Ministry of higher Education and Scientific Research
University of Saad Dahlab - Blida 1
Faculty of Science
Department of informatics

Performance evaluation of a service differentiation mechanism in Wireless Sensor Network

Report submitted for the fulfillment of the Master degree
Domain: MI
Affiliation: Informatics
Specialization: Computer Systems and Networks

Presented by:
Demerdji Hacene
Ferhaoui Souhaib

In front of the jury:
Supervisor Mrs. Boutoumi Bachira
President Mrs. Cherfa Imane
Examiner Mrs. Guessoum Dalila

Academic year: 2022/2023

ACKNOWLEDGEMENT

First and foremost, we would like to praise and thank God, the Almighty, who has granted us countless blessings and knowledge so that we have finally been able to accomplish the thesis.

Apart from our effort, the success of this thesis depends largely on the encouragement and guidelines of many others. We take this opportunity to express our gratitude to the people who have been instrumental in the successful completion of this thesis.

We would like to show our heartfelt gratitude to our supervisor Mrs. BACHIRA BOUTOUMI for her guidance and insightful, constructive suggestions throughout the planning and execution of this research project. In addition, her generosity in donating her time has been greatly appreciated.

Special thanks are extended to our parents for their constant love, support, and motivation through our studies.

Our sincere thanks also go to the members of the jury for their interest in our project by agreeing to examine the work and enrich it with their suggestions and recommendations. Finally, we would like to thank everyone who participated either directly or indirectly in the achievement of this work.

ABSTRACT

Wireless sensor networks (WSNs) have undergone significant advancements in recent years, increasing their use in various applications. WSNs can now collect and transmit large amounts of data from remote and harsh environments, enabling real-time monitoring and control of physical systems. They are used in environmental monitoring, precision agriculture, industrial automation, healthcare, and smart cities.

Energy is a critical issue in wireless sensor networks (WSNs) due to the limited battery life of sensor nodes. The energy problem in WSNs arises because sensor nodes are typically deployed in remote and harsh environments, making replacing or recharging their batteries difficult or impossible. As a result, energy management is a crucial aspect of WSN design and operation.

In order to mitigate the energy problem in WSNs, we propose models based on the vacation queueing using Generalized stochastic Petri nets (GSPNs). Then we used the model's corresponding Continuous Time Markov Chain (CTMC) to deduct the algorithm of the infinitesimal generator. Finally, based on these models, the formulas of the main performance indices are inferred for the evaluation of wireless sensor nodes.

Keywords:
WSN, Service Differentiation, Vacation Queueing, N-Policy, GSPNs, IEEE 802.15.4, CTMC.

Résume

Les réseaux de capteurs sans fil (RCSF) ont connu des progrès considérables ces dernières années, ce qui a conduit à leur utilisation accrue dans diverses applications. Les réseaux de capteurs sans fil sont désormais capables de collecter et de transmettre de grandes quantités de données dans des environnements éloignés et difficiles, ce qui permet de surveiller et de contrôler des systèmes physiques en temps réel. Ils sont utilisés dans des applications telles que la surveillance de l'environnement, l'agriculture de précision, l'automatisation industrielle, les soins de santé et les villes intelligentes.

L'énergie est un problème critique dans les réseaux de capteurs sans fil (RCSF) en raison de la durée de vie limitée de la batterie des nœuds de capteurs. Le problème de l'énergie dans les réseaux de capteurs sans fil est dû au fait que les nœuds de capteurs sont généralement déployés dans des environnements éloignés et difficiles, ce qui rend difficile, voire impossible, le remplacement ou la recharge de leurs batteries. Par conséquent, la gestion de l'énergie est un aspect crucial de la conception et du fonctionnement des RCSF.

Afin d'atténuer le problème de l'énergie dans les RCSF, nous proposons des modèles basés sur la file d'attente de vacances à l'aide de réseaux de Petri stochastiques généralisés (RPSG). Nous avons ensuite utilisé la chaîne de Markov à temps continu (CMTC) correspondante du modèle pour déduire l'algorithme du générateur infinitésimal. Enfin, sur la base de ces modèles, les formules des principaux indices de performance sont déduites pour l'évaluation des nœuds de capteurs sans fil.

Mots clés :

RCSF, différenciation de services, File d'attente des vacances, N-Policy, RdPSG, IEEE 802.15.4, CMTC.

ملخص

شهدت شبكات الاستشعار اللاسلكية تطورات كبيرة في السنوات الأخيرة ، مما أدى إلى زيادة استخدامها في مختلف التطبيقات. وهي الآن قادرة على جمع ونقل كميات كبيرة من البيانات من البيئات البعيدة والقاسية ، مما يتيح المراقبة والتحكم في الوقت الحقيقي للأنظمة المادية. يتم استخدامها في تطبيقات مثل المراقبة البيئية والزراعة الدقيقة والأتمتة الصناعية والرعاية الصحية والمدن الذكية.

تعد الطاقة مشكلة حرجة في شبكات المستشعرات اللاسلكية نظرًا لعمر البطارية المحدود المستشعرات. تنشأ مشكلة الطاقة من حقيقة أن عُقد المستشعرات يتم نشرها عادةً في بيئات نائية وقاسية ، مما يجعل من الصعب أو المستحيل استبدال بطارياتها أو إعادة شحنها، نتيجة لذلك تعد إدارة الطاقة جانبًا مهمًا من تصميم وتشغيل أجهزة الاستشعار.

من أجل التخفيف من مشكلة الطاقة في WSNs ، نقترح نماذج قائمة على قوائم الانتظار باستخدام شبكات بيتري العشوائية المعممة (GSPNs). استخدمنا بعد ذلك سلسلة ماركوف ذات الوقت المستمر المقابلة (CMTC) للنموذج لاشتقاق خوارزمية المولد متناهية الصغر. أخيرًا ، بناءً على هذه النماذج ، يتم استخلاص صيغ مؤشرات الأداء الرئيسية لتقييم المستشعر اللاسلكي.

الكلمات الدالة:

WSNs ، تمايز الخدمة ، الإجازات في قائمة الانتظار ، CTMC ، IEEE 802.15.4 ، GSPNs ، N-Policy

Contents

LIST OF ABBREVIATIONS

WSN: Wireless Sensor Network.
WV: Working Vacation.
GSPN: Generalized stochastic petri net.
CTMC: Continuous Time Markov Chain
DTMC: Discrete Time Markov Chain.
PN: Petri Net.
RG: Reachability Graph.
GreatSPN: Graphical Editor and Analyzer for Timed and Stochastic Petri Nets.
OSI: Open Systems Interconnections.
ISO: International Organization of Standardization.
MAC: Medium Access Control.
QoS: Quality of Service.
IEEE: Institute of electrical and electronics engineers.
WPAN: Wireless Personal Area Network.
RF: Radio Frequency.
RFD: Reduced Function Devices.
FFD: Full Function Devices.
CSMA/CA: Carrier Sense Multiple Access Collision Avoidance.

TABLES LIST

Equations List

FIGURES LIST

GENERALE INTRODUCTION

Over the past few years, wireless network technology has been growing steadily thanks to the notable technological development seen in various fields related to microelectronics and wireless communication, opening the horizon for a new generation of smart devices called sensors [1]. These devices are usually deployed in huge numbers to ensure maximum coverage of a region of interest and connectivity between devices. Each device is capable of monitoring and recording the physical conditions of an environment and eventually forwarding the collected data to a central location through a wireless network for processing and analysis. A collection of these devices is known as a wireless sensor network or WSN. Due to their reduced size flexibility and relatively low cost, WSNs have been adopted in various fields (Industrial, Medical, environmental, Military service, etc.) for different applications: patient monitoring, security, threat detection, logistics, seismic and volcanic activity monitoring.

Despite the great potential of wireless sensor networks, the hardware offers minimal resources in transmission range and data processing and storage, mainly because of access difficulties, especially in harsh environments and the scarcity of available energy. When programming the nodes of the network, the management of available resources can be challenging. That is why much research today aims to extend the lifetime of a sensor network by jointly applying different techniques to reduce the amount of energy consumed by the battery during network activities [2].

One main approach, duty cycling, focuses on designing MAC protocols in the MAC sub-layer of resource-constrained WSNs. The duty cycle of a node represents the ratio of the active period and the cycle time, cycle time is the duration of the active period and the sleep period of a sensor. To reduce energy consumption, nodes alternate between activation and dormancy, hence the active and sleep periods. During the active period, the sensor can function at full capacity, transmitting and receiving data. During the sleep period, the sensor's transceiver, considered the main energy consumer, is turned off. During this time, a node will not transmit data to other sensor nodes [3].

The current project proposes a performance evaluation and analysis of an energy-saving and service differentiation framework for a wireless sensor based on duty cycling techniques. We will rely on vacation policies from the theory of queuing with vacation and Generalized stochastic Petri nets (GSPNs) to model and analyze the sleep/wake pattern of the sensor node. From the Petri net of each model, we can observe the marking in the different states and eventually form the corresponding time-continuous Markov chain. After analysis, we can create the algorithm of the infinitesimal generator of the CTMC. To measure the performance parameters, we developed formulas to measure and analyze the node's energy consumption in a particular model.

OUTLINE OF THE REPORT

The remainder of the report is organized as follows:

Chapter 1: Provides a technical background on WSNs and their applications. The important factors that can affect their power consumption, including their architecture, topologies and operation. It also presents one of IEEE's main communication protocols 802.15.4.

Chapter 2: It goes through the formal methods used in modeling the WSNs, starting with a reminder on Petri nets, their types and properties, analysis methods, an overview of queueing theory, and Markov chains in the stochastic process.

Chapter 3: Presents some related work for our problematic, and describes our modeling approach for vacation policy in a sensor node using petri nets, as well as the algorithms used to produce the infinitesimal matrices of the CTMC of each proposed model and the formulas used to calculate the main performance parameters.

Chapter 4: Uses the algorithmic approach in order to conduct extensive comparison between the suggested models and discuss the obtained performance results.

Conclusion: Summarizes the main results drawn from our comparative study and discusses some possible directions for future research work.

CHAPTER 1

BACKGROUND ON WSNS

1 Introduction

Sensor nodes offer a powerful combination of distributed sensing, computing, and communication. The ever-increasing capabilities of these tiny sensor nodes, which include sensing, data processing, and communicating, enable the realization of WSNs based on the collaborative effort of several other sensor nodes. They enable a wide range of applications and, at the same time, offer numerous challenges due to their peculiarities, primarily the stringent energy constraints to which sensing nodes are typically subjected. WSNs incorporate knowledge and technologies from three fields; Wireless Communications, Networking, and Systems and Control theory. In order to realize the existing and potential applications for WSNs, sophisticated and highly efficient communication protocols are required.

2 Components of sensor node

A sensor node is composed of multiple units:

2.1 Processing unit

The sensor node processing unit refers to the central component within a sensor node that manages the overall operation and functionality of the node. It acts as the brain of the sensor node, coordinating the activities of the sensors, data acquisition, and communication modules. The processing unit is responsible for controlling the sensor node's power management, data storage, and data processing tasks. It may include a microcontroller or microprocessor, along with associated memory and input/output interfaces.

2.2 Sensing unit

The main functionality of the sensing unit is to produce a measurable response signal to a change in a physical condition such as temperature, pressure, humidity, or any number of other environmental phenomena. The continual analog signal sensed by the sensors is digitized by an analog-to-digital converter and sent to the embedded processor for further processing. A sensor node can have one or several types of sensors integrated into or connected to the node.

2.3 Transmission unit

This unit is responsible for all data transmission and reception via a wireless communication medium. The various transmission media choices include Radio Frequency (RF), Laser, and Infrared.

2.4 Power unit

The sensor node power unit refers to the component within a sensor node that manages and supplies power to the various components of the node. It is responsible for converting and regulating the power source to meet the requirements of the sensors, processing unit, and communication modules.[19]

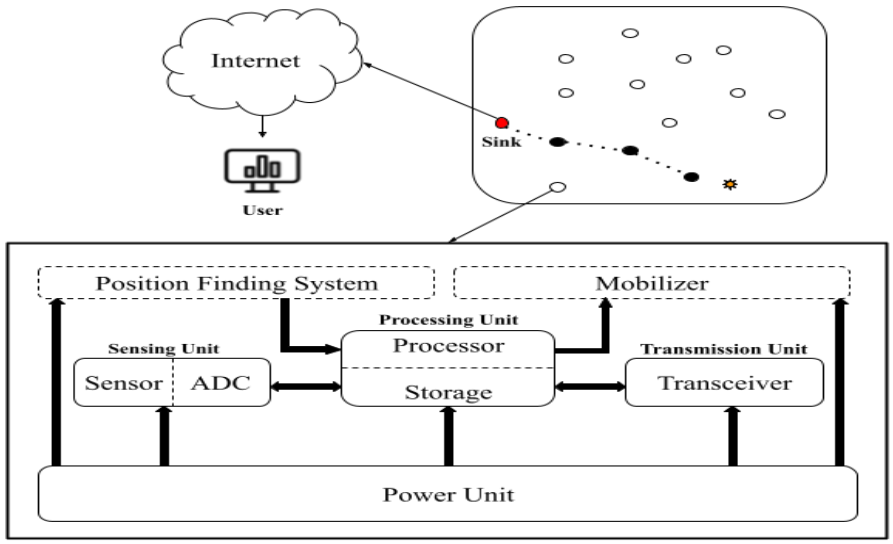

Figure 1: Physical architecture of a sensor node.[3]

3 Energy consumption in a sensor

Energy consumption is the most important factor in determining a sensor network's life because sensor nodes are driven by batteries. Sometimes energy optimization is more complicated in sensor networks because it involves not only reducing energy consumption but also prolonging the network's life as much as possible. The optimization can be done by having energy awareness in every aspect of design and operation. This ensures that energy awareness is also incorporated into groups of communicating sensor nodes, the entire network, and not only the individual nodes. [1]

- **Sensing unit energy:** The capture is carried out by the acquisition components which translate the physical phenomena into an electrical signal. In many cases, it isn't very important compared to the energy consumed by the processing and communication modules.
- **Transmission unit energy:** Process energy consists of two types of energy: switching energy and leakage energy. The supply voltage determines the switching energy, and the total capacity switched at the software level (by running the software). On the other hand, the energy leakage corresponds to the energy consumed when the calculation unit is not performing any treatment. Generally speaking, the processing energy is small compared to the energy of processing necessary for communication.
- **Communication unit energy:** The energy exchanged comprises three components: received, emitted, and standby. The amount of data to be sent, the transmission

distance, and the physical parameters of the radio module all influence the amount of power used. The strength of a signal determines its transmission. The signal range is extensive, and the power consumption energy is essential for high transmission power. Plus, the communication energy required by the sensor nodes accounts for most of the total energy consumed by the sensor nodes.

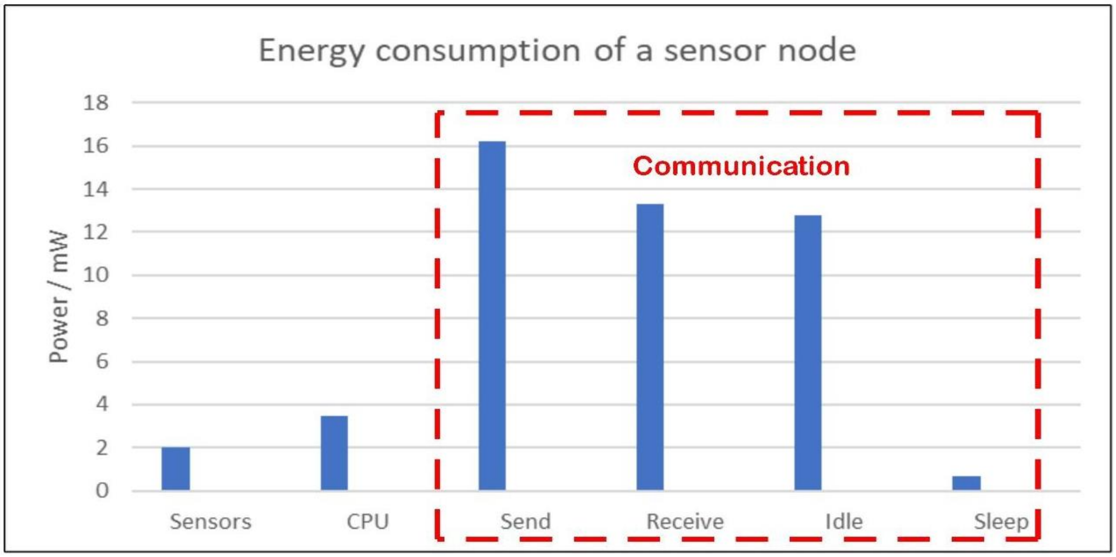

Figure 2: Energy consumption for Sensing, processing and communication [38]

4 Energy saving techniques in WSNs

The network protocol stack has multiple layers; each layer can use a different technique to save energy. Some of the standard techniques are:

- Energy saving can be achieved at the physical layer by using low-power transceivers, adaptive modulation and coding schemes, power control, and directional antennas.
- At the MAC layer, energy saving can be achieved using duty cycling schemes, contention-free protocols, collision avoidance mechanisms, and cooperative communication.
- At the network layer, energy saving can be achieved by using efficient routing protocols, data aggregation and compression, topology control and clustering.
- Energy saving can be achieved at the transport layer by using reliable and congestion-aware protocols, flow control, and error recovery mechanisms.
- At the application layer, energy saving can be achieved by using adaptive sampling rates, event-driven data collection, and cross-layer optimization.

To lower the power usage of the sensor nodes and prolong the duration of the network is applying these methods separately or together. However, Different scenarios and applications have different optimization techniques for saving energy. Hence, energy efficiency is one of many factors to consider when designing such techniques. Other performance metrics, such as latency, throughput, reliability, and security, also affect the trade-offs involved.

5 Wireless sensor network

WSNs refer to networks of spatially dispersed and dedicated sensors that monitor and record the physical conditions of the environment and forward the collected data to a central location. WSNs can measure environmental conditions such as temperature, sound, pollution levels, humidity, and wind [1].

5.1 Wireless sensor network architecture

Wireless Sensor Networks (WSNs) can be defined as self-configured and distributed wireless networks to monitor physical or environmental conditions, such as temperature, sound, vibration, pressure, motion, or pollutants, and to cooperatively pass their data through the network to a sink where the data can be observed and analyzed. A sink or base station acts as an interface between users and the network. One can retrieve required information from the network by injecting queries and gathering results from the sink. Typically, a wireless sensor network contains hundreds of thousands of sensor nodes. The sensor nodes can communicate among themselves using radio signals [2].

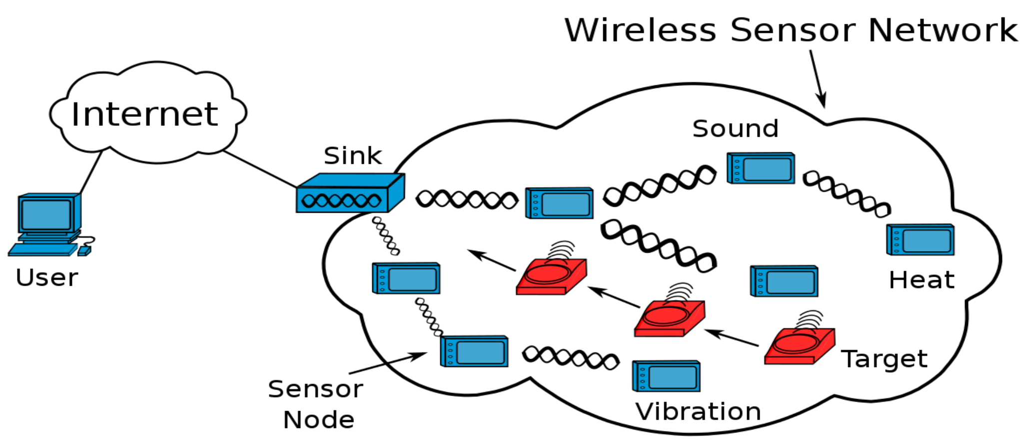

Figure 3: Architecture of a wireless sensor network (WSN) [3].

5.2 Wireless sensor network topologies

The topology depicts how nodes are connected with each other via wireless communication. It indicates the placement of the nodes in the sensor field. These deployments aren't random but follow a design that serves a purpose.[4]

- **Tree topology:** The network uses a central hub called a root node as the main communication router.

- **Star topology:** Star networks are connected to a centralized communication hub (sink), and the nodes cannot communicate directly. The entire communication must be routed through the centralized hub.
- **Mesh topology:** Mesh topologies involve messages that can take several paths from source to destination. A mesh network in which every node connects to every other is called a full mesh.
- **Circular topology:** In this topology, there is a circular sensing area, and the sensing area has a sink. The sensor nodes sense the event of interest and transmit this data to the sink.

5.3 Wireless sensor network application fields

In recent years, Wireless sensor networks have gained popularity and been used in various domains due to the flexibility and ease of deployment and maintenance and the fact that they are entirely wireless.[5] The main application fields of WSNs are:

- **Military applications:** The rapid deployment, self-configuration, and fault tolerance of sensor networks make this type of network a valuable tool in this field. Deployment on a strategic or difficult-to-access location to monitor all enemy forces' activities or to analyze the terrain before sending troops there (for example, the detection of chemical, biological, or radiation agents).
- **Health applications**: The health monitoring field represents a rapidly growing market for wireless sensor networks. These sensor networks can ensure permanent monitoring of vital human organs (blood sugar monitoring, cancer detection) thanks to micro-sensors that can be swallowed or implanted under the skin. They can also facilitate the diagnosis of some diseases by performing physiological measurements such as blood pressure and heart rate, using sensors, each with a specific task.
- **Environmental applications:** The term "Environmental Sensor Networks (ESNs)" has been developed to cover several benefits of WSNs to environmental and earth science studies. This comprises sensing oceans, seas, glaciers, atmosphere, volcanoes, forests, etc. However, there are presently some biosensors that have been developed for use in agricultural and environmental sustainability. Some other vital aspects are; air contamination monitoring and management, forest fires discovery/detection, and Landslide discovery/detection.
- **Industrial applications:** In an industrial environment, sensors are usually used in hazardous areas that are hazardous or inaccessible to workers to monitor the operation of equipment, assets, or environmental conditions. WSNs have been advanced for "Technological Condition-based Maintenance (TCMB)" since they could offer significant cost reductions/investments and allow innovative functionalities.
- **Agricultural applications:** The employment of WSNs has been reported to assist farmers in various aspects such as greenhouse (GH) monitoring and management, the

maintenance of wiring in a complex environment, and irrigation mechanization, which aids more resourceful water use and reduction of wastes.

5.4 Challenges and constraints

A variety of unique challenges and constraints influence a sensor network. These constraints impact the design and serve as a guideline for designing and implementing a WSN.

5.4.1 Design constraints

- **Energy:** The constraint often associated with sensor network design is that sensor nodes operate with limited energy budgets. Typically, they are powered through batteries, which must be replaced or recharged (e.g., using solar power) when depleted. For some nodes, neither option is appropriate. They will be discarded once their energy source is depleted. Whether or not the battery can be recharged significantly affects the energy consumption strategy. [6]
- **Fault tolerance:** Sensor nodes generally fail due to lack of power, physical damage, or environmental interference, and such failures should not affect the WSN's designated task. Fault tolerance is the ability of a WSN to sustain sensor functionalities without interruptions due to node failures.

5.4.2 Hardware constraints

- **Size:** The small size of the sensors can have many advantages, allowing for flexible and simple deployment of the network. However, the reduced size of the sensor may affect the battery's size and the sensor's processing power.
- **Processing power:** Most sensors utilize very low-power, low-frequency microcontrollers to operate autonomously and efficiently to accomplish tasks and minimize power consumption.

6 Stack protocol architecture

The Open Systems Interconnection (OSI) seven-layer model, proposed by the International Organization for Standardization (ISO), forms the basis for the design of the WSN protocol stack. However, unlike the seven-layer OSI model, which consists of the physical layer, the data link layer, the network layer, the transport layer, the session layer, the presentation layer, and the application layer, the WSN protocol stack does not adopt all the seven layers of the OSI model. In reality, the seven-layer OSI model has too many layers making it overly complex and challenging to implement. Consequently, the protocol stack adopted by WSN consists of only five layers [6].

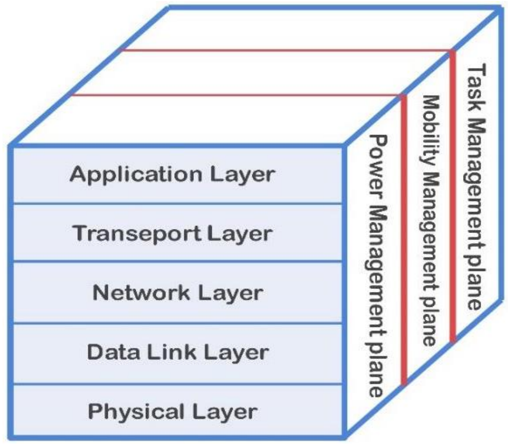

Figure 4: Stack protocol.[6]

- **Physical layer:** The physical layer converts bit streams from the data link layer to signals suitable for transmission over the communication medium. For this purpose, it must deal with various related issues, such as transmission medium, frequency selection, and data encryption. [9]
- **Data link layer:** The data link layer is responsible for data stream multiplexing, data frame creation and detection, medium access, and error control to provide reliable point-to-point and point-to-multipoint transmissions.
- **Network layer:** It handles the routing of data the transport layer provides. It establishes the routes between the sensor and sink nodes and selects the best path regarding energy, transmission delay, throughput.
- **Transport layer:** The transport layer is responsible for reliable data delivery required by the application layer between sensor nodes and the sink(s). Due to sensor nodes' energy, computation, and storage constraints, traditional transport protocols cannot be applied directly to sensor networks without modification.
- **Application layer:** The application layer contains a variety of application layer protocols to generate various sensor network applications. This layer performs sensor network applications, such as Query dissemination, node localization, time synchronization, and network security.
- **Management planes:** The management planes of the protocol stack are the components that optimize the performance of a wireless sensor network (WSN) according to different metrics.
- **Power management:** A wireless sensor node requires only a limited power source. The node's life shows a strong dependence on the battery's life. The energy management plan must manage how the nodes use their energy.

- **Mobility management:** A mobility management system must exist since nodes can be mobile. Such a system must be able to record the node's movements in order to help it locate itself.
- **Task management:** The task management level ensures the balancing and distribution of tasks among the different nodes of the network in order to ensure cooperative and energy-efficient work and, consequently, to extend the network's life.

7 WSNs standards and technology

Several standards have been developed to ensure interoperability and compatibility between different WSNs. These standards define the protocols and specifications for communication, data transfer, and power management in WSNs. The availability of these standards has significantly simplified the development and deployment of WSNs, enabling researchers and practitioners to focus on the application-specific aspects of their projects.

7.1 IEEE 802.15.4

IEEE 802.15.4 is a standard defined by IEEE 2003 for wireless communication in low-power, low-data-rate, and low-cost applications. It defines the physical and MAC (media access control) layer specifications for wireless personal area networks (WPANs), including the protocol for low-rate wireless networks such as ZigBee, Wireless HART, and ISA100.11a. 802.15.4 operates in the ISM radio bands and has a data rate of 20 up to 250 kbps. It supports star and peer-to-peer network topologies and operates at a short distance of up to 10 meters. The IEEE 802.15.4 standard is used in various applications such as home automation, building automation, industrial automation, and healthcare. It is also used in wireless sensor networks (WSNs) for monitoring and controlling physical or environmental conditions. It has very low complexity and ultra-low power consumption.[35]

7.1.1 802.15.4 Device architecture

The architecture of an 802.15.4 device is typically composed of several layers, including the physical layer, which usually contains the radio frequency (RF) transceiver along with its low-level control mechanism, A medium access layer (MAC) responsible for coordinating access to the shared wireless medium and managing data transmission between devices. The upper layers include a network layer providing networking services like addressing and routing, plus an application layer containing the description of the device’s intended function. [21]

7.1.2 Device types

There must be one and only one PAN coordinator in an 802.15.4 network. The PAN Coordinator assigns a PAN ID to the network and assigns itself a short address. It handles requests from other devices that wish to join the network and assigns them a short address. It is also responsible for performing an energy scan to select the most appropriate channel to use for communications. Depending on the topology, it relays all or some of the messages sent. In practice, the coordinator is the network's edge and bridges the network toward the Internet. Given that the coordinator makes use of another network interface which may or may not be incapable of supporting sleep mode communication of 802.15.4 networks, which in some cases does not support the PAN coordinator to enter sleep mode, this device type requires more power usage than other nodes in the network, making it less suitable to run on battery.

- **The local coordinator:** also known as the router, is a device capable of relaying messages from and to other nodes. In a network, there can exist many local coordinators, each capable of handling requests when other nodes wish to join the network or relay their messages.
- **The end devices:** are nodes connected to either a PAN Coordinator or Local Coordinator, which cannot relay other nodes' messages. Usually, these are low-power-consuming devices with a high sleep-to-work ratio and are only set to wake up to handle incoming messages or transmit data. These end devices can be battery-powered with a very long battery-operating time.

7.1.3 Device classes

The IEEE 802.15.4 distinguishes two device classes on the MAC layer:

- **Full Function Device (FFD):** This class of devices can support the full functionality of the 802.15.4 MAC layer and can be found in every topology. It can operate in three different roles, as a PAN coordinator, a simple coordinator, or an end device. It sends out beacon frames and communicates with any device class. In addition, it can handle various services like synchronization, communication, and network join services.
- **Reduced Function device (RFD):** This class of devices supports the minimum functionality of the 802.15.4 standard and is able to operate in one mode only, and that is an end device. It needs at least one FFD in the same network to communicate and relay data and can only be found in a star topology.

7.1.4 Topologies

The IEEE 802.15.4 standard supports several topologies: star topology, point-to-point (peer-to-peer), and tree (cluster-tree) topology. It is possible to have several PANs operating in the same area but independently of each other, each using a different PAN identifier.

- **Tree topology:** The star topology requires communications to be established directly and only between the node that manages the previously defined network (PAN coordinator) and the nodes in its wireless range. The coordinator PAN will be the initiator node of the network, and all traffic will have to pass through this node.

- **The point-to-point topology:** In the point-to-point topology, each node can communicate with any other node of the network thanks to the collaboration of the intermediate nodes (Coordinator) that are solicited in order to relay packets to the destination. This topology makes it possible to create much larger extended networks, but this will require the addition of a routing protocol.
- **Cluster-Tree Topology:** Cluster-tree topology is just a particular case of peer-to-peer topology. The only difference is that the network will be hierarchical, with a PAN coordinator, coordinators, or the cluster heads managing the nodes connected to them and node terminals communicating only with their coordinator.

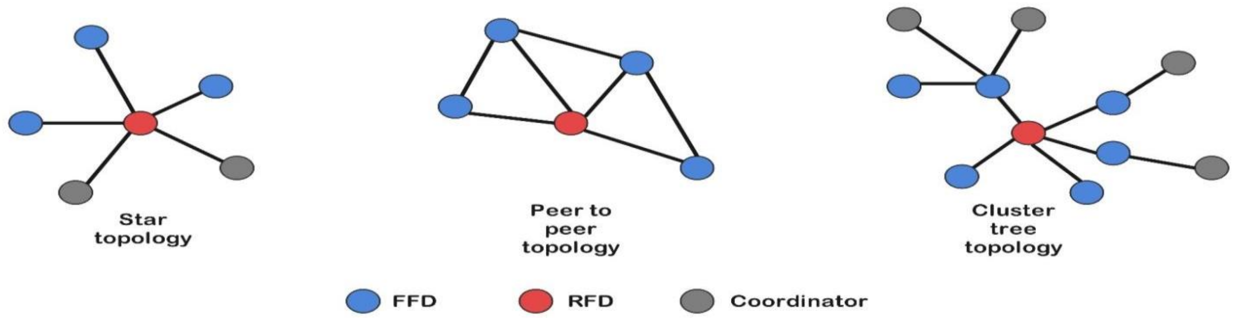

Figure 11: IEEE 802.15.4 topologies.[35]

7.1.5 IEEE 802.14.5 frame types

Four main types of frames are defined in the IEEE 802.14.5 standard:

- The data frame contains the data being exchanged between nodes.
- A *control frame* is a particular frame used to communicate special commands like network association.
- Acknowledgment frames notify a sender that a frame has been received.
- Beacon frames are sent only by the coordinator to manage the network.

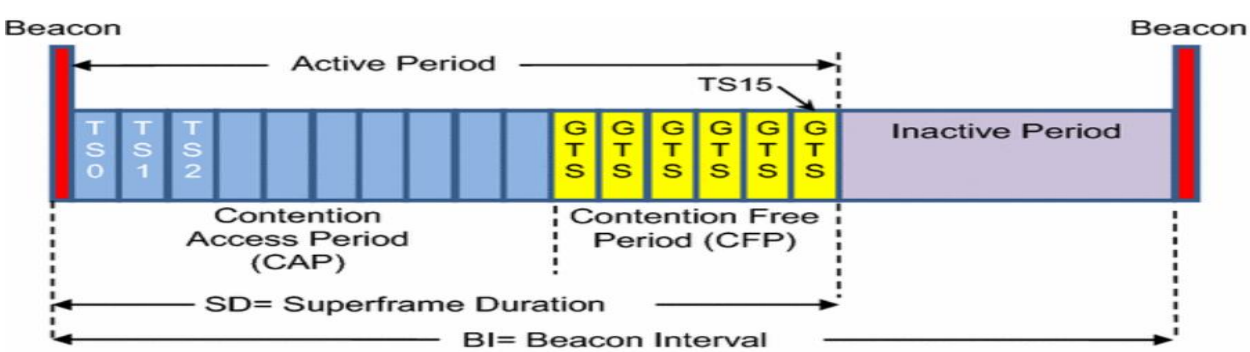

Figure 12: IEEE 802.15.4 Superframe structure.[22]

7.1.5.1 Operating modes in IEEE 802.15.4

IEEE 802.15.4 is a standard for low-rate wireless personal area networks (LR-WPANs) that supports various applications such as sensor networks, smart home, and industrial automation. The standard defines two operating modes: beacon-enabled mode and non-beacon-enabled mode. In beacon-enabled mode, a coordinator periodically broadcasts beacons to synchronize the devices in its network and to allocate time slots for data transmission. This mode is suitable for applications that require low latency and deterministic behavior. In non-beacon-enabled mode, there is no beacon transmission and the devices use a carrier sense multiple access with collision avoidance (CSMA/CA) mechanism to access the channel. This mode is suitable for applications that have low duty cycle and sporadic traffic.

7.1.5.2 Data transfer procedure

- **From device to coordinator:** When a device wants to send a data packet to the coordinator, it is verified if it has allocated GTS. If so, it wakes before the time slot starts and sends the packet immediately without running a carrier sense or collision-avoiding operation. If not, it sends its data packet during the CAP using the slotted CSMA protocol.
- **From coordinator to the device:** When the device has allocated a receive GTS and the packet acknowledgment/IFS cycle fits into these, the coordinator will transmit the packet in the allocated time slot without further coordination. If the coordinator cannot receive GTS, a handshake is performed. The coordinator announces a Bufford packet to the device by including its address in the pending address field of the beacon frame.

8 Conclusion

In this chapter, we have introduced wireless sensor network networks, their architecture, characteristics, and communication of sensor networks, as well as the fields of application of this type of network. Plus, a brief introduction to the IEEE 802.15.4 standard. Despite the extensive use of WSNs in various fields and the evolution they witnessed in recent years, the technology still needs to improve from an energy standpoint. To mitigate this problem, special techniques are used to make sensors more efficient and therefore extend the network's lifetime.

CHAPTER 2

FORMAL METHODS

1 Introduction

Analysis of wireless sensor network systems is typically performed by simulation. Due to the limitations of wireless sensor networks, such as Memory, power consumption, execution rate, and communication, a new approach is preferred. It consists of using formal forms for identification. These limitations, therefore, give more accurate results. Among these models, our study Three methods are considered. The Generalized Stochastic Petri nets (GSPNs) are used first, then Markov Chains are introduced for performance evaluation of the sensor node.

2 Generalized stochastic petri nets (GSPNs)

Petri Nets are commonly used for modeling and evaluating the performance of systems involving concurrency, non-determinism, and synchronization, such as parallel and distributed computer architectures and communication networks.

A Petri net comprises events called transitions and resource containers called places and arcs that link them. Places contain marks that usually represent conditions, resources, or products. A GSPN is a special kind of Petri net. Its transitions can be one of two types: immediate and timed. The transitions of the first type do not need time to fire, whereas timed transitions need time to fire. An inhibitor arc is a means to forbid the firing of an event. When a place is linked with a transition by an inhibitor arc, this transition stays inactive until the number of marks in the place becomes lower than the weight of the inhibitor arc.

Moreover, after modeling the system using GSPNs formalism, two classes of analysis can be conducted. The qualitative analysis includes liveness, boundedness, robust connectivity, ergodicity, and others. Quantitative analysis will be performed in the last steps and consists of calculating the performance parameters of the equilibrium state. Solving the net in the equilibrium state provides the vector of stationary probability called π. [19]

2.1 Formal definition

A GSPN is a tuple (P, T, I, O, H, W, Π, M0) where:

- P is the set of places.
- T is the set of timed and immediate transitions.
- I: T → bag(P) is the input function (represented by arcs from places to transitions).
- O: T → bug(P) is the output function (represented by arcs from transitions to places).
- H: T→ bug(P) is the inhibition function (circle-headed arcs from places to transitions).
- W: T→ R defines the negative exponentially distributed firing rate in the case of timed transitions and the weight in the case of immediate transitions.
- Π: T→ N is the priority function. Timed transitions have a priority level of 0, and immediate transitions have a priority level > 0.
- M0:P→N is the initial marking. [15]

2.2 Graphical representation

In addition to the algebraic representation, GSPNs are represented graphically. Where places are represented as circles containing several points representing tokens (a place has an infinite capacity by default), timed transitions are represented by small hollow rectangles, and a solid rectangle represents immediate transitions, neither can contain tokens. Connecting the places and transitions are directed arcs that have a capacity of 1 by default. If other than 1, the capacity is marked on the arc, with the rule that arcs can only connect places to transitions and vice versa. To control the firing of transitions, inhibitor arcs are placed between a place and a transition and represented by a directed arc finishing with a small circle.[20]

2.3 Incidence matrix

The Petri net, initially a graphical object, not very exploitable under this aspect despite the user-friendliness it brings, can be transcribed algebraically. The incidence matrix, which will be described, synthesizes all the links between places and transitions of the PN. This matrix is generally rectangular and has a number of columns equal to the number of transitions of the net and a number of lines equal to the number of places of the net. Each element of the matrix reflects the presence or absence of a link between each place and each transition, as well as the weight attached to each transition. Each transition and the weight attached to the arc in question. The sign of the element in question transcribes the direction of this arc. [34]

- Post-Conditions matrix or W+ is a matrix defining the relation between transitions and places.
- Pre-Conditions matrix or W- is a matrix defining the relation between places and transitions.

The incidence matrix or W is defined as $W = W^{+} - W^{-}$

$$W = \begin{pmatrix} 1 & -1 & 0 \\ -1 & 0 & 1 \\ 0 & 1 & -1 \\ 0 & 0 & 0 \end{pmatrix} \qquad W^{+} = \begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{pmatrix} \qquad W^{-} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix}$$

2.4 The evolution of a GSPN

Stochastic generalized Petri nets (GSPNs) evolve in time by firing transitions based on a probability distribution. The probability of each transition firing is determined by the transition rate and the number of tokens in the input places; this process is repeated until a stopping condition is met, such as reaching a specific marking or a certain number of iterations. There are two types of transitions in a Petri net:

- **Timed transitions:** A timed transition has a firing rate associated with a random variable to determine the execution time of the different activities of the transitions in the net. When the random variable of the transition follows the am negative exponential distribution.
- **Immediate transitions:** An immediate transition fires in zero time and are usually used to model prioritized actions that are not time-consuming, like synchronization and logical operations. Timed transitions have a higher priority than timed transitions. In addition, multiple priority levels can be defined between immediate transitions by assigning a specific weight to each transition.

2.4.1 Enabling rule

A transition t ∈T is enabled when the number of tokens in each input place exceeds the multiplicity of input arcs. The number of the tokens in each inhibitor input place is less than the multiplicity of the inhibitor arcs, i.e., T is enabled ⇔ M(Pi) ≥ T(Pi) for any P, where M(Pi) represents the number of tokens in place P. Once a transition is enabled, it can fire at any time.

2.4.2 Firing a transitions

The firing of an enabled transition removes as many tokens as the multiplicity of the input arcs from the input places. It adds as many tokens as the multiplicity of the output arcs to the output places. In a given marking Mi, firing an enabled transition generates another marking Mj, which is said to be directly reachable from Mi (Mi → Mj).[16]
Considering the following example (Figure 15) to illustrate the firing of a transition in a Petri net:

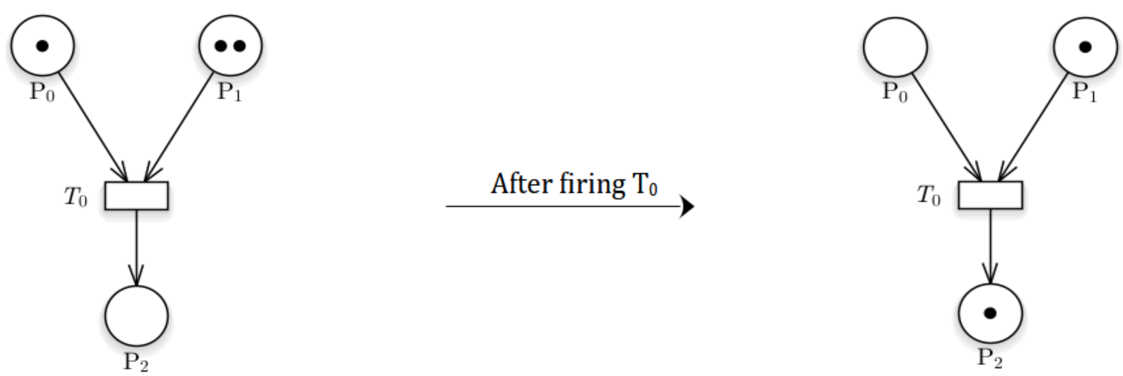

Figure 15: Firing a transition in a petri net.[16]

2.4.3 Transition firing sequence

The natural extension of the concept of transition firing is the firing of a transition sequence (or execution sequence). A transition sequence σ = t (1), · · ·, t (k) can fire starting from marking M if and only if there exists a sequence of markings M = M (1), · · ·, M(k+1) =

M’ such that ∀i = (1, · · ·, k), M(i) [t(i) >M(i+1). We denote by M[σ>M’ the firing of a transition sequence and say that M’ is reachable from M.

2.4.4 Inhibitor arc

An inhibitor arc is used to indicate when a local state disables a transition rather than enables it. An inhibitor arc from a place to a transition means the transition cannot fire if there is a token in the place; It can fire when there is no token in the place if the places connected to its input arcs do contain tokens [17].
In addition, a transition t is not fireable until the marking of each input place is less than the valuation of the inhibitor arc linking that place to the transition.

Figure 16: Graphical representation of an inhibitor arc.[17]

2.5 The reachability graph

The reachability graph (RG) is a fundamental tool for analyzing a Petri Net model. Given a Petri net, from its initial marking M0, we can obtain as many “new” markings as the number of the enabled transitions. From each new marking, we can again reach more markings. This process results in a reachability tree representation of the markings. Nodes represent markings generated from M0 (the root) and its successors, and each arc represents a transition firing, which transforms one marking into another. [14]
Considering the GSPN in Figure 17, the corresponding reachability graph would be:

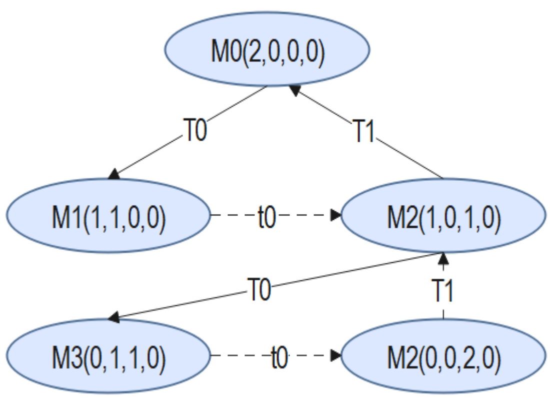

Figure 9: Reachability graph for GSPN.

2.6 Properties of Petri nets

The behavior properties of a Petri net depend on its initial marking and the firing policy of the net and are sometimes called Marking-dependent properties. These properties are usually used for designing dynamic systems because they depend on the system's behavior, not structure.[17]

- **Reachability:** A marking Mn is said to be reachable from a marking M0 if there exists a sequence of firings that transforms M0 to Mn.
- **Boundedness:** A Petri net (N, M0) is said to be k-bounded or simply bounded if the number of tokens in each place does not exceed a finite number k for any marking reachable from M0.
- **Liveliness:** A Petri net (N, M0) is said to be alive if, no matter what marking has been reached from M0, it is possible to ultimately fire any transition of the net by progressing through some further firing sequence.
- **Reversibility and Home State:** A Petri net (N, M0) is said to be reversible if, for each marking M in R(M0), M0 is reachable from M. Thus, in a reversible net one can always get back to the initial marking or state.
- **Coverability:** A Petri net is said to be coverable if there exists a sequence of transitions that can be fired to reach the given marking from the initial marking.
- **Persistence:** A Petri net (N, M0) is said to be persistent if, for any two enabled transitions, when the firing of one transition will not disable the other. A transition in a persistent net, once it is enabled, will stay enabled until it fires.
- **Fairness:** Two transitions, t1, and t2, are said to be in a bounded-fair (or B-fair) relation if the maximum number of times that either one can fire while the other is not firing is bounded.

2.7 Structural properties

The structural properties depend on the structure of the Petri net, not the firing policy or the initial marking. For that reason, they are mainly used for designing static systems because they depend on the layout and not how the system behaves. Most structural properties can be easily verified using algebraic techniques. The principal structural properties of Petri nets are:

- **Consistency:** A Petri net is considered consistent if an initial marking M0 and a firing sequence σ from M0 back to M0 such that every transition fires at least once in σ.
- **Repetitive:** A Petri net is said to be repetitive if there exists an initial marking M0 and a fireable sequence σ, with each transition appearing an unlimited number of times.
- **Controllability:** A Petri net is said to be utterly controllable if any marking is reachable from any other marking.
- **Conservativeness:** A Petri net is conservative if all transitions fire tokens preserving, i.e., All transitions add precisely as many tokens to their output places as they subtract from their input places.

2.8 The stochastic process associated with GSPN

Due to the presence of immediate transitions, the set of accessible markings of a GSPN contains two types of markings.

Tangible markings represent states where the modeled system spends a certain amount of time. However, vanishing markings represent states in which the time spent is zero. The stochastic process associated with a GSPN is a semi-Markovian stochastic process, where the sojourn time distribution in the markings is a composition of negative exponential distributions and zero deterministic distributions.

The average sojourn time in a vanishing marking is zero. In contrast, the sojourn time in a tangible marking M is a random variable corresponding to the minimum Firing time of the fireable transitions by this marking. [18]

2.9 Performance parameters

The quantitative analysis of a GSPN consists in calculating stationary probabilities and performance indices. It is based on the continuous-time Markov chain (CTMC) associated with the GSPN. This chain can be constructed from the graph of accessible markings. The states of the Markov chain are the tangible markings. Evanescent markings are merged with their successors (tangible markings).[18] The transition rates of the CTMC are the firing rates of the GSPN transitions. We obtain the infinite generator of this CTMC, which is then a square matrix of dimension (r × r) (r is the finite number of tangible markings of the GSPN) that groups all the transition rates from one marking to another. from one marking to another. The probability distribution π = (π1, π2, ..., πn) can then be obtained by solving the system of equations of the following linear equation system:

$$\begin{cases} \pi Q & = 0; \\ \sum_{i=1}^{r} \pi & = 1. \end{cases}$$

Equation 1

3 Markov chain

A Markov chain is a mathematical model used to describe a system that transitions between different states over time. A *Markov chain* is a stochastic process that follows the Markov property, which states that the probability of transitioning to a new state depends only on the current state and not on any previous states. It is used in a wide range of applications due to its ability to model systems that exhibit random behavior or are subject to external influences that cannot be predicted with certainty. A Markov chain is a stochastic Markov process {X (t), t ∈ T} with discrete state space E (E ⊂ N). Depending on the values of the index t, a Markov chain can be discrete-time (CMTD: Discrete-Time Markov Chain) or continuous-time (CMTC: Continuous-Time Markov Chain).[10]

3.1 The Exponential distribution [11]

A continuous random variable X is said to have an exponential distribution with parameter λ>0 if its pdf is given by:

$$F(t) = \begin{cases} \lambda e^{-\lambda t} & \text{si } \; t \geq 0 \\ 0 & \text{otherwise} \end{cases}$$

Equation 2

Its distribution function is:

$$F(t) = \int_0^t \lambda e^{-\lambda x} dx = 1 - e^{-\lambda t}, t \geq 0$$

Equation 3

The average for a variable following the Exponential law is:

$$E(x) = \frac{1}{\lambda}$$

Equation 4

3.2 Random variable

A random variable is an application X of a probability space Ω in a state space E, defined as a set of values that the random variable X could take.[12]
There are two types of random variables:

- When E ⊂ Z, X is said to be a **discrete** random variable that is defined by its state probabilities: p(n) = P[X=n], n = -∞, ..., +∞, with:

$$\sum_{k=-\infty}^{+\infty} P(n) = 1$$

Equation 5

- When E ⊂ R, X is said to be a **continuous** random variable that is defined by its function probability density fx(x) for x ∈] - ∞, +∞ [, such that:

$$\int_{-\infty,}^{+\infty} Fx\,(X)dx = 1$$

Equation 6

3.3 The stochastic process

A stochastic process is defined as a family of random variables {Xt: t ⊆ T} where each random variable Xt is indexed by parameter t ⊆ T, usually called the time parameter

if T ⊆ℝ+ = [0, ∞).

The set of all possible values of Xt (for each t ⊆ T) is known as the state space S of the stochastic process. Suppose a countable, discrete-parameter set T is encountered. In that case, the stochastic process is called a discrete-parameter process, and T is commonly represented by (a subset of) No = {0,1, . . .}; otherwise, we call it a continuous-parameter process. The state space of the stochastic process may also be continuous or discrete. Generally, we restrict ourselves here to the investigation of discrete state spaces and, in that case refer to the stochastic processes as chains, but both continuous- and discrete-parameter processes are considered.[13]

A stochastic process $\{X_t: t \subseteq T\}$ constitutes a Markov process if for all $0 = t_0 < t_1 < \ldots < t_n < t_{n+1}$ and all $s_i \subseteq S$ the conditional CDF of X_{tn+1} depends only on the last previous value X_{tn} and not on the earlier values $X_{t0}, X_{t1}, \ldots, X_{tn-1}$:

$$P\,(X_{tn+1} \leq S_{n+1} \mid X_{tn} = S_n, X_{tn-1} = S_{n-1} \ldots, X_{t0} = S_0) = P\,(X_{tn+1} \leq S_{n+1} \mid X_{tn} = S_n)$$

3.4 Discrete-Time Markov Chain (DTMC)

Discrete-parameter Markov chains are considered. First, Markov processes are restricted to a discrete, finite, or countably infinite state space, S, and a discrete-parameter space T. For the sake of convenience, we set T ⊆ No. The conditional Probability Mass Function (PMF) reflecting the Markov property for discrete-time Markov chains, is summarized in the following definition:

A given stochastic process $\{X_0, X_1, \ldots, X_{n+1}, \ldots\}$ at the consecutive points of observation 0,1, . . ., n+1 constitut.es a DTMC if the following relation on the conditional pm, that is, the Markov property, holds for all n E No and all s⊆ S:

$$P\,(X_{tn+1} \leq S_{n+1} \mid X_{tn} = S_n, X_{tn-1} = S_{n-1} \ldots, X_{t0} = S_0) = P\,(X_{tn+1} \leq S_{n+1} \mid X_{tn} = S_n).$$

Given an initial state, the DTMC evolves, that is, step by step, according to one-step transition probabilities. The right-hand side reveals the conditional PMF of transitions from state sn (at time step n) to state sn+1 (at time step n + 1). Without loss of generality, let S = (0, 1,2, . . .} and write the following shorthand notation for the conditional PMF of the process's one-step transition from static i to state j at time n conveniently:

$$P_{ij}(n) = P\,(X_{n+1} = s_{n+1} = j \mid Xn = S_n, = i).$$

In the homogeneous case, when the conditional PMF is independent of epoch n, it is reduced to:

$$P_{ij} = P_{ij}(n) = P\,(X_{n+1} = j \mid X_n = j \mid X_n = i) = P\,(X_1 = j \mid X_0 = I)\ \forall n \subseteq T$$

Graphically, a finite-state DTMC is represented by a state transition diagram (also referred to as state diagram), a finite directed graph, where state *a vertex depicts me of the chain*, and a one-step transition from state *i* to state *j* by an edge marked with one-step transition probability p_{ij}.[13]

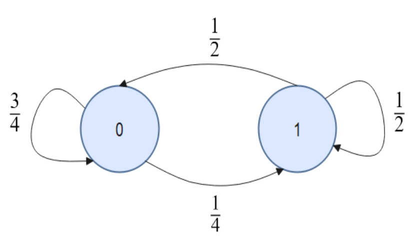

Figure 10: Transitions diagram of CTMC.

2.5.1 Stationary regime of a DTMC [14]

The initial distribution of a discrete-time Markov chain designates the state where the system is at the start of the analysis. A probability vector represents it:

$$\pi^{(0)} = [\pi_0^{(0)}, \pi_1^{(0)} \dots \pi_i^{(0)}]$$

where $\pi(0)$ is the probability that the system is in state *i* at the initial moment,

$$\pi_i^{(0)} = P\,[X_0 = i]\ for\ i = 1\dots, s.$$

When the system is initially in state i, we have: $\pi_i(0) = 1$, and $\pi_j(0) = 0$, $\forall j \neq i$.

The study of the steady state of a CMTD consists in determining the stochastic vectors probabilities of states at different times $\pi^{(n)} = \{\pi_i^{(n)}, i \in E\}$, where: $\pi_i^{(n)} = P\,[X_n = i]$ is the probability that the system is in state *i* at time n. Indeed, we have the following equality:

$$\pi_{(n)} = \pi^{(n-1)}.\, P = \dots = \pi^{(n)}.\, P^n$$

When n becomes very large (n → ∞), after the lapse of an infinite time, under certain conditions, the vector of the probabilities of the states converges towards a vector π:

$$\pi = \mathrm{Lim}_{n\to\infty}\pi(n)$$

In this case, we say that the steady state is reached, and the probabilistic distribution corresponding then remains stable throughout the process after this time. This gives the possibility to calculate several stationary performance parameters of the system.

The vector $\pi = \{\pi i, i = 1, \cdots, s\}$ is said vector of stationary probabilities, and it is the unique solution of the following system of linear equations:

$$\begin{cases} \pi . P = \pi \\ \sum_{i \epsilon E} \pi_i = 1 \end{cases}$$

Equation 7

2.6 Continuous- Time Markov Chain (CTMC)

Continuous- and discrete-time Markov chains provide different yet related modeling paradigms, each having its application domain. For the definition of CTMCs, we refer back to the definition of general Markov processes and specialize it to the continuous parameter, discrete state-space case. CTMCs are distinct from DTMCs in that state transitions may occur at arbitrary instants of time and not merely at fixed, discrete time points, as with DTMCs. Therefore: we use a subset of the set of non-negative real numbers IW: to refer to the parameter set T of a CTMC, as opposed to No for DTMCs:

A given stochastic process $\{Xt: t \in T\}$ constitutes a CTMC if, for arbitrary

$$t_i \in EX: with\ 0 = t_0 < t1 < \dots < t_n < t_{n+1},\ \forall n \in N, and\ \forall s_i \in S = \mathbf{N}_0$$

for the conditional PMF, the following relation holds:

$$P\,(X_{tn+1} = s_{n+1}\, l\, X_{tn}, = s_n, X_{tn-1} = s_{n-1} \dots X_{t0} = s_0) = P(\, X_{tn+1} = s_{n+1}\, l\, X_{tn}, = s_n)$$

Like DTMCs, the formula expresses the Markov property of continuous-time Markov chains. If we further impose homogeneity, then because the exponential distribution is the only continuous-time distribution that provides the memoryless property, the state sojourn times of a CTMC are necessarily exponentially distributed.[13]

Again, the right-hand side is referred to as the transition probability p_{ij} (u, w) of the CTMC to travel from state *i* to state *j* during the period *[u,v)*, with *u, w* $\in$ *T* and $u \le v$:

$$p_{ij}(u,v) = P(X_v = j \mid X_u = i).$$

For $u = v$ we define:

$$p_{ij}(u,v) = f(x) = \begin{cases} 1, & i = j \\ 0, & Otherwise \end{cases}$$

If the transition probabilities *Pij(u, v)* depend only on the time difference $t = v - u$ and not on the actual values of *u* and *v*, the simplified transition probabilities for time-homogeneous CTMC result:

$$P_{ij}(t) = P_{ij}(0, t) = P(X_{u+t} = j \mid X_u = i) = P(X_t = j \mid X_0 = i) \text{ for } \forall u \in T$$

Graphically, CTMCs can be described either by a state diagram or by a transition rate matrix called an infinitesimal generator.

2.6.1 Infinitesimal generator [14]

In continuous-time Markov chains, in addition to transition probabilities Pij (The probability of visiting *j* while leaving *i*), we consider the rates of transition μij. When the process enters state *i*, there remains a random duration of exponential distribution with parameter μi, then jumps instantaneously to the state $j \neq i$, with the probability pij, the transition time from *i* to *j* is exponential with parameter μij = μi × pij. Thus, μij is the average number of transitions from state *i* to state *j* per unit of time. The infinitesimal generator Q is a square matrix of order s = |E|, whose elements qij, (i ≠ j) correspond to the transition rates μij; qij, = μij, the elements of the diagonal qij are, by definition, equal to the opposite of the sum of the other elements in the row:

$$q_{ij} = \begin{cases} \mu_{ij} & \text{if} \quad i \neq j \\ -\sum_{k=1, k\neq i}^{s} \mu_{ik} & \text{if} \quad i = j \end{cases}$$

Equation 8

2.6.2 Transitions Diagram

Continuous-time Markov chains are represented by their transition diagram, a directed graph whose vertices are the states, and the arcs are the transitions with a nonzero transition rate in Q. There will be an arc from state i to state *j* if q (i, j) ≠ 0. Moreover, we never represent a transition between i and i.[14]

As an example, a CTMC with three states E={1,2,3} and Q its infinitesimal generator:

$$Q = \begin{pmatrix} -0.025 & 0.02 & 0.005 \\ 0.3 & -0.5 & 0.2 \\ 0.02 & 0.4 & -0.42 \end{pmatrix}$$

The transition diagram corresponding to the CTMC would be:

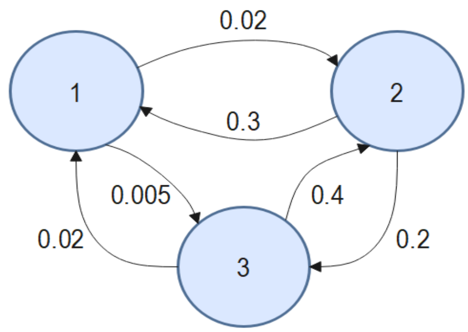

Figure 24: Transitions diagram of CTMC.

2.6.3 Study of a continuous-time Markov chain [14]

There is a close relationship between CMTCs and CMTDs. Indeed, for each CMTC defined by its infinitesimal generator Q, we can define a CMTD called an included Markov chain (IMC). This last is defined by its transition probability matrix P. The terms of the matrix P are obtained as follows:

We know that: $q_{i,j} = q_i . P_{i,j}$.

Then we can say that:

$$P_{i,j} = \begin{cases} q_{i,j} / q_i & \text{si } i \neq j \text{ et } q_i \neq 0, \\ 0 & \text{si } i \neq j \text{ et } q_i = 0\ ; \end{cases} \quad \text{and} \quad P_{i,i} = \begin{cases} 0 & \text{si } q_i \neq 0\ , \\ 1 & \text{si } q_i = 0. \end{cases}$$

Equation 9

Thanks to this result, the study of CMTCs is greatly facilitated. Indeed, to ensure the existence of a steady state, the CMTC must be irreducible from the following results:

- **Result 1**: A CMTC is irreducible if and only if it includes CMTD is irreducible.
- **Result 2**: A finite and irreducible CMTC is ergodic.
- **Result 3**: Consequently, a finite and irreducible CMTC tends towards a stable distribution π after the passage of infinite time. The vector π is the unique solution of the system of matrix equations following:

$$\begin{cases} \pi . Q = 0 \\ \sum_{i \in E} \pi_i = 1 \end{cases}$$

Equation 10

4 Queueing Systems

A queuing phenomenon can be described as a system composed of a certain number of (finished or unfinished) waiting places on one or more servers and customers arriving at random moments. Customers wait, get served according to specified rules, and exit the system. It is common in many real-world systems, such as traffic networks, communication systems, service facilities, and manufacturing processes.

The identification of traditional queue systems is mainly based on three elements: the stochastic process describing the arrival of clients in the system, the service mechanism (the number of servers and the probabilistic law describing the duration of services), and the discipline of waiting.[36]

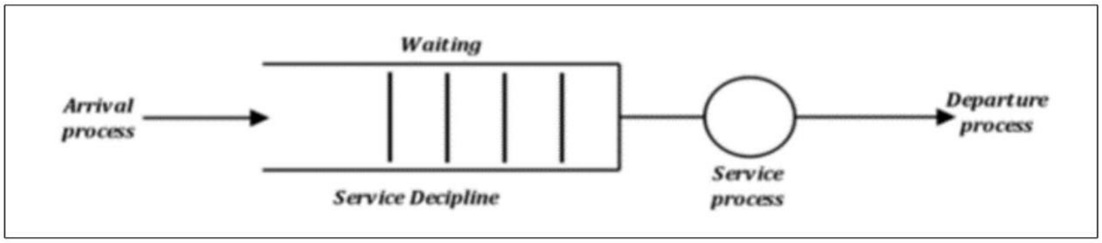

Figure 25: Schematic representation of a classic queue.

4.1 Characteristics of a queueing system [36]

A queuing system is usually described by five essential characteristics of the queuing process:

- **Arrival process** defines the distribution of the intervals separating two consecutive arrivals. Queue arrivals can be deterministic or random, dependent or independent, individual or grouped, homogeneous or heterogeneous.
- **Calling population:** The calling population is the potential number of customers who require a service from the system. It can be finite or infinite. Most systems with a substantial number of customers are considered infinite.
- **Service process:** It can be measured by the number of served customers by the system per unit of time. The service process and time are considered identical and independent in most systems.
- **The number of servers:** The server is the entity that provides the service to the customers waiting in the queue. A system may possess one to multiple servers. The number of servers can affect the system's performance as it can serve multiple customers simultaneously.
- **Queueing discipline:** It can be described as the method used by the server to choose the next client to be served after the server has completed the service of the current customer.
    - There are many queueing disciplines:
- **FIFO** (First in, First out): Customers are served one at a time according to their arrival time.
- **LIFO** (Last in First Out): Customers are served one at a time, but the last customer entering the queue will be served first.
- **Shortest job first:** The next customer to be served has the shortest service time.
- **Priority:** All customers are assigned a given priority level; customers having the highest priority are served first.

- **Random:** The next customer to be served is chosen randomly from the queue.

4.2 Kendall's notation [35]

Kendall's notation represents a queueing system: A /B / C/D /E. Its terms are as follows:

- **A:** Inter-arrival time or the arrival process.
- **B:** Type of service time.
- **C:** The number of servers in the system.
- **D:** The system capacity.
- **E:** The queue discipline or order of service.

4.3 Vacation queueing systems

In classical queuing models, servers are always available. However, in many practical queuing systems, servers may become unavailable for a while for various reasons. This period of server absence may represent the server's working on supplementary jobs, being checked for maintenance, or simply taking a break.[37]

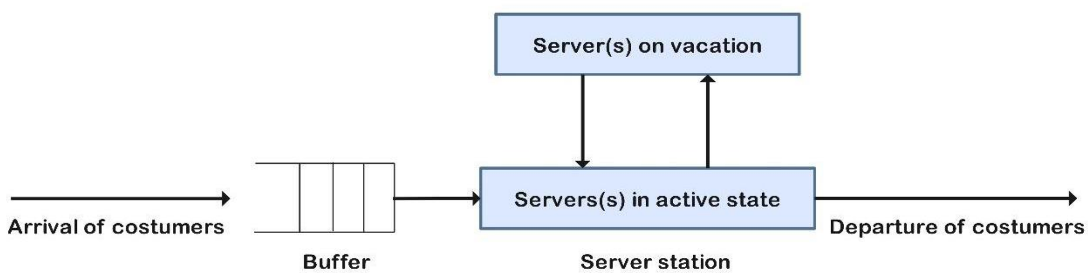

Figure 26: Schematic representation of a vacation queueing system.

A vacation queueing system consists of the same parts of a classical queueing system and two more parts that describe the vacation of the server(s). A vacation queueing system can be characterized as follows:

- **The vacation start-up** rule determines when the server starts a vacation. There are two major types, namely exhaustive and non-exhaustive services. With an exhaustive service, the server cannot take a vacation until the system becomes empty. On the other hand, the server in a non-exhaustive service can take a vacation even when the system is not empty. A semi-exhaustive service rule may be used in a multi-server system if some servers take a vacation. Another vacation start-up rule is the service interruption during the progress of customer service. The service interruption may represent a machine failure during this operation.
- **Vacation termination** rule determines when the server resumes serving the queue. Two popular rules are the multiple vacation policy and the single vacation policy. A multiple vacation policy requires the server to keep taking vacations until it finds at

least one customer waiting in the system at a vacation completion instant. In contrast, under a single vacation policy, the server takes only one vacation at the end of each busy period. After this single vacation, the server serves the waiting customers, if any or stays idle.

- **Vacation duration period:** The duration of the vacation depends on the policy followed. They are considered independent and identically distributed random variables with a particular law of probability, depending on the system's characteristics.

4.3 Priority queueing

Queuing with priority is a specific type that assigns different priorities to requests or data packets. This allows for the efficient management of resources and ensures that high-priority requests are processed before low-priority ones. Queuing with priority is widely used in various applications, including network traffic management, operating systems, and real-time systems.

- **Relative priority:** characterized by the fact that a higher-priority customer may not, under any circumstances, interrupt the service of a lower-priority customer. It must wait until the lower-priority customer's service is finished before being served. The queuing system with relative priority can also be seen as a waiting system with an unreliable server. The failure is considered after the client being served has been freed. This model is known as "Breakdown postponable interruption."
- **Absolute priority:** Unlike the previous case, here, a higher-priority customer has the right to interrupt the service of a lower-priority customer. Service to a lower-priority customer. The interrupted service will then be resumed from the point it was suspended.

5 Conclusion

This chapter first discussed the fundamental concepts of stochastic processes, especially discrete and continuous-time Markov chains. Then we started with PN and GSPN, the latter being powerful forms for analyzing the performance of different systems. On the one hand, GSPNs can efficiently represent instantaneous events (synchronization, blocking mechanism, and competition) and non-instantaneous events that require a particular execution time (processing a request).

We introduced queues as a valuable modeling formalism. Then we presented the fundamental concepts of a queuing system, Kendall's notation, and a brief description of vacation and priority queuing systems and their operation. In the next chapter, we will review some research on the utilization of formal methods for performance evaluation of WSNs. Then we will proceed to propose energy savings models based on the vacation and priority disciplines.

CHAPTER 3

MODELING AND ANALYSIS OF WSNs

1 Introduction

Wireless sensor networks (WSNs) are increasingly important as they are present in many applications, such as industrial process automation, air traffic control systems, and hospital patient monitoring. The nodes forming these networks capture information about the environment, process it and communicate it to the end user with possible time constraints while consuming little energy. Hence the introduction of the duty-cycle technique, which periodically switches the node's network interface from the active state to the sleep state to save more energy.

To optimize energy consumption in WSNs, we propose six models based on vacation and priority queueing theory. In the first model, we used N-policy to minimize energy consumption by reducing the busy period. In the second and third, we improved the N-policy model by adding the DI state to reduce the node's state switching energy. Moreover, in the fourth and fifth models, we considered two types of traffic for the node; in the last one, we implemented reliability insurance for essential frames.

In the next section, we present the related research concerning the use of formal methods for evaluating the performance of WSNs. Then we present models based on working vacation disciplines using the GSPNs. These will be described and analyzed to generate the TCMCs. Then, we will give the transition rates between the different states of the TCMCs, allowing us to obtain the algorithm of the infinitesimal generator. Finally, we will end this chapter with a conclusion.

2 Related Works

Several solutions have been proposed in the literature to reduce the power consumption of sensor nodes in WSNs. Vacation queuing models are widely used to model and analyze the behavior of the nodes from an energy consumption and delivery delay perspective. Some work was realized with an infinite buffer queueing system and infinite source without considering the limited buffer size of the device for simplicity's sake. Of those, we can cite the following:

- S. Ghosh & S. Unnikrishnan [23] aimed to entangle one of the sensor node's challenges, which is power consumption, caused by the difficulty of replacing their batteries when deployed in remote or hostile areas. Their approach alleviates this problem using the N-policy M/M/1 queuing model. They studied the behavior of the sensor node in the idle state and the busy state. Then the authors used mathematical preliminaries of the Markovian queuing system to determine the rates of transitions. To show the effectiveness of their approach, they simulated the model for a single node in WSN with different values of the queue threshold N. They concluded that the mean power consumption could fall by a considerable amount by selecting an optimum threshold value validating the theoretical results and effectiveness of the model.

- In another research, S. Ghosh and S. Unnikrishnan [24] proposed a new solution to mitigate the N-policy incurred delay in WSNs. They use the N-Policy M/M/1 Queuing model to the sensor node. The sensor starts its radio transmitter in its model when the buffer contains at least N packets. The node switches to the idle state when the buffer is empty. Transitioning between idle and busy states can be energy-consuming, hence the aim of reducing the number of transitions between states. They resolve analytically the Markovian queuing system. After analysis, they concluded that power consumption decreases by setting a threshold N and that the delay is proportional to the threshold N since the increase in the arrival rate leads to the faster filling of the buffer and which leads to shorter waiting times. After performing a simulation, they confirmed the power consumption reduction, especially when an optimal value of N is set. Furthermore, the effectiveness of the Min (N, T) policy was proven to reduce the latency incurred by an N–policy.
- R. Maheswar and R. Jayaparvathy [25] performed their study on Heterogeneous Sensor Networks (HSN), a network divided into clusters. Each cluster consists of two types of sensor nodes: an H-sensor and an L-sensor. The L-sensors are distributed around the H-sensor, which is each cluster's head. An H-sensor can be in one of two operational modes:
    - The sleep mode, where the energy consumption of the node is minimal, and the sensor has no interaction with the external world.
    - The active mode, where the node may be in the idle state where the node can listen to the channel, receive packets, generate data or transmit data packets.

In addition, two more sub-operational states during the active mode, the IDLE and the BUSY state. The H-sensor remains in the IDLE state and switches to BUSY state when the node’s buffer is filled at least with a threshold number of packets (N), i.e., queue threshold, and the node switches back from BUSY state to IDLE state when there are no packets in the buffer. The proposed analytical model considers channel contention and is based on the M/G/1 queuing model. Regarding performance parameters, the tests were based on the average energy consumption and the mean delay, which showed an increase of up to 68% for the optimal threshold value. When compared to the simulation results, the obtained values validate the approach.

- R. Sudhesh and M. Shapique[26] performed a transient analysis to achieve an optimal power consumption scheme for wireless sensor networks. In their system, the node transits between four different states. After serving all the events (packets) in the queue, the server would go to a shutdown state for a random duration V, where events can join the queue but will not be served until the number of queued packets reaches the threshold K. Moreover, it requires start-up time and a change of state to resume service. If the number of packets in the system is less than k and the state is the shutdown state, the server goes into the inactive period. If not, it goes into the wake-up state. In the case of this model, the transient and steady-state system size probabilities of the system are obtained in a closed form, and mean, variance, probability of the system being in a power-saving mode, and mean power consumption are computed to measure the system's performance. Eventually, they concluded that there exists a threshold value for each arrival rate to reduce the system's power consumption.

- D. Nageswari et al. [27] proposed an N-policy M/M/1 and studied sensor nodes in clusters. A node in a cluster would either be a cluster member (CM) or a cluster head (CH). During their active period, all nodes cycle between two states, the IDLE state, and the BUSY state:
  - To switch from the IDLE state to the BUSY state, the number of parquets in the buffer must reach the threshold (N).
  - The node switches back to the IDLE state when the buffer is empty.

When in the Busy state, the arrival of data packets to sensor nodes is assumed to follow a Poisson process with a mean arrival rate per node λ1 and λ2, respectively.

To measure the performance of the proposed approach, an analytical study of the model is conducted, and the average energy consumption and mean delay are the chosen performance parameters to be evaluated. An optimal value of the queue threshold N is deducted for minimal power consumption. Finally, a simulation is performed, and the obtained values are compared to the analytical results to validate them and confirm the accuracy of the approach.

Other researchers opted for a finite buffer vacation queueing system for WSNs as a means to obtain better overall performance from the system:

- In order to improve the efficiency of the energy-saving mechanism and reduce delay time in wireless sensor networks, B. Boutoumi and N. Gharbi [28] combined normal vacation and working vacation policy to introduce a new energy-efficient technique for full duplex WSN they called the “two thresholds working vacation policy.” They used an M/M/1/K queueing system based on N-policy to model an individual sensor node. Then the notion of the two thresholds working vacation policy is introduced to the same model. Generalized Stochastic Petri Nets (GSPNs) are used to model and analyze the behavior of the sensor node with different policies. Finally, they developed the formulas of the main stationary performance indices, energy consumption, and latency. They compare the proposed policy with the N-policy and find that the “two thresholds working vacation policy” is more energy efficient and reduces the latency and the blocking probability, which confirm the effectiveness of the technique in improving the lifetime of the sensor network.

- Aiming to solve the Energy Hole Problem (EHP) that is caused by the amount of energy consumed by the nodes that are close to the sink, Der-Chen Huang et al. [29] proposed a new energy conservation scheme based on the N-policy. The system is modeled by an M/M/1/K. They solved the Markovian system analytically and computed at a steady state the formulas of the main performance indices and energy consumption. Then, they conducted experiments and analyses to show the effectiveness of their proposition. They also realized an experiment to determine an optimal threshold N per mean arrival rate that would minimize energy consumption.
- Jyun-Fan Ke et al. [30] modeled a sensor node by M/M/1/K with N-policy. The novelty of this research is to consider two priority classes of packets. The node would be in one of the four states, Sleep State, Idle State, Busy State, and Transmit State, and would have two separate queues, each queue with different input and service rate according to the priority of the packets in the queue. Each queue would receive its packets respectively until reaching the limit K in both idle and busy periods. When a queue reaches the threshold, N, the sensor tries to access the channel to transmit. If successful, no more packets are received. Based on the queuing theory and the birth-death process, they build a mathematical model capable of quantifying various energy consumption parameters. From the obtained results, an optimal value of the threshold N can be deducted, which would solve the energy hole problem (EHP) in the system and result in an overall longer life span of the network.
- For the same objective as the previous research, Changzhen Zhang et al. [31] suggest an adaptive N-policy sleep scheduling for sensor nodes. They model the system with an M/M/1/C queue with N-policy. To minimize the additional waiting delay incurred by the N-policy, the authors introduce a new state called the wait state. The sensor node model was analyzed analytically. The formulas of energy consumption and the waiting delay are computed. Using the performed Monte Carlo simulation, the timely reliability was estimated. They found that the results of numerical experiments demonstrate that the proposed sleep scheduling with N-policy and wait time outperforms the existing N-policy sleep scheduling on the node energy consumption, packet delay, and the timely reliability of WSNs.
- B. Boutoumi and N. Gharbi [32] also modeled the sensor nodes as a finite vacation queueing system and proposed the N-policy as a wake-up mechanism for the queue to switch between the idle to the busy state, where the node would switch to the busy state when the number of packets stored in the buffer would reach the threshold of N packets, this would reduce the amount of energy consumed in the busy state. They also proposed a new policy aiming to reduce the energy consumption of the node as

well as the latency delay of the packets. In this model, the transition from the idle to the busy state can happen when one of two conditions is fulfilled:

- When the Nth has arrived and is stored in the buffer (N-policy).
- At the end of a random vacation, that is exponentially distributed at random duration even if the number of stored packets is less than the threshold N.

The GSPNs are used to model and analyze the comportment of the system. Several performance indices are computed, including energy consumption and waiting delay. The experiment results show that the proposed policy (Hybrid policy) reduces more energy and waiting delay than the N-policy.

- Due to the N-policy incurred waiting delay in WSN, B. Boutoumi, and N. Gharbi [33] opted for a priority M/M/1/K vacation queuing system with N-policy to model the comportment of the sensor node. A node switches between two states: idle and busy. They consider two classes of data traffic: High priority with the mean arrival rate λ1 and low priority traffic with the mean arrival rate λ2. The mean service rate is the same for the two classes μ. The sensor node, which is initially idle, turns to a busy state if the number of queued packets of the two classes reaches N. At this moment, it can start transmitting packets exhaustively according to their priority level. The CTMC is used to model and analyze the system. The main stationary performance indices and energy consumption formulas were determined as a function of the network parameters. Numerical results demonstrated the proposed N-policy priority queueing model's efficiency in energy and latency.

Table 1 illustrates the related works cited above:

Table 1: Comparative table for the different preexisting models.

References	Proposed model	Methodology	Performance indices	Paquet types
[23]	M/M/1 Queueing with N-policy.	-Analytical model. -Simulation.	-Average power consumption.	One type
[24]	M/M/1 Queueing with N-policy min (N, T) policy.	-Continuous Time Markov Chain. -Analytical model.	-Power consumption. -Average delay.	One type
[25]	M/G/1 Queueing model with N-policy.	-Analytical model. -Simulation.	-Average power consumption and -Average delay	One type
[26]	M/M/1 Queuing with start-up times and N-policy.	-Analytical model.	-Mean variance. -Mean mode probabilities -Mean power consumption	One type
[27]	M/M/1 Queuing with N-policy.	-Analytical model. -Simulation	-Average energy consumption -Energy Savings	One type

			-Mean delay.	
[28]	M/M/1/K Queuing with N-policy.	-Generalized Stochastic Petri Nets (GSPNs).	-Average energy consumption -Mean sojourn time. -Blocking probability.	One type
[29]	M/M/1/K Queuing with N-policy.	-Analytical model.	-Average energy consumption.	One type
[30]	M/M/1/K priority Queuing with N-policy.	-Case study.	-Average power consumption. -Average delay	Two types
[31]	M/M/1/C Queuing with Adaptive N-policy.	-Analytical model. -Simulation.	-Average power consumption and -Average delay. -Timely reliability.	One type
[32]	M/M/1/K Queueing with N-policy and Hybrid policy.	Generalized Stochastic Petri Net (GSPNs).	-Energy consumption -Waiting delay	Two types
[33]	M/M/1/K Priority Queueing with N-policy.	Continuous Time Markov Chain (CTMC).	-Energy consumption -waiting delay -blocking probability	Two types

3 Modeling of WSNs with the GSPN

The behavior of a single wireless sensor node can be considered as a finite queue with a single server and infinite arriving frames, the arrival process is a Poisson process with parameter λ, and the service time follows the exponential law with the parameter μ.

The following models all share the ordinary concept as a behavior base, with different additions for different use cases.

3.1 Ordinary model

An ordinary sensor is modeled using the GSPN to make a baseline performance index for comparison with other models. Therefore, an analysis will be made to generate the CMTC of the model, which allows for obtaining the infinitesimal generator.

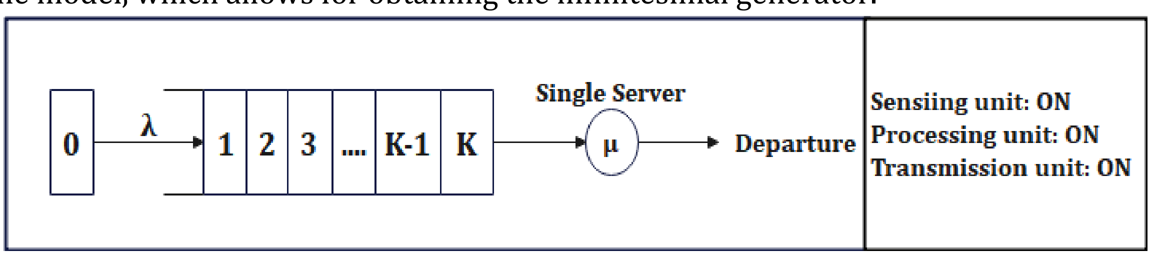

Figure 27: Pictorial representation of the model.

3.1.1 Description of the GSPN

The GSPN modeled in Figure 28 represents the M_0 model, in which:

- The place Capacity: represents the buffer capacity of a sensor node.

- The place Buffer: represents the buffer of the sensor node.
- The place Free: represents the state of the transmission unit.
- The Arrival timed transition: represents the arrival of a packet in the sensor node.
- The transmit timed transition: represents the transmission of frames by the sensor node.

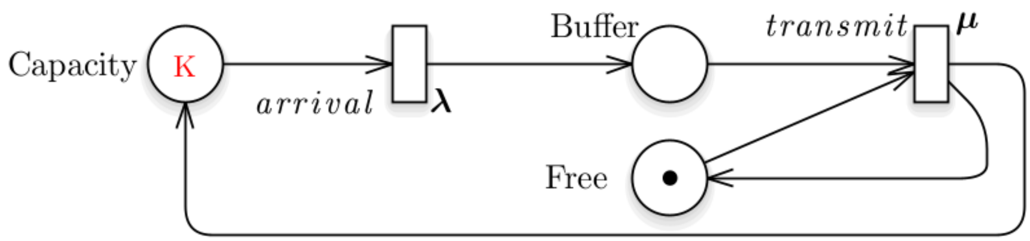

Figure 28:Ordinary model (M0).

3.1.2 Analysis of the GSPN

The presentation of the M_0 model follows this system of equations for whatever value for K.

$$\begin{cases} M(Capacity) + M(Buffer) = K \\ M(Busy) = 1 \end{cases}$$

Based on these two equations, the system can be described in this manner:

- i represent the number of frames in the buffer.
- j represents the value that always indicates the presence of a token in the space Busy.
- The marking for the different places can be concluded in this manner:

Table 2: The Marking corresponding to the GSPN of model M0.

M(Capacity) = K-i	0≤i≤K
M(Buffer) = i	
M(Free) = j	j=1

The CMTC obtained after this analysis is shown in Figure 29.

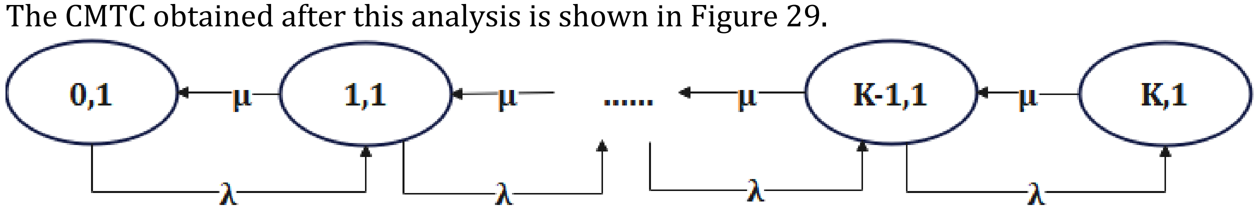

Figure 29: The CTMC corresponding to the GSPN of the Ordinary model.

3.1.3 Construction algorithm of the infinitesimal generator

After the analysis of the CTMC presented in Figure 29, the number of states of the CTMC which corresponds to the number of tangible markings is equal to K+1 states.

The infinitesimal generator is constructed as follows:

$$GI[(i,j),(x,y)] = \begin{cases} gi[(i,j),(x,y)] if (i,j) \neq (x,y) \\ -\sum gi[(i,j),(x,y)] if (i,j) = (x,y) \end{cases}$$

gi[(i,j),(x,y)] represents the transition rate between the state (i,j) and the state (x,y). Table 3: Representation of transition rates between states for M0.

Table 3: Representation of transition rates between states for M0.

Description	Outgoing State	Transition Rate	Incoming State
0≤i≤K, j=1	(i, j)	Λ	(i+1, j)
0≤i≤K, j=1	(i, j)	μ	(i-1, j)

These transitions allow us to build the algorithm given below, to compute the values of the different elements of the infinitesimal generator GI.

Algorithm M_0: Construction of the infinitesimal generator for the ordinary model

for (int i=0; i<K; i++) {
for (int j=1; i<K+1; i++) {
GI [(i, i)] [(i+1, j)] = λ;
}
}
for (int i=K; i<0; i--) {
for (int j=K-1; j≤K; j--) {
GI [(i, j)] [(i-1, j+1)] = μ;
}
}

K: Buffer Capacity; λ: Arrival rate; μ: Service rate.

3.2 Energy saving proposed models

The sensor node in each model here is a Full Function Device (**FFD**) that supports the 802.15.4 mac layer. This node acts as a pan coordinator in a star topology. During its cycle, it will be either Idle.

It is considered a finite buffered vacation queuing system, in which the frames (relayed and sensed) arrive according to the Poisson distribution, with the mean **λ**, and the service process is distributed exponentially with the mean rate **μ**.

The node will remain in the idle state to conserve energy which corresponds to the vacation state of a vacation queueing model, switches to busy state when the buffered frames sensed or relayed from the neighbor nodes reach the threshold number (N), and the node switches back from busy state to delayed idle state when there are no frames in the buffer, in some models the node keeps track of how many times it went into the delayed idle state and can't exceed a predetermined value (M), it will stay in this state until either the 1 period expires or a new frame arrives before that.

3.2.1 Model M1(N-policy)

A description of a generic sensor is made using the GSPN. Subsequently, an analysis of the latter will be made to generate the CTMC of the model, which allows the obtaining of the infinitesimal generator.

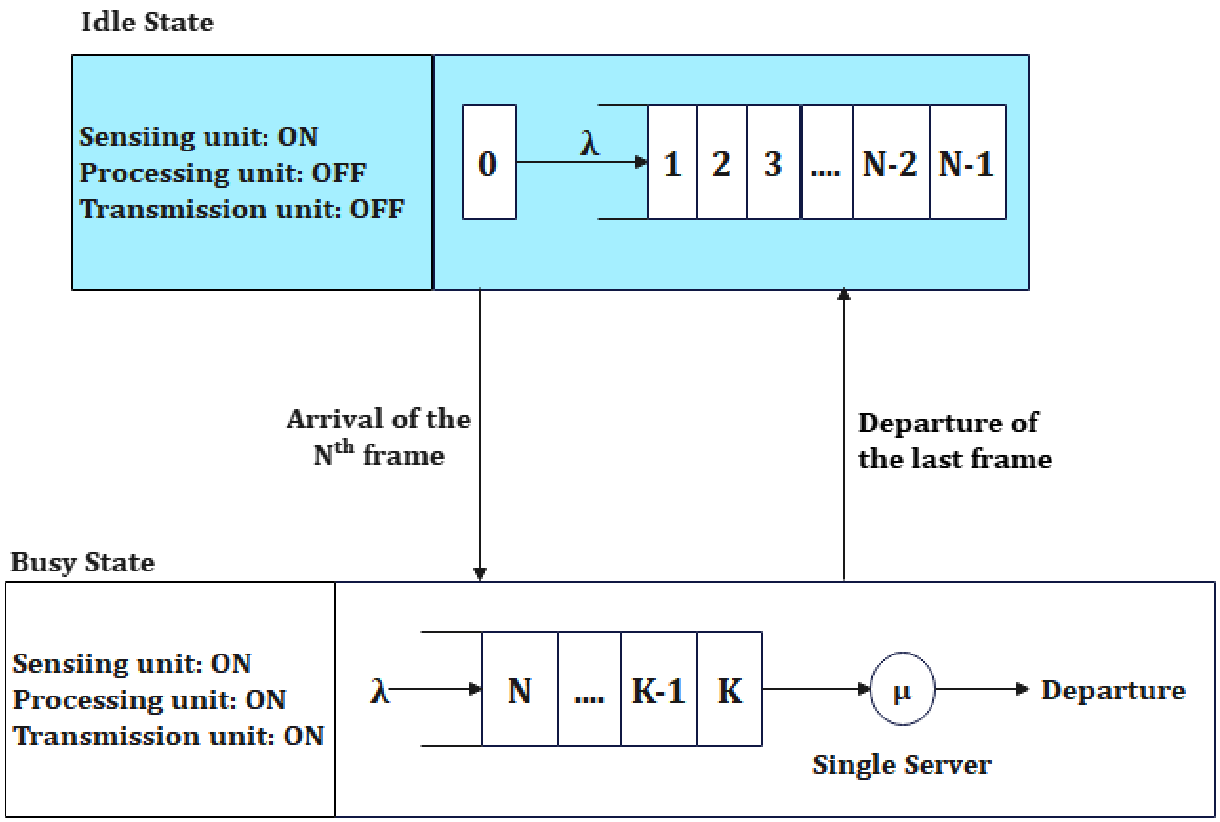

Figure 30: Pictorial representation of the M1.

3.2.1.1 Description of the GSPN

The GSPN modeled in Figure 30 represents the M_1 model which consist of adding these places and transitions to the M_0 model:

- The place Idle: represents that the node is in idle state.
- The Vac_Begin transition: it allows the sensor to take a working vacation.
- The Vac_End timed transition: represents the transmission of frames by the sensor node.

The N-policy is used to pass from the idle state to the busy state. If the number of frames in the buffer is lower than the threshold N, then our system is in vacation state. When the number of frames is equal to N, the system enters the busy state by firing the transition Vac_End. The sensor node goes back to idle state when the buffer is empty, meaning firing the transition Vac_begin.

Thus, the initial marking is represented as follows:

M0={M(capacity), M(Buffer), M(Free), M(Vacation)=(K,0,0,1)}

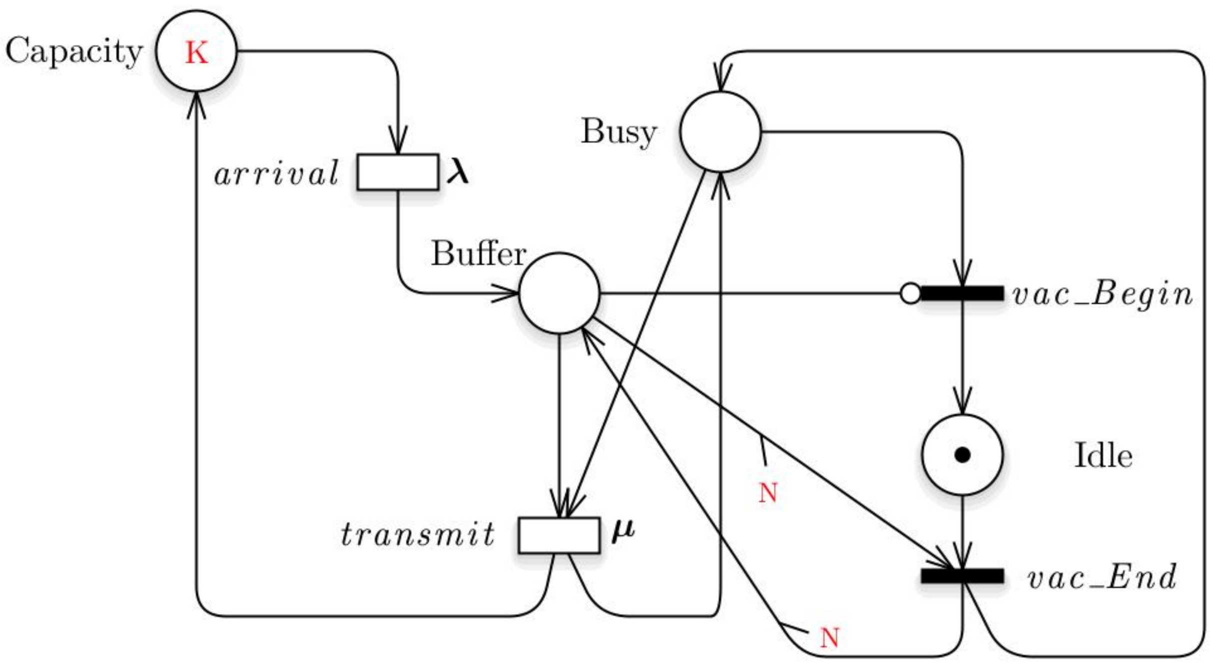

Figure 31: Petri net of the N-policy model M1.

3.2.1.2 Analysis of the GSPN

The presentation of the M_1 model follows this system of equations for whatever value for K.

$$\begin{cases} M(Capacity) + M(Buffer) = K \\ M(Busy) + M(Idle) = 1 \end{cases}$$

Based on these two equations, the system can be described in this manner:

- i represent the number of frames in the buffer waiting for transmission.
- j represents the state of the transmission unit either Busy or Idle.

The marking for the different places can be concluded in this manner:

Table 4: The Marking corresponding to the GSPN of model M1.

M(Capacity)= K-i	
M(Buffer) = i	0≤i≤K

M(Busy) = j	
M(idle)=1-j	0≤j≤1

The CMTC obtained after this analysis is shown in Figure 32.

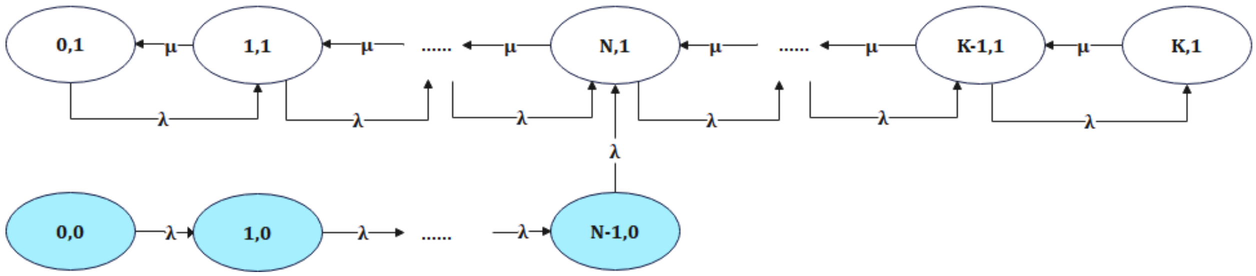

Figure 32: The CTMC corresponding to the GSPN of the M1 model.

3.2.1.3 Construction algorithm of the Infinitesimal generator

After the analysis of the CTMC presented in Figure 32, the number of states of the CTMC which corresponds to the number of tangible markings, is equal to K+Nstates.

The infinitesimal generator is constructed as follows:

$$GI[(i,j),(x,y)] = \begin{cases} gi[(i,j),(x,y)] if (i,j) \neq (x,y) \\ -\sum gi[(i,j),(x,y)]\, if\, (i,j) = (x,y) \end{cases}$$

gi[(i,j),(x,y)] represents the transition rate between the state (i,j) and the state (x,y).
Table 5: Representation of transition rates between states for M1.

Table 5: Representation of transition rates between states for M1.

Description	Outgoing State	Transition Rate	Incoming State
0≤i≤N-2, j=0	(i, j)	Λ	(i+1, j)
0≤i≤K-1, j=1	(i, j)	Λ	(i+1, j)
i=N-1, j=0	(i, j)	Λ	(i+1, j+1)
0≤i≤K-1, j=1	(i+1, j)	μ	(i, j)
i=1, j=1	(i, j)	μ	(i-1, j-1)

These transitions allow us to build the algorithm given below, to compute the values of the different elements of the infinitesimal generator GI.

Algorithm M1: Construction of the infinitesimal generator for the N-policy model

for (int i=1; i<K; i++) {
GI[(i,1)] [(i+1,1)] = λ;
GI[(i+1,1)] [(i,1)] = μ;
}
for (int i=0; i<N; i++) {
GI [(i,0)] [(j+1,0)] = μ;

} GI[(N-1,0)] [(N,1)] = λ; GI[(1,1)] [(0,0)] = λ;
K: Buffer Capacity; N: Threshold; λ: Arrival rate; μ: Service rate.

3.2.2 Model M2 (DI+N-policy with one type of traffic and no counter)

A description of a generic sensor is made using the GSPN. Subsequently, an analysis of the latter will be made to generate the CTMC of the model, which allows the obtaining of the infinitesimal generator.

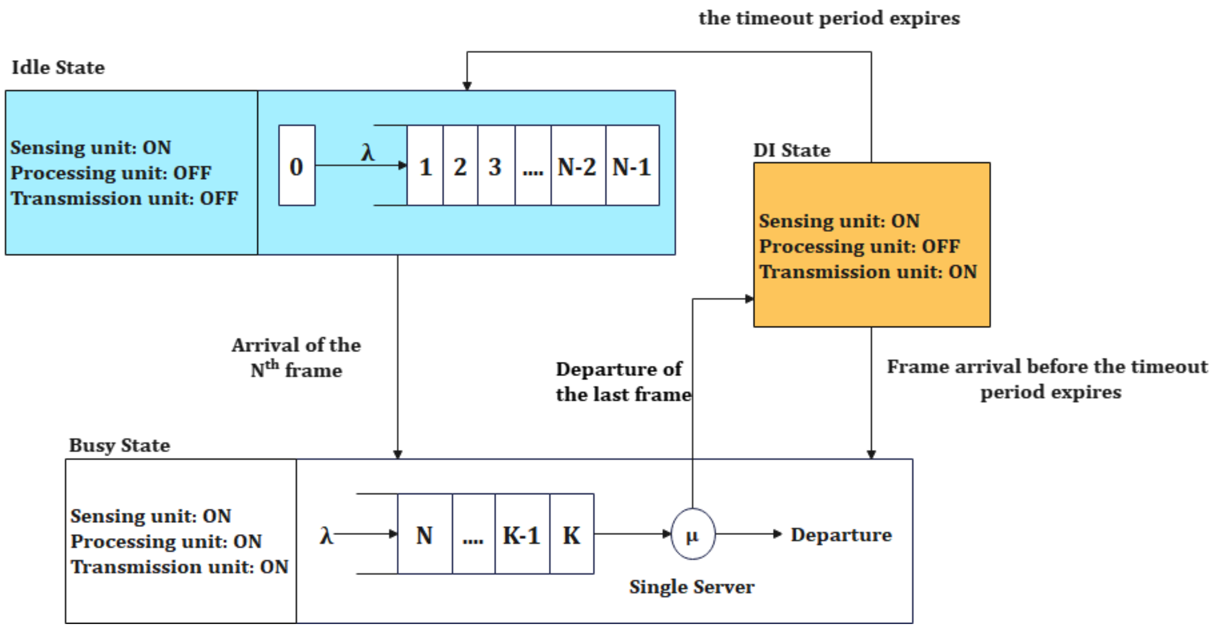

Figure 33: Pictorial representation of the M2.

3.2.2.1 Description of the GSPN

This model combines two policies into one. The node passes to the Busy state as soon as the number of frames waiting in the buffer reaches the threshold N. The objective of choosing this model is to avoid buffer saturation and to minimize the transition energy from the idle state to the standby state. And the node switches back from busy state to delayed idle state when there are no frames in the buffer. It will stay in this state until either the1 period expires or a new frame arrives before that.

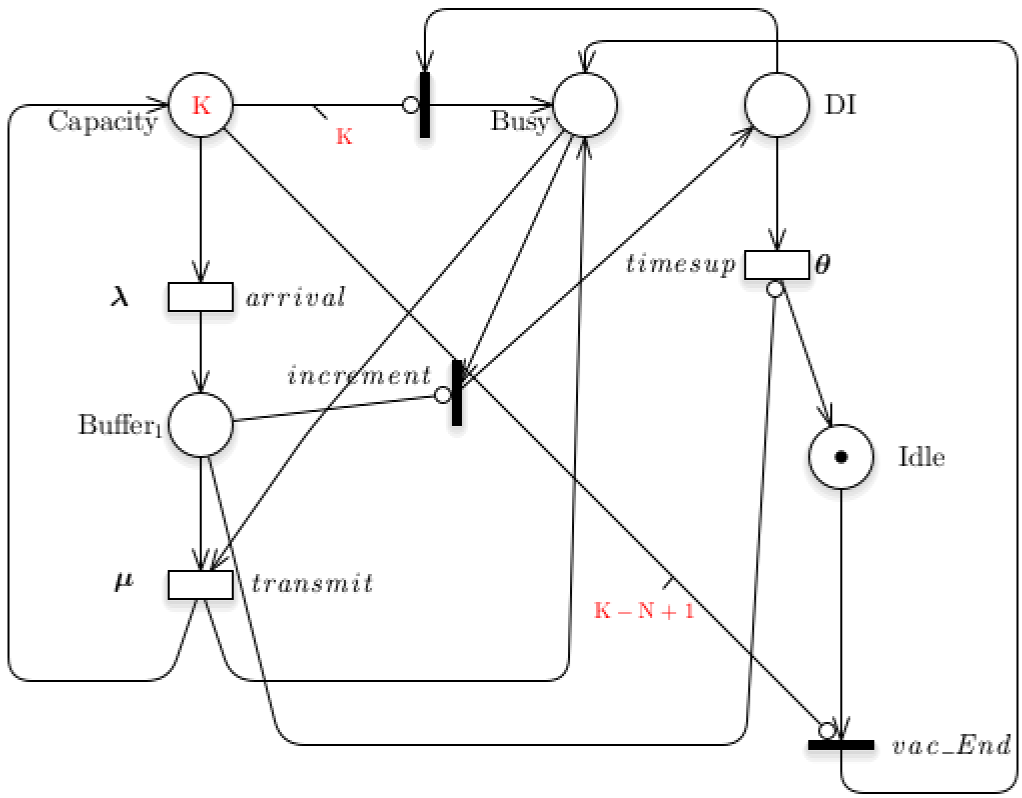

Figure 34: Petri net of model M2

3.2.2.2 Analysis of the GSPN

The presentation of the M3 model follows this system of equations, for whatever value for K.

$$\begin{cases} M(Capacity) + M(Buffer) = K \\ M(Busy) + M(Idle) + M(Delayed\_Idle) = 1 \end{cases}$$

Based on these two equations, the system can be described in this manner:

- i represent the number of frames in the buffer.
- u, v represents the state of the transmission unit in its different modes: Free, Idle, Delayed idle.
- The marking for the different places can be concluded in this manner:

Table 6:The Marking corresponding to the GSPN of model M2

M(Capacity) = K-i	$0 \leq i \leq K$
M(Buffer) = i	
M(Busy) = u	u+v=1
M(Idle) = 1-u-v	

M(Delayed_Idle) = v

The CMTC obtained after this analysis is shown in Figure 35.

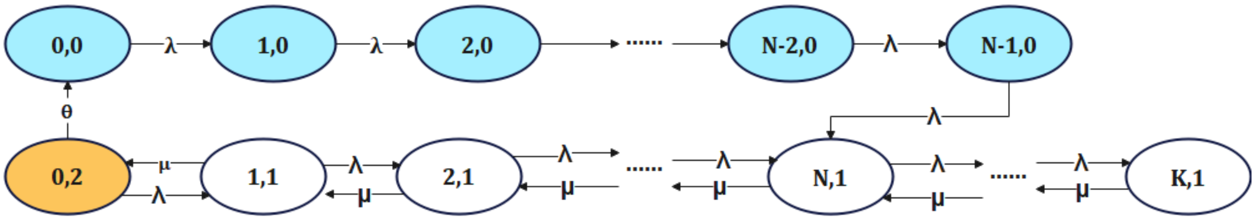

Figure 35:: The CTMC corresponding to the GSPN of model M2.

3.2.2.3 Construction algorithm of the Infinitesimal generator

After the analysis of the CTMC presented in Figure 35, the number of states of the CTMC which corresponds to the number of tangible markings is equal to K+N+1 states.

The infinitesimal generator is constructed as follows:

$$GI[(i,j),(x,y)] = \begin{cases} gi[(i,j),(x,y)] if (i,j) \neq (x,y) \\ -\sum gi[(i,j),(x,y)]\, if (i,j) = (x,y) \end{cases}$$

gi[(i,j),(x,y)] represents the transition rate between the state (i,j) and the state (x,y).

Table 7:Representation of transition rates between states.

Description	Outgoing State	Transition Rate	Incoming State
0≤i≤N-2	(i, 0,)	λ	(i+1, 0)
1≤i≤K-1	(i, 1)	λ	(i+1, 1)
1≤i≤K-1	(i+1, 1)	μ	(i, 1)
i=0	(i, 2)	λ	(i+1, 1)
i=0	(0, 2)	Ɵ	(0, 0)
i=0	(i+1, 1)	μ	(i, 2)

These transitions allow us to build the algorithm given below, to compute the values of the different elements of the infinitesimal generator GI.

Algorithm M2: Construction of the infinitesimal generator for Delayed Idle and N-policy policies

```
for (int i=0; i<N-1; i++) {
 GI [(i, 0)] [(i+1, 0)] = λ;
}
for (int j=1; j<K; j++) {
GI [(i, 1)] [(i+1, 1)] =λ;
GI [(i+1, 1)] [(i, 1)] = μ;
}
GI [(N-1, 0)] [(N, 1)] = λ;
GI [(0, 2)] [(1, 1)] = λ;
```

GI [(1, 1)] [(0, 2)] = μ; GI [(0, 2)] [(0, 0)] = ϴ;
K: Buffer Capacity; N: Threshold; M: Max value for the Counter; λ: Arrival rate; μ: Service rate; ϴ: DI rate.

3.2.3 Model M3 (DI + N-policy with counter and one type of traffic)

A description of a generic sensor is made using the GSPN. Subsequently, an analysis of the latter will be made to generate the CTMC of the model, which allows the obtaining of the infinitesimal generator.

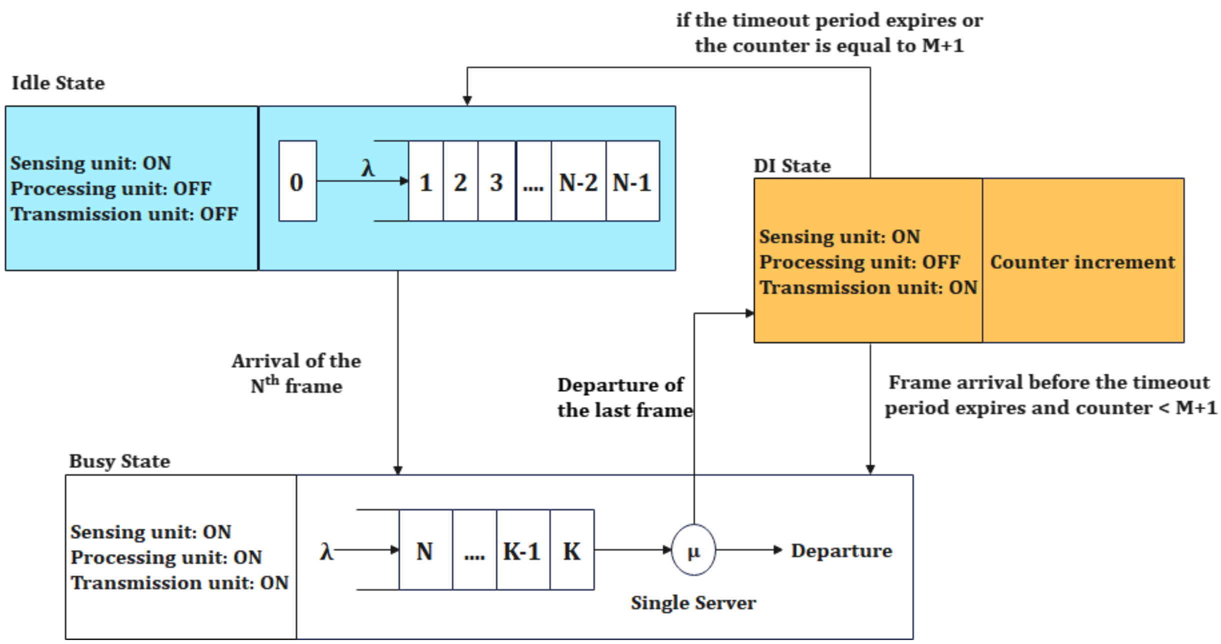

Figure 36:Pictorial representation of the M3.

3.2.3.1 Description of the GSPN

The GSPN modeled in Figure 37 represents the M3 model, which consists of adding these transitions and places to the M2 model:

- The place Counter: represents the number of times the node went to the state Delayed Idle.
- The timesup timed transition: represents the end of the time permitted in delayed idle state after a random time $1/\theta$.
- The di_end transition: represents the passage from delayed idle state to free state.

The initial marking is represented as follows:

M0={M(capacity), M(Buffer), M(Free), M(Transmission), M(DI), M(Counter), M(Vacation)=(K,0,0,0,0,0,1)}

This marking represents the absence of packets in the sensor node at the initial phase. While the buffer is empty or the number of packets is less than the threshold N, there will be no transmission to be made. Once the number of tokens in the buffer reaches the threshold, the transition vac_End will be fired, which transfers the token to the free state, therefore the start of the transmission with a transition rate μ.

After the transmission of all the packets (no token in the Buffer) the node can pass to delayed idle state and increment the counter by firing the transition increment. This can only be for a set number of times (M). depending on which event happen first, either an arrival of a new token so the t0 transition will be fired and going to state Free, or the passing of a random time $1/\theta$ so the token will be fired to state vacation, and counter will be rested to the initial marking.

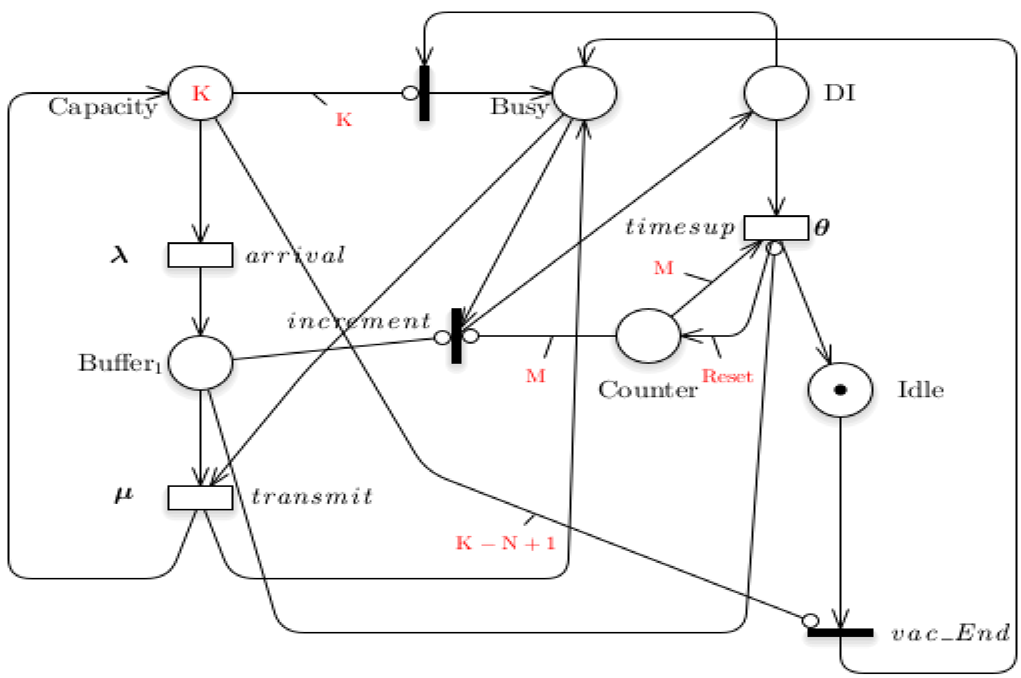

Figure 37:Petri net of M3 model.

3.2.3.2 Analysis of the GSPN

The presentation of the M3 model follows this system of equations, for whatever value for K.

$$\begin{cases} M(Capacity) + M(Buffer) = K \\ M(Busy) + M(Idle) + M(Delayed\_Idle) = 1 \end{cases}$$

Based on these two equations, the system can be described in this manner:

- i represent the number of packets in the buffer.
- u, v represents the state of the transmission unit in its different modes: Busy, Idle, Delayed idle.
- c represents the number of times the node went in Delayed_Idle mode.

The marking for the different places can be concluded in this manner:

Table 8:Marking corresponding to the GSPN of model M3.

M(Capacity) = K-i	0≤i≤K
M(Buffer) = i	
M(Busy) = u	u+v=1
M(Idle) = 1-u-v	
M(Delayed_Idle) = v	
M(Counter)=c	0≤c≤M

The CMTC obtained after this analysis is shown in Figure 38.

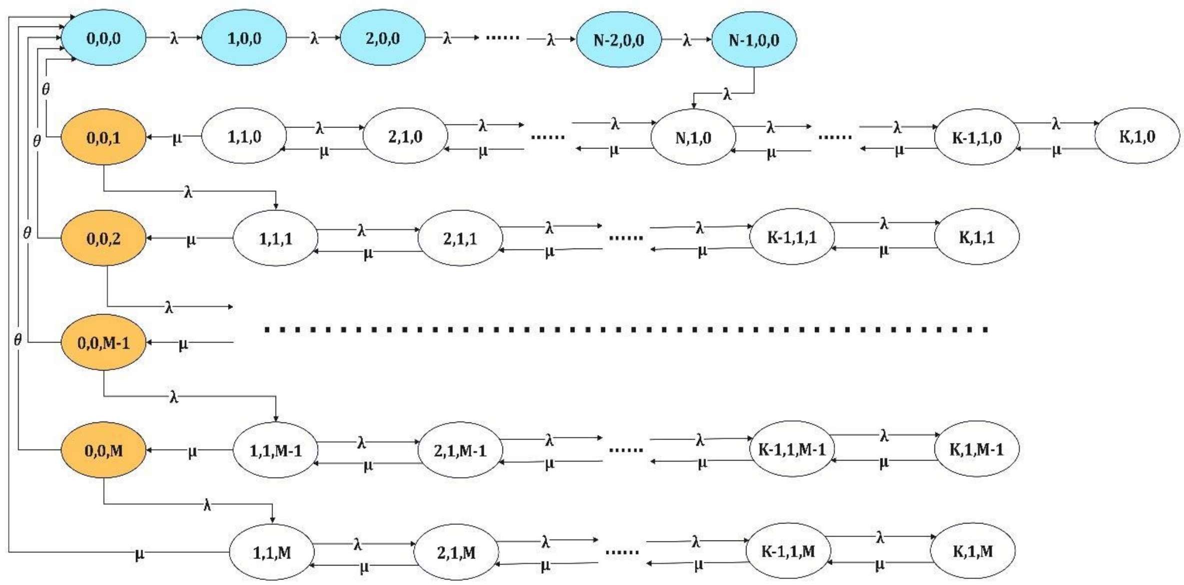

Figure 38:The CTMC corresponding to the GSPN of model M3.

3.2.3.3 Construction algorithm of the Infinitesimal generator

After the analysis of the CTMC presented in figure 38, the number of states of the CTMC which corresponds to the number of tangible markings is equal to K*(M+1)+M+N states.

The infinitesimal generator is constructed as follows:

gi[(i,j,c),(x,y,z)] represents the transition rate between the state (i,j,c) and the state (x,y,z).

Table 9:Representation of transition rates between states.

Description	Outgoing State	Transition Rate	Incoming State
0≤i≤N-2	(i, 0, 0)	λ	(i+1, 0, 0)
1≤i≤K-1, 0≤c≤M	(i, 1, c)	λ	(i+1, 1, c)
1≤i≤K-1, 0≤c≤M	(i+1, 1, c)	μ	(i, 1, c)
0≤c≤M-1	(1, 1, c)	μ	(0, 0, c+1)
1≤c≤M	(0, 0, c)	λ	(1, 1, c)
1≤c≤M	(0, 0, c)	θ	(0, 0, 0)
	(1, 1, M)	μ	(0, 0, 0)

These transitions allow us to build the algorithm given below, to compute the values of the different elements of the infinitesimal generator GI.

Algorithm M3: Construction of the infinitesimal generator for Delayed Idle and N-policy policies

```
for (int i=0; i<N; i++) {
if(i>=N-1) GI[(i,0,0)] [(i+1,1,0)] = λ;
else GI[(i,0,0)] [(i+1,0,0)] = λ;
}
for (int i=0; i<=M; i++) {
for (int j=1; j<K; j++) {
GI [(j,1, i)] [(j+1,1, i)] =λ;
GI [(j+1,1, i)] [(j,1, i)] = μ;
}
}
for (int i=1; i<=M; i++) {
GI [(0, 0, i)] [(0,0,0)] = θ;
GI [ (0,0, i)] [(1,1, i)] = λ;
}
for (int i=1; i<M; i++) {
GI [(1, 1, i)] [(0, 0, i+1)] = μ;
}
 GI [(1,1, M)] [(0,0,0)] = μ;
```

K: Buffer Capacity; N: Threshold; M: Max value for the Counter

3.2.4 Model M4 (DI +N-policy no counter with 2 types of traffic)

The sensor node is considered as a finitely buffered vacation queuing system and service differentiation, in which the control and data frames arrive according to the Poisson distribution, with the mean **λ1, λ2, t**he high priority for service is given to the control frames over the data frames, where the service process is distributed exponentially with the mean rate **μ1**, **μ2**.

A description of a generic sensor is made using the GSPN. Subsequently, an analysis of the latter will be made to generate the CTMC of the model, which allows the obtaining of the infinitesimal generator.

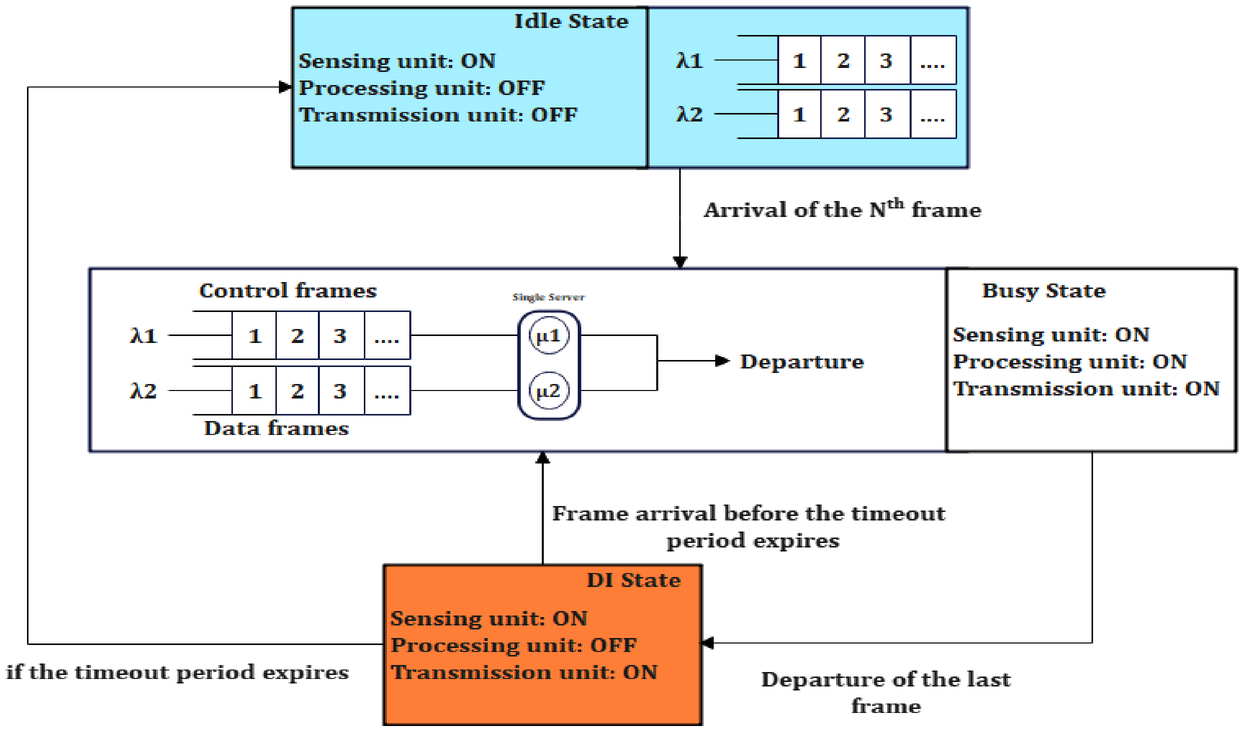

Figure 39:Pictorial representation of the M4.

3.2.4.1 Description of the GSPN

The GSPN modeled in Figure 40 represents the M4 model, which consists of adding these transitions and places to the M2 model:

- The place Buffer1: represents the buffer of the sensor node, containing the control frames.
- The place Buffer2: represents the buffer of the sensor node, containing the data frames.
- The Arrival1 timed transition: represents the arrival of the control frames in the sensor node.
- The transmit1 timed transition: represents the transmission of control frames by the sensor node.
- The Arrival2 timed transition: represents the arrival of the data frames in the sensor node.
- The transmit2 timed transition: represents the transmission of data frames by the sensor node.

The initial marking is represented as follows:

M0={M(capacity), M(Buffer1), M(Buffer2), M(Busy), M(DI), M(Counter), M(Idle)=(K,0,0,0,0,0,1)}

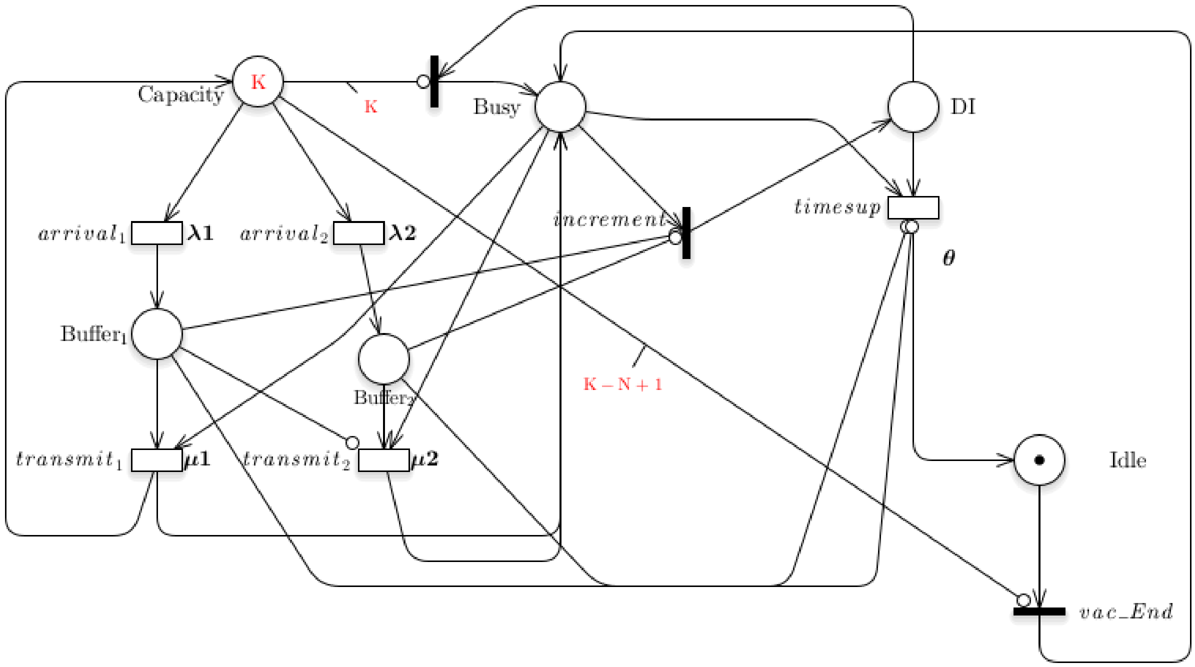

Figure 40:Petri net of M4 model

3.2.4.2 Analysis of the GSPN

The representation of the M4 model follows this system of equations, for whatever value for K.

$$\begin{cases} M(Capacity) + M(Buffer1) + M(Buffer2) = K \\ M(Busy) + M(Idle) + M(DI) = 1 \end{cases}$$

Based on these two equations, the system can be described in this manner:

- i1 represents the number of packets in the buffer1.
- i2 represents the number of packets in the buffer2.
- u, v represents the state of the transmission unit in its different modes: Busy, Idle, Delayed idle.

The marking for the different places can be concluded in this manner:

Table 10:The Marking corresponding to the GSPN of model M4.

M(Capacity) = K-i1-i2	0≤i≤K
M(Buffer1) = i1	
M(Buffer2) = i2	
M(Busy) = u	u+v=1
M(Idle) = 1-u-v	
M(Delayed_Idle) = v	

The CMTC obtained after this analysis is shown in Figure 41:

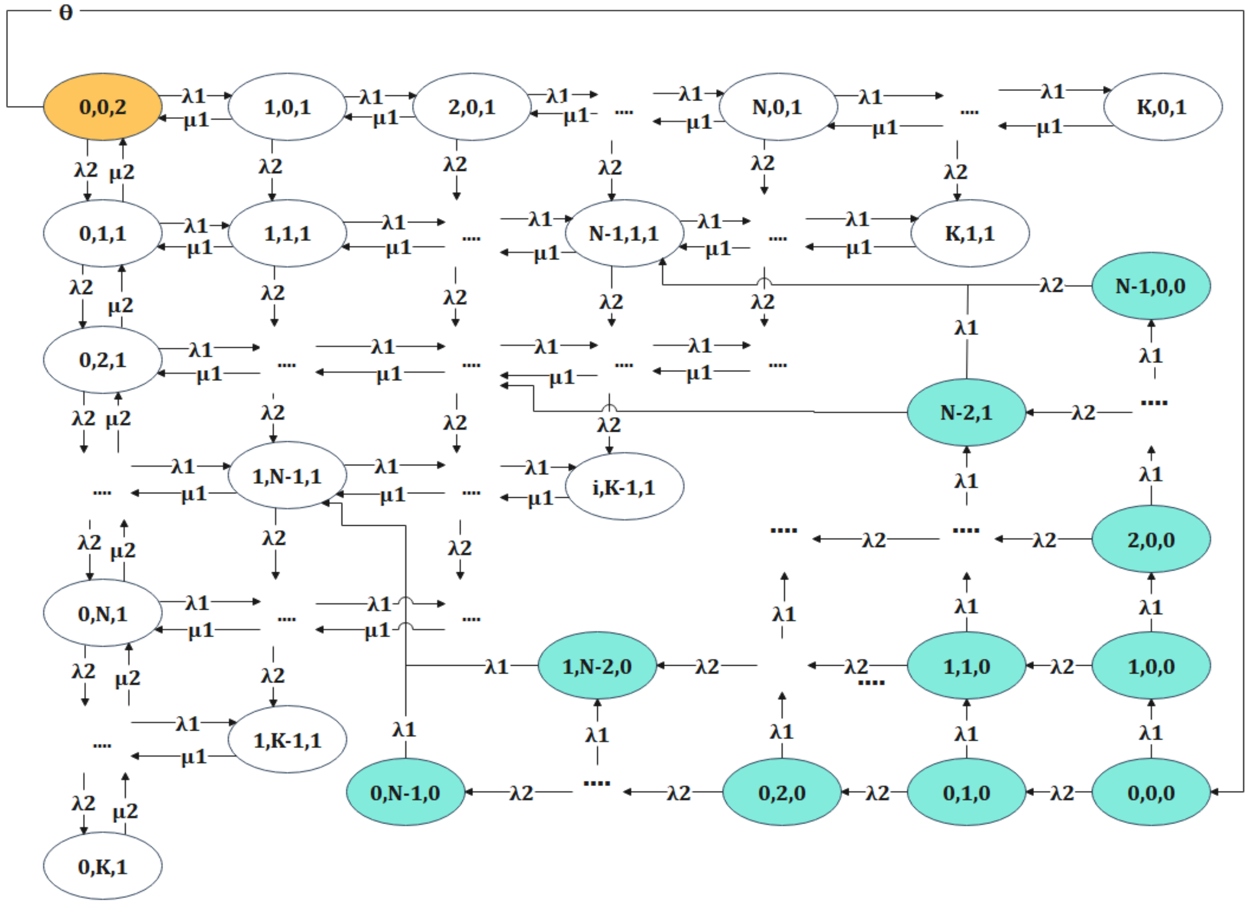

Figure 41:The CTMC corresponding to the GSPN of model M4.

3.2.4.3 Construction algorithm of the Infinitesimal generator

After the analysis of the CTMC presented in Figure 41, the number of states of the CTMC which corresponds to the number of tangible markings is equal to

$\frac{2+(K+1)(K+2)+N(N-1)}{2}$ states.

The infinitesimal generator is constructed as follows:

gi[(i,j,c),(x,y,z)] represents the transition rate between the state (i,j,c) and the state (x,y,z).

Table 11:Representation of transition rates between states.

Description	Outgoing State	Transition Rate	Incoming State
0≤i≤N-2, 0≤j≤N-2-i	(i, j, 0)	λ1	(i+1, j, 0)
0≤i≤N-2, 0≤j≤N-2-i	(i, j, 0)	λ2	(i, j+1, 0)
1≤i≤K-1,1≤k≤K-j-1	(i, j, 1)	λ1	(i+1, j, 1)
1≤i≤K-1,1≤k≤K-j-1	(i, j, 1)	λ2	(i, j+1, 1)
1≤i≤K-1, 1≤j≤K-1-i	(i, j, 1)	µ1	(i-1, j, 1)
1≤j≤K-1	(0, j, 1)	µ2	(0 j-1, 1)

	(1, 0, 1)	μ1	(0, 0,2)
	(0, 1, 1)	μ2	(0, 0,2)
	(0, 0, 2)	λ1	(1, 0, 1)
	(0, 0, 2)	λ2	(0, 1, 1)
	(0, 0,2)	ϴ	(0, 0, 0)

These transitions allow us to build the algorithm given below, to compute the values of the different elements of the infinitesimal generator GI.

Algorithm M4: Construction of the infinitesimal generator for model M4

```
for (int i=0;i<N-1;i++) {
        for (int j=0 ;j<N-1-i;j++) {
                GI[(i, j,0)][(i+1, j, 0)]=λ1;
                GI[(i, j,0)][(i, j+1, 0)]=λ2;
        }
}
for (int i=0 ;i<N;i++) {
        GI[(i, N-i-1, 0)][(i+1, N-i-1, 1)]=λ1;
        GI[(i, N-i-1, 0)][(i, N-i, 1)]=λ2;
}
If(c>0) {
        GI[(0, 0, 2)][(0, 0, 0)]=ϴ;
        GI[(0, 0, 2)][(1, 0, 1)]= λ1;
        GI[(0, 0, 2)][(0, 1, 1)]= λ2;
}
GI[(1, 0, 1)][(0, 0, 2)]=μ1;
GI[(0, 1, 1)][(0, 0, 2)]=μ2;
for (int i=0;i<K;i++) {
        for (int j=0;j<K-i;j++) {
                if(i>0 || j>0) {
                GI[(i, j, 1)][(i+1, j, 1)]= λ1;
                GI[(j, i, 1)][( j, i+1, 1)]= λ2;
                GI[(i+1, j, 1)][(i, j, 1)]=μ1;
                If(j==0) GI[(j, i+1, 1)][(j, i, 1)]= μ2;
        }
        }
}
```

K: Buffer Capacity; N: Threshold; M: Max value for the Counter

3.2.5 Model M5 (DI + N-policy with counter and two type of traffic)

The sensor node is considered as a finitely buffered vacation queuing system and service differentiation, in which the control and data frames arrive according to the Poisson distribution, with the mean **λ1, λ2, t**he high priority for service is given to the control frames over the data frames, where the service process is distributed exponentially with the mean rate **μ1**, **μ2**.

A description of a generic sensor is made using the GSPN. Subsequently, an analysis of the latter will be made to generate the CTMC of the model, which allows the obtaining of the infinitesimal generator.

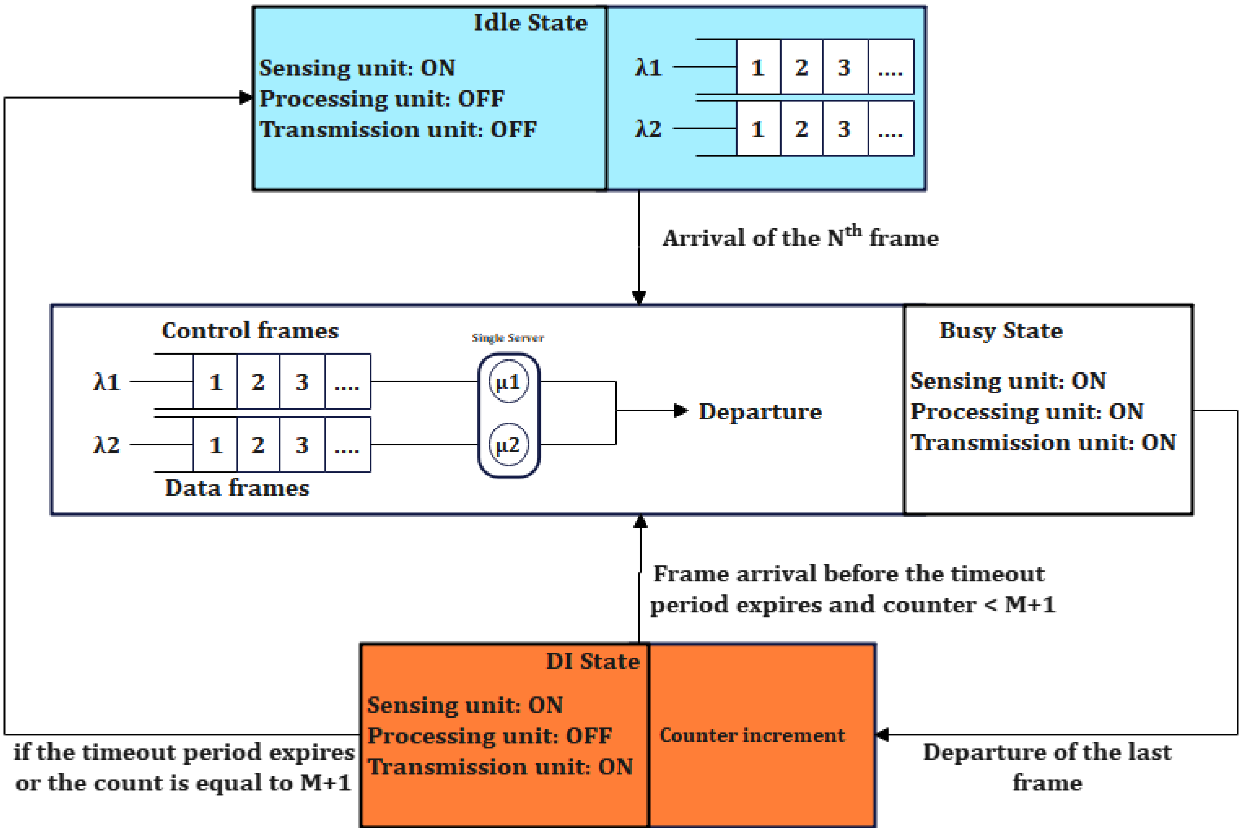

Figure 42:Pictorial representation of the M5.

3.2.5.1 Description of the GSPN

The GSPN modeled in Figure 43 represents the M4 model, which consists of adding these transitions and places to the M4 model:

- The place Counter: represents the number of times the node went to the state Delayed Idle.
- The timesup timed transition: represents the end of the time permitted in delayed idle state after a random time $1/\theta$.
- The di_end transition: represents the passage from delayed idle state to Busy state.

The initial marking is represented as follows:

M0={M(capacity), M(Buffe1), M(Buffe2), M(Busy), M(DI), M(Counter), M(Idle)=(K,0,0,0,0,0,1)}

This marking represents the absence of packets in the sensor node at the initial phase. While the buffer is empty or the number of packets is less than the threshold N, there will be no transmission to be made. Once the number of tokens in the buffer reaches the threshold, the transition vac_End will be fired, which transfers the token to the free state, therefore the start of the transmission with a transition rate μ.

After the transmission of all the packets (no token in the Buffer) the node can pass to delayed idle state and increment the counter by firing the transition increment. This can only for a set number of times (M). depending on which event happen first, either an arrival of a new token so the t0 transition will be fired and going to state Free, or the passing of a random time $1/\theta$ so the token will be fired to state vacation, and counter will be rested to the initial marking.

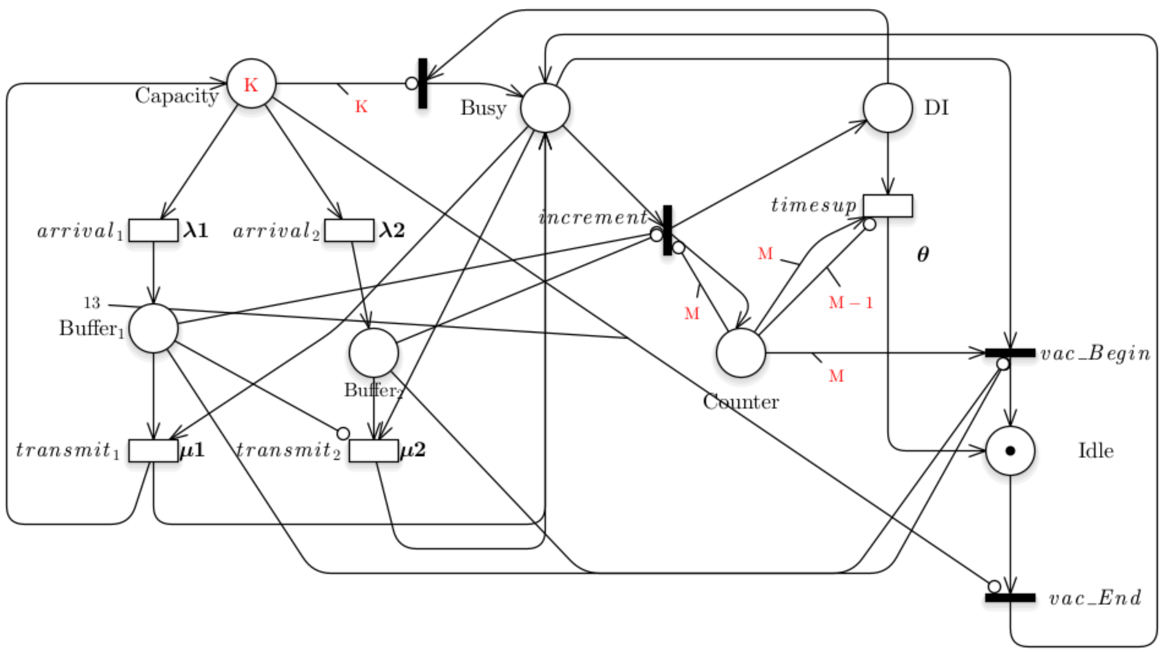

Figure 43:Petri net of M5 model.

3.2.5.2 Analysis of the GSPN

The presentation of the M5 model follows this system of equations, for whatever value for K.

$$\begin{cases} M(Capacity) + M(Buffer1) + M(Buffer2) = K \\ M(Busy) + M(Idle) + M(Delayed\_Idle) = 1 \end{cases}$$

Based on these two equations, the system can be described in this manner:

- i1 represents the number of packets in the buffer1.
- i2 represents the number of packets in the buffer2.
- u, v represents the state of the transmission unit in its different modes: Busy, Idle, Delayed idle.

- c represents the number of times the node went in Delayed_Idle mode.

The marking for the different places can be concluded in this manner:

Table 12:The Marking corresponding to the GSPN of model M5.

M(Capacity) = K-i1-i2	
M(Buffer1) = i1	0≤i≤K
M(Buffer2) = i2	
M(Busy) = u	
M(Idle) = 1-u-v	u+v=1
M(Delayed_Idle) = v	
M(Counter)=c	0≤c≤M

The CMTC obtained after this analysis is shown in Figure 44.

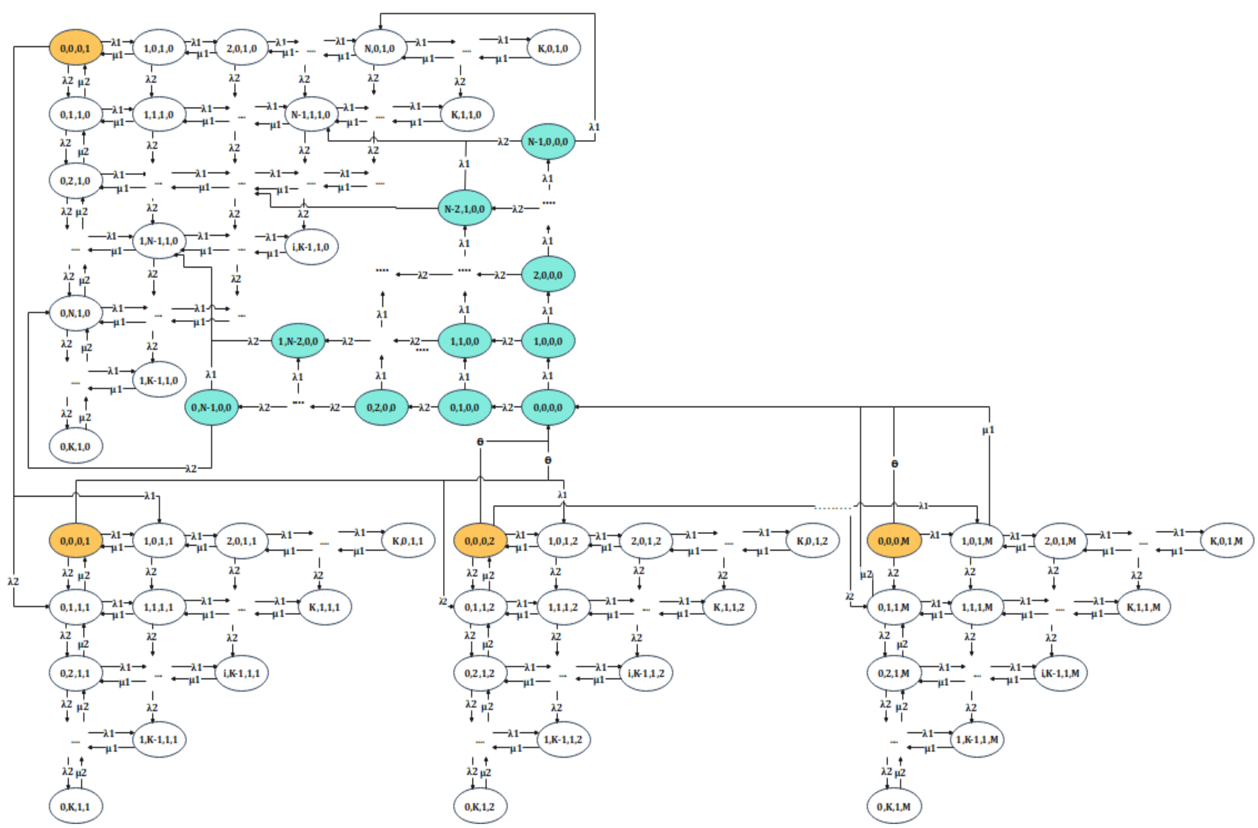

Figure 44:The CTMC corresponding to the GSPN of model M5.

3.2.5.3 Construction algorithm of the Infinitesimal generator

After the analysis of the CTMC presented in figure 44, the number of states of the CTMC which corresponds to the number of tangible markings, is equal to $\frac{2(M+1)+(K+1)(K+2)(M+1)+N(N-1)}{2}$ states.

The infinitesimal generator is constructed as follows:
gi[(i,j,c),(x,y,z)] represents the transition rate between the state (i,j,c) and the state (x,y,z).

Table 13:Representation of transition rates between states.

Description	Outgoing State	Transition Rate	Incoming State
0≤i≤N-2, 0≤j≤N-2-i	(i, j, 0, 0)	λ1	(i+1, j, 0, 0)
0≤i≤N-2, 0≤j≤N-2-i	(i, j, 0, 0)	λ2	(i, j+1, 0, 0)
1≤i≤K-1,1≤k≤K-j-1, 0≤c≤M	(i, j, 1, c)	λ1	(i+1 ,j , 1, c)
1≤i≤K-1,1≤k≤K-j-1, 0≤c≤M	(i, j, 1, c)	λ2	(i, j+1, 1, c)
1≤i≤K-1, 1≤j≤K-1-i, 0≤c≤M	(i, j, 1, c)	µ1	(i-1, j, 1, c)
1≤j≤K-1, 0≤c≤M	(0, j, 1, c)	µ2	(0j-1, 1, c)
0≤c≤M-1	(1, 0, 1, c)	µ1	(0, 0, 0, c+1)
0≤c≤M-1	(0, 1, 1, c)	µ2	(0, 0, 0, c+1)
1≤c≤M	(0, 0, 0, c)	λ1	(1, 0, 1, c)
1≤c≤M	(0, 0, 0, c)	λ2	(0, 1, 1, c)
1≤c≤M	(0, 0, 0, c)	ϴ	(0, 0, 0)
	(1, 0, 1, M)	µ1	(0, 0, 0)
	(0, 1, 1, M)	µ2	(0, 0, 0)

These transitions allow us to build the algorithm given below, to compute the values of the different elements of the infinitesimal generator GI.

Algorithm M5: Construction of the infinitesimal generator for model M5

```
for (int i=0;i<N-1;i++) {
        for (int j=0 ;j<N-1-i;j++) {
                GI[(i, j,0, 0)][(i+1, j, 0, 0)]=λ1;
                GI[(i, j,0, 0)][(i, j+1, 0, 0)]=λ2;
        }
}
for (int i=0 ;i<N;i++) {
        GI[(i, N-i-1, 0, 0)][(i+1, N-i-1, 1, 0)]=λ1;
        GI[(i, N-i-1, 0, 0)][(i, N-i, 1, 0)]=λ2;
}
for (int c=0;c<M+1;c++) {
        If(c>0) {
                GI[(0, 0, 0, c)][(0, 0, 0, 0)]=ϴ;
                GI[(0, 0, 0, c)][(1, 0, 1, c)]=λ1;
                GI[(0, 0, 0, c)][(0, 1, 1, c)]=λ2;
        }
        GI[(1, 0, 1, c)][(0, 0, 0, (c+1)%(M+1))]=µ1;
        GI[(0, 1, 1, c)][(0, 0, 0, (c+1)%(M+1))]=µ2;
        for (int i=0;i<K;i++) {
        for (int j=0;j<K-i;j++) {
        if(i>0 || j>0) {
                        GI[(i, j, 1, c)][(i+1, j, 1, c]=λ1;
```

```
                    GI[(j, i, 1, c)][( j, i+1, 1, c]= λ2;
                    GI[(i+1, j, 1, c)][(i, j, 1, c]=μ1;
                    If(j==0) GI[(j, i+1, 1, c)][(j, i, 1, c]= μ2;
            }
            }
            }
    }
```

K: Buffer Capacity; N: Threshold; M: Max value for the Counter

3.2.6 Model M6(DI + N-policy with acknowledgment for data frames)

The sensor node is considered as a finite buffered vacation queuing system and service differentiation, in which the control and data frames arrive according to the Poisson distribution, with the mean **$\lambda 1$, $\lambda 2$,** the highest priority is given to the control traffic due to them being necessary for managing the operation of the IEEE 802.15.4 network, this type of frames are broadcasted so they don't require an acknowledgment, their service process is distributed exponentially with the mean rate **$\mu 1$**, after synchronizing the information between the pan coordinator and the other devices in the network, the node starts processing the data frames with the mean rate **$\mu 2$**, and require an ACK after receiving, if not it will resend them after the expiration of the send window.

A description of a generic sensor is made using the GSPN. Subsequently, an analysis of the latter will be made to generate the CTMC of the model, which allows the obtaining of the infinitesimal generator.

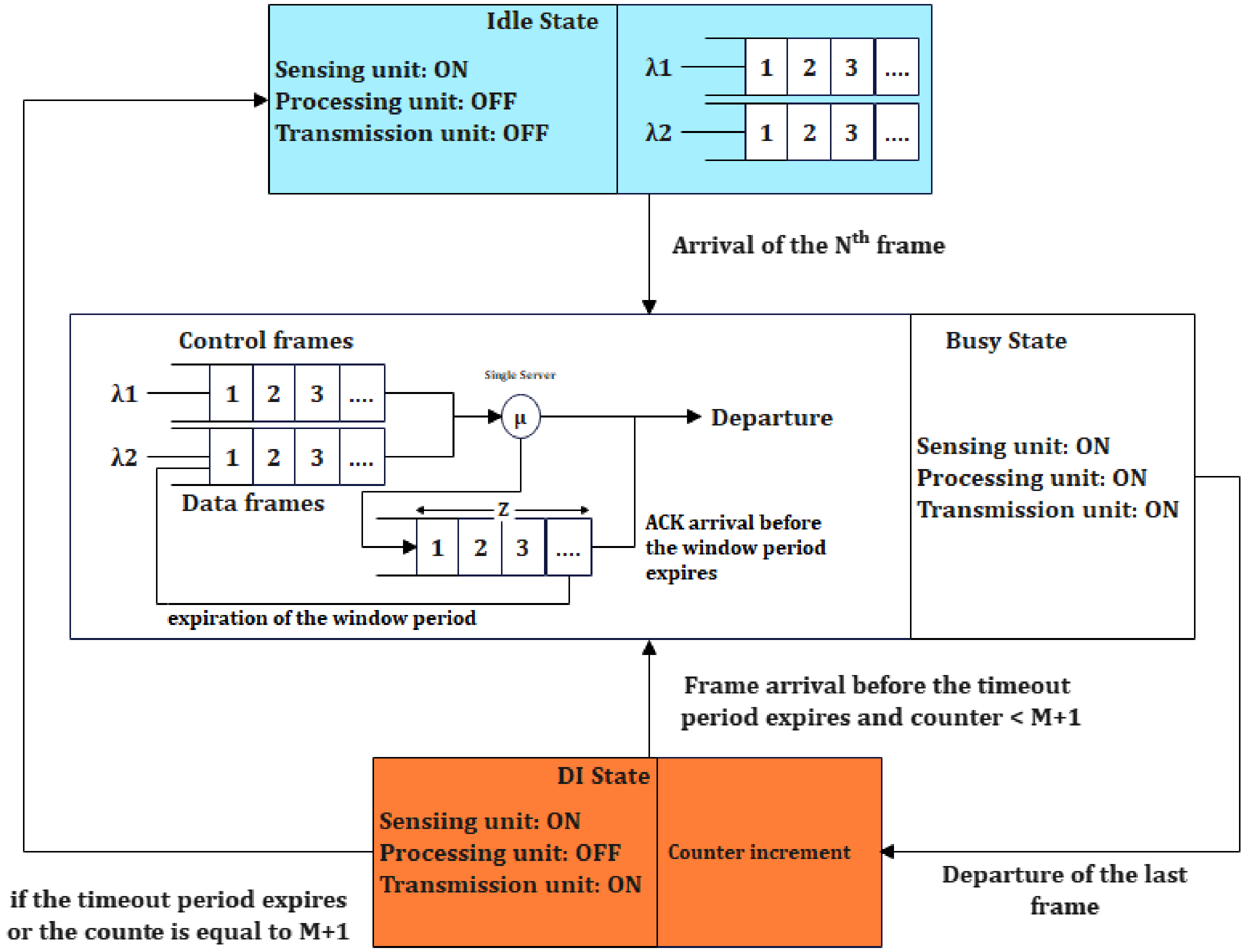

Figure 45:Pictorial representation of the M6.

3.2.6.1 Description of the GSPN

The GSPN modeled in Figure 46 represents the M4 model which consist of adding these transitions and places to the M4 model:

- The place BufferR: contains the data frames that have been sent and waiting for the acknowledgment.
- The Acknowledge timed transition: represents the arrival of an acknowledgment.
- The Nacknowledge transition: represents the expiration of the ACK waiting window permitted before resending the frames.

The initial marking is represented as follows:

M0={M(capacity), M(Buffe1), M(Buffe2), M(BufferR), M(Busy), M(DI), M(Counter), M(Idle)=(K,0,0,0,0,0,0,1)}

This marking represents the absence of packets in the sensor node at the initial phase. While the buffer is empty or the number of packets is less than the threshold N, there will be no transmission to be made. Once the number of tokens in the buffer reaches the threshold, the transition vac_End will be fired, which transfers the token to the free state, therefore the start of the transmission with a transition rate μ.

After the transmission of all the packets (no token in the Buffer) the node can pass to delayed idle state and increment the counter by firing the transition increment. This can only be for a set number of times (M). depending on which event happen first, either an arrival of a new token so the t0 transition will be fired and going to state Free, or the passing of a random time $1/\theta$ so the token will be fired to state vacation, and counter will be rested to the initial marking.

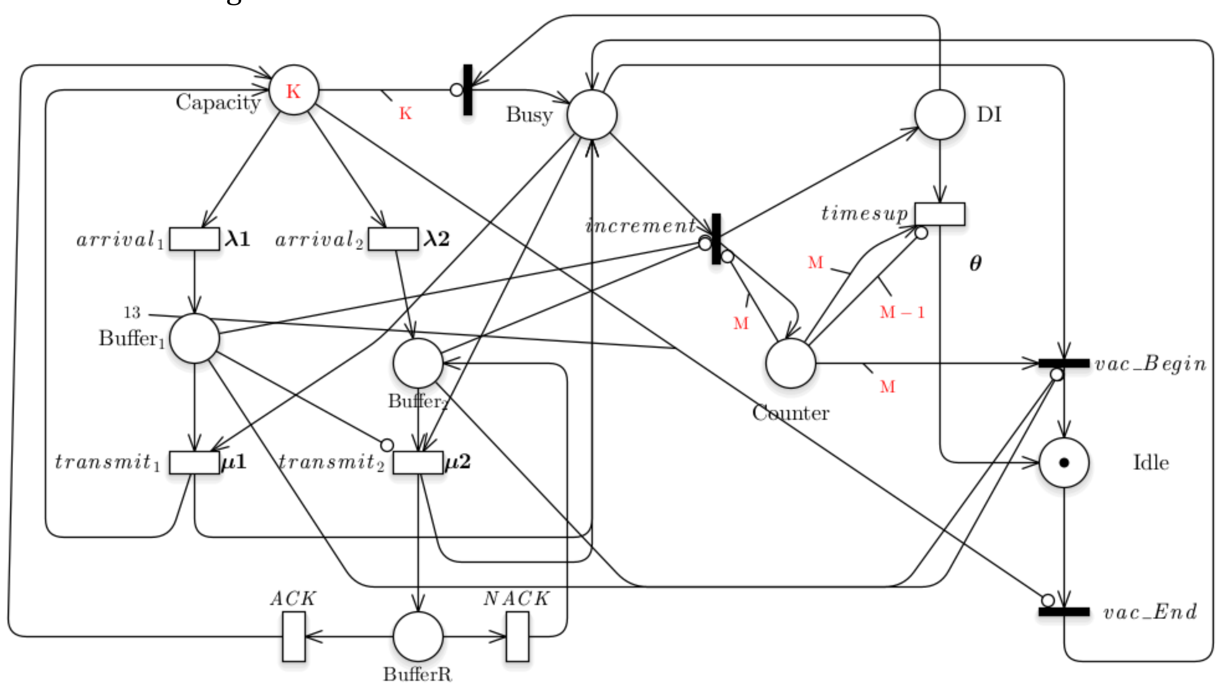

Figure 46:Petri net of M6 model.

3.2.6.2 Analysis of the GSPN

The representation of the M4 model follows this system of equations for whatever value for K.

$$\begin{cases} M(Capacity) + M(Buffer1) + M(Buffer2) + M(BufferR) = K \\ M(Busy) + M(Idle) + M(Delayed\_Idle) = 1 \end{cases}$$

Based on these two equations, the system can be described in this manner:

- i1 represents the number of packets in the buffer1.
- i2 represents the number of packets in the buffer2.
- i3 represents the number of packets in the bufferR.

- u, v represents the state of the transmission unit in its different modes: Busy, Idle, Delayed idle.
- c represents the number of times the node went in Delayed_Idle mode.

The marking for the different places can be concluded in this manner:

Table 14:The Marking corresponding to the GSPN of model M6.

M(Capacity) = K-i1-i2	0≤i≤K
M(Buffer1) = i1	
M(Buffer2) = i2	
M(BufferR) = i3	
M(Busy) = u	u+v=1
M(Idle) = 1-u-v	
M(Delayed_Idle) = v	
M(Counter)=c	0≤c≤M

The CMTC obtained after this analysis is shown in Figure 47.

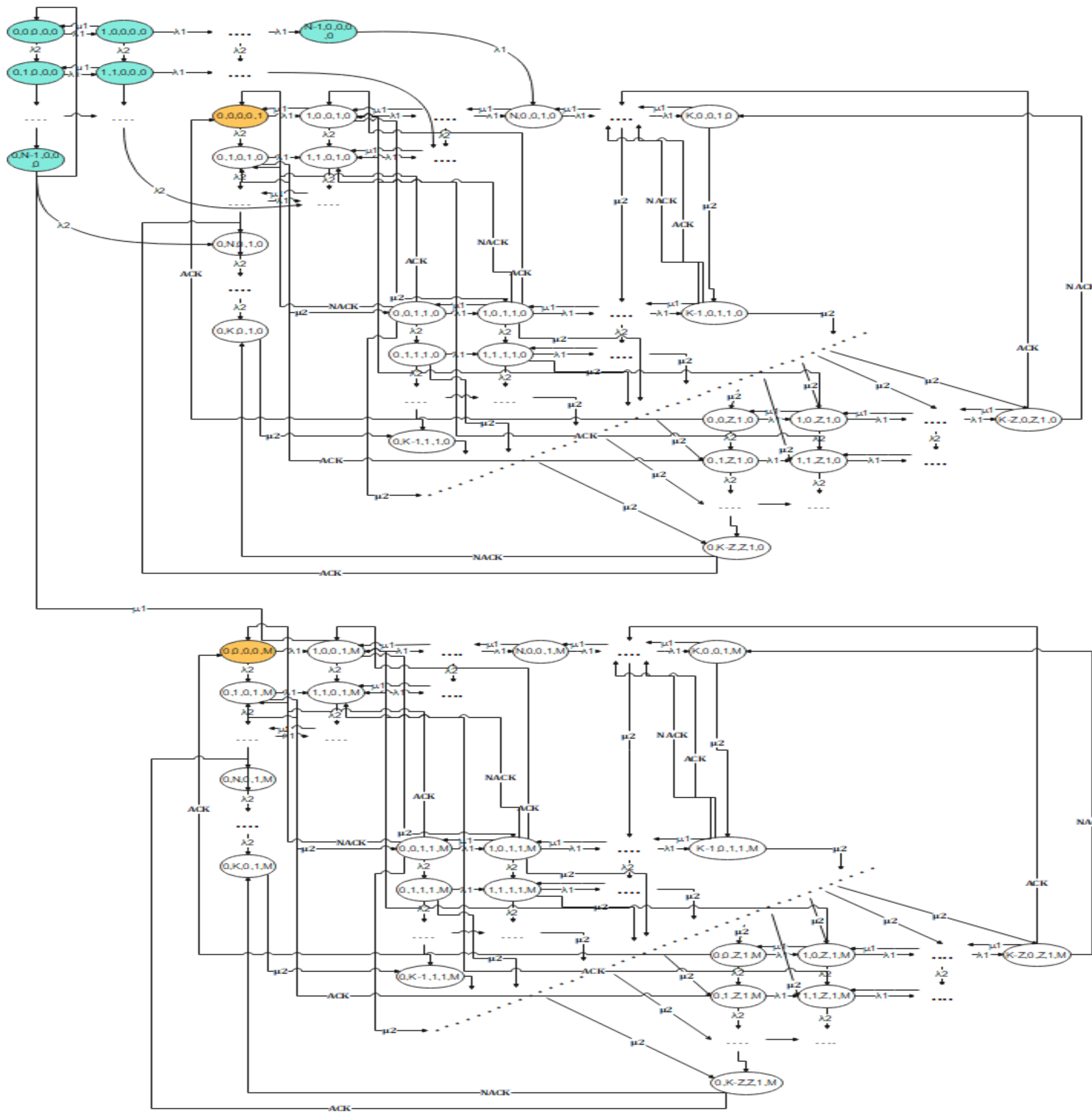

Figure 47:The CTMC corresponding to the GSPN of model M6.

3.2.6.3 Construction algorithm of the Infinitesimal generator

After the analysis of the CTMC presented in figure 47, the number of states of the CTMC which corresponds to the number of tangible markings, is equal to

$$\frac{(Z+1)(K^2+3K+2)+N(N+1)}{2} - \frac{Z(Z+1)+(2Z+1)}{6} \text{states.}$$

The infinitesimal generator is constructed as follows:

gi[(i,j,c),(x,y,z)] represents the transition rate between the state (i,j,c) and the state (x,y,z).

Table 15:Representation of transition rates between states.

Description	Outgoing State	Transition Rate	Incoming State
0≤i≤N-2, 0≤j≤N-2-i	(i, j, 0, 0, 0)	λ1	(i+1, j, 0, 0, 0)
0≤i≤N-2, 0≤j≤N-2-i	(i, j, 0, 0, 0)	λ2	(i, j+1, 0, 0, 0)
1≤i≤K-1, 1≤k≤K-j-1, 0≤c≤M, 0≤t≤Z-j	(i, j, t, 1, c)	λ1	(i+1, j, 0, 1, c)
1≤i≤K-1, 1≤k≤K-j-1, 0≤c≤M, 0≤t≤Z-j	(i, j, t, 1, c)	λ2	(i, j+1, 0, 1, c)
1≤i≤K-1, 1≤j≤K-1-i, 0≤c≤M, 0≤t≤Z-j	(i, j, t, 1, c)	μ1	(i-1, j, t, 1, c)
1≤i≤K-1, 1≤j≤K-1-i, 0≤c≤M, 0≤t≤Z-j	(i, j, t, 1, c)	NACK	(i, j+t, 0, 1, c)
1≤j≤K-1, 0≤c≤M, 0≤t≤Z	(0, j,t, 1, c)	μ2	(0 j-1, t+1, 1, c)
0≤c≤M-1	(1, 0, 0, 1, c)	μ1	(0, 0, 0, 0, c+1)
0≤c≤M-1	(0, 1, 0, 1, c)	μ2	(0, 0, 0, 0, c+1)
0≤c≤M-1, 0≤t≤Z	(0, 0, t, 1, c)	ACK	(0, 0, 0, 0, c+1)
0≤t≤Z	(0, 0, t, 1, M)	ACK	(0, 0, 0, 0, 0)
1≤c≤M	(0, 0, 0, 0, c)	λ1	(1, 0, 0, 1, c)
1≤c≤M	(0, 0, 0, 0, c)	λ2	(0, 1, 0, 1, c)
1≤c≤M	(0, 0, 0, 0, c)	ϴ	(0, 0, 0, 0, 0)
	(1, 0, 0, 1, M)	μ1	(0, 0, 0, 0, 0)
	(0, 1, 0, 1, M)	μ2	(0, 0, 0, 0, 0)

These transitions allow us to build the algorithm given below, to compute the values of the different elements of the infinitesimal generator GI.

Algorithm M6: Construction of the infinitesimal generator for model M5

```
for (int i=0;i<N-1;i++) {
        for (int j=0 ;j<N-1-i;j++) {
                GI[(i, j, 0, 0, 0)][(i+1, j, 0, 0, 0)]=λ1;
                GI[(i, j,0, 0, 0)][(i, j+1, 0, 0, 0)]=λ2;
        }
}
for (int i=0 ;i<N;i++) {
        GI[(i, N-i-1, 0, 0, 0)][(i+1, N-i-1, 0, 1, 0)]=λ1;
        GI[(i, N-i-1, 0, 0, 0)][(i, N-i, 0, 1, 0)]=λ2;
}
for (int t=0;t<Z+1;t++){
        for (int c=0;c<M+1;c++) {
                If(c>0) {
                        GI[(0, 0, t, 0, c)][(0, 0, 0, 0, c+1)]=ACK;
                        GI[(0, 0, 0, 0, c)][(0, 0, 0, 0, 0)]=ϴ;
                        GI[(0, 0, 0, 0, c)][(1, 0, 0, 1, c)]= λ1;
                        GI[(0, 0, 0, 0, c)][(0, 1, 0, 1, c)]= λ2;
                }
                GI[(1, 0, 0, 1, c)][(0, 0, 0, 0, (c+1)%(M+1))]=μ1;
                GI[(0, 1, t, 1, c)][(0, 0, t+1, 0, c)]=μ2;
        for (int i=0;i<K;i++) {
        for (int j=0;j<K-i-t;j++) {
        if(i>0 || j>0 || t>0) {
```

```
                GI[(i, j, t, 1, c)][(i+1, j, t, 1, c]= λ1;
                GI[(j, i, t, 1, c)][( j, i+1, t, 1, c]= λ2;
                GI[(i+1, j, t, 1, c)][(i, j, t, 1, c]=μ1;
                GI[(i, j, t, 1, c)][(i, j, 0, 1, c]= ACK;
                GI[(i, j, t, 1, c)][(i, j+t, 0, 1, c]= NACK;
                If(j==0) GI[(j, i+1, t, 1, c)][(j, i, t+1, 1, c]= μ2;
        }
        }
        }
}
```

K: Buffer Capacity; N: Threshold; M: Max value for the Counter

4 Performance Indices

In order to evaluate the performance of the sensor node modeled according to the different policies previously proposed, the stationary probability vector $\pi_i = (\pi_1, \pi_2, \dots, \pi_n)$ need to be computed, which is the unique solution of equation 7.

$$\begin{cases} \pi . P = \pi \\ \sum \pi_i = 1 \end{cases}$$

The GreatSPN software system is used to validate the correctness (qualitative analysis) of the developed GSPN models.

- The blocking probability of packets (P_{Block}): It corresponds to the buffer saturation probability.
- The probability that a sensor node is in idle state (P_{Idle}): It corresponds to the probability that the place Idle contains one token.
- The probability that a sensor node is in Busy state (P_{Busy}): It corresponds to the probability that the place Busy contains one token.
- The probability that a sensor node is in delayed idle state (P_{DI}): It corresponds to the probability that the place Idle contains one token.
- The average length of an idle period ($\overline{\boldsymbol{I}}$): It corresponds to the average duration of time when the place Idle contains one token, which represents the average time the node goes into idle state.
- The average length of a busy period ($\overline{\boldsymbol{B}}$): It corresponds to the average duration of time when the radio transmits frames.
- The average length of a delayed idle period ($\overline{\boldsymbol{DI}}$): It corresponds to the average duration of time when the place Delayed Idle contains one token, which represents the average time the node goes into delayed idle state.

- The mean number of frames in a sensor node ($\overline{Q}$): It corresponds to the mean number of tokens in the place Buffer.
- The mean number of frames in a sensor node ($\overline{Q1}$): It corresponds to the mean number of tokens in the place Buffer1.
- The mean number of frames in a sensor node ($\boldsymbol{Q2}$): It corresponds to the mean number of tokens in the place Buffer2.
- The frames reception throughput ($\overline{\lambda}$): It corresponds to the effective rate of packet reception by the sensor node.
- The mean waiting delay of packets ($\overline{W}$): It corresponds to the mean time tokens spent in the Buffer.
- The Average duration of a cycle ($\overline{C}$)
- The energy consumption at a sensor node (EC)

  EC_{Busy}: node energy consumption in busy state.
  EC_{Idle}:node energy consumption in idle state.
  EC_{DI}: node energy consumption in delayed idle state.
  $EC_{Holding}$: node energy consumption when the buffer is filled with frames.
  $EC_{Switching}$: node energy consumption when switching between states.

Performance Indices	M0	M1	M2	M3	M4	M5
P_{Block}	$\sum_{i:M(Capacity=K)} \pi_i$					
P_{Idle}	$\sum_{i:M(Idle=1)} \pi_i$					
P_{Busy}	$\sum_{i:M(Busy=1)} \pi_i$					
P_{DI}	-	-	$\sum_{i:M(Delayed\_Idle=1)} \pi_i$			
$\overline{I}$	-	$N/\overline{\lambda}$			$\frac{N(N+1)}{\lambda_1+\lambda_2}$	
$\overline{B}$	$\overline{Q}/\mu$				$\overline{Q1}/\mu1 + \overline{Q2}/\mu2$	
$\overline{DI}$	-	-	$\left(\frac{1}{\theta}+\frac{M}{\lambda}\right)/2$			
$\overline{Q}$	$\sum_{i:M_i \epsilon E} M_i(Buffer).\pi_i$				$\sum_{i:M_i \epsilon E} M_i(Buffer1, Buffer2)\pi_i$	
$\overline{Q1}$	-				$\sum_{i:M_i \epsilon E} M_i(Buffer1)\pi_i$	

$\overline{Q2}$	-	$\sum_{i:M_i \epsilon E} M_i(Buffer2)\pi_i$
$\bar{\lambda}$	$\lambda * \sum_{i:M_i \epsilon E} \pi_i$	$(\lambda 1 + \lambda 2) * \sum_{i:M_i \epsilon E} \pi_i$
$\overline{W}$	$\bar{Q} / \bar{\lambda}$	
$\bar{C}$	$\bar{B} + \bar{I}$	$\bar{B} + \bar{I} + \overline{DI}$
EC	$EC_{Busy} * P_{Busy} + EC_{Idle} * P_{Idle} + EC_{Holding} * \bar{Q} + EC_{Switching} / \bar{C}$	$EC_{Busy} * P_{Busy} + EC_{Idle} * P_{Idle} + EC_{DI} * P_{DI} + EC_{Holding1} * \overline{Q1} + EC_{Holding2} * \overline{Q2} + EC_{Switching} / \bar{C}$

5 Conclusion

This chapter was dedicated to presenting our approach for modeling and evaluating a wireless sensor node from an energy perspective. First, we begin by reviewing some pre-existing models based on vacation queueing for energy consumption optimization in WSN, and then we use GSPNs to model the different vacation policies. Next, we analyze each model to construct its corresponding CTMC and eventually obtain the algorithm for the infinitesimal generator, which we use to compute the stationary probabilities vector. In the next chapter, we use the obtained values to conduct experiments and derive the main performance indices and calculate the impact of the different parameters on the energy consumption of the node.

CHAPTER 4

EXPERIMENTAL RESULTS AND DISCUSSIONS

1 Introduction

In this chapter, we will use the models described in Chapter 4 to carry out the performance comparison of each model while varying different parameters that affect the performance of a sensor node. To evaluate their impact, a series of experiments are conducted using an application that simulates the network behavior and calculates the values of different performance indices, such as energy consumption and latency. Another tool is used to validate the accuracy of the application, called GreatSPN (Graphical Editor and Analyzer for Timed and Stochastic Petri Nets), which is a well-known method for modeling and analyzing complex systems. After that, we will present the results along with the analysis.

2 Experimental study

The results of the experiments obtained from the application are compared with the GreatSPN results. Table 16 summarizes the main findings for different values of N.

Table 16:Values of states probabilities as a function of N, calculated and verified with GreatSPN.

N	State probabilities	M1		M2		M3		M4		M5	
		App	GreatSPN	App	GreatSPN	App	GreaSPN	App	GreatSPN	APP	GreatSPN
	Busy	0.450007	0.450007	0.45	0.45007	0.45	0.45	0.450007	0.450007	0.45	0.45
3	Idle	0.549993	0.549993	0.52381	0.523802	0.523865	0.52	0.523803	0.523803	0.523865	0.523865
	DI	-	-	0.02619	0.02619	0.026135	0.026135	0.02619	0.02619	0.026135	0.026135
	Busy	0.450008	0.450008	0.45	0.450008	0.45	0.449999	0.450006	0.450006	0.45	0.449999
4	Idle	0.549992	0.549992	0.530121	0.530112	0.530163	0.530164	0.530114	0.530114	0.530163	0.530164
	DI	-	-	0.01988	0.019879	0.019837	0.019837	0.019879	0.019874	0.019837	0.019837
	Busy	0.450008	0.450007	0.45	0.450008	0.45	0.449998	0.450006	0.450006	0.45	0.449998
5	Idle	0.549992	0.549993	0.533981	0.533973	0.534015	0.534017	0.533975	0.533975	0.534015	0.534017
	DI	-	-	0.016019	0.016019	0.015985	0.015985	0.016019	0.016019	0.015985	0.015985
	Busy	0.450008	0.450008	0.45	0.450008	0.45	0.449997	0.450006	0.450006	0.45	0.449997
6	Idle	0.549992	0.549992	0.536586	0.536577	0.536615	0.536618	0.536579	0.53657	0.536615	0.536618
	DI	-	-	0.013415	0.13415	0.013386	0.013358	0.013415	0.013415	0.013386	0.013385
	Busy	0.450007	0.450007	0.449999	0.450007	0.449999	0.449996	0.450005	0.450005	0.449999	0.449996
7	Idle	0.549993	0.549993	0.538462	0.538454	0.538487	0.538491	0.538457	0.538457	0.538487	0.538491
	DI	-	-	0.011538	0.011538	0.011513	0.011513	0.011538	0.011538	0.011513	0.011513
	Busy	0.450007	0.450007	0.449998	0.450007	0.449998	0.449994	0.450005	0.450005	0.449998	0.449995
8	Idle	0.549993	0.549993	0.539879	0.53987	0.539901	0.539905	0.539873	0.539873	0.539901	0.539905
	DI	-	-	0.010123	0.010123	0.010101	0.010101	0.010123	0.010123	0.010101	0.010101
	Busy	0.450006	0.450006	0.449997	0.450006	0.449997	0.449992	0.450002	0.450002	0.449997	0.449993
9	Idle	0.549994		0.540987	0.540977	0.541007	0.541011	0.540982	0.540982	0.541007	0.541011
	DI	-	-	0.009016	0.009016	0.008997	0.008997	0.009016	0.009016	0.008997	0.008997
	Busy	0.450003	0.450003	0.449993	0.450003	0.449993	0.449989	0.449998	0.449998	0.449993	0.449989
10	Idle	0.549997	0.549997	0.541879	0.541869	0.541896	0.541901	0.541874	0.541874	0.541896	0.541901
	DI	-	-	0.008128	0.008128	0.00811	0.00811	0.008128	0.008128	0.00811	0.00811
	Busy	0.449997	0.449997	0.449986	0.449998	0.449986	0.449982	0.449991	0449991	0.449986	0.449982
11	Idle	0.550003	0.550003	0.542614	0.542603	0.542631	0.542635	0.54261	0.54261	0.542631	0.542635
	DI	-	-	0.007399	0.007383	0.007383	0.007383	0.007399	0.007399	0.007383	0.007383
	Busy	0.449982	0449982	0.449972	0.449982	0.449972	0.449967	0.449977	0.449977	0.449972	0.449967
12	Idle	0.550018	0.550018	0.543238	0.543228	0.543252	0.543257	0.543233	0.543233	0.543252	0.543257
	DI	-	-	0.00679	0.00679	0.006776	0.006776	0.00679	0.00679	0.006776	0.006775
	Busy	0.449952	0.449952	0.449942	0.449952	0.449942	0.449938	0.449946	0.449946	0.449942	0.449938
13	Idle	0.550048	0.550048	0.543783	543773	0.543797	0.543802	0.54378	0.54378	0.543797	0.543802
	DI	-	-	0.006275	0.006272	0.006261	0.006260	0.006274	0.006274	0.006261	0.00626
	Busy	0.449891	0.449891	0.449881	0.449892	0.449881	0.449876	0.449884	0.449884	0.449881	0.449876
14	Idle	0.550109	0.550109	0.544287	0.544277	0.544300	0.544305	0.544284	0.544284	0.5443	0.544305
	DI	-	-	0.005832	0.005832	0.005819	0.005819	0.005832	0.005832	0.005819	0.005819
	Busy	0.449761	0.449761	0.449753	0.449766	0.449753	0.449748	0.449755	0.449755	0.449753	0.449748
15	Idle	0.550239	0.550239	0.544799	0.544786	0.544811	0.544816	0.544797	0.544797	0.544811	0.544816
	DI	-	-	0.005448	0.005448	0.005436	0.005436	0.005448	0.005448	0.005436	0.005436
	Busy	0.449484	0.449484	0.449485	0.449492	0.449485	0.449480	0.449489	0.449489	0.449485	0.449481
16	Idle	0.550516	0.550516	0.545402	0.545395	0.545413	0.545418	0.545398	0.545398	0.545413	0.545418
	DI	-	-	0.005113	0.005113	0.005102	0.005102	0.005113	0.005113	0.005102	0.005102
	Busy	0.448923	0.448923	0.448922	0.448931	0.448922	0.448917	0.448922	0.448922	0.448922	0.448917
17	Idle	0.551077	0.551077	0.546258	0.546249	0.546269	0.546274	0.546258	0.546258	0.546269	0.546274
	DI	-	-	0.004820	0.004820	0.004809	0.004809	0.00482	0.00482	0.004809	0.004809
	Busy	0.447716	0.447716	0.447731	0.447733	0.447731	0.447726	0.447726	0.447726	0.447731	0.447726
18	Idle	0.552284	0.552284	0.547704	0.547703	0.547714	0.547720	0.54771	0.54771	0.547714	0.54772
	DI	-	-	0.004564	0.004564	0.004554	0.004554	0.004564	0.004564	0.004554	0.004554
	Busy	0.445162	0.445162	0.445200	0.445199	0.445200	0.445195	0.44519	0.44519	0.4452	0.445195
19	Idle	0.554838	0.554838	0.550454	0.550456	0.550464	0.550469	0.550464	0.550464	0.550464	0.550469
	DI	-	-	0.004346	0.004346	0.004336	0.004336	0.004346	0.004346	0.004336	0.004336
	Busy	0.439696	0.439696	0.439763	0.439768	0.439763	0.439758	0.439754	0.439754	0.439763	0.439758
20	Idle	0.560304	0.560304	0.556066	0.556062	0.556076	0.556081	0.556076	0.556076	0.556076	0.556081
	DI	-	-	0.004170	0.004171	0.004161	0.004161	0.004171	0.004171	0.004161	0.004161

The parameters values used as inputs in the analyzed models are summarized in Table 17:

Table 17:Tests parameters.

Parameters	Value
K	Range from 10 to 40
N	Range from 1 to 20
M	3
λ	Range from 0.25 to 2.5
µ	5.0
Ɵ	Range from 0.1 to 15.0
Idle EC	10
Busy EC	500
DI EC	40
Switching Energy	300
Frames Holding Energy	Data frames=5.0 Control frames=1.5

2.1 The arrival rates

The charts in the figures (figure 43, figure 44) represent the different values obtained by measuring the average energy consumption and the average waiting time with different values for the arrival rate.

Parameters values used for test: K=20, N=10, µ=5.0, Ɵ=1.0.

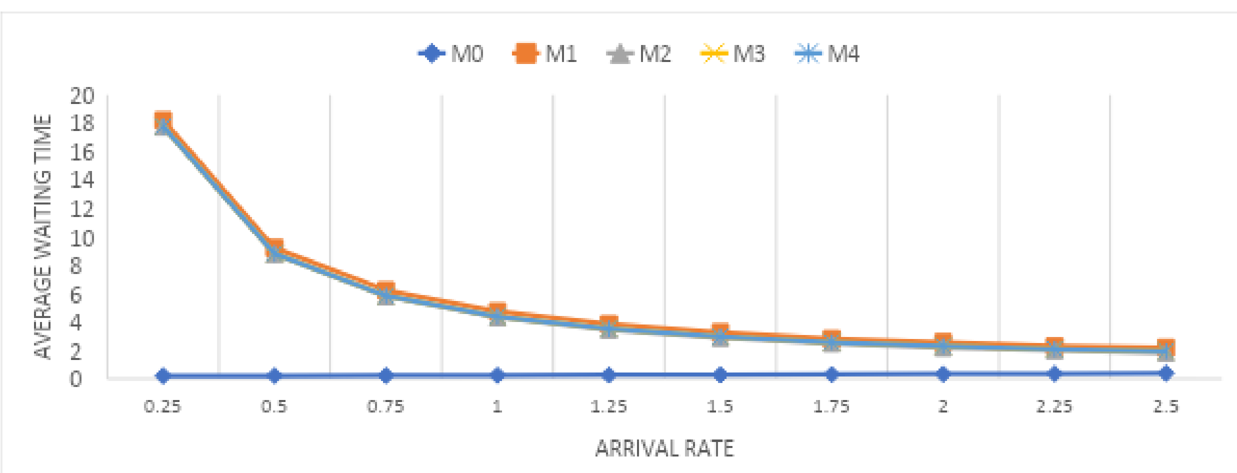

Figure 35:the average waiting time according to the arrival rate.

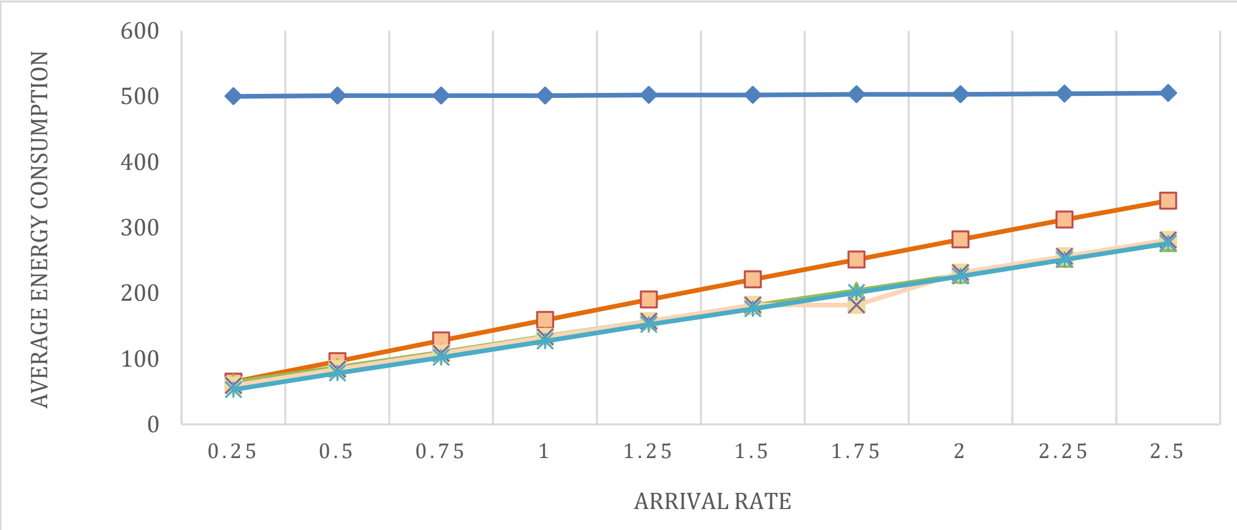

Figure 36: the average energy consumption according to the arrival rate.

Analysis:

The increase of λ influences the average energy consumption and the average waiting time. The energy consumption increases in the M1, M2, M3, and M4 models, while it remains stable in the generic model M0 because the more the arrival rate increases, the more the node consumes its energy resources to process the frames arriving in the system. On the other hand, the average waiting time decreases in the M1...M4 models because increasing the arrival rate increases the frames needing to be processed in the buffer, so the node will spend longer periods in the active state, hence reducing the waiting time.

2.2 Threshold N

The charts in the figures below represent the different values obtained by measuring the average energy consumption and the average waiting time with different values for the threshold N while keeping the other parameters values as follows: K=20, λ=2.25, μ=5.0

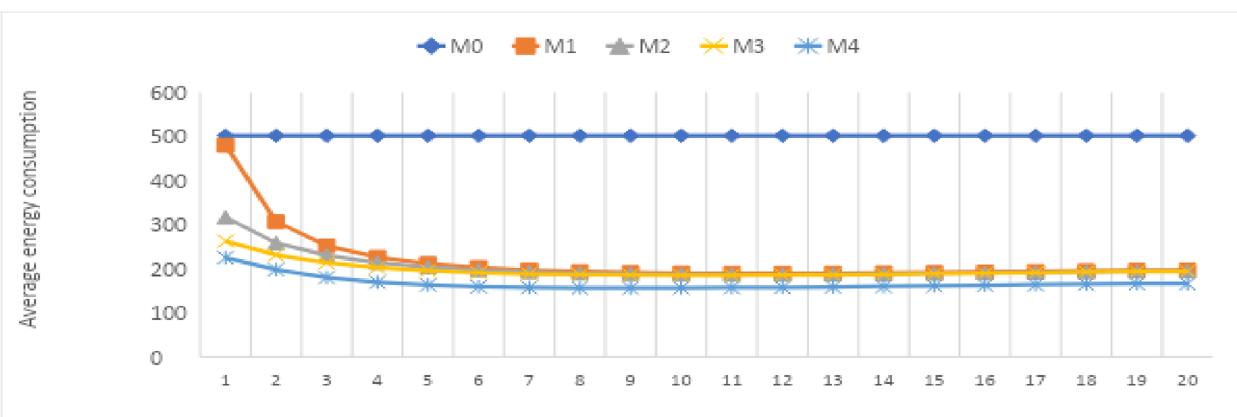

Figure 37:the average energy consumption according to the threshold N (θ=1.0)

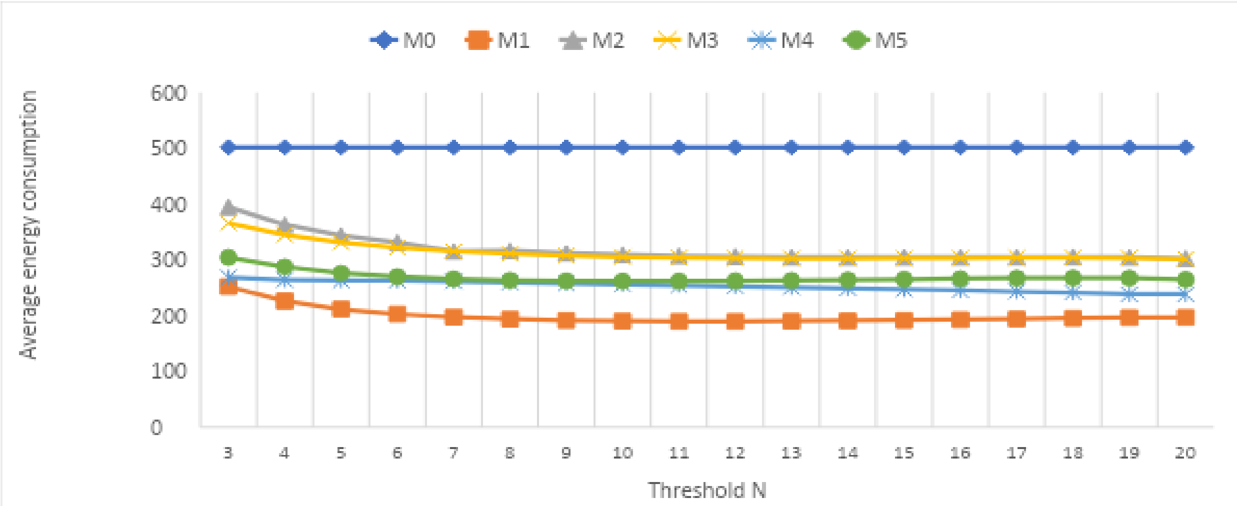

Figure 40:the average energy consumption according to the threshold N (Θ=15.0).

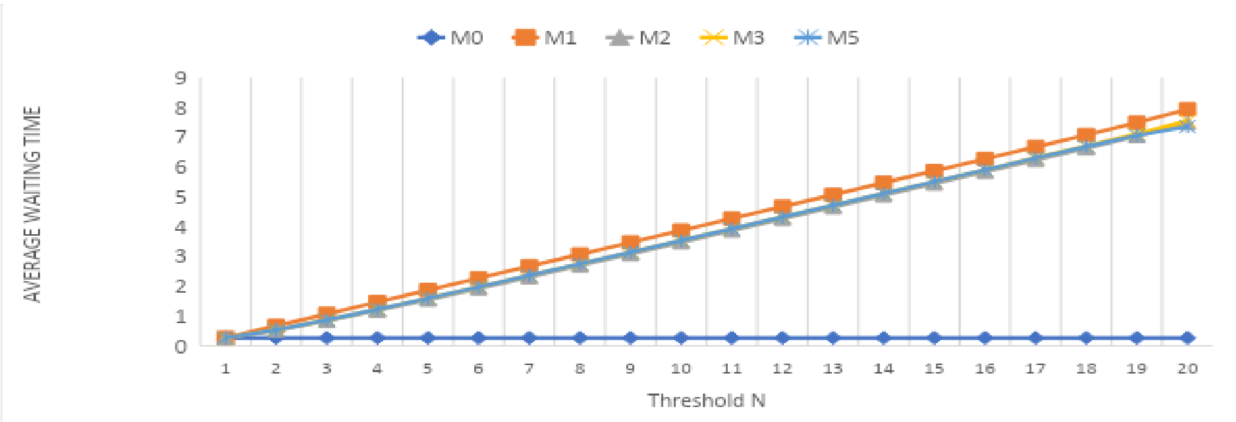

Figure 39:the average waiting time according to the threshold N (Θ=1.0)

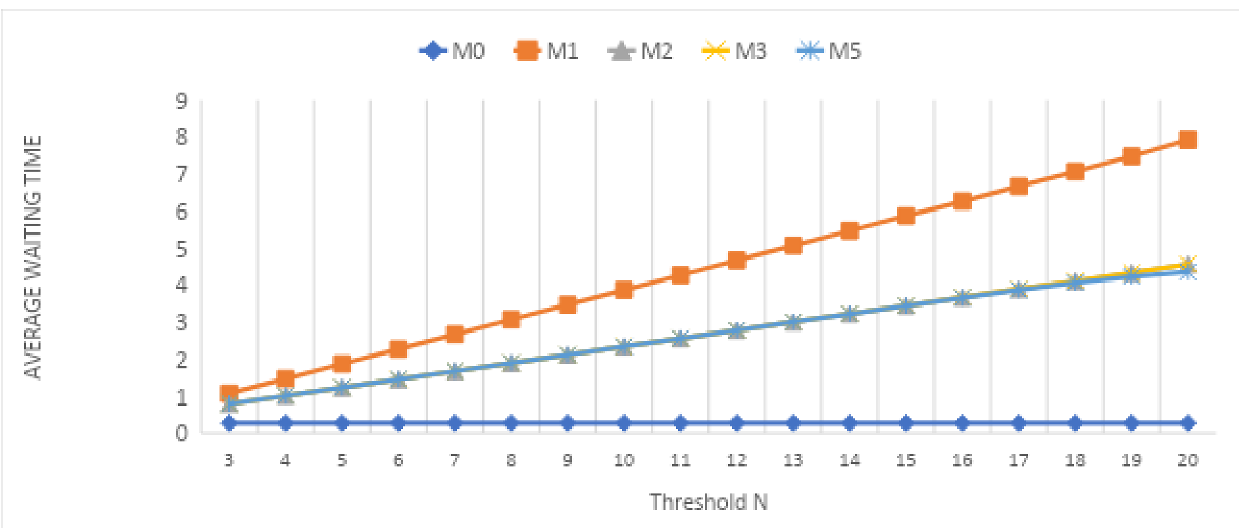

Figure 38:the average waiting time according to the threshold N (Θ=15.0)

Analysis:

The energy consumption decreases, and the average waiting time increases by increasing the threshold N in all the models except the ordinary one because it takes longer duration for buffered frames to reach the threshold N. Thus, the time a node spends in the idle state will be prolonged more; therefore, consumed energy will be reduced.

The consumption of the N-policy model is higher than that of the delayed-idle models, while it remains far better than the ordinary model.

The delayed idle models show better performance in conserving energy while maintaining lower waiting time than the N-policy. This is due to the node going into DI state when the buffer is empty, rather than going into idle state directly and spending energy in switching states and adding the time necessary to reach the threshold to the overall delay each time the node goes back to work.

When λ>ϴ, the DI models with counter have a higher delay than the ones without, as shown in Figure 45, this is due to the counter forcing it to go to the idle state after the count reaches M. This will increase the delay when it has to wait for the buffer to reach the threshold N in each cycle, this phenomenon will affect the energy consumed in a positive manner, where spending more time in idle periods will reduce the energy consumed, and for the No-counter models it will keep looping between the busy and the DI, and won't go to idle due to (λ>ϴ).This highlights the influence of the threshold N on energy consumption and waiting time.

2.3 Buffer Capacity

The charts in the figure below represent the different values obtained after measuring the average energy consumption and the average waiting time while varying the buffer capacity and fixing the other parameters in all models as follows:

N=10, M=3, λ=2.25, ϴ=1.0, μ=5.0.

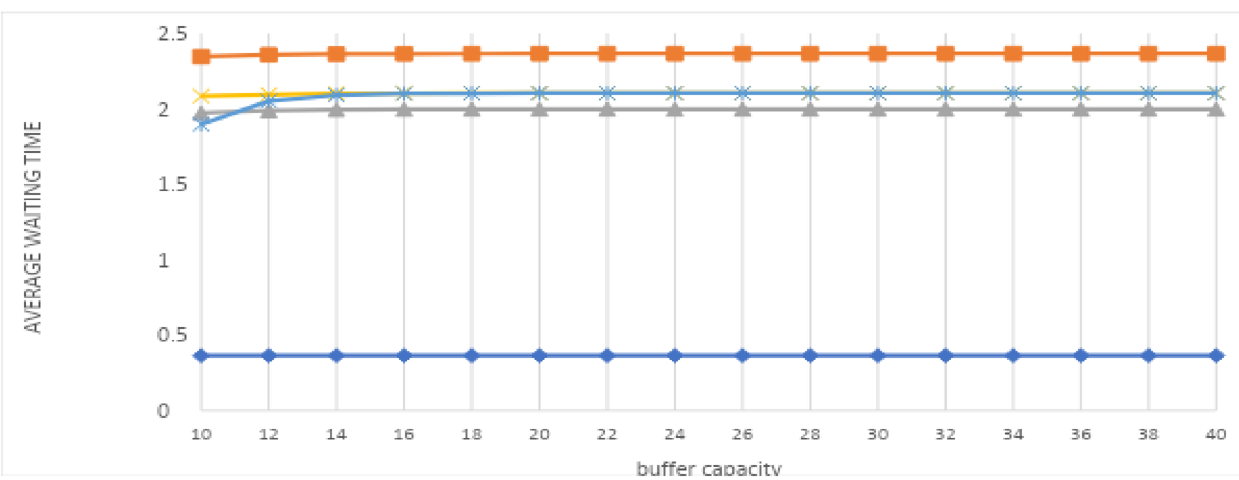

Figure 41:the average waiting time according to the Buffer capacity.

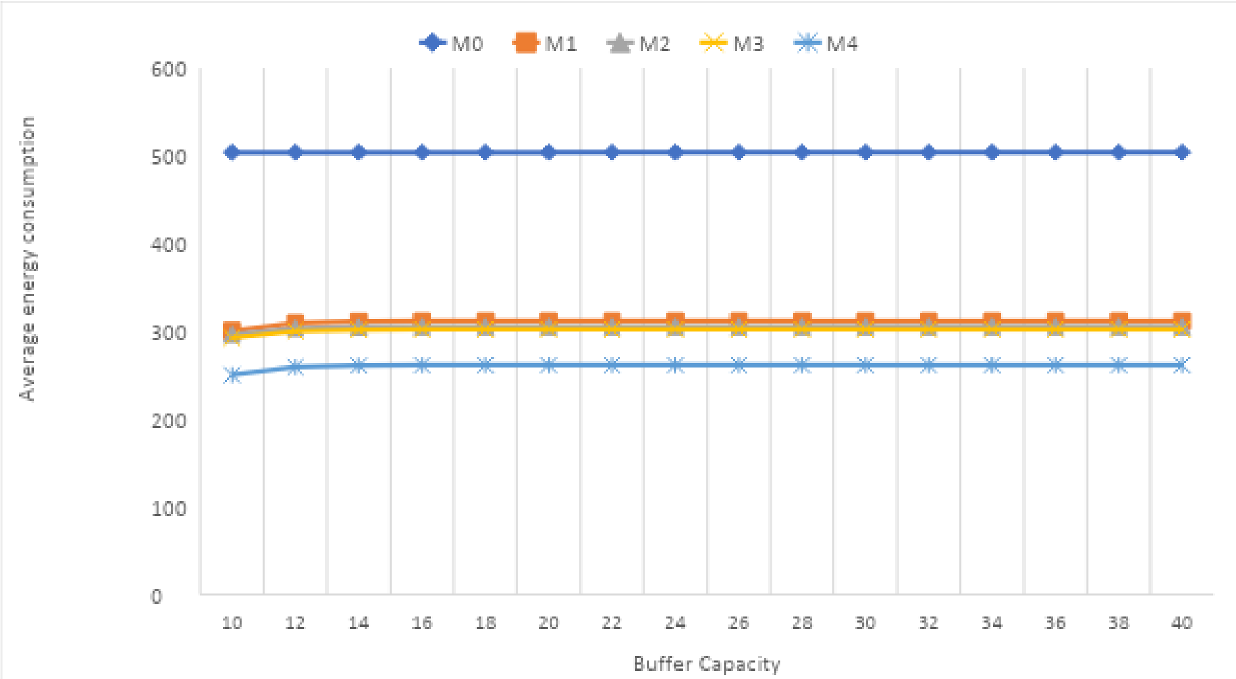

Figure 42:the average energy consumption according to the Buffer capacity.

Analysis:

The energy consumption and the waiting time of frames slightly increase in the four proposed models. At the same time, it remains stable in the generic model because the more the capacity of the buffer increases, the more the number of frames transmitted increases, hence the increase in energy consumption. This increase is only noticeable in the small values of K. It does not affect the delay of the one type traffic models.

2.4 DI rate Θ

The charts in the figure below represent the different values obtained after measuring the average energy consumption and the average waiting time while varying Θ and fixing the other parameters in all models as follows:

K=20, N=10, λ=2.25, µ=5.0, M=3.

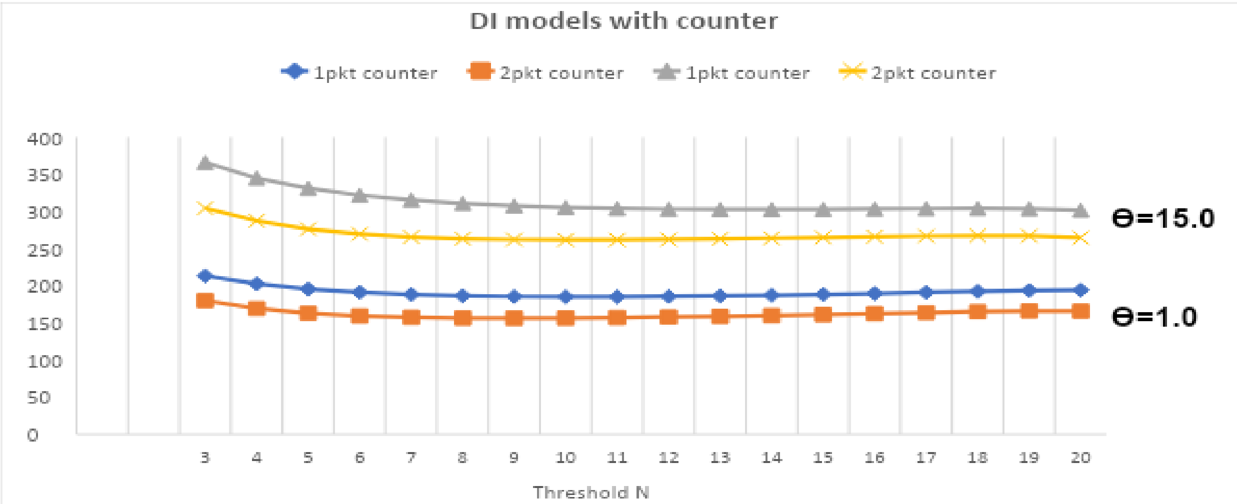

Figure 56:the average energy consumption according to the Buffer capacity.

Analysis

The energy consumption increases, and the average waiting time increases by increasing Ø in all the models because it will spend longer duration in the DI state that has higher energy consumption, and for the delay, it will increase more in the models with no counter, due to the node keep going back to idle and waiting for buffered frames to reach the threshold N, The consumption of the N-policy model is higher than that of the delayed-idle models, while it remains far better than the ordinary model. The delayed idle models show better performance in conserving energy based on the results from the previous test of the N-policy. This highlights the influence of threshold N on energy consumption and waiting time.

2.5 Improvement Factor

The improvement factor in the energy consumption can be expressed as follows:

$$IF = \frac{ECO - ECM}{ECO} * 100$$

ECO: energy consumption of the ordinary model.
ECM: energy consumption of the model under study.

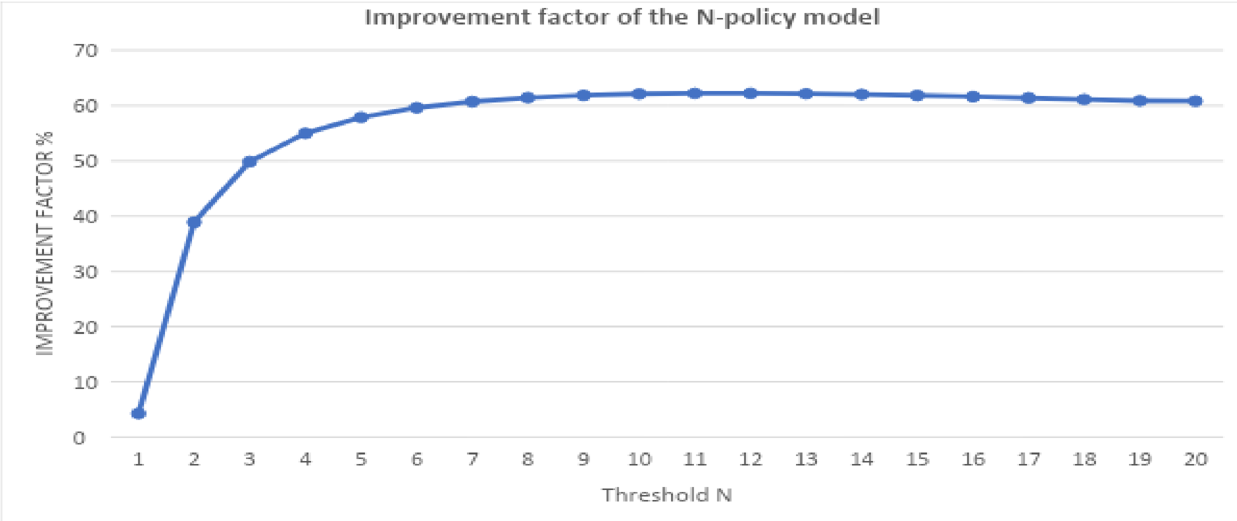

Figure 57:Improvement factor of the N-policy model.

Analysis:

It is noted that the energy gain increases rapidly for the value of N between 1 and 5. Then, this gain reaches its peak for the values between 9 and 14. This is justified by the extension of the idle duration of the sensor when increasing the threshold. Therefore, the average energy consumption will be reduced. Moreover, the number of pending frames increases.

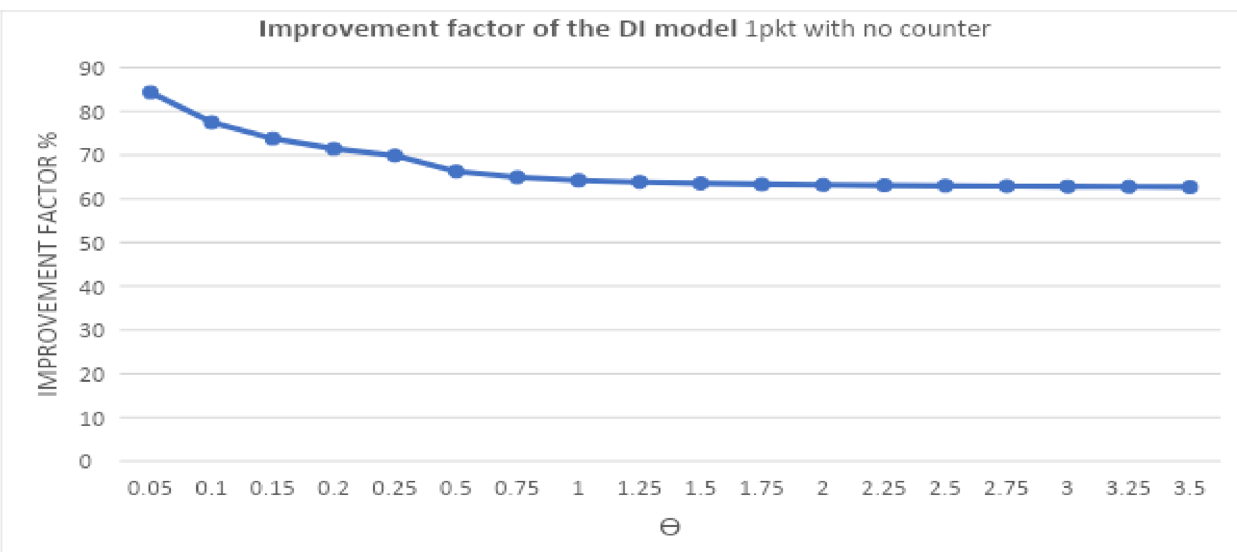

Figure 58:Improvement factor of the DI 1 pkt model with no counter.

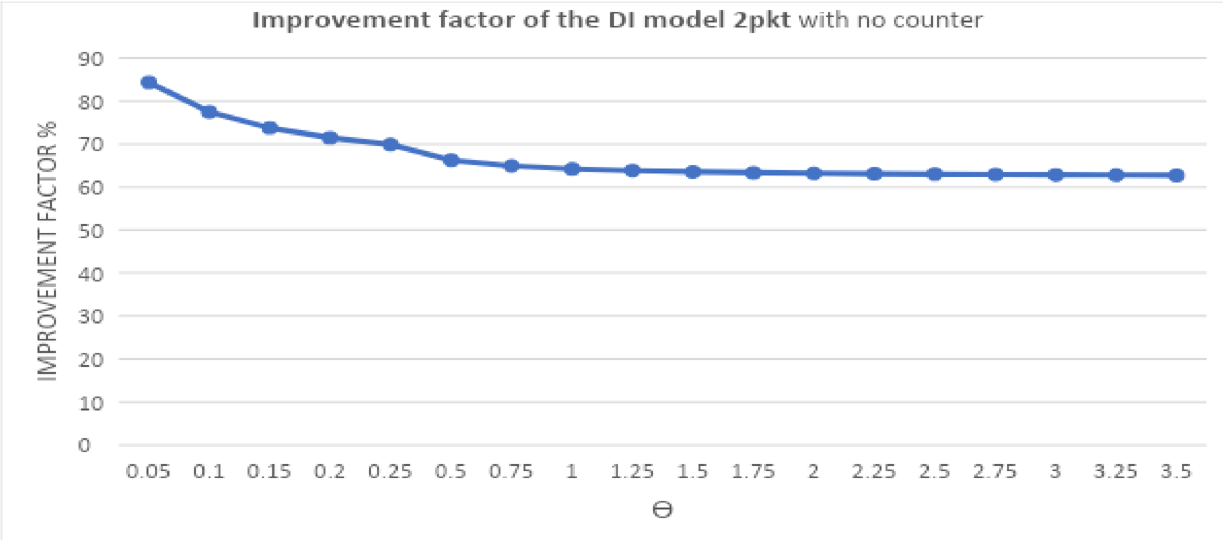

Figure 59:Improvement factor of the DI model 2pkt with no counter

Analysis:

It is noted that the energy gain decreases the more Θ increase, so the node will spend short periods $(\frac{1}{\theta})$ in the DI state and goes into the idle state. These results are logical because spending longer periods in the idle state rather than any other state will definitely show a good result in terms of energy saved, but this positive effect for energy is a negative effect on the delay, as shown in figures 47 and 48, where the node will have to uphold to the N-policy rule once again.

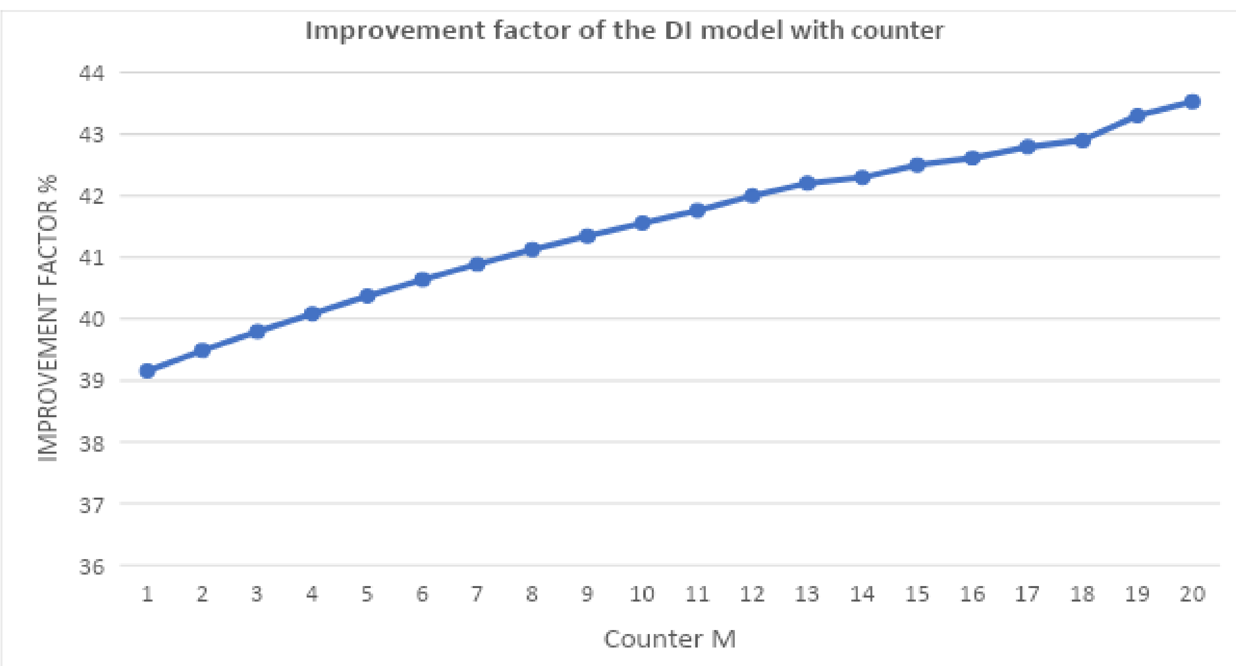

Figure 47:Improvement factor of the DI model with counter.

Analysis:

The increase in M showed great improvement in the energy consumed for the DI models, which directly relates to reducing the switching energy consumed each time a node hopped between states.

3. Conclusion

In this chapter, we have implemented our algorithmic approach to prove the impact of our system parameters on the network's performance with work vacation policies. We deduced from an experimental study the effect of the network parameters on the relevant performance measures to achieve optimized energy consumption and a shorter waiting time. We conclude that the DI policy allows better energy savings for a system that must maintain a minimum delay.

GENERAL CONCLUSION

Wireless sensor networks (WSNs) have undergone significant advancements since their inception. Initially designed for specific applications, such as environmental monitoring and industrial automation, WSNs have become more versatile. They can now be used in various fields, including healthcare, agriculture, and environmental monitoring. Modern WSNs are equipped with advanced sensors and evolved communication protocols, enabling them to collect and transmit large amounts of data over long distances while consuming minimal power.

However, the increasing demand for real-time monitoring and control of physical systems, coupled with the limited battery life of sensor nodes, makes energy optimization a critical aspect of WSNs. Therefore, energy-efficient protocols and algorithms are required to prolong the network's lifetime and ensure reliable data transmission.

This work aims to optimize the energy consumption of wireless sensor nodes using vacation policies. The proposed approach includes using Generalized Stochastic Petri nets (GSPNs) to model a single sensor node. Then an analysis is made to generate the reduced CMTC and the infinitesimal generator, which is used to compute the main performance indices for the system and evaluate the proposed models.

From the evaluation results of the different models, we can testify to the effectiveness of the vacation policy in reducing the power consumption of the sensor node while maintaining good performance.

Finally, as a perspective for this work, it would be interesting to apply our approach in several different fields and propose other formal methods to evaluate the performance of sensor nodes and achieve maximum efficiency.

Bibliography

[1] S. L. Ullo and G. R. Sinha, “Advances in Smart Environment Monitoring Systems Using IoT and Sensors,” Sensors, vol. 20, no. 11. MDPI AG, p. 3113, May 31, 2020. Doi: 10.3390/s20113113.
[2] M. A. Matin and M. M. Islam, “Overview of Wireless Sensor Network,” Wireless Sensor Networks - Technology and Protocols. Intech, Sep. 06, 2012. Doi: 10.5772/49376.
[3] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, “Wireless sensor networks: a survey,” Computer Networks, vol. 38, no. 4. Elsevier BV, pp. 393–422, Mar. 2002. doi: 10.1016/s1389-1286(01)00302-4.
[4] Sharma, D. & Verma, Sandeep & Sharma, K. (2013). Network topologies in wireless sensor networks: A review. Int. J. Electron. Commun. Technol. 4. 93-97.
[5] K. Eghonghon Ukhurebor, I. Odesanya, Soo Tyokighir, R. George Kerry, A. Samson Olayinka, and A. Oluwafemi Bobadoye, “Wireless Sensor Networks: Applications and Challenges,” Wireless Sensor Networks - Design, Deployment and Applications. IntechOpen, Sep. 15, 2021. doi: 10.5772/intechopen.93660.
[6] X. Cui, X. Guo, M. Zhou, J. Wang, S. Qin, and L. Qi, "A discrete whale optimization algorithm for disassembly line balancing with carbon emission constraint," IEEE Robotics and Automation Letters, 2023.
[7] Kingsly, C. S., & Chandran, J. G. C. (2013). Critical Study on Constraints in Wireless Sensor Network Applications. International Journal of Engineering Research & Technology (IJERT), 2(7).
[8] S.-H. Yang, “Hardware Design for WSNs,” Signals and Communication Technology. Springer London, pp. 49–72, Oct. 24, 2013. doi: 10.1007/978-1-4471-5505-8_3.
[9] L. Filipe, F. Fdez-Riverola, N. Costa, and A. Pereira, “Wireless Body Area Networks for Healthcare Applications: Protocol Stack Review,” International Journal of Distributed Sensor Networks, vol. 2015. SAGE Publications, pp. 1–23, 2015. doi: 10.1155/2015/213705.
[10] A. A. Markov (1971). "Extension of the limit theorems of probability theory to a sum of variables connected in a chain". reprinted in Appendix B of: R. Howard. Dynamic Probabilistic Systems, volume 1: Markov Chains. John Wiley and Sons.
[11] Fortier, Paul & Michel, Howard. (2003). Computer Systems Performance Evaluation and Prediction. Computer Systems Performance Evaluation and Prediction.
[12] Boufenneche, Y. (2014). Influence des messages répétés et des vacances sur les Performances et la consommation d'énergie dans les réseaux de capteurs sans fil (Doctoral dissertation).
[13] Gunter Bolch, Stefan Greiner, Hermann de Meer, and Kishor Shridharbhai Trivedi. 2005. Queueing Networks and Markov Chains. Wiley-Interscience, USA.
[14] Charabi, L. (2011). Proposition d'une Approche Numérique pour l'Évaluation des Performances des Systèmes avec Rappel et Serveurs Hétérogènes (Diplôme de Magister).
[15] Becker, Matthias & Szczerbicka, Helena. (1999). PIMiQ: Integration of queuing networks in generalized stochastic Petri nets. Software, IEE Proceedings -. 146. 27 - 32. 10.1049/ip-sen:19990153.
[16] Wang, Jiacun. (2007). Petri Nets for Dynamic Event-Driven System Modeling. Handbook of Dynamic System Modeling.
[17] Murata, T. (1989). Petri nets: Properties, analysis and applications. Proceedings of the IEEE. Institute of Electrical and Electronics Engineers, 77(4), 541–580. doi:10.1109/5.24143.
[18] Séminaire Mathématique de Béjaïa (LaMOS) Volume 12, 2013, pp. 93 – 100.
[19] Nourredine, Oukas & Boulif, Menouar & Badis, Lyes. (2022). A New GSPNs-Model for Sensors in Solar EHWSNs, Considering Seasonal Sunshine Levels and Sleeping Mechanism Based on Channel Polling Schedule. 10.1007/978-3-031-12097-8_16.

[20] H. M. W. Verbeek, M. T. Wynn, W. M. P. van der Aalst and A. H. M. ter Hofstede, "Reduction Rules for Reset/Inhibitor Nets," in Journal of Computer and System Sciences, vol. 76, no. 2, pp. 125-143, March 2010, doi: 10.1016/j.jcss.2009.06.003.
[21] Hassan, M.N. & Stewart, Robert & Murphy, Liam. (2012). Traffic Differentiation and Dynamic Duty Cycle Adaptation in IEEE 802.15.4 Enabled WSN for Real-Time Application. Telecommunication Systems. 62. 10.1007/s11235-015-0074-x.
[22] Gutiérrez, José. (2007). On the Use of IEEE Std. 802.15.4 to Enable Wireless Sensor Networks in Building Automation. IJWIN. 14. 295-301. 10.1007/s10776-007-0063-4.
[23] Ghosh, Sanjeev & Unnikrishnan, Srija. (2017). Reduced power consumption in wireless sensor networks using queue-based approach. 1-5. 10.1109/ICAC3.2017.8318794.
[24] Ghosh, Sanjeev & Unnikrishnan, Srija. (2019). Delay Incurred in Wireless Sensor Networks Using N-Policy M/M/1 Queuing Model and its Mitigation. 8. 40-52. 10.4018/IJECME.2019070103.
[25] Rajagopal, Maheswar & Jayaparvathy, R... (2011). Performance Analysis of Cluster based Sensor Networks Using N-Policy M/G/1 Queueing Model. European Journal of Scientific Research. 58. 2.
[26] Sudhesh, Ramupillai & Shapique, Mohammed. (2022). Transient analysis of power management in wireless sensor networks with start-up times and threshold policy. Telecommunication Systems. 80. 10.1007/s11235-022-00879-1.
[27] D, Nageswari & Rajagopal, Maheswar & Kanagachidambaresan, G... (2019). Performance analysis of cluster based homogeneous sensor network using energy efficient N-policy (EENP) model. Cluster Computing. 22. 10.1007/s10586-017-1603-z.
[28] Boutoumi, Bachira & Gharbi, Nawel. (2018). Two Thresholds Working Vacation Policy for Improving Energy Consumption and Latency in WSNs. 10.1007/978-3-319-93736-6_13.
[29] Jiang, Fuu-Cheng & Huang, Der-Chen & Yang, Chao-Tung & Leu, Fang-Yie. (2012). Lifetime elongation for wireless sensor network using queue-based approaches. The Journal of Supercomputing. 59. 1312-1335. 10.1007/s11227-010-0537-5.
[30] Ke, Jyun-Fan & Chen, Wen-Ju & Huang, Der-Chen. (2015). A Life Extend Approach Based on Priority Queue N Strategy for Wireless Sensor Network. EAI Endorsed Transactions on Cognitive Communications. 1. 10.4108/eai.19-8-2015.2260111.
[31] Timely reliability modeling and evaluation of wireless sensor networks with adaptive N-policy sleep scheduling,Reliability Engineering & System Safety,Volume 235,2023,109270,ISSN 0951-8320.
[32] Boutoumi, Bachira & Gharbi, Nawel. (2017). An energy saving and latency delay efficiency scheme for wireless sensor networks based on GSPNs. 0645-0650. 10.1109/CoDIT.2017.8102667.
[33] Boutoumi, Bachira & Gharbi, Nawel. (2023). N-policy Priority Queueing Model for Energy and Delay Minimization in Wireless Sensor Networks Using Markov Chains. 1-6. 10.1109/ICAECCS56710.2023.10104978.
[34] M. Bourcerie, La Modélisation des Systèmes de Production par Réseaux de Petri,Master Recherche Systèmes Dynamiques et Signaux, ENSTA Bretagne, 2010-2011.
[35] Vergne, A., & Comte, C. (2018). Files d'attente
[36] Fiems, D., Tian, N., & Zhang, Z. G. (2008). Vacation Queueing Models: Theory and Applications. Applied Stochastic Processes, 1726.
[37] Tian, Nai-shuo and Zhe George Zhang. "Vacation Queueing Models Theory and Applications." Operations Research and Management Science (2006).
[38] Srbinovska, Mare et al. "Energy consumption estimation of wireless sensor networks in greenhouse crop production." IEEE EUROCON 2017 -17th International Conference on Smart Technologies (2017): 870-875.