vahdat, *Application-specific network management for energy aware streaming of popular multimedia formats*, in USENIX Annual technical conference, 2002.

[23] R. Rajkumar, C. Lee, J. Lehoczky and D. Siewiorek *A Resource Allocation Model for QoS Management*, IEEE RTSS, 1997.

[24] S. Mohapatra, R. Cornea, N. Dutt, A. Nicolau, and N. Venkatasubramanian, *Integrated Power Management for Video Streaming to Mobile Devices*, in ACM Multimedia, 2003.

[25] X. Liang, S. Xuemin, J.W. Mark, *Fair resource allocation with guaranteed statistical QoS for multimedia traffic in wideband CDMA cellular network*, Mobile Computing, IEEE Transactions on Volume 4, Issue 2, pp. 166-177, March-April 2005.

[26] A. L. Peressini, R. E. Sullivan and J. J. Uhl, *Convex Programming and the Karish Kuhn-Tucker conditions*, Springer Verlag 1980.

[27] R. Mangharam, M. Demirhan, R. Rajkumar, D. Raychaudhuri, *Size matters: size-based scheduling for MPEG-4 over wireless channels*, SPIE Multimedia Computing and Networking Conference. Jan 2003.

[28] Chen Lee and Dan Siewiorek, *An Approach for Quality of Service Management* In Tech. Report CMU-CS-98-165, 1998.

[29] R. Mangharam, S. Pollin, B. Bougard, R. Rajkumar, F. Catthoor, L. Van der Perre, I. Moerman *Optimal fixed and scalable energy management for wireless networks*, IEEE Conference on Computer Communications (INFOCOM'05) , Miami, Fl, March 2005.

[30] M. Lacage, M. Hossein Manshaei, T. Turletti, *IEEE 802.11 Rate Adaptation: A Practical Approach*, ACM MSWiM 2004.

[31] T. H. Lee, *The Design of CMOS Radio-Frequency Integrated Circuits*, Cambridge University Press, 1998.

[32] M. Zargari, et al. *A 5-GHz CMOS Transceiver for IEEE 802.11a Wireless LAN*, IEEE JSSC, vol. 37, no. 12, Dec. 2002.

[33] Microsemi LX5506 InGaP HBT 4.5 6GHz Power Amplifier

[34] B. Bougard et al., *A scalable 8.7-nJ/bit 75.6-Mb/s Parallel Concatenated Convolutional Turbo Codec*, IEEE ISSC, Feb. 2003.

[35] F. Fitzek, M. Reisslein. *MPEG-4 and H.263 Video Traces for Network Performance Evaluation*, IEEE Network, 15: 40-54, 2001.

[36] V. Erceg et al., *TGn channel models*, IEEE 802.11-03/940r2, Jan 2004.

[37] JG Proakis, *Digital Communications*, New York: McGraw-Hill, 4th ed., 2001.

[38] IEEE 802.11 WG, Part 11: *Medium Access Control Enhancements for Quality of Service*, IEEE Std 802.11e/D4.0, Nov. 2002.

[39] *ns-2 Network Simulator*, http://www.isi.edu/nsnam/ns