A Novel Model Based on Window-Pass Preferences for Data-Emergency-Aware Scheduling in Computer Networks

Mahdi Jemmali, Mohsen Denden, Wadii Boulila *Senior Member, IEEE,* Gautam Srivastava *Senior Member, IEEE,* Rutvij H. Jhaveri *Senior Member, IEEE,* and Thippa Reddy Gadekallu *Senior Member, IEEE*

***Abstract*—The breakdown of vital communication infrastructures is one of the most widely common characteristics of all disasters. It can cause severe communication problems such as time delays and data loss, which cause deterioration in system performance. New techniques are needed to cope with such situations, many of which have been made possible due to the ongoing evolution of artificial intelligence technologies. In this study, we consider the case of a network consisting of several router allocation problems in situations of high priority and emergency data allocation. A novel network component called the scheduler is introduced and window constraints for routers are imposed. To solve the studied problem, four different algorithms are developed in this work. These algorithms were then applied in a particular scenario consisting of several routers and 2200 instances. In terms of the gap and running time, the proposed algorithms provide acceptable results. The best performances were achieved using the critical packet algorithm for 80% of instances with an average gap value of 0.009 and an average time of 0.209 seconds.**

***Index Terms*—Router constraints, user preferences, emergency network, window pass slot-time, Emergency data resource allocation.**

I. Introduction

Computer networks are often affected tremendously by circumstances such as natural disasters and all the conditions that may result, such as resource limitations, infrastructural damages, dynamic changes, and more. These issues can disrupt and restrict network communications, causing services to become unavailable during crucial periods. Today, disaster data management constitutes an urgent area of research. In this work, a particular network case is considered for those instances where standard techniques cannot provide the capability of monitoring and managing the process of sending data. This occurs in specific circumstances such as natural disasters, pandemics, safety crises, war, and armed conflict, or other circumstances needing exceptional transmissions, such as spying or the transmission of confidential, critical, important, or high-privilege data. In such cases, user processes may ask for different preferences before transmitting these data [1], [2]. These preferences can include window passes, data availability times, latency requirements, and other types of particular preference(s). For example, if one wishes to send specific data before others or if the sending process is meant to follow a specific plan, such situations cannot be realized using the typical processes for operation routers. To resolve this issue, a novel component called the "scheduler", which can plan out a particular transmission process is proposed. The main goal of this component is to allow users to send data according to their needs, which may be based on preference, priority, or any other selected condition.

Wang *et al.* in [3] and [4] presented several solutions and approaches that could be utilized for emergency cases and catastrophes, but ultimately they did not approve any satisfactory results. The solutions they offered require further enhancement, perhaps by understanding situations causing disaster areas. In such cases, the administrator must select the pertinent data that must be transmitted with the maximum priority. Other information should be temporarily blocked if necessary to expedite sending this priority data.

Many advanced machine learning (ML) techniques are evident in the literature in this area, where researchers demonstrate their uses in solving, enhancing, and optimizing problems related to scheduling, resource allocation, and more. Similar situations have also been recorded in research such as [5] and [6]. In particular, Jemmali *et al.* in [5] considered an NP-hard problem for a single router. The scenario described here cannot cover the case of an entire network, which is the type of solution that modern issues require.

Several works related to wireless HART and TSN technology were studied [7], [8], [9], [10], [11], [12]. The used algorithms can be applied to the studied problem.

Corresponding Author: Gautam Srivastava (email: srivastavag@brandonu.ca

Co-Corresponding Author: Mahdi Jemmali (email: m.jemmali@mu.edu.sa)

M. Jemmali is with the Department of Computer Science and Information, College of Science at Zulfi, Majmaah University, Majmaah, 11952, Saudi Arabia (e-mail: m.jemmali@mu.edu.sa; mah_jem_2004@yahoo.fr); MARS Laboratory, University of Sousse, Tunisia; Department of Computer Science, Higher institute of computer Science and mathematics, University of Monastir, Monastir, 5000, Tunisia

M. Denden is with the Department of Computer and Information Technologies, College of Telecommunication and Information Riyadh CTI, Technical and Vocational Training Corporation $TVTC$ Saudi Arabia; Department of Computer Science, Higher Institute of Applied Sciences of Sousse, Sousse University, Sousse, 4000, Tunisia

W. Boulila is with the Robotics and Internet-of-Things Laboratory, Prince Sultan University, Riyadh, Saudi Arabia; RIADI Laboratory, National School of Computer Science, University of Manouba, Manouba, Tunisia

G. Srivastava is with the Dept. of Mathematics & Computer Science Brandon University, Canada as well as with the Research Centre for Interneural Computing, China Medical University, Taichung, Taiwan. (e-mail: srivastavag@brandonu.ca)

R. H. Jhaveri is with the Department of Computer Science and Engineering, School of Technology, Pandit Deendayal Energy University, India. (e-mail: rutvij.jhaveri@sot.pdpu.ac.in)

Thippa Reddy Gadekallu is with the Department of Information Technology, Vellore Institute of Technology, India. (e-mail: thippareddy.g@vit.ac.in)

The focus of this paper is motivated by these studies and their promise as well as their shortcomings. In this work, we propose new algorithms on a model containing multiple routers to solve for the best allocation of computing resources in specific cases. Furthermore, an NP-hard problem has been considered for various routers in the case of a critical network constraint where the traffic cannot be controlled using typical techniques.

The remainder of this paper is organized as follows. Section II presents related works from the literature on machine learning (ML) and scheduling problems. Section III focuses on the motivation and contribution of the research work and gives an industrial case for the studied problem, while Section IV presents a formal description of the target problem. Section V proposes several approximate solutions, while Section VI shows experimental results with their interpretation and analysis before our conclusion in Section VII.

II. Related works

Several researchers have used scheduling solution methods and heuristics to solve resource allocation and routing problems. For instance, Jemmali and Alquhayz proposed a time slots algorithm for a critical constraint computer network. They aimed to consider a network with one router and how to develop algorithms capable of assigning preferences to priority data and block others [5] and [6].

In another work, Madni *et al.* in [13] and [14] used heuristics to control resource allocation. Here, the goal was to minimize the time and cost of resources reserved in the cloud space to achieve this work.

Meanwhile, Lin and Chong described an allocation technique in cloud computing. Their approach consists of minimizing the computation time needed, which is an essential factor that will influence the speed and potential for response to cloud users' requests [15].

In the article [4], authors presented a form of security architecture for disaster and emergencies. According to their project and results, these solutions were intended to guarantee that minimum security thresholds were met for important data.

According to Schweissguth *et al.* [16], scheduling is a complex NP-hard problem, but its complexity can be reduced by adopting specific abstractions. One typical approach is to separate routing and scheduling problems that are solved either successively or using heuristic coupling. These researchers also presented a novel integer linear program ILP formulation, which simultaneously solved several routing and scheduling problems. Multiple scheduling issues could be resolved with this formulation, such as the case of using the shortest fixed path routing with separate scheduling.

Hanzálek *et al.* [17] affirmed that there are no algorithms currently available that can solve complex NP-hard problems efficiently for a variety of topologies and different instances. Instead, a typical approach, as introduced in [17] and [18], consists of solving planning problems. This approach requires developing a formal mathematical description of all the resources and application constraints before attempting to resolve them.

Flow scheduling approaches with an exclusive routing were used in [19] in an attempt to reduce network energy consumption. However, these researchers considered global scenarios in data center networks and proposed a novel energy-efficient flow scheduling and routing algorithm for software-defined networks (SDN).

Elsewhere, Zhang *et al.* [20] proposed a novel differentiated scheduling approach that considers the link workloads to find separate paths for elephant and mice flows. This method schedules mice flow using a proactive weighted multi-path routing algorithm, while an algorithm based on path setup was used for elephant flows. To balance traffic in SDNs, they also designed an algorithm to dynamically reschedule data flows based on various links' current state.

A novel routing technique in the presence of jams in multi-hop wireless networks was proposed by Mageswari *et al.* [21]. The authors presented a novel technique that combines the best path with a routing internet protocol in the presence of jams within multi-hop wireless sensor networks. This algorithm selects routes based on the detection of jams and operates based on the QoS link.

In [22], the authors proposed an approach that generates a mobile synching environment to transfer data and packets with a positive acknowledgment. Bandwidth management and performance prediction for IoT are studied in [23], [24]. The proposed algorithms in the paper can be utilized in later works.

Meanwhile, the authors of [25] proposed a novel protocol that operates by selecting zones based on their residual energy, link-state, and metric for mobile ad hoc networks. In this approach, a selected zone is titled the zone leader (ZL). Each ZL selects multiple paths from border nodes to destinations, and traffic is distributed along these paths.

The authors of [26] used the dolphin echolocation algorithm to establish optimal links based on an adjacent-position trust verification protocol to construct a mesh-based multi-path routing scheme that can identify all possible secure paths.

In [27], the authors proposed that the age of data can provide useful criteria for measuring the performance of a network. Here, the age of any particular information age represents its freshness or how much time has elapsed since its generation, as seen by end-users. The researchers studied these criteria from a scheduling perspective using broadcast networks. Similarly, several recent applications of scheduling problems to networks and routers were studied in [28].

Summarizing the literature study, only two studies treated the problem with window-pass preferences into network [5] and [6]. These windows are time intervals that represent periods during which regular data transmissions are blocked and pass permission is granted to the administrator's critical data. These latter works studied only a single router problem which is a simple case of the generalization of the several routers problem.

In our paper, we propose a novel network architecture that is well-suited to the emergency scenarios discussed earlier, and that builds from many of the insights mentioned elsewhere in the literature. The goal was to select urgent data for classification and calculate estimated times for crossing networks. The availability of routers is a determining factor in calculating estimated times to delivery. Our novel "scheduler" component applies various algorithms to determine the optimal data allocation according to various temporal factors. The scheduler also regularly updates the global network state to refine the selection of data preferences. The administrator also specifies preferences, such as the required window for sending critical data.

The main objectives of this study can be summarized as follows: (i) to define an intelligent network architecture for an exceptional environment, (ii) to give extreme privilege to crucial and confidential data that must be transmitted in predetermined-time intervals, (iii) to guarantee high exploitation of resources, and (iv) to solve the problem of permission scheduling.

III. Motivations and contributions

In this paper, the studied problem is a generalization of an already studied problem cited in literature [5] and [6]. It is an NP-hard problem, which constitutes a difficult challenge to search for an approximate solution using several heuristics to solve the problem in a polynomial time. A heuristic among proposed ones utilizes the exact solution of the problem of one router. Indeed, in this paper, we developed an exact solution for the one router problem.

To the best of the authors' knowledge, the problem of multiple routers under user preferences constraints has not been studied in the literature. The only work in literature that treated the same problem has used only one router.

The constraints of window passes constitute the interval of priority given to the administrator in the network. These constraints make the hardness of the problem. The common scheduling techniques do not consider the proposed constraints. This is the reason that these techniques cannot be an efficient choice to be applied to the presented problem. Therefore, we developed several new heuristics dedicated to solving the studied problem. Two novel techniques are proposed in this work. The first one aims to use an exact solution for one router to solve the multiple routers problem. The second one presents the "Critical Packet" to adopt a new window pass and be considered a newly added constraint to solve the problem.

Suppose that there is an Intranet where every employee can send data through the network in an industry where the volume of data is big. In this case, the limitation of routers' resources can be a problem. To enable priority to administrators for specific traffic, the proposed architecture provides an obligation to routers to keep the free path transmission of administrator data.

IV. Problem Description

This study's goal was to solve the NP-hard network scheduling problem for a specific network with several routers. The aim was to develop novel algorithms that could be deployed in both existing emergency networks as well as future networks.

Currently, the selection of data priorities is managed according to the nature of application protocols. Data priority allows applications to guarantee QoS and sensitive administrator data is transmitted using special channels to maintain confidentiality. However, if the network topology changes suddenly due to any of the phenomena mentioned above, a portion of the data will be lost. Moreover, even backup operations and instant storage do not facilitate all data recovery.

To allow an administrator to send urgent and confidential data (UCD) at any time, network administrators must provide preferences regarding data and constraints to routers. This allows network administrators to prohibit data transmission through a network during fixed time windows (WPs). In a particular network characterized by the confidentiality of transmitted data, it is crucial to encrypt UCD and define a date or deadline for the disclosure of UCD. The network administrator is the only person who can configure the timing of UCD divulgence, or in other words, to decide the times at which confidential data will be revealed to other authorized users. For each router in the network, the administrator will set one or more WPs. Table I illustrates all variables used in this paper.

TABLE I
VARIABLES DESCRIPTION

Variable	Description
P_j	Packet j
n_P	Total number of packets
P	Set of packets
u_j	Uptime for packet P_j
tr_j	Estimated transmission time for packet P_j
l_j	Latency for packet P_j
s_j	Start time of sending packet P_j
R_i	Router number i
WP_i^k	Window pass number k for router R_i
n_w^i	Number of WPs for router R_i
n_r	Number of routers
a	Identifier linked to the lower limit of the window pass
B	Identifier linked to the upper limit of the window pass
$WP_i^k(a)$	Lower limit of the window pass
$WP_i^k(b)$	Upper limit of the window pass

The window pass interval can be written as $[WP_i^k(a), WP_i^k(b)]$. For a fixed $WP_i^k = [x, y]$, we let $WP_i^k(a) = x$ and $WP_i^k(b) = y$.

Example 1: Assume that a network contains five routers. The network administrator has several UCD that must be transmitted through these five routers. Suppose that the network administrator chooses to fix the WPs according to the preferences presented in Fig. 1.

In Fig. 1, we can see that the WPs are dispatched by the network administrator as follows:

- For R_1, there are two WPs called WP_1^1 and WP_1^2.
- For R_2, there is only one WP called WP_2^1.
- For R_3, there are two WPs called WP_3^1 and WP_3^2.
- For R_4, there is only WP called WP_4^1.
- For R_5, there are two WPs called WP_5^1 and WP_5^2.

The WPs allows administrators to control access to all input data over 24h.

In this study, the network architecture we have adopted is illustrated in Fig. 2, which presents a diagram of a typical

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TII.2022.3149896, IEEE Transactions on Industrial Informatics

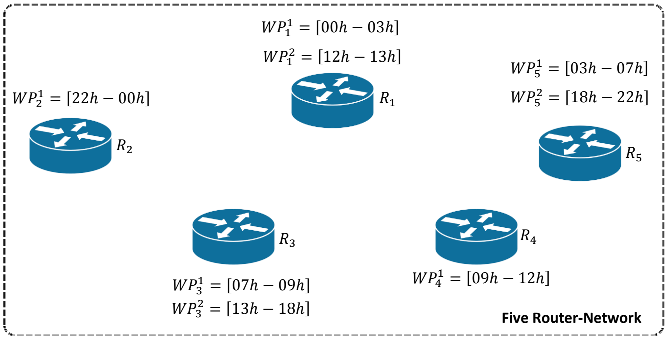

Fig. 1. Dispatching window passes (WPs) in a five router network

computer network. The different components in this figure also represent the pieces of our proposal. We called "packet uptime", the availability time of a packet P_j to be sent.

Each chunk of data is controlled by the network administrator, who is responsible for defining all preferences related to UCD. These preferences include the WPs, u_j and l_j.

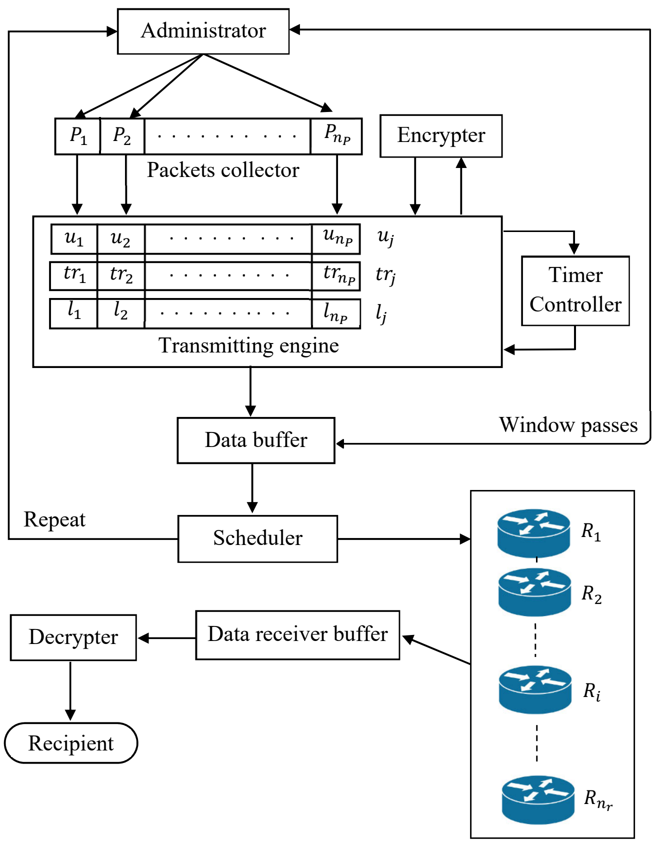

Fig. 2. Emergency network with administrator preferences

The method we propose revolves around the component labelled as the "scheduler" in Fig. 2. This component receives all packets sent as data by the network administrator and solves several scheduling problems to assign packets to the appropriate routers—an appropriate schedule yields efficient network resource management. The model described in this paper is used to show different steps followed in case of unavailability of all routers. Developed algorithms are used to dispatch data regarding the availability of routers. Data preferred by the administrator will be indexed and transmitted first within the minimum time.

Here, let f_j denote the completion time of transmission of the packet P_j. The time required for all packets to reach the component named "Data receiver buffer" in our model (see Fig. 2) and be regrouped to reconstruct the original files is calculated as: $T_{max} = max(s_j + tr_j + l_j), \forall j \in \{1, \ldots, n_P\}$. The goal here is to minimize T_{max}.

Likewise, let T_j denote the time required for packet P_j to reach the component "Data receiver buffer". This latter time is calculated as $T_j = s_j + tr_j + l_j, \forall j \in \{1, \ldots, n_P\}$. Let T_{R_i} denote the time required for router R_i to finish the transmission of all packets.

Example 2: Assume that the network administrator has one file of UCD. Suppose that the file will be divided into six packets. The "packet collector" receives the five packets using two routers. Assume that the administrator sets the WPs as follows. For R_1, $WP_1^1 = [2-4]$ and $WP_1^2 = [9-10]$. For R_2, $WP_2^1 = [3-6]$. The information regarding packets is provided in Table II.

TABLE II
INSTANCE WITH TWO ROUTERS AND SIX PACKETS

j	1	2	3	4	5	6
u_j	12	4	15	7	10	0
tr_j	4	3	5	5	6	2
l_j	14	9	8	3	1	12

Now, the scheduler must solve a scheduling problem based on the constraints listed in Table II. By applying the algorithms implemented by the scheduler, we achieved the results presented in Fig. 3.

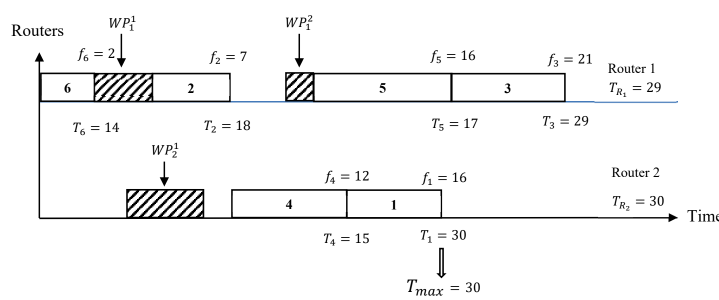

Fig. 3. Scheduling example with two routers and six packets

V. APPROXIMATE SOLUTIONS

In this section, four algorithms have been presented as means of solving the target problem. The first is based on the increasing order of the l_j algorithm (IA), while the second is based on the utilization of the exact solution for each router algorithm (ES). A critical-packet-based algorithm represents the third algorithm denoted by CP. Finally, the CE algorithm is based on the critical-packet-and-exact-solution. These proposed algorithms are based on different approaches. This difference is important to generate a new algorithm based on the best solution of all proposed algorithms. This is can be reached by running all the proposed algorithms and picking the best solution. Therefore, this new algorithm can serve to give a more efficient one.

A. Increasing order algorithm (IA)

First, a list Ls has been determined that contains all available packets at time t. Indeed, when $u_j \leq t$ the packet P_j is available, otherwise the time must be fixed with the minimum value of u_j. Then we sort Ls according to the non-decreasing order of l_j. Denote as LP_i the last packet on the router R_i. The completion time of transmission of each LP_i is denoted as C_i. The first packet in the list Ls will be sent through the router that has the minimum C_i $\forall i \in \{1, \ldots, n_r\}$ and so on until scheduling all packets in Ls. This algorithm is based on the dispatching rule. If two packets have the same l_j, the packet with the big tr_j will be assigned first. The complexity of this algorithm is $\mathcal{O}(nlogn)$ because this algorithm uses a heapsort algorithm which is a $\mathcal{O}(nlogn)$.

B. One router exact solution algorithm (ES)

After applying the IA, a schedule is obtained for each router. Let P_{R_i} denote the set of packets scheduled on router R_i. before an exact solution is calculated for each router by applying the one-router algorithm. We develop the exact solution of the one router problem. This exact solution is based on the utilization of the $1|r_j; d_j|C_{max}$ problem. The function $Onebb()$ is described in Algorithm 1. Let $Onebb()$ denote the function that yields an exact solution for a problem with one router. Algorithm 2 summarizes the process for the proposed ES. The complexity of the one-router algorithm is $\mathcal{O}(n)$. Therefore, the complexity of the ES algorithm is $\mathcal{O}(nlogn)$ because this algorithm uses a heapsort algorithm which is a $\mathcal{O}(nlogn)$.

Algorithm 1 $Onebb()$ function

```
Determine bounds
Determine fictive packets based on window-passes
Adjusting problem
Solve 1|r_j; d_j|C_max
Repeat: GOTO 2.
Choose the best solution
```

Algorithm 2 ES algorithm

```
Call IA()
for (i = 1 to i = n_r) do
    Call Onebb(P_{R_i})
    Calculate T_{R_i}
end for
T_max = max_{1≤ i≤n_r} (T_{R_i})
Return T_max
```

C. Critical packet algorithm (CP)

The first step in this algorithm is to apply one of the above algorithms denoted by A to solve the target problem. The algorithm A will return a schedule Sc. Next, we choose the packet with T_{max}. If there are multiple packets tied for T_{max}, then the packet with the greatest l_j is chosen. This packet, denoted as P_0, is called the "critical packet". We let u_0, l_0 and tr_0 denote the packet uptime, latency time and estimated transmission time of P_0, respectively. Next, the start time of sending packet P_0 (s_0) to its corresponding u_0 is set. The interval between the start time of sending packet and the completion time of transmission of P_0 will be considered as a new WP and denoted as WP_0. Therefore, $WP_0 = [u_0, u_0 + tr_0]$. Next, the algorithm A is applied to the set $P \setminus P_0$ with WPs $\cup[WP_i^k(a) - WP_i^k(b)] \cup WP_0, \forall i \in \{1, \cdots, n_r\}, \forall k \in \{1, \cdots, n_w^i\}$. This procedure is repeated until there are no additional packets that can be transformed into WPs. We let $A(P, WP)$ denote the function that applies the algorithm A to the set of packets P and intervals WP. Algorithm 3 summarizes the process for the proposed algorithm CP. The complexity of this algorithm is $\mathcal{O}(nlogn)$ because this algorithm uses a heapsort algorithm which is a $\mathcal{O}(nlogn)$.

Algorithm 3 CP algorithm

```
Initialize P_β = P, WP_β = WP
for (j = 1 to j = n_P − 1) do
    Call A(P_β, WP_β)
    Determine P_0
    Set WP_β = WP_β ∪ WP_0
    Set P_β = P_β \ P_0
end for
Assign the last packet to the most available router
Calculate T_max
Return T_max
```

In this study, A was set to be the IA.

D. Critical packet and one router exact solution algorithm (CE)

In this algorithm, the first step is to call the previous algorithm (CP). Then the router R_{max} with T_{max} is selected, and the next step is to solve the scheduling problem for R_{max} exactly. Here, Algorithm 4 summarizes the process for the proposed CE. The complexity of this algorithm is $\mathcal{O}(nlogn)$ because this algorithm uses a heapsort algorithm which is a $\mathcal{O}(nlogn)$.

VI. Experimental Results

To evaluate the performance of the algorithms we developed here, an extensive experimental study. All the proposed algorithms and instances were coded in C++ and run on a machine with Intel(R) Core(TM) i5-6200U CPU @ 2.30GHz 2.40 GHz and 8 GB RAM. The proposed procedures are tested on a set of instances that are detailed in Subsection VI-B.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TII.2022.3149896, IEEE Transactions on Industrial Informatics

Algorithm 4 CE algorithm

```
Initialize P_β = P, WP_β = WP
for (j = 1 to j = n_P − 1) do
    Call H(P_β, WP_β)
    Determine P_0
    Set WP_β = WP_β ∪ WP_0
    Set P_β = P_β \ P_0
end for
Assign the last packet to the most available router
select the router R_max with T_max
Call Onebb(P_{R_max})
Update T_max
Return T_max
```

A. Network Structure

This subsection describes the experimental results obtained by our developed algorithms. Fig. 4 shows the new network structure. It displays the component layer of the scheduler (controller layer), the preferences ordered by the administrator (Application Layer), and finally, the infrastructure layer presented with the set of routers. The algorithms developed must take into account the window-pass constraints of the routers. In this organizational model, we describe the contribution of the new component and the behaviour of the developed algorithms according to the routers constraint. The model is similar to the SDN architecture, but it takes into consideration the abnormal state of routers.

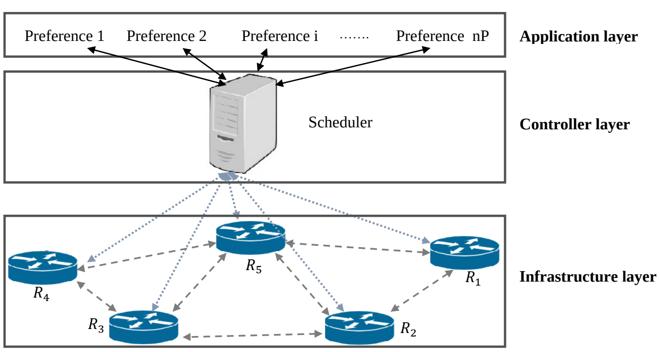

Fig. 4. Novel Network structure

B. Tested instances

The proposed algorithms were assessed using two classes of randomly generated data. The pair (n_P, n_r) has several possible combinations of values and our choice of the latter pair is demonstrated in Table III.

TABLE III
GENERATION OF (n_P, n_r)

n_P	n_r
10	2,3,5
20	2,3,5,10
50,100,300	2,3,5,10,15

For each fixed n_w^i in R_i, 10 varieties of WP were generated. For each variety of WP, 10 instances are generated. Therefore, based on Table III for the choice of pair (n_P, n_r) there are $(3+4+3\times5)\times10\times10 = 2200$ instances in total for the proposed Class. Meanwhile, the total number of instances tested in this paper was 2200.

The generation of instances is inspired by the analysis presented in [6] and [5].

tr_j, u_j and l_j are generated as follows. tr_j is generated uniformly in $[1-10]$. While, u_j and l_j are generated uniformly in $[0-30]$.

n_w^i is generated uniformly in $[1-5]$ for each router. The generation of the WP is detailed below.

For a fixed router R_i, $WP_i^1(a)$ is generated uniformly in $[1-20]$ and $WP_i^1(b) = WP_i^1(a) + U$, where U is an integer generated uniformly in $[1-10]$. Additionally, $\forall k > 1, WP_i^k(a) = WP_i^{k-1}(b) + U_k$ where U_k is generated uniformly in $[1-20]$ and $WP_i^k(b) = WP_i^k(a) + V_k$ where V_k is generated uniformly in $[1-10]$.

C. Indicators

To measure the performance of each algorithm described in Section **??**, several different indicators were evaluated. These indicators are defined as follows.

- A_b: The best (minimum) value obtained after applying all algorithms.
- A: The value returned by the presented algorithm.
- M_p: The percentage of instances for which the value returned by the presented algorithm is equal to A_b.
- $G_p = \frac{A-A_b}{A_b}$.
- A_g: The average G_p for a fixed number of instances.
- $Time$: The time required to execute an algorithm for the corresponding instances. This time is measured in seconds and we recorded it as "-" if the time is less than 0.001 seconds.

D. Discussion results

1) Overall results:

Table IV lists the overall results achieved for the developed instances, revealing that the best-performing algorithm is CP, with a M_p value of 80%, A_g value of 0.003, and an average time of 0.209 seconds.

TABLE IV
OVERALL RESULTS FOR PROPOSED ALGORITHMS

	IA	ES	CP	CE
M_p	31%	56%	80%	71%
A_g	0.017	0.009	0.003	0.003
$Time$	0.005	0.043	0.209	0.044

Table IV shows that the best algorithm is CP in 80% of cases. This means that there isn't any algorithm that is the best in 100% of cases. This proves the non-dominance of the proposed algorithms. This non-dominance can generate a new algorithm by running all the proposed algorithms and picking

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TII.2022.3149896, IEEE Transactions on Industrial Informatics

the best value. Therefore, the proposed algorithm can serve to give a more efficient one.

2) Behavior of algorithms when n_P and n_r changes: Table V details the behavior of algorithms when n_P changes. This makes clear that the best algorithm CP, as shown in Table IV, is inefficient when $n_P = \{10, 20, 50\}$. Indeed, for CP algorithm, $M_p = 93\%$, 79%, and 75% for $n_P = 10$, $n_P = 20$, and $n_P = 50$, respectively. However, the best algorithm given the maximum value of M_p when $n_P = \{10, 20, 50\}$ is CE. Indeed, the corresponding values of M_p for each value of $n_P = \{10, 20, 50\}$ are 96%, 82%, and 76%, respectively.

The maximum value of M_p for each algorithm is reached when $n_P = 10$.

The minimum average gap value of 0.001 is obtained for the algorithm CP when $n_P = 300$. In addition, for algorithm CE the minimum average gap value of 0.001 is obtained when $n_p = 10$. One can see that M_p decreases significantly as n_P increases for all algorithms excluding CP.

TABLE V
BEHAVIOR OF ALGORITHMS WHEN n_P CHANGES

n_P	IA		ES		CP		CE	
	$M_p(\%)$	A_g	$M_p(\%)$	A_g	$M_p(\%)$	A_g	$M_p(\%)$	A_g
10	65	0.015	79	0.009	93	0.002	96	0.001
20	48	0.022	71	0.011	79	0.006	82	0.005
50	39	0.016	68	0.007	75	0.004	76	0.004
100	18	0.017	52	0.008	74	0.002	68	0.003
300	3	0.013	24	0.008	84	0.001	44	0.003

Table VI presents the behavior of algorithms when n_r changes. The maximum M_p value of 95% is obtained by the algorithm CP when $n_r = 15$. However, the minimum M_p value of 21% is obtained for the algorithm IA when $n_r = 3$.

For the CP algorithm, the minimum average gap value of 0.001 is obtained when $n_r = \{10, 15\}$. For CE algorithm, the minimum average gap value of 0.002 is obtained when $n_r = \{10, 15\}$. One can see that M_p increases significantly as n_r increases for CP and CE.

TABLE VI
BEHAVIOR OF M_p AND A_g FOR ALL ALGORITHMS WHEN n_r CHANGES

n_r	IA		ES		CP		CE	
	$M_p(\%)$	A_g	$M_p(\%)$	A_g	$M_p(\%)$	A_g	$M_p(\%)$	A_g
2	22	0.017	57	0.006	66	0.005	60	0.005
3	21	0.020	58	0.009	71	0.004	63	0.004
5	33	0.017	56	0.010	83	0.002	74	0.003
10	45	0.013	56	0.009	94	0.001	81	0.002
15	44	0.013	53	0.009	95	0.001	81	0.002

3) Behavior of algorithms when Ind changes: The results show that the maximum value of $M_p = 100\%$ is obtained 3 times for the algorithm CP. The first time when $n_P = 10$ and $n_r = 5$. The second time when $n_P = 20$ and $n_r = 10$. The third time when $n_P = 50$ and $n_r = 15$. On the other hand, for algorithm CE, $M_p = 100\%$ was obtained 3 times for the same values of n_P and n_r reached for CP. For the remaining algorithm, though, $M_p = 100\%$ was never obtained. However, the minimum value of $M_p = 16\%$ for CP is obtained when $n_P = 50$ and $n_r = 3$.

Each pair (n_P,n_r) value is denoted by Ind. Fig. 5 illustrates the behavior of average gap A_g according to Ind for algorithms IA and ES. The algorithm ES has the minimum average gap compared with IA. It is important to note that four Ag relative maximums are obtained for the same values of Ind for the two algorithms IA and ES. Indeed, for $Ind = \{5, 10, 16, 22\}$ the curves IA and ES record their Ag relative maximums.

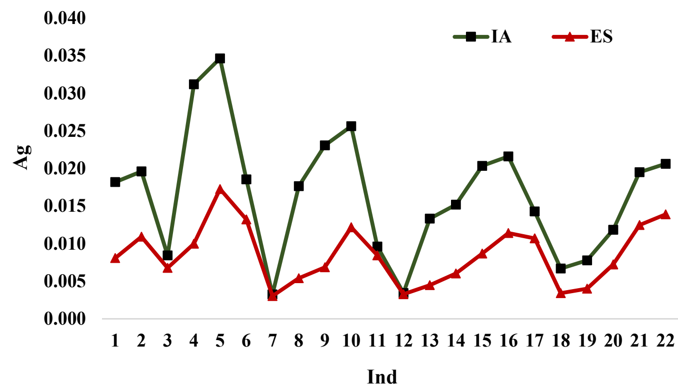

Fig. 5. Behavior of average gap A_g according to Ind for algorithms IA and ES

Fig. 6 illustrates the behavior of average gap A_g according to Ind for algorithms CP and CE. The algorithm CE has the minimum average gap compared with CP when $Ind \leq 12$.

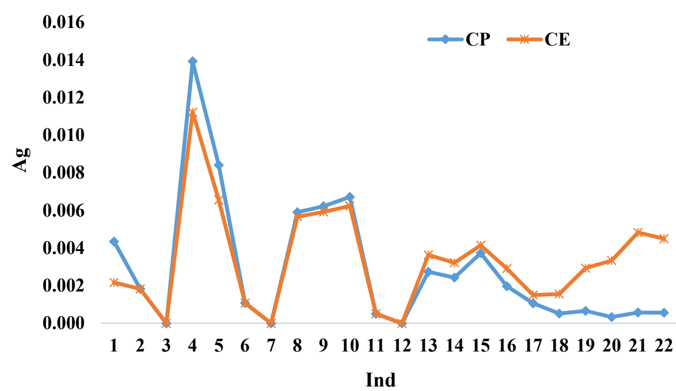

Fig. 6. Behavior of average gap A_g according to Ind for algorithms CP and CE

4) Time variation of algorithms: Table VII shows the $Time$ variation according to n_P for all algorithms. This table shows that the most time-consuming of 0.887 seconds is recorded to algorithm CP when $n_P = 300$.

Table VIII shows the $Time$ variation according to n_r for all algorithms. This table shows that the most time-consuming of 0.325 seconds is recorded for algorithm CP when $n_r = 15$. The minimum $Time$ is obtained for IA with a value of 0.004 seconds when $n_r = 2$.

The results show that the maximum $Time$ value of 0.952 seconds is obtained for CP when $n_P = 300$ and $n_r = 15$.

TABLE VII
$Time$ VARIATION ACCORDING TO n_p FOR ALL ALGORITHMS

n_P	IA	ES	CP	CE
10	-	-	0.002	0.003
20	-	0.003	0.001	0.003
50	-	0.008	0.004	0.008
100	0.002	0.014	0.026	0.016
300	0.018	0.163	0.887	0.163

TABLE VIII
$Time$ VARIATION ACCORDING TO n_r FOR ALL ALGORITHMS

n_r	IA	ES	CP	CE
2	0.004	0.095	0.168	0.055
3	0.004	0.050	0.173	0.053
5	0.005	0.026	0.189	0.036
10	0.005	0.014	0.242	0.033
15	0.007	0.010	0.325	0.037

VII. Conclusion and Prospects

A. Conclusion

In this paper, we have proposed and analyzed a potential new computer network model that could be used for specific cases such as natural disasters, which create certain resource limitations and restrictions on network communications. The model we have proposed here enables administrators to determine constraints on routers and data. This model and the algorithms developed to work with its function by finding available routers and fixing corresponding time slot intervals. A new component called the "scheduler" was added to our model to achieve this. This component runs four algorithms to solve the target scheduling problem. The experimental results we achieved by testing this model then demonstrated the excellent performance achieved by our proposed algorithms. There is no dominant relationship between the algorithms presented here, and their execution times are satisfactory. Our results demonstrate that the best algorithm is CP, which achieved 80% satisfactory results. While the algorithms analytic portion and results are essential, the "scheduler" as the main component of our proposed model must be capable of learning about previous network states to refine its results. The proposed approach needs to consider the state routers' prediction at all times. We are not sure that our algorithm's running time will be synchronized with the global network delay; a deep study should be done in this context.

B. Prospects

There are four research axes we wish to develop in the future. First, carry out more detailed discussions with different classes of instances against the main techniques discussed in the literature. This offers academics and researchers more directions on how to handle the window-passes preferences problem. Second, the proposed algorithms can be used as initial solutions for the utilization of the revolutionary algorithms as Genetic Algorithm and Particle Swarm Optimization Algorithm. Third Several lower bounds can be proposed and compared with the proposed algorithms, which constitute the upper bounds of the studied problem. This comparison can prove a near-optimal solution. Fourth, an exact solution can be developed to optimally solve the problem using the branch and bound method and the proposed algorithms as an upper bound in the branch and bound.

VIII. Acknowledgments

The authors would like to thank the Deanship of Scientific Research at Majmaah University for supporting this work under Project Number No. R-2022-42.

References

[1] L. K. B. Melhim, M. Jemmali, B. AsSadhan, and H. Alquhayz, "Network traffic reduction and representation," *International Journal of Sensor Networks*, vol. 33, no. 4, pp. 239–249, 2020.

[2] L. K. B. Melhim, M. Jemmali, and M. Alharbi, "Network monitoring enhancement based on mathematical modeling," in *2019 2nd International Conference on Computer Applications & Information Security (ICCAIS)*. IEEE, 2019, pp. 1–4.

[3] J. Wang, Y. Wu, N. Yen, S. Guo, and Z. Cheng, "Big data analytics for emergency communication networks: A survey," *IEEE Communications Surveys & Tutorials*, vol. 18, no. 3, pp. 1758–1778, 2016.

[4] A. Seba, N. Nouali-Taboudjemat, N. Badache, and H. Seba, "A review on security challenges of wireless communications in disaster emergency response and crisis management situations," *Journal of Network and Computer Applications*, vol. 126, pp. 150–161, 2019.

[5] M. Jemmali and H. Alquhayz, "Time-slots transmission data algorithms into network," in *2020 International Conference on Computing and Information Technology (ICCIT-1441)*. IEEE, 2020, pp. 1–4.

[6] H. Alquhayz and M. Jemmali, "Fixed urgent window pass for a wireless network with user preferences," *Wireless Personal Communications*, pp. 1–27, 2021.

[7] A. E. Kalør, R. Guillaume, J. J. Nielsen, A. Mueller, and P. Popovski, "Network slicing in industry 4.0 applications: Abstraction methods and end-to-end analysis," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 12, pp. 5419–5427, 2018.

[8] D. Yang, J. Ma, Y. Xu, and M. Gidlund, "Safe-wirelesshart: A novel framework enabling safety-critical applications over industrial wsns," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 8, pp. 3513–3523, 2018.

[9] G. D. Di Girolamo and A. D'Innocenzo, "Codesign of controller, routing and scheduling in wirelesshart networked control systems," *International Journal of Robust and Nonlinear Control*, vol. 29, no. 7, pp. 2171–2187, 2019.

[10] A. Manjunathan, E. K. Ruby, W. E. Santhkumar, A. Vanathi, P. Jenopaul, and S. Kannadhasan, "Wireless hart stack using multiprocessor technique with laxity algorithm," *Bulletin of Electrical Engineering and Informatics*, vol. 10, no. 6, pp. 3297–3302, 2021.

[11] M. Safaei, A. S. Ismail, H. Chizari, M. Driss, W. Boulila, S. Asadi, and M. Safaei, "Standalone noise and anomaly detection in wireless sensor networks: A novel time-series and adaptive bayesian-network-based approach," *Software: Practice and Experience*, vol. 50, no. 4, pp. 428–446, 2020.

[12] A. M. Romanov, F. Gringoli, and A. Sikora, "A precise synchronization method for future wireless tsn networks," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 5, pp. 3682–3692, 2020.

[13] S. H. H. Madni, M. S. Abd Latiff, Y. Coulibaly *et al.*, "Resource scheduling for infrastructure as a service (iaas) in cloud computing: Challenges and opportunities," *Journal of Network and Computer Applications*, vol. 68, pp. 173–200, 2016.

[14] S. H. H. Madni, M. Latiff, Y. Coulibaly, and S. M. Abdulhamid, "An appraisal of meta-heuristic resource allocation techniques for iaas cloud," *Indian Journal of Science and Technology*, vol. 9, no. 4, pp. 1–14, 2016.

[15] Y.-K. Lin and C. S. Chong, "Fast ga-based project scheduling for computing resources allocation in a cloud manufacturing system," *Journal of Intelligent Manufacturing*, vol. 28, no. 5, pp. 1189–1201, 2017.

[16] E. Schweissguth, P. Danielis, D. Timmermann, H. Parzyjegla, and G. Mühl, "Ilp-based joint routing and scheduling for time-triggered networks," in *Proceedings of the 25th International Conference on Real-Time Networks and Systems*, 2017, pp. 8–17.

[17] Z. Hanzálek, P. Burget, and P. Sucha, "Profinet io irt message scheduling with temporal constraints," *IEEE Transactions on Industrial Informatics*, vol. 6, no. 3, pp. 369–380, 2010.

[18] K. Neubauer, P. Wanko, T. Schaub, and C. Haubelt, "Enhancing symbolic system synthesis through aspmt with partial assignment evaluation," in *Design, Automation & Test in Europe Conference & Exhibition (DATE), 2017*. IEEE, 2017, pp. 306–309.

[19] G. Xu, B. Dai, B. Huang, J. Yang, and S. Wen, "Bandwidth-aware energy efficient flow scheduling with sdn in data center networks," *Future Generation computer systems*, vol. 68, pp. 163–174, 2017.

[20] H. Zhang, F. Tang, and L. Barolli, "Efficient flow detection and scheduling for sdn-based big data centers," *Journal of Ambient Intelligence and Humanized Computing*, vol. 10, no. 5, pp. 1915–1926, 2019.

[21] R. U. Mageswari and S. Baulkani, "Routing technique for data transmission under jammer in multi-hop wireless network," *Journal of Ambient Intelligence and Humanized Computing*, pp. 1–9, 2020.

[22] N. Kumar and D. Dash, "Flow based efficient data gathering in wireless sensor network using path-constrained mobile sink," *Journal of Ambient Intelligence and Humanized Computing*, vol. 11, no. 3, pp. 1163–1175, 2020.

[23] L. Xu, X. Zhou, X. Li, R. H. Jhaveri, T. R. Gadekallu, and Y. Ding, "Mobile collaborative secrecy performance prediction for artificial iot networks," *IEEE Transactions on Industrial Informatics*, 2021.

[24] R. H. Jhaveri, S. V. Ramani, G. Srivastava, T. R. Gadekallu, and V. Aggarwal, "Fault-resilience for bandwidth management in industrial software-defined networks," *IEEE Transactions on Network Science and Engineering*, vol. 8, no. 4, pp. 3129–3139, 2021.

[25] R. Shanthy and T. Padma, "A zone routing protocol incorporated with sleep scheduling for manets," *Journal of Ambient Intelligence and Humanized Computing*, pp. 1–11, 2020.

[26] G. M. Borkar and A. Mahajan, "A secure and trust based on-demand multipath routing scheme for self-organized mobile ad-hoc networks," *Wireless Networks*, vol. 23, no. 8, pp. 2455–2472, 2017.

[27] Y.-P. Hsu, E. Modiano, and L. Duan, "Scheduling algorithms for minimizing age of information in wireless broadcast networks with random arrivals," *IEEE Transactions on Mobile Computing*, 2019.

[28] M. Jemmali and H. Alquhayz, "Equity data distribution algorithms on identical routers," in *International Conference on Innovative Computing and Communications*. Springer, 2020, pp. 297–305.

Mahdi Jemmali . He received his Engineer degree in Computer science and information and M.S. degree in mathematics engineering from the Tunisia Polytechnic School, in 2005. He received a Ph.D. degree in Computer Science from the University of Tunis in 2011. He is currently an Associate professor at the Department of Computer Science and Information, Majmaah University. His interests include optimization algorithms, parallel machines, scheduling, and artificial intelligence. He is attracted by the application of several optimization techniques to solve and implement real-life hard problems from different domains.

Mohsen Denden He received the B.S. in 2000 and Ph.D in Telecommunication Engineering from National Engineer School ENIT Tunis in 2009. Currently, he is an Assistant professor at the College of Telecommunication and Information Riyadh Saudi Arabia. His current research interests include telecommunication and computer networks, information systems and cloud computing.

Wadii Boulila (SMIEEE) Received the B.Eng. degree (1st Class Honours with distinction) in computer science from the Aviation School of Borj El Amri, in 2005, the MSc. degree from the National School of Computer Science (ENSI), University of Manouba, Tunisia, in 2007, and the Ph.D. degree conjointly from the ENSI and Telecom-Bretagne, University of Rennes 1, France, in 2012. He is currently an associate professor of computer science and a senior researcher with Prince Sultan University, Saudi Arabia. He is also a senior researcher with the RIADI Laboratory, University of Manouba, and previously a Senior Research Fellow with the ITI Department, University of Rennes 1, France. He participated in many research and industrial-funded projects. His primary research interests include big data analytics, deep learning, cybersecurity, data mining, artificial intelligence, uncertainty modelling, and remote sensing images. He has served as the Chair, a Reviewer, and a TPC member for many leading international conferences and journals. He is an IEEE Senior Member and a Senior Fellow of the Higher Education Academy (SFHEA), U.K

Gautam Srivastava (SMIEEE) was awarded his B.Sc. degree from Briar Cliff University in Math and Computer Science in the U.S.A. in the year 2004, followed by his M.Sc. and Ph.D. degrees in Computer Science from the University of Victoria in Victoria, British Columbia, Canada in the years 2006 and 2012, respectively. From there in the year 2014, he joined a tenure-track position at Brandon University in Brandon, Manitoba, Canada, where he currently is active in various professional and scholarly activities. He was promoted to the rank of Associate Professor in January 2018.

Rutvij H. Jhaveri (SMIEEE) is an Assistant Professor in the Department of Computer Science & Engineering, Pandit Deendayal Energy University, Gandhinagar, India. He conducted his postdoctoral research at Nanyang Technological University, Singapore. He has 19+ years of experience in teaching and research. He obtained his Ph.D. from CHARUSAT University, India in 2016. In 2017, he was awarded Pedagogical Innovation Award by Gujarat Technological University, India. He authored 100+ publications in various prestigious journals and conferences. He is top 2% of the scientists around the world with h-index 21. He also has 3 Australian patents to his credit. He is a life member of several professional bodies such as CSI and ISTE. His research interests include Network Security, Routing Challenges, Software-Defined Networking, IoT/CPS and Machine Learning.

Thippa Reddy Gadekallu (SMIEEE) is currently working as Associate Professor in School of Information Technology and Engineering, VIT, Vellore, Tamil Nadu, India. He obtained his B.Tech. in CSE from Nagarjuna University, India, M.Tech. in CSE from Anna University, Chennai, Tamil Nadu, India and completed his Ph.D in VIT, Vellore, Tamil Nadu, India. He has more than 14 years of experience in teaching. He has published more than 100 international/national publications. Currently, his areas of research include Machine Learning, Internet of Things, Deep Neural Networks, Blockchain, Computer Vision. He is an editor in several publishers like Springer, Hindawi, Plosone. He also acted as a guest editor in several reputed publishers like IEEE, Springer, Hindawi, MDPI. He is recently recognized as one among the top 2% scientists in the world as per the survey conducted by Elsevier in the year 2021.