Journal Paper

A Cloud-Based Robot-Assisted Search And Rescue System

Omar Cheikhrouhou

Anis Koubâa*

Anis Zarrad

A Cloud-Based Robot-Assisted Search And Rescue System

Omar Cheikhrouhou, Anis Koubâa*, Anis Zarrad   1

*CISTER Research Centre

Polytechnic Institute of Porto (ISEP P.Porto)

Rua Dr. António Bernardino de Almeida, 431

4200-072 Porto

Portugal

Tel.: +351.22.8340509, Fax: +351.22.8321159

E-mail: o.cheikhrouhou@tu.edu.sa, aska@isep.ipp.pt

https://www.cister-labs.pt

Abstract

The combination of wireless sensor networks (WSNs) and 3D virtual environments opens a new paradigm for their use in natural disaster management applications. It is important to have a realistic virtual environment based on datasets received from WSNs to prepare a backup rescue scenario with the acceptable response time. This paper describes a complete cloud-based system that collects data from wireless sensor nodes deployed in real environments and then builds a 3D environment in near real-time to reflect the incident detected by sensors (fire, gas leaking, etc.). The system 19s purpose is to be used as a training environment for a rescue team to develop various rescue plans before they are applied in real emergency situations. The proposed cloud architecture combines 3D data streaming and sensor data collection to build an efficient network infrastructure that meets the strict network latency requirements for 3D mobile disaster applications. As compared to other existing systems, the proposed system is truly complete. First, it collects data from sensor nodes and then transfers it using an enhanced Routing Protocol for Low-Power and Lossy Networks (RLP). A 3D modular visualizer with a dynamic game engine was also developed in the cloud for near-real-time 3D rendering. This is an advantage for highly-complex rendering algorithms and less powerful devices. An Extensible Markup Language (XML) atomic action concept was used to inject 3D scene modifications into the game engine without stopping or restarting the engine. Finally, a multi-objective multiple traveling salesman problem (AHP-MTSP) algorithm is proposed to generate an efficient rescue plan by assigning robots and multiple unmanned aerial vehicles to disaster target locations, while minimizing a set of predefined objectives that depend on the situation. The results demonstrate that immediate feedback obtained from the reconstructed 3D environment can help to investigate what 13if scenarios, allowing for the preparation of effective rescue plans with an appropriate management effort.

Article

A Cloud Based Disaster Management System

Omar Cheikhrouhou [1,2,*], **Anis Koubaa** [3,4] **and Anis Zarrad** [5,*]

1   College of Information Technology and Computer Science, Taif University, Taif 26571, Saudi Arabia
2   Computer and Embedded Systems Laboratory, Sfax University, Sfax 3029, Tunisia
3   CISTER/INESC-TEC, Polytechnic Institute of Porto, 4200 Porto, Portugal; anis.koubaa@gmail.com
4   Department of Computer Science, Prince Sultan University, Riyadh 12435, Saudi Arabia
5   School of Computer Science, University of Birmingham Dubai, Dubai 73000, UAE
*   Correspondence: cheikhrouhou@gmail.com (O.C.); anis.zarrad@gmail.com (A.Z.)

check for
updates

Abstract: The combination of wireless sensor networks (WSNs) and 3D virtual environments opens a new paradigm for their use in natural disaster management applications. It is important to have a realistic virtual environment based on datasets received from WSNs to prepare a backup rescue scenario with an acceptable response time. This paper describes a complete cloud-based system that collects data from wireless sensor nodes deployed in real environments and then builds a 3D environment in near real-time to reflect the incident detected by sensors (fire, gas leaking, etc.). The system's purpose is to be used as a training environment for a rescue team to develop various rescue plans before they are applied in real emergency situations. The proposed cloud architecture combines 3D data streaming and sensor data collection to build an efficient network infrastructure that meets the strict network latency requirements for 3D mobile disaster applications. As compared to other existing systems, the proposed system is truly complete. First, it collects data from sensor nodes and then transfers it using an enhanced Routing Protocol for Low-Power and Lossy Networks (RLP). A 3D modular visualizer with a dynamic game engine was also developed in the cloud for near-real time 3D rendering. This is an advantage for highly-complex rendering algorithms and less powerful devices. An Extensible Markup Language (XML) atomic action concept was used to inject 3D scene modifications into the game engine without stopping or restarting the engine. Finally, a multi-objective multiple traveling salesman problem (AHP-MTSP) algorithm is proposed to generate an efficient rescue plan by assigning robots and multiple unmanned aerial vehicles to disaster target locations, while minimizing a set of predefined objectives that depend on the situation. The results demonstrate that immediate feedback obtained from the reconstructed 3D environment can help to investigate what–if scenarios, allowing for the preparation of effective rescue plans with an appropriate management effort.

Keywords: RLP; wireless sensor networks; cloud computing; disaster management; smart 3D virtual environment; traveling salesman problem

1. Introduction

The integration of wireless sensor networks (WSNs), mobile robots, and 3D virtual environments has contributed to a wide range of applications, such as disaster management [1], emergency preparedness fire scenarios, sophisticated military training, and search and rescue. In such applications, different team members must collaborate together in order to achieve a specific mission. The synergistic combination of WSNs and 3D graphics technologies results in a near-real-time 3D true-to-life scenario based on sensor data received from the real environment, which can be viewed on a web-based 3D visualization platform.

With the proposed platform, we aim to provide a smart solution by which rescue team supervisors can use real data collected from the disaster area through WSNs to understand the situation and decide upon the necessary workforce, including firefighters, medical staff, and policemen, before moving to the disaster location. In the case of a severe disaster, several autonomous agents, such as ground and/or aerial robots, are remotely sent to the disaster location to intensely investigate the disaster and send more details to the cloud concerning the number and state of victims, the fire spread patterns, etc. An instant 3D virtual environment (3D VE) must then be created to reflect the new conditions based on the observations of the agents/robots and wireless sensors. It is also important to consider the facial avatars' [2] emotions and behavior when dealing with a fire disaster to represent realistic appearance to the system user.

The use of data collected using WSNs to build a near-real-time 3D environment remains a challenge due to the computation time needed for complex rendering algorithms, bandwidth constraints, and signal fading. To this end, cloud architecture has been integrated to address these problems. Cloud computing provides services and resource solutions for disaster management applications.

The capability to transform and modify a 3D collaborative virtual environment (CVE) on the fly without the need to stop its operations is a challenging task, especially when applied in important applications like disaster management programs and military exercises. The contributions of this work are as follows:

-   First, we designed and implemented a routing protocol for WSNs tailored to the requirements of disaster management applications. These requirements include reliability, scalability, efficiency, and latency. We extended the routing protocol for low-power and lossy networks (RPL) by designing a new objective function, Cyber-OF (Cyber-Objective Function), that improves upon the existing objective functions of RPL, which do not take into account the cyber-physical properties of the environment. The Cyber-OF objective function is designed to satisfy a combination of metrics instead of a single metric, as well as to be adaptive to possible changes in data criticality during a disastrous event, such as a fire.
-   Second, we developed a complete 3D platform system to render the 3D environment remotely in the cloud and then stream the scenes to clients over the Internet. To reflect the real physical environment exactly, the resulting 3D environment is generated based on data received from WSNs deployed in the physical environment. Additionally, we integrated a scripting story interpreter component in the game engine to handle and monitor avatar face animation changes. The designers programmed the game engine to distinguish the programming language applied in implementing the game from the game's different scenarios, and to provide flexible and simple systems to users when changes are necessary.
-   Third, we aimed to come up with an efficient rescue plan scenario for the resulting 3D environment; within an acceptable response time to solve the previous drawbacks happened in reality. We formulated the problem as a multi-objective multiple traveling salesman problem (MTSP), wherein we had to assign resources (robots, drones, firefighters, etc.) to the target disaster location while minimizing a set of objectives. A three-phase mechanism based on the analytical hierarchy process (AHP) was proposed. The benefits of AHP usage include effectively assigning weights to objectives based on their level of importance in the system and not just common sense.

Figure 1 shows the flowchart of the proposed system. The challenge lies in how to collect and transfer sensor data in a dynamic disaster environment, and in how to construct a rich and detailed 3D virtual environment on the fly based on received sensor data and wireless network resources, in which bandwidth is highly dynamic and high computation time is required. For this, we can benefit from cloud computing to provide an attractive disaster recovery system, offering a more rapid and efficient rescue plan with dedicated infrastructure. To the best of our knowledge, this is the first proposed complete system that considers both WSNs and a 3D rendering engine using cloud computing.
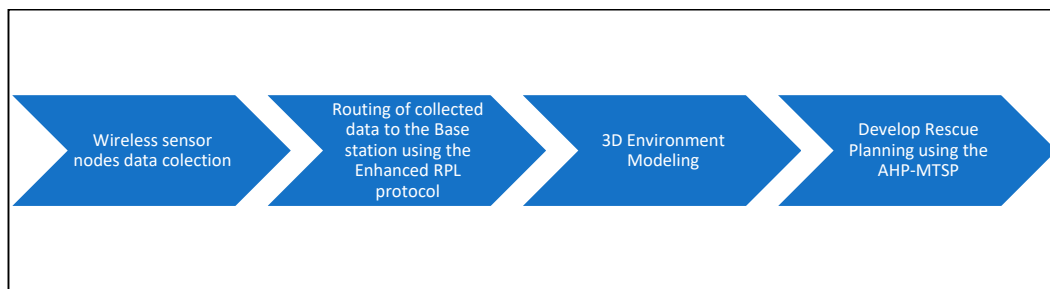
Figure 1. System Flow Chart.

In summary, this paper improves the design of a cyber-physical objective function, tailored for disaster management applications, that addresses the above-mentioned gaps between application requirements and network requirements. Initial simulation and results analysis demonstrate the efficiency of Cyber-OF in handling dynamic changes in the importance of event data and in availing a reasonable performance balance between contradictory performance metrics such as energy and delay. Additionally, a modular game engine was developed to address and solve the following problems: simplifying the 3D scene modifications and facial avatars in a real time fashion, and removing the need for designers and professional programmers. The application of a virtual environment markup language (XVEML) as a dynamic event-based state machine language can ease development processes and fasten them during the runtime. A multi-objective solution was proposed for the formation of an efficient customized rescue plan. All components were integrated with cloud computing to gain increased efficiency and rapid services.

The remainder of this paper is organized as follows. In Section 2, we discuss the main related works that have focused on cloud based systems, WSN communication protocols, game engines, and new real time 3D reconstruction. Section 3 describes the overall proposed system architecture and the modeling scenario used to implement the proposed approach. A description of the enhancement proposed to the RPL routing protocol is provided in Section 4, followed by a description in Section 5 of the 3D near real-time reconstruction approach we used to build and/or extend a virtual environment (VE) application during runtime. In Section 6, we present an MTSP approach to develop an optimized rescue plan. The simulation results and 3D outputs are given in Section 7. Finally, we conclude the paper and give recommendations for future directions of search and rescue applications.

2. Related Works

Several studies have discussed disaster recovery using cloud computing [3–5] in the business context and for enterprises such as storage recovery [3]. The authors in [4,5] discussed business continuity requirements, such as sudden service disruption, that directly impact business objectives and cause significant losses. In contrast, many works have been presented for cloud gaming systems [6–9]. Some authors [8,9] have proposed a 3D graphics streaming solution in which the cloud servers intercept graphics commands, then compress these commands, and finally stream them to clients. In [10], the authors developed an approach by which the cloud was used to render 3D graphics commands into 2D videos, then compress the videos, and stream them to clients. Analysis of the main cloud gaming platforms was discussed in [6], highlighting the uniqueness of their framework design. Clearly, disaster recovery in natural disaster situations is completely different from the business perspective. Users require powerful resources for rapid rendering and instant reaction. Zarrad [11] proposed a traditional cloud computing approach to manage a large amount of data and wide variety of organizations for natural and man-made emergencies.

The most challenging issues in disaster management systems are the communications networks and 3D modeling approaches. For instance, in each disaster situation, there is a need to transfer information between different bodies, such as the disaster management organizations, people, police, and other actors, in an efficient way to provide a quick and appropriate response. Moreover, representing the

disaster situation on the fly in a modern and realistic 3D modeling to execute an emergency response plan for the management of disaster incidents is an important and necessary capability to improve the ability to monitor, manage and coordinate the disaster situation in a particular region.

Many studies have shown the importance of network communication in disaster management systems. The authors in [12] presented a reliable communication mechanism based on 4G. A Mobile Pico is implemented to carry out data transfer between the events station and the accident site. In [13,14] authors proposed communication solution based on IoT technology to minimize the damage and risks involved with disasters. Sakhardande et al. [13] developed a distributed communication solution using IoT technology. The advantage of the proposed approach is that the data transmission between all actors' does not require any existing network infrastructure. A smart city case study is implemented to test the effectiveness of the protocol. In [15], the authors proposed a communication protocol to guarantee continuous connectivity between all players participating in the rescue activity. WiFi direct technology was used to overcome the limitations of the ad hoc networks. The system is specifically developed to offer a stable communication between victims and rescuers. Also, the authors used the WiFi legacy to optimize inter-group communication between different WiFi direct groups.

The Sherpa project [16,17], which aims to improve the rescuing activities through the smart collaboration between humans and robots, has studied the impact of different network technologies used for cloud robotics and concluded that the satellite technology best suit the search and rescue scenario. Moreover, the authors in [17] proposed a new architecture called Dew robotics inspired by the concepts of Dew computing. The proposed architecture leverages the collaboration between robots to fit the search and rescue applications requirements.

Today, research concerning the 3D VE tends to lean towards easy and fast runtime without the need for programming skills. VEs are used in several contexts, such as movies, mobile games, medical and educational visualization, and architectural visualization. The most common solutions used in literature are the general game engines: the Unity engine [18], the Unreal engine [19,20], the Gamebryo engine [21], the CryEngine [22], and the Software's Source engine [23]. Such engines are limited to specific tasks however, and their characteristics are paired with proposed game features. Hence, any change or extensions that take up a new characteristic in the system needs a game engine reboot. Such applications show great potential provided they address a particular application that does not need 3D content changes during runtime. Furthermore, only full-time programmers can change the VEs found within games due to their complexity. Deciding on an adequate game engine is dependent on the aim, platform, and speed at which changes may be needed. The formation of a 3D environment is based on 3D geometric modeling and interactions with light. The process of synthesizing an image by simulating light behavior is called rendering [24].

In building information modelling [25], a significant chunk of graphic designers and developers are necessary to stand for human behavior within a real-time game environment to accommodate a fire evacuation in order to perform effective information communication between the avatars and building. Furthermore, numerous studies have discarded animation concepts and avatars' behavior. The authors in [26] introduced a fire training simulator to enable students to experience a practical fire set-up and gauge different rescue plans in a graphic surrounding. The authors completely ignore mock-ups of human behavior in the case studies that have been implemented, which affects the validity and true-to-life idea of the VE. Chen et al. [27] provide complex instantaneous facial recognition through 3D shape regression. The algorithm in the animation utilizes training data produced from 2D facial images, and the accuracy of the system improves with more captured image and training data. This characteristic requires the process of massive data, which may lead to system failure in real time.

Telmo et al. [28] described a 3D VE meant to improve the understanding of airport emergency measures, whereby each user has a different role in specific emergencies. The concept draws from replication and does not mirror a realistic context. The authors propose a solution in [29] for decision makers and specialists to help them better comprehend, evaluate, and forecast natural disasters to

mitigate damage and to rescue lives. One significant factor not considered is the behavior dynamism within the disaster. System disruption is necessary to assume new scenarios.

Several methods have been put forward within the literature to concentrate on extensibility and 3D alterations. The Bamboo system [30] employed a microkernel-based design to distinguish the system components from the core to add, remove, and change throughout runtime. Regrettably, this highly-approved method invites a lot of complications, principally in terms of helping interaction between elements written in diverse languages. Furthermore, extensibility needs an intensive understanding of programming language. Oliviera et al. [31] developed a java adaptive dynamic environment (JADE) drawn from Java designs, which have a lightweight cross platform core that allows system evolution throughout runtime. However, the acceptance of JADE does not present effective solutions to problems for expanding CVE systems. Boukerch et al. [32] developed a new script language to extend virtual environment during the run time. The proposed approach ignores the avatar face animations and only focuses on the game scenario.

Regarding description languages for games, they focus on specific categories of applications (e.g., [33–36]). Game architectures are highly paired with the proposed programming language, without support for hierarchical ideas. Therefore, producing a playable game is a drawn-out procedure not necessarily designed principally for human communications. The virtual environment markup language (VEML) [37,38] was supported by non-linear story ideas defined by Szilas [39] to construct and/or expand CVE systems. This model permits story progress simultaneously with the simulation, and hence executed separately from the 3D settings programming. In [37], the authors modeled an atomic repetitive simulation in a rejoinder to similar procedures. For instance, the store that Real Madrid FC has for virtual shopping may run numerous sales all-year round. To manipulate this circumstance, the VE regulator will submit a script file VEML to participating customers every time a sale occasion is set. Thus, by conveying numerous descriptions, the files repeatedly affect network bandwidth. Furthermore, changes within the script file should be made manually. This can result in unexpected atomic simulation and loss of real application appearances.

Numerous systems have embraced script language technology to conceive body and facial gestures within animations meant for avatars. The virtual human markup language (VHML) [39] is an autonomous XML-based language utilized in MPEG-4 systems that include a markup language devoted to body animation. Perlin et al. [40] explained an IMPROV system utilizing high-level script languages for producing real-time behavior that possesses non-repetitive gestures, whereas Arafa et al. [41] explained an avatar markup language built on XML that combines text-to-speech and facial and body animations, in a singular approach with sufficient synchronization. Table 1 summarizes the main differences between the mentioned systems.

Table 1. 3D modeling Systems Comparison.

Headings	Characteristics	Systems
Change/extend the VE	Manual Approach	[28–31,37]
	Automated Approach	[18,23,26,27,32,38]
	Near Real -time rendering	[32,37,38]
	Need programming skills	[18,22,23,25,29–31]
	System interruption	[20–23,29,30]
User features	Realistic appearance	[18,23,31]
	Include body and facial animation	[27,30,39–41]
	Rescue Plan	[26,28]
	Efficient Network protocol	[13,14,26,32]

The primary interest in the proposed effort is the intricacy of execution necessary for extending VE with a realistic avatar facial animation. In addition, supportive architectures are characteristically designed to manage only avatar behaviors. Consequently, it is hard to borrow a relevant code from one system and adapt it to another system. We present a complete system with many operational

advantages, including near future VE extension without system interruption and with minimum programming effort, rescue plan, realistic appearance, and cloud architecture to alleviate network latency which is important for such systems.

3. Proposed System Overview

Our research aimed to design and implement a complete system to help rescue teams and first responders to visualize, monitor, and plan interventions in a 3D environment. The proposed system combines various technologies, including WSN, routing protocols, cloud computing, 3D rendering approaches, and multi-objective optimization (Figure 1).

The proposed disaster management system work as follows. First, a set of wireless sensors are deployed in the region under control. Upon detecting an incident such as a fire disaster or a crowded zone, a protocol for gathering data, such as location and severity, is set. The data are forwarded to the central station thanks to a routing protocol. Once data are received on the cloud side, a near real-time rendering approach starts in the cloud to construct a 3D environment closely reflecting reality and is then transferred to a user's machine (firefighter, manager, etc.). After that, a rescue plan is created using the AHP-MTSP solution to determine the optimized plan, including robots, firefighters, police, etc., for the different objectives.

A 3D representation of the current situation is constructive for police, military, and medical personnel, allowing them to react with suitable management effort when they reach the incident site. To construct an efficient and accurate plan, it is essential to have a real-time realistic 3D virtual environment that precisely reflects the physical environment. Any disaster events must be communicated and displayed after rendering using the pre-built geometry approach (GRB) [42]. In GRB, the scene contains many geometrically-defined surfaces with different properties, such as color, transparency, and lighting conditions. In contrast, image-based rendering (IRB) [43,44] depends on a scene description for the plenoptic function [45]. Scene photographs are needed from different viewpoints to create the 3D environment. We favor GRB in our solution because IRB requires a high-quality photo from the source, which may be unclear/blurry due to the presence of smoke, water leakage, etc., in the original environment. Additionally, the use of the plenoptic function requires too much data to handle.

Generally, the rescue implementation depends on several parameters such as the environment condition, the equipment capacity, and the victim's situation, etc. In this work, we use the facial behavior of avatars to imitate the real situation described by the sensor data during the runtime. Figure 2 describes the proposed system architecture. Three entities are identified: (1) real environment; (2) the corresponding 3D virtual environment-user side; and (3) the cloud computing.
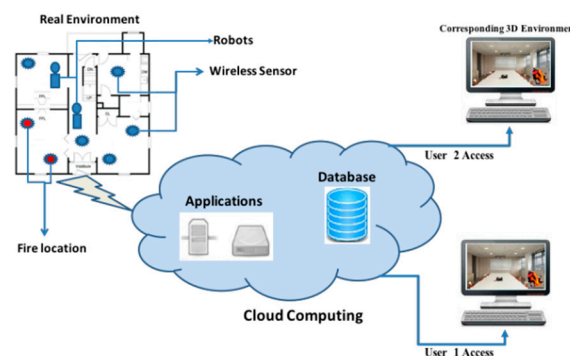

Figure 2. The overall system architecture.

To guarantee quick access and modeling, we proposed to preload the initial 3D representation of the VE into the cloud. Moreover, a modular game engine is implemented in the cloud to manage the VE modifications and extensions, including the avatars' facial animations.

The implementation of the main game engine components in the cloud offers the user the freedom to use different kinds of devices (desktop, mobile device, etc.) when running the system. Additionally, users are allowed to interact with applications without installing or configuring a 3D development environment, regardless of the amount of storage and memory required.

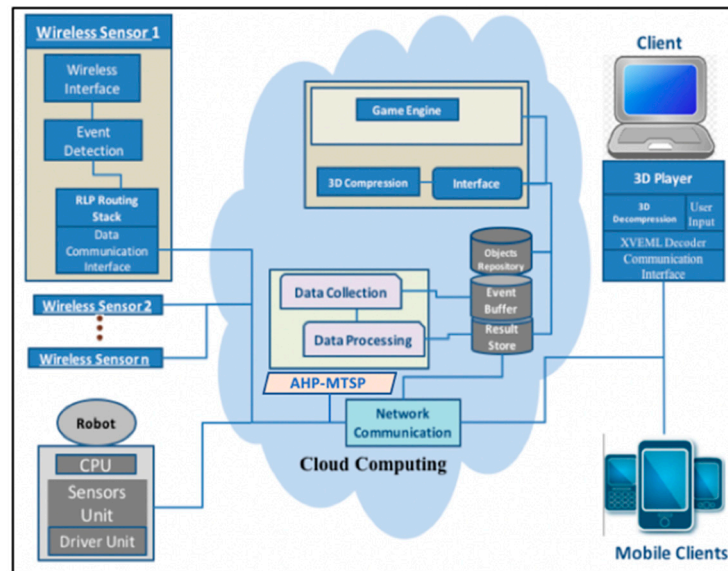The detailed cloud architecture used for our system is shown in Figure 3.

Figure 3. Detailed cloud architecture.

When sensors identify a fire within the actual environment, the system collects numerous events from the real surroundings and sends them back to a central database. RPL [46,47] is the typical de-facto standard protocol currently used for routing data from sensors to an Internet gateway. Security protocols, such as [48,49], are applied at the sensor layer to protect integrity and authentication of messages. A rendering protocol within the cloud infrastructure is utilized to image a comprehensive live depiction of the accident site in a 3D fashion.

To address the fire disaster, numerous human resources are necessary such as ambulances, firefighters, policemen, etc. The emergency service (ES) gets an emergency call, possibly from a witness reporting the developing disaster somewhere within the city. The ES responds by informing the closest fire station that possesses sufficient resources. Furthermore, the ES may also inform the ambulance service and the police if there are injuries. The players here include the witnesses, the firefighters, the policemen, and the ambulance staff who handle the fire and the injured. Depending on fire density, two fire stations may be involved in the rescue efforts. The story structure can be divided into many chapters, from "Chapter 1: Fire scene description and details", to "Chapter 2: Fire stations and police preparation", to "Chapter n: Extinguish the fire".

We designed each chapter independently utilizing the state machine idea. Every phase has a title, which helps in determining the state of the game scenario at any moment. Some phases are prevented because a specific external event is required. Participating avatars dynamically change their facial behaviors (frightened, happy, etc.) to reflect the surrounding situation in the VE depending on the current conditions, such as fire intensity, weather condition, and the number of injuries.

It is important to model the avatars facial animation [50] based on the data received from sensors networks to increase the realism and offer a realistic simulation and address the challenges of effective training environment. In the fire disaster management system, face animation is the main instrument for non-verbal communication, and for defining a figure's mood and personality avatars.

4. Extending the RPL Protocol with Cyber-Physical Objective Function

The idea of our proposed routing protocol is to extend RPL by adding the cyber-physical objective function (Cyber-OF) to improve network performance. The IETF ROLL working group has proposed the IPv6 routing protocol for lossy and low power systems (RPL) [46] as a standard routing protocol for low power and lossy networks (LLNs). Although RPL can be executed on top of any MAC layer, it is mainly designed for low-power IEEE 802.15.4 and its variants [51]. The authors in [52] proposed a protocol to cover the diverse requirements imposed by different applications running on static or dynamic constraints that are suitable for RPL. Primary link and node routing metrics are used to report the number of traversed nodes along the path. The path length is expected to be minimized and network overhead is reduced. In [53], the authors addressed a routing metric, called TXPFI to minimize the node energy cost and deliver energy-efficient routing in RPL. Clearly, the topological distance between source and destination nodes and the expected number of frame transmissions have an impact on the TXPFI metric.

The protocol specifies some basic mechanisms for point-to-point traffic essentially transmitting such point-to-point traffic between two sensors via the controller [54], and builds a destination oriented directed acyclic graph (DODAG) using an objective function and a set of metrics/constraints. Nodes in the network operate autonomously and manage several objective functions.

In the literature, several objectives functions and extensions were proposed for RPL to improve its performance to specific use cases [55,56]. In RLP [57], routing metrics and objectives functions (OF) are responsible for the construction of directed acyclic graph (DAG). In the IETF [58] standard, the routing metrics are not imposed. Thus, the parent selection is implementation-specific, which makes it an open research issue worth investigating.

Considering the RPL OFs mentioned, it appears that they do not take into account environmental cyber-physical properties. Relying on a single measure or more in a significant condition (disaster, fire, hurricane, etc.) may consequently be ineffective and not meet the demands of the smart cities' application profile. For instance, applying the hop-count metric in emergencies may not be the speediest method to broadcast the network. Furthermore, the use of a single static OF may not meet the requirements of similar systems with different levels of event importance.

Therefore, we modeled the Cyber-OF to facilitate its adjustment to the complex tree configuration concurrently in relation to the cyber-physical characteristics of the surroundings depending on the event importance. In fact, for ordinary data packets, the aim is to increase network life span; hence, the OF improves energy expenditure. In the case of a serious event, the system adapts its topology to reduce end-to-end delays. Consequently, we assumed an adjective behavior by taking into account two routing OFs, each utilizing a specific metric of interest, namely:

- **Energy metric:** this measure represents energy expenditure within an RPL node. With this metric, it is feasible to prolong the system life-span. It is necessary to think about this measure for systems that have energy-efficiency concerns.
- **End-to-end delay:** this is the typical time it takes for a packet to move from node to sink. This measure should be reduced for systems requiring real-time assurances.

In normal mode, the Cyber-OF uses the energy metric in the attempt to reduce energy consumption. This means that RPL will choose the routes that minimize energy consumption. For the case of a critical event (i.e., alarm due to a fire), Cyber-OF switches to the delay metric (latency-OF) so that the routing protocol emphasis more the route that will deliver faster the critical messages. We use the latency-OF mode in case of a detected fire for example, while energy-OF mode is used during normal situations.

Figure 4 sums up the operations of Cyber-OF. First, the OF based on the energy metric is activated. RPL routing will consider this OF until a critical event (e.g., fire) is detected. This means that every router chooses a parent that will optimize the energy metric in the case of normal operation. When a critical event is detected in a packet en route, RPL automatically switches to the latency OF, which allows it to find reliable minimum-latency paths. The measure utilized by Cyber-OF is decided by

the metrics found within the DIO metric container. Thus, the route from the node to the gateway router will change so as to reduce the end-to-end delay, rather than focusing on energy, considering the criticality of the event to be transported. Thus, in ordinary circumstances, the OF increases the system lifespan. In emergencies where serious events are discovered, a node required in advancing this process to the border router must utilize a new OF that minimizes end-to-end delays.
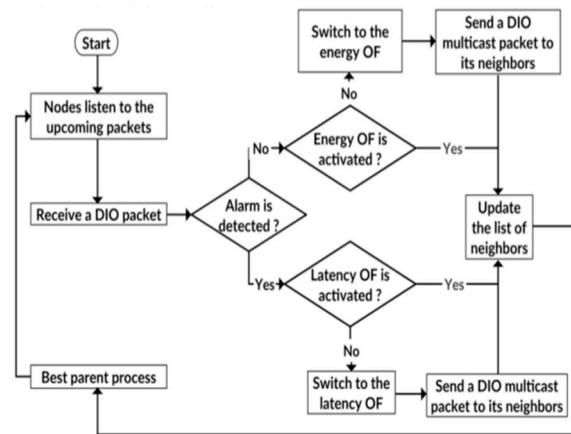

Figure 4. Flowchart of the Cyber-Physical Objective Function (Cyber-OF).

It is clear that the design of Cyber-OF allows for consideration of the cyber-physical properties of the environment, enabling network topologies to adapt to these conditions. In fact, when a high temperature is detected, the routers understand that this event is critical and must switch the routing strategy to meet the application real-time requirements.

In our design of Cyber-OF, we considered the energy metric for normal situations and the delay metric for critical situations. As these two metrics are only representative, other metrics that would better represent each situation can be considered instead.

In the fire disaster context, we can identify several unexpected behaviors. Early detection of fires in cities and urban areas is essential in order to prevent more losses and the spread of fire.

All of these emergency situations require early detection to provide an alarm in real-time. This is the main reason why we proposed using end-to-end delay as a routing metric in the case of a critical event.

From the implementation perspective, the end-to-end delay metric represents the sum of link latencies for the path of RPL routers traversed by the packet. Thus, every node in the network should maintain a link latency measure to all its neighbor nodes, which will be used to select the next router in the path for every packet. The next-hop selected is the RPL router that has the lowest latency. This strategy will be used the latency-OF method that improves the real-time performance in critical situations.

Regarding the energy metric, every RPL router estimates its remaining battery level. The next RPL router selected in the path is the router that has the highest remaining energy. This strategy will be used the energy-OF method that improves the network lifetime performance in normal situations.

5. Game Engine Architecture and Visualization Approach

Our objective is to change and/or extend the VE during the run, based on the data received from sensor networks. The proposed approach has an advantage because most changes can be applied with local action only, simplifying the job by increasing engine speed without a restart. Currently, most applications need wide-ranging programming activities and partnerships with diverse professionals to manage extended action. The dual designing of our VE system permits us to quickly and easily block alterations in the VE simulation throughout modeling.

If modifying the simulation scenario is necessary through the addition or subtraction of a chapter from the overall scenario, it is necessary to alter the matching state machine to mirror the new alterations and automatically produce the new SCXML file (Figure 5). Therefore, when the simulation scenario varies, it must be followed by instant modification. Thus, a new IXML file must be generated as well. Conversely, any changes in instance designs do not need an alteration in the class model. Such a technique makes code that would otherwise be complex and difficult to modify become simple and comprehensible to novices. In addition, the difficulties of developing SCXML and IXML script codes for deployment in large-scale VE application can totally avoid the limits in traditional development methods. The resultant files are conveyed to all users taking part in the VE throughout runtime, without disrupting the system.
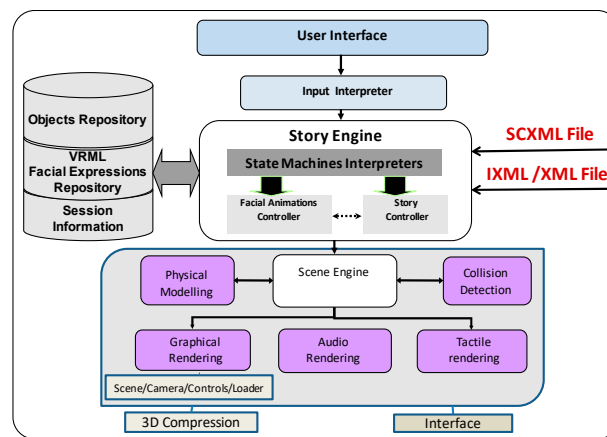

Figure 5. Detailed game engine architecture.

Furthermore, it is necessary to handle existing avatars in the missing chapter to ensure the application produced reflects a natural and realistic way. Two methods can work: (1) handling the singleton manually through a direct change of the avatar characteristics within the objects' repository database, or, (2) produce a certain state machine known as a migration state machine that eliminates avatars from the ghost chapter B and any other chapter.

The architecture put forward for the game engine is particularly modeled to connect the extension of the 3D VE during runtime and system operational interruption. We utilized modular architecture to enable the incorporation and management of new elements and offer better qualities to the system produced. On the customer side of the system, the design and animation engines are based on the web and animation outcomes are viewable locally within the end-user's browser.

The framework includes seven key components:

- User interface GUI and VRML viewer. The GUI (graphical user interface) of the application that transmits user instructions, such as altering scene content and avatar animations. The VRML [59] viewer pictures the VE contents and VRML avatars' face animations.
- The game engine is the primary element in this framework. It replicates the simulation set-up within the proposed application and avatar facial performance through the state machine concept. This component fundamentally consists of:

  ○ State machines interpreter: contributes its input once drawing from an SCXML file which provides the scenario story. It also replicates the states and transitions and handles all possible shifts via the distinct states. There are numerous instances of the IXML file and every one represents an example of a particular scenario.
  ○ Story controller: This element transforms SCXML files into instructions for the story engine.
  ○ Face animation controller/validate: This element handles IXNML files while undertaking key functions in the implementation of facial animations. Visage Software Development

Kit is incorporated into this element to authenticate animations produced and check expressions of conflict.

- The physics engine: This element consists of physical replication and collision recognition. It handles calculation of physics aspects such as object deformation due to gravitational forces. It also manages internal contradictions in facial animation, for instance when designers cannot combine two animations (e.g., happy face and angry face).
- The VRML animation and facial expression record holds facial descriptions. Each facial aspect is executed as a VRML transform node, which explains the shape and other characteristics. This also holds all the new facial animations produced during the simulation process.
- The scene engine controls illustrating a 3D world and portraying it on-screen. In addition, the scene engine determines which view is observable by the player on-screen. Three sub-elements (audio, graphical and tactical rendering,) are related to the scene engine. The audio rendering element produces sounds as the game runs; tactical rendering provides feedback support for accurate positioning and screen touch while the graphical rendering illustrates the 3D world.
- The 3D compression module is important for fast storage, access, and transmission of data to connected clients within bandwidth limitations.
- The interface is used to plan how communications will be conducted among all parties.

Regarding the server side, we implemented three main components: session manager, synch manager, and message controller. The session manager produces sessions for groups of users who share similar interests in the VE. The message regulator exchanges and manages messages between the main server and the users while the synch manager controls synchronization between all users taking part in the same session to preserve the same view within the shared VE that is 3D. When the server calls for new alterations, such as amending the list of expressions for a specific avatar, an XVEML file is produced in accordance with the change and inserted into the client's side. The validator then decodes the inserted file and transmits it to the controller of facial animations for implementation. The game engine then verifies if it can trigger the expression. If so, this action triggers events that it sends to the scene engine for display within the 3D space, then all users working the particular session get notifications of the change.

Our main focus in this work was to develop a 3D environment to help first responders and save lives, rather than looking to produce a 3D environment with high-quality visuals. The illumination of the fire scene was therefore simulated by applying a shading model [44] that utilizes a simple method that subjects color intensities to linear interpolations computed at the points of the rendered polygon across the insides of the polygon. This mechanism was implemented in the graphical rending module inside the scene engine. For easy facial animation, we adapted the rigging process [60]. We produced a "custom controller" to enable every expression to attain genuine avatar emotion within the face animation controller module. Execution of the firefighter set-up and the recorded avatar expressions when handling fire, shown in Figure 9, where frightened and happy expressions are displayed in screenshots. Expressions were performed within the 3D space in conformity to the circumstances surrounding the avatar. When the density of fire was high, the avatar displayed fear. However, when the fire was extinguished in successful operation, the avatar displayed happiness.

6. Optimized Rescue Plan Development

Based on the wireless sensor network, our proposed system determines the locations of the targets that need to be visited by the robots, rescue team, and fighters. These targets can be victims' locations, fight locations or any zone that need urgent intervention. Moreover, an optimal intervention needs to be planned and executed in order to save victims and minimize loss.

Therefore, in this section, we propose AHP-MTSP, a multi-objective optimization approach that helps to develop a rescue plan while minimizing several metrics of interest. Once the data was received from different sensors deployed in the environment. The system should prepare a training rescue plan

to deal with the fire situation. The aim is to find an effective target locations assignment for the set of robots, such that each target is visited by only one robot at an optimal cost. Optimal cost includes the minimization of the total travelled distance that all the robots cover, minimization of the maximum tour length for all the robots, minimization of the overall mission time, minimization of the energy consumption, and balancing in the targets allotment.

As mentioned in the architectures of Figures 1 and 2, the rescue team of robots and drones are connected to the cloud server through the Internet using cloud-based management system for controlling and monitoring robots, such as the Dronemap Planner system [61–63]. This enables the robots and drones to feed the cloud server with up to date information about the drones, including their locations, their internal status (battery level, altitudes, speed, accelerations,) which are used by the centralized agent server to dynamically develop the rescue plan using our proposed AHP-MTSP. The states of the robots and drones are exchanged using the MAVLink protocol [64] or the ROSLink protocol [65], which are messaging protocols that were designed to exchange messages between robots/drones and ground stations.

It is imperative to note that multi-objective optimization problems take into account various conflicting objectives. This connotes that an effective resolution for one objective could be an ineffective resolution for another objective. Specifically, traditional optimization techniques do not offer answers that are effective for all objectives within a problem in focus. A multi-objective optimization problem can come about via a mathematical model characterized by a collection of p objective functions that should undergo simultaneous minimization or maximization. Officially, a multi-objective problem is definable as:

$$
\begin{aligned}
&\min/\max f_1(X)\\
&\min/\max f_2(X)\\
&\min/\max f_3(X)\\
&\qquad\cdots\\
&\min/\max f_p(X)
\end{aligned}
\tag{1}
$$

where X represents the decision space.

For the multi-objective MTSP, we considered a set of m robots $\{R_1, \ldots, R_i\}$ which are initially located at depots $\{T_1, \ldots, T_i\}$ The m robots should visit n targets $\{T_{m+1} \cdots T_{m+n}\}$, where each robot should visit each target only once. Each robot Ri begins at its depot Ti, then visits each of the *ni* designated targets $\{T_{i1} \cdots T_{ini}\}$ sequentially then returns to its depot. The cost of travelling from target T_i to T_j is represented as $C(T_i;T_j)$, where cost may be Euclidean space, energy consumption, time spent, or anything else. Furthermore, it is possible to classify the objective functions into three groups. The first group focus on reducing the total costs for all the robots, such as reducing the total traveled distance or reducing the total energy consumption. This group of objective functions can be defined as:

$$
\text{minimize} \sum_{k=1}^{m} \sum_{i=1}^{n+m} \sum_{j=1}^{n+m} x_{ijk} C\left(T_i, T_j\right)
\tag{2}
$$

subject to:

$$
\sum_{k=1}^{m} \sum_{i=1}^{n+m} x_{ijk} = 1; \ \forall j = 1, \ldots, n+m
\tag{3}
$$

$$
\sum_{k=1}^{m} \sum_{j=1}^{n+m} x_{ijk} = 1; \ \forall i = 1, \ldots, n+m
\tag{4}
$$

$$
\sum_{i=1}^{n+m} x_{ijk} = 1; \ \forall k = 1, \ldots, m
\tag{5}
$$

$$\sum_{j=1}^{n+m} x_{ijk} = 1; \; \forall k = 1, \ldots, m \tag{6}$$

$$x_{ijk} \in \{0,1\}; \; \forall i, j = 1, \ldots, n+m; and \; k = 1, \ldots, m \tag{7}$$

Equations (3) and (4) guarantee that each robot visits each node only once. Equations (5) and (6) ensure that each robot begins and returns to its specific depot. Finally, restrictions brought about by Equation (7) show that the decision variables are twofold.

The second group of objective function covers those that reduce the maximum cost for all robots, such as reducing the maximum mission time or reducing the maximum tour length. It is possible to model this group of objective functions as:

$$\text{minimize} \max_{k \in 1 \ldots m} \left(\sum_{i=1}^{n+m} \sum_{j=1}^{n+m} x_{ijk} C(T_i, T_j) \right) \tag{8}$$

Subject to the same restrictions expressed within Equations (3)–(7).

The third group of objective functions relates to balancing the workload between the robots, balancing the lengths of the tour, balancing mission time, as well as the number of targets allocated. It is possible to model this group of objective functions as:

$$\begin{aligned} &\textit{minimize} \; \sum_{k=1}^{m} \left| C_k - C_{avg} \right| \\ C_K &= \left(\sum_{i=1}^{n+m} \sum_{j=1}^{n+m} x_{ijk} C(T_i, T_j) \right), \; K \in [1, m] \\ C_{avg} &= \frac{\sum_{k=1}^{m} \sum_{i=1}^{n+m} \sum_{j=1}^{n+m} x_{ijk} C(T_i, T_j)}{m} \end{aligned} \tag{9}$$

where C_k is the tour cost for the robot k (cost can be distance, energy, time, etc.). C_{avg} signifies the average cost of all tours. In our application model, it was assumed each robot possessed global knowledge of the targets within the disaster site requires a visit as well as the fire location. In addition, each robot is able to compute an estimation cost between its present location and the target to be visited, such as Euclidean distance, energy and time spent, subject to circumstance.

The proposed AHP-MTSP solution put forth is weighted, meaning that we assigned the objectives to be optimized with different weights. We defined the global cost as the sum of the weighted costs of the different objective functions under consideration. Formally, let $w = (w_1 \cdots w_p)$ be a weight vector, where $0 < w_i < 1 \; \forall i = 1, \ldots p$ and $\sum_{i=1}^{p} w_i = 1$. The problem then consists of minimizing the function:

$$\begin{aligned} \text{minimize } g(x|W) &= \sum_{i=1}^{p} w_i f_i(x) \\ \text{subject to} &: \; x \in \Omega \end{aligned} \tag{10}$$

where Ω is the decision space and $f_i()$ is an objective function.

To construct the weight vector W, we used the AHP technique [66]. AHP is a decision-making approach with multi-criteria that can be used to solve complex decision problems [66]. The relevant data was obtained by means of a series of pairwise comparisons, which are used in terms of each individual decision criterion to obtain the weights of the targets and the relative performance measures of the alternatives.

Figure 6 shows an overview of the proposed AHP-MTSP approach. First, the user defines his preference and priority of each objective function in a comparison matrix. This comparison matrix is then introduced to AHP to generate a weight vector. This weight vector is then used to compute the global cost, following Equation (10).
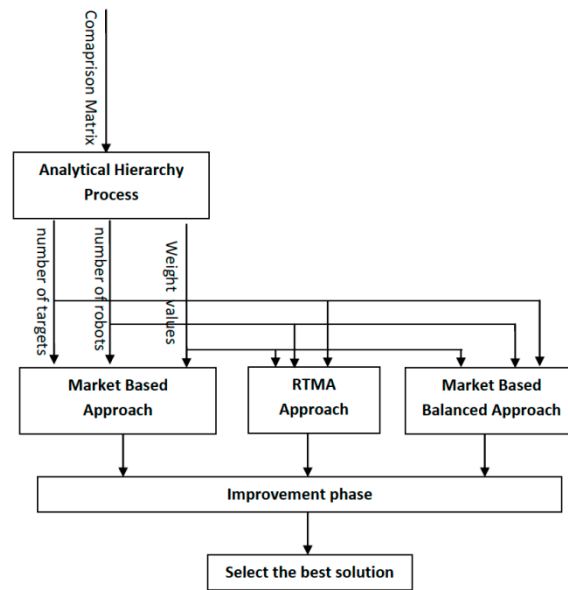
Figure 6. Flowchart showing the proposed AHP-MTSP approach.

Three different approaches were then executed, as shown in Figure 6. Finally, the best solutions for these three approaches were selected.

With respect to the disaster management application, three objective functions came under consideration: total travel distance (TTD), maximum tour (MT), and deviation rates for the tour lengths (DR). Certainly, in systems such as fire disasters, the most critical issue is mission time, which is directly proportional to MT. Furthermore, minimizing the TTD and DR permits the minimization of total energy consumption and balancing of vehicle workloads. We then considered the following comparison matrix:

$$A_{i,j} = \begin{pmatrix} & \text{TTD} & \text{MT} & \text{DR} \\ \text{TTD} & 1 & 1/2 & 1/3 \\ \text{MT} & 2 & 1 & 1/2 \\ \text{DR} & 3 & 2 & 1 \end{pmatrix} \quad (11)$$

This matrix shows that MT has double the priority of the TTD, and the DR has triple the priority of the TTD and double the priority of the MT.

The proposed AHP-MTSP algorithm was designed to select the best result from three approaches namely the market-based approach, the RTMA approach and the balanced. However, for lack of space, we present here only the market-based approach and we refer the reader to paper [67–69] for more details.

The market-based approach can leverage the use of the global knowledge of the system in the cloud server to dynamically adapt the allocation of tasks to robots based on the dynamic changes of an emergency situation in real-time through dynamic auction-bidding transactions for every new event when it occurs. In fact, in market-based, the coordinator of the cloud server will receive events in real-time from sensors spread in the environment and dynamically generates new auctions for the rescuer team (robots), which will bid on these tasks using the AHP-MTSP algorithm, and accordingly adapts the rescue plan in real-time.

Following the market-based approach, existing robots compete to reach currently available targets. Therefore, each robot selects the best target, which provides the least local cost. The weighted sum of the OF costs is defined as local cost for a specific robot. After the target selection, the robots relay their bid to a central machine, with each bid containing the specific chosen target and corresponding costs for each OF. Once it receives the different bids, the central machine calculates global costs in each bid,

then allocates the most appropriate target to the corresponding robot. The target with the least global cost is the best target.

To illustrate the market-based approach, we considered a scenario with two robots, six targets, and two OFs (TTD and MT), with the weight vector W = {0.66, 0.33} (TTD has a priority two times greater than that of MT; Figure 7). More details about the market-based approach are described in our previous paper [67].
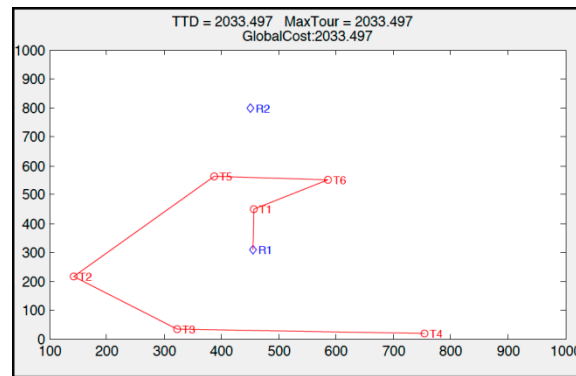

Figure 7. Market-based approach using fire scenario with two robots and six tasks.

First, R1 chooses T1 and R2 chooses T5. Since global cost when the central machine assigns T1 to R1 is lower than when it assigns T5 to R2, the allocation goes to R1. This process goes on until the central machine allocates all robots their targets, as evident in Figure 7, and the TTD cost is promoted more by the central machine than the MT.

7. Simulation Results and Scenario Modeling

This section discusses the performance evaluation of the different proposed components of our complete search and rescue system.

7.1. Cyber-OF Performance Evaluation Results

The study implemented a Cyber-OF in ContikiOS while utilizing the Cooja simulator to gauge overall performance. The study examined the output from three implemented OFs:

(1)　A latency-based OF, which considers the latency.
(2)　An energy-based OF, which considers the energy.
(3)　The Cyber-OF, which considers an adaptive behavior, described in Figure 8.

Simulation experiments were conducted utilizing a 2D-grid surface on a network topology that has 10, 20, or 30 sensors. The DAG architecture consisted of one border router that acts as the data sink while the rest acted as UDP servers producing data. The depth of the resulting DAG is equivalent to 6. For the simulation, a fire alarm is activated by one node, which would then transmit a unicast packet containing the alarm to the sink (Figure 8a). In response, the sink transmits the alarm to all the nodes within the DAG to adjust to the topology consequently (Figure 8b).

Next, the study presents the outcomes of the assessment of Cyber-OF, which also examines the influence of the following parameters:

- 　End-to-end delays: Duration from the beginning of packet transmission to its reception at the DAG root.
- 　Network lifetime: The time required for the initial sensor to expend all its energy.

Figure 9 compares the average end-to-end delay within the three OFs under investigation: the energy-OF, the latency-OF, and Cyber-OF. It is apparent they displayed similar delay values when the

network consists of less than 20 nodes. However, a slight difference was noted in the energy-OF, which permitted a higher average delay. This was anticipated because the energy-OF only extends network lifetime and the selection of the best performing parent is subject only to the remaining energy within the node.
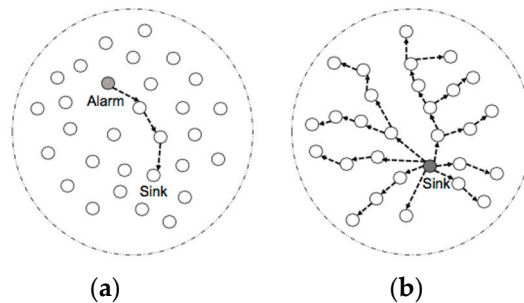

Figure 8. A multicast packet of the whole network simulation. (**a**) Unicast alarm packet sent to sink; (**b**) Sink forward alarm packet by multicast.
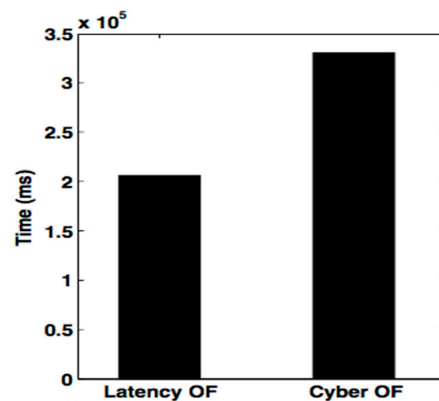

Figure 9. Comparison of the network lifetime of a latency-based objective function (latency-OF) and the cyber-physical objective function (Cyber-OF).

The study also noted that when the network consists of more than 20 nodes, Cyber-OF underwent lower average delay values than the energy-OF. This verifies the predisposition of the Cyber-OF to diminish the holdup when serious events (a fire alarm in this model) are identified. Figure 10 characterizes the energy that the latency-OF and Cyber-OF consume, signifying how Cyber-OF can reduce energy consumption and maximize network lifespan better than the latency-OF during a five-minute demonstration.
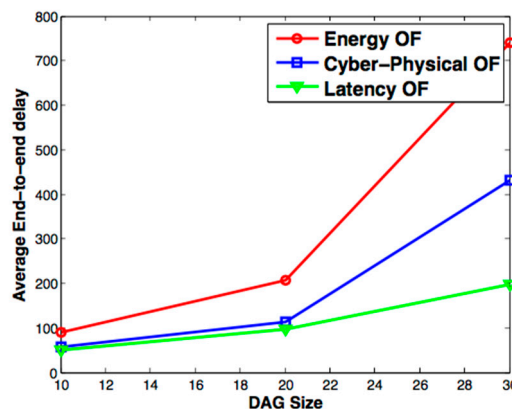

Figure 10. Average end-to-end delay of an energy-based objective function (energy-OF), a latency-based objective function (latency-OF), and the cyber-physical objective function (Cyber-OF), in relation to directed acrylic graph (DAG) size.

7.2. Evaluation Results for the Rescue Plan

In this section, we present the performance evaluation of the proposed market-based AHP-MTSP solution. We considered three objective functions, namely: TTD, MT, and DR. The considered global cost was therefore:

$$Globalcost = w_1 \sum_{k=1}^{m} C_k + w_2 \max_{k\in1...m}(C_k) + w_3 \sum_{k=1}^{m}|C_k - C_{avg}| \qquad (12)$$

where C_k, C_{avg} are as previously defined (Section 6) and $C(T_i;T_j)$—Euclidian distance between targets T_i and T_j.

The following results were obtained using three targets and the number of robots varies in nr = [35101520]. Furthermore, a target location is randomly selected from a 1000 × 1000 space. For each arrangement of robot and target numbers, the study randomly generated 30 possible set ups then mapped the average of the results from those 30 set ups.

In the multi-objective approach, a weight vector of W = {0.26; 0.19; 0.54} came under consideration. In addition, to examine a single objective function, either TTD, MT, or DR, the following weight vectors came under examination: W = {1; 0; 0}, W = {0; 1; 0} and W = {0; 0; 1}, respectively. Figure 11a–c displays the contrasting outcomes of the proposed multi-objective method to the mono-objective approach, in which only TTD, MT, or DR, came under consideration, respectively. With respect to the global cost considering the three objective functions, it is evident from the outcomes that the multi-objective method outperforms the mono-objective one and provides a minimal global cost (presented by the curve in black color with circle in Figure 11).
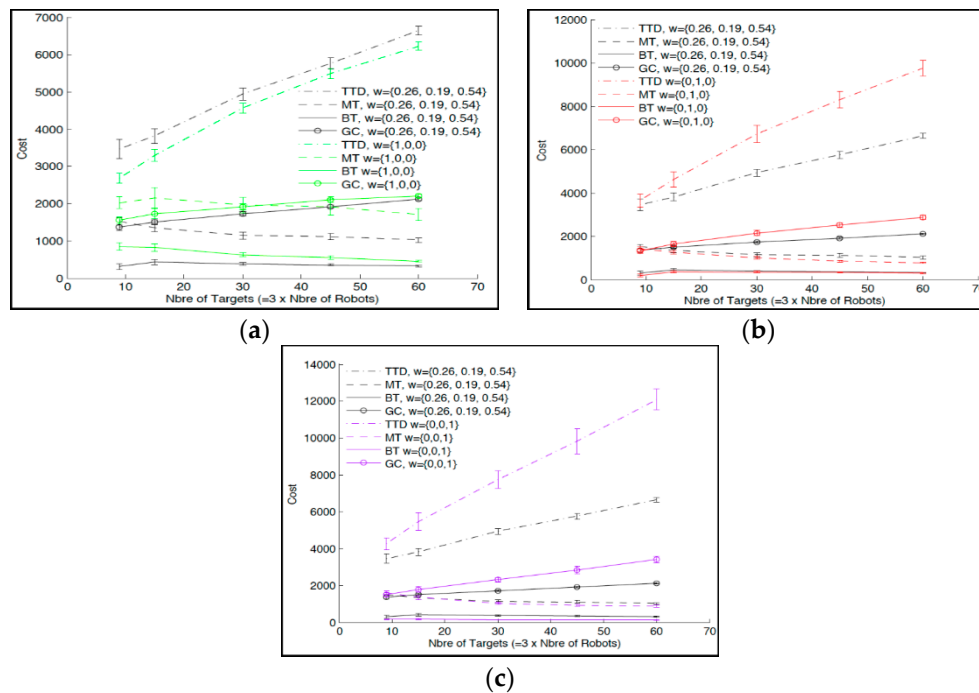

Figure 11. (**a**) AHP-MTSP comparison with mono-objective approach considering TTD. (**b**) AHP-MTSP comparison with mono-objective approach considering MT. (**c**) AHP-MTSP comparison with mono-objective approach considering DR.

The single objective method provides superior results in the case of the specific objective function under consideration. For example, when examining the mono-objective approach considering TTD (Figure 11a), a lower TTD cost was obtained compared to the multi-objective approach. This is anticipated, as the mono-objective approach leans towards only one criterion and excludes the others.

7.3. 3D Modeling System Implementation

To offer robustness, this system was developed using the ThreeJS, Unity, and Blender technologies [70]. A WebGL render was integrated into the scene engine to run code on both desktop/laptop systems and on mobile devices, using web technology. With WebGL, JavaScript will have native access to the API, in any compliant browser-based 3D graphics.

To render the fire, we used the script "VolumetricFire.js", which only required texture files and several parameters: fireWidth, fireHeight, fireDepth, and sliceSpacing. These parameters change based on the intensity. We used MTLLoader to load a 3D object (obj and mtl format). The "VolumetricFire.js" script gives the developed the flexibility to easily control and modify the 3D fire element based on received data from sensors. Under various factors, they have to response with the changes in the environment quickly. Figure 12 shows an example of an XML file describing a fire.

```xml
<xs:element name="fire" maxOccurs="unbounded" minOccurs="0">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="description">
        <xs:complexType>
          <xs:sequence>
            <xs:element type="xs:string" name="intensity"/>
            <xs:element type="xs:byte" name="positionX"/>
            <xs:element type="xs:byte" name="positionY"/>
            <xs:element type="xs:byte" name="position Z"/>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
```

Figure 12. XML file containing a fire description.

This file can be loaded to visualize the result in a near real-time 3D model. Figure 13 shows the interface of the proposed system. Users can visualize two models in real time, both the original one and the modified one, with a fire animation based on the alert and data received from the physical environment. With our proposed solution, we can visualize different locations catching fire. The system shows the output when the file shown in Figure 12 is used. Two locations are detected and modeled. In addition, we included several avatar facial animations, namely frightened, stressed, neutral, and happy, depending on the situation context.

Figure 13. The 3D modeling system output.

The XVEML file was first written by serializing the expression data. We then wrote into the file using the standard C# IO functions. We used sixteen features points on the avatar face. Texturing was conducted using Photoshop. Figure 14 shows an extract of the face animation XML file.

For easy facial animation, we adapted the rigging process [60]. We built a "custom controller" for every feature to obtain real avatar emotion in the Face Animation Controller module. MPEG-4 Face and Body Animation are utilized as a yardstick to verify avatar facial animations in a fire set up. The study incorporated the Visage SDK [71] as a guideline for avatar animations within the graphics engine (Figure 15). The selection of Visage SDK drew from its usefulness, wide usage, and full support of MPEG-4. Figure 15 illustrates three animations samples associated with the Visage SDK face track using 8.6 frames/second.

```xml
<?xml version="1.0"?>
<data>
  <ExpressionList>
    <Expression>
      <Firefighter_ID>1</Firefighter_ID>
      <ExpressionName>Frightened</ExpressionName>
      <ExpressionFile>Frightened</ExpressionFile>
      <Parameters>
        .......
      </Parameters>
    </Expression>
    <Expression>
      <Firefighter_ID>2</Firefighter_ID>
      <ExpressionName>Stressed</ExpressionName>
      ..............
    </Expression>
    <Expression>
      <Firefighter_ID>2</Firefighter_ID>
      <ExpressionName>Happy</ExpressionName>
      <Parameters>
        <Replay value="No" >
          < Start value="10">
            ............
      </Parameters>
    </Expression>
  </ExpressionList>

</data>
```
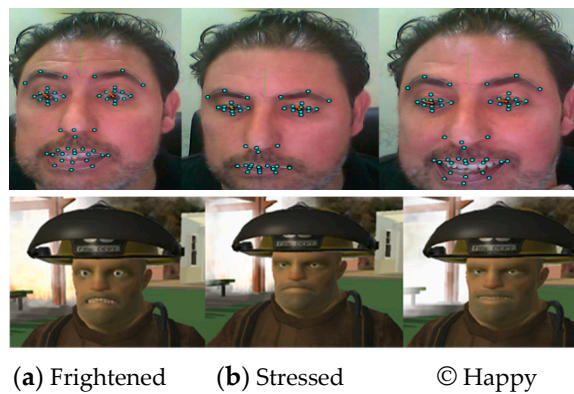
Figure 14. XML file for face control.

(**a**) Frightened   (**b**) Stressed   © Happy

Figure 15. Facial animation samples and corresponding MPEG-4 validation.

Whenever the study needs to insert a new expression, it has to be validated first to provide full realism to the proposed method [72]. When the firing density reduces, the firefighter's expression changes to happy. The state machine interpreter module transmits events to the face animation controller for reflection on the firefighter's face. Thereafter, the suitable facial expression file loads for rendering.

8. Conclusions

In this paper, we presented a new disaster management systems using cloud technology to improve mobility and availability. We integrated three packages: a 3D engine, an RLP for WSNs, and a rescue plan optimization approach. The proposed system forms a rich infrastructure for any disaster management application.

In the first package, we proposed a new OF called Cyber-OF for RPL, which was designed to meet the requirements of a disaster management application. Simulation results demonstrated the effectiveness of this OF in providing a better trade-off between energy consumption and real-time constraints.

In the second package, the problem of multi-objective multiple TSP came under consideration. The AHP-MTSP method was proposed. Simulation results demonstrated that the proposed multi-objective solution is superior to the single objective approach.

Finally, we developed an extensible game engine for 3D disaster management application that is based on an automatic concept that focuses on usability to easily extend set ups during the experiment. Novice developers with a weak programming background and 3D knowledge are hence capable

of developing games utilizing the game engine that is proposed. Avatars and object responses are modeled independently to provide more flexibility and vigor whenever updates are necessary.

Compared to other exiting systems discussed in Table 1, we developed a complete unified service platform for disaster management. The platform includes all the necessary features to monitor and simulate the situation and can achieve VE extension including avatar animation with minimum programming effort and without system interruption. In addition, the platform not only provides a 3D representation of the disaster, but also recommends an attack plan with possible resources. More importantly, it can provide users with more convenient and secure computing, storage, and other services using a cloud architecture.

In future work, generated scripts could be exploited to fit a wide range of other VE applications, such as military training, e-health, and e-learning. Moreover, we need to take into consideration the capacity of robots during the building of the optimized rescue plan.

References

1. Alotaibi, E.T.; Alqefari, S.S.; Koubaa, A. LSAR: Multi-UAV Collaboration for Search and Rescue Missions. *IEEE Access* **2019**, *7*, 55817–55832. [CrossRef]
2. Hagsand, O. Interactive MUVEs in the DIVE System. *IEEE Comput.* **1996**, *3*, 30–39.
3. Wood, T.; Cecchet, E.; Ramakrishnan, K.; Shenoy, P.; van der Merwe, J.; Venkataramani, A. Disaster recovery as a cloud service: Economic benefits & deployment challenges. In Proceedings of the 2nd USENIX Conference on Hot Topics in Cloud Computing, Boston, MA, USA, 22–25 June 2010.
4. Alhazmi, H.; Malaiya, K. Evaluating disaster recovery plans using the cloud. In Proceedings of the 2013 Proceedings Annual Reliability and Maintainability Symposium (RAMS), Orlando, FL, USA, 28–31 January 2013.
5. Wang, L.; Harper, E.; Mahindru, R.; Ramasamy, V. Disaster Recovery for Cloud-Hosted Enterprise Applications. In Proceedings of the 2016 IEEE 9th International Conference on Cloud Computing (CLOUD), San Francisco, CA, USA, 27 June–2 July 2016.
6. Shea, R.; Liu, J.; Ngai, E.C.-H.; Cui, Y. Cloud gaming: Architecture and performance. *IEEE Netw.* **2013**, *27*, 16–21. [CrossRef]
7. Zarrad, A. An Extensible Game Engine to Develop Animated Facial Avatars in 3D Virtual Environment. *Int. J. Virtual Communities Soc. Netw.* **2016**, *8*, 12–27. [CrossRef]
8. Eisert, P.; Fechteler, P. Low delay streaming of computer graphics. In Proceedings of the 2008 15th IEEE International Conference on Image Processing, San Diego, CA, USA, 12–15 October 2008.
9. Dhoedt, B.; Demeester, P. A hybrid thin-client protocol for multimedia streaming and interactive gaming applications. In Proceedings of the International Workshop on Network and Operating Systems Support for Digital Audio and Video, Newport, RI, USA, 22–23 November 2006.
10. Holthe, O.; Mogstad, O.; Ronningen, L. GeelixLiveG-ames: Remote playing of video games. In Proceedings of the 6th IEEE Consumer Communications and Networking Conference (CCNC'09), Las Vegas, NV, USA, 10–13 January 2009.
11. Zarrad, A. A Dynamic Platform for Developing 3D Facial Avatars in a Networked Virtual Environment. *Int. J. Comput. Games Technol.* **2016**, *2016*, 8489278. [CrossRef]
12. Chen, F.; Chang, R.; Lin, W.; Chen, S.; Chen, Y.; Li, C. Disaster and Emergency Management System. In Proceedings of the 15th International Symposium on Wireless Personal Multimedia Communications, Taipei, Taiwan, 24–27 September 2012; pp. 34–41.

13. Sakhardande, P.; Hanagal, S.; Kulkarni, S. Design of disaster management system using IoT based interconnected network with smart city monitoring. In Proceedings of the International Conference on Internet of Things and Applications (IOTA), Pune, India, 22–24 January 2016.

14. Wellington, J.; Ramesh, P. Role of Internet of Things in disaster management. In Proceedings of the 2017 International Conference on Innovations in Information, Embedded and Communication Systems (ICIIECS), Coimbatore, India, 17–18 March 2017.

15. Alnashwan, R.; Mokhtar, H. Disaster Management System Over Wifi Direct. In Proceedings of the 2nd International Conference on Computer Applications & Information Security (ICCAIS), Riyadh, Saudi Arabia, 1–3 May 2019.

16. Botta, A.; Cacace, J.; Lippiello, V.; Siciliano, B.; Ventre, G. Networking for Cloud Robotics: A case study based on the Sherpa Project. In Proceedings of the International Conference on Cloud and Robotics (ICCR), Saint Quentin, France, 23–23 November 2017.

17. Botta, A.; Gallo, L.; Ventre, G. Cloud, Fog, and Dew Robotics: Architectures for next generation applications. In Proceedings of the 7th IEEE International Conference on Mobile Cloud Computing, Services, and Engineering (MobileCloud), Newark, CA, USA, 4–9 April 2019; pp. 16–23.

18. Unity. Available online: http://www.unity3d.com (accessed on 2 January 2020).

19. Tim Sweeny. Unreal Script Language Reference. 1998. Available online: http://unreal.epicgames.com/UnrealScript.htm (accessed on 2 January 2020).

20. Epic Games. Epic Games' Unreal Development Kit Eclipses 50,000 Users in One Week [Online 2015]. Available online: http://www.udk.com/udk50k (accessed on 2 January 2020).

21. Emergent Game Technologies. Gamebryo. Available online: http:www.emergent.net/en/Products/Gamebryo/ (accessed on 2 January 2020).

22. CryEngine. Available online: http://www.crytek.com/cryengine (accessed on 2 January 2020).

23. Valve Software. Source Engine. Available online: http://source.valvesoftware.com/ (accessed on 2 January 2020).

24. Mehta, K. 3D Modeling and Rendering of 2D Medical Image. In Proceedings of the 2012 International Conference on Communication Systems and Network Technologies (CSNT), Rajkot, India, 11–13 May 2012.

25. Wang, B.; Li, H.; Rezgui, Y.; Bradley, A.; Ong, N. BIM Based Virtual Environment for Fire Emergency Evacuation. *Sci. World J.* **2014**, *2014*, 589016. [CrossRef] [PubMed]

26. Ren, A.; Chen, C.; Luo, Y. Simulation of emergency evacuation in virtual reality. *Tsinghua Sci. Technol.* **2008**, *13*, 674–680. [CrossRef]

27. Cao, C.; Weng, Y.; Lin, S.; Zhou, K. 3D shape regression for real-time facial animation. *ACM Trans. Graph. (TOG)* **2013**, *32*, 232–250.

28. Zarraonandia, T.; Vargas, M.R.R.; Díaz, P.; Aedo, I. A Virtual Environment for Learning Aiport Emergency Management Protocols. In *Human-Computer Interaction. Ambient, Ubiquitous and Intelligent Interaction, Lecture Notes in Computer Science*; Springer: Berlin/Heidelberg, Germany, 2009; Volume 5612, pp. 228–235.

29. HashemiBeni, L.; Mostafavi, M.A.; Pouliot, J. 3D Dynamic Simulation within GIS in Support of Disaster Management. In *Geomatics Solutions for Disaster Management. Lecture Notes in Geoinformation*; Springer: Berlin/Heidelberg, Germany, 2007; pp. 165–184.

30. Magerko, B.; Laird, J.E. Building an Interactive Drama Architecture. In Proceedings of the 1st International Conference on Technologies for Interactive Digital Storytelling and Entertainment, Darmstadt, Germany, 24–26 March 2003; pp. 226–237.

31. Oliveira, M.; Crowcroft, J.; Slater, M. Component Framework Infrastructure for Virtual Environments. In Proceedings of the Third International Conference on Collaborative Virtual Environments, San Francisco, CA, USA, 1 September 2000; pp. 139–146.

32. Boukerche, A.; Duarte, D.D.; Araujo, R. A Language for Building and Extending 3D Virtual Web-Based Environments. In Proceedings of the 2th Latin American Web Congress and10th Brazilian Symposium on Multimedia and the Web, Ribeirao Preto, Brazil, 15 October 2004; pp. 52–60.

33. Browne, C. Evolutionary game design. *IEEE Trans. Comput. Intell. AI Games* **2011**, *2*, 11–21.

34. Maggiore, G.; Spanò, A.; Orsini, R.; Bugliesi, M.; Abbadi, M.; Steffinlongo, E. A formal specification for Casanova, a language for computer games. In Proceedings of the 4th ACM SIGCHI Symposium on Engineering Interactive Computing Systems, Copenhagen, Denmark, 25–26 June 2012; pp. 287–292.

35. McGugan, W. *Beginning Game Development with Python and Pygame: From Novice to Professional*; Apress: New York, NY, USA, 2007.

36. Schaul, T. An Extensible Description Language for Video Games. *IEEE Trans. Comput. Intell. AI Games* **2014**, *6*, 325–331. [CrossRef]

37. Boukerche, A.; Duarte, D.; Araujo, R.; Andrade, L.; Zarrad, A. A novel solution for the development of collaborative virtual environment simulations in large scale. In Proceedings of the Ninth IEEE International Symposium on Distributed Simulation and Real-Time Applications, Montreal, QC, Canada, 10–12 October 2005; pp. 86–97.

38. Zarrad, A.; Bensefia, A. A Novel Approach to develop Large-Scale Virtual Environment Applications Using Script Language. In Proceedings of the 9th International Conference on Innovations in Information Technology (IIT), Abu Dhabi, UAE, 17–19 March 2013; pp. 169–174.

39. Szilas, N. IDtension: A narrative engine for Interactive Drama. In Proceedings of the Technologies for Interactive Digital Storytelling and Entertainment (TIDSE) Conference, Darmstadt, Germany, 24 March 2003.

40. Perlin, K.; Goldberg, A. Improv: A System for Scripting Interactive Characters in Virtual Worlds. In Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques, New Orleans, LA, USA, 4–9 August 1996; ACM Press: New York, NY, USA, 1996; pp. 205–216.

41. Arafa, Y.; Kamyab, K.; Kshirsagar, S.; Magnenat-Thalmann, N.; Guye-Vuille, A.; Thalmann, D. Avatar Markup Language. In Proceedings of the Eigth Eurographics Workshop on Virtual Environments, Barcelona, Spain, 30–31 May 2002; pp. 109–118.

42. Chow, M.M. Optimized geometry compression for real-time rendering. In Proceedings of the Visualization '97 (Cat. No. 97CB36155), Phoenix, AZ, USA, 24 October 1997.

43. Seitz, S.M.; Dyer, C.M. View morphing. In Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques, New Orleans, LA, USA, 4–9 August 1996; pp. 21–30.

44. Lhuillier, M.; Quan, L. Image interpolation by joint view triangulation. In Proceedings of the 1999 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (Cat. No PR00149), Fort Collins, CO, USA, 23–25 June 1999; Volume 2, pp. 139–145.

45. Adelson, E.H.; Bergen, J.R. The Plenoptic Function and Elements of Early Vision. In *Computational Models of Visual Processing*; MIT Press: Cambridge, MA, USA, 1991; pp. 3–20.

46. Gaddour, O.; Koubaa, A. RPL in a nutshell: A survey. *Comput. Netw.* **2012**, *56*, 3163–3178. [CrossRef]

47. Gaddour, O.; Koubaa, A.; Rangarajan, R.; Cheikhrouhou, O.; Tovar, E.; Abid, M. Co-rpl: Rpl routing for mobile low power wireless sensor networks using corona mechanism. In Proceedings of the 9th IEEE International Symposium on Industrial Embedded Systems (SIES 2014), Pisa, Italy, 18–20 June 2014; pp. 200–209.

48. Cheikhrouhou, O.; Koubaa, A.; Boujelben, M.; Abid, M. A lightweight user authentication scheme for Wireless Sensor Networks. In Proceedings of the ACS/IEEE International Conference on Computer Systems and Applications—AICCSA 2010, Hammamet, Tunisia, 16–19 May 2010; pp. 1–7.

49. Cheikhrouhou, O.; Koubaa, A.; Dini, G.; Abid, M. RiSeG: A ring based secure group communication protocol for resource-constrained wireless sensor networks. *Pers. Ubiquitous Comput.* **2011**, *15*, 783–797. [CrossRef]

50. Noh, J.; Neumann, U. *A Survey of Facial Modeling and Animation Techniques*; Technical Report 99-705; USC: Los Angeles, CA, USA, 1998.

51. Kurunathan, H.; Severino, R.; Koubaa, A.; Tovar, E. IEEE 802.15. 4e in a nutshell: Survey and performance evaluation. *IEEE Commun. Surv. Tutor.* **2018**, *20*, 1989–2010. [CrossRef]

52. Vasseur, J.P.; Kim, M. (Eds.) *Routing Metrics Used for Path Calculation in Low-Power and Lossy Networks*; IETF: Fremont, CA, USA, 2012; RFC 6551.

53. Sarakis, L.; Leligou, H.C.; Karkazis, P.; Velivassaki, T.H.; Zahariadis, T. Evaluation of a Transmission Count-Efficient Metric for Energy-Aware Routing in WSNs. In Proceedings of the International Conference on Telecommunications and Multimedia (TEMU), Heraklion, Greece, 28–30 July 2014; pp. 98–103.

54. Gaddour, O.; Koubaa, A.; Abid, M. Quality-of-service aware routing for static and mobile ipv6-based low-power and lossy sensor networks using RPL. *Ad Hoc Netw.* **2015**, *33*, 233–256. [CrossRef]

55. Aissa, Y.B.; Grichi, H.; Khalgui, M.; Koubâa, A.; Bachir, A. QCOF: New RPL Extension for QoS and Congestion-Aware in Low Power and Lossy Network. In Proceedings of the 14th International Conference on Software Technologies, Prague, Czech Republic, 26–28 July 2019; pp. 560–569.

56. Kurunathan, H.; Severino, R.; Koubaa, A.; Tovar, E. DynaMO—Dynamic Multisuperframe Tuning for Adaptive IEEE 802.15.4e DSME Networks. *IEEE Access* **2019**, *7*, 122522–122535. [CrossRef]

57. Gaddour, O.; Koubaa, A.; Baccour, N.; Abid, M. Of-fl: Qos-aware fuzzy logic objective function for the rpl routing protocol. In Proceedings of the 2014 12th International Symposium on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks (WiOpt), Hammamet, Tunisia, 12–16 May 2014; pp. 365–372.

58. Ohta, M. IETF and Internet standards. *IEEE Commun. Mag.* **1998**, *36*, 126–129. [CrossRef]

59. VRML. Available online: http://www.w3.org/MarkUp/VRML/ (accessed on 2 January 2020).

60. Baran, I.; Popovi, J. Automatic Rigging and Animation of 3D Characters. *ACM Trans. Graph. (TOG)* **2007**, *26*, 72. [CrossRef]

61. Koubâa, A.; Qureshi, B.; Sriti, M.F.; Javed, Y.; Tovar, E. A service-oriented Cloud-based management system for the Internet-of-Drones. In Proceedings of the 2017 IEEE International Conference on Autonomous Robot Systems and Competitions (ICARSC), Coimbra, Portugal, 26–28 April 2017; pp. 329–335.

62. Koubaa, A.; Qureshi, B.; Sriti, M.F.; Allouch, A.; Javed, Y.; Alajlan, M.; Cheikhrouhou, O.; Khalgui, M.; Tovar, E. Dronemap Planner: A service-oriented cloud-based management system for the Internet-of-Drones. *Ad Hoc Netw.* **2019**, *86*, 46–62. [CrossRef]

63. Koubaa, A.; Qureshi, B. DroneTrack: Cloud-Based Real-Time Object Tracking Using Unmanned Aerial Vehicles Over the Internet. *IEEE Access* **2018**, *6*, 13810–13824. [CrossRef]

64. Koubaa, A.; Allouch, A.; Alajlan, M.; Javed, Y.; Belghith, A.; Khalgui, M. Micro Air Vehicle Link (MAVlink) in a Nutshell: A Survey. *IEEE Access* **2019**, *7*, 87658–87680. [CrossRef]

65. Koubaa, A.; Alajlan, M.; Qureshi, B. ROSLink: Bridging ROS with the Internet-of-Things for Cloud Robotics. In *Robot Operating System (ROS)*; Springer: Cham, Switzerland, 2017; pp. 265–283.

66. Saaty, R. The analytic hierarchy process-what it is and how it is used. *Math. Model.* **1987**, *9*, 161–176. [CrossRef]

67. Trigui, S.; Cheikhrouhou, O.; Koubaa, A.; Zarrad, A.; Youssef, H. An analytical hierarchy process-based approach to solve the multi-objective multiple traveling salesman problem. *Intell. Serv. Robot.* **2018**, *11*, 355–369. [CrossRef]

68. Cheikhrouhou, O.; Koubâa, A.; Bennaceur, H. Move and improve: A distributed multi-robot coordination approach for multiple depots multiple travelling salesmen problem. In Proceedings of the 2014 IEEE International Conference on Autonomous Robot Systems and Competitions (ICARSC), Espinho, Portugal, 14–15 May 2014; pp. 28–35.

69. Trigui, S.; Koubâa, A.; Cheikhrouhou, O.; Qureshi, B.; Youssef, H. A Clustering Market-Based Approach for Multi-robot Emergency Response Applications. In Proceedings of the 2016 International Conference on Autonomous Robot Systems and Competitions (ICARSC), Braganca, Portugal, 4–6 May 2016; pp. 137–143.

70. Takala, M.; Meeri, M.; Hamalainen, P. Immersive 3D modeling with Blender and off-the-shelf hardware. In Proceedings of the 2013 IEEE Symposium on 3D User Interfaces (3DUI), Orlando, FL, USA, 16–17 March 2013.

71. Pejsa, T.; Pandzic, I.S. Architecture of an animation system for human characters. In Proceedings of the 2009 10th International Conference on Telecommunications, Zagreb, Croatia, 8–10 June 2009; pp. 171–176.

72. Pandzic, I.; Forchheimer, R. *MPEG-4 Facial Animation: The Standard, Implementation and Applications*; John Wiley & Sons, Ltd.: Hoboken, NJ, USA, 2002.