Service oriented fault monitoring in Internet of Things device management

Prasenjit Maiti
Department of Computer Science
and Engineering
National Institute of Technology
Rourkela,Odisha,India
Email: pmaiti1287@gmail.com

Bibhudatta Sahoo
Department of Computer Science
and Engineering
National Institute of Technology
Rourkela,Odisha,India
Email: bibhudatta.sahoo@gmail.com

Ashok Kumar Turuk
Department of Computer Science
and Engineering
National Institute of Technology
Rourkela,Odisha,India
Email: akturuk@gmail.com

***Abstract*—This paper investigates the issue of device clustering for fault monitoring in Internet of Things(IoT) device management system. To detect the faulty device quickly, fault monitoring must be conducted regularly and often. Therefore, it is desirable to reduce the communication cost for fault monitoring .Fault recovery is also an important factorin device management system. We model our problem as a multidepot fixed destination multiple traveling salesman problem in an integer programming (IP) formulation.**

keywords:- clustering, integer programming, TSP, fault, IOT

I. Introduction

Internet of Things(IoT) is one of the most upcoming technology which has created a great impact in the present technology era. It is a collection of network enabled devices (objects) which are controlled remotely. This IoT(Internet of Things)is nothing but a mixture of wireless sensor network (WSN) , embedded systems [1], [2]. Associating with these devices it has anticipated that the IOT will creat a mass effect in day to day life and also will acceptable in numerous application,for example,health care, smart home, smart car etc [3], [4]. With the help of heterogeneous sensors and embedded technology numerous IoT devices have been composed . The main challenge in IoT is to manage and maintain large number of devices and react smartly according to the data generated by them.

Adaptation to non-critical failure is essential for any IoT application like health care and smart car. Fault tolerance is also an important factor in any IoT application. Device is a collection of heterogeneous sensors where different battery energy and functionality are used. IoT nodes or devices are deployed with multiple multi-purpose sensors, where as the wireless sensor nodes are single-purpose sensors. Each device in IoT system perform a particular task. A service in a IoT system is a subset of IoT sensor devices perform a particular task [5], [6]. For example,all kinds of small appliances, light bulbs, door sensors, and other home products can automatically turn on and off when triggered by certain actions. Smart homes make life more comfortable, and they're typically designed to be power-efficient, which could save you money.IoT system provide a multiple set of services.Service management or device management is a big challenge in IoT.

Minimize the overall network traffic is a critical factor in energy constraint devices [7], [8], [9], [10]. Sen Zhou et al. [11] proposed a device clustering approache for fault monitoring, but they are not consider link failure between devices and one device can connect multiple devices and not consider any special loop(those nodes can't form a loop) . Our objective is to design a fault monitoring mechanism for IoT middleware to reduce communication overhead over a heterogeneous devices or nodes from diverse location in an IoT network to achieve the longest system lifetime. We propose a device clustering mechanism according to the service provided by the IoT system. Here we select a group of sensor device as a cluster which perform a particular service. We propose a framework which makes an energy efficient , less interference paths to reduce communication cost on the same network. We model our problem as a multidepot fixed destination multiple traveling salesman problem and formulate with integer programming. Each depot has one and only one salesman.

The motivation for fault monitoring in IoT is identify faulty devices in a service. The simplest route for fault monitoring is to utilize a central controller. The controller sends surveys to device intermittently to request their status. In the event that any strange conduct is recognized, or if a device doesn't answer, there might be a fault present. The central controller will then consider which service might be at faulty on which device.

But it faces scalability issue due to the number of devices increases in a system or network. IoT system's lifetime also degraded because of more fault monitoring combinations. Message transfer time for fault monitoring also increases. The effect of this communication is waste of energy and resources. Therefore we divide the IoT system into a number of cluster according to the service provided by the devices.Each cluster monitored individually and managed by Cluster Head(CH). Each CH of each cluster keep the backup information of each device. If any faulty device detect by other device in a cluster and report to the CH then CH replace the service with backup ones.

This paper is organized as follows. Specifically, in section II, we present the related work of fault monitoring in IoT devices. Then in section III and section IV , we discuss system model and proposed technique of loop based clustering of devices. We conclude the paper in section V.

II. Related Work

Wireless sensors nodes change their topology due to high mobility.For IoT nodes high mobility is not required. IoT nodes is not high mobility , so stable clustering structure save communication cost. All associating nodes not communicate continuously in IoT system. Most of the nodes following event-driven communication . If any event occurs then it send the changed value to the other nodes of the system. Nonetheless, this additionally implies that if no message is received, an accepting device can't tell whether nothing has happened or the sender has fizzled. Therefore a control packet is send periodically for detecting faulty device. So beacon receiving device know that the sending device is alive. The traditional clustering model LEACH was proposed by W. R. Heinzelman et al. LEACH based on distributed algorithm where CH selected by energy level of neighbors. Many researcher have been proposed other model which are based on LEACH. However, all clustering methods are star based which show themselves with the accompanying issue like number of packet transfer for fault monitoring is increase the communication overhead. We propose a loop based clustering technique to save communication cost for sending beacon message. CH load also reduce.This loop based technology has : (i) There is no critical cluster-heads defined in a loop , for that reason suffers from chain reactions caused by changes of CHs. (ii) Since every node is necessary to have knowledge of other nodes within the loop, if the information of the local loop reserved in one node is corrupt, by querying the neighbor nodes, the loop knowledge can be recovered, which provides the network with better robustness. (iii) one of the important features of a loop is that there are two paths situates between every two nodes i.e a backup route or path will be there for message transfer during connection less condition. A big cluster emerge monitoring overhead and scalability problem . For, scalability and load balancing issue we have a lower bound (L) and upper bound (U) of IoT nodes. Lower bound is subject to the adaptation to fault tolerance necessity of all applications in the framework. Upper bound is picked in view of the most extreme number of nodes in a space (e.g. smart room or Body area Network).Putting nodes from various spaces into one bunch will bring additional monitoring cost and is undesired. Setting an upper bound can likewise restrict the search space for the monitoring. A timestamp is used in all the device to measure control packet period. If the control packet is not receiving within the period then reciver timestamp stimulate the device to report the CH for faulty device. Our aim is to find a CH and decide the sequence for monitoring of nodes in each cluster which use minimal communication cost.

III. System Model and Problem Formulation

We model the service oriented device clustering for fault monitoring as a graph $G = (V, A)$ where V is the set of vertices(i.e. all the nodes in the network) and A is the set of arcs are assumed to be undirected. We consider that the network is a mesh network where every vertex(also referred to as a node) is reachable from every other vertex through one or more hops in a network. Message period(timestamp) is same for all the device. So, hop count measure the communication cost. If $\forall\ u, v \in V$, $\exists\ (u, v) \in A \wedge (v, u) \in A$, then (u, v) is a bidirectional link . c_{uv} is the cost of the path or arc (u, v). Device clustering problem (DCP) is to divide V into t sets $D = \{\ c_1\ ,\ c_2\ ,\ c_3\ , ..., c_p, ..., c_t\ \}$ as clusters and select A', a subset of A, as a packet traverse toor in a cluster for fault monitoring in each cluster, so that :

$*\ \forall\ u \in V$ and $u \in c_p$ i.e. u one and only belongs to one c_p.

$*\ c_p \in D$, the size of each cluster define by $|c_p|$, is between lower bound L and upper bound U , or $L \leq |c_p| \leq U$.

$*\ \forall\ u \in V$ is more than one incoming edge and more than one outgoing edge in each cluster.

$*\ \forall\ (u, v) \in A'$ so that $u \in c_{p_1}$ and $v \in c_{p_2}$ and $p_1 = p_2$.

$*$ The cost of edges in A' , $\sum_{(u,v)\,\in A'} c_{u,v}$ is minimized in a cluster.

We model the fault monitoring problem as a multidepot fixed destination multiple traveling salesman problem .

We first define the binary selection variable $X = x_{uvp}$.

$$x_{uvp} = \begin{cases} 1, & if\, arc(u,v)\, is\, used\, on\, the\, tour\, (u,v) \in c_p \\ 0, & otherwise \end{cases} \tag{1}$$

x_{uvp}=1 means control packet visit v immediately after u in p^{th} set.

The integer programming problem formulation is:

Objective:

$$min \sum_{(u,v)\,\in\, A'} c_{u,v} * x_{uvp} \tag{2}$$

subject to constraints:

$$\sum_{v} x_{uvp} = 1 \quad ,\forall (u) \in A' and (u,v) \in c_p \tag{3}$$

Control packet has to leave every node. Every node is monitoring exactly one other node.

$$\sum_{u} x_{uvp} = 1 \quad ,\forall (v) \in A' and (u,v) \in c_p \tag{4}$$

Control packet has to enter every node. Every node is monitored by exactly one other node.

$$\sum_{u} x_{uvp} = \sum_{u} x_{vup} \ , \forall v \in A' \tag{5}$$

Two edges of the same node belongs to the same cluster.

For a control packet , s_u is the number of nodes visited on that path from the origin up to node u(the visit number of the u^{th} node).s_u variable use for traversing the node u in a cluster. If u is a CH in a cluster then u initialize by 1 and increment by 1 after visiting each node in a cluster. U is the maximum number of nodes in a cluster, thus $1 \leq s_u \leq U, \forall u \geq 2$. In addition L minimum number of nodes in a cluster for fault tolerance and load balancing, $L \leq s_u \leq U$ must be satisfied.

$$s_u + (U-2) \sum_{p \in D} x_{pup} - \sum_{p \in D} x_{upp} \leq U-1, \, u \in A' \quad (6)$$

$$s_u + \sum_{p \in D} x_{pup} + (2-L) \sum_{p \in D} x_{upp} \geq 2, \, u \in A' \quad (7)$$

Constraint (6) and (7) impose bound of the number of nodes a control packet visit.

$$x_{pu} + x_{up} \leq 1, \, p \in D, U \in A' \quad (8)$$

Constraint (7) prohibit a control packer serving only a single node.

Sub-tour elimination constraints are:

$$s_u - s_v + U x_{uv} + (U-2) x_{vu} \leq U-1, \, u \neq v, u, v \in A' \quad (9)$$

$$x_{uv} \in 0, 1, \; \forall u, v \in A \quad (10)$$

Here $O(tn^2)$ binary variables and $O(n^2)$ constraints. Since Multidepot fixed destination multiple traveling salesman problem is an NP-hard problem. In our model each CH there is only one control packet . So, our problem also an NP-hard problem.

IV. Proposed Technique

Here we propose a loop-based clustering technique for fault monitoring in IoT device management. It executes on an on-demand basis and discovers a configurable loop topology for the fault monitoring. Because the nodes in sensor networks are likely to be fixed and the transmission radious of each sensor is configurable , the algorithm assumes that (i) it is applied in a fully connected network with only bidirectional links, (ii) each node has a unique identifier throughout the network , and (iii) during one execution of the algorithm , topology of the algorithm keeps unchanged. A loop is a bidirectional path which begins and ends with the same node. There is at most one connection between every two node, that is, if there exist two edges e_i and e_j connecting v_x and v_y , then e_i and e_j must be identical, so a path from v_n to v_m can be defined as a sequence of only vertices's $\{v_n, v_{n+1,} ...., v_m\}$. We define a loop as a sequence of vertices $\{v_n, v_{n+1,} ...., v_m\}$ where $v_i \neq v_j (i \neq j)$ for any $n \leq i \leq m-1$ and $n \leq j \leq m-1$, and $v_n = v_m$. The length of a loop is the number of hops from v_i to v_j, equal to the number of nodes on the loop - 1. Let l be a loop. When $len(l)$ is smaller than 3, either the node is isolated or l is a round trip between two nodes. A loop with only two nodes is regarded as a special loop.

Message structure:

Hello Message($Hello$): each node periodically broadcasts $Hello$ message to its adjacent nodes within its range to discover neighbors and establish a link. The content of $Hello$ message includes message type , sender node ID and neighbor-list.

Loop Discovery Request(LDR): this message is used to discover loops in the network and its content includes message type, sender node ID, message sequence number and link information. These are stored into an array.

The basic principle of this process is to establish clusters based on loops in the heterogeneous sensor devices with bidirectional links. Loop based clustering for fault monitoring consists of the following components :

1. Neighbor discovery and link formation
2. Cluster discovery and delete nested Cluster
3. Finding monitoring sequence of each cluster
4. Faulty device detection
5. Cluster reconstruction

- Neighbor discovery and link formation

Every node periodically broadcast $Hello$ message to its neighbors in every t_1 seconds. Each sensor node can detect only events that are within certain range of it. Shape of the detected area is rectangular or square because sensors are deployed in a room or in a office. How can we use a minimum number of sensor nodes achieving requirement of coverage and connectivity is a NP-hard problem. It also continuously listens for similar transmission from other nodes. Each node has its own unique ID. When a node receive a $Hello$ message , it checks whether its own ID is in the neighbor-list of Hello. If it is true then discard the $Hello$ message. If its own ID is not present in the neighbor-list then receiver node add its ID to the $Hello$ message and establish a link with the sender node and broadcast the $Hello$ message. Receiver node and sender node will store the neighbor IDS and link information obtained from $Hello$ message in its neighbor-table. If an node can't receive $Hello$ message from one neighbor in t_2 seconds, it will decide that the link between them is already failed and delete this neighbor from neighbor-table.

- Cluster discovery and delete nested cluster

Searching the loop in network firstly, and considering a discovered loop as a cluster, then each node in a cluster acquires the routing to other nodes in the same cluster. Initially, all the nodes in the network are initial nodes. If any node found a loop, then we consider that node as a $ClusterHead$ (CH).For loop building, we choose a range L and U for the number of nodes in one loop. Any nested loops must be deleted. We notice that each node is prone to join the earliest discovered loop and does not take into account whether it is good . So the result of loop-based technique discovered is hard to predict. The simplest criterion for determining good loops is the length of the loop. If the network is clustered with bigger loops , the communication cost is minimized and it is easier for routing . However, since there are more nodes on one loop , the possibility of connection loss increases and then the topology becomes less stable. So, to choose a good loop length is a trade-off between communication cost and stability.

The detailed steps are described in Algorithm 1, Algorithm 2 and Algorithm 3.

Algorithm 1: $Cluster\_Configuration(V, L, U)$

```
Input: Set of nodes $V = \{1, 2, ....., n\}$; lower bound $L$ and
       upper bound $U$ of cluster size
Result: $D = \{\ c_1,\ c_2,\ c_3, ...,\ c_p, ...,\ c_t\ \}$ , $CH_p$ is the head
        of cluster $c_p \in D$
1  while $|V| \neq 0$ do
2      Select a node $s \in V$ as a source node arbitrarily from
         network(in general convenient for operation and
         management)
3      $Send(s, Neighbor\_List(s), LDR)$;
4      $Receive(r, LDR)$; /* $\forall r \in V$ */
5      if $r.ID \notin LDR$ then
6          $Update(r.ID, LDR)$;
7          $Send(r, Neighbor\_List(r), LDR)$;
8      else
9          Discard LDR ;
10     if $Receive(r, LDR)$>1 then
11         $c_p \leftarrow Each\_Cluster\_Formation(r, LDR[])$;
12         $CH_p \leftarrow r$ ;
13         $Cluster\_Aware\_Message(CH_p$,$c_p)$;
14     Add($c_p$,D);
15     $V \leftarrow V - c_p$; /*$c_p \in D$*/
16
```

Algorithm 2: $Each\_Cluster\_Formation(r, LDR[])$

```
Input: Set of LDR messages $LDR = \{1, 2, ....., n\}$, r
Result: $c_p, r \leftarrow CH_p$
1  $LDR_m = \{y_1, y_2, ...., y_i, ....., y_m, r\}$ ;
2  $LDR_n = \{z_1, z_2, ...., z_j, ....., z_n, r\}$;
3  if $(y_i == z_j\ and\ y_x \neq z_y) and (i < x < m\ and\ j < y < n)$
     then
4      if $(\{r, y_m, ....., y_i, z_j, ...., z_n, r\} ==$
         $TRUE\ and \{r, z_n, ...., z_j, y_i, ...., y_m, r\} == TRUE)$
         then
5          $c_p \leftarrow cluster(r)$;
6      if $(L \leq |c_p| \leq U)$ then
7          $c_p \leftarrow accepted$;
8      else if $(|c_p| < L\ \ and\ \ Neighbor\_Cluster(c_p) > L)$ then
9          Share nodes $Neighbor\_Cluster(c_p)\ to\ c_p$;
10     else if $(|c_p| <$
         $L\ and\ (can't\ form\ a\ cluster\ sharing\ with\ neighbor\ cluster))$
         then
11         $c_p \leftarrow Special\_Cluster$;
12     else
13         $c_p \leftarrow rejected\ \ and\ add\ c_p\ to\ S$;
14 Nested cluster should be deleted and bigger one (contain
     maximum nodes)should be remain.
15 return $c_p$
16
```

- Finding monitoring sequence of each cluster

After deciding the monitoring clusters, we need to find the monitoring sequence so that the total communication cost is minimized. This is the classical travelling salesman problem (TSP) , a well known NP-hard combinatorial problem.

Here we find the possible link of each node in a cluster c_p and select the shortest path. Let $c_p = (V', A')$ is a complete weighted graph which is used to represent a TSP, where V' is the set of n devices and A' is the set of arcs(paths) fully connecting all devices in a cluster. Each arcs $(u, v) \in A'$ is assigned a communication cost c_{uv}, which is the distance between devices u and v. In a TSP problem , the arficial ants are distributed randomly to these n devices . Each ant will choose the next to visit according to the pheromone trail remained on the paths. Then artificial ants have "memory" which is used to remember the nodes (devices) they have visited and therefore they would not select those nodes again. The artificial ants are know the communication distance netween two nodes and prefer to choose the nearby nodes from their positions. Therefore , the probability that node v is selected by ant k to be visited after node u could be written as follows:

Algorithm 3: $Cluster\_Aware\_Message(CH_p, c_p)$

```
Input: Set of nodes of a cluster $c_p = \{1, 2, ....., n\}$ ,$CH_p$
Result: $Add\ nodes\ \ of\ c_p\ to\ cluster(c_p)\_List$
1  $CH_p \leftarrow Generate(CAM)$ /*Cluster Aware Message(CAM)*/;
2  while $\forall nodes\ cluster(c_p) \leftarrow visited$ do
3      $Send(CH_p$,v,CAM);   $/ * v \in$
         $Neighbor\_List(CH_p),\ \ i.e\ \ v = y_m\ \ or\ \ v = z_n * /$
         $Receive(v, CAM)$
4      if $v\ \ already\ \ received\ \ CAM$ then
5          discard CAM
6      else
7          $Add(v, cluster(c_p)\_List)$;
8          $Send(v, t, CAM)$ /*$t \in Neighbor\_List(v)$ */
9
```

$$p_{uv}^k = \begin{cases} \frac{[\tau_{uv}]^\alpha . [\eta_{uv}]^\beta}{\sum_{a \in W_k(u)} [\tau_{ua}]^\alpha [\eta_{ua}]^\beta} & if\, v \in W_k(u) \\ 0 & otherwise \end{cases} \tag{11}$$

where τ_{uv} is the intensity of pheromone trail between nodes (u, v), α the adjustable positive parameter that control τ_{uv}, η_{uv} is the heuristic visibility of node v from node u, which is always $1/c_{uv}$, β the adjustable positive parameter that control η_{uv} and $W_k(u)$ is a set of nodes which remain to be visited when the ant is at node u.

At the beginning, r ants are placed to the n nodes randomly. Then each ant decides the next node to be visited according to the probability p_{uv}^k given by Eq. (11). Every ant completes a tour after n iterations of this process. Obviously, the ants with shorter tours should leave more pheromone than those longer tours. Therefore,the trail levels are updated as on a tour each ant leaves pheromone quantity given by Q/L_k, where Q is a constant and L_k the length of its tour, respectively. After completing a tour each ant updating the pheromone amount on each path could be written as follows:

$$\tau_{uv}\,(t+1) \,=\, \rho .\, \tau_{uv}\,(t) \,+\, \Delta\tau_{uv}\,(t) \tag{12}$$

In this equation,

$$\Delta\tau_{uv}\,(t) = \sum_{k=1}^{r} \Delta\tau_{uv}^k(t) \tag{13}$$

$$\Delta\tau_{uv}^{k}\,(t) = \{Q/L_k \;\;, if\, ant\, k\; travels\, on\; arc\,(u,v) \quad 0 \qquad otherwise \tag{14}$$

where t is the iteration counter , ρ $(0 < \rho < 1)$ the parameter to regulate the reduction of τ_{uv}, $\Delta\tau_{uv}\,(t)$ the total increase of trail level on arc (uv) and $\Delta\tau_{uv}^{k}(t)$ the increase of trail level on edge(uv) caused by ant k, respectively. After the pheromone trail updating process, the next iteration $t+1$ will start.

Ant colony optimization for TSP, the probability that node v is selected by ant k to be visited after node u is computed according to two factors, namely, that the pheromone trail quantity distributed on the paths and the visibility of node v from node u. A simple idea is that the probability should also be computed by Eq.(11). , where the set $W_k(u)$ is the set of nodes which are not visited when the ant k is at node u. And the updating rule of τ_{uv} is also followed by Eqs. (12)-(14). Denote VN_k as the set of visited nodes by ant k. In this Algorithm 4 describe the process of monitoring sequence of each cluster.

Algorithm 4: $Monitoring\_Sequence\_Each\_Cluster(C_p, CH_p)$

```
Input: Cluster $c_p$ , $CH_p$
Result: $S$ is the set of arcs for shortest tour
1  $t = 0$; /* t is time counter */
2  $\tau_{uv} = c$; /*$\forall(u,v) \in A'$*/
3  $\Delta\tau_{uv} = 0$ /*$\forall(u,v) \in A'$*/
4  for $k = 1$ to $r$ do
5      Place ant k on a node x randomly;
6      $VN_k[] \leftarrow x$;
7  for $k = 1$ to $r$ do
8      Choose the next node to be visited according to the
         probability $p_{uv}^{k}$ given by Eq. (11);
9      Move the ant k to the selected node;
10     Insert the selected node in $VN_k$;
11 for $k = 1$ to $r$ do
12     Move the ant k from $VN_k(n)$ to $VN_k(1)$;
13     Compute the tour length $L_k$ traveled by ant k ;
14     Update the shortest tour found.
15 for $k = 1$ to $r$ do
16     Update the pheromone trail density $\tau_{uv}$ according to
         Eq.(12)-(14);
17     $t = t + 1$;
18 if $t < TIME\_MAX$ then
19     Empty all $VN_k$; Goto step 4;
20 else
21     Print the shortest tour $S$.
```

- Faulty device detection

Fault detection in a network is an essential task before data transmission. In our model we have used a beacon message Neighbor Alive Message(NAM) as a control packet is send periodically for detecting faulty device(DE). We check whether the devices(DEs) are working or not because unnecessary time loss occurs in sending data to faulty devices. Algorithm 5 describe the process.

Algorithm 5: $Faulty\_Devices\_Detection(C_{p_{ms}}, CH_p, NAM)$

```
Input: Monitoring sequence of a cluster at time
         $t_1$, $C_{p_{ms}} = \{d_p, d_q, d_r, ....., d_t, d_p\}, d_p \leftarrow$
         $CH, NAM$
Result: Faulty devices
         $DE = \{d_a, d_b, ...., d_c\} where DE \in$
         $C_{p_{ms}} and\ \ |DE| < |c_{p_{ms}}|$
1  Each device of $C_{p_{ms}}$ send() and receive() NAM message
     at $\Delta t$ time interval;
2  Let devices $x, y \in C_{p_{ms}}$ and monitoring sequence is left
     to right in cluster $C_{p_{ms}}$;
3  $y\ is\ the\ right\ neighbor\ of\ x$
4  Device x send NAM message at $\Delta t$ time interval to y
     and y send NAM message at $\Delta t$ time interval to right
     neighbor of y.;
5  Send(x,y,NAM);
6  Send(y,$Right\_Neighbor(y)$,NAM);
7  Device y receive NAM message at $\Delta t$ time interval from
     x and ;
8  if !$Receive(y, x, NAM)$ /*y not receive NAM message
     from x at $\Delta t$ time*/ then
9      Device x is faulty and report to $d_p$.;
10     Add x to DE and $d_p$ provide backup;
11
```

- Cluster recovery

If any link failure occurs then a cluster node send link failure notification to it's neighbors in the same cluster and try to find any alternate link to form a new cluster.

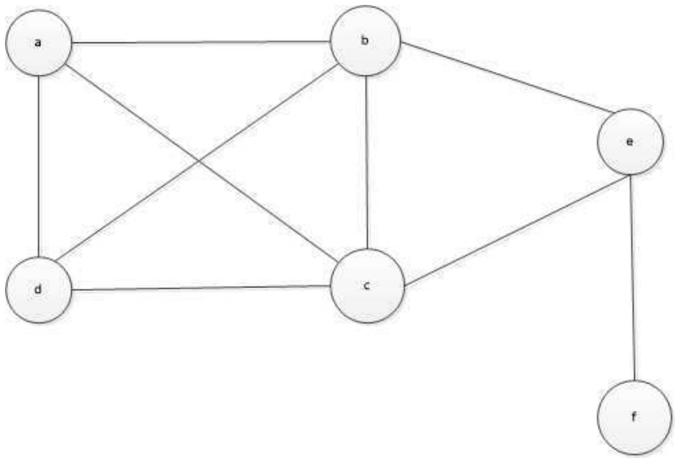

Fig. 1. Simulation results for the network.

For example, as shown in Fig (1) node a, b, c, d, e, f periodically broadcast $Hello$ message to its neighbors in every

T_1 seconds. Each node has an unique ID. These 6 nodes can obtain their neighbor information and establish a bidirectional link through the process of neighbor discovery. Suppose LDR messages are being transfered from node a to b, c and d. Here a is the source node which is taken arbitrarily. Node b, c and d receives LDR message , add their ID at the end of LDR and send to its neighbors. When a gets LDR forwarded from b, c and d, those messages will be discard because a is found in LDR.Let $L = 4$ and $U = 10$. Node c receives two LDR messages a, b, c and a, c. But loop$\{c, b, a\}$ can't form because total no of nodes is less than L value. When node c receives two LDR message a, b, c and $\{a, d, c\}$ then loop or cluster$\{c, d, a, b\}$ form where total no of nodes greater than L and less than U. Cluster Aware message (CAM) send to all the nodes of a cluster for being a member of a cluster. We can see the loop e, c, d, a, b has a nested loop $\{c, d, a, b\}$. So, we delete nested loop $\{c, d, a, b\}$. We consider loop $\{e, c, d, a, b\}$ as an example. Each node send NAM message to its neighbor.Suppose node c send NAM message to its neighbor $\{b, d\}$. If $\{c - b\}$ link failure then $\{c, a, b, d\}$ loop is form. Node f can not form a loop. so $\{e, f\}$ is a special loop.

V. Conclusion

We proposed loop-based clustering for service oriented fault monitoring system in device management of IoT. This is designed to neighbor discovery and link formation , cluster discovery and delete nested cluster,finding monitoring sequence of each cluster,faulty device detection and cluster reconstruction. This takes lower communication overhead.

VI. Acknowledgment

This work is funded by Media Lab Asia (Visvesvaraya Ph.D. Scheme for Electronics and IT, Project Code-CS-VSE) under the department of DeitY government of India. The work is done at Department of Computer Science and Engineering ,NIT Rourkela, India.

References

[1] T. T. Mulani and S. V. Pingle, "Internet of things," *International Research Journal of Multidisciplinary Studies*, vol. 2, no. 3, 2016.

[2] A. M. Magazine, "Internet of things."

[3] J. Gubbi, R. Buyya, S. Marusic, and M. Palaniswami, "Internet of things (iot): A vision, architectural elements, and future directions," *Future Generation Computer Systems*, vol. 29, no. 7, pp. 1645 – 1660, 2013, including Special sections: Cyber-enabled Distributed Computing for Ubiquitous Cloud and Network Services &amp; Cloud Computing and Scientific Applications Big Data, Scalable Analytics, and Beyond. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0167739X13000241

[4] G. Misra, V. Kumar, A. Agarwal, and K. Agarwal, "Internet of things (iot)–a technological analysis and survey on vision, concepts, challenges, innovation directions, technologies, and applications (an upcoming or future generation computer communication system technology)," *American Journal of Electrical and Electronic Engineering*, vol. 4, no. 1, pp. 23–32, 2016.

[5] S. K. Datta and C. Bonnet, "A lightweight framework for efficient m2m device management in onem2m architecture," in *Recent Advances in Internet of Things (RIoT), 2015 International Conference on*, April 2015, pp. 1–6.

[6] Y. B. Lin, Y. W. Lin, C. Y. Chih, T. Y. Li, C. C. Tai, Y. C. Wang, F. J. Lin, H. C. Kuo, C. C. Huang, and S. C. Hsu, "Easyconnect: A management system for iot devices and its applications for interactive design and art," *IEEE Internet of Things Journal*, vol. 2, no. 6, pp. 551–561, Dec 2015.

[7] Z. Sheng, H. Wang, C. Yin, X. Hu, S. Yang, and V. C. M. Leung, "Lightweight management of resource-constrained sensor devices in internet of things," *IEEE Internet of Things Journal*, vol. 2, no. 5, pp. 402–411, Oct 2015.

[8] A. Sehgal, V. Perelman, S. Kuryla, and J. Schonwalder, "Management of resource constrained devices in the internet of things," *IEEE Communications Magazine*, vol. 50, no. 12, pp. 144–149, December 2012.

[9] A. Zanella, N. Bui, A. Castellani, L. Vangelista, and M. Zorzi, "Internet of things for smart cities," *IEEE Internet of Things Journal*, vol. 1, no. 1, pp. 22–32, Feb 2014.

[10] M. A. Marotta, C. B. Both, J. Rochol, L. Z. Granville, and L. M. R. Tarouco, "Evaluating management architectures for internet of things devices," in *2014 IFIP Wireless Days (WD)*, Nov 2014, pp. 1–7.

[11] S. Zhou, K. J. Lin, and C. S. Shih, "Device clustering for fault monitoring in internet of things systems," in *Internet of Things (WF-IoT), 2015 IEEE 2nd World Forum on*, Dec 2015, pp. 228–233.

[12] Y. Li, X. Wang, F. Baueregger, X. Xue, and C. K. Toh, "Loop-based topology maintenance and route discovery for wireless sensor networks," in *GLOBECOM '05. IEEE Global Telecommunications Conference, 2005.*, vol. 5, Dec 2005, pp. 5 pp.–3111.

[13] E. Shakshuki, M. Younas, M. G. Khair, B. Kantarci, and H. T. Mouftah, "The 2nd international conference on ambient systems, networks and technologies (ant-2011) / the 8th international conference on mobile web information systems (mobiwis 2011) heterogeneous clustering of sensor network," *Procedia Computer Science*, vol. 5, pp. 939 – 944, 2011. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1877050911004571

[14] V. Mhatre and C. Rosenberg, "Homogeneous vs heterogeneous clustered sensor networks: a comparative study," in *Communications, 2004 IEEE International Conference on*, vol. 6, June 2004, pp. 3646–3651 Vol.6.

[15] T. Bektas, "The multiple traveling salesman problem: an overview of formulations and solution procedures," *Omega*, vol. 34, no. 3, pp. 209 – 219, 2006. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0305048304001550

[16] I. Kara and T. Bektas, "Integer linear programming formulations of multiple salesman problems and its variations," *European Journal of Operational Research*, vol. 174, no. 3, pp. 1449 – 1458, 2006. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0377221705003188

[17] J. Yang, X. Shi, M. Marchese, and Y. Liang, "An ant colony optimization method for generalized {TSP} problem," *Progress in Natural Science*, vol. 18, no. 11, pp. 1417 – 1422, 2008. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1002007108002736

[18] Z. C. S. S. Hlaing and M. A. Khine, "An ant colony optimization algorithm for solving traveling salesman problem," in *International Conference on Information Communication and Management, Singapore, IPCSIT*, vol. 16, 2011.