Towards a better battery model for INET

Laura Marie Feeney
Uppsala University
lmfeeney@it.uu.se

***Abstract*—Battery lifetime is often an important performance metric when simulating networks containing wireless devices. When evaluating system or protocol performance, there are cases where it is useful for the battery model to more accurately reflect the complex electrochemical processes involved in battery discharge.**

This paper describes an implementation of the KiBaM battery discharge model for INET. With regard to the INET implementation itself, the main technical issue is to extend the INET power model to more effectively handle both constant output voltage energy sources (mains power and simple battery models) and variable output voltage sources (complex battery models).

A custom battery testbed allows us to measure the discharge behavior of Li-coin cells under controlled conditions. The testbed is capable of coarse-grain modeling of the current draw associated with an IEEE 802.15.4 radio transmitting a frame. This enables a preliminary comparison between the simulation model and real discharge behavior.

I. Introduction

Operational lifetime is an important performance metric for battery-powered wireless devices, especially ones that are expected to operate for months or years without maintenance.

Such long lifetimes mean that experiments that directly measure battery lifetime under real loads are generally impractical. Instead, the device’s power consumption is measured over a short period of time and it is this value that is used as the figure of merit. Usually, the operating system is instrumented to record the amount of time that device drivers spend in various states (e.g. sleep, transmit, receive). The power consumption is then inferred from hardware specifications for the device. There are also a few testbeds, such as [3], that directly measure the current that the device draws from the battery. However, this requires specialized hardware.

In both approaches, the focus is on the load that the device puts on the battery. The battery itself is treated as a linear source of charge (mA-h) that is drained by operations that consume i mA for t time.

In reality, a battery is an electrochemical system and a device fails when the chemical reactions in the battery are no longer able to maintain a sufficiently high output voltage under load (i.e. the device cut-off voltage). For a given battery chemistry and physical structure, the discharge process is determined by the timing and intensity of the applied load, as well as by external factors such as temperature. There is also considerable manufacturing variation, even between nominally identical batteries.

In short, battery behavior is complex and non-linear. This limits the value of “accelerated” testbed experiments that use unrealistically high loads and/or duty cycles to obtain a more practical run time: The battery discharge behavior is qualitatively different under these conditions.

This complexity also suggests that there are aspects of performance evaluation in which it may be important to consider battery discharge behavior in more detail. These include comparing protocol performance, absolute lifetime estimation, the accuracy of real-time state of charge estimation, and the battery/device interface. The last three of these are important practical topics which are particularly dependent on improved battery models. This work provides a step toward that goal.

Simulating battery lifetime requires modeling three components: the load created by the device, the battery’s response to load, and the device’s response to changes in battery state. This paper describes a preliminary implementation of the ’hybrid KiBaM’ model [2] for OMNeT++’s [8] INET framework. The battery model is parameterized for the Panasonic CR 2032 Li coin cell [5], which is a 225 mA-h battery suitable for small devices.

The implementation of the load and device response are largely based on existing INET infrastructure for power modeling: The load is defined by tracking changes in device state, using OMNeT+’s signaling mechanism. The implementation and parameterization of the battery model itself are mostly independent of the OMNeT++/INET environment; the interested reader is referred to [7].

The main INET technical issue is to suitably extend the INET power model to more effectively handle both constant output voltage energy sources (mains power and simple battery models) and variable output voltage sources (complex battery models). Due to issues in the templates for managing units, it may be challenging to do this in a backwards compatible way.

A custom testbed for discharging batteries under controlled loads is described in [1]. The test hardware can be used to mimic the load patterns generated by an INET simulation and apply them to real batteries. The simulation results can then be compared to results obtained from the testbed.

II. Batteries and modeling

Battery modeling has been an important engineering technique in many fields for many years: A good overview for the non-expert is found in [6].

In most existing simulations of power consumption, the battery’s residual capacity or state of charge is models using “coulomb counting”. The battery has a known initial charge capacity (mA-h) which is drawn from the battery at a rate defined by the current $I(t)$. The consumed charge is the integral of current over time. In practice, $I(t)$ is treated as

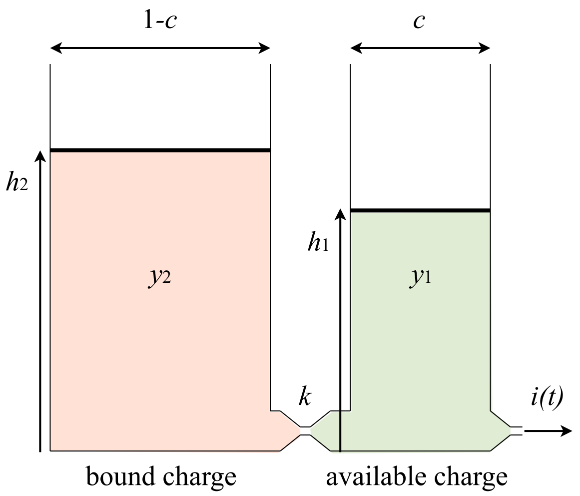

Fig. 1. Abstract view of the KiBaM model for battery state of charge.

a piecewise constant function that is the sum of fixed currents associated with each of the operations taking place on the device at any given time. The battery is empty and the device fails when the residual capacity reaches zero. (A power source, such as a solar panel, is similarly modeled as providing charge to the battery.)

The coulomb counting approach does not directly support modeling of the battery output voltage, which is mostly ignored. Generally, it is assumed to be equal to the battery's nominal output voltage as long as the battery has non-zero residual capacity and zero otherwise. In fact, it is the battery output voltage, and especially the voltage response when a load is placed on the battery, that actually determines operational lifetime of a device. The device electronics do not operate correctly when the input voltage is below some (device dependent) threshold value. Depending on the interface between the device and the battery, its behavior may also be affected by decreasing output voltage even before the cut-off.

A. *KiBaM: State of charge modeling*

Battery discharge exhibits two important non-linear behaviors[1]: The first is the rate-capacity effect: Discharging the battery at a higher current (rate at which charge is drawn from the battery) reduces its effective capacity. The second is charge recovery: An intermittent discharge is more efficient than a continuous one.

KiBaM [4] is an analytic battery model that is intended to reflect these behaviors by treating the battery as a pair of logical "wells", connected by a "valve" (Figure 1). One well contains "available" charge, which can used by the device. The current is the rate at which the charge is drained from this well. The battery fails when the device is unable to drain any more charge from the "available" well. The other well contains "bound" charge, which cannot be used by the device. Charge from the "bound" well flows into the "available" well though the "valve", at a rate proportional to the difference in height between the two wells. This interaction can be represented by a straightforward set of differential equations, which provide a good balance between accuracy and computational complexity.

[1]Temperature dependence is another important factor, but is not addressed in this work.

Qualitatively, we see that this model captures several of the key points above: A high current discharges the battery disproportionately more quickly than a low current (the available well drains more quickly than it can be replenished by the bound well). An intermittent current discharges the battery more efficiently than a continuous one, since the available charge is replenished from the bound charge.

The KiBaM model is parameterized in terms of the relative sizes of the two wells and the valve between. The parameterization is highly battery specific and is based on detailed measurements of the battery in question under various kinds of controlled loads. The implementation reported here is parameterized for the Panasonic CR2032 Li coin cell.

B. *Output voltage modeling*

The ability of the device electronics to function as specified is determined by the battery's ability to maintain a sufficiently high output voltage under load. The battery output voltage can be seen as evolving on two different time scales. Under constant load, the output voltage is roughly constant or slightly decreasing over much of its lifetime, before decreasing sharply near its end of life.

Under an intermittent load, finer gain details emerge. When a load is applied to the battery, the output voltage immediately drops in proportion to its internal resistance, which increases as the battery is discharged. The output voltage continues to drop over the duration of the load; the magnitude of this effect reflects on the how well the electrochemical processes in the battery are able to "keep up" with the ongoing demand for charge. This ability decreases with decreasing state of charge, as the active species in the battery become depleted. It is the minimum voltage reached during this time that defines the battery lifetime. When the load is removed, the internal resistance of the battery is no longer present, so the output voltage increases. Due to relaxation and charge recovery processes in the battery, the voltage continues to recover for some time.

Hybrid KiBaM [2] combines the KiBaM state of charge model with an equivalent circuit model of the battery output voltage. The circuit consists variable voltage source (the battery's open circuit voltage), a resistor (the internal resistance) and an RC circuit. The circuit parameters are exponential functions of the KiBaM state of charge, with the constants determined experimentally for the battery in question.

Details of the Hybrid KiBaM model and its parameterization for the CR2032 Li coin cell are described in [7]. The results presented in that paper suggest the potential for significant improvements in the model. The voltage model is therefore only partially implemented in INET.

III. INET MODEL

INET 3.2.4 includes a power consumption modeling framework, with key structures highlighted in Figure 2. The main components are the *EnergyStorage* and *Consumer* modules. A simple battery model is implemented in *SimpleEnergyStorage*.

The *EnergyStorage* module models the state of the energy store and makes this information available to other modules using OMNeT++ signals. It is where the battery discharge model itself is implemented. *EnergyStorage* can also invoke the *lifeCycleController* to propagate shut-down commands to simulate battery depletion. *Consumer* modules act as interfaces between *EnergyStorage* and other modules that simulate activities that consume power, such as a *Radio* module. There can be multiple consumers, which can be dynamically added and removed, *EnergyStorageBase* handles the relevant housekeeping[2].

Having a *Consumer* module separates modeling of the power consumption from the modeling of activities that consume power. This means that these modules do not need to explicitly include power consumption in their implementations, which allows for cleaner code. In particular, it allows the – possibly very hardware specific – calculation of the load caused by various operation to be developed separately. This means that each *Consumer* module is highly specific to the module whose power consumption it is representing. And of course, the module must somehow make the necessary information available to the *Consumer*. A *Consumer* can also control the level of detail with which it models the resulting load, allowing the *Consumer* to match its fidelity with that of the chosen *EnergyStorage*.

The *StateBasedConsumer* is part of the wireless physical layer and models the radio energy consumption. The *Radio* module emits signals announcing transitions among a range of radio states. The *StateBasedConsumer* queries the *Radio* module for detailed mode information and translates this into power consumption values. The power consumption of various radio modes is parameterized in the *StateBasedConsumer*.

The current implementation of the *StateBasedConsumer* assumes a fixed transmit power. To implement variable transmit power (and the associated differences in power consumption), it would be necessary for the *StateBasedConsumer* to also query the *Radio* for information about the transmit power being used when the radio is transmitting. The *StateBasedConsumer* would also need to be parameterized with relevant information about the power consumption of various transmit output powers.

The interface between *Consumer* and *EnergyStorage* that actually applies the simulated load to the simulated battery is instantiated via a function call *setPowerConsumption()*. This function takes Watts as its argument (as do related *get* and recording functions). This is deeply embedded in the power framework: the *EnergyStorageBase* classes in the INET power structure are defined in terms of Watts, via templates in the C++ code and @units in the .ned files.

The KiBaM battery model mathematics are straightforwardly implemented in a class derived from *Storage*. *Consumer* modules act as interfaces between *EnergyStorage* and modules that simulate activities that consume energy. However, KiBaM and other advanced battery models take current $I(t)$ as an input and generate either state-of-charge $SoC(t)$ and/or voltage $V(t)$ as an output.

The current KiBaM implementation deals with this incompatibility in rather unsatisfactory ways: *KiBaMEnergyStorage* provides a *set**Current**Consumption()* interface, which in turn requires a new current-based *KibamStateBasedConsumer*. To interact with the EnergyStorage "housekeeping" functions, the current implementation is based on some rather ugly code to bypass the Watts-oriented interfaces. (This cannot be used as a final implementation.)

Of course, it is possible to address this by implementing a parallel "current"-oriented class hierarchy. However, it would be helpful to be able to avoid duplicate code by creating a more integrated structure. The challenge is to make use of the safety and automatic conversions provided by the OMNeT++ @units templates, while still allowing for flexibility.

More generally, support for both power- and current-oriented power models is important for extending INET power modeling to other domains. In system and network simulation, there are (at least) two topic areas in which energy efficiency is important. The first area is reducing the power consumption of computing and communication infrastructure, such as data centers. These systems use mains power, which is intended to operate as a constant voltage energy source (e.g. 240V). In this case, consumption is naturally expressed in terms of Watts and Watt-hours. The second area is maximizing the effective lifetime of battery powered devices. A battery's output voltage depends heavily on its state-of-charge and load, as well as exogenous factors such as temperature. In this case, consumption is naturally expressed in terms of Amps and Amp-hours.

For simple battery models, this distinction is not an issue in practice because the battery voltage is treated as constant (the nominal output voltage) over its lifetime. Even if when loads are specified in terms of current, current is easily converted to power ($P = I \times V$) and this constant factor is simply carried through everywhere – the problem is only one of semantics.

However, more complex battery models expressly simulate the output voltage of the battery under load. Current is drawn from the battery at whatever voltage the battery supplies. This makes the use of power-based interfaces problematic. The relationship between the battery voltage and the current consumed for a given operation is non-trivial. In practice, it depends on device hardware and may be described in data sheets. This makes it important to have clean support for both battery modeling strategies, as well as to help ensure that users sensibly combine *EnergyStorage*, *Consumer*, and simulation modules.

[2] *EnergyStorage* is actually comprised of *EnergySource* and *EnergySink*, which handle power consumed from and added to the *EnergyStorage*, via *Consumer*s and *Generator*s, respectively. For non-rechargeable batteries, only the former is relevant and is the focus of this discussion. Codewise, the two interfaces are analogous.

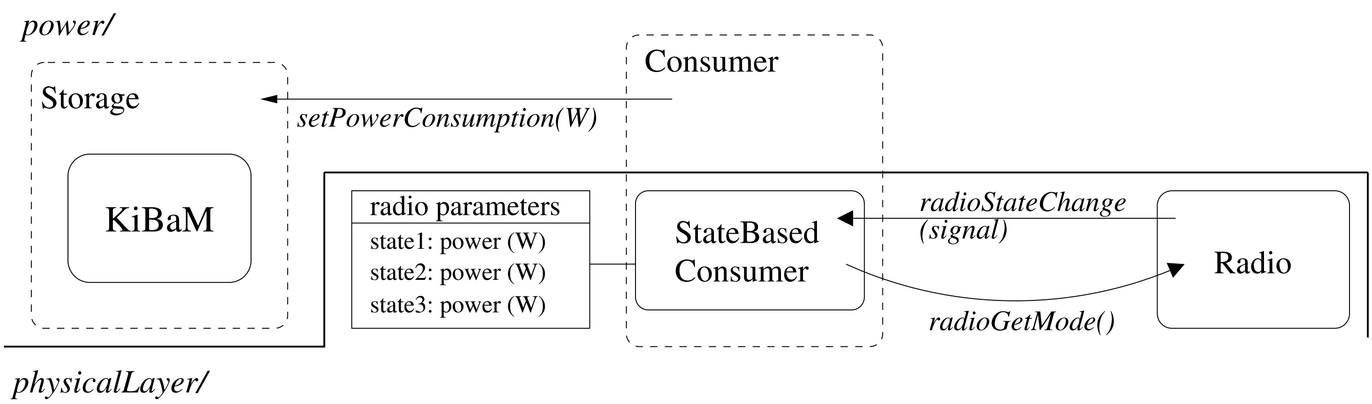

Fig. 2. Overview of the INET power framework. Power consumption is modeled by the *Consumer*, which translates information about device operations (e.g. of the *Radio*) into information about the load placed on the *Source*. Power generation is modeled using an analogous structure (not shown).

IV. Testbed

Because it is difficult to control the load generated by radios operating in real wireless environments, it is useful to be able to discharge batteries under more controlled conditions.

There are commercial testbeds that support current- and voltage- controlled loads and fine-grain measurements, but this approach is extremely expensive for large scale experiments. A large scale low-cost custom battery testbed is described in [1]. This hardware supports only resistive loads and allows only rather coarse grain load timing and voltage measurement.

In this work, the testbed is configured to generate load patterns that resemble those caused by transmissions from an IEEE 802.15.4 radio. To focus on the battery rather than variation associated with wireless communication, the load assumes that there is no contention or loss.

The load transitions through four operating states (Figure 3): First, it is in the receive state for one of eight randomly chosen intervals, ranging from 0.2 to 2.6 ms. This approximates the random backoff (n * 0.32 ms) and rx-to-tx transition. Then it is in the transmit state for 4.0 ms, the time it takes to transmit 125B. Then it returns to the receive state for 0.8 ms, which approximates the tx-to-rx transition and receiving the ACK. The nominal receive current is 20mA and the nominal transmit current is 28mA. Otherwise the load is 0mA, reflecting an idealized sleep mode. "Frames" are "transmitted" every 100ms, which results in an average duty cycle of about 7%.

The testbed was configured to discharge seven batteries over a period of 10 days. Figure 4 shows two sets of voltage measurements. The V_{oc} measurements are taken just before the load is applied and reflect the maximum recovery from the previous load. This is approximately the open circuit voltage. The V_{load} measurements are taken just before the load is removed and show the battery's decreasing ability to maintain output voltage under load.

The INET IEEE 802.15.4 model can be used to create a simulation scenario that puts a similar load on the simulated battery. To generate the simple load used in the testbed, the simulation consists of two *Host*s, a sender and a receiver,

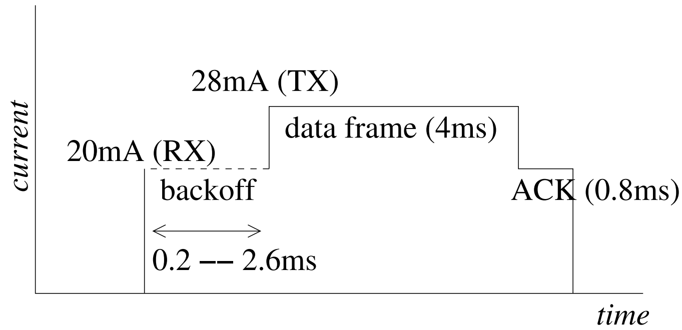

Fig. 3. Load used in simulation and in testbed experiments. Note that because the testbed provides a resistive load, rather than a constant current load, the actual current decreases over the lifetime of the experiment.

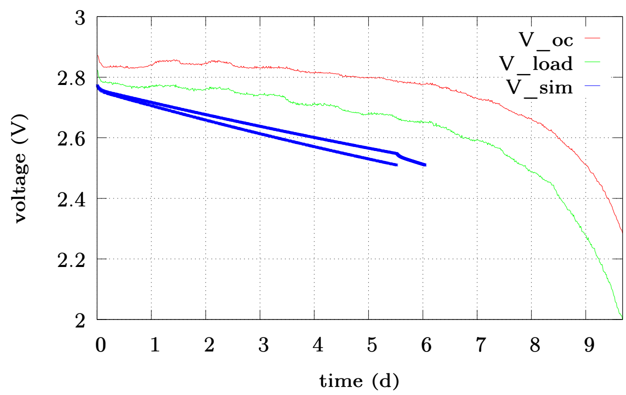

Fig. 4. Testbed and simulated voltages.

connected on an *idealChannel*. Every 100ms, the *Host*s' radios wake up and the sender's *udpAPP* module generates an 88B payload, to which IPv4 and IEEE 802.15.4 headers are added. The *ieee802154NarrowBandMAC* performs a random backoff, transmits (the transmit time is ∼4ms), then waits for and receives an ACK. The *Consumer* parameters are set to the same values that are used in the testbed, resulting in a similar battery load.

This simulation scenario is intended to exercise the battery model, rather than reflect the contention and collision behaviors, wakeup protocols, and multi-hop traffic that would be found in more realistic wireless network simulations and would result in more complex battery loads. In principle, it is possible to take any OMNeT++ simulation trace of the *Consumer* module and convert it to a scripted battery load for the testbed (modulo some testbed hardware limitations). (This is potential future work.)

Some substantial differences are visible between the two results. In particular, the KiBaM model takes the state-of-charge to zero before the experimental discharge and before the equivalent circuit model brings the open circuit voltage to null. The former discrepancy may be partly due to the non-current-controlled load in the testbed, which results in a lower effective load and longer lifetime. The latter discrepancy may be because, although the transient voltages are represented in the KiBaM model, they are not yet fully implemented in the equivalent circuit model. In other words, the battery has a measurable output voltage, but fails as soon as a load is applied – consistent with behavior observed in reality. Nevertheless, careful further study of the validity of the model and its parameterization for the Li coin cell are needed.

V. Conclusion

This paper has presented a preliminary implementation of the KiBaM hybrid battery model for INET. This is a step forward in improving INET's value as a tool for simulating and improving the behavior of networks of battery powered devices. There are, of course, still many limitations and open issues in the work.

With regard to the INET and simulation context, the most immediately visible issue is that the necessary changes to the INET power models is not backwards compatible – the current KiBaM implementation is an untenable hack. An alternative implementation that allows interfaces with different unit parameters would be most desirable.

To make meaningful use of a more detailed battery model also puts some requirements on simulation and model developers. The fundamental problem is ensuring that modeling of the load is appropriately detailed as well.

One factor is the impact of high currents: A current spike associated with a device state change might be ignored under a linear battery model because it consumes relatively few mA-h or occurs only infrequently. However, these operations might have a significant impact on the battery state. This can only be captured in the more sophisticated battery model if this aspect of the load is represented.

It also becomes more important to model all of the components of a device. When using simple battery models based on coulomb counting, the mA-h consumed by the radio, the sensor(s), and the processor could be treated independently. In particular, when evaluating or comparing the energy efficiency of various protocols, the latter could be ignored. A better understanding of both device loads and battery models is therefore needed to give appropriate guidance to developers and encourage more whole system simulation.

Independently of INET, there is considerable future work in improving and validating battery models, including extending the work to other types of batteries, especially rechargeable batteries. Another goal is incorporating temperature into models – this is particularly important for modeling SoC estimation and load balancing.

Acknowledgments

Christian Rohner has made many contributions to the analysis of the hybrid KiBaM model and the creation of the battery testbed. Parts of the OMNeT++ implementation were done in connection with the Computer Networking Project Course (Datakom III) at Uppsala University by students Felix Färjsjö, Andreas Gäwerth, Jonas Nilson, and Eric Stenberg.

References

[1] L. M. Feeney, C. Rohner, P. Gunningberg, A. Lindgren, and L. Andersson. How do the dynamics of battery discharge affect sensor lifetime? In *11th IEEE/IFIP Conf on Wireless On-demand Network Systems and Services (WONS)*, 2014.

[2] T. Kim and W. Qiao. A hybrid battery model capable of capturing dynamic circuit characteristics and nonlinear capacity effects. In *IEEE Transactions on Energy Conversion*, volume 26, 2011.

[3] R. Lim, F. Ferrari, M. Zimmerling, C. Walser, P. Sommer, and J. Beutel. Flocklab: A testbed for distributed, synchronized tracing and profiling of wireless embedded systems. In *Information Processing in Sensor Networks (IPSN), 2013 ACM/IEEE International Conference on*, pages 153–165. IEEE, 2013.

[4] J. Manwell and J. McGowan. Lead acid battery storage model for hybrid energy systems. *Solar Energy*, 50(5), 1993.

[5] Panasonic. Manganese dioxide lithium coin batteries: Individual specifications. http://www.panasonic.com.

[6] R. Rao, S. Vrudhula, and D. Rakhmatov. Battery modeling for energy aware system design. *IEEE Computer*, 36(12), 2003.

[7] C. Rohner, L. M. Feeney, and P. Gunningberg. Evaluating battery models in wireless sensor networks. In *11th Int'l Conf on Wired/Wireless Internet Communications (WWIC)*, 2013.

[8] A. Varga. OMNeT++. In K. Wehrle, M. Günes, and J. Gross, editors, *Modeling and Tools for Network Simulation*. Springer-Verlag, 2010.