Routing in Wireless Sensor Networks based on Generalized Data Stack Programming Model

Hala Elhadidy1, Rawya Rizk[1]
1Electrical Engineering Department, Port Said
University Port Said, Egypt

Hassan T. Dorrah[2]
[2]Electrical Engineering Department, Cairo University,
Giza, Egypt

Abstract—**Generalized Data Stack Programming (GDSP) model describes that any affected activity or varying environment is intelligently self-recorded inside the system in form of stack-based layering types where stack is re-defined to be one of six classes. The multi-stacking network is presented thinking of that any system connects to other system to work properly. This addresses a novel way to investigate and analyze any system. Wireless Sensor Networks (WSNs) monitor the environment and take action accordingly. However, WSN suffers from some weakness due to nodes failure or interference which affects the network topology and the routing table at each node. In this paper, the GDSP model is applied on the routing problems in WSNs. A history matrix at the user side is proposed to retrieve and backward the events affected the network.**

Keywords— GDSP; history matrix; multi-stacking network; network topology; routing table; WSN

I. INTRODUCTION

Any physical system is a collection of interconnected physical parts or elements to perform desired function [1]. Since all physical elements have properties that change with environment and age, we cannot always consider the parameters of a control system to be completely stationary over the entire operating life of the system [2]. Therefore, the controller in any system updates when new measurement data arrives or at a specific time or both as the system can be event-driven or time-driven system [3].

Wireless sensor network (WSN) is an example of the event-driven systems where many of its nodes have the ability to sense data from environment, perform simple computations and transmit this data using wireless interfaces to the base station (sink) either directly or in a multi-hop fashion through neighbors to take action accordingly [4]. All the nodes are working co-operatively to accomplish pre-specified mission objectives, such as minimum exploration time or complete monitoring coverage [5, 6]. Routing in WSNs is a challenge issue [7]. Routing protocols need to adapt to any changes to the application requirements and network dynamics during run-time of WSNs. Routing typically begins with neighbor discovery. Nodes send rounds of messages (packets) and build local neighbor tables. Routing tables in WSNs include the minimum information of each neighbor location, nodes' remaining energy, delay via that node, and an estimate of link quality [8]. Once the tables exist, in most WSN routing algorithms, messages are directed from a source location to a sink address based on geographic coordinates. Due to node mobility and failure in WSN, the network topology may be continuously changing with time and therefore the routing table of each node is also changing dynamically.

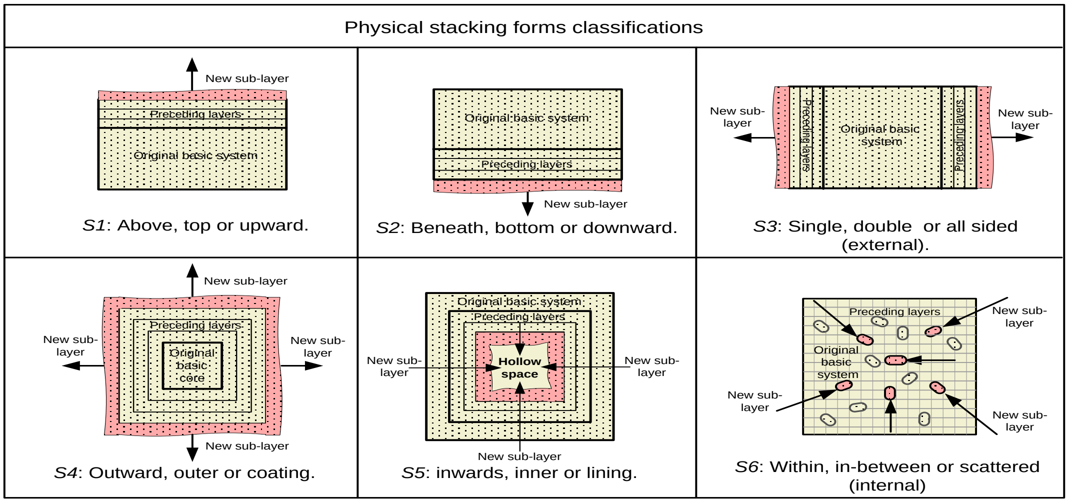
On the other hand, system changes are transferred into stack-based changes in physical system layers with different stack classes and categories in time-driven systems. However, system changes are transferred at each event step to corresponding infinite stack-based parameters changes at basic system level and to changes in consolidity index at the higher level. The definition of the stack has been changed to be more realistic and express any system [9, 10]. Based on this new stack definition, a new Generalized Data Stack Programming (GDSP) model was presented in [11]. Since any system has many elements and each of them somehow can be represented by stack, if any operation is done to one of them, it will affect the other neighbors. Therefore the new vision of the system as a network of stacks rises and is covered in this paper. Also, this paper presents how to apply the new theory of the stack on WSNs in the case of changing the topology of the network and how the stacking network works in this case.

The rest of this paper is organized as follows: Section II presents the GDSP model. Section III presents the node failure problem in WSNs. Section IV introduces how GDSP is applicable to WSNs. Section V presents the multi-stack layering network and how it is applied to WSNs. Section VI presents conclusions and future works..

II. GENERALIZED DATA STACK PROGRAMMING (GDSP) MODEL

The proposed GDSP model depends on the idea of simulating what happen in the physical world where any object can be a stack. System changes in real life may be logically conceived to be internally stacked in the form of a new sub-layer arranged in some form in relations to the other preceding existing layer(s). This stack uses matrices to represent the system which can grow or shrink according to the change occurred to the system. The matrix formulation allows fast, direct design and reconfiguration of discrete event controllers [12]. It provides a better dynamical description and high-level interface than other popular tools for discrete event systems, such as Petri Nets. The GDSP model includes six classes of stacking which effectively forms *modular blocks* for handling various applications. Fig. 1 shows that the stack is redefined to be one of six classes. At S1, the insertion or deletion can be done at the top. The insertion or deletion can be done also at the bottom, at one or both sides, as an outer ring around the object, as an inner ring inside the object, or at any place in the object at S2, S3, S4, S5, or S6 respectively.

Physical stacking forms classifications



S1: Above, top or upward.

S2: Beneath, bottom or downward.

S3: Single, double or all sided (external).

S4: Outward, outer or coating.

S5: inwards, inner or lining.

S6: Within, in-between or scattered (internal)

Fig.1. Schematic sketches of various physical stacked-based change classifications at the system basic level based on the directions of system sub-layers or sub-stratum change

TABLE.I. GDSP MODEL

Class	Features	Matrix representation	Applications
S1	Traditional form, S1$^+$ or S1$^-$ done at top of an object.	$\xrightarrow{S1^+}$ Growing matrix $\begin{bmatrix}+&+&+\\x&x&x\\x&x&x\end{bmatrix}$ $\xrightarrow{S1^-}$ Shrinking matrix $\begin{bmatrix}x&x&x\end{bmatrix}$	The computer memory.
S2	S2$^+$ or S2$^-$ done at the bottom of an object	$[x$ $\xrightarrow{S2^+}$ Growing matrix $\begin{bmatrix}x&x&x\\x&x&x\\+&+&+\end{bmatrix}$ $\xrightarrow{S2^-}$ Shrinking matrix $\begin{bmatrix}x&x&x\end{bmatrix}$	The marine algae under water.
S3	S3$^+$ or S3$^-$ done to a side or two sides of an object	$\begin{bmatrix}x&x&x\\x&x&x\end{bmatrix}\xrightarrow{S3^+}$ Growing matrix $\begin{bmatrix}+&x&x&x\\+&x&x&x\end{bmatrix}$ or $\begin{bmatrix}+&x&x&x&+\\+&x&x&x&+\end{bmatrix}$ $\xrightarrow{S3^-}$ Shrinking matrix $\begin{bmatrix}x&x\\x&x\end{bmatrix}$ or $\begin{bmatrix}x\\x\end{bmatrix}$	External rust on an industrial pipe.
S4	S4$^+$ or S4$^-$ done to the outer ring of an object	$\begin{bmatrix}x&x&x\\x&x&x\\x&x&x\end{bmatrix}\xrightarrow{S4^+}$ Growing matrix $\begin{bmatrix}+&+&+&+&+\\+&x&x&x&+\\+&x&x&x&+\\+&x&x&x&+\\+&+&+&+&+\end{bmatrix}$ $\xrightarrow{S4^-}$ Shrinking matrix $[\,x\,]$	The tree trunk.
S5	S5$^+$ or S5$^-$ done to the inner ring of an object under a condition that the object has an empty center.	$\begin{bmatrix}x&x&x&x&x\\x&0&0&0&x\\x&0&0&0&x\\x&0&0&0&x\\x&x&x&x&x\end{bmatrix}\underset{S5^-}{\overset{S5^+}{\rightleftarrows}}\begin{bmatrix}x&x&x&x&x\\x&+&+&+&x\\x&+&0&+&x\\x&+&+&+&x\\x&x&x&x&x\end{bmatrix}$	The fats can deposit inside the artery
S6	Add an element at (a,b) then rearrange the matrix horizontally or vertically. Removing an element does not mean to remove a column or row.	$\begin{bmatrix}x&x&x\\x&x&x\\x&x&0\end{bmatrix}\xrightarrow{S6+}$ Horizontall $\begin{bmatrix}x&x&x&0\\x&y&x&x\\x&x&0&0\end{bmatrix}$ or vertical $\begin{bmatrix}x&x&x\\x&y&x\\x&x&0\\0&x&0\end{bmatrix}$ $\xrightarrow{S6^-}$ Horizontal $\begin{bmatrix}x&x&x\\x&x&0\\x&x&0\end{bmatrix}$ or vertical $\begin{bmatrix}x&x&x\\x&x&x\\x&0&0\end{bmatrix}$	Deposit of sand or minerals in the kidney, fruit worm and having a tumor under skin.

Table 1 summarizes the basic features and the matrix representation of these six classes with their applications. The codes and the mathematical representation for each class are found in [11]. It is assumed that the new layer is uniform – adding a whole row, column or ring depends on the stack class. However, the layer could be non-uniform containing segments of different nature and appearance. This case is exactly like the deposits of fats inside the human artery where the fats are not accumulated in a complete ring shape. So its matrix can be as follow:

$$\begin{bmatrix} x & x & x & x & x \\ x & 0 & 0 & 0 & x \\ x & 0 & 0 & 0 & x \\ x & 0 & 0 & 0 & x \\ x & x & x & x & x \end{bmatrix} \quad \text{will be} \quad \begin{bmatrix} x & x & x & x & x \\ x & x & 0 & 0 & x \\ x & x & 0 & 0 & x \\ x & x & 0 & 0 & x \\ x & x & x & x & x \end{bmatrix} \quad \text{or}$$

$$\begin{bmatrix} x & x & x & x & x \\ x & 0 & 0 & 0 & x \\ x & 0 & 0 & 0 & x \\ x & x & x & x & x \\ x & x & x & x & x \end{bmatrix} \text{ or even } \begin{bmatrix} x & x & x & x & x \\ x & 0 & x & x & x \\ x & 0 & 0 & 0 & x \\ x & x & x & x & x \\ x & x & x & x & x \end{bmatrix}.$$

There are many applications that use more than one stack class like in the soil. The deposits of soil will be added up increasing the height of it ($S1^+$). The decayed animals and plant remains in the soil affected its fertility ($S2^+$). Also there are some insects and animals live inside the soil. The presence of these insects looks like adding an element in a specific place ($S6^+$). WSN is an example of the applications that uses more than one stack class as will be explained in the next section.

If a node failure is discovered in a WSN, one of the solutions to maintain the reliability of the network is that, the nodes following the failed one which are lying in the opposite side of the sink move to new locations to optimize network performance while simultaneously confirming the connectivity requirements. The optimal configuration of the nodes is lying evenly on the line from the source to the sink as proven in [13]. However, moving the nodes could affect the coverage area of the network which has an arbitrary shape and therefore cause degradation in the network performance [14] as shown in Fig. 2.

III. The Node Failure Problem in WSNs

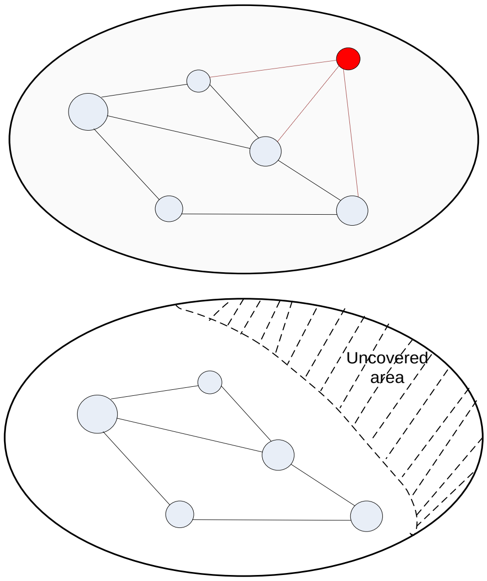


Fig.2. the difference between the covered area of a WSN before and after removing a failed node

The network topology can be represented by a matrix (N×N) where N is the number of nodes. m(a,b)= 0 or 1 where 1 means there is a connection (route) between node a and b. For example, the topology of the network shown in Fig. 3 can be represented as

$$\begin{bmatrix} 1 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 \end{bmatrix}.$$

The routing table of each node indicates the minimum information about the neighbors of the node. It may contain the remaining energy of each node as shown in Table 2 that represents the routing table of Node 3.

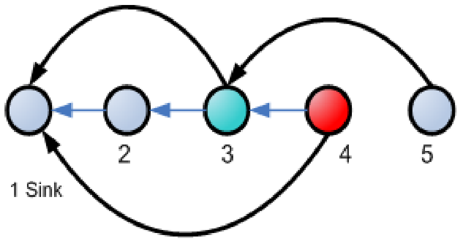


Fig.3. An example of a WSN

TABLE.II. ROUTING TABLE OF NODE 3

Node ID	Position x	Position y	Remaining energy
1	50	50	100
2	60	50	80
4	80	50	10
5	90	50	40

If a failure happens to a node, a message will pass through the network (which happens regularly) to change the topology according to the routing algorithm that is used in this network. In the example shown, if Node 4 has a failure, there are two cases [15]. First, Node 5 will take the position of Node 4 if it is able to move. Then, the topology changes according to its communication range. If this moving node can covers the communication range, then the topology of a WSN becomes as shown in Fig. 4(a) and its matrix can be represented as $\begin{bmatrix} 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 \end{bmatrix}$.

If the sensing area is shrunk which affects the coverage area and then causes degradation in network performance, a new node needs to be inserted at the end of the network as shown in Fig. 4(b) and its final matrix can be considered as $\begin{bmatrix} 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 \end{bmatrix}$.

Table 3 shows the final routing table of Node 3 in this case. It is shown from the table that, the remaining energy of Node 1 is 100% since it is important to plug in the sink node to be always alive [16] while the remaining energy of the other nodes decreases with the usages.

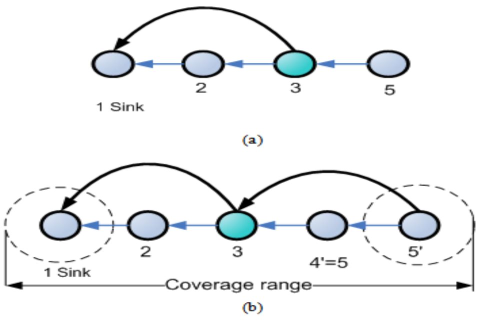


Fig.4. The network after changing the topology if Node 5 is able to move. (a) If the sensing area doesn't change, (b) If a new node is added for complete coverage.

TABLE.III. THE NEW ROUTING TABLE AFTER ADDING A NEW NODE AT THE END

Node ID	Position x	Position y	Remaining energy
1	50	50	100
2	60	50	77
4'(5)	80	50	39
5'	90	50	100

Second, if Node 5 is a static node, a new node needs to be inserted instead of Node 4 and according to its communication range the topology will change as in Figure 5 for example and its matrix can be represented as $\begin{bmatrix} 1 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 \end{bmatrix}$.

In this case, the final routing table for Node 3 is shown in Table 4.

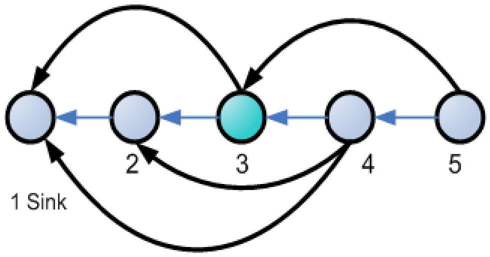


Fig.5. The network after adding a new powerful node at 4 if Node 5 is static

TABLE.IV. THE NEW ROUTING TABLE AFTER ADDING A NEW NODE AT THE POSITION OF NODE 4

Node ID	Position x	Position y	Remaining energy
1	50	50	100
2	60	50	77
4	80	50	100
5	90	50	39

IV. GDSP IMPLEMENTATION ON WSNS

By comparing the previous matrices, we can consider that the WSN is a multi-stacking application of GDSP as any physical system. In the first case, $S6^-$ will be used repeatedly for all nodes to remove the failed one m(*i*,*4*) where *i*=1,2,3,5 and then rearrange horizontally and remove m(*4*,*i*) then rearrange vertically. Adding a column at the end ($S3^+$ one side) and row at the end ($S2^+$) with values 0 and 1 according to the new node's sensing rang are equivalent to inserting the new node. In the second case, $S6^-$ will be used repeatedly for all nodes to remove the failed one m(*i*,*4*) where i=1,2,3,5 and then rearrange horizontally and remove m(*4*,*i*) then rearrange vertically. After that a new node is inserted instead of the failed Node 4 (as in the example) ($S6^+$) so m(*i*,*4*) is added with a value 1 or 0 according to the sensing range of the node and in this case will be 1 all the time then rearrange horizontally and add m(*4*,*i*)=1 and rearrange vertically.

On the other hand, it is mentioned that if the topology is changed, a round of messages is moved through the network so that each node updates its routing table. To save energy, this message will move around the failed node's neighbors only and their routing tables only are updated. In the first case and when Node 4 is removed, all the following nodes to it will move towards it then add a new one at the end. This means $S6^-$ will be applied to the routing table then rearrange the table vertically with changing the nodes ID then $S2^+$ with the information of the new node.

In the second case, another node (new 4) will be inserted. Therefore, a new entry in the routing table with the same ID of the failed node but with the new information is inserted. Applying GDSP, $S6^-$ is used to remove the failed node without rearranging either horizontally or vertically leaving an empty row. After adding a new node, its information will be in this empty row ($S6^+$ without rearranging).

At the user side and since that the user is not all the time monitoring the network, we propose a new matrix P(*i*,*j*) with a numerical entry between 0 and 1 which represents the residual energy of each node. This matrix is updated if an unusual event happens like if there is a failed node and it is removed or a new node is inserted. The new matrix begins with P(*1*,*N*) where *N* is the number of nodes in the network then adding a new row at the top every time there is a new event. This matrix represents the history of the network and it is lasted as required (programmed) a day, a week or a month. In addition, this matrix can be 3D matrix where the third dimension represents the time at which that event happens.

In the previous example, the matrix at the beginning is $[1 \quad 0.8 \quad 1 \quad 0.1 \quad 0.4]$. Then after removing Node 4, the matrix will be $\begin{bmatrix} 1 & 0.77 & 0.9 & 0 & 0.395 \\ 1 & 0.8 & 1 & 0.1 & 0.4 \end{bmatrix}$. When the new node is inserted at the previous node's location, the matrix will be $\begin{bmatrix} 1 & 0.77 & 0.94 & 1 & 0.39 \\ 1 & 0.77 & 0.95 & 0 & 0.395 \\ 1 & 0.8 & 1 & 0.1 & 0.4 \end{bmatrix}$. The user is also able to see the topology matrix that was explained before.

V. Multi-Stack Layering Network

It is always difficult to isolate changes in any component of a system from its neighboring component within in its operation domain. Each component of any system is represented by stack and the interaction between each other is done in one or both directions as in Fig. 6. To maintain the laws of preservation of mass and energy, it is important to find the reaction of each component of the system.

The evaporation process is an example of a multi-stack layering network. In the hot days, water evaporates from any water body (i.e. lakes, ponds, seas,..) makes it decrease ($S1^-$). The water particles push the air particles to replace them in the atmosphere above the water body. Water is going up to condense in the upper cold layer forming clouds.

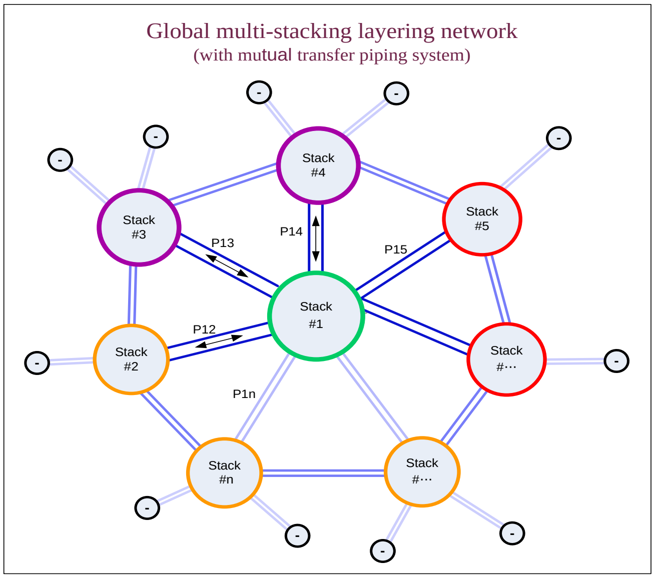
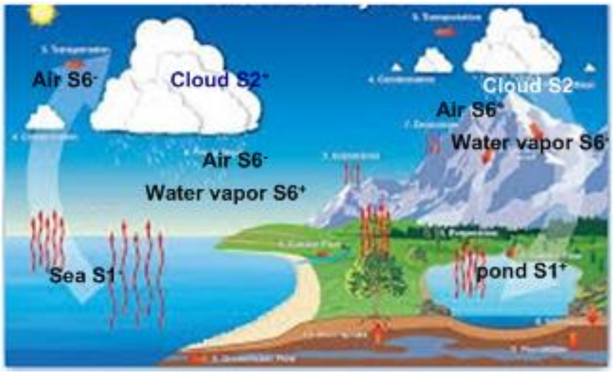
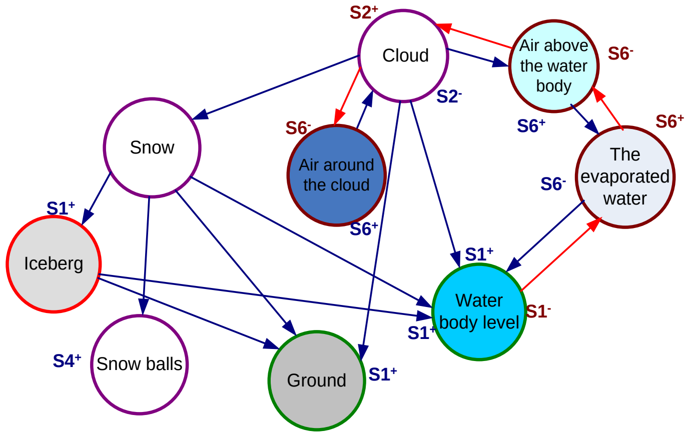


Fig.6. A general view of a multi-stack layering network [9].

The more water condenses the bigger or more clouds form. The cloud replaces the air that exists in this region pushing the air particles which they re-spread in this region. The opposite of the evaporation process is the rain in the worm regions. While if there is a cold region, the snow will be formed instead. In this case, iceberg could be formed or the snow will raise the sea level. If the snow falls on the ground, snowballs could be formed ($S4^+$) or the snow covers the ground ($S1^+$). This operation is a multi-stacking network as the system has many components and if anything happens to one of them, the rest reacts accordingly. This network is illustrated in Fig. 7. From this figure, we can say that all the multi-stack layering networks found anywhere are connected together to represent the universe.



(a)



(b)

Fig.7. General representation for the evaporation and condensation processes. (a) The evaporation and condensation processes. (b) multi-stack layering network in this processes.

In the case of WSN, as explained before there is a connection between the topology matrix, the routing table and the proposed history matrix as in Fig. 8. There are two options if the covering area is shrunk; either to insert the new node at the end of the network or insert it at the place of the failed one. In the first case (the left branch), the topology is changed as the nodes following the failed one are moved toward the sink then a new node is added at the end. The routing table is changed too with rearranging means all the following nodes will get one row up in the table after removing the row of the failed one (S6^{-}) adding a node at the end means adding a row at the end of the routing table (S2^{+}). In the second case, the failed node is removed from the network (S6^{-}) without changing the topology and in the routing table, the row of the failed one will be removed then adding a new node instead (S6^{+}) means adding a row in the routing table in the place of the removed one (S6^{+}) as in the right branch of Fig. 8.

VI. Conclusion and Future Works

It is crucial to search for new tools to analyze and model any system which provides us the chance for more understanding and deep control. The GDSP is one of them which helps us look at the system from different view and analyze it as layers connecting together. On the other hand, WSN is a rich application with many details which can be used to apply the new theory. A new matrix is proposed in this paper to retrieve and backward the history of the network so that the user follows the details of the network and does not miss any event. This history matrix is considered as a layer of the multi-stacking network in the WSN beside the routing table and the topology matrix.

What introduced in this paper is just opening the door for more works with more details about WSNs and trying to apply GDSP model from different respects. The required are; studying the effect of the new history matrix on the WSN and studying more systems and practical applications after applying GDSP to see the advantages of the new model where this paper is just putting a new theory with some applications without details.

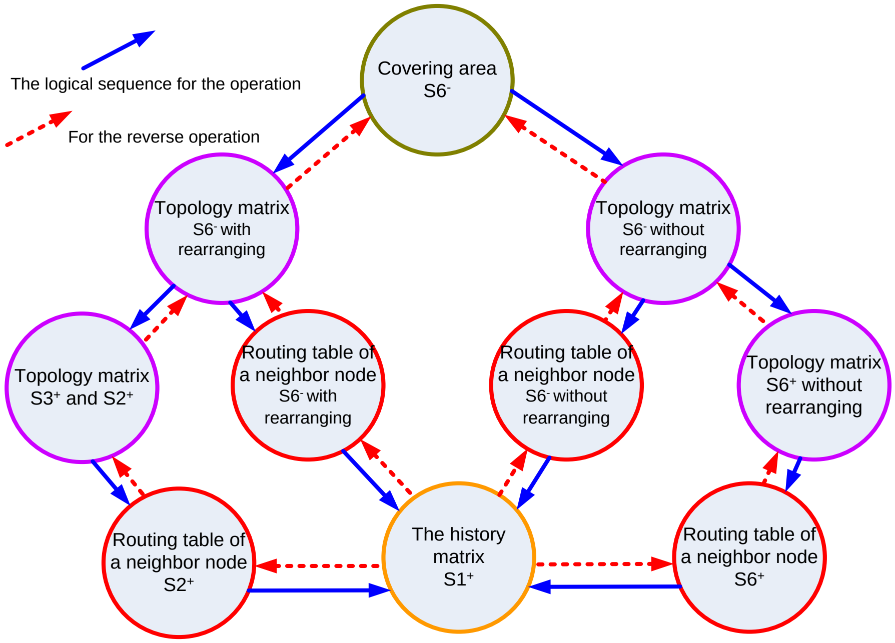

The logical sequence for the operation
For the reverse operation
Covering area S6-
Topology matrix S6- with rearranging
Topology matrix S6- without rearranging
Topology matrix S3+ and S2+
Routing table of a neighbor node S6- with rearranging
Routing table of a neighbor node S6- without rearranging
Topology matrix S6+ without rearranging
Routing table of a neighbor node S2+
The history matrix S1+
Routing table of a neighbor node S6+

Fig.8. A multi-stacking network of WSN

References

[1] *Y. Singh and A. Agrawal," Control systems, theory, problems and solutions," Galgotia publications Put. Ltd., India, 1st edn, 2008.*

[2] B. C. Kuo," Automatic control systems," Prentice-Hall of India, 7th edn., 1999.

[3] Ch. G. Cassandras and S. Lafortune," Introduction to Discrete Event Systems," Springer Science Business Media LLC, 2nd edn., 2008.

[4] R. Rizk, "Quality of Services in Wireless Sensor Networks. In Wireless Sensor Networks: From Theory to Applications, " Ibrahiem M.M.EI Emary and Ramakrishnan S. (Ed.), CRC Press, Taylor & Francis Group, USA, August 2013, http://www.crcpress.com/product/isbn/9781466518100.

[5] R. Rizk, S. Magdy, and F. Zaki, "Energy efficiency of virtual multi-input, multi-output based on sensor selection in wireless sensor networks," Wireless Communications and Mobile Computing, John Wiely & Sons, doi: 10.1002/wcm.2310, 2012.

[6] J. Zhang, X. Shen, G. Dai, Y. Feng, S. Tang, and C. Lv, "Energy-efficient lossy data aggregation in wireless sensor networks" , Adhoc & Sensor Wireless Net, vol 11, no. 1-2, pp. 11-35, 2011.

[7] J. A. Stankovic, "Research challenges for wireless sensor networks," Special issue on embedded sensor networks and wireless computing (SIGBED), vol 1, no.2, July 2004.

[8] Y. Pyokin, E. Jung, Y. Park, A radio-aware routing algorithm for reliable directed diffusion in lossy wireless sensor networks, Sensors, 2009, 9, (10), pp. 8047-8072.

[9] H. T. Dorrah, "Consolidity; Stack-based systems change pathway theory elaborated," Ain Shams Eng. J., 2014, http://dx.doi.org/10.1016/j.asej.2013.12.002

[10] H. T. Dorrah, "Supplement to consolidity: moving opposite to built-as-usual practices, " Elsevier Ain Shams Eng. Journal, vol 4, no. 4, pp. 783-803, 2013.

[11] H. Elhadidy, R. Rizk, and H. T. Dorrah, "A new generalized data stacking programming (GDSP) model", Proc. of 9th ICENCO Egypt, Dec. 2013, pp. 78-84, DOI:10.1109/ICENCO.2013.6736480.

[12] J. Mireles and F. Lewis, "Intelligent material handling: development and implementation of a matrix-based discrete event controller," IEEE Transactions on Industrial Electronics, vol. 48, no. 6, pp. 1087– 1097, Dec. 2001.

[13] D. Goldenberg, J. Lin, A. Morse, B. Rosen, and Y. Yang, "Towards mobility as a network control primitive," proc. of the ACM Int. Symp. on Mobile Ad Hoc Net. and Comp. (MobiHoc), pp. 163-174, May 2004.

[14] L. Lazos, R. Proovendran, and J. A. Ritcey, "Probabilistic detection of mobile targets in heterogeneous sensor networks," proc. of the 6th int. Conf. on Information Processing in Sensor Net. (IPSN'07), pp. 519-528, April 2007.

[15] R. Rizk, H. Elhadidy and H. Nassar, "Optimised mobile radio aware routing algorithm for wireless sensor networks," IET Wirel. Sens. Syst., vol 1, no. 4, pp. 206–217, 2011.

[16] M. Chen, T. Kwon, and Y. Choi, "Energy-efficient differentiated directed diffusion (EDDD) in wireless sensor networks," Elsevier Computer Comm., special issue on Dependable WSNs, vol. 29, no. 2, pp. 231-245, 2006.