Journal of
Sensor and
Actuator Networks

MDPI

Article

Resilient Green Cellular IoT for Landslide Monitoring Using Voice Channels

Sangeeth Kumar [1,*], Subhasri Duttagupta [2] and Venkat P. Rangan [1]

[1] Center for Wireless Networks and Applications (WNA), Amrita Viswa Vidyapeetham, Amritapuri 690525, India; venkat@amrita.edu
[2] Department of Computer Science and Engineering, Amrita Viswa Vidyapeetham, Amritapuri 690525, India; subhasrid@am.amrita.edu
* Correspondence: sangeethk@am.amrita.edu

Abstract: A wide-scale outdoor remote deployment involves a large number of low-cost nodes that are powered by green energy, such as solar. We deal with such a system for landslide monitoring where the tiny nodes with ultra-low memory as little as 2 KB are directly connected to the Internet using cellular networks, thereby constituting Cellular IoT's (C-IoT). This makes them vulnerable to a wide range of Denial of Service (DoS) attacks during their collaborative communications. Further, due to memory constraints, the nodes are not able to run resource-hungry security algorithms. Existing IoT protocols also cannot offer resiliency to DoS attacks for these memory-constrained devices. This paper proposes the Voice Response Internet of Things (VRITHI), which addresses the above issues by using the voice channel between the nodes. To the best of our knowledge, this is the first solution in the IoT domain where both the voice and data channels are being used for collaborative communications. Evaluation results demonstrate that VRITHI is able to reduce external DoS attacks from 82–65% to less than 28% and improves real-time communications in such a memory-constrained environment. In addition, it also contributes to green IoT energy saving by more than 50% in comparison with other IoT protocols.

Keywords: Internet of Things; green IoT; real-time; sensor applications; landslides

check for updates

Citation: Kumar, S.; Duttagupta, S.; Rangan, V.P.; Resilient Green Cellular IoT for Landslide Monitoring Using Voice Channels. *J. Sens. Actuator Netw.* **2021**, *10*, 59. https://doi.org/10.3390/jsan10030059

Academic Editor: Lei Shu

Received: 31 July 2021
Accepted: 6 September 2021
Published: 17 September 2021

1. Introduction

In recent years, Internet of Things is experiencing a greater push to provide green solutions for a sustainable environment. Green Internet of Things is more relevant in remote outdoor areas without easy access to electricity. Further, a large scale IoT deployment for practical applications [1,2] involving hundreds of nodes can drastically reduce the energy consumption by adopting Green IoT.

For remote monitoring and data collections, the IoT nodes mostly connect through a resource-rich gateway in the field to the data center. However, setting up a Local Area Network through a gateway is quite challenging in a hilly site [3,4] for reasons, such as steep terrain, higher vegetation density, channel conditions, inadequate Fresnel Zone clearance [5], and coverage issues. Hence, the Local Area Network protocols, such as RPL [6] (Routing Protocol for Low-Power and Lossy Networks), that is based on IEEE 802.15.4 are not suitable to use in our environment. Alternatively, the IoT nodes in large sites can directly connect to cellular networks. Due to its pervasive and superior connectivity compared to other wireless solutions, several recent works [7,8] have advocated the use of cellular networks to connect IoT devices in a large-scale deployment. This new paradigm of connecting IoT devices is referred to as C-IoT.

This paper deals with such a C-IoT deployment in hilly terrains for a multi-site landslide early warning system. The deployed nodes are directly connected to the Internet, which makes them extremely susceptible to various Denial of Service (DoS) attacks from public networks during real-time collaborative communications. Our first landslide deployment site, sprawling around 7 acres, uses an IoT gateway and a LoRa Local Area Networks, along with a fog computing framework [9], for local communications among the nodes. The IoT gateway protects the nodes from external DoS attacks. On the other hand, the second site sprawls a large area of 72 acres (Figure 1) and uses C-IoT for communications. The Tier 1 nodes in Figure 1 monitor crown part of the hill, whereas Tier 2 and Tier 3 nodes monitor the middle and toe part of the hill, respectively.

1.1. Challenges

We faced many challenges for real-time communications among the C-IoT nodes deployed on this site, and some of the challenges are given below:

1. The nodes in the site use solar power for functioning during the daytime and at the same time harvest the energy. The energy harvesting is used to power the node at night. Being a hilly area with heavy rains, there is always a lack of sunlight. To keep the energy consumption *low*, we choose ultra-low-power nodes with only a few KB RAM. *Resource constraints* on the nodes prevent them from running popular security protocols, as mentioned in References [10,11], including the most recent one [12]. Moreover, existing AI and machine learning algorithms requiring more memory for packet anomaly detection in case of DoS attacks cannot be used on these nodes. Hence, the solution must work on a low memory hardware.

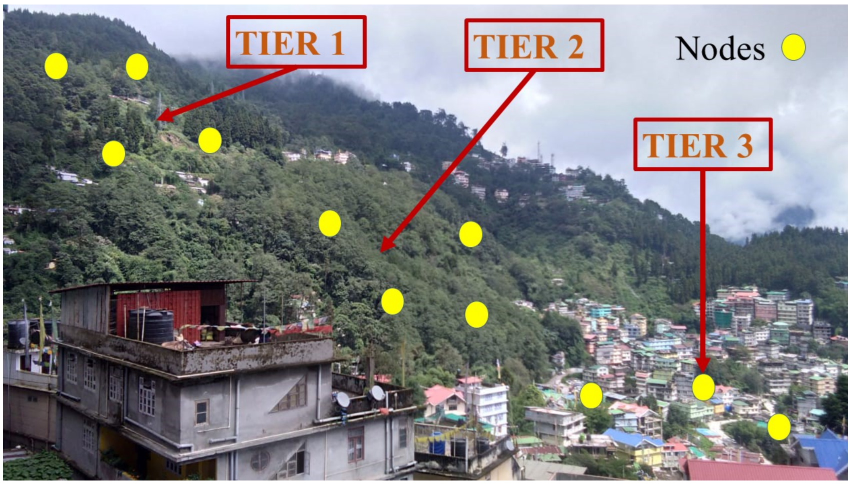


Figure 1. Landslide deployment site in a hilly terrain.

2. The nodes in the large deployment site may use *heterogeneous carriers* to connect to their cellular networks. Though there are some solutions [13] using deep convolutional neural networks (CNN) for detecting DoS attacks, there are no inbuilt security mechanisms against DoS attacks in the cellular hardware. In addition, the end-users may be unaware of security solutions adopted by different carriers. Hence, the solution requires to be both carrier and cellular hardware agnostic in reducing DoS attacks.
3. The C-IoT nodes are connected with various sensors, such as moisture, pore pressure, and movement sensors, and they collaboratively *communicate with each other in real-time* to track complex phenomena, such as landslides.
   Two prevalent cloud-based node interaction models in the IoT domain are: Request-Response and Publish-Subscribe [14].
   (a) In *Request-Response* (denoted by R/R), the client node sends a message to the message-broker on the cloud. The receiver node connects to the broker periodically to check for messages and fetches them. The drawback of this method is that periodical polling with the broker results in non-real time communication between the nodes.

(b) In the *Publish-Subscribe* (denoted by P/S) mechanism, the client IoT node sends a message to the message-broker on the cloud. The receiver nodes subscribe to the message-broker and listen for incoming messages on an open port in the public network. The message-broker broadcasts these messages to the subscribers when it receives the message. This mechanism provides real-time communication between two nodes but makes the nodes vulnerable to DoS attacks in public networks due to continuous listening for incoming messages.

1.2. Contribution

Hence, there is a need for a practical, carrier, and hardware independent deployment solution for DoS attacks in C-IoT that facilitates Green IoT nodes with very low memory in performing sustained real-time collaborative communications among themselves.

The communication mechanism proposed in this paper is referred to as the Voice Response Internet of Things (VRITHI). VRITHI proposes a new category of interaction model called dual-channel communication, which multiplexes voice and data channel. It neither polls nor listens continuously, such as request-response or publish-subscribe mechanisms. It is interrupt-driven, provides real-time communication, and incurs low power consumption. Further, qualitative feature analysis of VRITHI, along with a few prominent IoT protocols, is done in Table 1. The solution is applicable even to tiny Green IoT devices with ultra-low memory and low power hardware, as shown in Figure 2.

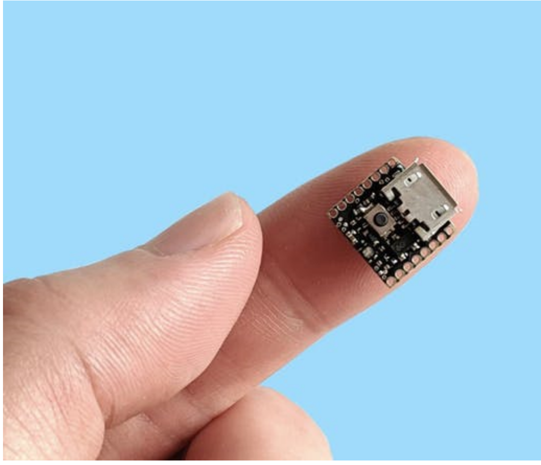


Figure 2. Arduino Atto Board with 2 KB RAM, 3.3 V operating voltage.

The main contributions of this paper in the order of priority are as follows:

- VRITHI provides a higher resilience towards DoS attacks during collaborative interactions of nodes. The DoS attack reduces from 82–65% to less than 28% in the nodes.
- The proposed solution operates in an ultra-low memory scenario and still results in improved real-time communications than the existing protocols, such as MQTT, AMQP, XMPP, and CoAP. Moreover, it increases green IoT energy savings by more than 50% in comparison with these protocols.
- Due to coverage issues and poor connectivity in the mountains during the rainy season, the cellular network falls back from 4G or 3G network to 2G only network without any IP-address connection. During this time, our system is able to send selective sensor data through voice channel to the data center.

Table 1. IoT protocols characteristics (P/S): Publish/Subscribe, (R/R): Request/Response.

Protocol	Type	Real-Time	Vulnerability Level	Power Consumption
MQTT	P/S	Y	High	High
AMQP	R/R,P/S	N	Low	Moderate
CoAP	R/R,P/S	N	Low	Moderate
XMPP	R/R,P/S	N	Low	Moderate
VRITHI	Dual Channel	Y	Low	Low

2. Related Work

Since DoS attacks in IoT nodes is a well-analyzed topic, we first discuss a few comprehensive reviews that cover more than 225 contributions in this area. Then, we consider research articles that examine DoS attacks specific to resource-constrained devices. Finally, we provide a comparative analysis of these papers based on a number of important features in Table 2.

Neshenko et al. [10] review various state of art work in IoT security and highlight research trends in IoT security. They also present a unique taxonomy by comparing and analyzing nearly 100 research contributions in this field. However, none of these papers address green IoT with ultra-low resources while enhancing Industrial IoT security to minimize DoS attacks. Centikiya et al. [15] provides an overview of the recent denial of service attacks in networked control systems. It provides an outlook based on game theory and uses optimization, and probabilistic techniques for modeling these attacks. Another work [16] studies distributed DoS attacks in IoT devices as well as in the cloud environment. They also discuss various multi-vector DDoS attacks.

Y. Lu et al. [11] review the current research topics of cyber-security in each of the layers, such as sensing layer, networking layer, middleware layer, and application layer. Another comprehensive survey on various security threats on real IoT devices and vulnerabilities in IoT technologies is presented in Reference [17]. The authors do a comparison of IoT technologies with respect to security parameters, like integrity, anonymity, confidentiality, access control, authentication, etc. The work in Reference [18] gives an overview of various security considerations in real-world applications, such as landslide monitoring and warning system.

The work in Reference [19] uses statistical learning methods to determine the device behavior and flag the deviations as anomaly packets in resource constraint devices. The methods are evaluated using Beagle Bone blackboards with 512 MB RAM and dual micro-controller architecture. However, for ultra-low resource hardware, such as 2 KB of RAM and 8 Mhz of a micro-controller, these methods are not applicable, and a simpler method to reduce DDoS attack is required.

A method to defend large-scale DDoS attacks in resource-constrained devices is described in Reference [20], where a third party algorithm is introduced to handle the risk. The algorithm is implemented at the gateway router, and the gateway handles low rate attacks (Blacklist, Greylist) and filters them. However, in our case, the nodes are densely deployed and connected to open public Internet without any gateway or firewall. Mouratidis et al. [21] propose a method called security-by-design for the Industrial Internet of Things that runs in two different levels-modeling and simulation. However, the entire software architecture needs high processing speed and memory.

The work in Reference [22] discusses possible attacks, such as DoS and DDoS, that can happen in resource-constrained devices. The designed algorithm is implemented at a 6LoWPAN Border Router to filter out all the malicious junk packets at an early stage itself. The proposed method deals with two algorithms called Primary-check in which incoming packets with source IPs belonging to the blacklisted IPs are dropped, and, in Secondary-check, packets from all the non-blacklisted IPs (Grey listed IP) undergo a payload inspection for detecting DoS and DDoS attacks. The method is suitable wherever an access gateway or firewall is placed as a proxy for the IoT devices. In contrast, in our case, the devices are connected directly to the Internet.

Table 2. A comparison of security-related approaches in green IoT.

Ref.	Description	Node Memory Usage > 2 KB	Node Power	C-IoT	Require Gateway or Firewall?	Gateway Resource	Reason for not Using
[23]	Security in green fog and cloud computing	N	Low	N	Y	High	Requires gateway for Internet
[19]	Learning methods to characterize the packets anomalies	Y	Moderate	Y	N	NA	Higher node memory
[24]	Green IoT in data centers for switches and routers	N	Low	N	Y	High	Requires gateway for Internet
[21]	Security framework for IoT node	Y	High	Y	N	NA	Higher memory and processing power
[20,22]	Filtering DDoS packets in local gateway connecting GreenIoT devices	N	Moderate	N	Y	Moderate	Requires gateway or border router for Internet
[25,26]	Hierarchical key policy encryption (H-KP-ABE) for IoT Health care	Y	Moderate	Y	N	NA	High memory and moderate power

After analyzing various research and review papers, we find that none of the approaches meet the requirement of handling DoS attacks in the C-IoT nodes with less than 2 KB of RAM that are directly connected to the Internet. Hence, we came up with the design of *voice response Internet of Things (VRITHI)* to meet the requirements in our deployment. We do a qualitative comparison of VRITHI with selective state of the art work on IoT security for resource-constrained devices. In Table 2, we list down important features of these papers and provide the reasons why their solutions cannot be applied in our scenario.

Y. Fang et al. [27] explain the major advantages and disadvantages of symmetric and asymmetric PSK systems in resource-constrained devices for mitigating the attacks. Even though the key-based approach is a better authentication technique, it is not favorable in practical situations due to computational and communication complexities. Moreover, it leads to new DoS attacks in the network due to authentication and power depletion. Bogus message injection, such as DDoS attack, can also happen in WSN networks. However, the proposed scheme also requires more memory and mandates powerful hardware that is not suitable for Green IoT.

The authors in Reference [26] develop a lightweight authentication protocol that could function with a similar hardware as ours for cloning attacks but on 802.15.4 modules. The cloning attack is different from DoS attack and requires physical access to hardware to clone the device. Further, the authentication scheme does not consider flooding attacks on the IP stack. The author also state that deploying the traditional security models with such resource constraint devices with less than 2 Kb RAM and 32 KB program memory is not a feasible option.

The review article, in Reference [7], discusses challenges in providing green solutions for cellular IoT where IoT devices are connected to cellular networks, especially in 5G network. Reference [8] mentions different machine learning-based solutions for connecting IoT devices with cellular network. However, resource constraints on our device prevent them from adopting these solutions. Techniques attaining *Green Internet of Things* are discussed in References [23,24]. The work in Reference [24] proposes a technique for green Industrial-Tactile IoT that employs deterministic traffic flows in layer-2 and -3 with switches and routers creating an intrusion detection system. The strategy in Reference [23] achieves a green environment using fog computing in a cloud network. It also introduces a security *wall* between the cloud network and the Internet to protect the privacy of data.

Additional papers in the area security and *real-time communications* in practical scenarios are References [9,25]. Reference [25] describes a lightweight key policy attribute-based encryption scheme (H-KP-ABE) for packet exchanges in an IoT health care system. Reference [9] deals with real-time communications using IoT network to monitor and establish a landslide warning system. The authors in Reference [28] provide a classification algorithm to detect forged nodes that are compromised and in turn perform machine-to-machine DoS attacks. Reference [29] discusses the prevention and detection of ARP spoofing and poisoning of IP addresses. *Ethical hacking* on cyber-physical systems and LTE networks is described in References [30,31].

3. VRITHI: System Design and Characteristics

This section discusses the DoS attack scenario in our environment and system setup of the nodes, and then it presents the design of VRITHI protocol, along with its advantages.

3.1. DoS Attack Scenario in VRITHI

In this section, we mention the nodes in the deployment and their vulnerability to security attacks. The IoT nodes deployed in our scenario are directly connected to the cellular network, as shown in Figure 3. The IoT nodes have two main components: a sim5300E wireless cellular OEM (original equipment manufacturer) hardware module and a small controller (with only a few bytes of RAM). The wireless module connects the controller to the voice and data connection of cellular provider. This is the typical deployment scenario used in field implementations of IoT testbed without a proxy gateway.

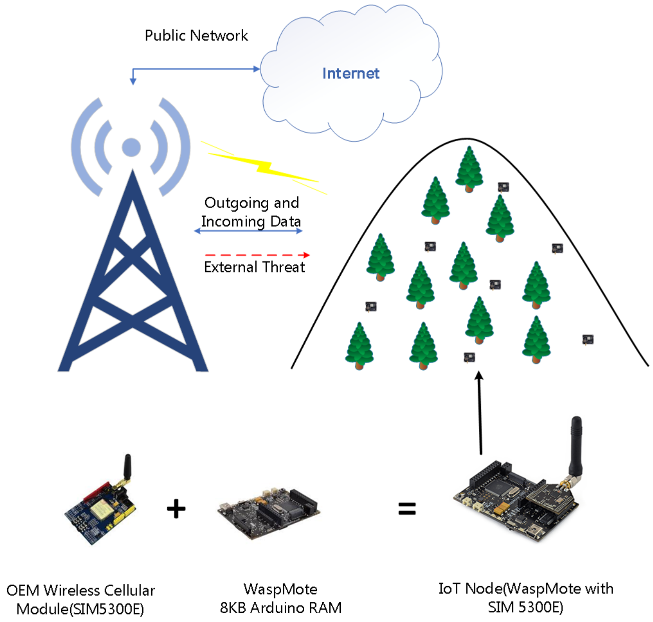

Public Network
Internet
Outgoing and Incoming Data
External Threat
+
=
OEM Wireless Cellular Module(SIM5300E)
WaspMote 8KB Arduino RAM
IoT Node(WaspMote with SIM 5300E)

Figure 3. DoS attack scenario of IoT nodes directly connected to Cellular Network.

During the node startup, the modem is connected through the voice channel of the cellular provider. However, it does not suffer from DoS attacks at this time due to the non-availability of data connection. Later on, as the controller requires to send data, it requests the modem for Internet connection. Subsequently, the modem obtains an IP address from the cellular provider and gets exposed to the public Internet. The botnets on the Internet scan the IP addresses and possible vulnerable ports open in the modem, resulting in DoS/DDoS attack whenever the ports are not protected.

In addition, attacks, such as TCP SYN Flood attack, affect the TCP stack, whereas they do not affect the voice connections since the voice communication is handled by GSM stack in the modem. This aspect is employed in the design of VRITHI, where the voice network between the IoT nodes acts as an overlay over the IP network. Without the voice network, it is not possible for other nodes to know whether the peer node is affected by DoS attack or not. The nodes could only find that the peer (affected) node is non-responsive.

3.2. System Setup

Figure 4 illustrates three nodes that are deployed at different tiers as mentioned in Figure 1. Each of the nodes in the field has four key components: (1) IoT module, (2) sensor interfacing circuit, (3) power harvesting system, and (4) sensors. The IoT module uses the wasp motes from Libelium, Inc. (libelium.com, Zaragoza, Spain) integrated with a tiny sim5300E modem. The reason for choosing this module is that it has many add-on industrial interfaces, such as RS485, Modbus, etc., and also has many sensors components off the shelf for future use. The IoT module has a processing speed of 16 MHz and 8 KB of RAM. The node has an ultra-low power consumption of 7 µA during hibernation, 30 µA during sleep, and 17 mA during the active state. The sim5300E modem connects the cellular network to the IoT node, and the sensor interfacing circuit helps the IoT node to communicate with sensors. The modem has both voice and data connection functionalities. In the node, sensor drivers use 4 KB of RAM in the controller; processing routines use 2 KB leaving only 2 KB for network routines. The sensors used are pore pressure sensors, movement sensors, and moisture sensors. The sensors are buried inside the soil to sense deep earth movements and characteristics. The power system uses solar energy to function the node. It harvests energy for operating during the nights and rainy conditions.

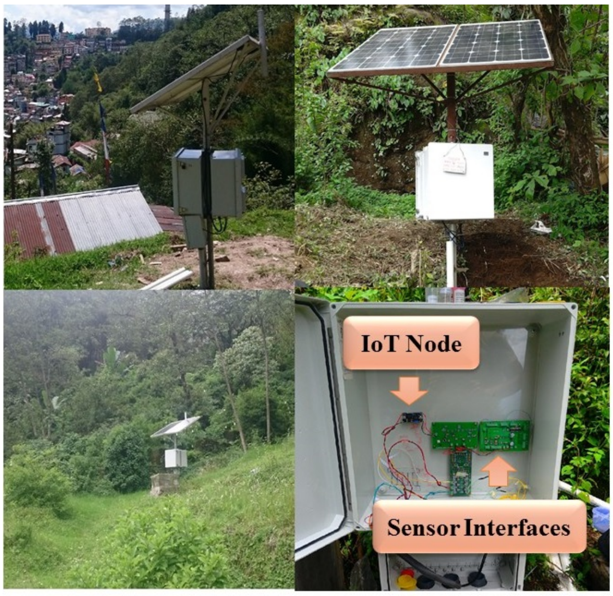


Figure 4. Field nodes (top-left to bottom-right): Node 2 (Tier1), Node 5 (Tier2), Node 10 (Tier3), and their components.

3.3. Data Flow in VRITHI

In this section, we elaborate on the design of VRITHI in terms of the control channel and the data channel. The reference diagram of the data flow using these two channels of the Telecom network is shown in Figure 5. It shows that the control channel and data channel take different paths to reach the peer IoT node. The control channel uses the voice traffic and Public Switched Telephone Network (PSTN) to reach the peer IoT node. Here, both the IoT nodes are connected using the International Mobile Subscriber Identity (IMSI) in PSTN, where the traditional DoS attack from botnets is not possible. However, the control channel has minimal bandwidth and could be used only for control parameters of data channel, such as changed IP addresses, the port for listening, duration of port open, etc. In contrast, the data channel is used for sensor data traffic that consumes bandwidth. During communications, both the control and data channel function simultaneously in an IoT node. Based on the parameters received in the control channel, the data channel is dynamically configured at run-time.

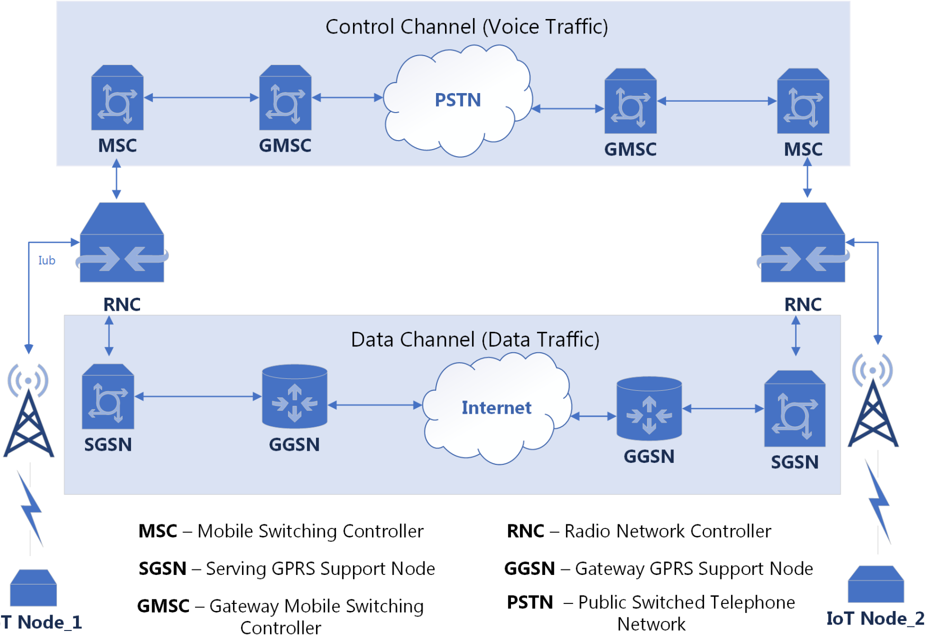

Control Channel (Voice Traffic)
MSC
GMSC
PSTN
GMSC
MSC
Iub
RNC
RNC
Data Channel (Data Traffic)
SGSN
GGSN
Internet
GGSN
SGSN
IoT Node_1
IoT Node_2
MSC – Mobile Switching Controller
SGSN – Serving GPRS Support Node
GMSC – Gateway Mobile Switching Controller
RNC – Radio Network Controller
GGSN – Gateway GPRS Support Node
PSTN – Public Switched Telephone Network

Figure 5. Communication architecture.

The parameters are encoded in voice traffic using Dual Tone Multi-frequency (DTMF) tones [32]. DTMF is a method used for converting the bits to tones. The tones are sent in the form of voice signals between the peer IoT nodes using voice frequency bands. The DTMF table, along with different frequencies and corresponding hexadecimal bits, is shown in Table 3, and a detailed protocol handshake process between the sender and receiver is explained in Section 3.4.

The data channel uses the usual IP (Internet protocol) channel to communicate between the nodes. By default, none of the nodes listen for any incoming messages in the data channel. The data communication takes place after the configuration parameters are received through the control channel. This prevents unnecessary listening in the data channel and saves a significant amount of power in the node.

Table 3. DTMF Frequency table.

	1209 Hz	**1336 Hz**	**1477 Hz**	**1633 Hz**
697 Hz	1	2	3	A
770 Hz	4	5	6	B
852 Hz	7	8	9	C
941 Hz	*	0	#	D

3.4. Communication Phases in VRITHI

As aforementioned, VRITHI uses both the control and data channel simultaneously for real-time collaboration between IoT nodes. Every communication sequence between two IoT nodes goes through four phases: (1) Node Initialization, (2) Control Channel Initialization, (3) Node Collaboration, and (4) Connection Closure. Each of these phases are briefly discussed below:

- *Node Initialization phase:* in this phase, the node is powered up, and the modem of the node is connected to the cellular network. At this point, the node is assigned a signal bearer by the cellular provider. A bearer is a tunnel that connects a specific service from the Telecom network to the user equipment (UE) viz modem. Different services, such as voice, Internet data, video call, etc., have different bearers from the Telecom network to the UE. Signal bearers are used by the Telecom network to signal the UE for services, such as UE registration, incoming call signaling, etc. Voice bearers are used to carry voice traffic between the UEs, and data bearers are used to connect the UE to the Internet. The voice bearer is given the highest priority and guaranteed bit rate compared to other bearers by the Telecom network. As landslides are more susceptible during rainy conditions, such a priority for voice traffic is more helpful for the landslide early warning system.
- *Control Channel Initialization phase:* The nodes enter this phase if they require to start collaborative real-time communication with other IoT nodes. In this phase, the UE uses the control channel to connect to other node, and a dedicated voice bearer VB_0 is created between the nodes by the Telecom network. The entire communication sequence between two IoT nodes is illustrated in Figure 6. The voice bearer is shaded in brown color, the data bearer in blue color, and the signal bearer in black color. In the figure, the control channel Initialization is shown as the message sequence (2).
- *Node Collaboration phase:* In this phase, an IoT node uses the control channel to exchange the control parameters. Protocol message sequence (3) to (6), in Figure 6, are exchanged during this phase. This protocol message exchange is iterative until the entire sensor data is communicated between the nodes. The block marked as loop in Figure 6 takes care of changing the IP addresses and ports at run-time. In brief, the message sequence (3) is used by the modem to signal its interest in establishing the IP data connection. In sequence (4), the IoT nodes mutually exchange their source IPs (assigned by cellular provider), a nounce, and CRC through voice bearer VB_0. The nounce is used for a simple XoR mechanism to differentiate between each exchanged message, and CRC is used to detect errors during the conversion of payload into voice codec. In sequence (5), additional parameters, such as acknowledgment or no acknowledgment, TCP or UDP protocol, and the number of ports to open, are exchanged. After successful exchanges of these messages, both the nodes agree to start mutual communication of IP data. It is only at this stage that one of the nodes starts to listen on the data port, and the IP message sequence (6) is sent to that port by creating data bearer DB_0.
- In the *Connection Closure* phase, one of the peer IoT nodes participating in the real-time communication indicates the end of conversation in the control channel. Hence, the control channel and data channel are closed at both the ends. The voice and data bearers,VB_0 and DB_0 allocated by the Telecom network are released.

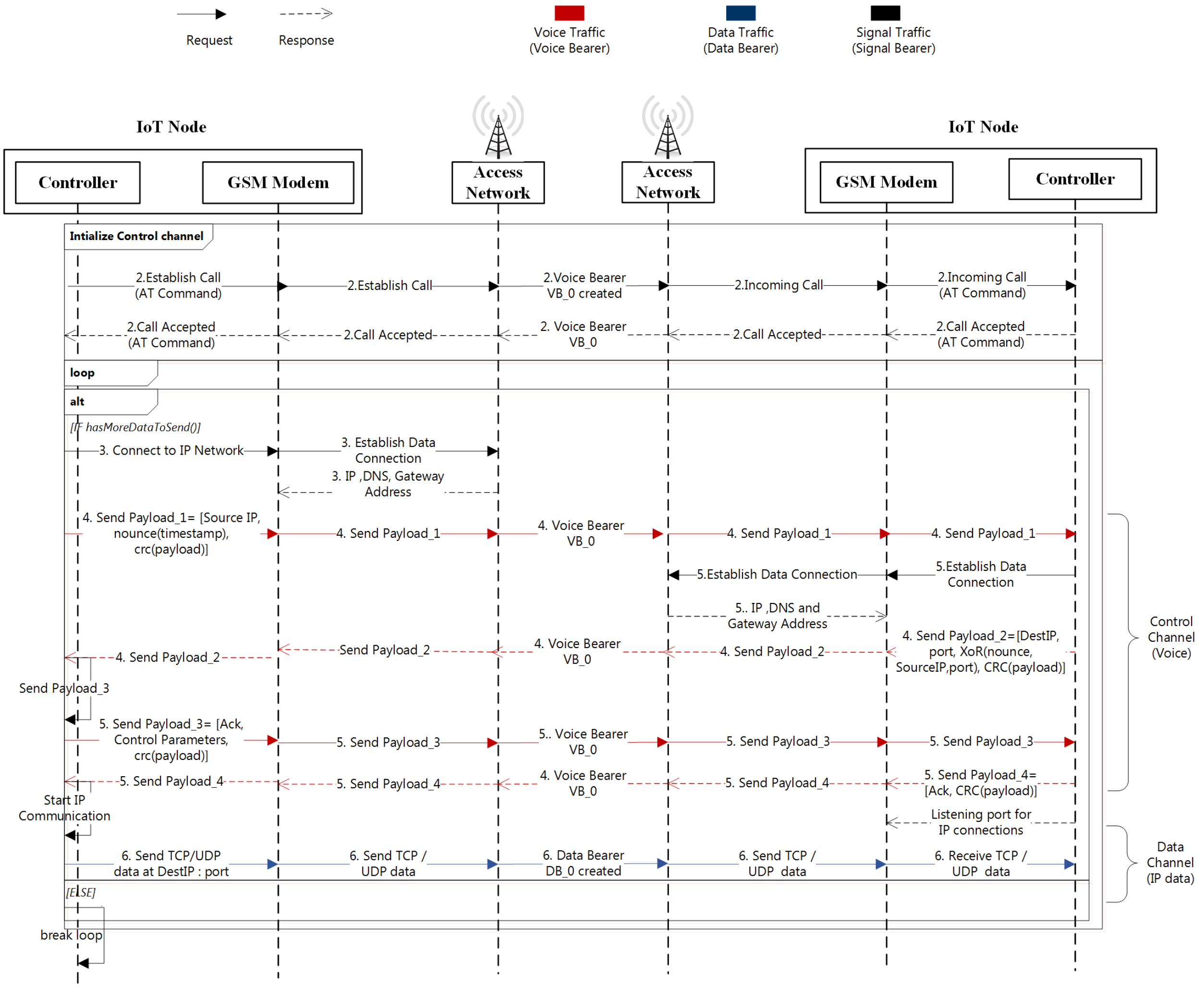

Request
Response
Voice Traffic (Voice Bearer)
Data Traffic (Data Bearer)
Signal Traffic (Signal Bearer)
IoT Node
Controller
GSM Modem
Access Network
Access Network
IoT Node
GSM Modem
Controller
Intialize Control channel
2.Establish Call (AT Command)
2.Establish Call
2.Voice Bearer VB_0 created
2.Incoming Call
2.Incoming Call (AT Command)
2.Call Accepted (AT Command)
2.Call Accepted
2. Voice Bearer VB_0
2.Call Accepted
2.Call Accepted (AT Command)
loop
alt
[IF hasMoreDataToSend()]
3. Connect to IP Network
3. Establish Data Connection
3. IP ,DNS, Gateway Address
4. Send Payload_1= [Source IP, nounce(timestamp), crc(payload)]
4. Send Payload_1
4. Voice Bearer VB_0
4. Send Payload_1
4. Send Payload_1
5.Establish Data Connection
5.Establish Data Connection
5.. IP ,DNS and Gateway Address
4. Send Payload_2
Send Payload_2
4. Voice Bearer VB_0
4. Send Payload_2
4. Send Payload_2=[DestIP, port, XoR(nounce, SourceIP,port), CRC(payload)]
Send Payload_3
5. Send Payload_3= [Ack, Control Parameters, crc(payload)]
5. Send Payload_3
5.. Voice Bearer VB_0
5. Send Payload_3
5. Send Payload_3
5. Send Payload_4
5. Send Payload_4
4. Voice Bearer VB_0
5. Send Payload_4
5. Send Payload_4= [Ack, CRC(payload)]
Start IP Communication
Listening port for IP connections
6. Send TCP/UDP data at DestIP : port
6. Send TCP / UDP data
6. Data Bearer DB_0 created
6. Send TCP / UDP data
6. Receive TCP / UDP data
[ELSE]
break loop
Control Channel (Voice)
Data Channel (IP data)

Figure 6. Communication handshake for peer-to-peer communication.

3.5. Advantages of Control Channel

Using the control channel offers many benefits, as listed below:

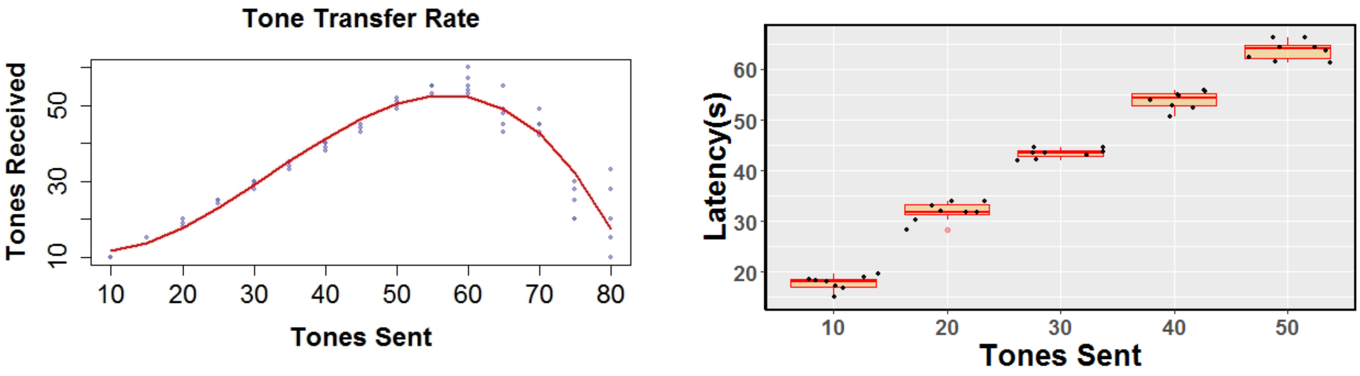
- Unlike the data channel, when the communication is happening between the control channel of the sender and receiver nodes, the third node receives a busy tone and cannot initiate another communication or flood the receiver, causing a DoS attack. In contrast, during communications through the data channel, any node on the Internet can flood the receiver with IP packets that results in DoS attack.
- In the control channel, the communication happens using IMSI number, which can drastically reduce the traditional botnet attacks.
- If the botnet DoS attacks compromise the modem, the TCP/IP stack becomes non-responsive, but the GSM stack that has established voice bearer remains active. Hence, voice connections continue to operate.
- Listening in the data channel is performed only when it is required for communication (or signaled by the control channel). This helps in saving power, as shown in Section 4.5.
- The IP assigned to the IoT node by the Telecom provider keeps changing over time. The control channel takes care of such dynamic-IPs assigned to the node. This saves memory space by not having a separate Dyn-DNS client running in the main memory of the node.

- During disasters or emergency conditions, voice traffic has a higher priority than Internet data traffic. Hence, having a control channel in voice traffic is advantageous for signaling in such scenarios.

3.6. Communication Characteristics of Control Channel

Before concluding this section, we would like to derive specific characteristics of the control channel in terms of the maximum DTMF tone transfer rate in the voice channel and latency for sending the tone. Figure 7a shows the transfer rate of DTMF tones on encoded data in the channel. The plot reflects that average of four different test cases for sent versus received tones. A cubic polynomial-fit is found suitable in this plot, with p value less than 2.2×10^{-16}, and with R^2 value as 0.8927.

Initially, the tones are tested for unidirectional communication. It is observed that, after the rate of 58 tones transmitted unidirectionally, there is a decline of tone rate due to the mixture of DTMF tones at the receiver side of the modem. The resultant packets were corrupted and, hence, could not be used for communications. However, the channel can be reset by sending an ACK bit back to the sender. The test is conducted without a Cyclic Redundancy Check (CRC) mechanism to know the maximum tone rate that we could achieve with the modem.



(**a**) Data rate in Control Channel (**b**) Latency in Control Channel

Figure 7. Characteristics of tone transfer. (**a**) Data rate in Control Channel; (**b**) Latency in Control Channel.

The latency of the tones is measured from sender to receiver. The experiment is repeated eight times, and the distribution is plotted in Figure 7b using a box plot for five values of the number of tones sent. It can be observed that, to send 30 tones, it incurs an average latency of 46.5 s.

4. Evaluation of VRITHI

In this section, we evaluate the effectiveness of VRITHI in achieving increased protection, efficient real-time communications, and reduced energy consumption of the nodes. Then, we also compare its performance with a number of popular IoT protocols.

4.1. Test Setup

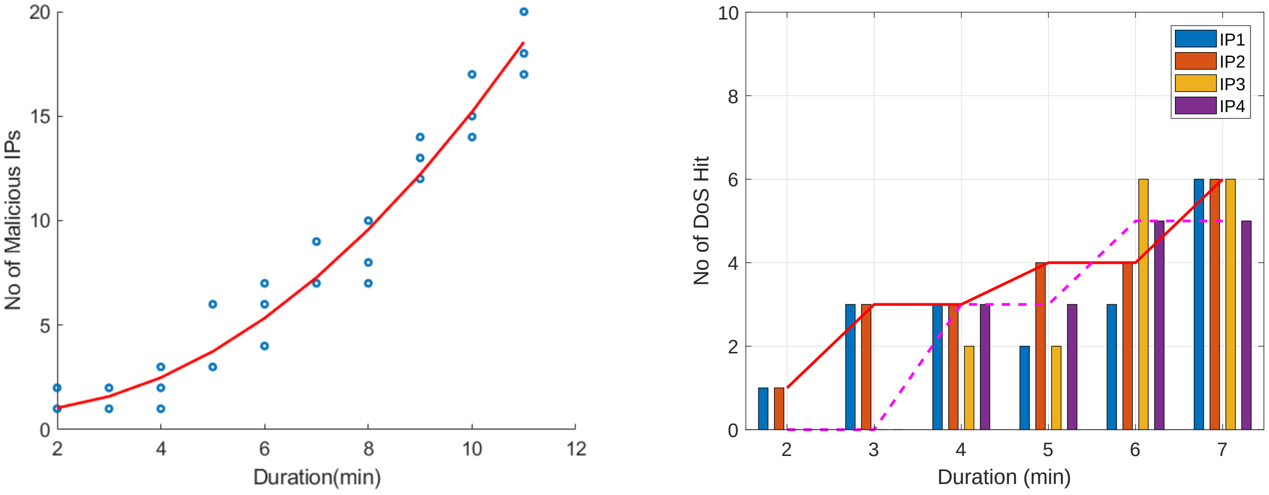
The nodes are connected directly to the cellular network in the field. They periodically send sensor data to the data management center (DMC). The nodes have ultra-low resources, and, when a DoS attack compromises a node, it stops transmitting data to DMC. It also stops responding to incoming messages. This discontinuity in periodic transmission of data to DMC could be due to various reasons, such as (i) lack of node power, (ii) lack of cellular signal strength, (iii) absence of data communications while only the voice connection is available, or (iv) due to DoS (denial of service) attack. We log the nodes health (power), GSM signal strength, the IP address of the peer node establishing a connection with the current node, and the timestamp of the incoming IP address. These are stored in non-volatile EEPROM memory on the node. This metadata is also sent to DMC, along with sensor data.

If there is no incoming data in DMC from a node, we first analyze the above metadata sent by the node. When a DoS attack happens, we observe that reasons (i) and (ii) are not the primary cause, but nodes still fail abruptly to send data to DMC. Hence, we start to analyze the logged IP addresses in the EEPROM. We find that the nodes have been hit by IP addresses with different octets in Most Significant Bit (MSB) than the IP addresses of the subscribed local cellular network domain. Hence, we concluded that these IP addresses are not in the subscribed local cellular network domain. By tracking these IP addresses, we derive the intensity of DoS attacks. The analysis is done using three IoT devices connected to public network as these nodes engage in mutual communication. The deployment is done using ten such devices deployed in the field at strategically selected locations.

4.2. Intensity of DoS Attacks

We retrieve from log details the time of attacks and different IP addresses from which attacks happened. We then plot the number of malicious IPs (the addresses that do not belong to our network) over a duration of 10 min, as illustrated in Figure 8a. The blue dots in the scatter plot represent the number of malicious IP addresses with three different test cases. The trend line (in red) reflects the rate at which the malicious IPs are targeting the node.

The *hit rate* defines the frequency of the same IP address hitting the sockets in the node. Some IP addresses hit the node at the same rate, whereas other IP addresses increase the rate when it finds that the socket is open. We took four sample IP addresses and plotted the hit rate graph. The hit rate graph in Figure 8b illustrates the number of times these IP addresses hit the nodes. The x-axis represents the duration of observations, and the y-axis represents the number of times a malicious IP hits the node in a minute. We observe that some of the IPs have the same hit rate, whereas, for others with open sockets, the hit rate increases. This may be due to random botnets hitting the node or the botnets acting as malicious nodes, again strategically making the DDoS attack with other botnets. In rare scenarios, the hit rate decreases, but we speculate that it may be due to the loss of packets in the WAN connection from the malicious nodes.



(**a**) Ratio of malicious IP

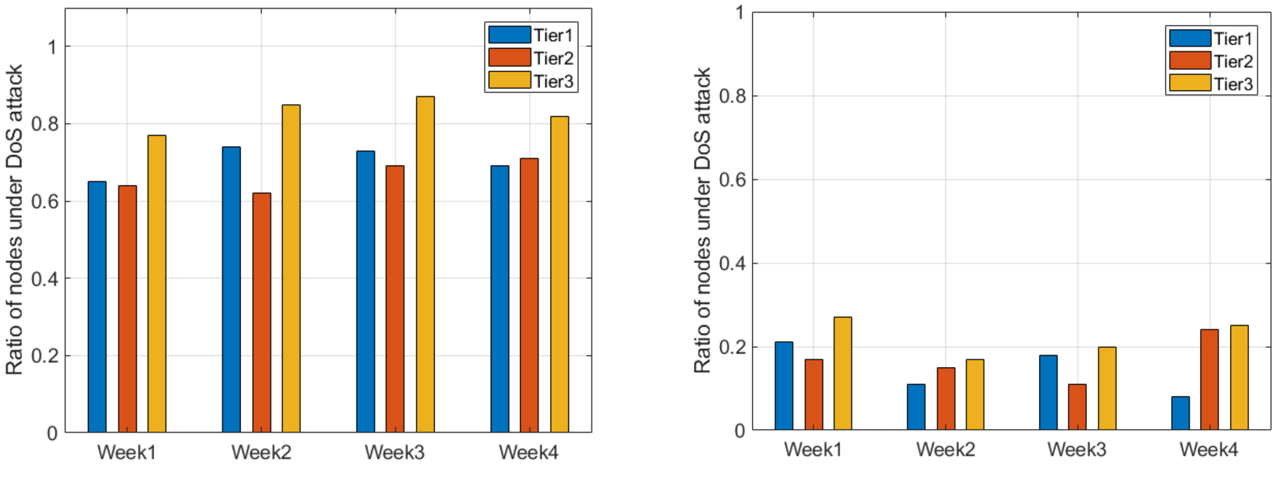
(**b**) DoS Hit Rate

Figure 8. Intensity of DoS attacks before VRITHI. (**a**) Ratio of malicious IP; (**b**) DoS Hit Rate.

4.3. Effectiveness in Reducing DoS Attacks

We mention earlier that nodes under DoS attacks in the network obtain IP addresses that are not in the subscribed local cellular networks domain. The ratio of the total number of IPs the nodes received (from both inside and outside the network), and the total number of IPs received from outside the network, is an important metric to track the proportion of DoS attacks affecting the nodes. The nodes are observed for malicious IPs over one month, and this ratio taken for each week is plotted in Figure 9a. The plot illustrates that, before implementing VRITHI, more than 50% of the nodes are under DoS attack in many situations, and the nodes become not useful for valid operations.

Effectiveness of VRITHI is readily reflected in the reduction of DoS attacks, as illustrated in Figure 9b. The modems used to connect to the cellular network have a default setting in which the control channel is active while the data transfer is going on. During the process of data transfer, socket connections are changed at run-time. The IP packets have the sequence numbers tagged to them such that they can be put in correct sequence at the destination. This helps in achieving a lower DoS attack ratio. From Figure 9b, it can be noted that DoS attack after applying VRITHI is reduced from 65–82% to less than 28% and is achieved using hardware with as little as 2 KB of RAM.



(**a**) Before VRITHI (**b**) After VRITHI

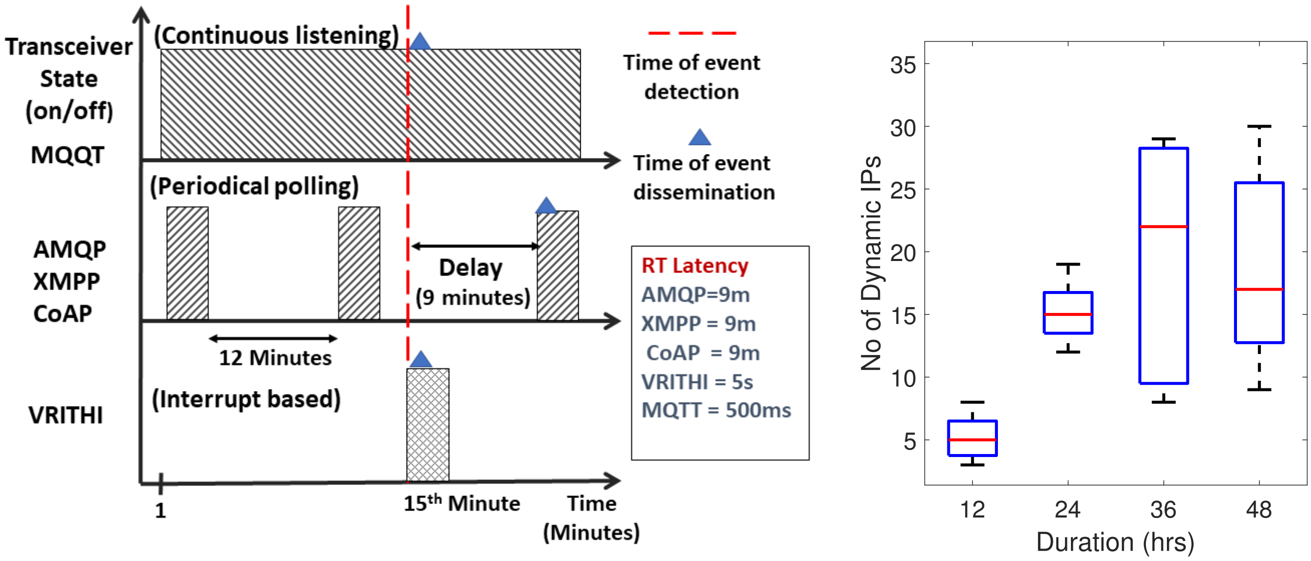
Figure 9. VRITHI leads to reduction of ratio of nodes under DoS attacks: (**a**) Before VRITHI, (**b**) After VRITHI.

4.4. Improved Real-Time Communications

VRITHI provides better real-time communication with less power consumption and less message delay. This section compares VRITHI with other prominent IoT protocols, such as MQTT, AMQP, CoAP, and XMPP.

The MQTT protocol is based on publish-subscribe mechanism and uses cloud brokers for intercommunication among nodes. In MQTT, the subscriber continuously listens to the publisher. Whenever an event in the sensor is detected, it is disseminated immediately to the subscribers, as shown in Figure 10a. However, listening continuously (or periodically) induces a lot of power consumption. In the figure, the grey area in MQTT indicates whether transceiver state is ON or OFF.

AMQP, XMPP, and CoAP are other prominent protocols that can act in either publish-subscribe or request-reply mechanism. Primarily, they use a Request-Reply mechanism. AMQP is mostly used in a business messaging scenario where mobile handsets communicate with the back-office data centers. It can work in both TCP or UDP connections but requires higher resource utilization compared to other protocols. It needs a third party broker, such as Rabbit Messaging Queuing (RabbitMQ), for inter-node communications. XMPP uses XML-based messaging predominantly over TCP connection. However, they additionally need a jabber-id and broker server, such as AMQP, for communications. CoAP uses less energy as compared to these protocols. However, it also needs a polling mechanism and third-party broker (CoAP Mongoose server) for functioning. In Figure 10a, the grey areas in AMQP, XMPP, CoAP indicate the period of polling for every 12 min. We can observe that, while an event is happening at the 15th minute, it is disseminated only after a delay of 9 min during the next polling. Moreover, periodical polling results in higher power consumption.



(**a**) Latency in Real-time Communications (**b**) Dynamic IP in WAN

Figure 10. Real-time communication characteristics using VRITHI. (**a**) Latency in Real-time Communications; (**b**) Dynamic IP in WAN.

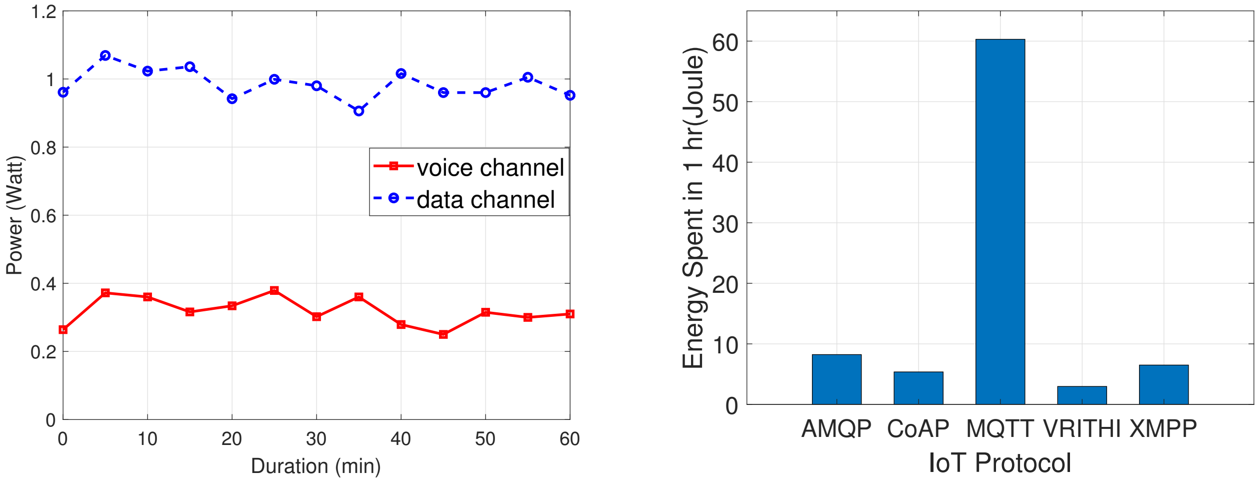
None of the above protocols provide direct peer-to-peer connection without the help of a brokerage service. This disadvantage of message delay is overcome by the VRITHI framework, along with reduced energy consumption. VRITHI follows an interrupt-based mechanism. In this, a node is in standby mode, and the data channel is disabled. Whenever there is an event in the sensor data, the control channel initiates an handshake with the appropriate node through their corresponding International Mobile Subscriber Number (IMSI). The data channels are then enabled. For IP communication to happen, the control channel is used to exchange their corresponding IP addresses and ports, as described in Section 4. In Figure 10a, the grey area in VRITHI indicates the dissemination of interrupt as soon as the event is detected.

During the course of communications, the assigned IP address may get changed by the cellular provider. In that case, the control channel takes care of updating it in the other node. Unlike other protocols, there is no need for a DNS or third party client in nodes memory which is helpful with nodes having only 2 KB RAM. We observe a specific node for the number of dynamic IPs assigned over 48 h during the peer-to-peer communication using sim5300E modem, and Figure 10b reflects the number of times IP address changes in four 12-h intervals.

4.5. Reduced Energy Consumption

The nodes in VRITHI are in stand-by mode with only voice network enabled. Before a peer-to-peer connection, the control channel handshake is initiated with the receiver through voice channel. It saves a considerable amount of power. Figure 11a shows the power consumption by sim5300E modem for the data channel and control channel averaged on a hourly basis for a node in a single tier. It is measured using Agilent Source measuring unit B2902A. Since all the three tiers use the same hardware and same carrier networks, their power consumption on voice channel and data channel are the same for a given payload. It can be observed that the control channel has less than half the power of the data channel. This is a significant power saving in Green IoTs. Secondly, whenever a disaster happens, the voice channel has the highest guaranteed bit rate in the mobile switching center of the Telecom network. It happens due to higher congestion in the incoming calls and the need to provide uninterrupted connectivity at crisis time. Hence, having the control channel for signaling helps significantly.

Figure 11b compares the energy consumption of VRITHI with that of other IoT protocols. In this experiment, observations are taken over one hour, and the y-axis shows the total energy consumption during that interval. MQTT is in always-listening mode; hence, it has the maximum power consumption. For polling-based mechanisms, the polling period is taken the same as in Figure 10a, and, at every polling cycle, we measure the power consumption and then obtain the sum up over 1 h. VRITHI, being interrupt-based, incurs power consumption only while doing the data communications, and, during our experiment, it happens only once. Hence, VRITHI has the lowest energy consumption, and energy saving increases from 41% to 97.2%.



(**a**) Power of Data and Voice Channels

(**b**) Comparison with IoT Protocols

Figure 11. Energy consumptions using VRITHI. (**a**) Power of Data and Voice Channels, (**b**) Comparison with IoT Protocols.

5. Conclusions

We propose a novel method called VRITHI for the Green Industrial Internet of Things with ultra-low resource hardware (as less than 2 KB RAM) where nodes are directly exposed to Internet without any intermediate nodes, such as gateway. VRITHI employs dual-channel communication mechanism where a voice channel is used for control signals followed by a data channel for the actual communications. This paper discusses the design of VRITHI in terms of detail communication sequence using voice channel and data channel. Being an interrupt-based technique, VRITHI achieves increased energy efficiency and improved real-time communications with respect to other prominent IoT protocols in the Industry. A thorough analysis of the state of the art in IoT reflects that the domain lacks such a solution for ultra-low memory hardware. Evaluation of VRITHI demonstrates that it is able to provide increased protection against DoS attacks, efficient real-time communications, and reduced energy consumption. The proposed solution is applicable to any IoT scenarios with tiny ultra-low memory nodes, such as smart rings, smart watches, etc. In the future, we would like to provide information coding techniques on the tones in control channels used in VRITHI. This will enable sending additional control information among the nodes and help in further energy saving. Secondly, in the current scenario, we assume that the core network is a trustful carrier network. In the future, a complete protocol and key exchange handshake between the nodes could be implemented. This helps to protect the data from spoofing by a core network and provide an end-to-end protection between the nodes.

6. Patents

Sangeeth kumar, Venkat Rangan, Maneesha Vinodini Ramesh, "Method of reducing the DoS attacks using Voice Response in IoT Systems", U.S. Patent 16/924,468 filed 9 July 2020.

Author Contributions: Conceptualization, S.K.; methodology, S.K.; validation, S.K.; data curation, S.K.; writing-original draft preparation, S.K.; writing-review and editing, S.D.; supervision, V.P.R. All authors have read and agreed to the published version of the manuscript.

Funding: This work is funded by Visvesvaraya Ph.D scheme of Government of India.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Acknowledgments: The authors would like to express gratitude for the immense amount of motivation and research solutions provided by Sri. Mata Amritanandamayi Devi (AMMA), The Chancellor, Amrita University. The authors would also like to thank the UN recognized NGO, Mata Amritanandamayi Math, for the help, continuous support, and co-funding of the landslide project, which is a societal beneficial system.

Conflicts of Interest: The authors have no conflict of interest to declare that are relevant to the content of this article.

References

1. Abas, K.; Obraczka, K.; Miller, L. Solar-powered, wireless smart camera network: An IoT solution for outdoor video monitoring. *Elsevier Comput. Commun.* **2018**, *118*, 217–233. [CrossRef]
2. Tanyingyong, V.; Olsson, R.; Hidell, M.; Sjödin, P.; Ahlgren, B. Implementation and Deployment of an Outdoor IoT-Based Air Quality Monitoring Testbed. In Proceedings of the 2018 IEEE Global Communications Conference (GLOBECOM), Abu Dhabi, United Arab Emirates, 9–13 December 2018.
3. Olasupo, T.O. Wireless Communication Modeling for the Deployment of Tiny IoT Devices in Rocky and Mountainous Environments. *IEEE Sens. Lett.* **2019**, *3*, 1–4. [CrossRef]
4. Iova, O.; Murph, A.L.; Picco, G.P.; Ghir, L.; Molten, D.; Ossi, F.; Cagnacci, F. LoRa from the City to the Mountains: Exploration of Hardware and Environmental Factors. In Proceedings of the 2017 International Conference on Embedded Wireless Systems and Networks, Uppsala, Sweden, 20–22 February 2017.
5. From Wikipedia, the Free Encyclopedia. Available online: https://en.wikipedia.org/wiki/Fresnel_zone (accessed on 2 September 2018).
6. Panagiotis, K.; Panagiotis, T.; Zahariadis, T.; Hatziefremidis, A.; Helen, L. RPL modeling in J-Sim platform. In Proceedings of the Ninth International Conference on Networked Sensing, Antwerp, Belgium, 11–14 June 2012.
7. Ren, Y.; Zhang, X.; Lu, G. The Wireless Solution to Realize Green IoT: Cellular Networks with Energy Efficient and Energy Harvesting Schemes. *Energies* **2020**, *13*, 5875. [CrossRef]
8. Sharma, S.; Wang, X. Toward Massive Machine Type Communications in Ultra-Dense Cellular IoT Networks: Current Issues and Machine Learning-Assisted Solutions. *IEEE Commun. Surv. Tutor.* **2020**, *22*, 426–471. [CrossRef]
9. Kumar, S.; Duttagupta, S.; VenkatRangan, P.; Ramesh, M.V. Reliable network connectivity in wireless sensor networks for remote monitoring of landslides. *WIreless Netw.* **2019**, *26*, 2137–2152. [CrossRef]
10. Neshenko, N.; Bou-Harb, E.; Crichigno, J.; Kaddoum, G.; Ghani, N. Demystifying IoT security: An exhaustive survey on IoT vulnerabilities and a First empirical look on Internet-Scale IoT exploitations. *IEEE Commun. Tutor.* **2019**, *21*, 2702–2733. [CrossRef]
11. Lu, Y.; Xu, L.D. Internet of Things (IoT) Cybersecurity Research: A Review of Current Research Topics. *IEEE Internet Things* **2019**, *6*, 2103–2115. [CrossRef]
12. Khan, M.N.; Rao, A.; Camtepe, S. Lightweight Cryptographic Protocols for IoT-Constrained Devices: A Survey. *IEEE J. Internet Things* **2021**, *8*, 4132–4156. [CrossRef]
13. Hussain, B.; Du, Q.; Sun, B.; Han, Z. Deep Learning-Based DDoS-Attack Detection for Cyber–Physical System Over 5G Network. *IEEE Trans. Ind. Inform.* **2021**, *17*, 860–870. [CrossRef]
14. Dizdarevic, J.; Carpio, F.; Jukan, A.; Masip-Brui, X. A survey of communication protocols for Internet of Things and related challenges of Fog and Cloud Computing integration. *ACM Comput. Surv.* **2019**, *51*, 1–29. [CrossRef]
15. Cetinkaya, A.; Ishii, H.; Hayakawa, T. An overview on denial-of-service attacks in control systems: Attack models and security analyses. *Entropy* **2017**, *21*, 210. [CrossRef] [PubMed]
16. Salim, M.M.; Rathore, S.; Park, J.H. Distributed denial of service attacks and its defenses in IoT: A survey. *J. Supercomput.* **2020**, *11*, 5320–5363. [CrossRef]

17. Meneghello, F.; Calore, M.; Zucchetto, D.; Polese, M.; Zanella, A. IoT: Internet of Threats? A Survey of Practical Security Vulnerabilities in Real IoT Devices. *IEEE Internet Things* **2019**, *6*, 8182–8201. [CrossRef]
18. Karthik, A.; Kumar, S.; Rao, S.N. Security considerations for a real time Landslide Monitoring System. In Proceedings of the IEEE International Conference on Computational Intelligence and Computing Research, Coimbatore, India, 14–16 December 2017.
19. Li, F.; Shinde, A.; Shi, Y.; Ye, J.; Li, X.Y.; Song, W. System statistics learning-based IoT security: Feasibility and suitability. *IEEE Internet Things* **2019**, *6*, 6396–6403. [CrossRef]
20. Adat, V.; Gupta, B.B.; Yamaguchi, S. Risk transfer mechanism to defend DDoS attacks in IoT scenario. In Proceedings of the IEEE International Symposium on Consumer Electronics, Kuala Lumpur, Malaysia, 14–15 November 2017.
21. Mouratidis, H.; Diamantopoulou, V. A security analysis method for industrial Internet of Things. *IEEE Trans. Ind. Inform.* **2018**, *14*, 4093–4100. [CrossRef]
22. Kajwadkar, S.; Jain, V.K. A novel algorithm for DoS and DDoS attack detection in Internet of Things. In Proceedings of the Conference on Information and Communication Technology, Yogyakarta, Indonesia, 6–7 March 2018.
23. Stergiou, C.; Psannis, K.E.; Gupta, B.B.; Ishibashi, Y. Security, privacy and efficiency of sustainable Cloud Computing for Big Data and IoT. *Sustain. Comput. Inform. Syst.* **2018**, *19*, 174–184. [CrossRef]
24. Szymanski, T.H. Security and privacy for a green Internet of Things. *IT Prof. Mag.* **2017**, *19*, 34–41. [CrossRef]
25. Tan, S.Y.; Yeow, K.W.; Hwang, S.O. Enhancement of a lightweight attribute-based encryption scheme for the Internet of Things. *IEEE Internet Things* **2019**, *6*, 6384–6395. [CrossRef]
26. Sathyadevan, S.; Achuthan, K.; Doss, R.; Pan, L. Protean Authentication Scheme – A Time-Bound Dynamic KeyGen Authentication Technique for IoT Edge Nodes in Outdoor Deployments. *IEEE Access* **2019**, *7*, 92419–92435. [CrossRef]
27. Fang, Y.; Zhu, X.; Zhang, Y. Securing resource-constrained wireless Ad hoc Networks. *IEEE Wirel. Commun.* **2009**, *16*, 24–130. [CrossRef]
28. Selis, V.; Marshall, A. A classification-based algorithm to detect forged embedded machines in IoT environments. *IEEE Syst.* **2019**, *13*, 389–399. [CrossRef]
29. Hijazi, S.; Obaidat, M.S. A new detection and prevention system for ARP attacks using static entry. *IEEE Syst.* **2019**, *13*, 2732–2738. [CrossRef]
30. Li, C.Y.; Tu, G.H.; Peng, C.; Yuan, Z.; Li, Y.; Lu, S.; Wang, X. Insecurity of voice solution VoLTE in LTE mobile networks. In Proceedings of the 22nd ACM Conference on Computer and Communications Security, Denver, CO, USA, 12–16 October 2015; pp. 316–327.
31. Jeon, H.; Eun, Y. A stealthy sensor attack for uncertain cyber-physical systems. *IEEE Internet Things* **2019**, *6*, 6345–6352. [CrossRef]
32. Neetha, K.K.; Koya, A.M. A compressive sensing approach to DCT watermarking system. In Proceedings of the 2015 International Conference on Control Communication & Computing India (ICCC), Trivandrum, India, 19–21 November 2015.