

Research Article

A Cross-Layer Framework for Network Management in Wireless Sensor Networks Using Weighted Cognitive Maps

Amr El-Mougy and Mohamed Ibnkahla

Department of Electrical and Computer Engineering, Queen's University, Kingston, ON, Canada K7L 3N6

Correspondence should be addressed to Mohamed Ibnkahla; ibnkahla@post.queensu.ca

Received 30 September 2012; Accepted 3 January 2013

Academic Editor: Marc St-Hilaire

Copyright © 2013 A. El-Mougy and M. Ibnkahla. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Achieving the end-to-end goals and objectives of Wireless Sensor Networks (WSN) is a highly challenging task. Such objectives include maximizing network lifetime, guaranteeing connectivity and coverage, and maximizing throughput. In addition, some of these goals are in conflict such as network lifetime and throughput. Cross-layer design can be efficient in proposing network management techniques that can consider different network objectives and conflicting constraints. This can be highly valuable in challenging applications where multiple Quality of Service (QoS) requirements may be demanded. In this paper, a novel cross-layer framework for network management is proposed that particularly targets WSN with challenging applications. The proposed framework is designed using the tool known as Weighted Cognitive Map (WCM). The inference properties of WCMs allow the system to consider multiple objectives and constraints while maintaining low complexity. Methods for achieving different objectives using WCMs are illustrated, as well as how system processes can operate coherently to achieve common end-to-end goals. Using extensive computer simulations, the proposed system is evaluated. The results show that it achieves good performance results in metrics of network lifetime, throughput, and Packet Loss Ratio (PLR).

1. Introduction

Wireless Sensor Networks (WSN) are an enabling paradigm for a wide variety of applications. Due to their low cost, flexibility, and ease of deployment, they have already been applied in many fields such as environmental monitoring, food safety, intelligent transportation, and smart grids. In order to unlock the vast potential of WSN, several known challenges have to be addressed, such as limited energy and processing capabilities, scalability, and fault-tolerance [1–4]. These challenges impose restrictions on network management protocols that can be used, thus hindering the possibility of using WSN in some applications, such as those with strict Quality of Service (QoS) requirements.

In addition to traditional challenges and QoS requirements, end-to-end goals of WSN are sometimes in conflict. For example, maximizing network lifetime might require reducing the frequency at which nodes transmit their data, which in turn reduces throughput. In another example, maximizing network lifetime may mean that a large number

of nodes will have to be switched to sleep mode, which impacts connectivity and coverage. Some WSN also require the support of different applications. For example, a WSN for highway safety may periodically transmit information regarding road or traffic conditions. This information usually has low volume and requires low transmission frequency. However, if an accident occurs, the WSN may be required to transmit images or even low-resolution video from the scene of the accident to aid first responders. In another example, a WSN for animal tracking may occasionally need to transmit images of the location of the animal if any dangers are detected (which may be indicated by, e.g., an elevation in the animal's heart rate). WSN protocols have to adapt quickly to such changes in application requirements and guarantee the desired performance levels.

Performance assurance in WSN has been addressed in many ways by the research community. Unfortunately, the research is rather scattered and usually focuses on specific issues. For example, there is significant research on the areas of connectivity and coverage, routing, and congestion control

(see Section 2). However, there is not as much research activity on networks where several conflicting objectives need to be considered. Cross-layer design provides means to consider different issues from multiple layers in order to improve network efficiency. Nevertheless, there are challenges that are typically associated with cross-layer design that need to be addressed. One important challenge that can have significant effects on network performance is adaptation loops, where improving a specific network issue may lead to deteriorating performance in other issues. For example, transmit power adaptation affects levels of interference in the network and thus may have an impact on routing, congestion, energy lost due to collisions, among other factors. In another example, sleep/awake scheduling directly impacts network connectivity and may affect load balancing between nodes (e.g., if a set of nodes are chosen to remain constantly active, they may become depleted, while other nodes remain unused). Furthermore, WSN protocols are required to have low complexity due to the limited capabilities of nodes. In order to consider complex dynamics within a network, and ensure that network elements are operating coherently towards end-to-end goals, new network management protocols are needed.

In this paper, a novel cross-layer framework for network management is proposed for WSN that run challenging applications. The mathematical tool known as Weighted Cognitive Maps (WCMs) is considered as a tool to provide a parameterized representation of conflicting system processes, in order to perform reasoning while considering multiple conflicting goals and constraints. In WCMs, each process, environment variable, or end-to-end goal is simply represented as a concept in the system, and edges of the map connect concepts that are causally related (an overview of WCMs is given in Section 3). The inference properties of WCMs enable conflicting interactions within the network to be represented as simple mathematical operations, requiring only information about causal relationships between processes. Thus, multiple objectives and constraints can be considered with low complexity, avoiding long processing times that may be associated with optimization problems.

To the best knowledge of the authors, WCMs have not been considered in the design of network management systems for WSN. Although WCMs have been considered in [5] to design cognitive nodes, those nodes operated independently and the design did not consider network interactions between nodes. The design proposed in this paper primarily targets the needs of WSN. WCMs are utilized to design a framework for WSN that monitors network interactions and achieves its end-to-end goals. We illustrate how multiple objectives, constraints, system processes, and environment variables can be translated into concepts of a WCM and how protocols can be implemented using their underlying causal relationships. The proposed system is evaluated through extensive computer simulations to illustrate its capabilities.

The remaining sections of this paper are organized as follows; in Section 2, some recent related research efforts are reviewed; Section 3 provides an overview of WCMs and some fundamental design concepts; Section 4 explains the details of our proposed cross-layer framework; Section 5 provides

simulation results that illustrate the capabilities of our proposal; and finally, Section 6 offers concluding remarks.

2. Related Work

Several areas in WSN have been explored by the research community. These areas include coverage and connectivity, routing, topology management, transmit power, and data rate adaptation. For example, the network management protocol proposed in [6] attempted to improve energy-efficiency and network lifetime while considering routing and coverage constraints in a clustered network architecture. An optimization problem was formulated, labeled (OPT-ALL-RCC), which minimized energy consumption and achieved load balancing while guaranteeing coverage and connectivity. OPT-ALL-RCC was shown to be NP-complete, and a heuristic algorithm named TABU-RCC was proposed to achieve a compromise between efficient performance and processing time. Another network management protocol known as Energy-Efficient m -Coverage and n -Connectivity Routing (EECCR) was proposed in [7]. It considered the routing problem under coverage and connectivity constraints. EECCR has two main phases. In the first phase, the network was divided into mutually exclusive scheduling sets, and sets that can guarantee m -coverage were switched on. Then, routing paths were set up to achieve n -connectivity. The second phase in EECCR was the data transmission phase, where the set-up routing paths were utilized to relay data to the sink node. In [8], a framework called Topology-Aware Resource Adaptation (TARA) was proposed with the primary goal of alleviating congestion in WSN. The idea was to activate a larger number of nodes in periods of congestion in order to increase network resources and reduce congestion. Network topology and traffic patterns were considered in order to propose heuristics that can detect congestion, activate the correct number of nodes, and discover alternative routing paths that can relay packets away from congested spots. Another protocol for congestion control called Enhanced Congestion Detection and Avoidance (ECODA) was proposed in [9]. ECODA used two buffer thresholds, Q_{\min} and Q_{\max} , to detect congestion. Once buffer capacity dropped below Q_{\max} , the protocol started to filter packets based on their delay requirements. If buffer capacity dropped below Q_{\min} , most packets were rejected. A dynamic scheduler was also proposed that ensured fairness in packet delivery between nodes that were close to the sink and nodes that were far away from the sink. In addition, source sending rate control was used to mitigate packet dropping due to congestion.

Guaranteeing connectivity and coverage was addressed in [10], where a distributed algorithm was proposed to schedule the activation of nodes in every time slot. The algorithm assumed that distances between adjacent nodes were known and used this information to find a schedule that considered the remaining battery power of nodes in order to maximize network lifetime. In another paper [11], the problem of finding the optimal transmit power to maximize network lifetime while guaranteeing connectivity was studied. A physical layer-oriented QoS constraint was considered, based on the maximum allowable Bit Error Rate (BER) at the end

of any multihop path. However, the analytical framework in [11] was proposed for specific routing and Medium Access Control (MAC) protocols. In addition, it was assumed that all nodes used the same transmit power.

In [12], a protocol for adapting transmit power, data rate, and duty cycle was proposed with the goal of improving energy efficiency and throughput. In this protocol, nodes that observed good channel conditions transmitted at higher data rates and therefore could conclude their transmissions earlier and stay in sleep state for longer periods. Thus, by adapting the duty cycle according to the observed channel conditions, considerable energy savings were achieved. Another protocol named Symphony was proposed in [13] and targeted transmit power and data rate adaptation while considering throughput and energy efficiency. Algorithms were proposed to ensure that frequent changes in transmit power and data rate were avoided and that fairness in accessing the wireless medium was ensured among interfering links. In [14], a protocol named Throughput Plus Fairness Optimization (TPFO) was proposed. In this protocol, an optimization problem was formulated with the goal of maximizing throughput while ensuring that asymmetric channel access is avoided. Given a certain set of data rates used at different nodes, the optimization problem was solved to find the optimum packet lengths and contention window sizes that would maximize throughput and guarantee fairness.

To support QoS in WSN, the Multipath Multi-SPEED (MMSPEED) protocol was proposed in [15] to address delay and reliability metrics. If delay was the required QoS parameter, every node maintained an estimate of the delay, called “progress speed,” needed to transmit to each of its active neighbours. The neighbour with the greatest progress speed was chosen as the next hop. A threshold, *SetSpeed*, was defined. As long as the progress speed at every hop was greater than *SetSpeed*, the end-to-end delay across the network was bounded by *SetSpeed* and the distance between the source and the destination. On the other hand, MMSPEED supported reliability of packet delivery by allowing multiple paths to deliver packets. Intermediate nodes determined how many paths should be used based on an error metric. As the required reliability increased, more paths were utilized to ensure low Packet Loss Ratio (PLR). However, utilizing multiple paths can increase interference in the network and increase the chance of collisions. Alternatively, the Distributed Aggregate Routing (DARA) protocol [16] also supported latency and reliability metrics in a WSN using multiple sinks. Packets with strict delay requirements were delivered using the shortest paths to the sink, while packets with loose delay requirements used longer paths. Reliability was achieved by transmitting multiple copies of the packets to the sink nodes. Packet scheduling was performed at forwarding nodes by prioritizing packets with strict delay requirements. The disadvantage of DARA is that transmitting multiple copies of packets is not energy efficient and creates interference. To address the combined issue of energy efficiency and QoS support, the Optimized Energy-Delay Subnetwork Routing (OEDSR) was proposed in [17]. Routing in OEDSR was based on a metric that considered the available energy, average end-to-end delay, and distance between the

source and the destination. Performance results showed that OEDSR achieves low energy consumption and lower average end-to-end delay. However, OEDSR was only compared to classic protocols such as Ad Hoc On-Demand Distance Vector (AODV) routing and Dynamic Source Routing (DSR), which are outdated and have no capabilities to support QoS.

Therefore, it can be concluded that new research directions that can consider multiple conflicting constraints with low complexity are required to achieve the desired end-to-end goals of WSN. In this paper, WCM was selected to design a cross-layer framework for WSN due to its versatility and wide range of capabilities. Even though there have been limited efforts on using WCMs in wireless networks [5], as mentioned before, there have been extensive analytical efforts on developing the theory of WCMs since its original proposal in [18]. For example, the work in [19] included comprehensive mathematical description as well as construction methods of WCMs. In addition, the use of WCMs in various applications such as precision agriculture, drought management, and modeling business processes was discussed. In another example [20], the analytical and modeling capabilities of WCMs were extended by proposing methods of transforming one cognitive map into another. In [21], the inference capabilities of WCMs were studied, and different inference methods were compared and analyzed. Furthermore, the idea of conditional edge weights, where edges are activated only under certain conditions, was proposed in [22]. It was shown that conditional edge weights can provide an increased degree of flexibility in system design.

3. Fundamental Design Concepts of WCM

To understand how WCMs can be used in the design of wireless networks, this section provides an overview of the main properties of WCMs. Afterwards, we propose some fundamental guidelines for using WCMs in the design of a cross-layer framework for any network.

3.1. Overview of WCM. A WCM (also known as fuzzy cognitive map) is a graphical model used to represent dynamic systems through their underlying causal relationships [18–20]. Each vertex in the WCM is called a concept and represents a particular process or event in the system being modeled. For example, in WSN, a concept in the WCM can represent the processes of transmit power adaptation or routing, or it can represent environment variables such as PLR or Expected Transmission Time (ETT) at a particular node. A concept can also represent end-to-end goals or constraints such as network lifetime or connectivity. Each concept C_i is characterized by a scalar A_i that represents the value of the concept or its activation level in the real system. These characterizing scalars can take values either in the range $[0, 1]$ or $[-1, 1]$. If the allowed interval is $[0, 1]$, then a concept can be inactive ($A_i = 0$), fully active ($A_i = 1$), or partially active. On the other hand, if the allowed interval is $[-1, 1]$, then C_i can be increasing ($0 < A_i \leq 1$) or decreasing ($-1 \leq A_i < 0$). Edges of the WCM connect concepts that are causally related. Edge weights can take on any value from the interval $[-1, 1]$. Negative edge weights

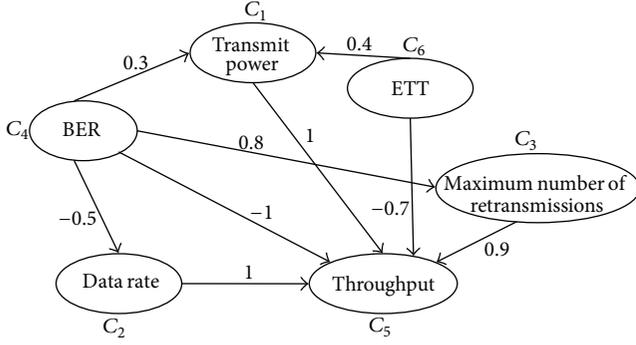


FIGURE 1: WCM representing processes of a wireless node.

imply negative causality and positive edge weights imply positive causality, while zero edge weights imply the absence of a causal relationship between concepts. WCMs can be qualitative or quantitative. Qualitative WCMs only represent causal relationships between concepts, while quantitative WCMs can also represent different levels of granularity in concepts.

One of the main advantages of WCMs lies in their inference capabilities. To illustrate, consider the following WCM representing processes of a wireless node, depicted in Figure 1.

The WCM of Figure 1 models the relationships between 6 processes (concepts) affecting a wireless node, namely, transmit power, data rate, maximum number of retransmissions allowed before a packet is dropped, BER, throughput, and ETT. The edge weights shown represent the strength of causality between concepts. We can classify concepts in any WCM into end-to-end goals, environment variables, and processes. All processes interact to achieve the end-to-end goals. On the other hand, environment variables cannot be manipulated directly, but only as a result of other actions. It can be seen from Figure 1 that the end-to-end goal is throughput (C_5), since all other concepts can cause changes in it. It can also be seen that there are two environment variables considered, namely, BER and ETT, since they cause changes in other processes but cannot be changed directly by them. They can only be affected by changes in the environment or as a result of actions taken by the WCM. The processes available to the WCM are transmit power and data rate adaptation and changing the maximum number of retransmissions. The WCM can be represented in matrix form as

$$W = \begin{matrix} & C_1 & C_2 & C_3 & C_4 & C_5 & C_6 \\ \begin{matrix} C_1 \\ C_2 \\ C_3 \\ C_4 \\ C_5 \\ C_6 \end{matrix} & \begin{pmatrix} 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0.9 & 0 \\ 0.3 & -0.5 & 0.8 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0.4 & 0 & 0 & 0 & -0.7 & 0 \end{pmatrix} \end{matrix}. \quad (1)$$

For a WCM with n concepts, its status at time t can be given by

$$A(t) = [A_1 \ A_2 \ A_3 \ \cdots \ A_n], \quad (2)$$

where A_i is a scalar value representing the activation level of concept C_i . The inference process of WCMs is the one by which the values of concepts change according to their underlying causal relationships. Thus, according to the inference properties of WCMs, the status at time $t + 1$ can be given by

$$A(t+1) = f(A(t)W), \quad (3)$$

where $f(x)$ is a threshold function that determines the type of WCM [21]. The inference process is initialized when a particular concept is triggered, causing its activation value to change, for example, due to a sudden increase in BER. Therefore, the input to the WCM at time t is $A(t)$, including the new value of the concept that was just triggered. The triggered concept influences other concepts according to W , thus producing an output to the WCM, $A(t+1)$, as given by (3). This output provides the new activation values of the concepts, thus indicating the adaptations that need to take place. For example, if an increase in concept C_4 causes an increase in concept C_1 (see Figure 1), this means that the increase in BER requires that the system increases the transmit power.

WCMs have significant advantages over other tools, such as Bayesian and neural networks [23]. They allow for feedback loops, which are not present in Bayesian networks. Concepts in WCMs can also represent events or processes from the real system. This is not available in neural networks, which can be seen as a black box trained to model a particular system and may not faithfully reproduce its characteristics. The simple inference properties also make WCMs attractive for systems that require low complexity, such as WSN. It is also worth noting that WCMs have some disadvantages [23]. They rely on expert knowledge to design the system, which may be challenging, especially in quantitative WCMs. Also, there is no research on how to build a WCM with a centralized view of the network, where multiple WCMs may exist and have to work together to achieve end-to-end goals.

3.2. Design Guidelines for WCM. In order to design a reasoning machine based on WCM, we have to identify the concepts to be considered and the causal links between them. As mentioned in Section 3.1, concepts of the WCM can be classified into end-to-end goals, processes, and environment variables. Thus, the first step is to identify the end-to-end goal(s) of the WCM. These are the concepts that will be constantly monitored by the WCM in order to ensure that they are achieved. Thus, all concepts will interact to achieve these goals. The goals of the WCM must relate directly to the end-to-end goals of the application and the requirements of the network. In addition, it is desirable to design the goals in a way that facilitates monitoring by the WCM. For example, if the network is required to remain operational for a specific period of time, then a suitable goal of the WCM might be energy consumption, which the WCM monitors and takes actions when it is above an unacceptable threshold. On the other hand, if the network is expected to support QoS applications that require the guarantee of certain parameters such as bandwidth, reliability, and delay, these parameters can be directly incorporated as end-to-end goals of the WCM.

Thus, the WCM will monitor these parameters and take actions at appropriate nodes when required. Furthermore, a goal of the WCM can incorporate several parameters simultaneously. For example, a goal can be the ratio of source load to transmission rate, which can be used to avoid congestion.

After determining the goals of the WCM, the next step is to identify the processes that will be used to achieve these goals. Any process can be incorporated in the WCM, depending on the capabilities of the nodes in the network. For example, some of the processes can be transmit power and data rate control, adjusting the size of the contention window, adjusting the sleep/active schedule of devices, or routing. The task of the WCM is to determine when to activate these processes in response to environmental changes. Thus, after determining the processes, the environment variables that trigger the WCM to operate have to be specified. For example, if transmit power and data rate control are considered, a suitable environment variable that would trigger these processes can be PLR, BER, ETT, or others depending on the system and the application requirements. Choosing the processes and environment variables ultimately depends on the issues that the WCM must take into account. Thus, if congestion is a probable event that must be addressed, processes such as adjusting the source loading rate and routing might be incorporated, and the environment variables in this case might be remaining buffer capacity and channel utilization.

After determining all the concepts of the WCM, the next step is to identify the edge weights and causal relationships between concepts. These edge weights depend on the nature of the processes (or the protocols) and design principles utilized. For example, the edge between the concept of PLR and the concept of transmit power control should have a positive weight. This is because an increase in PLR requires an increase in transmit power, and vice versa. Conversely, the edge between PLR and data rate should have a negative edge weight because an increase in PLR requires a decrease in data rate, and vice versa. On the other hand, causal relationships may also be affected by the system design itself. For example, if the system considers both transmit power and data rate control, it may not be desirable to activate both concepts every time there is a change in PLR. To address this issue, the idea of conditional edge weights can be used [22], where only specific edges will be activated depending on the status of the system. For example, if there is an increase in PLR and the WCM has the option of either increasing the transmit power or decreasing data rate, the WCM can make the decision based on the current value of energy consumption and throughput. Thus, if energy consumption is too high, the WCM may opt to decrease data rate to avoid excess drainage in battery power. On the other hand, if throughput is too low, then the WCM may choose to increase transmit power to avoid any impact on the speed of data delivery. This way, multiple objectives and conflicting constraints can be considered by the WCM when executing every action.

An important issue arises in WCM design, which is how to determine the specific weights of the edges. As explained in Section 3.1, a WCM can be qualitative or quantitative. A quantitative WCM (where edge weights may be fractional) may provide an accurate representation of system

interactions but will ultimately require a continuous supervised learning operation to determine the exact value of the edge weights depending on network conditions [19]. This imposes a heavy processing burden on the system, which may not be acceptable in networks with limited capabilities such as WSN. Moreover, the efficiency of the overall system will be highly affected by the accuracy of the learning process. Thus, in this paper, focus will be given to qualitative WCM (where edge weights are either -1 , 0 , or 1), which do not require supervised learning processes. For example, the edge weight between PLR and transmit power would simply be equal to 1 because an increase in PLR causes the system to use the next higher transmit power. The choice of edge weights may also depend on the situations where the system designer specifies that these edges must be activated. For example, the designer may specify that routing should be activated when congestion is detected at a certain node. Thus, the edge weight between the congestion detection concept and the routing concept should be set to 1 . Nevertheless, in order to incorporate some of the advantages of quantitative WCM, some design methods will be proposed in the following sections that illustrate how to give the system some quantitative traits while maintaining the qualitative nature of the WCM.

The final issue to be considered in the design of a WCM framework is implementation. Particularly, distributed versus centralized implementation has to be studied. The advantages and disadvantages of each of these options are well known and will not be repeated here. A hybrid implementation can also be adopted with WCMs, where concepts that require centralized operation are implemented at a central node, while other concepts are distributed over other nodes. In Section 4, we propose a framework for challenging WSN applications based on WCMs, in order to illustrate how the design guidelines that were explained in this section can be implemented. It is important to stress that WCM design is not restricted to the concepts included in the system proposed in Section 4. They are only being used as examples of how WCMs can be used to achieve multiple conflicting goals with low complexity.

4. Designing the Cross-Layer Framework Based on WCM

This section illustrates how a cross-layer framework for WSN can be designed using WCMs. First, the system model is illustrated, and then the design of WCMs to achieve different objectives of the system is explained.

4.1. System Model. The WCM system can be implemented using any WSN architecture. Without loss of generality, and to simplify the concepts to the reader, a clustered hierarchy is adopted in this paper. Nodes where the WCM is implemented are labeled “intelligent nodes,” and they naturally will require higher energy and processing capabilities than regular sensor nodes. However, not all nodes in the network are required to be intelligent. Intelligent nodes can make decisions that are executed by regular nodes.

As shown in Figure 2, the WCM is implemented at the sink node and cluster heads (CH). Each CH will gather

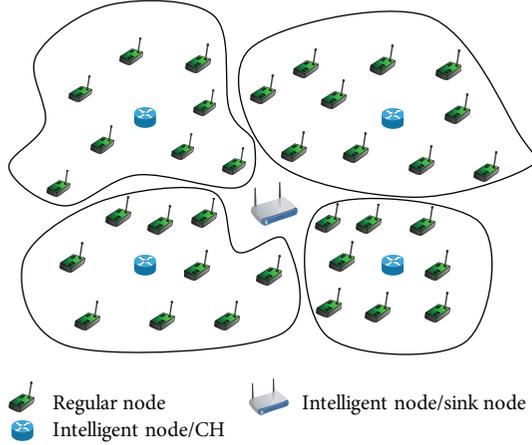


FIGURE 2: Proposed WSN topology.

information from its cluster (such as PLR and residual battery power of nodes) and use the WCM to make decisions about different system parameters, such as transmit power and data rate, which will then be applied by regular nodes. The sink node will be responsible for monitoring WCM concepts that require a centralized view of the network, such as connectivity and coverage. It is also responsible for ensuring that end-to-end goals of the network are achieved. The sink node and CHs are placed in strategic geographic positions in the network to maximize the performance gain from their deployment. The regular sensor nodes can be deployed at random or in predetermined (specific) patterns, depending on the application requirements.

Figure 2 shows regular sensor nodes randomly deployed throughout the area to be covered, with the sink node at the center of the network. There are four clusters with four CHs strategically deployed in positions such that the number of regular sensor nodes managed by each CH is fairly even. It is assumed that nodes are synchronized and that time is slotted. Every time slot has the same duration as one duty cycle. Thus, in one time slot, a node may wake up, sense the medium, transmit one packet, and go back to sleep mode. Moreover, it is assumed that the active portion of the duty cycle is adaptive according to the data rate utilized, similar to the algorithm proposed in [12], which has been previously reviewed in Section 2. If there are redundant nodes in the network, which is probable WSN, not all nodes will wake up in every duty cycle. Thus, scheduling which nodes to wake up takes place every M time slot. This means that, for every M consecutive time slots, the same group of nodes will wake up in every duty cycle. If some nodes fail during this period or need to be switched off for any reason, the WCM will be responsible for making decisions that guarantee connectivity and coverage in every time slot. We also assume that the locations of nodes are known to the WCM. This can be done using GPS devices or any other method for node localization.

In addition, it is assumed that every node transmits using one transmit power from the set $Tx = \{Tx_1, Tx_2, \dots, Tx_L\}$, and one data rate from the set $Rate = \{R_1, R_2, \dots, R_K\}$. Also, every node can choose from a discrete set of duty

cycles, $DC = \{D_1, D_2, \dots, D_K\}$. Furthermore, nodes have limited buffer capacity of B packets. Thus, the set of occupied buffer slots is given by $BF = \{BF_1, BF_2, \dots, BF_B\}$. A simple routing protocol is assumed, where nodes formulate a tree that identifies the shortest path to the associated CH. Thus, multiple hops are allowed within a cluster.

In the following sections, the design of different parts of our WCM will be illustrated. The objective is to design a WSN that considers multiple objectives and conflicting constraints, for example, the energy efficiency versus throughput tradeoff, while maintaining low complexity. Therefore, without loss of generalization, the proposed WCM will include the processes of transmit power, data rate, and duty cycle control; routing; and congestion control. The constraints to be satisfied are connectivity and coverage, and we define an end-to-end goal that considers energy consumption as well as traffic load. The authors emphasize that the proposed WCM is not restricted to the processes or protocols being adopted, but the goal is to illustrate the capabilities of WCMs and provide examples of how they can be designed.

4.2. End-to-End Goal of the WCM. As mentioned before, the first step in designing the WCM is to determine the end-to-end goal. Typical choices of end-to-end goals in WSN include maximizing network lifetime, minimizing energy consumption, and maximizing the amount of data that the network can collect before it reaches its lifetime. In this paper, “Network Efficiency” was chosen as the end-to-end goal of the WCM, which is defined as traffic load divided by energy consumption. The reason behind this choice is to give the network user the flexibility to request higher volumes of data throughout the lifetime of the network. Higher volumes of data will ultimately require higher energy consumption, which has to be considered. Choosing energy consumption only or network lifetime only may cause the network to deny the user higher volumes of data in order to achieve the end-to-end goal. The energy consumption of a node is the amount of battery declination within a given time window (M time slots) divided by its residual battery power. Every node calculates how much the battery declines within the time window and then transmits this value along with its remaining battery power in every packet. Thus, the energy consumption of node i can be given by

$$\text{Eng_consump}_i = \frac{E_{\text{start_window}} - E_{\text{end_window}}}{E_{r_i}}, \quad (4)$$

where E_{r_i} is the residual energy of node i , $E_{\text{start_window}}$ is the remaining energy of node i at the beginning of the time window, and $E_{\text{end_window}}$ is the remaining energy at the end of the window. Equation (4) considers remaining battery power as well as load balancing. In particular, if the residual battery power of a node is low, then its efficiency will decrease, prompting the WCM to take action. On the other hand, if its energy consumption increases, say from being involved in excess traffic, its efficiency will also decrease, again prompting the WCM to take action (specific actions will be explained in later sections). This ensures that nodes will not be overused and will be avoided if their battery power is low.

Traffic load is defined as the amount of traffic dispensed by any node within a given time window (M time slots). Thus, the traffic load of node i can be expressed as

Load _{i}

= Number of packets dispensed within M time slots

/Total number of packets generated at i within M time slots

$$= \frac{P_{\text{Trans}(i)}}{P_{\text{Gen}(i)}}, \quad (5)$$

where $P_{\text{Trans}(i)}$ is the number of packets transmitted at i within M time slots and $P_{\text{Gen}(i)}$ is the number of packets generated at i within M time slots. Thus, if the number of transmitted packets is significantly lower than the number of generated packets, the WCM will be prompted to take action. Therefore, the efficiency of a cluster with number of nodes G is given by

$$\text{Efficiency} = \sum_{i=1}^G \frac{\text{Load}_i}{\text{Eng_consump}_i}. \quad (6)$$

Cluster efficiency, as calculated in (6), is the end-to-end goal that will be considered by the CHs. Since decisions made by one CH can affect the efficiency of another cluster (e.g., routing decisions), the sink node will make sure that the efficiency of all CHs is above the required threshold. Proper actions will be taken if the efficiency of one cluster drops below the threshold, as will be shown in the following sections.

In order to determine the appropriate threshold for efficiency, $\text{Eff}_{\text{Thresh}}$, that would trigger the WCM to take action, application requirements have to be considered. For example, if the initial battery power of nodes is E mAhr and the target lifetime of the network is X time slots, the task of the WCM is to make sure that energy consumption does not exceed E/X mAhr per time slot per node or MGE/X mAhr per M time slots per cluster. On the other hand, the average value of Load _{i} should be close to 1 so that the memory of nodes is not consumed and nodes are not overused in relaying packets for other nodes. Thus, the WCM will be prompted to take action if Load _{i} is less than 0.9 (10% of the nominal value).

4.3. Designing a WCM for Transmit Power, Data Rate, and Duty Cycle Control. Controlling transmit power, data rate, and duty cycle are powerful tools for improving performance. In order to design a WCM that performs these adaptations, the first step is to identify the environment variables, processes, and end-to-end goal that will make up the concepts of the WCM. The second step is to identify the protocols that will define the causality between the considered concepts.

The processes used in this WCM are transmit power, data rate, and duty cycle control, while the end-to-end goal is network efficiency, given by (6), which is the goal of the overall system. The environment variables are PLR and ETT. PLR was chosen because it is affected by channel conditions and interference and can thus give a clear indication about

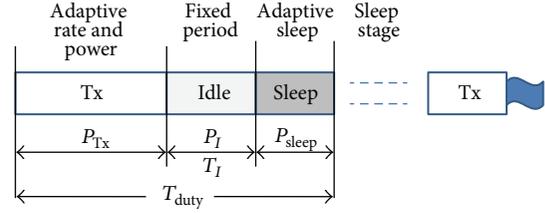


FIGURE 3: Duty cycle with adaptive sleep period.

the quality of the wireless link. ETT is the expected amount of time needed to successfully transmit a packet. It accounts for interference between links, number of retransmissions needed to successfully transmit a packet, and the data rate used at interfering links. Therefore, it can be used to improve throughput and ensure fairness between interfering links. For example, when the ETT of one node is significantly larger than that of another, this means that the node is not getting fair access to the wireless medium. By taking actions to keep ETT close among interfering nodes, fairness can be achieved.

The next step is to determine the set of rules and interactions that will determine the causality between concepts of the WCM. Whenever there are changes in the environment variables that require intervention by the CH, the WCM is triggered to take action. This action can be to adapt transmit power, data rate, or both. Duty cycle adaptation directly follows data rate adaptation, similar to the idea proposed in [12]. If the data rate is increased, the duty cycle is decreased, and vice versa. The WCM decides whether to adapt transmit power or data rate by considering the energy consumption resulting from such adaptations. To illustrate, consider the duty cycle shown in Figure 3.

Figure 3 shows a duty cycle with length T_{Duty} . It is divided into an active period, where packet transmissions take place; a fixed idle period, with length T_I ; and an adaptive sleep period. P_{Tx} is the energy consumed during the active transmission period in Watts, P_{idle} is the energy consumed during the idle period in Watts, and P_{sleep} is the energy consumed during sleep period. Note that P_{Tx} , P_{idle} , and P_{sleep} are determined by hardware characteristics. Also P_{idle} and P_{sleep} are constant parameters, but P_{Tx} is directly proportional to the transmission power used. Adapting transmit power and data rate takes place during the active period. The energy consumed at a single node during one duty cycle can be expressed as

$$E = P_{\text{Tx}} \times N \frac{L}{R} + P_{\text{idle}} \times T_I + P_{\text{sleep}} \left(T_{\text{Duty}} - T_I - N \frac{L}{R} \right), \quad (7)$$

where N is the number of packets transmitted during the active period, L is the packet length in bits, and R is the data rate used. The WCM uses (7) to compare between the energy consumption resulting from adapting transmit power or data rate. For example, if there is an increase in PLR that affects link reliability, the WCM needs to compare between increasing transmit power and decreasing data rate. After

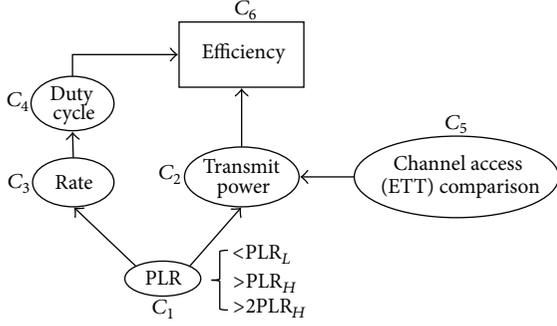


FIGURE 4: WCM for transmit power, data rate, and duty cycle adaptation. The concepts are labelled C_1 – C_6 .

removing the constant parameters from (7), this comparison can be expressed as

$$\frac{P_{Tx-Old}L}{R_{New}} - P_{sleep} \frac{L}{R_{New}} \geq \frac{P_{Tx-New}L}{R_{Old}} - P_{sleep} \frac{L}{R_{Old}}, \quad (8)$$

where P_{Tx-Old} and R_{Old} are the current transmit power and data rate values, while P_{Tx-New} is the value of transmit power if it will be increased and R_{New} is the value of data rate if it will be decreased. If the comparison in (8) is “True,” then this means that decreasing the data rate will result in higher energy consumption, and thus the transmit power will be increased. On the other hand, if it is “False,” then the data rate will be decreased instead. A similar process takes place if there is a decrease in PLR. Also, if there is a sharp decrease in PLR, the WCM has to increase transmit power and decrease data rate simultaneously in order to address the problem. After determining the end-to-end goal, environment variables, processes, and causality between concepts, the resulting WCM is shown Figure 4.

In order to implement the adaptation algorithm, conditional edge weights are used. Thus, when the PLR crosses a high or low threshold, a comparison according to (8) is performed, and a weight matrix, W , is formed similar to the one in (1). Edges $W_{(1,2)}$ or $W_{(1,3)}$ are activated according to the following rule:

$$\begin{aligned} &\text{If } PLR < PLR_L \text{ Or } PLR > PLR_H \text{ and (8) is} \\ &\quad \text{True, } W_{(1,2)} = 1 \text{ and } W_{(1,3)} = 0 \\ &\quad \text{False, } W_{(1,2)} = 0 \text{ and } W_{(1,3)} = -1 \end{aligned} \quad (9)$$

If $PLR > PLR_{2H}$

$$W_{(1,2)} = 1 \text{ and } W_{(1,3)} = -1,$$

where PLR_H , PLR_L , and PLR_{2H} are the values chosen for high, low, and too high PLR thresholds, respectively, and their specific values are determined according to the reliability requirements of the application. For example, if the application requires a reliability level of 98%, then $PLR_H = 0.02$, $PLR_L = 0.02 - 10\% = 0.018$, and $PLR_{2H} = 0.02 + 10\% = 0.022$. The value 10% was chosen to avoid the WCM from triggering frequent adaptations.

The rules in Expression (9) simply implement the algorithm detailed above, with the precaution that if $PLR >$

PLR_{2H} , then edges $W_{(1,2)}$ and $W_{(1,3)}$ are activated simultaneously to address the deterioration in channel conditions. Note that the WCM is triggered to take action if PLR changes considerably, or if the ETT of one node is 10% larger than that of an interfering node. The imbalance in ETT is addressed by activating $W_{(5,2)} = 1$, which leads to an increase in transmit power to give the node a better chance in accessing the medium. In addition, an increase in data rate will lead to the activation of $W_{(3,4)} = -1$, so that the duty cycle will be decreased. Also, an increase in transmit power or duty cycle will activate $W_{(2,6)} = -1$ and $W_{(4,6)} = -1$, respectively, prompting the WCM to recalculate the new value of efficiency, in order to ensure that the adaptations made do not violate the end-to-end goal.

After formulating the weight matrix, the CH formulates an array $A(t)$ similar to (2), reflecting the current activation levels of the concepts in Figure 4. For example, if $PLR > PLR_H$, then $A_1 = 1$, and if $PLR < PLR_L$, then $A_1 = -1$. According to (3), a multiplication is performed to determine $A(t + 1)$. The new values of concepts are then used to determine the actions needed to be taken. It is important to stress that the result of this multiplication process will not specify the exact level of transmit power or data rate to be used, but only if they should be increased or decreased.

4.4. Designing a WCM to Guarantee Connectivity and Coverage. This section illustrates how to design a WCM to satisfy connectivity and coverage constraints. Typical WSN have redundant nodes in order to extend their lifetime. Nodes in the network wake up every duty cycle with probability P . This value should ensure that every point in the area is within the sensing range of at least k sensors, and every node can find a routing path to the sink node. We assume that every node has a circular sensing range with radius r_s . The value P is lower as the number of redundant nodes increases and should increase to 1 as nodes die throughout the network’s lifetime. This is because, as nodes die, the number of redundant nodes in the network decreases, and thus the remaining nodes need to wake up with higher probability to guarantee connectivity and coverage.

The objective of the WCM in this section is to adapt P in every duty cycle in order to ensure connectivity and coverage. The environment variable that triggers the activity of this WCM is node failure. Thus, the CH will gather information from its cluster to determine how many nodes are active in every duty cycle. This can be done by monitoring traffic from regular sensor nodes in every duty cycle, since every node that wakes up senses and transmits information. Once the CH detects that the number of active nodes is close to violating the conditions of connectivity and coverage, the WCM will be triggered to increase P in order to ensure that more nodes will wake up in the following duty cycle.

In order to perform this task, we use the theorem derived in [24], which states that for sensing range r_s and communication range r_c at every node. When $\alpha = r_s/r_c \leq 1$, the area (D) is almost surely connected- k -covered if, for some growing function $\varphi(nP)$, P and r_s satisfy

$$nP\pi r_s^2 \geq \log(nP) + k \log \log(nP) + \varphi(nP), \quad (10)$$

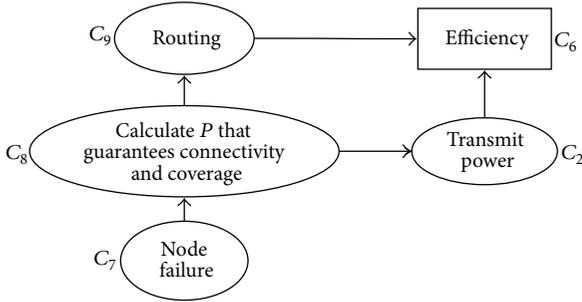


FIGURE 5: WCM to guarantee connectivity and coverage.

where n is the number of nodes in the network. The expression “connected- k -covered” means that every point in the network is covered by at least k sensors and can find a path to the sink node. Almost surely connected- k -covered means that as n increases to infinity, the probability of connected- k -coverage goes to 1.

Note that as nodes die in the network, n will decrease accordingly. Thus, the WCM will attempt to find P that satisfies Inequality (10) in every duty cycle according to the current value of n . Section 5 will illustrate how an appropriate activation function $\varphi(nP)$ is chosen to ensure high probability of connectivity and coverage when Inequality (10) is satisfied. The selected theorem achieves connectivity and coverage in a dynamic way, by adjusting P using simple mathematical operations, thus avoiding intensive processing operations typically associated with optimization problems used to address this issue. The WCM for guaranteeing connectivity and coverage is depicted in Figure 5.

As this figure shows, the WCM is triggered when node failure (C_7) occurs, causing a reduction in the number of active nodes. The edge $W_{(7,8)} = 1$ is activated, and the CH formulates an input array $A(t)$ as in (2), including the current activation values of the concepts ($A_7 = 1$). New activation values of concepts $A(t + 1)$ are then calculated using (3), causing concept C_8 to be activated. This will prompt the CH to calculate P that satisfies Inequality (10) according to the new value of n (after node failure). Based on the new value of P , rerouting (C_9) may need to be activated ($W_{(8,9)} = 1$) to find paths for the newly activated nodes. Note that the end-to-end goal of this WCM is again efficiency, since this is the goal of the overall system. Figure 5 also shows that an edge from C_8 – C_2 is added, so that the transmit power of nodes can be increased ($W_{(8,2)} = 1$) if the current values do not guarantee connectivity. Similar to the WCM in Figure 4, changes in transmit power or routing will activate $W_{(2,6)} = -1$ and $W_{(9,6)} = -1$, respectively, in order to ensure that adaptations do not violate the end-to-end goal.

4.5. Designing a WCM for Congestion Control. Congestion typically occurs when a particular set of nodes in the network becomes exposed to an amount of traffic larger than what the nodes can handle. This can cause queue buildups and significant transmission delays. The traditional way of dealing with congestion is to instruct the source node to reduce its

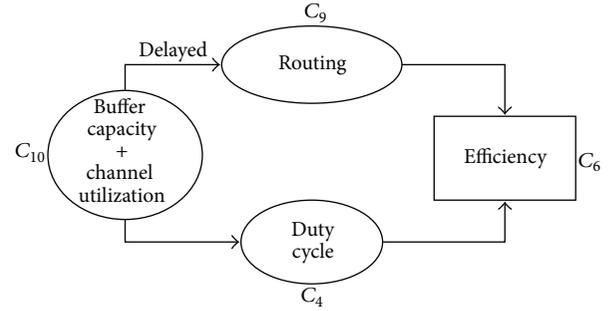


FIGURE 6: WCM for congestion control.

loading rate, which is the number of bits inserted in the transmission queue per second. Note that this is different from the node’s data rate, which corresponds to the rate at which bits are transmitted from the transmission queue. Although reducing source loading rate may reduce congestion, it has a significant disadvantage from the point of view of the QoS observed by the user. Decreasing the loading rate reduces the resources given to the user at the specific time when she/he is requesting more resources [8]. In WSN, there is a possibility for another solution to this problem, which is to increase the amount of resources available in the network. This approach was explored in [8], where topology-aware resource adaptation (TARA) was proposed. In TARA, sleeping nodes are instructed to wake up around congested areas, and data is routed through these nodes to alleviate the problem. This approach may not always work over the entire network lifetime if nodes die and no more redundant nodes are available to be activated. It also requires significant knowledge of network topology.

However, redundant nodes are not the only way of increasing network resources. Since duty cycle control is already being utilized, as previously explained, it can be exploited to keep the nodes active for longer periods of time when congestion occurs in order to dispense larger volumes of traffic. In the proposed WCM system, nodes will be instructed to extend their duty cycles to accommodate extra traffic. However, route maintenance will still be needed to disperse paths that are causing congestion but without the need for significant topology knowledge to learn which nodes need to be woken up. Using adaptive duty cycle will also work throughout network lifetime, since it does not depend on redundant nodes.

In order to implement this congestion resolution algorithm in a WCM, the first step is to identify the appropriate parameters to detect congestion. The parameters of channel utilization and buffer capacity are chosen, since they can be easily measured in real networks. Channel utilization is the fraction of time that a node detects the channel to be busy during a predefined interval, while buffer capacity is the remaining slots available in the buffer. Both parameters are required for an accurate indication of congestion. The WCM implementation is shown in Figure 6.

Once buffer capacity drops below a specific threshold “AND” channel utilization is above a specific threshold,

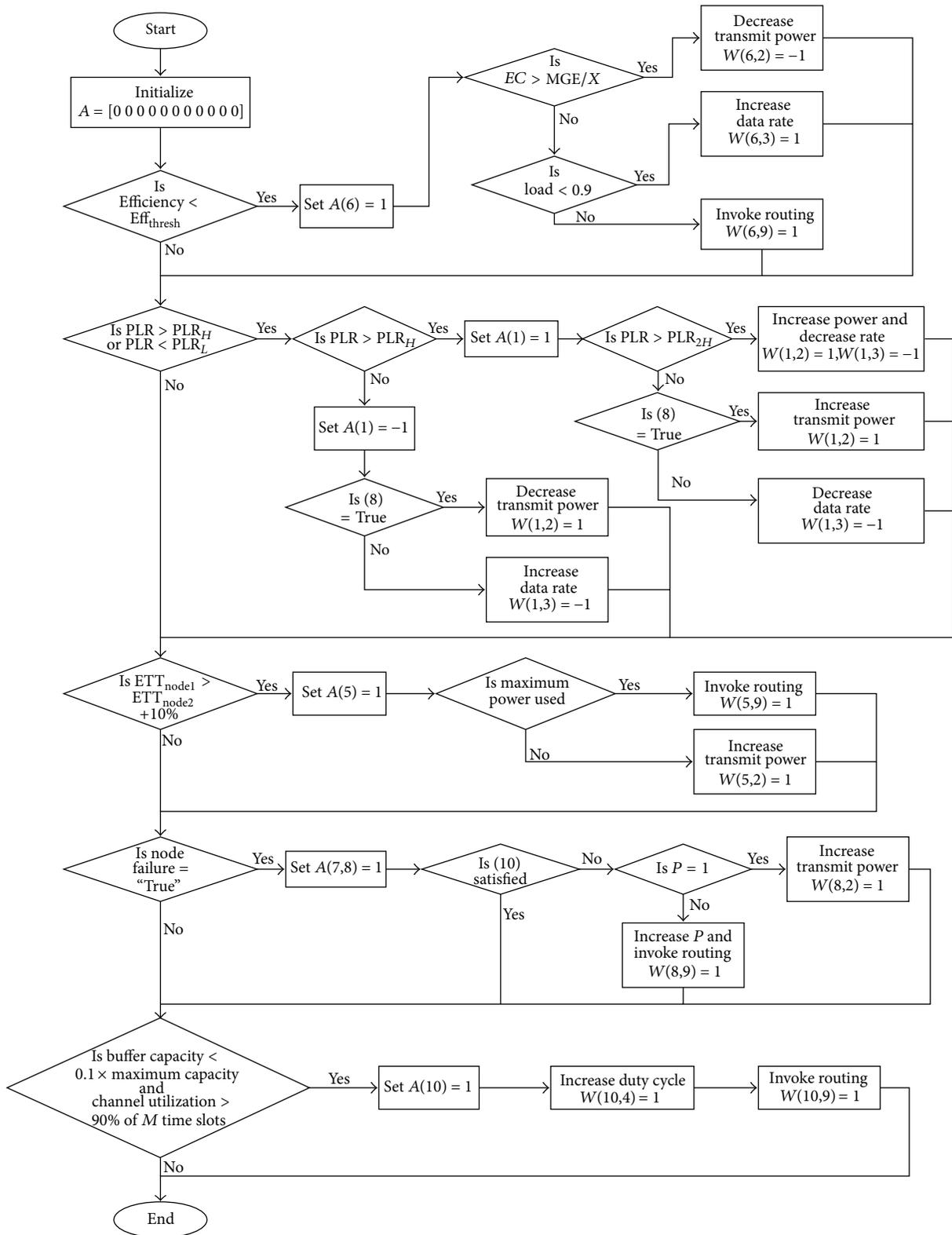


FIGURE 8: Process to determine the edge weights when the WCM is triggered.

TABLE 1: Simulation parameters.

Parameter	Value
Grid size	500 × 500 m
Number of nodes	169, 256, 400, 625, 900, 1225, and 1600
Transmit power levels	[−12, −6, 0, 2, 4] dBm
Data rates	[20, 40, 250] kbps
Cycle period	100 ms
Duty cycle	[0.25, 0.5, 0.75, 1] × 100 ms
Packet size	80 bytes
Packet generation rates (λ)	[0.007, 0.009, 0.02] pckts/sec
Initial battery power	5 Ahr
	Rx: 26 mAhr
Power consumption	Tx: 26 mAhr + (Tx power × packet size/data rate)
	Sleep: 0.3 μ Ahr × sleep time
Initial buffer capacity	100 packets

network simulator was created in MATLAB, in which we consider different network topologies to study the performance of the system under different scenarios. The lognormal propagation model with shadowing and multipath fading is used, and correlated shadowing is also considered to ensure a realistic propagation environment. Packet generation at every node is done according to a random Poisson process with rate λ pckts/sec. The main simulation parameters are shown in Table 1. As this table illustrates, the transmit power and data rate parameters of IEEE Standard 802.15.4 have been utilized, since it is the standard generally used in WSN.

The WCM system is compared to several known protocols. Particularly, the network management protocols TABU-RCC [6] and EECCR [7], and the transmit power and data rate adaptation protocol known as Symphony [13] are chosen. TABU-RCC and EECCR were chosen because their end-to-end goal is to maximize network lifetime in the presence of coverage and connectivity constraints, which is similar to the goals and constraints considered by our WCM system. Since TABU-RCC and EECCR do not consider transmit power or data rate adaptation, the WCM system is also compared against a network with the Symphony adaptation protocol. In addition, comparisons of the WCM system with a reduced set of features are included. Particularly, the comparisons include a WCM with no transmit power, data rate, or duty cycle adaptation, designated as “WCM-no adaptation,” and a WCM where the capabilities to adjust the value of P are switched off, designated as “WCM-no management.” As a baseline for the comparisons, a regular network with no transmit power or data rate adaptations and no network management protocols is included. Note that all systems are simulated with the MAC layer of the IEEE Standard 802.15.4. Also, all simulations are repeated enough times to ensure a confidence level of 95%.

5.1. Evaluation Using a Uniform Random Topology. In this section, the performance of the proposed system is evaluated using a uniform random topology, similar to the one that was previously illustrated in Figure 2. The metrics of network

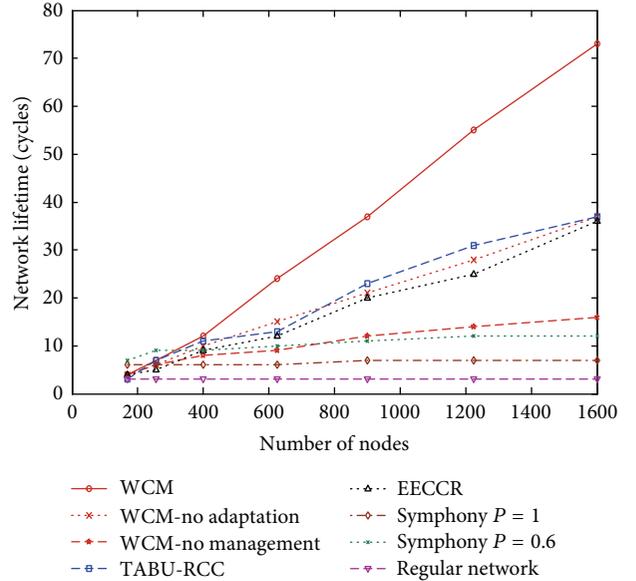


FIGURE 9: Network lifetime of WCM compared to other systems.

lifetime, throughput, and PLR and the amount of information that the network is able to dissipate before it reaches its lifetime are considered. Note that the amount of information dissipated before the network expires reflects the efficiency of the system in utilizing the available resources in transmitting packets. Network lifetime is defined as the time spanning from the start of network operation until the time when the remaining nodes can no longer guarantee full coverage and connectivity of the network. This particular definition was chosen as it reflects the capability of the systems to guarantee the QoS constraints of the network.

In all simulations conducted, time is divided into cycles. In every cycle, nodes wake up with probability P . In WCM, WCM-no adaptation, TABU-RCC, and EECCR, P changes throughout network lifetime to maximize energy efficiency. In Symphony, WCM-no management, and the regular network, there will be a predefined value for P that will not change throughout the lifetime of the network, since these protocols do not define methods for adapting P .

Nodes will transmit packets according to the packet generation rate, routing path, transmit power, and data rate specified by the WCM or the systems under comparison. The same packet generation rate, initial buffer capacity, initial battery power, packet size, and cycle period are used in all simulations to ensure fair comparisons between different protocols. The basic “shortest path” routing protocol is also used in all systems. In the first experiment, the lifetime of the proposed system is evaluated under different numbers of nodes. The results are shown in Figure 9.

As Figure 9 shows, the proposed WCM system achieves the highest lifetime results compared to other systems, especially in larger networks. In smaller networks, the WCM is forced to switch most nodes in the network to active mode in every duty cycle to guarantee connectivity and coverage. Therefore, there are not as many redundant nodes to be managed by the WCM. In larger networks the capabilities

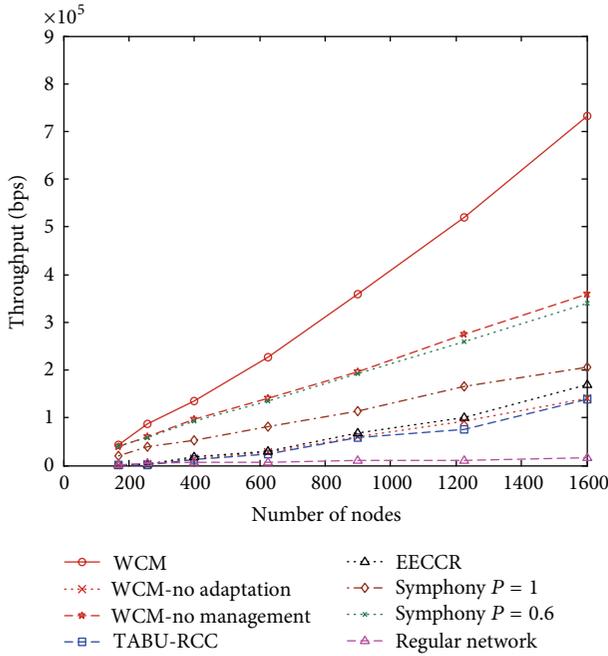


FIGURE 10: Throughput of WCM compared to other systems.

of management systems become more apparent. TABU-RCC comes second as it has capabilities to choose a suitable number of nodes to be active in every duty cycle. Figure 9 also shows that when transmit power, data rate, and duty cycle adaptation capabilities of the WCM system are switched off (WCM-no adaptation), the lifetime results approach those of TABU-RCC and EECCR, since the processes considered within those systems are similar to WCM-no adaptation. The Symphony protocol with $P = 0.6$ achieves longer lifetime than Symphony with $P = 1$. This is because, with $P = 1$, all nodes are switched on in every duty cycle, causing network resources to become depleted quickly. Note that $P = 0.6$ was chosen because it is the minimum value that can guarantee connectivity and coverage in the smallest network (169 nodes). Thus, $P = 0.6$ and $P = 1$ are included since they account for the minimum and maximum values that can be used. When the capabilities of the WCM to adjust the value of P are switched off (WCM-no management), the results are close to the Symphony system, since they both consider similar processes. WCM-no management was also simulated using $P = 0.6$, and it achieves slightly better results than Symphony, since it considers congestion. In all systems with no management protocols, the performance does not improve significantly as the number of nodes increases. This is because, with constant P , a larger number of nodes are switched on in every duty cycle as the total number of nodes in the network increase. Thus, redundant nodes are not being utilized efficiently.

In the next experiment, the throughput and PLR of the proposed WCM system are compared to other protocols. The results are shown in Figures 10 and 11. Note that the PLR results of the regular network and Symphony with $P = 1$ are

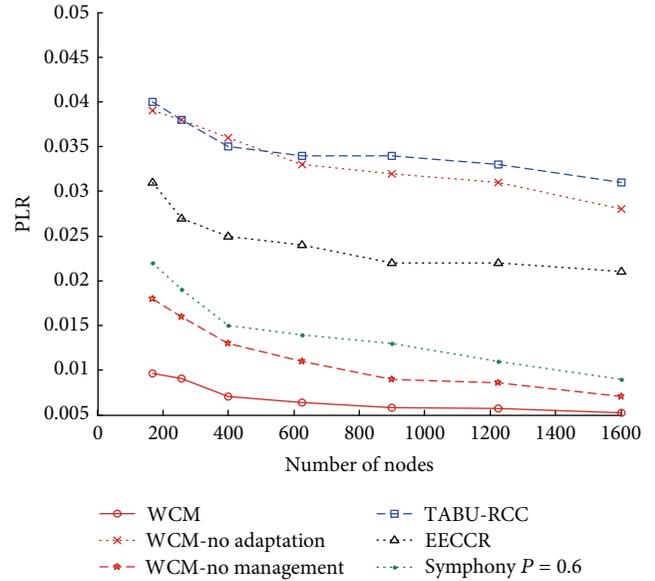


FIGURE 11: PLR of WCM compared to other systems.

not shown, since their PLR was significantly larger than other protocols.

As Figures 10 and 11 show, the proposed WCM system achieves the highest throughput and the lowest PLR results. In larger networks, the difference in throughput is larger in the favor of WCM, showing its ability to adapt system parameters efficiently. The WCM system considers data rate in the end-to-end goal (6) and tries to maximize it whenever possible as long as other constraints, such as PLR, are not violated. In addition, transmit power cannot be decreased unless $PLR \leq PLR_L$. This ensures that decreasing transmit power will not lead to packet failures, which leads to low PLR and high throughput. WCM-no management and Symphony with $P = 0.6$ also achieve high throughput, showing the impact of transmit power and data rate adaptation in achieving good throughput results. Figures 10 and 11 show that Symphony with $P = 0.6$ achieves similar throughput and PLR results to WCM-no management, and WCM-no adaptation achieves similar results to EECCR, confirming the results in Figure 9.

In the next experiment, the capability of the proposed system to utilize resources in disseminating information is evaluated. This is done by counting the total number of bits transmitted by all nodes in the network throughout its lifetime. The results are shown in Figure 12.

The results in this figure show that the WCM system is able to transmit the largest number of bits before the network reaches its lifetime. If this observation is combined with those of Figures 10 and 11, we can see that WCM is able to utilize network resources efficiently, since a large number of packets are transmitted with high throughput and low PLR. The networks employing Symphony also transmit large numbers of packets, especially in larger networks. This is because the fixed value of P forces a larger number of nodes to be switched on and transmit packets. However, from Figure 10, we have

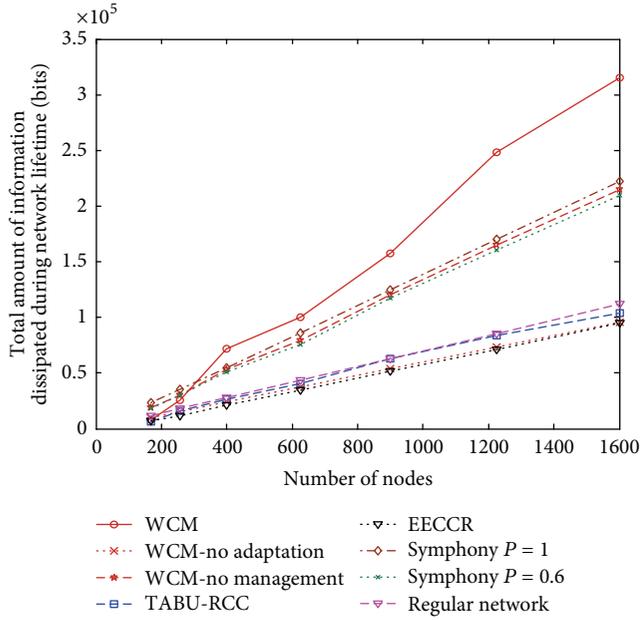


FIGURE 12: Ability of WCM to dissipate information compared to other systems.

seen that Symphony with $P = 1$ does not achieve good throughput results, due to high interference and collisions. As with previous experiments, WCM-no adaptation achieves similar results to TABU-RCC and EECCR, and WCM-no management achieves similar results to Symphony.

5.2. Evaluation Using a Grid Topology. In this section, the performance of the proposed system is evaluated using a grid topology, in order to test the system operation in different scenarios. As in random topology simulations, the same routing protocol, packet generation rates, initial buffer capacity, initial battery power, packet size, and cycle period are used for all systems under comparison. The topology used in this section is illustrated in Figure 13.

The same experiments that were performed for the random topology simulations are repeated in this section. The simulation results for network lifetime, throughput, and PLR are shown in Figures 14, 15, and 16, respectively. Note that the results for the total amount of information dissipated during network lifetime were not shown due to its similarity with the results of the random topology (Figure 12).

Figures 14–16 confirm the results of the random topology experiments. The proposed WCM system achieves good results in terms of network lifetime, throughput, and PLR. The performance also improves significantly in larger networks, since management capabilities become more apparent as redundant nodes increase. This clearly illustrates the efficiency of the proposed protocol in utilizing network resources in different network scenarios. Also, WCM-no adaptation achieves similar results to EECCR, and WCM-no management achieves similar results to Symphony. Note that TABU-RCC does not achieve good lifetime results in the

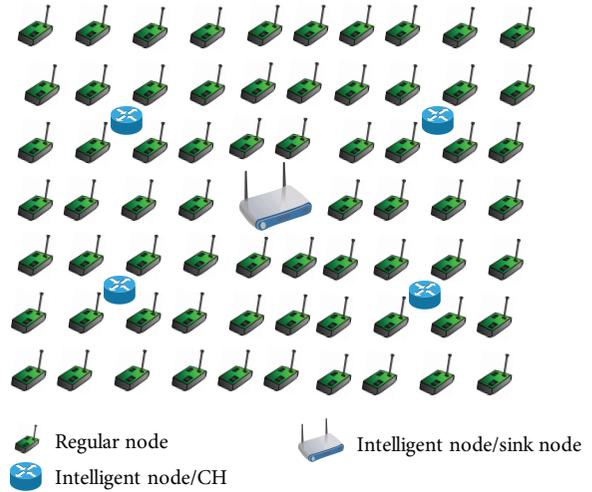


FIGURE 13: Grid topology used in simulations.

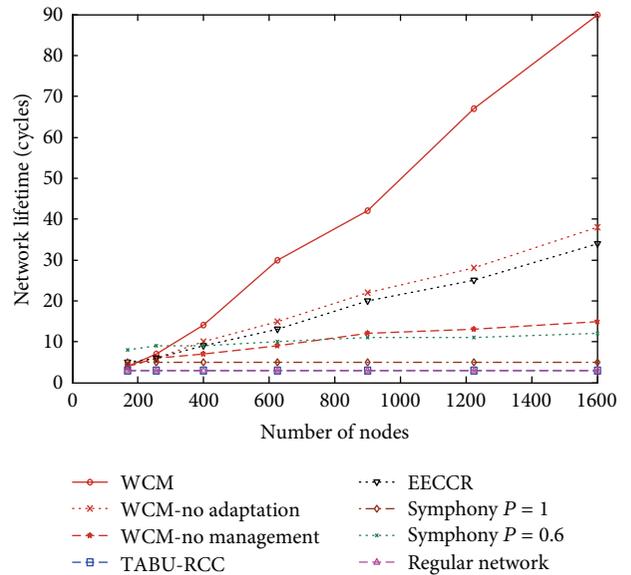


FIGURE 14: Network lifetime of WCM and other systems in grid topology.

grid topology because it was primarily designed for random topologies.

It is important to note that the efficient performance of WCM is achieved while maintaining low computational complexity. The execution of WCM is simply a matrix multiplication operation. The activated concepts determine the input array, and the status of the nodes determines the WCM matrix. On the other hand, in TABU-RCC, the best network configuration is discovered in every duty cycle through an extensive search operation. The algorithm starts with the configuration where all nodes are active, and then nodes are randomly switched off to find new energy-efficient configurations. The algorithm ends when switching off more nodes will no longer guarantee connectivity and coverage. Also, EECCR needs to determine new scheduling sets every

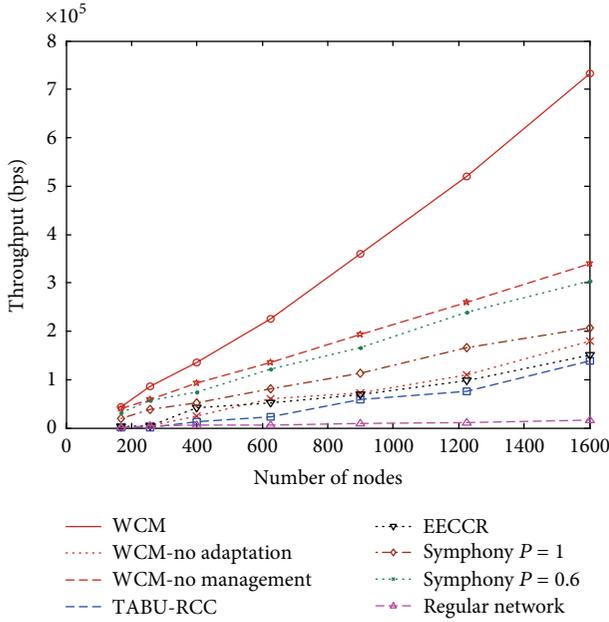


FIGURE 15: Throughput results of WCM and other systems in grid topology.

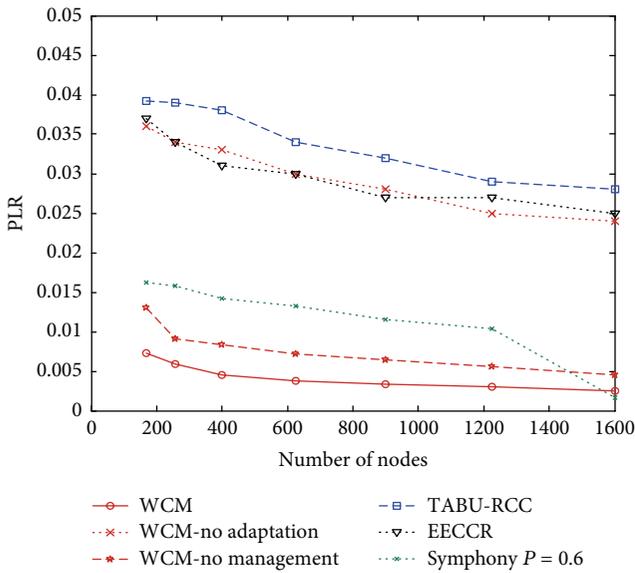


FIGURE 16: PLR of WCM and other systems in grid topology.

time there is a change in network topology. Thus, the processing time of TABU-RCC and EECCR are significantly longer than WCM. If TABU-RCC and EECCR are executed every few duty cycles, they will not be able to react quickly to network changes.

Note that WCM, TABU-RCC, and EECCR are network management protocols that may be implemented on top of existing network functionalities. For example, they all require the existence of routing protocols that can establish paths between nodes. The complexity of WCM is adjustable according to the concepts required. For example, if network

load is light, concepts that deal with congestion can be removed. This leads to a smaller WCM matrix and shorter processing time. The qualitative nature of WCM enables it to avoid long training processes. However, the proposed design ensures that it is able to produce some decisions that can be implemented quantitatively, such as transmit power and data rate adaptations. For concepts where quantitative decisions are produced, there is no need to run parallel protocols. The proposed WCM can also be implemented in a distributed way between the CHs and the sink node, thus avoiding the need for heavy processing at the sink node.

Furthermore, it is important to note that the complexity of the WCM system grows with the size of the network. An additional WCM machine needs to be implemented in every new CH. Also, if the number of nodes within a cluster grows, there will be a higher probability that the WCM will be triggered to operate. However, this is true of most network management systems that are implemented in a clustered hierarchy. Since the WCM system avoids long search processes, it will be quite scalable with larger networks and will respond faster than systems that employ optimization problems. In addition, the WCM system can be programmed to avoid frequent triggers simply by adjusting the thresholds that cause the WCM to operate. Moreover, the operation of the WCM can be delayed for a specific time after a threshold has been violated in case the problem is temporary and can resolve itself.

WCM does not require a significant increase in communication overhead. Information necessary for the execution of WCM, such as PLR levels and battery consumption levels, can be piggybacked on transmitted data packets. Occasional overhead packets can be sent if certain nodes require fast intervention. Note that in WSN nodes transmit sensed information regularly, so there will always be data packets on which information can be piggybacked.

5.3. On Choosing an Appropriate Activation Function for Guaranteeing Connectivity and Coverage. In Section 4.4, it was mentioned that an appropriate choice for the activation function $\varphi(nP)$ is necessary for Inequality (10) to be used efficiently in guaranteeing connectivity and coverage. The choice of this function depends on the area to be covered as well as sensing and communication ranges of nodes, and it is done offline. This section illustrates how $\varphi(nP)$ is chosen and the impact of this choice on network performance.

An experiment that is divided into two parts is conducted. In the first part, computer simulations are performed with parameters $k = 1$, $r_s = 100$ m, and $r_c = 200$ m. The area to be covered is a square of size $500 \text{ m} \times 500 \text{ m}$. The number of nodes $n = [100, 110, 120, \dots, 1600]$ nodes. For every value of n , 1000 simulation runs are performed, whereby every run consists of a new random deployment of nodes. In every run, the current deployment of nodes is checked to find if it achieves connectivity and coverage. After 1000 runs, the probability that the network is connected and covered using this value of n is calculated. This is repeated for values of $P = [0.1, 0.15, \dots, 0.3]$. Simulation results are plotted in Figure 17.

In the second part of this experiment, Inequality (10) is used with the same values of k , r_s , r_c , and P that

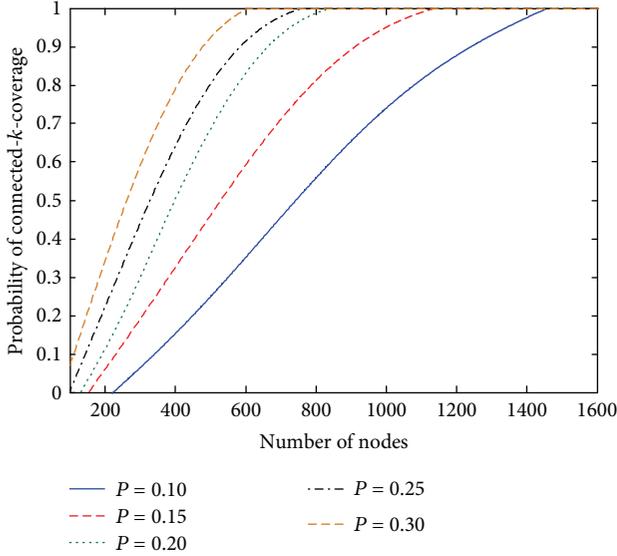


FIGURE 17: Probability of connectivity and coverage using different values of P .

were used in the simulations. A slowly growing function $\varphi(nP) = (\log \log(nP))^P$ is chosen to provide fine granularity for choosing P that satisfies Inequality (10), and the value of ρ provides some flexibility in the speed at which this function grows. However, a different $\varphi(nP)$ can be chosen provided that it is slowly growing. A fast growing function may lead to values of P that are either too conservative (activates too many nodes) or too low to guarantee connectivity and coverage. We use values of $\rho = [4.0, 4.1, 4.2, \dots, 5.5]$ for experimentation. For every value of ρ , the minimum number of nodes n that would satisfy Inequality (10) is evaluated. Then, simulation results from Figure 17 are used to check the probability of connectivity and coverage using this value of n . This is repeated for every value of P used in the simulations. The results are shown in Table 2.

As Table 2 shows, with $\rho = 4.0$, the minimum values of n that satisfy Inequality (10) for different values of P are 876, 584, 438, 351, and 292 nodes, respectively. However, the simulation results specify that the probability of connected- k -coverage is less than 0.63 using this value of n for all values of P . Therefore, $\rho = 4.0$ does not guarantee connectivity and coverage. On the other hand, with $\rho = 5.4$, the minimum values of n that satisfy Inequality (10) for different values of P are 1652, 1102, 826, 661, and 551 nodes, respectively. Here, the simulation results specify that the network is connected- k -covered with probability greater than 0.96 for all values of P . Thus, this value of ρ provides a better guarantee for connectivity and coverage. We have used $\rho = 5.4$ in all simulations in Sections 5.1 and 5.2. Thus, in order to use Inequality (10) in the WCM system, a small offline experiment can be performed using the given target area to be covered and the sensing ranges of nodes to determine the appropriate value of ρ . It is important to note that $\varphi(nP)$ was chosen in this way so that there is no need to change it for different areas and sensing ranges of nodes. Only the

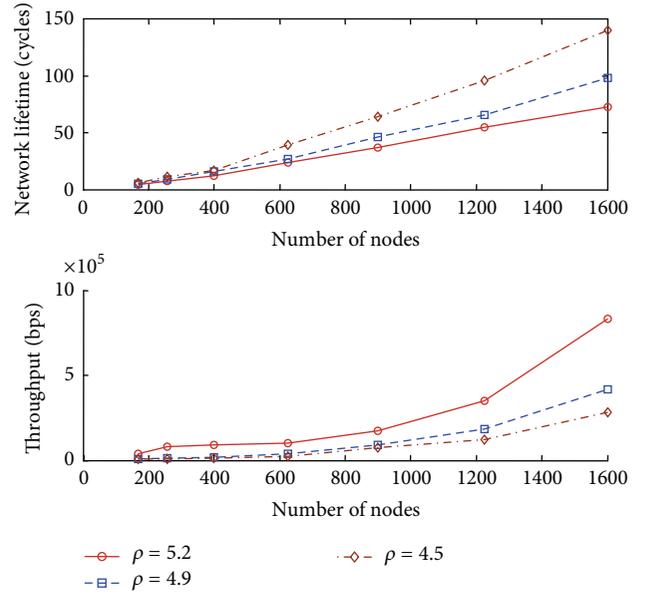


FIGURE 18: Network lifetime and throughput results at different values of ρ .

appropriate value of ρ needs to be determined for every network once before the network becomes operational.

To illustrate the impact of varying ρ on network performance, network simulations are performed similar to the ones in Sections 5.1 and 5.2, using the WCM system. A uniform random topology with the same simulation parameters as the ones used in Section 5.1 is utilized, and system performance is evaluated under values of $\rho = \{4.5, 4.9, 5.4\}$. Values of ρ lower than 4.5 provide poor guarantees for connectivity and coverage, as shown in Table 2, and thus were not considered. Metrics of network lifetime and throughput are considered, and the results are illustrated in Figure 18.

This figure shows that network lifetime increases with lower values of ρ , while throughput decreases. This is because, with lower values of ρ , a smaller number of nodes are switched on every duty cycle. Thus, the rate of consumption of network resources decreases causing lifetime to increase. However, switching on a smaller number of nodes with lower values of ρ means that the network is less dense. Therefore, the total number of transmitted packets decreases and connectivity decreases, which may cause higher probability of transmission failure and lower throughput.

In order to test the efficiency of the chosen function in guaranteeing connectivity and coverage, an experiment is performed, where the average percentage of the total area that is covered by at least one sensor node is measured in every time slot during the lifetime of the network. This percentage is calculated by measuring the fraction of the area that is within the sensing range (r_s in Section 4.4) of at least one sensor node and then multiplying by 100. In this experiment, a maximum number of nodes of 625 is considered, since connectivity and coverage problems are more probable in networks with smaller node densities. In addition, node deployment is done in a uniform random fashion, since

TABLE 2: Values of ρ and their corresponding probability of connected- k -coverage.

ρ	$P = 0.1$		$P = 0.15$		$P = 0.2$		$P = 0.25$		$P = 0.3$	
	Min. n	Probability	Min. n	Probability	Min. n	Probability	Min. n	Probability	Min. n	Probability
4.0	876	0.63	584	0.57	438	0.57	351	0.55	292	0.57
4.1	904	0.66	603	0.60	452	0.60	362	0.57	302	0.60
4.2	934	0.68	623	0.62	467	0.63	374	0.59	312	0.62
4.3	966	0.71	644	0.65	483	0.66	387	0.62	322	0.64
4.4	1002	0.74	668	0.67	501	0.69	401	0.64	334	0.67
4.5	1041	0.77	694	0.70	521	0.72	417	0.67	347	0.69
4.6	1084	0.80	723	0.74	542	0.75	434	0.70	362	0.72
4.7	1131	0.83	754	0.77	566	0.79	453	0.73	377	0.75
4.8	1184	0.87	789	0.80	592	0.82	474	0.77	395	0.78
4.9	1242	0.90	828	0.84	621	0.86	497	0.80	414	0.81
5.0	1307	0.93	871	0.87	654	0.89	523	0.83	436	0.85
5.1	1379	0.97	919	0.92	690	0.92	552	0.87	460	0.88
5.2	1460	1.00	973	0.95	730	0.95	584	0.90	487	0.91
5.3	1550	1.00	1034	0.97	775	0.97	620	0.93	517	0.94
5.4	1652	1.00	1102	0.99	826	1.00	661	0.96	551	0.97
5.5	1767	1.00	1178	1.00	884	1.00	707	1.00	589	1.00

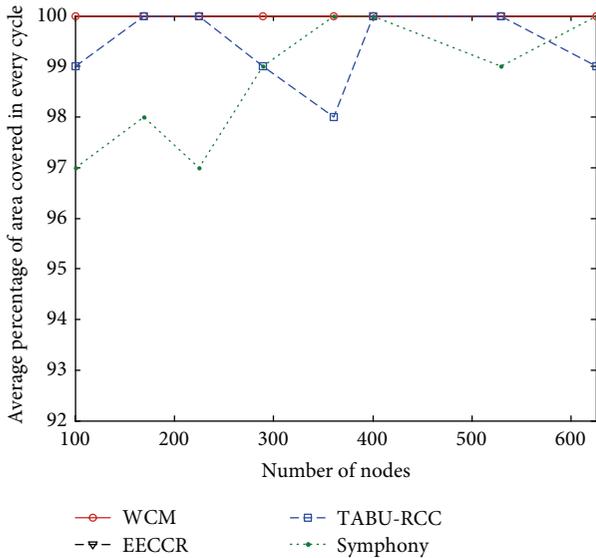


FIGURE 19: Average coverage results of WCM compared to other protocol.

this is the more challenging scenario for connectivity and coverage. The performance of WCM is compared to EECCR, TABU-RCC, and Symphony. Simulation results are shown in Figure 19.

The results in this figure show that the WCM system achieves 100% connectivity and coverage in every duty cycle for all network sizes. This proves that the chosen function performs efficiently and is able to switch an appropriate number of nodes to active mode during network lifetime, without the need for complicated optimization operations. Furthermore, if we combine these results with the ones in Figure 9, it can

be deduced that the number of activated nodes in every cycle with WCM achieves good energy efficiency that leads to the maximization of network lifetime. Moreover, Figure 19 shows that EECCR also achieves 100% connectivity and coverage, while TABU-RCC and Symphony do not. This is because Symphony has no means of guaranteeing connectivity and coverage, while TABU-RCC is a heuristic algorithm that may not achieve 100% coverage and connectivity in every cycle.

6. Conclusions

In this paper, a cross-layer framework for network management in WSN based on the WCM tool was presented. The proposed system is able to perform efficient reasoning while considering multiple objectives and constraints. By maintaining an overview of all network elements, the WCM is able to ensure that they operate coherently. The WCM continuously monitors the required QoS levels specified by the user and takes fast and efficient actions whenever those levels are violated. This is achieved while avoiding high complexity typically required by optimization problems or long search operations.

To evaluate the performance of the proposed system, extensive computer simulations were conducted, and the WCM was compared against other well-known protocols. The WCM system showed ability to utilize network resources efficiently and adapt to different network scenarios and conditions by adjusting system parameters accurately. For these reasons, the WCM outperforms other protocols in metrics of lifetime, throughput, and PLR.

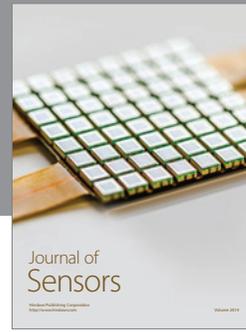
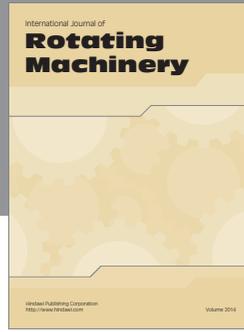
In future work, the authors will present a theoretical framework that models the behavior of the proposed system. This model will be used to further analyze system functionality and discover new methods to improve system performance.

Acknowledgments

The authors would like to thank the NSERC DIVA Network and the Ontario Research Fund (ORF) WiSense project for supporting this work.

References

- [1] M. Ibnkahla, *Adaptation and Cross-Layer Design in Wireless Networks*, CRC Press, Boca Raton, Fla, USA, 2008.
- [2] N. M. Freris, H. Kowshik, and P. R. Kumar, "Fundamentals of large sensor networks: connectivity, capacity, clocks, and computation," *Proceedings of the IEEE*, vol. 98, no. 11, pp. 1828–1846, 2010.
- [3] A. Ananda, M. Chan, and W. Ooi, *Mobile, Wireless, and Sensor Networks: Technology, Applications, and Future Directions*, John Wiley & Sons, Hoboken, NJ, USA, 2006.
- [4] M. Ibnkahla, *Wireless Sensor Networks: A Cognitive Perspective*, CRC Press, Boca Raton, Fla, USA, 2012.
- [5] C. Facchini and F. Granelli, "Towards a model for quantitative reasoning in cognitive nodes," in *Proceedings of the IEEE Globecom Workshops*, pp. 1–6, Honolulu, Hawaii, December 2009.
- [6] A. Chamam and S. Pierre, "On the planning of wireless sensor networks: Energy-efficient clustering under the joint routing and coverage constraint," *IEEE Transactions on Mobile Computing*, vol. 8, no. 8, pp. 1077–1086, 2009.
- [7] Y. Jin, L. Wang, J.-Y. Jo, Y. Kim, M. Yang, and Y. Jiang, "EECCR: an energy-efficient m-Coverage and n-Connectivity routing algorithm under border effects in heterogeneous sensor networks," *IEEE Transactions on Vehicular Technology*, vol. 58, no. 3, pp. 1429–1442, 2009.
- [8] J. Kang, Y. Zhang, and B. Nath, "TARA: topology-aware resource adaptation to alleviate congestion in sensor networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 18, no. 7, pp. 919–931, 2007.
- [9] L. Q. Tao and F. Q. Yu, "ECODA: enhanced congestion detection and avoidance for multiple class of traffic in sensor networks," *IEEE Transactions on Consumer Electronics*, vol. 56, no. 3, pp. 1387–1394, 2010.
- [10] G. S. Kasbekar, Y. Bejerano, and S. Sarkar, "Lifetime and coverage guarantees through distributed coordinate-free sensor activation," *IEEE/ACM Transactions on Networking*, vol. 19, no. 2, pp. 470–483, 2011.
- [11] S. Panichpapiboon, G. Ferrari, and O. K. Tonguz, "Optimal transmit power in wireless sensor networks," *IEEE Transactions on Mobile Computing*, vol. 5, no. 10, pp. 1432–1446, 2006.
- [12] X. Zhao, E. Bdira, and M. Ibnkahla, "Joint adaptive modulation and adaptive MAC protocols for wireless sensor networks," Internal Report, Queen's University, 2012.
- [13] K. Ramachandran, R. Kokku, H. Zhang, and M. Gruteser, "Symphony: synchronous two-phase rate and power control in 802.11 WLANs," *IEEE/ACM Transactions on Networking*, vol. 18, no. 4, pp. 1289–1302, 2010.
- [14] Y.-L. Kuo, K.-W. Lai, F. Y.-S. Lin, Y.-F. Wen, E. H.-K. Wu, and G.-H. Chen, "Multirate throughput optimization with fairness constraints in wireless local area networks," *IEEE Transactions on Vehicular Technology*, vol. 58, no. 5, pp. 2417–2425, 2009.
- [15] E. Felemban, C.-G. Lee, and E. Ekici, "MMSPEED: multipath Multi-SPEED Protocol for QoS guarantee of reliability and timeliness in wireless sensor networks," *IEEE Transactions on Mobile Computing*, vol. 5, no. 6, pp. 738–754, 2006.
- [16] M. A. Razzaque, M. M. Alam, M. Mamun-Or-Rashid, and C. S. Hong, "Multi-constrained QoS geographic routing for heterogeneous traffic in sensor networks," in *Proceedings of the 5th IEEE Consumer Communications and Networking Conference (CCNC '08)*, pp. 157–162, Las Vegas, Nev, USA, January 2008.
- [17] S. Ratnaraj, S. Jagannathan, and V. Rao, "OEDSR: Optimized energy-delay sub-network routing in wireless sensor network," in *Proceedings of the IEEE International Conference on Networking, Sensing and Control, (ICNSC'06)*, pp. 330–335, April 2006.
- [18] J. A. Dickerson and B. Kosko, "Virtual worlds as fuzzy cognitive maps," in *Proceedings of the IEEE Annual Virtual Reality International Symposium*, pp. 471–477, September 1993.
- [19] M. Glykas, *Fuzzy Cognitive Maps: Advances in Theory, Methodologies, Tools, and Applications*, Springer, Berlin, Germany, 2010.
- [20] Y. Miao, C. Miao, X. Tao, Z. Shen, and Z. Liu, "Transformation of cognitive maps," *IEEE Transactions on Fuzzy Systems*, vol. 18, no. 1, pp. 114–124, 2010.
- [21] A. K. Tsadiras, "Comparing the inference capabilities of binary, trivalent and sigmoid fuzzy cognitive maps," *Information Sciences*, vol. 178, no. 20, pp. 3880–3894, 2008.
- [22] D. E. Koulouriotis, I. E. Diakoulakis, and D. M. Emiris, "Realism in fuzzy cognitive maps: Incorporating synergies and conditional effects," in *Proceedings of the 10th IEEE International Conference on Fuzzy Systems*, pp. 1179–1182, Melbourne, Australia, December 2001.
- [23] X. Luo, X. Wei, and J. Zhang, "Guided game-based learning using fuzzy cognitive maps," *IEEE Transactions on Learning Technologies*, 2008.
- [24] G. Yang and D. Qiao, "Critical conditions for connected-k-coverage in sensor networks," *IEEE Communications Letters*, vol. 12, no. 9, pp. 651–653, 2008.
- [25] A. El Mougy, *A cognitive framework for WSN based on WCM and Q-learning*, [Ph.D. thesis], Queen's University, Kingston, Canada, 2012.



Hindawi

Submit your manuscripts at
<http://www.hindawi.com>

