Singapore Management University
Institutional Knowledge at Singapore Management University

Research Collection School Of Information Systems | School of Information Systems

6-2017

Virtualization in wireless sensor networks: Fault tolerant embedding for internet of things

Omprakash KAIWARTYA

Abdul Hanan ABDULLAH

Yue CAO

Jaime LLORET

Sushil KUMAR

See next page for additional authors

Follow this and additional works at: http://ink.library.smu.edu.sg/sis_research

 Part of the Digital Communications and Networking Commons, and the OS and Networks Commons

Citation

KAIWARTYA, Omprakash; ABDULLAH, Abdul Hanan; CAO, Yue; LLORET, Jaime; KUMAR, Sushil; SHAH, Rajiv Ratn; PRASAD, Mukesh; and PRAKASH, Shiv. Virtualization in wireless sensor networks: Fault tolerant embedding for internet of things. (2017). *IEEE Internet of Things*. Research Collection School Of Information Systems.
Available at: http://ink.library.smu.edu.sg/sis_research/3791

This Journal Article is brought to you for free and open access by the School of Information Systems at Institutional Knowledge at Singapore Management University. It has been accepted for inclusion in Research Collection School Of Information Systems by an authorized administrator of Institutional Knowledge at Singapore Management University. For more information, please email libIR@smu.edu.sg.

Author
Omprakash KAIWARTYA, Abdul Hanan ABDULLAH, Yue CAO, Jaime LLORET, Sushil KUMAR, Rajiv Ratn SHAH, Mukesh PRASAD, and Shiv PRAKASH

This journal article is available at Institutional Knowledge at Singapore Management University: http://ink.library.smu.edu.sg/sis_research/3791

Published in IEEE Internet of Things Journal,
2017 June
http://doi.org/10.1109/JIOT.2017.2717704

Virtualization in Wireless Sensor Networks: Fault Tolerant Embedding for Internet of Things

Omprakash Kaiwartya, *Member IEEE*, Abdul Hanan Abdullah, *Member IEEE*, Yue Cao, *Member IEEE*, Jaime Lloret, *Senior Member IEEE*, Sushil Kumar, *Member IEEE*, Rajiv Ratn Shah, Mukesh Prasad, Shiv Prakash

***Abstract*—Recently, virtualization in wireless sensor networks (WSNs) has witnessed significant attention due to the growing service domain for IoT. Related literature on virtualization in WSNs explored resource optimization without considering communication failure in WSNs environments. The failure of a communication link in WSNs impacts many virtual networks running IoT services. In this context, this paper proposes a framework for optimizing fault tolerance in virtualization in WSNs, focusing on heterogeneous networks for service-oriented IoT applications. An optimization problem is formulated considering fault tolerance and communication delay as two conflicting objectives. An adapted non-dominated sorting based genetic algorithm (A-NSGA) is developed to solve the optimization problem. The major components of A-NSGA include chromosome representation, fault tolerance and delay computation, crossover and mutation, and non-dominance based sorting. Analytical and simulation based comparative performance evaluation has been carried out. From the analysis of results, it is evident that the framework effectively optimizes fault tolerance for virtualization in WSNs.**

***Index Terms*—IoT, Virtualization, Wireless sensor networks.**

I. Introduction

Network virtualization has got significant attention as an enabling technology for service-oriented heterogeneous network for Internet of Things (IoT) [1]. The rigid communication architecture of Internet is one of challenging issues in IoT. Network virtualization enables Internet to retain its communication architecture while enlarging and transforming as IoT. In IoT enabled smart cities, service-oriented communication architecture is required for smart applications (see Fig.1). It can be addressed by implementing network as a service through virtualization [2]. Sensing as a service can be implemented in collaboration with network as a service, to support heterogeneous networking and sensing resource optimization for smart applications in IoT.

The research is supported by Ministry of Higher Education Malaysia in collaboration with Research Management Centre, Universiti Teknologi Malaysia (UTM) under Grant: Q.J130000.2528.14H81

O. Kaiwartya, A.H. Abdullah are with the Universiti Teknologi Malaysia, Johor Bahru, 81310, Malaysia. Email: omprakash@utm.my; hanan@utm.my

Y. Cao (Corresponding Author) is with the Department of Computer and Information Sciences, Northumbria University, Newcastle upon Tyne, NE2 1XE, UK. Email: yue.cao@northumbria.ac.uk

J. Lloret is with the Polytechnic University of Valencia, Camino de Vera 46022, Valencia, Spain. Email: jlloret@dcom.upv.es

S. Kumar is with Jawaharlal Nehru University (JNU), New Delhi, 110067, India. Email: skdohare@mail.jnu.ac.in

R. R. Shah is with the Singapore Management University (SMU), Singapore, 188065. Email: rajivshah@smu.edu.sg

M. Prasad is with University of Technology Sydney, Ultimo, New South Wales 2007, Australia. Email: mukesh.prasad@uts.edu.au

S. Prakash is with the Indian Institute of Information Technology Kottayam (IIIT-K), 686635, Kerala, India, Email: shivprakash16@gmail.com



Fig. 1. Heterogeneous networks in Internet of Things (IoT)

Network virtualization is implemented using virtual network embedding (VNE) algorithms [3]. Due to the inception and evolution of VNE, the role of Internet service provider has been divided into two parts namely, infrastructure provider and service provider. Infrastructure provider is responsible for deployment and maintenance of physical infrastructure whereas, service provider is responsible for range of customizable services to end user. Various virtual network embedding techniques have been suggested focusing on wired-networks [4]. The two key issues in virtual network embedding include resource optimization [5], and survivability [6].

Due to the recent developments in IoT technology, network virtualization in WSNs has been envisioned [7]. Some of the initial works on the virtualization focuses on service-oriented network architecture to optimize sensing resources. An four-layered architecture for virtualization in WSNs has been suggested, based on reducing redundant deployment of sensor networks for different IoT applications [8]. Another five-layered virtualization architecture has been explored to support network diversity and increase resource utilization in IoT [2]. Brain-inspired adaptive architecture has been presented for embedding and running IoT applications on virtual wireless sensor networks [9]. The aforementioned architectures did not consider communication failure on virtual networks, caused by the communication failure on physical WSNs. The fault tolerance (reactive capability of handling communication failure in physical networks) is a pressing issue in virtualization in wired networks [10]. The

fault tolerance in WSNs environments is more challenging as compared to its wired counterparts, due to the dynamic wireless channel based connectivity. Bandwidth reservation based fault tolerance enhancement decreases resource utilization [11]. Moreover, estimation of guaranteed connectivity, which is utilized to enhance fault tolerance in VNE for wired-networks [12], is quite challenging in WSNs.

In this context, this paper proposes a framework to maximize fault tolerance and minimize communication delay for VNE in WSNs environments focusing on IoT services. It is a reactive optimization of fault tolerance and communication delay for service-oriented heterogeneous virtual networks in IoT. Specifically, the major components of the framework are listed below as contributions to literature:

1) The fault tolerance optimization problem is mathematically formulated considering fault tolerance and communication delay as the two conflicting objectives in WSNs environments.
2) An adapted non-dominated sorting based genetic algorithm (A-NSGA) is developed to solve the optimization problem.
3) The solution framework is consisted of chromosome representation, fault tolerance and delay computation, crossover and mutation, and non-dominance based sorting.
4) Simulations are performed to analyze the performance of A-NSGA in optimizing fault tolerance for virtualization in WSNs.

The rest of the article is organized into following sections. Section II reviews related literature on fault tolerance in VNE considering three categorizations including survivability, topological knowledge for trust, and reliability using single objective optimization. Section III presents the detail of mathematical formulation of the multi-objective optimization problem, and A-NSGA for solving the problem. Section IV discusses simulation setting, metrics and comparative performance evaluation, followed by conclusion made in section V.

II. Related Work

In this section, related literature on fault tolerance in VNE is reviewed. It is divided into three categories including resource reservation, topological knowledge, and single objective optimization based survivability. Next, virtualization in WSNs is critically explored as the contribution area.

A. Resource Reservation based Survivability

A proactive survival virtual network embedding (SVNE) technique has been suggested for single link failure using bandwidth reservation [13]. Specifically, SVNE is mathematically formulated and solved using two heuristic considerations including hybrid policy heuristic and baseline policy heuristic. The preserved quota for backup on each link has been utilized for restoration and protection. Although link restoration and protection based survivability measure is quite effective in business perspective. The management of restoration information for each link reduces the applicability of the approach in realistic network scenario specifically in case of wireless environments. Moreover, fault tolerance capability of links and communication delay are not considered in the survivability measurement. SVNE has been enhanced by addressing node failure along with link failure using dynamic recovery (DR-SVNE) technique [11]. An algorithm has been developed to discover alternative path end points of the failed link as source and destination. Node failure has been addressed by discovering a set of alternative paths for each link of the node. Although back up resources have not been utilized in dynamic recovery yet, it manages backup information for recovering each link. It did not explore all the alternative paths to select best available alternative path rather finds the first alternative path satisfying the requirement.

Another survival virtual network embedding technique has been suggested based on optimal resource allocation for both working and backup resources [14]. The problem of single node failure in the network has been viewed as NP complete multi-commodity network flow problem. Integer linear programing model based solution has been suggested to utilize available and backup resources optimally. Auxiliary protection graph is constructed to facilitate single node survivability. Heuristic based algorithm is developed for embedding auxiliary protection graph to physical network. Although heuristic based mapping effectively addresses single node failure. The consideration of backup resources, and possibility of multiple nodes and links failure are the undesirable aspects of this embedding technique. SVNE has been enhanced considering failure dependent protection and transformed virtual network request [10]. A backup node has been considered corresponding to each node. Enhanced virtual network has been designed from the initial virtual network request using complete connected graph, for minimizing the requirement of resources to survive in case of failure. Binary quadratic programing and mixed integer linear programing have been utilized to formulate failure dependent protection and transform virtual network, respectively. Heuristic based algorithms have been developed for solving the embedding problem. The undesirable aspect of failure dependent protection is the management of backup resources.

B. Topological Knowledge based Survivability

Topology aware virtual network embedding has been suggested to improve utilization of resources, and thus, maximize revenues due to the better utilization [15]. Six topological characteristics have been suggested to rank both node and link during mapping. The characteristics include degree, strength, closeness, between-ness, eigen vector, and Katz centralities. Heuristic embedding algorithms have been developed by utilizing these topological characteristics apart from resource requirements. The topological characteristics have been devised focusing on better utilization of resources without considering the fault tolerance capability. Topology aware embedding has been improved considering convergence degree to avoid embedding of virtual link on longer physical path [12]. The degree of virtual nodes has been considered as convergence degree. Maximum convergence degree based embedding algorithm has been developed to ensure shorter physical paths for virtual links. Although survivability improves due to the shorter path consideration yet, no direct contribution on fault tolerance. Geographical location of physical resources has been utilized in VNE to enhance survivability with lower operational cost [16]. Location constrained survival virtual network embedding (LSNE)

problem has been vied as multi-commodity network flow problem. The problem has been addressed using integer linear programing model for small size network whereas, heuristic based algorithm has been utilized for larger size network. The impact of location information is negligible in case of smaller network due to the high speed of physical links. The topological knowledge significantly increase revenue gain, and indirectly enhance survivability. The undesirable aspects of topology based survivability is the direct impact of topology change on virtual networks. It could force re-embedding for revenue gain.

C. Single-Objective Optimization based Survivability

Fault tolerance in virtual network embedding with redundant resource has been modelled as an integer linear programming problem [17]. A single-objective optimization technique namely, discrete particle swarm optimization (PSO) has been utilized to solve the embedding problem. The consideration of redundant resource as backup reduces resource utilization of the PSO-based survivability enhancements. Another single-objective optimization technique namely, ant colony optimization (ACO) has been utilized to improve survivability of VNE, by guarantying completeness of virtual network despite a node failure [18]. The survival embedding problem has been modelled as integer linear programming problem. To solve the problem, firstly, primary-cut set of has been identified from the sub-network of the virtual network obtained after the removal of failed node. Secondly, ACO has been utilized to find optimal embedding of virtual network. The performance of some well-known single-objective optimization techniques and their variants has been investigated comparatively for virtual network embedding [19]. ACO-based algorithms differ in the order of virtual node embedding and computing pheromone trail whereas, PSO-based algorithms differ in calculating rank of nodes. Although number of network parameters are considered for procuring survival embedding yet, optimal values of the parameters not obtained due to the usage of single-objective optimization techniques.

D. Virtualization in WSNs for IoT: The Contribution Area

Recently, virtualization in WSNs has witnessed significant attention due to the advancements in IoT technology. Although not much works has been done in the area yet, some potential initiatives have been made towards VNE for WSNs infrastructure. Network virtualization for IoT has been introduced focusing on resource constrained devices [7]. It has focused on the two key issues in IoT environments, including the lack of end-to-end connectivity among sensors in gateway based communication, and access control and authentication in direct communication without using gateways. Some use cases have conceptualized the virtual networking in IoT. An architecture has been suggested for virtualization of WSNs, to reduce redundant deployment of sensor networks for different IoT applications [8]. The architecture includes four layers considering physical, virtual sensor, virtual sensor access, and application. Two communication paths including data and control, and five interfaces have been also considered in the architecture. Another network virtualization architecture for IoT has been suggested to support network diversity and increase resource utilization [2]. A five-layered architecture namely, smart service system has been developed for implementing network virtualization in IoT. A four-phase service delivery model for IoT has been designed considering registration, discovery, description, and execution of services. The virtual network resource management model for IoT environments has been presented to handle service delivery with optimal resource utilization. Brain-inspired adaptive architecture has been presented for embedding and running IoT applications on virtualized WSNs [9]. Specifically, hierarchical modular structure of brain has been utilized for constructing different level of virtualization.

Most of the proposal on virtualization in WSNs focuses on enhancing resource (i.e., sensor) utilization, by executing application-centric multiple tasks in sensors and abstracting sensors based on application (i.e., virtual sensors). This is evident from recent surveys focusing on critical investigation of node and network level virtualization in WSNs for IoT [20, 21] and applications [22-25]. Different levels of abstraction have been considered for the same purpose including four layers [8], five layers [2], and hierarchical layers [9]. Complementary to these proposal, we focus on fault tolerance aspect in virtualization in WSNs, which can be utilized to aid the aforementioned resource-centric proposals. Specifically, we utilize multi-objective optimization to maximize fault tolerance and minimize communication delay parallelly.

III. Fault Tolerant Virtualization

In this section, the detail of fault tolerant aspect of virtualization in WSNs is presented. A four-layered network architecture is considered (see Fig. 2). The bottom most physical layer is represented by the actual sensor nodes, i.e., (s_1, s_2, s_3, s_4), capable of performing different types of tasks. Task based virtualization of sensors is performed in the sensor virtualization layer, i.e., $(s_{11}, s_{12}, \dots, s_{43})$, resulting in more number of sensors than the number of actual physical sensors. Different wireless networks are generated at the access layer based on fault tolerant embedding of task oriented sensors. Each embedded network is considered as access agent. The application layer represents smart applications of IoT. The proposal is implemented in the access layer.

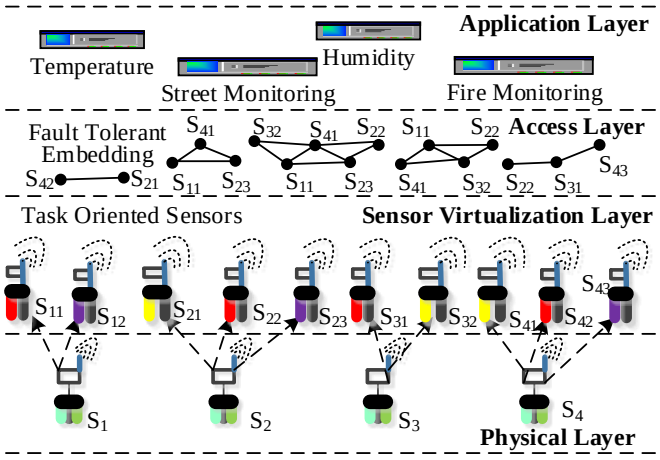


Fig. 2. Fault tolerance embedding at virtual sensor access layer

Table I. Nomenclature

Notation	Description
S_i or S_{ij}	i^{th}sensor or j^{th}virtualization of i^{th}sensor
r	Distance of nearest neighbour sensor
Δr	Small increment in r
N_A	Network area
λ	Density of sensors in the network
$P^c_{r\|(r+\Delta r)}$	Probability of closest sensor between r and $r+\Delta r$
$P^s_{r\|(r+\Delta r)}$	Probability of some sensor between r and $r+\Delta r$
$P^0_{<r}$	Probability of no sensor at less than $r$distance
N_n	Number of sensors in the network
$f_r(r)$	Pdf of closest neighbour distance
R	Transmission range of sensors
$E(r)$	Expected closest neighbour distance
s^p	Source physical sensor
d^p	Destination physical sensor
D	Distance between $s^p$and d^p
$(N_{in}-2)_{C_k}$	Number of paths with k intermediates
N^p	Total number of paths between $s^p$and d^p
N_{in}	Number of intermediate nodes between $s^p$and d^p
FT_i^p	Fault tolerance of i^{th}path
FT_i^l	Fault tolerance of i^{th}link
CD_i^p	Communication delay of i^{th}path
CD_i^l	Communication delay of i^{th}link
CH_i	i^{th}chromosome
N_{re}	Number of retransmissions for a success
$e_{i,j}$	Packet error rate of a link between nodes i and j
$d^l_{i,j}$	Degree estimation of a link between nodes i and j
d_i^e	Degree of i^{th}node
α	Decision variable
$d_{i,j}$	Distance between nodes i and j
S_p	Propagation speed
S_t	Transmission speed
S_{pkt}	Size of packet
s^v and d^v	Virtualized source and destination sensors of a link failure
S^{pop}	Size of chromosome population
N^{gen}	Number of generations during solution optimization

A. The Optimization Problem

Let there is a link failure between two virtual sensors s^v and d^vdue to the path failure embedded for the link. The path for the link in wireless sensor network is between the physical sensor nodes s^p and d^p. All the available paths between s^p and d^p need to be explored to obtain an alternative path between these nodes with maximum fault tolerant capacity. The knowledge of expected number of paths with intermediate node is required for exploring these paths.

To find the total number of paths, expected distance of the nearest neighbor node qualified for communication establishment is computed. The expected distance of closest neighbor sensor can be determined once the probability density function (pdf) of the location of sensors is known. To determine pdf of the location of sensors, the probability of a neighbor sensors between the two distances r and $(r+\Delta r)$ is derived, where r is a distance within transmission radius and Δr represents a small incremental distance.

It is assumed that sensors are distributed across a physical wireless sensor network area N_A with uniform density λ. Therefore, the probability of presence of a sensor in the area is 1, which can be expressed as given by Eq. (1)

$$\int_{N_A} \lambda \, dN_A = 1 \Rightarrow \lambda = \frac{1}{N_A} \tag{1}$$

The probability $P^c_{r|(r+\Delta r)}$ of closest neighbor sensor at the distance between r and $(r+\Delta r)$ is the joint probability of $P^s_{r|(r+\Delta r)}$ presence of some neighbor sensor at this distance, and probability $P^0_{<r}$ of presence of no other sensor closer than the distance r. The probability $P^c_{r|(r+\Delta r)}$ can be expressed as given by Eq. (2)

$$\begin{aligned} P^c_{r|(r+\Delta r)} &= P^0_{<r} \cdot P^s_{r|(r+\Delta r)} \\ &= [1 - P^s_{<r}] \cdot \left[P^s_{r|(r+\Delta r)}\right] \end{aligned} \tag{2}$$

Considering only half of the area in transmission range towards destination d^p with N_n sensor nodes in the network, it is simplified as given by Eq. (3).

$$\begin{aligned} P^c_{r|(r+\Delta r)} &= \left[1 - \sum_{j=1}^{N_n} \binom{N_n}{j} \left(\frac{\lambda\pi r^2}{2}\right)^j \left(1 - \frac{\lambda\pi r^2}{2}\right)^{N_n - j}\right] \cdot \\ &\quad \left[\sum_{j=1}^{N_n} \binom{N_n}{j} \int_r^{r+\Delta r} \left(\frac{2\lambda\pi r \,.\, dr}{2}\right)^j dr \cdot \int_r^{r+\Delta r} \left(1 - \frac{2\lambda\pi r \,.\, dr}{2}\right)^{N_n - j} dr\right] \\ &= (1-\lambda\pi r^2)^{N_n}\left[1 - \left(1 - \lambda\pi(rdr + dr^2)\right)^{N_n}\right] \\ &= (1-\lambda\pi r^2)^{N_n}\left[1 - \left\{1 - \binom{N_n}{1}.\left(\lambda\pi(rdr+dr^2)\right) + \binom{N_n}{2}.\left(\lambda\pi(rdr+dr^2)\right)^2 \ldots\right\}\right] \\ &= (1-\lambda\pi r^2)^{N_n}\left[N_n\lambda\pi r dr + N_n\lambda\pi dr^2 - \binom{N_n}{2}.\left(\lambda\pi(rdr + dr^2)\right)^2 \ldots\right] \end{aligned} \tag{3}$$

The probability density function $f_r(r)$ of closest neighbor distance can be obtained by considering limit in Eq. (3) as:

$$f_r(r) = \lim_{dr\to 0} \frac{P^c_{r|(r+\Delta r)}}{dr} = N_n\lambda\pi r(1-\lambda\pi r^2)^{N_n} \tag{4}$$

Considering R as transmission range of sensors in Eq. (4), the expected closest neighbor distance $E(r)$can be expressed as given by Eq. (5).

$$\begin{aligned} E(r) &= \int_0^R r f_r(r) dr = \int_0^R N_n\lambda\pi r^2(1-\lambda\pi r^2)^{N_n} dr \\ &= \left[\frac{-r(1-\lambda\pi r^2)}{\lambda\pi(N_n+1)}\right]_0^R + \int_0^R \frac{(1-\lambda\pi r^2)^{N_n+1}}{\lambda\pi(N_n+1)} dr \\ &= \left[\frac{1}{\lambda\pi(N_n+1)} \sum_i^{N_n+1} \binom{N_n+1}{i} \frac{(-\lambda\pi r^2)^i r}{i+1}\right]_0^R \\ E(r) &= \frac{\sqrt{N_A}}{\lambda\pi^{3/2}(N_n+1)} \sum_i^{N_n+1} \frac{(-1)^i}{i+1} \end{aligned} \tag{5}$$

Now, the number of paths from s^p to d^p with k number of intermediate nodes is $(N_{in}-2)_{C_k}$ where $k = \left\{1,2,3,\ldots\left(\left\lfloor D/E(r)\right\rfloor - 1\right)\right\}$, total intermediate nodes $N_{in} = \left(\left\lfloor D/E(r)\right\rfloor - 1\right)$ and D represents distance between s^p and d^p. The total number of paths N^p from s^p to d^pcan be expressed as given by Eq. (6).

$$\begin{aligned} N^p &= (N_{in}-2)_{C_1} + (N_{in}-2)_{C_2} + \cdots + (N_{in}-2)_{C_{(N_{in}-2)}} \\ &= \left\{(N_{in}-2)_{C_0} + (N_{in}-2)_{C_1} + \cdots + (N_{in}-2)_{C_{(N_{in}-2)}}\right\} - 1 \\ &= 2^{N_{in}} - 1 \end{aligned} \tag{6}$$

Using the number of paths in Eq. (6), the maximization function for Fault Tolerance (FT) can be expressed as:

$$Maximize\ FT = \max_{i=1,2,\ldots,N^p} (FT_i^p) \tag{7}$$

Where, FT_i^p represents fault tolerance of i^{th} path from s^p to d^p. The normalized fault tolerance of a path can be expressed as:

$$FT_i^p = \frac{1}{\left(\left\lfloor D/_{E(r)}\right\rfloor - 1\right)} \sum_{i=s^p, j=1}^{i=\left(\left\lfloor D/_{E(r)}\right\rfloor - 1\right), j=d^p} FT_{i,j}^l \quad (8)$$

Where, $FT_{i,j}^l$ is the fault tolerance of a link between an adjacent pair of nodes, and $i,j \in S_i^{op} = \{s^p, 1,2, \dots, \left(\left\lfloor D/_{E(r)}\right\rfloor - 2\right), \left(\left\lfloor D/_{E(r)}\right\rfloor - 1\right), d^p\}$. The ordered set of nodes of i^{th} path is represented by S_i^{op}. Similarly, the minimization function for Communication Delay (CD) can be expressed as:

$$Minimize\ CD = \min_{i=1,2,\dots,N^p} (CD_i^p) \quad (9)$$

Where, CD_i^p represents delay of i^{th} path from s^p to d^p. The normalized CD of a path can be expressed as:

$$CD_i^p = \frac{1}{\left(\left\lfloor D/_{E(r)}\right\rfloor - 1\right)} \sum_{i=s^p, j=1}^{i=\left(\left\lfloor D/_{E(r)}\right\rfloor - 1\right), j=d^p} \left(\frac{CD_{i,j}^l}{CD_{max}^l}\right) \quad (10)$$

Where, $CD_{i,j}^l$ is the delay of a link between an adjacent pair of nodes, and $i,j \in S_i^{op}$. The maximum link delay among all the links is represented by CD_{max}^l. The constraints corresponding to the aforementioned optimization problem include $0 < FT_i^p \leq 1, 0 < FT_i^l \leq 1, 0 < CD_i^p \leq 1, 0 < \frac{CD_{i,j}^l}{CD_{max}^l} \leq 1$.

B. Adapted NSGA

An adapted NSGA is developed for solving the multi-objective optimization problem focusing on chromosome representation, fault tolerance and delay computation, cross over and mutation operations, and sorting chromosomes using non-dominance concept.

1) Chromosome Representation

An ordered set of intermediated nodes S_i^{op} starting from source s^p and ending with destination d^p nodes, represents a chromosome in the solution space of A-NSGA considering the optimization problem. Each node of the set represents a gene of the chromosome representation. An i^{th} chromosome can be represented as (see Fig. 3):

$$CH_i = \{s^p, 1,2, \dots, \left(\left\lfloor D/_{E(r)}\right\rfloor - 2\right), \left(\left\lfloor D/_{E(r)}\right\rfloor - 1\right), d^p\}^{FT,CD} \quad (11)$$

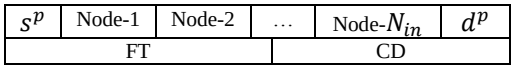


Fig. 3. Genotype representation of chromosome

2) Fault Tolerance and Communication Delay

The fault tolerance of links are utilized to obtain cumulative fault tolerance of a path. The fault tolerance of a link is derived considering packet error rate based link quality estimation, and neighbor density based degree estimation. It can be expressed as given by Eq. (12).

$$FT_{i,j}^l = \left(1 - \sum_{t=0}^{N_{re}} (e_{i,j})^t (1 - e_{i,j})\right) + d_{i,j}^l \quad (12)$$

Where N_{re} is the number of retransmission required for a successful transmission over the link, $e_{i,j}$ is the packet error rate of the link and $d_{i,j}^l$ is the degree estimation of the link. The degree estimation can be derived as expressed by Eq. (13).

$$d_{i,j}^l = \begin{cases} 1, & d_i^e = d_j^e = N_n - 1 \\ 1 - \alpha^{d_i^e}, & d_i^e = d_j^e < N_n - 1 \\ 1 - \alpha^{\frac{\left(d_i^e - d_j^e\right)^2}{d_i^e + d_j^e}}, & \left|d_i^e - d_j^e\right| > 0 \end{cases} \quad (13)$$

where d_i^e and d_j^e are the degrees of nodes i and j, respectively and α is a decision variable varies between 0 and 1.

The communication delay is computed considering interference for the link determined by the link quality, and propagation and transmission delay. It can be expressed as given by Eq. (14).

$$CD_{i,j}^l = \left(1 - \sum_{t=0}^{N_{re}} (e_{i,j})^t (1 - e_{i,j})\right) + \frac{d_{i,j}}{S_p} + \frac{S_{pkt}}{S_t} \quad (14)$$

Where, $d_{i,j}$ is the distance between the pair of nodes i and j, S_p represents propagation speed, S_{pkt} is the packet size and S_t represents transmission speed.

3) Crossover and Mutation

The crossover operation is performed by selecting two chromosomes (also termed as parent solution in optimization theory) from the population (all paths between s^p and d^p), and randomly exchanging a group of nodes between the chromosomes (see Fig. 4). The exchange is constrained to the reachability of the nodes from both downward and upward direction in the chromosome. The larger group size is considered in initial stage of the solution (at lower generations) whereas, smaller group size is preferred in latter stages. Thus, size of the group for crossover operation is based on the generation number and size of the chromosome pair. Due to the possibility of repetition of intermediate nodes, the chromosome after crossover operation (also termed as offspring in optimization theory) are repaired. The intermediate nodes present in the parent chromosome but not in the newly generated offspring, are considered while repairing the offspring.

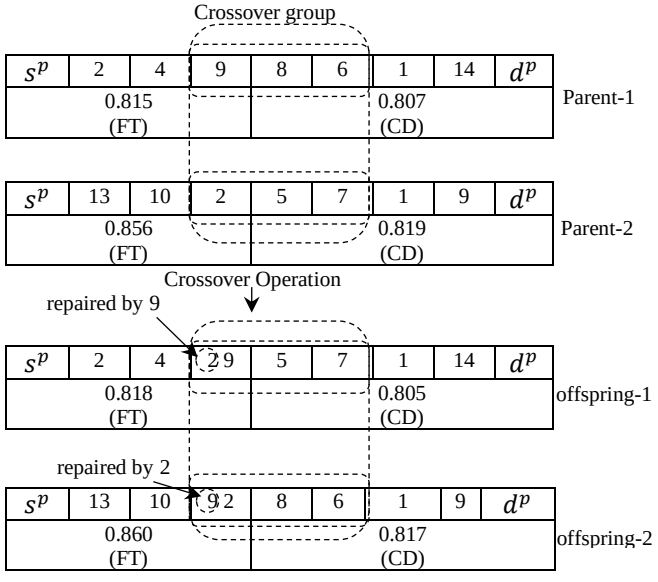


Fig. 4. The crossover operation

In the mutation operation, the order of two randomly selected nodes is exchanged in the chromosome, if the nodes are reachable (present as neighbor) from their downward nodes. Illustrative example of crossover and mutation operation is shown in Fig. 4 and 5, respectively.

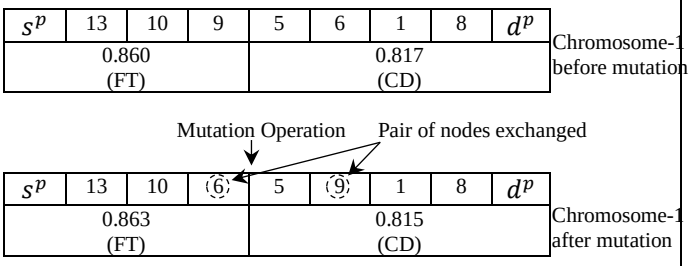


Fig. 5. The mutation operation

4) Non-dominance based Sorting for Chromosomes

The concept of non-dominance is used for sorting population of chromosomes. The sorting technique prioritizes chromosomes based on multiple conflicting objectives. Let us consider, two chromosomes $CH_i$and CH_j from the population. According to Pareto optimal, chromosome CH_i is said to dominate CH_j if at least one of the objectives fitness value of CH_i is better than the fitness value of CH_j, and the fitness value of the other objectives are equivalent. Parato optimal based prioritization is preferred in multi-objective optimization use cases in communication networks [26, 27]. For two objectives, it can be derived as:

$$CH_i > CH_j = \begin{cases} CH_i(FT) > CH_j(FT), \wedge\ CH_i(CD) \not< CH_j(CD) \\ CH_i(CD) > CH_j(CD), \wedge\ CH_i(FT) \not< CH_j(FT) \end{cases} \quad (15)$$

The chromosomes of the population are ranked in ascending order by comparing their fitness values using non-dominance concept. The chromosomes not dominated by any other chromosomes are ranked 1st in the population. The 2nd rank is assigned to the chromosomes dominated by only one chromosome in the population. The chromosomes dominated by two other chromosomes are ranked 3rd in the population. The crowding distance is calculated for each chromosomes of the population after ranking of the chromosome. Tournament selection approach is utilized to select population for the next generation.

5) Adapted NSGA

The complete set of steps utilized for solving the optimization problem is presented in Algorithm 1.

Algorithm 1: A-NSGA

Notations: $l_{S,D}$: Link between s^p and d^p; F_i^{path}:Fault tolerance of i^{th} path
D_i^s : Dominance set of i^{th} solution; S_i: i^{th} solution of the population
np =This is the number of solutions that dominate p; F_j: j^{th} front
$S^{child-pop}$: Size of child population; R_i: Rank of i^{th} solution
g : Number of generations used for looping; S^{pop}: Size of the old population considered for execution; C^{dist}: Crowding distance
N^{gen}: Number of generations; old^{pop}: Old population; $child^{pop}$: Child population;

Input: $l_{S,D}, F_i^{path}, S^{pop}, N^{gen}$

Process:

1. **Generate initial population of size** S^{pop} by random distribution of decision variable in given range (low, high). Save one copy of population as old^{pop}
2. **for each** $S_i \in old^{pop}$
   a. **Calculate** objective-1 normalized fault-tolerance using Eq. (8)
   b. **Calculate** objective-2 normalized delay using Eq. (10)
   endfor
   $g = 1$
3. **While** $(g \leq N^{gen})$
   Non-dominated_sorting(old^{pop}) // function for Non-Dominated sorting
4. **for each** $S_i \in old^{pop}$
   Calculate D_i^s
   endfor
   $j = 1,$
5. **For each** $S_i \in old^{pop}$
   If $(D_i^s = \phi)$
   $F_j = F_j \cup S_i$
   $R_i = 1$
   Endif
   Endfor
   $j = 2$
6. **For each** $S_i \in old^{pop}$
   If $(D_i^s \neq \phi \;\&\&\; R_i == j-1)$
   $F_j = F_j \cup S_i$
   $R_i = j$
   $j = j + 1$
   Endif
   Endfor // end of function Non-dominated_sorting
   Crowing_distance (old^{pop}) // function for crowing distanc
   Assume C^{dist} from boundary point (group of solution) to ∞ for any solution
7. **for each** $S_i \in old^{pop}$ // The crowing distance Start
   calculate C^{dist} from all point excluding boundary points
   endfor // end of crowding distance function
8. **Select** the best half population as $parent^{pop}$ considering R and C^{dist} using tournament selection approach.
   $child^{pop} = \Phi$
   $S^{child-pop} = 0$
9. **While**($S^{child-pop} \leq S^{pop}$)
   Randomly select two chromosomes from the parent population
   Perform crossover to produce two child chromosomes
   Update $child^{pop}$ and $S^{child-pop} = S^{child-pop} + 2$
   Randomly choose a chromosome from parent population
   Mutate chromosome to produce a child chromosome
   Update $child^{pop}$ and $S^{child-pop} = S^{child-pop} +$
   endwhile
10. **Generate** new population of size $(2\times S^{pop})$ by $old^{pop} \cup child^{pop}$
11. **Calculate** normalize fault-tolerance using Eq. (8)
12. **Calculate** normalized delay using Eq. (10)
13. **Non-dominated_sorting(** $old^{pop} \cup child^{pop}$)
14. **Crowing_distance (**$old^{pop} \cup child^{pop}$)
15. **Select** again the best half population as old^{pop} using rank and C^{dist}
    endwhile
16. **exit**

Output: optimized chromosomes

Explanation of A-NSGA

In step 1 and 2, population (paths from source and destination pair) is generated and initialized. The population is sorted following non-dominance based sorting in steps 3-6. The best half population is selected as parent in steps 7-8. In step 9, crossover and mutation operations are performed for generating better solutions from selected parent population. In steps 10-15, better half population is again selected from the combined (old and newly generated) population. These steps are repeated until the condition mentioned in step 3 satisfies (the predefined maximum number of generations). These steps are presented as flowchart in Fig. 6. The time complexity of A-NSGA is $O(2\times S^{pop}\times N^{gen})$, where S^{pop} is size of population and N^{gen} represents the number of generations.

The size of the network indirectly controls the number of generations, and thus execution time. Also, time require for each generation might vary, and depend on the hardware configuration of the system.

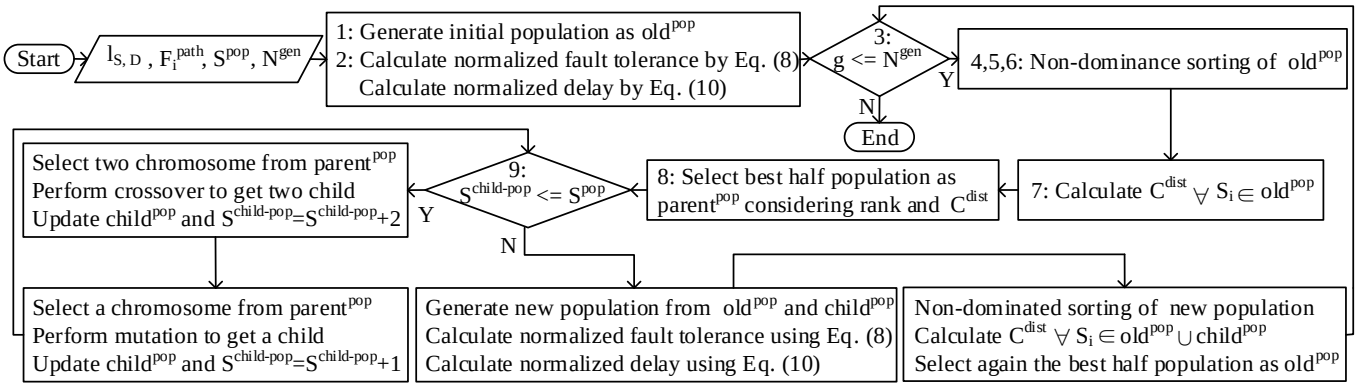


Fig. 6. Flowchart of A-NSGA

IV. Empirical Results

In this section, analytical, case study based simulation results are discussed for analyzing the performance of the proposed fault tolerance optimization framework for IoT. It is broadly divided into two parts. Analytical results are discussed in the first part, whereas simulation setting, metrics and comparative analysis of simulation results are discussed in the second part.

A. Analytical Results

The characteristics of the mathematical derivations are analytically analyzed using mathematical tool. The analytical analysis evaluates the performance of the mathematical formulations obtained for solving the optimization problem in IoT environments. Most of the parameters considered for generating analytical results are pointed in the results itself. The increment on network area has exponential impact on expected nearest node distance. The increment in number of paths with the increase in number of intermediate vehicles is initially slower but continuous and approximately exponential nature with higher intermediate nodes (see Fig. 7 (a) and (b)).

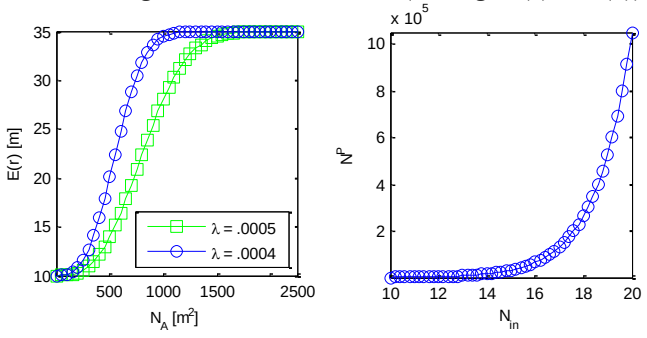


Fig. 7. Analytical results, (a) impact of N_A on $E(r)$, (b) impact of N_{in} on N^p

B. Simulation Results

In this section, simulations carried out to evaluate the performance of the framework is presented focusing on environment setting, metrics, and comparative analysis of results. The two objectives were aimed for performing case study based simulations. Firstly, effectiveness of fault tolerant optimization is measured considering the impact of number of generations on optimization. Secondly, efficiency of fault tolerant optimization is measured considering network density.

1) Simulation Setting and Metrics

The proposed optimization of fault tolerance and delay in virtual network is implemented in network simulator NS-2 using C++ programing language for implementing major classes of the simulation. The major classes of the simulation include 'NetworkNode', 'VirtualNode' 'RandomProvider', 'PathSearchNSGA' and 'MainApp'. All the characteristics of a node in a network such as position, list of neighbors, link delay with neighbors, fault tolerance of associated links are implemented in 'NetworkNode'. Interface based task processing is implemented at 'VirtualNode', For different simulation runs, different set of network nodes are randomly generated using 'RandomProvider'. The optimization of fault tolerance and delay for generating virtual network is implemented in 'PathSearchNSGA'. Two prototype IoT applications are implemented in 'MainApp' class. The simulation is performed in a machine having Intel Core i7-2500S 2.70 GHz processor, 16GB RAM, 64bit Linux OS. The other basic setting of parameters in simulation is approximately similar the parameter table and setting considered in [28, 29]. Three different sets of network with 100, 500 and 1000, 1500, 2000 nodes are generated following Poisson process. The adapted NSGA is executed up to 500, 1000 and 1500, and 2000 generations to optimize fault tolerance and communication delay in four deferent networks. The chromosomes of the last generation in results which represents the last optimized values.

2) Analysis of Results

A comparison of optimization performance between A-NSGA and SVNE is shown in Fig. 8(a)-(d) considering 100 nodes and 500~800 generations. It can be clearly observed that the optimization performance of A-NSGA is better as compared to SVNE for both the objectives including fault tolerance and communication delay. The observation affirms the capability of handling failure in virtualization of WSNs. Specifically, the optimized value of fault tolerance is approximately 0.65 whereas the optimized value of delay is approximately 0.02. This can be attributed to the accurate

prediction of fault tolerance using packet error rate based link quality estimation. In case SVNE, optimized value of fault tolerance is approximately 0.3 whereas the optimized value of delay is approximately 0.2. This is due to the degree of connectivity based fault tolerant estimation. The estimation is not suitable in wireless scenario. Moreover, large number of chromosomes has higher delay and lower fault tolerance values. Also, the impact of increment of number of generations on the optimized chromosome is quite lower due to the smaller network (100 nodes). As the difference between the successive results ((a)-(d)) is not clearly visible. This is due to the lesser number of path formation in smaller networks.

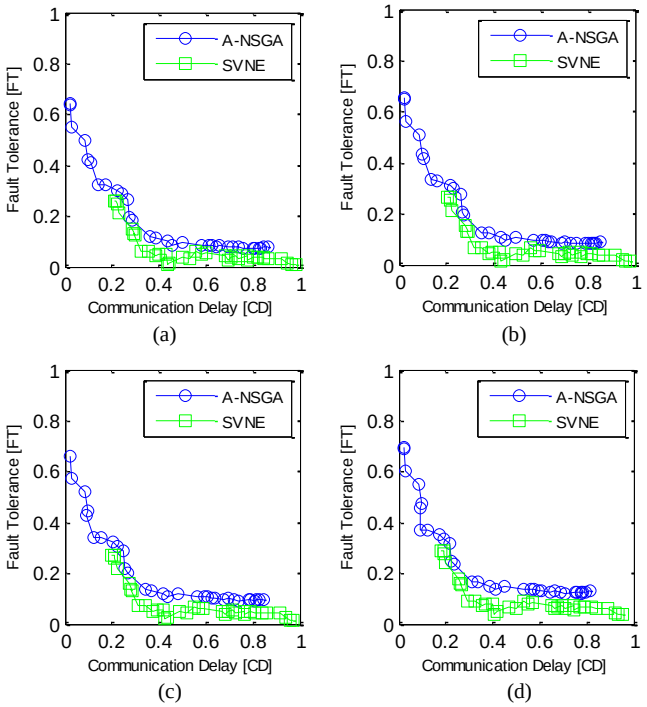


Fig. 8. Optimized chromosome with 100 nodes after (a) 500 (b) 600, (c) 700, (d) 800 generations

The network size is enlarged considering 500 nodes for magnifying the optimization performance difference in successive generations. A comparison of optimization performance the larger network size is shown in Fig. 9(a)-(d). It is evident from results that the optimization performance of A-NSGA is better as compared to SVNE considering both the objectives. In particular, the last optimized chromosome value of fault tolerance is approximately 0.8 whereas, the optimized chromosome value of communication delay is approximately 0.015. This can be attributed to the availability of higher number of paths with larger networks which helps in selection better quality links, with greater fault tolerance nad lower communication delay. The optimized chromosome value of fault tolerance is approximately 0.3 whereas the optimized value of delay is approximately 0.15. This is due to the slower convergence of the wired network based approach. Moreover, the number of optimized chromosomes has still lesser and the convergence rate towards optimal solution is quite lower. Also, the impact of increment of number of generations on the optimized chromosome is little bit better due to the larger network (500 nodes). As the difference between the successive results ((a)-(d)) is more visible.

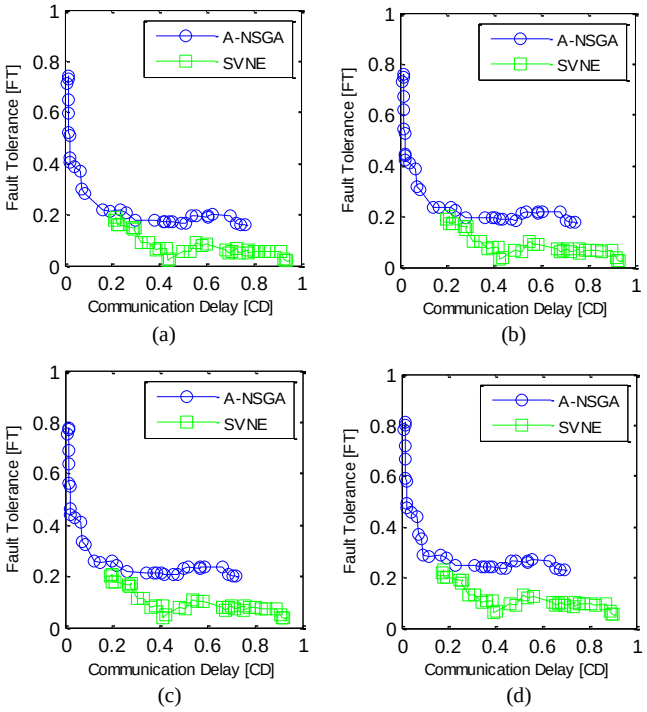


Fig. 9. Optimized chromosome with 500 nodes after (a) 500 (b) 600, (c) 700, (d) 800 generations

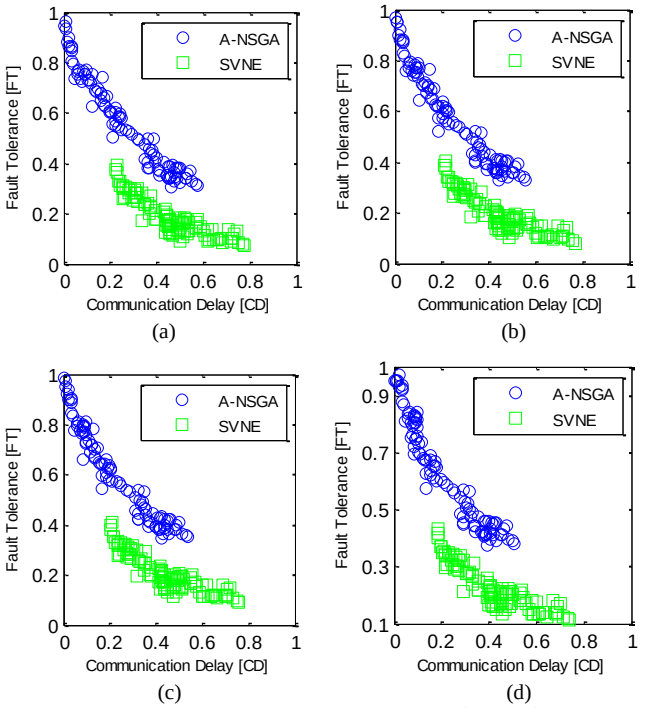


Fig. 10. Optimized chromosome with 1000 nodes after (a) 500 (b) 600, (c) 700, (d) 800 generations

The network size is further enhanced with 1000 nodes to improve the convergence rate towards optimal solution. A comparison of optimization convergence rate is shown in Fig. 10(a)-(d). It is evident from results that the optimization convergence rate of A-NSGA is better as compared to SVNE for both the objectives. The optimized chromosome value of

fault tolerance is approximately 0.98 whereas, the optimized chromosome value of communication delay is approximately 0.010. This can be attributed to the better felt tolerant path selection from the large number of available paths with scaled network size. In case of SVNE, the optimized chromosome value of fault tolerance is approximately 0.48 whereas the optimized value of delay is approximately 0.13. Moreover, the number of optimized chromosomes has significantly increased with scaled network size. The chromosomes are densely clustered for both approaches. The impact of increment of number of generations on the optimized chromosome is clearly visible in successive results ((a)-(d)) due to the greater availability of paths.

C. Summary of Observations

The optimization of fault tolerance in virtualization WSNs significantly improves the performance of IoT applications, requiring heterogeneous network architecture. The problem can be modelled as multi-objective optimization. The optimization problem can be solved using A-NSGA. The number of paths between source and destination pair increases exponentially with the increase of network density The fault tolerance can be represented as link quality and density of neighbor nodes in wireless network environments. The analysis of case study based simulation results attests the effectiveness of the optimization framework in handling failure of virtual networks. The convergence rate of the optimal solution is quite lower with smaller network size (100 nodes). The impact of number of generations on optimization of solutions is more visible with larger network size (1000 nodes), due to the higher number of paths. The optimized chromosomes are higher in numbers and densely clustered with scaled network size. Therefore, the fault tolerant approach is scalable, and suitable for the scaled network environments of IoT.

V. Conclusion and Future Work

In this paper, a framework for optimizing fault tolerance in virtualization in WSNs is presented focusing on heterogeneous network requirement for IoT applications. A multi-objective optimization problem is mathematically formulated considering fault tolerance and communication delay in virtualization. An A-NSGA is developed for solving the optimization problem. The optimization framework is more effective as compared the state of the art approaches. It is evident from the better optimization results obtained with lower number of generations. The optimization results are also obtained in shorter time as compared to the state of the art approaches. This affirms the efficiency of the proposed framework. In future research work, authors will consider more network parameter in the multi-objective optimization problem.

References

[1] Xu, K., Qu, Y. and Yang, K., 2016. A tutorial on the internet of things: from a heterogeneous network integration perspective. IEEE Network, 30(2), pp.102-108.

[2] Guo, Y., Zhu, H. and Yang, L., 2016. Service-oriented network virtualization architecture for Internet of Things. China Communications, 13(9), pp.163-172.

[3] Mosharaf, C.; Muntasir, R. R. and Raouf, B; "ViNEYard: Virtual Network Embedding Algorithms with Coordinated Node and Link Mapping", IEEE/ACM Transactions on Networking," vol. 20, no.1, pp. 206-219, 2012.

[4] Fischer, A.; Botero, J.F.; Till Beck, M.; de Meer, H.; Hesselbach, X.; "Virtual Network Embedding: A Survey," IEEE Communications Surveys & Tutorials, vol.15, no.4, pp.1888-1906, 2013.

[5] Zhang, S., Qian, Z., Wu, J., Lu, S. and Epstein, L., 2014. Virtual network embedding with opportunistic resource sharing. IEEE Transactions on Parallel and Distributed Systems, 25(3), pp.816-827.

[6] Guo, B., Qiao, C., Wang, J., Yu, H., Zuo, Y., Li, J., and He, Y., 2014. Survivable virtual network design and embedding to survive a facility node failure. Journal of Lightwave Technology, 32(3), pp.483-493.

[7] Ishaq, I., Hoebeke, J., Moerman, I. and Demeester, P., 2012, Internet of things virtual networks: Bringing network virtualization to resource-constrained devices. In Green Computing and Communications (GreenCom), 2012 IEEE International Conference On (pp. 293-300).

[8] Khan, I., Belqasmi, F., Glitho, R., Crespi, N., Morrow, M. and Polakos, P., 2015. Wireless Sensor Network virtualization: Early architecture and research perspectives. IEEE Network, 29(3), pp.104-112.

[9] Toyonaga, S., Kominami, D. and Murata, M., 2016. Virtual Wireless Sensor Networks: Adaptive Brain-Inspired Configuration for Internet of Things Applications. Sensors, 16(8), p.1323.

[10] Guo, B., Qiao, C., Wang, J., Yu, H., Zuo, Y., Li, J., ... & He, Y. (2014). Survivable virtual network design and embedding to survive a facility node failure. Lightwave Technology, Journal of, 32(3), 483-493.

[11] Bo, L. U., HUANG, T., SUN, X. C., CHEN, J. Y., & LIU, Y. J. (2014). Dynamic recovery for survivable virtual network embedding. The Journal of China Universities of Posts and Telecommunications, 21(3), 77-84.

[12] Cui, H; Tang, S.; Sun, F.; Xu, Y.; Yang, X.; "Topological Embedding Feature Based Resource Allocation in Network Virtualization," Mathematical Problems in Engineering, vol. 2014, no.1, pp. 1-10, 2014.

[13] Rahman, M. R. and Boutaba, R.; "SVNE: Survivable Virtual Network Embedding Algorithms for Network Virtualization," IEEE Transactions on Network and Service Management, vol.10, no.2, pp.105-118, 2013.

[14] Hu, Q., Wang, Y., & Cao, X. (2013). Survivable network virtualization for single facility node failure: A network flow perspective. Optical Switching and Networking, 10(4), 406-415.

[15] Liao, J.; Feng, M.; Li, T.; Wang, J.; Qing, S.;"Topology-aware Virtual Network Embedding Using Multiple Characteristics," KSII Transactions on Internet and Information Systems (TIIS), vol. 8, no. 1, pp. 145-164, 2014.

[16] Hu, Q., Wang, Y., & Cao, X. (2012, May). Location-constrained survivable network virtualization. In Sarnoff Symposium (SARNOFF), 2012 35th IEEE (pp. 1-5). IEEE.

[17] Yuan, Y.; Wang, C.; Zhang, C.; Zhu, N.; "Fault Tolerant Virtual Network Embedding Algorithm Based on Redundant Backup Resource," In Proceedings on the Third International Conference on Instrumentation, Measurement, Computer, Communication and Control (IMCCC), pp. 354-357, 21-23 September 2013.

[18] Xiu-Xiu, W., & Wei, J.; "A Hybrid Reliable Heuristic Mapping Method Based on Survivable Virtual Networks for Network Virtualization," Discrete Dynamics in Nature and Society, vol. 2015, no. 1, pp. 1-8, 2015.

[19] Chang, X. L., Mi, X. M., & Muppala, J. K. (2013). Performance evaluation of artificial intelligence algorithms for virtual network embedding. Engineering Applications of Artificial Intelligence, 26(10), 2540-2550.

[20] Khan, I., Belqasmi, F., Glitho, R., Crespi, N., Morrow, M. and Polakos, P., 2015. Wireless sensor network virtualization: A survey. IEEE Communications Surveys & Tutorials, 18(1), pp.553-576.

[21] Islam, M.M., Hassan, M.M., Lee, G.W. and Huh, E.N., 2012. A survey on virtualization of wireless sensor networks. Sensors, 12(2), pp.2175-2207.

[22] Kim, S., 2016. Game Based Certificate Revocation Algorithm for Internet of Things Security Problems. Adhoc & Sensor Wireless Networks, 32(1), pp.319-336.

[23] Shah, R.R., Hefeeda, M., Zimmermann, R., Harras, K., Hsu, C.H. and Yu, Y., 2016. NEWSMAN: Uploading Videos over Adaptive Middleboxes to News Servers in Weak Network Infrastructures. In Proceeding of the International Conference on Multimedia Modeling, pp.100-113. Florida USA.

[24] Cao, Y., Kaiwartya, O., Wang, R., Jiang, T., Cao, Y., Aslam, N. and Sexton, G., 2017. Toward Efficient, Scalable, and Coordinated On-the-Move EV Charging Management. IEEE Wireless Communications, 24(2), pp.66-73.

[25] Calhan, A., 2016. A Non-Preemptive Priority Scheduling Algorithm for Improving Priority Data Transmission Delay in Wireless Body Area Networks. Adhoc & Sensor Wireless Networks, 34(1), pp. 59-75.
[26] Kaiwartya, O., Kumar, S., Lobiyal, D.K., Tiwari, P.K., Abdullah, A.H. and Hassan, A.N., 2015. Multiobjective dynamic vehicle routing problem and time seed based solution using particle swarm optimization. Journal of Sensors, 15(1), pp.1-14.
[27] Zhou, B. and Chen, Q., 2016. On the Particle-Assisted Stochastic Search Mechanism in Wireless Cooperative Localization. IEEE Transactions on Wireless Communications, 15(7), pp.4765-4777.
[28] Kaiwartya, O., Abdullah, A.H., Cao, Y., Raw, R.S., Kumar, S., Lobiyal, D.K., Isnin, I.F., Liu, X. and Shah, R.R., 2016. T-MQM: Testbed-Based Multi-Metric Quality Measurement of Sensor Deployment for Precision Agriculture—A Case Study. IEEE Sensors Journal, 16(23), pp.8649-8664.
[29] Zhou, B., Chen, Q. and Xiao, P., 2017. The Error Propagation Analysis of the Received Signal Strength-based Simultaneous Localization and Tracking in Wireless Sensor Networks. IEEE Transactions on Information Theory, 63(6), pp.3983-4007.



Omprakash Kaiwartya (M'15) received his Ph.D., degree in Computer Science from School of Computer and Systems Sciences, Jawaharlal Nehru University, New Delhi, India in 2015. He is currently a Postdoctoral Research Fellow at Faculty of Computing, Universiti Teknologi Malaysia (UTM), Johor Bahru, Malaysia. He has recently selected as Visiting Scholar at the Department of Computer and Information Sciences, Northumbria University, UK. His research interests focus on Internet of Vehicles (IoV), Electronic Vehicles (EV), IoT use case of Wireless Sensor Networks (WSNs).



Abdul Hanan Abdullah (M'15) received his Ph.D. degree from Aston University in Birmingham, United Kingdom in 1995. He is currently working as a Professor and heading Pervasive Computing Research Group at Faculty of Computing, Universiti Teknologi Malaysia, Johor Bahru, Malaysia. He was the Dean of the faculty from 2004-20011. His research interests include Wireless Sensor Networks, Vehicular Adhoc Networks, Internet of Things, Network Security and Next Generation Networks.



Yue Cao (M'16) received his Ph.D. degree from the Institute for Communication Systems (ICS) formerly known as Centre for Communication Systems Research, at University of Surrey, Guildford, UK in 2013. Further to his PhD study, he was a Research Fellow at the ICS. Currently, he is working as Lecturer in Department of Computer and Information Sciences, Northumbria University, UK. His research interests focus on Delay/Disruption Tolerant Networks, Electric Vehicle (EV) charging management, Information Centric Networking (ICN), Device-to-Device (D2D) communication and Mobile Edge Computing (MEC).



Jaime Lloret (M'07, SM'10) received his M.Sc. in Physics in 1997, his M.Sc. in electronic Engineering in 2003 and his Ph.D. in telecommunication engineering in 2006. He is currently Associate Professor in the Polytechnic University of Valencia. He is currently Chair of the Internet Technical Committee (IEEE Communications Society and Internet society) and Working Group of the Standard IEEE 1907.1. He has authored 22 book chapters and has more than 340 research papers published in national and international conferences, international journals (more than 110 with ISI). He has been the co-editor of 38 conference proceedings and guest editor of several international books and journals. He is editor-in-chief of the Ad Hoc and Sensor Wireless Networks, and associate editor of 46 international journals (16 ISI). He has been involved in more than 320 Program committees of international conferences and more than 130 organization and steering committees. He has been general chair of 28 International workshops and conferences.



Sushil Kumar (M'14) received his Ph.D. degree in Computer Science from School of Computer and Systems Sciences, Jawaharlal Nehru University, New Delhi, India in 2014. He is currently working as Assistant Professor at the Jawaharlal Nehru University, India. His research interest includes Vehicular Ad-hoc Networks, and Wireless Sensor Networks.



Rajiv Ratn (M'16) Shah received his Ph.D. from School of Computing, National University of Singapore (NUS) in 2016. Currently, he is a Postdoctoral Research Fellow at the School of Information Systems, Singapore Management University (SMU). His main research interests focus on multimodal analysis of user-generated content in support of social media applications, location-based services in support of social media applications, multimedia analysis and retrieval.



Dr. Mukesh Prasad (M'13) received his Ph.D. degree in computer science from National Chiao Tung University, Hsinchu, Taiwan in 2015. He is currently a Lecturer at School of Software, University of Technology Sydney, Australia. He was principle engineer (R&D) with Taiwan Semiconductor Manufacturing Company, Hsinchu, Taiwan. His research interests include machine learning, pattern recognition, fuzzy systems, neural networks, artificial intelligence and brain computer interface.



Shiv Prakash received his Ph.D. in Computer Science from School of Computer and System Sciences, Jawaharlal Nehru University, New Delhi, India in 2014. Currently, he is working as Assistant Professor at the Indian Institute of Information Technology, Kottayam, India. His research interests include parallel and distributed system, cloud computing, artificial intelligence and machine learning.