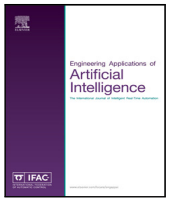
Engineering Applications of Artificial Intelligence 126 (2023) 106745

Contents lists available at ScienceDirect

Engineering Applications of Artificial Intelligence

journal homepage: www.elsevier.com/locate/engappai



HANNA: Human-friendly provisioning and configuration of smart devices

Carolina Fortuna [a,*], Halil Yetgin [b], Leo Ogrizek [a], Esteban Municio [c,d], Johann M. Marquez-Barja [c], Mihael Mohorcic [a]

[a] Jozef Stefan Institute, Jamova 39, Ljubljana, 1000, Slovenia
[b] Department of EEE, Bitlis Eren University, Bes Minare Mah, Bitlis, 13100, Turkey
[c] IDLab, University of Antwerp - IMEC, Sint-Pietersvliet 7, Antwerp, 2000, Belgium
[d] i2CAT - The Internet Research Center, C Gran Capità 2-4 Edifici Nexus, Barcelona, 08034, Spain

ARTICLE INFO

Dataset link: https://zenodo.org/record/7933500#.ZGAJYC8RqvU

Keywords:
Human-friendly
Provisioning
Initial configuration
Voice assisted
Wireless
IoT
Speech recognition
Vocabulary

ABSTRACT

Today, there are billions of connected IoT devices and their number continues to grow as they contribute to the digitalization of infrastructures. However, the deployment process of these smart wireless devices when delivered to customer premises is slow and error prone as each of them needs to be provisioned with authentication credentials to access the corporate network. In this paper, we propose HANNA, a human-friendly provisioning and configuration framework for smart devices, that extends the zero-touch paradigm to large IoT deployments by introducing voice assisted configuration in combination with large scale ad-hoc communications to overcome the initial installation effort of IoT deployments. The most prominent role in HANNA is played by the assisting device, which includes a voice assistant capable of correctly understanding a minimum number of keywords required for initial provisioning and configuration of the devices. The device's role is to interact with the user and ensure that all provisioning details are received. These are then converted into appropriate machine instructions for further use by the mass provisioning mechanism. We provide an example prototype implementation of HANNA and evaluate the performance of the assisting device in the human-to-machine communication phase and the performance of the selected communication technique in the machine-to-machine communication phase. Our results show the potential of existing speech-to-text engines for this application area and also reveal shortcomings with respect to the robustness of the engines in office-like working environments as well as with respect to user's gender and language proficiency level. Additionally we show that the proposed machine-to-machine provisioning approach is always faster compared to manual provisioning for cases with more than ten devices.

1. Introduction

The Internet of Things (IoT) has been a recurrent theme since the term was coined in the late 1990s. The concept has evolved from early work on Radio Frequency Identifier (RFID) technology which represented a hardware-related breakthrough that aimed to connect everyday objects to a network. This constituted the first wave of the IoT, which then developed beyond the initial hardware world innovation, and focused increasingly on developing new types of sensors and sensing materials, as well as on developing new communication technologies and protocols. As a result, a wide variety of new communication technologies emerged in the early years of the 21st century which were able to support the ubiquitous deployment of a wide variety of sensors. We refer to this as the second wave of IoT. In the last decade, the focus of IoT has shifted to data collection, processing, analytics and security aspects and this period is termed the third wave of IoT which we are witnessing today.

The large number of IoT devices being deployed on a daily basis makes device management a prominent issue for IoT platform providers (Davies and Fortuna, 2020). For example, according to a recent whitepaper from a device manufacturer, setting up ten thousand smart light bulbs in a factory can take nearly 2 years before they can actually commence data streaming (Wilhelm et al., 2017) with the provisioning process taking up to 45 min to complete per device when using traditional industry practices that currently provision each device manually (John Wilhelm, 2017). According to another manufacturer, the deployment represents 30% of the costs of a smart metering project (Pauzet, 2010). The main reason for such costs and overhead was that the traditional way of provisioning smart devices, i.e. connecting them to an access point, was a manually intense process where a universal serial bus (USB) or joint test action group (JTAG) cable was needed to connect the smart device to a computer. This physical connection was then followed by manual configuration using wireless credentials. Then, once connected to the local network, not necessarily

* Corresponding author.
E-mail address: carolina.fortuna@ijs.si (C. Fortuna).

https://doi.org/10.1016/j.engappai.2023.106745
Received 19 July 2022; Received in revised form 14 June 2023; Accepted 29 June 2023
Available online 17 July 2023

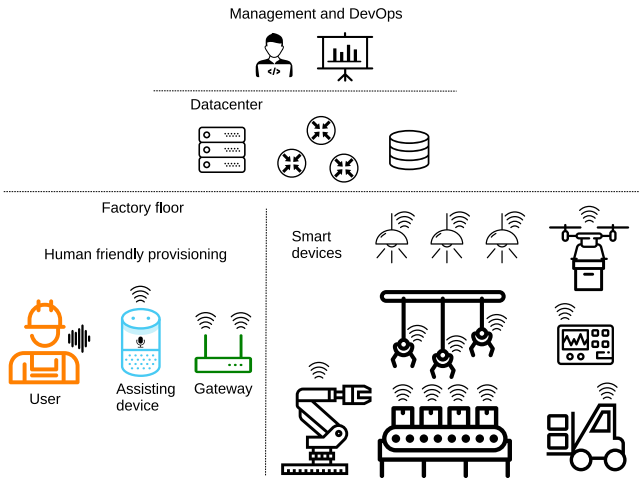


Fig. 1. Actors in the process of digitization of a factory infrastructure.

to the Internet, the devices are flashed and/or a configuration file with network credentials is transferred to them over a secure shell (SSH) connection. Finally, the access to the local network and/or to the Internet is tested (Boskov et al., 2020).

IoT platform providers increasingly wish to migrate from current device management processes, which are labour-intensive, error-prone (Berggren, 2018) and time-consuming activities (John Wilhelm, 2017), requiring individual provisioning of each device for a new solution or a new customer; and move towards automated provisioning and remote management. With the current state of the art, the physical connection is replaced by a wireless connection that the devices allow upon boot-up and the configuration files are automatically fetched. This process is referred to as zero-touch provisioning (ZTP) and initial studies have shown an up to 4 times faster provisioning time compared to the traditional ways (Boskov et al., 2020).

However, further automation could be achieved by removing the need to provision smart devices one by one. Assume a scenario such as depicted in Fig. 1 where *technicians* deploy and provision *smart devices* on the factory floor using an *assisting device* that intermediates the process of connecting the devices and machines to the inter/intranet through a *gateway* device. The assisting device can be seen as a domain-specific cobot (Weiss et al., 2021) that collaborates with humans forming a team. Instead of interacting with the assisting devices via a touch screen and a graphical user interface, which is a machine-centric interaction imposed on humans (Zhu et al., 2023), a more human-centric approach based on voice-based interaction can be used. Once deployed and configured, a regular physical and software maintenance regime commences through the network, data centre and DevOps thus smart devices get integrated into the industrial work process enabling more efficient and sustainable production processes (Sisinni et al., 2018).

It is well-known that provisioning and configuration of numerous devices, for example on a factory floor, can be a cumbersome and tedious process due to hazardous equipment/robots/cables and hard-to-reach areas (factory ceiling) (Boskov et al., 2020). Therefore, one critical need is to provision and configure these numerous and spatially separated devices in a quick and efficient manner without physical human interaction. This is a critical issue to be addressed because concurrent zero-touch mass provisioning and configuration reduces the need for technical expertise and helps manage and allocate human resources better due to significantly reduced deployment time. Since it is an automated and sequenced process that is carefully scripted with fail-safe procedures, human-induced configuration errors are prevented. Therefore, we aim at devising a zero-touch provisioning and configuration platform leveraging a voice assistant-equipped device, namely assisting device, with which a technician interacts to identify the relevant command for the provisioning and configuration tasks. Then, having identified the intended command, the assisting device expands the provisioning and configuration process towards other devices for mass-provisioning and -configuration through a communication protocol, namely 6TiSCH.

Broadly, we propose a new framework for human-friendly mass provisioning and configuration of smart devices which extends the state-of-the-art zero-touch provisioning (Boskov et al., 2020) paradigm and takes a new, fresh look at human-friendly large-scale provisioning and configuration through a human-centric voice-based interaction. We refer to this framework throughout the paper as Human-Friendly Provisioning and Configuration of Smart Devices (HANNA). To reiterate, the most prominent role in HANNA is taken by the *assisting device* that includes a voice assistant able to correctly understand a minimal set of keywords required for the initial provisioning and configuration of the devices. The device's role is to interact with the user and ensure that all provisioning details are received. These are then converted into appropriate machine instructions for further use by the mass provisioning mechanism. Through a reference implementation of HANNA, we evaluate the performance of voice configuration technologies and the scalability of the mass provisioning proof-of-concept.

The contributions of this paper are:

- HANNA, a human-friendly scalable mass-provisioning and -configuration framework that can speed up the deployment times and lower the skill level needed from deployment technicians. Our simulations show that for 100-node deployment, the proposed method can save up to 258 min on average in configuration time.
- A small domain-specific vocabulary that can be used in conversations according to a sequence diagram describing the configuration and provisioning process.
- We provide an example prototype implementation of HANNA and evaluate the performance of the assisting device in the human-to-machine communication phase and the performance of the selected communication technique within the machine-to-machine communication phase.
- We uncover the potential of existing speech-to-text engines for this application area and also reveal the shortcomings with respect to the robustness of the engines in office-like working environments as well as with respect to the user's gender and English language proficiency level. Additionally, we show that the proposed mass-provisioning approach always outperforms manual provisioning for cases with more than ten devices.

Such automation using an *assisting device* as depicted in Fig. 1 is becoming feasible especially due to the recent breakthroughs in AI, i.e. image and speech recognition, that facilitate a different level of human-to-machine interaction, best illustrated by the plethora of available voice assistants (Kim et al., 2019). This interaction will further continue to become ever more seamless (possibly also using gesture or brain signals) as the devices will get ever smaller yet able to support increasing processing complexity. Increased automation for smart devices involves recent advances in the areas of cloud computing and AI, in addition to expertise in industrial electronics, embedded devices, wireless communications and targeted application domain (Fortuna et al., 2022), thus the findings of this paper are relevant to a broad community.

The rest of the paper is structured as follows. Section 2 discusses the related work while Section 3 elaborates on the proposed HANNA framework. Section 4 presents the evaluation of the framework and Section 5 concludes the paper.

2. Related work

We group the related work into three subsections, the first concerned with voice assistants, the second with provisioning and initial configuration and the third with mass provisioning.

2.1. Voice assistants

Recently, voice recognition advanced to a level where speakers can be identified (Hansen and Hasan, 2015), their speech features can be utilized for biometric authentication and secured with privacy-preserving speech recognition techniques (Inthavisas and Lopresti, 2012), and voice assistants, such as Siri and Google are used daily. Additionally, specific vocabularies to be used for automatic network configuration purposes have also been investigated (Fortuna and Mohorcic, 2009). There are two distinct approaches to automatic speech recognition (ASR), language dependent and language independent (Watanabe et al., 2017). The first group is based on linguistic information and pronunciation dictionaries and the second is a language-independent design based on neural network architecture. The most influential online as well as offline ASR tools and APIs are reviewed and evaluated in Kim et al. (2019), Georgila et al. (2020) and Alibegović et al. (2020) revealing Google Speech-to-Text (STT) having a very good performance.

Interaction with computers based on voice is envisioned to take over increasingly demanding tasks such as computer programming (Caballar, 2021). There are many capable cloud-based ASR solutions such as the ones described in Kim et al. (2019). Besides online solutions, there are also many offline ones suitable for embedding into devices. Various online and offline solutions were previously evaluated in terms of the word error rate (WER) (Georgila et al., 2020). The results show that online ASR generally performed better than offline ones. However, for the special purpose of initial provisioning and configuration, an ASR solution trained specifically on a limited and more domain-specific vocabulary can perform reliably.

More recently, AI-based ASR engines are being applied in various areas such as for developing helper agents for IoT (Longo et al., 2021), enabling voice-driven configuration of software-defined networks (Chaudhari et al., 2019) and investigating the role of voice assistants with respect to productivity (Marikyan et al., 2022). More recently, the potential contribution of AI to increase efficiency and cut the cost of the initial deployment of smart devices has been recognized in Fortuna et al. (2022). A few studies related to the adoption of human-centric voice-based interactions have been recently proposed. The authors of Zhu et al. (2023) proposed a denoising technique in view of voice control for Industry 5.0 while in Chen et al. (2023) they study aspects of enabling voice-based interaction with consumer electronics. In Yan et al. (2021) they study the feasibility of injecting inaudible voice commands into voice assistants.

2.2. Provisioning and initial configuration

With the current industry practices, the *technician* uses a special or general purpose computer as *assisting device* and connects it via a universal serial bus (USB) or joint test action group (JTAG) cable to a computer (Boskov et al., 2020). This physical connection is then followed by the manual configuration of required wireless authentication credentials. Then, once connected to the local network, not necessarily to the Internet, the devices are flashed and/or a configuration file is transferred to them over a secure shell (SSH) connection. Finally, access to the local network and/or to the Internet is tested.

State-of-the-art methods also referred to as zero-touch provisioning (ZTP) are eliminating the physical configuration and part of the parameter configuration from the process. This is done by developing standards in which the *smart devices* boot up in a listening mode. A hand-held device provides an input option in the form of a touch screen or keyboard and serves as an *assisting device* that connects wirelessly to the listeners and provisions them, often using automation scripts with minimal required user input (Boskov et al., 2020). For WiFi-based networks, the software-enabled access point (soft-AP) has been proposed, where a device to be provisioned can be booted temporarily in soft-AP mode as a WiFi hotspot (Lee et al., 2019). Then, a user attempts to connect to this soft-AP hotspot using a mobile device either with the help of the underlying operating system or by directing the technician to manually connect. Then, the technician enters the required credentials (SSID, passkey) of his/her private gateway into a web form served from the temporary soft-AP via a browser or mobile application and this way allows access to the private WiFi network and the Internet. However, this provisioning method is time-consuming and can introduce human-made errors due to the manual input of credentials, especially when multiple gateways for large-scale deployment are considered (Lee et al., 2019).

Other provisioning methods proposed in the literature are based on near field communication (NFC), quick response (QR) code, ultrasound or are manufacturer specific (Boskov et al., 2020). However, both NFC- and QR-based provisioning methods are short-range solutions and cannot be leveraged for provisioning of large-scale deployment, since the interaction with each device is time-consuming. More recently, a proposal to improve the provisioning procedure in a mesh network has been put forth. Their implementation allows Bluetooth non-mesh devices to be provisioned and to take part in a Bluetooth mesh network, making it possible to continue using current devices (Hortelano et al., 2021). Other recent provisioning works are concerned with managing the computational loads automatically (Grasso et al., 2022) and enabling multi-tenant access networks (Bonati et al., 2023).

Only Martini et al. (2022) and Ridhawi et al. (2022) are concerned with increased automation of the provisioning and configuration process. The first one proposed an intent-based zero-touch service chaining layer that provides the programmable provision of service chain paths in edge cloud networks; while the second highlights and proposes potential solutions towards zero-touch networks for tactile internet. To date, a study on the potential improvements from increasing automation enabled by AI is yet to be performed, therefore the motivation for this work.

2.3. Mass configuration

The zero-touch configuration has its origin in 1997 and was introduced by Stuart Cheshire under the term “Zeroconf” in a post of Net-Thinkers emailing list (Cheshire, 2005-2006). Ultimately, this term was renamed “Bonjour” by Apple in 2002. The main idea of Zeroconf was to automatically provision a network by enabling the devices to obtain a dynamic host configuration protocol (DHCP) address and making a request to a remote server for maintaining the latest software configuration data (Cheshire and Aboba, 2005). However, one issue with Zeroconf was the assumption that the devices are automatically assigned an IP address. Later, the authors of Anon (0000) circumvented this assumption by providing a solution with link-local addressing, which is perhaps one of the most significant steps towards a true zero-touch provisioning method. A link-local address is a network address that is assigned solely for communications within the network segment to which the host is connected. These addresses are often only utilized in circumstances, when no external address configuration exists, such as DHCP.

Nonetheless, a new standard, namely WiFi Aware, also known as neighbour awareness networking (NAN) is being standardized in the WiFi Alliance enabling the certified devices to continuously discover nearby services, applications and devices while operating in the background with less energy consumption (Camps-Mur et al., 2015). The concept of Wi-Fi Aware enables real-time, energy-efficient discovery, ranging and connectivity for peer-to-peer connections and multicast applications (Camps-Mur et al., 2015). There is only a paucity of contributions in the literature that address zero-touch provisioning using Wi-Fi Aware. For example, unintended connection request arriving from neighbour devices is one of the inherent characteristics of Wi-Fi Aware and is addressed in Cheshire (2005-2006). NAN-based IoT solutions including mass-provisioning of IoT devices are still in their infancy despite the fact that they can be greatly beneficial for the

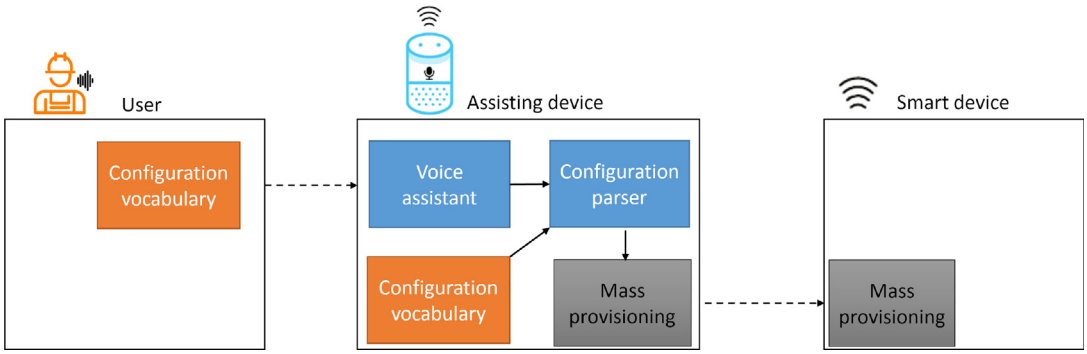


Fig. 2. The four functional components of HANNA: configuration vocabulary, voice assistant, configuration parser and mass provisioning.

zero-touch provisioning of massive IoT devices owing to their rapid connectivity (quick discovery and ranging procedures) and ease-of-use characteristics.

Perhaps 6TiSCH (Kalita and Khatua, 2022) and its various implementations are the most used approach for industrial IoT. Whisper (Municio et al., 2018), one such implementation, has already exploited the idea of adding an external node to automatically configure the network in terms of routing and scheduling. Scalability, especially in industrial environments is still being improved (Orozco-Santos et al., 2022) as their wireless nature and autonomous operation are challenging. While most of the current 6TiSCH security implementation relies on the secure joining mechanisms defined in Vučinić et al. (2019), also such aspects are being improved in terms of authentication in view of industrial applications (Haj-Hassan et al., 2022).

3. HANNA framework: Design and realization

In the following, we propose a human-friendly mass provisioning and configuration framework HANNA which consists of four main blocks, namely the configuration vocabulary, the voice controller, the configuration parser and the mass provisioning as depicted in Fig. 2. The first three blocks are embedded in the assisting device and ensure human-friendly machine communication while the last one involves both assisting and smart devices ensuring scalable machine-to-machine communication.

3.1. Configuration vocabulary

The configuration vocabulary requires the definition of domain-specific vocabulary which is significantly more technical than general-purpose vocabulary (Fortuna and Mohorcic, 2009). The vocabulary needs to be defined in such a way as to cover all the settings needed for efficient configuration and provisioning. However, the emphasis needs to be on human friendliness hence making it less intimidating in terms of technical knowledge, so that non-expert users with minimal or no training are able to use it. The vocabulary has to be consistent between the user and the assisting device as also visually represented in Fig. 2 since the device expects certain keywords that the user needs to specify in order to be able to trigger the provisioning.

A vocabulary for automatic speech recognition can be classified into three groups depending on its size i.e. the number of words it uses. The vocabulary is considered small if it uses less than 100 words, medium if it uses between 100 and 1000 words and large if it uses more than 1000 words (Sneha et al., 2018).

As no domain-specific vocabulary is available, we propose a small vocabulary since it uses mostly use-case-specific words, therefore it can provide very high detection accuracy (Qiao et al., 2010). First, in Table 1 we propose a list of configuration keywords that cover all aspects of IoT provisioning and configuration. It can be seen from the last three columns of the table that the keywords are grouped in three depending on their scope. According to the first column of the table, the first four keywords address operations regarding the powering of devices, then according to the second column of the table the fourth to seventh keywords address network-related operations, while the last keyword is dedicated to credentials-related operations as per the last column of the table. These keywords should be recognized with high fidelity by the voice controller as they are essential in triggering the automated scripts for mass configuration.

Table 1
Configuration settings keywords.

	Power	Network	Credentials
ON	✓	✗	✗
OFF	✓	✗	✗
standby	✓	✗	✗
Reset	✓	✓	✗
Connect	✗	✓	✗
Scan	✗	✓	✗
Address	✗	✓	✗
Distribute	✗	✗	✓

The first is powering the device which includes settings “ON”, “OFF”, “standby” and the response to the user when the setting changes. The second important setting is related to the network which can also be restarted in case of errors. However, the most important is the “connect” which enables the user to connect to the internet based on voice command which returns “success” or “fail” to the user. The last setting is reserved for mass provisioning and is concerned with the distribution of the network credentials once the assisting device is connected. It also returns “success” or “fail” depending on how the provisioning is finished.

Based on the configuration keywords, we propose the following natural language dialogues between the technician and the assisting device for each of the three scopes identified in Table 1: power, network and credentials. In the following command/response sequence boxes (natural language dialogues), C denotes the commands expected from the technician while R abbreviates the responses of the assisting device as a follow-up of the execution of the command instructions.

First, we provide vocabulary related to power operations:

C1: **HANNA power ON/OFF / reset/standby**.
R11: Success: The power setting **ON/OFF / reset/standby** was successful.
R12: Fail: The power configuration **ON/OFF / reset/standby** was not successful. Proceed with a manual check.
C2: **HANNA reset** the **power** of **device** with **address FF**.
R21: Success: The reset was successful.
R22: Fail: Reset was not successful. Proceed with a manual check.

Next, we provide an example list of vocabulary related to networks:

C3: **HANNA scan/connect / reset** the **network**.
R31: Scan: The following networks are available. Chose the one to connect.
R32: Success: Connection to the network was successful
R33: Fail: Connection to the network was not successful. A possible reason is that the authentication credentials did not match. Try choosing a different network or resetting the network controller.
C4: **HANNA reset** the **network** of **device** with **address FF**.
R41: Success: The reset was successful.
R42: Fail: Reset was not successful. Proceed with a manual check.

Finally, we provide an example list of vocabulary related to connection credentials:

C5: **HANNA distribute** the connection **credentials** between **devices**.
R51: Success: The connection credentials were distributed successfully.
R52: Fail: The distribution of connection credentials was not successful for devices with the following addresses FF, EE. Try resetting the network and power on these devices or proceed with a manual check.

3.2. Voice assistant

The voice assistant is a fundamental building block of HANNA and it is incorporated in the assisting device as depicted in Fig. 2. Its purpose is to translate the voice commands of the user to computer-generated text which can be then parsed by the configuration parser to identify the requested command based on the configuration vocabulary. Additionally, it needs to be adapted to noisy industrial environments where there may be background noise from both humans and machines during the smart device deployment process.

In principle, a voice controller uses ASR techniques that translate the analogue acoustic signal into computer-generated text. The typical ASR architecture consists of four main building blocks (Saon and Chien, 2012; Kumar and Singh, 2019): (1) feature extraction, (2) acoustic model, (3) language model and (4) decoder. Feature extraction is responsible for acoustic signal processing and the translation into feature vectors. The acoustic model forms a statistical representation of sounds that correspond to words. The language model is a list of words including their probabilities of appearing in the detected order while the decoder is responsible for matching the detected sounds with the words from the language model.

For the realization of the voice controller we selected Google Speech-to-Text[1] as the best performing ASR according to the evaluations in Kim et al. (2019) with a 4.1% WER. Additionally, we also evaluate CMU Sphinx (Lamere et al., 2003) (Open source solution, offline), Wav2Vec (Baevski et al., 2020) (Open source solution based on transformers, offline) and TextFromToSpeech[2] (Commercial solution, online) as alternative engines able to transcript audio and then pass the transcript to the configuration keyword parser.

[1] https://cloud.google.com/speech-to-text/
[2] https://www.textfromtospeech.com/

3.3. Configuration parser

Assuming that manual configuration scripts and tools already exist, the only missing link is the configuration mapping from the natural language commands interpreted from the user's voice to the system tools and scripts. We envision using static configuration keyword parsers form the transcript based on ASR as well as recently introduced more advanced solutions based on AI capable of learning based on the context and thus significantly reducing WER by adding vectorized contextual learning capabilities to ASR (Pundak et al., 2018). The text-to-script mapping, which can take the shape of a declarative language, gets automatically parsed by the infrastructure automation software (i.e. embedded software) and triggers mass provisioning and/or configuration as depicted in Fig. 2.

The configuration parser was realized as a mapping between the keywords identified in Table 1 and Unix-compatible tools and scripts. The specific parameters for these commands are provided by the technician during the conversation with the voice assistant for instance exemplified in **C3** above. To better emphasize the described approach we provide an example which illustrates the specific system command executed under a Unix-compatible OS when a network reset command **C3** is transcribed from the users voice:

```
$ /etc/init.d/networking restart
```

3.4. Mass provisioning with 6TiSCH protocol

The mass provisioning block is concerned with distributing provisioning and configuration-related information from the *assisting device* to all the *smart devices* on site thus realizing the scalable machine-to-machine functionality of HANNA. A compliant realization of the framework requires that the assisting device as well as all smart devices utilize the same communication protocol that ensures mass provisioning. Therefore, the most suitable protocols to realize mass provisioning are the ones in the category of ad-hoc, multi-hop communication protocols (Broch et al., 1998). It is important that the authentication parameters reach all *smart devices*, even the ones not having a direct communication link with the *assisting device*.

For the mass provisioning mechanism used by the assisting device and smart devices, we choose 6TiSCH as one of the most adopted industrial-grade IoT technologies currently available. We choose 6TiSCH since the Time Slotted Channel Hopping (TSCH) mode is designed to provide reliable, low-latency communication in a multi-hop and scalable Industrial Internet of Things (IIoT). The TSCH link layer protocol allows nodes to change their physical channel after each transmission to eliminate interference and degradation due to multipath fading. It builds upon the IEEE 802.15.4 PHY layer, normally using the 2.4 GHz license-free band.[3] Within 6TiSCH, some approaches like Whisper (Municio et al., 2018) have already exploited the idea of adding an external node to automatically configure the network in terms of routing and scheduling. For security provisioning, most of the current 6TiSCH implementation relies on the secure joining mechanisms defined in Vučinić et al. (2019).

To give an overview of the functional structure of Fig. 2, once the user has spoken a particular command from the predefined vocabulary, it is received by the voice assistant and translated into computer-generated text that can then be parsed by the configuration parser to identify the requested command based on the configuration vocabulary. Once the requested command is received, the associated configuration file is transmitted from the assisting device to other devices to be configured in a multi-hop manner owing to the 6TiSCH protocol.

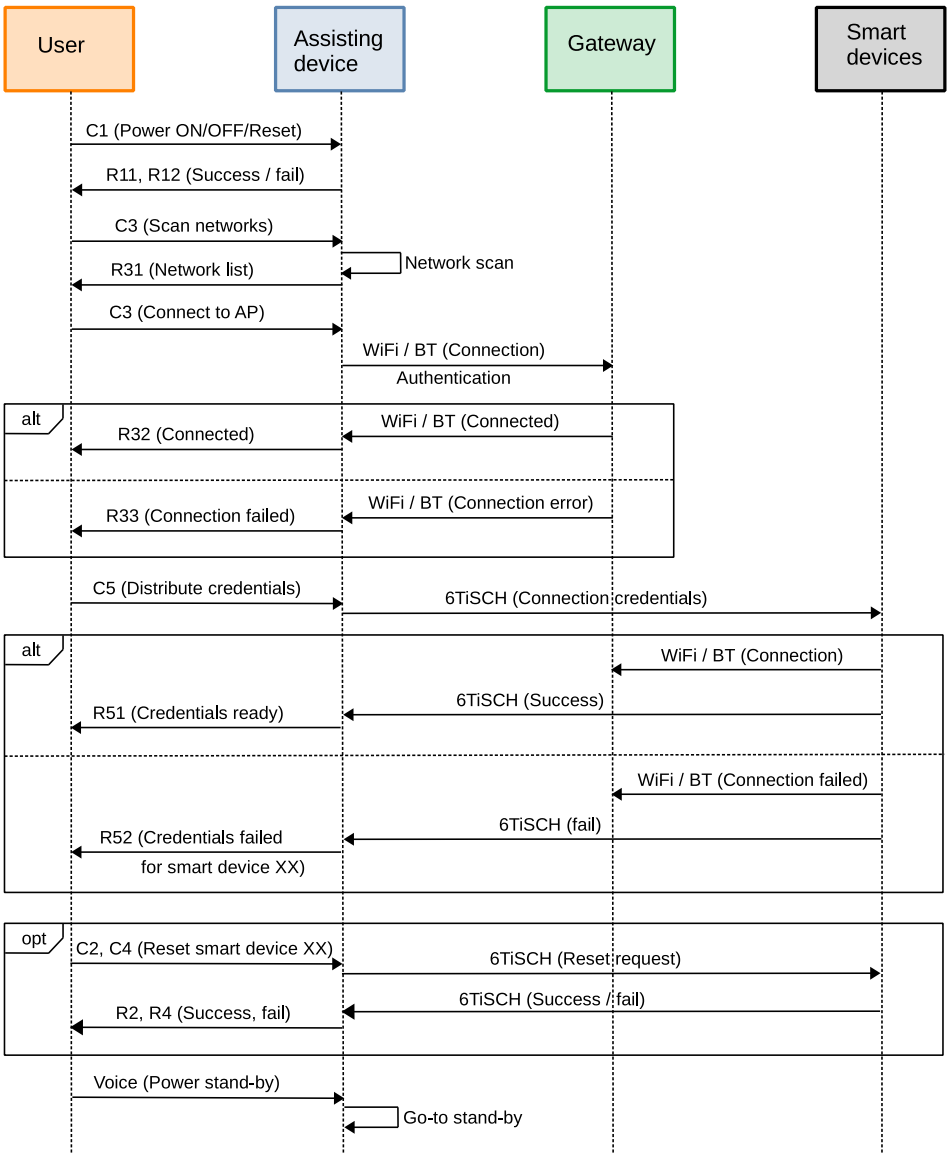


Fig. 3. HANNA reference implementation sequence diagram using the proposed vocabulary, the selected commercially available voice controller, custom parser and 6TiSCH as the selected mass-provisioning protocol.

3.5. HANNA communication diagram and fail-safe procedures

The communication diagram used in HANNA is presented in Fig. 3 in the form of a sequence diagram explaining the interaction between the actors and selected techniques for the realization of the proposed framework.

The process in Fig. 3 starts by initiating a voice command to power on the assisting device **C1** which responds with "Success" **R11** or "Fail" **R12** as proposed in the vocabulary in Section 3.1. Next, if there is no network connectivity in the assisting device, the user can issue a command **C3** to scan available networks which results in the list of available networks provided within **R31**. The user then selects the network connection which instructs the device to connect to the gateway over WiFi or Bluetooth depending on the underlying system. When the device successfully connects to the gateway it responds to the user with "Connected" **R32**. Alternatively, the device can reply with connection failed **R33** in case something went wrong during the connection and advise the user to retry. The next command **C5** instructs the assisting device to distribute the connection credentials to smart devices based on a multi-hop wireless mesh network for IoT devices 6TiSCH. When the smart devices receive the connection credential from assisting device, they can connect to the main network and are now fully operational. After the smart devices were successfully connected to the gateway the assisting device responds with "Credentials ready" **R51** to the user. Alternatively, in the case that the distribution of credentials fails, the voice assistant with the response **R52** suggests resetting or manually checking some devices. Therefore, the next step, restarting smart devices with commands **C2** or **C4** is optional. The last step is to put the assisting device to standby with the command **C1** until there are more configuration settings to perform.

Additionally, only when a spoken command is matched with the predefined vocabulary, it is transmitted to other devices to be configured. Otherwise, the user has to speak the command again until it is matched with one of the vocabularies. Due to this fail-safe procedure, it is quite unlikely that a spoken command from the known command list, such as reset, may be understood as another predefined command, such as connect. Whichever device receives a command, it initiates the provisioning and then the configuration.

[3] The 868/915 MHz bands are also available at different bit rates.

Table 2
Voice recordings summary.

Parameter	Value
Use Cases (UC)	5 background noise types (baseline, factory, office, talking, traffic)
Users	5 female, 5 male
Language	English (all accents)
Recordings	50
Encoding	WAV
Sampling rate [Hz]	48000

A device can crash or encounter a failure/conflict/error in the logic above 6TiSCH or within 6TiSCH itself. Assuming the issue (crash/conflict/error) happens in the logic above 6TiSCH (e.g., when reading the conf and trying to connect to the WiFi), the solution is to re-send a provisioning message (with the connection credentials) in unicast (in a multi-hop manner) to those nodes that failed as per C5 in Fig. 2. It is known which nodes failed because they issue a Fail command over 6TiSCH, or they do not issue a command at all (are silent). If a device crashes or is unresponsive, then it auto-reboots after a certain amount of duration and may require a manual check.

If the failure/conflict/error happens in 6TiSCH, the protocol-specific procedures are triggered. For instance, a bootloader triggers a process that restarts automatically the 6TiSCH radio upon failure and re-joins the network autonomously (it will receive the credentials in the joining phase from the gateway). Also, a keepalive message that is sent every 5 min to every 6TiSCH node can be configured. If any of the nodes do not respond on 3rd attempt, the user is notified that the node is malfunctioning and needs to be manually checked.

4. Experimental evaluation

In order to evaluate HANNA, we focus on two aspects. Firstly, we assess the performance of the *assisting device* in interacting with the user and generating the configurations needed for triggering the mass provisioning. For this, we defined five use cases (UC) in which the user (i.e. technician) from Fig. 1 may be male or female and may talk to the device under various noise conditions. The first UC represents the *baseline* and assumes a quiet room where the user interacts with the assisting device.

For UCs 2–5, four types of representative noisy environments were selected. In the second UC, we assume the devices are provisioned and configured around a metalworking *factory* background noise and are intended to evaluate usage in industrial environments. In the third UC, we consider that the devices are being provisioned and configured in a large *office* environment with background noise containing quiet background talk, phones ringing, people walking etc. In the fourth UC, we assume *talking* in the background acquired from an informational video, which tests the engines' ability to isolate the correct speaker. Finally, for the fifth UC, we assume a busy street *traffic* that represents smart city-like deployment scenarios with sounds of motorized vehicles and pedestrians.

Recordings of all the conversations proposed in Section 3.1 according to the sequences proposed in Section 3.5 were recorded for UC1-5 with different background noises for different speakers, five male and five female. The recordings were all recorded in English from subjects that are native speakers to subjects that have mild or stronger accents[4] with parameters as summarized in Table 2. The noise was injected as a random slice from a prerecorded audio file and overlaid over the original audio with volumes normalized in such a way that SNR = 3 dB. This was done 50 times for every combination of recording and noise, to increase the reliability of the performance evaluation.

[4] Recordings, noise and generation scripts, https://zenodo.org/record/7933500#.ZGAJYC8RqvU

Table 3
Detection performance for speech-to-text engines averaged over all users and all UCs.

Engine \ Metric	SR	SP	KD	WER
Google	0.45/0.24	0.47/0.24	0.60/0.28	0.43/0.27
Sphinx	0.01/0.03	0.01/0.03	0.06/0.07	0.68/0.16
TFTS	0.28/0.26	0.33/0.30	0.49/0.37	0.25/0.19
Wav2Vec	0.17/0.16	0.21/0.19	0.47/0.24	0.42/0.21

The performance of the four speech-to-text engines selected in Section 3.2 was evaluated on these recordings to assess their suitability for powering the assisting device according to the following metrics. The word error rate (WER) shows how many words were correctly recognized in a sentence and was used to evaluate the correct identification of the words in the human-to-machine conversations listed in Section 3.2. The keyword detection rate (KD) measures the recognition performance as the ratio of correctly identified keywords presented in Table 1. The sentence recognition rate (SR) measures how often full sentences were correctly recognized. For a sentence to be considered fully correct all keywords and parameters must be recognized. To remove inconsistencies due to pronunciation, a parser was used for eliminating common mistakes such as *standby* being interpreted as *stand by*, *power off* as *power of*, *address* as *a dress* etc. The last metric, sentence recognition rate after parser fixes (SP) determines how many sentences are fully recognized after mistakes were fixed by the parser described in Section 3.3. All metrics are provided as both mean and standard deviation values calculated across the entire group relevant to the given experiment in the form of "mean/STD".

Secondly, we evaluate the average provisioning time required by the mass provisioning mechanism selected in Section 3.4 against the traditional manual provisioning and the state-of-the-art zero-touch technique proposed in Boskov et al. (2020).

4.1. Evaluation of the human-to-machine provisioning

A. Overall findings. Table 3 presents the performance of different speech-to-text engines across all four metrics: SR, SP, KD and WER. This data aggregates across all users and all UCs.

Google STT and TFTS exhibit the best KD and SP values, which are the most critical metrics for evaluating the usability of such a provisioning and configuration system. These metrics gauge the efficacy of recognizing the necessary keywords and parameters for provisioning. Google STT yielded a KD of 0.60 and an SP of 0.47, with standard deviations of 0.28 and 0.24, respectively. TFTS presented a KD of 0.49 and an SP of 0.33, with standard deviations of 0.37 and 0.30, respectively. The relatively high variance of results can be attributed to synthetically added noise, introducing additional complexity to the recognition problem.

Wav2Vec exhibited somewhat worse performance, with a KD of 0.47 and an SP of 0.21. The significant standard deviation again reflects the influence of synthetically added noise on the system's performance.

CMU Sphinx demonstrated a very low KD of 0.06 and an SP of 0.01. To anticipate, these results suggest that CMU Sphinx may not be robust to noise, affecting its ability to be deployed across a wide range of UCs. This is also reflected in its high WER, which indicates the overall poor performance and is confirmed by the following tables in this section.

Comparing the values for SR and SP highlights the parser's impact on the overall performance. The parser noticeably enhanced performance in all cases, scoring an additional 2–5 percentage points, except for CMU Sphinx, where it did not significantly improve the outcome. This performance could potentially be further amplified by designing a more effective parser.

Table 4
Effect of working environment on speech-to-text engines performance.

Use case	Baseline		Factory		Office		Talking		Traffic	
Engine	KD	SP	KD	SP	KD	SP	KD	SP	KD	SP
Google	0.76/0.27	0.64/0.28	0.75/0.24	0.58/0.23	0.55/0.25	0.43/0.20	0.36/0.20	0.27/0.15	0.75/0.24	0.60/0.23
Sphinx	0.30/0.13	0.08/0.09	0.03/0.04	0.00/0.02	0.04/0.05	0.01/0.03	0.08/0.07	0.01/0.03	0.08/0.07	0.01/0.03
TFTS	0.85/0.12	0.55/0.21	0.56/0.38	0.42/0.31	0.46/0.38	0.31/0.30	0.56/0.31	0.34/0.25	0.36/0.38	0.25/0.30
Wav2Vec	0.75/0.14	0.49/0.18	0.49/0.25	0.22/0.17	0.50/0.23	0.23/0.18	0.31/0.18	0.08/0.12	0.59/0.21	0.30/0.20

Table 5
Effects of gender on speech to text engine performance.

Gender	Male		Female	
Engine	Baseline KD	Noise KD	Baseline KD	Noise KD
Google	0.90/0.12	0.73/0.14	0.62/0.31	0.53/0.24
Sphinx	0.36/0.10	0.10/0.07	0.24/0.13	0.04/0.06
TFTS	0.88/0.05	0.54/0.38	0.82/0.16	0.35/0.31
Wav2Vec	0.81/0.06	0.60/0.10	0.69/0.17	0.39/0.24

B. Effect of background noise. Table 4 displays the performance of the selected speech-to-text engines, each listed row-wise, in different working environments, presented column-wise. The environments include the baseline, factory, office, talking, and traffic noise conditions. The table provides the keyword detection (KD) and sentence recognition rate after parser fixes (SP) metrics for each engine in each noise setting.

In the baseline setting, TFTS exhibits the best performance with a KD of 0.85 and an SP of 0.55. Both Google and Wav2Vec show similar performance, with Google having a KD of 0.76 and an SP of 0.64, and Wav2Vec presenting a KD of 0.75 and an SP of 0.49. CMU Sphinx, on the other hand, demonstrates a relatively low performance with a KD of 0.30 and an SP of 0.08.

When exposed to factory noise, Google maintains its higher SP with a KD of 0.75 and an SP of 0.58, while TFTS has a KD of 0.56 and an SP of 0.42. Wav2Vec's performance drops slightly in this environment, with a KD of 0.49 and an SP of 0.22. CMU Sphinx's performance further decreases, with a KD of 0.03 and an SP of 0.00.

In the office noise setting, Google's KD and SP scores drop to 0.55 and 0.43, respectively, while TFTS has a KD of 0.46 and an SP of 0.31. Wav2Vec presents a KD of 0.50 and an SP of 0.23, and CMU Sphinx's performance remains low with a KD of 0.04 and an SP of 0.01.

Under the talking noise condition, TFTS demonstrates better performance with a KD of 0.56 and an SP of 0.34, while Google's performance decreases further, with a KD of 0.36 and an SP of 0.27. Wav2Vec has a KD of 0.31 and an SP of 0.08. CMU Sphinx exhibits a KD of 0.08 and an SP of 0.01.

Finally, in traffic noise, Google's performance returns to near baseline levels with a KD of 0.75 and an SP of 0.60. Wav2Vec exhibits a KD of 0.59 and an SP of 0.30, while TFTS presents a KD of 0.36 and an SP of 0.25. CMU Sphinx maintains its low performance with a KD of 0.08 and an SP of 0.01.

Based on the results presented in Table 4, it can be observed that the performance of the speech-to-text engines is affected differently by different types of noise. Statistical analysis revealed significant differences in the keyword detection rates for different noise conditions. Specifically, the p-values for the t-tests comparing the no noise condition to the traffic noise, office noise, talking noise, and factory noise conditions were 8.87e−05, 1.06e−07, 2.72e−15, and 0.00038, respectively. These results indicate that the presence of noise has a significant impact on the performance of speech-to-text engines, with talking noise having the strongest effect and traffic and factory noise having the least impact.

C. Effect of gender. Table 5 illustrates the effects of gender on the performance of various speech-to-text engines. The table presents the mean baseline keyword detection (KD) scores and the mean KD scores under noisy conditions for both male and female speakers.

For male speakers, Google STT achieved a baseline KD of 0.90 and a noise KD of 0.73. TFTS followed closely with a baseline KD of 0.88 and a noise KD of 0.54, with a notably high standard deviation of 0.38 under noisy conditions, consistent with results from 4. Wav2Vec exhibited a baseline KD of 0.81 and a noise KD of 0.60, while CMU Sphinx had a baseline KD of 0.36 and a noise KD of 0.10.

For female speakers, TFTS displayed the best performance with a baseline KD of 0.82 and a noise KD of 0.35, similarly with a relatively high standard deviation of 0.31 under noisy conditions. Wav2Vec followed with a baseline KD of 0.69 and a noise KD of 0.39. Google STT achieved a baseline KD of 0.62 and a noise KD of 0.53, while CMU Sphinx had the lowest performance, with a baseline KD of 0.24 and a noise KD of 0.04.

The table demonstrates a noticeable disparity in performance between male and female speakers, with speech-to-text engines generally exhibiting higher KD scores for male speakers in both baseline and noisy conditions. The t-test performed to compare the performance of speech-to-text engines across genders resulted in an extremely low p-value of around 9.92e−110, indicating a statistically significant difference between male and female users in terms of keyword detection rates. This highlights the importance of considering gender as a factor when evaluating the performance of speech-to-text engines and the need for further research to improve their performance for female users.

D. Effect of accent. Table 6 presents the performance of speech-to-text (STT) engines with respect to users' language proficiency levels, which are categorized as native, proficient, and inexperienced. The results are reported in terms of keyword detection (KD) rates under both baseline and noise conditions.

For native speakers, Google's STT engine exhibits the highest performance in both baseline (KD: 0.96, STD: 0.00) and noisy conditions (KD: 0.76), followed by TFTS (baseline KD: 0.93, noise KD: 0.76), Wav2Vec (baseline KD: 0.89, noise KD: 0.54), and Sphinx (baseline KD: 0.48, noise KD: 0.13).

Among proficient speakers, TFTS performs best in both baseline (KD: 0.89) and noisy conditions (KD: 0.64), followed by Wav2Vec (baseline KD: 0.82, noise KD: 0.54) and Google (baseline KD: 0.78, noise KD: 0.61). Sphinx has the lowest performance (baseline KD: 0.32, noise KD: 0.07).

For inexperienced speakers, TFTS has the highest baseline KD rate (0.81), followed by Google (KD: 0.71), Wav2Vec (KD: 0.67), and Sphinx (KD: 0.24). In noisy conditions, Google outperforms the other engines with a KD rate of 0.56, followed by Wav2Vec (KD: 0.40), TFTS (KD: 0.31), and Sphinx (KD: 0.04).

4.2. Discussions on the performance of the voice-assistant

Overall, the performance of all engines is negatively affected by the presence of noise and decreases with the users' English proficiency level. The p-value for comparing native and proficient speakers is 8.12e−11, indicating a statistically significant difference between the two groups. Similarly, the p-value for comparing native and inexperienced speakers is 1.45e−66, indicating a highly significant difference in performance between the two groups. These results demonstrate that language proficiency has a significant impact on the performance of speech-to-text engines.

The quality of recording considerably affects the accuracy of detection, i.e. the signal-to-noise ratio (SNR) needs to be at a certain level for

Table 6
Effects of language proficiency on speech to text engines detection performance.

Proficiency	Native		Proficient		Inexperienced	
Engine	Baseline KD	Noise KD	Baseline KD	Noise KD	Baseline KD	Noise KD
Google	0.96/0.00	0.76/0.13	0.78/0.16	0.61/0.25	0.71/0.36	0.56/0.32
Sphinx	0.48/0.00	0.13/0.08	0.32/0.10	0.07/0.06	0.24/0.13	0.04/0.05
TFTS	0.93/0.00	0.76/0.06	0.89/0.03	0.64/0.33	0.81/0.16	0.31/0.35
Wav2Vec	0.89/0.00	0.54/0.17	0.82/0.02	0.54/0.20	0.67/0.15	0.40/0.26

Table 7
6TISCH mass provisioning protocol simulation parameters.

Parameter	Value
Area	1 km^2
Frequency	2.4 GHz
Num. nodes	10-100
Topology	Random
Avg. number of hops	1.7
Min PDR	0.8
Num. channels	16
Time slot duration	10 ms
Slotframe lenght	101
EB probability	33%
TX queue size	10 pkts
SF	MSF
RPL OF	OF0
DAO period	60 s

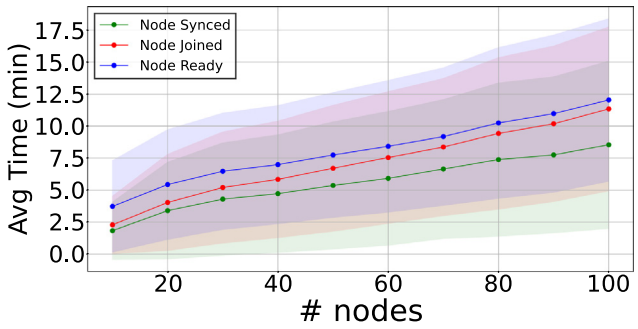
detection to work adequately. Moreover, the speed of pronunciation, as well as the speaker's volume, are important factors for accurate detection. The errors typically appear either because of an incomplete transcript or phonetically similar words i.e. *connected* instead of *connect*. There are still some random unusually transcribed words left after parser error correction i.e. *a dress* or *Edwards* instead of *address*, which would be very difficult to statically correct in advance since there are too many possible outcomes. However, this could be greatly improved by calculating a vectorized distance between the configuration settings and recordings i.e. using the Hamming distance or enhancing the detection by adding contextual learning capabilities to ASR (Pundak et al., 2018).

4.3. Evaluation of machine-to-machine mass provisioning

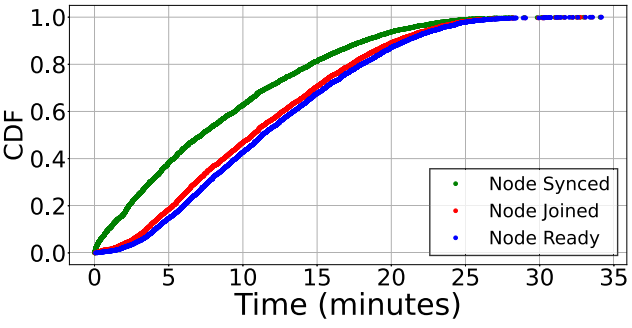
In order to compare how our proposed mass provisioning method would behave when integrated with existing provisioning approaches, the following approaches have been evaluated and simulated:

Multi-hop. 6TiSCH protocol stack as the mass provisioning method is used to present results on automatic mass provisioning triggered in one single node in a multi-hop network. To evaluate the provision time we have used the 6TiSCH Simulator (Municio et al., 2019), using the parameters described in Table 7. Some simulation parameters include Packet Delivery Ratio (PDR), Enhanced Beacon (EB) probability, 6TiSCH Minimal Scheduling Function (MSF), RPL Objective function (OF) Zero and Destination Advertisement Object (DAO) messages period. We also assume that the 6TiSCH provisioning guidelines are similar to those in Boskov et al. (2020).

The provisioning time in 6TiSCH networks can be divided into three components: a) synchronization time, which is the time it takes for a node to receive a first EB and synchronize with the TSCH schedule, b) joining time, which is the time it takes for a node to securely join the network using shared cells following Vučinić et al. (2019), and c) ready-to-send time, i.e. the time it takes for a node to allocate its first dedicated cell and is able to send data packets. For different network sizes, we show the evolution of the average time of each component in Fig. 4(a). The main takeaway of these results is that for network sizes of 100 nodes or lower, the *Node Ready* time is always under 30 min for the 99.7% of the nodes, while 50% of the nodes are always ready and provisioned within 11.2 min as seen in Fig. 4(b).



(a) Time required by a 6TiSCH network to get ready to receive/send data for different network sizes.



(b) CDF of synchronization time for a 100-node 6TiSCH network.

Fig. 4. Machine-to-machine based provisioning time improvements.

Zero-touch provisioning. ZTP provisioning methods improve the provisioning times through the use of zero-touch mechanisms using both WiFi and BT. However, it also requires an assisting input device for provisioning, and thus the evolution of the provision time with the number of devices can also be assumed linear. The provisioning time considered for both technologies is 37.13 s and 21.12 s, respectively (Boskov et al., 2020). The numbers represent averages for each technology separately performed under two distinct communication conditions, i.e. line-of-sight (LOS) and non-LOS (NLOS).

Manual provisioning. The manual provisioning method is used as a benchmark and includes two types of manual provisioning: one performed by a non-expert operator and one by an expert operator. The average provision time is calculated linearly, assuming 131.87 s and 46.88 s per device respectively for each mode (Boskov et al., 2020). Manual provisioning is a baseline evaluation method to explicitly understand the performance improvements that the automated ZTP solutions introduce, where the device was provisioned by one expert over 15 times that was familiar with the provisioning procedures, and by 15 other non-experts with no previous knowledge following step-by-step provisioning guidelines, as provided by Boskov et al. (2020).

To date, there are no directly and fairly comparable works in the literature, which also reveals the novelty of our platform. This is why we opt to compare current results with our previous paper findings in Boskov et al. (2020) as a benchmark on provisioning duration. Assuming the command recognition functionality works well (evaluated in Section 4.1 and discussed in Section 4.2), Fig. 5 compares the

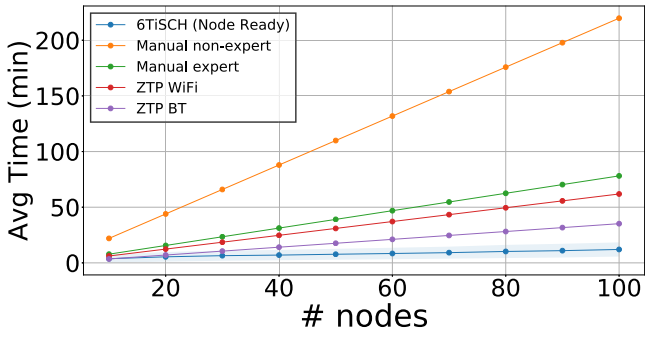


Fig. 5. Provisioning time comparison of each solution for each network size.

estimated time to provision to configure a 100 node network using a manual approach (Boskov et al., 2020), using a ZTP approach (Boskov et al., 2020) and using HANNA.

A comparison of the provision time of each approach for different network sizes is presented in Fig. 5. While for networks with less than 10 nodes under some conditions such as early after startup approaches like the ZTP BT and WiFi can outperform the mass provisioning while 6TiSCH based mass provisioning approach is always faster than others (Manual and ZTP) when there are more than 10 nodes in the network. The simulation in Fig. 5 shows that for 100 nodes, the proposed method can save up to 258 min on average in configuration time. This is mainly due to the linear behaviour of the manual approaches that do not leverage multi-hop broadcasting mechanisms as the 6TiSCH mass provisioning does. These results evidence the convenience of using our proposed human-friendly voice-assisted provisioning mechanisms together with automated multi-hop provision mechanisms (e.g., 6TiSCH) to realize a full end-to-end human-friendly voice-assisted mass provisioning system.

5. Conclusions and open challenges

In this paper, we addressed the problem of massive IoT deployments specifically the initial provisioning of authentication parameters as well as the initial configuration of smart devices. We introduced the HANNA framework which addresses the problem in a human-friendly manner by proposing the voice-based interaction in natural language and the distribution of authentication parameters using large-scale ad-hoc communications. Furthermore, we provided a reference implementation to illustrate the benefits of HANNA which indicate that HANNA-compliant frameworks can be realized using current state-of-the-art protocols and tools. Finally, we performed the evaluation of the HANNA reference implementation.

For human-to-machine communication, the results show the potential of existing speech-to-text engines for this application area and also reveal shortcomings with respect to the robustness of the engines in office-like working environments as well as with respect to user's gender and English language proficiency level. Out of four engines, the Google STT and Amberscript engines perform the best in all considered use cases. The parser that succeeds the STT engine, provides a meaningful increase in performance scoring an additional 3–4 percentage points. The keyword detection performance on traffic and factory background noises is generally within 10 percent of the baseline with a silent background. The office noise is handled slightly worse and a performance degradation of around 15% can be noticed. In our experiments, all the engines show inferior performance on females regardless of the type of noise considered for the respective use case, dropping around 15 percentage points in detection performance. Our results also show a noticeable drop in performance between native and non-native speakers, with Google STT and CMU Sphinx engines dropping almost 20 percentage points. The differences between proficient and inexperienced are comparatively much lower: around 5 percentage points on average.

With respect to machine-to-machine communication, the results suggest that our proposed approach performs better than the traditional sequential approaches of initial provisioning i.e. ZTP, in all evaluated categories, especially in situations where a large number of devices need to be provisioned and configured simultaneously. The proposed mass provisioning approach always outperforms manual provisioning for cases with more than ten devices.

Given the findings of this paper, as a future work, it would be interesting to develop a specialized domain-specific speech-to-text tool that is configurable or able to adapt to various background noises and capable of solving the gender robustness and language proficiency barriers by enabling multilingualism or training the engine on a more extensive set of language levels.

CRediT authorship contribution statement

Carolina Fortuna: Conceptualization, Methodology, Writing – original draft, Supervision, Data curation. **Halil Yetgin:** Conceptualization, Writing – review & editing. **Leo Ogrizek:** Methodology, Software, Data curation. **Esteban Municio:** Methodology, Software, Writing – original draft, Visualization. **Johann M. Marquez-Barja:** Supervision, Funding acquisition. **Mihael Mohorcic:** Writing – original draft, Funding acquisition.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data is available at https://zenodo.org/record/7933500#.ZGAJYC8RqvU.

Acknowledgements

This work was funded in part by the Slovenian Research Agency under the grant P2-0016.

References

Alibegović, B., Prljača, N., Kimmel, M., Schultalbers, M., 2020. Speech recognition system for a service robot - A performance evaluation. In: 2020 16th International Conference on Control, Automation, Robotics and Vision. ICARCV, pp. 1171–1176. http://dx.doi.org/10.1109/ICARCV50220.2020.9305342.

Anon, 0000. Zero Configuration Networking (Zeroconf), URL http://www.zeroconf.org/.

Baevski, A., Zhou, Y., Mohamed, A., Auli, M., 2020. wav2vec 2.0: A framework for self-supervised learning of speech representations. Adv. Neural Inf. Process. Syst. 33, 12449–12460.

Berggren, V., 2018. Towards zero-touch network operations, The Ericsson Blog, https://www.ericsson.com/en/blog/2018/9/towards-zero-touch-network-operations.

Bonati, L., Polese, M., D'Oro, S., Basagni, S., Melodia, T., 2023. NeutRAN: An open RAN neutral host architecture for zero-touch RAN and spectrum sharing. arXiv preprint arXiv:2301.07653.

Boskov, I., Yetgin, H., Vucnik, M., Fortuna, C., Mohorcic, M., 2020. Time-to-provision evaluation of IoT devices using automated zero-touch provisioning. In: GLOBECOM 2020 - 2020 IEEE Global Communications Conference. pp. 1–7. http://dx.doi.org/10.1109/GLOBECOM42002.2020.9348119.

Broch, J., Maltz, D.A., Johnson, D.B., Hu, Y.-C., Jetcheva, J., 1998. A performance comparison of multi-hop wireless ad hoc network routing protocols. In: Proceedings of the 4th Annual ACM/IEEE International Conference on Mobile Computing and Networking. MobiCom '98, Association for Computing Machinery, New York, NY, USA, pp. 85–97. http://dx.doi.org/10.1145/288235.288256.

Caballar, R.D., 2021. Programming by voice may be the next frontier in software development. IEEE Spectrum: Technol. Eng., and Science News https://spectrum.ieee.org/computing/software/programming-by-voice-may-be-the-next-frontier-in-software-development.

Camps-Mur, D., Garcia-Villegas, E., Lopez-Aguilera, E., Lambert, P., Raissinia, A., 2015. Enabling always on service discovery: WiFi neighbor awareness networking. IEEE Wirel. Commun. 22, 118–125. http://dx.doi.org/10.1109/MWC.2015.7096294.

Chaudhari, A., Asthana, A., Kaluskar, A., Gedia, D., Karani, L., Perigo, L., Gandotra, R., Gangwar, S., 2019. VIVoNet: Visually-represented, intent-based, voice-assisted networking. Int. J. Comput. Netw. Commun. (IJCNC) 11 (2), http://dx.doi.org/10.5121/IJCNC.2019.11201.

Chen, M., Zhang, Q., Song, Q., Qian, X., Guo, R., Wang, M., Chen, D., 2023. Neural-free attention for monaural speech enhancement towards voice user interface for consumer electronics. IEEE Trans. Consum. Electron..

Cheshire, S., 2005-2006. Zero configuration networking : The definitive guide / stuart cheshire and daniel h. Steinberg.. Zero Configuration Networking : The Definitive Guide. O'Reilly, Beijing.

Cheshire, S., Aboba, B., 2005. Dynamic configuration of IPv4 link-local addresses. IETF Internet Draft.

Davies, J., Fortuna, C., 2020. The Internet of Things: From Data to Insight. John Wiley & Sons.

Fortuna, C., Mohorcic, M., 2009. Dynamic composition of services for end-to-end information transport. IEEE Wirel. Commun. 16 (4), 56–62. http://dx.doi.org/10.1109/MWC.2009.5281256.

Fortuna, C., Yetgin, H., Mohorcic, M., 2022. Smart infrastructures: Artificial intelligence-enabled lifecycle automation. IEEE Ind. Electron. Mag. 2–12. http://dx.doi.org/10.1109/MIE.2022.3165673.

Georgila, K., Leuski, A., Yanov, V., Traum, D., 2020. Evaluation of off-the-shelf speech recognizers across diverse dialogue domains. In: Proceedings of the 12th Language Resources and Evaluation Conference. European Language Resources Association, Marseille, France, pp. 6469–6476, URL https://www.aclweb.org/anthology/2020.lrec-1.797.

Grasso, C., Raftopoulos, R., Schembra, G., 2022. Smart zero-touch management of UAV-based edge network. IEEE Trans. Netw. Serv. Manag. 19 (4), 4350–4368. http://dx.doi.org/10.1109/TNSM.2022.3160858.

Haj-Hassan, A., Imine, Y., Gallais, A., Quoitin, B., 2022. Zero-touch mutual authentication scheme for 6TiSCH industrial IoT networks. In: 2022 International Wireless Communications and Mobile Computing. IWCMC, IEEE, pp. 354–359.

Hansen, J.H., Hasan, T., 2015. Speaker recognition by machines and humans: A tutorial review. IEEE Signal Process. Mag. 32 (6), 74–99. http://dx.doi.org/10.1109/MSP.2015.2462851.

Hortelano, D., Olivares, T., Ruiz, M.C., 2021. Providing interoperability in Bluetooth mesh with an improved provisioning protocol. Wirel. Netw. 27, 1011–1033.

Inthavisas, K., Lopresti, D., 2012. Secure speech biometric templates for user authentication. IET Biometrics 1 (1), 46. http://dx.doi.org/10.1049/iet-bmt.2011.0008.

John Wilhelm, S.M., 2017. A Device Manufacturer's Perspective: Addressing a Key Customer Need to Onboard Iot Devices Securely and Efficiently. Tech. Rep., Kaiser Associates, Inc.

Kalita, A., Khatua, M., 2022. 6TiSCH–IPv6 enabled open stack IoT network formation: A review. ACM Trans. Internet Things 3 (3), 1–36.

Kim, J.Y., Liu, C., Calvo, R., McCabe, K., Taylor, S., Schuller, B., Wu, K., 2019. A comparison of online automatic speech recognition systems and the nonverbal responses to unintelligible speech. arXiv abs/1904.12403.

Kumar, Y., Singh, N., 2019. A comprehensive view of automatic speech recognition system - A systematic literature review. In: 2019 International Conference on Automation, Computational and Technology Management. ICACTM, pp. 168–173. http://dx.doi.org/10.1109/ICACTM.2019.8776714.

Lamere, P., Kwok, P., Gouvea, E., Raj, B., Singh, R., Walker, W., Warmuth, M., Wolf, P., 2003. The CMU SPHINX-4 speech recognition system. In: Ieee Intl. Conf. on Acoustics, Speech and Signal Processing, Vol. 1. Icassp 2003, Hong Kong, pp. 2–5.

Lee, B.M., Patil, M., Hunt, P., Khan, I., 2019. An easy network onboarding scheme for Internet of Things networks. IEEE Access 7, 8763–8772. http://dx.doi.org/10.1109/ACCESS.2018.2890072.

Longo, C.F., Longo, F., Santoro, C., 2021. Caspar: Towards decision making helpers agents for IoT, based on natural language and first order logic reasoning. Eng. Appl. Artif. Intell. 104, 104269. http://dx.doi.org/10.1016/j.engappai.2021.104269, URL https://www.sciencedirect.com/science/article/pii/S0952197621001160.

Marikyan, D., Papagiannidis, S., Rana, O.F., Ranjan, R., Morgan, G., 2022. "Alexa, let's talk about my productivity": The impact of digital assistants on work productivity. J. Bus. Res. 142, 572–584.

Martini, B., Gharbaoui, M., Castoldi, P., 2022. Intent-based zero-touch service chaining layer for software-defined edge cloud networks. Comput. Netw. 212, 109034.

Municio, E., Daneels, G., Vučinić, M., Latré, S., Famaey, J., Tanaka, Y., Brun, K., Muraoka, K., Vilajosana, X., Watteyne, T., 2019. Simulating 6TiSCH networks. Trans. Emerg. Telecommun. Technol. 30 (3), e3494. http://dx.doi.org/10.1002/ett.3494.

Municio, E., Marquez-Barja, J., Latré, S., Vissicchio, S., 2018. Whisper: Programmable and flexible control on industrial IoT networks. Sensors 18 (11), http://dx.doi.org/10.3390/s18114048, URL https://www.mdpi.com/1424-8220/18/11/4048.

Orozco-Santos, F., Sempere-Payá, V., Silvestre-Blanes, J., Vera-Pérez, J., 2022. Scalability enhancement on software defined industrial wireless sensor networks over TSCH. IEEE Access 10, 107137–107151.

Pauzet, O., 2010. Cellular Communications and the Future of Smart Metering. Sierra Wireless, Inc.

Pundak, G., Sainath, T.N., Prabhavalkar, R., Kannan, A., Zhao, D., 2018. Deep context: End-to-end contextual speech recognition. In: 2018 IEEE Spoken Language Technology Workshop. SLT, pp. 418–425. http://dx.doi.org/10.1109/SLT.2018.8639034.

Qiao, F., Sherwani, J., Rosenfeld, R., 2010. Small-vocabulary speech recognition for resource-scarce languages. In: Proceedings of the First ACM Symposium on Computing for Development. In: ACM DEV '10, Association for Computing Machinery, New York, NY, USA, http://dx.doi.org/10.1145/1926180.1926184.

Ridhawi, I.A., Aloqaily, M., Karray, F., Guizani, M., Debbah, M., 2022. Realizing the tactile internet through intelligent zero touch networks. IEEE Netw. 1–8. http://dx.doi.org/10.1109/MNET.117.2200016.

Saon, G., Chien, J.-T., 2012. Large-vocabulary continuous speech recognition systems: A look at some recent advances. IEEE Signal Process. Mag. 29 (6), 18–33. http://dx.doi.org/10.1109/MSP.2012.2197156.

Sisinni, E., Saifullah, A., Han, S., Jennehag, U., Gidlund, M., 2018. Industrial Internet of Things: Challenges, opportunities, and directions. IEEE Trans. Ind. Inform. 14 (11), 4724–4734. http://dx.doi.org/10.1109/TII.2018.2852491.

Sneha, V., Hardhika, G., Jeeva Priya, K., Gupta, D., 2018. Isolated kannada speech recognition using HTK—A detailed approach. In: Saeed, K., Chaki, N., Pati, B., Bakshi, S., Mohapatra, D.P. (Eds.), Progress in Advanced Computing and Intelligent Engineering. Springer Singapore, Singapore, pp. 185–194. http://dx.doi.org/10.1007/978-981-10-6875-1.

Vučinić, M., Simon, J., Pister, K.S., Richardson, M., 2019. Constrained join protocol (CoJP) for 6TiSCH. Internet Eng. Task Force.

Watanabe, S., Hori, T., Hershey, J.R., 2017. Language independent end-to-end architecture for joint language identification and speech recognition. In: 2017 IEEE Automatic Speech Recognition and Understanding Workshop. ASRU, pp. 265–271. http://dx.doi.org/10.1109/ASRU.2017.8268945.

Weiss, A., Wortmeier, A.-K., Kubicek, B., 2021. Cobots in Industry 4.0: A roadmap for future practice studies on human–robot collaboration. IEEE Trans. Hum.-Mach. Syst. 51 (4), 335–345. http://dx.doi.org/10.1109/THMS.2021.3092684.

Wilhelm, J., Williams, J., Macy, S., 2017. Whitepaper on IoT Onboarding - A Device Manufacturer's Perspective. Kaiser Associates White Paper, https://www.intel.com/content/dam/www/public/us/en/documents/white-papers/kaiser-associates-iot-onboarding-for-device-manufacturers-whitepaper.pdf.

Yan, C., Zhang, G., Ji, X., Zhang, T., Zhang, T., Xu, W., 2021. The feasibility of injecting inaudible voice commands to voice assistants. IEEE Trans. Dependable Secure Comput. 18 (3), 1108–1124. http://dx.doi.org/10.1109/TDSC.2019.2906165.

Zhu, H., Zhang, Q., Gao, P., Qian, X., 2023. Speech-oriented sparse attention denoising for voice user interface toward industry 5.0. IEEE Trans. Ind. Inform. 19 (2), 2151–2160. http://dx.doi.org/10.1109/TII.2022.3206872.