A Framework for Energy-Scalable Communication in High-Density Wireless Networks

Rex Min and Anantha Chandrakasan
Massachusetts Institute of Technology, 50 Vassar St, Room 38-107
Cambridge, MA 02139
{rmin, anantha}@mtl.mit.edu

ABSTRACT

Power-aware communication is essential for maximizing the lifetime of energy-constrained wireless devices. Applications running on such devices can cooperatively reduce communication energy by trading communication latency, reliability, or range for energy savings. We introduce a framework that exposes these high level trade-offs to a power-aware communication subsystem featuring variable-strength convolutional coding, an adjustable power amplifier, and a voltage-scaled processor. An application programming interface (API) exposes an application's minimum quality constraints on the communication. These constraints are translated into energy-efficient parameter settings for the communication hardware. We apply our framework to improved communication energy models and measurements from a wireless microsensor node to effect over an order of magnitude of energy scalability.

Categories and Subject Descriptors

C.2.1 [**Network Architecture and Design**] — *wireless communication, network communications.* C.2.3 [**Network Operations**] C.4 [**PERFORMANCE OF SYSTEMS**]. D.2.2 [Design Tools and Techniques]

General Terms

Performance, Design, Reliability

Keywords

power awareness, energy scalability, wireless sensor networks, distributed microsensors, μAMPS, API design, dynamic voltage scaling, forward error correction, transmit power, macromodels, energy models

1. INTRODUCTION

The shrinking size and increasing density of wireless devices have profound implications for the future of wireless communication. Today's laptops and wireless phones may soon be outnumbered by ubiquitous computing devices such as microsensors, micro-robots, and "smart-dust" [11]. With smaller, more ubiquitous wireless elements come reduced battery capacities and unprecedented node density. Thus, there is an urgent need for hardware and software design techniques that encourage energy-efficient communication fabrics in ultra-high-density networks.

The distributed microsensor network [1] is an excellent example of a wireless network with high node density and an unprecedented demand to energy efficiency. Microsensor nodes are placed at high densities to provide both fault-tolerance and rich, high-resolution observations. Wireless communication enables inter-node collaboration and the transmission of data to a remote base station. Microsensor nodes are expected to operate from 5-10 years from an amount of energy equivalent to an "AA" cell [10], requiring careful attention to energy-efficient communication. These application characteristics of microsensor networks serve as a helpful guide and design driver for our work.

ISLPED'02, August 12-14, 2002, Monterey, California, USA.

Power aware hardware reacts gracefully to constantly changing operational demands. As performance demands increase or decrease, power aware hardware scales energy consumption accordingly to adjust its performance on-the-fly. Graceful energy scalability is highly desirable for any energy-constrained wireless node since the operational demands on a real-world node constantly change, and the peak performance of the node is rarely needed. Energy scalability is effected by key parameters that act as knobs to adjust energy and performance simultaneously.

Several such "knobs" are available in a power aware wireless communication subsystem. In general, the communication subsystem consists of digital processing for error correction and protocol handling, and a radio transceiver for the actual transmission and reception. The performance of digital processing can be adjusted with *dynamic voltage scaling*. Reducing processor voltage slows computation, permitting a graceful exchange of energy for latency. The workload on the processor can be varied through adaptive forward error correction (FEC) coding. "Stronger" codes that are more resilient to errors generally require more processing, and therefore more computation energy. The radio transceiver is energy-scalable as well; a an adjustable power amplifier in the transmitter allows energy to scale with transmission range. Voltage scaling, convolutional code strength, and radio transmission power are three crucial knobs for power awareness that will be increasingly present in modern wireless nodes.

While power-aware hardware is a significant stride toward energy-efficient communication, application designers that utilize wireless communication typically do not wish to concern themselves with low-level parameters such as processor voltage or transmit power. Since energy conservation is so crucial to wireless nodes, however, we must provide a way for the application to take advantage of hardware energy scalability. Figure 1 illustrates our approach. We introduce an application programming interface (API) and middleware layer that bridge the gap between these low-level "knobs" for energy scalability and performance metrics more relevant to an application. Performance metrics for communication are expressed through the API and translated by middleware into energy-efficient parameter settings for the communication hardware.

2. COMMUNICATION API

We define the *performance of communication* with four high-level parameters: *range*, *reliability*, *latency*, and *energy*. These parameters are an application's fundamental bases for specifying its com-

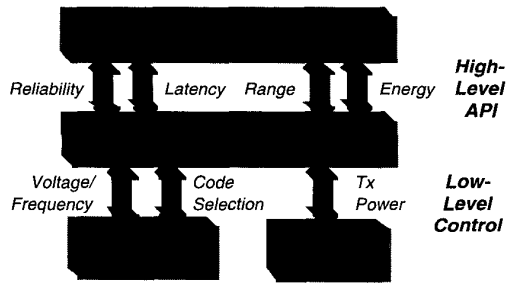

Figure 1: A power-aware interface bridges an application's quality requests for communication and the hardware's energy scalability.

munication needs. Using these parameters, we can define a minimal API for imposing bounds on these parameters.

The required range of a transmission is determined by its destination(s). To encourage an application to take advantage of range scalability, we allow it to specify the explicit destination or destinations of its communication, whether it be a unicast, multicast, or broadcast message.

With cooperation from a protocol layer that maintains approximate distances to—and numbers of—neighboring nodes, the communication range desired by an application can be expressed through four API calls:

- `set_destination(Node n)`
- `set_destination(Nodes n[])`
- `set_range(int numberOfNearestNeighbors)`
- `set_range(Distance d)`

The `set_destination` calls allow the range of a communication to be expressed in terms of its intended recipients. The unicast form directly implies a transmission range equal to the distance to the receiver. The multicast form, which eliminates the need for redundant unicast calls by the application, additionally implies a range that varies with direction, a fact that can be utilized by clever algorithms to generate asymmetric multi-hop routes.

The `set_range` calls allow an explicit specification of transmission radius and are especially useful primitives for broadcast. Range is specified either in terms of the number of nearest neighbors reached by the transmission, or as an outright distance in meters.

The remaining parameters of latency, reliability, and energy should be boundable by the application. This is achievable through the following calls:

- `set_max_latency(double usecs)`
- `set_min_reliability(double ber)`
- `set_max_energy(double ujoules)`

Any or all bounds can be specified, but an appropriate exception is thrown if the combination of bounds requested exceeds the capability of the system. To guide the user, reasonable defaults and parallel `get` calls to fetch property values should be provided.

3. COMMUNICATION ENERGY MODELS

With the hardware's energy scalability identified and application-level parameters defined, we now model the energy consumed in a point-to-point wireless transmission.

3.1 Radio Transmission Energy

Ultra-low-energy systems typically transmit at a low duty cycle: the transmitter electronics are only powered occasionally when a burst of data is ready for an immediate departure. The energy required to transmit a single burst of data from an initially powered-down transmitter can be expressed as follows:

$$E_{tx}(N, R_c, P_{amp}) = P_{start}T_{start} + \frac{N}{R_c R}(P_{txElec} + P_{amp}) \quad (1)$$

The two terms in the expression represent the energies of startup and transmission respectively. P_{start} and T_{start} represent the power and latency of radio startup [4], P_{txElec} the active transmission power, P_{amp} the *dissipated* amplifier power (not the power radiated), N the number of data bits before FEC, R the radio bit rate, and R_c the convolutional code rate. Typically, $P_{start} < P_{rxElec}$, since only the VCO and PLL require a settling time T_{start}. The remainder of the transmission circuit, such as the power amplifier, starts quickly enough to be omitted from the startup energy term.

In (1), the energy of transmission is expressed as a function of the low-level parameters R_c and P_{amp}. Our goal now is to relate these values to the application level parameters for range and reliability: the transmission distance d and bit error rate P_b. We achieve this by first relating P_b to the power incident at the receive antenna, and then computing the transmit power required to attain this receive power over a distance d.

An explicit relation between P_b and received power under Rayleigh fading and convolutional coding is well beyond the scope of this paper. Simulations of convolutional codes' performance under Rayleigh fading [7], plotted in Figure 2, are regressed into a function for the received power P_{rcvd} needed to achieve a bit error rate P_b. We denote each function $P_{rcvd}(P_b,R_c,K_c)$. Since R_c and K_c together identify a distinct convolutional code in this work, P_{rcvd} can be viewed as an array of functions on P_b, with R_c and K_c serving as indices to identify the particular BER-to-receive power relationship for each code.

Computing the total amplifier power at the transmitter required to achieve P_{rcvd} at the receiver requires consideration of the path loss and amplifier inefficiencies:

$$P_{amp}(d) = \alpha_{amp} + \beta_{amp}P_{1mAtt}d^n P_{rcvd}(P_b, R_c, K_c) \quad (2)$$

where d is distance, α_{amp} and β_{amp} parameters representing the linearized efficiency of the power amplifier, P_{1mAtt} the attenuation

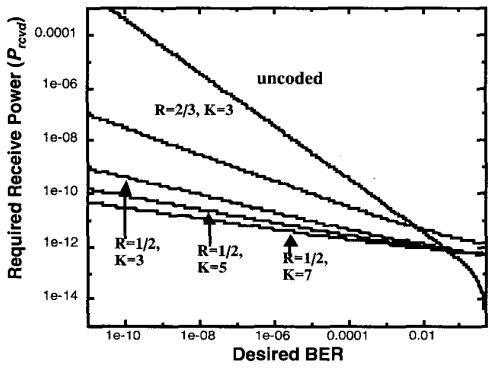

Figure 2: Receive power P_{rcvd} required to attain a desired bit error rate for various coding schemes, using performance parameters of a commercial radio [7].

at one meter from the sender including antenna effects, and n the path loss index. The following results are based on $P_{1mAtt} = 30$ dB and $n = 3.5$. In practice, P_{1mAtt} and n vary heavily by environment. Substituting (2) into (1) yields a complete expression for $E_{tx}(P_b,d,N)$, the energy in terms of reliability and range, for each code (R_C, K_C):

$$E_{tx}(P_b, d, N) = P_{start}T_{start} + \frac{N}{R_c R}[P_{txElec} + (\alpha_{amp} + \beta_{amp}P_{1mAtt}d^n P_{rcvd}(P_b, R_c, K_c))] \quad (3)$$

3.2 Decoding and Receive Energy

The energy required to receive a packet is the sum of the energies dissipated by radio startup, the active receiver electronics, and the digital circuits used by Viterbi decoding:

$$E_{rx}(N, R_c, E_{decbit}) = P_{start}T_{start} + P_{rxElec}\frac{N}{R_c R} + E_{decbit}N \quad (4)$$

E_{decbit} is the decoding energy per information bit; the other parameters have been introduced in the previous section.

The decoding energy E_{decbit} models the energy consumed by the Viterbi algorithm on digital hardware. The energy consumed per bit is expressed as the sum of digital switching and leakage energies [8]:

$$E_{decbit}(V_{DD}, C_{bit}, T_{bit}) = C_{bit}V_{DD}^2 + T_{bit}\left(V_{DD}I_0 e^{\frac{V_{DD}}{nV_T}}\right) \quad (5)$$

C_{bit} is the switched capacitance per bit, V_{DD} the supply voltage, which is adjustable through dynamic voltage scaling, and T_{bit} the computational time required per bit. I_0 and n, which model digital leakage current, are functions of the process technology. V_T is the thermal voltage.

C_{bit} and T_{bit} are themselves functions of the convolutional code and the use of dynamic voltage scaling. The computational workload of Viterbi decoding is exponential with the constraint length K_c [7] such that

$$C_{bit}(K_C) = C_0\alpha_c^{K_c} \quad (6)$$

and

$$T_{bit}(f, K_C) = T_0\alpha_t^{K_c}\frac{f_{max}}{f} \quad (7)$$

where f_{max} represents the maximum clock frequency, and f represents the actual frequency which may have been reduced due to dynamic voltage scaling. The constants C_0, α_c, T_0, and α_t can be regressed for the hardware being modeled; note that C_0 and T_0 will vary by orders of magnitude depending on the implementation fabric (i.e., ASIC versus microprocessor).

To first order, the relation between core voltage V_{DD} and frequency f is linear, arising from the roughly inverse relationship between and circuit latency.

$$f(V_{DD}) = K_{proc}(V_{DD} - c_{proc}) \quad (8)$$

where the constants K_{proc} and c_{proc} adjust the slope and intercept. Now, given a latency constraint of T on the decoding of N useful bits of data, we can solve for the required V_{DD}.

$$V_{DD}(K_C, T) = \frac{f_{max}NT_0\alpha_t^{K_C}}{K_{proc}T} + c_{proc} \quad (9)$$

Substituting (6),(7), and (9) into (5) provides $E_{decbit}(T)$, the decoding energy per useful bit in terms of a latency constraint on coding. Assuming an expected MAC overhead time T_{mac} for the data link, we substitute $E(T_{tot} - T_{mac}, K_C)$ into (5) to yield $E_{rx}(T_{tot}, N, R_c, K_c)$, the receive energy as a function of the tolerable latency T_{tot} over the link:

$$E_{rx}(T_{tot} - T_{mac}, N) = P_{start}T_{start} + N\left(\frac{P_{rxElec}}{R_c R} + C_0\alpha_c^{K_c}V_{DD}^2 + (T_0\alpha_t^{K_c})\frac{f_{max}}{f}V_{DD}I_0 e^{\frac{V_{DD}}{nV_T}}\right)$$

with V_{DD} defined in (9) above.

4. POWER-AWARE MIDDLEWARE

The energy models derived in Section 3 characterize the system energy required to attain the performance dictated by the API of Section 2. These expressions dictate the operational policy of the power aware middleware layer that translates the API calls for communication performance bounds into the minimum-energy hardware settings that meet those bounds. We now provide a concrete example of a real-world middleware policy by evaluating our energy models with measured parameters for the µAMPS-1 microsensor node [7, 8, 9]. µAMPS-1 is the first prototype node for the MIT µAMPS project (Adaptive Multidomain Power-Aware Sensors) [4] which is developing a power aware hardware and software foundation for microsensor nodes. Figure 3 illustrates the architecture of µAMPS-1. The parameter values extracted from µAMPS-1 are listed in Table 1.

µAMPS-1 supports the hardware "knobs" of processor voltage scaling, variable FEC, and adjustable radio power. The processing subsystem, implemented primarily by a SA-1110 low-power microprocessor, is customized to support dynamic voltages scaling from 0.9-1.5 V. Forward error correction, implemented through convolutional encoding and Viterbi decoding at the SA-1110, is scalable by altering the constraint length and puncturing of the base code. (Note that an FPGA or dedicated hardware would be a lower-energy solution.) The radio subsystem consists of a 2.4 GHz, 1 Mbps FSK transceiver with time-division media access and features two power amplifiers for low (+0 dBm) or high (+20 dBm) power transmission.

4.1 Node-to-Base Station Communication

When a node is communicating with an energy-unconstrained base station, only the energy of transmission need be considered. Given d and P_b, we can evaluate (3) over all codes supported by the node

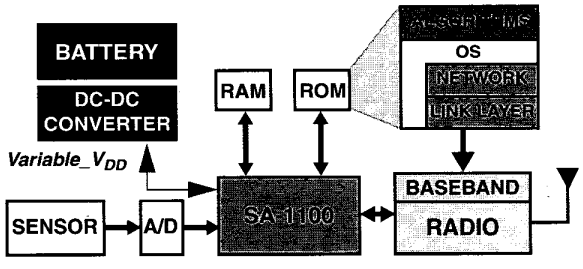

Figure 3: Architecture of the µAMPS-1 microsensor node.

TABLE 1 Model parameters and values for Sections 3 and 4

Symbol	Description	Value in Section 4
α_{amp}	amplifier inefficiency, constant term	174 mW
α_c	switched cap/bit, exponential base	2.62
α_t	decode time/bit, exponential base	2.99
β_{amp}	amplifier inefficiency, linear coeff.	5.0
C_0	switched cap/bit, linear coefficient	51.6 nF
c_{proc}	processor f,V_{DD} relation, const. term	659 mV
d	transmission range	0-100 m
f	processor frequency	59-206 MHz
f_{max}	maximum processor frequency	206 MHz
I_0	proc. subthresh. leakage, linear coeff.	1.196 mA
K_C	conv. code constraint length	3, 5, 7
K_{proc}	processor f,V_{DD} relation, linear coeff.	245 MHz/V
N	bits per transmission	1000
n	path loss exponent	3.5
n_0	subthresh. leakage, exponential term	21.26
P_{1mAtt}	path loss, one-meter attenuation	30 dB
P_{amp}	transmit amplifier power	179, 674 mW
P_b	bit error rate (BER) at receiver	10^{-3} to 10^{-9}
P_{rxElec}	receive electronics power	279 mW
P_{start}	radio startup power	58.7 mW
P_{txElec}	transmit electronics power	151 mW
R	radio transmit data rate	1 Mbps
R_C	convolutional code rate	1/2, 2/3
T_0	decode time/bit, linear coefficient	219 ns
T_{start}	radio startup time	470 µs
V_{DD}	processor supply voltage	0.9-1.5V
V_T	thermal voltage (room temperature)	26 mV

and choose the lowest-energy code. Back substituting the selected code (R_C, K_C) into (2) provides the required amplifier power setting. As convolutional encoding is a trivial computation, and the data rate for µAMPS-1 is high compared to the expected media access delay T_{mac}, latency constraints are decoupled from the hardware operational policy. We have plotted the least-energy coding and transmission power policies in Figure 4 for transmission of a 1000-bit packet to a base station, As the evaluated version of µAMPS-1 supports only two power levels, range and reliability can be increased with greater energy-efficiency by lowering the code rate and prolonging the transmit time, rather than switching to a higher power amplifier.

4.2 Node-to-Node Communication

Summing the receive and transmit energies E_{rx} and E_{tx} defined in and (3), provides the total energy of communication between two energy-constrained wireless nodes:

$$E_{tot}(P_b, T_{tot}, d, N) = E_{rx}(T_{tot} - T_{mac}, N) + E_{tx}(P_b, d, N) \quad (10)$$

The minimum energy operational policy is selected by choosing the least-energy code (R_C, K_C) from (10), sufficient amplifier power P_{amp} from (2), and the decoding processor voltage V_{DD} from (9), such that all performance constraints are met.

Figure 5 evaluates (10) to illustrate range scalability through the variation of transmit power and convolutional coding scheme. Both receive and transmit energy are considered for a 1000-bit packet (N=1000), a BER constraint of P_b=10^{-5}, and no latency constraint. As the required range increases, the transmit power amp is switched from low to high power and higher-rate (and higher K_c) codes are applied. Varying both parameters together has a dramatic impact on range scalability: for communication under 100 meters, the total system energy is less than linear with distance. A judicious middleware policy that accounts for the energy consumption characteristics of the hardware ensures that communication range does *not* scale as a power law with distance.

Figure 6 extends this analysis to two dimensions, illustrating the least-energy coding scheme given a choice of d and P_b, under no latency constraint. These parameter selections are appropriate for communication between two energy-constrained devices. Due to the high processing energy of µAMPS-1, it is now more desirable to utilize high-power transmission instead of convolutional coding

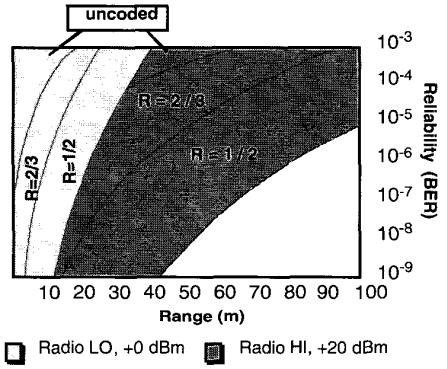

Figure 4: Least-energy hardware (transmit power and code rate) policy for single-hop communication given a specified reliability and range, considering *only the energy of transmission*. K_c = 3 for coded communication. N = 1000.

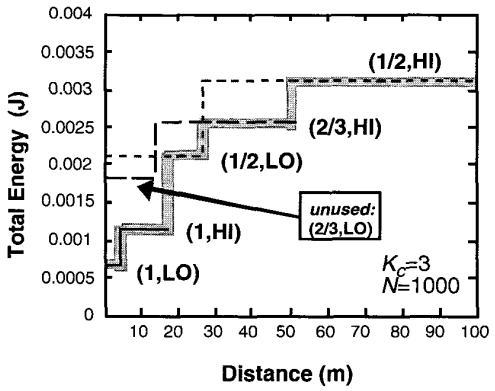

Figure 5: As communication distance increases, the operational policy (R_c,P_{tx}) is continuously adjusted for minimum energy. For a target BER of 10^{-5}, the least-energy policy for each d is shaded above. Note that our code and transmit power variation removes path loss effects at these distances.

when additional performance is needed for node-to-node communication. The choice between radio and processor scalability is hardware-dependent and highlights the importance of building detailed energy models for an energy-efficient operational policy. Figure 7 illustrates the total network energy (sum of the energies dissipated by the sender and receiver) consumed for the modes selected in Figure 6. In short, energy is scalable over nearly two orders of magnitude, realizing range scalability to well over 100 meters and BER scalability across several decades.

In the previous discussion, no latency constraint is considered, and therefore the processor is run at its lowest operating frequency of 59 MHz. Figure 8 illustrates the impact of latency scalability on energy consumption. Viterbi decoding on the SA-1110 takes a fair amount of time; expanding that allowable time allows the SA-1110 to be run at a reduced frequency and voltage. For K_c = 7, relaxing the latency constraint to four times its minimum value enables a 60% energy savings for the *entire* communication. The leftmost point on each curve is the minimum possible latency due to processor limitations; the jaggedness of the tradeoff curves is due to the eleven discrete voltage/frequency pairs supported by the SA-1110.

It is instructive to consider the impact of alternate hardware on the middleware policy. μAMPS-1 utilizes a 2.4 GHz radio; radios in lower frequency bands tend to consume less energy and have a lower data rate R. As a result, the latency and energy penalties of coding would become more significant, and a N/R term (which was previously neglected) would be added to the latency models.

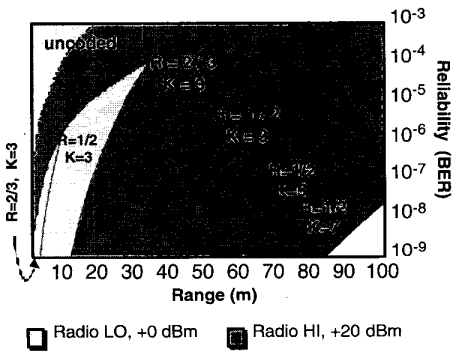

Figure 6: Least-energy hardware policy for single-hop communication given a specified reliability and range, considering both transmit and receive energy. N = 1000.

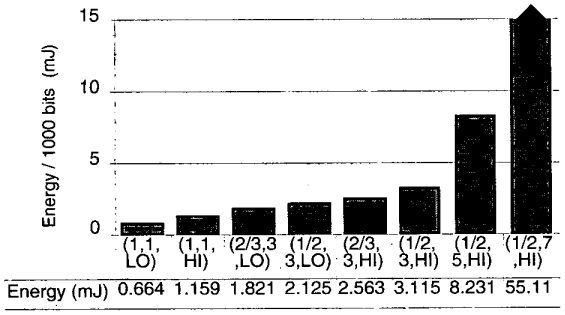

Figure 7: Energy (in millijoules) to transmit a 1000-bit packet using the transmit power and coding combinations (R_c,K_c,P_{tx}) of Figure 6. (Note that the last bar exceeds the dimensions of the graph.)

μAMPS-1 utilizes a general-purpose processor for Viterbi decoding. A dedicated ASIC for Viterbi decoding would decrease time and energy substantially, perhaps allowing us to neglect E_{decbit} and T_{bit} entirely. As a result, the node would favor stronger FEC coding, and the influence of the radio and MAC would grow.

5. REAL-WORLD ROUTING SCENARIOS

So far we have only considered the energy of a single point-to-point transmission. We now interpret these results in the context of two larger examples, data aggregation and multihop routing. Guided by the preceding results, our application-level view of communication exposes additional overhead in multihop routing and aggregation that has been overlooked in the past.

Data aggregation [9] fuses observations from S nodes, each of length N, into a single, high-quality data stream of length N. Hence, as depicted in Figure 9a, aggregation reduces the number of bits forwarded by the aggregating node from NS to N, suggesting an immediate and substantial energy savings due to reduced transmission and reception time. From an application's point of view, however, aggregated data is presumably of higher value than a single stream of data. Therefore, we would expect the application

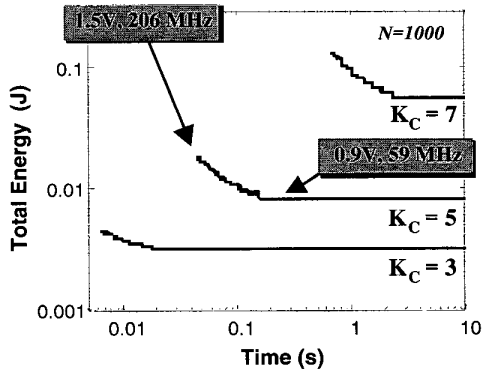

Figure 8: Dynamic voltage scaling enables latency scalability for Viterbi decoding. For K_c=7, the most energy-intensive code considered, extending the latency deadline by a factor of four allows a 60% total communication energy savings.

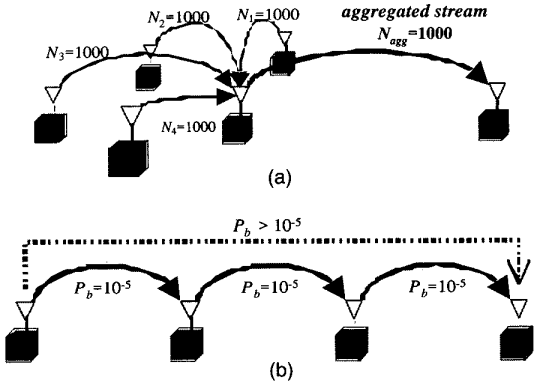

Figure 9: (a) *Data aggregation* fuses the observations from multiple sensors into a single, high-quality stream. (b) *Multihop routing* linearizes path loss by breaking a long-range transmission into several, shorter hops.

to demand a higher reliability bound when aggregated data is being forwarded—perhaps P_b/S. Hence, the operating point in Figure 6 moves vertically downward, and the total energy consumed for the transmission changes from $_{tot}(P_b/S, T_{tot}, d, NS)$ to $E_{tot}(P_b/S, T_{tot}, d, N)$. The energy savings from aggregation of S data streams is offset by any additional energy required to increase link reliability. For instance, if ten data streams of 1000 bits each are being transmitted over d = 30 meters with reliability $P_b = 10^{-4}$, then the least-energy transmission policy is HI radio power with no coding, resulting in E_{tot} = 11.1 mJ. Aggregating the ten streams into one increases the reliability requirement to $P_b = 10^{-5}$. The least-energy coding policy is now R_c=2/3, K_c=3, resulting in E_{tot} = 2.56 mJ. Although the transmission is 10% of its original length, the energy consumed is 23% of its original value before aggregation. Processing energy for the aggregation algorithm would further increase this figure.

Multi-hop routing [5], illustrated in Figure 9b, sacrifices additional nodes' receive energy E_{rx} in exchange for a reduction in transmission energy E_{tx}. There are two reasons why multihop routing is less desirable under our framework. First, both the end-to-end BER and latency grow with the number of wireless hops, so that the total energy consumed by an h-hop transmission becomes roughly $E_{tot}(P_b/h, T_{tot}/h, d/h, N$. (Obtaining the exact relationship between hop count and end-to-end P_b is rather involved; P_b/h is offered for illustrative purposes.) If an application demands an upper bound on either parameter, a multihop scheme must tighten the bounds for the individual hops, potentially requiring a more energy-intensive operational policy as with our aggregation example above. Second, our distance-scalable communication policy already mitigates transmission path loss over moderate distances, as seen by the energy *vs.* distance curve of Figure 5. Hence, a 60-meter transmission under the assumptions of Figure 5 consumes 3.115 mJ, while four 15-meter "hops" require a total of 4.636 mJ, even before any reliability constraints are considered. Multihop provides no energy benefits for µAMPS-1 until the desired transmission range exceeds the radio range at +20 dBm with a R_c=1/2, K=3 convolutional code at d > 100 m.

6. CONCLUSION AND FUTURE WORK

The four parameters that define communication to an application are range, reliability, latency, and energy. These high-level parameters, however, are somewhat removed from the actual hardware "knobs" that allow energy to scale gracefully with performance. In a communication subsystem, these knobs include the amplifier's transmit power, the convolutional code, and processor voltage.

We have linked these hardware knobs to application-level parameters in two ways: first, through a basic API that exposes an application's explicit performance requirements for communication, and second, through a comprehensive fusion of mathematical relations grounded in hardware but capturing communication energy as a function of application-level parameters alone. Dynamic voltage scaling for the node's processor effects end-to-end latency scalability in communication, and code and transmit power variation together enable scalability in reliability and range. Power-law path losses are mitigated to the point that the need for multihop routing is dramatically reduced.

While we have chosen the µAMPS-1 sensor node as our application example and the source of hardware parameter measurements, our methodology is easily adaptable to new hardware. Our framework is extensible to any additional sources of energy scalability in the hardware, such as the key length of data encryption [2], voltage overscaling [3], and the variation of M in M-ary modulation [6,10]. Communication techniques such as ARQ and alternate codes can be modeled as well. Such extensions are simply additional "knobs" that can be incorporated into the model by solving for their impact on communication range, reliability, and latency. Considering the energy consumption of communication in these terms yields new insight into the efficacy of other communication energy reduction techniques such as multihop and data aggregation.

It is our hope that the bridging of low-level hardware hooks and communication software will catalyze energy-efficient protocol design and spur further development of power aware API and middleware layers. As we expect hardware to become increasingly energy scalable, the algorithms used by power management layers, and the functionality they expose to an application, will be instrumental to the success of power-aware communication.

7. ACKNOWLEDGMENTS

The authors thank Seong-Hwan Cho, Raul Blasquez-Fernandez, and Alice Wang for helpful discussions and superior insight; Ben Calhoun, Nathan Ickes, Theodoros Konstantakopoulos, Fred Lee, and Piyada Phanaphat for their work on µAMPS-1; and the anonymous reviewers for their detailed suggestions. This research is sponsored by the Defense Advanced Research Project Agency (DARPA) Power Aware Computing/Communication Program and the Air Force Research Laboratory, Air Force Materiel Command, USAF, under agreement number F30602-00-2-0551; by the Army Research Laboratory (ARL) Collaborative Technology Alliance, through BAE Systems, Inc. subcontract RK7854; and by Hewlett-Packard under the HP/MIT Alliance. Rex Min is funded by an NDSEG Fellowship.

8. REFERENCES

[1] Asada, G. et al. Wireless Integrated Network Sensors: low power systems on a chip. Proc. ESSCIRC '98 (Sept. 1998).

[2] Goodman, J. and Chandrakasan, A. An energy-efficient IEEE 1363-based reconfigurable public-key cryptography processor. IEEE ISSCC 2001 (San Francisco, Feb. 2001), pp. 330-331, 461.

[3] Hedge, R. and Shanbhag, N. A low-power digital filter IC via soft DSP. Proc. CICC 2001, pp. 309-312.

[4] Min, R. et al. Low-Power Wireless Sensor Networks. Proc. 14th Int'l Conf. on VLSI Design (Bangalore, India, Jan. 2001), pp. 205-210.

[5] Royer, E. and Toh, C-K. A review of current routing protocols for ad hoc mobile wireless networks. IEEE Personal Communications, 6, 2 (April 1999), pp. 46-55

[6] Schurgers, C.; Aberthorne, O.; Srivastava. Modulation Scaling for Energy Aware Communication Systems, Proc. ISLPED 2001, pp. 96-99.

[7] Shih, E. et al. Physical layer driven algorithm and protocol design for energy-efficient wireless sensor networks. Proc. MobiCOM 2001 (July 2001).

[8] Sinha, A. and Chandrakasan, A.JouleTrack—a Web based tool for software energy profiling. Proc. DAC 2001, pp. 220-225.

[9] Wang, A. and Chandrakasan, A. Energy efficient system partitioning for wireless sensor networks. Proc. ICASSP 2001.

[10] Wang, A. et al. Energy efficient modulation and MAC for asymmetric RF microsensor systems. Proc. ISLPED 2001, Aug. 2001, pp. 106-111.

[11] Warneke, B.; Atwood, B.; Pister, K.S.J. Smart dust mote forerunners. Proc. 14th IEEE Int'l Conference on MEMS (Jan. 2001), pp. 357-360.