International Journal of Applied Engineering Research ISSN 0973-4562 Volume 13, Number 8 (2018) pp. 6125-6137

Adaptive Protocol for Management Internet of Things Systems: Analysis, Design and Performance Evaluation

Omar Said
College of Computers and Information Technology, Taif University, Taif, KSA.
College of Science, Menoufia University, Shibin El Kom, Menoufia, Egypt.
Corresponding Author
ORCID: 0000-0002-8868-8000

Alaa Elnashar
College of Computers and Information Technology, Taif University, Taif, KSA.
College of Science, Minia University, Minia, Egypt

Osama Elshakankiry
College of Computers and Information Technology, Taif University, Taif, KSA.
Faculty of Electronic Engineering, Menoufia University, Menouf, Menoufia, Egypt.

Abstract

Nowadays, the Internet of Things (IoT) became able to communicate milliard of heterogeneous things on Earth. So, IoT systems should collect and handle massive amount of data. Hence, field IoT management systems became one of the most important challenged fields due to the special nature of IoT. Simple Network Management Protocol (SNMP) is a famous management protocol. To apply SNMP on the IoT systems, it should be adapted. In this paper, an adapted version of SNMP is introduced. In this version, the special specs of IoT such as rapid scalability, heterogeneity, huge number of traps, and big data are considered. In addition, the proposed version comprises different forms of SNMP by assigning a specific form for each class of things. Furthermore, the Management Information Base (MIB) is reconstructed to comprise additional objects for the IoT systems. Finally, the network simulation package, NS2, is used to measure the performance of the proposed adapted SNMP. The performance metrics, which are used to test its performance, are percentage of number of management messages, IoT-SNMP state transformation percentage, bandwidth consumption, delay, packet loss percentage, throughput, and energy consumption. Moreover, results of IoT- SNMP are compared with the most recent version of traditional SNMP. The results proved that the proposed IoT-SNMP outperforms the traditional one.

Keywords: Internet of Things; SNMP; IoT Simulation; IoT Management; Network Management.

INTRODUCTION

Recently, the Internet of Things (IoT) becomes a hot topic in many fields such as industry, marketing, security, military, etc. Each daily life object can be equipped with Radio frequency identifier (RFID) to communicate such object with others using wired or wireless technologies. Early, industrial equipments were connected using IoT technology. Today, IoT concept is changed to communicate everything on the plant, i.e. active or passive (inanimate) things in addition to animals, peoples, and living organisms. Furthermore, information can be communicated with human beings as in case of e-health applications. Hardware and software are used to implement the IoT devices. Special hardware components such as microcontrollers are used to engage IoT nodes with the physical world to perform complicated tasks. The interaction between IoT devices is accomplished using special operating systems. Also, an important feature of IoT is that the communication between its devices are accomplished using the Internet, based on large cloud-base servers. This concept guides us to use Internet protocols but after some sort of adaptation process to become suitable to work with the IoT systems [1-7].

Network management can be defined as a group of functions, methods, procedures, and administrative tools that manage the network. The network management comprises network administration, network operation, network maintenance, and network provisioning. The network administration is the process of tracking network devices and resources such as routers, servers, etc., i.e. it monitors the network resources performance. In addition, it distributes the software applications that are used by network users. The network operation process is related to smooth network functions such as network address monitoring or fixing simple problems that may be occurred frequently. The network maintenance process involves repairing and upgrading network resources as well as preventing the possible failures by communicating with network administrators. The network provisioning process concerns with network resources configuration to adapt special requirements of network users or devices such as increasing of voice broadband to facilitate more users [8-11].

Simple Network Management Protocol (SNMP) is one of the most famous network management protocols. The basic function of SNMP is to collect information about the network and extract its performance parameters values. In addition, it predicts the problems that may occur in the future. SNMP is an application layer protocol that uses UDP as a transport layer

protocol. It consists of SNMP manager, SNMP agent, and SNMP Management Information Base (MIB). The SNMP manager communicates with SNMP agents, which are located in the network devices, reply the agents' queries, get agents' feedback, set agents' variables, and acknowledge the synchronized events, which may occur in agents. SNMP agent is a software package that is installed on each network device to collect device information. SNMP MIB is a database that is constructed by SNMP agent, stored in the network device and used by SNMP manager [12-18]. In this paper, SNMP is used to manage the IoT systems. The most recent version of SNMP is adapted to meet the IoT requirements i.e. modifications are carried out in SNMP to accomplish its function without causing an extra overhead in control bits or computations. Also, Simplicity of the resultant version is a desired goal of this paper.

This paper is proceeded as follows: In Section 2 the paper problem is introduced. In Section 3, the related works are discussed. In Section 4, the paper contribution is demonstrated. In Section 5, the proposed IoT-SNMP is discussed. In Section 6, simulation and evaluation of the proposed IoT-SNMP are demonstrated. Finally, the conclusion is introduced in Section 7.

PROBLEM FORMULATION

It is clear that IoT contains billions of nodes. Each node produces data in an automated manner. In addition, variety of nodes implies variety of collected data, which requires many complex protocols, algorithms, and techniques to manipulate such collected data. Furthermore, the continuous increase in the number of devices that are connected to the internet makes the size of the collected data also increases dramatically. The rapid increasing of data size badly affects scalability, transmission, and processing. So, the main challenge for this paper is how to manage the IoT systems using SNMP tacking into consideration all of special specs of the IoT environment.

RELATED WORK

The previous trails that utilized SNMP in IoT systems were related to the IoT management. F. Sallabi, et al., [ref] proposed architecture for healthcare management system. This architecture has many advantages such as reliability, effectiveness, well performing, and security. The implementation of the developed architecture is weak and is considered as special purpose architecture [19]. Some related issues were discussed by T. Brooks in [20] such as security and management of IoT. S. Choi, et al., proposed two management schemes, CoAP-PMIP and CoAP-DPMIP. These schemes are based on proxy mobile IPv6. CoAP-PMIP is used to provide mobility support for sensors. CoAP-DPMIP is used to determine the optimized path which should be followed to transmit data in the IoT. The simulation of these proposed schemes are accomplished and their results proved that CoAP-DPMIP scheme has better performance compared to both of CoAP and CoAP-PMIP schemes. The simulation of these schemes didn't reflect the real environment of IoT. So, the results are considered not accurate [21]. B. Gateau, et al., proposed a model for automatic treatment which is called e-comfort management model. There are two layers which are used to discuss the proposed model, low-level layer and upper layer of software agents. This model formulates negotiation to provide human users with the required treatment. This model is considered as a special purpose one and didn't consider the scientific meaning of network management [22]. J. Chen, et al., proposed a solution for the problem of resource and power allocation for energy consumption in the LTE-A relay networks. This solution minimized the consumption of energy and guarantee quality of service (QoS). This solution is only energy consumption minimization oriented (not network management oriented) [23]. J. Kim proposed a network management protocol to manage WSN. This protocol is considered as a special purpose protocol because it is designed for only WSN which is considered as a core of IoT but not IoT itself. In addition, there is no implementation or simulation for this proposed protocol [24]. D. Gao, et al., used service-oriented architecture (SOA) and geographic information systems (GIS) to design monitoring management platform for wide-area electric vehicle charging-swap networks. This platform architecture is considered as a special purpose [25] platform. Z. Xinhua, et al., proposed a network management system which is designed for heterogeneous networks. This system has many advantages such as low energy consumption, flexibility, automatic configuration. This proposed system is used more frequently for WSN than for IoT [26]. H. Hui-Ping, et al., proposed a strategy to use SNMP for management WSN in IoT. This work is considered the most closed related work to our paper idea. The proposed strategy has many weak points such as no implementation or simulation exists, used only for WSN, and the adaptation process of SNMP is not clear [27].

PAPER CONTRIBUTION

In this paper an adapted version of SNMP, which is called IoT-SNMP, is proposed. The proposed IoT-SNMP infrastructure is based on multilevel management tiers. These tiers comprises the following objects; general manager, sub managers, general agents, and subagents. These tiers are communicated with each other to complete the management mission of the IoT environment nodes. Furthermore, the IoT-SNMP has three statuses, which could be extended, to be consistent with the IoT systems (Fig. 1). A part of Fig. 1 is taken from [28]. The SNMP messages are adopted and their sending and receiving processes are controlled. In addition, MIB is extended to comprise new things that are related to the IoT environment. Furthermore, the NS2 simulation package is used to construct an IoT environment to test performance of the proposed IoT-SNMP. Finally, simulation results of the IoT-SNMP are compared to that of the traditional SNMP.

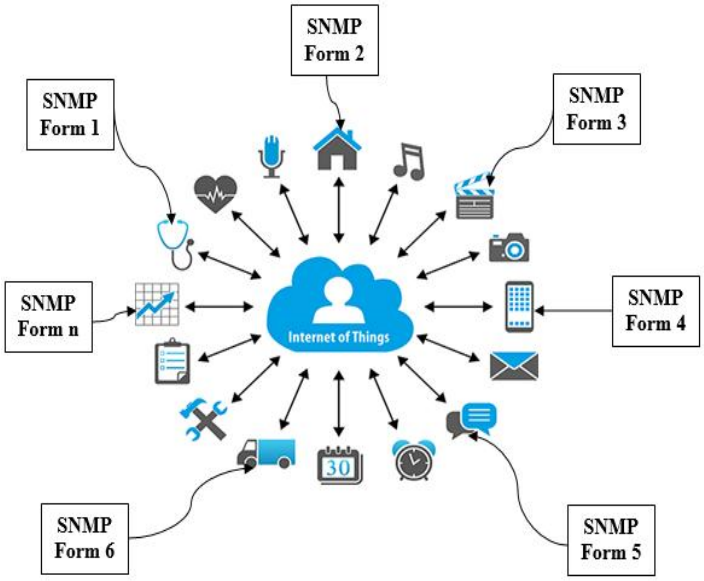

Figure 1: Idea of the adapted version of SNMP.

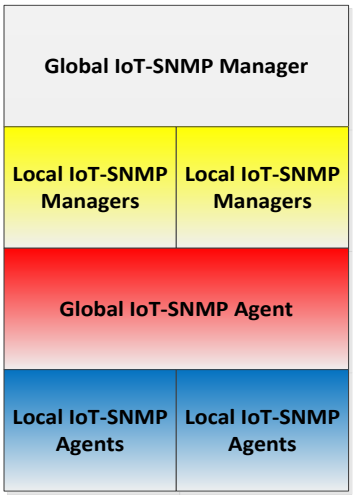

Figure 2: IoT-SNMP hierarchy.

IoT-SNMP

In this section, the components of IoT-SNMP are briefly discussed. Then, the way the IoT-SNMP works is described. Finally, the reconstruction of MIB is introduced.

Components of IoT-SNMP

IoT-SNMP consists of four main components that are distributed over the IoT environment and are communicating with each other in harmony to accomplish the management functions, see Fig. 2. These components are stated as follows:

1. Global IoT-SNMP manager: This component is used to manage the entire IoT system. It has most of the management functions and communicates with other management components of the IoT-SNMP. The global IoT-SNMP manager is considered as a central system that is transformed to a distributed system depending on the number of active devices in the IoT environment. The global IoT-SNMP manager should have powerful specs to be able to achieve the management functions efficiently.
2. Local IoT-SNMP managers: These local managers are distributed over the IoT environment. The IoT environment is divided into groups or domains. Each local manager is responsible for one or more IoT domains. It is not required to use powerful specs devices for small managers. In case of overlapped domains, the global IoT-SNMP will select the local manager depending on predetermined parameters such as specs of selected local managers, specs of the domain, type of devices, etc.
3. Global IoT-SNMP agent. This agent is responsible for monitoring the global IoT things. It communicates with the local IoT-SNMP agents in the IoT environment.
4. Local IoT-SNMP agent: these local agents are distributed over the IoT environment to monitor things (active or passive) in the IoT environment.

How IoT-SNMP works

The IoT environment comprises different networks with a large number of heterogeneous nodes that are communicated by the Internet. Therefore, SNMP is selected to manage this type of environment. SNMP is a simple protocol constructed to manage Internet nodes in an easy manner. IoT-SNMP keeps this simplicity feature with additive functions to manage the IoT environment. Hence, IoT-SNMP comprises managers, agents, and MIB in addition to messages to communicate the components with each other. Furthermore, IoT-SNMP protocol distributes many agents over each main network or sub-network in the IoT environment. These agents are organized in tree architecture such that each agent has a parent agent. Also, IoT-SNMP managers are organized in tree architecture such that each manager has a parent manager. Firstly, IoT-SNMP local agent connects to MIB to access the values which are required by its local manager. Then, the IoT-SNMP local manager sends this data to the upper layer (parent) IoT-SNMP manager, which sends the data to the upper layer until the targeted data researched to the global IoT-SNMP manager. The main challenge is to decrease the number of transmitted management data. To achieve this target, the number of IoT-SNMP messages should be decreased. This reduction in the number of messages should be related to the IoT system statuses. In other words, in case of healthy IoT network, the number of management messages may be increased. But, in case of bad status of IoT network, the number of messages should be decreased. The global IoT-SNMP manager monitors status of the entire IoT system. So, it can be decided by system administrator to decrease or increase the IoT-SNMP management messages. It is well known that the SNMP messages are GET, GET NEXT, Trap (NOTIFY), SET, Get BULK [14]. The traditional usage of the first SNMP message, which called GET, is to access data from a specific agent after a request from the SNMP manager. This traditional meaning should be changed to be more restricted such that GET message should be fired not only upon SNMP manager request but also upon the IoT system status. Hence, GET message should be constructed periodically depending on the time schedule which is determined by the IoT-SNMP. This

time schedule is determined depends on three parameters; number of users, number of transmitted bits (size of data) and status level of the IoT system. If the number of IoT users more than a predetermined threshold, the transmitted data will be massive, which leads to IoT system starvation, the GET message transmission will be postponed. On the other hand, in case of low number of users, which means low transmitted data that leads to healthy network, so the period between consecutive transmissions of GET messages will became short. For GET NEXT message, it should be merged into GET message. In other words, GET message should send all of required data to SNMP managers. The GET message should contains new values of predetermined values for the MIB objects provided that the change in the values should be smooth. For the GET Next message, it can be used in normal status of the IoT system. In the GET message (PDU), the request ID filed should be deleted and the period filed should be added, see Fig. 3 and Fig. 4. This will provide a flexibility and new SNMP IoT versions can be created. To achieve this target, a new message should be sent from global IoT-SNMP manager to inform the IoT-SNMP agents about the new schedule of their sending messages. This new message will be sent using the same technique in broadcast message and will be called "Scheduling", see Fig. 5. Also, the meaning of SET message should be changed to become more suitable with the IoT environment. It's well known that the traditional meaning of SET message is change or add a new value for MIB object by SNMP manager. In addition, the nature of IoT environment objects is that they periodically changed. So, SET message may be used repeatedly which will consume extra bandwidth. Hence, SET message should be divided into two main messages. The first one is used to SET urgent values and called USET. The second one, which called HSET, is used to change normal IoT objects' values but in case of IoT system starvation, the process of this type of SET message will be hold. As regards the TRAP message, it should be fired in case of sudden event occurrence. The TRAP message should not have any changes because it is so danger if this message became restricted at any time of IoT system working. But, there is a new message which should be added in the IoT-SNMP. This message is called prediction message that can be sent from IoT-SNMP manager to IoT-SNMP agent and via versa. This message also may be sent from IoT-SNMP manager to other IoT-SNMP managers and from IoT-SNMP agents to other IoT-SNMP agents. The message target is to predict the new values of MIB objects in addition to the events which may be occurred suddenly. To construct this message, the prediction system, which is found at [29], should be installed in IoT-SNMP to analysis the available IoT data periodically and predicts the future events. Also, the prediction system should analysis solutions of the sudden problems that are occurred previously. From the old solutions, new suggested ones may be introduced by local IoT-SNMP managers which decrease the time consumption. For the IoT-SNMP prediction message contents, see Fig. 6.

Upon the discussion of IoT environment, it is found that these types of environments comprise many different networks. So, the nature of IoT-SNMP agent will be different with respect to the network type. For example, the agent in the mobile ad hoc network should monitor the network and the transmitted data even the network management system not active. This is because the MANET users join and leave the network frequently. For WSN, the agent should send the target data without delay because these types of networks are based on environmental monitoring. In addition, the nodes in WSN are mostly sensors which have not powerful processing unites, so alight version of SNMP agent should be designed such that only it can send the required data to its local IoT-SNMP manager. For RFID network, it should be managed by agents that are setup on active devices such as servers. For satellite network, the agent should have an ability to send and receive the data to/from long distances. In IoT-SNMP, there is a uniform design of IoT-SNMP agent, which is adapted to work according to privileges and restrictions. As stated above that the agent component should have multiple forms to be suitable for different networks in the IoT environment. To solve this problem, each feature in the global IoT-SNMP agent has two states, active and inactive. The active state means that the feature is ready to use by the agent and inactive state means that the feature cannot be run for a predefined time interval. This idea creates many forms of the IoT-SNMP agents. These forms can be adapted to work in harmony with the IoT network depending on its natures. Furthermore, the global IoT-SNMP agents can be constructed by making all of its features in active state. So, it can be used to manage the local IoT-SNMP agents. For general view of the IoT-SNMP, see Fig. 7.

The relation between global IoT-SNMP managers and local IoT-SNMP managers should be determined. Each local IoT-SNMP manager has full authority to manage its cluster (network part). It can send and receives management messages to/from agents what are. In addition, it can summarize the collected data from agents and sends it to the global IoT-SNMP manager. Furthermore, one or more IoT-SNMP can replace the global IoT-SNMP manager in case of its failure. Also, selection of local IoT-SNMP managers which will be used to replace the global one is based on its technical specs and the distance from the global IoT-SNMP manager in addition to the number of IoT-SNMP agents per IoT-SNMP manager area. The number of local IoT-SNMP agents in each cluster is determined depending on the number of users (size of network). For the global IoT-SNMP agent, it should connect to the local IoT-SNMP agents to measure the efficiency of each agent regards monitoring its device. Furthermore, it can connect directly to the global IoT-SNMP manager by summarizing the data of local IoT-SNMP agents.

In traditional SNMP, there is a communication between managers- especially when there are many levels- can be accomplished using messages. But the communication between agents should be achieved using multi-agent system feature [23]. Using multi-agent system will be useful in the IoT environment because it reduces the latency of communication in addition to prevent bottlenecks in the SNMP system. Furthermore, it increases the scalability of the system by adding more agents easily. Moreover, agents can achieve other missions while waiting for the SNMP manager even local or global. Also, it can recover failure agents easily.

Version Number	Authentication	PDU Type	Request ID	Interval	Error Status	Error Index	Variable Bindings

Version Number	Authentication	PDU Type	Replay	Interval	Error Status	Error Index	Variable Bindings

Figure 3: IoT-SNMP request/reply messages

Version Number	Authentication	PDU Type	Enterprise	Agent Addr.	Generic Type	Specific Trap Type	Time Stamp	Variable Bindings

Figure 4: IoT-SNMP trap message

Version Number	Authentication	PDU Type	IoT-SNMP Local Managers' Add	IoT-SNMP Local Agents' Add	New Interval Schedule

Version Number	Authentication	PDU Type	IoT-SNMP Local Managers' Add	IoT-SNMP Local Agents' Add	New Interval Schedule

Figure 5: IoT-SNMP message that determine the time scheduling of management messages transmission

Version Number	Authentication	PDU Type	IoT-SNMP Local Managers' Add	Problem	Soultion

Figure 6: IoT- SNMP prediction message

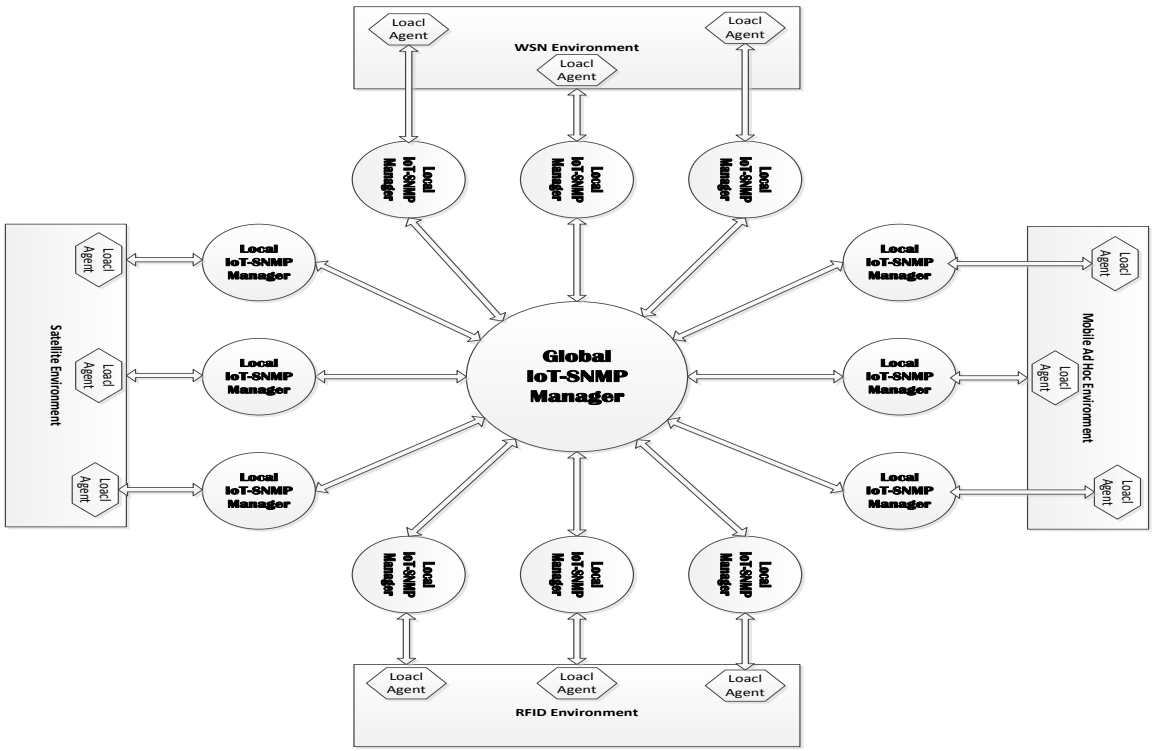

Figure 7: General view of the IoT-SNMP

IoT-SNMP MIB

One of the most important components in SNMP is MIB. MIB comprises objects that their values are gathered by SNMP agents and accessed by SNMP managers. The traditional SNMP environment may be constructed from WSN, Mobile Ad hoc or RFID networks. But these networks may be found individually. The communication of these types of networks and others by the Internet will construct the IoT environment. Hence, IoT-SNMP comprises objects of all of these networks. Furthermore, MIB for IoT-SNMP is extended to accept more objects in the future. To clarify the MIB of the IoT-SNMP, each network MIB is simply introduced.

For WSN MIB, there are many classes that should be managed. These classes include energy, topology, transceptor, hierarchy, processor, sensor, and administrative. Firstly, the energy objects can be abstracted in many fields such as Energy Source, Residual Energy, Operation Voltage, Manufacturer, and Battery Type. Secondly, the topology class has many fields such as mobile, motion, velocity, location type, coordinates, direction, and neighbors. Thirdly, the Hierarchy class has an identifier of each group, members of the group, active members, and level. Fourthly, the processor class has manufacturer, RAM, ROM, frequency, and speed. Fifthly, the transceptor class has Operational State, Range, Manufacturer, Transmission Consumption, and type. Sixthly, the sensor class has Manufacturer, Data Buffer, Range, Last Calibration, Sensing Interval, Consumption, and Operational State. Seventhly, the administrative class contains Data Messages Sent, Management Messages Sent, Is Common, Administrative State, and Is Access Point.

For Mobile Ad-Hoc MIB, the objects are organized into four classes. These classes are capabilities, configuration, state, and performance. The capabilities class comprises objects that are used by network devices for advertising to clarify their locations such as RSSAs. The configuration class comprises objects that are related to multicast performance and option configuration for the network devices. The state class comprises objects that contain information about network devices with their neighbors. The performance class contains objects which are related to the performance of each device in the mobile ad-hoc network.

For RFID network MIB, there are many classes such as ReaderDevice (getDescription, etDescription, and getLocationDescription), NotificationChannel (getLastNotificationAttempt, setAdminStatus, etOperStatusAlarmControl, and getLastSuccessfulNotification), AlarmChannel (create, getName, getAddress, and setAddress), and AntennaReadPoint (getIdentificationCount, getMemReadCount, getFailedMemReadCount, getWriteCount, and getKillCount). For the satellite network MIB, it comprises many objects such as satelliteInfo, satelliteNumber, masterSettings, satelliteUpNotification, colubrisSatelliteManagementMIBCompliances, satelliteIpAddress, satelliteDetectionPort, satelliteDeviceMacAddress, and satelliteGroupName.

The IoT-MIB is a collection of MIBs of individual networks that the IoT environment comprised, see Fig. 8. The objects at every network are classified into three classes. The first class comprises the objects that should be monitored and their values should be accessed periodically by IoT-SNMP managers (important objects). The second class comprises the objects which cause traps frequently (critical objects). The third class comprises the traditional objects ((Normal Objects)). This IoT-MIB organization makes IoT-SNMP version flexible to the dynamic changes that may be occurred in the IoT environment. Accordingly, in case of network starvation, the number of required objects is also decreased. This leads to minimizing the size of transmitted data. The organization of IoT-MIB objects into three categories is achieved using queuing theory [30] by creating three queues, one for each category. The first queue is assigned to the first category and handling process of objects that should be determined. The second and third queues are assigned to second and third categories respectively. The administrator of each local region in the IoT environment is responsible for determining the number of objects in each category, the status of IoT system part, and the mechanism that should be executed in each queue.

Neglecting of one or more MIB object should be achieved by each local IoT-SNMP manager. The global IoT-SNMP manager can reactive any neglected object depending on its importance within its region or over a larger region. For object classification style, see Fig. 9

Upon the above discussion of the IoT-SNMP and MIB, the states of IoT-SNMP should be clarified. The proposed IoT-SNMP comprises three statuses and can be extended to contain more other statuses. The first status is full SNMP version which will be used in case of small IoT system. This version comprises the same strategies of traditional SNMP. The second version (status) is used when the IoT environment has large size and there are many users transactions join/leave to/from the environment within a short time interval. This version of IoT-SNMP is divide the IoT environment into many clusters such that one or more local IoT-SNMP managers in addition to one or more local IoT-SNMP agents are reserved for each cluster depends on its size. Furthermore, this version classifies the MIP objects into 3 categories. The third version of IoT-SNMP is used when the IoT environment has more bottlenecks. So, the number of IoT-SNMP messages should be decreased because the system resources will be limited. Hence the number of accessed objects should be decreased. In this case the prioritization idea is applied on the MIB objects. Accordingly, after the object classification, the important objects should be reclassified to most important objects and less important ones. Also, the number of messages which come from many local IoT-SNMP agents and received by local IoT-SNMP managers should be abstracted provided that the abstract message should be sent to the upper layer of IoT-SNMP hierarchal through a bottleneck. The transformation from one IoT-SNMP status to another is achieved dynamically, see Fig. 10.

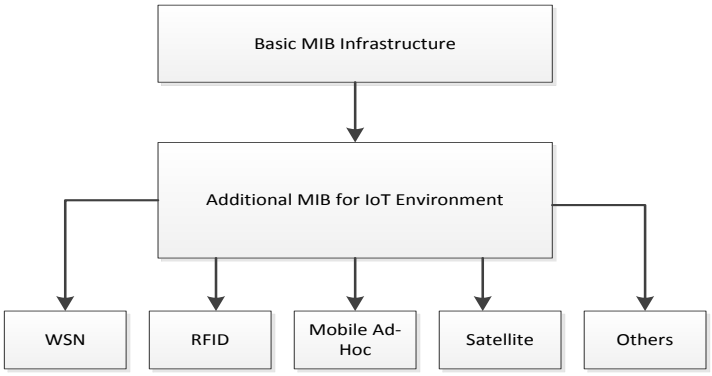

Figure 8: IoT-SNMP MIB

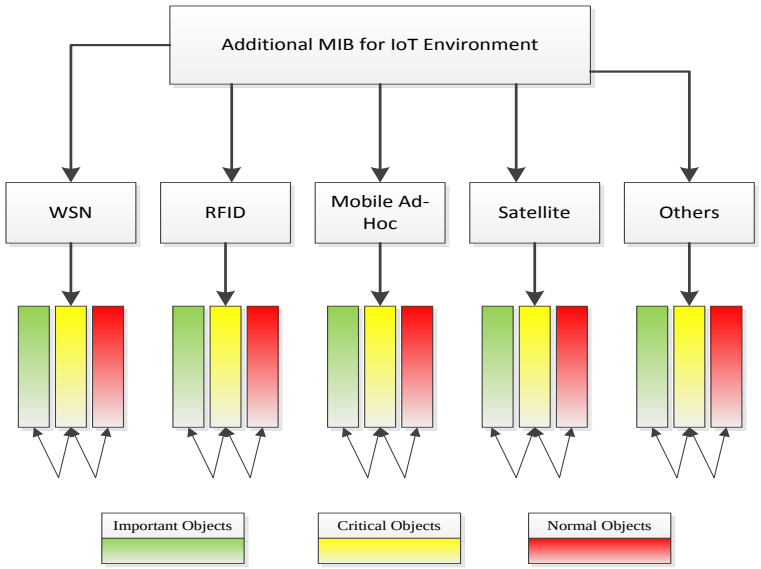

Figure 9: The MIB objects' classification in the IoT-SNMP

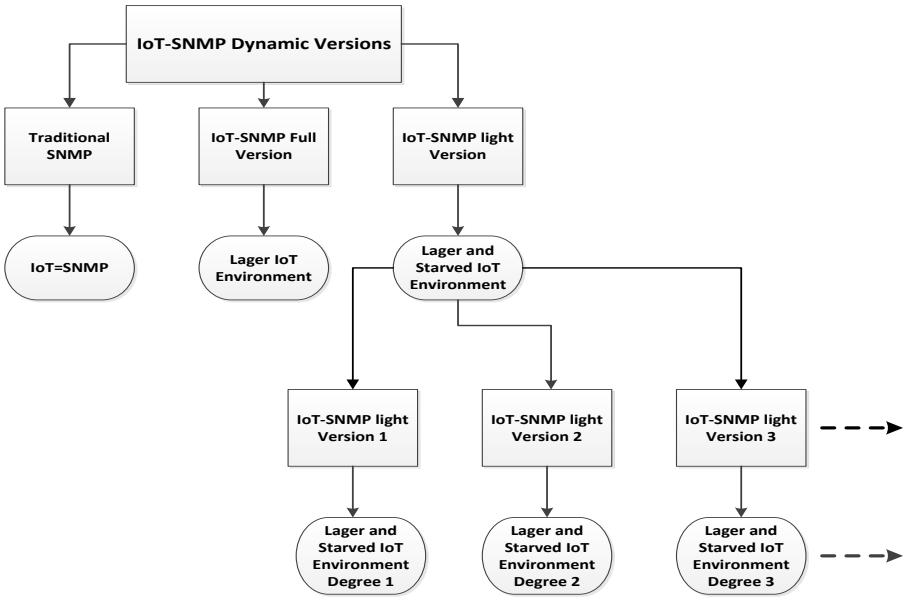

Figure 10: General view of the IoT-SNMP versions (statuses)

SIMULATION AND EVALUATION

This section measures and compares the performance of the proposed IoT-SNMP to the performance of the traditional SNMP. Firstly, the infrastructure of the simulation environment of IoT system that will be used to test the performance of the IoT-SNMP is introduced. Secondly, the simulation results are discussed.

Simulation setup

Measurement of IoT-SNMP performance is an important issue to determine how the IoT-SNMP enhances the traditional SNMP when it is applied in the IoT environment. NS2 is used to construct the simulation environment [31]. As stated before, IoT comprises a massive number of heterogeneous nodes that have different specs. To represent this type of environment, different networks should be involved in the simulation infrastructure. So, WSN, MANET, and RFID networks are represented in the simulated IoT environment. Recently, the Internet is considered as a famous transmission medium where it has been widely spread in our communities. Accordingly, the Internet acts as a communication medium between the IoT different networks. In addition, the IoT environment may comprise active and passive things. Active things are represented easily because most of networks such as WSN or MANET comprise active nodes. In addition, passive things (i.e. things without processing unit) are also represented. To achieve the representation of passive things, RFID network is used. Furthermore, each network comprises a large number of nodes to reflect the nature of IoT in accurate manner. The factors that affect the IoT are stated as follows; number of nodes, channel specs, network size, protocols, overlapped areas, number of passive things, etc. Hence, changing of IoT in the simulated environment is represented by controlling in the channel specs in addition to the number of nodes that are joined/left to/from the IoT system. Autonomies nature of IoT is represented by the determination of events and actions. Events and actions are stored in two different files. The two files are mapped such that when a specific event, which is listed in the first file, occurs the suitable action, which is listed in the second file, is executed automatically.

To introduce the simulation environment, each network simulation is discussed. WSN simulation is represented by a number of sensors that communicate with each other. This number is increased or decreased automatically within a simulation time. The sensors are organized into clusters and distributed randomly with dynamic shape exchange. At each sensors' cluster, there is one or more sink node. The number of sink nodes is determined depending on the number of sensors at each cluster. Each sensor has a location that is determined by its coordinates. For the simulation parameters of WSN see table 1.

RFID simulation is represented by tags, readers, sites, and applications as they are the main components of RFID networks and determine the RFID network scalability. RFID things are randomly distributed at a square area. The simulated RFID network comprises active and passive things. The communication between active things is easy to achieve. But, the communication of passive things is accomplished using the information that is distributed through the RFID network (by TCP/IP protocol suite). The distribution of management tasks is the strategy which is implemented in the simulation of RFID network using network controllers. For the simulation parameters of RFID, see table 2.

The network area in MANET simulation is represented as a square area with two types of nodes, i.e., clients and server. The client node is considered as source of network data in addition to transitional node which is used to transmit this data form one node to another. The server node is used to response to the client node requests. Creation of requests and replies is achieved randomly. The users are distributed in MANET using uniform distribution. The client requests may be accepted or rejected. The direction of each user is determined randomly. The distance between users depends on the ability of his coverage. The simulation parameters of MANET are found in table 3.

To complete the simulation scenario, the Internet is comprised in the simulation infrastructure. It is simulated by a number of nodes that communicate with each other using TCP/IP. To reflect the nature of the Internet, each node has its protocols, specs and statistics. The routing mechanism that is used in the simulation is found at [32]. The traditional SNMP simulation is found at [33]. For general view of the proposed simulation infrastructure, see Fig. 11.

Table 1: WSN simulation parameters

WSN Simulation Parameters			
#	**Parameter**	**Value**	**Unite**
1	RX	94	dB m
2	Battery	1250 and 1.5	mAH and V
3	Transmission mode (current drain)	11	mA
4	Receiving mode (current drain)	19.7	mA
5	Size of coverage area	1000 × 1000	m
6	Power of RF	10	dB m
7	TX	250	kb/s
8	Number of sensors	1100	Sensor
9	FRQ	2400	MHz

Table 2: RFID simulation parameters

RFID Simulation Parameters			
#	**Parameter**	**Value**	**Unite**
1	SNR Range	7–10	dB
2	Data Rate	2	Mbps
3	RFID transmission power	45	dB m)
4	Sensitivity of RX	91	dB m)
5	Range of reading	1.62	m
6	Number of nodes	1500	nodes
7	Range of interference	7.1	m
8	Fading	Null	Null
9	Interference of inter-channel	Null	Null
10	FRQ of control channel	930	MHz
11	Range of sensing	5.4	m
12	Threshold of RX	81	dB m
13	FRQ of Data channel	915	MHz

Table 3: MANET simulation parameters

MANET Simulation Parameters			
#	**Parameter**	**Value**	**Unite**
1	Size of coverage area	500 x 500	m
2	Speed of node Range	30-60	km/h
3	Distance of Transmission Range	30-210	m
4	Time to live Random range	4-7	ms
5	Number of transmitted requests	2000	request
6	The interval between transmitted requests	500	ms
7	Number of nodes	100	node
8	Availability of link Range	0-1	tu
9	Size of packet	1	mb

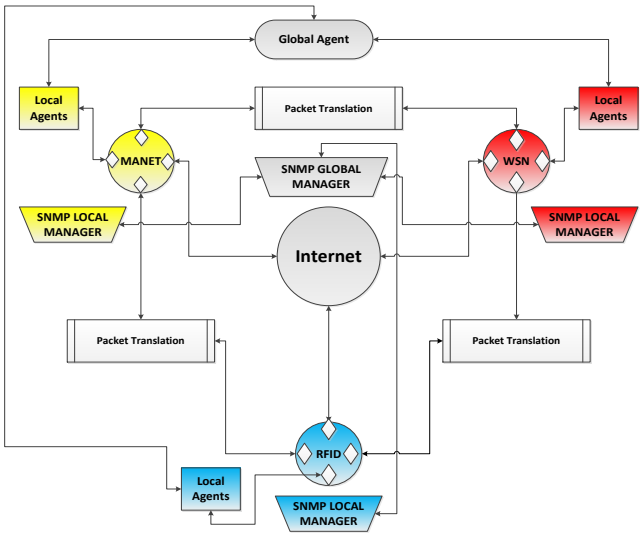

Figure 11: Simple view of the simulation scenario

Simulation results

The performance metrics, which are used to measure the performance of IoT-SNMP, are the number of transmitted IoT-SNMP messages, state transformation, bandwidth consumption, end-to-end delay, packet loss percentage, throughput, and energy consumption. In this subsection, the results of performance parameters are discussed and explained to clarify the effect of the IoT-SNMP in the management of IoT environment and prove that it outperforms the traditional SNMP.

The number of management messages in case of IoT-SNMP and traditional SNMP is considered as the most important parameter in this performance measurement process. This is because if the number of management messages is high, QoS, which are available in the IoT system, will be consumed. Then, the main messages will be starved in their transmission trip which affects the entire IoT system performance. Fig. 12 and Fig. 13 show the number of transmitted messages in the IoT-SNMP and the traditional SNMP respectively. The X axis represents the simulation time in minutes and the Y axis represents the number of messages in both of the IoT-SNMP and the traditional SNMP. It is notable that the number of management messages in the IoT system when using IoT-SNMP is less than that of the traditional SNMP. In addition, for the traditional SNMP, at multiple of simulation time points, the number of management messages is increased as the number of users in the IoT system is increased. So, there is no control over the number of management messages in the traditional SNMP. On the other hand, for the IoT-SNMP, when the number of management messages increases, the IoT-SNMP changes its status to use lighter version that decreases the number of management messages. Hence, it provides more chance for other IoT main data to be transmitted which increase the efficiency of the IoT system.

The second performance metric is the IoT-SNMP status transformation. This metric clarify how the IoT-SNMP changes its status depending on the state of IoT environment. Fig. 14 shows the results of this performance metric. The X

axis represents the IoT-SNMP statuses and the Y axis represents the percentage that the status is executed within the simulation time. It is notable that status number 2 is executed most of simulation time. This is because the second IoT-SNMP status represents the normal state of IoT. The first IoT-SNMP status represents the light version of IoT environment which its results come after the second IoT-SNMP status. But, the third status represents the starved version of IoT environment, so it takes fewer periods than the other statuses. Also, the results show that there is a transformation between the three IoT-SNMP versions which proves that the proposed IoT-SNMP has a flexibility to work in dynamic environments such as IoT.

The bandwidth consumption is measured to clarify the effect of the proposed IoT-SNMP on bandwidth usage. This performance metric is calculated by the number of management packets which is transmitted between global IoT-SNMP manager and local IoT-SNMP managers, between global IoT-SNMP manager and local IoT-SNMP agents, between local IoT-SNMP managers and local IoT-SNMP agents, and between global IoT-SNMP agent and local IoT-SNMP agents. The bandwidth consumption for the traditional SNMP is calculated by the number of management packets transmitted between SNMP manager and SNMP agents. Fig. 15 shows the results of bandwidth consumption. The X axis represents the simulation time in minutes and the Y axis represents the bandwidth consumption in bits. It is notable that the bandwidth consumption for the IoT-SNMP is less than that of the traditional SNMP. This is due to the reduction of the number of management messages for the IoT-SNMP. This is explained by changing the status in IoT-SNMP depending on the status of IoT system. But, for the traditional SNMP, the number of management messages increases when the number of things increases regardless the status of IoT and the available QoS in the IoT networks. The hesitations in the two bandwidth consumption blocks comes from the fast dynamic changing in the IoT status by join/ leave users/things to/from the IoT systems.

The end-to-end delay is one of the most important performance metrics in computer networks field. This is because it can be used to judge on the entire efficiency of a network. The end-to-end delay in this simulation is calculated by the summation of transmission, queuing, propagation, and processing delays. Fig. 16 shows the results of end-to-end delay. The X axis represents the simulation time in minutes and the Y axis represents the end-to-end delay values in millisecond (ms). The end-to-end delay is taken as an average over different networks that are installed in the simulation environment. The end-to-end delay for the IoT-SNMP is less than that of the traditional SNMP version. This is because the IoT-SNMP has an ability to transform itself to a suitable status that decreases the number of management messages which reduces the delay. On the other hand, the traditional SNMP isn't flexible with the IoT environment that changes its state periodically. In addition, hesitations in the two end-to-end delay plots are resulted from rapid join/leave of users to/from the IoT environment that leads to the change in the number of management messages in addition to data messages.

The packet loss performance metric is measured to determine the effect of the management message transmission on the main message transmission. The increase in the number of management messages means the increase of the total number of messages that is transmitted through the IoT channels. This increase may lead to congestion that causes high percentages of packet loss even for the management messages or the main messages. So, measuring the packet loss performance metric is very important. Fig. 17 shows the packet loss percentage results. The X axis represents the simulation time in minutes/10 and the Y axis represents the packet loss percentage. The results proved that the packet loss percentage for the IoT-SNMP is less than that of the traditional SNMP. This happens because of the channel congestion that is occurred in the traditional SNMP due to the high number of management messages during the simulation time. At simulation time point 5, the packet loss percentage for IoT-SNMP is larger than that for the traditional SNMP. This is explained by the sudden change of IoT system status and the IoT-SNMP takes a short time to change its status to be adapted with the new status of IoT.

The throughput performance metric is measured to prove that the management messages don't affect the main messages that are transmitted among things in the IoT system. The throughput is calculated by the number of kilobytes (kbs) which are transmitted and correctly reached to its destinations. Because the IoT environment contains different networks, the throughput is measured for each network individually and the average is taken over the simulation time. Fig. 18 shows the simulation results of the throughput performance metric. The X axis represents the simulation time in minutes and the Y axis represents the average of throughput values. It is notable that the throughput for IoT-SNMP is larger than the throughput of the traditional SNMP. This is explained by the large packet loss ratio which is occurred when the traditional SNMP applied in the IoT environment. The large number of main messages and management messages led to starvation in IoT system that causes a high percentage of packet loss. On the other hand, the IoT-SNMP decreases the number of management messages when the number of main messages increased. In addition, the sudden decrease in throughput values of the traditional SNMP is explained by high delay and packet loss ratio which result from a sever congestion in the IoT system.

It is well known that the energy is a main Skelton in most of IoT networks. Also, in order to increase the life of these types of networks, the energy consumption should be decreased. Hence, the energy consumption performance metric is measured to make sure that the energy based nodes in the IoT networks such as sensors, mobiles, taps, etc. work in normal manner for longer periods. Fig. 19 shows the average of energy consumption for the IoT-SNMP and the traditional SNMP. The X axis represents the simulation time in minutes and the Y axis represents the energy consumption in Joules. The energy consumption of the IoT-SNMP is less than the energy consumption of the traditional SNMP. This is explained by the number of management messages in case of the IoT-SNMP is less than the number of management messages of the traditional SNMP. In addition for the two

plots in Fig. 19, the energy consumption values increase over the simulation time. This is because the number of the main messages increases due to the increase of the number of users of IoT environment.

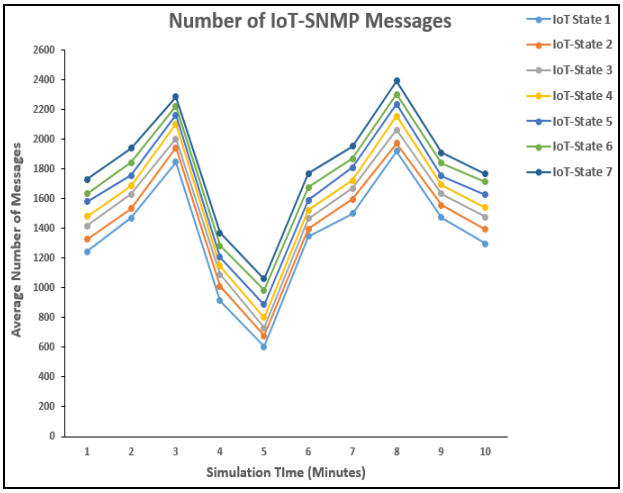

Figure 12: Number of the IoT-SNMP management messages.

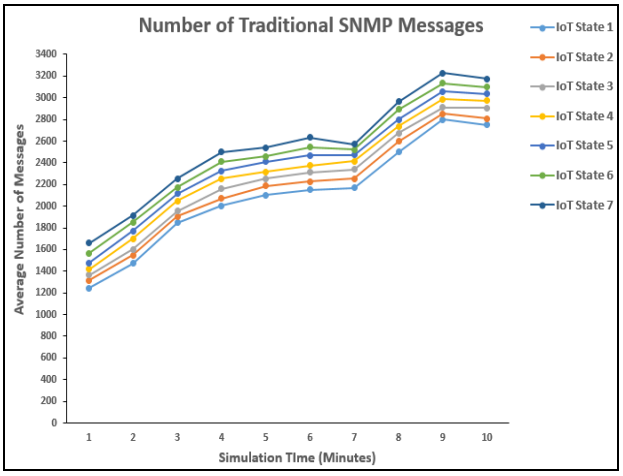

Figure 13: Number of the traditional SNMP management messages.

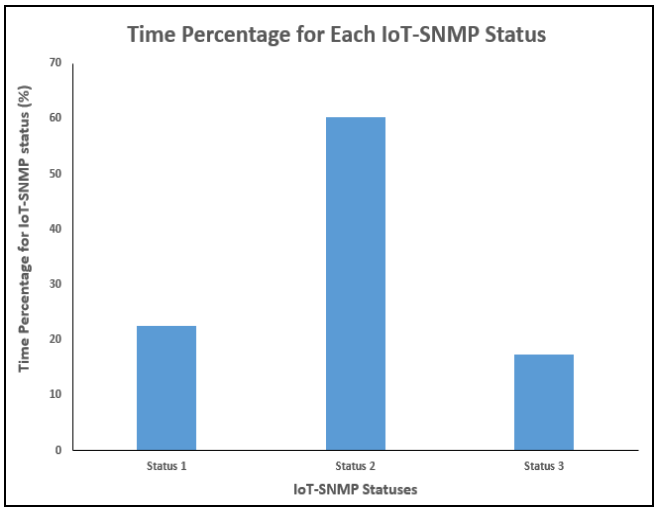

Figure 14: IoT-SNMP statuses' transformation.

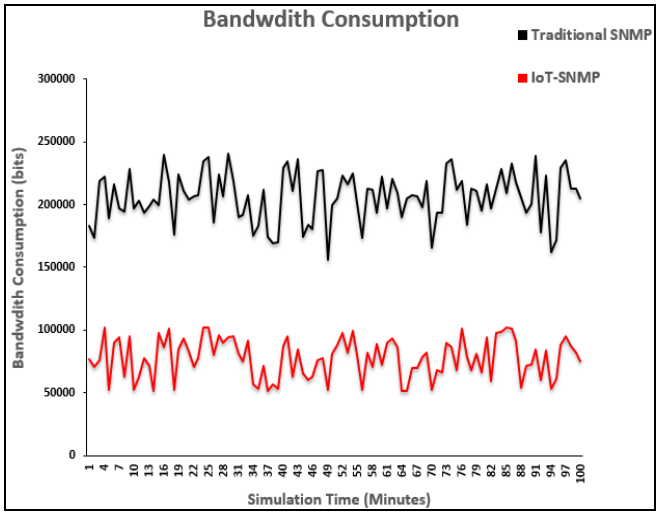

Figure 15: The bandwidth consumption

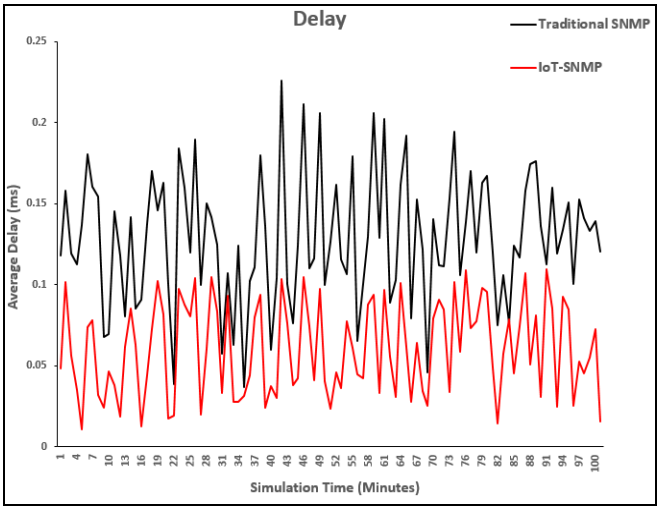

Figure 16: The end-to-end delay of the IoT-SNMP and the traditional SNMP

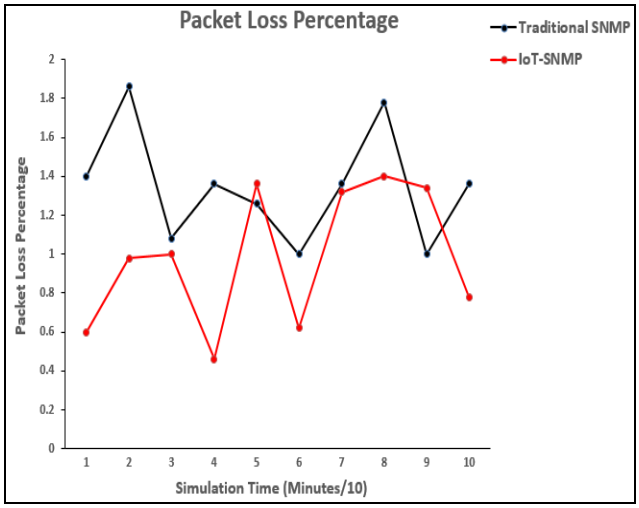

Figure 17: The packet loss

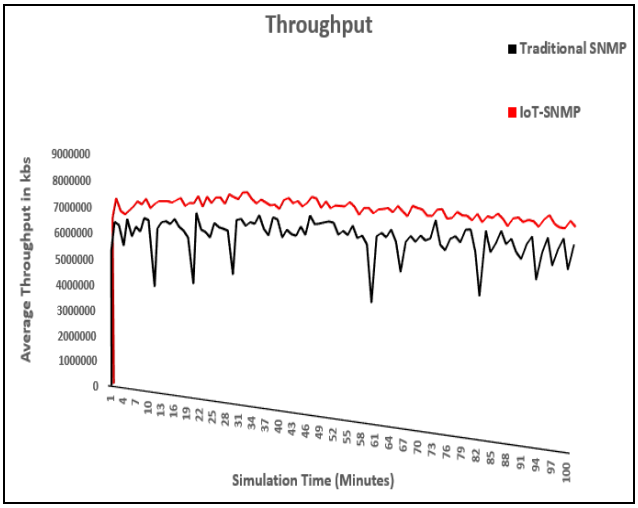

Figure 18: The throughput of the IoT-SNMP and the traditional SNMP

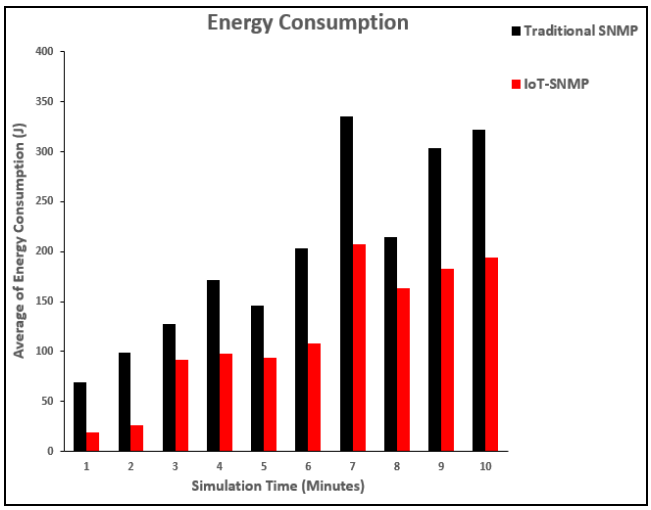

Figure 19: Energy consumption of the IoT-SNMP and the traditional SNMP

CONCLUSION

In this paper, SNMP is adapted to work in the IoT environments. This adapted version is called IoT-SNMP. IoT-SNMP considers the special specs of IoT environments such as scalability, heterogeneity, huge number of traps, and big data. Furthermore, it comprises three different statuses to be consistent with the nature of IoT. Moreover, MIB is reconstructed to comprise additional IoT objects. In addition, IoT-SNMP recognizes the IoT MIB by classifying its objects according to their importance. To measure the performance of IoT-SNMP, the NS2 simulation package is used to construct a simulation environment. The performance metrics are the number of management messages, IoT-SNMP status transformation, end-to-end delay, throughput, and energy consumption. The simulation results of the IoT-SNMP are compared to that of the traditional SNMP. The simulation results proved that the proposed IoT-SNMP outperformed the traditional SNMP as follows, the number of management messages is decreased by 31.507% ↓, the IoT-status 1 is executed 22.45% of simulation time, the IoT-SNMP status 2 is executed 60.24% of simulation time, and the IoT-SNMP status 3 is executed 17.31% of simulation time, the bandwidth consumption is decreased by 39.391% ↓, the end-to-end delay is decreased by 53.027% ↓, the packet loss percentage is decreased by 26.746 ↓, the throughput is increased by 12.795% ↑, and the average energy consumption is decreased by 40.536% ↓. Finally, upon the simulation results, the IoT-SNMP is recommended to be used in the IoT environments.

ACKNOWLEDGEMENT

We are very grateful to Taif University for supporting this research.

REFERENCES

[1] P. Amitangshu, K. Krishna, IoT-Based Sensing and Communications Infrastructure for the Fresh Food Supply Chain, IEEE Computer, Volume: 51, Issue: 2, pp. 76 – 80, February 2018.

[2] T. Qu, et al., Internet-of-Things-based just-in-time milk-run logistics routing system, 2015 IEEE 12th International Conference on Networking, Sensing and Control (ICNSC), Taipei, Taiwan, pp. 258 – 263, 9-11 April 2015.

[3] A. Alvi, W. Shah, Mahmood, Energy efficient green routing protocol for Internet of Multimedia Things, 2015 IEEE Tenth International Conference on Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP), Singapore, pp. 1 – 6, 7-9 April 2015

[4] A. Musaddiq, et al., A Survey on Resource Management in IoT Operating Systems, IEEE access, Volume: PP, Issue: 99, 2018.

[5] Z. Zhou, et al., E-CARP: An Energy Efficient Routing Protocol for UWSNs in the Internet of Underwater Things, IEEE Sensors Journal, Vol. PP, Issue: 99, 2015.

[6] X. Hua-Mei, Y. Kun, Routing Protocols Analysis for Internet of Things, 2nd International Conference on Information Science and Control Engineering (ICISCE), Shanghai, China, pp. 447 – 450, 24-26 April 2015.

[7] F. Turjman, Information-centric framework for the Internet of Things (IoT): Traffic modeling & optimization Future Generation Computer Systems, Volume 80, pp. 63-75, March 2018

[8] O. Said, M. Masud, "Towards Internet of Things: Survey and Future Vision," International Journal of Computer Networks (IJCN), Volume 5, Issue 1, pp. 1-17, 2013.

[9] P. Diogo, P. Reis, N. Lopes, Internet of Things: A system's architecture proposal, 9th Iberian Conference on Information Systems and Technologies (CISTI), Barcelona, Spain, pp.1 – 6, 2014.

[10] K. Ponnusamy, N. Rajagopalan, Internet of Things: A Survey on IoT Protocol Standards. In: Saeed K., Chaki N., Pati B., Bakshi S., Mohapatra D. (eds) Progress in Advanced Computing and Intelligent Engineering. Advances in Intelligent Systems and Computing, Volume 564, Springer, Singapore, pp. 651-663, 2018.

[11] M. Kaidan, et al., Configuration of network management for energy efficiency in optical transport networks using GMPLS and OBS techniques, Simulation Modelling Practice and Theory, Volume 74, pp. 17-27, 2017

[12] B. Mokhtar, et al., Big data and semantics management system for computer networks, Ad Hoc Networks, Volume 57, pp. 32-51, 2017

[13] W. Zhangchao, et al., An algorithm and implementation of network topology discovery based on SNMP, 2016 First IEEE International Conference on Computer Communication and the Internet (ICCCI), China, pp. 283 – 286, 13 - 15 Oct. 2016

[14] K. Bharadwaj, et al., Developing a Scalable SNMP Monitor, IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW), China, pp. 1043 – 1047, 2016

[15] M. Slabicki; K. Grochla, Performance evaluation of CoAP, SNMP and NETCONF protocols in fog computing architecture, IEEE/IFIP Network Operations and Management Symposium, Istanbul, Turkey. 25-29 APRIL 2016

[16] W. Cerroni, et al., Decentralized detection of network attacks through P2P data clustering, of SNMP data, Elsevier Computers & Security, Volume 52, pp. 1-16, 2015.

[17] Raphael C.-W. Phan Cryptanalysis of the application secure alternative to SNMP (APSSNMP), Computer Standards & Interfaces, Vol.31, No. 1, pp. 63-65, 2009

[18] H. Lamaazi, et al., Challenges of the Internet of Things: IPv6 and Network Management, Eighth International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing (IMIS), Birmingham, United Kingdom, pp. 328-333, 2-4 July, 2014.

[19] F. Sallabi, K. Shuaib, Internet of things network management system architecture for smart healthcare, 2016 Sixth International Conference on Digital Information and Communication Technology and its Applications (DICTAP), Mevlana University, Konya, Turkey, 21-23 July 2016.

[20] T. Brooks, Security and Trust Management for the Internet of Things: An RFID and Sensor Network Perspective, Wiley-IEEE Press, 17 Dec. 2016.

[21] S. Choi, S. Koh, Use of Proxy Mobile IPv6 for Mobility Management in CoAP-Based Internet-of-Things Networks, IEEE Communications Letters, Volume 20, No. 11, pp. 2284 – 2287, Nov. 2016.

[22] B. Gateau, J. Rykowski, Personal e-comfort modeling and management based on multi-agent system and Internet of Things network, International Conference on Pervasive and Embedded Computing and Communication Systems (PECCS), Angers (France), February 11-13, pp. 83-89, 11-13 Feb. 2015

[23] J. Chen, J. Liang, Z. Chen, Energy-efficient uplink radio resource management in LTE-advanced relay networks for Internet of Things, 2014 International Wireless Communications and Mobile Computing Conference (IWCMC), pp. 745-750, 04 Aug - 08 Aug 2014.

[24] J. Kim, Short paper: Wireless sensor network management for sustainable Internet of Things, 2014 IEEE World Forum on Internet of Things (WF-IoT), 6-8 March 2014. DOI: 10.1109/WF-IoT.2014.6803147

[25] D. Gao, et al., Information Perception and Intelligent Management for Electric Vehicle Charging-Swap Networks with the Internet of Things, IEEE 12th International Conference on Computer and Information Technology (CIT), Chengdu, Sichuan, China, 27-29 Oct. 2012. DOI: 10.1109/CIT.2012.80

[26] Z. Xinhua, L. Hong, A Self-Reconfigurable Sensor Network Management System for Internet of Things Paradigm, 2012 International Conference on Computer Science and Service System, pp. 311-314, August 11 - 13, 2012.

[27] H. Hui-Ping, X. Shi-De, M. Xiang-Yin Applying SNMP Technology to Manage the Sensors in Internet of Things, the Open Cybernetics & Systemics Journal, Volume 9, pp. 1019-1024, 2015.

[28] https://iot.do/friendly-technologies-iot-solutions-2016-10

[29] S. Samarah, A Data Predication Model for Integrating Wireless Sensor Networks and Cloud Computing, Procedia Computer Science, Volume 52, pp. 1141-1146

[30] O. Said and A. Elnashar, Probabilistic Queueing Scheme for Servicing E-Mail Using Markov Chains, Journal of Theoretical and Applied Information Technology (JATIT), (2013), Volume 56, No.2, pp. 314-323

[31] The Network Simulator – ns-2. 2008, Available from: http://www.isi.edu/nsnam/ns/

[32] O. Said, "Analysis, design and simulation of Internet of Things routing algorithm based on ant colony optimization," International Journal of Communication Systems, Wiley, 2016, DOI: 10.1002/dac.3174

[33] R. Dobrescu, F. Ionescu, Large Scale Networks: Modeling and Simulation, CRC Press, 2016