

Outline:

- 1 The need for a theory of processing power
- 2 Modeling decoding implementation:
VLSI model of decoding implementation
- 3 Node model of power consumption
- 4 Wire model of power consumption
- 5 Summary and conclusions

Pulkit Grover^{†*}, Anant Sahai[†], Karthik Ganesan^{*} and Jan Rabaey^{*}
[†] Wireless Foundations, UC Berkeley
^{*} Berkeley Wireless Research Center, UC Berkeley

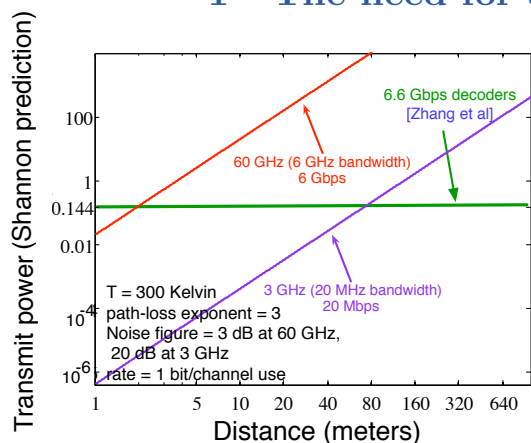
“Green codes with short wires at the decoder: Fundamental limits and constructions”

ITA Workshop, San Diego Feb. 10, 2011.

Slides available: www.eecs.berkeley.edu/~pulkit/files/ITA11Slides.pdf

This Handout: .../~pulkit/files/ITA11Handout.pdf

1 The need for a theory of processing power



Transmit power vs distance for two wireless communication situations. Most current systems are somewhere in between these two cases. Decoding power tends to be larger than Tx power at short distances.

The new short-distance communication systems are challenging the conceptual separation between communication theory and circuit design.

Traditional communication theory focuses *almost exclusively on transmit power*. While at long distances, transmit power dominates all processing power, this is no longer true at short distances (see figure). The decoding power itself can be larger at short distances.

Can we still design codes and circuits in isolation? Or do we now need an expanded theory of information that accounts for circuit power as well?

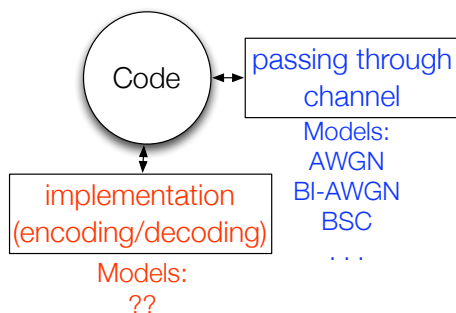
2 Modeling decoding implementation

2.1 Do we need a theory of decoding power?

The most compelling reason comes from the observation that there is the *disconnect* between the code-design community and those who implement them. For instance, while high-degree irregular LDPCs are believed to be good in theory, in practice engineers often use regular LDPCs (e.g. the 10-GBASE-T standard [Zhang *et al.*]), or sometimes no coding at all [Marcu *et al.* '10]!

Evidently, designing codes and decoders in isolation, as we have done for decades, is not working any longer. But why do we need a *theory*? Imagine making transmit-power efficient codes without Shannon theory to guide us. In absence of a theory for decoding power, the sheer number of code/decoding possibilities is too many to experiment.

2.2 The importance of abstractions of decoding power



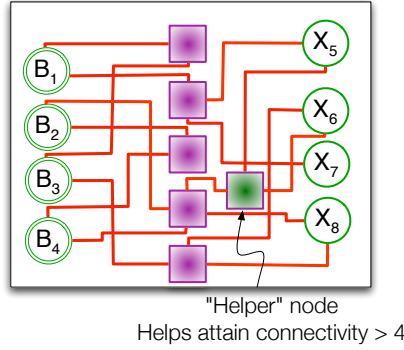
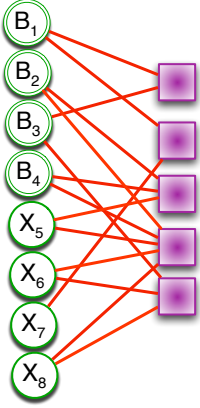
A code is an abstract mathematical object that interfaces with the physical world during implementation (encoding/decoding), and when passing through a channel. Each of these stages consumes power. Our current models are only for the channel, and thus they allow us to construct codes that are transmit-power efficient. In order to be efficient at a system level, just as we have channel models, we need models of decoding and encoding power consumption.

We introduce a model of decoding implementation and two models of decoding power consumption to show that there is hope that such models can be built and used to gain insight into code/decoder design.

A code's interfaces with the physical world

We need models for **implementation** just as we have models for the **channel**

2.3 VLSI model of decoding implementation



A message-passing decoder VLSI abstraction of decoder implementation

The decoding implementation is assumed to consist of computational **nodes** and **wires**. Wiring-density constraints dictate that each node is connected to at most α other nodes ($\alpha = 4$ in the figure). At each time-step of the decoding process (also called an “iteration”), each node at the decoder passes messages to all the nodes that it is connected to [Grover, Woyach, Sahai JSAC '11].

The computational nodes are either message-nodes (denoted by bits B_i), channel-output nodes (X_i, B_i), “helper” nodes, or any combination thereof. The helper nodes simply help in the decoding process (e.g. by computing, such as check nodes, or by increasing connectivity, as shown in the figure).

This model is based on VLSI model of computation introduced by Thompson [Thompson '79].

3 Node model of power consumption for the VLSI model of decoding

Power is consumed in computational nodes and the interconnecting wires. The node model [Grover, Woyach, Sahai JSAC '11] ignores the power consumed in the wires. For simplicity, it assumes that the power consumed per node per iteration is a constant, E_{node} . Typical values of E_{node} are between 0.1 picojoules to 50 picojoules.

The model is a good approximation when the nodes perform complex computations, and wires are short.

3.1 Bounds on computational complexity

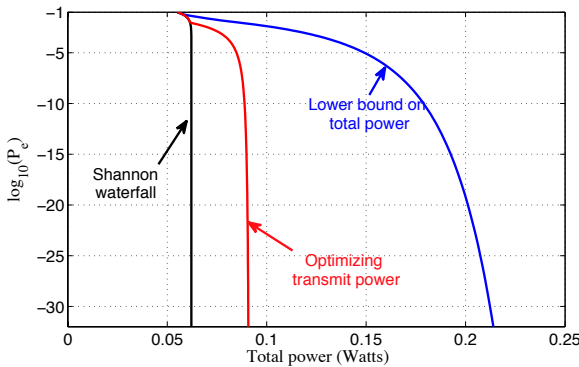
A lower bound on number of decoding iterations t for message-passing decoding is derived in [Sahai, Grover '07] for any code and any message-passing decoding algorithm.

$$t \gtrsim \frac{1}{\log(\alpha - 1)} \log \left(\frac{\log \frac{1}{P_e}}{(C(P_T) - R)^2} \right),$$

based on deriving error exponents for bit-error probability with respect to **neighborhood sizes**.

The bounds predict a **doubly-exponential decay in P_e** with the number of iterations, t , as is evident from a $\log \log \frac{1}{P_e}$ term in the above bound. A “sphere-packing” large-deviations bound with respect to *neighborhood sizes* (instead of the usual block-length) gives the first exponential. The second exponential comes from the observation that the neighborhood size increases exponentially with the number of iterations, with base $\alpha - 1$.

3.2 Fundamental bounds on power consumption



Lower bounds on total power vs error-probability. The total power now diverges to infinity as $P_e \rightarrow 0$. Also shown is the optimizing transmit power which is bounded away from the Shannon limit.

Using the node model with E_{node} joules of energy per node per iteration, we obtain a lower bound on total power which at low P_e takes the form [Grover, Woyach, Sahai JSAC '11]

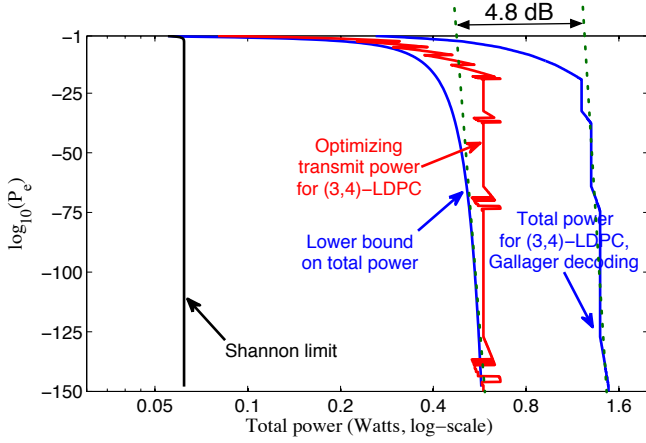
$$P_{total} \gtrsim \inf_{P_T: C(P_T) > R} P_T + \frac{\gamma E_{node}}{\log(\alpha - 1)} \log \left(\frac{\log \frac{1}{P_e}}{(C(P_T) - R)^2} \right).$$

Here γ denotes the “exchange rate” between transmit and decoding power, and depends on the distance between transmitter and receiver and other parameters.

Morals:

- Unlike the traditional “waterfall” curve, the *total* power must diverge to infinity as P_e goes to zero.
- If P_T is small, so that $C(P_T)$ is close to R , then decoding power blows up. Thus there is a **fundamental tradeoff** between transmit and decoding power.
- The optimizing transmit power is bounded away from the Shannon limit. Moral: stay away from capacity!

3.3 Regular LDPCs can be “order optimal”

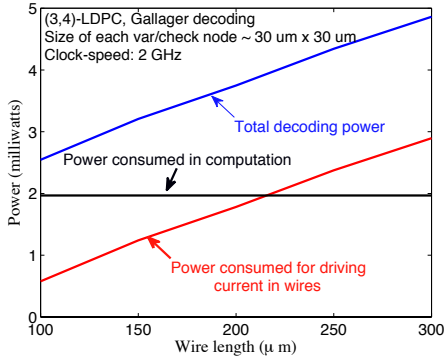


The gap (in log-scale, i.e. dBs) between performance of (3,4)-LDPC and rate 1/4 lower bound is bounded by a constant even as $P_e \rightarrow 0$

- Regular LDPCs use order-optimal total power, *i.e.* as $P_e \rightarrow 0$, they attain within bounded dBs of the optimal [Grover, Woyach, Sahai JSAC '11] (to show this, the x-axis is in log-scale). This is because the error probability for regular- (d_v, d_c) LDPCs decays doubly-exponentially with the number of iterations (for $d_v \neq 2$), which matches the lower bound in order.
- Capacity-approaching LDPCs are highly suboptimal in this model of total power consumption. This is because they require a non-zero fraction of degree-2 nodes, which means that their error-probability decays only exponentially with the number of iterations.

4 Wire model of power consumption for VLSI model of decoding

4.1 The inspiration: a teaching and research “experiment”



Scaling of power consumption with length for LDPC decoders. We assume that the nodes perform simple Gallager-A decoding. Each node at the decoder has size 30 μm x 30 μm .

Our model in [Grover, Woyach, Sahai JSAC '11] considers only node power consumption and ignores power consumed in the wires. We conducted a teaching and research “experiment” where 51 students were asked to design simple LDPC decoders. The results showed that power consumption in *decoder wires can exceed* the power consumption in computational nodes [EE 141 project].

Why do wires consume power at all? A wire induces self-capacitance between itself and the substrate, and also cross-capacitance between wires. It requires power to charge and discharge this capacitance.

The increase of power with wire-length is verified in [Ganesan, Grover, Rabaey '11 in prep.], where we also show that decoders with short wires can consume substantially smaller power.

4.2 Fundamental bounds on wire-lengths [Grover, Sahai CISS '11]

Is designing decoders with shorter wires a problem of pure implementation? The following two results show that the answer is ‘no’. In the following, we assume that each node at the decoder requires A_{node} units of area on-chip

Bounds for regular LDPCs: For decoding a regular (d_v, d_c) -LDPC code of girth g using belief-propagation decoding in a fully-parallel implementation, the length of longest wire W_{max} is lower bounded by

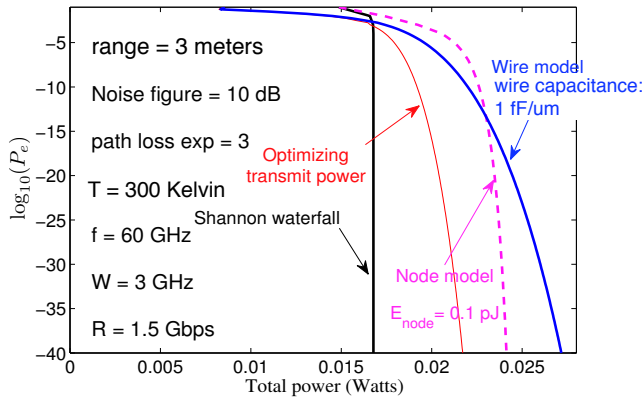
$$W_{max} \gtrsim \frac{\sqrt{A_{node}}}{\sqrt{\pi} \left(\frac{g}{2} - 1\right)} \left((d_v - 1)^{\lfloor \frac{g}{4} - \frac{1}{2} \rfloor} (d_c - 1)^{\lfloor \frac{g}{4} - \frac{3}{4} \rfloor} \right).$$

Bounds for any code: The following bound holds on $t \times W_{max}$, the product of the number of decoding iterations (t) and the length of the longest interconnect (W_{max}):

$$tW_{max} \gtrsim c\sqrt{A_{node}} \frac{\sqrt{\log \frac{1}{P_e}}}{(C(P_T) - R)^2}.$$

The bounds are derived by assuming a two-dimensional chip. The area covered in t iterations is at most $\pi(tW_{max})^2$. Along with the bound on neighborhood sizes [Sahai, Grover '07], this yields the second bound. The first bound is derived similarly using the observation that until time-step $\frac{g}{2} - 1$, there is no node that is repeated in a decoding neighborhood. The bounds appear in [Grover, Sahai CISS'11].

4.3 Bounds on power consumption in the wire model



The behavior of lower bounds on power for the wire model is tighter than that for node model. Further, the optimizing transmit power for the wire-model diverges to infinity as $P_e \rightarrow 0$

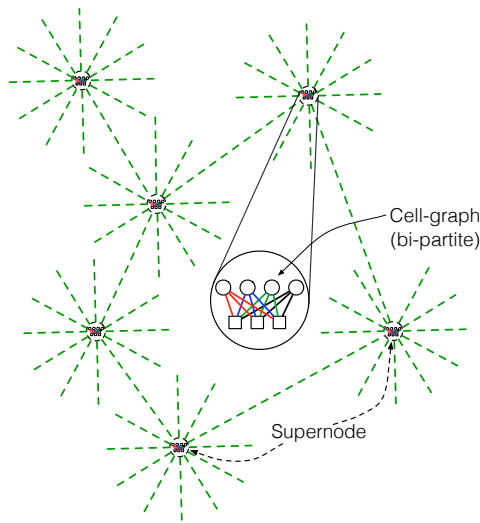
Total power scales at least proportionally to tW_{max} , yielding the following bound

$$P_{total} \gtrsim \inf_{P_T: C(P_T) > R} P_T + \gamma \sqrt{A_{node}} \frac{\sqrt{\log \frac{1}{P_e}}}{(C(P_T) - R)^2}.$$

As a crude bound, we assume that the average length of on-chip wires is same as the maximum.

- These bounds scale to infinity much faster than the earlier “waterslide” bounds.
- The optimizing *transmit* power diverges to infinity.
- Traditional approach is to maximize the girth of a graph. However, this approach can consume more decoding power than the minimum required if the designed codes can attain error-probability lower than desired. Codes should be designed for just the desired performance, and no better!

4.4 Short wires using “cage graphs”



Our constructions for code graphs are based on a “skeleton-graph” constituted by “Supernodes”. “Cell-graphs” are embedded in these supernodes. An edge-exchange between various cell-graphs is used to obtain a bi-partite graph whose girth is at least as large as the girth of the underlying “skeleton-graph”. The problem thus becomes one of searching for regular graphs of high girth and small blocklength.

We provide a simple construction to obtain upper bounds on the required wire-length [Grover, Sahai CISS’11]. We use the fact that density-evolution provides precise error-probabilities for regular LDPC codes as long as each decoding neighborhood is cycle-free. The problem thus becomes of designing codes with specified girth. We need upper bounds on the size of such graphs.

We use constructions for regular graphs of specified girth [Lazebnik, Ustimenko, Woldar ’97] to construct bipartite graphs of specified girth. This is done by first embedding a “cell-graph” that is bipartite into each node of the regular graph. An edge-exchange mechanism then yields a bi-partite graph of specified girth.

Upper bounds on wire-lengths are then estimated by calculating the length of the the wire connecting the farthest nodes. This is a crude metric because in actual implementations, intersecting wires travel in different metal layers on the chip.

Using these crude bounds yields $t \times W_{max} = O\left(\left(\log \frac{1}{P_e}\right)^\zeta\right)$ for some $\zeta > 0$ that depends on the channel and the code construction.

5 Summary

This talk intends to bring out the following ideas:

- Just as we have models for channel to understand transmit power, **we need models of decoding implementation** in order to understand the increasingly relevant decoding power.
- The problem of obtaining tractable models not hopeless. We propose a “VLSI model” of decoding implementation. Using this implementation model, we propose two models of decoding power consumption:
 - **Node model**: total power diverges to infinity as $P_e \rightarrow 0$. Optimal transmit power bounded away from Shannon limit (but bounded above in the limit $P_e \rightarrow 0$). Moral: stay away from capacity!
 - **Wire model**: even optimizing transmit power diverges to infinity as $P_e \rightarrow 0$. There is a fundamental tradeoff between code performance and wire-lengths at the decoder. Moral: design codes for *just* the desired error-probability in order to avoid paying unnecessarily large wiring costs.