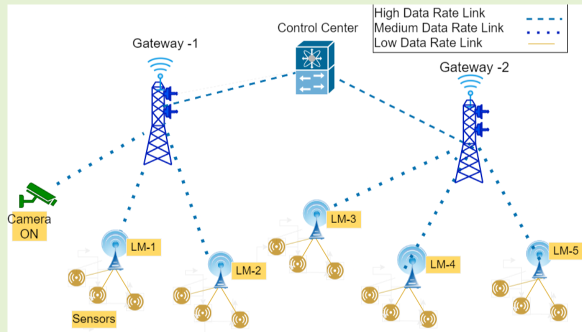
Dynamic Load Balancing in multiple gateway based Industrial IoT Networks

Saikat Mondal, Anand Prakash Rawal and Prasenjit Chanak

***Abstract*—The phrase ”Industry 4.0” refers to a new way of thinking about how technology, particularly the Internet of Things (IoT) and Cyber-Physical Systems (CPS), may be applied in production. The total digitalization of the production process is one of the key goals of this fourth industrial revolution in order to realize the interconnectedness of people, machines, and gadgets. In this process, Wireless sensor networks(WSN) will play a crucial role. As the Cyber Physical Production System(CPPS) grows in Industrial IoT(IIoT) , the aggregation and reliable management of large amounts of data and dynamic reconfiguration of data links under spatio-temporal variation of data are vital aspects of this paradigm. With this broader goal, this paper proposes a load balancing scheme that effectively monitors wireless connections, dynamically reconfigure under spatio-temporal variation of data, detects possible risks of channel saturation, and distributes data load among available nodes with minimizing the number of reconfigurations, total energy consumption.**



***Index Terms*—CPS, IoT, IIoT, WSN, Industry 4.0, CPPS, load balancing, Channel saturation**

I. Introduction

INDUSTRY 4.0 is a paradigm that focuses on the digitalization and integration of existing industries to create smart and adaptable factories [1]. A robust commnunication infrastructure is required for Cyber-Physical Production Systems (CPPS) to become ubiquitously connected [1] [2]. Although protocols like ISA100.11a and WirelessHart are used by conventional industrial wireless networks (IWN), they can only bear data transfer at low bandwidth.

The periodic generation of extensive amounts of data from devices like IP cameras or 3D scanners requires high bandwidth connection links. Additionally, existing centralized network system leads to signaling overhead, prolong reconfiguration periods, and scalability issues. To cope up with these challenges, hierarchical IWNs that can co-ordinate several subnetworks each managed by various wireless technology can be used to dynamically react under spatio-temporal variations of data requirement and distribution to avoid saturation of wireless links and loss of precious data.

A. Background

In wireless sensors networks, a proper load balancing scheme should have the following capabilities:

- Keeping track of the status of wireless connections: The load balancing scheme should be able to monitor the status of all wireless connections in the network. This includes the number of connected devices, signal strength, and available bandwidth.
- Detecting probable danger of channel saturation: The scheme should be able to identify when a wireless channel is becoming congested and could potentially cause performance degradation [3].
- Effectively distributing the data load among nodes: Once the load balancing scheme has gathered information about the present status of wireless connections and identified potential saturation risks, it should distribute data load among available nodes [4].

By effectively monitoring wireless connections, detecting risks of channel saturation, and distributing data load among available nodes, a load balancing scheme can enhance the overall performance and reliability of wireless networks [5].

The focus of this scheme is to balance the load of data links between gateway node and sink node , and it is capable of handling spatio-temporal variations of data. Additionally, it aims to reduce the number of reconfigurations required for wireless links in deploying self-organizing IWNs as well as minimize the total energy consumption of the system .By optimizing the load balancing and reducing the need for frequent reconfiguration, this scheme can improve the performance and reliability of self-organizing IWNs.

II. Related Works

Different approaches for load balancing have been suggested in traditional cellular and wireless networks to enhance network performance. For example, So as to keep the load at any access point (AP) equal to the average network load, the load balancing scheme in IWN proposed in [7] distributes devices across the APs. An AP's load is calculated as the ratio of total bandwidth needed to total bandwidth actually available at the AP. The suggested scheme in [7] assumes all the links of same AP experience same Packet Error Rate(PER). The link quality of links can not be neglected since if a device has bad link quality, it would need more bandwidth to send the same data packets compared to a device with better link quality. So, not considering it can result in imbalanced network traffic and poor network performance. The schemes proposed in [8] [9] uses the data queue length of a node as an alternative statistic for load balancing. This metric calculates how much data the node has still to send overall.This scheme proved to be impressive in improving throughput and reducing delay. However, a limitation of using data queue length as a measure of congestion in [8] and [9] is that it does not give adequate insight about the required bandwidth to transmit data to its destination. It is also essential to consider the link quality of the wireless connection and its variations while estimating the required bandwidth. In [10] a gradient based method using Newton-Raphson, Gauss-Seidel algorithms is proposed.The main drawback of this methodology is its high complexity and high execution time. In [11] fog load balancing scheme using random walk is applied , this scheme has low response time but has very low scalibiltiy. In [12], a load balancer based on fuzzy logic is deployed. This proposed scheme in [13] provided low latency and low energy consumption but it had low reliability and less security. In [14] , a meta-heuristic approach based on Bat algorithm is used to efficiently perform load balancing . But in [14] , the drawbacks were high complexity, possibility of bottle neck, low scalability. There are very few works in this field that considered dynamical reconfiguration of links under spatio-temporal variations of data. The proposed scheme mentioned in [1] as CUBE(Channel Utilization Load Balancing Scheme) is designed for self-organizing Industrial Wireless Networks (IWNs) and involves the deployment of hierarchical IWNs with many sub-networks. In [1] , Local Manager (LM) nodes that act as both the sink node and manager for each sub-network. The gateway nodes which are connected to the LM nodes accumulate data from various LM nodes. Large capacity (fixed or wireless) backhaul links can be used to connect the gateway nodes to on-site or remote control, data centres, and servers (orchestrator), as well as other devices. The focus of this scheme is to balance the load of connections between LM and gateway nodes in such a manner that the scheme is capable of handling spatio-temporal variations of data. Additionally, it aims to reduce the number of inter-gateway channel switching required for wireless links in deploying self-organizing IWNs. [1] is by far the most impressive in terms of results. However, the main drawbacks of [1] are - the load balancing algorithm in this scheme is executed even when load is balanced in network , there are scenarios when it is optimal for LM to change serving gateways to balance load among gateway channels but CUBE prioritizes changing channels within serving gateways, it is not energy efficient.

A. Motivation

The noble scheme proposed in this paper is based on the modified version of CUBE [1]. In Industrial IoT (Internet of Things), all gateway nodes and LM's position are mostly fixed . Whenever load among gateway channels is not balanced , Channel Utilization Balancing schemE (CUBE) is executed and dynamic auto allocation and reconfiguration of gateway channels happen in such a manner that objective function is minimized. In objective function, two parameters are minimized maximum load of any channel and LM's changing their serving gateway i.e. intergateway channel switching.

By minimizing the maximum load and number of inter-gateway channel switching, it is evident that energy efficiency is achieved to a certain extent but using the fact that all positions of LM nodes and gateway nodes are fixed in IIoT, if we take into account a new parameter i.e. distance between LM node and its serving gateway, maximum energy efficiency can be achieved. To do so, our proposed algorithm should ensure that if LM's are changing their gateway then it should change to its nearest optimal gateway. Also, this can improve network performance and reduce the likelihood of congestion and latency issues.

III. Network Model

Hierarchical IWNs with multiple sub-networks is deployed in this proposed scheme. The components of this network is -

- Sensors, actuators, IP camera, 3D scanner etc.
- Local Manager(LM)
- Gateway Node(GW)
- Orchestrator(Control Center)

In this scheme, each sub-network has a local manager (LM) which acts like both a sink node and a network manager i.e. aggregates all data within its sub-network and manages the links in its sub-network. Different protocols like ZigBee, IEEE 802.15.4e, ISA100.11a, WirelessHART are possible choices to be implemented in each subnetwork based upon its required bandwidth or communication range. LM nodes are connected to some gateway via IEEE 802.11 (or WiFi). One gateway serves mutliple LM nodes and amass data from various LM nodes.This gateway nodes are connected to remote or onsite control center through extensive (fixed or wireless) backhaul links. This schemes aims to balance the load of data links between gateway nodes and LM nodes, and is capable of handling spatio-temporal variations of data with minimizing the number of inter-gateway channel switching energy-efficiently.An example of dynamic link reconfiguration under spatio temporal variation of data is illustrated in Fig2. Initially, due to the network configuration in (Fig2.a), LMs were able to collect all data from sensors and was able to transmit it to the Control Center via the Gateway Nodes but as soon as IP camera became active, (Fig2.b) load in gateway 1 increased as a result to the balance the load and avoid channel saturation, LM2 shifted to gateway 2.

IV. Load Balancing Proposal

In this scheme,the load of a channel is calculated to be the percentage of time the channel has been utilizied by all

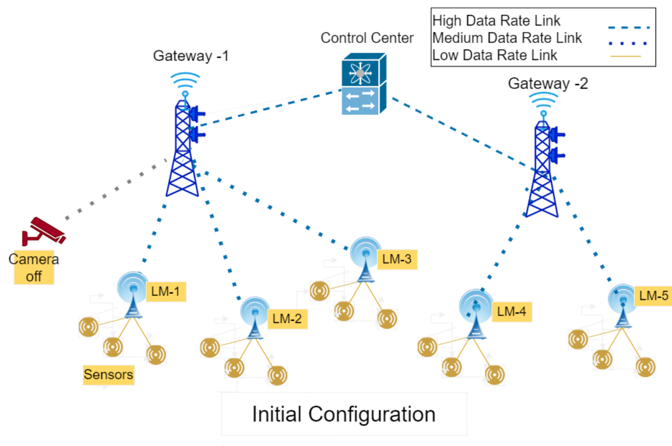


Fig. 1

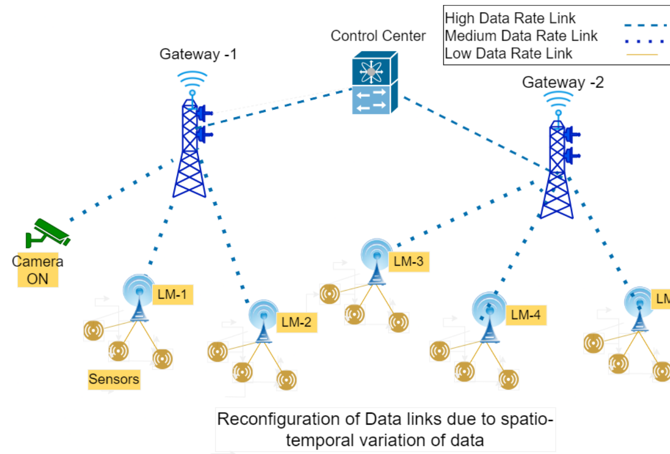


Fig. 2

the LMs it serves [1]. This new statistic is mentioned as CU or Channel Utilization percentage, and the suggested load balancing scheme is named as MCUBE (**M**odified **C**hannel **U**tilization Load **B**alancing schem**E**).Here, IEEE 802.11 (or WiFi) is taken into account to wirelessly link gateway nodes and LM nodes. The Gateway nodes function as access points (APs) and use IEEE 802.11a with Point Coordination Function (PCF) to regulate the access to the channel of the attached LM nodes and to avoid packet collisions. Due to this, many LMs can be served by a Gateway's channel. The Gateway nodes and LMs relentlessly track the link quality (specifically, Signal to Noise ratio, SNR) in every links and input data rate and transfer this reports to the control center. Based on the information received time to time from LM and Gateway nodes, the orchestrator in control center prompts execution of MCUBE to properly maintain load balanced in all channel of the gateways by delivering reconfiguration information to LM and Gateway nodes. Based on the instruction, the LM node can change its channel with in its serving gateway or even change to a new serving gateway. So, MCUBE first prioritizes balancing load among gateway channels and secondly LMs changing channel within serving gateway and thirdly LMs changing to nearest optimal gateway including the serving gateway. In our scheme, there are LM nodes denoted by i from 1 to L and Gateway nodes denoted by j from 1 to G and for each Gateway j there are channels denoted by c from 1 to C_j .

- $status[i][j][c]$ is a binary variable which is 1 if channel c is used by LM i to communicate with Gateway j and equal to 0 otherwise.
- $dist[i][j]$ denotes distance between Gateway j and LM i.
- $cnt[i][j][m]$ is a binary variable which denotes channel switching status from gateway j to gateway m for LM i . Simply , if before execution of MCUBE , LM i is connected to gateway j and after execution of MCUBE LM i changed its serving gateway to gateway m then cnt[i][j][m] is equal to 1 and equal to 0 otherwise.
- $gateways_{before}[i]$ denotes LM i is connected to which gateway before execution of MCUBE.
- $gateways_{after}[i]$ denotes LM i is connected to which gateway after execution of MCUBE.
- $\widehat{CU}_{\mathrm{jc}}$ is the calculated load of channel c within Gateway j.
- $\widehat{CU}_{\mathrm{ijc}}$ is the calculated load due to each LM i connected to Gateway j through channel c.

MCUBE seeks to minimize the maximum load among all channels by deciding the optimal network configuration i.e. which LM should connect to which gateway through which channel.

$$min \max_{j,c} \widehat{CU}_{\mathrm{jc}}, suchthat\ \widehat{CU}_{\mathrm{jc}} = \sum_{i=1}^{L} \widehat{CU}_{ijc} \cdot status[i][j][c]$$

This can be linearly expressed as following
$min\, X$
st:

$$(1) \sum_{i=1}^{L} \widehat{CU}_{ijc} status[i][j][c] \le X, \forall j \in [1,...,G], \forall c \in [1,...,C_j], X \in \mathbb{R}, X < 1$$

Let's assume, LM i is connected to Gateway j through channel c, i.e.$gateways_{before}[i] = j$. Assume that after execution of MCUBE, LM i is assigned to a new Gateway m with $m \neq j$ using channel n, $n \in [1, C_{\mathrm{m}}]$,i.e. $status[i][m][n] = 1, gateways_{after}[i] = m$ so,
Before execution of MCUBE : $status[i][j][c] = 1$
After execution of MCUBE : $status[i][m][n] = 1$
This implies $cnt[i][j][m] = 1$
Now, to minimize the number of inter-gateway channel switching, Gateway MCUBE seeks to minimize the following expression

$min\, Y$

$st:.$

$$(2) Y = \sum_{i=1}^{L} \sum_{j=1}^{G} \sum_{\substack{m=1 \\ m \neq j}}^{G} cnt[i][j][m]$$

$$(2.1) \sum_{j=1}^{G} \sum_{\substack{m=1 \\ m \neq j}}^{G} cnt[i][j][m] = 1 \text{ if LM i changes gateway}$$

$$(2.2) \sum_{j=1}^{G} \sum_{\substack{m=1 \\ m \neq j}}^{G} cnt[i][j][m] = 0 \text{ if LM i doesn't change gateway}$$

Now, to make scheme more energy-efficient , MCUBE should ensure LM's changing their gateway to nearest optimal gateway . To do so , it minimizes the difference of distance between LM i , current gateway(m) after execution of MCUBE and previous gateway(j) before the execution of MCUBE.
$min\, Z$
$st:.$

$$(3) Z = \sum_{i=1}^{L} \sum_{j=1}^{G} \sum_{\substack{m=1 \\ m \neq j}}^{G} cnt[i][j][m](dist[i][m] - dist[i][j])$$

$$(3.1) cnt[i][j][m] = \sum_{c=1}^{C_j} status[i][j][c]_{before} \& \sum_{c=1}^{C_m} status[i][m][c]_{after}$$

MCUBE also assures that no LMs are disconnected i.e. Lm is connected via some channel following the restriction

$$(4) \sum_{j=1}^{G} \sum_{c=1}^{C_j} status[i][j][c] = 1 \, \forall i \, \in \{1, ..., L\}$$

$$of:. (5) min\, X + \frac{1}{W_1} Y + \frac{1}{W_2} Z$$

$st:$

$$(5.1) \sum_{i=1}^{L} \widehat{CU}_{ijc} \cdot status[i][j][c] \leq X, \forall j \in [1, ..., G], \forall c \in [1, ..., C_j]$$

$$(5.2) Y = \sum_{i=1}^{L} \sum_{j=1}^{G} \sum_{\substack{m=1 \\ m \neq j}}^{G} cnt[i][j][m]$$

$$(5.3) Z = \sum_{i=1}^{L} \sum_{j=1}^{G} \sum_{\substack{m=1 \\ m \neq j}}^{G} cnt[i][j][m](dist[i][m] - dist[i][j])$$

$$(5.4) cnt[i][j][m] = \sum_{c=1}^{C_j} status[i][j][c]_{before} \& \sum_{c=1}^{C_m} status[i][m][c]_{after}$$

$$(5.5) \sum_{j=1}^{G} \sum_{c=1}^{C_j} status[i][j][c] = 1 \, \forall i \, \in \{1, ..., L\}$$

$$(5.6) X \in \mathbb{R}, X < 1$$

$$(5.7) W_1, W_2 \in \mathbb{R}, 0 < \frac{1}{W_2} < \frac{1}{W_1} <<< 1$$

In the objective function defined above the second term is multiplied by the factor $\frac{1}{W_1}$ and third term by $\frac{1}{W_2}$, where W_1 and W_2 are large numbers $W_2 > W_1$. As a result, MCUBE seeks minimizing X with prioritizing solutions-

- that minimize the count of LMs changing their serving gateway (Intergateway channel switching for LM).
- If LM's are changing the gateway to balance load among gateway channels , LM's should switch to one of the nearest optimal gateway to ensure MCUBE scheme is energy efficient.

The optimization problem is a mixed integer programming (MIP) problem with binary variable $status[i][j][c]$ and the real variable X.

All Gateway nodes calculate the estimated load of all channels under it and sends this information to the orchestrator every t_{MCUBE} seconds. The orchestrator periodically checks every t_{MCUBE} seconds the following conditions and executes MCUBE if the following condition becomes true -

- there exists a channel load(CU_{jc}) greater than the predeifend threshold load value (CU_{th})

After each execution of MCUBE, the value of CU_{th} is updated as a function of the optimum value of X (represented by X*) and $status[i][j][c] \, \forall i \, \forall j \, \forall c$ are updated according to the current configuration of gateway channels and LM.

A. Load Estimation

The channel load(CU_{ijc}) can be calculated using the method mentioned in [1].The process is briefly described below-

- Each LM i report
  - SNR value with gateways with in its communication range
  - $IRate_i$ the input data rate of LM in MAC sublayer
- Now, The Orchestrator estimates packet error rate using the SNR values and calculates the minimum output rate of data queue in LM so that data queue doesn't augment $ORate_i \geq \frac{IRate_i}{1-PER_{ijc}}$, PER_{ijc} is packet error rate of LM i with gateway j through channel c.
- $P_{ijc} = \lfloor \frac{ORate_i}{L_{max}} \rfloor$, L_{max} maximum length of the payload of a data packet transmitted
- Channel c should be used by LM i to send P_{ijc} data packets with a payload of Lmax bits, plus one extra data packet with a payload of L bits to gateway j where $L = ORate_i - P_{ijc} L_{max}$
- $\widehat{CU}_{ijc} = P_{ijc} \widehat{T}_{ijc}(L_{max}) + \widehat{T}_{ijc}(L)$
  $Tijc(L_{max})$ and $T_{ijc}(L)$ indicate an estimate of the amount of time that LM i spends on channel c when sending a data packet with a payload of L_{max} and L data bits, respectively, to Gateway j. By calculating these two values , CU_{ijc} can be calculated.

Algorithm 1

1 **if** $T_{ela} \% T_{MCUBE} == 0$ **then**
2 Check all channel load
3 **if** $CU_{jc} > CU_{th} \forall j \forall c$ **then**
4 Execute MCUBE
5 Update(CU_{th}),Update status[i][j][c] $\forall i \forall j \forall c$
6 Go To Line 2
7 **else**
8 $T_{ela} \leftarrow T_{ela} + 1$
9 **end if**
10 **else**
11 $T_{ela} \leftarrow T_{ela} + 1$
12 Go To Line 1
13 **end if**

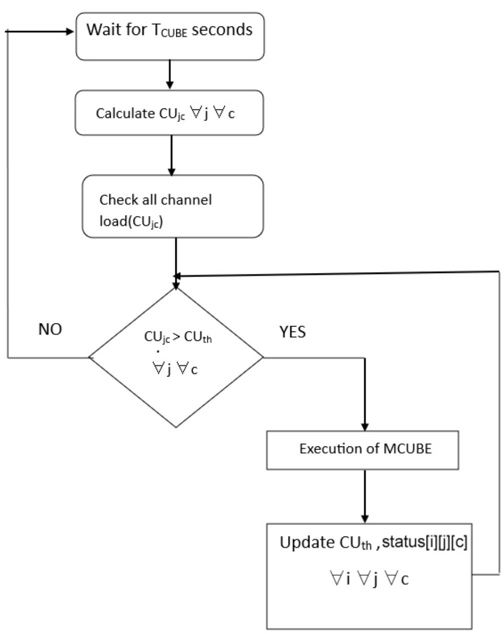


Fig. 3: Operation Of MCUBE

Algorithm 2 CU_{th} update ,$\beta_1, \beta_2 \in [0, 1]$

if $CU_{th} < X^*$ **then**
  $CU_{th} = X^* + \beta_1$
else
  $CU_{th} = CU_{th}.\beta_2$
  if $CU_{th} < X^*$ **then**
    $CU_{th} = X^* + \beta_1$
  end if
end if=0

B. Simulation Parameter

Parameter	Value
T_{MCUBE}	0.1s
β_1	0.05
β_2	0.95
Data Queue Capacity of LM nodes	1500kb
L	9
G	3
max(Cj)	5

V. Performance Evaluation

Authors developed a custom discrete-event simulator in C++ to evaluate the schemes and scenarios. The simulator incorporates all necessary aspects required for a precise evaluation of load balanced system performance in industrial wireless networks. The platform precisely replicates and models the LMs to Gateway connections that carry out the load balancing schemes under consideration. The simulator also uses SCIP to solve the mixed integer programming optimization problem. The schemes are assessed in a situation that simulates a 300 x 200 m industrial plant with halls that are 20 m [11] wide and arranged as shown in Fig. 3. This hypothetical situation is based on a real industrial facility with broad hallways and sizable rooms divided by concrete walls. This arrangement consists of 9 LMs and 3 gateway nodes. As the input data rate at the data queue of LM changes due to the fixed and mobile sensor nodes , the load in the channel connecting the LM to the gateway also changes. As a result, MCUBE scheme is executed automatically and dynamic reconfiguration of links happen in such a way that no channel within a gateway gets saturated, packet drops due to the overflow of data queue of LM is less, inter gateway channel switching is minimal and energy consumption is minimum.

The proposed MCUBE scheme is compared with fixedGW,QUEUE [9]and CUBE [1]. Here, fixedGW means fixed assignment of LM to Gateway i.e. there is no dynamic reconfiguration , all links are pre decided and fixed.

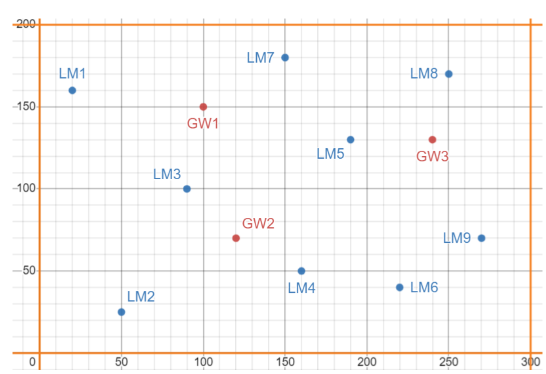


Fig. 4: Evaluation environment

A. Percentage data received

Fig 5 and 6 depict how all the data dispatched by the sensor nodes in the plant is split up among the LMs in CUBE

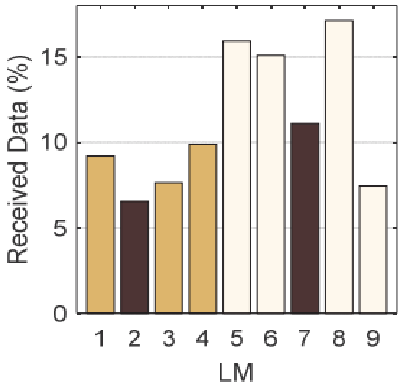


Fig. 5: Percentage of data received out of total data at LM in CUBE [1]

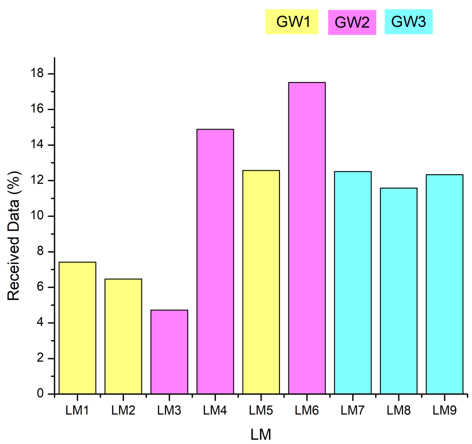


Fig. 6: Percentage of the total Percentage of data received out of total data at LM in proposed scheme MCUBE

and MCUBE. In CUBE, LM 5,6,7,8 received maximum percentage data while in MCUBE, LM 4,5,6,7,8,9 received more data. As compared to CUBE, MCUBE received more uniform data distribution and MCUBE received more data from sensors compared to CUBE. The colors of the LM bars in graph represent which gateway it is connected to in fixed assignment of gateways(fixedGW).

B. Percentage Packet loss

From Fig 7, 8 , In CUBE, LM1,LM2,LM3,LM4 received negligible packet losses where as in MCUBE LM1,LM2,LM3,LM7,LM8,LM9 experienced relatively less packet loss. In CUBE, LM5 in 2nd scenario experienced more than 2% packet loss . But in case of MCUBE, % packet loss never crossed 2%. So, MCUBE even after dealing with more data is able to minimze the maximum packet loss . So , in this point of view, MCUBE has outperformed all CUBE, QUEUE, FixedGW .

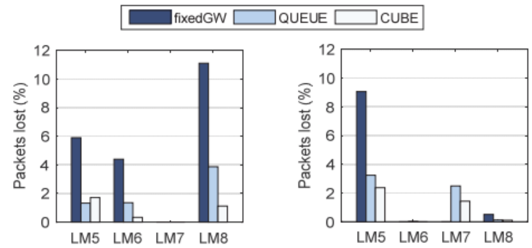


Fig. 7: packets loss percentage at LM in CUBE

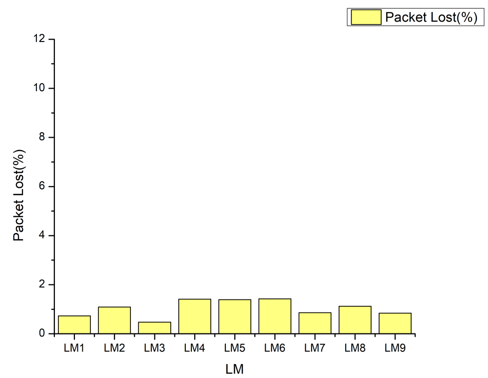


Fig. 8: packets loss percentage at LM in MCUBE

C. Inter-Gateway channel switching

In a network, Intergateway channel switching of LM causes more signalling overhead and require long reconfiguration period [1] . So, it is energy inefficient to change gateway most of the times [1].From Fig 9 and 10 , it is obvious that MCUBE has less intergateway channel switches.because in QUEUE and CUBE, there are 6 intergateway switches whereas in MCUBE only 4 switching occurs. So, MCUBE is better in reducing no of reconfiguration of data links.

.

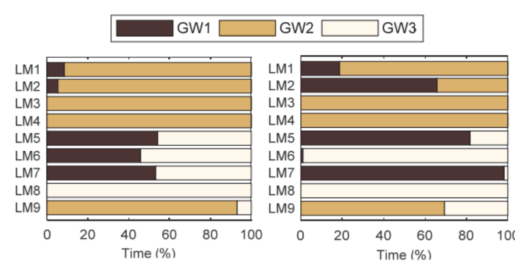


Fig. 9: Percentage time out of total time each LM is connected to Gateway in QUEUE and CUBE

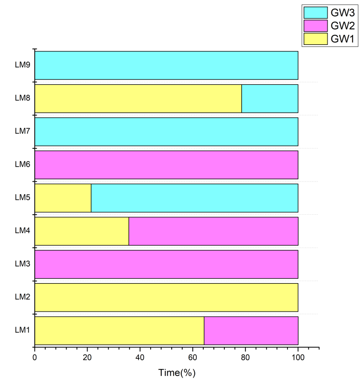


Fig. 10: Percentage time out of total time each LM is connected to a Gateway in MCUBE

D. Average Channel Utilization for Gateways

From Fig 11,12,13,14,15 , it is evident MCUBE outperforms QUEUE, fixedGW at all t=25%,50%,75%,100%.Because, in MCUBE all the three gateways are utilized almost same percentage i.e. load distribution is uniform.So, MCUBE ensures that none of the gateways get saturated and hence it can better handle spatio-temporal variation of data. Now, for comparison with CUBE, in CUBE GW3 is utilized more than 40% whereas in MCUBE , all gateway channel utilization is always less than 40%.

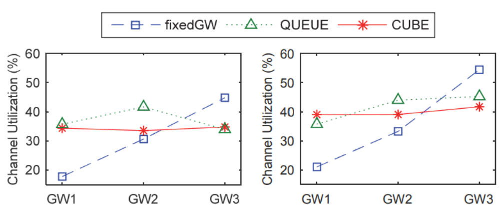


Fig. 11: Average Channel Utilization For Gateways

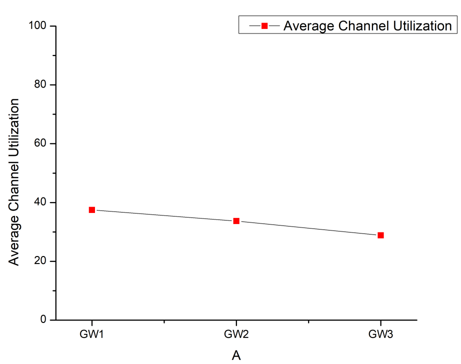


Fig. 12: Average Channel Utilization For Gateways in MCUBE for time=25%

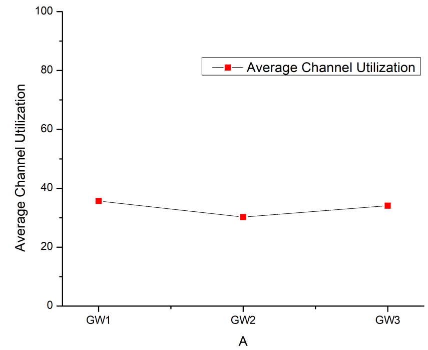


Fig. 13: Average Channel Utilization For Gateways in MCUBE for time=50%

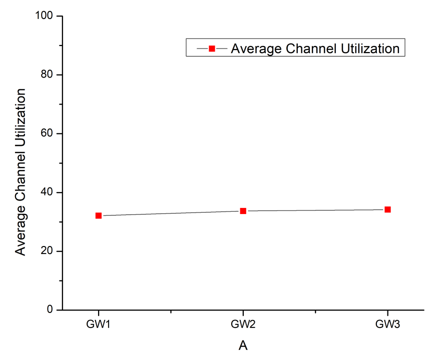


Fig. 14: Average Channel Utilization For Gateways in MCUBE for time=75%

E. Energy consumption

It is obvious that as transmission distance rises, so does the amount of energy used by each node. The proportionality between the energy consumed per unit meter and the transmission distance is the cause of the sharp increase in energy consumed per unit meter [15]. From Fig-4,9,10 it is evident in MCUBE, if LM's are changing their serving gateway then they are changing to their nearest optimal gateway so that maximum load of any channel and no of intergateway channel switching is minimized. For example, LM4, LM5, LM8 changes their serving gateway to nearest optimal gateway GW2, GW3, GW3. respectively.

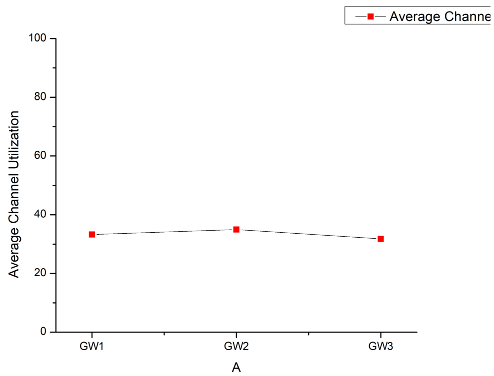


Fig. 15: Average Channel Utilization For Gateways in MCUBE for time=100%

VI. Conclusion

In this study, a dynamic load-balancing strategy for industrial wireless networks was described and assessed. The plan's goals were to accommodate the anticipated spatiotemporal changes in IIoT data as well as the implementation of dependable and self-organizing industrial wireless networks. The performed analysis has shown that, in comparison to conventional deployments where wireless links between nodes are often preset and fixed, the suggested load balancing technique greatly increases dependability . Also , this scheme outperforms both QUEUE [9] and CUBE [1] which were already one of the best schemes in the dynamic load balancing paradigm. The suggested load balancing scheme not only balances the load among channels but also minimizes the number of reconfiguration of links and energy consumption. The simulation results indicate that this proposed scheme can be implemented in real life in Industrial IoT.

References

[1] M. C. Lucas-Estañ and J. Gozalvez, "Load Balancing for Reliable Self-Organizing Industrial IoT Networks," in IEEE Transactions on Industrial Informatics, vol. 15, no. 9, pp. 5052-5063, Sept. 2019, doi: 10.1109/TII.2019.2898173.

[2] Usman Ahmed, Jerry Chun-Wei Lin, and Gautam Srivastava. 2022. Heterogeneous Energy-aware Load Balancing for Industry 4.0 and IoT Environments. ACM Trans. Manage. Inf. Syst. 13, 4, Article 46 (December 2022), 23 pages. https://doi.org/10.1145/3543859

[3] Kumar, Rohit Taneja, Kavita Taneja, Harmunish. (2021). Tackling control channel saturation for load balancing in multi-channel mobile ad hoc networks. International Journal of Information Technology. 13. 10.1007/s41870-021-00625-x.

[4] J. M. Shah, K. Kotecha, S. Pandya, D. B. Choksi and N. Joshi, "Load balancing in cloud computing: Methodological survey on different types of algorithm," 2017 International Conference on Trends in Electronics and Informatics (ICEI), Tirunelveli, India, 2017, pp. 100-107, doi:10.1109/ICOEI.2017.8300865.

[5] Zerguit, Youssef Youness, Hammoudi Idrissi, Idriss Derrhi, Mostafa. (2023). A RELAY-BASED AUTOMATIC BALANCING SYSTEM FOR THREE-PHASE LOADS IN INDUSTRY 4.0. Journal of Theoretical and Applied Information Technology. 101. 2012-2026.

[6] M. H. Kashani and E. Mahdipour, "Load Balancing Algorithms in Fog Computing," in IEEE Transactions on Services Computing, vol. 16, no. 2, pp. 1505-1521, 1 March-April 2023, doi: 10.1109/TSC.2022.3174475.

[7] M. Collotta, et al., "Dynamic load balancing techniques for flexible wireless industrial networks", Proc. IEEE IECON 2010, pp. 1329-1334, Glendale, USA, 2010.

[8]] U. Ashraf, S. Abdellatif, G. Juanole "Gateway Selection in Backbone Wireless Mesh Network", Proc. WCNC, pp. 1-6, Budapest, Hungary, 2009.

[9] C. Basaran, et al., "Hop-by-hop congestion control and load balancing in wireless sensor networks", Proc. IEEE 35th LCN, pp. 448-455, Denver, USA, 2010.

[10] E. Barros, M. Peixoto, D. Leite, B. Batista and B. Kuehne, "A Fog Model for Dynamic Load Flow Analysis in Smart Grids," in 2018 IEEE Symposium on Computers and Communications (ISCC), Natal, Brazil, 2018 pp. 1-6.

[11] R. Beraldi et al., "Randomized load balancing under loosely correlated state information in fog computing," in Proc. 23rd Int. ACM Conf. Model. Anal. Simul. Wireless Mobile Syst., 2020, pp. 123–127

[12] S. P. Singh, A. Sharma, and R. Kumar, "Design and exploration of load balancers for fog computing using fuzzy logic," Simul. Modell. Pract. Theory, vol. 101, 2020, Art. no. 102017.

[13] J. Yang, "Low-latency cloud-fog network architecture and its load balancing strategy for medical big data," J. Ambient Intell. Humanized Comput., 2020. Ed., 2nd ed. New York, NY, USA: McGraw-Hill, 1964, pp. 15-64. [Online]. Available: http://www.bookref.com.

[14] J. Gozalvez, M. Sepulcre, J.A. Palazón, "On the feasibility to deploy mobile industrial applications using wireless communications", Computers in Industry, vol. 65, no. 8, pp. 1136-1146, Oct. 2014.

[15] Ketshabetswe, Lucia Adamu, Murtala Mangwala, M. Chuma, Joseph Sigweni, Boyce. (2019). Communication protocols for wireless sensor networks: A survey and comparison. Heliyon. Heliyon 5. 1-43. 10.1016/j.heliyon.2019.e01591.

Saikat Mondal is a 2nd year student of Indian Institute of Technology Varanasi(BHU), currently pursing a B.Tech program in Computer Science and Engineering.

Anand Prakash Rawal received the B.E degree in Information Technology from Rajiv Gandhi Proudyogiki Vishwavidyalaya, Bhopal, India n 2006. He received M.Tech degree in Information Security from Maulana Azad National Institute of Technology, Bhopal, India in 2008. He is working as an Assistant Professor at Guru Ghasidas University(A central University) in Bilaspur, India. He is Ph.D research scholar at Indian Institute of Technology in the Department of Computer Science Engineering, BHU, Varanasi, India. His research interests are in the area of Wireless Sensor Networks and IoT.

Prasenjit Chanak received the M.Tech and Ph.D degree in Information Technology from Indian Institute of Engineering Science and Technology (IIEST), Shibpur, India in 2011 and 2016 respectively. He is currently an Assistant Professor at Indian Institute of Technology (BHU), Varanasi, India. He also served as an Assistant Professor in the Department of Information and Communications Technology at Atal Bihari Vajpayee- Indian Institute of Information Technology and Management, Gwalior, from December 2016 to August 2020. He has executed various Government of India-sponsored projects from DST, ICSSR and SERB. He has published more than 60 research papers in different reputed journals and conferences. He has published 5 Indian patents. His research interests are Wireless Sensor Networks, Internet of Things(IoT), Cyber-Physical Networks(CPN), Machine Learning, Consumer electronics systems.