This article has been accepted for publication in IEEE Access. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2024.3407672

IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier 10.1109/ACCESS.2022.Doi Number

Adaptive Priority Scheduling of Internet of Things Data for Disaster Management in Smart Cities

Marie C. Pognon[1,2], Alejandro Quintero[1], Member, IEEE, and Samuel Pierre[1], Senior Member, IEEE

[1]Department of Computer and Software Engineering, Polytechnique Montréal, Montreal, QC, Canada
[2]ISTEAH, Cité du savoir, Génipailler, Milot, Haïti

Corresponding author: Marie C. Pognon (e-mail: marie-carria.pognon@polymtl.ca).

ABSTRACT In the recent context of the emergence of smart cities, the massive amount of data generated by connected objects has led to unprecedented demands in terms of data transfer. The various constraints linked to their number, their characteristics, and their transmission are even greater and dim the effectiveness, in their regard, of traditional data planning schemes. As a result, the need to minimize the delivery time of urgent packets while reducing the average data delay, the difficulty in choosing and combining the appropriate criteria for classifying and prioritizing data, and the loss of packets are of continuing concern. In this paper, we propose an adaptive scheduling model based on multilevel priority packet classification, preemptive packet queuing with dynamic and adaptive reordering, contingency migration of packets in critical situations, and adaptive criticality-based selection of packet next-hop. We introduce two new parameters for scheduling decisions: the ratio of per-level deadlines reflecting the evolution of a packet in the network and the migration coefficient based on the experience of same-characteristic packets. Performance evaluation shows that the proposed model effectively prevents data loss and prioritizes the transfer of emergency data over a hierarchical wireless sensor network. Moreover, it guarantees the shortest delays for urgent data with an improvement of 31% and promotes fairness toward less urgent ones. The lowest delivery rate observed with the proposed method is 99.9%.

INDEX TERMS Adaptive scheduling, Internet of Things, queuing analysis, smart cities, wireless sensor network.

I. INTRODUCTION

In the quest to improve the quality of life of the citizens, the Internet of Things (IoT) in smart cities is becoming more and more established in our daily lives with various applications in areas such as building automation, parking management, energy management, traffic management, climate, and health monitoring, to name a few. However, the world's urban population growth [1]–[2] increases the demand for the consumption of services and products in cities, which, in turn, contributes to an ecological deterioration of the environment and, subsequently, to a growing risk of large-scale disasters. The infrastructure of smart cities must provide ways for better disaster management as the latter is one of the areas of intervention of municipalities in their support to citizens. The notion of urgency is intrinsic to the detection of cataclysms. Thus, to be useful to the application that uses them, the data generated by connected objects must meet the minimum condition of arriving at their destination, on time and without alteration. Indeed, to increase the effectiveness of relief operations, the workforce deployed during a disaster response must promptly receive vital information for appropriate and rapid use [3]. Several research works have been done on reducing the transfer time of critical IoT data. However, most of the methods proposed for this purpose do not always heed the expiration of lower priority packets and result in starvation or loss [4]–[7]. When scheduling packets according to their priority, the difficulty lies in combining methods that can both ensure better transmission times for urgent packets and provide a certain level of fairness in the handling of non-urgent data. On the other hand, packet loss is undesirable as it affects real-time transmission in IoT systems and, particularly, the reliability and performance of emergency applications [8].

This article has been accepted for publication in IEEE Access. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2024.3407672

The loss of warning messages (e.g., in case of fires, road accidents, natural disasters, etc.) leads to disastrous scenarios and erroneous or delayed decisions. Data retransmission has been explored for improving the reliability of transmission [9]. However, such a method generates more traffic and increases latency, due to the supplemental time it requires. Moreover, the additional use of bandwidth by the retransmitted information impacts the throughput in real-time.

IoT networks must ensure the timely transmission and delivery of disaster-related data despite the high probability of congestion in such situations. However, many parameters must be checked, and several requirements must be met for a successful data transfer. Traditional planning schemes are less and less suitable and have been outclassed by more recent and adapted solutions [4], [7], [10]–[12] which, to the best of our knowledge, cannot simultaneously prioritize urgent data and avoid data loss. A research challenge is to conceive a model that concurrently addresses these constraints. This could be accomplished by adopting data scheduling and routing strategies that adapt to changing network conditions. In this paper, we present an adaptive scheduling model that can achieve a fair prioritization of IoT data and a balanced queueing of packets which gives precedence to critical data, reduces packet delivery delay, and greatly decreases data loss.

The contributions of this paper are summarized as follows:

1) We design an adaptive scheduling method for IoT data in smart cities aimed at reducing transmission delays of urgent data by bringing into play an efficient criteria-based packet classification and prioritization, and a packet reordering strategy sensitive to the packet experience in the network.
2) We present an adaptive multilevel queuing strategy aimed at avoiding data loss by introducing a contingency migration plan using fallback queues for the migration of packets in critical situations.
3) We propose an adaptive strategy for the selection of packet next hop that takes the packet criticality into account. It is based on continuous monitoring and analysis of several status- and performance-related metrics of the potential next forwarding nodes.
4) We introduce two novel parameters for scheduling decisions: firstly, the ratio of per-level deadlines reflecting the evolution of a packet in the network and, secondly, the migration coefficient based on the experience of same-characteristic packets and reflecting the varying conditions of the network.

The results of our simulations show that our model effectively prioritizes data transfer over the network. It also prevents packet loss, guarantees the shortest delays for urgent data, and promotes fairness toward low-priority packets.

The rest of this paper is organized as follows. We define some concepts and terminologies and present the related works in Section II. We describe the proposed model in Section III. In Section IV, we discuss the results. Finally, we conclude the paper in Section V.

II. BACKGROUND AND RELATED WORKS

Municipalities are increasingly using information technologies to improve their management and enhance public services to citizens [24]. The expectation is to have a secure and resilient city through the provision of services to citizens without interruption [1]. Smart cities require intelligent, automated infrastructures capable of handling exponential data creation, long-term storage, rapid processing, and precise analysis.

The Internet of Things (IoT), with its widespread use of devices and infrastructure, is an enabler of smart cities and its most important data source. Several attempts have been made to define IoT from different perspectives [25]. More particularly, the International Telecommunications Union (ITU), cited by [26], equates IoT to "a global infrastructure for the information society, enabling advanced services by interconnecting elements (physical and virtual) based on existing, scalable and interoperable information and communication technologies". In general, within an IoT system, the information is collected by sensors and transferred to the servers of the control centers where it undergoes a processing and analysis phase [27]. This can lead to the actuation of different mechanisms depending on the needs identified during the interpretation of the data [28].

The management of disasters is also an intervention area of municipalities in their support to citizens. In disaster management systems, IoT finds its utility, during the preparation phase, in monitoring key parameters, detecting abnormal, dangerous and threatening conditions, and signaling these anomalies through triggering alerts and broadcasting emergency messages [29]. Moreover, IoT is useful during the response phase in rescue interventions. It helps in locating victims, assessing the spread of incidents, and identifying evacuation routes, among others [29]–[32]. During the restoration phase, it is used in resource distribution for beneficiary identification and inventory monitoring. In combination with artificial intelligence [38], data analysis and virtual reality, IoT can contribute to disaster forecasting and predictive maintenance for infrastructure protection in the mitigation phase.

The effectiveness of relief operations is intrinsically linked to the prompt reception of vital information. Data planning, also called scheduling, is the strategy adopted to ensure the optimal transmission of data while adapting to resource constraints. Scheduling is the process that determines the departure order of data packets and the next packet to be transmitted [19]. In general, a data scheduling mechanism will be chosen to achieve a specific goal for particular conditions [15]. For this purpose, this mechanism must examine the characteristics of the analyzed traffic (e.g., the type, deadline, priority, etc.) [7], [20], [38]. The data latency tolerance threshold varies depending on different parameters, such as the type of application that produced it

IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

and the urgency, among others. The IEEE Standards Association [22] defines latency as "the delay incurred by a frame during its propagation between two points on a network, measured by the time difference between when a known reference point in the frame passes the first point and the moment the reference point in the frame passes the second point". In practice, latency is the sum of the time required to access the transmission channel and the propagation time from the source to the destination [23]. Too much latency can cause data loss. A packet is deemed lost when it is transmitted over the network and does not reach its intended destination [21]. To meet the needs of real-time analysis and processing of data provided by connected objects (IoT) in smart cities, dynamic traffic management is necessary to guarantee quality of service (QoS) requirements for the transiting packet on the network. These requirements are increased tenfold in disaster situations because time and precision are essential to survival. Following are some methods developed for the transfer of data generated by connected objects in the area of IoT data scheduling and routing. They are compared in Table I.

The authors in [4] adopted a preemptive scheduling model for hierarchical wireless sensor networks (WSN) using three priority queues and prioritizing packets coming from the lowest nodes. To guarantee fairness, lowest priority packets are transmitted before higher priority packets after a certain waiting time. However, this method could perform better, had it used the packets' deadline to sort them and avoid losing them because of timeout. The scheduling method for WSNs developed in [5] aims to decrease the end-to-end delay and provides three priority levels defined as in [4] for hierarchically distributed nodes. In this method, lower-priority packets suffer from the wait imposed at intermediate nodes for data aggregation, as they can become irrelevant because of this delay or be lost. In [6], the authors combined the Backpressure Scheduling technique and the Shortest Path First algorithm for choosing the next node when transmitting IoT packets. This solution only defines two levels of priority that do not reflect common situations. In addition, non-urgent packets run the risk of being blocked in the event of high-priority packets. The authors in [7] used preemption in favor of urgent packets and fairness toward non-urgent ones. The source node first informs the destination before sending the packet itself. However, the risk of data loss remains, in case of heavy traffic, if a packet expires before it leaves the queue. It would benefit from performing dynamic updating of the network when a node fails or a new one is added. The Long Hop First algorithm and the Dijkstra algorithm are both combined in [10] to reduce packet loss and energy consumption of IoT applications in a WSN where messages are sent from a sensor to the base station via the cluster head. Although this solution showed performance in terms of delay and packet loss, it does not address the case of urgent packets. A mechanism to resist node failure and to improve the performance of data packets transmission in real-time is proposed in [11]. The scheduling phase of their approach uses an improved version of the DMP model [4] by ordering packets of the same priority based on their deadline instead of their size. Faulty nodes are detected, and alternative paths are constructed to reconnect isolated subtrees to the main network. Although efficient in terms of urgent packet delay, end-to-end delay, and packet loss rate, this method is specific to tree-based networks. The model presented in [12] determines the packet route, in a WSN mesh topology, based on the strategy of the Shortest Path and Less number of Links (SPLL). For each path, to promote traffic load balance, the nodes that have the fewest neighbors are chosen. This method, however, does not address the case of urgent packets. In [13], the authors have developed the Multilevel Dynamic Feedback Scheduling (MDFS) algorithm to minimize transmission delay and increase the delivery rate. The time quantum determines a packet priority for which the model provides three levels, the highest being gpt urgent data. Priority management is supported at each node by a system of three queues connected by a feedback mechanism. The MDFS performs inter-queue migrations based on the comparison of the time quantum of the packet to the minimum and masimum limit values of the queue to avoid starvation problems. In the Packet Rank-Based Data Scheduling (PRBDS) model adopted in [16], each new packet that arrives at a node is placed in a queue corresponding to its rank which is calculated according to the priority, the deadline and the size of the packet, respectively. This model has the advantage of using several important parameters to classify the packets, which greatly reduces the wait time while promoting fairness. The method in [17] schedules packets based on type and priority and drops less important packets in the event of congestion. It selects the paths according to their score calculated from the path length, the nodes' energy state, the loss rate, and the congestion level. Higher-priority packets are routed through paths with a higher score. Although the path selection criteria are judicious, this method, however, encourages packet loss. A prediction-based dynamic scheduling mechanism is proposed in [18]. A weight coefficient for each queue is used and calculated based on the number of packets that arrived in the queue during the previous cycle. This solution encourages packet loss for low-priority data classes and does not guarantee fairness. The authors in [35] adopted a path elimination method to remove paths that may cause packet loss or delay exceeding the packet deadline. The best-suited path is selected based on the expected network performance for each packet. This paper shows that using previous network experience to determine the packet routing path can significantly improve the overall network performance in terms of packet loss and reduce average end-to-end delay. In [36], a packet next hop is selected by first analyzing the quality of the path (link) which is based on the number of neighbors of the receiving node and the residual energy. This

TABLE I
COMPARISON OF THE DATA SCHEDULING METHODS IN IOT

Method	Reduces Packet Loss	Reduces Average Delay	Considers Priority Vs Fairness	Considers Packet Deadline	Considers Emergency	Considers Multilevel Priority	Is adaptive based on network status evolution
Nasser *et al.* [4]		✓	✓		✓	✓	
Akila *et al.* [5]	✓	✓	✓		✓	✓	
Qiu *et al.* [6]	✓	✓		✓	✓	✓	
Qiu *et al.* [7]	✓	✓		✓	✓	✓	
Farhan *et al.* [10]	✓	✓					
Qiu *et al.* [11]	✓	✓		✓	✓	✓	
Farhan *et al.* [12]	✓						
Natarajan *et al.* [13]	✓	✓	✓	✓	✓	✓	
Kavitha *et al.* [14]	✓				✓	✓	
Mahendran *et al.* [16]	✓	✓	✓	✓	✓		
Abd El Kader *et al.* [17]	✓					✓	✓
Sharma *et al.* [18]	✓	✓				✓	✓
Al-Turjman *et al.* [35]	✓	✓					✓
Farhan *et al.* [36]	✓						
Ullah *et al.* [37]	✓	✓					

model is adaptive only on the basis of the node's energy state and mentions the probability of congestion without, however, proposing a solution to this problem. The model established in [37] introduces two schemes for path selection for which the criterion for both methods is a weighting function of the highest residual energy, highest signal-to-noise reduction (SNR), smallest distance, and Bit Error Rate (BER). The chosen weighed criteria are good avenues for more in-depth analysis in similar research.

As described in previous paragraphs, dynamic planning methods explore different criteria for data classification and prioritization. Some adopt the urgency, the deadline [13], or the size [4], [16]. Others preferentially use the number of hops and distance [10], [12], the required throughput, and the packet's tolerance degree to delay [14]. As the end-to-end delay is a determining factor for the usefulness of data, several models aim to improve that of urgent data [4]–[5], [11], [15], sometimes to the detriment of lower priority data, which find themselves blocked during the influx of critical packets [5]. However, none of these methods guarantees, at the same time, reduction of delay, reduction of data loss, priority given to urgency, and fairness toward low-priority data [4]–[6], [10]–[11], [14]–[15]. Many of these dynamic data planning approaches for IoT are not adaptive. Some suppose the evolution of traffic conditions and the state of the network to be static, and the latter are only accounted for when initiating the data transfer. For the methods claiming to perform adaptive scheduling, most of them are oriented toward energy efficiency [33]–[34], [36]–[37] and are more focused on path planning. Their schemes do not ponder the effect of network traffic characteristics [33], the fairness toward low-priority packets [18] and the emphasis on urgent data prioritization [34] for emergency situations.

III. PROPOSED MODEL

In this section, we first present the model architecture, then we describe the adaptive delay-aware scheme for data scheduling and routing, which consists of two aspects, namely the queuing process and the next-hop selection.

A. NETWORK MODEL

The proposed model, as shown in Fig. 1, is composed of a wireless sensor network inspired by the (PRDBS) [16] model and which uses software-defined network technology. This is a hybrid architecture where the sensors are grouped into clusters around a cluster head (CH) and the clusters are arranged into levels. The CHs are themselves virtually organized hierarchically according to the number of hops that separate each from the base station. Thus, a CH at level 1 is one hop away from the base station and one level higher or closer than another CH at level 2. To reach the base station, a packet generated in a specific cluster will first be sent to the related CH; it will then be forwarded to one of the reachable CHs located at the immediate upper level, and so forth until it reaches the base station. At network initialization, sensors are identified, levels are established, clusters are formed, and cluster heads are assigned. Clustering is a technique widely adopted for reducing communication overhead in large-scale networks [39]. It enables a significant reduction in the number of nodes that need to communicate directly with the base station by allowing each sensor node to communicate with its respective CH, which then forwards the data to the base station. The hierarchical organization of a clustered network provides scalability and makes it possible to efficiently handle a vast number of nodes. Moreover, the hierarchical structure also offers redundancy and fault tolerance. In the event of a cluster member node failing within a cluster, neighboring nodes can continue to ensure monitoring of the area. Similarly, if a CH fails, neighboring nodes or higher-level CHs can take over its responsibilities depending on the chosen strategy. Hence, hierarchical clustering is an efficient and reliable method for collecting data in large-scale environmental monitoring projects.

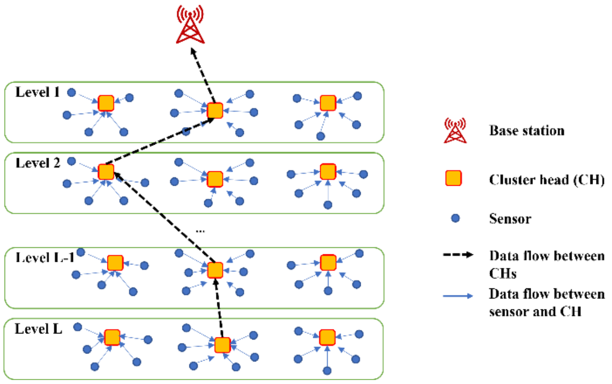

FIGURE 1. **Proposed network model.**

B. PACKET CLASSISFICATION AND PRIORITIZATION

Based on criticality, packets are divided into three categories, with the first one being the most urgent data and the last one being the least urgent data. Based on origin, we establish two categories when studying packet movement through a CH: the data detected locally, meaning by a sensor belonging to the same cluster as the CH, and those generated remotely, meaning at a level lower than the CH's level. Combining these two criteria, we establish four classes of packets: class 1 is for packets of criticality 1, whether or not they are detected locally or remotely; class 2 is for packets of criticality 2, whatever their origin; class 3 is for packets of criticality 3, sent from lower-level CHs; and class 4 is for local packets of criticality 3, received from nodes belonging to the same cluster as the CH. Priorities are assigned, as depicted in Table II, to the packets according to the classification explained: priority 1 and priority 2 are given respectively to class 1 and class 2 packets; priority 3 and priority 4 are assigned to class 3 and class 4 packets, respectively. The cluster head is responsible for managing the priority using the queues.

TABLE II
VARIOUS LEVELS OF PACKET PRIORITY

Priority	Criticality	Origin
Very high (1)	Urgent (1)	Any
High (2)	Medium (2)	Any
Medium (3)	Low (3)	Remote cluster
Low (4)	Low (3)	Local cluster

For the transmission of packets from the sensors to the cluster head, a timeslot is assigned to each sensor of the cluster consecutively. Even though energy management is not in the scope of this research work, sensor energy consumption is tracked using some reference formulas in [40]. When a sensor member of a cluster is dead, the packets in its buffer are lost, and it is no longer scheduled for timeslot assignment.

C. ADAPTIVE PACKET QUEUING

The increased usage demands immediately following a disaster decrease network performance, causing congestion and packet loss. The adaptive packet queuing method aims at minimizing data loss. In this section, we present the packet reordering method, the packet contingency migration strategy, and the queue service process.

1) ADAPTIVE PACKET REORDERING

Within each CH node is implemented, with preemption, a system of four queues, each corresponding to one of the priority levels defined in the previous section. These are the default queues. A new packet arriving at a CH node is placed in the queue matching its priority.

To promote fair transmission, we perform the reordering of packets upon the arrival of a new one in a queue according to a series of criteria presented below. In other words, the arriving packet is compared to each packet in the receiving queue to find the correct position at which it should be inserted.

DEADLINE-BASED CRITERION

Packet deadline is an important characteristic to explore for packet scheduling in applications that are time-sensitive such as in disaster monitoring. The packet with the shortest deadline should be transmitted first, as this criterion aligns perfectly with the objective of delivering packets before the exhaustion of their time limit. However, as two packets with the same deadline are progressing toward the base station, how can we better discriminate them based on the direction of the evolution of their situation in terms of elapsed time?

As it is established that our network is organized by level as mentioned in Section III.A, we define d as the packet global deadline, n as the level at which the packet is generated, p as the level where the packet currently is, with $p \leq n$, and a as the time the packet has lived so far. At any moment, the remaining time before a packet reaches its global deadline represents its updated deadline; it depends on the time already lived and is expressed as *(d-a)*. Therefore, knowing that as a packet moves toward the base station, it goes from one level to another, and by bringing the analysis back to the current level where the packet is located, we define, respectively, in (1) and (2) the initial per-level deadline, noted *IPLD*, and the updated per-level deadline, noted *UPLD*. *IPLD* represents the initial time provision a packet has for completing each level to reach the base station before its global deadline expiration; its value is fixed but may be different for each packet depending on the application settings. *UPLD* denotes the updated time provision for each remaining level as time passes; its value can be smaller or greater than the packet *IPLD* and depends on how quickly a packet has gone through previous levels.

$$IPLD = d / n \tag{1}$$

$$UPLD = (d - a) / p \tag{2}$$

Let's analyze a queue containing a certain number M of packets, each packet being at position i, with $1 \leq i \leq M$. If we chose the deadline (or the *IPLD*) as the reordering criterion and gave precedence to the packet presenting the lowest value, at the arrival of a new packet X, we would compare d_X with d_i (or $IPLD_X$ with $IPLD_i$). If $d_X < d_i$ (or $IPLD_X < IPLD_i$), packet X is inserted into the queue just before the packet at position i. Now, let's ponder this additional information: for packet X, we have $IPLD_X < UPLD_X$, meaning that its situation deadline-wise has improved; and for packet at position i, let's say we have $IPLD_i > UPLD_i$, meaning that its situation deadline-wise has worsened. By inserting packet X before the packet at position i, we have made the condition of the latter even worse and increased its chances of being dropped. Invertedly, given the previous additional information, putting packet X after the packet at position i in the queue amounts to keeping in view the evolution of the condition of both packets.

The direction of the evolution of the situation of a packet in relation to its deadline can be translated by the $UPLD/IPLD$ ratio. When its value is below 1, the situation has worsened, and when it is greater than 1, the condition has improved. To maintain a certain fairness in packet reordering, the hypothesis is that it is better to compare packets based on the $UPLD/IPLD$ ratio – which reflects the evolution of the packet in the network – than just the deadline or the IPLD itself, which have a fixed value.

PACKET ORIGIN

The origin level of a packet is the level at which it is generated in the network, namely the same level where the generating sensor is located. The origin level tells how many hops separate the related cluster head from the base station. A packet of distant origin in terms of hops count is subject to a longer path than another one sensed at a higher level of the network. This increases the risk of being discarded before reaching the base station. Hence the need to give precedence to the one coming from afar in the queue.

PACKET SIZE

The transmission time of a packet is proportional to its size. A packet of a smaller size will take less time to be put on the transmission link. Therefore, it should be transmitted before a packet with a larger size. This strategy ensures that, of two packets having the same remaining short time on their deadline, at least one can be saved.

ORDER OF ARRIVAL

The order of arrival refers to the time at which a packet enters a queue. In the case of a tie between two packets with respect to another criterion, precedence could be given to the one that arrived first in the queue according to the principle of First Come, First Served (FCFS).

Regarding the packet characteristics presented in the previous paragraphs, the strategy is to use them as a series of ordered criteria of which only the first one is guaranteed to be applied in the packet reordering process; the subsequent ones will be invoked individually and as needed to break ties related to the precedent criteria. Using $UPLD/IPLD$ ratio as the first criterion and the origin level as the second criterion in the case of tie could increase the delay for the packets coming from further away. However, there is a chance that the latter will begin to experience an increasingly reduced $UPLD/IPLD$ as they progress along their journey in case of high packet traffic, which would tend to work in their favor regarding their positioning in the queues. On the other hand, using the origin level as the first criterion and the $UPLD/IPLD$ ratio as the second criterion makes it possible to reduce the delay for packets coming from further away at all times. A packet generated one hop from the base station may experience difficulty being routed there. Therefore, what we gain in terms of time on the average delay for packets coming from far away, we risk losing it on the side of those that are closer to the base station. Another hypothesis is that it is better to have $UPLD/IPLD$ ratio as the first criterion for packet reordering and keep the origin level as the second in case of a tie. The packet size will be used as the third criterion, whereas the time of arrival will serve as the fourth criterion. The packets are ordered in the queues according to these criteria, as described by Algorithm 1, and then transferred to the next node – a CH at the next upper level – according to the order of priority of the queue hosting them.

Algorithm 1 Adaptive packet reordering

Require: *Pkt* ▷ Packet to be inserted
Require: *Elt* ▷ Packet already in queue
Require: *NumElements* ▷ Number of packets in queue

if Queue is empty **then**
  Insert *Pkt* at position $i = 1$.
else
  $i \leftarrow 1$
  $a,b \leftarrow$ UPLD/IPLD ratio of *Pkt*, UPLD/IPLD ratio of Elt_i
  $c,d \leftarrow$ Origin Level of *Pkt*, Origin Level of Elt_i
  $e,f \leftarrow$ Size of *Pkt*, Size of Elt_i
  $g,h \leftarrow$ ToA of *Pkt*, ToA of Elt_i
  while $\neg ((a < b) \vee ((a == b) \wedge (c < d)) \vee ((a == b) \wedge (c == d) \wedge (e < f)) \vee ((a == b) \wedge (c == d) \wedge (e == f) \wedge (g < h)) \wedge j < NumElements$ **do**
    $i \leftarrow i + 1$
    $b \leftarrow$ UPLD/IPLD ratio of Elt_i
    $d \leftarrow$ Origin Level of Elt_i
    $f \leftarrow$ Size of Elt_i
    $h \leftarrow$ ToA of Elt_i
  end while
  if $\neg ((a < b) \vee ((a == b) \wedge (c < d)) \vee ((a == b) \wedge (c == d) \wedge (e < f)) \vee ((a == b) \wedge (c == d) \wedge (e == f) \wedge (g < h))$ **then**
    $i \leftarrow i + 1$
  end if
  for $j \leftarrow NumElements : -1 : i$ **do**
    $Elt_{j+1} \leftarrow Elt_j$
  end for
  Insert *Pkt* at position i.
end if

2) ADAPTIVE PACKET CONTINGENCY MIGRATION

To mitigate contingency situations that may arise during a packet lifecycle, we devised the contingency migration

This article has been accepted for publication in IEEE Access. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2024.3407672

strategy, as presented in Algorithm 2, that is based on the fallback queues, the definition of a critical situation, the migration conditions, and the migration coefficient calculation.

Algorithm 2 Adaptive contingency migration

Require: Pkt ▷ Packet in queue
Require: N ▷ Number of packets in queue
$i \leftarrow 1$
while $i \leq N$ **do**
  $AD_{CL} \leftarrow$ average delay of packets similar to Pkt_i
  $Age \leftarrow$ age of Pkt_i
  $Deadline \leftarrow$ deadline of Pkt_i
  $H_{completed} \leftarrow$ number of hops completed by Pkt_i toward BS
  $H_{initial} \leftarrow$ number of initial hops needed for Pkt_i to reach BS
  $MC_P \leftarrow AD_{CL}/Deadline$ ▷ Migration coefficient of Pkt_i
  $UPLD \leftarrow$ updated per level deadline of Pkt_i
  $IPLD \leftarrow$ initial per level deadline of Pkt_i
  if $Age / Deadline > H_{completed} / H_{initial}$ **then**
    if $MC_P \times IPLD \leq UPLD$ **then**
      Migrate Pkt_i to fallback queue
    else
      Drop Pkt_i as lost case
    end if
  end if
  $i \leftarrow i + 1$
  $N \leftarrow N - 1$
end while

FALLBACK QUEUES

The four default queues within a CH node ensure the transfer of data according to its priority, which is linked to its criticality. The series of criteria chosen for the dynamic and adaptive reordering in the queues reinforce the contextual processing of each packet and contribute to its arrival at its destination. However, certain situations, such as heavy data flow or technical failure, can still make it difficult to deliver packets. This concern leads to envisage having one fallback queue (FQ) per CH for packet migration purposes, in critical situations, to avoid packet loss as much as possible. This fallback queue can receive packets from any other queue within the same CH, has the highest priority during queue service, and its packets are transmitted on an FCFS basis. Once its packets are dispatched, it is disabled.

PACKETS IN CRITICAL SITUATION

As a packet makes its way toward the base station, it is expected that its progression on the path will be proportional to time progression since its generation. A packet whose time progression would noticeably be greater than its path progression is regarded to be in a critical situation. This is expressed by (3), where a is the packet age, d is the packet deadline, $h_{completed}$ is the number of completed hops toward the base station, and $h_{initial}$ is the initial count of hops necessary to reach the base station.

$$a / d > h_{completed} / h_{initial} \tag{3}$$

MIGRATION CONDITION

Equation (3) is a necessary but not sufficient condition for migrating a packet to the fallback queue. Indeed, at the time of said migration, a packet may already have no chance of reaching the base station in time, considering both the path it still has to travel and its life expectancy from that moment on. To prevent such a packet from continuing to use resources unnecessarily and to speed up the transmission of those that follow, it might be better to drop the packet once it has been identified as a lost case.

In a sensor network of n levels, let $T_{RemGlob}$ be, at any moment, the time remaining to a packet before its global deadline expiration, and T_{RemCL} represent the time limit to spend at the current level to have the chance to reach the base station. T_{RemCL} is not a fixed value; it evolves with the age of the packet. On the other hand, let T_{ToBS} be the time it will take for the packet to reach the base station, and T_{ToNL} be the time that the packet will have to spend at the current level before being transmitted to the next level. Ideally, as expressed in (4) T_{ToBS} should not exceed the overall remaining time to avoid packet loss. Likewise, as stated by (5), T_{ToNL} should not go over the provisioned time for the level.

$$T_{ToBS} \leq T_{RemGlob} \tag{4}$$

$$T_{ToNL} \leq T_{RemCL} \tag{5}$$

The variations in waiting times in the queues can lead to having $T_{ToNL} \leq$ IPLD or $T_{ToNL} >$ IPLD. The IPLD being constant for a packet, the value of T_{ToNL} can be expressed as a function of it as in (6), with k' belonging to $\mathbb{R}_+^*$. Then, equation (5) is updated to (7).

$$T_{ToNL} = k' * IPLD \tag{6}$$

$$k' * IPLD \leq UPLD \tag{7}$$

Equation (7) states that the updated time limit for the current level T_{RemCL}, represented by the UPLD, must be greater than some proportion of the initial time limit per level (IPLD) for the packet to have the chance to reach the destination before the global initial time limit (the deadline) runs out. The coefficient k' represents this proportion, which is called to change according to the packet waiting time in the previous levels. Let k be the minimum value of the coefficient k' for which the packet no longer has any chance of reaching the destination on time. From (7), we have the following:

$$k * IPLD \leq k' * IPLD \leq UPLD \tag{8}$$

$$k * IPLD \leq UPLD \tag{9}$$

Equation (9), where k represents the migration coefficient (MC), is the second condition that must be met by a packet for its migration to the fallback queue. If it is not met, the packet will be discarded.

MIGRATION COEFFICIENT CALCULATION

As described in the previous section, the migration coefficient is the minimum value the UPLD/IPLD ratio can have for a packet migration to take place. It is possible to set this value

This article has been accepted for publication in IEEE Access. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2024.3407672

empirically to a fixed and unique number, but it would not be reflective either of the varying conditions of the network or of the experience of the packets in the network. Besides, the criteria to select the best value would have to be defined and would probably depend on the type of application the network is built for. Additionally, as illustrated in Table III, a sensitivity analysis of the migration coefficient that we ran was not very conclusive about the best threshold value or interval for this parameter that would support the achievement of both reduced delays and reduced packet loss.

TABLE III
SENSITIVITY ANALYSIS OF MIGRATION COEFFICIENT VALUES

Values (decimal)	Delivery Ratio (> 0.9)	Average Delay (kept to minimum)
0.125	Yes	No
0.25	Yes	No
0.375	Yes	No
0.5	Yes	No
0.625	No	Yes
0.75	No	Yes
0.875	No	Yes

Therefore, we propose not to establish a default MC value for all packets, but rather compute it for each packet based on the network experience of similar packets. Let us examine a packet *P*, of criticality *C*, generated at level *L*, and with a deadline *d*. Let us denote by AD_{CL} the current average delay of all delivered packets of the same criticality and generated at the same level as packet *P*. The migration coefficient for packet *P* is given by (10):

$$MC_P = AD_{CL} / d \tag{10}$$

3) QUEUE SERVICE
A queue is served when its data is transmitted. The priority of the queues determines the order in which they are served, with the highest priority being priority 0 when the fallback queue is activated, and 1 in the case of only the four default queues. In the absence of packets in a queue of higher priority, the arrival of packets in any queue (of priority j=0, …, p-1) can preempt the transmission of data from all the other queues of lower priority p (p>j). Therefore, when the fallback and the priority 1 queue are empty, the arrival of data in the priority 2 queue will preempt the transmission of priority 3 or priority 4 data that was in progress.

D. ADAPTIVE NEXT-HOP SELECTION
Path selection for data packets aims to determine the best route based on network conditions to optimize data transmission and avoid latency due to congestion. It takes the packet criticality into account. It is also based on continuous monitoring and analysis of several status- and performance-related metrics for the potential next forwarding nodes, the latter being the nodes at the immediate upper level toward the base station. Following is the list of criteria used to select the next hop for any packet, as presented in Algorithm 3.

1) ENERGY AVAILABILITY
Energy is a limited resource in WSNs with nodes running on batteries. Once the battery runs out, the node is no longer operational. Therefore, even though energy management is not the purpose of this paper, it is an important aspect to think about for efficient packet scheduling and ensuring not to forward an additional packet to a node that is about to die. When choosing the next hop for a packet, the first criterion is to check that the candidate node has enough residual energy to process not only the packet to be sent, but also all the packets of the same and higher priority than the latter's that it currently holds. As stated in [36], communication is the most energy-consuming activity of a sensor node, accounting for 51%, while sensing corresponds to 6%. For the purpose of this paper, we compute node energy consumption using only the communication and sensing aspects.

2) TRAFFIC LOAD
With the rationale of reducing packet delay, sending the latter to a node that is less loaded gives a better chance for the packet to be forwarded earlier to the next-hop and finally to the base station. As explained previously, a CH node memory is organized in queues for packet priority management purposes. At packet reception, it is placed in the CH buffer and then put in the queue hosting same-priority packets. Therefore, as a second criterion, we check both queue and buffer occupancy to choose the node that is less loaded with same-priority packets both in the related queue and in the buffer.

3) SOJOURN TIME IN QUEUE
The experience of same-characteristics packets having transited through a CH is another aspect included in the analysis of the CH suitability as the next-hop for a specific packet. The sojourn time of a packet in a queue is defined as the time between its arrival and its departure from that queue. The idea is to compare the candidates forwarding nodes based on the sojourn time, in their respective queues of related priority, of packets originated at the same level and presenting the same criticality as the one to be transferred. The selected CH is the one presenting the shortest sojourn time for same-characteristics packets.

4) RESIDUAL ENERGY VALUE
Finally, in the perspective of not overworking a node, the value of the residual energy is used in the case of ties for the previous criteria, and the CH with the highest resource level is chosen.

IV. PERFORMANCE EVALUATION
In this section, we report the experimentations performed to evaluate the proposed model.

A. SIMULATION ENVIRONMENT
The model was implemented and simulated using the MATLAB development environment, version 23.2.0.2380103 (R2023b). The main hardware that was used is a laptop with the following specifications: 12th Gen Intel(R)

Core(TM) i5-1235U 1.30 GHz processor; 16.0 GB RAM; 64-bit operating system, x64-based processor.

Algorithm 3 Adaptive next-hop selection

Require: *Pkt* ▷ Packet to be sent to next level
Require: *C* ▷ Criticality of *Pkt*
Require: *L* ▷ Current level of *Pkt*
$i \leftarrow 1$
if $L = 1$ **then**
  Send *Pkt* to base station
else
  $List \leftarrow$ available CHs at level $L - 1$
  for *Elt* in *List* **do**
    $E_a \leftarrow 0 \vee 1$ status on energy availability
    $O \leftarrow$ *Elt*'s buffer and queue load of criticality C
    $S \leftarrow$ sojourn time in *Elt* of packets similar to *Pkt*
    $E_r \leftarrow$ residual energy value of *Elt*
  end for
  Order *List* by descending E_a, ascending O then S, desc. E_r
  Send *Pkt* to Elt_i of *List*
end if

B. ASSUMPTIONS

For the implementation of the model, the following assumptions are made:

- Each network node is located within a cluster and is allocated a time slot to transmit its packets to the cluster head. The length of the time slot is the same for all cluster nodes.
- Cluster head nodes are equipped with more energy resources than the regular cluster member nodes.
- The distribution of the network elements is fixed. At the launch of the network, the clusters contain the same number of sensors, and the number of clusters at each level is the same.
- The sensor coverage area is not considered in the simulations.
- No data aggregation is performed at intermediate levels.
- The data processing delay within a sensor is overlooked.

C. PERFORMANCE METRICS

The proposed model is evaluated based on the following metrics: average delay, delivery rate, and number of critical situations that arise. The average delay represents the time required to route packets from a source node to a destination node. It includes the waiting time, the transmission delay, and the propagation time. The waiting time is the time a packet spends in the queue before being transferred to the transmission medium. The transmission delay (D_{Tx}) is the time required to put the packet on the transmission medium; it is represented by the ratio between the size (Sz_P) of the packet in bits and the bandwidth (Sp_{Tx}) in bits per second, as described by (11):

$$D_{Tx} = Sz_P / Sp_{Tx} \tag{11}$$

The propagation delay (D_P) is the time it takes a packet to travel the distance between the source and destination nodes; it is determined by the ratio between the distance ($L_{src\text{-}dest}$) which separates the two nodes, and the speed of propagation (Sp_{Prop})in the medium, as described by (12):

$$D_P = L_{src\text{-}dest} / Sp_{Prop} \tag{12}$$

Delivered packets are those that have reached the destination to which they were sent. Lost packets are those that were sent to a destination that they could not reach for various reasons. Let N_S be the number of packets sent and N_R the number of packets received at the destination. The packet delivery rate is given by (13):

$$Del_{Rate} = N_R / N_S \tag{13}$$

We recall that (3) defines the critical situation for a packet. A packet may fall into a critical situation more than once during its journey from the source to the destination. For this metric, we simply count the number of critical situations that arose during the duration of the simulation, regardless of the packets or their characteristics. It helps measure the efficiency of the model in facilitating packet progress.

D. SIMULATION PLAN AND PARAMETERS

The simulation plan involves two phases. The first phase consists of short simulations repeated three times for the comparison of fifteen cases. It aims to evaluate clearly and separately the impact of the strategies of packet reordering, contingency migration, and adaptive next-hop selection, as well as some combinations of these ones. Table IV illustrates the cases examined in the first phase of simulations.

Case 1 represents the baseline against which the performance of the others is compared. Cases 2 to 5 only evaluate the performance of the packet reordering strategies using different orders for a series of reordering criteria. Cases 6 to 8 merely assess the efficiency of the migration strategies using whether a unique fallback queue served with the highest priority when activated, or three FQ with different priorities in service. Cases 9 to 11 solely measure the impact of the next-hop selection strategies involving the average packet sojourn time in queue, the CH occupancy in terms of same-priority packets, or an ordered series of relevant criteria. Cases 12 to 15 combine the best-performing strategies from each of the previous categories. The second phase entails longer simulations repeated five times for the comparison of the most efficient strategies for each category among those evaluated during the previous phase. Table V presents the simulation parameters for both phases:

- We supposed that not every sensor senses the data at the same time for different reasons: network energy saving in case of having redundant nodes, different types of sensors having different minimum warmup times, and sensing intervals, among others. So, during a timeslot representing a fraction of a second, only a fraction of the whole sensor population will capture data concurrently.

The percentage of network generating packets in each slot has been chosen empirically, based on the abovementioned reasons and our computational constraints for the simulations.

- The migration coefficient parameter is part of the proposed method, and its value is computed during the simulation according to (10).
- We considered 3 different values for the packet size to introduce more variability and heterogeneity in the network data flow. The size of 512 bytes is commonly found in the literature [13], [16].
- We made the supposition of an infinite queue because we wanted to evaluate the performance of the proposed model and explore how the system behaves under ideal conditions without the queue size as a limiting factor causing potential packet loss in case of queue overflow.
- Packets of each criticality level have an equal chance of being generated, leading to a balanced representation of criticality levels in the simulated packet data. Due to the stochastic nature of the process, the exact counts may vary slightly but each category sensibly represents approximately a third of the total number of packets generated during a simulation. If criticality is associated with sensor purpose (e.g. temperature, smoke, or fire sensor), we may assume that the monitored area could be equally equipped with all types. Therefore, the same reading intervals might have been set for all three types of sensors, which would result in a balanced representation of criticality levels in the simulated packet data. If criticality is associated with the extent to which the sensed data exceed preset threshold values regardless of the type of sensor, ensuring that packets of each criticality level have an equal chance of being generated helps test the model efficiency under various conditions and not focusing disproportionately on one category of criticality levels, which could lead to biased results.
- The values chosen for the timeslot duration and the bandwidth are similar to those found in the literature [16].
- The values selected for the size of the network as well as the simulation length and numbers have been chosen empirically, keeping in mind our computational constraints for the simulations. Regarding the network size, during phase 1, for a moderately dense situation that would help in selecting the best strategies that will be tested further in phase 2, we had 5 levels of 10 clusters of 10 sensors. It means that the farthest cluster heads are at 5 hops from the base station, each cluster head has 10 options for forwarding its data to the next level towards the base station, and 3000 packets are generated in one second. Whereas for phase 2, we proceed with a range of different sizes for the network.

TABLE IV
DESCRIPTION OF SIMULATION CASES

Case	Packet reordering	Migration	Next-hop selection criteria
Case 1	None (FIFO)	None	Queue occupancy
Case 2	Deadline → Origin → Size → ToA	None	Queue occupancy
Case 3	UPLD/IPLD→ Origin → Size → ToA		
Case 4	Origin → Deadline → Size → ToA		
Case 5	Origin → UPLD/IPLD → Size → ToA		
Case 6	None (FIFO)	With 1 FQ (service position: 1)	Queue occupancy
Case 7		With 3 FQ (service positions: 1-3-5)	
Case 8		With 3 FQ (service positions: 1-2-3)	
Case 9	None (FIFO)	None	Avg. sojourn time in dedicated queue
Case 10			Occupancy of CH for same criticality pkts
Case 11			Energy availability → CH occupancy → Sojourn time → Residual energy
Case 12	None (FIFO)	Best between cases 6, 7, 8	Best between cases 9,10, 11
Case 13	Best between cases 2, 3, 4, 5	None	Best between cases 9,10, 11
Case 14	Best between cases 2, 3, 4, 5	Best between cases 6, 7, 8	Queue occupancy
Case 15	Best between cases 2, 3, 4, 5	Best between cases 6, 7, 8	Best between cases 9,10, 11

E. SIMULATION RESULTS

In this section, we present and analyze the results of the simulation of the proposed model. The final proposed solution is represented by case 15. It combines packet classification and prioritization, adaptive packet queueing (which incorporates the packet reordering, the contingency migration and the preemptive queue service), and the adaptive next-hop selection as described in Section III.

TABLE V
SIMULATION PARAMETERS

Parameters	Phase 1	Phase 2
Percentage of network generating packets in each slot	3%	
Migration coefficient	Adaptive	
Packet size	512, 1024 and 2048 bytes	
Queue size	Infinity	
Packets generation rate according to their criticality	Criticality 1→ 1, Criticality 2→ 1 Criticality 3→ 1	
Bandwidth	2 Mbps	
Duration of timeslot	5 ms	
Quantity of sensors, clusters, and levels in network	500 – 50 – 5	[100-10-5] to [700-70-5]
Simulation time	5000 ms	20000 ms
Number of simulations	3	5

1) PERFORMANCE OF THE PROPOSED STRATEGIES

For this first phase of simulations, the number of sensors is set to 500, the quantity of clusters is maintained at 50 with 10 nodes per cluster, and the depth of the network is fixed at 5

This article has been accepted for publication in IEEE Access. This is the author's version which has not been fully edited and
content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2024.3407672

levels. We compare cases 1 to 15 based on the metrics introduced in Section IV.C.

Fig. 2 depicts the performance in terms of average delay. Among the proposed packet reordering strategies, case 3 is the most efficient in terms of delay reduction when compared to the baseline (case 1). Specifically, the results demonstrate that having the UPLD/IPLD ratio (case 3) instead of the deadline (case 2) as the first criterion is better and outperforms cases 4 and 5 which first compare the packets based on origin level. We also note that all the migration strategies contribute to increasing the delay, and the less aggravating one is case 6, which uses only one fallback queue. Among the next-hop selection strategies, case 11 is the best performer, with a slight advance on case 10; they both drastically reduce the average delay, but the former capitalizes on a series of criteria, including the CH occupancy. Cases 12 and 15 are the most efficient among the combined strategies.

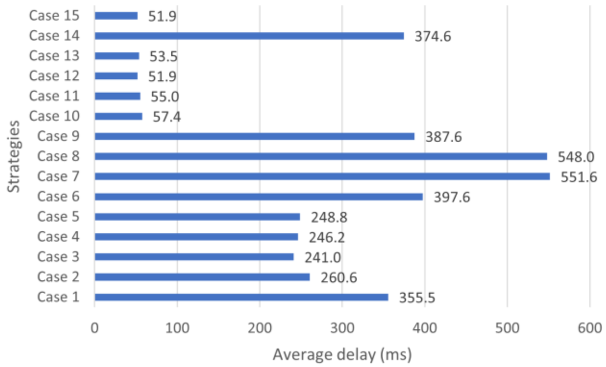

FIGURE 2. Average delay of packets for each simulated strategy for the fixed network size.

Fig. 3 presents the delivery ratio performance. We note that all the packet reordering strategies decrease the ratio, whereas all the migration strategies increase it. However, we reached the maximum delivery rate with the next-hop selection and the combined strategies.

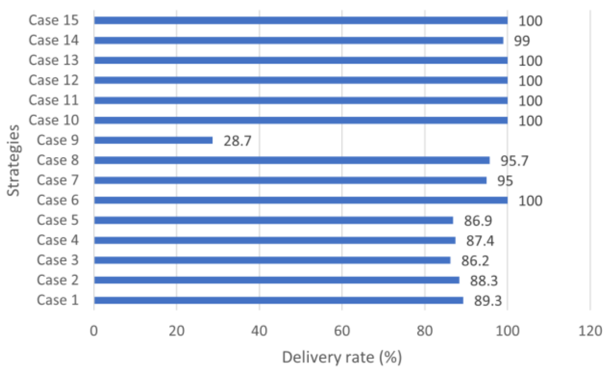

FIGURE 3. Delivery ratio of packets for each simulated strategy for the fixed network size.

We report on the critical situations number in Fig. 4. Except for case 9, which uses the average sojourn time as the next-hop selection criterion, all the strategies reduce this metric value to a certain degree; we obtain the best result with case 15, which combines the best strategies from each category.

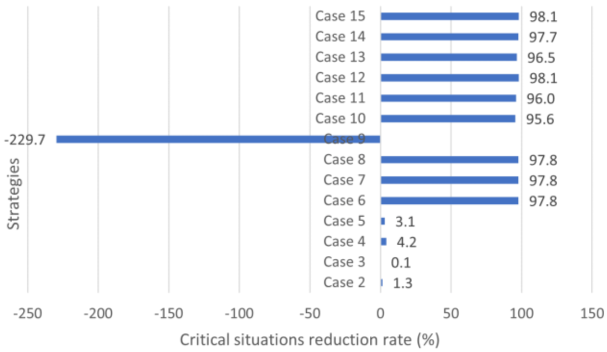

FIGURE 4. Critical situation reduction rate for each strategy for the fixed network size. A negative value indicates an increase.

For the next phase of simulations, we retained case 3 for packet reordering, case 6 for contingency migration, case 11 for next-hop selection, and case 15 for the combined strategies.

2) IMPACT OF NETWORK SIZE

For this second phase of simulations, the depth of the network is fixed at 5 levels, while varying the quantity of sensors between 100 and 700 and the number of clusters between 10 and 70, with 10 nodes per cluster. We compare the following strategies: case 1 as the baseline; case 3 as the most efficient for packet reordering; case 6 as the best one for contingency migration; case 11 as the most efficient for next-hop selection; and finally, case 15 as the best combination. The impact of the network size on the average delay is reported in Fig. 5 to Fig.8.

First, we present the results for the global average delay in Fig. 5 where we see that the selected packet reordering method (case 3) effectively reduces the numbers compared to the baseline (case 1), but with a slight tendency to increase with the network size. We note that the retained migration strategy (case 6) performs well in terms of delay only for the smallest network size and presents huge increases for the others. Both the chosen next-hop selection method (case 11) and the combined strategies method (case 15) manage not only to reduce the delay but also to decrease it along with the increase in network size.

Moreover, we perform a criticality-wise analysis of the average delay. As shown in Fig. 6 to Fig. 8, the proposed strategies, whether individually or combined, contribute to its reduction for packets of criticality 1, 2 and 3, except for case 6. The latter, however, is more efficient in terms of delay for criticality 3 packets.

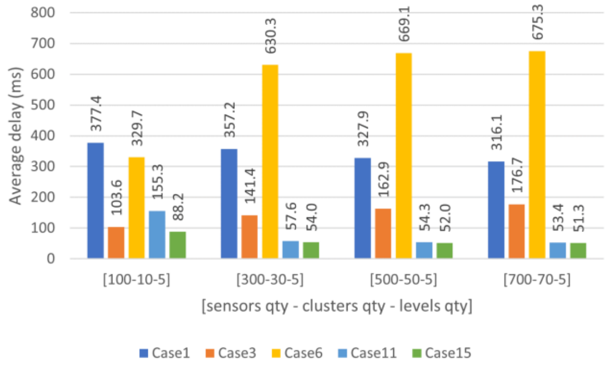

FIGURE 5. Impact of the network size on the global average delay achieved by each strategy.

For the packet delivery rate, as presented in Fig. 9, case 3 is the only bad-performing strategy. However, the combined-strategies method is the most efficient one, showing a constant value despite the increase in the network size.

Contingency migration addresses the packet loss issue, essentially trying to provide a fast lane to packets that need it and that are anticipated to be able to complete the track. We observe, indeed, its efficiency in the packet delivery ratio. However, we note that, compared to case 1, this selected contingency migration strategy (case 6) raises the global average delay, whereas packet loss lessens. This can be explained by its efficiency at saving packets from being dropped, thus causing a growth in overall traffic on the network. The network being more crowded causes packets to experience increased queuing delays at nodes or along the communication path toward the destination. Although global delay increases with case 6, we note an unexpected reduction in delay for criticality 3 packets. Although no packet reordering or migration is applied in case 1, the queue service still favors criticality 1 packets, ensuring they are transmitted first to meet their deadlines, before processing criticality 2 packets. Because of that, criticality 3 packets may experience delays due to their lower priority, leading to some of them missing their deadlines. The formula for the migration coefficient that influences the migration decision, given by (10), is the ratio of the current average delay of all delivered alike packets to the packet's deadline. As the deadline for criticality 3 packets is set greater than that of others, the value of the ratio for their migration coefficient tends to be smaller, giving more opportunity for criticality 3 packets to be migrated to the fast lane (the fallback queue) using the condition in (9). This contributes to reducing their delay and eventually increasing that of other criticality packets. Because of that, lower-priority packets may be regarded as provided with more resources, ensuring their timely processing and transmission, and we can say that a certain level of fairness is achieved toward them through the migration process.

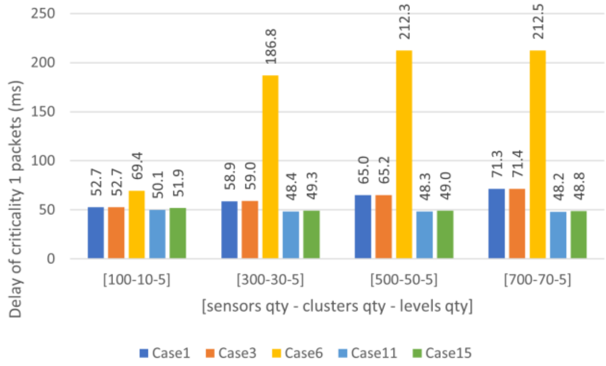

FIGURE 6. Impact of the network size on the average delay achieved by each strategy for packets of criticality 1.

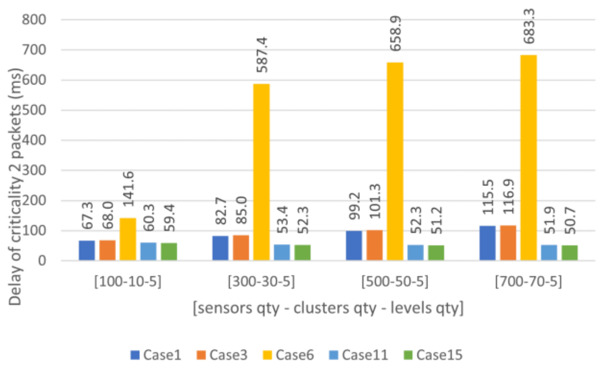

FIGURE 7. Impact of the network size on the average delay achieved by each strategy for packets of criticality 2.

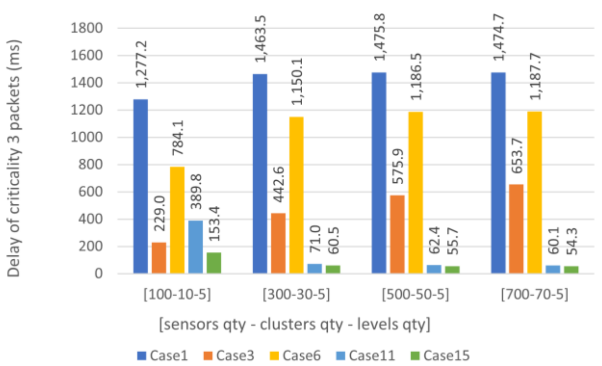

FIGURE 8. Impact of the network size on the average delay achieved by each strategy for packets of criticality 3.

The number of critical situations arisen during simulations is reduced by all strategies, as shown in Fig. 10, the most efficient being case 15. A packet may be in critical situation several times during its journey. Moreover, we noted a great reduction of the maximum number of times where a packet would be candidate for migration in cases 6, 11, and 15.

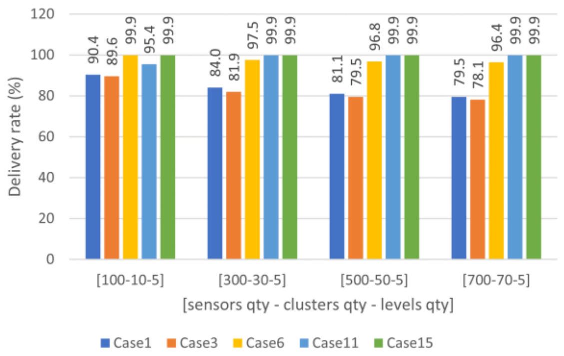

FIGURE 9. **Impact of the network size on the delivery rate achieved by each strategy.**

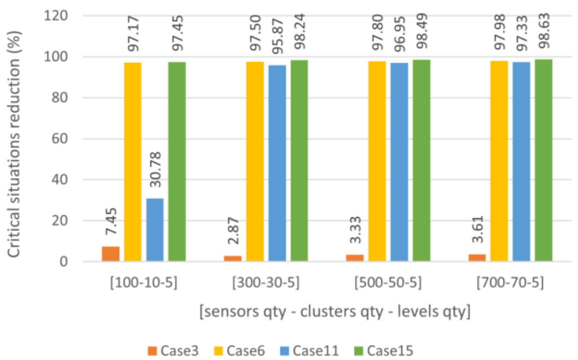

FIGURE 10. **Impact of the network size on the amount of critical situation reduction achieved each strategy.**

V. CONCLUSION

In this paper, we proposed an adaptive scheduling method for IoT data in smart cities keeping in view the multilevel priority of the latter. We designed a model to reduce transmission delays, avoid data loss, and improve overall network performance while favoring urgent data. The proposed model classifies, prioritizes, queues, and transfers packets according to criticality and origin, the two criteria that define the priority levels. It performs the adaptive scheduling of packets through three main strategies supported by preemptive queue service in favor of high-priority data: adaptive packet reordering, contingency migration, and network-experience-based next-hop selection. The first strategy is applied when inserting the packets into the default queues, according to a set of four successively applied criteria: UPLD/IPLD ratio, origin, size, and order of arrival. The second uses an additional queue called the fallback queue for the migration of packets falling into a critical situation but not being a lost case. This is accomplished using the migration coefficient value that is dynamically computed according to the previous experience of similar packets in the network. The third strategy invokes another set of four successively applied criteria: the energy status, the node overall occupancy, the sojourn time of alike packets in the targeted node, and the residual energy value.

We evaluated our work through simulation using the MATLAB (R2023b) development environment, with a focus on the impact on the model of the size of the network by varying the number of sensors and clusters. We achieved an average delay of 48.8 milliseconds for the most urgent data (criticality 1) and 54.3 milliseconds for the less urgent ones (criticality 3) for the biggest quantity of nodes in the network; these numbers respectively represent an improvement of 31% and 96% compared to the baseline method. The lowest delivery rate observed is 99.9%. The results show that the proposed model avoids data loss and guarantees the lowest delays to the most urgent data. The overall network experience is improved by effectively using parameters that reflect the variation in network conditions, such as the UPLD/IPLD ratio, the migration coefficient, the node occupancy, or the average sojourn time of packets with similar characteristics.

Extensions of this work could include the following aspects: 1) the model adaptation to varying levels of traffic in the network; 2) the dynamic adjustment of the model parameters using machine learning algorithms; and 3) the combination of the model with an energy-efficient node deployment and scheduling approach for retaining coverage and connectivity across the network and make it more resilient to node failure.

AKNOWLEDGMENT

This work was funded by the Natural Sciences and Engineering Research Council of Canada (NSERC), Prompt, Flex Group, and ISAME.

REFERENCES

[1] L. B. Elvas, B. M. Mataloto, A. L. Martins, and J. C. Ferreira, “Disaster Management in Smart Cities,” *Smart Cities*, vol. 4, no. 2, Art. no. 2, Jun. 2021, doi: 10.3390/smartcities4020042.
[2] Department of Economic and Social Affairs, “2018 Revision of World Urbanization Prospects,” United Nations, 2018. [Online]. Available: https://www.un.org/development/desa/publications/2018-revision-of-world-urbanization-prospects.html
[3] A. Sinha, P. Kumar, N. P. Rana, R. Islam, and Y. K. Dwivedi, “Impact of internet of things (IoT) in disaster management: a task-technology fit perspective,” *Ann. Oper. Res.*, vol. 283, no. 1, pp. 759–794, Dec. 2019, doi: 10.1007/s10479-017-2658-1.
[4] N. Nasser, L. Karim, et T. Taleb, « Dynamic Multilevel Priority Packet Scheduling Scheme for Wireless Sensor Network », *IEEE Transactions on Wireless Communications*, vol. 12, no. 4, p. 1448-1459, Apr. 2013, doi: 10.1109/TWC.2013.021213.111410.
[5] V. Akila, T. Sheela, et G. A. Macriga, « Efficient packet scheduling technique for data merging in wireless sensor networks », *China Communications*, vol. 14, no. 4, p. 35-46, Apr. 2017, doi: 10.1109/CC.2017.7927575.
[6] T. Qiu, R. Qiao, et D. O. Wu, « EABS: An Event-Aware Backpressure Scheduling Scheme for Emergency Internet of Things », *IEEE Transactions on Mobile Computing*, vol. 17, no. 1, p. 72-84, Jan. 2018, doi: 10.1109/TMC.2017.2702670.
[7] T. Qiu, K. Zheng, M. Han, C. L. P. Chen, et M. Xu, « A Data-Emergency-Aware Scheduling Scheme for Internet of Things in Smart Cities », *IEEE Transactions on Industrial Informatics*, vol. 14, no. 5, p. 2042-2051, May 2018, doi: 10.1109/TII.2017.2763971.
[8] S. Hammoudi, Z. Aliouat, et S. Harous, « Challenges and research directions for Internet of Things », *Telecommun Syst*, vol. 67, no. 2, p. 367-385, Feb. 2018, doi: 10.1007/s11235-017-0343-y.
[9] N. Maalel, E. Natalizio, A. Bouabdallah, P. Roux, et M. Kellil, « Reliability for Emergency Applications in Internet of Things », in *2013 IEEE International Conference on Distributed Computing in*

Sensor Systems, May 2013, p. 361-366. doi: 10.1109/DCOSS.2013.40.

[10] L. Farhan, L. Alzubaidi, M. Abdulsalam, A. J. Abboud, M. Hammoudeh, et R. Kharel, « An efficient data packet scheduling scheme for Internet of Things networks », in *2018 1st International Scientific Conference of Engineering Sciences - 3rd Scientific Conference of Engineering Science (ISCES)*, Jan. 2018, p. 1-6. doi: 10.1109/ISCES.2018.8340518.

[11] T. Qiu, K. Zheng, H. Song, M. Han, et B. Kantarci, « A Local-Optimization Emergency Scheduling Scheme With Self-Recovery for a Smart Grid », *IEEE Transactions on Industrial Informatics*, vol. 13, no. 6, p. 3195-3205, Dec. 2017, doi: 10.1109/TII.2017.2715844.

[12] L. Farhan, R. Kharel, O. Kaiwartya, M. Hammoudeh, et B. Adebisi, « Towards green computing for Internet of things: Energy oriented path and message scheduling approach », Sustainable Cities and Society, vol. 38, p. 195-204, Apr. 2018, doi: 10.1016/j.scs.2017.12.018.

[13] M. Natarajan et S. Subramanian, « A cross-layer design: energy efficient multilevel dynamic feedback scheduling in wireless sensor networks using deadline aware active time quantum for environmental monitoring », International Journal of Electronics, vol. 106, no. 1, p. 87-108, Jan. 2019, doi: 10.1080/00207217.2018.1501615.

[14] K. Kavitha et G. Suseendran, « Priority based Adaptive Scheduling Algorithm for IoT Sensor Systems », in 2019 International Conference on Automation, Computational and Technology Management (ICACTM), Apr. 2019, p. 361-366, doi: 10.1109/ICACTM.2019.8776691.

[15] V. Gabale, B. Raman, P. Dutta, and S. Kalyanraman, “A Classification Framework for Scheduling Algorithms in Wireless Mesh Networks,” IEEE Commun. Surv. Tutor., vol. 15, no. 1, pp. 199–222, First 2013, doi: 10.1109/SURV.2012.022412.00068.

[16] N. Mahendran, S. Shankar, et T. Mekala, « EMA-PRBDS: efficient multi-attribute packet rank based data scheduling in wireless sensor networks for real-time monitoring systems », *International Journal of Electronics*, vol. 107, no. 6, p. 879-898, Nov. 2019, doi: 10.1080/00207217.2019.1692244.

[17] M. E. E. D. Abd El Kader, A. A. A. Youssif, and A. Z. Ghalwash, “Energy Aware and Adaptive Cross-Layer Scheme for Video Transmission Over Wireless Sensor Networks,” *IEEE Sens. J.*, vol. 16, no. 21, pp. 7792–7802, Nov. 2016, doi: 10.1109/JSEN.2016.2601258.

[18] R. Sharma, N. Kumar, N. B. Gowda, and T. Srinivas, “Packet Scheduling Scheme to Guarantee QoS in Internet of Things,” *Wirel. Pers. Commun.*, vol. 100, no. 2, pp. 557–569, May 2018, doi: 10.1007/s11277-017-5218-8.

[19] IEEE Standard Association, “IEEE Standard for a Real-Time Operating System (RTOS) for Small-Scale Embedded Systems,” IEEE Std 2050-2018, pp. 1–333, Aug. 2018, doi: 10.1109/IEEESTD.2018.8445674.

[20] M. Shariat, A. U. Quddus, S. A. Ghorashi, and R. Tafazolli, “Scheduling as an important cross-layer operation for emerging broadband wireless systems,” IEEE Commun. Surv. Tutor., vol. 11, no. 2, pp. 74–86, Second 2009, doi: 10.1109/SURV.2009.090206.

[21] IEEE Standard Association, “IEEE Standard for Broadband over Power Line Networks: Medium Access Control and Physical Layer Specifications,” IEEE Std 1901-2010, pp. 1–1586, Dec. 2010, doi: 10.1109/IEEESTD.2010.5678772.

[22] IEEE Standard Association, “IEEE Standard for Local and Metropolitan Area Network–Bridges and Bridged Networks,” IEEE Std 8021Q-2018 Revis. IEEE Std 8021Q-2014, pp. 1–1993, Jul. 2018, doi: 10.1109/IEEESTD.2018.8403927.

[23] IEEE Standard Association, “IEEE Standard for Universal Power Adapter for Mobile Devices,” IEEE Std 1823-2015, pp. 1–123, May 2015, doi: 10.1109/IEEESTD.2015.7106444.

[24] S. Mallapuram, N. Ngwum, F. Yuan, C. Lu, and W. Yu, “Smart city: The state of the art, datasets, and evaluation platforms,” in 2017 IEEE/ACIS 16th International Conference on Computer and Information Science (ICIS), May 2017, pp. 447–452. doi: 10.1109/ICIS.2017.7960034.

[25] L. Atzori, A. Iera, and G. Morabito, “Understanding the Internet of Things: definition, potentials, and societal role of a fast-evolving paradigm,” Ad Hoc Netw., vol. 56, pp. 122–140, Mar. 2017, doi: 10.1016/j.adhoc.2016.12.004.

[26] H. Atlam, R. Walters, and G. Wills, “Internet of Things: State-of-the-art, Challenges, Applications, and Open Issues,” Int. J. Intell. Comput. Res., vol. 9, pp. 928–938, Sep. 2018, doi: 10.20533/ijicr.2042.4655.2018.0112.

[27] A. Zanella, N. Bui, A. Castellani, L. Vangelista, and M. Zorzi, “Internet of Things for Smart Cities,” IEEE Internet Things J., vol. 1, no. 1, pp. 22–32, Feb. 2014, doi: 10.1109/JIOT.2014.2306328.

[28] J. Venkatesh, B. Aksanli, C. S. Chan, A. S. Akyurek, and T. S. Rosing, “Modular and Personalized Smart Health Application Design in a Smart City Environment,” IEEE Internet Things J., vol. 5, no. 2, pp. 614–623, Apr. 2018, doi: 10.1109/JIOT.2017.2712558.

[29] D. Perera, O. Seidou, J. Agnihotri, H. Mehmood, et M. Rasmy, « Challenges and Technical Advances in Flood Early Warning Systems (FEWSs) », in Flood Impact Mitigation and Resilience Enhancement, G. Huang, Éd. IntechOpen, 2020. doi: 10.5772/intechopen.93069.

[30] A. Khan, S. Gupta, et S. K. Gupta, « Multi-hazard disaster studies: Monitoring, detection, recovery, and management, based on emerging technologies and optimal techniques », International Journal of Disaster Risk Reduction, vol. 47, p. 101642, Aug. 2020, doi: 10.1016/j.ijdrr.2020.101642.

[31] L. Yang, S. H. Yang, et L. Plotnick, « How the internet of things technology enhances emergency response operations », Technological Forecasting and Social Change, vol. 80, no. 9, p. 1854-1867, Nov. 2013, doi: 10.1016/j.techfore.2012.07.011.

[32] P. P. Ray, M. Mukherjee, et L. Shu, « Internet of Things for Disaster Management: State-of-the-Art and Prospects », IEEE Access, vol. 5, p. 18818-18835, 2017, doi: 10.1109/ACCESS.2017.2752174.

[33] E. Fitzgerald, M. Pióro, et A. Tomaszwski, « Energy-Optimal Data Aggregation and Dissemination for the Internet of Things », IEEE Internet of Things Journal, vol. 5, no. 2, p. 955-969, Apr. 2018, doi: 10.1109/JIOT.2018.2803792.

[34] M. Mathew et N. Weng, « Quality of Information and Energy Efficiency Optimization for Sensor Networks via Adaptive Sensing and Transmitting », IEEE Sensors Journal, vol. 14, no. 2, p. 341-348, Feb. 2014, doi: 10.1109/JSEN.2013.2282774.

[35] F. Al-Turjman, L. Mostarda, E. Ever, A. Darwish, et N. Shekh Khalil, « Network Experience Scheduling and Routing Approach for Big Data Transmission in the Internet of Things », IEEE Access, vol. 7, p. 14501-14512, 2019, doi: 10.1109/ACCESS.2019.2893501.

[36] L. Farhan, R. Kharel, O. Kaiwartya, M. Quiroz-Castellanos, U. Raza and S. H. Teay, "LQOR: Link Quality-Oriented Route Selection on Internet of Things Networks for Green Computing," 2018 11th International Symposium on Communication Systems, Networks & Digital Signal Processing (CSNDSP), 2018, pp. 1-6, doi: 10.1109/CSNDSP.2018.8471884.

[37] U. Ullah, A. R. Shahid, M. Irfan, J. Qadir, M. Nawaz and R. Qureshi, "A Stable and Reliable Short-Path Routing Scheme for Efficient Acoustic Wireless Sensor Networks (AWSNs)," in IEEE Access, vol. 8, pp. 1458-1474, 2020, doi: 10.1109/ACCESS.2019.2962004.

[38] Yaseen FA, Alkhalidi NA, Al-Raweshidy HS. She networks: Security, health, and emergency networks traffic priority management based on ml and sdn. IEEE Access. 2022 Aug 31;10:92249-58.

[39] A. Shahraki, A. Taherkordi, Ø. Haugen, and F. Eliassen, “Clustering objectives in wireless sensor networks: A survey and research direction analysis,” Comput. Netw., vol. 180, p. 107376, Oct. 2020, doi: 10.1016/j.comnet.2020.107376.

[40] M. Halgamuge, M. Zukerman, K. Ramamohanarao, et H. Vu, « An Estimation of Sensor Energy Consumption », Progress In Electromagnetics Research B, vol. 12, p. 259-295, Jan. 2009, doi: 10.2528/PIERB08122303.

MARIE C. POGNON received the Dipl.Eng. degree in electronics engineering from the Faculté des Sciences de l’Université d’État d’Haïti and the M.Sc.A. degree in computer engineering and information technology from the Institut des Sciences, des Technologies et des Études Avancées d’Haiti (ISTEAH), Republic of Haiti. She is currently pursuing the Ph.D. degree in computer engineering with the Mobile Computing and Networking Research Laboratory (LARIM),

École Polytechnique de Montréal, Canada, where she has been a member since January 2021. Her research interests include data prioritization and queuing, wireless sensor networks, resilience in IoT networks, and smart city applications.

ALEJANDRO QUINTERO (Senior Member, IEEE) received the Engineer degree in computer engineering from Los Andes, Colombia, in 1983, the Diploma degree in advanced studies from the INPG Grenoble in June 1989, and the Doctoral degree in computer engineering from Université Joseph Fourier, Grenoble, France, in 1993. He is currently a Full Professor with the Department of Computer Engineering, Polytechnique Montréal, Canada. He has coauthored one book, as well as more than 60 other technical publications, including journal and proceedings papers. His main research interests include services and applications related to mobile computing, network security, networks infrastructures, and next generation mobile networks.

SAMUEL PIERRE (Senior Member, IEEE) is currently a Professor with the Department of Computer and Software Engineering, Polytechnique Montreal, Canada, and the Director of the Mobile Computing and Networking Research Laboratory (LARIM). He has authored or coauthored more than 600 technical publications, including papers in refereed archival journals, textbooks, patents, and book chapters. His research interests include wired and wireless communications, mobile computing and networking, cloud computing and e-learning. He was the recipient of several awards, including the Prix Poly 1873 for excellence in teaching and training (2001 and 2005), and the Knight of the National Order of Quebec in 2009. He is a Fellow of the Engineering Institute of Canada (2003) and Canadian Academy of Engineering (2008). In December 2011, he was appointed Member of the Order of Canada. In May 2014, he was the recipient of an Honorary Doctorate from University of Quebec at Trois-Rivieres (UQTR) and another from University of Quebec in Outaouais (UQO) in November 2016. In 2017, he received the El Fassi Prize from the Agence universitaire de la Francophonie (AUF) to highlight the action of a person who has exerted a significant influence through the quality of his expertise and the innovative nature of his achievements at the international level in the fields of research, training, development and international cooperation, governance and/or transfer of knowledge or skills. He received the Grand Prize for Professional Excellence from the Order of Engineers of Quebec (2020), the Gold Medal from Engineers Canada (2021), the promotion of Officer of the National Order of Quebec in 2022, the JM Ham Award from IEEE Canada (2023) for "outstanding contributions to training the next generation of scientists and engineers in the areas of mobile computing, networking and e-learning", and the Julian C. Smith Medal of the Engineering Institute of Canada "for distinguished achievements in the development of Canada" (2024).