Future Generation Computer Systems 92 (2019) 1051–1065

Contents lists available at ScienceDirect

Future Generation Computer Systems

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

AGRA: AI-augmented geographic routing approach for IoT-based incident-supporting applications

Dmitrii Chemodanov [a,*], Flavio Esposito [b], Andrei Sukhov [c], Prasad Calyam [a], Huy Trinh [a], Zakariya Oraibi [a]

[a] University of Missouri-Columbia, USA
[b] Saint Louis University, USA
[c] Samara National Research University, Russia

HIGHLIGHTS

- We propose an Artificial Intelligence (AI)-augmented geographic routing approach (AGRA). Our approach uses deep learning over satellite imagery available at the edge clouds to enable IoT-based incident-supporting applications.
- We leverage a novel repulsive forwarding technique that theoretically guarantees local minimum avoidance and approximates the shortest path algorithm with a 3.291 path stretch bound.
- Using both large scale numerical and NS-3 event-driven simulations we show how our proposed approach outperforms classical geographic routing algorithms performance, e.g:, goodput and enable real-time situational awareness from the IoT devices located at the disaster scene.

ARTICLE INFO

Article history:
Received 15 March 2017
Received in revised form 1 July 2017
Accepted 8 August 2017
Available online 25 August 2017

Keywords:
Incident-supporting applications
IoT
Deep learning
Geographic routing
Local minimum avoidance
Electrostatics

ABSTRACT

Applications that cater to the needs of disaster incident response generate large amount of data and demand large computational resource access. Such datasets are usually collected in real-time at the incident scenes using different Internet of Things (IoT) devices. Hierarchical clouds, *i.e.*, core and edge clouds, can help these applications' real-time data orchestration challenges as well as with their IoT operations scalability, reliability and stability by overcoming infrastructure limitations at the ad-hoc wireless network edge. Routing is a crucial infrastructure management orchestration mechanism for such systems. Current geographic routing or greedy forwarding approaches designed for early wireless ad-hoc networks lack efficient solutions for disaster incident-supporting applications, given the high-speed and low-latency data delivery that edge cloud gateways impose. In this paper, we present a novel Artificial Intelligent (AI)-augmented geographic routing approach, that relies on an area knowledge obtained from the satellite imagery (available at the edge cloud) by applying deep learning. In particular, we propose a stateless greedy forwarding that uses such an environment learning to proactively avoid the local minimum problem by diverting traffic with an algorithm that emulates electrostatic repulsive forces. In our theoretical analysis, we show that our Greedy Forwarding achieves in the worst case a 3.291 path stretch approximation bound with respect to the shortest path, without assuming presence of symmetrical links or unit disk graphs. We evaluate our approach with both numerical and event-driven simulations, and we establish the practicality of our approach in a real incident-supporting hierarchical cloud deployment to demonstrate improvement of application level throughput due to a reduced path stretch under severe node failures and high mobility challenges of disaster response scenarios.

1. Introduction

Providing technologies in response to a natural or man-made disaster is challenging, due to traditional infrastructure assumptions that may fail given the damage made by man or natural-caused disasters. Additionally, there is a need to handle large

* Corresponding author.
E-mail addresses: dycbt4@mail.missouri.edu (D. Chemodanov), espositof@slu.edu (F. Esposito), amskh@yandex.ru (A. Sukhov), calyamp@missouri.edu (P. Calyam), hntzq4@mail.missouri.edu (H. Trinh), zaonr5@mail.missouri.edu (Z. Oraibi).

http://dx.doi.org/10.1016/j.future.2017.08.009
0167-739X/

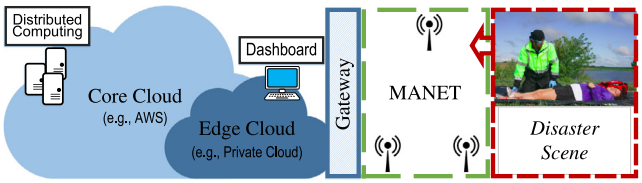


Fig. 1. During incidents caused by disasters, large dataset generated on-site needs large computation/storage resources. Consequently, data needs to be transferred to an edge cloud at a high constant speed through an incident-supporting wireless edge comprising Mobile Ad-Hoc Network (MANET).

datasets from multiple incident scenes and disaster relief coordination between first responder agencies such as fire, police or hospitals.

Applications designed for disaster incident response may benefit from the latest hierarchical cloud trends [1–3], *i.e.*, core and edge clouds. Such technologies are integrated with network virtualization and dynamic on-demand access to networking, computation and storage resource, wherever available. In particular, applications that provide visual situational awareness are crucial for first responders, as they are based on data collected in the real-time at the incident scenes using different Internet of Things (IoT) devices, such as sensors, wearable heads-up display devices, bluetooth beacons, etc [3]. For example, tablets, wearable heads-up display or smartphone devices can be used for the real-time video conferencing with the incident commander featuring face recognition of disaster victims [4], or to detect children in attempt to reunite them with their guardians [5], whereas virtual beacons can be mainly used to track their location.

In Fig. 1 we illustrate our "Panacea's Cloud" system deployment, *i.e.*, a hierarchical cloud setup that leverages a IoT-based incident-supporting application to provide real-time situational awareness *e.g.*, medical triage communications for paramedics and other first responders [6–8]. For an effective disaster incident response, the design of the hierarchical cloud needs to overcome the challenges of application's real-time data movement requirements and any infrastructure limitations at the network edge. To cope with the potential loss of infrastructure in a disaster scene, a mobile ad-hoc wireless network (MANET) needs to be operational for transferring media-rich visual information from the disaster scene as quickly as possible to the edge cloud gateway. Such (visual) data can be used, *e.g.*, within a medical application context to transfer high-definition video streams generated by paramedics' wearable heads-up display devices from the disaster triage scene to a dashboard located at the edge cloud, or by a first responder for a live remote medical consultation. The incident response and resource allocation decision making *e.g.*, ambulance routing to scene, medical supply replenishment tracking, requires significant computational resources that can be augmented on demand by a core cloud cluster (see Fig. 1).

For such edge cloud network scenarios to be operational, traffic generated by the MANET needs to be handled dynamically and with low-latency constraints. Geographic routing-based approaches are generally suitable for these applications; however, there is a lack of incident-supporting forwarding and routing approaches in MANETs capable of providing sustainable high-speed data delivery to the edge cloud gateway [9]. Specifically, there is a need to design a better performant greedy-forwarding solutions that does not suffer from the *local minimum problem in presence of (non-arbitrary) node failures and mobility*. This problem is caused by the lack of global routing knowledge of the greedy forwarding algorithm [10–13] that may deliver packets to nodes that do not have neighbors closer to the destination than themselves.

Existing algorithms provide only partial solutions to the local minimum problem and can be classified into stateful or stateless solutions. Existing stateless greedy forwarding solutions may fail to find a path even if it exists [12,13], or they stretch such paths significantly [11] by visiting almost all possible nodes to "desperately" find a way. Existing stateful greedy forwarding algorithms instead (*e.g.*, those relying on the network topology knowledge including spanning trees [14,15] or partial paths [16]) are sensitive to frequent node failures and mobility [16–18], a typical scenario within regions and infrastructures hit by a natural or man-made disaster. Consequently, such algorithms lead to poor or unacceptable performance in incident-supporting applications that need constant high-speed data transfer to provide crucial real-time situational awareness.

Our contributions. To cope with the above geographic routing limitations, in this paper we leverage the latest advances in the Artificial Intelligence (AI) area and present a novel AI-augmented geographic routing approach (AGRA). The main contributions of our AGRA are as follows:

- Our main ***design*** contribution lays in our AGRA approach that utilizes a knowledge about physical obstacles presence in a geographic area obtained by deep learning [19,20] from the satellite imagery (maps) available at the system's dashboard (see Section 3). In addition to widely adopted geographic coordinates, the obtained geographical obstacle knowledge is then used to build a conceptually different greedy forwarding approach that avoids the *local minima problem* while supporting high-speed data delivery, *e.g.*, of high-definition data streams and multi-modal data, across large disaster incident scene areas.
- Our main ***theoretical*** contribution lays in our results pertaining to a geographic routing model that guarantees a local minimum avoidance as well as the shortest path approximation in (mobile) ad-hoc networks (see Section 4). Specifically, our stateless greedy forwarding approach builds upon electrostatics principles such as the Green's function [21] to model packets as charges immersed in an electrostatic potential field from the source to the destination, and we steer their route by charging regions containing obstacles accordingly, hence guaranteeing local minimum avoidance. Moreover, when packets are forwarded using gradient descent on the Green's function potential field (along lines of electrical force), we show a ***3.291*** path stretch approximation bound. To our knowledge, this is the first greedy forwarding algorithm that has theoretical guarantees on the path stretch length without any strong assumptions on wireless ad-hoc network such as symmetrical links or unit disk graphs.
- Our main ***algorithmic*** contribution lays in our two novel algorithms, *viz*. *Attractive Repulsive Greedy Forwarding* (ARGF) and *Attractive Repulsive Pressure Greedy Forwarding* (ARPGF). Both algorithms use the notion of *electrostatic repulsion* to enhance greedy forwarding (see Section 5). ARGF does not theoretically guarantee the local minimum avoidance due to the complexity of computing the exact theoretical potential field on multiple obstacles of arbitrary shape and due to discrete node distribution. For this reason, we extend ARGF with a known pressure recovery technique (ARPGF) to guarantee delivery at expense of a small path stretch.

We evaluate our algorithms using numerical simulations with asymmetrical connectivity and obstacles of complex convex shape, and with an event-driven simulations obtained considering an actual incident-supporting hierarchical cloud deployment.[1] We

[1] The source code of both our simulators is publicly available under a GNU license at https://github.com/duman190/AGRA.

found that our ARGF and ARPGF algorithms outperform related stateless greedy forwarding solutions such as Greedy Forwarding [12], Greedy Perimeter Stateless Routing (GPSR) [13] (*face routing* representative), and Gravity Pressure Greedy Forwarding (GPGF) [11] (*pressure forwarding* representative). Particularly, ARPGF outperforms GPGF under practical Time To Live (TTL) constraints. Moreover, we show that when the packet TTL $\leq$ 128 and under legitimate assumptions, ARPGF data could fit in the available IP packet header space, hence it has minor overhead. At the same time, we found that GPSR performs worse than the ARGF algorithm due to the former's unrealistic assumptions on the underlaying network graph, *i.e.*, unit disk or planar graph. Finally, our NS-3 [22] event-driven simulations also confirm superior ARGF (and hence ARPGF) goodput performance compared to GPSR and other stateful reactive routing protocols, such as Ad-hoc On-demand Distance Vector (AODV) [23] and IEEE 802.11 s standard Hybrid Mesh Network Protocol (HWMP) [24], especially in challenging disaster response conditions of severe node failures and high mobility.

Paper organization. In Section 2, we discuss related work. In Section 3, we motivate the local minimum problem and describe how deep learning can be applied to derive physical obstacle information from the satellite imagery available at the edge cloud. In Section 4, we introduce our theoretical electro-static-based forwarding model to include physical obstacle knowledge, and show how the *repulsive* field guarantees the shortest path approximation as well as the local minimum avoidance. Section 5 describes our practical approximation of the theoretical model. Section 6 presents our evaluation methodology, performance metrics and results that show effectiveness of our proposed approach. Section 7 concludes the paper.

2. Related work

Edge routing and IoT. Recent advances in the IoT have brought challenges in storage, networking and computation management and under several scenarios, including mobile edge computing [3,25], wireless sensor networking [26,27] or cognitive networking [28,29]. Among the most severe IoT challenges for data marshaling, we have seen scarce energy, high mobility and frequent failures [16,29]. To overcome some of these challenges in bridging IoT devices with the gateway, edge and core clouds, recent advances in the Software-Defined Networking (SDN) and Network Function Virtualization (NFV) have been adopted [3,25]. For example, a control plane has been used to dynamically find low-latency and high bandwidth paths that satisfy IoT-based application demands [3,30]. Existing solutions that leverage data transfer to a gateway commonly require knowledge of the network topology such as spanning trees [14,15] or network clusters [27]. In this paper, we consider a special case of data delivery from the disaster incident scenes to the gateway over MANETs, that is in contrast to typical wireless sensor network scenarios. Because of the high node mobility (*i.e.*, due to use of MANETs) and severe node failures (*e.g.*, due to intermittent energy supply), solutions that rely on a logically centralized network control [3,25] or on the network topology knowledge [15–17,27] can be inadequate. Our approach copes with the above limitations in disaster incident scenarios via a stateless greedy forwarding protocol that finds high bandwidth and low-latency paths by improving packets delivery and minimizing path stretch. Moreover, our approach does not directly address energy scarceness of the IoT devices, but copes with it in a best effort manner by using stateless greedy forwarding that avoids usage of routing protocols that can drain a battery of IoT devices relatively fast.

Physics in computer networks. Applying physics laws to solve computer network problems is not a novelty. The first successful attempt, to our knowledge is the popular result by Shannon, who created the basics of information theory relying on the entropy definition from physics [31]. To justify network effects new models such as the "small world" effect, cluster models, network correlation, random graph model, network growth model and many others have been developed. All these models rely on physics to some extent. A survey of these models can be found in [32].

Narrowing our attention to routing and forwarding schemes using potential fields, we found a few routing and forwarding schemes using potential fields similar to our approach [33–35]. Their solution is aimed at balancing the network load by the natural property of electrostatic lines of force to be geo-spatially dispersed. In [33], authors use the aforementioned property to select a path trajectory so that a greedy forwarded packet can reach the destination without facing a local minimum. In [34], authors use numerical calculations to optimize network load for one-to-many and many-to-many communication patterns. However, both schemes do not address the local minimum problem due to presence of obstacles directly and can benefit from using our approach. In [35], the authors use a potential field to repulse traffic in excess from heavy loaded sensors to reduce congestion. Our proposed approach can be used similarly to directly deal with congestions and other disaster incident-supporting geographic routing problems.

Geographic routing and MANET. The literature on geographic routing and greedy forwarding is also vast, and here we focus on the most valuable works that help to highlight our contributions. Many geographic routing algorithms that can recover from a local minimum have been proposed. One of the first geographic routing solutions which guarantees delivery were Greedy Perimeter Stateless Routing (GPSR) [13] and GFG [36]. To recover from a local minimum, both protocols use face routing which requires strong assumptions such as unit disk and planar graphs. However, planar graphs can be disconnected when graphs have arbitrary shapes, nodes are mobile and real physical obstacles appear [37]. Kim et al. [37] propose a solution which overcomes planar graph limitations in practice by introducing Cross-Link Detection Protocol (CLDP) complication. As later works have show [15,17], CLDP requires expensive signaling to detect and remove crossed edges. Authors in [15] proposed Greedy Distributed Spanning Tree Routing (GDSTR) which requires less expensive distributed spanning tree construction (to maintain one or several spanning trees) to guarantee delivery and recover from a local minimum. GDSTR is also extended to a 3D case [38]. Kleinberg et al. [14] use spanning trees for greedy embedding, *i.e.*, for an assignment of virtual coordinates to greedy forward a packet without facing a local minimum. More recent works [16–18] show that spanning trees are sensitive to dynamic topologies and mobility. Moreover, most of the aforementioned solutions were designed for static sensor networks which are limited in dynamics. Our approach obtains better path stretch results, also works in 3D spaces, but does not require the time and space complexity of a spanning trees construction.

More recent protocols such as MTD and WEAVE [16,17] can cope with topology dynamics to some extent. For example MTD requires construction of Delaunay triangulation (DT) graphs for local minimum recovery. When topology changes, nodes may loose their Delaunay neighbors which are needed for recovery from a local minimum. Contrary to ours, all of the aforementioned protocols are stateful — *i.e.*, they rely on global or partial topology knowledge and therefore their performance degrades under node mobility or failures — common for disaster scenarios. Moreover, all these algorithms build around greedy forwarding and hence, they can benefit from using our *repulsive* field to proactively avoid local minima created by obstacles. To our knowledge, we are the first to introduce a theoretical solution to the local minimum avoidance

that approximates with a bound the shortest path in ad-hoc networks by creating conceptually different forwarding decision rules.

Geographic routing and the Internet. Geographic routing has been also proposed for the Internet [11,18,39] that is less dynamic than wireless ad-hoc networks. The authors in [11] build upon the work of Kleinberg et al. [14] and show that due to inaccurate greedy embedding caused by topology dynamics, packets can get stuck in a local minimum. To this aim, they propose the Gravity Pressure Greedy Forwarding (GPGF) [11] protocol which is shown to have guaranteed packet delivery on graphs of an arbitrary shape. To recover from local minimum, GPGF counts the number of node visits (storing that information in packet headers) to press packets from local minima until greedy forwarding can resume. The key idea beyond pressure recovery is a greedy forwarding gradient descent property — once a packet reaches a location closer to the destination, there is no way how the packet can be forwarded back to the previous location of a local minimum. However, such a recovery needs expensive packet header space [18] and can stretch path significantly to undesired levels.

AGRA: AI-augmented geographic routing. In this paper, we use a conceptually different *repulsive* field for geographic routing to benefit from a static physical obstacle knowledge obtained by using deep learning-based detectors [19,20] over any available edge satellite maps. The proposed approach based on the electrostatic potential of Green's function theoretically guarantees avoidance of a local minimum as well as the shortest path approximation. The electrostatic field guiding packets has a gradient descent property, with minimum at the destination. This means that greedy forwarding can be also complemented with the gravity pressure mode of GPGF for a local minimum avoidance. Our proposed algorithms, *i.e*. Attractive Repulsive Greedy Forwarding (ARGF) and Attractive Repulsive Pressure Greedy Forwarding (ARPGF) use both repulsive and attractive fields to greedy forward a packet (in 2D or 3D Euclidean spaces). As shown in our simulations, such a greedy forwarding synergy enhances the path stretching property of GPGF and hence the delivery ratio (limited by the packet's Time To Live), making ARPGF suitable for the incident-supporting wireless edge (*i.e.*, ad hoc) networks. Due to the static obstacle knowledge, proposed algorithms can cope better with high mobility and severe node failures, which results in an overall greater goodput during disaster scenarios, crucial for most of the incident-supporting situational awareness applications. In the absence of obstacles knowledge or with obstacle location miscalculations by deep learning, the performance of such proposed algorithms degrades with respect to their respective predecessor performances, *i.e.*, ARGF to GF [12] and ARPGF to GPGF [11].

3. Problem motivation and AI relevance

In the highly mobile and frequent node failure conditions during a disaster-incident response, we cannot rely on the network topology knowledge such as routing tables, spanning trees, etc. Thus, most of the geographic routing solutions designed for static sensor networks and available for MANETs today are poorly applicable for the disaster-incident case. On the other hand, local minimum of the geographic routing often appears near large physical obstacles (especially, of concave shapes) such as man-made (*e.g.*, buildings) or natural (*e.g.*, close to lakes or ponds). Figs. 2(a) and 2(b) illustrate these potential physical obstacles.

Fortunately, due to the design of our "Panacea's Cloud" system [6–8], we have satellite maps available (pre-uploaded) at the edge cloud which contain information about the disaster-incident area. Fig. 3 shows maps of Joplin, MO area before and after tornado damages that occurred on May 22nd, 2011. The tornado response imagery of Joplin, MO is available at [40]. We can see how information (*e.g.*, size and location) of the Joplin High School (see Figs. 3(a) and 3(b)) and the Joplin Hospital (see Figs. 3(c) and 3(d)) buildings (and other physical objects) are slightly impacted by the tornado disaster incident. Thus, we can benefit even from currently available maps of the disaster incident area by addressing the following problems — (i) how to extract information about potential physical obstacles (*e.g.*, buildings, lakes, etc.) from the available maps of the disaster scene; and (ii) how to use this knowledge within geographic routing to improve its goodput (*i.e.*, application layer throughput) sufficient for the *real-time visual situational awareness*.

3.1. Obstacle detector architecture

The first problem can be solved by manually labeling all potential obstacles on the map. However, due to any given disaster-scene scale and due to the fact that time is critical for first responders, labeling physical obstacles on the map all the time (*i.e.*, the manual approach) can be intractable. Thus, we adopt an approach that involves automation of that process.

The artificial intelligence, and more specifically the pattern recognition areas today are ideally suited for such automation processes. The pattern recognition field today includes many approaches related to an object detection (finding the object location and its size) in the given image (*e.g.*, satellite imagery). These approaches include: Support Vector Machines, Nearest Neighbor, Deep Learning and other techniques. However, the most accurate detectors rely on deep learning approaches [19,20]. For example, the You Only Look Once (YOLO) [19] deep-learning detector predicts objects in images using only a single neural network composed by 26 layers, an easier task for the edge cloud. At the same time, it can have worse performance than more sophisticated deep learning detectors [20]. Note that alternatives to object detection include also geographical object-based image analysis [41] that relies on the spectral information extracted from image pixels that may require additional and more expensive LiDAR hardware [3] not necessarily available during incident response scenarios.

However, the deep learning object detectors may not find obstacles or misclassify them in some cases. Thus, they still require some human interaction *i.e.*, labeling of some of the detected and correctly classified or misclassified samples. To cope with deep learning complexity, in our "Panacea's Cloud" architecture shown in Fig. 4, we move deep learning to the core cloud due to limited edge cloud storage and computation resources. Further, we assume that training samples are collected and partly labeled during the previous incident responses at the edge cloud. These samples are then used for supervised or semi-supervised deep learning [42] to enhance performance of the detector in future. The up-to-date detector version then can be pre-uploaded to the edge cloud and used *off-line* during disaster-incident response activities within a lost infrastructure region. Once detected, physical obstacles are then propagated to the MANET through a gateway.

Note that in this paper, we do not focus on finding the best (*i.e.*, the most accurate) approach for the obstacle detection on the satellite maps. Instead, we address the second problem — how to utilize obtained physical obstacle knowledge in the geographic routing to support real-time visual situational awareness under challenging disaster-incident conditions.

4. Repulsive field model

In this section, we describe a theoretical solution that can be used to incorporate the physical obstacle's knowledge extracted from the maps within the geographic routing. We will use an approximation of this theory as a design principle for our algorithms described in subsequent sections.

Let us consider a wireless network with nodes uniformly distributed over the continuous $\mathbb{R}^2$ (or $\mathbb{R}^3$ in 3-dimensional case)

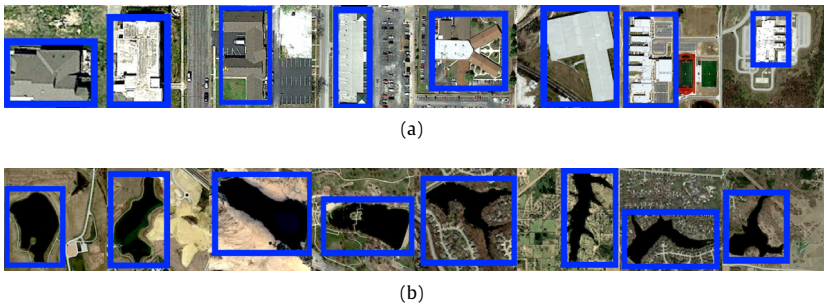


Fig. 2. Satellite imagery examples of various physical obstacles which can be used for training purposes including both man-made *e.g.*, buildings (a) and natural *e.g.*, lakes or ponds (b).

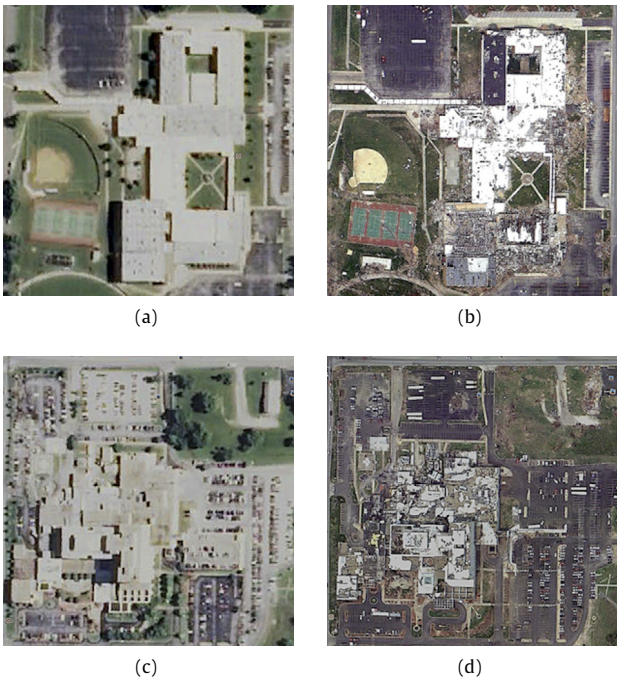


Fig. 3. Satellite imagery of Joplin, MO area including Joplin High School (*top row*) and Joplin Hospital (*bottom row*) buildings before (a, c) and after (b, d) tornado damages on May 22nd, 2011: we can see how captured on satellite images information about *size* and *location* of buildings (and other physical objects) was slightly impacted by the disaster incident.

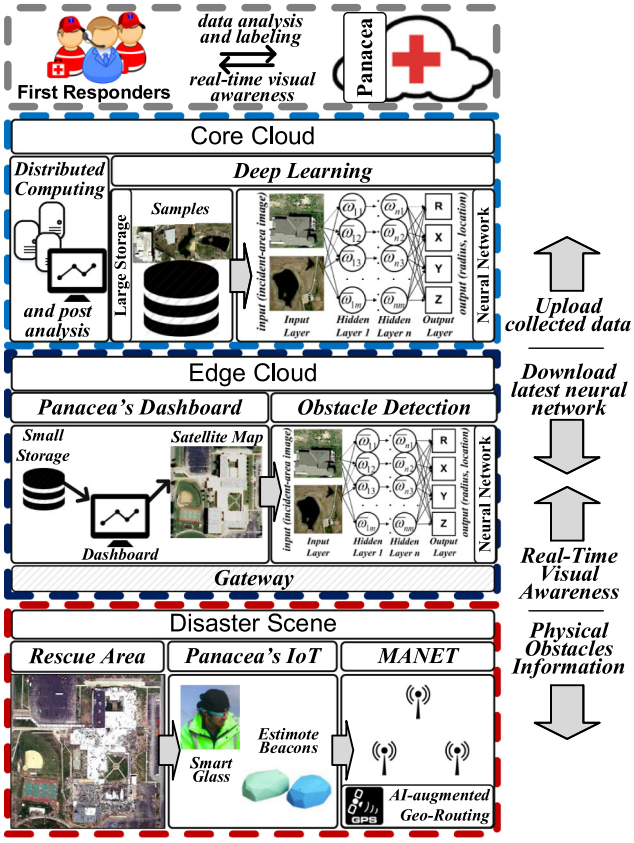


Fig. 4. To cope the deep learning complexity of the obstacle detector we are moving deep learning to the core cloud. The up-to-date detector version then can be pre-uploaded to the edge cloud and used *off-line* during disaster-incident response activities within a lost infrastructure region to enhance geographic routing.

plane, with limited radio range. Let us also assume that our wireless network is static and does not have any obstacles (voids). In a greedy forwarding algorithm, nodes need to be aware of the (euclidean) coordinates of the destination as well as of all their neighbors [11,13,15–17]. Packets are then forwarded to the neighbor closest to the destination.

We model greedy forwarding based on an analogy from the electrostatics literature: specifically, we model a packet as a positive electric charge (or test charge) and its movement from the source with an electrostatic field created by the destination (with point negative charge). Packets are forwarded towards lines of force of the electrostatic field till they reach the destination. The potential field φ generated by the negative charge at the destination on the test charge is modeled as:

$$\varphi = -\frac{Q}{r} \tag{1}$$

where r is a distance between the node which currently holds a packet and the charge located at the destination, and Q represents the intensity of such a charge. Following the laws of electrostatics, each node forwards its packets to the neighbor with the lowest electrostatic energy *i.e.*, the node with the lowest electrostatic potential. When there are no obstacles between the source and destination, nodes always forward packets to the node closer to the destination. In this paper, we refer to this potential field generated with Eq. (1) as the *attractive* field.

In presence of obstacles, to generate the repulsive potential, we first ground each obstacle region making its surface equipotential (zero potential) by generating additional charges within this region. We then sum to the main potential described in Eq. (1), an additional potential created by each induced charge. In the rest of this section we first show how nodes would compute a potential for a single grounded spherical obstacle region. We then extend

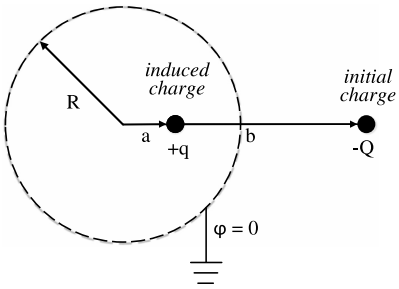


Fig. 5. Electrical images method: a single negative point charge induces a single positive point charge (image) inside a grounded sphere making its surface equipotential (zero potential).

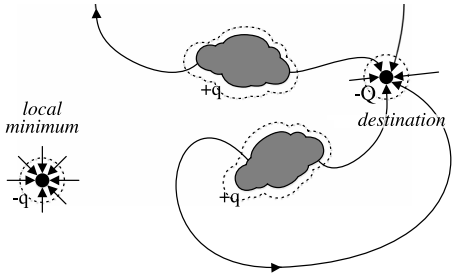


Fig. 6. A potential field has a local minimum if there is a point or a surface which creates an additional negative potential field somewhere else besides the destination, so that the lines of force are directed inwards the local minimum.

our computation to a network with multiple obstacles of arbitrary shapes. We finally show how our proposed *repulsive* field model approximates the shortest path.

4.1. Electrical images solution for a single spherical obstacle

To "ground" a single spherical obstacle we use the electrical images method [21]. In particular, we induce a single positive point charge inside the obstacle as shown in Fig. 5. The potential field generated by such point charge is modeled by the following equation:

$$\varphi = -\frac{Q}{r} + -\frac{q}{|\vec{r} - \vec{b} + \vec{a}|} \tag{2}$$

where $\vec{b}$ is a vector modeling the distance between the center of the sphere and the destination node, and $\vec{a}$ is a vector modeling the distance between the center of the sphere and the induced charge. The value of the induced charge q is then: $q = \frac{R}{b}Q$, where R is the radius of the sphere. The final location of the induced charge $\vec{a}$ is instead: $\vec{a} = \frac{R^2}{b^2}\vec{b}$.

4.2. Green's function solution for obstacles of arbitrary shape

Let us now consider a more general case of a network with multiple obstacles of arbitrary shape. To model the positive potential generated by such scenario, we use the Green's function [21]:

$$G = -\frac{Q}{r} + \chi \tag{3}$$

where χ is the potential induced by the charges on the obstacles within the network graph. We assume the origin of the coordinate system at the destination node. The effect of such a positive potential is then summed to the negative potential field generated by the point negative charge at the destination to generate the line of force, and hence the trajectory of the packet. In this paper, we refer to the potential field generated with Eq. (3) (or its approximation) as the *repulsive* field. Thus, we have the following result:

Fact 1. *Given a network with a system of grounded conductors and a single point charge at the destination, we can emulate a unique potential field without a local minimum.*

Proof. Let us assume by contradiction that we have a local minimum created by the Green's function potential field for several grounded obstacle surfaces and a negative point charge located at the destination. Moreover, let us assume by contradiction that this potential field is also not unique. Having a local minimum implies that all lines of force have to be directed inwards of the local minimum, which is possible if and only if there is a point or a surface that creates an additional negative potential field besides the destination (see Fig. 6). From the Green's Reciprocation Theorem [21] we derive the closed-form expression for a potential between any point a located on the grounded surface and a single point charge located at the destination as follows:

$$\varphi(a) = -\oint_S \varphi_S \vec{\nabla} G(a)\, dS \tag{4}$$

where φ_S is the potential on the surface S and φ is the potential generated by a point charge located at the destination. Note that Eq. (4) can be rewritten in a discrete form as: $\varphi(a) = -\sum_{j=1}^{n} \varphi_{jS} q_{jS}$. From Eq. (4) and its discrete form we conclude that the potential field created by some obstacle of arbitrary shape has always an opposite sign of the potential field created by the point charge. This means that an additional potential field is always positive when the point charge located at the destination is negative. In addition, based on the Green's function Uniqueness Theorem [21] such solution is unique. ■

Corollary 1. *Repulsive greedy forwarding always avoids local minimum.*

Proof. Within a repulsive field, each node forwards packets to the neighbor with minimum emulated potential energy. Based on Fact 1, the line of force of such potential function can lead only to the destination, or to infinity in case of a disconnected network. Hence, if a path between the source and the destination exists, then packet delivery by *repulsive* greedy forwarding is guaranteed. ■

4.3. Path stretch approximation bound

The goal of this subsection is to show a bound on the path stretch obtained using *repulsive* greedy forwarding. To this aim, we first show that the maximum path stretch arises only in presence of a single spherical obstacle; we then estimate such stretch upper bound using properties of the electrostatic potential field.

Fact 2. *The maximum path stretch of repulsive forwarding arise when both source and destination are located at the two extreme points of a single obstacle's diameter.*

Proof. Since the potential field created by an obstacle is inversely proportional to the distance r to this obstacle (Eq. (2)), the closer the shortest path lies to an obstacle region, the greater the strength of the obstacle's potential field on such packet. This means that the path stretch is as high as the path gets closer to the obstacle. In the worst-case, the shortest path length equals to half of the obstacle's perimeter, *e.g.*, when the source and the destination are located on the opposite sides of the obstacle's diameter. From the Green's function [21] we know that the potential field generated

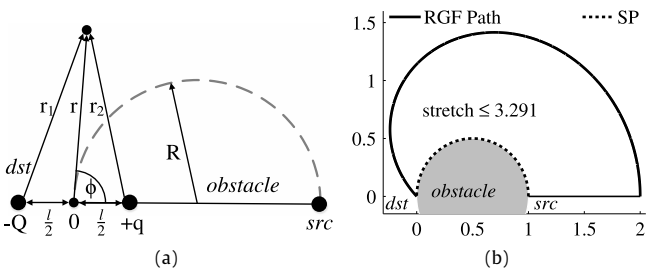


Fig. 7. (a) Potential field in presence of a single grounded sphere in polar coordinate system: when the destination locates very close to the surface of a sphere, the induced charge creates a potential field of a point dipole. (b) Using electrostatic potential field properties and numerical calculations we estimated the worst-case *repulsive* greedy forwarding (RGF) stretch of the shortest path (SP) as $\approx 3.2907221 \pm 10^{-7}$.

by the grounded obstacle is directly proportional to its volume. This means that the greater is the obstacle's volume (or area) and the shorter is its perimeter, the higher is the possible path stretch. Finally, the more obstacles we have in the forwarding region, the weaker will be their potential fields due to secondary induced charges, since the obstacle potential fields weaken each other. Due to this fact and based on the isoperimetric property of a sphere (circumference) [43], we conclude that *repulsive* greedy forwarding shows a maximum path stretch only in the presence of a single spherical obstacle, when both source and destination are located at opposite sides of the obstacle's diameter. ■

Corollary 2. *Repulsive forwarding has a 3.291 path stretch approximation bound with respect to the shortest path.*

Proof. Based on Fact 2, the *repulsive* greedy forwarding maximum path stretch arise when a single spherical obstacle has both source and destination located at opposite sides of its diameter. The Green's function (see Eq. (3)) of this obstacle is $G = \frac{q}{r_1} - \frac{Q}{r_2}$ as shown in Fig. 7(a), where Q is a point charge located at the destination; $q = \frac{RQ}{R+\frac{l}{2}}$ is a point induced charge located within the obstacle (see Eq. (2)), and l is the distance between them. Since $l \ll r$, using the cosine theorem and the Taylor series expansion on l we have:

$$\begin{cases} r_1 = \sqrt{r^2 + \frac{l}{4}^2 - rl\cos\phi} = r - \frac{l}{2}cos\phi + O(l^2) \\ r_2 = \sqrt{r^2 + \frac{l}{4}^2 + rl\cos\phi} = r + \frac{l}{2}cos\phi + O(l^2) \end{cases} \tag{5}$$

Similarly, q can be written as $q = Q - \frac{l}{2R}Q + O(l^2)$, and hence, using Eq. (5) the Green's function G in its first l approximation can be rewritten in polar coordinates as:

$$G(r,\phi) = \frac{Q - \frac{l}{2R}Q}{r - \frac{l}{2}cos\phi} - \frac{Q}{r + \frac{l}{2}cos\phi} = \frac{Ql(cos\phi - \frac{r}{2R})}{r^2} \tag{6}$$

Note that when $r \ll R$, the potential field in Eq. (6) becomes a potential field of a point dipole. Moreover, on the obstacle's circumference (*e.g.*, when $r = 2Rcos\phi$), Eq. (6) equals to 0, which is also expected since a grounded conductor (an obstacle) has a zero-potential surface.

Initially, packets are repelled away from the obstacle along the source–destination line until they reach the minimum potential point, which we can find by differentiating Eq. (6) with respect to r and subsumed $\phi = 0$ as following:

$$\frac{dG(r,0)}{dr} = -\frac{2Ql}{r^3} + \frac{Ql}{2Rr^2} = 0 \tag{7}$$

Solving Eq. (7) gives us $r = 4R$, and hence, initially packets are repelled away from the obstacle on $L_0 = 2R$ distance. If we model a path with the closed-form of packet trajectory obtained by *repulsive* greedy forwarding, we can find its residual length from the gradient symmetry of the electrostatic potential field [21]:

$$\frac{dr}{\nabla_r G(r,\phi)} = \frac{rd\phi}{\nabla_\phi G(r,\phi)} \tag{8}$$

Further, by subsuming Eq. (6) to Eq. (8) we obtain the following differential equation:

$$\dot{r} + r\frac{1}{2R\sin\phi} - 2ctg\phi = 0 \tag{9}$$

where $\dot{r} = \frac{dr}{d\phi}$. Moreover, $r(0) = 4R$ (see Eq. (7)).

As both shortest path and *repulsive* greedy forwarding path lengths are proportional to the obstacle's size, without loss of generality we can assume unitary obstacle diameter. Hence, solving Eq. (9) with $R = \frac{1}{2}$ gives us the following packet trajectory closed-form:

$$r = \frac{2\phi}{tg\frac{\phi}{2}} - 2 \tag{10}$$

The maximum path stretch is then equal to $\frac{L}{\pi R} = \frac{2l}{\pi}$ where $L = \int_{\phi_1}^{\phi_2}(r^2 + \dot{r}^2)^{\frac{1}{2}}d\phi + L_0$ is a length of the found trajectory. Note that $\phi_1 = 0$, and $\phi_2 \approx \frac{3\pi}{4}$ which can be found by subsuming $r = 0$ to Eq. (10), *i.e.*, by solving $\phi = tg\frac{\phi}{2}$.

Solving the integral using numerical calculations, we estimated the length of the trajectory to be $L = 5.1690541 \pm 10^{-7}$. The maximum path stretch is then $\approx 3.2907221 \pm 10^{-7}$ (see Fig. 7(b)). We then conclude that — *repulsive* greedy forwarding is an approximation algorithm for the shortest path. ■

In the rest of this paper, we first present a practical algorithms which incorporates *repulsive* field model. Using numerical and event-driven simulations, we then show how a path stretch minimization with proposed algorithms improves an overall network goodput.

5. Practical repulsive forwarding

In the previous section, we have shown how to use physical obstacle information within geographic routing to avoid the local minima by greedy forwarding packets as if they are hypothetically immersed in a potential field generated by a point charge at the destination. Obstacles were modeled as conductors with zero potential surface resulting in additional repulsing fields. Herein, we describe how we approximate the Green's function to capture the potential field generated by multiple randomly shaped obstacles. We first compute our approximation assuming global knowledge of all obstacles at every node. Next, we extend our solution to a local knowledge case: nodes are only aware of obstacles present at some specified distance called "repulsion zone". Then, we discuss how our forwarding algorithm can be coupled with existing stateless local minimum recovering scheme called "Pressure" proposed in [11] for a guarantee delivery.[2]

Why do we need an approximation? Our repulsion field builds upon the continuous $\mathbb{R}^2$ (or $\mathbb{R}^3$) plane; a discrete node distribution instead creates discontinuity of potential fields which can form artificial local minima. For example, nodes may be scattered in a way of creating artificial voids (i.e., not radio-covered space without physical obstacles). Moreover, to find a closed-form of the potential induced on every node on the network, the Green's

[2] As we show later guaranteed delivery is possible only if there is no path length limit such as e.g., Time To Live (TTL).

function requires an integration over the obstacles' surface with respect to every other obstacle, including the impact of secondary charges. Solving such an integral (at every forwarding decision) may be unfeasible, due to complex obstacle shapes or by the dynamic nature of the network. The mirror effect further complicates the computation of the repulsive potential field induced by the obstacles: when an object is located between two mirrors, infinite number of images (induced charges) appear. Similarly, secondary charges recursively induce progressively weaker additional fields, to be included in the Green's function solution.

Next, we show the impact of the secondary charges by approximating any obstacle shape to a circle (a sphere) with a point charge located on its center of mass. For our approximation of the obstacle electrostatic field we require:

Sign: the potential induced by each obstacle region has to be positive, *i.e.*, each obstacle is replaced with a positive point charge to repulse a packet.

Direction: the potential induced has to be greater inside, and smaller outside than a potential field created by a negative point charge located at the destination. This is necessary to correctly drive the packet in direction of the destination.

Intensity: the total potential field must be equal to 0 at ∞.

The above requirements allow the greedy forwarding algorithm to use a gradient descent on the repulsion field to converge to the destination by forwarding packets to the neighbor whose electrostatic energy is minimum. We describe the local minima recovering mechanism in Section 5.3.2.

Obstacles' shape approximation. We approximate the potential field of an obstacle by circumscribing it to a circle (or sphere) j to capture the worst path stretch it can cause (see Fact 2). Having the set of obstacle's pixels the detector (see Section 3) computes the center and the radius of the circumscribing circle. We locate the center of the circle with the center of mass of such an obstacle. To locate the coordinates of the center of mass C_j we average the coordinates of the N pixels S_i of the obstacle as follows:

$$\begin{cases} x_{C_j} = \dfrac{\sum_{i=1}^{N} x_{S_i}}{N} \\ y_{C_j} = \dfrac{\sum_{i=1}^{N} y_{S_i}}{N} \end{cases} \tag{11}$$

We then assign to the radius of the circumscribing circle R_j, the distance between the center C_j and the furthest (border) pixel, which can be computed as follows:

$$R_j = \max_{i=1,N} dist(C_j, S_i). \tag{12}$$

Remark. Note that instead of using an AI-based obstacle detectors over satellite imagery at edge clouds, the proposed *Repulsive* field can be used in conjunction with existing dynamic obstacle detection algorithms (e.g., localization techniques based on node's signal strength) running in MANETs [44]. To this aim, Eqs. (11) and (12) can use coordinates of obstacle's border nodes. The cons of a dynamic obstacle detection is the overhead as well as the dependency on the network awareness, which can be aggravated by the geographic routing performance under node failure and mobility conditions.

Obstacle's Potential Approximation. To neglect the impact of self-induced charges we approximate the potential field of the obstacle with a decaying potential whose strength magnitude is a parameter n. To approximate the potential field of the circumscribed circle, we place a positive point charge q_j in the obstacle center C_j. We

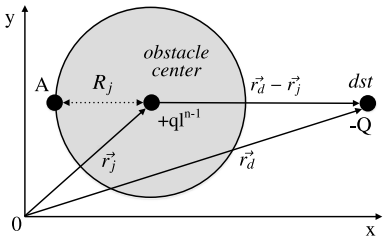


Fig. 8. Use of border conditions in A to find values $q_j l^{n-1}$.

approximate the potential field generated by the obstacle with a potential whose strength diminishes with the r_j^n:

$$\varphi_j = \frac{q_j l^{n-1}}{r_j^n} \tag{13}$$

where l^{n-1} is a normalizing constant with units of length elevated to the power of $n-1$. To minimize the impact of the secondary induced charges in presence of several obstacles, we could use $n \geq 1$. Note how a potential of a negative point charge located at the destination always depends merely on the distance r_d ($n = 1$).

5.1. Computing electrostatic potential with obstacles global knowledge

Let us assume that all nodes are aware of both the center and the radius of each obstacle j, $\langle C_j, R_j \rangle$. This assumption is suitable for routing schema in which the global knowledge of the topology is available, or in the case of a static known man-made or natural obstacle, such as a building or a pond. With such information, we can compute the electrostatic potential at any node e as:

$$\varphi_e = -\frac{Q}{|\vec{r_d} - \vec{r_e}|} + \sum_{j=1}^{M} \frac{q_j l^{n-1}}{|\vec{r_j} - \vec{r_e}|^n} \tag{14}$$

where $\vec{r_d}$ is a radius vector directed towards the destination, $\vec{r_j}$ is a radius vector directed towards the obstacle's center C_j, and $\vec{r_e}$ is a radius vector directed towards the node e. To compute $q_j l^{n-1}$ we equate the electrical fields created by the obstacle j and the destination d on the obstacle's border, so that they are equal in magnitude and opposite in direction, *i.e.*, $\vec{\nabla}\varphi_d = -\vec{\nabla}\varphi_j$, where:

$$\begin{cases} \vec{\nabla}\varphi_d = -\dfrac{Q}{|\vec{r_d} - \vec{r_A}|^3}(\vec{r_d} - \vec{r_A}) \\ \vec{\nabla}\varphi_j = \dfrac{n q_j l^{n-1}}{|\vec{r_j} - \vec{r_A}|^n + 2}(\vec{r_j} - \vec{r_A}) \end{cases} \tag{15}$$

From Eq. (15) we derive the dependence of $q_j l^{n-1}$ by Q (see Fig. 8):

$$q_j l^{n-1} = \frac{Q R_j^{n+1}}{n(|\vec{r_d} - \vec{r_j}| + R_j)^2} \tag{16}$$

5.2. Computing electrostatic potential without global knowledge of obstacles' location

In this section we describe a simple method to compute forwarding decisions based solely on local knowledge of obstacles' location. This is motivated by the need to reduce network overhead and to optimize memory and storage needed to deal with the propagation and storage of obstacles information (especially in case of large-scale networks). In our evaluation section, we quantify how such local knowledge is enough for a performant greedy forwarding strategy with minimum path stretch. Note how, since

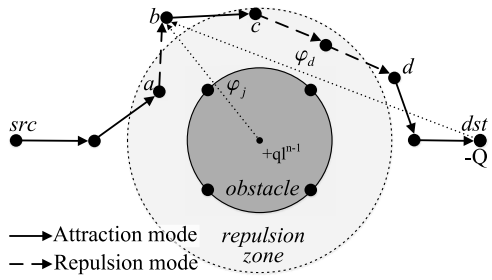


Fig. 9. Attractive/Repulsive Greedy Forwarding (ARGF) example: ARGF starts in *Attraction* mode until it reaches the *repulsion zone*; at node *b* (located outside the repulsion zone), ARGF returns back to its *Attraction* mode. ARGF can again switch to the *Repulsion* mode with guaranteed progress towards destination. At node *d*, ARGF returns to the *Attraction* mode and continue forwarding in this mode to destination.

the electrostatic field intensity diminishes with distance, when computing the potential, forwarding nodes do not need to consider obstacles "far away".

To find the distance at which the obstacle's contribution to the total potential field is still significant, *i.e.*, the obstacle's *repulsion zone* R_z, we first decompose the second term of Eq. (14) with a Taylor series expansion obtaining:

$$f(R) = \frac{q_j l^{n-1}}{R^n} - \frac{n q_j l^{n-1}}{R^{n+1}}(r - R) + O((r - R)^2) \tag{17}$$

and then we neglect the first two terms of higher order and substitute R_z to r in Eq. (17). The radius of repulsion zone is therefore:

$$R_z = \left(1 + \frac{1}{n}\right) R \tag{18}$$

Thus, nodes need only to store obstacles located closer than R_z. Once a node moves further away from an obstacle's R_z, it can remove its states concerning that obstacle. On contrary, when a new node arrives within the obstacle's repulsion zone R_z, the information about such obstacle should not be used. As we show in Section 6, such information can be exchanged among neighbors during their regular position beaconing communication with negligible additional overhead (*i.e.*, using the same packet).

5.3. Greedy forwarding algorithms

Note how, due to discrete node distribution and since we approximate the Green's function of the real obstacles to a potential of spherical regions, Eqs. (14) and (16) do not guarantee avoidance of a local minimum. To this end, we first propose a solution that alternates forwarding in both attractive and repulsive fields, *viz.* Attractive Repulsive Greedy Forwarding (ARGF), to increase the chance of escaping or avoiding all local minima and deliver a packet with the minimum path stretch. To guarantee 100% packet delivery, we then extend ARGF with stateless Pressure local minimum recovery scheme proposed in [11]. The resulting Attractive Repulsive Pressure Greedy Forwarding (ARPGF) applies recovery when in pressure mode to both attractive and repulsive fields (details in Section 5.3.2).

5.3.1. Attractive/Repulsive Greedy Forwarding

Algorithm 1 outlines how each node forwards packets using the Attractive Repulsive Greedy Forwarding (ARGF) strategy, when either local or global information about obstacles are known. Fig. 9 illustrates the following forwarding process: upon receiving a packet to be forwarded, node *e* checks if the packet should proceed further in *Repulsion* mode or in *Attractive* mode. To this end, ARGF first check that *e* in *repulsion zone* R_z and that previously found potential in *Repulsion* mode attached to the packet P_φ_{rep} (which is initially set to max value) is greater than current potential of *e* (line 3). This statement is important to ensure that packets progress in *Repulsion* mode towards the destination without possibility of returning to previous local minimum of repulsive field. Thus, if $P\_\varphi_{rep} > \varphi_{total}(e, P, e.\vec{C}, e.\vec{R})$, *e* stores its potential in packet header (line 4) and computes neighbors *Nbrs* potential φ_{total} (line 5) using its information about known obstacle centers $e.\vec{C}$ and their radius $e.\vec{R}$ via Eqs. (14) and (16).

If no neighbor *n* found has a potential φ_{total} lower than node *e* potential, or if *e* is not in R_z, *e* switches to an *Attraction* mode, where it computes its neighbors' potential φ_d using Eq. (1) (line 12). If both *Repulsion* and *Attractive* modes are unavailable to find next hop, ARGF returns a potentially detected obstacle condition and terminates.

Algorithm 1: Attractive/Repulsive Greedy Forwarding

```
   /* Upon receiving a packet P at node e                                  */
 1 if e ≠ dst then
 2     next ← NIL
 3     if e ∈ R_z and P_φ_rep > φ_total(e, P, e.C⃗, e.R⃗) then
           /* Repulsion mode                                               */
 4         P_φ_rep ← φ_total(e, P, e.C⃗, e.R⃗)
 5         n ← argmin_{n∈Nbrs(e)} φ_total(n, P, e.C⃗, e.R⃗)
 6         if φ_total(n, P, e.C⃗, e.R⃗) < φ_total(e, P, e.C⃗, e.R⃗) then
 7             next ← n
 8             forward(P, next)
 9         end
10     end
11     if next == NIL then
           /* Attraction mode                                              */
12         n ← argmin_{n∈Nbrs(e)} φ_d(n, P)
13         if φ_d(n, P) < φ_d(e, P) then
14             next ← n
15             forward(P, next)
16         else
17             exception ("ARGF faced a local minimum")
18             alert ("Potentially unknown obstacle detected")
19             terminate
20         end
21     end
22 else
23     terminate
24 end
```

5.3.2. A/R Pressure Greedy Forwarding

Although empirically (as we show in Section 6) our ARGF outperforms traditional Greedy Forwarding in terms of packet delivery, it does not theoretically guarantee 100% packet delivery. To this end, we devised Attractive Repulsive Pressure Greedy Forwarding (ARPGF), which builds upon a known *Gravity-Pressure* scheme that has been shown to provide guarantee packet delivery [11].

Algorithm 2 outlines how each node forwards packets using Attractive Repulsive Pressure Greedy Forwarding (ARPGF): the algorithm also starts by alternating *Repulsive* and *Attractive* fields in ARGF, when it needs to forward packets. However, similarly to the last known potential of *Repulsion* field φ_{rep}, it also saves the last known potential in the *Attractive* field variable φ_{attr} (line 12) to ensure in a possibility of the progress in *Attractive* mode *e.g.*, after resuming from *Pressure* mode. When both *Attractive* and *Repulsive* forwarding fail to find next node for forwarding, ARPGF switches to *Pressure* mode (line 21).

The key idea behind recovery in *Pressure* mode is to forward packet to the closest to the destination neighbor among the least visited neighbors (line 23).

Algorithm 2: Attractive/Repulsive Pressure Forwarding

```
   /* Upon receiving a packet P at node e                                  */
1  if e ≠ dst then
2      next ← NIL
3      if e ∈ R_z and P_φ_rep > φ_total(e, P, e.C⃗, e.R⃗) then
           /* Repulsion mode                                               */
4          P_φ_rep ← φ_total(e, P, e.C⃗, e.R⃗)
5          n ← argmin_{n∈Nbrs(e)} φ_total(n, P, e.C⃗, e.R⃗)
6          if φ_total(n, P, e.C⃗, e.R⃗) < φ_total(e, P, e.C⃗, e.R⃗) then
7              next ← n
8              forward(P, next)
9          end
10     end
11     if next == NIL and P_φ_attr > φ_d(e, P) then
           /* Attraction mode                                              */
12         P_φ_attr ← φ_d(e, P)
13         n ← argmin_{n∈Nbrs(e)} φ_d(n, P)
14         if φ_d(n, P) < φ_d(e, P) then
15             next ← n
16             forward(P, next)
17         else
18             alert ("Potentially unknown obstacle detected")
19         end
20     end
21     if next == NIL then
           /* Pressure mode                                                */
22         visits_min ← min_{n∈Nbrs(e)} P_visits(n)
23         Candidates ← {n ∈ Nbrs(e) and P_visits(n) == visits_min}
24         n ← argmin_{n∈Candidates} φ_total(n, P, e.C⃗, e.R⃗)
25         P_visits(n) ← P_visits(n) + 1
26         next ← n
27         forward(P, next)
28     end
29 else
30     terminate
31 end
```

6. Evaluation results

In this section, we establish the practicality of our electrostatics-based approach by evaluating its performance in several scenarios that result into the following salient findings:
(i.a) *Local obstacles knowledge is enough*. Our repulsive greedy forwarding approach is not affected by a lack of global knowledge on obstacles' position.
(i.b) *Local obstacles introduce negligible storage and no network overhead*. To maintain a local knowledge on obstacles, our routing protocols only requires $<$ 0.25 KB of storage space, and hence, that information can be piggybacked and propagated with the keep-alive beaconing message to update their position at no (or negligible) additional network overhead.
(ii) *Our ARPGF outperforms related stateless greedy forwarding solutions* [11,13] *in terms of delivery ratio, and the required information to run it, which can fit in available IP packet header space with* 99% *probability (i.e., its overhead is extremely low).*
(iii) *The repulsive field (and hence both ARGF and ARPGF) improve network's goodput in challenged disaster incident wireless edge networks.*[3] By reducing a path stretch due to a physical obstacles knowledge, ARGF (and hence ARPGF) results into a higher network throughput than related solutions. The first two results emerge from our numeric simulations, while we found our third result analyzing using more realistic ns-3 event-driven simulations (see Sections 6.1 and 6.2).

[3] Improvement were observed when both coordinates and radius of physical obstacles are known (see Section 6.2).

6.1. Performance tuning under static obstacles of complex concave shapes

Simulation Settings. Our Java-based simulation environment is composed by an Ubuntu OS GNU/Linux *x*86_64 machine with an *Intel(R) Xeon(R)* processor with CPU 2.1 GHz and 1 GB RAM. We generate a 1 km^2 area and place nodes into each 10 $\times$ 10 m cell (for a total of 10K nodes). To remove the "Unit Disk"" graph assumption, we set the radio range of each node from 50 to 40 m, unless stated differently. We then applied the random graph generation model $G(n, 1-p)$ [11,17] with probability $p = 0.05$. With this parameters when two nodes are within the reciprocal radio range, there is a 5% probability that one of these nodes is not detected by the other. We refer to this condition as lack of symmetrical link assumption, that in turn leads to a network asymmetrical connectivity.

We generate circular obstacles with a radius ranging from 10 to 100 m in random locations. When overlapping, such obstacles create complex concave shapes, which stress greedy forwarding to the limit [16]. We run our simulations with 0, 10, 30, 50 and 100 obstacles that occupy ≈0%, 10%, 25%, 40% and 60% of the available routing space, respectively.

Remark. Note that some of the recent similar solutions demonstrate valuable performance degradation after only 30% of obstacles occupancy [17]. In a disaster scenario, this would be common and such performance degradation unacceptable.

After setting up the environment, we attempt to deliver traffic among 1000 random pairs $\langle src, dst \rangle$. In this scenario, our main goal is to stress our greedy forwarding algorithms with obstacles of complex concave shapes, and hence, we do not use node mobility, as it leads to a frequent network partitioning under high obstacle occupancy which hides greedy forwarding algorithms' potential.[4] For the same reason, we do not generate obstacles at the area edges. All our results show 99% confidence interval over 50 trials, and our randomness lies in both the source–destination pairs and the formed network topologies.

Comparison Methods and Metrics. To empirically evaluate which potential field best approximates an obstacle of arbitrary shape, we tested the performance of both ARGF and ARPGF under different potential field attenuation orders n. We then leverage our finding ($n = 2$ for ARGF and $n = 1$ for ARPGF) in our other experiments.

We compare our ARGF and ARPGF algorithms with three other stateless greedy forwarding exiting approaches, that is, the original Greedy Forwarding (GF) algorithm (also known as compass routing [12]), a *face routing* algorithm, GPSR [13] and a *gravity pressure forwarding* algorithm GPGF [11]. In addition, we also compare ARGF algorithm coupled with GPSR using a face routing recovery policy, which we call ARPSR. The related solutions are compared across two metrics: the packet delivery success ratio, *i.e.*, the number of delivered packets divided by the total number of attempted delivery (counting only cases in which a path for $\langle src, dst \rangle$ pair exists), and the average path stretch, calculated as the average path length ratio of delivered packets and the shortest paths computed with a simple Breadth-First Search algorithm. Finally, we also compare packet header sizes, which are dynamic for ARPGF and GPGF.

(i.a) Local obstacle knowledge is enough. During the potential field formation w.r.t. the different attenuation order n, we found

[4] Note how mobility does not affect stateless greedy forwarding under the assumption that $t_1 \ll t_2$, where t_1 is time needed to greedy forward a packet to the next hop, and t_2 is time needed for the node to move out of its neighbor radio range. To convince the reader that this is a practical assumption, we later apply (high) mobility in our event-driven simulations.

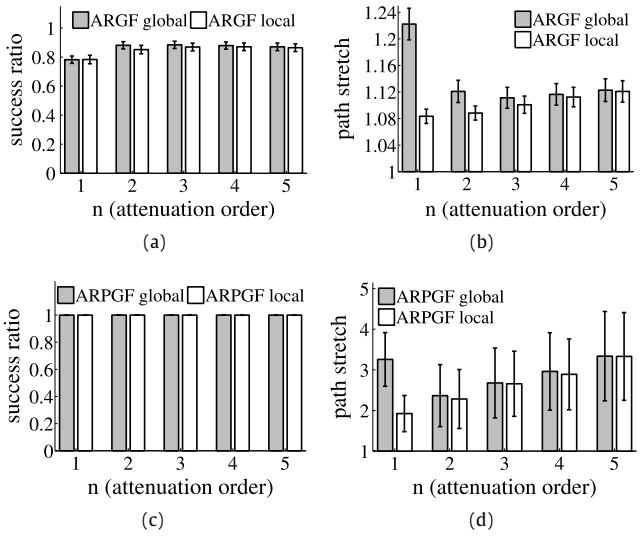


Fig. 10. Ignoring the obstacle mutual electrostatic influence during a potential field computation leads to a lower (a) success ratio and/or (b,d) higher path stretch (attenuation order $n < 2$). Performance of both ARGF and ARPGF does not improve for attenuation order $n \geq 2$.

Table 1
Obstacle data exchange packet payload.

Obstacle ID	Center coordinates	Radius
2·*n* bytes	12·*n* bytes	4·*n* bytes

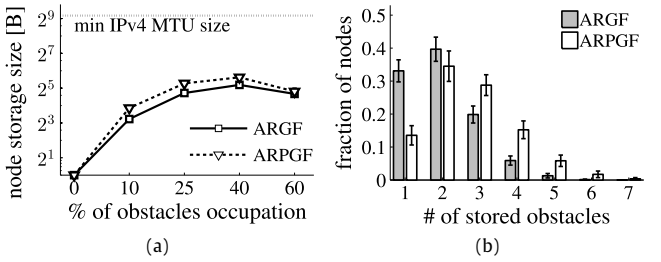


Fig. 11. Local knowledge on obstacles maintenance for both ARGF (with $n = 2$) and ARPGF (with $n = 1$ — the largest *repulsion zone*) requires low storage (a); even for ≤1% of nodes for the (worst-case) 40% obstacles occupation (b) which store up to 7 unique obstacles the storage space needed is $7 \cdot 18 < 256$ B.

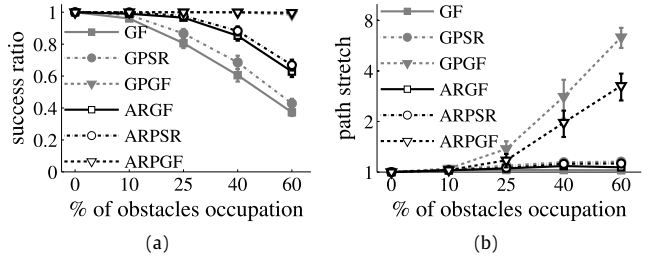


Fig. 12. (a) All *pressure forwarding* algorithms (*i.e.*, GPGF and ARPGF) have a guaranteed packet delivery when there is no TTL policy. Due to the asymmetrical links and no Unit Disk graph guarantees, ARGF (without local minimum recovery) outperforms GPSR — known *face routing* algorithm. These results are confirmed also by our event-driven simulations. (b) Recovering from a local minimum in GPGF and ARPGF may stretch path significantly; however, applying a *Repulsion* field to GPGF (using ARPGF algorithm) shows a halved path stretch.

that based on local obstacles knowledge, both ARGF and ARPGF perform similarly (aside from a better path stretch for $n \leq 2$) to the case when all obstacles are known a priori (Figs. 10(a), 10(b), 10(c) and 10(d)). A path stretch difference under $n \leq 2$ is due to a potential field approximation inaccuracy: we ignore an obstacle's mutual influence by disregarding the secondary induced charges (creating artificial local minima), while local obstacles knowledge mitigates this problem. We can see how both global and local versions of ARGF show the lowest path stretch and the highest success ratio for $n = 2$, which match with the attenuation order of a point dipole, and therefore in line with our theoretical model (see Eq. (6)). At the same time, the local version of ARPGF shows the lowest path stretch when $n = 1$, as in this case we gain the best ratio between secondary induced charges mitigation and the radius of *repulsion zone*, which achieves maximum length decreasing the number of Attractive/Repulsive field alternations during the *Pressure* recovery mode. We use $n = 2$ for ARGF and $n = 1$ for ARPGF results for the rest of experiments.

(i.b) Local obstacles introduce no network overhead. To maintain information about local obstacles for our repulsive field, we need to store them and periodically exchange with the neighbors. The (3D) GPS coordinates of the obstacle location in the worst case (*i.e.*, without converting them to grid coordinates) takes no more than 12 bytes (4 bytes for each coordinate). To store obstacle radius we need no more than 4 bytes. Finally, to distinguish obstacles we can also store its id which takes no more than 2 bytes (up to 65K unique obstacles). To exchange that information we can use the following packet payload structure shown in Table 1. Fig. 11(a) shows how maintenance of the local knowledge for both ARGF (with $n = 2$) and ARPGF (with $n = 1$) on obstacles requires low amount of the node storage space, *i.e.*, even in rear cases of storing 7 unique obstacles as shown in Fig. 11(b) we need $7 \cdot 18 < 256$ B (or < 0.25 KB). Thus, local obstacle information can be exchanged during a periodical node's neighbors beaconing to update their position at no additional network overhead (*i.e.*, within a single packet).

(ii) ARPGF can fit its data in the available IP header space. Our simulations show how all *pressure forwarding* algorithms (*i.e.*, GPGF and ARPGF) have a guaranteed packet delivery when there is no path length restrictions, such as a set time to leave (TTL) — see Fig. 12(a). However, both ARPGF and GPGF lead to large path stretches when obstacles occupy most of the available space, *i.e.*, the ARPGF and GPGF average path lengths are ≈3 and 6 times larger than a shortest path, respectively (Fig. 12(b)). Although *Repulsive* field usage allows ARPGF to have a halved path stretch than GPGF (paths are 2x shorter), these path stretches may force large end-to-end delays and network congestions that may jeopardize applications usage. We can also see how the packet delivery of the *face routing* algorithms (*i.e.*, GPSR and EPSR) degrades due to asymmetrical links and variation of the nodes' radio range, leading to disconnected planar graphs (these results are in line with previous works [37]. Surprisingly, ARGF without a local minimum recovery outperforms GPSR with the local minimum recovery. These results are in line with our event-driven simulations in ns-3 (Section 6.2).

In the last two simulated scenarios, where 100 obstacles are present on ≈60% of the area, we first limit the path length setting different TTL policies (for a maximum of 256 path length) having fixed the average node degree (nodes' radio range ranging from 50 to 40 m). We then set TTL to 128, and we vary the average node degree by reducing an interval of node's radio range distribution by 10 m, until all nodes have a minimal network connectivity radio range of 10 m. As expected, the *repulsion* field usage allows ARPGF to achieve the best packet delivery ≈90% (Fig. 13(a)) that gradually decreases as the network become less dense (see Fig. 13(c)). We

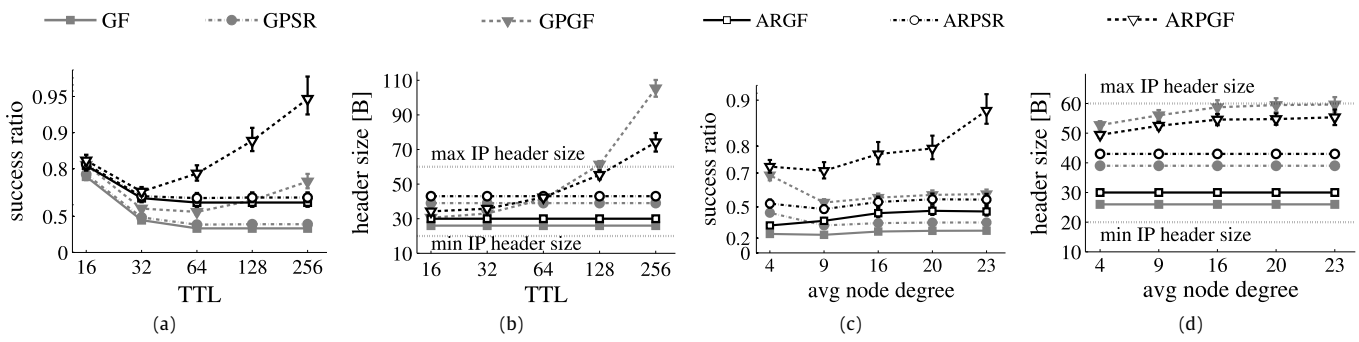


Fig. 13. Success ratio and header size. (a, c) In presence of 100 obstacles (≈60% of obstacles occupancy) and under various TTL path length restrictions as well as for various network densities (with TTL = 128) ARPGF outperforms related greedy forwarding algorithms by showing the highest success ratio. (b, d) Under the same conditions and under some legitimate assumptions, all algorithms may utilize the available space in the IP packet header to use greedy forwarding.

Table 2
ARGF packet header.

Source coordinates	Destination coordinates	Repulsive potential
3 bytes	3 bytes	4 bytes

can also see how in 99% of the cases, both GPGF and ARPGF fits their required data for greedy forwarding in the available IP packet header space, as long as the TTL $\leq$128 (see Figs. 13(b) and 13(d)). We computed the packet header sizes under the following assumptions:

- *2D or 3D node coordinates have total size of 3 bytes;* this can be achieved by *e.g.*, converting GPS coordinates into coordinates of a finite grid which spans regions covered by the network as in [16].
- *Both node ID and number of nodes visits have size of 1 byte.* As the path length is limited by the TTL, we cannot visit more than TTL-1 nodes during the *Pressure* packet recovery phase. Hence, it is possible to find a hash function of *e.g.*, an IP address to map a node's ID between 0 and 255, with a minimum collision probability. When operating in same subnetwork, we can just use the last byte of an IP address as a node ID.

Let us now analyze the overhead of ARGF and ARPGF; with the above assumption, we need the total of 6 bytes (to store source and destination coordinates) + 4 bytes (to store last potential in *Repulsion* mode), so only 10 bytes an extra space for ARGF protocol (see Table 2). ARPGF header size is then 10 bytes (as for ARGF) + 4 bytes (to store last potential in *Attraction* mode) + $2 \cdot n$ bytes (to node visits during *Pressure* recovery) = $14 + 2 \cdot n$ bytes, where n — number of unique node visits (see Table 3). Having 40 bytes of available space in packet header allows ARPGF track up to 13 unique nodes during *Pressure* recovery.

6.2. Incident-supporting application case study results

Simulation Settings. To evaluate the impact of the path stretch on the performance of higher layer protocols under potentially failing and mobile MANET nodes, we compared two stateless greedy forwarding algorithms — *i.e.*, the proposed ARGF and the known *face routing* GPSR protocol [13] using the NS-3 simulator [22]. We have implemented our ARGF protocol by extending the GPSR protocol in the NS-3 with our *repulsive forwarding* mode (see Sections 4 and 5). The details of the GPSR protocol implementation can be found in [45]. Note that in this simulation we do not use the *pressure* forwarding mode to evaluate impact on throughput of the proposed *repulsion* field. To compare with stateful ad-hoc routing solutions, we use the known reactive Ad-Hoc On Demand Distance Vector (AODV) protocol [23]. We also use Hybrid Wireless Mesh Network (HWMP) protocol of 802.11 s standard [24] which combines reactive (with AODV) as well as proactive routing (using the spanning tree algorithm).

We use realistic disaster scenes of damaged by tornado Joplin High School and Joplin Hospital buildings in Joplin, MO (2011) to evaluate the performance of stateless greedy forwarding algorithms under mobility and (severe) node failures. To recreate these disaster scenes, we used the available satellite maps of Joplin, MO, tornado response imagery [40]. In our disaster-incident scenario, we simulate the 5 Mbps high-definition video streaming over a TCP connection from a heads-up display device worn by a paramedic *e.g.*, Google Glass acting as a visual data source.

The paramedic stays for 3 min at each patient location and moves at a jogging speed (2.8 m/s) between these locations. The simulations are designed to cause a geographic routing to face a local minimum when the paramedic source is near the second (first scenario) or third (second scenario) patient locations. Aside from the source mobility, in the node failure simulation scenario (see Fig. 14(a)), nodes around an obstacle can fail for the next 30 s[5] with a probability sampled from the interval [5%, 50%] (from low to severe node failures). Under these failure conditions, the goodput degrades due to losses (*e.g.*, caused by packet collisions) that increase with the path length or path reconstruction of the stateful routing approaches. Note that when nodes fail for continues periods, any "store and forward" solution is inadequate [46,47]. We evaluate the impact of nodes mobility in a second simulation scenario (see Fig. 14(b)), where paramedics can communicate with the gateway only through moving vehicles on the road which speed is sampled uniformly from the interval [5, 20] m/s (from low $\approx$ 10 mph to high $\approx$ 40 mph mobility cases).

Finally, nodes are placed on a grid ranging from 50–150 m step, each node has a radio range of 250 m, and an obstacle (a building) is located approximately in the center of this grid. Each node has roughly 3–10 neighbors for resilience purposes. Table 4 summarizes our simulation details.

(iii) The repulsive field improves a network's goodput. For low node failures (5%), ARGF delivers all packets with a TCP throughput of ≈3 Mbps (Fig. 15(a)). GPSR instead has a 20% failure rate in delivering packets. When GPSR enters the recovering mode it uses planarization which, in turn, can significantly stretch paths. As a result, GPRS shows lower TCP throughput (<2 Mbps) than ARGF due to a lower congestion window size (see Fig. 15(e)), and only 40% of the time it shows similar performance. Under severe

[5] Such behavior is expected due to possibility of an intermittently available power supply, or due to a physical damage caused by rescue workers near the disaster scene.

Table 3
ARPGF packet header.

Source coordinates	Destination coordinates	Repulsive potential	Attractive potential	$Node_{1..n}$ ID	$Node_{1..n}$ visits
3 bytes	3 bytes	4 bytes	4 bytes	n bytes	n bytes

Table 4
Simulation environment settings.

Topology:		**Physical/Link layers:**	
Number of nodes:	30–40	Frequency:	2.4 GHz
Grid placement:	50–150 m	Tx power:	20 dBm
1st obstacle size:	600 × 300 m	Tx gain:	6 dB
2nd obstacle size	400 × 400 m	Rx gain:	0 dB
Radio range:	250 m	Detection threshold:	−68.8 dBm
Avg node degree:	≈3 − 10	Delay prop. model:	CONSTANT SPEED
Overall settings:		Loss prop. model:	TWO-RAY
Node failure period:	≈0.033 Hz	Technology:	802.11g/s
Node failure probability:	0.05–0.5	Modulation:	OFDM
Mobile nodes speed:	5–20 m/s	Data rate:	54 Mbps
Time at each location:	180 s	**Transport/App layers:**	
Src speed:	2.8 m/s	Transport protocol:	TCP
Simulation time:	720–780 s	Payload:	1448 bytes
Beaconing frequency:	1–4 Hz	Application bit rate:	5 Mbps

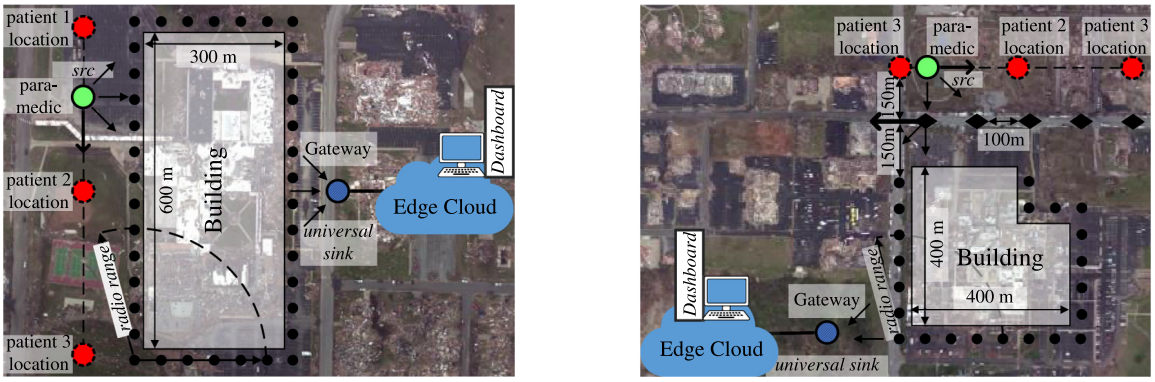


Fig. 14. AI-augmented geographic routing evaluation using recreated disaster scenes of damages by a tornado at the Joplin High School (a) and at the Joplin Hospital (b) buildings in Joplin, MO (2011): a paramedic acts as a source sending data to the gateway (universal sink) over a resilient ad-hoc network. Video streams gathered on-site are sent over a TCP session to the dashboard located in an edge-cloud for further data processing in conjunction with a core cloud. (a), we evaluate our approach under severe failures, (b) under high mobility. We assume that the information regarding a damaged buildings (*e.g.*, its center coordinates and radius) was provided from the edge-cloud through a Gateway using proposed obstacle detector (see Section 3) on pre-uploaded satellite maps.

node failure conditions (nodes fail 50% of the time they receive a packet to forward), we observe similar behaviors (Fig. 15(b)): GPSR experiences lower TCP throughput 40% of the time compared to ARGF, caused again by the congestion window size (see Fig. 15(f)) when GPSR faces a local minimum. Under such severe failures, both GPSR and ARGF fail to deliver packets ≈45% and 35% of the time.

For low node mobility (5 m/s), both ARGF and GPSR deliver all packets with a TCP throughput of ≈1–2 Mbps (Fig. 15(c)). However, when GPSR enters the recovering mode near patient location 3 (after ≈500 s), due to its planarization (which stretches paths), it shows lower TCP throughput (≤1 Mbps) than ARGF. That is in line with a lower congestion window size (see Fig. 15(g)). At the same time, 60% of the time (first 500 s) it shows similar performance. Under high node mobility conditions (20 m/s), we again observe similar behaviors (Fig. 15(d)): GPSR experiences lower TCP throughput 40% of the time compared to ARGF, caused again by the planarization when GPSR faces a local minimum. That is confirmed by congestion window size (see Fig. 15(f)). Under such high mobility, both GPSR and ARGF are still able to deliver all packets, which makes geographic routing more attractive to disaster-incident response activities which benefit from the real-time situational awareness.

Even though both AODV and HWMP have advantages over pure proactive stateful routing solutions, in a challenged disaster scenario they do not show acceptable throughput level, leading to service outages caused by disconnections (from 20% to 90% percent of the time). Recent solutions in stateful greedy forwarding literature can help cope with some disaster incident challenges [16,17]. For example, recent stateful greedy forwarding solutions have shown promising results under severe node failures [16]. However, we found no stateful greedy forwarding algorithm which can cope with both severe node failures and high mobility. The superior performance of ARGF is due to its knowledge about a static physical obstacle located within the disaster scene, which in most cases allows local minima avoidance by using our proposed *Repulsion* forwarding.

7. Conclusion

In this work, we addressed the lack of suitable geographic routing approaches for IoT-based incident-supporting applications in edge computing, that can provide constant high-speed data delivery to an edge cloud gateway to enhance their scalability, reliability

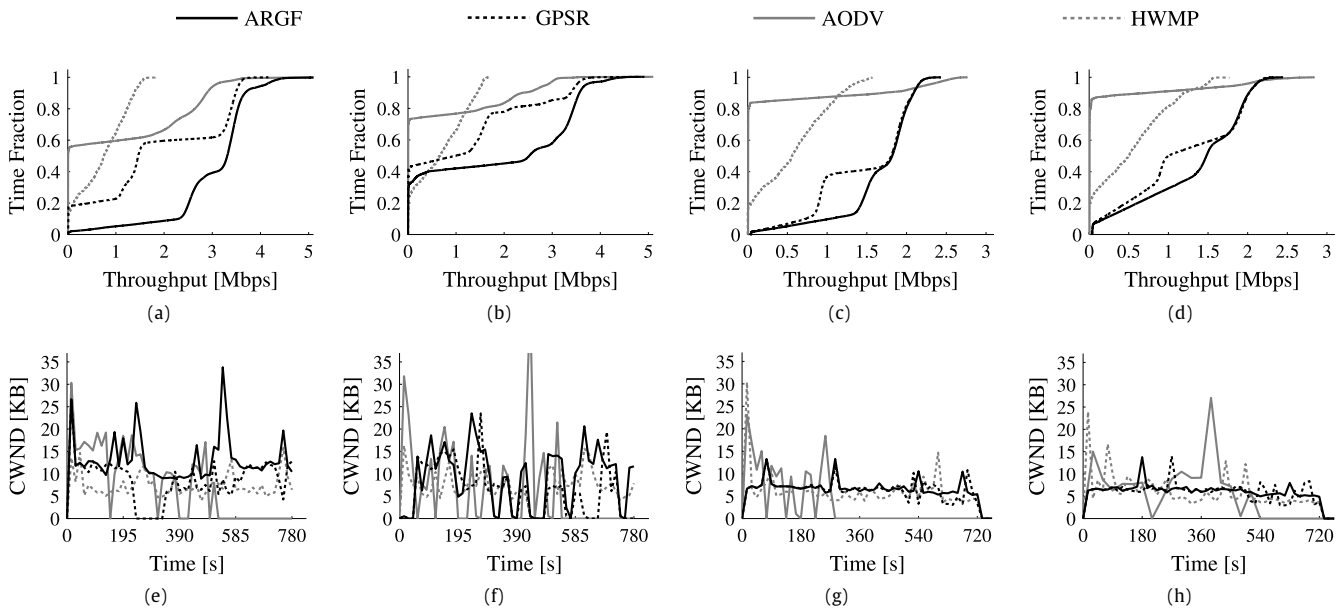


Fig. 15. Time fraction of the TCP throughput (top row) and congested window (CWND) size (bottom row) averaged over 1 and 15 s, respectively, under (a, e) low node failures (5%) and (b, f) severe node failures (50%) and under (c, g) low node mobility (5 m/s) and (d, h) high node mobility (20 m/s): half of the time GPSR faces a local minimum showing two times worth throughput and lower CWND than ARGF due to higher path stretch caused by GPSR planarization. Both AODV and HWMP stateful routing solutions show worse throughput level within a disaster scene, due to its challenging conditions. As expected, performance of all algorithms degrades as we increase node failures or high node mobility.

and stability. Specifically, we presented a novel AI-augmented geographic routing approach (AGRA), which relies on the physical obstacle information obtained from satellite imagery (available at the edge cloud) by applying deep learning. We then proposed a novel *repulsive* field strategy based on electrostatic potential of Green's function to incorporate physical obstacle knowledge within geographic routing. Our approach theoretically guarantees avoidance of a local minima as well as shortest path approximation. Due to inaccuracies in the obstacles' potential field approximation and the discrete node distribution, in practice the approach cannot guarantee the *local minima avoidance*. To this end, we introduced a novel Attractive Repulsive Greedy Forwarding (ARGF) algorithm which can alternately forward packets in both *repulsive* and *attractive* field modes, to maximize the chances of escaping from, or avoiding local minima. Furthermore, to guarantee packet delivery, we coupled our ARGF algorithm with a known gravity pressure recovery algorithm. As emulating both repulsive and attractive fields allows gradient descent to the destination, the recovery schema can be applied to also minimize the path stretch. Using extensive simulations, we have shown that our proposed algorithms outperform related stateless greedy forwarding solutions in terms of packet delivery success ratio and path stretch. Considering an actual incident-supporting hierarchical cloud deployment scenario, we have also analyzed how ARGF has better goodput performance than other stateless *face routing* solutions (such as, GPSR) as well as stateful reactive mesh routing (*i.e.*, AODV and HWMP).

As part of future work, one can plan to apply our Green's function based approach to other known problems in wireless networking. First, the emulated *repulsive* field can be used for traffic engineering by inducing additional charges on heavily-loaded nodes to repulse unbalanced network traffic, and thereby improve the overall network utilization. Second, by inducing additional electrostatic charges in network segments of malicious or selfish behavior, one can improve the overall security. Finally, to improve the overall ad-hoc wireless mesh network vitality, additional electrostatic charges can be induced on nodes with low battery levels. Moreover, our AGRA approach can be also used synergistically with frequency division-based forwarding techniques [28,29] to further improve routing energy efficiency within IoT devices.

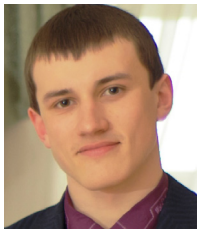
Acknowledgments

This work has been partially supported by the National Science Foundation award CNS-1647084, by the Coulter Foundation Translational Partnership Program, by RFBR according to the research project 16-07-00218a and the public tasks of the Ministry of Education and Science of the Russian Federation (2.974.2017/4.6). Any opinions, findings or conclusions expressed in this publication are those of the author(s) and do not necessarily reflect the views of the funding agencies.

References

[1] N. Fernando, S.W. Loke, W. Rahayu, Mobile cloud computing: a survey, in: Proc. of Future Generation Computer Systems, 2013.
[2] I. Stojmenovic, W. Sheng, The fog computing paradigm: scenarios and security issues, in: Proc. of IEEE FedCSIS, 2014.
[3] D. Chemodanov, et al., Incident-supporting visual cloud computing utilizing software-defined networking, in: IEEE TCSVT, 2016.
[4] H. Trinh, D. Chemodanov, S. Yao, Q. Lei, B. Zhang, F. Gao, P. Calyam, K. Palaniappan, Energy-aware Mobile edge computing for low-latency visual data processing, in: proc. of IEEE FiCloud, 2017.
[5] S. Chung, C. Mario Christoudias, T. Darrell, S. Ziniel, L.A. Kalish, A novel image-based tool to reunite children with their families after disasters, Acad. Emerg. Med. 19 (11) (2012) 1227–1234.
[6] J. Gillis, P. Calyam, A. Bartels, et al., Panacea's glass: Mobile cloud framework for communication in mass casualty disaster triage, in: Proc. of IEEE MobileCloud, 2015.
[7] M. Vassell, O. Apperson, P. Calyam, J. Gillis, S. Ahmad, Intelligent dashboard for augmented reality based incident command response co-ordination, in: Proc. of IEEE CCNC, 2016.
[8] F. Demir, S. Ahmad, P. Calyam, D. Jiang, R. Huang, I. Jahnke, A next-generation augmented reality platform for mass casualty incidents, J. Usability Stud. (2017).
[9] J. Burchard, D. Chemodanov, J. Gillis, P. Calyam, Wireless mesh networking protocol for Sustained Throughput in Edge Computing, in: Proc. of IEEE ICNC, 2017.
[10] A. Sukhov, D. Chemodanov, The neighborhoods method and routing in sensor networks, in: Proc. of IEEE ICWISE, 2013.

[11] A. Cvetkovski, M. Crovella, Hyperbolic embedding and routing for dynamic graphs, in: Proc. of IEEE INFOCOM, 2009.
[12] E. Kranakis, H. Singh, J. Urrutia, Compass routing on geometric networks, in: Proc. of Canadian Conference on Computational Geometry.
[13] B. Karp, H.T. Kung, GPSR: greedy perimeter stateless routing for wireless networks, in: Proc. of ACM MobiCom, 2000.
[14] R. Kleinberg, Geographic routing using hyperbolic space, in: Proc. of IEEE INFOCOM, 2007.
[15] B. Leong, B. Liskov, R. Morris, Geographic routing without planarization, in: Proc. of ACM NSDI, 2006.
[16] M. Krol, E. Schiller, F. Rousseau, A. Duda, WEAVE: Efficient geographical routing in large-scale networks, in: Proc. of EWSN, 2016.
[17] S.S. Lam, C. Qian, Geographic routing in d-dimensional spaces with guaranteed delivery and low stretch, in: Proc. of ACM SIGMETRICS, 2011.
[18] S. Sahhaf, et al., Experimental validation of resilient tree-based greedy geometric routing, in: Elsevier Computer Networks, Vol. 82, 2015, pp. 156–171.
[19] J. Redmon, S. Divvala, R. Girshick, A. Farhadi, You only look once: Unified, real-time object detection, in: Proc. of IEEE Conference on Computer Vision and Pattern Recognition, 2016.
[20] S. Gidaris, N. Komodakis, Object detection via a multi-region and semantic segmentation-aware cnn model, in: Proc. of the IEEE Computer Vision Conference, 2015.
[21] W.K. Panofsky, M. Phillips, Classical Electricity and Magnetism, Courier Corporation, 2005.
[22] T.R. Henderson, et al., Network simulations with the ns-3 simulator, in: Proc. of ACM SIGCOMM Demo, 2008.
[23] C. Perkins, E. Belding-Royer, S. Das, Ad hoc on-demand distance vector (AODV) routing, RFC 3561, 2003.
[24] G.R. Hiertz, et al., IEEE 802.11 s: the WLAN mesh standard, IEEE Wirel. Commun. 17 (1) (2010) 104–111.
[25] T. Taleb, K. Samdanis, B. Mada, H. Flinck, S. Dutta, D. Sabella, On multi-access edge computing: a survey of the emerging 5g network edge architecture & orchestration, IEEE Commun. Surv. Tutor. (2017).
[26] I.F. Akyildiz, W. Su, Y. Sankarasubramaniam, E. Cayirci, A survey on sensor networks, IEEE Commun. Mag. 40 (8) (2002) 102–114.
[27] T. Umer, M. Amjad, M.K. Afzal, M. Aslam, Hybrid rapid response routing approach for delay-sensitive data in hospital body area sensor network, in: Proc. of ACM Conference on Computing Communication and Networking Technologies, 2016.
[28] A.A. Khan, M.H. Rehmani, A. Rachedi, Cognitive-radio-based internet of things: applications, architectures, spectrum related functionalities, and future research directions, in: Proc. of IEEE Wireless Communications, 2017.
[29] F. Akhtar, M.H. Rehmani, M. Reisslein, White space: Definitional perspectives and their role in exploiting spectrum opportunities, in: Elsevier Telecommunications Policy, Vol. 40, No. 4, pp. 319–331.
[30] F. Esposito, Catena: A Distributed architecture for robust service function chain instantiation with guarantees, in: Proc. of IEEE NetSoft, 2017.
[31] C.E. Shannon, Prediction and entropy of printed english, Bell Syst. Tech. J. 30 (1) (1951) 50–64.
[32] M.E. Newman, The structure and function of complex networks, SIAM Rev. 45 (2) (2003) 167–256.
[33] N.T. Nguyen, A.I. Wang, P. Reiher, G. Kuenning, Electric-field-based routing: a reliable framework for routing in MANETs, in: Proc. of ACM SIGMOBILE Mobile Computing and Communications Review, 2004.
[34] M. Kalantari, M. Shayman, Routing in wireless ad hoc networks by analogy to electrostatic theory, in: Proc. of IEEE Communications, 2004.
[35] S. Jung, M. Kserawi, D. Lee, J.K.K. Rhee, Distributed potential field based routing and autonomous load balancing for wireless mesh networks, IEEE Commun. Lett. (2009).
[36] P. Bose, P. Morin, I. Stojmenovic, J. Urrutia, Routing with guaranteed delivery in ad hoc wireless networks, Wirel. Netw. (2001).
[37] Y.J. Kim, R. Govindan, B. Karp, S. Shenker, Geographic routing made practical, Proc. of ACM NSDI, 2005.
[38] J. Zhou, Y. Chen, B. Leong, P.S. Sundaramoorthy, Practical 3D geographic routing for wireless sensor networks, in: Proc. of ACM SENSYS, 2010.
[39] F. Papadopoulos, D. Krioukov, M. Boguna, A. Vahdat, Greedy forwarding in dynamic scale-free networks embedded in hyperbolic metric spaces, in: Proc. of IEEE INFOCOM, 2010.
[40] National Oceanic and Atmospheric Organization. http://storms.ngs.noaa.gov/storms/joplin/. (Last accessed March 2017).
[41] T. Blaschke, Object based image analysis for remote sensing, ISPRS J. Photogramm. Remote Sens. 65 (1) (2010) 2–16.
[42] J. Weston, F. Ratle, H. Mobahi, R. Collobert, Deep learning via semi-supervised embedding, in: Neural Networks: Tricks of the Trade, 2012.
[43] R. Osserman, The isoperimetric inequality, Bull. Amer. Math. Soc. 84 (6) (1978) 1182–1238.
[44] D. Chen, P.K. Varshney, A survey of void handling techniques for geographic routing in wireless networks", IEEE Commun. Surv. Tutor. 9 (1) (2007) 50–67.
[45] A. Fonseca, A. Camoes, T. Vazao, Geographical routing implementation in NS3, in: Proc. of ACM ICST, 2012.
[46] A. Lindgren, A. Doria, O. Schelen, Probabilistic routing in intermittently connected networks, in: Proc. of ACM SIGMOBILE CCR, 2003.
[47] E. Kuiper, S. Nadjm-Tehrani, Geographical routing in intermittently connected ad hoc networks, in: Proc. of IEEE AINAW, 2008.



Dmitrii Chemodanov received his B.S. and M.S. degrees from the Department of Computer Science at Samara State Aerospace University, Russia in 2012 and 2014, respectively. He is currently a Ph.D. student in the Department of Computer Science at University of Missouri-Columbia. His current research interests include distributed and cloud computing, network and service management, and peer-to-peer networks.

Flavio Esposito is an Assistant Professor in the Computer Science Department at SLU and a Visiting Research Assistant Professor in the CS Dept. at University of Missouri, Columbia. He received his Ph.D. in CS at Boston University in 2013, and his M.S. in Telecommunication Engineering from University of Florence, Italy. His research interests include network management, network virtualization and distributed systems. Prior to joining SLU, Flavio worked at Exegy, St. Louis, MO, at Alcatel-Lucent, Italy, at Bell Laboratories, NJ, at Raytheon BBN Technologies, MA, and at EURECOM, France. He was also a visiting researcher at the Center for Wireless Communications, Finland.

Andrei Sukhov is a Professor of Samara National Research University, Russia and was awarded a Ph.D. in Moscow, in Physics and Mathematics in 1993. In 2007 he received Dr.Sc. degree in computer networking at Moscow State University of Electronics and Mathematics (MIEM HSE). Over the last 20 years he has been involved in acting as an investigator for more then 15 telecommunication projects supported by the Russian government, RFBR, INTAS, NATO, ESA, etc. His research area is the computer networks and security.

Prasad Calyam received his MS and Ph.D. degrees from the Department of Electrical and Computer Engineering at The Ohio State University in 2002 and 2007, respectively. He is currently an Assistant Professor in the Department of Computer Science at University of Missouri-Columbia. His current research interests include distributed and cloud computing, computer networking, and cyber security. He is a Senior Member of IEEE.

Huy Trinh received the B.S. degree in Computer Science at University of Missouri, Columbia, USA, in 2015. He is currently a Graduate Research Assistant working toward the M.S degree in the Department of Computer Science, University of Missouri, Columbia, USA. His research interests include image processing, cloud computing and networks.

Zakariya Oraibi received his B.S. and M.S. degrees from the University of Basrah, Iraq in 2007 and 2010, respectively. He is currently pursuing his Ph.D. degree in Computer Science at the University of Missouri-Columbia, and is a recipient of HCED scholarship. His research interests include image processing and machine learning.