ABSTRACT In order to mitigate the problem of data congestion, increased latency, and high energy consumption in Wireless Sensor Networks (WSNs), Mobile Agent (MA) has been proven to be a viable alternative to the traditional client-server data gathering model. MA has the ability to migrate among network nodes based on an assigned itinerary, which can be formed via Single Itinerary Planning (SIP) or Multiple Itinerary Planning (MIP). MIP-based data gathering approach solves problems associated with SIP in terms of task duration, energy consumption, and reliability. However, the majority of existing MIP approaches focus only on reducing energy consumption and task duration, while the Event-to-sink throughput has not been considered. In this paper, a Clone Mobile-agent Itinerary Planning approach (CMIP) is proposed to reduce task duration while improving the Event-to-sink throughput in real-time applications, especially when the MA is assigned to visit a large number of source nodes. Simulation results show that the CMIP approach outperforms both Central Location-based MIP (CL-MIP) and Greatest Information in Greatest Memory-based MIP (GIGM-MIP) in terms of reducing task duration by about 56% and 16%, respectively. Furthermore, CMIP improves the Event-to-sink throughput by about 93% and 22% as compared to both CL-MIP and GIGM-MIP approaches, respectively.

INDEX TERMS Data gathering, Clone, Mobile Agent, Static Itinerary, Dynamic Itinerary, MIP

I. INTRODUCTION

The merging of low-cost wireless communication, computation, and sensing have spawned a new generation of small and cheap intelligent devices. The deployment of tens to thousands of these tiny devices in self-organizing networks has established a new network type known as Wireless Sensor Networks (WSNs). A WSN can be generally described as the deployment of a vast number of tiny sensor nodes that connect to each other via wireless communication. The advantages of these sensor nodes; such as easy installation, low cost, small size and low power consumption; make this type of networks very useful for many applications in different fields of life such as agriculture, industrial automation, transportation, health care, and military [1].

The main objective of WSN deployment is to gather environmental data and deliver it to end users [2]. Typically, in a WSN, sensor nodes are deployed in a field of interest where every sensor node is able to perform several tasks such as sensing, surveillance, environmental monitoring, processing and storing the sensed data. The sensed data at any sensor node needs to be transmitted to a collection point in the network, called a processing unit or sink, for further analysis. The transmission of data is done via single or multi-hop wireless communication. This type of transmission is based on the conventional client-server model, where every
involved sensor node transmits data packets to the sink. The WSNs have a very limited link bandwidth compared to wired networks, a set of connected wireless links need to be established from those nodes to the sink in order to forward data packets. Consequently, this could increase network data traffic and, as a result, it could increase the consumption of network resources. As a solution to this overwhelming data traffic, [9] has proposed an MA-model for data gathering in WSNs. In this model, a software component, called MA, is forwarded throughout the network and to visit source nodes one by one following an itinerary to perform data gathering process locally at each source node. Upon completion of data gathering, the MA brings back the aggregated data to the sink. Consequently, this data combination decreases the number of packets transmitted to the sink, which results in a decrease in communication costs, bandwidth utilization, network congestion, and energy consumption.

In WSNs, MA-based data gathering employs two schemes of the itinerary: Single Itinerary Planning (SIP) and Multi Itinerary Planning (MIP). SIP utilizes single MA to roam networks for data gathering [4–7], while MIP utilizes multiple MAs that work in parallel to perform data gathering [8, 9]. In large-scale networks, SIP introduces many drawbacks [10] such as long delays, increase in MA packet size, low reliability, and high probability of MA packet loss, due to the migrating to a large number of source nodes.

To overcome these drawbacks, MIP [8, 9] was proposed, in which several MAs are distributed into the network to visit groups or partitions of source nodes concurrently, where each MA is assigned to only one group of source nodes. This process reduces the MA packet size, which further leads to a lower energy consumption compared to SIP schemes. Moreover, the task duration in MIP is minimized, due to the distribution of concurrent aggregation tasks among multiple MAs.

As known, the MA itinerary is the source-visiting sequence that the MA needs to follow during its migration, where the MA migration itinerary planning is still a challenging issue in MA-based data gathering. In spite of MIP advantages, it worth noting that finding an optimal itinerary for the MA is a Non-deterministic Polynomial (NP-complete or NP-hard) problem [7, 11]. A sub-optimal MA itinerary may lead to a highly inefficient overall network performance.

In MA-based data gathering, the determination of MA’s itinerary can be classified as static, dynamic or hybrid itinerary [11, 12]. In static itinerary, the sequence of source nodes that will be visited by MA is calculated at the sink before the MA starts its migration, while in the dynamic itinerary, the sequence of source nodes that will be visited by MA is determined on the fly at each source node. In hybrid itinerary, the source nodes to be visited are selected at the sink, but the visiting sequence is computed on the fly by the MA. However, dynamic or hybrid itineraries consume valuable node energy resources and employ larger MA. This is because the MA needs to carry the next hop computation code to be executed at each node during the migration process. On the other hand, static itinerary consumes less energy compared to dynamic or hybrid itinerary since the MA carries only a pre-determined itinerary that has been calculated at the sink.

The applications of WSNs are designed to accomplish specific objectives or desired tasks. These objectives or tasks are defined and optimized according to the application user. The main goal of WSNs is to deliver the sensed data to the relevant processing unit (sink) for further analysis and decision making, where the collected data must reflect the current state of the environment. In most cases, the collected data is valid only for a limited period of time due to the constant changes in the monitored environment. Thus, the importance of these data changes from accurate to less accurate until it becomes totally inaccurate to reflect the current state of the environment as time progresses [13]. In real-time WSNs’ applications, it is very crucial to deliver the collected data to the sink without delay to make timely actions or decisions. Moreover, collecting as much data as possible is also very important to the WSN application user to allow him or her to improve the quality of evaluation and analysis to make precise decisions [14].

Indeed, MIP approaches focus only on reducing energy consumption and task duration, while neglecting the amount of delivered data to the sink within the time period. However, collecting as much data as possible requires the distributed MAs to visit a larger number of source nodes, which definitely will increase the delay. To solve this issue, a MIP approach that is able to minimize task duration and maximize collected data is highly needed to support the real-time applications. However, this solution will be at the expense of energy efficiency.

This paper proposes a novel MIP-based data gathering approach in WSNs, namely CMIP, which aims to reduce task duration of data gathering process while maximizing the volume of collected data. The remainder of the paper is organized as follows: Section II explains the related works, and Section III presents the proposed CMIP approach, while Section IV presents the simulation setup. Further, the performance evaluation and experimental results are explained in Section V. Finally, Section VI concludes the work and presents the future directions for the potential improvements.

II. RELATED WORKS
In the last few years, several MIP-based data gathering approaches have been proposed. The main objective of those approaches was to reduce the energy consumption and task duration. This section reviews several MIP-based approaches proposed for data gathering, discusses their basic concepts, and highlights their advantages and shortcomings.

In [15], a Near-Optimal Itinerary Design (NOID) algorithm was proposed to address the problem associated with calculating the number of near-optimal routes for
MAs. NOID algorithm adapts the Esau-Williams heuristic method [16] that were designed for the Constrained Minimum Spanning Tree (CMST) problem in network designing. NOID algorithm iteratively groups sensor nodes in the network to separate sub-trees that are connected progressively to the Processing Element (PE) or sink. Finally, each sub-tree is assigned to an individual MA, where the number of needed MAs is equal to the number of trees in the relevant network.

Later, the Tree-Based Itinerary Design (TBID) algorithm [17] was proposed, which outperformed NOID algorithm in terms of low-cost itineraries. TBID not only determines the optimal number of MAs but also creates low-cost itineraries for each individual MA, by partitioning the area around the PE/sink into concentric zones. The number of nodes lying within the radius of the first zone of PE/sink represents the starting points of itineraries for the needed MAs. The radius of the first zone can be calculated as $r_{max}$, where $r_{max}$ is the maximum transmission range of any sensor node and the input parameter $a \in [0,1]$. The subsequent zones width is calculated as $r_{max}/2$, such that each sensor node in any zone can only communicate with sensor nodes belonging to the previous, current, and next outer zones. In each iteration, MAs itineraries start from inner zones and proceed to outer zones connecting the pair of adjacent sensor nodes $u$ and $v$, where $u$ is linked to an itinerary and $v$ is not yet linked. This minimizes the edge Potential Cost (PC), which is equal to the cost of an itinerary derived in the case of incorporating the edge in the itinerary. This ensures low-cost itineraries to be created for each individual MA. The end output of TBID is $k$ trees, where each tree is rooted at the nodes lying within the radius of the first zone of PE/sink. The itinerary of each MA in these trees is derived by a post-order traversal.

[18] introduced a novel algorithm for the energy-efficient itinerary planning of MAs. This algorithm adopts a meta-heuristic method called Iterated Local Search (ILS) to derive the hop sequence of multiple traveling MAs over the deployed source nodes. Like other tree-based MIP algorithms, ILS determines the number of itineraries (MAs) by considering a circular zone around the sink. The nodes that are lying in the sink zone will be the starting points of each MA itinerary. However, as opposed to other tree-based MIP algorithms, ILS iteratively examines the energy cost of potential attachment of each candidate node $u$ (among those that remain unconnected) with any pair of already attached subsequent nodes. Such that the minimum itinerary cost among the examined candidate nodes will be added to the current MA’s itinerary. As a result, the ILS approach builds low energy cost of MAs’ itineraries.

Although NOID, TBID, and ILS perform better than SIP approaches, the MA in these tree-based schemes consumes twice as much energy due to the reverse routes that the MA take, especially when there are many branches. This results in an increase in the energy consumption and task duration of MA’s migration.

The central location-based MIP (CL-MIP) is another algorithm proposed by [19], where the determination of the optimal number of MAs can be divided into four parts; (1) Visiting Central Location (VCL) selection algorithm, (2) Source grouping algorithm, (3) Source-visiting order determination SIP-based algorithm, (4) Iterative algorithm; to ensure that all source nodes have been assigned to their MAs. CL-MIP groups all source nodes according to the node density (gravity algorithm). The basic idea of VCL algorithm is to distribute each source node’s impact factor to other source nodes. Let $n$ represent the source node number; then each source node will receive $(n-1)$ impact factors from other source nodes, and one from itself. At each iteration, the location of the source node with the highest accumulated impact factor will be selected as a VCL. Then, all source nodes within the radius of VCL are grouped in a cluster and assigned to an MA. The above process repeats until all remaining source nodes in the network are assigned to their MAs, then, the itinerary for each MA can be planned by a SIP algorithm. In CL-MIP, the itinerary for each distributed MA is determined by local closest first (LCF) [3]. In LCF, the MA looks for the next hop node with the shortest distance from the current location. However, CL-MIP algorithm assumes that the relevant source nodes are geographically distributed in several clusters, which limits the use of this algorithm in a broad range of applications.

In [20], the author proposed an Optimal Multi-agents Itinerary Planning (OMIP) algorithm similar to CL-MIP, where the source nodes are grouped into clusters. OMIP adopts Efficient Clustering Routing Protocol (ECRP) to partition the network into $v$ clusters, select a medoid node $MN_i$ in each cluster, and then minimizes the average distance between $MN_i$ and other nodes in the cluster. The $MN_i$ in OMIP functions as the cluster-head as well as the starting and arrival node of the MA. In OMIP, the sink dispatches an MA to every cluster in the network. The MA starts its data gathering process from $MN_i$, roams the source nodes within the cluster and returns back to the $MN_i$. After the MA completes data gathering, it travels back to the sink with the aggregated data. OMIP approach suffers from the same drawbacks of CL-MIP, where the clustering method highly depends on the distribution of source nodes. OMIP generates clusters with few source nodes to be visited by the MA. As a result, MAs’ itineraries becomes imbalanced which directly affects the network performance in terms of energy consumption and task duration.

Multi-mobile Agent itinerary planning-based Energy and Fault aware (MAEF) was proposed by [21]. Similar to (OMIP), MAF also groups the source nodes into clusters and each cluster has a Cluster Head (CH). It adopts the same idea presented in CL-MIP in which a distribution of density impact factor is used to select the CHs. Also, all source nodes, which lie within the maximum transmission range of each selected CH, are grouped together. MAF differs from CL-MIP and OMIP, as the MAs’ itineraries are planned only among the CHs and determined using the MST. Furthermore, the number of itineraries are equal to the number of nodes.
located within the sink’ transmission range. In MAEF, once the MA reaches the CH for the first time, it notifies the CH’s source nodes to send their data to the CH. When the MA reaches the last CH in its itinerary, it starts gathering data from CHs on its way back to the sink.

Although the MAEF algorithm deploys a new MIP strategy, where the distributed MAs visit only CHs, it has two main drawbacks. First, the CHs’ transmission range (which used to group the source nodes into clusters) can generate overlap between the clusters which can increase the overhead. Second, each MA needs to traverse its itinerary twice; to notify the CH’s source nodes to send data, and to gather data from the CHS. This would result in an increase of energy consumption and task duration.

In most of the aforementioned MIP approaches, the geographic information of the sensor nodes is the main parameter used to determine the optimal number of MAs and their itineraries. In [22], the Greatest Information in the Greater Memory-based MIP (GIGM-MIP) algorithm was proposed. The GIGM-MIP algorithm considers geographic information to partition the network and takes into account the data size in each partition to formulate the optimal number of MAs.

In GIGM-MIP, the data size in each partition plays an important role in determining the number of MAs. After partitioning the network, GIGM-MIP calculates the data size of the source nodes in each partition and starts with an initial number of MA. Then, GIGM-MIP updates this number by assigning source nodes to the MAs that have free payload memory size, such that each partition may have more than one MA. Using this strategy, GIGM-MIP approach balances the data load among distributed MAs, thereby reducing task duration. However, exchanging the assigned source node among the MAs results in long MA itinerary, especially when the location of the assigned source node is far from the current MA’s location. Moreover, distributing more than one MAs to one partition means that each distributed MA should carry its processing code (aggregation code). Multiple MAs carrying the same aggregation code within a single partition would result in an increase in the MA’s migration hops, which subsequently increases the energy consumption of the network.

In the above-reviewed MIP approaches, many solutions have been proposed to determine the number of MAs and their itineraries, which is the main challenging issue associated with MIP paradigm. Majority of the proposed MIP approaches are focusing on minimizing the energy consumption and task duration, while the amount of the data delivered to the sink within the time period is omitted. In some WSNs applications (e.g. real-time applications) a MIP approach that is able to minimize the task duration and maximize the data collected at the sink is really needed even if it is not optimizing the energy consumption. The proposed CMIP approach in this paper adopts the cloning concept of the MA to reduce the task duration of data gathering process while maximizing data collection.

III. CMIP APPROACH

This section presents the proposed CMIP, which includes two parts, the MA’s cloning mechanism and the mechanism of determining the itinerary used by MA(s) to visit the source nodes.

A. CLONING MECHANISM OF MA IN CMIP

As mentioned in [8], [19], the task duration in MIP is the delay experienced by the last MA, which returns to the sink node. In other words, this MA has the longest delay among the other dispatched MAs. In [5], the delay of MA’s itinerary to complete its data gathering task (T_{MA}) can be expressed as follows:

\[ T_{MA} = T_p + T_{roam} + T_{back}, \]  

(1)

where \( T_p \) is the delay of the MA migration from the sink to the first source node, \( T_{roam} \) is the MA migration delay from the first source node to the last source node, and \( T_{back} \) is the MA migration delay from the last source node to the sink. The \( T_p \) can be simplified as below:

\[ T_p = \left( \frac{pc}{D_r} + t_{ctrl} \right) \times H, \]  

(2)

where \( pc \) is the MA’s size (processing code plus MA packet header), \( D_r \) is the data rate at MAC layer, \( t_{ctrl} \) is the delay of control messages and \( H \) is the number of hops between the sink and the first source node. The \( T_{roam} \) can be expressed as follows:

\[ T_{roam} = \sum_{i=1}^{M} \left( t + S_{data} + S_{ma} + \frac{pc}{D_r} + t_{ctrl} \right) \times H, \]  

(3)

where \( M \) is the number of source nodes, \( t \) is the MA access delay (the time required to mount MA’s processing code in the target source node), \( S_{data} \) is the size of sensed data at the source node, \( D_p \) is the data processing rate, and \( S_{ma} \) is the size of the aggregated data carried by the MA at source node \( i \). The \( T_{back} \) can be decomposed into:

\[ T_{back} = \left( \frac{S_{ma} + pc}{D_r} + t_{ctrl} \right) \times H, \]  

(4)

where \( S_{ma} \) is the size of MA at the last source node.

Based on Equation 3 the delay of MA is highly associated with the number of source nodes and the number of MA’s hops. The more source nodes to be visited by MA, the larger MA’s hops which consequently implies a higher delay. Additionally, when the MA visits a large number of source nodes, its data payload size becomes large, which can further increase the MA delay.

The time required by an MA to collect data from a large number of source nodes can be minimized by dispatching two MAs since each MA will be assigned to fewer source nodes. However, this strategy would lead to increased energy consumption due to the large number of hops utilized by the two separated itineraries. Moreover, based on Equation 2 using two separated itineraries would mean each MA is required to carry its own processing code (aggregation
FIGURE 1: MA itinerary planning based on: (a) GIGM-MIP; (b) CMIP

The FSN is selected to be the cloning point, where the MMA clones itself (for example node number 4 in Figure 1(b)). At the FSN, both MMA and CMA will start data collection from the source nodes. The reason behind selecting the FSN to be the cloning point because the operation of data gathering from the farthest source node first towards the sink consumes less energy and time as compared to the one starts from the nearest source node (more details are discussed in the following section).

B. MA’S ITINERARY DETERMINATION IN CMIP

The delay and energy consumption of an MA itinerary depends on the efficiency of source nodes visiting order. In GIGM-MIP, this visiting order is determined by the LCF algorithm, which selects the nearest source node to the current MA location as the next MA’s hop. However, selecting the closest source node as the next MA’s hop does not always ensure the optimal solution. For example, in Figure 1(a), the output of the LCF algorithm shows that the MA needs to migrate 5 hops after data collection until it reaches the sink. In this context, 5 hops will be at the cost of high energy dissipation and will increase the delay since it carries a large amount of data over many hops. Indeed, it will be more efficient if the nearest source node to the sink becomes the last destination of MA (before reaching the sink). Therefore, in CMIP, the source nodes’ visiting order is first determined by the LCF algorithm, then the output itinerary is reversed, such that the last source node in the itinerary will be visited first.

The energy consumption and delay between two source nodes is proportional to the hop count between them with respect to the size of MA. Therefore, the hop count between these two source nodes is the key metric to evaluate the MA’s migration cost in terms of energy consumption and delay. In order to evaluate the MA’s migration cost in the proposed CMIP, Figure 2 illustrates the comparison of cost between LCF itinerary and reversed itinerary. In this figure, there are four source nodes to be visited by the MA. In Figure 2(a), the source nodes’ visiting order determined by LCF is 1, 2, 3 and...
where source node 1 will be visited first. Let assume that the size of the MA is 1 when dispatched from the sink and it will increase by 1 after each source node visit. By multiplying the current size of MA by the number of hops, the cost of the MA’s itinerary can easily be calculated. The results show that the total cost of the LCF itinerary is 20 when source node 1 is visited first. In contrast, from Figure 2(b), the total cost of the reversed itinerary is 16. Thus, the reversed itinerary can further decrease the energy consumption and delay.

In the proposed CMIP, if the FSN is the last source node in the itinerary, the whole itinerary is reversed and there is no cloning process. On the other hand, when the FSN is not the last source node, the reversing procedure will be different. The first part of the itinerary from the sink to the FSN will be reversed where the second part from the FSN to the sink will remain unchanged. Then, each part of the itinerary will be assigned to an individual MA, where all MAs works concurrently. Figure 1(b) above demonstrate an example of this reversed process. In this figure, the FSN is the source node 4, so the itinerary from the sink to the FSN is reversed with a visiting order 4, 3, 2, 1, till it reaches the sink. On the other hand, the second part of the itinerary, which begins from 4 to 5, 6 and 7, till it reaches the sink, will remain unchanged. This ensures that the data collection performed by the MA always starts from the farthest source node and proceeds towards the nearest nodes to the sink, which minimizes the delay and energy consumption. Algorithm 1 details the pseudo-code of the proposed CMIP approach.

**C. TASK DURATION IN CMIP**

As known, the task duration of MIP approach equals the delay of the MA from the dispatching time to it returns to the sink. However, since the CMIP consists of MMA and CMA, the task duration equals the delay from the dispatching time of the MA to the arrival of both MMA and CMA to the sink, whichever is longer. Additionally, since the itinerary of the MMA begins and ends at the sink node and it is also based on Equation (1), the total delay of its itinerary can be calculated as follows:

$$ T_{MMA} = T_{p}^{MMA} + T_{roam}^{MMA} + T_{back}^{MMA}, $$

Algorithm 1: Pseudo-code of CMIP approach

1. **Notation:**
   - $FSN$ is the farthest source node from sink
   - $SN_{S}$ denotes the set of source nodes to be visited by MA
   - $Num_{iti}$ denotes the number of MA itineraries
   - $f(SN_{S}, sink)$ is a function that returns the shortest path between the sink and FSN
   - $f(SN_{S}, FSN)$ is a function that returns the next source node determined by LCF Algorithm
   - $FSN_{SP}$ denotes the MA hop sequence determined by $f(SN_{S}, sink)$
   - $VSN_{S}$ denotes the source nodes’ visiting order determined by $f(SN_{S}, FSN)$

2. **Initialization:**
   - $N$ ← Number of sensor nodes
   - $MA_{DPT}$ ← The threshold value of the main MA data payload

3. **Partitioning the network using k-means algorithm by calculating the distance among N:**
   - $K$ ← Number of partitions (specified by user as in GIGM-MIP)

4. **for $j$ = 1 to $K$ do**
   - $S$ ← Number of source nodes in set $p$
   - $NumMAs$ ← Number of MAs in set $p$

5. **for $m$ = 1 to $NumMAs$ do**
   - Determine the MA itinerary using LCF Algorithm
   - Find $FSN$ from the sink using Equation (5)
   - if $FSN$ is not the last SN in the MA’s itinerary then
     - Split the itinerary into two itineraries from $FSN$
     - for each splited itinerary do
       - $FirSN$ ← $FSN$
       - $FSN_{SP}$ ← $f(FSN, sink)$
       - $VSN_{S}$ ← $f(SN_{S}, FSN)$
       - add $FSN_{SP}$ to $VSN_{S}$
     - Reverse the MA itinerary determined by LCF Algorithm
   - end

6. **end**

7. **Return $Num_{iti}$**

where $T_{p}^{MMA}$ is the delay of the MMA migration from the sink to the FSN, and $T_{roam}^{MMA}$ is the MMA migration delay from the FSN to the last source node, and $T_{back}^{MMA}$ is the MMA migration delay from the last source node to the sink. As for the case of the CMA, since its itinerary begins at...
the FSN and ends at the sink, its total task duration can be calculated as follows:

\[ T_{CMA} = T_{MMA}^p + T_{CMA}^{roam} + T_{CMA}^{back}, \]

where \( T_{MMA}^p \) is the delay incurred due to the MMA’s migration from the sink to the FSN plus the delay from FSN to the first source node in CMA’s itinerary. \( T_{CMA}^{roam} \) is the CMA migration delay from the first source node to the last source node, and \( T_{CMA}^{back} \) is the CMA migration delay from the last source node to the sink. Noting that the delay of \( T_{MMA}^p \) is added to the delay of CMA in Equation (7) because the delay incurred to reach the FAN is the same for both MMA and CMA.

IV. SIMULATION SETUP

The proposed CMIP approach has been implemented and tested on a simulation developed via MATLAB R2017b (student version). The same network model used in [8], [19], [22], [26] is adopted which is the most popular network model in data gathering-based MIP. We used the same energy consumption model as in [8], [27]. More specifically, a large-scale network, consisting of 800 static nodes dense and uniformly deployed, is considered in the experiments in order to validate the scaling of the proposed CMIP approach. Each node has a transmission range of 60 meters. The sink node has a continuous energy supply and it is located at the center of the network. The generated MAs’ itineraries are statically predetermined at the sink node before the MAs are dispatched to the network. In each data gathering task, a random number of source nodes, varying from 10 to 80 by the step of 5, is selected. Each compared MIP algorithm was tested with the same number of selected source nodes. The simulation parameters are listed in Table 1.

### TABLE 1: Simulation parameters

| Network Parameters | Value |
|--------------------|-------|
| Size of network    | 1000 m × 500 m |
| Number of deployed nodes | 800 |
| Number of source nodes | 10 - 80 |
| Transmission range  | 60 m |
| Raw data size      | 2048 bits |

| MA Parameters       | Value |
|--------------------|-------|
| MA processing code  | 1024 bits |
| MA accessing delay  | 10 ms |
| Raw data reduction ratio | 0.8 |
| Aggregation ratio   | 0.9 |
| Data processing rate| 50 Mbps |

V. PERFORMANCE EVALUATION

This section evaluates the performance of the proposed CMIP in terms of task duration, Event-to-sink throughput, and energy consumption, where the GIGM-MIP and CL-MIP approaches are chosen for benchmarking. The performance evaluation is done based on two scenarios: (1) a scenario with variable number of source nodes and (2) a scenario with variable aggregation ratio.

A. IMPACT OF VARYING SOURCE NODES

In this scenario, the number of source nodes is increased from 10 to 80 by a step of 5, while keeping all other parameters, as in Table 1, unchanged. In fact, the number of source nodes has a direct impact on the performance of the studied algorithms in terms of energy consumption, Event-to-sink throughput, and task duration because the more the source nodes to be visited, the larger the MA size.

Figures 3, 4, and 5 show the impact of varying the number of source nodes on the task duration, Event-to-sink throughput and energy consumption, respectively. In Figure 3 it is clear that the proposed CMIP achieves the minimum task duration by about 56% and 16% reduction compared to CL-MIP and GIGM-MIP approaches, respectively. Moreover, CMIP decreases the task duration linearly compared to GIGM-MIP when the number of source nodes increases, as a result of using the cloning mechanism, which splits the MA itineraries that have a large number of source nodes. This split operation helps each distributed MA to take less number of source nodes, which in turn helps them to complete the data gathering journey much faster.

Moreover, the reduction in task duration noticed in CMIP also improves the Event-to-sink throughput. In WSNs, the Event-to-sink throughput \( E \) is defined as the number of packets received at the sink over the time period \( T \) [28].

FIGURE 4: Event-to-Sink Throughput

FIGURE 5: Task Duration
Since the MAs in MA-based data gathering algorithms are responsible for collecting and delivering the aggregated data to the sink within the task duration, thus $E$ can be calculated as follows:

$$ E = \frac{T_{Data}}{T_{Duration}}. \quad (8) $$

where $T_{Data}$ is the total aggregated data that has been successfully delivered to the sink by all MAs and $T_{Duration}$ is the task duration from beginning till the end.

Figure 4 shows that the CMIP is able to outperform the compared schemes (GIGM-MIP and CL-MIP) by attaining higher Event-to-sink throughput due to the ability of the involved MAs in CIMP to complete their round-trip of delivering the data to the sink with lower task duration. In contrast, GIGM-MIP and CL-MIP show lower Event-to-sink throughput due to the longer time needed for the MAs to accomplish their data gathering process. In GIGM-MIP and CL-MIP, some MAs are assigned to a large number of source nodes, which in turn increases the number of hops and MA data payload size that results in task duration increase.

**FIGURE 5: Energy consumption**

Figure 5 presents the energy consumption needed to accomplish the task of CMIP compared to GIGM-MIP and CL-MIP. In MIP, the task energy consumption is the accumulated energy spent by all distributed MAs to perform the data gathering task from all source nodes [8], [18]. It includes the energy spent on transmitting, receiving, and exchanging control message. Thus, the total task energy consumption can be calculated as follows:

$$ C_{total} = \sum_{i=1}^{|I|} IC^{t_i}, \quad (9) $$

where $IC^{t_i}$ is the energy cost of itinerary $I^i$ covered by the MA, and $IC^{t_i}$ can be simplified to:

$$ IC^{t_i} = \sum_{j=1}^{|I^i|} (jdf + pc)c_{i,j}, \quad (10) $$

where $|I^i|$ represents the number of visited nodes in the itinerary $I^i$ by the relevant MA, $j$ is the visited sensor node, $jdf$ is the size of data collected by the MA at the sensor node $j$ after it is aggregated by a ratio of $f$, $pc$ is the MA’s initial size (processing code plus MA packet header), and $c_{i,j}$ is the energy consumption of the MA to migrate from sensor node $i$ to sensor node $j$. Noting that $j$ could act as a source node (has data to be collected by the MA) or as an intermediate node (forwarding node).

The proposed CMIP has a compatible performance with GIGM-MIP when there are 10-20 source nodes. However, it has lower energy consumption compared to CL-MIP scheme. As the number of source nodes increase, CMIP starts to consume higher energy compared to both GIGM-MIP and CL-MIP schemes because CMIP starts to clone its MA for the itineraries that have a larger number of source nodes which in turn results in a slightly higher number of hops. It is worth noting that the increase in hops does not affect the MA round trip since the data collected by the MMA and CMA is well distributed. Consequently, the MAs could successfully complete their tasks with less delay and high Event-to-sink throughput.

**B. IMPACT OF VARYING AGGREGATION RATIO:**

This section evaluates the impact of varying data aggregation ratio $f$ on the performance of CMIP. In this scenario, the varied $f$ ratio represents different redundancy and compression function of the collected data. Thus, different sizes of collected data can be delivered to the sink, which has a direct impact on the proposed CMIP in terms of task duration, Event-to-sink throughput, and energy consumption. In this experiment, $f$ ratio is varied from 0.1 to 0.9 for 80 source nodes.

**FIGURE 6: Task Duration**

Figures 6, 7 and 8 show the impact of varying the ratio of $f$ on task duration, Event-to-sink throughput and energy consumption, respectively. As shown in Figure 6, the proposed CMIP outperforms GIGM-MIP and CL-MIP in terms of task duration, although the size of MA becomes larger whenever the value of $f$ becomes smaller. This is due to the cloning strategy deployed by the CMIP, which facilitates the collaboration between the involved MAs (MMA and CMA) allowing them to collect larger data collaboratively in a shorter duration of time.
On the other hand, Figure 7 shows that the Event-to-sink throughput of CMIP is higher compared to GIGM-MIP and CL-MIP schemes. Although the size of the collected data changes on varying \( f \), CMIP still performs better than the other schemes due to the large amount of data collected by the MMA and CMA within a shorter time, which improves the Event-to-sink throughput. As the size of the collected data in GIGM-MIP and CL-MIP increases, the MA size increases. Thus, receiving and transmitting an MA with large data at any hop consumes a significant amount of time, which increases the total task duration and directly decreases the Event-to-sink throughput.

Additionally, receiving and transmitting an MA that carries a large amount of data at any hop consumes more energy, which may shut down that hope due to having a low level of energy. As a result, the MA with its carried data will be lost, which negatively affects the Event-to-sink throughput. However, the increase in energy consumption caused by the large size of MA has a negligible impact on the Event-to-sink throughput in CMIP, as shown in Figure 8 due to the use of the cloning strategy that deploys two MAs to gather throughput in CMIP, as shown in Figure 8, due to the use of the cloning strategy that deploys two MAs to gather

![Figure 7: Event-to-Sink Throughput](image1)

![Figure 8: Energy consumption](image2)

VI. CONCLUSION

In this paper, a Clone Mobile-agent Itinerary Planning (CMIP) approach has been proposed. The proposed CMIP adopt the cloning concept of MA to reduce the task duration when the MA’s itinerary has a large number of source nodes to be visited. The CMIP mitigates this issue by splitting the itinerary into sub-itineraries, each of which is assigned to an individual MA. Moreover, a reversed MA’s itinerary is proposed to further decrease the energy consumption and task duration. Further, simulation experiments have been conducted to evaluate the performance of CMIP. The results show that CMIP significantly outperforms the compared schemes in terms of task duration and Event-to-sink throughput.

REFERENCES

[1] Daniele Puccinelli and Martin Haenggi. Wireless sensor networks: applications and challenges of ubiquitous sensing. IEEE Circuits and systems magazine, 5(3):19–31, 2005.
[2] Jamal N Al-Karaki and Ahmed E Kamal. Routing techniques in wireless sensor networks: a survey. IEEE wireless communications, 11(6):6–28, 2004.
[3] Hairong Qi, Yingyue Xu, and Xiaoling Wang. Mobile-agent-based collaborative signal and information processing in sensor networks. Proceedings of the IEEE, 91(8):1172–1183, 2003.
[4] Hairong Qi and Feiyi Wang. Optimal itinerary analysis for mobile agents in ad hoc wireless sensor networks. Proceedings of the IEEE, pages 147–153, 2001.
[5] Min Chen, Taekyoung Kwon, Yong Yuan, Yanhee Choi, and Victor Leung. Mobile agent-based directed diffusion in wireless sensor networks. EURASIP Journal on Applied Signal Processing, 2007(1):219–219, 2007.
[6] Qishu Wu, Nageswara SV Rao, Jacob Barhen, SS Iyengar, Vijay K Vaishnavi, Hairong Qi, and Krishnendu Chakrabarty. On computing mobile agent routes for data fusion in distributed sensor networks. Knowledge and Data Engineering, IEEE Transactions on, 16(6):740–753, 2004.
[7] Min Chen, Victor Leung, Shiwen Mao, Taekyoung Kwon, and Ming Li. Energy-efficient itinerary planning for mobile agents in wireless sensor networks. In Communications, 2009. ICC’09. IEEE International Conference on, pages 1–5. IEEE, 2009.
[8] Wei Cai, Min Chen, Takahiro Hara, Lei Shu, and Taekyoung Kwon. A genetic algorithm approach to multi-agent itinerary planning in wireless sensor networks. Mobile Networks and Applications, 16(6):782–793, 2011.
[9] Mostefa Bendjima and Mohamed Feham. Multi mobile agent itinerary for wireless sensor networks. International Journal of Emerging Trends & Technology in Computer Science, 1(1):6–11, 2012.
[10] Huthiafa Q Qadori, Zuriati A Zulkarnain, Zurina Mohd Hanapi, and Shamala Subramaniam. Multi-mobile agent itinerary planning algorithms for data gathering in wireless sensor networks: A review paper. International Journal of Distributed Sensor Networks, 13(1):1550147716684841, 2017.
[11] Min Chen, Sergio Gonzalez, and Victor Leung. Applications and design issues for mobile agents in wireless sensor networks. Wireless Communications, IEEE, 14(6):20–26, 2007.
[12] Aristides Mpiotiopoulos, Damianos Gavalas, Charalampos Konstantopoulos, and Grammati Pantziou. Mobile agent middleware for autonomic data fusion in wireless sensor networks. Autonomic computing and networking, pages 57–81, 2009.
[13] Ousmane Diallo, Joel JPC Rodrigues, and Mbaye Sene. Real-time data management on wireless sensor networks: A survey. Journal of Network and Computer Applications, 35(3):1013–1021, 2012.
[14] Saied Pourroostaei Ardakani. Wireless Sensor Network Routing Protocols for Data Aggregation. PhD thesis, University of Bath, 2014.
[15] Aristides Mpiotiopoulos, Damianos Gavalas, Charalampos Konstantopoulos, and Grimirati Pantziou. Deriving efficient mobile agent routes in wireless sensor networks with noid algorithm. In Personal, Indoor and Mobile Radio Communications, 2007. PIMRC 2007. IEEE 18th International Symposium on, pages 1–5. IEEE, 2007.
16] Larry R. Esau and Kenneth C. Williams. On teleprocessing system design: part ii a method for approximating the optimal network. IBM Systems Journal, 5(3):142–147, 1966.

17] Charalampos Konstantopoulos, Aristides Mpitziopoulos, Damianos Gavalas, and Grammati Pantziou. Effective determination of mobile agent itineraries for data aggregation on sensor networks. Knowledge and Data Engineering, IEEE Transactions on, 22(12):1679–1693, 2010.

18] Damianos Gavalas, Ioannis E Venetis, Charalampos Konstantopoulos, and Grammati Pantziou. Energy-efficient multiple itinerary planning for mobile agents-based data aggregation in wsns. Telecommunication Systems, pages 1–5, 2016.

19] Min Chen, Sergio Gonzalez, Yan Zhang, and Victor CM Leung. Multi-agent itinerary planning for wireless sensor networks. In International Conference on Heterogeneous Networking for Quality, Reliability, Security and Robustness, pages 584–597. Springer, 2009.

20] Amine Rais, KHALID BOURAGBA, and Mohamed Ouzzif. Determination of itinerary planning for multiple agents in wireless sensor networks. International Journal of Communication Networks and Information Security (IJCINS), 10(1), 2018.

21] Mohamed El Fissaoui, Abderrahim Beni-Hssane, and Mostafa Saadi. Multi-mobile agent itinerary planning-based energy and fault aware data aggregation in wireless sensor networks. EURASIP Journal on Wireless and Mobile Communications and Networking, 2018(1):92, 2018.

22] Imene Aloui, Okba Kazar, Laid Kahloul, and Sylvie Servigne. A new itinerary planning approach among multiple mobile agents in wireless sensor networks (wsn) to reduce energy consumption. International Journal of Communication Networks and Information Security, 7(2):116, 2015.

23] Aristides Mpitziopoulos, Damianos Gavalas, Charalampos Konstantopoulos, and Grammati Pantziou. Obid: a scalable method for distributed data aggregation in wsns. International Journal of Distributed Sensor Networks, 6(1):206517, 2010.

24] Soheil Javadi, Mohammad H Hajiesmaili, Behzad Moshiri, and Ahmad Khonsari. Clone-based mobile agent itinerary planning using separate trees for data fusion in wsns. International Journal of Wireless & Mobile Networks, 4(4):227, 2012.

25] Govind P Gupta, Manoj Misra, and Kumkum Garg. Energy and trust aware mobile agent migration protocol for data aggregation in wireless sensor networks. Journal of Network and Computer Applications, 41:300–311, 2014.

26] Junfeng Wang, Yin Zhang, Zhuanti Cheng, and Xuan Zhu. Emip: energy-efficient itinerary planning for multiple mobile agents in wireless sensor network. Telecommunication Systems, pages 1–8, 2015.

27] Min Chen, Laurence T Yang, Taekyoung Kwon, Liang Zhou, and Minh Jo. Itinerary planning for energy-efficient agent communications in wireless sensor networks. IEEE Transactions on Vehicular Technology, 60(7):3290–3299, 2011.

28] Chen Wang and Li Xiao. Improving event-to-sink throughput in wireless sensor networks. In International Conference on Distributed Computing in Sensor Systems, pages 50–63. Springer, 2007.

Huthiafa Q Qadori received his bachelor degree in computer science from Faculty of Computer Science and Information Technology, Universiti of Al-Anbar, Iraq, in 2006, and his master of Information Technology from University Tenaga National (Uniten), Malaysia, in 2012. He is currently pursuing his Ph.D degree at the Department of Communication Technology and Networks, Universiti Putra Malaysia. His research interests are in computer networks, wireless sensor networks, routing algorithms and the Internet of things.

Zuriati Ahmad Zukarnain (M’10) received her bachelor and master degrees in physics and education from Universiti Putra Malaysia (UPM) in 1997 and 2000, respectively, and her Ph.D. degree in quantum computing and communication from the University of Bradford, U.K., in 2005. Now, she is a professor in the Faculty of Computer Science and Information Technology, UPM. She was appointed as Head of Department for Communication Technology and Networks from 2006 to 2011. She also being appointed as the Head of the Section of High-Performance Computing, Institute of Mathematical Research, UPM, from 2012 to 2015. She taught several courses for the undergraduate students such as data communication and networks, distributed system, mobile and wireless networks, network security, computer architecture, and assembly language. For postgraduate students, she taught few courses such as advanced distributed computing and research method. Also, she is a member of the IEEE. Her areas of interest are computer networks, distributed system, mobile and wireless networks, network security, quantum computing, and quantum cryptography.

Mohamed A. Alrshah (M’13–SM’17) received his BSc degree in Computer Science from Naser University - Libya, in 2000, and his MSc and Ph.D. degrees in communication technology and networks from Universiti Putra Malaysia in May 2009 and Feb 2017, respectively. Now, he is a Senior Lecturer in the Department of Communication Technology and Networks, Faculty of Computer Science and Information Technology, Universiti Putra Malaysia (UPM). Also, he is a senior member of the IEEE. He has published a number of articles in high impact factor scientific journals. His research interests are in the field of high-speed TCP protocols, high-speed wired and wireless network, parallel and distributed algorithms, WSN, IoT, and cloud computing.

Zurina Mohd Hanapi (M’10) received her bachelor in Computer and Electronic System Engineering from Strathelyde University in 1999, and her master in Computer and Communication Systems Engineering from UPM in 2004, and her Ph.D. in Electrical, Electronic, and System Engineering from Universiti Kebangsaan Malaysia in 2011. Now, she is an Associate Professor in the Faculty of Computer Science and Information Technology, University Putra Malaysia (UPM). She has received an Excellence Teaching Awards in 2005, 2006 and 2012 and she has received a silver medal in 2004 and bronze medal in 2012. She is a leader of some research projects and she has published many conference and journal papers and she is a member of the IEEE. Her research interests in Routing, Wireless Sensor Network, Wireless Communication, Distributed Computing, Network Security, Cryptography, and Intelligent Systems.
Shamala Subramaniam (M’10) received her Bachelor, Master, and Ph.D in Computer Science from University Putra Malaysia (UPM) in 1996, 1999, and 2002, respectively. Now, she is a professor in the Department of Communication Technology and Network, Faculty of Computer Science and Information Technology, Universiti Putra Malaysia. Also, she is a member of the IEEE. Her research interests are Computer Networks, Simulation and Modeling, Scheduling and Real-Time System.