Reconfigure and Reuse: Interoperable Wearables for Healthcare IoT

Nidhi Pathak[1*], Anandarup Mukherjee[2†], Sudip Misra[2‡]
[1]Advanced Technology Development Center, [2]Department of Computer Science and Engineering
Indian Institute of Technology Kharagpur, India
Email: *nidhipathak@iitkgp.ac.in, †anandarupmukherjee@ieee.org, ‡smisra@cse.iitkgp.ac.in

***Abstract*—In this work, we propose Over-The-Air (OTA)-based reconfigurable IoT health-monitoring wearables, which tether wirelessly to a low-power and portable central processing and communication hub (CPH). This hub is responsible for the proper packetization and transmission of the physiological data received from the individual sensors connected to each wearable to a remote server. Each wearable consists of a sensor, a communication adapter, and its power module. We introduce low-power adapters with each sensor, which facilitates the sensor-CPH linkups and on-demand network parameter reconfigurations. The newly introduced adapter supports the interoperability of heterogeneous sensors by eradicating the need for sensor-specific modules through OTA-based reconfiguration. The reconfiguration feature allows for new sensors to connect to an existing adapter, without changing the hardware units or any external interface. The proposed system is scalable and enables multiple sensors to connect in a network and work in synchronization with the CPH to achieve semantic and device interoperability among the sensors. We test the implementation in real-time using three different health-monitoring sensor types – temperature, pulse oximeter, and ECG. The results of our real-time system evaluation depict that the proposed system is reliable and responsive in terms of the achieved packet delivery ratio, received signal strength, and energy consumption**

***Index Terms*—Internet of Things, interoperability, universal adapter, e-healthcare, WBAN**

I. Introduction

Achieving a generalized unification of networks and devices through interoperability is one characterized by significant challenges and constraints such as cross-domain communication, heterogeneous devices, proprietary technologies, data privacy, security, and others. Often, in the absence of any domain-specific standards, considering both hardware and software, significant communication and networking challenges arise between IoT devices – both syntactic and semantic. Networked deployments in IoT, requiring communication between different devices at various layers of implementation and middleware benefit greatly from interoperable solutions, specific to domains [1]. Interoperability is the ability of devices to understand and use the data generated by one another irrespective of their vendor or hosting network. Interoperability types are generally grouped into – 1) semantic, 2) syntactic, 3) device, 4) network, 5) platform, and 6) technology [2]. The availability of present-day IoT solutions, which typically incorporate a set of heterogeneous sensors and devices [3], finds widespread usage across various domains and application areas. However, these solutions are proprietary in terms of the platform, protocols, data formats, and compatibility with other similar entities [2], [4].

IoT healthcare is one such domain that involves a multitude of sensors working in synchronization towards achieving a common goal of monitoring a subject's health over networks [5] [6]. Networked wearable-based health monitoring often involves various heterogeneous sensors and platforms interacting with one another, gathering data using pre-configured devices, accumulating the data into one platform, and parsing the data for multiple formats before transmitting them over a backbone network. This approach not only increases the computational complexities involved, but also increases the traffic on the network [7]. Additionally, such an arrangement requires the device type to be known and its ontologies predefined in the system to allow other devices to identify and understand them.

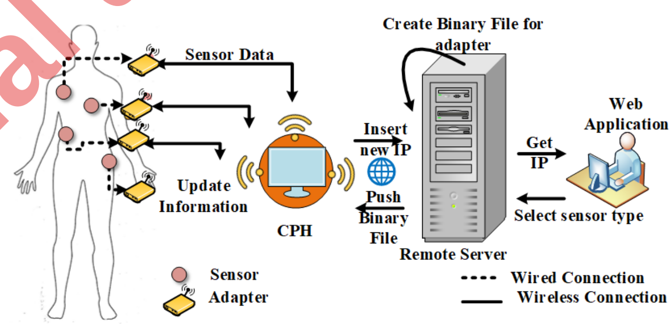


Fig. 1: An end-to-end functional overview of the proposed architecture showing different components of the system and their interconnections.

In this work, we propose Over-The-Air (OTA)-based reconfigurable IoT health-monitoring wearables to eliminate the dependency of predefined devices in an IoT-enabled health monitoring system. Similar to existing systems, multiple health monitoring sensors can be connected to a central processing hub (CPH) through wireless adapters using the proposed system and approach. Fig. 1 shows the basic outline and overview of the proposed system. The wearable proposed in the system consists of an adapter connected to a sensor unit, a battery, and a charger circuit. The adapter can wirelessly connect any health sensor type to the CPH without any specific ontology. Once the CPH accumulates data from the wearables, it processes the data before transmitting it further to a remote server. The adapter enables interoperability between the sensors and the central unit by removing the restriction of integrating any specific sensor types. The proposed system enables device interoperability, is highly scalable, and supports

TABLE I: Comparison of the proposed architecture with the existing device management schemes in IoT.

Schemes	Target devices	Network Overhead	Update frequency	Mode of operation	Focus area	References
OMADM	Mobile devices	High	Periodic	Remote	Device management	[8], [9]
TR-069	Network devices	High	Periodic	Remote	Device management	[10], [11]
LwM2M	End devices	Relatively low	Periodic	Remote	Device & data management	[12], [13]
OTA-based	Sensor adapters	Very low	On-demand	Remote	Adapter registration & update	–

multiple and new devices, which enhances the reusability of the adapters by allowing any sensor to be connected and reconfigured as per actual on-field requirements of the user. The proposed system provides the flexibility of selecting the type of sensor by the end-user through a web application. Upon successful selection, the adapter is configured using OTA with the necessary configuration files and updates to support the sensor-integrated wearable unit.

We highlight the usability of the proposed approach – a temperature sensor can just as easily integrate with the adapter as a pulse sensor, or an electrocardiogram (ECG) sensor, without the user having to program the device in the field separately, or follow a predefined connection mechanism or configuration. The information and configuration of the newly attached sensor are remotely updated to the adapter using OTA updates from a remote server through the CPH. The adapters can also be quickly and easily deployed in other non-healthcare application areas – anything requiring IoT-based wireless sensing solutions enabling cross-domain convergence.

A. Motivation

Under real-life operating conditions, it is necessary to have a system, which can accommodate the diversity in devices and systems and allows for seamless interoperability among them. The present-day solutions are not fully capable of accommodating devices and systems beyond the specified ontologies and also suffer from restricted support in terms of flexibility of including new devices and ensuring system scalability [14]. The existing wearable technologies in the market offer wireless sensing and recording of physiological data. Vendors such as Shimmer [15] and APDM Wearable Technologies [16] offer a good range of sensors for health monitoring. However, the sensors and hardware are vendor-specific and require the acquisition of new hardware units for new sensor integration. Moreover, the selection of a sensor is also limited to specific suppliers supported by the system. Some of the major lacunae of the existing systems, which motivated us to pursue this work of enabling dynamic sensor integration are:

1) Vendor-specific solutions for interoperability cannot sustain the long-term evolution of the fast-paced technological changes due to rapidly emerging technologies every day.
2) Dynamic addition of new sensors to existing systems is not feasible beyond the ontological description provided in the system.
3) The interoperability solution approaches proposed so far are specific to data formats and platforms [17], [18].

B. Contribution

Towards achieving a universal, device-independent technology, we propose an IoT healthcare monitoring framework with sensor-independent reconfigurable and reusable wireless wearables. The proposed system addresses the issue of interoperability in an IoT-enabled remote healthcare monitoring system using specially designed wearables that can integrate multiple sensor units with the system. We have made the following distinct contributions in this work:

- We propose an IoT-enabled healthcare monitoring system to support multiple on-demand sensors. The system is designed to exhibit a high degree of scalability and flexibility, which enables the connection of new undefined sensors without any knowledge of prior tags or ontologies of these sensors.
- We propose a sensor-independent reconfigurable adapter to facilitate interoperability among different sensor-integrated wearable units. These adapters are capable of receiving OTA configurations from a remote server based on its user's choice through a web-based application.
- We propose a system for the inclusion of newly defined sensors in the system without making any changes to the existing hardware units. The new sensor provisioning is done at the remote server based on user request or system update.
- We implement and evaluate the proposed system and the concept at lab-scale using three different sensors – 1) temperature, 2) pulse-oximeter, and 3) ECG. We evaluate the system based on various network parameters such as packet delivery ratio, the effect of distance and number of connected wearables on the system performance, accuracy of the data received, and the energy consumption of the wearables during their operation.

II. Related Works

In the presence of no single agreed-upon standard, both in terms of hardware and software, the communication between heterogeneous IoT devices is often difficult to achieve. Interoperability faces significant challenges with varying scenarios in different domains and industries. We discuss in details, the works, and challenges in the field of IoT healthcare concerning semantic, device, and network interoperability.

A. Semantic Interoperability

The need for semantic interoperability arises as a result of heterogeneous devices using different syntax to encode and decode their information [2]. When a sink device receives a data format from a source device that cannot be decoded into a form that is understandable by the sink, reliable communication with the source device is interrupted. Enabling semantic interoperability allows various heterogeneous devices to understand one another's data format and process it seamlessly [19]. Many works have been proposed, which use standard data formats for devices and platforms. These mainly include uniform XML and JSON data formats, enabling the parsing and retrieving of data without altering its original meaning. Health level 7 (HL7) and HL7 FHIR are standards defined to regulate and format clinical data to allow medical devices from different vendors and based on different technologies [20] [21] to interoperate with one another. The inclusion of metadata and tags in the sensor data using different ontologies is another popular means to enable semantic interoperability [22].

B. Network Interoperability

Integrating multiple heterogeneous IoT systems requires interoperability at different network and protocol layers to enable efficient functioning and conjunction [2]. Approaches such as frameworks supporting multi-layered IoT middlewares, protocols, and services are quite popular within the IoT ecosystem [23]. Interoperable mechanisms have been proposed to enable the coexistence of Named Data Networking (NDN) and Internet Protocol (IP) in IoT environments [24]. Additional network interoperability approaches such as interoperability between smart city networks and military networks (Federated Mission Networks) have proved to be crucial in disaster management, rescue missions, and other civilian-military co-operative operations [25].

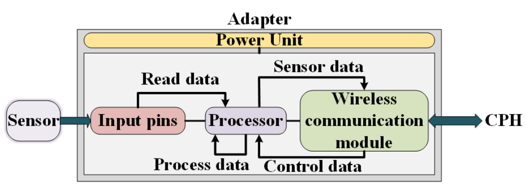


Fig. 2: Architecture of the proposed reconfigurable wearable showing it's internal components and flow of data.

C. Device Interoperability

Systems to transmit bio-medical data from the IoT devices to servers are quite complicated and the solutions to which involve multiple layers of the TCP/IP stack. The use of CoAP to integrate the ISO/IEEE 11073 and IHE PCD-01 standards [26] is one such solution. The shaping of data into a common format at the edge device level, before its transmission over the network, is yet another useful approach [27]. Device-based interoperability solutions have also been known to have incorporated smartphone-based mobile gateways [28], and model-driven enterprise monitoring solutions [29].

There are device and driver management protocols that were designed to manage connected devices in a network [30] [31]. Open Mobile Alliance-Device Management (OMA-DM) proposed by the OMA is targeted towards mobile devices with changing IPs [9]. TR-069 is yet another device management protocol by the Broadband Forum, which is widely used by the telecommunication industry for managing and provisioning their routers and gateways devices. The main target of TR-069 is a gateway and other devices with similar properties, such as VoIP phones and digital set-top boxes [11]. Both OMA-DM and TR-069 are highly structured and complex, which results in large network overheads. The large overheads and complex structure of these protocols make them heavy for an IoT end-device. A comparative analysis of the proposed system with the existing device management protocols is tabulated in Table I.

Data privacy and security are of utmost importance in IoT environment. Many frameworks and protocols have been proposed to ensure the safety of the data [32] [33] in IoT environment. However, we limit the scope of the proposed work to implementing an interoperable wearables system.

D. Synthesis

The existing systems use predefined ontologies, metadata tags, and model-driven solutions to allow communication between various medical devices. However, most of these systems are not scalable and designed for adopting the rapidly emerging newer hardware and software changes to existing systems in a remote manner. The addition of new devices would require predefining the ontologies of newer devices and technologies to make the deployed systems aware of their existence. Ontologies and metadata vary on a large scale as the size of data increases. Eventually, it becomes difficult to arrive at a global set of ontologies to cover all aspects of data gathering and exchange over the Internet. Summarily, there is a definite need for systems, which can enable dynamic changes to deployed devices system, extend interoperability, and allow for its reusability, without affecting its real-time operation.

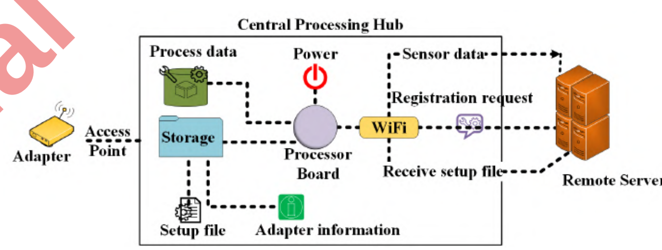


Fig. 3: Schematic view of a CPH in the proposed architecture.

III. System Model

The proposed *Reconfigure and Reuse* system consists of reconfigurable wearables, a gateway hub, and a remote server as the essential components for its operation. The wearable consists of an adapter, which is a low-power processor module with in-built WiFi capabilities (NodeMCU), a power module, and a sensor connected in a wired manner. The adapter is designed to wirelessly connect to a user-defined access point for data transfer, as shown in Fig. 2. The adapter is designed to be instantaneously usable with multiple sensor types, without any prior information about the sensor. The gateway, which we refer to as the Central Processing Hub (CPH), creates an access point to which the wearables can connect and forward their data to the Internet. The CPH, which is a Raspberry Pi unit, consists of a processing unit, memory unit, and a wireless communication unit. The CPH collects data from the wearables and forwards it to the remote server over the Internet, as shown in Fig. 3. The CPH is also responsible for accumulating data from different sensor types, identifying the sensor type based on a *Key* field and adapter ID from the received packets, and forwarding appropriate sensor data to their respective tables in the remote database. Fig. 4 shows the internal workflow of the remote server where the server hosts a web application to be accessed by an end-user. This web application enables the user to select the type of sensor for a particular wearable and reconfigure the wearable, on-demand. The complete process of wearable configuration to data collection follows a series of steps that are discussed in detail in the following sections.

A. System Architecture

The wearable consists of an adapter connected to a sensor. The adapter consists of a NodeMCU and ESP8266 module, which enables wireless transmission of data from the wearables. The CPH creates an access point for the wearables to connect and dynamically assigns IPs to all the connected wearables, and forwards the collected information to the remote server over the Internet. This IP-based Wireless Body Area Network (WBAN) allows for a significantly large number of wearables to coexist in a gateway as compared to other commercial solutions, which either use Bluetooth or wired connections. A user may access the current status of the connected devices through a web application hosted at the remote server. The web application enables the user to select the available devices and configure them according to the attached sensor. The reconfiguration data and files are first transmitted to the CPH, which then reprograms the adapter using OTA according to the user-defined configuration.

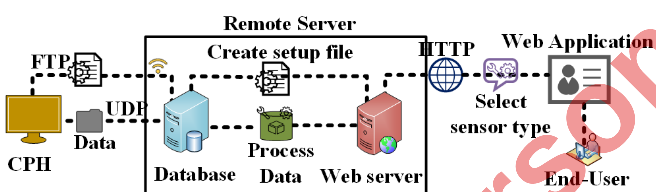


Fig. 4: A schematic view of the remote server.

B. Network Architecture

Fig. 5 shows the data flow for both control and sensed value packets, in the system using the implemented protocols, which follows a hybrid network architecture by using different communication protocols at various stages. The connection of the wearable to the CPH uses the IEEE 802.11 WiFi standards. Initially, the wearables are configured according to Algorithm 1 to send its IP and MAC address to the CPH upon a successful connection. The connected adapters' information is transferred to the remote server by the CPH. The user selects the type of sensor connected to each of the devices. This information is sent back to the corresponding CPH through a UDP-based connection. Simultaneously, an FTP-based connection is used to transfer the necessary adapter configuration files to the CPH. The CPH reconfigures the wearables through OTA using the appropriate configuration files from the remote server. After successful reconfiguration, the wearables start sensing data from the sensors and forward it to the CPH. Fig. 6 shows some sample signals received from ECG, pulse, and temperature sensor-based adapters at the CPH during its operation.

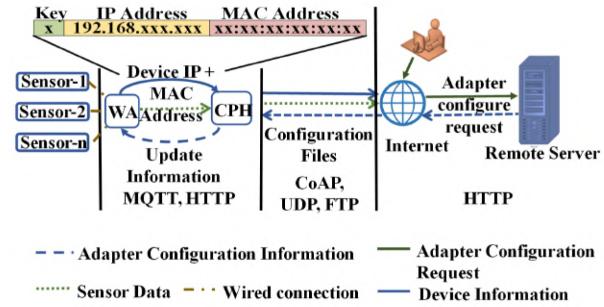


Fig. 5: Data flow and communication protocols used at various stages in the proposed architecture.

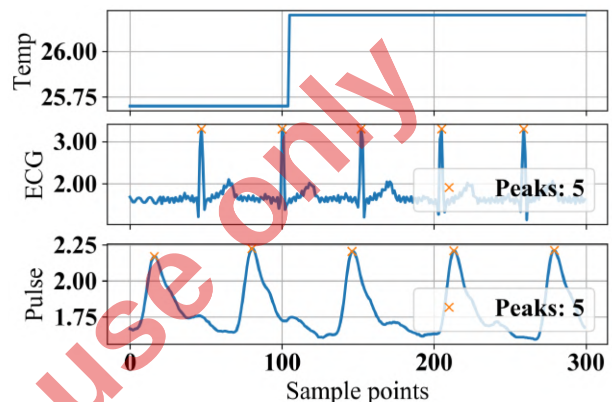


Fig. 6: Plots showing samples of the received data from different wearables.

C. Channel Model

We model the channel between the wearables' adapter and the CPH by first considering that the adapters connected to a CPH are represented as a set $A = \{a_1, a_2, a_3, \dots a_n\}$. The maximum number of adapters that can be connected to the CPH is given as n_{max}, whereas n is the number of adapters connected to CPH at any time instant such that $n \leq n_{max}$. The CPH offers a total bandwidth of β to the connected adapters at any point of time. The data-rate of each individual adapter is represented as a set $DR = \{dr_1, dr_2, dr_3, \dots dr_n\}$. The total utilized bandwidth β^{util} of the connected adapters is restricted by the available bandwidth of the CPH as,

$$\beta^{util} = \sum_{i=1}^{n} dr_i, \; \beta^{util} \leq \beta \tag{1}$$

Further, the channel link between an adapter a_i and the CPH has a channel gain of $g_i = \frac{1}{\ell_i^{\alpha}}$, where ℓ_i is the distance between a_i and CPH and α is the path loss parameter. The Signal to Interference plus Noise (SINR) ratio received at the CPH for adapter a_i is given as $\rho_i g_i \phi^{-2}$, where ρ_i is the transmission power of a_i and ϕ^2 is the Gaussian white noise power. According to Shannon capacity formula [34], the maximum allowable data-rate of each of the adapters is given as,

$$dr_i^{max} = \frac{\beta}{n} log_2(1 + SINR_i) = \frac{\beta}{n} log_2(1 + \rho_i \ell_i^{-\alpha} \phi^{-2}) \tag{2}$$

We assume that the bandwidth from the CPH is equally distributed among the connected adapters, owing to the homogeneous nature of these adapters. The total generated data packets for transmission time t can be represented as $P(t) = \beta_t^{util} \times t$.

1) Delay: The system suffers delays in data transmission due to the adapter setup and the network-related factors such as transmission delay, propagation delay, processing delay, and queuing delay.

- Transmission delay is calculated as the ratio of the total number of packets generated to the transmission rate of the adapters connected to the CPH. It is denoted as $\Delta_{trans} = (\beta_t^{util} \times t)/(\sum_{i=1}^{n} dr_i)$.
- Propagation delay of the network is defined as the ratio of the distance between the adapter and the CPH to the speed of transmission and is denoted as $\Delta_{prop} = d_i/S$, where d_i is the distance of adapter a_i from the CPH, and S is the speed of light.
- Considering the queue length at the CPH as ℓ_{queue}, the queuing delay is expressed as a product of the queue length and the transmission delay such that $\Delta_{queue} = \ell_{queue} \times (\beta_t^{util} \times t)/(\sum_{i=1}^{n} dr_i)$.
- The data at the CPH is processed before forwarding it to the remote server. Considering the processing time for each data packet as t_{proc}, the total processing time is calculated as,

$$\Delta_{proc} = t_{proc} \times \sum_{i=1}^{n} dr_i \times t \tag{3}$$

Finally, the total delay, Δ_{total}, is represented as,

$$\begin{aligned}\Delta_{tot} &= \Delta_{trans} + \Delta_{prop} + \Delta_{proc} + \Delta_{queue} \\ &= \frac{\beta_t^{util} \times t}{\sum_{i=1}^{n} dr_i}[1 + \ell_{queue}] + \frac{d_i}{S} + t_{proc} \times \sum_{i=1}^{n} dr_i \times t\end{aligned} \tag{4}$$

A one time delay is incurred by each of the adapters when connecting for the first time during its registration phase. Section IV discusses the registration phase and its associated delay in details.

IV. OTA-Based Configuration of Wearables

The proposed system uses an OTA-based reconfiguration scheme for the connected wearables. Once the wearable connects to the CPH, it sends an information packet to the CPH. Each information packet from the adapter includes the IP and MAC address of the adapter. The CPH logs the data in a local file and forwards it to the remote server. The remote server updates its database, which is reflected in the web application accessible to the end-user. The end-user can select the IP and MAC address of the connected wearable and select the type of sensor that is connected to the wearable. Once the selection process is complete, a configuration file with the desired adapter configuration is created at the server and is subsequently pushed to the CPH connected to the wearable. Meanwhile, the connected wearable scans for the configuration file in the CPH's database. Once the configuration file is located, it is pushed to the wearable's adapter through a local HTTP server using a PHP script. We refer to this entire process as the *registration phase* of the wearable. After successful configuration and registration of the wearable, it starts transmitting the data collected from the sensor to the CPH. The CPH then forwards the received data to the remote server. The delays incurred during the registration phase of the adapter can be represented as,

$$\Delta_{reg} = \Delta_{file} + \Delta_{conf} + \Delta_{update} + \Delta_{net} \tag{5}$$

where, Δ_{file} is the time taken to create the configuration file, Δ_{conf} is the delay occurred during transferring the configuration file from the remote server to the CPH, Δ_{update} is the time taken by the wearable's adapter to update itself with the new configuration file, and Δ_{net} is the delay due to regular network parameters. The information packet of size P_{info} is transmitted at an interval of t_{loop} until the CPH acknowledges it. The number of packets in the queue Q at the CPH at time instant t' is given as $P_Q = \sum_{i=1}^{n-1} \times t'$, where $n-1$ number of wearables are connected to the CPH immediately before addition of the new wearable. Assuming that the packets in the queue are successfully transmitted to the remote server, the network delays are calculated as follows,

$$\Delta_{net}^{trans} = \frac{\beta_{t'}^{util} \times t'}{\sum_{i=1}^{n-1} dr_i + \frac{P_{info}}{t_{loop}}} \tag{6}$$

such that, Δ_{net}^{trans} is the transmission delay of the network during the registration phase. Further, the queuing delay Δ_{net}^{queue} is represented as,

$$\Delta_{net}^{queue} = \ell_{queue} \times \frac{\beta_{t'}^{util} \times t'}{\sum_{i=1}^{n-1} dr_i + \frac{P_{info}}{t_{loop}}} \tag{7}$$

As the propagation and processing delays follow the same equations as discussed in the previous section, the registration delay is finally denoted as,

$$\begin{aligned}\Delta_{reg} = \frac{\beta_{t'}^{util} \times t'}{\sum_{i=1}^{n-1} dr_i + \frac{P_{info}}{t_{loop}}}[1 + \ell_{queue}] + t_{proc} \times \sum_{i=1}^{n} dr_i \times t' + \\ \frac{d_i}{S} + \Delta_{conf} + \Delta_{update} + \Delta_{file}\end{aligned} \tag{8}$$

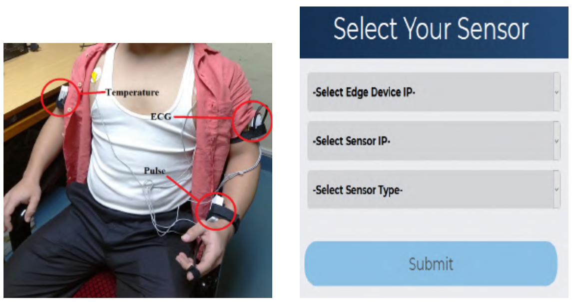
Once the wearable is reconfigured with the received configuration file, it starts sensing the physiological data and transmits to the CPH.

Algorithm 1 ASA: Adapter Setup Algorithm

```
if Adapter connected to CPH then
    Send information packet, wait for acknowledgement
    if Acknowledgment received then
        Scan for setup file
        if Setup file found then
            Update adapter and reboot
        else Resend information packet
        end if
    end if
end if
```

V. Implementation

The proposed system is implemented and tested in real-time in a laboratory environment using three different types of sensors, which include ECG, temperature, and pulse oximeter. Fig. 7 shows the 3D-printed wearables placed on the body of a subject. The wearables initially check for a connection with the CPH. Upon connection establishment, an information packet is sent to the CPH, formatted, as shown in Fig.5. Subsequently, the wearables scan the specified location for a binary file with its MAC address as the name of the file. If the desired file is located, the wearable's adapter is reprogrammed with the binary file, which also reboots the adapter before the start of sensing. If the binary file is not found, it resends the information packet to the CPH and repeats the process. The CPH incorporates Algorithm 2 and accepts connections from wearables.



(a) On-body placement of wearables (b) Web application to select the sensor type.

Fig. 7: Real-time implementation of the proposed architecture with 3D-printed wearables and web application hosted at a remote server.

Algorithm 2 SSA: Sensor-type Selection Algorithm

if If connection found and data received **then**
    Check key
    if Received packet is information packet **then**
        Send IP and MAC address to remote server
    end if
else Insert sensor data into database
end if

Depending on the type of data, information, or sensor readings (as indicated by the 'Key' field), the CPH forwards the contents of a packet to the appropriate database in the remote server. We use a Python-based UDP socket to transmit data between the remote server and the CPH. An Apache-based web server is hosted at the remote location and acts as an access medium between the user and the wearables. A web application is provided to the user to access the list of connected wearables with the CPH. A user can select the combination of CPH and a wearable's IP address along with the type of sensors to configure the wearable. Once the selected information is submitted through the web application, Algorithm 3 is executed to create the corresponding setup file at the remote server. The setup file is then pushed to the CPH using an FTP-based connection.

Algorithm 3 ARS: Algorithm for Remote Server

Inputs Sensor type, IP, and MAC address

Receive input from web application
Create setup file with selected MAC address name
Push setup file to selected IP address

VI. Evaluation

We evaluate the implemented system with three different types of sensors – pulse sensor, ECG sensor, and temperature sensor. We record the data from each sensor for a duration of 60 seconds for all the experimental conditions. Various system parameters are varied to analyze the performance of the proposed system. The parameters are as follows:

1) *Distance:* We place the wearables at a distance of $1m$, $3m$, and $5m$ from the CPH to evaluate the effect of distance on the performance of the system. These tests are performed under non-controlled conditions in a closed room environment.
2) *Connected Wearables:* We increase the number of wearables connected to the CPH to evaluate its effect on the network parameters and, eventually, the system's performance. The implementation includes three wearables with heterogeneous sensors.
3) *Sensor Types:* The implemented system is tested with three heterogeneous sensors. Each of these three sensors has a unique data signature – temperature can have a single value in a given time interval. In contrast, ECG requires a collection of values to generate intelligible information for the same time interval.

A. Performance Metrics

We use the following performance metrics to evaluate the performance of the proposed system:

1) *Wearable registration time:* The proposed wearable is reconfigurable with an OTA update. It is the time taken to send an adapter registration request to the CPH, which is forwarded to the remote server for receiving the corresponding adapter's configuration file in CPH at a specified location.
2) *Packet delivery ratio:* Packet delivery ratio (PDR) is the ratio of the number of received packets to the actual number of packets sent by a wearable. A higher PDR shows a reliable system with high accuracy.
3) *Generated data volume:* The generated data volume is the combined amount of data being pushed into the network by all the connected wearables. We analyze the data-rate of the wearables and the effect of data volume on the channel capacity.
4) *Errors in the received data:* We study the effect of the dropped packets on the received data by analyzing the peak counts of the received signals for varying distances of the wearables from the CPH.
5) *Information content:* We evaluate the data collected by the implemented system for its content of information. The purpose of this evaluation is to identify the sensors with high information content for a given period. A sensor

data with high information content is more likely to be transmitted with a higher data rate compared to a signal with low information content.

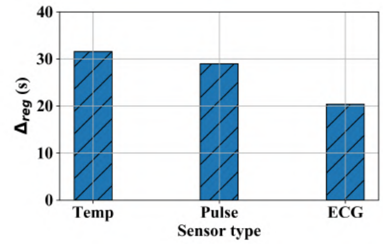


Fig. 8: Registration time of adapters integrated with different sensors.

6) *Energy consumption:* As an IoT-based healthcare device, the power consumption of the wearable is an important aspect in deciding the efficiency and reliability of the system.

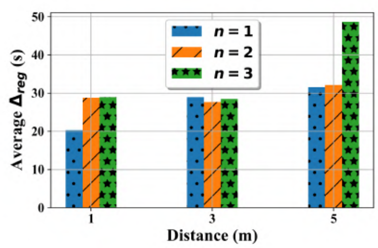


Fig. 9: Effect of the number of connected wearables on the adapter registration time with increasing distance.

B. Wearable Registration Time

Fig. 8 shows the average wearable registration time for each sensor type connected to the adapter. We attribute the difference in the registration time for the three sensors to the different configuration file sizes. The Δ_{file}, Δ_{conf}, and Δ_{update} (refer equation 5) varies with dependent header files, libraries for sensors, and ultimately affects the registration time of the wearables. In continuation, we evaluate the effect of the number of connected adapters in the network on the registration time of wearables. Fig. 9 shows the variation in the registration time with varying number of wearables at three different distances from the CPH. We observe that, beyond $3m$, the average adapter registration time increases. We attribute this gradual increase to packet collisions resulting in packet loss and the loss due to distance and physical obstructions in the implementation environment, which is evident from the increased negative RSSI values (refer Fig. 13(d)), signifying deploring signal strength between the CPH and the adapters.

C. Information Content

We analyze the information content in terms of the statistical variance of the data. This identification enables the system to optimize the data generation rate and implement selective data transmission, which utilizes the available channel bandwidth

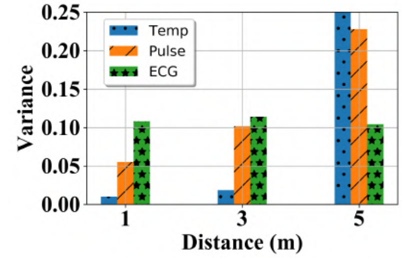
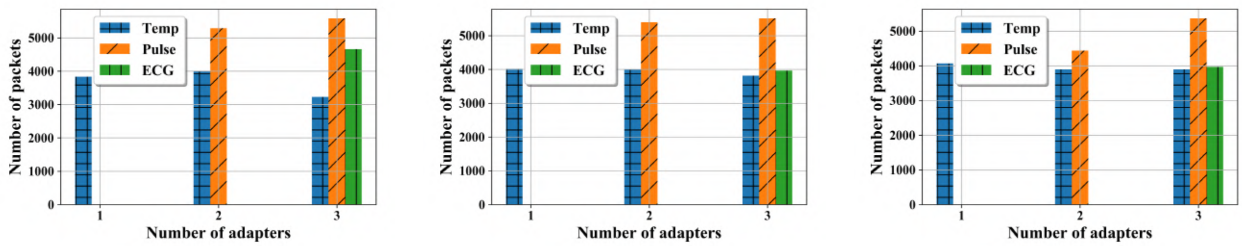


Fig. 10: Variance in the information generated from the three sensors with increasing distance from the CPH.

to the fullest by allowing more time-critical data to be sent through the channel with minimum loss of information. Fig. 10 shows the information content of the received signal. The variance of the data increases with distance due to the loss of information during data transmission caused by packet drops. As a general trend, the variance of ECG data is found to be higher than the other two sensors under similar deployment conditions due to its complex composition. This indicates higher information content and the time-critical nature of the ECG signal. However, for 5m, deviating from the trend, we observe a sudden surge in the variation of the typically low-variance temperature sensor, which results due to sensorial and adapter hardware errors, rather than network errors. Summarily, we observe that for our selected sensors up to a distance of $3m$ from the CPH, ECG has the highest variance, followed by pulse, and finally, the temperature. We also decide the criticality and timeliness of transmitting signals from these sensors in the same order.

D. Packet Delivery Ratio

We analyze the PDR for each of the sensors at varying distances. We observe in Fig. 13(c) that the overall performance of the implemented system shows high reliability with a maximum PDR of 0.996 and a minimum PDR of 0.951. We attribute this decrease in PDR to intermittent interference from external sources. The variation is also dependent on the type of sensor connected to the wearable and its packet generation rate. With a high packet generation rate, the number of successfully sent packets remains the same due to the restricted channel bandwidth, which increases the number of dropped packets. This phenomenon results in a low PDR and hampers the performance of the system. Additionally, from Fig. 11, we observe that as the number of adapters connecting to the CPH gradually increases, the previous adapter's received packets marginally decline, which is due to the sharing of the channel by more adapters. Similarly, for an increase in distance between the CPH and the adapters, we observe a marginal reduction of packets received for each sensor type due to intermittent interference from other $2.4GHz$ band ISM devices in the experimental environment. However, the implemented system exhibits a reasonably good packet delivery ratio for the three wearables, proving the reliability of the channel and the overall system. The PDR can be maintained by optimizing the rate of data generation and also selective forwarding of data based on the information content of the data.



(a) $\ell_i = 1m$ (b) $\ell_i = 3m$ (c) $\ell_i = 5m$

Fig. 11: Number of packet received at the CPH with different number of connected wearables in the network.

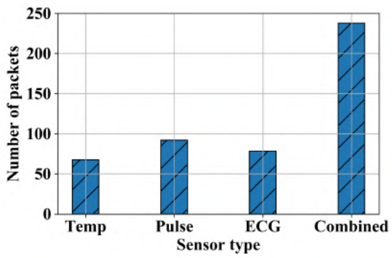
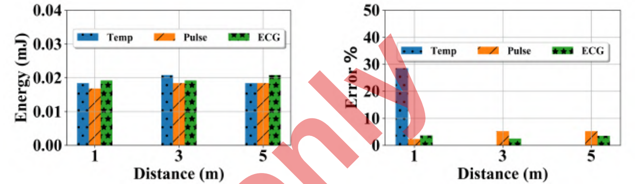
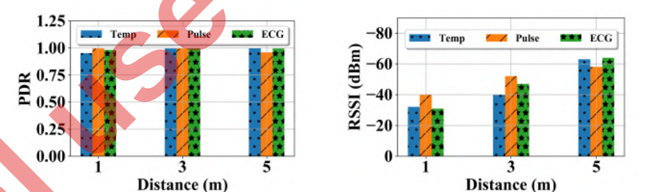


Fig. 12: Generated data volume during individual transfer of the three sensor types and their combined volume for a period of 1 second.



(a) Energy consumed by different wearables during data transmission. (b) Error percentage in the received data at the CPH.



(c) Packet delivery ratio of the CPH for varying distance. (d) Received signal strength index of different wearables.

Fig. 13: Analysis of the variations in the system parameters with increasing distance between the CPH and the wearables.

E. Data Volume

Each wearables' adapter is configured with a data rate dr_i, at which the data is read from the connected sensor and processed. This rate affects the overall data being pushed into the network of the connected wearables. An increased data rate increases the overall data volume of the system, saturating the utilized bandwidth, β^{util}. The data volume contributed by each of the sensors is shown in Fig. 12. The implemented system uses the same data rate for all the sensors to verify the information content of the data received from different sensors. However, the actual rate of generation varies for different sensors due to delays introduced by the processing Δ_{proc} for each sensor type connected to the adapter. For example, the analog to digital conversion for ECG incurs significant processing in the adapter as compared to the digital reading of the temperature values from the sensor. The combined data volume for $1s$ is significantly higher than the individual sensor types, signifying the utilization of much more of the channel's resources. Towards ensuring fair usage of the available channel bandwidth, the value of dr_i can be set according to the information content of the data being generated.

F. Errors in Received Data

We analyze the accuracy and reliability of the implemented system based on the errors in the received data. The error is measured as a decrease in the peak counts of the ECG and pulse signals, whereas the error in temperature data is measured in terms of missed transitions in temperature values. The observation in Fig.13(b) shows an average raise in error percentage with increasing distance. The overall performance of the system shows high reliability as the temperature sensor shows an error of 0% for $3m$ and $5m$ distances. The ECG and pulse sensors show an increasing error percentage with distance. The high error percentage at $1m$ distance is due to the packet loss and low PDR values of the temperature sensor. This low PDR and packet loss are linked to interference from external sources and the technical errors in the hardware performances. Also, the data volume of the network results in packet collision at the CPH, which results in packet loss. The performance of the system shows a substantial scope of improvement with regulated data generation and selected data transmission. As the proposed system uses on-body sensors, the CPH can be placed near the subject, which will help in maintaining a better link quality between the wearables and the CPH.

G. Energy Consumption

The energy consumption of the wearables during their operation is evaluated against the distance of the wearables from the CPH. The small size of the wearables makes it crucial to have a system with low-power consumption during data processing and transmission. A major portion of the energy is consumed during the data transmission to the CPH. A comparative analysis of the three adapters shows that the energy consumption of the ECG sensor is slightly higher than the temperature and pulse sensor. We attribute this to the processing of the ECG data at the wearable's adapter before transmitting it to the CPH. Also, the data generation rate of the ECG signal is higher, as previously discussed. This results in more number of transmissions, resulting in overall high energy consumption. The energy consumption is also affected by the signal strength of the connected network. In the implemented system, the temperature sensor shows a sudden rise in energy consumption at a distance of $3m$. We attribute this behavior to the transmission rate of the sensor and also the received signal strength. Fig. 13(a) shows that the energy consumption of the adapters is in the range of $0.02mJ$ for a period of $60s$. A $300mAH$ Lithium-ion battery is estimated to operate a wearable for a period of about 270 hours in a single charging cycle.

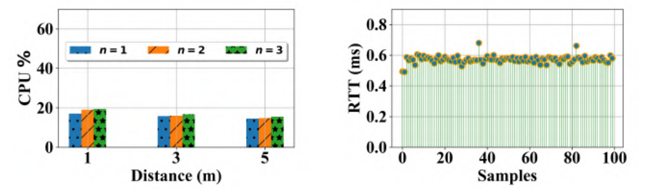
H. CPH Performance

The parameters evaluated so far are for the connection between the CPH and the wearables. Here, we evaluate the performance of the CPH and the connection between the CPH and the remote server. We test the processor usage of the CPH during data reception for a varying number of adapters at different distances. The processor usage mainly depends on the combined volume of data being received at the CPH. Eventually, a higher PDR value will mean high processor usage. Fig. 14(a) shows that with increasing distance, the processor usage decreases, while the processor usage increases for an increasing number of wearables. We attribute this phenomenon to the generated data volume and packet loss. As the number of wearables, n, increases, the additional generated data contributes to the overall data volume being pushed in the network. As a result, the number of data packets at the CPH increases, resulting in higher consumption of the CPH's processing resources. Considering the distance factor, an increase in distance between the CPH and the wearables decreases the PDR due to the channel's path loss and reduced link quality. The number of packets lost is more as compared to packet loss at a smaller distance. The packet loss eventually contributes to less processing at the CPH.

Further, we evaluate the link quality between the CPH and the remote server by conducting a ping test from the CPH to the remote server. The test is repeated for 80 samples, which is shown in Fig. 14(b). The maximum Round Trip Time (RTT) is observed to be $0.68ms$. The RTT signifies the network delay for a two-way transfer. Hence, the system suffers a maximum network delay of $0.34ms$ for data transmission between the CPH and the remote server. The network delay is dependent on the network quality of the access point and is bound to vary with different connection source.

I. Interoperability

The sensor modules used in the implemented system vary in terms of device vendors, pin configuration, and data transmission protocols. The OTA updates allow the same adapter to be reconfigured and used for different sensors, enabling device interoperability between the CPH and the wireless adapter. This feature of our system enables it to include a wide range and variety of sensors to the same underlying hardware framework. Additionally, the OTA configuration enables the adapter to transmit data to the CPH in readable formats irrespective of the sensor type and vendors, which enables semantic interoperability in the proposed system.



(a) Processor usage % of CPH with varying number of wearables and distance.

(b) Round trip time for network link between the CPH and the remote server.

Fig. 14: Performance analysis of the CPH's processor usage and its communication link with the remote server.

VII. Conclusion

The paper proposed reconfigurable wearables to incorporate interoperability in an IoT-based healthcare monitoring framework. The wearables can be reconfigured remotely, based on the type of sensor connected to it. The proposed system is implemented in real-time with three wearable adapters connected to three different sensors – ECG, temperature, and pulse. The results of extensive evaluations show high accuracy and reliability of the system, with marginal differences concerning intermittently varying network conditions. The performance evaluation also highlights possible areas that can be targetted for an increase in the efficiency of the system further.

In the future, we plan to evaluate the scalability over a large-scale implementation of the system. More advanced and robust communication protocols, along with proper scheduling and a slotting mechanism, will be implemented for the optimization of data transmission. Additionally, the security and privacy of the collected data will also be taken as the future scope of the proposed work.

VIII. Acknowledgement

The authors gratefully acknowledge SensorDrops Networks Pvt Ltd for lending their healthcare monitoring IoT platform for implementing the work reported in this paper. Additionally, the third author acknowledges the funding support received from INAE (Sanction letter no. INAE/121/AKF, Dt. 13-02-2019), and SERB/IMPRINT-II (Sanction letter no SERB/F/12680/2018-2019;IMP/2018/000451, Dt. 25-03-2019) for executing parts of this project.

References

[1] T. Fronimos, M. Koutsoubelias, S. Lalis, and T. Bartzanas, "A Service-Based Approach for the Uniform Access of Wireless Sensor Networks and Custom Application Tasks Running on Sensor Nodes," in *Integration, Interconnection, and Interoperability of IoT Systems*. Springer, 2018, pp. 77–101.

[2] M. Noura, M. Atiquzzaman, and M. Gaedke, "Interoperability in internet of things: Taxonomies and open challenges," *Mobile Networks and Applications*, vol. 24, no. 3, pp. 796–809, Jun. 2018.

[3] L. Atzori, A. Iera, and G. Morabito, "The internet of things: A survey," *Computer networks*, vol. 54, no. 15, pp. 2787–2805, 2010.

[4] G. Hatzivasilis, I. Askoxylakis, G. Alexandris, D. Anicic, A. Bröring, V. Kulkarni, K. Fysarakis, and G. Spanoudakis, "The Interoperability of Things: Interoperable solutions as an enabler for IoT and Web 3.0," in 23^{rd} *International Workshop on Computer Aided Modeling and Design of Communication Links and Networks (CAMAD)*. IEEE, Sep. 2018, pp. 140–147.

[5] S. R. Islam, D. Kwak, M. H. Kabir, M. Hossain, and K.-S. Kwak, "The internet of things for health care: a comprehensive survey," *IEEE Access*, vol. 3, pp. 678–708, 2015.

[6] Y. Bhatt and C. Bhatt, "Internet of things in healthcare," in *Internet of things and big data technologies for next generation HealthCare*. Springer, 2017, pp. 13–33.

[7] S. Din and A. Paul, "Smart health monitoring and management system: Toward autonomous wearable sensing for internet of things using big data analytics," *Future Generation Computer Systems*, vol. 91, pp. 611–619, 2019.

[8] "OMA Device Management Protocol Approved Version 1.3 24 May 2016," Accessed: 2019-07-30. [Online]. Available: http://www.openmobilealliance.org/release/DM/V1_3-20160524-A/OMA-TS-DM_Protocol-V1_3-20160524-A.pdf

[9] C. Guangyu, C. Jian, G. Haisheng, G. Zhipeng, and Q. Xuesong, "OMA-DM based mobile device diagnostics and monitoring mechanism," in *Global Mobile Congress*, Oct. 2009, pp. 1–6.

[10] "TR-069 CPE WAN Management Protocol Issue: 1 Amendment 5 IssueDate: November2013 CWMPVersion: 1.4," Accessed: 2019-07-30. [Online]. Available: https://www.broadband-forum.org/download/TR-069_Amendment-5.pdf

[11] B. A. G. Hillen, I. Passchier, E. F. Matthijssen, F. T. H. d. Hartog, and F. Selgert, "Remote management of mobile devices with broadband forum's tr-069," in *The* 13^{th} *International Telecommunications Network Strategy and Planning Symposium*, Sep. 2008, pp. 1–19.

[12] "Lightweight Machine to Machine Technical Specification: Core," Accessed: 2019-07-30. [Online]. Available: http://www.openmobilealliance.org/release/LightweightM2M/V1_1-20180710-A/OMA-TS-LightweightM2M_Core-V1_1-20180710-A.pdf

[13] I. F. Trentin, S. Berlemont, and D. A. C. Barone, "Lightweight M2M protocol: Archetyping an IoT device, and deploying an upgrade architecture," in *IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops)*, Mar. 2018, pp. 403–408.

[14] T. Adame, A. Bel, A. Carreras, J. Melià-Seguí, M. Oliver, and R. Pous, "CUIDATS: An RFID–WSN hybrid monitoring system for smart health care environments," *Future Generation Computer Systems*, vol. 78, pp. 602–615, 2018.

[15] "Shimmer: Discovery in Motion," Accessed: 2019-07-30. [Online]. Available: http://www.shimmersensing.com/

[16] "APDM Wearable Technologies," Accessed: 2019-07-30. [Online]. Available: https://www.apdm.com/

[17] A. Bröring, A. Ziller, V. Charpenay, A. S. Thuluva, D. Anicic, S. Schmid, A. Zappa, M. P. Linares, L. Mikkelsen, and C. Seidel, "The BIG IoT API - Semantically Enabling IoT Interoperability," *IEEE Pervasive Computing*, vol. 17, no. 4, pp. 41–51, Oct. 2018.

[18] P. Pace, G. Aloi, G. Caliciuri, R. Gravina, C. Savaglio, G. Fortino, G. Ibanez-Sanchez, A. Fides-Valero, J. Bayo-Monton, M. Uberti, M. Corona, L. Bernini, M. Gulino, A. Costa, I. De Luca, and M. Mortara, "INTER-Health: An Interoperable IoT Solution for Active and Assisted Living Healthcare Services," in 5^{th} *IEEE World Forum on Internet of Things (WF-IoT)*, Apr. 2019, pp. 81–86.

[19] S. D. Nagowah, H. B. Sta, and B. Gobin-Rahimbux, "An Overview of Semantic Interoperability Ontologies and Frameworks for IoT," in 6^{th} *International Conference on Enterprise Systems (ES)*, 2018, pp. 82–89.

[20] M. Azarm and L. Peyton, "An Ontology for a Patient-Centric Healthcare Interoperability Framework," in *IEEE/ACM International Workshop on Software Engineering in Healthcare Systems (SEHS)*, May 2018, pp. 34–41.

[21] D. Bender and K. Sartipi, "Hl7 fhir: An agile and restful approach to healthcare information exchange," in *Proceedings of the* 26^{th} *IEEE international symposium on computer-based medical systems*, 2013, pp. 326–331.

[22] F. Ullah, M. A. Habib, M. Farhan, S. Khalid, M. Y. Durrani, and S. Jabbar, "Semantic interoperability for big-data in heterogeneous IoT infrastructure for healthcare," *Sustainable cities and society*, vol. 34, pp. 90–96, 2017.

[23] G. Fortino, C. Savaglio, C. E. Palau, J. S. de Puga, M. Ganzha, M. Paprzycki, M. Montesinos, A. Liotta, and M. Llop, "Towards multi-layer interoperability of heterogeneous IoT platforms: the INTER-IoT approach," in *Integration, Interconnection, and Interoperability of IoT Systems*. Springer, 2018, pp. 199–232.

[24] J. Quevedo, R. Ferreira, C. Guimarães, R. L. Aguiar, and D. Corujo, "Internet of Things discovery in interoperable Information Centric and IP networks," *Internet Technology Letters*, vol. 1, no. 1, 2018.

[25] M. Pradhan, N. Suri, C. Fuchs, T. H. Bloebaum, and M. Marks, "Toward an architecture and data model to enable interoperability between federated mission networks and IoT-enabled smart city environments," *IEEE Communications Magazine*, vol. 56, no. 10, pp. 163–169, Oct. 2018.

[26] W. Li, C. Jung, and J. Park, "IoT Healthcare Communication System for IEEE 11073 PHD and IHE PCD-01 Integration Using CoAP," *KSII Transactions on Internet & Information Systems*, vol. 12, no. 4, pp. 1396–1414, 2018.

[27] P. Pace, G. Aloi, R. Gravina, G. Caliciuri, G. Fortino, and A. Liotta, "An edge-based architecture to support efficient applications for healthcare industry 4.0," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 1, pp. 481–489, Jan. 2019.

[28] G. Aloi, G. Caliciuri, G. Fortino, R. Gravina, P. Pace, W. Russo, and C. Savaglio, "Enabling IoT interoperability through opportunistic smartphone-based mobile gateways," *Journal of Network and Computer Applications*, vol. 81, pp. 74–84, 2017.

[29] A. Ahmed, M. Kleiner, and L. Roucoules, "Model-Based Interoperability IoT Hub for the Supervision of Smart Gas Distribution Networks," *IEEE Systems Journal*, vol. 13, no. 2, pp. 1526–1533, Jun. 2019.

[30] M. H. Elgazzar, "Perspectives on M2M protocols," in 7^{th} *IEEE International Conference on Intelligent Computing and Information Systems (ICICIS)*, Dec. 2015, pp. 501–505.

[31] S. K. Roy, S. Misra, and N. S. Raghuwanshi, "Senspnp: Seamless integration of heterogeneous sensors with iot devices," *IEEE Transactions on Consumer Electronics*, vol. 65, no. 2, pp. 205–214, May 2019.

[32] D. Miorandi, S. Sicari, F. De Pellegrini, and I. Chlamtac, "Internet of things: Vision, applications and research challenges," *Ad hoc networks*, vol. 10, no. 7, pp. 1497–1516, 2012.

[33] S. Sicari, A. Rizzardi, D. Miorandi, C. Cappiello, and A. Coen-Porisini, "Security policy enforcement for networked smart objects," *Computer Networks*, vol. 108, pp. 133–147, 2016.

[34] C. E. Shannon, "Communication in the presence of noise," *Proceedings of the IRE*, vol. 37, no. 1, pp. 10–21, 1949.