Computing offloading method with low power consumption constraint for smart grid

Yuebin WU\textsuperscript{1,a}, Pengcheng LU\textsuperscript{1,b}, Xiande BU\textsuperscript{2,c}, Shidong LIU\textsuperscript{2,d}, Cheng ZHONG\textsuperscript{1,e}, Chuan LIU\textsuperscript{2,f}

\textsuperscript{1}Xiongan New Area Power Supply Company, State Grid Hebei Electric Power Co. LTD, Xiongan, China
\textsuperscript{2}Global Energy Interconnection Research Institute, State Grid Laboratory of Electric Power Communication Network, Nanjing, China
\textsuperscript{a}xa_wuyb@he.sgcc.com.cn; \textsuperscript{b}13398619196@163.com; \textsuperscript{c}buxiande@geiri.sgcc.com.cn; \textsuperscript{d}liushidong@geiri.sgcc.com.cn; \textsuperscript{e}hebeizhongc@163.com; \textsuperscript{f}liuchuan@geiri.sgcc.com.cn

Abstract—With the development of the Power Internet of Things, power transmission and transformation business have a wide range of operation, large regional differences, a wide variety of equipment and other problems, making the access and deployment of mass sensors with low power consumption crucial. In order to realize the cloud-side collaborative computing of the Internet of Things, it is necessary to build a low-power information acquisition and transmission network, and realize the low-power deployment of the low-power sensor network and equipment. The embedded node device can be used as the basis of low-power transmission network to realize low-power transmission of cloud-side collaborative computing information, combined with the existing low-power transmission and transformation equipment.

1. INTRODUCTION
Internet of Things technology has been widely used in smart transmission networks. Edge computing devices have been deployed on transmission lines to realize intelligent perception and local intelligent processing of line status. However, the application of high-performance equipment improves the intelligent level of line monitoring, but also raises the requirements on the power supply of equipment. Only improving the capacity of battery power supply cannot fully meet the power supply requirements of the equipment. Therefore, it is necessary to reduce the requirements on the power supply of the equipment based on the working mechanism of the equipment.

2. LOW POWER CONSUMPTION REALIZATION MECHANISM OF EMBEDDED DEVICE
An important characteristic of embedded system is the non-uniformity and dynamic variation of workload. Since the workload of an embedded system typically changes over time, a balance between system performance and power consumption can be achieved by switching off the device or dynamically adjusting the operating voltage of the processor. After years of research and development, these two technologies have now been proposed at many levels of the system and have become the dominant techniques in the dynamic low power technology design process [1-2].
2.1 Dynamic Power Management (DPM)

The basic prerequisites for DPM application are as follows. A system, or system unit, will be under an uneven workload during normal operating hours. The non-uniformity of workload is a very common phenomenon in embedded systems and most interactive systems. The essence of DPM technology is to selectively set system resources to low-power mode according to the changes of system workload, so as to achieve the purpose of reducing system energy consumption. System resources can be used to construct a corresponding model using an abstract diagram of working state, in which each state is a compromise between performance and power consumption. For example, a system resource may have two working modes: normal and sleep. Among them, sleep state has low power consumption, but it takes some time and energy cost to return to Normal state. Switching behavior between states is controlled by commands sent from the Power Management (PM) unit, which observes the workload to determine when and how to switch modes of work. The power minimization (or maximization) strategy model under performance constraints is a constrained optimization problem.

The basic idea of DPM is to view a workload as a collection of multiple task requests. For example, for hard disks, task requests are read and write commands. For the network adapter, the task request contains two parts of packet sending and receiving. The device is in busy state when there are requests. Otherwise, the device is in idle state. Based on this concept, during T1-T4 period, the device is in idle state, in which it is possible to enter sleep low-power working mode. The device is turned off at point T1 and woken up at point T4 when a task request is received. It takes time for this state to change. In the case of hard drives or displays, it can take a few seconds to wake up these devices. It also takes extra power to wake up a device in sleep mode. In other words, changes in the operating mode of the device will inevitably incur additional overhead. Without this additional overhead (both time and energy), DPM itself would not be necessary. Any device enters idle once, status immediately turns it off. Therefore, a device should go into Sleep mode only if the energy savings justify the extra cost. The rules that determine whether a device is worth shutting down are called policies. In the process of power management, we generally only consider the power consumption of the device in Idle state, but not in Busy state.

2.2 Dynamic Voltage Regulation (DVS)

Recently, with the development of commercial CMOS chip power supply technology, it is possible to adjust the operating voltage of processor core in real time during operation. The emergence of efficient DC-DC voltage converter also provides conditions for the dynamic adjustment of the operating voltage of the processor core. In addition, in soft real-time system, tasks only need to be completed before the specified deadline to meet the performance requirements of the system, but there is no requirement for immediate response of the system. DVS technology dynamically adjusts the processor operating voltage according to the urgency of the task to achieve a balance between task response time and low power consumption of the system.

DPM technology can provide significant energy savings for non-real-time systems. However, DPM cannot be applied to real-time system due to its inherent probabilistic characteristics and uncertainty. DVS technology is a good solution to the performance and power consumption requirements of embedded real-time system, which can adjust the processor operating voltage according to the performance requirements of the current running task. DVS technology is primarily based on the fact that the energy consumption of the processor is square proportional to the operating voltage. If only the frequency of the processor is adjusted, the energy savings will be limited. This is because power consumption is inversely proportional to cycle time, and power consumption is proportional to execution time and power consumption. The early DVS principles set the speed of the processor based on its utilization and did not take into account the different requirements of running tasks. Some voltage regulation strategies have been proposed for real-time systems. For example, processors with continuous and discrete voltage variations are discussed. To reduce the voltage that continuously changes processor power consumption, a specific voltage needs to be found for each task to extend the
entire execution time to the corresponding deadline. For processors with discrete voltage variations, at least two execution voltages need to be found for each task. The problem of joint online scheduling of periodic and non-periodic tasks is considered. This principle can ensure that the deadlines of all periodic tasks are met and the number of non-periodic tasks accepted is maximized [3-5].

3 LOW POWER NODE DEVICES ADAPTED TO POWER TRANSMISSION AND TRANSFORMATION IOT

3.1 Hardware Architecture
In view of the growing power business of the IoT, such as overhead line monitoring, cable multidimensional perception and other services, node equipment is required to have higher processing capacity. The high-intensity work of node equipment inevitably brings power consumption problem. In this section, the hardware module circuit design and low power scheme design are processed.

Fig.1. Hardware architecture design of node devices adapted to power transmission and transformation Internet of Things.

Edge computing node devices are mainly used in transmission lines and substation multi-service scenarios. The hardware of the device includes the CPU module, power module, and sensor unit, as shown in Figure 4-1. This section studies the hardware architecture design of node devices adapted to the IoT.

3.1.1 Power supply and management unit: In the transmission and transformation environment, it is difficult to get power, and more battery power is used. The power supply mainly uses lithium battery, and has battery management unit, which mainly completes battery charge and discharge management, electric quantity detection, temperature monitoring, system power supply management.

3.1.2 Sensor unit: Real-time information interaction with sensor nodes. According to the business type of the IoT, a variety of sensors are applied in different scenarios, including temperature, humidity, tension and other sensors.

3.1.3 Control unit: With low-power MCU as the core, on the one hand, edge computing can be carried out, and at the same time, local communication can be carried out through wireless modules. Cloud-side collaborative information interaction can also be carried out with the main station of power transmission and transformation business through remote communication units.

3.1.4 Telecommunication unit: According to the real-time demand of power transmission and transformation service, communication unit should have strong data transmission capability. The communication network of power-saving equipment should be equipped with the functions of special wireless power networks, 4G or 5G public network communication modules.

3.2 Low Power Circuit Method
In view of the low power performance requirements of power transmission and transformation edge node equipment, this section studies the low power of hardware units from the aspects of circuits of each module. The devices include digital CMOS circuits, sensor unit circuits, and CPU circuits to reduce hardware complexity, miniaturize and integrate devices, and reduce low power consumption, as shown in Figure 2.
For low power circuit design, the principle of simplifying circuit should be followed, to reduce useless power consumption. Firstly, the power consumption can be effectively reduced by reducing the length of the wire and the load capacitance and the working frequency. When the signal activity is zero, even if the load capacitance is large, the circuit does not consume energy. Therefore, in practical work, when a certain system or module of the circuit does not work and is in a dormant state, the clock of these systems can be shielded, so that part of the circuit can stop working and flipping, so as to reduce the power consumption of the circuit.

### 3.2.1 Digital CMOS circuit
Digital CMOS circuit is a common low power circuit, which should be selected first for hardware modules and units in node devices. Because short-circuit current power consumption and driving voltage, switching current power consumption is mainly related to load capacitance, driving voltage, turnover frequency. The main source of CMOS circuit power is dynamic power, which consists of two parts: switching current and short-circuit current. Dynamic power consumption is proportional to the square of the operating voltage, and proportional to the switching frequency and load capacitance.

### 3.2.2 Power design
In power supply design of transmission and transformation equipment, reducing working voltage is an effective way to reduce power consumption. If the rest of the device's power circuit is exactly the same, reducing the operating voltage to 3V can save 75% of power consumption, which indicates the effect of reducing voltage to save energy is of great significance. Therefore, a careful design of the operating voltage of the circuit and the design of an efficient power supply to reduce the power loss in the voltage conversion can reduce the power consumption of the device.

### 3.2.3 Component selection
The selection of low-power components of the equipment should meet the logic circuit that saves the number of devices and select small devices to reduce the load capacitance, so as to effectively reduce the power consumption of the equipment. In order to give consideration to low power consumption and high efficiency at the system level, the requirements of practical applications must be considered. For example, in special power transmission and transformation environment, the device sensor can turn off the MCU master clock and CPU, and only turn on the low-frequency clock and wake up the peripheral circuit for detection. When an event that meets the specified conditions occurs, the CPU to process the event should be quickly started, so that the CPU can maintain the standby state of the sensor network on the node device.

In view of the special application scenarios of the Power transmission and transformation Internet of Things, the hardware circuits of edge computing node devices are analyzed in detail. According to the actual situation, the design satisfies the different hardware circuit low power consumption scheme. The design of low power circuit follows the principle of digital CMOS circuit. Low power circuits are enhanced according to specific functions between different cell modules [6-7].

## 4 Offloading and Offloading Strategies of Mobile Edge Computing in the Internet of Things
Mobile edge computing offloading technology refers to a technology that can unload part or all of the tasks to the cloud with more superior and sufficient resources when the resource capacity of the device cannot meet the requirements of the tasks with limited resources and high service demands. This
approach addresses many of the shortcomings of traditional networks, such as allocating user resources, improving service performance and reducing power consumption.

4.1 Off-loading Process
The task unloading process of mobile edge computing is mainly composed of six steps, which are described as environment perception, task allocation, unloading decision, unloading task transmission, MEC task execution and task processing result return. The decision to unload a task is critical throughout the process. That is, the user chooses the appropriate decision-making mode according to the task characteristics, which is the key factor to determine the user service experience in the system. Figure 3 shows the process for off-loading tasks

According to the previous introduction of the system framework for moving Edge Computing (MEC), MEC is an infrastructure architecture that operates as a virtual machine. In the MEC system, when a user unloads a task, the task will be first unloaded to the MEC within the communication range. The edge server develops a VM environment corresponding to the task submitted by the user, and then provides the corresponding service to the user. When the MEC server completes the computing task off-loaded by the user, the processing result of the task is fed back to the client through the backhaul link. The MEC server then destroys the virtual machine, along with some reclaiming of available resources. In general, the MEC paradigm can be divided into two categories: binary unload and partial unload. With binary off-loading, computing tasks at the user cannot be partitioned, but must be executed at the user or at the edge cloud. If partial unload is used, tasks can be divided at the user's office so that local computation and unload can be performed.

4.1.1 Off-loading environment awareness: When users in the system have tasks to arrive, UE will first perceive the current network environment. Such as detecting the current wireless channel environment and the working status of edge computing servers that are expected to provide services for their own tasks. These factors will determine the user's off-loading decision to some extent. Therefore, this stage can be regarded as the preparation stage before task off-loading.

4.1.2 Task assignment: When a task is request from the user, it can be divided to some extent before unload. This division can make it easier for users to make decisions on tasks. If the processing capabilities of the MEC in the system are inconsistent, users can select the MEC based on the task requirements. For example, if a computationally intensive task is performed, users may be willing to offload to a server with better resources in order to achieve lower latency.

4.1.3 Off-loading decision: The essence of the user's off-loading decision is to decide whether the current task needs to be off-loaded, the appropriate proportion of off-loading, and the appropriate power to submit the off-loaded data, etc. In the system, when the user makes the decision of task unloading, it usually adopts the appropriate decision algorithm, along with various indicators of task requirements. For example, time constraint of task processing, task energy consumption, service experience, user cost and many other factors are taken to weigh it and then make the best unloading plan for itself.
4.1.4 Task transfer: According to the steps mentioned above, after the user has made the off-loading decision, the task can be delivered with the MEC and submitted to the edge cloud server for execution. Edge computing servers have more processing power and superior resources than end users, and are geographically closer to users than traditional cloud centers, which leads to very direct benefits in terms of low latency and low energy consumption. Therefore, it has more advantages than traditional cloud offloading, so it can bring better and more convenient service to users.

4.1.5 MEC server execution: After the user makes the decision about the task and submits it to the MEC server, the edge server assigns virtual machines to the off-loaded task and uses its own resources, such as computing resources, to process the task submitted by the user.

4.1.6 Results back: This is the last step in the unloading process. After the edge cloud completes the task delivered by the user by using its own resources, the result of task processing will be sent back to the user through the backhaul link. Users receive the feedback results of MEC, and then process and use them according to their own needs. The user at this stage may unload the delivery again in response to the received volume processing results, or terminate the request and disconnect from the edge server.

4.2 Off-loading Model
The unloading model of tasks directly affects the decision making of unloading. For the user's computing task model, there are mainly two unloading models, namely the two-state unloading model and the partial unloading model of the task.

4.2.1 Binary off-loading: The dimorphic unloading model considers each computing task as an indivisible and independent unit. An independent computing task can be processed locally (without unload) or unloaded to the MEC server (unload). Typically in this type of unloading model, the task is represented by a triplet $\lambda (D,C,T)$. D represents the number of input bits of the task. C represents the computing resources required by the task, the number of available CPU cycles. T represents the time limit for completing the task.

4.2.2 Partial off-loading: For partial off-loading, computing tasks can be regarded as further subdivided into sub-task modules. For each subtask, either execute locally or off-loading can be processed. Thanks to the development of code decomposition and parallel processing technology, this model has been paid more and more attention. Due to the refinement of unloading granularity, through the parallel execution of each sub-task, computing tasks can be completed more effectively and faster and computing resources can be utilized more rationally, thus improving unloading efficiency. In addition, there are dependencies among subtask modules in the partial unload model (The output of one module may be the input of another module). The order in which each subtask is executed should be considered. Although the partial unloading model further refines the unloading granularity and improves the unloading efficiency, the partial unloading model must consider the execution order between sub-tasks. However, in multi-user scenarios, it is too complex to consider the execution order of each user's tasks, unload decisions and resource allocation at the same time. Therefore, the two-state unloading model of tasks is generally considered in multi-user scenarios, while the partial unloading model is more common in single-user scenarios.

4.3 Formulation of Task Unload Decision
Edge computing technology is applied to transmission Hub Net, which aims to realize local data processing and analysis and transmission station operation and maintenance control. However, due to the limitations of computing, storage and power consumption, Hub Net can unload the monitoring video analysis, electricity analysis and other computation-intensive tasks to the master station server for operation. Only sending computing tasks and receiving results are needed. There is no need to take
up local computing and storage resources, which can effectively solve the problem of local resource constraints. The computational unload technology actually solves the joint optimization problem of computing resources and communication resources allocation in the technical support system of transmission stations.

The decision of task unloading mainly considers the execution location (where), execution time (when), which tasks (what) to off-loading, and how much wireless resources and computing resources to allocate to each task. At present, there are two main decision-making modes: centralized control and distributed control.

4.3.1 Central station control
Centralized computational unload control is a centralized optimal control based on a global perspective by integrating the wireless channel state in the network, the number of resources in the network, the number of tasks requested at the same time and the requirements of each task. The objective is to obtain the optimal results of scheduling, resource allocation and unloading decisions. Generally, the role of the centralized control unit is assumed by the base station or MEC server. In the multi-cell collaboration scenario, the centralized control unit can be a macro station, a unified aggregation point, or an SDN controller with a global perspective to achieve the separation of control signaling and specific execution.

4.3.2 Distributed control
Distributed computing offload control, which focuses on the specific preferences of each user/computing task, is an equilibrium rather than optimal result for multiple users to maximize their utility function. Each user is not only the initiator but also the decision maker of task unloading, which is usually modeled by game theory or auction model. In specific research, users can make the choice according to the focus and scene of the research problem. Centralized control can obtain optimal (or sub-optimal) task scheduling results from the global perspective, but the complexity is large. Distributed control is widely used in the Internet of Things. Because of the massive device connection, the distributed control method can get the decision result quickly with low complexity.

4.4 Calculation unloading politics
The main factors affecting the decision of off-loading are network transmission technology, basic resources of Hub Net, off-loading of application program and basic resources of host server. The most important factors to be considered are delay and energy consumption of Hub Net transmission.

4.4.1 Unloading decisions aimed at reducing delay
For computing tasks requiring high real-time performance, the offloading decision aiming at reducing delay should be adopted so as not to affect the quality of service when the computing tasks are offloaded to the master station. If the computing task is executed locally, the time taken is the computing time, i.e.,

$$T_E = \frac{m}{v_E}$$  \hspace{1cm} (1)

Where, $T_E$ is the local execution time of a computing task, $m$ is the computing resources required to complete the computing task, and $v_E$ is the local execution speed.

If the computing task is unloaded to the primary server, the total time spent is composed of three components.

That is, the time for transmitting the data required for computing to the master site, the time for the master site to perform the computing task, and the time for the master site to receive the data returned from the master site. That is,
In the formula, $T_C$ is the time required for unloading the calculation task to the master station, and $b_1$ is the size of data transmitted to the master station; $B_1$ is the current network bandwidth; $v_c$ calculates the speed of the master station; $B_2$ is the size of the returned data; $B_2$ is the network bandwidth at return time.

If $T_E > T_C$, that is, the local execution time exceeds the execution time when the computing task is off-loaded to the primary site, the computing task is off-loaded to the primary site to meet the minimum delay requirement.

If $T_E < T_C$, that is, the local execution time is less than the off-loading time to the primary site, the computing task is executed locally to meet the minimum delay requirement.

### 4.4.2 Unloading decisions aimed at reducing energy consumption

Low power consumption is an important index of Hub Net. Unloading some computing tasks to the master station will greatly reduce the energy consumption of Hub Net transmission. Calculate the power consumption of a locally executed task.

$$E_E = P_E \cdot \frac{m}{v_E}$$

In the formula, $E_C$ is the energy consumption of the local execution of the task. $P_E$ is the operating power of Hub Net.

The energy consumption of a computing task unloaded to the master station consists of two parts: the energy consumption of unloading data transferred to the cloud and the energy consumption of receiving returned data.

$$E_C = P_1 \cdot \frac{b_1}{B_1} + P_2 \cdot \frac{b_2}{B_2}$$

In the formula, $E_C$ is the energy consumption of the unloading task to the main station. $P_1$ is the transmission power during data upload. $P_2$ is the transmitted power when data is received.

If $E_E > E_C$, that is, the energy consumption of the local computing task exceeds that of the local computing task being off-loaded to the primary site, off-load the computing task to the primary site to meet the minimum power consumption requirement.

If $E_E < E_C$, that is, the energy consumption of the local computing task is less than that of the off-loading to the primary site, the local computing task is executed locally to meet the minimum power consumption requirement.

### 4.4.3 Unloading decision aiming at tradeoff delay and energy consumption

Different application scenarios have different requirements for delay and energy consumption. How to take two optimization objectives into consideration is an important factor in unloading decision. The total transmission cost can be set as $Q = \alpha T + \beta E$, $\alpha + \beta = 1$, $\alpha, \beta$ are the weight coefficients of delay and energy consumption respectively. Therefore, unloading decision is based on joint optimization of computing and communication resource allocation.

### 5 Tests and Results

Figure 4 shows the relationship between the total energy consumption of the system and the calculation demand. The results show that the energy consumption of the three schemes increases with the increase of computing requirements. Similarly, when only local calculation is performed, the energy consumption is the largest, and the MEC scheme has the middle performance, and the complete unloading scheme has the lowest power consumption. The performance of the scheme in this
paper is significantly better than other schemes. With the increasing of computing tasks, the gap between the proposed scheme and other schemes is basically expanding. In order to meet the requirement of task delay, when computing tasks increase, more tasks will be unloaded to optimize the system energy efficiency.

![Energy consumption VS computing demand](image)

**Fig.4.** Power consumption in different task off-loading modes

## 6 Conclusion

This paper proposes a low-power implementation method for the edge devices of the Power Internet of Things. It uses computing offloading technology to solve the problem of the distribution and joint optimization of computing and communication resources when energy is limited, and occupies less local computing and storage resources, so as to effectively solve the problem of local resource limitation. At present, the computational unloading technology has been widely used in the transmission lines of urban distribution network. In the state of no charge, the equipment can be extended twice the working time. In the future, the hibernation technology can be combined to further extend the working time of the equipment and effectively reduce the power consumption of the equipment.

## Acknowledgment

This work is supported by the Science and Technology Project of State Grid Corporation of China: Research and application of key technologies for wireless communication network coverage of urban power underground pipe gallery (Grant No. 5700-202113189A-0-0-00).

## References

[1] Y. Chen, L. Wei, and Q. Jiao, “A Low Power Communication Technology for Power Line Status Surveillance,” Automation of Electric Power System, vol. 36, No.24, pp. 69-73, December 2012.

[2] Y. Xie, “Design of Low Power Consumption and Real-Time Monitoring System of Micro-meteorological for Power Transmission Line”, Process Automation Instrumentation, Vol.39, No.4, pp.33-36. April 2018.

[3] L. Gou, J.Q. Jia and S. Cao, “Analysis of Special Network of Energy and Power IoT Based on LPWAN Technology,” Telecom Power Technology, Vol.37, No.15, PP.127-129, August 2020.

[4] L. Gou, J.Q. Jia and S. Cao, “Analysis of Special Network of Energy and Power IoT Based on LPWAN Technology,” Telecom Power Technology, Vol.37, No.15, PP.127-129, August 2020.

[5] Internet of Things in 2020, Roadmap for the future. http://www.smart-systems-integration.org/public/internet-ofthings, 2010.

[6] Z. Nie, J.M. Zhang and H.W. Fu, “Key Technologies and Application Scenario Design for Making Distribution Transformer Terminal Unit Being a Containerized Edge Node,” Au

[7] M. Young, The Technical Writer’s Handbook. Mill Valley, CA: University Science, 1989