OPEN ACCESS

International Journal of Innovative Technology and Exploring Engineering (IJITEE)
ISSN: 2278-3075 (Online), Volume-9 Issue-2, December 2019

Adaptive Data Transmission Optimization in Internet of Things

Saniya Zahoor, Roohie Naaz Mir

Abstract: ***In most of the IoT applications, exchange of data among various physical and virtual IoT devices having different data flows, energy and delay constraints is a challenging task in such environments. This imposes constraints in IoT applications at the node, network and application level, and to meet such constraints, we propose an adaptive IoT system that adapts to different data flows in IoT network having different time and energy constraints. The proposed scheme consists of two algorithms viz., coarse grain transmission path algorithm for low-deadline IoT applications, where time, traffic load and energy consumption are considered as the main parameters; and a fine-grain algorithm for high-deadline situations, where low latency and power constraints are the important performance parameters. Finally, the performance of proposed strategy is evaluated by simulation. The results of the proposed scheme in this paper outperform the existing algorithms in terms of energy, power, number of alive nodes and delay. The proposed scheme is used for data transmission optimization in delay-sensitive resource-constrained IoT applications.***

Keywords: ***Coarse-grain transmission, Data aggregation, Edge, Fine-grain transmission, Internet of Things.***

I. INTRODUCTION

Over the years the Internet of Things (IoT) has gradually developed to cater many fields of applications. IoT comprises of large number of nodes capable of sensing data, deployed in a large geographical area and edge devices that can communicate with the IoT nodes over wireless communication link. The IoT nodes are placed in number of ways in the network viz., single-hop, multi-hop or mesh or grid topology. There are constraints in the IoT ecosystem at the node, network and application level, but network lifetime is the main constraint in IoT networks. In resource constrained IoT applications, once the network is deployed, it is undesirable to replace or recharge IoT nodes. In many applications, the replacement of batteries is impracticable. The solution for energy efficiency at energy techniques only offers a partial solution, therefore, attention has to be paid at protocol level as well, where data is considered as one of the main parameters that can be managed to conserve energy in such systems.

Revised Manuscript Received on December 30, 2019.
***** Correspondence Author

Saniya Zahoor*, Department of Computer Science and Engineering, NIT Srinagar, India, Email: saniyazahoor@nitsri.net

Roohie Naaz Mir, Department of Computer Science and Engineering, NIT Srinagar, India

In IoT applications, more energy is consumed due to transmission of data as compared to the local processing of data, therefore the data to be transmitted needs to be minimized. To design an energy efficient IoT, the main emphasis should be on data aggregation techniques. In delay-sensitive energy-constrained IoT applications, in addition to power requirements, there are latency constraints that need to be met. We have identified scenarios in IoT applications based on power and delay constraints and these include: power and delay constraints, power constraint and no delay constraint, no power constraint and delay constraint, and no power and no delay constraints.

In the low-deadline situations where time, traffic load and energy consumption are important design parameters, a coarse grain transmission path algorithm is used for meeting such constraints in IoT. In high-deadline situation lwhere we have low latency and power constraints, a fine-grain algorithm is used for meeting such requirements. In this paper, we have improved algorithms presented in [1] and proposed an adaptive routing mechanisms based on path difference degree so as to adapt the different data flows in IoT with different time and energy constraints. Comparative analysis of these algorithms shows that our proposed algorithm gives better results as compared to the algorithms proposed in [1]. The organization of paper is as follows, Section II discusses the literature survey, Section III presents the data flows in IoT data aggregation, Section IV illustrates the proposed system, Section V presents the mathematical proofs, Section VI presents the simulation results and Section VII presents the conclusions.

II. LITERATURE SURVEY

Internet of Things embodies a vision of merging heterogeneous objects to establish seamless interactions among various logical and physical devices. The logical and physical devices in IoT are resource constrained in terms of processing, storage and bandwidth. Such an environment poses additional challenges to the miniature and unattended IoT devices deployed in IoT applications. There are numerous constraints in resources which need to be taken care of both at the software and hardware level. In addition to the resource constraints, there are many limitations due to characteristics of an individual node as well, (e.g., limited energy), the behavior of the network (e.g., topology change) and constraints at the application level (e.g., latency).

Further, these IoT devices are typically battery operated; as such the one-time deployment can further impose constrains on IoT application especially in terms of energy.

The problem of energy consumption in IoT is a grave issue; therefore an efficient energy management is required.

Most of the energy consumption in such resource constrained IoT devices occurs because of the RF trans-receivers and flash memory components [2]. The local processing consumes less energy, than transmitting the data for remote processing on edge or some other IoT node [3]. In IoT applications that are delay and energy sensitive, it becomes a challenge to transmit the data to the sink with low delay and energy savings. To address this, research has been carried out to address packet delivery ratio and delay[4][5], while others address energy and packet error rates [6],[7].

There has been research in data gathering schemes to address the delay and lifetime of the sensor nodes in network. The data-centric approach has two main advantages i.e. minimum communication overhead and efficient in-network processing. In in-network processing, the content moving through the network is identifiable by intermediate nodes, resulting in increased resource consumption which can be managed by efficient data aggregation and compression techniques e.g., threshold-sensitive energy-efficient sensor network protocol (TEEN) [8]. Several researchers have studied the combination of information gathering in a WSN by combining routing with in-network compression [9]. The type of compression can only be application specific such as LEACH protocol [10], distributed source routing [11], routing and compression approaches used in prediction based monitoring [12] and distributed regression framework for model-based compression [13].

Predominant work has also been done on comparative analysis of data routing algorithms based on performance metrics such as network lifetime, robustness, security, delay, etc [14][15][16]. The growing research on Internet of things show that clustered network has advantages in terms of resource conservation. That is why the hierarchical structure is widely adopted in IoT that emphasize on real time application requirements. On the other hand, due to huge data sensing, the communication traffic of IoT applications are increasingly growing [17] [18].

Data transmission path plays an important role in IoT systems that determine how the data is to be communicated among the IoT devices. Optimizing transmission of data along the communication path provides an efficient way of data transmission in wide area IoT networks. Various optimization techniques have been adopted viz., Ant optimization [19], data transmission for high reliability in IoT environments [20], etc. Various Edge based IoT solutions have been adopted to provide fast computing on low-resource IoT devices. Authors in [21] presents an edge based solution to enhance energy conservation at IoT node level via optimal data transmission in IoT network. This paper presents coarse-grain and fine-grain data transmission algorithms for low-deadline and high-deadline situations in IoT.

III. DATA FLOWS IN IOT DATA AGGREGATION

Internet of Things is the network of physical devices which monito the physical world. Connectivity, sensing, and interactivity among devices are considered as the main features of the IoT. Due to the large-scale deployment of distributive and pervasive IoT nodes, there has been explosive development of redundant data in such applications. Further, the smaller sized IoT nodes are typically resource constrained, powered by limited batteries, storage, processing and communication capabilities. To optimize the use of limited resources in such IoT applications, various data aggregation mechanisms are used that aim to route the data via minimal resource consumption routes. Therefore, the purpose of data aggregation is to aggregate and collect the data packets in an effective manner in order to optimize the use of resources.

As shown in Fig.1, for data aggregation in an IoT environment, deployed IoT nodes collect data by sensing the environment; the aggregated data is forwarded to the edge node. The sensed information is carried to the edge node either directly or in hops via intermediate IoT nodes. In data intensive IoT applications which require periodic monitoring of surrounding environment, it is possible that an intermediate node receives redundant data from its child IoT nodes. Populating such sensed information has an impact on consumption of resources especially energy. To avoid this, data aggregation mechanisms can be employed in which the intermediate IoT nodes can forward only appropriate data to the edge node rather than redundant values, therefore enhancing optimization of resource utilization in IoT applications. Data aggregation mechanism offers several benefits to resource constrained IoT applications such as: improving the efficiency and accuracy of information, eliminating the unnecessary redundant information, minimizing the traffic load, saving energy of the nodes, optimizing the storage utilization in memory constrained IoT nodes, optimizing the processor utilization in IoT nodes, etc.

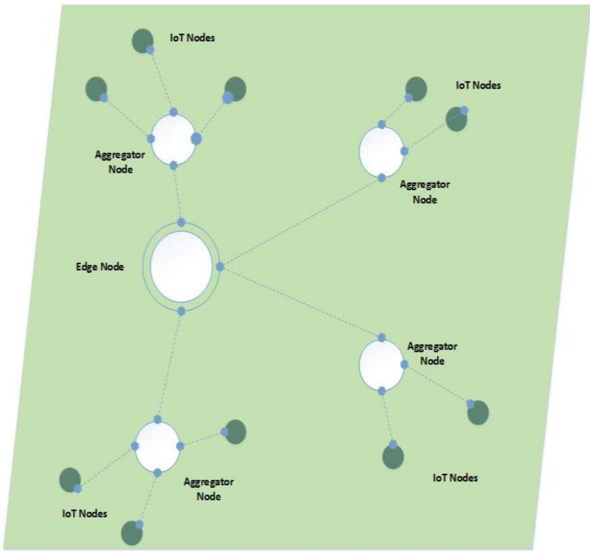

Fig.1. Data Aggregation mechanism

In resource constrained IoT applications, nodes have limited energy. Therefore, devising the energy conserving solutions for such applications becomes necessary; clustering in data aggregation schemes provides the essential candidate solutions. To ensure resource efficiency in such networks, clustering has become an emerging mechanism for building robust and energy efficient IoT environment.

Retrieval Number: B7859129219/2019©BEIESP
DOI: 10.35940/ijitee.B7859.129219
Journal Website: www.ijitee.org

Published By:
Blue Eyes Intelligence Engineering & Sciences Publication

In clustering, the entire network is separated into various clusters; each cluster consists of IoT nodes wherein an aggregator IoT node forwards the aggregated data to the edge node.

In such edge based clustering in IoT networks, edge nodes are placed at the network edge to bring the computing resources closer to the resource constrained IoT nodes, and data aggregation is performed in parallel at the edge nodes. To minimize the amount of aggregated data from the IoT nodes and push the burden from the resource-constrained IoT nodes, adaptive data aggregation approach can also be used as an efficient solution in such networks where resource consumption can occur at a higher speed.

In edge based IoT applications, nodes are usually heterogeneous in hardware, there may be a node(s) capable of processing the data, such set of nodes whose resources are sufficient are referred to as high end nodes e.g., edge nodes. The number of edge nodes depends on the number of clusters formed and size of data processed. Typically, data aggregation approaches for cluster based edge-IoT networks can be classified into two: (a) aggregator node directly sends data to the edge node (Direct transmission), and (b) the aggregator node sends data to the edge node via other node in multiple hops (Indirect transmission) as shown in Fig. 2 and Fig. 3.

In direct transmission, each aggregator sends the sensed data from its IoT nodes to the sink. As Fig. 2 shows, in clusters C_1, C_2 *and* C_3, active IoT nodes I_1, I_2, I_3, I_4, and I_5 sends the sensed data to the respective aggregator nodes A_1, A_2 and A_3 , which then sends the aggregated data to the edge node E1. We can see that there are five data flows of sensed data from the active IoT nodes. A_1 aggregates three data flows into one. As a result, there are three data flows from the aggregator nodes to the edge node. This approach proves to be an inefficient data aggregation approach because of multiple redundant data transmissions.

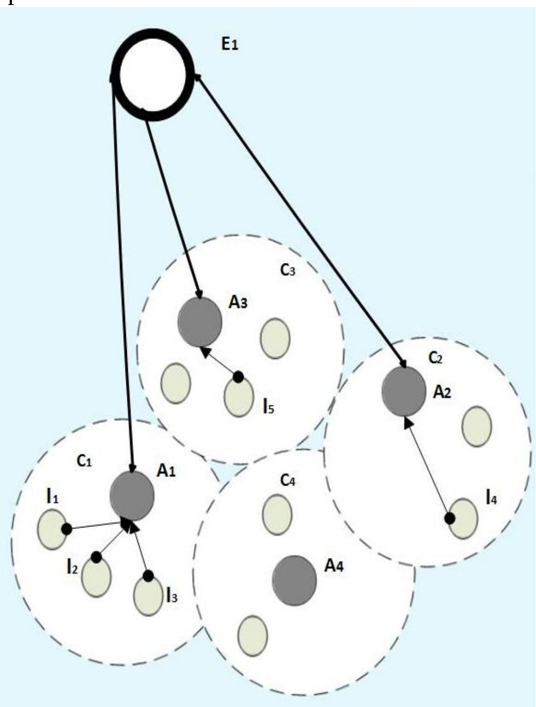

Fig. 2. Data aggregation via direct transmission

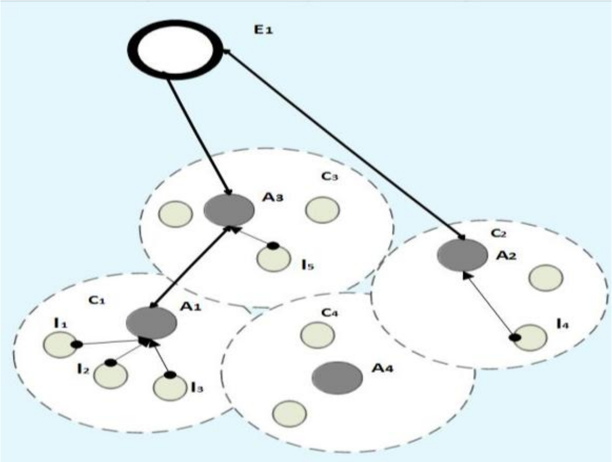

Fig. 3. Data aggregation via indirect transmission

While in indirect transmission, a hierarchical data aggregation model is used. According to this hierarchy, the data aggregated on an aggregator node will further be aggregated on another aggregator node at a higher level. Therefore, the data can be aggregated hop by hop through multiple intermediate nodes to the edge node. As Fig. 3 shows, in clusters C_1, C_2 and C_3, active IoT nodes I_1, I_2, I_3, I_4, and I_5 sends the sensed data to the respective aggregator nodes A_1, A_2 and A_3. A_3 further aggregates the data flow from aggregator node A_1 and IoT node I_5 into one data flow to the edge node E1. Therefore, data aggregation via indirect transmission allows for reduction in redundant data transmissions, compared to data aggregation via direct transmission. Efficient data aggregation algorithms are required to accommodate different data flows in IoT environment.

IV. PROPOSED SYSTEM

The proposed system consists of aggregator nodes that collect the sensed data from IoT nodes and push the data to the corresponding cluster head. The nodes and cluster heads have a many to one relation. We assume the cluster heads have a higher configuration as compared to IoT nodes. After the data to be transmitted is collected by a cluster head, hop to hop delivery is started until data reaches the destination cluster head. The last cluster head then forwards the data to the sink. The entire network (lnter-cluster/Intra-cluster connections) is controlled by a SDN (software Defined Networks) controller which adjusts the network in real time. Edge computing is utilized by the cluster heads to reduce computational load on the SDN controller. Once a path has been set by the SDN controller, the path remains the same for the entire data transmission duration. For the next data transmission, the path is recalculated. Further storage optimizations is performed by storing all results in a hash table and using them when needed.

A. Assumptions

Following are the assumptions made.

- Consider the network to be made up of nodes and edges similar to a graph.

- Let R the communication range of nodes in the case of low real-time performance.
- Let C be the set of all cluster heads where C_i, C_j, C_k ,…, C_n are its members.
- A pair of cluster heads C_i, C_j are said to be connected if the distance between them is less than R.
- Let T_{Rcc} be the rate of communication between cluster heads.
- Let T_{Rcn} be the rate of communication between cluster heads and IoT nodes.
- Each node is connected to the nearest cluster head.
- Let T_{Rij} be the transmission rate between network members i and j.
- Let V_i be the node of network
- Let C_i be the Cluster head of network
- Let $Path\_ALL_{CH}$ be the set of all possible paths between cluster heads
- Let T be the time taken for transmission
- Let T_c be the time constraint in which message needs to be sent
- Let $Path_{ij}$ be the path that has already been selected
- Let $CP(vi, vj)$be the pair of cluster heads closest to V_i and V_j
- Let $D(C_i, C_j)$be the distance calculated between Cluster head i and cluster head j
- Let Path_Cost be the sum of the weights of all the edges along a path

B. *Definitions*

Following definitions are used in proposed algorithms:

- *Edge Weight:* The connection between the network members is of two types: Cluster head to cluster head and Cluster head to aggregator node. The higher the weight of an edge, better the communication.
- *Loss_Probability:* A running average of packet loss percentage is calculated using a sliding window approach i.e., percentage of last 100 transmission that were successful. This value is multiplied for every node along a path to get a path loss probability value.
- *Path_Energy: Path_Energy* is defined as the ratio of the current available energy in all the nodes along a path to the initial available energy in all the nodes.
- Power_Cost: As the algorithm decides to use a cluster head instead of a regular node for transmission, the total energy needed for transmission increases. Path energy cost is the difference between the final cost of the path with increased number of cluster heads and the initial cost.
- *Average Time Delay:* The average time required for transmission of data source IoT node to the destination IoT node.
- *Path Difference Degree (PDD):* It measures the balance of transmission path. The energy consumption and load are dependent on this performance metrics.

C. Proposed Algorithms

In this section, we have proposed two algorithms viz., coarse grain optimal path algorithm for low deadline situation and adaptive transmission algorithm for power optimization for high deadline situations. Comparison of these algorithms has been done with the algorithms outlined in [1]. The improved algorithm 1 and Algorithm 2 are shown in Table I (a) and (b). In Table I(a), the decision parameters α, β and ¥ decides which path to use, and are set according to the application constraints. For IoT applications where the network doesn't need load balancing, α is set low. For applications where the node to node packet loss is high, β is set low to reduce the likelihood of a packet drop. And for networks where the Path energy is low, ¥ is set low to reduce the likelihood of the path being selected.

Table – I (a): Improved Algorithm 1: Pseudocode of coarse grain optimal path for delay optimization

Require: vi, vj, C, Path_AllCH, TC, TR, W
Ensure: Path_selection V<vi,vj>,Pathij , 0 0
From the set C, search the cluster head pair (Ci, Cj)
Construct the communication over cluster head pairs corresponding CHPair< Ci,Cj> of vi and vj
Select all paths of Pathij between cluster head Ci and Cj, from Path_AllCH, according with W
For i = 0 to |Pathij| do
Calculating every path time
If the time taken for path, T is less than latency requirements, TC
Add path to set of paths under consideration
Endif
Endfor
For j = 0 to |PathSS|
Calculate path difference degree W j
Calculate Loss_Prob
Calculate Path_Energy
*Calculate Wj * α + Loss_Prob* β + Path_Energy * ¥*
Find the maximum value for path_Cost
Endfor
Return Max_Path

Table – I (b) : Improved Algorithm 2: Pseudocode of adaptive transmission for power Optimization

Require: v_i, v_j, $Path_{ij}$, T_c, T_{min}
Ensure: Path_new
For i = 0 to |Path_all |
Construct the communication over all cluster head pair corresponding of vi and vj, stored in $CP(v_i,vj)$ *from the head for* $Path_{ij}$
For m=0 to |CP|
Calculate distance of cluster head pair, d(cm,cm+1)=|cm,cm+1|
If $R < d_{(cm,cm+1)} < R_{max}$
$P_{t(cm)} = P(d_{(cm,cm+1)})$ *//increase the current transmission power*
$T_{Rm,m+1} = T_{RC}$ *// update the transmission rate*
End if
Calculate the Tm
If a new path, $T_i\ T_C\ T_{min}$ *meets the application requirement*
NewPathSet ⟵ Path_all[m]
Endif
End or
Endfor
For k = 0 to |NewPathSet|
Calculate a new power changed cost, C_{pk}
Calculate Path_Loss, Path_Energy, Path_Cost
If $\alpha * C_{PK} + \beta$ ** Path_Loss+ ¥* Path_Eenrgy*
Path_new ← NewpathSet[k]
Endif
Endfor
Return Path_New

In Table I (b), the decision parameters are set according to requirements, but here they serve a different purpose.

For IoT applications where the network has strict energy constraints, α is set low, to reduce the likelihood of high *Path_Cost*, for networks where the node to node packet loss is high, , β is set low to reduce the likelihood of a packet drop and vice versa. And for networks where the *Path_Energy* is low, ¥ is set low to reduce the likelihood of the path being selected.

V. MATHEMATICAL FORMULATION

In aggregation of data from IoT nodes to aggregator to sink, we have improved on various parameters such as power, failure rate, packet loss and robustness in coarse grain algorithm stated in [1]. The improvements are discussed as under:

- *Power Improvement:* Power is the ratio of total energy of path if all nodes were full and actual energy present in all nodes along the path. In calculation of time taken in simulation, multiplying the time by a calculated constant increases the likelihood of a path having more energy left in its cells being used. This helps with load balancing and increasing network lifetime.
- *Failure rate Improvement:* Failure rate is inversely proportional to failure percentage of nodes along the path. Multiplying the time by a calculated constant increases the likelihood of a path having lesser failure rate along its nodes being used. This also ensures that shorter paths are selected, to improve network transmission times.

  The above improvements can be combined to give improved power consumption and failure rates.
- *Packet loss improvement:* Since paths are calculated according to the time constraint which takes into account weight of edges, using Lemma 1 and Lemma 2, we can show that reweighing the edges favors paths with lesser packet loss.

  Lemma 1: The edges are reweighed to favor edges with lesser packet loss, given as:

$$W = W * |packets - sent| \; |packets - received| \quad (1)$$

Between a pair of nodes, the number of packets sent is a constant amount so the above equation reduces to:

$$W = |packets\ constant - received| \quad (2)$$

The weight becomes inversely proportional to the packets received.

QED

Lemma 2: The path is made up of edges with lesser packet loss. While calculating all possible paths, paths with higher weights than a certain constraint are rejected. And if the weight of an edge is less, the path which includes this edge will have less total weight. QED

The power is also improved and can be proved in a similar way as Lemma 2. These two reweighing schemes can be combined to give an algorithm with lesser packet loss and power improvement

- *Robustness:* In addition to packet loss and power improvements, the proposed modified algorithm 1 ensures robustness as supported by Lemma 3.

Lemma3: More the number of nodes / hops in path, more is the expected value of a failure occurring.

According to the algorithm if message is sent along one path, then any node can be a source of single point of failure. Consider the set of all such nodes in the selected path, let that set be Si. We assume sending the message along multiple paths; therefore, single point of failure can only occur at nodes which are common to all paths. Let the set of common nodes to those paths be {S1} ∩{S2}∩{S3}, ... , ∩ {Si}. It is clear that the size of the second set is bounded by the set with the least number of elements, which implies:

$$N_{fi} \leq N_{fo} \quad (3)$$

Where N_{fi} is the number of sources of failure in improved algorithm and N_{fo} is the number of sources of failure in algorithm 1 in [1] .

We have improved on parameters like power, failure rate, and time delay in algorithm 2 stated in [1]. Power and failure improvements can be stated and proven in the same way as for improved algorithm 1. However, the mathematical proof for improvement in time delay is stated in lemma 4:

Lemma 4: An edge is included between cluster head pairs only if it improves the time constraints of the path. If the statement holds for every edge included in path, the path will always be better than a path that doesn't include the intermediate cluster head pairs. Therefore the path is at least as good as a path chosen otherwise by algorithm 2 of [1].

VI. SIMULATION SETUP AND RESULT ANALYSIS

The proposed graph structure are programmed and simulated in C++. The simulations are run using omnet++. The simulation parameters are set as shown in Table II.

A. Performance Metrics

Following is the performance metrics used:

- *Network area versus rounds:* It is assumed that every node contributes area equal to P_I * R*R overlap of node areas is counted twice. It is calculated three different times for three different location schemes.
- *Energy per round:* Total energy is calculated and divided by number of rounds.
- *Packet delivery ratio:* It is assumed that every cluster head has a failure rate between 0% and 20%. It is assumed that every node has a failure rate between 0% and 30%. Any packet which has < 60 % failure rate is considered as failed.
- *Rounds versus Alive Nodes:* Every node is given 20j, one transmission costs 1j.

B. Result Analysis

The performance of improved algorithm 1 is evaluated in terms of round number versus number of alive nodes, network area versus number of rounds, power/time versus number of rounds and rounds versus energy/alive node.

Table-II :Simulation Parameters

Parameters	Value
Simulation Area	150 * 150 square meter
Data aggregators	50
Cluster Heads	10
Maximum time threshold	500 seconds
Cluster to cluster transmission of unit data	1 second
Cluster to data aggregator transmission of unit data	100 seconds
1 transmission of data cost	1 joule
Initial Node Energy	20 joules

Fig. 4 shows energy left per node versus number of rounds. It is evident from the graph that the energy of a node is more in improved algorithm 1 as compared to algorithm 1 stated in [1]. Fig 5 shows power per node versus number of rounds. It is evident from the graph that the power of a node is more in improved algorithm 1 as compared to algorithm 1 stated in [1]. Fig. 6 shows the nodes in improved algorithm 1 lasts for a longer time as compared to algorithm 1 stated in [1], but dies at approximately same time of simulation. Fig. 7 shows average energy of a node across simulations. With different initial energy taken for different simulations, the energy of a node is analyzed after 20 rounds, and it is evident that the energy of a node in algorithm 1 of [1] is less than the energy of a node in our improved algorithm 1. Fig. 8 shows average power of a node across simulations. With different initial power taken for different simulations, the power of a node is analyzed after 20 rounds. it is evident that the power of a node in algorithm 1 of [1] is less than the power of a node in our improved algorithm 1.

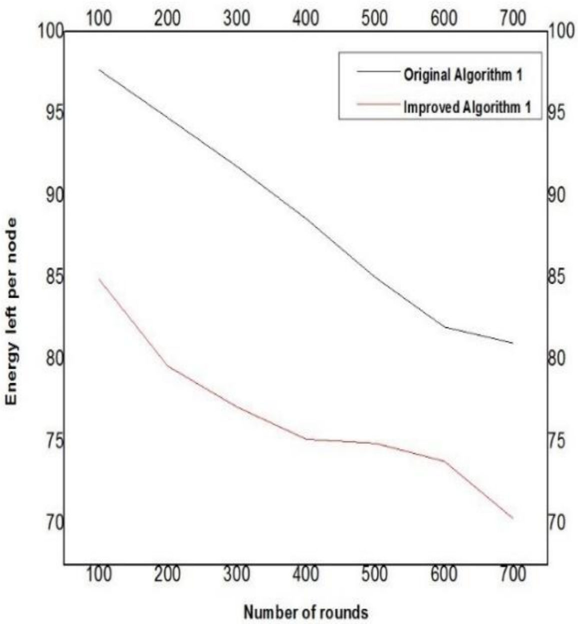

Fig. 4. Energy left per node versus Number of Rounds

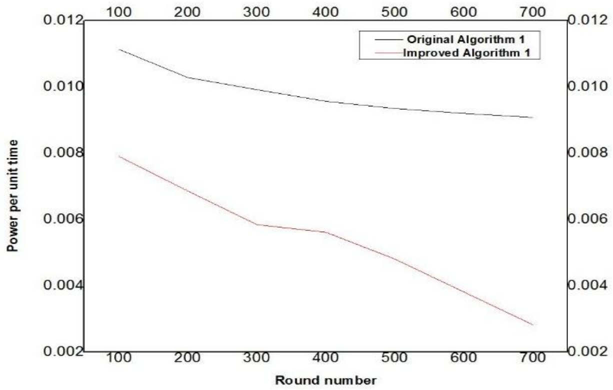

Fig. 5. Power per unit time versus Round number

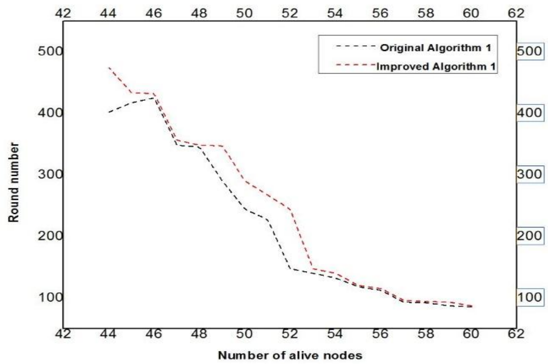

Fig 6. Number of alive nodes versus Number of Round

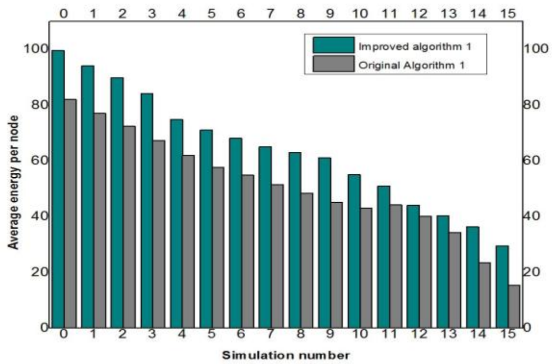

Fig 7. Average energy per node across simulations

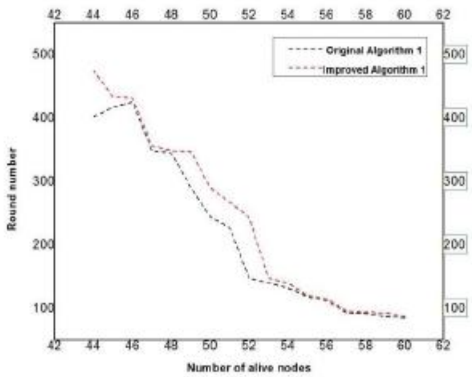

Fig 6. Number of alive nodes versus Number of Round

Retrieval Number: B7859129219/2019©BEIESP
DOI: 10.35940/ijitee.B7859.129219
Journal Website: www.ijitee.org

Published By:
Blue Eyes Intelligence Engineering & Sciences Publication

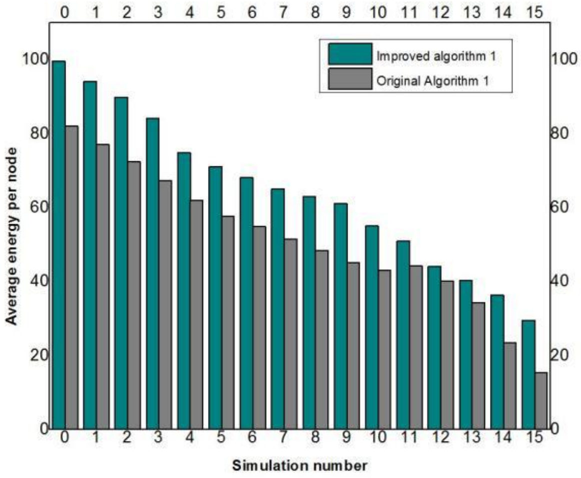

Fig 7. Average energy per node across simulations

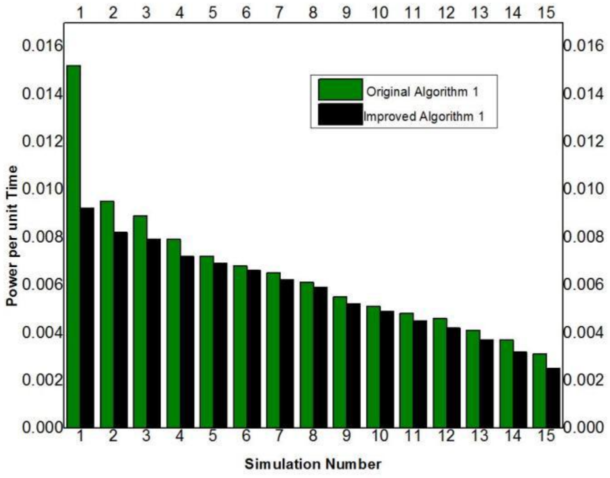

Fig 8. Power per unit time across simulations

The improved algorithm 2 is only designed for fast network transmission times; we have considered two different types of graphs - dense graphs and sparse graphs, each with power constraint and no power constraint. Results are shown in Table III to Table VI.

Table - III: Dense graph, with no power

Simulation	Time taken in improved algorithm 2	Time taken in algorithm 2 of [1]	Cluster Head time in improved algorithm 2	Cluster head time in algorithm 2 of [1]	Cluster heads in path in improved algorithm 2	Cluster heads in algorithm 2 of [1]
0	1.91842E+06	1.93076E+06	29907	42254	20417	32764
1	1.91766E+06	1.93294E+06	30141	45423	20656	35938
2	1.92286E+06	1.93464E+06	29971	41747	20459	32235
3	1.9205E+06	1.93297E+06	31398	43862	21905	34369
4	1.92504E+06	1.93914E+06	30754	44859	21235	35340
5	1.91816E+06	1.93159E+06	31636	45071	22156	35591
6	1.92219E+06	1.93634E+06	30890	45040	21386	35536
7	1.91845E+06	1.93012E+06	29742	41409	20251	31918
8	1.91968E+06	1.93326E+06	27988	41568	18482	32062
9	1.92355E+06	1.93688E+06	28271	41609	18747	32085
10	1.91829E+06	1.93313E+06	29778	44619	20288	35129

Table- IV: Dense graph, with power constraint

Simulation	Time taken in improved algorithm 2	Time taken in algorithm 2 of [1]	Cluster Head time in improved algorithm 2	Cluster head time in algorithm 2 of [1]	Cluster heads in path in improved algorithm 2	Cluster heads in algorithm 2 of [1]
0	1.0742E+06	1.08105E+06	17311	24165	12000	18854
1	1.14256E+06	1.15169E+06	17608	26740	11955	21087
2	1.1155E+06	1.12216E+06	18816	25468	13305	19957
3	1.05648E+06	1.06458E+06	16706	24802	11481	19577
4	1.0704E+06	1.07902E+06	16494	25117	11198	19821
5	1.05659E+06	1.06397E+06	17014	24396	11790	19172
6	1.19939E+06	1.20807E+06	19516	28200	13587	22271
7	1.1163E+06	1.12326E+06	18416	25376	12899	19859
8	1.23671E+06	1.24622E+06	19028	28540	12909	22421
9	1.19807E+06	1.20707E+06	18595	27595	12668	21668
10	1.12638E+06	1.13528E+06	17352	26251	11779	20678

Table – V: Sparse graph, with no power constraint

Simulation	Time taken in improved algorithm 2	Time taken in algorithm 2 of [1]	Cluster Head time in improved algorithm 2	Cluster head time in algorithm 2 of [1]	Cluster heads in path in improved algorithm 2	Cluster heads in algorithm 2 of [1]
0	1.60234E+06	1.6086E+06	30642	36894	22744	28996
1	935848	941409	16667	22228	12048	17609
2	1.58618E+06	1.59169E+06	31789	37299	23978	29488
3	1.35081E+06	1.35702E+06	26071	32281	19414	25624
4	1.22757E+06	1.23326E+06	23425	29110	17374	23059
5	1.0499E+06	1.05393E+06	20670	24698	15498	19526
6	1.73318E+06	1.73885E+06	35709	41377	27179	32847
7	1.69251E+06	1.69802E+06	33045	38564	24706	30225
8	1.73953E+06	1.74622E+06	33898	40596	25327	32025
9	1.63861E+06	1.64508E+06	32480	38949	24409	30878
10	1.39216E+06	1.39794E+06	27417	33197	20559	26339

Table – VI: Sparse graph, with power constraint

Simulation	Time taken in improved algorithm 2	Time taken in algorithm 2 of [1]	Cluster Head time in improved algorithm 2	Cluster head time in algorithm 2 of [1]	Cluster heads in path in improved algorithm 2	Cluster heads in algorithm 2 of [1]
0	740817	743904	13074	16161	9417	12504
1	857995	862569	15429	20003	11195	15769
2	782055	784592	14910	17447	11055	13592
3	762371	766088	13534	17251	9771	13488
4	823862	827930	14330	18398	10262	14330
5	728802	731089	13795	16082	10202	12489
6	869430	871382	17511	19463	13230	15182
7	848171	851797	15754	19380	11571	15197
8	938366	941436	18190	21260	13566	16636
9	804449	807325	15414	18290	11449	14325
10	809693	812900	14688	17895	10693	13900

VII. CONCLUSIONS

In IoT applications, the communication among the IoT nodes is an important aspect in determining the network topology. Clustered topology are considered as manageable structures for IoT. But these frameworks are constrained in terms of latency, bandwidth, coverage, and unbalanced deployment of computing resources. In order to manage these issues, Software Defined Networks and Edge computing are integrated into IoT to constitute our proposed framework. The IoT system should provide an effective connectivity and control among various physical and virtual IoT devices. An efficient data transmission scheme is a need of the hour for most of the IoT systems that are energy constrained. The different delay constraints in data flows is a challenging issue in such systems. To optimize the performance of data transmission, we propose an adaptive IoT system that adapts different data flows in IoT meeting various application requirements. The proposed work consists of coarse grain transmission path algorithm and a fine-grain algorithm for low-deadline and high-deadline IoT applications.

The proposed work is validated by simulations. The simulation results are supported by mathematical proof of the proposed algorithms used in the work. The proposed scheme provides improved solution for different data transmissions in Edge-IoT applications with delay constraint or no delay constraint and with power constraint or no power constraint scenarios.

REFERENCES

1. Li, X., Li, D., Wan, J., Liu, C. and Imran, M., 2018. Adaptive transmission optimization in SDN-based industrial Internet of Things with edge computing. IEEE Internet of Things Journal, 5(3), pp.1351-1360.
2. Forster, A., 2016. Introduction to wireless sensor networks. John Wiley & Sons.

3. Mukherjee, N., Neogy, S. and Roy, S., 2015. Building Wireless Sensor Networks: Theoretical and Practical Perspectives. CRC Press.
4. Deng, X., He, L., Li, X., Liu, Q., Cai, L. and Chen, Z., 2016. A reliable QoS-aware routing scheme for neighbor area network in smart grid. Peer-to-Peer Networking and Applications, 9(4), pp.616-627.
5. Parvez, I., Jamei, M., Sundararajan, A. and Sarwat, A.I., 2014, December. RSS based loop-free compass routing protocol for data communication in advanced metering infrastructure (AMI) of Smart Grid. In 2014 IEEE Symposium on Computational Intelligence Applications in Smart Grid (CIASG) (pp. 1-6). IEEE.
6. Shah, G.A. and Akan, O.B., 2013, June. Spectrum-aware cluster-based routing for cognitive radio sensor networks. In 2013 IEEE International Conference on Communications (ICC) (pp. 2885-2889). IEEE.
7. Fadel, E., Faheem, M., Gungor, V.C., Nassef, L., Akkari, N., Malik, M.G.A., Almasri, S. and Akyildiz, I.F., 2017. Spectrum-aware bio-inspired routing in cognitive radio sensor networks for smart grid applications. Computer Communications, 101, pp.106-120.
8. Sharma, S. and Jena, S.K., 2011, February. A survey on secure hierarchical routing protocols in wireless sensor networks. In Proceedings of the 2011 international conference on communication, computing & security (pp. 146-151). ACM.
9. Scaglione, A. and Servetto, S., 2005. On the interdependence of routing and data compression in multi-hop sensor networks. Wireless Networks, 11(1-2), pp.149-160.
10. Heinzelman, W.R., Chandrakasan, A. and Balakrishnan, H., 2000, January. Energy-efficient communication protocol for wireless microsensor networks. In Proceedings of the 33rd annual Hawaii international conference on system sciences(pp. 10-pp). IEEE.
11. Cristescu, R., Beferull-Lozano, B. and Vetterli, M., 2004, March. On network correlated data gathering. In IEEE INFOCOM 2004 (Vol. 4, pp. 2571-2582). IEEE.
12. Goel, S. and Imielinski, T., 2001. Prediction-based monitoring in sensor networks: taking lessons from MPEG. ACM SIGCOMM Computer Communication Review, 31(5), pp.82-98.
13. Guestrin, C., Bodik, P., Thibaux, R., Paskin, M. and Madden, S., 2004, April. Distributed regression: an efficient framework for modeling sensor network data. In Proceedings of the 3rd international symposium on Information processing in sensor networks (pp. 1-10). ACM.
14. Nasser, N., Arboleda, C. and Liliana, M., 2006, May. Comparison of clustering algorithms and protocols for wireless sensor networks. In 2006 Canadian Conference on Electrical and Computer Engineering (pp. 1787-1792). IEEE.
15. Xu, D. and Gao, J., 2011. Comparison study to hierarchical routing protocols in wireless sensor networks. Procedia Environmental Sciences, 10, pp.595-600.
16. Xu, D. and Gao, J., 2011. Comparison study to hierarchical routing protocols in wireless sensor networks. Procedia Environmental Sciences, 10, pp.595-600.
17. Zhang, Y., 2016. GroRec: a group-centric intelligent recommender system integrating social, mobile and big data technologies. IEEE Transactions on Services Computing, 9(5), pp.786-795.
18. Song, L., Chai, K.K., Chen, Y., Schormans, J., Loo, J. and Vinel, A., 2017. QoS-aware energy-efficient cooperative scheme for cluster-based IoT systems. IEEE Systems Journal, 11(3), pp.1447-1455.
19. Kotagi, V.J., Singh, F. and Murthy, C.S.R., 2017, May. Adaptive load balanced routing in heterogeneous IoT networks. In 2017 IEEE International Conference on Communications Workshops (ICC Workshops) (pp. 589-594). IEEE.
20. Zhu, R., Zhang, X., Liu, X., Shu, W., Mao, T. and Jalaian, B., 2015. ERDT: Energy-efficient reliable decision transmission for intelligent cooperative spectrum sensing in industrial IoT. IEEE Access, 3, pp.2366-2378.
21. Peralta, G., Iglesias-Urkia, M., Barcelo, M., Gomez, R., Moran, A. and Bilbao, J., 2017, May. Fog computing based efficient IoT scheme for the Industry 4.0. In 2017 IEEE International Workshop of Electronics, Control, Measurement, Signals and their Application to Mechatronics (ECMSM) (pp. 1-6). IEEE.

AUTHORS PROFILE

Saniya Zahoor is a research scholar in the Department of Computer Science & Engineering at NIT Srinagar, INDIA. She received her B.E in Computer Science & Engineering from Kashmir University, Kashmir (India) in 2011 and M.E in Computer Engineering from Pune University, Maharashtra (India) in 2015. Her research interests include Internet of Things, Virtualization, and Pervasive Computing.

Roohie Naaz Mir is a professor in the Department of Computer Science & Engineering at NIT Srinagar, INDIA. She received B.E. (Hons) in Electrical Engineering from University of Kashmir (India) in 1985, M.E. in Computer Science & Engineering from IISc Bangalore (India) in 1990 and Ph.D. from University of Kashmir, (India) in 2005. She is a Fellow of IEI and IETE India, senior member of IEEE and a member of IACSIT and IAENG. She is the author of many scientific publications in international journals and conferences. Her current research interests include reconfigurable computing and architecture, mobile and pervasive computing, security and routing in wireless Ad hoc, sensor networks, and Internet of Things.