Hindawi
Mobile Information Systems
Volume 2017, Article ID 3273917, 17 pages
https://doi.org/10.1155/2017/3273917

Research Article

SACA: Self-Aware Communication Architecture for IoT Using Mobile Fog Servers

Vishal Sharma,[1] Jae Deok Lim,[2] Jeong Nyeo Kim,[2] and Ilsun You[1]

[1]*The Department of Information Security Engineering, Soonchunhyang University, Asan-si 31538, Republic of Korea*
[2]*Electronics and Telecommunications Research Institute, Daejeon, Republic of Korea*

Correspondence should be addressed to Ilsun You; ilsunu@gmail.com

Received 19 January 2017; Revised 19 February 2017; Accepted 26 February 2017; Published 11 April 2017

Academic Editor: Eric Rondeau

Internet of things (IoT) aims at bringing together large business enterprise solutions and architectures for handling the huge amount of data generated by millions of devices. For this aim, IoT is necessary to connect various devices and provide a common platform for storage and retrieval of information without fail. However, the success of IoT depends on the novelty of network and its capability in sustaining the increasing demand by users. In this paper, a self-aware communication architecture (SACA) is proposed for sustainable networking over IoT devices. The proposed approach employs the concept of mobile fog servers which make relay using the train and unmanned aerial vehicle (UAV) networks. The problem is presented based on Wald's maximum model, which is resolved by the application of a distributed node management (DNM) system and state dependency formulations. The proposed approach is capable of providing prolonged connectivity by increasing the network reliability and sustainability even in the case of failures. The effectiveness of the proposed approach is demonstrated through numerical and network simulations in terms of significant gains attained with lesser delay and fewer packet losses. The proposed approach is also evaluated against Sybil, wormhole, and DDoS attacks for analyzing its sustainability and probability of connectivity in unfavorable conditions.

1. Introduction

Sustainable communication is one of the key demands of network devices. With a large number of network devices making continuous requests, it becomes important to provide architectural support for continuous connectivity. With a tremendous growth in the number of devices already observed and expected in the near future, sustainable and context-aware communication is in high demand [1].

Internet of things (IoT) has bridged the gap between the network technology and the devices operating over it. Most of the devices are able to utilize the network as a service for sharing data between them. Availability and easy access are the critical issues to be handled with a high number of devices operating on a common platform for service capturing [2, 3]. Infrastructure and network play a key role in providing services to all the devices. A novel architecture can enhance the session quality and can provide prolonged connectivity even in the nonsupporting conditions such as failures, network breakdowns, or security breaches.

With a demand of efficient data management, continuous connectivity, security, and context-aware service provisioning, it becomes important to provide an architecture which can maintain connections between nodes [4, 5]. Most of the authors have defined sustainability according to the domain and area of application such as IoT for water management [6]. Use of artificial intelligence and machine learning approaches can be two of the key solutions for sustainable IoT. Efficient networks can provide enhanced management and control over the devices, whereas a slight irregularity in the network can make it vulnerable to many issues such as privacy, trust, and session hijacking [7–9]. A novel network can provide a common solution for sustainable IoT and can form the backbone of most of the approaches. There are several approaches that aim at the formation of sustainable IoT by considering the service-level solutions, which are specific to a particular domain. However, the success of such approaches depends on the novelty in network layouts. This is an inefficient way of enhancing connectivity until the underlying network is not robust enough to support the service solutions. Thus,

in order to overcome the issue of network breakdowns for sustainable IoT, a novel architecture is proposed in this paper, which forms an intelligent solution for node management. The proposed approach uses a key concept of fog computing but in a novel way by placing fog servers on a train network.

Fog computing/fogging is an innovative nomenclature given to the near user cloud to reduce the latency involved in the flow of data [10]. With major of its properties derived from cloud computing, fogging also utilizes the key concept of mobile clouds [11]. However, it should not be confused with the mobile cloud operations, since mobile cloud aims at utilizing the mobile devices as a platform for implementing cloud applications which are handled as batches, whereas, in this paper, mobile fog computing refers to placing fog servers on fast-moving platforms which can handle the demand for continuous connectivity irrespective of the number of users.

A multitier architecture is proposed in this paper, which provides a self-aware communication setup for handling services across the network. The proposed approach uses unmanned aerial vehicles (UAVs) as intermediate flying routers between the fixed on-ground nodes to provide immediate and efficient connectivity. UAVs can fly in controlled as well as autonomous formations [12, 13]. UAVs have already proven their utility in the upcoming networks and can provide coverage over the large areas [14–17]. Although server computations can be performed on the aerial vehicles, this requires consideration of payload which is a constraint in the utilization of UAVs for operations that require heavy equipment.

Train networks form the key part of the proposed solution which is the actual near user site for placing the fog servers and is an intermediate between the user network and the core network. A distributed node management (DNM) module is used by station terminals for managing all the network nodes considering a state diagram which is defined over the order of connectivity.

The proposed self-aware communication architecture (SACA) is a multimodular hybrid network approach which utilizes the train network and UAV network to exploit the service-guaranteeing features of upcoming 5G networks. Further, sensor feeding, sensor signatures, and content-based server allocation policies are used for managing the load by forming optimization problems using Wald's maximum model [18]. The key contributions of the proposed solution are listed as follows:

(i) A novel hybrid architecture comprising train and UAV networks using the concept of fogging

(ii) Sensor signatures and content-based server allocation for load balancing

(iii) Highly reliable and sustainable network formation even during node/link failures as well as during network attacks

(iv) Intelligent decision-making in the case of network threats and attacks using network state dependency and DNM

The rest of the paper is organized as follows: Section 2 presents the related work. Section 3 presents the motivation and problem statement. Section 4 gives the details of proposed work along with the theoretical analyses. Section 5 evaluates the performance of the proposed approach. Section 6 presents discussions, open issues, and comparison with the existing state-of-the-art approaches. Finally, Section 7 concludes the paper.

2. Related Work

Internet of things aims at providing connectivity to all and connectivity on the go. With a large number of devices generating a huge amount of data and service requests, it becomes important to provide a sustainable strategy that can withstand such tremendous demand for connectivity. Over the years, a lot of attempts have been made for designing sustainable architectures and models to support a large number of IoT devices.

2.1. Sustainable IoT. IoT has made life much easier for humans but complex for the devices and technology handling it. Handling a large number of requests, data dropouts, and security issues and connection stability are the key metrics for defining the sustainable IoT [2, 19]. Many architectures and models exist which have utilized one or other features to provide prolonged connectivity between the IoT devices and network infrastructure. Some of them have focused on device-to-device approach [20], while others emphasized on device-to-infrastructure-to-device methodology [21, 22].

Riedel et al. [23] used web-service gateways along with the code generation to enhance the sustainability of IoT networks. However, depending heavily on the client side gateway can add up to the issues of congestion in a network comprising a large number of simultaneously operating devices. Designing of efficient systems on a chip for providing energy efficient connectivity can also provide sustainable IoT [1]. El Kaed et al. [4] developed a semantic query system for industrial IoT. The authors utilized the concept of semantic tagging of products for making a sustainable IoT for industrial applications. However, the primary focus of their approach is in the selection of gateways which ignores other key factors such as reliability and fault-tolerance.

2.2. Sustainable Fogging. Cloud and fog computing based IoT can provide a vast range of applications using the service selection strategies [24, 25]. The operability of an efficient fog computing environment depends on the efficient policy formation [26]. With a focus on the application-oriented near user cloud formation, policy driver architectures can provide sustainable connectivity. Embedding existing wireless sensor solutions into the cloud environment allows the formation of an efficient fog computing environment [27].

Chen [28] considered a food chain as a cyberphysical system and proposed an intelligent approach for food traceability using the concept of fog computing. The author proposed an architecture that can handle the dynamics involved in food traceability. Luan et al. [29] defined the credibility of fog computing in bridging the gap between the mobile applications and the cloud computing. The authors emphasized the virtual resource utilization and location-based

service allocation as important aspects in building fog environments.

Okay and Ozdemir [35] defined a fog computing model for smart grids. The authors proposed a model which acts as a pivot between the cloud environment and smart grids. Their work focused on laying down the key points required in the formation of sustainable fog architecture, such as latency, self-healing, adaptability, security, and proximity. Tang et al. [36] developed a hierarchical architecture for analyses of big data systems in smart cities. Their architecture aimed at combining the multiple components of smart cities together to perform experimental evaluations of the collected data using event-driven systems. The existing solutions are application specific implementation of the sustainable fog computing having a limited scope in scalability for generic implementations.

Apart from the above approaches, utilizing service as a component of fog computing is an important paradigm in defining reliability and sustainability. Al Faruque and Vatanparvar [37] presented energy management as a service over fog computing. Energy efficient approaches can provide a stable solution for managing the services across the network. However, dependency only on the energy as a paradigm may allow the network to operate for a longer duration but cannot guarantee continuity in the case of threats and node failures. Further, web applications can be improved by provisioning of intelligent and efficient fog environment as well as smart gateways by utilizing the edge cloud architecture over fogging [38, 39].

2.3. Train Networks. Train networks are predefined and periodically configured networks which relay data utilizing the access points on the stations and antennas over the trains [40]. Trains allow support for heavy traffic as large equipment can easily be deployed over them [41, 42]. With a fixed route and path, a periodical approach can help to sustain the connectivity over the train networks. However, speed and handovers are the keys constraints for utilizing the train networks for crucial data-sustaining applications [43–45].

Train network provides a sustainable topology which does not change very often and the route of the trains is changed occasionally [46]. Such property allows ease of governance over networks. Further, with a predefined movement, it is easier to localize the servers placed on trains, which provides a controlled facility movement across the entire network. Such key aspects make train networks suitable for mobile server applications.

Trains can be used as a pivot for placing servers which can provide the facility of fog computing on the move [47]. Vehicle-based fog computing can be readily applied to the train network since this provides better stability and topology control over the entire network [48]. Apart from the advantages of using train networks, it is important to fix a location of the off-site server which will interact with the train servers for connectivity. A central or distributed control authority is also required which can keep a track of server activities as well as the alterations in the topology of the train. Such servers are the intermediate access points in connecting the train servers to the outer network.

3. Motivation and Problem Statement

The information and communication technologies have seen a tremendous growth in the number of users over the last decade. With an exponential increase in the number of users across different platforms, continuity of services and provisioning of quality is of utmost importance. Connectivity between almost all the devices over the network demands service providers to facilitate fast and efficient data processing. More users generate a large amount of data for transmission over the internet, which causes a huge overhead. In the recent years, a solution to such problem is provided in the form of fog computing, which aims at the formation of private and personalized cloud near the user, which decreases the latency involved in the data-sharing over the internet. Although it can provide an efficient solution for latency, it cannot help in sustaining the continuously increasing demand of users. Amendments are required either in the entire network layout or in the data-handling strategies, which can provide a scalable and sustainable approach for connecting a large number of IoT devices with low complexity.

The problem deals with the enhancement of connectivity between the IoT devices along with the distribution of load appropriately with an aim of connectivity to all. The novel network architecture and service allocation approaches are required which can sustain a load of increasing number of network devices without failure and can handle the pressure of device failure during network attacks.

4. Proposed Approach

The proposed approach aims at the formation of an intelligent and sustainable architecture for IoT devices using multiple UAVs and train networks. The train networks form the key part of the proposed approach by serving as the 5G-enabled mobile fog servers. The concept of fogging is taken to another level by forming a private near user cloud system rather than the traditional static fog servers. This helps in maintaining the logistics of fogging and allows attaining flexibility in covering a large number of devices with efficient load balancing. Train networks already exist in literature, where trains serve as the mobile terminals in between the user layer and the eNodeB considering a 4G-enabled network. However, the existing train network does not consider the near user fogging as well as cloud formation to improve the connectivity which can help in sustaining the pressure of increasing incoming requests as well as network failures.

4.1. Hybrid Fogging Using UAVs and Train Networks. This section presents the details on the network architecture comprising UAVs and trains. The train network is used as a location site for placing the heavy payloads, that is, fog servers. The fog servers are the key part of the near user cloud systems, which allows data processing within the communication zone of a user rather than transmitting it over the internet. The mobile servers reduce the cost involved in setting a new private cloud system, which otherwise remains static and required multiple connections for covering more users.

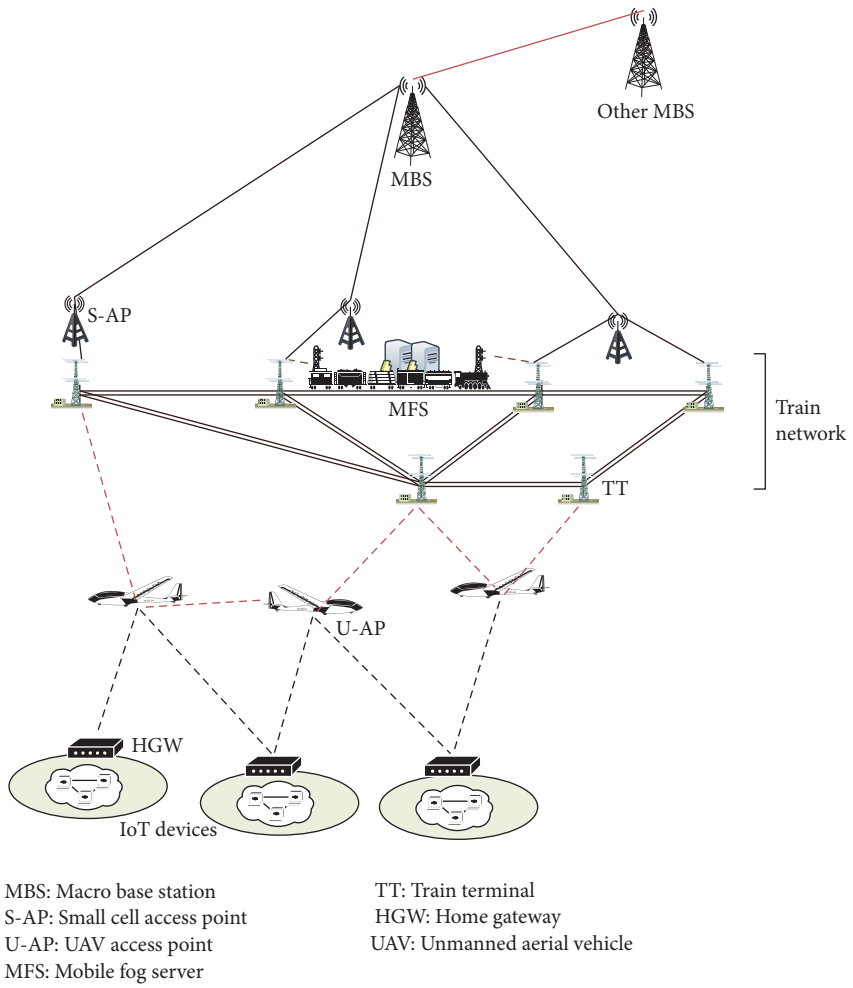

Figure 1: A representative illustration of hybrid fogging using train and UAV networks for sustainable IoT.

The mobile fogging approach allows fog servers to be traveling across the terminals with the maintenance of continuous links without consuming extra space. The storage on-the-go is the other key feature of mobile fogging. Further, the data can be easily distributed across the different trains depending on the terminals traversed. A query may arise for using UAVs as a direct fogging server between the small cell access point (S-AP) and the home gateways (HGW), which raises concerns regarding the payload supported by the aerial vehicles. Since fogging deals with the fast evaluations over data without transmitting it across the internet to a core public/private cloud, it requires heavy servers to be placed near user site, which is difficult for UAVs to accommodate. Thus, trains are used as a support for mobile fog servers.

In addition to this, a train system is a well-planned network, which seldom changes over the years with predecided and fixed route of each train. Considering all these aspects, a train system can provide strong support in the formation of sustainable mobile architectures. A representative illustration of the hybrid fogging using train and UAVs networks is shown in Figure 1 with a hierarchical view in Figure 2. The model comprises multiple macro base stations (MBS), each covering a zone which contains S-APs. The S-APs are the small cells which form the bridge between the train network and the MBS.

The underlying train network comprises two main components, namely, fog server (FS) which forms the key part of mobile fog cloud and train terminals (TT), which are the access points for connectivity between the FS and other network equipment. The direct communication with FS can also be considered; however, since the primary task of the proposed system is to form a sustainable communication setup, TT allows an extra layer of protection which helps in maintaining the continuous connectivity in the network. The connectivity between the HGW which are connected to multiple IoT devices is provided via an additional layer of UAVs which serves as UAV access points (U-AP).

An existing layer of femtocell can be used for connectivity between the HGW and the TT, but, for new network layouts, it is recommended to use dynamic UAVs as these aerial vehicles can reduce the cost involved in the implementation of static networks. Since the distance between the train network and HGW is less, UAVs can be used to provide wireless connectivity. This network can be operated over high-frequency wave system, but this would require low flying support from aerial vehicles as well as a dense network

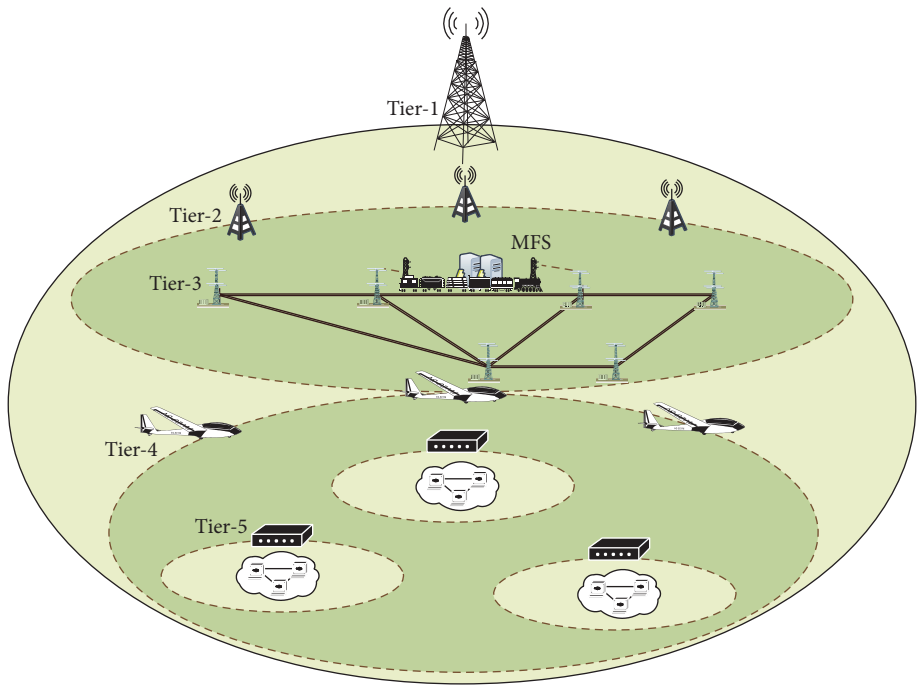

Figure 2: A hierarchical representation of hybrid fogging using train and UAV networks.

formation between them as high-frequency systems have a limited wavelength, which can be overcome by enhancing power and antenna characteristics.

4.2. System Model. The network model presented in the above section is modeled to satisfy the criteria for reliable and sustainable communications. Let B be the set of S-APs which connects the underlying components to the MBS. The underlying components include mobile FS and UAV layer which is the bridge between the HGW and the fog servers.

Let Z be the set of station terminals (TT) which are the transceiver antennas for connecting FS to the main network line. Let R be the set of trains and let F be the set of fog servers on each train. Let U be the set of UAVs which replaced the traditional femtocell of the networks, and let X be the set of users or IoT devices making continuous requests for services over the network.

The network aims at the formation of a dynamic graph $G(V, E)$ which updates after a certain interval when the nodes go beyond the transmission range or there is a change in the network topology due to node/link failure. Here, V is the set of vertices comprising network components and E is the set of edges representing transmission link between the nodes.

The link failure refers to the nonavailability of a route between the nodes despite the nodes being active, whereas the node failure refers to the nonavailability of nodes for intermediate relaying. The selection of the nodes for transmission is carried on the basis of reliability score R_s, sustainability S_t, and the output from a distributed node management (DNM) system which uses Wald's maximum model to optimize the connectivity between the nodes.

In the considered network, the reliability considering the graph $G = (V, E)$, where $V \in \{B \cup Z \cup R \cup F \cup U \cup X\}$, is given using [49] as

$$R_s = \sum_{i=1}^{n} G_i' P^i (1 - P)^{n-i}, \quad n = |V|, \tag{1}$$

where G_i' is the number of subgraphs with exactly i number of nodes when the nodes fail with a probability $Q = 1 - P$ and P is the probability of nodes without failure.

Thus, reliability can be defined as the probability of the existence of a route to a node in the set V during all time of connectivity. This means that a network can be reliable if there exists a graph G comprising the nodes x and y as a source and destination, respectively. Reliability can be considered as the only measure for sustainable networking as done by most of the solutions, but this can lead to inappropriate network formations because there can be a network with $R_s = \max$ that does not contain desired x and y. Hence, it is important to consider the reliability over links along with the reliability over nodes.

Thus, considering the reliability of connections, (1) is altered using [6], such that

$$R_s = \sum_{i=1}^{|E|} G_{x,y}' P^i (1 - P)^{m-i}, \quad m = |E|, \tag{2}$$

where $G_{x,y}'$ are the subgraphs containing x and y as vertices. Now, the probability of nonfailed nodes and graph formation

P is given as the ratio of available components for connections to the total number of components, such that

$$P = \frac{1}{|L|}\sum_{i=1}^{|L|}\left(\frac{C_a}{C_m}\right)_i, \quad (C_a)_i \neq 0. \tag{3}$$

Here, $\sum_{i=1}^{|L|}(C_a)_i = (|B'|+|Z'|+|R'|+|F'|+|U'|+|X'|)$ such that $|B'| \neq 0$, $|Z'| \neq 0$, $|R'| \neq 0$, $|F'| \neq 0$, $|U'| \neq 0$, and $|X'| \neq 0$, where every entity defines the set of available components on the particular layer. C_m is the number of components on all layers in set L such that $\sum_{i=1}^{|L|}(C_m)_i = (|B|+|Z|+|R|+|F|+|U|+|X|)$ and $|L| = 6$ as there has to be at least one vertex from all the six possible components. If any of these considerations is unsatisfied, the network fails.

Sustainability is defined in terms of connectivity over the graph between the nodes such that despite the number of failures there is always a route between the source and the destination. Out of the available connections, how many actually provides route defines the sustainability of the network; that is, if the network is active and a node guarantees connectivity in terms of R_s, it cannot guarantee sustainability until or unless it supports communication between the nodes.

Consider a scenario where the UAVs are overoccupied with the load; now the network is reliable, since there exists a path to support existing communications, but, to ensure further connections, the overheads induced due to a sudden increase in the services must be handled immediately to provide sustainable connectivity. Thus, the network sustainability is calculated as the ratio of free links to the available number of links; that is,

$$S_t = \frac{1}{|L|}\sum_{i=1}^{|L|}\left(\frac{C_b}{L_a}\right)_i, \quad (C_b)_i \neq 0, \tag{4}$$

where $\sum_{i=1}^{|L|}(C_b)_i = (|B''| + |Z''| + |R''| + |F''| + |U''| + |X''|)$, such that every entity defines the free links on each layer and L_a are the total links supported at each layer. The value 0 for free components refers to either a link or a node failure.

A graph can be reliable if it ensures the presence of end nodes in the subgraphs and it can be sustainable if it ensures the presence of a link between the end nodes of the subgraphs. The link stability can be attained by minimizing the interference between the nodes operating over the same spectrum as well as by keeping a minimum distance between the UAVs and TTs.

4.3. Decision Modeling and Optimization Problem. A decision system is formed over the reliability and sustainability of the network which helps to sustain the connectivity for longer duration without falling prey to a node or link failures. The decision system uses Wald's maximum model [18, 50] which is used in the case of involvement of two players that participate in a decision-making strategy in a sequential order. This model differs from other decision-making models by the case that the second player always knows the decision taken by the first player. This model fits well to the situation considered in this paper. The model can be applied either as a maximin formulation or as a minimax formulation.

In the considered model, the focus is on the reliability, sustainability, and the degree of connectivity. If $W\ (= T_r/T_p)$ is the weight assigned to the requests generated by the network, then the decision system aims at finding a state h in the network such that

$$h = \min_{x,y\in V} \max\left(\frac{T_p - T_r}{T_p}\right), \tag{5}$$

which refers to minimizing the maximum difference between the links available for handling the pending requests and the demanded links. Here, T_r is the number of links remaining and T_p is the total demanded links. (5) holds when the ideal state considers allocating single link to every service request. However, for load balancing, multiple links are utilized to distribute the load across the network; thus, (5) deduces to

$$h = \min_{x,y\in V} \max\left(k - \frac{T_r}{T_p}\right), \tag{6}$$

where k is the number of links fixed by the ideal state. The model can also be applied to deviation in the case if a decision is to be taken on the basis of multiple instances of subgraphs that are available over the same network. In such scenario, (6) can be represented as minimizing the maximum deviation between the states; that is,

$$h = \min_{x,y\in V} \max\left(\sqrt{\frac{1}{g}\sum_{i=1}^{g}\left(W_i - \overline{W}\right)^2}\right), \tag{7}$$

where g is the number of instances of subgraphs and $\overline{W}$ is the mean weight. Instead of mean weight, an ideal value can also be calculated to support continuous connectivity and Wald's model can be applied with respect to the ideal weight.

The entire network is subjected to three major paradigms which on successful optimization can provide highly balanced and sustainable computing for handling IoT devices. Out of the three optimization problems, one is given in (7) and the other two are as follows:

$$\begin{aligned} &\max \min\left(R_s\right), \\ &\max \min\left(S_t\right). \end{aligned} \tag{8}$$

Equation (8) aims at maximizing the minimum reliability and sustainability of the network.

4.4. Self-Aware and Sustainable Communication in IoT. The proposed architecture involves the hybridization of mobile nodes to provide a sustainable network which can guarantee a reliable and efficient communication over the IoT. The IoT devices require data evaluations to be performed at a rapid pace so as to enhance the reply time to the query maker. The data evaluations depend on the structuring of data, which is not in the scope of this paper; but the pace depends on the type of network and location of the server to perform evaluations, which is considered in this paper.

With the concept of fog computing, the near user site cloud formation provides extensive support for storage,

retrieval, caching, and mining of information without any latency. However, the cost and the periodicity of user requests affect the performance of the existing static fog computing. Thus, the proposed SACA utilizes a mobile infrastructure to sustain as well as grow the processing and computation power of a network. This allows handling of a large number of users even in the scenario of network threats, attacks, and node failures.

The proposed SACA utilizes the unique sensor feeding, sensor signatures, and knowledge-depth graphs for the formation of a sustainable architecture, which keeps a track of network states and helps in allocating the server on the basis of load and network alliance. This strategy allows the selection of efficient links and allows load migrations in the case of urgency or operational issues.

4.4.1. Sensor Feeding and Signatures. The proposed SACA utilizes the existing sensor features to find optimality in the network which can guarantee optimization over reliability, sustainability, and deviation issues. The network comprises IoT devices which have a network card installed providing a unique signature to every device. In the proposed approach, the registration is done in two ways to attain reliability. Every device which registers itself for the connectivity over the proposed model is given a unique registration number by the corresponding FS; however, FS is unaware of the registration sequence and the device to which it is allocated. This process is termed as the sensor feeding.

In sensor feeding, each device in the network requiring connections makes a request to its HGW, which keeps a track of its physical address and gathers all the information about the device activity including the type of data it operates on and allocates a unique sequence counter. Now, as soon as HGW maintains a list of incoming devices, it demands registration IDs from the FS via intermediate access points (UAVs). It is to be noted that this paper does not consider any intelligent activity on the UAVs and treat them only as a forwarding router. FS gives a set of registration IDs to an HGW and also shares the IDs across the other FSs, which helps to maintain a connection on the move. The HGW allocates the registration IDs randomly to every connected device, thus maintaining an abstraction of the sensor signatures from the FSs.

After an initial agreement between the IoT device, HGW, and FS, the HGW sends the property list to FS which now knows the type of data it will receive for the particular registration ID but is unaware of the device making the request. On receiving, FS acknowledges if it can provide the requested services; or, otherwise, it sends the request to the DNM which is placed at the station terminal. Thus, the TT not only acts as train access points but also has a capability of deciding another FS which can handle a service request that is initially declined by an initial FS. The same procedure of sensor signatures and feeding is given in Figure 3.

DNM is invoked only if an FS is unable to handle the service request; otherwise, the network operations continue without involving new network operations. This helps in maintaining lower latency by reducing the operational impact of DNM. However, in a highly overloaded condition, DNM proves to be handy as it helps to take a predecision and allows efficient allocation of service requests to the available servers. Contrary to this, invoking DNM on every service request increases the handling overhead; thus, the emphasis is given on the policy of invoking DNM only when required.

4.4.2. Knowledge-Depth Graphs. The DNM forms an integral part of the network and is invoked in the absence of service validation from a requested FS. Usually, the DNM allocates services to FS randomly from the point of view of just handling them. However, it also keeps a record for each state of the network and handles the situation when an FS declines to handle the request despite availability.

The DNM utilizes the concept of Knowledge-Depth (KD) Graphs. The KD graphs are formed by the union of two dynamic graphs, one with knowledge as the property assigned to each of its vertex and the other with the depth of knowledge as a property. The knowledge graphs are represented as $G_1^* = (V, E, K_g)$, where K_g is the knowledge set for the vertices in V. Value of each element in K_g is calculated as the ratio of total degree of the node (D_t) to the total edges in the network; that is,

$$K_{g,i} = \frac{D_{t,i}}{|E|}, \quad i \in V. \tag{9}$$

Higher value for the degree of a node represents better knowledge of the network; similarly, the depth graphs are defined as the level of knowledge, which is expressed as the ratio of the sum of direct links (D_l) in all the subgraphs containing the node to the total links available on the layer/tier to which the node belongs, such that

$$K_{d,i} = \frac{D_{l,i}}{L_{a,i}} \tag{10}$$

and $G_2^* = (V, E, K_d)$. Thus, for each set of vertices and edges, there are two graphs available which are termed as the KD graphs. These graphs help in taking a decision on the basis of requirement of the load. A depth graph is used when the load is to be transferred across the nodes of layers other than FS, whereas knowledge graph is utilized when the load is to be managed across the FS. However, the network can operate using a single KD graph by generalizing the weight associated with the vertices such that the optimal graph is given as $G_f^* = (V, E, K_w)$, and

$$K_{w,i} = \eta_1 K_{g,i} + \eta_2 K_{d,i}, \tag{11}$$

where η_1 and η_2 are the balancing constants for managing a relation between knowledge and depth such that $0 \leq \eta_1 \leq 1$ and $\eta_1 \leq \eta_2 \leq 1$ as depth is more important when the knowledge is available.

4.4.3. Load-Based Server Allocation. The proposed SACA aims at the formation of a sustainable network which guarantees connectivity even in the case of node/link failures. The proposed approach utilizes the KD graphs to take a decision on the basis of network load. The DNM takes a decision on allocating the server on the basis of trivial approach by

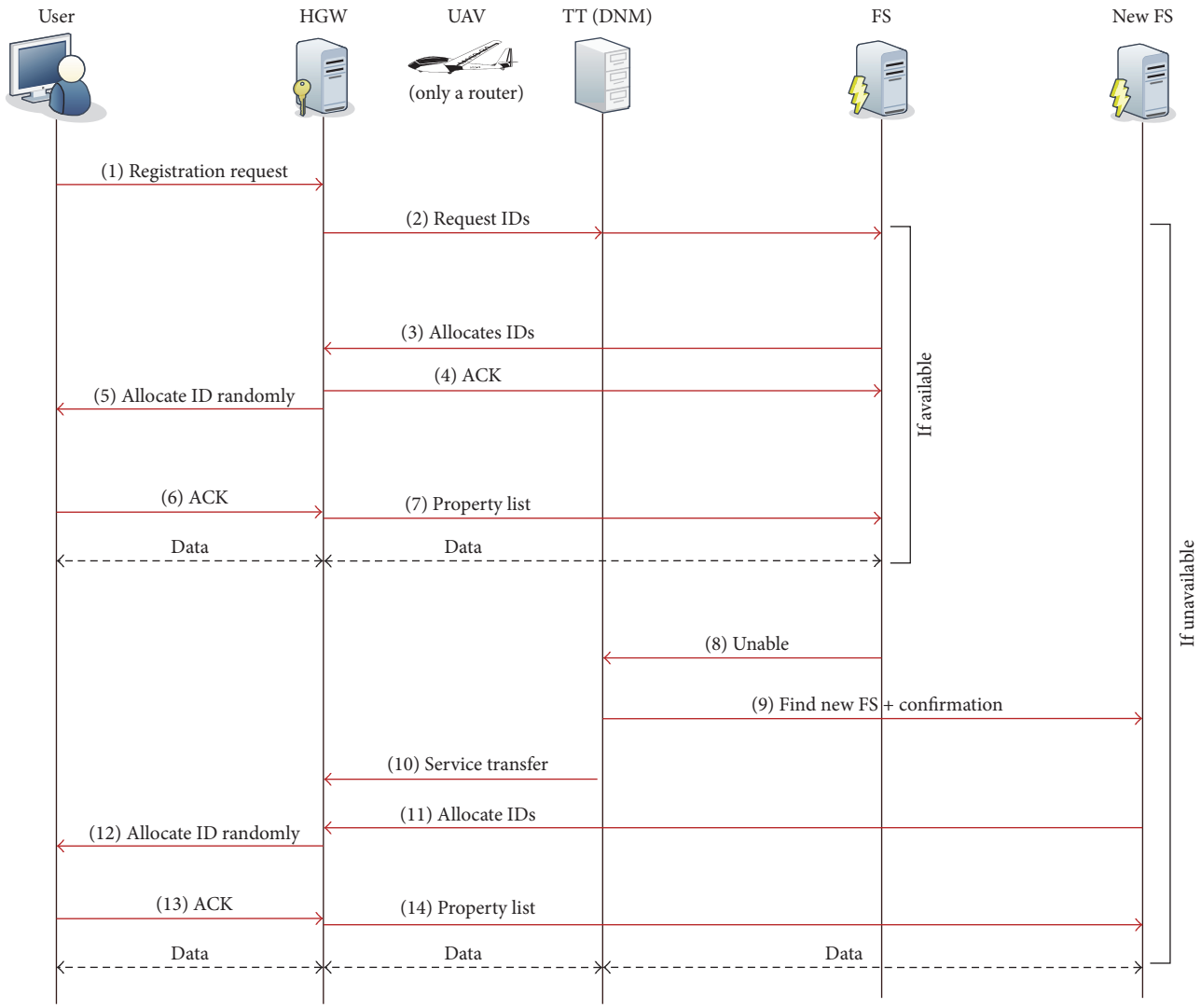

Figure 3: An illustration of sensor feeding and signature registration procedure.

checking the current load of every available server and the KD properties.

The KD graph formed is used to check the available servers for hosting the requested services by an HGW via UAVs and TTs. DNM can take the decision either on the basis of knowledge graph or depth graph or by using the common weight graph by utilizing (9)–(11). DNM is also capable of selecting multiple servers for handling the data from the same source by dividing the operations between the multiple servers. In the case of availability of multiple servers with similar load handling capabilities, the one with better KD graph properties is selected.

4.4.4. State Maintenance and Learning. Continuous connectivity depends on the state maintenance and learning about the network situation for maintaining reliability and sustainability. The state maintenance is performed by defining the order of connection. Learning can be achieved only for the first-order connections, since the nodes can have an exact status of the other node. A detailed overview of the state dependency diagram for learning is shown in Figure 4. The dependency diagram is formed by considering the connectivity between the different tiers of the network. For example, the nodes with direct connectivity are given first-order dependency and nodes with an intermediate are given second-order dependency and so on. This allows easy learning mechanisms and maintenance of "who is connected to whom."

The state maintenance and learning depend on the DNM which play a pivotal role in handling transmission across the entire network. The depth of connectivity defines the learning mechanism of a network. The utilization of mobile FS in handling large processing and storage requests is supported by the formation of a learning system which can be updated with low complexity.

Whenever a selection operation is performed across the FS, the DNM maintains a log on the basis of inputs received from the handling FS. The FS manages the traffic and provides support to DNM for watching the trend in the upcoming

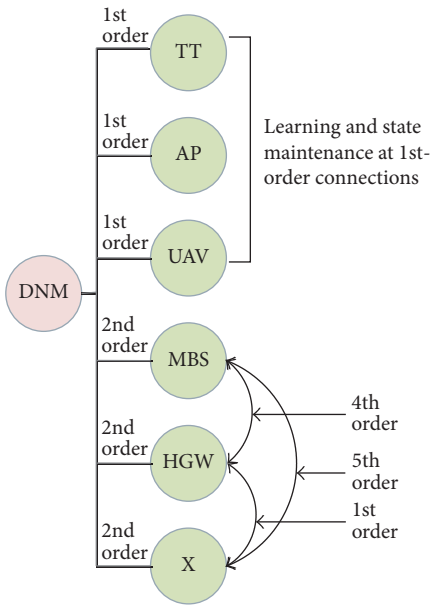

Figure 4: An illustration of state dependency for learning.

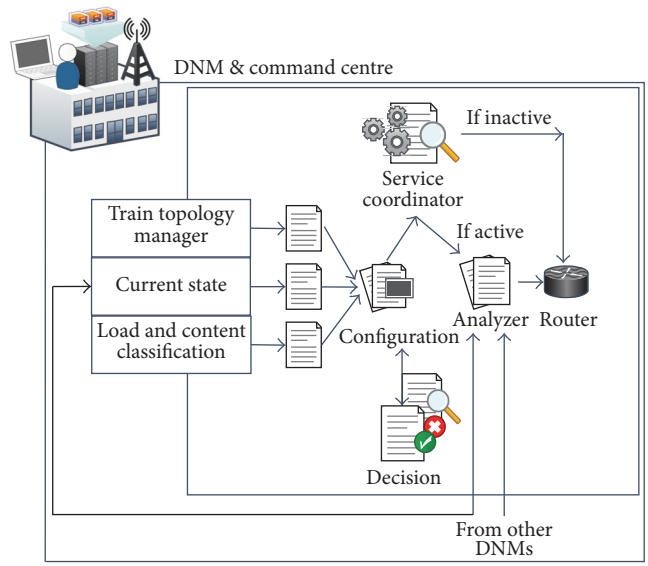

Figure 5: An illustration of DNM deployment as a spy model.

traffic and taking a decision for regulating the role of nodes involved as a fist-order connection as well as the selection of new routes in the case of failures.

The DNM and FS are equipped with the feature of sending warning signals across the entire network beforehand so as to prevent any network threat as well as an anomaly. A remedy to threat is made by not considering the server or node for further communication until the problematic server confirms positive role by sharing its consistent logs without participating in the communication.

4.4.5. Spy-Based Deployment. The entire network is laid as the initial architecture defined comprising train and UAV network. The operations are performed as a regular network except for the fact that the FS plays a key role in providing near user site cloud services for handling a large number of service requests with low latency. An intelligent system is formed over the TTs, which is termed as DNM which manages and controls the entire network by managing the network state, node configurations, and state logs. An illustration of spy-based deployment of various features over a TT-DNM is shown in Figure 5.

The deployment as a spy allows close control over the FS and a virtual control over the entire network. The train topology, current network state, and traffic controller form the key part of DNM. All these are passed as an input file to form the main configuration file. This file simply maps the network state to the train topology and the traffic condition by following the time as a controlling metric. Then, a configuration analyzer is invoked which takes a decision on the correctness of network states, after which a service coordinator takes a call on selecting the appropriate server for handling the user requests.

An intermediate analyzer is also provided which takes input from the service coordinator and the inputs from other DNMs to maintain a list of available FS. An API support is provided on the DNM server to easily manage and configure the network policies and maintain a connectivity state. The service coordinator is responsible for managing the information flow across the network. The learning is provided by maintaining file logs of each state which are then passed to the current state analyzer.

4.4.6. Network Alliance for Failures Detection. The network alliance is a node cooperative system managed by DNMs for preventing network against induced failures which may or may not be caused due to a vulnerability or network attack. The network alliance aims at handling network in the case of node/link failures without affecting its performance. The procedure for network alliance is simple and depends on the state dependency model, virtual DNM control, and periodic analyses of the sustainability value.

DNM periodically shifts the control to one of the nodes of every tier which operates as a virtual DNM. All the nodes in the tiers register and share their calculated sustainability value with the virtual DNM node. The virtual DNM node then provides all the accessed information to the TT-DNM which matches the attained information with its state diagram. Such application allows DNM to possess a virtual control over the entire network.

The procedure of network alliance can be conducted either periodically or during failure in the network. The value of R_s and S_t can also be used to decide on conducting network alliance procedures. The steps for network alliance are presented in Algorithm 1. The network alliance helps in understanding the failures in the network, which allows taking a decision on changing interaction procedures with the nodes so as to reduce the delay.

Lemma 1. *The depth, knowledge, and probability of connection between the nodes increase with a higher degree of connectivity which improves the reliability of the network.*

(1) **Input**: Current State, DNM - S_t and R_s
(2) **Output**: Decision for continuity or change link
(3) set interval
(4) **while** Transmission Continues **do**
(5) check for interval
(6) request periodic update
(7) select most communicating node from each layer
(8) send virtual control to the node
(9) receive and map state dependencies
(10) **if** ambiguous $\|S_t < \overline{S_t}\| \; R_s < \overline{R_s}$ **then**
(11) eliminate node from state-dependencies
(12) update neighbours
(13) **else**
(14) continue
(15) **end while**
(16) **end while**

Algorithm 1: Network alliance for prevention against failures.

Proof. From (9)–(11), the knowledge and depth increase affecting the connectivity between the nodes; this connectivity directly affects the probability of connections between the nodes as the number of subgraphs containing the source and destination will increase with an increase in the overall degree of the nodes. Using (2), with the increase in probability and number of subgraphs, the reliability of the network increases. □

Lemma 2. *With reliability attaining a maximum value, the sustainability increases and maximizes if available degree per node is equal to k times the remaining links* (T_r).

Proof. Reliability of a network can maximize with an increase in the number of subgraphs containing the source and destination nodes, which further increases with an increase in the probability of connectivity. However, a reliable network does not always guarantee sustainable formations, which depends on the number of connections available and supported by each node. From (5), if $T_p = kT_r$, maximum number of links are available, which increases the number of free links (see (4)), thus, maximizing the sustainability. □

Remark 3. Reliability and sustainability can simultaneously maximize their minimum value when $P = 1$.

Proof. With a higher degree per node, a sufficient number of subgraphs are available for connectivity between the nodes, which makes $P = 1$ and, from (1), R_s attains maximum. Now, with R_s at maximum and $P = 1$, $C_b = L_a$, which makes S_t attain a maximum value. □

Remark 4. Traffic variations and an increase in the number of users making connection demands affect the network reliability.

Proof. With an increase in the number of users making continuous service requests, the probability of falling in different subgraph increases as all the users may not fall in the same subgraph; this decreases the reliability of the network, and alternative paths are required for transmissions. This is the condition for the requirement of load balancing. □

Remark 5. The minimum number of connections required to sustain communication is greater than or equal to k, where k defines the minimum rule for connectivity. The number of station terminals required to allow this transmission depends on the rule of $2r + Q$, where r is the radio range of terminals and Q is the length of a train.

Proof. From (5) and (6), $T_p = kT_r$, which defines the condition for minimum number of connections as stated by the lemma. Now, for the number of terminals, the trivial rule of $2r + Q$ is followed as a measure of distance between two TTs, as shown in Figure 6. Considering wireless multihop transmissions, the antennas are placed on the both ends of train which are connected to each other via FS; now, the antennas over train and TT are assumed to have common radio range r, which means the maximum gap between two TTs can be maximum up to $2r+Q$. The placement of TTs in the proposed architecture should be governed by this rule. □

5. Performance Evaluation

The proposed SACA is evaluated in three parts. The first part presents the numerical analyses for reliability and sustainability, the second part presents evaluation of SACA in simulation environment, and the third part evaluates SACA for its sustenance in the presence of intruders and attackers resulting into node failures.

5.1. Numerical Analyses. In the numerical analyses, the results are presented for the variation of a variable (connections) and its impact over the network reliability and sustainability. The numerical analyses are conducted using Matlab™ with configurations given in Table 1.

A total of 1000 users made consistent connection demand in a network operating with 5 tiers. The results are recorded

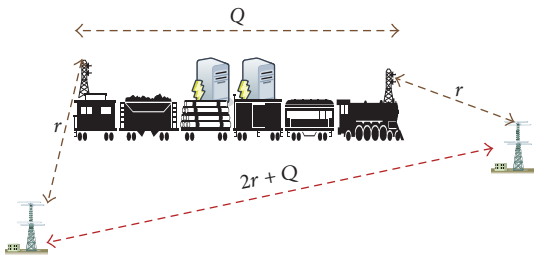

Figure 6: An illustration of maximum distance sustainable between two TTs.

Table 1: Parameter configurations for numerical analyses.

Parameter	Value	Description
$\|L\|$	5	Number of tiers
$\|X\|$	1000	Number of IoT devices
$\|B\|$	2	Access points
$\|Z\|$	5–20	Station terminals
$\|R\|$	4	Number of trains
$\|F\|$	4	Number of fog servers
$\|U\|$	5	Number of UAVs
k	2	Number of connections per node
G	Random	Graph for analyses
η_1, η_2	0.5	Balancing constants

for variation in the number of TTs with respect to the connections demanded by the nodes. The variation in the probability of connectivity affects the performance of the network. With a larger number of free connections, the network capability in handling more users increases. Figure 7 presents the results for variation in the network reliability with variation in the number of terminals available for connectivity operating with 5 UAVs. Also, the graph includes the impact of variation in the number of subgraphs including the source and the destination. With a higher value of intermediate nodes and terminals for connectivity, more links are available for connections. Also, the increase in the number of links is accompanied by an increase in the alternative routes between the source and destination which increases the probability of connectivity of the overall network, thus resulting in an increase in the overall reliability of the network.

A reliable network may or may not provide sustainable connectivity as it may have a connection for one set of nodes, while on the other hand it may not provide connectivity between the requested nodes. Since the numerical simulations are performed in a common graph, network sustainability attained a higher value with an increase in the number of TTs as shown in Figure 8. The increase is marked by an increase in the overall probability of connectivity for the users making continuous requests. Thus, it is concluded from the numerical analyses that the number of intermediate nodes and the connection supported by them heavily impact the performance of the network. However, deployment of more number of TTs and other intermediate nodes will increase the overall cost of network. Thus, selection of an optimal value for the number of intermediate nodes can be

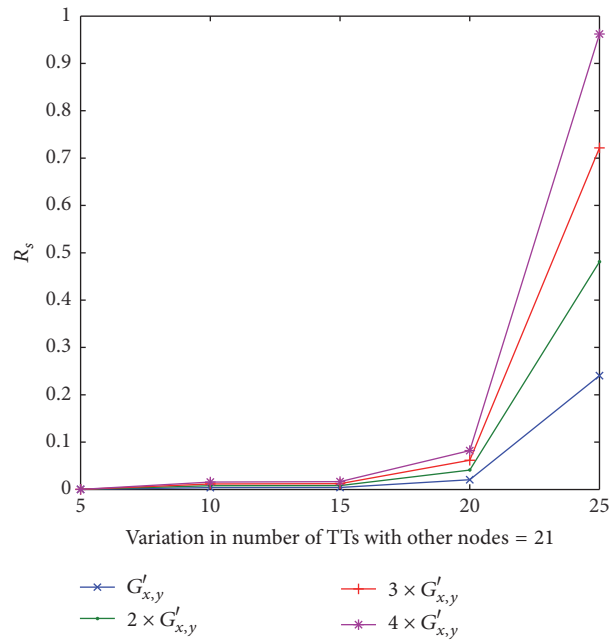

Figure 7: R_s versus variation in the number of TTs and subgraphs containing source and destination.

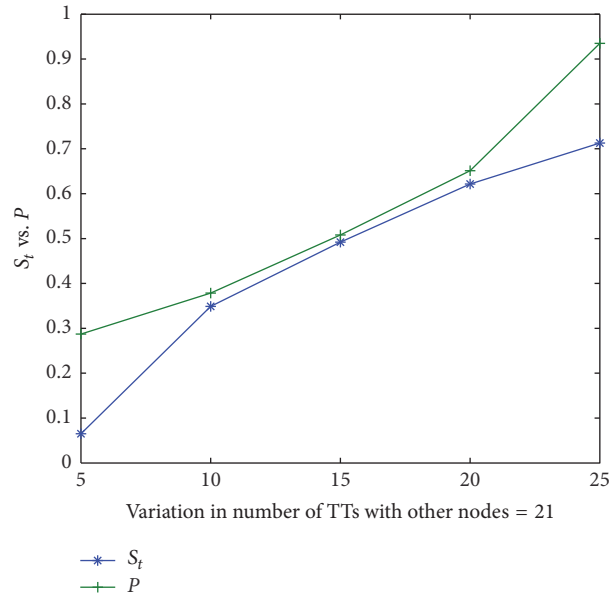

Figure 8: S_t and probability of connectivity versus variation in the number of TTs.

performed by considering the level of stability which should be sufficient enough to maintain continuous connections.

5.2. Network Simulation Analyses. The network simulations are conducted using Matlab by creating a scenario comprising all the components as network nodes operating using the configurations of a wireless network. The range of each node is kept fixed at 500 m. The length of a train is taken to be 50 m. The number of aerial nodes varied between 5 and 20

Table 2: Parameter configurations for simulations.

Parameter	Value	Description
$\|L\|$	5	Number of tiers
$\|X\|$	200–1000	Number of IoT devices
$\|B\|$	50–100	Access points
$\|Z\|$	20–50	Station terminals
$\|R\|$	5	Number of trains
$\|F\|$	5	Number of fog servers
$\|U\|$	5–20	Number of UAVs
k	2	Number of connections per node
Mobility-IoT	Random waypoint	Mobility model ground
Mobility-UAVs	Cooperative mode	Mobility model aerial
UAV speed	50 kmph	UAV movement
Train speed	150 kmph	Train movement
r	500 m	Radio range
Q	50 m	Length of train
Runs	20	Simulation runs
Traffic type	CBR	Traffic over TCP
Packet size	1024 bytes	Average packet size
Buffer	5000–20000	Buffer capacity
Interval	2 p/s	Time to wait before sending
Initial rate	256 kbps	Initial transmission rate

and the traffic is created using Poisson distribution. A total of 20 simulation runs are traced for analyzing the performance of the proposed SACA in terms of end to end delay and packet loss.

A variation in the number of users is considered with the intermediate links in dynamic state operating in an environment with a failure rate of 10% and 20%. The failures in simulations are dynamically induced over the random links. During simulations, the source and destination are kept at a distance having at least 2 intermediate hops from the failed nodes. Other configuration details for simulations are shown in Table 2.

The IoT devices are modeled using random waypoint, whereas cooperative framework [51] is used for the aerial nodes. In the performed simulations, five trains are made to run with fixed periodicity at a speed of 150 kmph. The IoT devices operate in "Request" mode, which means every device in the network demands authority for transmission during simulations. The baseline of the proposed approach is defined with no failures and complete connectivity between the nodes. Link state routing is applied using K_w as the weight metric. The simulation results are evaluated for the end to end delays and packet loss. The end to end delay ($E2D$) is calculated using [52] as

$$E2D = D_{\text{transmission}} + D_{\text{propagation}} + D_{\text{queue}} + D_{\text{processing}}, \tag{12}$$

where $D_{\text{transmission}}$ is calculated as the ratio of number of bits transferred to the link speed (rate of transmission), $D_{\text{propagation}}$ is the ratio of distance between the nodes to the channel speed, D_{queue} is the waiting time of packets before the beginning of processing, and $D_{\text{processing}}$ is the delay induced during the forwarding of packets. The packet loss is calculated as the ratio of lost packets to the total transmitted over the network.

With a variation in the failure of nodes from 10% to 20%, the end to end delay is recorded 45.7% and 48% higher than the baseline which observed a maximum end to end delay of 27 seconds. The higher values include the entire session delays as shown in Figure 9. With an increase in the number of users, more requests are made in the network, which affects the availability of connections per component. This increase is reflected in terms of average waiting time, which is included in the end to end delay graphs. However, focusing on the level of complexity considered in the simulations and the number of users making simultaneous requests over limited resources, the lesser delays justify the efficiency of the proposed model.

Further, with lesser values of delay, the overall performance of the proposed SACA is very high. The average packet loss for the entire session is very less as shown in Figure 10. The baseline operations recorded 20.8% and 34.2% lower packet loss in comparison with operations at 10% and 20% failure rate of nodes. The lowest value of 1.21% is recorded for packet loss in the proposed approach. However, with an increase in the number of users and constraints by the increase in average waiting time, the packet loss increases but does not go beyond a value which can affect the network. The overall delivery ratio of the proposed approach remains higher than 90% even in the scenarios of induced failures.

5.3. Sustainability Analyses. The network is sustainable if it provides strong connectivity support even in the case of

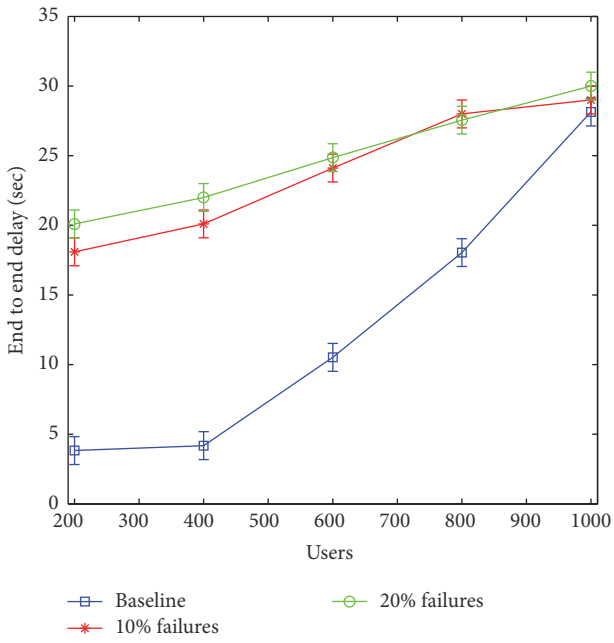

Figure 9: End to end delay versus users.

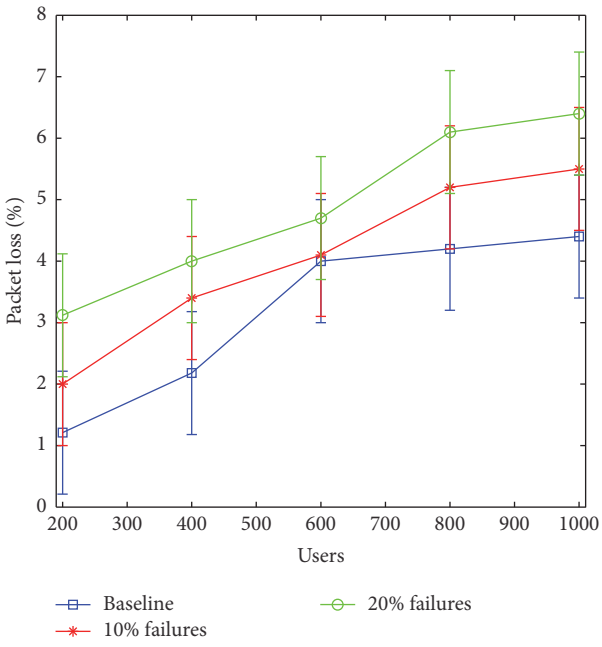

Figure 10: Packet loss (%) versus users.

node/link failures. The maximum cases of failures arise in a network when an attack is induced in the network. An attack can be as severe as resulting into the leakage of information, compromising of user accounts, or even the shutdown of the entire network.

Three different attack scenarios are considered for evaluation of the sustainability of the proposed SACA model, namely, Sybil attack, distributed denial of service (DDoS), and wormhole attack. Sybil attack is vulnerability caused when the users of a system induce false reputation for the intruder node [53]. A DDoS attack is caused by multiple

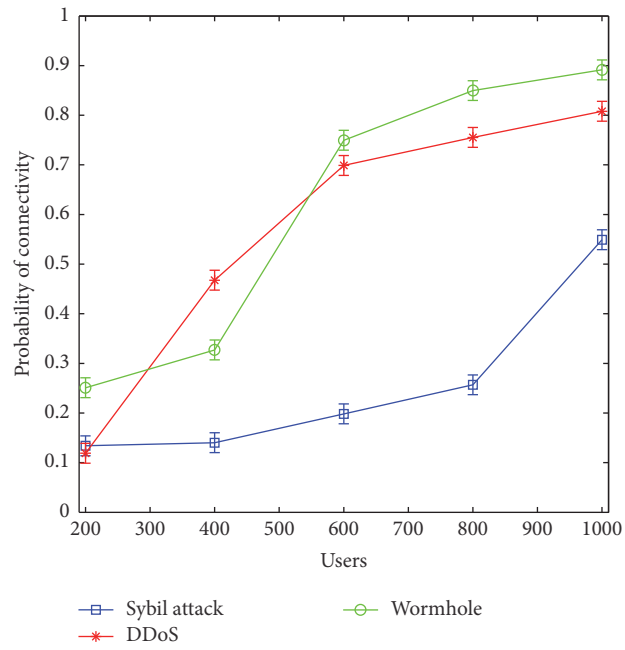

Figure 11: Probability of connectivity versus users.

flooding over a single path by network nodes affecting the traffic flow over a particular link [54, 55]. Wormhole attack is a type of false routing by making a glimpse of the presence of an alternative shorter route to the actual node [56]. All these attacks are highly critical for any type of network. These attacks are also time dependent as they take some time to penetrate into the entire network. The results shown in the paper are evaluated for each simulation cycle of 100 seconds.

The proposed approach is evaluated for its probability of connectivity and degree of sustainability in the presence of these attacks. A similar network as that used in the simulations is considered for the evaluation of proposed model in the presence of attacker nodes. The attacks are induced over 20% of the total nodes in the network. For Sybil attack, a false reputation index is given to the attacker nodes and the network pretends to consider these nodes as legitimate. Any data transmitted to these nodes is considered as a dropped packet and results are recorded. Similarly, for DDoS, 20% of the network nodes preoccupy the links leaving a lesser number of subgraphs with source and destination, and, for the wormhole, 20% of the nodes give similar next hop address for generating false routes.

Initially, the results are recorded for the probability of connectivity as shown in Figure 11. Sybil attack affected the proposed model more in comparison with the DDoS and wormhole attack. This is because of difficulty in identifying the nodes by the DNM while performing network alliance. The dependency over a false node for network alliance causes a reduction in the transmission rate as well as the probability of connectivity. However, with an increase in the number of nodes, more alternative paths are available, which allows an increase in the connectivity. The key advantage of the proposed approach is its high provisioning of sustainable networking. Despite the presence of attacker nodes in the

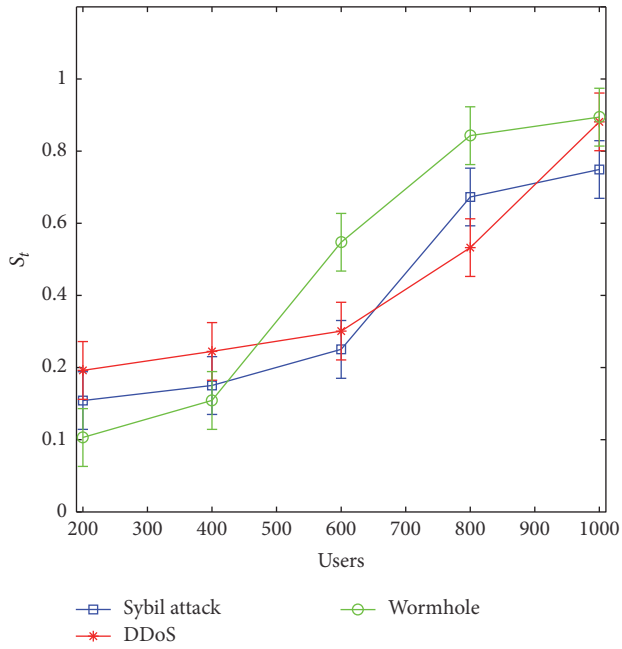

FIGURE 12: Sustainability (S_t) versus users.

network, the proposed model which utilizes a periodic concept of network alliance via DNM over TTs is capable of keeping aloof the vulnerable nodes from the selected path. Further, the quick succession of next hop in the case of node/link failure during an attack in the proposed model allows a high value for network sustainability as shown in Figure 12. With an increase in the number of users, the sustainability of the network further increases as the number of subgraphs containing the source and destination increases along with an increase in the number of available links for connectivity.

6. Discussions and Open Issues

The proposed SACA is capable of providing continuous services even in the adverse conditions which prevent data forwarding as well as decision-making on selecting next hop. SACA uses a DNM module which operates over one or all TTs of train network and allows efficient control over the network. The use of state diagram dependency allows resolution of conflicts involved in the selection on next hop and network alliance is used to handle the failures. The proposed approach is efficient in providing low delay and low packet loss transmission with high reliability, probability of connectivity, and sustainability as proven by the results.

There are other certain architectures which aim at provisioning of sustainable and robust connectivity over IoT devices but in a particular application scenario. Most of them use the existing underlying network model and do not present any variation in network formation. The existing solutions only provide a service-level solution for enhancing the connectivity for IoT devices. A state-of-the-art comparison is presented in Table 3 which presents key contributions for sustainable IoT along with their ideologies.

Despite the existing and proposed solution for sustainable IoT, there are several other issues which must be handled for the robust and reliable communication in IoT devices. These include the following:

(i) Handling energy constraint for reliable communications

(ii) Approaches for privacy preservation and fast authentication of IoT devices

(iii) Handling handovers efficiently to provide less latent architectures

(iv) Service divisibility and abstraction for context-aware IoT

(v) Self-organization and autonomous decision-making for IoT devices to handle unexpected failures and changes in the network

7. Conclusion

IoT demands high support from the underlying network. An efficient network can help in sustaining the services across the devices with prolonged connectivity. In this paper, the problem of sustainable and reliable flow of information is considered over a hybrid network formation. The solution proposed in this paper uses a concept of hybrid multimodular self-aware architecture which helps in providing fault-free communication. The proposed model uses a distributed node management (DNM) system which takes care of network connections and helps in identifying nodes which are reliable and can sustain the pressure of increasing demand of users. An optimization problem is formulated using Wald's maximum model. Analyses show that the proposed approach is capable of providing sustainable and reliable connectivity with lesser delay and fewer packet losses. The proposed approach is also capable of handling transmissions even in the scenarios that are under the threat of Sybil, wormhole, and DDoS attacks. By intelligent decision-making and self-awareness regarding the state of network components, the proposed model can guarantee connectivity even in unfavorable conditions.

In the future, we shall be aiming at extending the features of the proposed DNM to more realistic scenarios and testing it using hardware-assisted emulations.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported by Institute for Information & communications Technology Promotion (IITP) grant funded by the Korea government (MSIP) (B0190-16-2032, Development of Operating System Security Core Technology for the Smart Lightweight IoT Devices) as well as by the Soonchunhyang University Research Fund (no. 20150690).

Table 3: State-of-the-art architectures for efficient connectivity in IoT.

Approach	Ideology	Sustainable	Key features	Application-specific	Secure	Network novelty
Cirani et al. [7]	Authorization service architecture	Yes	Constrained application protocol	No	Yes	No
Zgheib et al. [30]	Semantic data driven architecture for healthcare	Yes	Semantic Sensor networks	Yes	No	No
Gupta et al. [31]	Cloud centric architecture	Yes	XML web services	Yes	Yes	No
Barbosa et al. [32]	Architecture for emotional smartphones	No	Happy system architecture	Yes	No	No
Sarkar et al. [33]	Distributed architecture for IoT	No	Automated service management	No	Yes	No
Flauzac et al. [34]	SDN-based architecture	No	Distribution of security rules	No	Yes	Yes
Proposed SACA	Hybrid mobile fog servers	Yes	Distributed node management	No	Yes	Yes

References

[1] D. Bol, J. De Vos, F. Botman et al., "Green socs for a sustainable internet-of-things," in *Proceedings of the IEEE Faible Tension Faible Consommation (FTFC '13)*, pp. 1–4, IEEE, Paris, France, 2012.

[2] L. Fritsch, A.-K. Groven, and T. Schulz, "On the internet of things, trust is relative," in *Proceedings of the International Joint Conference on Ambient Intelligence*, pp. 267–273, Springer, Amsterdam, The Netherlands, November 2011.

[3] N. K. Giang, J. Im, D. Kim, M. Jung, and W. Kastner, "Integrating the epcis and building automation system into the internet of things: a lightweight and interoperable approach," *Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications*, vol. 6, no. 1, pp. 56–73, 2015.

[4] C. E. El Kaed, I. Khan, H. Hossayni, and P. Nappey, "Sqeniot: Semantic query engine for industrial internet-of-things gateways," in *Proceedings of the IEEE 3rd World Forum on Internet of Things (WF-IoT '16)*, pp. 204–209, Reston, Va, USA, December 2016.

[5] B. B. Snchez, D. S. de Rivera, and L. Snchez-Picot, "Building unobtrusive wearable devices: an ergonomic cybernetic glove," *Journal of Internet Services and Information Security (JISIS)*, vol. 6, pp. 37–52, 2016.

[6] T. Robles, R. Alcarria, D. Martín et al., "An iot based reference architecture for smart water management processes," *Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications*, vol. 6, no. 1, pp. 4–23, 2015.

[7] S. Cirani, M. Picone, P. Gonizzi, L. Veltri, and G. Ferrari, "IoT-OAS: an oauth-based authorization service architecture for secure services in IoT scenarios," *IEEE Sensors Journal*, vol. 15, no. 2, pp. 1224–1234, 2015.

[8] S. M. Ghaleb, S. Subramaniam, Z. A. Zukarnain, and A. Muhammed, "Mobility management for IoT: a survey," *Eurasip Journal on Wireless Communications and Networking*, vol. 2016, no. 1, article 165, 2016.

[9] G. Marques, N. Garcia, and N. Pombo, "A survey on IoT: architectures, elements, applications, QoS, platforms and security concepts," in *Advances in Mobile Cloud Computing and Big Data in the 5G Era*, vol. 22 of *Studies in Big Data*, pp. 115–130, Springer International Publishing, Cham, Switzerland, 2017.

[10] K. Kai, W. Cong, and L. Tao, "Fog computing for vehicular Ad-hoc networks: paradigms, scenarios, and issues," *Journal of China Universities of Posts and Telecommunications*, vol. 23, no. 2, pp. 56–96, 2016.

[11] K. Hong, D. Lillethun, U. Ramachandran, B. Ottenw, and B. Koldehofe, "Mobile fog: a programming model for large-scale applications on the internet of things," in *Proceedings of the 2nd ACM SIGCOMM Workshop on Mobile Cloud Computing*, pp. 15–20, ACM, Hong Kong, August 2013.

[12] V. Sharma and R. Kumar, "Teredo tunneling-based secure transmission between UAVs and ground ad hoc networks," *International Journal of Communication Systems*, 2016.

[13] V. Sharma, I. You, and R. Kumar, "Energy efficient data dissemination in multi-UAV coordinated wireless sensor networks," *Mobile Information Systems*, vol. 2016, Article ID 8475820, 13 pages, 2016.

[14] V. Sharma and R. Kumar, "Cooperative frameworks and network models for flying ad hoc networks: a survey," *Concurrency Computation: Practice and Experience*, 2016.

[15] M. Mozaffari, W. Saad, M. Bennis, and M. Debbah, "Drone small cells in the clouds: design, deployment and performance analysis," in *Proceedings of the 58th IEEE Global Communications Conference (GLOBECOM '15)*, pp. 1–6, December 2015.

[16] M. Mozaffari, W. Saad, M. Bennis, and M. Debbah, "Unmanned aerial vehicle with underlaid device-to-device communications: performance and tradeoffs," *IEEE Transactions on Wireless Communications*, vol. 15, no. 6, pp. 3949–3963, 2016.

[17] A. Merwaday and I. Guvenc, "UAV assisted heterogeneous networks for public safety communications," in *Proceedings of the IEEE Wireless Communications and Networking Conference Workshops (WCNCW '15)*, pp. 329–334, IEEE, New Orleans, La, USA, March 2015.

[18] A. Wald, "Statistical decision functions," in *Breakthroughs in Statistics*, Springer Series in Statistics, pp. 342–357, Springer, New York, NY, USA, 1992.

[19] D. Kyriazis, T. Varvarigou, D. White, A. Rossi, and J. Cooper, "Sustainable smart city IoT applications: heat and electricity management & Eco-conscious cruise control for public transportation," in *Proceedings of the IEEE 14th International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM '13)*, pp. 1–5, June 2013.

[20] S. Wen, X. Zhu, X. Zhang, and D. Yang, "QoS-aware mode selection and resource allocation scheme for Device-to-Device (D2D) communication in cellular networks," in *Proceedings of the IEEE International Conference on Communications Workshops (ICC '13)*, pp. 101–105, IEEE, Budapest, Hungary, June 2013.

[21] S. Mayer, D. Guinard, and V. Trifa, "Searching in a web-based infrastructure for smart things," in *Proceedings of the 3rd International Conference on the Internet of Things (IOT '12)*, pp. 119–126, IEEE, Wuxi, China, 2012.

[22] M. Kovatsch, S. Mayer, and B. Ostermaier, "Moving application logic from the firmware to the cloud: towards the thin server architecture for the internet of things," in *Proceedings of the 6th International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing (IMIS '12)*, pp. 751–756, July 2012.

[23] T. Riedel, N. Fantana, A. Genaid, D. Yordanov, H. R. Schmidtke, and M. Beigl, "Using web service gateways and code generation for sustainable IoT system development," in *Proceedings of the Internet of Things (IoT '10)*, pp. 1–8, IEEE, Tokyo, Japan, December 2010.

[24] F. Tao, Y. Cheng, L. D. Xu, L. Zhang, and B. H. Li, "CCIoT-CMfg: cloud computing and internet of things-based cloud manufacturing service system," *IEEE Transactions on Industrial Informatics*, vol. 10, no. 2, pp. 1435–1442, 2014.

[25] L. Gao, T. H. Luan, B. Liu, W. Zhou, and S. Yu, "Fog computing and its applications in 5g," in *5G Mobile Communications*, pp. 571–593, Springer, 2017.

[26] C. Dsouza, G.-J. Ahn, and M. Taguinod, "Policy-driven security management for fog computing: preliminary framework and a case study," in *Proceedings of the 15th IEEE International Conference on Information Reuse and Integration (IEEE IRI '14)*, pp. 16–23, August 2014.

[27] L. Prieto González, C. Jaedicke, J. Schubert, and V. Stantchev, "Fog computing architectures for healthcare: wireless performance and semantic opportunities," *Journal of Information, Communication and Ethics in Society*, vol. 14, no. 4, pp. 334–349, 2016.

[28] R. Chen, "An intelligent value stream-based approach to collaboration of food traceability cyber physical system by fog computing," *Food Control*, vol. 71, pp. 124–136, 2017.

[29] T. H. Luan, L. Gao, Z. Li, Y. Xiang, G. We, and L. Sun, "A view of fog computing from networking perspective," https://arxiv.org/abs/1602.01509.

[30] R. Zgheib, E. Conchon, and R. Bastide, "Engineering IoT healthcare applications: towards a semantic data driven sustainable architecture," in *eHealth 360◦*, vol. 181 of *Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering*, pp. 407–418, Springer International Publishing, Cham, 2017.

[31] P. K. Gupta, B. T. Maharaj, and R. Malekian, "A novel and secure IoT based cloud centric architecture to perform predictive analysis of users activities in sustainable health centres," *Multimedia Tools and Applications*, 2016.

[32] R. Barbosa, D. Nunes, A. Figueira et al., "An architecture for emotional smartphones in Internet of Things," in *Proceedings of the IEEE Ecuador Technical Chapters Meeting (ETCM '16)*, vol. 1, pp. 1–5, Guayaquil, Ecuador, October 2016.

[33] C. Sarkar, A. U. N. Sn, R. V. Prasad, A. Rahim, R. Neisse, and G. Baldini, "Diat: a scalable distributed architecture for iot," *IEEE Internet of Things Journal*, vol. 2, no. 3, pp. 230–239, 2015.

[34] O. Flauzac, C. Gonzalez, A. Hachani, and F. Nolot, "SDN based architecture for IoT and improvement of the security," in *Proceedings of the 29th IEEE International Conference on Advanced Information Networking and Applications Workshops (WAINA '15)*, pp. 688–693, March 2015.

[35] F. Y. Okay and S. Ozdemir, "A fog computing based smart grid model," in *Proceedings of the International Symposium on Networks, Computers and Communications (ISNCC '16)*, pp. 1–6, Yasmine Hammamet, Tunisia, May 2016.

[36] B. Tang, Z. Chen, G. Hefferman, T. Wei, H. He, and Q. Yang, "A hierarchical distributed fog computing architecture for big data analysis in smart cities," in *Proceedings of the ASE BigData and SocialInformatics (ASE BD and SI '15)*, ACM, October 2015.

[37] M. A. Al Faruque and K. Vatanparvar, "Energy management-as-a-service over fog computing platform," *IEEE Internet of Things Journal*, vol. 3, no. 2, pp. 161–169, 2016.

[38] J. Zhu, D. S. Chan, M. S. Prabhu, P. Natarajan, H. Hu, and F. Bonomi, "Improving web sites performance using edge servers in fog computing architecture," in *Proceedings of the IEEE 7th International Symposium on Service-Oriented System Engineering (SOSE '13)*, pp. 320–323, IEEE, San Francisco, Calif, USA, March 2013.

[39] M. Aazam and E.-N. Huh, "Fog computing and smart gateway based communication for cloud of things," in *Proceedings of the 2nd International Conference on Future Internet of Things and Cloud (FiCloud '14)*, pp. 464–470, Barcelona, Spain, August 2014.

[40] C.-X. Wang, F. Haider, X. Gao et al., "Cellular architecture and key technologies for 5G wireless communication networks," *IEEE Communications Magazine*, vol. 52, no. 2, pp. 122–130, 2014.

[41] Y. Hu, H. Li, Z. Chang, and Z. Han, "Scheduling strategy for multimedia heterogeneous high-speed train networks," *IEEE Transactions on Vehicular Technology*, 2016.

[42] L. Lei, J. Lu, Y. Jiang et al., "Stochastic delay analysis for train control services in next-generation high-speed railway communications system," *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 1, pp. 48–64, 2016.

[43] B. Hui, J. Kim, H. Chung, and I. Kim, "Creation and control of handover zone using antenna radiation pattern for high-speed train communications in unidirectional networks," in *Proceedings of the International Conference on Information and Communication Technology Convergence (ICTC '16)*, pp. 737–740, IEEE, Jeju, Korea, October 2016.

[44] Z. Li, Y. Chen, H. Shi, and K. Liu, "NDN-GSM-R: a novel high-speed railway communication system via named data networking," *EURASIP Journal on Wireless Communications and Networking*, vol. 2016, no. 1, article 48, pp. 1–5, 2016.

[45] E. A. Ibrahim, E. F. Badran, and M. R. Rizk, "An optimized LTE measurement handover procedure for high speed trains using WINNER II channel model," in *Proceedings of the 22nd Asia-Pacific Conference on Communications (APCC '16)*, pp. 197–203, IEEE, Yogyakarta, Indonesia, August 2016.

[46] S. Xu, G. Zhu, B. Ai, and Z. Zhong, "A survey on high-speed railway communications: a radio resource management perspective," *Computer Communications*, vol. 86, pp. 12–28, 2016.

[47] L. Gao, T. H. Luan, S. Yu, W. Zhou, and B. Liu, "FogRoute: DTN-based data dissemination model in fog computing," *IEEE Internet of Things Journal*, vol. 4, no. 1, pp. 225–235, 2016.

[48] H. Kopetz and S. Poledna, "In-vehicle real-time fog computing," in *Proceedings of the 46th Annual IEEE/IFIP International Conference on Dependable Systems and Networks Workshop (DSN-W '16)*, pp. 162–167, Toulouse, France, June 2016.

[49] O. Goldschmidt, P. Jaillet, and R. LaSota, "On reliability of graphs with node failures," *Networks*, vol. 24, no. 4, pp. 251–259, 1994.

[50] A. Wald, "Statistical decision functions," *Annals of Mathematical Statistics*, vol. 20, pp. 165–205, 1949.

[51] V. Sharma and R. Kumar, "A cooperative network framework for multi-UAV guided ground ad hoc networks," *Journal of Intelligent and Robotic Systems: Theory and Applications*, vol. 77, no. 3-4, pp. 629–652, 2015.

[52] V. Sharma, M. Bennis, and R. Kumar, "UAV-assisted heterogeneous networks for capacity enhancement," *IEEE Communications Letters*, vol. 20, no. 6, pp. 1207–1210, 2016.

[53] K. Zhang, X. Liang, R. Lu, and X. Shen, "Sybil attacks and their defenses in the internet of things," *IEEE Internet of Things Journal*, vol. 1, no. 5, pp. 372–383, 2014.

[54] S. T. Zargar, J. Joshi, and D. Tipper, "A survey of defense mechanisms against distributed denial of service (DDOS) flooding attacks," *IEEE Communications Surveys and Tutorials*, vol. 15, no. 4, pp. 2046–2069, 2013.

[55] T. Booth and K. Andersson, "Network security of internet services: eliminate DDoS reflection amplification attacks," *Journal of Internet Services and Information Security*, vol. 5, pp. 58–79, 2015.

[56] L. Wallgren, S. Raza, and T. Voigt, "Routing attacks and countermeasures in the RPL-based internet of things," *International Journal of Distributed Sensor Networks*, vol. 2013, Article ID 794326, 2013.