iJRASET
International Journal For Research in
Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 8 **Issue:** VII **Month of publication:** July 2020

DOI: https://doi.org/10.22214/ijraset.2020.30770

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.429
Volume 8 Issue VII July 2020- Available at www.ijraset.com

Design and Development of an Open Platform for Software-Defined Heterogeneous Wireless Sensor Networks

Shaliny Stanly A[1], Shijin Knox G U[2]
[1, 2]*Dept. Information Technology Government Engineering College Trivandrum, Kerala, India*

Abstract: Software Defined Networking (SDN) is an emerging technology which can reduce the complexity of network management and configuration. SDN was mainly focused on wired networks but now SDN has extended its scope to wireless sensor networks (WSN). Software Defined Wireless Sensor Networks (SDWSN) allows dynamic configuration of sensor nodes and centralised control over entire network which can save energy of sensor nodes and easy management of WSNs. Heterogeneity of sensor nodes is an important factor to be consider in the WSN. In this paper we design and develop an open platform for Heterogeneous Software Defined Wireless Sensor Network. Also we have designed a Device-control table for device level control and management. This can be useful for the implementation of SDWSN in real time applications.

Index Terms: SDN, WSN, SDWSN, Controller

I. INTRODUCTION

Applications of Wireless Sensor Networks are increasing day by day in fields such as military applications, environment monitoring, home and office applications etc. WSN resources such as energy, memory and processing capacity are limited and it is the main limitation of these networks. Several researches are carried out to overcome these limitations and it has become necessary to tackle those since the importance of WSN in modern technology.[1] This has lead to the proposal of integrating SDN technology in Wireless Sensor Networking. Software Defined Networking technology has developed for the wired network for the ease of network management and dynamic control via a logically centralised controller. In order to make it possible, the control plane and data plane are separated from the network device[2]. Data plane consist of network devices such as switches and routers where as control plane consist of one more controllers. These logically cen- tralised controller is capable of configuring and managing the entire network. Controller has global view of network and the routing decisions are given by the controller. Data plane device responsibility has limited to forwarding only according to the flow rule. SDN supports virtualization and thus reduces the hardware cost and provide efficient management of network. Dynamic configuration can be done easily by the controller. All these advantages can be utilised for the WSN also. This leads to the proposal of Software-Defined Wireless Sensor Networks.

Similar to SDN, SDWSN decouples the control plane from the sensor nodes. Every sensor node consist of flow table that contains flow rules given by the controller to handle the packets. Controller in SDWSN is usually a PC which has no limitation of energy like sensor nodes. Since the routing decisions and processing are done by controller energy of sensor nodes can be saved and thus improves network lifetime. Dynamic configuration and management of sensor nodes are not possible in WSNs,ie; once the devices are deployed reprogramming and configuring those nodes are not too easy task,it costs high[3]. By the use of SDWSN it becomes possible to dynamically program the nodes and manage whole network. Even though the SDWSN results in these advantages, it has not implemented in any real time applications yet. Why this technology is still resides in researching area only and not used in any applications? This has lead to the factor heterogeneity. Heterogeneity of sensor networks includes difference in processing and computation power, energy, protocols used platforms etc. In real time applications, usually consist of sensor nodes with different abilities. The sensor node with high sensing range and performance are placed far from the sink node and low sensing range nodes placed near sink in order to achieve balance of performance, improved network lifetime and reduce the WSN cost[4]. This mixed deployment method balance the performance of network in the scenario; the sensor node near sink needs only small sensing range where as sensor nodes far from sink needs to find nodes that are closer to sink in order to get shortest path, this will improve performance. If we are placing low energy devices near sink, they will ran out quickly since it requires more energy for relaying every packets from other nodes which result in reducing the network lifetime. Thus the importance of heterogeneous network is very much clear and this factor is not considered yet in SDWSN.

In this paper, we proposes an open platform for Software- Defined Heterogeneous Wireless Sensor Networks along with a device level control which saves energy and lifetime of the network and improved performance.

II. LITERATURE SURVEY

Deze Zeng et.al [3] proposes Software Defined wireless Sensor Network and its enabling technologies. The technologies needed for the realisation of SDWSN such as Soft- ware defined Radio, Field-Programmable Gate Array Technique (FPGA),operating systems, Over-The-Air-Programming Technique. All these technologies are existing and more ad- vanced and the convergence all these will help to realise SD- WSN efficiently. Tie Luo et.al[5] proposes integration of SDN and OpenFlow along with a flow creator module and developed an algorithm called Control Message Quenching(CMQ) for reducing flow setup latency and elevating network throughput. Bruno T. de Oliveira et.al[7] proposes TinyOS based SDN called Tiny SDN. It was the first implementation that uses the TinyOS. Alejandro De Gante et.al[6] proposes a smart WSN management using SDN and proposes an architecture for base station. Localization and tracking algorithms are proposed for accurate mapping and maintaining node positions. Using the proposed architecture WSN can be easily managed and energy can be saved. Laura Galluccio et.al[8] proposes an SDWSN solution for WSN called SDN WISE. It is a stateful solution with objectives of reducing the amount of transmission between controller and sensor nodes and make sensor nodes programmable as finite state machines. It provides flexibility in managing the network. Proposes a SDWSN whose role can be redefined after the deployment of sensor nodes which uses two main technologies, role generation and delivery mechanisms and re-configurable WSN with number of re-configurable sensor nodes. It uses FPGA and MCU together in order to reduce the overhead of MCU units in the networks. Sensing and data processing tasks are given to FPGA and thus the overhead of MCU unit can be reduced. Tryfon Theodorou et.al[9] proposes an SDN solution for WSN which uses intelligent centralised mechanism for dynamically controlling the network and supports elasticity to the challenging require- ments of the WSN called CORAL SDN. Scalability ofnetwork can also improved by this proposed architecture. CORAL-SDN uses its own protocol for the forwarding of information and CORAL-SDN controller module acts as the controller of the network which manages the entire network.

Pradeepa R et.al[10] proposes SDN enabled SPIN protocol. SPIN is a negotiation protocol used for efficient information forwarding in WSN. SPIN running based on the energy level of the host and the process depends on the remaining energy level of the host node. This advantage is taken into consider- ation for the SDWSN. TinySDN[7] uses the Collection Tree Protocol[CTP] for the routing. Compared to CTP protocol and normal SPIN protocol ,SDN- SPIN is good at energy saving. In case of hops generated, SDN-SPIN is better than the normal SPIN protocols. Normal WSN uses gossiping and flooding for the information forwarding, and it wastes much energy of sensor nodes for broadcasting information to the entire network. Cui Ding[11] uses SDN- enabled sensor nodes and uses Zigbee protocol for the communication. SDN-WISE[8] developed its own protocol stack called SDN-WISE protocol stack. It provides better performance in terms of efficiency. Pangu[12] added an additional condition for the routing called modality matching. Modality is a representation of grouping. Identified by it, nodes could be grouped by the data types, the capacities of their hardware, or even the priorities of sensing tasks, etc. Thus enables network users to centrally define the forwarding behaviors of nodes in the underlying opportunistic routing protocol. With the help of centrally- defined modality properties, the network is partitioned into multiple desirable sub-nets that can satisfy the upper-level requirements for system performance.

Zhu.Y.Zhang et.al[13] proposes a localization node selection algorithm based on Cramer-Rao Lower Bound(CRLB).Existing power allocating solutions are mostly distributed for the lack of global network knowledge. By making use of the global network knowledge provided by the SDN controller, they formulate the localization node selection into a 0-1 programming problem on the premise of energy satisfaction. Calculated the contributions of each anchor node in the metric of CRLB for a specific agent node and choose the most reliable nodes for localization. An anchor scheduling scheme is created in [18] which tries to maximise the lifetime of the network by limiting the amount of active anchor nodes within the network. It is done as the extension of CRLB method. This allows for nodes with shorter timeouts to contribute more to localisation. Cloete et.al[14] reviewed the existing localization methods in Wireless Sensor Networks and proposes a SDN-based localisation and compares it with some existing literature. The review shows that how traditional RSSI ranged based localisation methods can be improved by the centralised nature of the SDWSN controller. The centralised nature and global overview of the controller improves the accuracy of positional estimations but also enables management decision to be taken based on these results.

Jiangwei et.al[15] present a new clustering algorithm called Extending Dynamic Sub network Scheme—Low En- ergy Adaptive Clustering Hierarchy (EDSS-LEACH) based on software-defined wireless sensor network (SDWSN). Sensor nodes in WSNs have limited energy and the nodes near sink nodes usually ran out quickly than other nodes. Thus an imbalance in energy consumption occurs and results in low utilisation rate and short network lifetime. The proposed algorithm helps to extend the network life time. Clu-Flow[16] propose a cluster-based flow management approach that makes a trade-off between the granularity of monitoring by an SDN controller and the communication overhead of flow manage- ment. A network is partitioned into clusters with a minimum number of border nodes.

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.429
Volume 8 Issue VII July 2020- Available at www.ijraset.com

Instead of having to handle the individual flows of all nodes, the SDN controller only manages incoming and outgoing traffic flows of clusters through border nodes. CluFlow minimizes the number of border nodes and the communication overhead used for SDN control.

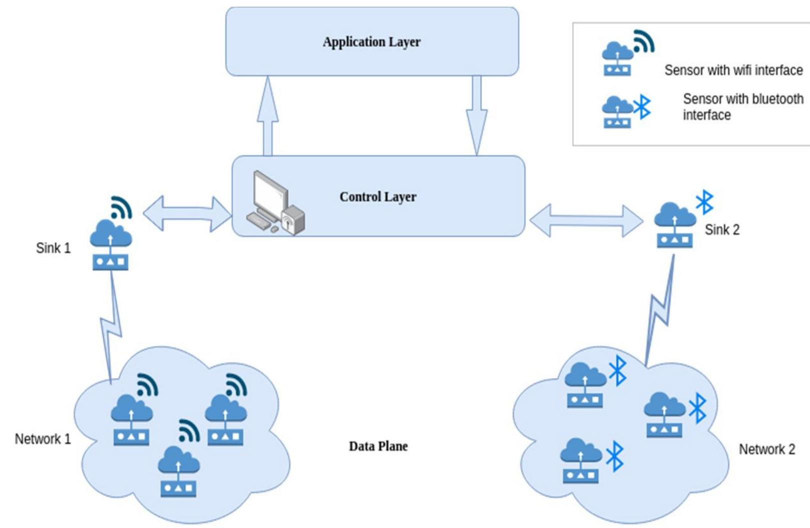


Fig. 1. Architecture of proposed SDWSN

III. PROPOSED SCHEME

The figure.1 shows the basic architecture of SDWSN. similar to SDN, it mainly consist of three layers. Application layer consist of sensor applications which can communicate with the network through controller. Control plane consist of controllers which has global view of the network and able to control the entire network. Controller is directly connected to sink node of each network. For realising the centralised control, its necessary to collect the view of the network. For this topology discovery protocol have to be implemented.

A. Topology Discovery

Topology discovery protocol is based on periodical trans- mission of beacon packet. The process is initiated by the sink node by broadcasting the beacon packet through the wireless channel. The beacon packet consist of the battery level and distance. The sensor nodes that receives the beacon packet will add the address of node to the neighbour list along with the battery level. If the address is already present in the neighbour list, then update the battery level. The distance is updated by adding one with the value in distance field of received packet.The neighbour list are cleared periodically and update the neighbour list. The collected neighbour-list are transmitted to the controller using report packet.Report packet are relayed through the shortest distance towards the controller.

B. Packets used

The figure.2. shows the packet header format.

1) *Network id:* Identity of the network from which the packet is sending.
2) *Length:* Length of the packet.
3) *Source:* address of the source node that sends packet.
4) *Destination:* Address of node to which the packet is sending.
5) *Type:* Type of packet that is sending.
6) *Time To Leave (TTL):* Lifetime of rule in the flow table.
7) *Next hop:* Address of next node in the path to reach destination.

Network ID	Length	Source	Destination	Type	TTL	Next hop

Fig. 2. Packet header format

a) Beacon Packet: Beacon packet is used for finding the neighbour list of each node. Type is set to ’0’ in the type field to identify the received packet is beacon.
b) Data Packet: Used to send the sensed data to con- troller. Type is set to ’1’.Packet is forwarded according to the flow rule.
c) Report Packet: Report packet is used to send the neighbor list to the controller. Type is set to 2.
d) Request Packet: Request packet is sent to the controller to request the flow entry. If the request packet with additional header information exceeds the payload limit and unmatched packet does not fit into one request, packet is split into 2parts. Type is set to 3.
e) Response Packet: Send by the controller in response to the flow request. Packet contains rule along with header. Type is set to 4.

C. Packet Handling

When a node receives the packet, it search for a matching rule from the flow table. If a match found the packet is handled according to the flow rule. Flow table format is shown in figure.3 with some example values. The first field is operator field which consist of relational operators to compare the received value and threshold value. For example if the received size of payload 2 and the threshold value we set to 2 and operator is ’=’ then the rule matched if the destination address is 0, then the packet is forwarded to the next hop and increment the counter. If no match is found in the flow table, the node sends a request packet to the controller. The controller will send the response packet with new rule to handle the that type of flow. Each rule will be removed after the TTL expires.

1) Shortest Path Calculation: In order to handle and for- ward the packets effectively, shortest path has to calculated by the controller. Here we used Dijkstra’s shortest path algorithm along with RSSI value. RSSI value is used as the cost function. A threshold value is set to RSSI and if the received RSSI of node is greater or equal then that node can be added toshortest path.

D. Device Control

We have introduced a device level control in SDWSN. Figure.4 shows the device control table and figure.5 shows the packet formats used for it. Through the device level control the administrator can control the device. For example if we need sensed data from 5 particular nodes in the network at a time, administrator can easily turn off the radio of all other sensor nodes using device-out packet with the specified time is set in TTL field. In the same way another controllable parameters can control using this technique. There will be different parameters that can control for different platforms. For example, in Memsic IRIS motes, we can control only the radio, where as the raspberry pi or other platforms have other parameters that can control.

Matching rule

Operand	Size	Value	Address	Action	TTL	Counter
=	2	x	0	Forward	100	10
>	2	x	1	Drop	122	23
<	0	x	2	Modify	125	5
=	1	x	10	Drop	100	66

Fig. 3. Flow Table format

Action	TTL
Radio On	100
Radion OFF	60

Fig. 4. Device control Table format

Using this technique we can save much energy of sensor nodes. Device-in packet is used by the sensor nodes to inform the controller about the battery level. ie, packet is send when the battery level is down below the threshold.Thus the admin can do appropriate action.

IV. IMPLEMENTATION

We have simulated the proposed scheme using Contiki OS and Cooja simulator. We have used sky motes and Z1 motes for the simulation of network . SDN-WISE framework and protocol is used for the implementation. Total average Power consumption of nodes is analysed using PowerTrace application in cooja. Device control can be used according to the platform used and parameters available that are controllable. Thus a generalized format is proposed and for testing radio on/off action is implemented. Thus by the use of device control power consumption has reduced from 2.55mV to 1.23 mV. Total simulation time was 30 minutes and carried out with and without device control for the analysis of power consumption.

V. CONCLUSION

We have proposed an Open platform for heterogeneous Software Defined Wireless Sensor Network along with a device level control. This scheme will be very helpful for the realisation of SDWSN in real time applications. Advantages of SDN had utilised efficiently to implement device-control. This work can also be tested with the Conitiki supported hardware platforms.

Fig. 5. (a) Device-out packet (b) Device-in packet

REFERENCES

[1] Sharma, S., Bansal, R. K., Bansal, S. (2013). Issues and Challenges in Wireless Sensor Networks. 2013 International Conference on Machine Intelligence and Research Advancement. doi:10.1109/icmira.2013.18

[2] Tatang, D., Quinkert, F., Frank, J., Ropke, C., Holz, T. (2017). SDN- Guard: Protecting SDN controllers against SDN rootkits. 2017 IEEE Conference on Network Function Virtualization and Software Defined Networks (NFV-SDN)

[3] Zeng, D., Miyazaki, T., Guo, S., Tsukahara, T., Kitamichi, J., Hayashi, T. (2013). Evolution of Software-Defined Sensor Networks. 2013 IEEE 9th International Conference on Mobile Ad-Hoc and Sensor

[4] Wu, C.-H., Chung, Y.-C. (n.d.). Heterogeneous Wireless Sensor Net- work Deployment and Topology Control Based on Irregular Sensor Model. Lecture Notes in Computer Science, 78–88

[5] Luo, Tony Tan, Hwee Quek, Tony Q.S.. (2012). Sensor OpenFlow: Enabling Software-Defined Wireless Sensor Networks. IEEE Communi- cations Letters. 16. 1896–1899.10.1109/LCOMM.2012.092812.121712.

[6] De Gante, A., Aslan, M., Matrawy, A. (2014). Smart wireless sensor network management based on software-defined networking. 2014 27th Biennial Symposium on Communications (QBSC).

[7] De Oliveira, B. T., Margi, C. B., Gabriel, L. B. (2014).TinySDN: Enabling multiple controllers for software-defined wireless sensor net- works. 2014 IEEE Latin-America Conference on Communications (LAT-INCOM). doi:10.1109/latincom.2014.7041885

[8] Galluccio, L., Milardo, S., Morabito, G., Palazzo, S. (2015). SDN-WISE: Design, prototyping and experimentation of a stateful SDN solution for WIreless SEnsor networks. 2015 IEEE Conference on Computer Communications (INFOCOM)

[9] Theodorou, T., Mamatas, L. (2017). CORAL-SDN: A software-defined networking solution for the Internet of Things. 2017 IEEE Conference on Network Function Virtualization and Software Defined Networks (NFV- SDN).

[10] Pradeepa R, Pushpalatha, M. (2016). SDN enabled SPIN routing pro- tocol for wireless sensor networks. 2016 International Conference on Wireless Communications, Signal Processing and Networking (WiSP- NET).

[11] Ding, C., Shen, L. (2017). Design and Implementation of Programmable Nodes in Software Defined Sensor Networks. 2017 IEEE 85th Vehicular Technology Conference (VTC Spring).

[12] Guo, J., He, Y., Zheng, X. (2017). Pangu: Towards a Software-Defined Architecture for Multi-function Wireless Sensor Networks. 2017 IEEE 23rd International Conference on Parallel and Distributed Systems.

[13] Zhu, Y., Zhang, Y., Xia, W., Shen, L. (2016). A Software-Defined Network Based Node Selection Algorithm in WSN Localization. 2016 IEEE 83rd Vehicular Technology Conference (VTC Spring)

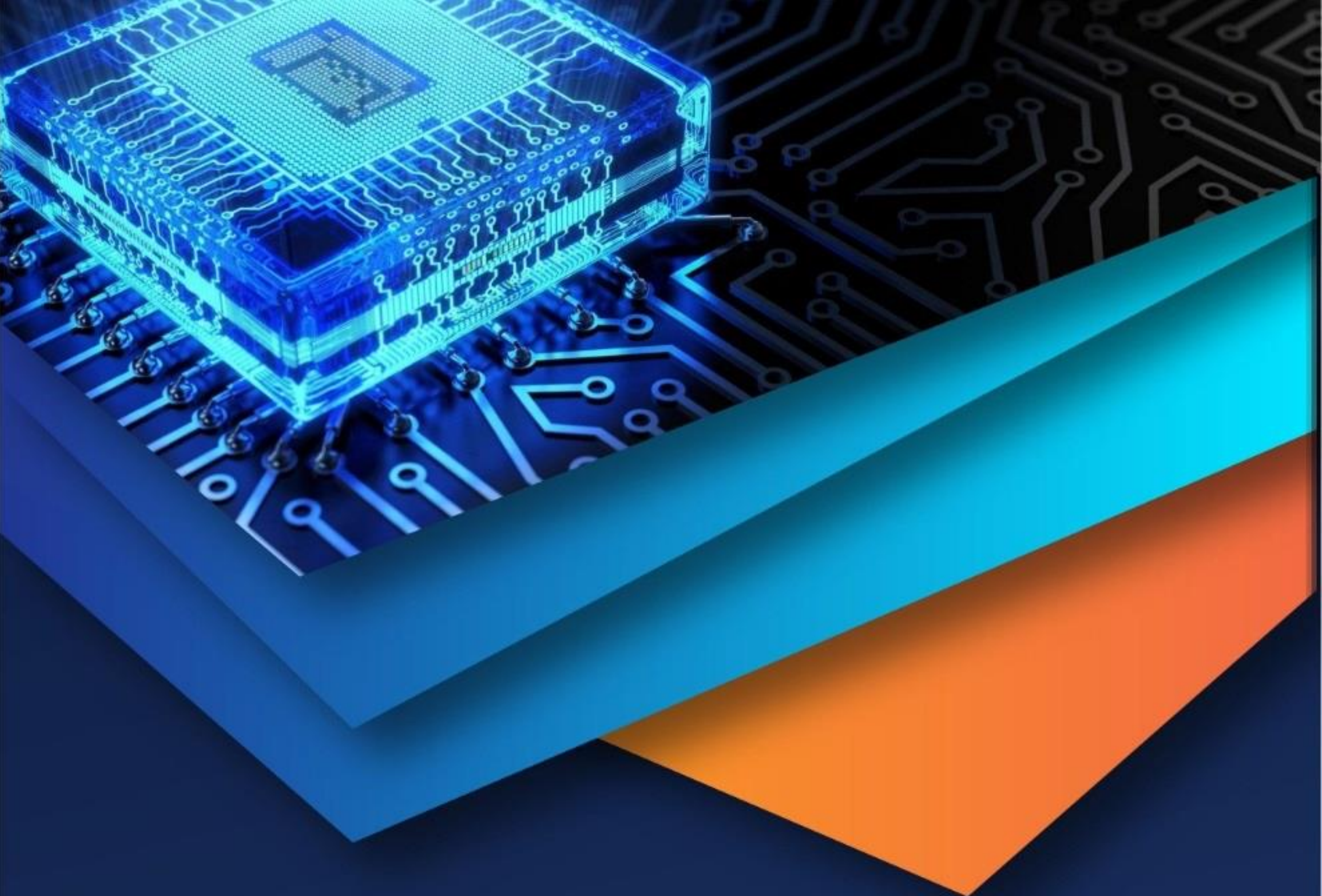
[14] Cloete, O. P., Abu-Mahfouz, A. M., Hancke, G. P. (2019). A Review of Wireless Sensor Network Localisation Based on Software Defined Net- working. 2019 IEEE International Conference on Industrial Technology (ICIT). doi:10.1109/icit.2019.8755075

[15] Jiangwei, S., Muqing, W., Zhihao, Z. (2018) Clustering Algorithm Based on Extending Dynamic Subnetwork Scheme for Software-Defined Wireless Sensor Networks. 2018 IEEE/CIC International Conference on Communications in China (ICCC Workshops.

[16] Liu, Q., Ozcelebi, T., Cheng, L., Kuipers, F., Lukkien, J. (2019). CluFlow: Cluster-based Flow Management in Software-Defined Wireless Sensor Networks. In 2019 IEEE Wireless Communications and Net- working Conference, WCNC 2019 (Vol. 2019-April). [8885485] IEEE.

[17] Mukherjee, M., Shu, L., Zhao, T., Wang, D., Wang, H. (2017). Lightweight flow management for software-defined wireless sensor networks with link fault in data plane. 2017 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS).

[18] Y. Zhu, F. Yan, Y. Zhang, R. Zhang, and L. Shen, “Sdn-based anchor scheduling scheme for localization in heterogeneous wsns,” IEEE Com- munications Letters, vol. 21, pp. 1127–1130, May 2017.



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY