Power Management for Bluetooth Sensor Networks

Luca Negri[1] and Lothar Thiele[2]

[1] Politecnico di Milano, Dept. of Electrical Eng. and Comp. Science (DEI)
[2] ETH Zurich, Computer Eng. and Networks Lab (TIK)

Abstract. Low power is a primary concern in the field of wireless sensor networks. Bluetooth has often been labeled as an inappropriate technology in this field due to its high power consumption. However, most Bluetooth studies employ rather over–simplified, fully theoretical, or inadequate power models. We present a power model of Bluetooth including scatternet configurations and low–power sniff mode and validate it experimentally on a real Bluetooth module. Based on this model, we introduce a power optimization framework employing MILP (Mixed–Integer Linear Programming) techniques, and devise optimal power management policies in the presence of end–to-end delay constraints. Our optimizations, if backed by power–aggressive hardware implementations, can make Bluetooth viable for a wider range of sensor networks.

1 Introduction

Low power consumption has always been a primary goal in the design of wireless sensor networks. Moreover, communication accounts for a relevant power contribution on sensor nodes, fact that is true even for more complex mobile systems [1]. Communication among sensors can be implemented with custom solutions or standardized radio interfaces. If on one hand custom solutions carry the greatest power optimization potential, the choice of widespread wireless communication standards guarantees interoperability as well as ease of connection with existing commercial devices.

Communication protocols are often endowed with power/performance tradeoffs, which can be used to match application requirements with power consumption; such protocols are known as *power–aware protocols* [2]. Power–aware is a broad term and may denote a protocol that (i) manages power/performance tradeoffs offered by lower layer protocols, (ii) exposes further tradeoffs to the layers above or (iii) both. Examples of power/performance tradeoffs are modulation scaling [3] and power control [4] at the physical layer, Bluetooth's low–power modes (hold, sniff, park) and WiFi's Power Save Protocol (PSP) at the MAC layer. In order to exploit these features an abstraction describing power behavior on tradeoff curves is required: such abstraction is called a *power model*.

Bluetooth (BT) is a leading standard for short–range ad–hoc connectivity in the Personal Area Networks (PAN) field. Although initially designed for simple point–to–multipoint cable replacement applications, Bluetooth has proved

K. Römer, H. Karl, and F. Mattern (Eds.): EWSN 2006, LNCS 3868, pp. 196–211, 2006.

very appealing also to build multi–hop ad–hoc networks (called *scatternets*) [5] and even for high bandwidth sensor networks [6]. BT provides low–power modes (hold, sniff, park) which trade throughput and latency for power. We believe these features, if backed by a power–aggressive hardware implementation, can make the protocol fit for a wider range of sensor networks, allowing for an appropriate tuning of communication performance (and power) on the application requirements. Nevertheless, to achieve this, a power model describing all possible states (number of links, low–power mode of each link etc.) is necessary.

There is indeed a lack of such a model in the literature to date. Many Bluetooth power–optimization proposals, such as [7] and [8], are based on over–simplified power models, not considering number and role (master vs. slave) of links. Also, such models are normally not based on experimental measurements, but rather on theoretical assumptions. Other BT–related studies employ rather old and inadequate power models that were derived for other wireless systems [9]. Finally, the few power measurements for Bluetooth in the literature (see [10] and [6]) do not cover low–power modes and scatternet configurations.

In this paper we describe a full power model of Bluetooth in a complex *scatternet scenario* where each link can be in active or low–power *sniff* mode. The model is experimentally characterized and validated (RMS error below 4%) for the *BTnode*, a BT–based ad–hoc network prototyping platform developed at ETH Zurich [11]. We employ then the power model to build a flexible power optimization framework based on Mixed–Integer Linear Programming (MILP), which can be used to solve a number of power vs. Quality of Service (QoS) problems. In particular, in this paper we focus on the *power/delay* tradeoff offered by BT's *sniff mode* and determine the best network configuration that grants the lowest power consumption while meeting given end–to–end delay requirements. Such a *power management* policy can be either centrally determined and distributed to all nodes when needed or pre–computed for different requirements sets and stored in the nodes themselves.

After Section 2 briefly introduces Bluetooth and describes the BTnode platform, the main contributions of this work are presented: the power model of BT in Section 3, the power/delay problem in Section 4 and its solution for selected topologies in Section 5. Section 6 concludes the paper, outlining further possible usages of the optimization framework and its re–application to different protocols and scenarios.

2 Bluetooth and the BTnode Platform

The Bluetooth standard is based on 79 independent channels working at 1 Mbit/s ($1\mu s$ symbols) selected thorough a *frequency hopping* algorithm. The MAC layer is based on a TDMA (Time Division Multiple Access) scheme using slots of $625\mu s$ each, and supports up to 8 devices within the same *piconet* (set of nodes sharing the same hopping sequence), one of them being the *master* of the piconet and polling the other *slave* devices. Master/slave communication is handled in a TDD (Time Division Duplexing) fashion, where the master uses even slots and

Fig. 1. The BTnode rev3 node

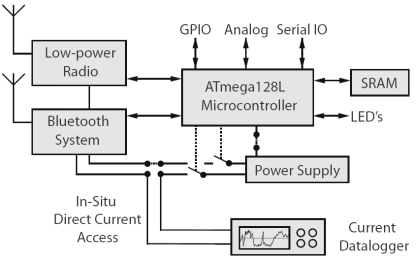


Fig. 2. Experimental setup with current datalogger connected for Bluetooth power consumption measurements

the polled slaves respond in the odd ones. Nodes are allowed to participate in more than one piconet in a time–sharing fashion, to form a *scatternet*.

During normal piconet operation (*active* mode), a master regularly polls its attached slaves every T_{poll} slots. However, slaves are completely unaware of the polling algorithm and are required to listen to the channel at the beginning of each master slot, to find out whether a packet is sent to them. *Sniff* mode allows for a lower duty cycle on both sides, with a master polling its slaves at regularly spaced beacon instants. Since beacon spacing can be in the range of seconds, rather than tens of slots (as for T_{poll}), this mode allows for power savings. More precisely, sniff mode is regulated by three parameters called Sniff Interval (SI), Sniff Attempt (SA) and Sniff Timeout (ST), which are specified in number of slot pairs[1]. SI is the time between beacons. At each beacon the slave listens to its master for SA slot pairs, during which it is allowed to send data if polled. The slave continues then listening for an extra ST slot pairs after the last packet received from the master.

The BTnode (see Figure 1) is a versatile, lightweight, autonomous platform based on a Bluetooth radio, a second independent low-power radio and a microcontroller [12]. The device is designed for fast prototyping [11] of ad-hoc and wireless sensor network applications and is well suited to investigate different protocols, operation parameter tradeoffs and radio alternatives. The Bluetooth radio is a Bluetooth 1.2 compliant device (Zeevo ZV4002) with radio circuits, baseband, MAC and link controller and an ARM7 core integrated on a single system-on-chip. The Atmel ATmega128l microcontroller serves as Bluetooth host controller and interfaces to the Host Controller Interface of the ZV4002 via UART. Embedded applications are integrated into BTnut, a custom C-based threaded operating system that offers drivers for the peripherals as well as communication protocol stacks for the radios. Benefits of this platform are a small form factor of 5x3 cm and comfortable programmability while maintaining interoperability through its standardized wireless interface. Simple sensors and actuators can be attached and powered through generic interfaces. Three direct

[1] However they are often specified in second in this document.

current access points are available where *in-situ* measurements of the power consumption of the radios and the microcontroller core can be performed (see Figure 2). This allows for very fine grained and subsystem-specific power consumption measurements in the live system under standard operating conditions as opposed to an artificial lab setup with developer boards only.

3 Power Model of Bluetooth

3.1 Experimental Phase

The TDMA, connection–oriented nature of Bluetooth makes it substantially different from other systems employing contention–based MAC protocols (e.g. 802.11). This reflects in a different power model, where power contributions also exist to merely keep links alive, even with no data transfer. In [13] and [14] we presented a complete power model of BT for the *point–to–point* case, i.e. limited to a device being master or slave of a single link. Such a model highlights three major contributions: (i) a *standby* power consumption P_{stby}, always present, (ii) a *Link Controller* (LC) power consumption varying if the device is master (P_{master}) or slave (P_{slave}) of the link and (iii) an additional *data* level consumption for transmission (P_{tx}) and/or reception (P_{rx}) of data over the link. In this model 'stby', 'master', 'slave', 'tx' and 'rx' are called *logical activities*, and the model is said to be *characterized* for a specific BT implementation once a value has been assigned to the correspondent P_{stby}, P_{master}, etc.

The work in [13] shows that the modeling abstraction of summing up power related to useful data transmissions and to link maintenance activities holds well for the point–to–point case when validating the model for a real BT device. We have run some tests in the presence of multiple links, and verified that the same property also holds for a multipoint scenario[2]. Therefore, we concentrate here on the Link Controller layer model and *extend* it to the *piconet* and *scatternet* cases, allowing for an arbitrary number of master/slave, active/sniff connections (within the limits of BT specifications).

In order to extend the model we have followed the same methodology outlined in [13] and [14]. We have run a set of roughly 100 experiments on BTnodes, tracing in each experiment the current draw of the Zeevo BT chip on the BTnode (see Section 2) for 20 seconds via a bench multimeter. The voltage, which we assumed constant during the experiments, was previously measured at 3.3 V and the multimeter was set to operate at 50 samples/s (integration time 20 ms). The following parameters have been varied among the experiments: (i) *number* of nodes connected to the device under test and *role* of these connections (maximum 7 slaves and 3 masters supported by the Zeevo chip) (ii) *mode* of these connections (active vs. sniff). In sniff mode, Sniff Interval (SI), Sniff Attempt (SA) and Sniff Timeout (ST) were also varied.

Figure 3 and Figure 4 compare the current consumption curve in active and sniff mode, for a master and slave role connection, respectively. We denote with

[2] In particular, this is true for low duty cycle or bursty traffic patterns.

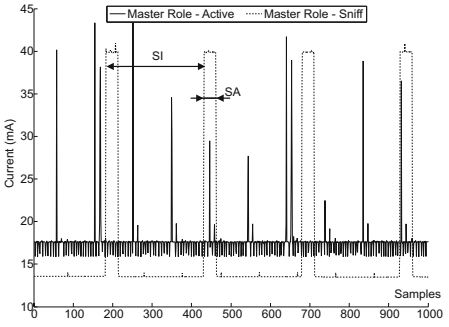


Fig. 3. Current in active and sniff mode on a master (SI=5.12 s, SA=ST=0.64 s)

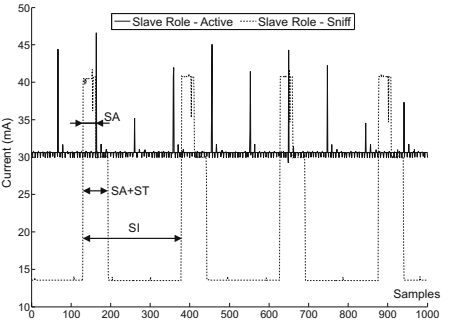


Fig. 4. Current in active and sniff mode on a slave (SI=5.12 s, SA=ST=0.64 s)

master role connection a connection to a slave (since the device is master of the link) and with *slave role connection* a connection to a master. The active–mode slave role curve (around 30 mA) is significantly *higher* than the master one (just above 15 mA); we believe this is due to the continuous listening activity a slave is required to perform.

Figure 3 and Figure 4 also highlight the bursty behavior of sniff mode, with periodic peaks every SI slot pairs. The baseline value is lower than the active master and active slave ones, and is equal to the *standby* current, which had been previously measured at 13.51 mA. Conversely, the height of the peaks surpasses both active master and slave values, reaching 40 mA; this can be justified with the increased frequency of POLL packets sent by the master during the SA (Sniff Attempt), which also causes higher power consumption on the slave receiving them. According to BT specs, the slave continues then listening for an extra ST slot pairs, and this justifies the wider peaks in Figure 4.

Figure 5 concentrates on the effects of multiple active–mode connections on the total power consumption. The left cluster of bars represents the *average* current in *piconet* mode, with an increasing number of connections to slaves (0 to 6), but no master attached. These values exhibit the interesting property that each additional slave after the first one brings a nearly *constant* power penalty.

The right cluster of bars in Figure 5 is the average current when an increasing number of slaves are attached (0 to 6, as before) but when the device also has a *master*; in this situation the BTnode is in *scatternet* mode. The values are higher than in piconet mode, and all lie in the neighborhood of 30 mA, which is the *active slave role* consumption as discussed for Figure 4. This can be explained as follows: with no data transfers, the only duty of the node in its piconet (as a master) is to poll its slaves, which accounts for a small time fraction; hence, the node spends far more time in slave mode, listening in the second piconet, and its current consumption is then much closer to the slave than to the master one. A second interesting property emerges: the total power is only slightly affected by the number of active slaves attached if an active master is present. In one word, the slave role *dominates* the master one.

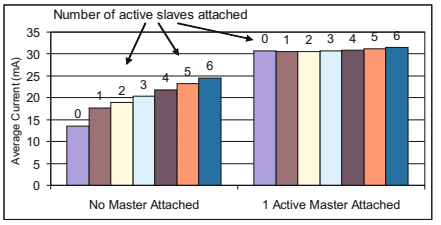


Fig. 5. Average current with sole active–mode links. Left cluster: 0 to 6 slaves attached (piconet mode); right cluster: 0 to 6 slaves plus one master (scatternet mode).

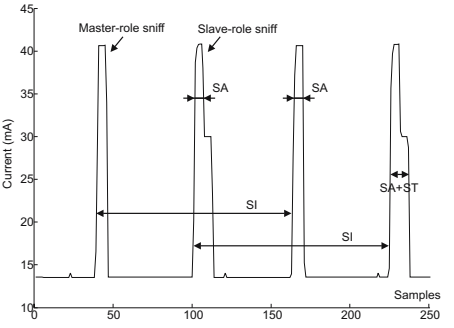


Fig. 6. Current with mixed master (one) and slave (one) sniff–mode links (SI=2.56 s, SA=ST=0.16 s)

Figure 6 refers to scatternet mode, but where all links are in *sniff mode*; more specifically, one master and one slave role links are present here. The exhibited behavior is a simple extension of that with a single link in sniff mode (as in Figure 3 and Figure 4), with the BT module scheduling each sniff attempt as far away as possible from the others.

Figure 7 shows the current plots in the case of multiple coexisting links in active and sniff mode; all links are here towards slaves (master roles) and the device is not in scatternet mode. The graph represents a single period of the 20 seconds experiments, whose waveforms are periodic with SI, namely 2.56 seconds or 128 samples. The first curve (dashed) represents the case of 3 active slaves, and shows no major peaks. The second curve (solid, 2 active and 1 sniff slaves) has a lower baseline average value but exhibits one peak of width SA. The third curve (solid with dots, 1 active and 2 sniff slaves) presents an even lower baseline value but features two peaks of width SA, according to what said for Figure 6. Finally, the fourth curve (solid with squares) has a baseline value

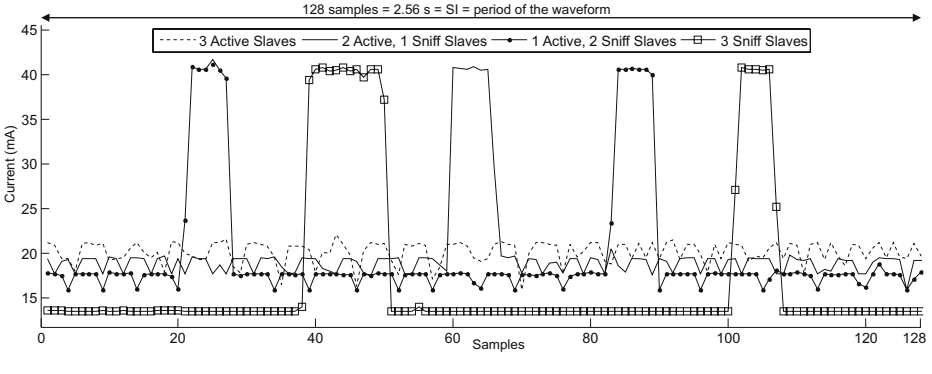


Fig. 7. Current with active and sniff master roles: 3 slaves attached, switched from active to sniff mode one after the other in a sequence (SI=2.56 s, SA=ST=0.16 s)

equal to *standby* but contains three sniff attempts, two of which are clustered together in a peak of width $2 \cdot SA$; this behavior is in line with the fact that all links are now in sniff mode. The rules emerging from Figure 7 are the following: (i) outside sniff attempt peaks power consumption is determined by the number of active–mode links; (ii) the height of the sniffing peaks is not influenced by the number of active links.

A set of experiments similar to that of Figure 7 has been performed on coexisting slave–role active and sniff–mode links (not shown here). The behavior in this case is slightly different: although there are still a baseline value and regular peaks due to sniff, the baseline value shows only *marginal fluctuations* around the value of active slave power consumption (circa 30 mA), regardless of the number of active slave roles (1, 2 or 3). The same holds true if active master roles (connections to slaves) are added, and this confirms the rule of the slave role dominance introduced earlier when discussing Figure 5.

3.2 Model Characterization and Validation

We extend here the set of LC *logical activities* of the point–to–point model (P_{master}, P_{slave}) to handle multiple connections and sniff mode. We seek a compact set of activities A_i, each with power consumption P_i, whose *linear combination* approximates with a reasonable error the actual consumption of the device in all cases. Our choice is driven by the knowledge gained in the experimental phase, which can be summarized in the following rules:

1. Power consumption is the sum of three terms: (i) a *standby* term (P_{stby}) always present, (ii) a *baseline* power value on top of standby due to active connections and (iii) periodic *peaks* due to sniff links.
2. When an *active slave role* connection exists, this fixes the baseline value at P_{slave}, regardless of additional active masters and/or slaves attached. ł
3. When *no active slave role* connection exists, baseline value is determined by the number of active master roles, with the first contributing P_{master} and each additional one contributing P_{add_slv} (with $P_{add\_slv} < P_{master}$).
4. On top of the previously determined baseline value, which shall be called P_{BAS}, contributions from sniff–mode peaks are added as follows, respectively for master roles (1) and slave roles (2):

$$P_{M,SN} = (P_{sniff} - P_{BAS}) \cdot (\frac{SA}{SI}) \quad (1)$$

$$P_{S,SN} = (P_{sniff} - P_{BAS}) \cdot (\frac{SA}{SI}) + (P_{slave} - P_{BAS})(\frac{ST}{SI}) \quad (2)$$

where P_{sniff} is the peak value during sniff attempts.

Figure 8 shows a generic BTnode having K masters and J slaves attached, that is K slave roles and J master roles. In this situation the total power consumption as predicted by the model is:

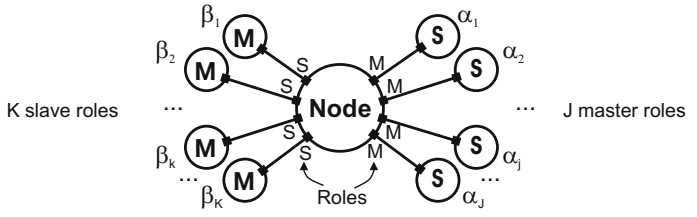


Fig. 8. The four main degrees of freedom of Link Controller state including multiple master/slave and active/sniff links

$$P = P_{stby} + \overbrace{\sum_{j=1}^{J}\Big[(1-\alpha_j)P_{M,SN}\Big]}^{\text{Sniff Master Roles}} + \overbrace{\sum_{k=1}^{K}\Big[(1-\beta_k)P_{S,SN}\Big]}^{\text{Sniff Slave Roles}} + \tag{3}$$

$$\underbrace{\beta_0 P_{slave}}_{\text{Active Slave Roles}} \underbrace{+(1-\beta_0)\alpha_0(P_{master}-P_{add\_slv}) + (1-\beta_0)\sum_{j=1}^{J}\Big[\alpha_j P_{add\_slv}\Big]}_{\text{Active Master Roles}}$$

where:

$$\alpha_j = 1 \Leftrightarrow \text{link to j–th slave is active } (1 \le j \le J)$$
$$\beta_k = 1 \Leftrightarrow \text{link to k–th master is active } (1 \le k \le K)$$
$$\alpha_0 = 1 \Leftrightarrow \exists \text{ one active link to a slave } (\alpha_0 = \alpha_1 \vee \alpha_2 \ldots \vee \alpha_J)$$
$$\beta_0 = 1 \Leftrightarrow \exists \text{ one active link to a master } (\beta_0 = \beta_1 \vee \beta_2 \ldots \vee \beta_K)$$

The extended set of logical activities that make up the model are hence P_{stby}, P_{master}, P_{add_slv}, P_{slave}, P_{sniff}. To characterize the model for the BTnode means to assign numeric values to these quantities, following the methodology fully described in [13]. For each experiment, the *average power* measured during the experiment is equaled to the prediction of the model according to (4). For a single experiment j, all α_j and β_k coefficients are fixed, and thus (4) becomes a linear combination the activity power consumptions:

$$V \cdot \overline{I}_j = \sum_{i=0}^{N} P_i t_{ji} \tag{4}$$

where $P_1 = P_{stby}$, $P_2 = P_{master}$, etc. are to be determined, $\overline{I}_j$ is the average current during the experiment, t_{ji} are coefficients determined by the values of the α_js and β_ks, and V is the operating voltage of 3.3 V.

Since the number of experiments is significantly higher than the number of unknowns, the equations (4) for all experiments, if taken together, form a strongly over–constrained linear system, which can be solved reliably with the Least Squares method. Doing so yields the values that best fit the experimental data, shown in Table 1.

Table 1. Numerical power model for the BTnode

Activity	Description	Value
P_{stby}	Always present	44.58 mW
P_{master}	Being master of 1 slave	12.97 mW
P_{add_slv}	Having additional slaves	4.55 mW
P_{slave}	Being slave	56.63 mW
P_{sniff}	Peak value in sniff mode	86.96 mW

We have validated our linear model using the LOO (Leave One Out) technique [15] as described in [13]. This implies solving the model repeatedly, excluding each time a different test, and using that test to calculate a residual (difference between measure and prediction). The RMS value of such residuals (actually, percentile residuals) is the *validation error* and amounts to 3.7%, whereas the maximum residual among all experiments is around 10%. Although the numbers in Table 1 might seem very specific to the BTnode, our experience and further measurements on other Bluetooth modules confirm that the power model and some of the trends highlighted by Table 1, such as $P_{sniff} > P_{slave} > P_{master} > P_{add\_slv}$, are common to most BT implementations [14].

4 The Power/Delay Problem in Bluetooth Scatternets

The presented power model, while guaranteeing an accuracy below 4%, is analytically simple enough to be used in solving a number of power management problems. We define a generic *power management problem* as the seek of a network configuration that minimizes some power figure (total power of the network, maximum power of a node, standard deviation of power over all nodes etc.) whilst satisfying some QoS requirements.

In particular, in this work we focus on the power/delay problem, which can be stated as follows: *given* a scatternet topology, a power model for each node and a set of end–to–end maximum delay requirements between a root node and all other nodes in the network, *determine* the best configuration (whether each link should be in active or sniff mode, and the value of SI, SA, ST in the second case) that *minimizes* the total power in the network, or alternatively optimizes some other objective functions. We approximate the delay introduced by a single link as equal to SI in sniff mode, or 0 if active. We limit our study here to tree topologies, which eliminates routing issues. Figure 9 visualizes the problem in graphical notation: the same delay requirement D_{max} is applied here to all leaves (which implies to all nodes) of the tree considering node 1 as the root, topology and master–slave orientations are given.

This kind of optimization problem suits well all situations in which the main limiting requirement is delay and not throughput, which maps to all applications that use Bluetooth but would not strictly require its whole bandwidth of 1 Mbit/s. This includes all sensor networks handling time–critical data with relatively low packet sizes, such as security, health and environmental monitoring

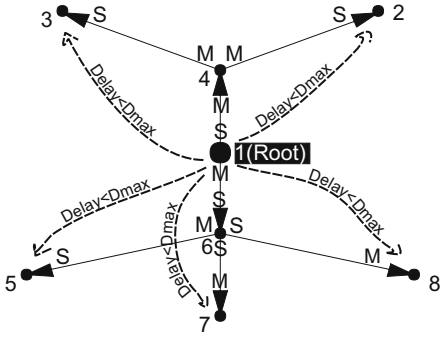


Fig. 9. The power/delay problem in a tree of BTnodes; links are oriented away from root, delay requirements are from node 1 (root) to all leaves, M and S indicate Master and Slave roles on links

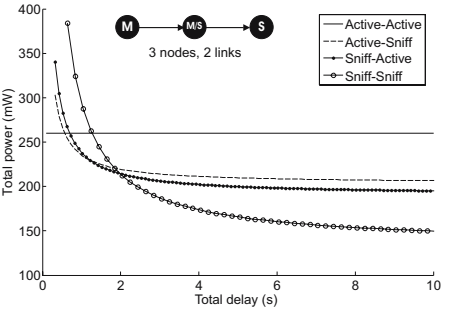


Fig. 10. Power vs. delay Pareto curves for a chain of three nodes (two links) varying SI with fixed SA,ST. Depending on the target delay, active or sniff mode are more convenient.

systems. The framework can be as well applied to throughput–constrained scenarios; this is discussed as ongoing work in Section 6.

Figures 3 and 4 in Section 3 suggest a power *tradeoff* between active and sniff mode as the Sniff Interval (SI) is varied. This is confirmed by Figure 10, which plots the power consumption of three nodes connected in a chain (two links) according to (4) as the mode of the links is switched between active and sniff, and as SI is varied (SA, ST fixed).

When the number of links in the network grows, it is inefficient to evaluate all possible combinations of link mode and sniff interval. Hence, we have chosen *MILP* (Mixed–Integer Linear Programming) as an optimization tool to solve the problem for bigger networks. The problem must be slightly modified before it can be handled by a MILP optimizer; in fact, the model is *linear* w.r.t. the power consumption of the logical activities P_{stby}, P_{master}, P_{slave}, P_{add_slv}, P_{sniff} but *not* linear w.r.t. other parameters, such as the link mode binary variables α_j and β_k and the Sniff Intervals SI. However, these issues can be tackled with standard techniques as adding additional variables and constraints, as well as approximating nonlinear functions (such as $P \propto 1/\mathrm{SI}$).

An initial complexity study on our power/delay linear programs (using Cplex on a Unix workstation) exhibits a quadratic behavior with the number of nodes for medium–size networks (up to 300 nodes) and an exponential growth thereafter, when the MILP optimizer starts employing different algorithms.

5 Selected Case Studies

5.1 Total Power in a Chain of Nodes

Figure 11 shows a chain of 10 BTnodes, where each node has different master/slave roles (M-M,M-S,S-M,S-S). In this case all sniff parameters are fixed at SI=2s, SA=ST=.01s, and only the mode (active vs. sniff) of each link is treated

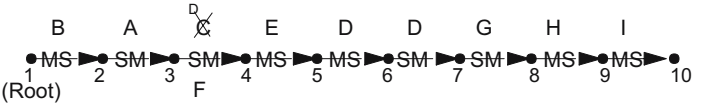
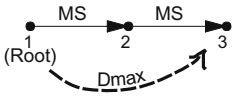


Fig. 11. Chain of 10 nodes, each link can be in active mode or sniff mode with SI=2s, SA=ST=.01s, objective is sum of power over all nodes. As delay requirement from root to other nodes is decreased from 18 s to 10 s links start switching from sniff to active mode; the order in which this happens is indicated by the letters (A, then B, etc.).



D_{max}	22	20	19	18	15	10	5	4	3	2	1	.5	.2
SI(1,2)	10	10	9	8	5.3	4.7	2.3	1.3	1.3	.7	A	A	A
SI(2,3)	10	10	10	10	9.7	5.3	2.7	2.7	1.7	1.3	1	.5	A

Fig. 12. Chain of 3 nodes, each link can be in active mode or sniff mode with SA=ST=.01s and variable SI between 0.1 s and 10 s, objective is sum of power over all nodes. Table shows best configuration for decreasing values of D_{max}. A stands for Active, numbers are SIs.

as an optimization variable. The objective function is here the sum of power over all nodes. In Figure 11, the optimization is run repeatedly as the end–to–end delay requirement D_{max} from node 1 (root) to all other nodes is gradually lowered from 18 s to 10 s. Initially, all links are in sniff mode; as D_{max} decreases, links start switching to active mode; the order in which this happens is indicated in figure by capital letters (A first, then B, etc.). The lessons learned with this simple experiment are:

- As delay requirement is decreased links switch from sniff to active mode.
- Active links stick together. This is convenient power–wise, as experiments have proved that additional active roles cost less than the first one.
- Active links appear first on S-S nodes, then on M-S and finally on M-M nodes. This can again be justified with the rules described in Section 3.2.

Figure 12 still refers to a chain of nodes (three in this case), but where the Sniff Interval is also an optimization variable, in the range $0.1s \leq \text{SI} \leq 10s$. The objective function is still the total power consumption in the network, and the end–to–end delay requirement is decreased from 20 s to 0.2 s. The table in Figure 12 shows the best combination of link mode and SI for both links. Further considerations are:

- The best combination of Sniff Intervals on a chain of links is the one in which all intervals are equal to $D_{max}/(\text{number of links})^3$.
- The switch from sniff to active happens earlier for higher values of SA/ST.

[3] This can be proved analytically. We have verified that the deviation from this 'symmetric' behavior exhibited by the values in Figure 12 is due to linearization.

5.2 Total Power in a Tree of Nodes

In Figure 13 a fixed–SI optimization (SI=2) is applied repeatedly to a tree topology. Again, for sufficiently high values of the maximum delay requirement D_{max}, all links are in sniff mode. Conversely, as D_{max} is decreased, links start switching to active mode, in the order indicated by the capital letters in Figure 13 (A first, then B, etc.). The observed behavior can be summarized as follows:

- The first candidates to become active are the links (thus the nodes) that serve a higher number of downstream nodes at the same time. They are followed by minor branches and, at last, by leaves. The tree in Figure 13 has a sort of *backbone* (main branch) along nodes $1, 18, 15, 14, 5, 11$, then branches from node 11, and finally numerous leaves along the backbone and branches: the above rule is clearly obeyed.
- In addition, the same rules found for the simple chain of nodes apply: S-S nodes are the best candidates, followed by M-S and M-M.

Figure 14 refers to the same tree shown in Figure 13, but now the Sniff Intervals are used as optimization variables ($0.1 \leq \text{SI} \leq 10$ s). The table is divided into three parts:

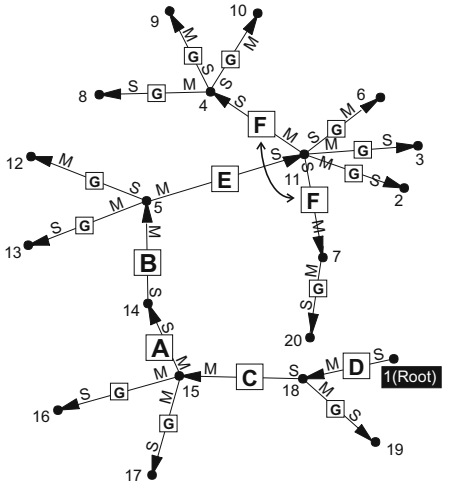


Fig. 13. Random tree, each link can be active or sniffing with SI=2s, SA=ST=.1s, objective is total power. D_{max} from root to other nodes is decreased from 15 s to 0 s; letters indicate order in which links switch from sniff to active (A first, then B, etc.).

	D_{max}			
Link	15	10	5	2.5
1,18	1.3	1.3	0.6	A
18,15	1.3	1.3	1	A
15,14	1.3	A	A	A
14,5	1.3	A	A	A
5,11	3	1.3	A	A
11,4	3.3	2.6	1.3	1.1
11,7	3.3	3.3	A	A
18,19	10	8.6	4.3	2.5
15,16	10	7.3	3.3	2.5
15,17	10	7.3	3.3	2.5
5,12	9.6	7.3	3.3	2.5
5,13	9.6	7.3	3.3	2.5
11,2	6.6	6	3.3	2.5
11,3	6.6	6	3.3	2.5
11,6	6.6	6	3.3	2.5
4,8	3.3	3.3	2	1.3
4,9	3.3	3.3	2	1.3
4,10	3.3	3.3	2	1.3
7,20	3.3	2.6	3.3	2.5

Fig. 14. Random tree (see Fig. 13), optimizing total power with variable SI, SA=ST=.1s; optimal SI (or A=Active) for different values of D_{max}

- The first five links are those of the backbone, and first switch to active.
- The second block contains links (11,4) and (11,7), which represent branches from node 11, each with its own leaves. Interestingly, for $D_{max} = 5s$, (11,7) is *active* but (11,4) is not, forcing two active slave roles on node 11; at the same time, (15,18) and (1,18) on the main backbone are *not* active. This suggests that a tradeoff exists between the previously devised rules concerning size of branches and node roles (S-S, M-S, M-M), and that the order of application of these rules is not fixed.
- The third and last block groups all "leaf links", whose sniff interval decreases with D_{max} as well as with the distance from the root (e.g. $SI(18,19) < SI(11,2) < SI(4,8)$). However, in certain cases, some values can increase for lower D_{max}, as other links closer to the root switch to active (see link (7,20) for instance).

5.3 Coping with Real World Constraints

The solutions we have presented so far do not take into account an important limitation of the BTnode's BT subsystem: if *multiple sniff links* are activated on a node they must have *equal SI*. This constraint, which simplifies link scheduling for the BT hardware, definitely changes the structure of the problem. Figure 15 and Figure 16 depict the optimal solutions for the same scenario of Figure 13 with this additional constraint, for $D_{max} = 15s$ and $D_{max} = 10s$ respectively. It is worth noting that:

- Applying the SI equality constraint implies that the *whole network* must use the same SI *unless* some active links exist.

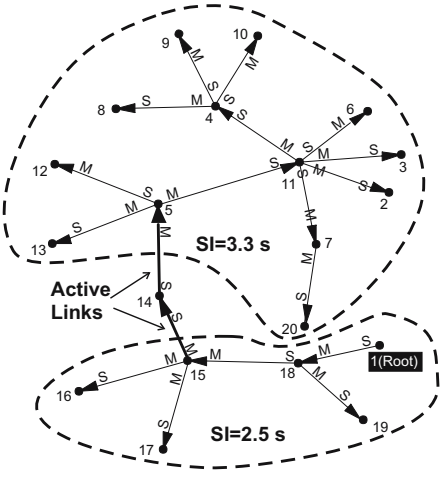


Fig. 15. Total power optimization with variable SI, but fixed SI for each node. $D_{max} = 15s$.

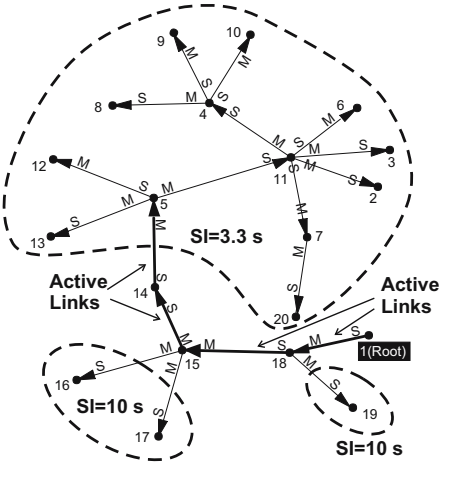


Fig. 16. Total power optimization with variable SI, but fixed SI for each node. $D_{max} = 10s$.

– If *active* links exist, they act as separators among *iso–SI clusters* of nodes.
– These two rules cause the best solution already for $D_{max} = 15$ (Figure 15) to contain two active links, as compared to all sniff links in Figure 14.

5.4 Total Power vs. Maximum Power Optimization

Table 2 compares the optimal solution for different values of D_{max} as different combinations of objective functions and constraints are employed:

– Columns *a* refer to the results already presented, taking the *total power* in the network (sum over all nodes) as objective to be minimized. This is equivalent to minimizing the average power consumption of a node, but with no check on the standard deviation. This causes the optimum to be quite unfair among the nodes, e.g. links (15,14) and (14,5) are switched early (as D_{max} decreases) to active mode as it is best for the whole network, however this quickly drains the battery of node 14.
– To overcome this limitation, columns *b* use a different objective function, namely the *maximum power* consumption of a *single node*[4]. The results for

Table 2. With reference Fig. 13, SA=ST=.1s, optimal mode and SI for each link with different objective functions and constraints: a) minimizing total power; b) minimizing maximum power of a single node; c) as in b but with equal SI on all links on each node

	$D_{max} = 15$			$D_{max} = 10$			$D_{max} = 5$			$D_{max} = 2.5$		
Link	a	b	c	a	b	c	a	b	c	a	b	c
1,18	1.3	1.3	2.14	1.3	0.6	1.42	0.6	0.6	ACT	ACT	ACT	ACT
18,15	1.3	1.3	2.14	1.3	1.1	1.42	1	0.6	ACT	ACT	0.6	1.25
15,14	1.3	1.3	2.14	ACT	1	1.42	ACT	ACT	ACT	ACT	ACT	ACT
14,5	1.3	2.5	2.14	ACT	1.3	1.42	ACT	ACT	ACT	ACT	ACT	ACT
5,11	3	2.5	2.14	1.3	1.3	1.42	ACT	0.6	1.6	ACT	ACT	ACT
11,4	3.3	2.6	2.14	2.6	2.4	1.42	1.3	1.3	1.6	1.1	ACT	ACT
11,7	3.3	4.7	2.14	3.3	3.1	1.42	ACT	1.6	1.6	ACT	ACT	ACT
18,19	10	10	2.14	8.6	9.3	1.42	4.3	4.3	ACT	2.5	2.5	1.25
15,16	10	10	2.14	7.3	8.2	1.42	3.3	3.6	5	2.5	1.8	1.25
15,17	10	10	2.14	7.3	8.2	1.42	3.3	3.6	5	2.5	1.8	1.25
5,12	9.6	8.5	2.14	7.3	5.8	1.42	3.3	3.6	1.6	2.5	1.8	1.25
5,13	9.6	8.5	2.14	7.3	5.8	1.42	3.3	3.6	1.6	2.5	1.8	1.25
11,2	6.6	6	2.14	6	4.5	1.42	3.3	3	1.6	2.5	ACT	ACT
11,3	6.6	6	2.14	6	4.5	1.42	3.3	3	1.6	2.5	ACT	ACT
11,6	6.6	6	2.14	6	4.5	1.42	3.3	3	1.6	2.5	ACT	ACT
4.8	3.3	3.3	2.14	3.3	2	1.42	2	1.6	1.6	1.3	ACT	ACT
4.9	3.3	3.3	2.14	3.3	2	1.42	2	1.6	1.6	1.3	ACT	ACT
4,10	3.3	3.3	2.14	3.3	2	1.42	2	1.6	1.6	1.3	ACT	ACT
7,20	3.3	1.3	2.14	2.6	1.3	1.42	3.3	1.3	1.6	2.5	1.8	1.25

[4] The actual objective is a linear combination of the maximum power of a single node and of the total power, where the latter has a lower weight.

$D_{max} = 10$ show lower sniff intervals and no active links, as compared to strategy a, meaning that power consumption, even though higher, is more evenly distributed among nodes. This trend is confirmed by the cases of $D_{max} = 5$ and $D_{max} = 2.5$, where sniff–mode links have low SIs, leveling their power consumption with that of active links.

– Finally, columns *c* have the same objective as b, but with the additional constraint introduced in Section 5.3, which imposes the same SI on all links on each node. Comparing the results with those of Figure 15 and Figure 16 it is apparent that iso–SI clusters begin forming for lower values of D_{max}. In fact, for $D_{max} = 10$, all links are here in sniff mode with SI=1.42 s, whereas in Figure 16 (still $D_{max} = 10$, but minimizing total power) 4 active links and 3 clusters exist.

6 Conclusions and Extensions of the Framework

We have presented a real–world power model of a Bluetooth device in a scatternet scenario, where links can be in active or low–power sniff mode. The model has been validated experimentally with an average validation error below 4%. Based on our model, we have described a framework which employs Mixed–Integer Linear Programming to solve power/delay optimization problems. Results have been shown for selected topologies ranging from chains to trees of nodes.

The results provide useful rules to determine the best network configuration power–wise given certain requirements. These rules constitute a power management policy for the network, which can be implemented in different ways. The policy can be centrally computed in the root and distributed to all nodes every time the requirements change; alternatively, the solution for the most used requirements sets (e.g. high responsiveness, low responsiveness, idle) for the network can be pre–computed and stored in the nodes as look–up table.

Building on top of this work, we are currently exploring a number of extensions, including mesh topologies (which implies routing), park and hold modes, mixed delay and throughput requirements as well as a traffic matrix on top of open connections. Additionally, heterogeneous networks (e.g. different power budgets in the nodes) and battery models could be taken into account.

Although the BTnode employed in this study presents a standby consumption which is too high to implement long–life sensor networks, the methodology we have followed could be easily re–applied in the future to more power–aggressive implementations of Bluetooth as well as to other protocols (e.g. Zigbee) that promise a better power/performance ratio. In this direction we are investigating the possibility of completely switching off and on the BT radio of the BTnode using the microcontroller to obtain a low–power mode that performs better than sniff for extremely low duty cycles.

References

1. Raghunathan, V., Pering, T., Want, R., Nguyen, A., Jensen, P.: Experience with a low power wireless mobile computing platform. In: Proc. ISLPED-04, ACM Press (2004) 363–368
2. Jones, C.E., Sivalingam, K.M., Agrawal, P., Chen, J.C.: A survey of energy efficient network protocols for wireless networks. Wireless Networks (**7**) 343–358
3. Schurgers, C., Aberthorne, O., Srivastava, M.: Modulation scaling for energy aware communication systems. In: Proc. ISLPED 2001, (ACM Press) 96–99
4. Rulnick, J.M., Bambos, N.: Mobile power management for wireless communication networks. Wirel. Netw. **3** (1997) 3–14
5. Kazantzidis, M., Gerla, M., Johansson, P., Kapoor, R.: Personal area networks: Bluetooth or ieee 802.11? Intl. Journal of Wireless Information Networks, Special Issue MANETs Standards, Research, Applications (2002)
6. Leopold, M., Dydensborg, M.B., Bonnet, P.: Bluetooth and sensor networks: a reality check. In: SenSys '03: Proc. 1st international conference on Embedded networked sensor systems, New York, NY, USA, ACM Press (2003) 103–113
7. Chakraborty, I., Kashyap, A., Rastogi, A., Saran, H., Shorey, R., Kumar, A.: Policies for increasing throughput and decreasing power consumption in bluetooth mac. (In: Proc. 2000 IEEE intl. conf. on Personal Wireless Comm.) 90–94
8. Zhu, H., Cao, G., Kesidis, G., Das, C.: An adaptive power-conserving service discipline for bluetooth. In: 2002 IEEE intl. conf. on Communication. Volume 1. (2002) 303–307
9. Ashok, R.L., Duggirala, R., Agrawal, D.P.: Energy efficient bridge management policies for inter-piconet communication in bluetooth scatternets. In: Proc. Vehicular Tech. Conf. (2003)
10. Meier, L., Ferrari, P., Thiele, L.: Energy-efficient bluetooth networks. Technical Report 204, Comp. Eng. and Networks Laboratory (TIK), ETH Zurich (2005)
11. Beutel, J., Kasten, O., Mattern, F., Römer, K., Siegemund, F., Thiele, L.: Prototyping wireless sensor network applications with BTnodes. In: Proc. 1st European Workshop on Sensor Networks (EWSN 2004). Volume 2920 of Lecture Notes in Computer Science., Springer, Berlin (2004) 323–338
12. Beutel, J., Dyer, M., Hinz, M., Meier, L., Ringwald, M.: Next-generation prototyping of sensor networks. In: Proc. 2nd ACM Conf. Embedded Networked Sensor Systems (SenSys 2004), ACM Press, New York (2004) 291–292
13. Negri, L., Sami, M., Macii, D., Terranegra, A.: Fsm–based power modeling of wireless protocols: the case of bluetooth. In: Proceedings of the 2004 international symposium on Low power electronics and design, ACM Press (2004) 369–374
14. Negri, L., Zanetti, D., Tran, Q.D., Sami, M.: Flexible power modeling for wireless systems: Power modeling and optimization of two bluetooth implementations. In: Proc. WoWMoM 05 - IEEE Intl. Symp. on World of Wireless, Mobile and Multimedia Networks. (2005) 408–416
15. Hassoun, M.: Fundamentals of Artificial Neural Networks. MIT Press (1995)