JOURNAL OF INFORMATION SCIENCE AND ENGINEERING 36, 795-819 (2020)
DOI: 10.6688/JISE.202007_36(4).0007

A Scalable Two-Hop Multi-Sink Wireless Sensor Network for Data Collection in Large-Scale Smart Manufacturing Facilities*

CONG GAO[1,2,+], ZHONGMIN WANG[1,2], YANPING CHEN[1,2]
AND ZHENZHOU TIAN[1,2]
[1]*School of Computer Science and Technology*
[2]*Shaanxi Key Laboratory of Network Data Analysis and Intelligent Processing*
Xi'an University of Posts and Telecommunications
Xi'an, Shaanxi, 710121 P.R. China

In industrial fields, wireless sensor networks have been massively deployed for the purpose of data collection. For the various application scenarios of smart manufacturing in Industry 4.0, versatile production tasks demand dynamic features both in production lines and manufacturing processes. Therefore, the design and performance of the corresponding data collection mechanisms are facing unprecedented challenges. In this work, we propose a unified data description and management framework. This framework possesses high flexibility that it is able to identify an unknown data type and accord an adequate description. Besides, the scalability of this framework enables the provision of handy interfaces for the exploitation of stored data. Then, we develop two network connectivity models in one dimension and two dimensions. These two models greatly facilitate the measurement of the level of connectivity for a wireless sensor network. At last, we elaborate a two-hop multi-sink routing scheme to alleviate the flooding problem. This scheme contains a novel r-Kruskal algorithm for the sink nodes and an efficient two-hop routing method for the whole network. The flooding effect can be neatly controlled with the two-hop scheme. Extensive experiments are conducted to evaluate our proposal. Simulation results show that our model has excellent adaptability to the scale of the network and possesses satisfactory performance in terms of both message overhead and data availability.

Keywords: data collection, wireless sensor networks, network connectivity, routing, data availability, message overhead

1. INTRODUCTION

Recent years, the Cyber-Physical Systems (CPS) have possessed large scale deployment in industrial manufacturing [1]. The CPS is defined as a collection of innovative technologies for regulating interconnected systems between their physical assets and computing power [2]. Generally speaking, the information comes from physical workshops and cyberspace is closely monitored and coordinated. By employing cutting-edge information analysis methods, the interconnected machines are able to operate collaboratively and efficiently. The above pattern has greatly facilitated the fourth revolution of industrial manufacturing (aka. Industry 4.0) [3]. With the rapid development of sensors, data

Received August 28, 2019; revised September 25, 2019; accepted November 7, 2019.
Communicated by Xiaohong Jiang.
+ The corresponding author.
* This work is supported in part by the Science and Technology Project of Shaanxi Province, China (Grant No. 2019ZDLGY07-08), the International Science and Technology Cooperation Program of the Science and Technology Department of Shaanxi Province, China (Grant No. 2018KW-049), and the Special Scientific Research Program of Education Department of Shaanxi Province, China (Grant No. 17JK0711).

acquisition devices, and computer networks, more and more enterprises rush to deploy sensors and interconnected machines for the purpose of improving competitiveness [4]. As a result, the factories armed to the teeth persistently produce massive data, namely big data [5]. Under the above circumstances, a further development of the CPS enables an efficient management of big data [6]. By taking advantage of the interconnectivity of machines, an intelligent, reconfigurable, and self-adaptive manufacturing system is feasible [7]. With the integration of the CPS and the production, logistics, and services within current industrial processes, the present traditional factories can be transformed to a new type of factory, which is known as smart factory [8, 9]. In 2014, a cooperative research by Fraunhofer [10] and Bitkom [11] claimed that German gross value can be boosted by a cumulative 267 billion euros by 2025 after introducing Industry 4.0 [12].

1.1 Four Design Principles

To bring about Industry 4.0, four principles are identified as guidelines [13].

1) Interconnection. As machines, devices, sensors, and people are closely interacted with each other over the Internet of Things (IoT) [14], a ubiquitous access demand is mandatory. Since traditional links based on a wired communication are unable to accord an omnipresent access support, wireless communication technologies are preferred. Thus, a flexible and efficient wireless communication mechanism which facilitate the connections in the Internet of Things is of great importance and acts as the basis of Industry 4.0.

2) Information Transparency. The fusion of physical world and virtual world enables a new form of information transparency [15]. Based on the networked machines and various sensor data, a virtual copy of the physical factory is established. This virtual copy consists of cyber components which correspond to the real world. An excellent implementation of the above information transparency enables a comprehensive view of the operation of the real world. The information transparency hides the complex composition of the real world and the corresponding multi-source heterogeneous data generated in the real world.

3) Decentralized Decisions. Equipments of the CPS share data with a digitalized network, and they are monitored and controlled autonomously and simultaneously by a certain management scheme. The global coordination conducted by the management scheme is crucial to the operation of the whole manufacturing environment. In most instances, decisions are made locally. This local processing manner reduces the burden of transmission and takes full advantage of the edge computing resources. While in case of exceptions, interferences, and conflicts, the issue of decision-making is committed to a higher level [16].

4) Technical Assistance. In a traditional factory, humans mainly focus on the operation of machines. For most tasks, the operation of machines are just routines. The involvement of humans are simple and straightforward. However, large amount of repetitive work requires a lot of workers. While in smart factories, humans are shifted to macroscopic planning and problem solving. As the role of humans is centered on technical and strategic issues, the number of works could be significantly reduced. This role transition also demands higher specialty literacy for a worker in a smart factory than that of a traditional factory. For instance, with the development of robotics, humans should be trained adequately for the human-machine collaboration [17].

1.2 Three Kinds of Integration

Besides the above four design principles, there are three critical aspects should be considered for realizing Industry 4.0 [18]: 1) horizontal integration through value networks, 2) vertical integration and interconnected manufacturing systems, and 3) end-to-end digital integration of engineering across the entire value chain. The above three integrations in Industry 4.0 are depicted in Fig. 1.

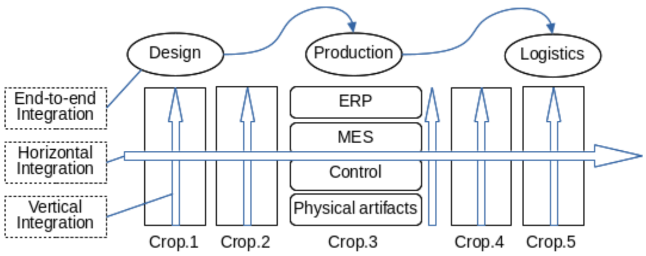

Fig. 1. Three kinds of integration.

The horizontal integration through value networks promotes the collaboration among different corporations. In practice, an individual corporation competes and cooperates with many other corporations. The horizontal integration enables the formation of an efficient ecosystem among related corporations. With the fluent exchange of information, capital and material among these corporations, fresh value networks, and new business models will spring up.

The vertical integration is conducted within an individual factory. It aims to construct a flexible and reconfigurable manufacturing system which consists of hierarchical subsystems. A conventional factory possesses several cyber-physical subsystems, such as cooperative planning system, driving system, signal sensing system, control system, production management system, and mechanic manufacturing system. To achieve high flexibility and reconfigurability, a vertical integration of hierarchical subsystems is needed. The top of the integration lies the enterprise resource planning (ERP) system. By the above integration, the machines within a factory form a self-organized system. The system is able to reconfigure itself for the purpose of adapting various production tasks. By collecting and analyzing the massive data contained in the production lines and manufacturing processes, the production flow of a product is clearly presented.

End-to-end digital integration of engineering across the entire value chain affords support for product customization. The product-centric value creation process involves a series of activities, such as customer requirement profiling, product design and development, services, maintenance, recycling and reusing.

As discussed above, the vertical integration of a manufacturing system networks is the context of a smart factory. It concentrates on the construction of a flexible and reconfigurable manufacturing system which consists of hierarchical subsystems. Smart factory is the primary application entity which holds up Industry 4.0 [19]. Its modern production lines and manufacturing processes contained various data, such as temperature, pressure, displacement, thermal energy, vibration and noise. Multiple forms of analysis can be conducted based on these data, such as equipment diagnostics, power consumption, quality assurance and automated logistics. A brief comparison between today's factory and an Industry 4.0 factory is presented in [20]: both factories are introduced/described/analyzed in three aspects: component, machine, and production system. The data source is listed as sensor, controller, and networked system. The attributes and technologies of the above three data sources are refinedly summarized. In short, the notable features of an Industry 4.0 factory are identified as: self-aware, self-predict, self-aware, self-predict, self-compare, self-configure, self-maintain, and self-organize.

During the manufacturing process of a final product, a series of complex operations are imposed upon raw material and semi-finished products. This process involves a variety of cyber-physical subsystems which are located on different layers, such as driving and sensing layer, control layer, manufacturing and executing layer, coordinative planning layer and production management layer. At present, information cannot circulate smoothly among the above subsystems, which compromises the continuity and consistency of a manufacturing process. Thus, a key issue in implementing industry 4.0 is the vertical integration of hierarchical subsystems. This integration transforms traditional factories to a smart factory which is highly flexible and reconfigurable.

The key tasks of the vertical integration of hierarchical subsystems are data collection and transmission. In traditional methods, researchers employed the cloud computing technologies. Cloud computing provides shared computing resources in the light of user demands. Data are aggregated and provided based on demands. For the wireless sensor networks, the concept *sensor cloud* [21] sprang up with the combination of wireless sensor network and cloud computing. As the name suggests, sensor cloud refers to an infrastructure within which physical sensors are connected to cloud for management. It provides users with cloud service instances in an automated way. The cloud service instance is called *virtual sensor*. A virtual sensor is an emulation of a physical sensor and its data are obtained from underlying physical sensors [22]. The term *virtual* means transparency to users. Namely, there is no difference between a cloud service instance and other physical resources in the system in terms of user experience. Before the concept of sensor cloud appears, the real-time communication of cloud computing has been discussed [23,24]. And extensive studies on the integration of sensors with a cloud framework. In [25], a detailed review of sensor cloud was given, including concepts, inherent natures, and application advantages. In addition, a comparison among the message types involved in different models was also conducted. An optimal gateway selection model is proposed for the purpose of maximizing bandwidth for data transmission. In [26], the challenges in front of the integration of wireless sensor network and cloud were highlighted, then a dedicated sensor cloud framework for Software-as-a-Service applications was proposed. A similar work was provided in [27], the challenges for comprehension of sensor cloud diversification, implementation of scalable functions, privacy protection were discussed. In [28], a simple virtual wireless sensor network infrastructure was proposed. The scheme is independent of the underlying protocols, and is able to combine with popular routing protocols and data aggregation protocols. In [29], a topology virtualization model was implemented by node self-organization for underwater sensor network. In [30], the authors proposed a cost-efficient virtual sensor management scheme for a large scale deployment of wireless sensor nodes. This scheme is aimed to efficiently map concurrent data sensing requests to the sensor cloud.

Most existing works have the following drawbacks: there is a lack of a unified data management framework for the multi-source heterogeneous data. A unified data management framework is crucial to the collection and storage of data. Besides, the connectivity of a wireless sensor network is not taken into account. The mobility and limited radio range of a wireless sensor node could significantly affect the usability of a a wireless sensor network. In addition, detailed routing protocols for the wireless sensor networks are not mentioned. Thus, the implementation of an actual system is infeasible.

Our contributions are summarized below.

- A unified data description and management framework. This framework is able to deal with existing data types and unknown data types. The flexible data description method of the framework accords high scalability and facilities the subsequent exploitation of stored data.
- Connectivity model of a wireless sensor network. We discussed the application scenario of a wireless sensor network which consists of mobile sensor nodes and sink nodes. Then, we developed the models of one-dimensional network connectivity and a two-dimensional network connectivity.
- A two-hop multi-sink routing scheme. In this scheme, we proposed an *r-Kruskal* algorithm to facilitate the communication activities in the network of sink nodes. Then, we designed a two-hop routing scheme for the whole network which includes both the mobile sensor nodes and the sink nodes.

The rest of this paper is organized as follows. Section 2 reviews the level of measurement and develops a unified data description and management framework. Section 3 describes the application scenario and proposes the one-dimensional network connectivity model. Section 4 introduces the two-hop model and formulates the level of connectivity. In addition, we develops the *r-Kruskal* algorithm and elaborates the two-hop multi-sink routing scheme for the whole network. In Section 5, we present well-designed experiments and make a detailed analysis of the experimental results. Section 6 contains the conclusions and directions for future work.

2. A UNIFIED DATA DESCRIPTION AND MANAGEMENT FRAMEWORK

For the diversity and complexity of data from different sources and different phases of the manufacturing processes, we propose a unified data description and management framework to facilitate the intelligent fusion for multi-source heterogeneous data from various devices of different domains.

2.1 Level of Measurement

During the cognition of information, plenty of work has been done to bring out a taxonomy of levels of measurement. The most influential definition is given by Stevens, which classified the level of measurement as: nominal, ordinal, interval and ratio [31-33]. The nominal type is able to handle a qualitative measurement, while the rest three are able to deal with a quantitative measurement.

- Nominal measurement focuses on classification and membership. It does not reflect an order of values.
- Ordinal measurement centers around comparison and level. It is able to accord an order of values. However, the differences between consecutive values are imprecise.
- Interval measurement concentrates on difference and affinity. It provides precise differences between the measured objects, which is conventionally described as distance. However, the meaning of zero is arbitrary (*e.g*. GPS coordinates).
- Ratio measurement is targeted at magnitude and amount, a ratio scale possesses a meaningful zero value (*e.g*. the absolute zero of the Kelvin temperature scale).

2.2 Data Description and Management

The unified data description and management contains a scalable data description model. The model is able to accommodate the data in existing data types. Moreover, by analyzing an unknown data type, new data formats could be readily created by our unified data description model. Thus, it is capable to deal with the data in unknown data types. Besides, this model is flexible to afford efficient interfaces for data query, data analysis, and data mining.

Definition 1: The measurement is denoted by m_i, such that $m_i \in M = \{m_1, m_2, m_3, m_4\}$. Specifically, the measurements nominal, ordinal, interval, and ratio are denoted by m_1, m_2, m_3, and m_4, respectively.

In a smart factory, considerable underlying information is contained during a manufacturing process. We decompose the physical world into a number of entities which act as data sources. These data sources are classified based on their own physical phenomena, such as temperature, vibration, sound, pressure, motion, light, humidity, gravity, magnetic fields, and electrical fields. Each data source is equipped with a corresponding type of industrial sensor. Then, large amount of diversified time series data are generated by these industrial sensors during the operation of the equipment.

Definition 2: The type of physical sensor is denoted by t_i, such that all possible physical sensor types in a smart factory are denoted by set $T = \{t_1, t_2, \cdots, t_\alpha\}$.

Definition 3: The physical phenomenon is denoted by p_i, such that all interested physical phenomena in a smart factory are denoted by set $P = \left\{p_1, p_2, \cdots, p_\beta\right\}$.

There are two mapping among the above sets M, T, and P. For nonempty sets P and M, we denote the mapping from P to M by $f_{pm} : P \to M$. Likewise, for nonempty sets M and T, the mapping from M to T is denoted by $f_{pm} : M \to T$. Specifically, for each element in set P, there always exists a unique corresponding element in set M. The same applies to set M and set T.

So far we have involved nothing about the type of data, which is a traditional attribute. In practice, data are often referred to as text, photo, audio, video, and so on. Furthermore, for different application-specific areas, the types of data could be multitudinous. To support higher level components of an application system, it is essential for a data description and management framework to enable the indication of the above information. Thus, we propose a human-oriented profiling mechanism to describe the type of data. In particular, there is a list of properties which is used for profiling a type. Each property has a number of values. We denote the list of properties and values by and all components of a smart factory are aware of this list. For instance, a property called device indicates the name of the device, and a property called temperature may have the values cold start temperature, operating temperature, alarm temperature, and so on. Based on investigations of the application area, a list of properties and the corresponding values could be obtained readily. The properties are denoted by set $property = \left\{pr_1, pr_2, \cdots, pr_{np}\right\}$. The corresponding values of pr_i are denoted by set $p_i = \{pv_{i1}, pv_{i2}, \cdots, pv_{im}\}$. Now, we introduce the definition of data type profile.

Definition 4: By data type profile we mean the character string $tp = \left\{pv_{1s}, pv_{2j}, \cdots, pv_{(np-i)k}\right\}$ which is formed by concatenating the values of $np - i$ properties sequentially, where $0 \leqslant i < np$ and $1 \leqslant s, j, k \leqslant m$. When $i = 0$, we call tp a *complete profile*. Otherwise, we call tp an *incomplete profile*. Thus, there are totally $\prod_{i=1}^{np} |p_i|$ different complete profiles.

The unified data description and management framework stipulates that the data obtained by one sensing operation should conform a format which contains six parts. As depicted in Fig. 2, the the first three parts denote phenomenon, measurement, and sensor type, respectively. The fourth part is the unique identification of a physical sensor, since several physical sensors might be deployed for one data source. The fifth part is the type profile. The last part indicates the binary form of the sensor data.

p_i	m_i	t_i	ID	TP	Content

Fig. 2. Data format.

For an unknown data type, the corresponding measurement and physical phenomenon are identified firstly. Then, the properties in set *property* is examined to determine a possible value. However, there are chances that some properties are not suitable or even have nothing to do with the current unknown data type. In this case, the particular properties are kept empty. Besides, an unknown data type may possesses some characteristics which cannot be represented by the existing properties in set *property*. Then, new properties are created in response to the new characteristics.

3. A PRELIMINARY MODEL OF NETWORK CONNECTIVITY

For simplicity, we divide the whole factory floor into $m \times n$ grids in a two-dimensional surface. The set of grids are denoted by set $gr = \{gr_{11}, gr_{12}, \cdots, gr_{ij}\}$, where $1 \leqslant i \leqslant m$ and $1 \leqslant j \leqslant n$.

The whole sensor network consists of two kinds of nodes: sink node and mobile sensor node.

A sink node is located at a fixed position, and it has a cabled power supply. Thus, there is no energy constraint to a sink node. All sink nodes follow the same networking protocol and constitute the backbone infrastructure of the sensor network. We denote the sink nodes by set $sn = \{s_1, s_2, \cdots, s_{N_s}\}$. The backbone infrastructure is denoted by SNN, which is short for sink node network.

A mobile sensor node relies on limited battery energy, which is a common issue of wireless network. Thus, its functionalities should be meticulously designed. All mobile sensor nodes form a mobile ad hoc network and act as the local structure of the sensor network. We denote the mobile sensor nodes by set $ms = \{ms_1, ms_2, \cdots, ms_{N_m}\}$. The local structure is denoted by MSN, which is short for mobile sensor network.

Based on the participating entities and involving scopes of communication, we divide the communication activities among the sensor network into the following three patterns:

1) Inter SNN. Since the location of a sink node in SNN is fixed, the topology of SNN is almost invariable under normal circumstances. However, the communication activities among the sink nodes in SNN are conducted wirelessly.

2) Inter MSN. Since MSN is a mobile ad hoc network, the topology of MSN keeps changing. The communication activities among the sensor nodes in MSN are conducted by a wireless network.

3) Mixed. Unlike the above two communication patterns, the Mixed pattern indicates the communication between a sensor node in MSN and a sink node in SNN.

3.1 Connectivity Model

Though the sink nodes in SNN are stationary, the communication activities among them are conducted wirelessly.

Furthermore, the level of connectivity of the manufacturing network should be considered in the perspective of the MSN and the SNN. In general, the connectivity between two wireless sensor nodes are dominated by their radio ranges. In the above manufacturing network, it is easy to understand that the connectivity of the whole network is more robust than a pure wireless network. For simplicity, we consider four sensor nodes within a two-dimensional plane. As depicted in Fig. 3, the Euclidean distance between node ms_1 and node s_1 is d_1, the Euclidean distance between node ms_2 and node s_2 is d_2, and the Euclidean distance between node s_1 and node s_2 is d_3. Suppose nodes ms_1, ms_2, s_1, and s_2 possess the same communication range R, where $R = d_1 = d_2$, and $R < d_3 < 2R$. When nodes ms_1, ms_2, s_1, and s_2 constitute a pure wireless sensor network, it is obvious that node ms_1 and s_1 are able to communicate with each other, and so do node ms_2 and s_2. As there exists isolated parts, the connectivity of the whole network is poor. In our model, node s_1 and node s_2 are two stationary nodes in SNN, and node ms_1 and node ms_2 are two mobile sensor nodes in MSN. As node s_1 and node s_2 have wired power supply, they are able to temporarily increase the radio power to achieve a larger communication range (*e.g.*, $2R$). As a result, node ms_1 and ms_2 are able to communicate with each other mediately. Namely, both node s_1 and node s_2 act as a relay node. Thus, the connectivity of the whole network gets improved.

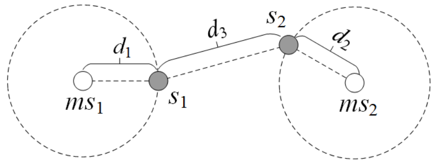

Fig. 3. Simple connectivity of four nodes.

Generally speaking, the connectivity of a wireless sensor network is the conditions of the radio links and the routes between sensor nodes. Related works focused on two major groups: one is the connectivity in diverse deployment circumstances and various operating statuses, and the other one is the variation of the connectivity for the entire life cycle of the network. To facilite the description of connectivity, a network can be routinely abstracted to an undirected graph $G(V,E)$. Here, V is the set of vertices which correspond to the nodes of the network, and E is the set of edges which correspond to the links in the network. Based on the level of connectivity, we accord three types of connectivity.

- *Fully-connected.* By a fully-connected wireless network, we mean there is always at least one link between two arbitrary vertices of the corresponding undirected graph $G(V,E)$. When there are always at least k uncrossed links between two arbitrary vertices, namely two arbitrary links of the k links share no vertices except for the two endpoints of the links, then the corresponding wireless network is a k-connected network. In practice, in a large randomly deployed wireless network, the probability an arbitrary node lies out of the radio ranges of all the other nodes is always larger than 0. Thus, the chances of a fully-connected network are slim to none.
- *Approximately-connected.* When the scale of a wireless network approaches infinity, if the probability that there is always at least one link between two arbitrary ver-

tices of the corresponding undirected graph $G(V,E)$ is 1, the corresponding wireless network is approximately-connected. Similarly, when the probability that there are always at least k uncrossed links between two arbitrary vertices is 1, the corresponding wireless network is a k approximately-connected network.

- *Partially-connected*. For an undirected graph G, if two arbitrary vertices are mutually reachable, then G is *strongly connected*. When the scale of a wireless network approaches infinity, there is one and only one strongly connected component C in the corresponding undirected graph $G(V,E)$, and there are infinite vertices in C. Then the corresponding wireless network is partially-connected.

3.2 One-dimensional Network Connectivity

To construct the model of wireless communication between two sensor nodes, we assume that each sensor node is equipped with an omni-directional antenna whose radio range is r_0. When the Euclidean distance between two sensor nodes $r_e \leqslant r_0$, they are able to directly communicate with each other. For a two-dimensional wireless network, the communication between a sensor node and a sink node might be relayed by several intermediate nodes. As shown in Fig. 4, the information flow is from node n_x to node n_s, the three Euclidean distances between the four nodes n_x, n_{x+1}, n_{x+2}, and n_s are d_1, d_2, and d_3, respectively. This two-dimensional network can be transformed to a one-dimensional network consists of nodes n'_x, n'_{x+1}, n'_{x+2}, and n'_s.

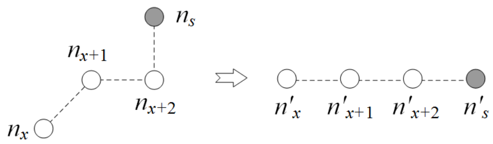

Fig. 4. Dimension reduction.

In a one-dimensional network, when a node is unable to communicate with the next node along the direction of the information flow, the node is disconnected from the network (*e.g.*, node n'_x is disconnected when $d_1 > r$). By a connected one-dimensional network, we mean iff there are no nodes which are disconnected from the network. Otherwise, the one-dimensional network is disconnected. In other words, as long as there exist two adjacent nodes between which the Euclidean distance is larger than r_0, then the network is disconnected. Theoretically speaking, the length of a one-dimensional network is infinite. Here, we consider a part of a one-dimensional network which is denoted by FN. And the length of FN is $L(FN) = x_b - x_a$, where x_b and x_a are the endpoints of FN. The number of sensor nodes within the interval $[x_a, x_b]$ is denoted by $N(x_a, x_b) = N(x_b) - N(x_a)$, where $x_b > x_a$. And the number of sensor nodes within FN is denoted by $N(FN)$. Generally, the actual number of sensor nodes is subject to a Poisson distribution. Thus, the probability of a specific number of sensor nodes lies in an interval is

$$P(N(x) = n) = \frac{E\,[N(x)]^n}{n!} e^{-E[N(x)]}, \tag{1}$$

where

$$E\,[N(x)] = \int_0^x k(x)dx, \tag{2}$$

and $k(x)$ in Eq. (2) denotes the sensor node density at x. Suppose two adjacent sensor nodes n'_x and n'_{x+1} are moving towards the access point along the one-dimensional network. The moments nodes n'_{x+1} and n'_x arrive at location l are t_{x+1} and t_x, respectively. At time t, the Euclidean distance between n'_x and n'_{x+1} is

$$d_{T_x}(t) = \int_{t-T_x}^{t} v(s)ds, \tag{3}$$

where $v(s)$ is the average velocity of sensor nodes. To facilitate the investigation of the connectivity of two adjacent sensor nodes, we introduce a threshold of the arrival interval which is denoted by T_0. When $T_x \leqslant T_0$, n'_x and n'_{x+1} stay connected during the entire movement. On the contrary, when $T_x > T_0$, the maximum Euclidean distance between n'_x and n'_{x+1} during the entire movement would be larger than r. The maximum Euclidean distance between two adjacent sensor nodes is

$$\max_{t\in\Omega}\left\{d_{T_0}(t)\right\} = \max_{t\in\Omega}\left\{\int_{t-T_0}^{t} v(s)ds\right\}, \tag{4}$$

where Ω is the set of time t_c at which n'_x and n'_{x+1} are simultaneously within FN. For a Poisson process with parameter α, we have

$$p_0 = P(T_x \leqslant T_0) = 1 - \mathrm{e}^{-\alpha T_0}, \tag{5}$$

where α denotes the average times of arrival.

Suppose the number of sensor nodes within the one-dimensional network FL is positive (*i.e.*, $N(FL) > 0$), the probability that FN is connected is

$$P_{con}(FN) = \frac{\sum_{j=0}^{\infty} p_0^{j-1} P(N(FN) = j)}{1 - P(N(FN) = 0)}. \tag{6}$$

Combining Eqs. (1) and (6), we obtain

$$P_{con}(FN) = \frac{\sum_{j=1}^{\infty} \frac{(p_0 \cdot E[N(FL)])^j}{j!} \cdot \mathrm{e}^{-E[N(FL)]}}{p_0(1 - \mathrm{e}^{-E[N(FL)]})} = \frac{\mathrm{e}^{p_0 \cdot E[N(FL)]}}{p_0(\mathrm{e}^{E[N(FL)]} - 1)}. \tag{7}$$

By Eq. (7), it can be inferred that the connectivity of a one-dimensional network is dominated by p_0 and $E\left[N(x)\right]$. By Eq. (5), p_0 is dominated by α and T_0. For a one-dimensional network whose sensor nodes are uniformly distributed, the threshold of the arrival interval is approximately r_0. Besides, the expectation of the number of sensor nodes within the one-dimensional network is also related to α.

For a sparsely deployed wireless sensor network, there is not much room for the selection of relay nodes. Meanwhile, the number of available communication links is often few. Besides, most of the involved sensor nodes of a communication link tend to move towards the access point. In this case, the one-dimensional network model is suitable to analyze the connectivity. However, the movement patterns of sensor nodes and the constitution of communication links are usually more complicated than the cases suitable for the one-dimensional network model.

4. TWO-HOP MULTI-SINK ROUTING SCHEME

4.1 Two-dimensional Network Connectivity

4.1.1 Two-hop model

As the communication activities among the sensor network is conducted by a node and its neighbor nodes, we introduce a two-hop model for the analysis the two-dimensional network connectivity. To facilitate the description of the two-hop model, we give the following definitions.

Definition 5: By a one-hop neighbor of a sensor node n_x, we mean a sensor node n_o which is one hop away from n_x, namely $d(n_x, n_o) \leqslant r_0$.

The sufficient and necessary condition for a sensor node n_o to be a one-hop neighbor of sensor node n_x is that they are able to directly communicate with each other.

The set of one-hop neighbors of n_x is calculated as

$$N_{(x,1)} = \left\{ n_{x1}, n_{x2}, \cdots, n_{xd_{(x,1)}} \right\}, \tag{8}$$

where $d_{(x,1)}$ is the number of one-hop neighbors of n_x, namely the degree of n_x.

Definition 6: By a two-hop neighbor of a sensor node n_x, we mean a sensor node n_t which is two hops away from n_x, namely $r_0 < d(n_x, n_t) \leqslant 2r_0$.

The sufficient and necessary condition for a sensor node n_t to be a two-hop neighbor of sensor node n_x is that they are not able to directly communicate with each other and n_t is a one-hop neighbor of one or more sensor node in set $N_{(i,1)}$.

The set of two-hop neighbors of n_x is calculated as

$$N_{(x,2)} = N_{(x1,1)} \cup N_{(x2,1)} \cup \cdots \cup N_{\left(xd_{(x,1)},1\right)} \backslash N_{(x,1)}. \tag{9}$$

Similar to the number of one-hop neighbors of n_x, we denote the number of two-hop neighbors of n_x by $d_{(x,2)}$, which can be calculated as

$$d_{(x,2)} = \left| \bigcup_{n_{xj} \in N_{(x,1)}} N_{(xj,1)} \backslash N_{(x,1)} \right| - 1. \tag{10}$$

As n_x is also a one-hop neighbor of its own one-hop neighbors, namely

$$n_x \in \bigcup_{n_{xj} \in N_{(x,1)}} N_{(xj,1)} \backslash N_{(x,1)}. \tag{11}$$

Hence, the calculation of $d_{(x,2)}$ excludes n_x itself. For a sensor node n_x in the sensor network, the two-hop model of n_x is composed of its one-hop neighbors and two-hop neighbors.

4.1.2 Level of connectivity

The connectivity of a two-dimensional network is more complicated than that of a one-dimensional network. We introduce a novel approach to quantify the level of connectivity for the two-dimensional network. This approach is based on the above two-hop model. For two arbitrary nodes n_x and n_{x+1}, we denote the relation between them by $f_k(n_x, n_{x+1})$, where k is the number of hops in the shortest path between n_x and n_{x+1}. When $k = 0$, n_x and n_{x+1} reside in different isolated parts of the sensor network, they are unable to communicate with each other. For $k = 1, 2, 3, 4$, and $k > 4$, five cases of the shortest path between n_x and n_{x+1} are illustrated in Fig. 5, where the relay nodes are denoted by n_{ri} ($i \in \mathbb{N}$).

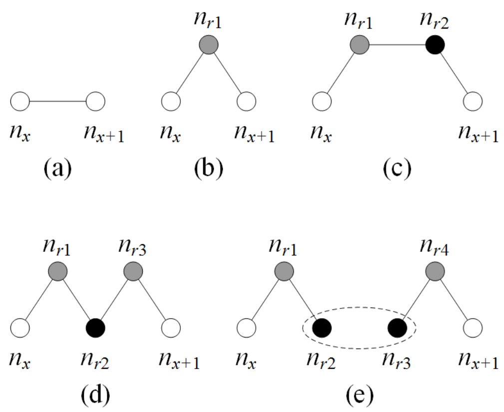

Fig. 5. Node relations based on the two-hop model.

In Fig. 5 (a), n_{x+1} is the one-hop neighbor of n_x and vice versa. For the rest four cases, the communication between n_x and n_{x+1} are conducted by the relay node(s). In Fig. 5 (b), n_{r1} is a one-hop neighbor of both n_x and x_{x+1}. For Fig. 5 (c), we consider n_{r1} and n_{r2} are a one-hop neighbor and a two-hop neighbor of n_x, respectively. And $d(n_{r2}, n_{x+1}) \leqslant r_0$, thus n_{x+1} is within radio range of the two-hop model of n_x. In Fig. 5 (d), n_{r2} is the two-hop neighbor of both n_x and n_{x+1}, thus the corresponding two two-hop models are overlapped. In Fig. 5 (e), n_{r2} and n_{r3} are a two-hop neighbor of n_x and n_{x+1}, respectively. And the relation between n_{r2} and n_{r3} are the same with that of n_x and n_{x+1} described in the previous four cases. In addition, more complex relations between n_x and n_{x+1} can be decomposed into the combinations of the above five cases.

We define the level of connectivity between n_x and n_{x+1} by $c(n_x, n_{x+1})$, which is formulated as

$$c(n_x, n_{x+1}) = \begin{cases} 0, & k = 0 \\ 1/k, & k = 1, 2, 3, 4 \\ \frac{1}{4 + 1/c(n''_x, n''_{x+1})}, & k > 4 \end{cases} \tag{12}$$

When n_x and n_{x+1} cannot communicate with each other, let $c(n_x, n_{x+1}) = 0$. For $k > 0$, $c(n_x, n_{x+1})$ monotonically decreases with the increase of k, namely the level of connectivity between two sensor nodes is inversely proportional to the number of hops between them. For $k = 1, 2, 3$, and 4, we use the reciprocal of k to represent $c(n_x, n_{x+1})$. For $k > 4$, $c(n_x, n_{x+1})$ is calculated recursively. n''_x and n''_{x+1} are the two-hop neighbors of n_x and n_{x+1}, respectively. Besides, they are located in the shortest path between n_x and n_{x+1}. For instance, the level of connectivity $c(n_x, n_{x+1})$ in Fig. 5 (e) is

$$c(n_x, n_{x+1}) = \frac{1}{4 + 1/d(n_{r2}, n_{r3})}. \tag{13}$$

As the two-hop model of a sensor node contains a one-hop neighbor and a two-hop neighbor, each time k is greater than a multiple of 4, an iterative calculation of the level of connectivity is needed. Thus, the number of iterations is

$$R = \lfloor (k-1)/4 \rfloor . \tag{14}$$

For an individual sensor node n_x, we denote the set of sensor nodes with which n_x is able to communicate by $N(x)_k = \{n_1, n_2, \cdots, n_k\}$. The level of connectivity of n_x can be computed as

$$c(n_x) = \sum_{i=1}^{k} c(n_x, n_i). \tag{15}$$

For a sensor network which contains N sensor nodes, the level of connectivity of the whole network is

$$C(N) = \sum_{\forall n_x \in N, \forall n_{x+1} \in N} c(n_x, n_{x+1}), n_x \neq n_{x+1}. \tag{16}$$

Theoretically speaking, provided the value of R is large enough, there always exists a path for two arbitrary sensor nodes. With the increase of R, the level of connectivity for the whole network $C(N)$ keeps increasing until it reaches $\hat{c}$. In other words, there is an inflection point $(\hat{r}, \hat{c})$. For $R < \hat{r}$, $C(N)$ monotonically increases; While $R \geqslant \hat{r}$, $C(N) \equiv \hat{c}$. Moreover, let $\hat{r}$ be the degree of convergence of the network. The smaller the value of $\hat{r}$ is, the better the network converges.

4.2 r-Kruskal Algorithm for SNN

As the location of a sensor node in SNN is fixed, and the nodes in SNN have a cabled power supply, thus the topology of SNN could be considered invariable for the most part. We denote the corresponding undirected graph of SNN by $G_s(V_s, E_s)$. Thus, the communication activities among the sensor nodes in SNN could be formulated as the solution to the minimum spanning tree for $G_s(V_s, E_s)$. There are three notable methods for the problem of minimum spanning tree of an undirected graph: Kruskal [34], Boruvka [35], and Prim [36]. The above three algorithms are based on the principle of greedy. Each algorithm employs its own step-by-step solving strategy. The Prim algorithm and the Kruskal algorithm are similar. For the Boruvka algorithm and the Kruskal algorithm, the latter is feasible when there are edges with the same weight value, while the former is not feasible.

In our model, we propose an *r-Kruskal* algorithm to establish the communication within SNN. This algorithm is based on the idea of the original Kruskal algorithm. Without loss of generality, suppose the N_s sink nodes $s_1, s_2, \cdots, s_{N_s}$ are randomly deployed in the $m \times n$ two-dimensional surface. We introduce an $m \times n$ matrix $D_s = (ds_{ij})$ to record the deployment of sink nodes in set sn. Each grid is able to accommodate one sink node at most, namely a grid is either empty or occupied by exactly one sink node. For a grid gr_{ij}, if it contains a sink node s_k, then $ds_{ij} = k$. Similarly, if it contains a mobile sensor node ms_k, then $ds_{ij} = -k$. When gr_{ij} is empty, let $ds_{ij} = 0$.

The application of the original Kruskal algorithm requires a prerequisite that the considered undirected graph is *fully-connected*. However, in practice, the actual deployment of sensor nodes is unable to meets this condition in the vast majority of situations. Thus, we propose an improved *r-Kruskal* algorithm for the purpose of obtaining a degraded minimum spanning tree which facilitates the communication activities among the sink nodes in SNN.

The basic idea behind our *r-Kruskal* algorithm is that specific operations are centered on the edge with the minimum weight value among the available edges. Specifically, for an undirected graph $G_s(V_s, E_s)$, where the number of vertices is $vs = |V_s|$ and the number of edges $es = |E_s|$. We construct a disconnected graph $T_s = \{V_s, E_c\}$, where $E_c = \varnothing$. Initially, there are no edges in graph T_s. In other words, graph $T_s = \{V_s, \varnothing\}$ and it only contains vs vertices. Each vertex itself constitutes an independent connected component. For all edges in set E_s, we consider one of the edges with the minimum weight value and denote it by e_0. When $w(e_0) > r_0$, the *r-Kruskal* algorithm terminates. While $w(e_0) \leqslant r_0$, the *r-Kruskal* algorithm proceeds as follows: if the two vertices of edge e_0 fall in different connected components, then e_0 is removed from set E_s and added to set E_c; otherwise, the two vertices of edge e_0 fall in the same connected component, then this particular edge e_0 is never considered again. Repeat the above edge operation until none of the edges in set E_s with a weight value which is less than or equal to r_0. The detailed *r-Kruskal* algorithm is shown in Algorithm 1.

Algorithm 1 *r-Kruskal*(E_s, V_s, r_0)

$E_c \leftarrow \varnothing, V_c \leftarrow \varnothing, C \leftarrow \varnothing, E_r \leftarrow \varnothing$
for $i = 1$ **to** $|V_s|$ **do**
  $E_{c_i} \leftarrow \varnothing, V_{c_i} \leftarrow \varnothing$
  $c_i = (E_{c_i}, V_{c_i} \leftarrow V_{c_i} \cup \{v_i\})$
  $C \leftarrow C \cup \{c_i\}$
end for
while $w(e_0 = e_i | \min_{1 \leqslant i \leqslant |E_s|} w(e_i)) \leqslant r_0$ **do**
  if $v_{e_{0,1}} \in c_m$ && $v_{e_{0,2}} \in c_n$ && $c_m \neq c_n$ **then**
    $E_s = E_s \setminus \{e_0\}, E_c = E_c \cup \{e_0\}$
    $V_c = V_c \cup \{v_{e_{0,1}}, v_{e_{0,2}}\}$
  else
    $E_r = E_r \cup \{e_0\}$
  end if
end while
return E_c, V_c, C, E_r

Algorithm 1 takes three input parameters: set E_s, set V_s, and the radio range of a sensor node r_0. Set V_s contains the corresponding vertices of all sensor nodes in set *sn*. Thus, the number of elements in set V_s is $|V_s| = N_s$. For an edge e_i, the two endpoints of e_i is denoted by $v_{e_{i,1}}$ and $v_{e_{i,2}}$. We use the weight value of e_i to represent the Euclidean distance between the corresponding two sensor nodes, namely $w(e_i) = d(v_{e_{i,1}}, v_{e_{i,2}})$. For set V_s, there are totally $N_s(N_s - 1)/2$ different pairs of vertices. Thus, the number of elements in set E_s is $|E_s| = N_s(N_s - 1)/2$.

When two arbitrary vertices in $G_s(E_s, V_s)$ are connected, $G_s(E_s, V_s)$ is a connected graph. The actual effect of Algorithm 1 degenerates into that of a classical Kruskal algorithm. In other words, a minimum spanning tree of $G_s(E_s, V_s)$ which contains all vertices in set V_s would eventually be obtained. On the contrary, when $G_s(E_s, V_s)$ is a disconnected graph, the effect of Algorithm 1 needs further investigation. Suppose there exists a connected component which is the maximal connected subgraph. Here, the term "maximal" indicates the number of vertices contained in the subgraph is maximal. We denote this subgraph by $G_c(E_c, V_c)$, where $E_c \subset E_s$ and $V_c \subset V_s$. The vertices which are not contained in $G_c(E_c, V_c)$ constitute set $V_r \equiv V_s \setminus V_c$, and $1 \leqslant |V_r| \leqslant vs - 2$. In other words, there are a connected component and $|V_r|$ isolated points. The connected component contains r edges and $r+1$ vertices. The number of isolated points $|V_r| = |V_s| - r - 1$.

As stated above, the sink nodes in SNN are stationary for a given application scenario. To determine the routing in SNN, we just need to run Algorithm 1 once, rather than regularly. When there is a minimum spanning tree of $G_s(E_s, V_s)$, the communication activities among all sink nodes in SNN could be done based the minimum spanning tree. On the contrary, when there exist isolated sensor node(s), possible communication activities between an isolated node and a node within the minimum spanning tree need to be relayed by the nodes in MSN. The mobility of the mobile sensor nodes has a considerable influence on the topology and connectivity of the network. Thus, both the path establishment and subsequent data transmission in the wireless sensor network are challenging and deserve efficient solutions. A proactive routing scheme regularly updates the routing table regardless of whether it is needed. Hence, when the frequency of data transmission is low, the cost performance of proactive path establishment methods is considered high. Furthermore, the mobility of sensor nodes may significantly degrades the actual performance. It is probable that a routing table becomes unable to work soon after the routing table gets updated. Besides proactive path establishment methods, there are also reactive and hybrid path establishment methods. The main idea of a reactive path establishment method is the formation of a routing path takes place upon a transmission demand. In specific, a reactive path establishment method does not maintain the routing table periodically. A routing path is generated based on the routing demand and the network status in real time. This feature facilitates the adaptation to changes of topology and avoids a substantial amount of inefficient regular routing table updates. However, the reactive strategy has a major drawback. During the establishment of a routing path, there is a considerable amount of message overhead. In specific, control messages for routing discovery and data messages for sensor data transmission are flooded. As the adoption of flooding is prone to incur network congestion, the detailed flooding policy should be prudently designed for the purpose of reducing message overhead.

4.3 Two-hop Routing

The two-hop multi-sink routing scheme is based on the two-hop model and the *r-Kruskal* algorithm. This scheme covers all communication activities among MSN and SNN. Mobile sensor nodes periodically send sensor data messages to a sink node in SNN. For a mobile sensor node n_m in MSN, we denote the sending frequency of a sensor data message by f_m. To investigate the operations of node n_m and its one/two-hop neighbors, we consider a time period $[t_a, t_b]$, where $t_a < t_b$. We make a premise that the time period $[t_a, t_b]$ is long enough to observe the behaviors of all nodes in the whole network. For the sake of simplicity, we assume that the sending frequencies of a sensor data message for all nodes in MSN remain constant during this period. For node n_m, it receives sensor data messages from all its one-hop neighbors. As the sending frequencies of a sensor data message vary from node to node, the sensor data messages from the one-hop neighbors arrive at node n_m asynchronously.

For node n_x in the whole network, we denote the total number of sensor data messages it received by $d_{(x,r)}$, which can be computed as

$$d_{(x,r)} = \sum_{n_{xj} \in N_{(x,1)}} \left\lfloor f_{xj} \cdot (t_b - t_a) \right\rfloor . \tag{17}$$

In general, a mobile sensor node and a sink node both forward the received sensor data messages. However, a sink node itself does not issue a sensor data message, while a mobile sensor node issues its own sensor data messages. Both a mobile sensor node and a sink node send sensor data messages to all its one-hop neighbors. For a mobile sensor node n_m, we denote the number of sensor data messages it sends by

$$d_{(m,s)} = d_{(m)} + d_{(m,f)}, \tag{18}$$

where $d_{(m)}$ is the number of sensor data messages issued by node n_m itself, namely $d_{(m)} = f_m \cdot \lfloor (t_b - t_a) \rfloor$. And $d_{(m,f)}$ is the number of sensor data messages forwarded by n_m. Similarly, for a sink node n_s, the number of sensor data messages it sends is denoted as

$$d_{(s,s)} = d_{(s,f)}, \tag{19}$$

where $d_{(s,f)}$ is the number of sensor data messages forwarded by node n_s. In our two-hop model, a node n_x is expected to forward the received sensor data message to all its one-hop neighbors. However, the sensor data messages come from the one-hop neighbors should not be sent back. Conventionally, packets with a positive TTL (Time-To-Live) value are forwarded in a network. In our two-hop model, a typical initial TTL value of a sensor data message is 4.

For node n_x, the TTL value of each sensor data message is within set $\{3,2,1,0\}$. Hence, the coverage of a sensor data message originally issued by node n_x is its one-hop neighbors and two-hop neighbors. This typical TTL value is expected to alleviate the flooding problem and reduce message overhead. Assume the TTL values of a received sensor data message obey a Poisson distribution

$$P\{\text{TTL} = k\} = \frac{\lambda_d^k \cdot e^{-\lambda_d}}{k!}, \lambda > 0, k = 0,1,2,3. \tag{20}$$

Thus, the number of sensor data messages forwarded by node n_x is

$$d_{(x,f)} = P_x \cdot \sum_{n_{xj} \in N(x,1)} (\lfloor f_{(xj)} \cdot (t_b - t_a) \rfloor \cdot (d_{(x,1)} - 1)), \tag{21}$$

where $P_x = \frac{5}{3} \cdot \lambda_d \cdot e^{-\lambda_d}$.

Combining Eqs. (17) and (21), we obtain

$$d_{(x,f)} = P_x \cdot d_{(x,r)} \cdot (d_{(x,1)} - 1). \tag{22}$$

Thus, for a mobile sensor node n_m, Eq. (18) is rewritten as

$$d_{(m,s)} = d_{(m)} + P_x \cdot d_{(m,r)} \cdot (d_{(m,1)} - 1), \tag{23}$$

where $d_{(m)} = f_m \cdot \lfloor (t_b - t_a) \rfloor$. Similarly, for a sink node n_s, Eq. (19) is rewritten as

$$d_{(s,s)} = P_x \cdot d_{(s,r)} \cdot (d_{(s,1)} - 1). \tag{24}$$

Note that Eq. (24) is a preliminary version. For a node n_x, a received sensor data message with the TTL value 0 will not be forwarded. Besides, for a sink node n_s, provided a received sensor data message is designated to it, the message will not be forwarded any more regardless of the TTL value. Suppose f_s percent of the received sensor data messages can be forwarded. Thus, a modified version of Eq. (24) is

$$d'_{(s,s)} = f_s \cdot P_x \cdot d_{(s,r)} \cdot (d_{(s,1)} - 1). \tag{25}$$

5. NUMERICAL RESULTS

5.1 Performance Metrics

Though wireless sensor networks has been applied to a variety of fields, data-centric is the primary feature for different kinds of monitoring applications based on wireless sensor networks. That is to say, the main purpose of a wireless sensor network is collecting data. Thus, data availability is a vital performance indicator for the operation of a wireless sensor network. Besides, our proposal is designed to tackle the flooding problem. It is of great importance to make an analysis of the message overhead.

5.1.1 Data availability

In the context of a communication network or an online application system, the term *data availability* usually involves a measurement about the relation of the number of received data replies dr and the number of data requests sent ds. For instance, there are three cases of dr and ds. When $dr = ds$, it is considered that the data availability is one hundred percent, and this is the simplest case. When $dr < ds$, either some data requests or some data replies are lost during the transmission due to unpredictable reasons. In general, possible reasons includes network congestion, routing failure (*e.g.*, node failure, routing error), malicious attacks, *etc*. In practice, when dr is observably smaller than ds, the data availability is considered unsatisfactory. To ensure acceptable performance in production environment, both data requests and data replies are broadcasted to some extent. Thus, it is possible that $dr > ds$. In this case, the data availability is greater than one hundred percent.

For a wireless sensor network, sensor data are periodically collected and aggregated to sink nodes. This unidirectional working mode simplifies the request-response message model. Hence, we turn to describe the data availability in terms of the sensor data messages sent and the received sensor data messages.

As the sensor data messages are transmitted throughout the whole network, it is useful to consider the degree of data availability for both mobile sensor nodes and sink nodes. For an individual node n_x, the data availability is defined as the ratio of the number of received sensor data messages to the number of sensor data messages sent, which is denoted as

$$da_x = \frac{d_{(x,r)}}{d_{(x,s)}} = \begin{cases} d_{(m,r)}/d_{(m,s)}, & n_x \in ms \\ d_{(s,r)}/d'_{(s,s)}, & n_x \in sn \end{cases}. \tag{26}$$

Based on the definition of data availability of an individual sensor node, we formulate the data availability of the network as

$$da_N = \frac{1}{N_m}\sum_{i=1}^{N_m}\frac{d_{(mi,r)}}{d_{(mi,s)}} + \frac{1}{N_s \cdot \rho}\sum_{j=1}^{N_s}\frac{d_{(sj,r)}}{d'_{(sj,s)}}. \tag{27}$$

As shown in Eq. (27), the data availability of the network is a composite of two parts. As the radio range of a sink node is ρ times as long as that of a mobile sensor node, the calculation of the data availability of the network should a conform to a uniform measure of data availability. Thus, the parameter ρ is involved for the group of sink nodes.

5.1.2 Message overhead

As mentioned above, to deal with packet loss and guarantee data availability, a traditional request-response message model usually adopts certain broadcast strategy. For a wireless sensor network, sensor data messages are duplicated and broadcasted. For a particular sensor data message, it is originally issued by a mobile sensor nodes, and expected to be received by a specific sink node. In other words, the original copy and the corresponding duplicates are destined for the same sink node. Let $sd_{(s,m,k)}$ be a sensor data message originally issued by mobile sensor node n_m, its destination is sink node n_s. We denote the extra duplicates of $sd_{(s,m,k)}$ by set $dup(sd_{(s,m,k)})$ and the corresponding messages eventually received by set $sr_{(s,m,k)}$, where $\left|sr_{(s,m,k)}\right| \leqslant \left|dup(sd_{(s,m,k)})\right| + 1$. During the time period $[t_a, t_b]$, node n_s received σ different kinds of sensor data messages, and the numbers of each kind are $n_1, n_2, \cdots, n_\sigma$. Here, a sensor data message is discriminated by the two parameters m and k. Namely, the number of different combinations of m and k is σ. We consider the message overhead from the perspective of a sink nodes. For node n_s, the message overhead can be calculated as

$$mo_s = \frac{1}{\sigma}\sum_{i=1}^{\sigma}\frac{\left|dup(sd_{(s,i)})\right| + 1 - n_i}{n_i}. \tag{28}$$

The message overhead of the network is formulated as

$$mo_s = \frac{1}{N_s}\sum_{i=1}^{N_s} mo_s. \tag{29}$$

5.2 Experiments and Analysis

To evaluate our proposal, we developed a simulation platform with NS-3 [37]. The experiments are specially designed to support the investigation of the model features, message overhead, and data availability.

To facilitate the presentation, we call our proposal THMS (Two-hop Multi-sink) for short. The THMS model defines the connectivity model between two individual nodes and accords detailed formulation about one-dimensional network connectivity and two-dimensional network connectivity. Based on these building blocks, we describe the degree of the connectivity of the network. As shown in Fig. 6, the average degree of connectivity of the network is depicted. The three curves denotes $N_m = 300$, 600, and 900, respectively. With the increase of the number of sink nodes, the average degree of connectivity of the network monotonically increases. Besides, when the current number of sink nodes are large, the same increment of the number of sink nodes results more significant improvement in the average degree of connectivity of the network than that of a small current number of sink nodes. In other words, the rising tendency becomes sharper as the number of sink nodes increases.

To investigate the message overhead of a sink node, we consider three settings of sink nodes and mobile sensor nodes. The ratio of the number of sink nodes to the number of mobile sensor nodes are kept as 1/10. We prefer to observe the differences of message overhead for a sink node in diverse scales of network. As shown in Fig. 7, the values of message overhead for $N_m = 600$ and $N_s = 60$ are gathered around 20%, namely the red crosses. Similarly, the values of message overhead for $N_m = 1200$ and $N_s = 120$ are gathered around 40%, namely the blue circles. For $N_m = 1800$ and $N_s = 180$, the overall distribution of the green asterisks exhibits no distinctive features. In fact, based on extensive simulations, we conclude that all the above three cases of values approximatively obey certain normal distributions. Moreover, with the increase of the scale of the network, the mean and the variance of a normal distribution both increase. For instance, the means of the red crosses, the blue circles, and the green asterisks in Fig. 7 are 20%, 40%, and 60%, respectively. While the corresponding variances are also proportionable as the means are.

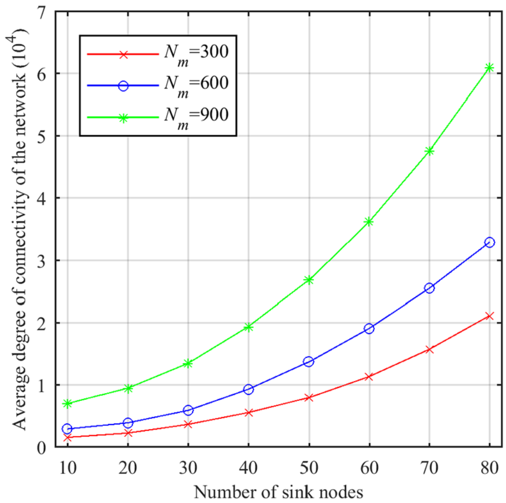

Fig. 6. Average degree of connectivity of the network.

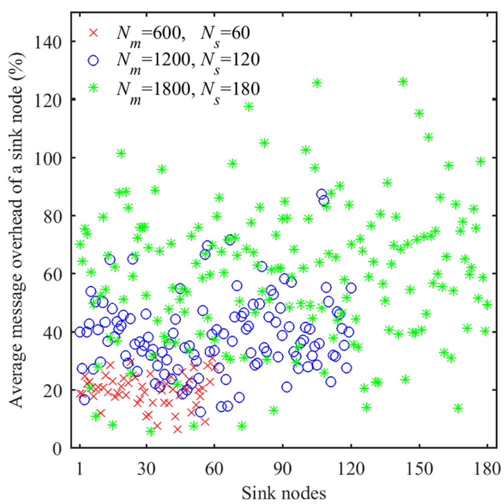

Fig. 7. Average message overhead of a sink node.

For the data availability of the network, we consider the number of mobile sensor nodes $N_m = 100$, 200, and 300. The number of sink nodes N_s is selected in the range of $[10, 160]$. As shown in Fig. 8, the overall trends of the three curves are analogous. Each curve possesses an inflection point $(\hat{N}_s, \hat{da}_N)$. Besides, the average data availability of the network increases when $N_s < \hat{N}_s$, while decreases when $N_s > \hat{N}_s$. At the left sides of the inflection points of $N_m = 100$ and $N_m = 200$, the average data availability of the network for $N_m = 100$ increases more dramatically than that of $N_m = 200$, so does $N_m = 200$ and

$N_m = 300$. Thus, the degree of the increase of the average data availability of the network is inversely proportional to the number of the mobile sensor nodes. Similarly, we come to the conclusion that the degree of the decrease of the average data availability of the network is also inversely proportional to the number of the mobile sensor nodes. To further

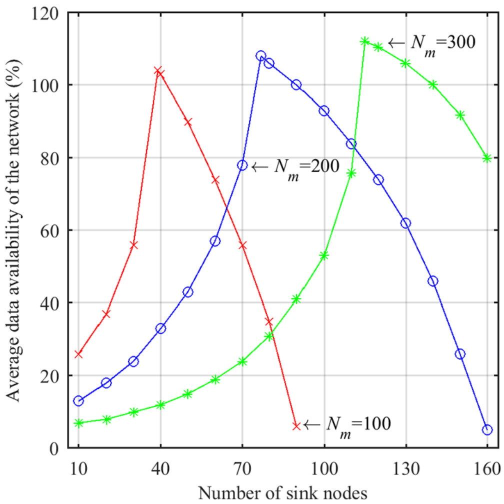

Fig. 8. Average data availability of the network.

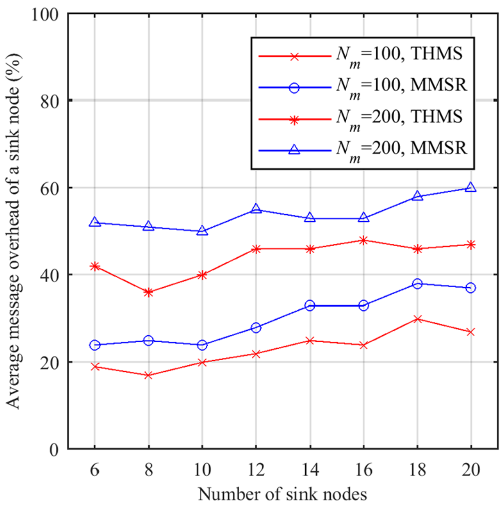

Fig. 9. Comparison of message overhead.

investigate the performance of our proposal, we make a comparison between THMS and MMSR [38]. The MMSR is based on three sink nodes, and all of them are mobile. As the sink nodes in THMS are stationary, to facilitate the comparison, we make the three sink nodes stationary and transform the mobility to the relative movement of mobile sensor nodes. Furthermore, the number of sink nodes are selected in $[6, 20]$. As shown in Fig. 9, the overall performance of our method is better than the MMSR in terms of the average message overhead of a sink node. With the increase of the number of sink nodes, both THMS and MMSR exhibit an slight increasing trend with some fluctuations. Besides, the increase of the number of mobile sensor nodes introduces considerable additional message overhead. This indicates that the amount of message overhead is proportional to the number of mobile sensor nodes. As mentioned above, data availability is a vital performance indicator for the operation of a wireless sensor network. Hence, the performance is worth discussing when the value of data availability is greater than a threshold. In practice, the data availability of a system running in a production environment should approaches 100% under normal circumstances. For the purpose of a comprehensive illustration, we let the threshold be 40%, which allows more information to be showed. As Fig. 10 depicted, the data availability of the network for both THMS and MMSR are illustrated from three aspects: maximum, minimum, and mean. For instance, when there are 30 sink nodes and 100 mobile sensor nodes, the leftmost red solid line shows the maximum, minimum, and mean of the data availability of the network for THMS. The maximum, minimum, and mean are 62, 53, and 56. Here, the mean is calculated for the 30 sink nodes. The maximum and the minimum are denoted by the two endpoints of a line segment. And the mean is denoted by a plus sign within the line segment. In general, when the plus sign is closer to the minimum than the maximum, it is considered that most of the values are smaller than the mean. On the contrary, when the plus sign is closer to the maximum than the minimum, it is considered that most of the values are greater than the mean. When $N_m = 100$, for $N_s \in [30, 70]$, the plus sign of MMSR is more closer to the minimum than that of THMS in each pair of line segment. In other words, the proportion of nodes which possess values of data availability which are smaller than the mean for MMSR is greater than that of THMS. This pattern also occurs for $N_s \in [60, 110]$ when $N_m = 200$. Besides, for a given acceptable lower bound of the data availability of the network, different numbers of mobile sensor nodes demand different ranges of number of sink nodes. That is to say, an appropriate number of sink nodes for a given number of mobile senso nodes is crucial to maintain a satisfactory data availability. When the value of data availability is greater than 100%, the cost performance of the system is degraded. In general, THMS is superior to MMSR in terms of data availability of the network. However, the range of number of sink nodes which retains the data availability greater than 100% is narrow.

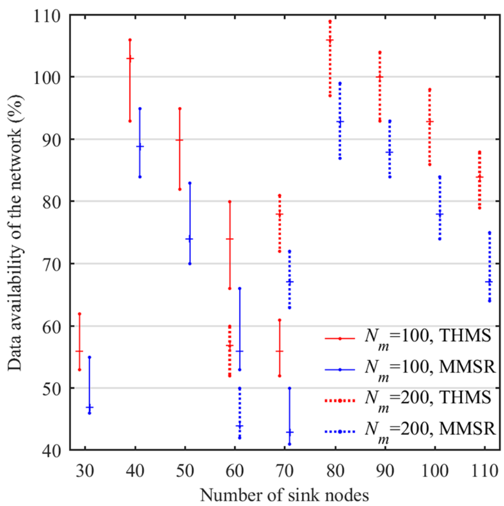

Fig. 10. Comparison of data availability.

6. CONCLUSION AND FUTURE WORK

We studied the problem of data collection in large-scale smart manufacturing facilities. To address the management of increasing amount of multi-source heterogeneous data generated in modern production lines and production process, we proposed a unified data description and management scheme. This scheme contains the measurement and physical phenomenon of a data type. In addition, the specific characteristics of a given data type are represented by a profile. This profile contains various properties which are used for describing a data type. The scalability of this scheme is that existing data types can be readily processed, while unknown data type can be accommodated by adding several extra properties. At last, both a complete and an incomplete description for a data record is feasible. For the message routing in a wireless sensor network, we discussed the connectivity model of an individual node and a network. In specific, a one-dimensional network connectivity model and a two-dimensional network connectivity model are proposed. The proposed *r-Kruskal* algorithm is able to deal with the cases of isolated nodes, which is more practical in a real world. The two-hop multi-sink routing scheme proposed in this paper aims to provide a cost-efficient message routing solution which is able to alleviate the flooding effect. Experimental results showed that our THMS model possesses excellent adaptability to the scale of the network. The characteristics of our model in terms of degree of connectivity, message overhead, and data availability are relatively steady and regular. Besides, the comparison between our model and the MMSR method showed that our model are be superior to the MMSR model in terms of both message overhead and data availability. However, there is still room for improvement. The transmission of a sensor data message involves many factors. For instance, as a considerable amount of relaying are involved, it is important to provide the analysis of message delay. Thus, more performance metrics should be modeled in a further study.

ACKNOWLEDGMENT

The authors would like to thank the anonymous reviewers whose comments and suggestions greatly helped us improve the quality and presentation of this paper.

REFERENCES

1. L. Monostori, B. Kadar, T. Bauernhansl, S. Kondoh, S. Kumara, G. Reinhart, O. Sauer, G. Schuh, W. Sihn, and K. Ueda, "Cyber-physical systems in manufacturing," *Cirp Annals*, Vol. 65, 2016, pp. 621-641.
2. R. Baheti and H. Gill, "Cyber-physical systems," *The Impact of Control Technology*, Vol. 12, 2011, pp. 161-166.
3. L. D. Xu, E. L. Xu, and L. Li, "Industry 4.0: state of the art and future trends," *International Journal of Production Research*, Vol. 56, 2018, pp. 2941-2962.
4. F. Tao, Q. Qi, A. Liu, and A. Kusiak, "Data-driven smart manufacturing," *Journal of Manufacturing Systems*, Vol. 48, 2018, pp. 157-169.
5. J. Lee, E. Lapira, B. Bagheri, and H.-A. Kao, "Recent advances and trends in predictive manufacturing systems in big data environment," *Manufacturing Letters*, Vol. 1, 2013, pp. 38-41.
6. L. D. Xu and L. Duan, "Big data for cyber physical systems in industry 4.0: a survey," *Enterprise Information Systems*, Vol. 13, 2019, pp. 148-169.

7. J. Lee, H. Davari, J. Singh, and V. Pandhare, “Industrial artificial intelligence for industry 4.0-based manufacturing systems,” *Manufacturing Letters*, Vol. 18, 2018, pp. 20-23.
8. J. Lee, E. Lapira, S. Yang, and A. Kao, “Predictive manufacturing system-trends of next-generation production systems,” *IFAC Proceedings Volumes*, Vol. 46, 2013, pp. 150-156.
9. J. Lee and E. Lapira, “Predictive factories: the next transformation,” *Manufacturing Leadership Journal*, Vol. 20, 2013, pp. 13-24.
10. https://www.fraunhofer.de/en.html, 2019.
11. https://www.bitkom.org/EN/index-EN.html, 2019.
12. S. Heng, “Industry 4.0: Huge potential for value creation waiting to be tapped,” *Deutsche Bank Research*, 2014.
13. M. Hermann, T. Pentek, and B. Otto, “Design principles for industrie 4.0 scenarios,” in *Proceedings of IEEE 49th Hawaii International Conference on System Sciences*, 2016, pp. 3928-3937.
14. L. Atzori, A. Iera, and G. Morabito, “The internet of things: A survey,” *Computer Networks*, Vol. 54, 2010, pp. 2787-2805.
15. H. Kagermann, “Change through digitization–value creation in the age of industry 4.0,” *Management of Permanent Change*, Springer, 2015, pp. 23-45.
16. M. T. Hompel and B. Otto, “Technik fur die wandlungsfahige logistik. industrie 4.0,” in *Deutscher Materialfluss-Kongress*, Vol. 23, 2014, pp. 117-126.
17. S. Jaschke, “Mobile learning applications for technical vocational and engineering education: The use of competence snippets in laboratory courses and industry 4.0,” in *Proceedings of IEEE International Conference on Interactive Collaborative Learning*, 2014, pp. 605-608.
18. H. Kagermann, J. Helbig, A. Hellinger, and W. Wahlster, ”Recommendations for implementing the strategic initiative INDUSTRIE 4.0: Securing the future of German manufacturing industry,” Final report of the Industrie 4.0 Working Group, Forschungsunion, 2013.
19. S. Wang, J. Wan, D. Li, and C. Zhang, “Implementing smart factory of industrie 4.0: an outlook,” *International Journal of Distributed Sensor Networks*, Vol. 12, 2016, p. 3159805.
20. J. Lee, “Industry 4.0 in big data environment,” *German Harting Magazine*, Vol. 26, 2013, pp. 8-10.
21. M. Yuriyama and T. Kushida, “Sensor-cloud infrastructure-physical sensor management with virtualized sensors on cloud computing,” in *Proceedings of IEEE 13th International Conference on Network-Based Information Systems*, 2010, pp. 1-8.
22. S. Madria, V. Kumar, and R. Dalvi, “Sensor cloud: A cloud of virtual sensors,” *IEEE Software*, Vol. 31, 2014, pp. 70-77.
23. C.-F. Lai, H. Wang, H.-C. Chao, and G. Nan, “A network and device aware qos approach for cloud-based mobile streaming,” *IEEE Transactions on Multimedia*, Vol. 15, 2013, pp. 747-757.
24. C.-F. Lai, H.-C. Chao, Y.-X. Lai, and J. Wan, “Cloud-assisted real-time transrating for http live streaming,” *IEEE Wireless Communications*, Vol. 20, 2013, pp. 62-70.
25. A. Alamri, W. S. Ansari, M. M. Hassan, M. S. Hossain, A. Alelaiwi, and M. A. Hossain, “A survey on sensor-cloud: architecture, applications, and approaches,” *International Journal of Distributed Sensor Networks*, Vol. 9, 2013, pp. 917-923.

26. M. M. Hassan, B. Song, and E.-N. Huh, "A framework of sensor-cloud integration opportunities and challenges," in *Proceedings of the 3rd ACM International Conference on Ubiquitous Information Management and Communication*, 2009, pp. 618-626.
27. M. Eggert, R. HauBling, M. Henze, L. Hermerschmidt, R. Hummen, D. Kerpen, A. N. Perez, B. Rumpe, D. ThiBen, and K. Wehrle, "Sensorcloud: Towards the interdisciplinary development of a trustworthy platform for globally interconnected sensors and actuators," *Trusted Cloud Computing*, Springer, 2014, pp. 203-218.
28. S. Olariu, A. Wada, L. Wilson, and M. Eltoweissy, "Wireless sensor networks: leveraging the virtual infrastructure," *IEEE Network*, Vol. 18, 2004, pp. 51-56.
29. T. Ojha, M. Khatua, and S. Misra, "Tic-tac-toe-arch: a self-organising virtual architecture for underwater sensor networks," *IET Wireless Sensor Systems*, Vol. 3, 2013, pp. 307-316.
30. C. Gao, Z. Tian, Y. Chen, and Z. Wang, "A cost-efficient virtual sensor management scheme for manufacturing network in smart factory," *Journal of Information Science and Engineering*, Vol. 35, 2019, pp. 1075-1097.
31. S. S. Stevens *et al*., "On the theory of scales of measurement,"*Science*,*NewSeries*, Vol. 103, 1946, pp. 677-680.
32. W. Kirch, (ed.), *Scale of Measurement*, Springer Netherlands, Dordrecht, 2008, pp. 1279-1279.
33. W. Kirch, (ed.), *Level of Measurement*, Springer Netherlands, Dordrecht, 2008, pp. 851-852.
34. J. B. Kruskal, "On the shortest spanning subtree of a graph and the traveling salesman problem," in *Proceedings of the American Mathematical Society*, Vol. 7, 1956, pp. 48-50.
35. J. Nesetril, E. Milkova, and H. Nesetrilova, "Otakar boruvka on minimum spanning tree problem translation of both the 1926 papers, comments, history," *Discrete Mathematics*, Vol. 233, 2001, pp. 3-36.
36. R. C. Prim, "Shortest connection networks and some generalizations," *The Bell System Technical Journal*, Vol. 36, 1957, pp. 1389-1401.
37. https://www.nsnam.org/, 2019.
38. J. Wang, L. Zuo, J. Shen, B. Li, and S. Lee, "Multiple mobile sink-based routing algorithm for data dissemination in wireless sensor networks," *Concurrency and Computation: Practice and Experience*, Vol. 27, 2015, pp. 2656-2667.

Cong Gao received the Ph.D. degree in Computer Architecture from Xidian University, Xi'an, China, in 2015. Currently, he is an Assistant Professor in the School of Computer Science and Technology at Xi'an University of Posts and Telecommunications, Xi'an, China. His current research interests include data sensing and fusion, network and information security, and service computing.

Zhongmin Wang received the Ph.D. degree from Beijing Institute of Technology, Beijing, China, in 2000. Currently, he is a Professor in the School of Computer Science and Technology at Xi'an University of Posts and Telecommunications. His current research interests include embedded intelligent perception, big data processing and application, and affective computing.

Yanping Chen received the Ph.D. degree from Xi'an Jiaotong University, Xi'an, China, in 2007. Currently, she is a Professor in the School of Computer Science and Technology at Xi'an University of Posts and Telecommunications, Xi'an, China. Her current research interests include service mining, service computing, and network management.

Zhenzhou Tian received the B.S. degree and Ph.D. degree in Computer Science and Technology from Xi'an Jiaotong University, Xi'an, China, in 2010 and 2016, respectively. He is currently a Lecturer in the School of Computer Science and Technology at Xi'an University of Posts and Telecommunications. His research interests include trustworthy software, software plagiarism detection and software behavior analysis.