Edge Energy Orchestration

Nima Valizadeh[1], Vijay Kumar,[1] Ioan Petri[2], Omer Rana[1], Ambarish Nag[3], Sagi Zisman[3], Charles Tripp[3], and Shuva Paul[4]

[1] School of Computer Science & Informatics, Cardiff University, UK
[2] School of Engineering, Cardiff University, UK – valizadehn@cardiff.ac.uk
[3] Computational Science Center, National Renewable Energy Lab (NREL), Colorado, USA – ambarish.nag@nrel.gov
[4] Energy Security and Resilience Center, NREL, Colorado, USA

Abstract. Edge computing devices have increased in number and capability over recent years. The ability to process data and execute machine learning in proximity to data generation and collection sources provides several advantages over using cloud- based data centers. We describe an orchestration mechanism that enables edge devices to make more effective use of energy resources in their proximity – a technique we refer to as "edge energy orchestration". A software "orchestrator" can take account of renewable generation to alter how task execution on edge devices is carried out. An application scenario is used to illustrate the use of the orchestrator in practice, followed by a discussion about how this approach can be generalized to a broader set of applications

Keywords: energy efficiency · renewable energy use · edge-cloud continuum · resource management

1 Introduction

An edge resource, positioned nearer to the data source, process data locally, thus reducing latency and bandwidth consumption. It acts as an intermediary, delivering faster response times and localized computing power. Ensuring continuous and efficient operation of the edge presents unique challenges and opportunities, especially in cases where edge devices are powered using renewable energy sources such as solar power, hydrogen fuel cells, and wind power. Renewable energy sources, while environmentally beneficial, are inherently variable and depend on factors such as weather conditions and time of day. This variability necessitates adaptive strategies to manage power consumption effectively. The primary goals are to maximize the utilization of edge energy, minimize reliance on cloud computation and data transmission to the cloud, and make the most efficient use of the computational capabilities of edge resources.

We explore three primary use cases at the edge: (i) the edge devices function primarily as a data aggregator, collecting data from various endpoints and transmitting it to the cloud for processing. This approach minimizes energy consumption at the edge of the network by offloading computational tasks to the cloud. However, energy consumption will depend on the size of the data transfer and will require a high network bandwidth depending on the quantity of data to be transferred; (ii) the edge devices perform all necessary data processing locally, providing rapid responses and minimizing the need for data transfer to the cloud. This use case leverages computational capabilities to maximize edge computing while relying on renewable energy sources; (iii) a hybrid approach balancing the computational load between the cloud and the edge. By dynamically partitioning tasks based on real-time energy availability and computational demand, this strategy aims to optimize both energy use and processing efficiency, striving to minimize cloud computation and data transmission.

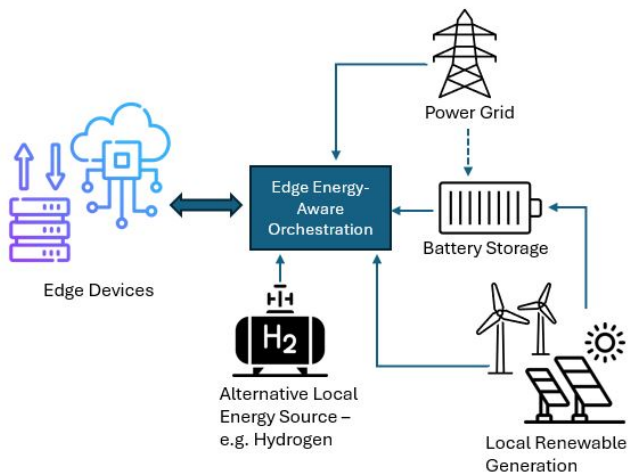

Fig. 1. Application execution using an orchestration engine that takes account of renewable edge energy sources

As illustrated in Figure 1, we describe an energy-aware orchestrator able to use locally available energy sources to schedule applications on edge resources. Such an orchestration mechanism aims to maximize the use of locally available energy sources before connecting to a power grid. Three types of energy resources are considered: (i) local renewable generation, which can have variable frequency and availability. This edge energy source enables devices to directly connect to these sources, and task execution on edge devices can be scheduled based on availability; (ii) a local battery which can be charged through renewable sources. The battery can remove fluctuations in generation from local renewable generation and provide a more reliable source of (local) energy. Specifying the size of the battery needed within a given context and monitoring its state-of-charge variation are two requirements when such a source is used; (iii) alternative local sources of (more expensive) energy may also be available, such as hydrogen, as an alternative. The orchestrator needs to create a schedule for data collection and task execution on available edge resources based on user requests and energy availability. The orchestrator can have several utility functions to influence its operations, such as maximizing the use of locally sourced energy, achieving a throughput target for task execution, and maximizing the use of renewable energy sources rather than the battery of the power grid. We investigates methods for adjusting power consumption at the edge to align with the availability of renewable energy. Identifying how a balance can be established between computational demand and energy availability, contributing to the development of more resilient and sustainable computing infrastructures, is a key contribution. This aspect is covered from two perspectives: (i) monitoring local renewable energy resources to assess their stability and availability profile over a predefined time window; this aspect is covered in 2. An accurate estimation enables direct use of local energy resources; (ii) adaptive use of renewable energy at the edge, smoothing out fluctuations in generation using battery storage, covered in Section 3.

2 Condition Monitoring of Renewable Energy Resources

The production of renewable energy depends on a complex interplay of factors, including the natural variability of resources and weather conditions. Beyond these external influences, the quality and operational condition of renewable energy infrastructure and storage systems are crucial for determining energy output. The efficiency and reliability of solar panels, wind turbines, and large battery storage systems are pivotal. Any faults or degradation in these technologies can cause significant fluctuations and reductions in energy production. Therefore, maintaining high standards in the construction, upkeep and monitoring of renewable energy infrastructure and storage is essential to ensure a stable and reliable energy supply. Schenato et al. [10] describe real time insights, conditional monitoring of renewable energy resources, optimizing distribution networks, enhancing building energy efficiency, and efficiently managing EV charging infrastructure.

Wind energy could benefit from edge processing by improving monitoring systems, reducing downtime and enabling predictive maintenance. Xu et al. [14] have devised an embedded multi-sensor architecture to detect incipient short-circuit in wind turbine electrical generators, that is robust to both false positives and negatives, and enables the testing of five different sensor settings in three feature extraction methods and four classifiers. Abdelmoula et al. [2] proposed a novel framework for monitoring decentralized photovoltaic systems within a smart city infrastructure – using edge computing to overcome challenges associated with costly processing via remote cloud servers.

3 Renewable Energy at the Edge

Lowering of the carbon footprint of edge computing systems in a sustainable way requires both energy efficiency techniques to save power consumption and the use of renewable energy (green energy) as the primary power supply and brown energy (fossil fuel-based energy) as the secondary energy supply. The relatively small energy demand of edge computing systems positions them to make effective use of renewable energy. Utilising a microgrid, distributed renewable energy sources in the same area can be effectively integrated to supply power to local users with less power loss due to transmission and distribution infrastructure and match the dynamic local demand with local supply in a more convenient way. Li et al. [5] proposed an energy management framework that systematically integrates edge computing and the microgrid so that these two systems can cooperate and complement each other to enhance the effectiveness and utilization of energy resources while still satisfying the requirements of IoT applications. The proposed integration methodology reinforces the sustainability of both microgrid and edge computing, by virtue of being tightly coupled with a renewable energy management workflow that enables efficient interaction and collaboration between the systems. For devices operating at the edge, (i) they may be run by single-use (non-rechargeable) batteries, (ii) they may be run by rechargeable batteries that store renewable energy or, (iii) they may be directly connected to the electric grid. However, the most energy-efficient scenario might be determined by a pareto-style optimization, where a linear combination of these three different options might turn out to be optimal.

Sustainable edge servers can also utilize photo-voltaic (PV) panels or micro wind turbines to harvest solar or wind energy from the surrounding environment to enable the scaling and sustainability. Recent experimental results indicate that when the solar power density reaches $600W/m^2$ or the wind speed reaches $11m/s$ (24.6 mph), a $2m^2$ PV panel with 20% energy conversion efficiency, or a 12 kilograms wind turbine with $1.2m^2$ rotor-swept area can generate more than $170W$ of power, which suffices to drive high-performance processors, such as AMD EPYC 7501 and Intel Xeon Gold 6328HL. Given this ability for energy harvesting, edge servers can be deployed outside the coverage of electric grids. A SES (sustainable edge server) needs to dynamically update its computing power based on the energy harvesting rate to achieve the best computational performance, since solar and wind-based energy production are not consistent but highly variable with time. Luo at al. [15] have proposed an optimal computing power management strategy to maximize the average computing power of the solar-powered SES in dynamic renewable energy environments. An energy harvesting model that supports a feedback loop between power consumption to energy generation of the SES is developed.

Edge computing can reduce energy consumption by cloud providers, as data transfer from edge devices and computation at a data center can be minimized. The pervasive nature of edge devices also allows workload balancing, enabling excess tasks to be offloaded to or from a cloud platform. Such mechanisms help coordinate resources and associated tasks by providing more intelligent access to distributed edge resources.

4 Power-Aware Edge implementation

Edge devices can offload computation to centralized servers to enhance user experience. However, this migration incurs energy costs. Jiang et al. [4] discuss offloading strategies, such as local execution, partial offloading [12], and full offloading [4]. Mao et al.,introduce dynamic computation offloading for mobile-edge computing with energy harvesting devices. This approach adapts offloading decisions based on the energy availability of devices. By leveraging energy harvesting information, it optimizes the trade-off between local execution and offloading, ensuring energy-efficient task execution [7]. Wang et al. propose a reinforcement learning-based algorithm, in which mobile users learn from network states and historical behaviors to find optimal energy consumption point and resource allocation policies [13]. By planning offloading base stations for user devices, they achieve a 28% reduction in total energy consumption while ensuring balanced traffic management across base stations [6].

These algorithms aim to strike a balance between energy efficiency and system performance. In another interesting work, Sun et al., propose a joint offloading and computing optimization approach in wireless powered mobile-edge computing systems. By considering both computation offloading and resource allocation, their strategy aims to maximize system throughput while minimizing energy consumption. It dynamically allocates resources to edge nodes, striking a balance between computation tasks and energy availability [11]. Furthermore, in another work by Ahvar et al., which underscores the energy efficiency of distributed computing architectures utilizing foundational energy model and evaluates the consumption of cloud-related architectures, including edge computing [1], our proposed edge energy orchestration mechanism aims to further optimize energy utilization by dynamically aligning task execution with renewable energy availability, which leads to enhancing the the sustainability of edge computing infrastructures. Additionally, the empirical analysis by Mocnej et al. on the impact of edge computing on IoT energy consumption [8] aligns with our objective to enhance energy efficiency, providing a case study that exemplifies the potential for edge computing to extend the operational lifespan of IoT devices through improved energy management.

Energy efficiency in computing devices can be enhanced using various hardware and software modifications, e.g.: (i) underclocking the CPU to reduce the CPU clock speed; (ii) disabling HDMI output when it is not in use; (iii) turning off onboard LEDs to conserve power. Utilizing an efficient power supply and disconnecting unnecessary peripherals, such as USB devices and external drives, further contributes to lower power consumption. Software optimizations involve employing a lightweight operating system to decrease system load, e.g. using Raspbian Lite on a Raspberry Pi can reduce power usage. Power management tools like `powertop` can identify and mitigate power-hungry processes and settings.

Devices can be configured to reduce their CPU clock speed based on the remaining battery power, thereby optimizing energy consumption. This process involves utilizing battery monitoring tools such as `upower` or `acpi` to continuously track the battery level. Software optimizations play a crucial role in enhancing energy efficiency. Utilizing built-in power modes is a primary strategy. For instance, the `nvpmodel` tool on devices like Jetson Nano allows for switching between different power modes, such as setting the device to a 5W mode (mode 1) to reduce power consumption. Similarly, the Jetson AGX Xavier provides more granular control over power consumption through various power modes that can be selected using the `nvpmodel` tool. Enabling Dynamic Voltage and Frequency Scaling (DVFS) on these devices dynamically adjusts the voltage and frequency of the processor according to the workload, optimizing energy usage. Additionally, disabling unused CPU cores and services can significantly reduce power consumption by ensuring that only necessary components are active. Employing power management tools such as `tegrastats` enables the monitoring of power consumption and resource usage, helping to identify and optimize power-hungry processes.

4.1 Power-aware Orchestration

An edge resource orchestrator (EO) is a software component that dynamically determines the placement and scheduling of user applications to: (i) improve utilisation of resources that are in proximity to a user; (ii) meet overall application execution constraints such as deadline, network latency and security. We consider the EO to be hosted on a network component (e.g the first hop router to a user) to undertake this process. The EO may: (i) schedule tasks on locally available edge resource(s), or forward tasks to a cloud system; (ii) aggregate/ divide tasks prior to forwarding these to a cloud system, described in [9]. An EO able to take account of local energy resources is illustrated in figure 1 – connecting computational devices at the network edge with energy sources. This component harnesses data from energy generation and storage, in proximity to edge devices, to influence scheduling of tasks, making most effective use of such energy. An EO in this instance is able to:

- Approximate task deployment based on resource proximity, as edge resources are identified based on geographical proximity facilitating advantages related to cost, latency and security.
- Reduce cost by deploying tasks efficiently: an edge orchestrator can find low cost edge resources where the overall execution is still compliant with a quality of service requirement identified by a user, whilst meeting an energy usage profile.
- Maximize performance: an edge orchestrator can reduce latency associated with data transfer based on existing placement requirements, i.e. when quality-of-solution is important, the edge orchestrator can search for high throughput resources.

5 Edge AI use-cases

We provide a number of scenarios using different types of energy resources – and associated orchestration.

Edge devices with single-use (non-rechargeable) batteries: The Internet of Things (IoT) enables citizens to take informed actions based on data. IoT-generated data can be categorized into logical layers based on where it is generated, how it is used, and the evolving roles of data collectors and users: personal, built environment, district and urban. At the personal level, wearable devices embedded with sensors, like activity trackers, utilize edge AI to gather and analyse data on various physical activities, e.g. counting steps, estimating calorie expenditure, tracking sleep patterns, and recording elevation changes, aiding individuals in monitoring their health. These wearable sensors also leverage edge AI to detect indicators of critical health events such as strokes or traumatic brain injuries (TBI) in patients. Moreover, sensors installed on surfaces (such as battery-powered sofas, chairs, beds) monitor metrics such as heart rate, respiratory signals, movement activity during sleep, and sleep quality – offering sleep analysis and detecting sleep-related issues and identify and prevent poor body postures that may negatively affect health and lead to discomfort and other complications. In these applications, sensors have their own battery that powers data collection. Data collection occurs at limited time intervals, minimising the amount of energy consumption of the device.

Edge devices with rechargeable batteries: recent support for AI/ML-based applications on the edge can make use of ultra-low-power devices with an energy cost below 1mW. This enables the development of many advanced applications in domains where edge computing is favored due to requirements such as high mobility, sustainability, low latency, privacy preservation, and continuous availability.

Optimised LLMs: TinyChat [3] provides an efficient and lightweight system for Large Language Models (LLM) deployment on the edge, that runs Meta's LLaMA-2 model at 30 tokens per second on NVIDIA Jetson Orin and can support different models and hardware. In this approach, direct embedding of LLMs into real-world systems, e.g. the copilot services (coding, smart reply and office) on laptops, in-car entertainment systems, vision-language assistants in robots or vehicular control interfaces enables users to instantly access responses and services without relying on a stable internet connection. Moreover, this approach often bypasses queuing delays associated with cloud services. Running LLMs on the edge not only improves user experience but also relieves privacy concerns, as sensitive data remains localized, which in turn, reduces the potential risk of breaches. A reduction in power can also lead to restricted memory bandwidth and limited peak computation throughput on the edge. Moreover, edge devices have restricted memory capacity. As an example, the NVIDIA Jetson Orin Nano, characterized by its 8GB DRAM, cannot accommodate even the most compact LLaMA-2 model in half precision. TinyChat provides a solution for weight quantization, enabling LLM inference on edge devices with limited memory.

Urban Observatories: Building Level: at a building level, electricity companies are implementing smart meters, enabling citizens to monitor their energy consumption at half-hourly, daily, monthly, and yearly intervals. This data aids in comprehending and itemizing electricity charges, pinpointing energy-intensive appliances by identifying electric usage signature at the edge using federated learning. Occupancy levels within buildings can now be accurately determined through innovative methods utilizing mobile phones and WiFi signals. Both domestic and non-domestic building require energy optimization that often involve deployment and execution of neural networks and genetic algorithms on edge devices. Additionally, depth-based cameras and edge AI analyse the functional movements of individuals, such as vulnerable individuals or patients and serve as alert mechanisms, capable of detecting potentially hazardous events like individuals on the verge of falling, monitoring walking patterns, and identifying specific types of dementia.

Edge devices with renewable power sources - Wind/Solar: At a district level, edge AI utilises the data available around the environment (air quality/temperature, wind speed/direction, traffic delays), location of EV charging points, car parking spots and availability to optimize traffic flow, manage waste collection, enhance public safety, and improve overall urban living conditions. Further, information about crime maps, neighborhoods, past activity can help to predict crime in near future.

At the urban level, the integration of edge AI technologies enhances the monitoring and management of environmental factors. Authorities often deploy a network of sensors equipped with edge AI capabilities to collect real-time data on air quality, temperature, humidity, and barometric pressure throughout the city and its surroundings. This advanced sensor network enables not only the identification of sources of air pollution, such as power plants, road transport, and industrial processes, but also the analyses of complex data patterns to predict air quality levels. By combining air pollution data with meteorological information authorities can accurately forecast air quality trends and understand how they are influenced by seasonal variations and weather conditions. Furthermore, edge AI-driven analytics provide insights into energy demand patterns by processing meteorological data, including rainfall and solar energy levels. This enables city councils to optimize energy management strategies and enhance the efficiency of household energy consumption.

City councils employ sensor technology, including cameras, microphones, and edge computing, to monitor street activities such as pedestrian and cyclist traffic in intersections and parks. This data helps authorities improve citizen services and manage crowds effectively, including coordinating with law enforcement when necessary. Thermal cameras near harbours and water bodies aid in detecting individuals at risk of falling into the water. Furthermore, data on vehicle purchases and ticketing for various modes of transportation enable transport authorities to analyse mobility patterns and optimize services, such as adding more trains to crowded stations. Automatic Plate Number Recognition (APNR) systems track vehicle movements at city borders and monitor the distribution of petrol, diesel, hybrid, and electric vehicles. This information informs efforts to address air pollution and gauge public acceptance of electric cars. Moreover, edge machine learning is utilized to enhance road infrastructure management,

with applications like road damage detection improving productivity and reducing costs for city councils. This technology ensures safer road conditions by addressing issues such as faded lane markings and graffiti on street signs, thereby enhancing overall traffic safety.
Decentralised AI and variable (decentralised) energy sources: Edge computing can also enable decentralized operation of AI systems, thereby reducing the need for large-scale data centers. Decentralized AI systems lower the risk of downtime and improve the reliability and availability of AI systems – in addition to reducing power requirements of centralised data centers. In edge computing, the volume of data traversing the network can be reduced greatly, which in turn can free up bandwidth. This is more efficient from both time and energy perspective to work with the data on the edge and send the data to the cloud only if it is really needed there for aggregation and other manipulations. Moreover, bypassing the requirement for voluminous data storage lowers the demand for power-hungry data centers. Ait

6 Conclusion

The need for supporting an energy orchestrator at the edge of the network has been identified. The orchestrator is able to utilise energy generation and usage “signals" to influence how computational tasks can be scheduled and managed on edge devices. Whereas previous work has primarily focused on undertaking partitioning of tasks between edge and cloud resources to meet quality of service targets (such as latency, throughput, response time, etc.) – this work highlights the need to also maximise the use of energy generation in proximity to edge resources.

A number of application *classes* have been identified that make use of different types of energy sources: from non-rechargeable battery use for limited data acquisition and transmission/storage, to city-scale infrastructure that is able to harness power generation across a number of locations across a city, able to take account of various renewables and alternative forms of energy (such as hydrogen). The Urban Observatory is used as a common infrastructure to illustrate these different uses of edge resources. An orchestrator that is able to respond to varying needs of these application classes, and able to adapt its behaviour is a key requirement identified in this work.

References

1. Ahvar, E., Orgerie, A.C., Lebre, A.: Estimating energy consumption of cloud, fog, and edge computing infrastructures. IEEE Transactions on Sustainable Computing **7**(2), 277–288 (2022). https://doi.org/10.1109/TSUSC.2019.2905900
2. Ait Abdelmoula, I., Idrissi Kaitouni, S., Lamrini, N., Jbene, M., Ghennioui, A., Mehdary, A., El Aroussi, M.: Towards a sustainable edge computing framework for condition monitoring in decentralized photovoltaic systems. Heliyon **9**(11), e21475 (Nov 2023). https://doi.org/10.1016/j.heliyon.2023.e21475, https://linkinghub.elsevier.com/retrieve/pii/S2405844023086838

3. Haotian Tang, Yang, S., Lin, J., Tang, J., Chen, W.M., Wang, W.C., Han, S.: TinyChat: Large Language Model on the Edge (Sep 2023), https://hanlab.mit.edu/blog/tinychat
4. Jiang, C., Fan, T., Gao, H., Shi, W., Liu, L., Cérin, C., Wan, J.: Energy aware edge computing: A survey. Computer Communications **151**, 556–580 (2020). https://doi.org/https://doi.org/10.1016/j.comcom.2020.01.004, https://www.sciencedirect.com/science/article/pii/S014036641930831X
5. Li, W., Yang, T., Delicato, F.C., Pires, P.F., Tari, Z., Khan, S.U., Zomaya, A.Y.: On Enabling Sustainable Edge Computing with Renewable Energy Resources. IEEE Communications Magazine **56**(5), 94–101 (May 2018). https://doi.org/10.1109/MCOM.2018.1700888, https://ieeexplore.ieee.org/document/8360857/
6. Lv, X., Ge, X., Zhong, Y., Li, Q., Xiao, Y.: Energy consumption optimization for edge computing-supported cellular networks based on optimal transport theory. Science China Information Sciences **67**(2) (Jan 2024). https://doi.org/10.1007/s11432-023-3855-5
7. Mao, Y., Zhang, J., Letaief, K.B.: Dynamic computation offloading for mobile-edge computing with energy harvesting devices. IEEE Journal on Selected Areas in Communications **34**(12), 3590–3605 (2016). https://doi.org/10.1109/JSAC.2016.2611964
8. Mocnej, J., Miskuf, M., Papcun, P., Zolotova, I.: Impact of edge computing paradigm on energy consumption in iot. IFAC-PapersOnLine **51**(6), 162–167 (2018). https://doi.org/10.1016/J.IFACOL.2018.07.147
9. Petri, I., Rana, O.F., Zamani, A.R., Rezgui, Y.: Edge-cloud orchestration: Strategies for service placement and enactment. In: IEEE International Conference on Cloud Engineering, IC2E 2019, Prague, Czech Republic, June 24-27, 2019. pp. 67–75. IEEE (2019). https://doi.org/10.1109/IC2E.2019.00020
10. Schenato, R.: Empowering the Energy Sector; edge computing solutions for a sustainable future (Feb 2024), https://sixsq.com/blog/discover/2024/02/27/edge-computing-solutions-for-energy-sector.html
11. Sun, H., Zhou, F., Hu, R.Q.: Joint offloading and computation energy efficiency maximization in a mobile edge computing system. IEEE Transactions on Vehicular Technology **68**(3), 3052–3056 (2019). https://doi.org/10.1109/TVT.2019.2893094
12. Tang, Q., Lyu, H., Han, G., Wang, J., Wang, K.: Partial offloading strategy for mobile edge computing considering mixed overhead of time and energy. Neural Comput. Appl. **32**(19), 15383–15397 (oct 2020). https://doi.org/10.1007/s00521-019-04401-8, https://doi.org/10.1007/s00521-019-04401-8
13. Wang, Y., Dai, X., Wang, J.M., Bensaou, B.: A reinforcement learning approach to energy efficiency and qos in 5g wireless networks. IEEE Journal on Selected Areas in Communications **37**(6), 1413–1423 (2019). https://doi.org/10.1109/JSAC.2019.2904365
14. Xu, Y., Nascimento, N.M.M., De Sousa, P.H.F., Nogueira, F.G., Torrico, B.C., Han, T., Jia, C., Rebouças Filho, P.P.: Multi-sensor edge computing architecture for identification of failures short-circuits in wind turbine generators. Applied Soft Computing **101**, 107053 (Mar 2021). https://doi.org/10.1016/j.asoc.2020.107053, https://linkinghub.elsevier.com/retrieve/pii/S1568494620309911
15. Yu Luo, Lina Pu, Liu, C.H.: Computing Power and Battery Charging Management for Sustainable Edge Computing (Jun 2024), https://my.ece.msstate.edu/faculty/chliu/papers/journal/CompPower.pdf