**INTRODUCTION:**

Fog Computing, which is also known as Edge Computing, as the ideal paradigm to support the resource-constrained IoT devices. Indeed, Fog Computing, which is not supposed to replace the centralized Cloud but to coexist and cooperate with it, distributes Cloud Computing technologies and principles anywhere along the Cloud-to-Things continuum and particularly at the network edge, in close proximity to the IoT devices. Instead of moving data to the cloud, it may be more efficient to move the applications and processing capabilities closer to the data produced by the IoT. This concept is referred to as data gravity, and fog computing is well suited to address this matter. In the era of Big Data, it may be inefficient to send the extraordinarily large amount of data that swarms of IoT devices generate to the cloud, due to the high cost of communication bandwidth, and due to the high redundancy of data. This research discussed how fog computing can improve multiple aspects in IoT applications.

**6. FOG COMPUTING AND THE INTERNET OF THINGS [6]:**

**a. Characterization of Fog Computing**

Fog Computing is a highly virtualized platform that provides compute, storage, and networking services between end devices and traditional Cloud Computing Data Centers, typically, but not exclusively located at the edge of network. we demonstrate the role the Fog plays in three scenarios of interest: Connected Vehicle, Smart Grid, and Wireless Sensor and Actuator Networks.

**b. Connected Vehicle (CV)**

A smart traffic light system illustrates the latter. The smart traffic light node interacts locally with a number of sensors, which detect the presence of pedestrians and bikers, and measures the distance and speed of approaching vehicles. It also interacts with neighboring lights to coordinate the green traffic wave. The data collected by the STLs is processed to do real-time analytics . The data from clusters of smart traffic lights is sent to the Cloud for global, long-term analytics.

**c. Smart Grid**

Smart Grid is another rich Fog use case. We defer section 4 a discussion of the interplay of Fog and Cloud in the context of Smart Grid.

**d. Wireless Sensors and Actuators Networks**

Most of these WSNs involve a large number of low bandwidth, low energy, low processing power, small memory motes, operating as sources of a sink, in a unidirectional fashion. TinyOS2 is the de-facto standard operating system. Motes have proven useful in a variety of scenarios to collect environmental data. Banka et al stress that emergent applications demand a higher bandwidth, collaborative sensing environment. CASA, a multi-year, multi-partner initiative led by UMASS, deployed a network of small weather radars, integrated with a distributed processing and storage infrastructure in a closed-loop system to monitor the lower troposphere for atmospheric hazards like tornados, hailstorms, etc. Zink et al provide technical details of the deployment. WSNs and WSANs.

**7. ISSUES AND CHALLENGES [7]**

**a. Scenarios of Mobile IoT In A Fog Environment**

The number of mobile devices connected all over the world is everyday increasing at an unprecedented rate. According to our research, global mobile devices were 7.6 billion in 2015, 8.0 billion in 2016 and are predicted to reach 11.6 billion by 2021. Our focus, though, is on a specific subset of mobile devices: the IoT ones. Mobile IoT is an ever-increasing phenomenon and forecasts are definitely impressive. Our research reports says that wearable devices, which are probably the best known example of mobile IoT devices, were 325 million all over the world in 2016 and are expected to become 929 million by 2021. This great amount of mobile IoT devices may take advantage of Fog Computing in order to enable services that can dramatically improve people's lives. We now discuss some scenarios where there is an integration between the IoT and Fog Computing, and in which mobility support is essential.

**b. Citizen's healthcare:**

All these devices are embedded in, or wirelessly connected to, the patient's smartphone, which always runs the frontend application component. The backend component is deployed at the Fog layer instead. It receives the collected data and performs resource-intensive computations in order to extract valuable information from them and determine the actions to be performed. The promptness through which these actions are triggered is of paramount importance and can often make the difference between life and death. The Fog application component is migrated to the patient's smartphone, when she gets across town. Finally, when the patient reaches the hospital, the Fog service is mi-grated to a Fog node installed in a nearby cellular base station.

**c. Drones for smart urban surveillance:**

Drones, also known as Unmanned Aerial Vehicles (UAVs), can be employed in plenty of situations and especially in those too dangerous for humans (e.g., search for a missing person, disaster and emergency management). Indeed, drones do not have a pilot on board and are considerably cheaper than the manned aircrafts. Several companies are investing in drones, and therefore, the global market for them is expected to reach $21.23 billion by 2022. Up to now, drones have been remotely controlled by humans, but a new generation of drones is coming. They are going to be completely autonomous, able to operate without human intervention. All processing required to analyze the collected data and make decisions may be actually performed onboard, but this would negatively affect the drone battery life and thus the overall duration of the flight. These computations are extremely resource intensive indeed, as the collected data to be analyzed are usually video streams and other sensors data, while the actions to be performed are the drone and camera control but sometimes also the action of grabbing an object.

**d. Problem Description:**

As we introduced in our paper [7] , to provide mobility support in a Fog-IoT environment means to preserve the Fog Computing advantages also when the mobile IoT nodes move away from the associated Fog application component. Since the aforementioned advantages are all made possible by the Fog proximity to the end devices, what has to be done is to migrate the Fog application component from one Fog node to another, keeping it close enough to the related mobile IoT nodes. Actually, in the event that the Fog service is stateless, there is no need to migrate it from one Fog node to another, as there is no state to be maintained. Thus, the Fog service can be simply re instantiated on the target Fog node, and the related mobile nodes be redirected to it. At a first glance, keeping the Fog component topologically close to the related IoT nodes might not seem such a hard job. Actually, the Fog service migration problem can be defined as the issue of determining When, Where, and How to migrate the service in order to achieve the desired compromise among all the factors under consideration. With regard to when to migrate, one could propose to continuously migrate the Fog application component, always deploying it to the best possible Fog node, that is the topologically closest to where the IoT nodes are currently. However, it is not convenient to migrate the Fog service too frequently, as each migration may be very CPU- and bandwidth-consuming and implicates a minimum of downtime. **As for where to migrate, it would be ideal to deploy the** Fog component to the Fog node which is topologically closest to the mobile IoT devices. Though, the migration cost towards that node and its current workload might be taken into consideration.

**8. Vehicular Fog Computing [8]**

**a. System Architecture**

A high-level architecture of vehicular fog computing is presented in Fig.1, which comprises three types of entities, namely smart vehicles as the data generation layer, roadside units/fog nodes as the fog layer, and cloud servers as the cloud layer. These servers will obtain the data uploaded by the fog nodes while performing computationally intensive analytics to make optimal decisions from a holistic perspective.

City-level decision

Fog layer

Data generation layer

Cloud Layer

Area-level decision

Data exploitation and analysis

Data fusion and Pre-processing

Data gathering and Pre-processing

Vehicle-level decision

Cloud servers

Roadside units/ fog nodes

Smart Vehicles

Figure 1. Architecture of vehicular fog computing.

**b. Cloud Servers**

Cloud servers provide city-level monitoring and centralized control from a remote location. These servers will obtain the data uploaded by the fog nodes while performing computationally intensive analytics to make optimal decisions from a holistic perspective (e.g., city-level decision).

**c. Smart Vehicles**

Smart vehicles play an important role as the key data generator in a vehicular fog computing system, due to their real-time computing, sensing (e.g., cameras, radars and GPS), communication, and storage capabilities. The amount of data collected by the various sensors in a smart vehicle has been estimated to be around 25 GB/h in a single day (e.g., 20–60 MB/s for cameras, 10 kB/s for radar, and 50 kB/s for GPS). Some of these data can be processed by the smart vehicle itself, in order to inform real time decision making (i.e., vehicle-level decision).

**d. Roadside Units/Fog Nodes**

Roadside units, generally deployed in different areas of a city, can easily be upgraded to act as fog nodes. This will allow the collection of data sent by smart vehicles, processing of the collected data, and reporting of the (processed) data to the cloud servers. These units/nodes also act as the middleware/intermediate devices on the function of a connecting link between the cloud servers and the smart vehicles in a vehicular fog computing system. Unlike existing vehicular networks, these units/nodes will have more functions and provide more diverse services for smart vehicles, such as navigation, video streaming, and smart traffic lights.

|  |  |  |
| --- | --- | --- |
| Application Type | Service | Description |
| Traffic Control | Smart navigation | Plan optimal routes for smart vehicles |
| Smart traffic lights | Schedule traffic lights of each intersection in the city to control traffic flows |
| Driving Safety | Road condition detection | Detect environment information of smart vehicles and make adjustments accordingly |
| Emergency warning | Broadcast emergency warning information to nearby smart vehicles, such as car accidents and work zones |
| Entertainment | Commercial advertisement | Publish advertisements of public interest (e.g., Amber alerts) to nearby smart vehicles |
| Multimedia | Provide multimedia services for smart vehicles, such as music and video |

Table 1. Application examples of vehicular fog computing.

**e. A Fog-Assisted Traffic Control System**

A fog-assisted traffic management system is designed to deliver benefits such as reducing road traffic congestion and car accidents. A typical implementation will consist of two subsystems: one responsible for the local area and one responsible for the global area.

**f.** **Security and Forensic Challenges in Vehicular Fog Computing**

Existing security research mainly focuses on the identification of potential attacks, threats, and vulnerabilities of fog-assisted vehicular applications. An external attacker is not equipped with key materials in a vehicular fog computing system, while an insider attack is one originating from compromised smart vehicles, fog nodes, or cloud nodes that hold the key materials. A passive attack does not destroy the functionality of a vehicular fog computing system but attempts to disclose private information. An active attack is an attempt to deliberately disrupt the operations of a vehicular fog computing system.

**g. Security and Forensic Requirements**

A secure vehicular fog computing implementation should provide the following baseline security and forensic properties. In a vehicular fog computing system, it is critical to meet the integrity requirement since unauthorized modification may result in serious and/or fatal consequences, especially in life-critical vehicular application contexts such as a traffic control system. A number of traffic management algorithms, which may be deployed in a fog-assisted traffic control system, have been proposed in recent years. These include the traffic scheduling algorithms, ITLC and ATL, for controlling the traffic lights of an isolated traffic intersection and the entire road network , and a distributed real-time routing algorithm to avoid traffic congestion.

In general, most of the above-mentioned security requirements can be achieved partly using cryptographic techniques. For example, fully homomorphic encryption primitives can be employed to achieve confidentiality and functionality at the same time. However, most security mechanisms only effectively defend against passive attacks, and there is no foolproof security solution. Once one or more fog nodes have been compromised , for example, to launch attacks within a fog-assisted traffic control system, more sophisticated security mechanisms will be necessary to detect and deter such attacks.

**9.** **FOG COMPUTING: TOWARDS MINIMIZING DELAY IN THE INTERNET OF THINGS [9]**

**a. GENERAL IOT-FOG-CLOUD MODEL**

Fig. 1 illustrates a general framework for an IoT-fog-cloud architecture that is considered in this work. There are three layers in this architecture: things layer, where the “things” and end-users are located, fog layer, where fog nodes are placed, and cloud layer, where distributed cloud servers are located. A *cloud server* can be composed of several processing units, such as a rack of physical servers or a server with multiple processing cores.

In each layer, nodes are divided into domains where a single IoT-fog-cloud application is implemented. For instance, a domain of IoT nodes is shown in dark green, and they communicate with a domain of fog nodes associated with the application. A domain of IoT nodes could comprise things in a smart home, temperature sensors in a factory, or soil humidity sensors in a farm where all the things in the vicinity are considered to be in a single domain. Normally the fog nodes in one domain are placed in close proximity to each other, for example, in a single zip-code or in levels of a building. Each domain of fog nodes is associated with a set of cloud servers for a single application. The basic way in which IoT nodes, fog nodes, and cloud nodes operate and interact is as follows. Cloud nodes process requests and send the response back to the IoT nodes.

4

4

3

2

1

Distributed Cloud layer (core)

Cloud

Cloud Domain 1

Aggregation

4

Fog Layer (Edge)

Fog

Fog Domain 2

Fog Domain 1

Server Rack

Things Layer

Fog Node

Generated Data by Things

Server Rack

Fig. 1. General framework for IoT-fog-cloud architecture. Each layer is partitioned into domains where a single application is implemented

**b. FOG NODE COLLABORATION POLICY**

In this section, we introduce the framework in which fog nodes collaborate with each other to fulfill the requests sent from IoT nodes to the fog layer. However, when the fog node is busy processing many tasks, it may offload the request to some other fog nodes or to the cloud. The concept of Load Sharing is well studied in the literature , , and we borrow similar concepts for the design of the policy by which fog nodes collaborate. In this subsection, we discuss the decision fog nodes make for processing or offloading a task to other fog nodes.  
  
Similarly, a license plate reading request in a recorded video of a vehicle, sent by a traffic camera to fog nodes is an example of heavy processing task. The procedure of processing or forwarding requests by fog nodes is shown in Fig2.

Fig. 2. Policy of Fog node *j* for handling received requests.

Yes

No

No

Yes

All neighbors visited

Estimated Waiting Time

Forward to cloud

Increment of request Forward to best neighbor

Place request in queue Update Waiting Time

Request received

**10. FOG COMPUTING FOR THE INTERNET OF THINGS: SECURITY AND PRIVACY ISSUES [10]**

**Security and Privacy Challenges in IoT**

Although the IoT can play a central role in delivering a rich portfolio of services more effectively and efficiently to end users, it could impose security and privacy challenges. In the following, we summarize the major security and privacy challenges in IoT environments.

**a. Authentication**

These resource-constrained devices can outsource expensive computations and storage to a fog device that will execute the authentication protocol. This model is based on public-key infrastructure using multicast authentication for secure communications. While traditional PKI based authentication could solve the problem, it wouldn’t scale well for IoT systems.

**b. Trust**

Trust models based on reputation have been successfully deployed in many scenarios such as online social networks. Kai Hwang and colleagues proposed a new approach to improve the trust in clouds, which combines security-reinforced data centers, data access, and virtual clusters directed by reputation systems. To design a trust model based on reputation in the IoT, we need to tackle how to maintain the service reliability and prevent accidental failures, handle and identify misbehavior issues, identify malicious behavior correctly, and bootstrap building a trust model based on reputation in large-scale networks.

**c. Rogue Node Detection**

Their approach protects the networks from rogue access points even if the adversaries use customized equipment. A rogue IoT node has the potential to misuse users’ data or provide malicious data to neighboring nodes to disrupt their behaviors.

**d. Privacy**

The privacy leakage of user information in IoT environments, such as data, location, and usage, is attracting the attention of the research community. Another privacy issue is the location privacy that can be used to infer the IoT device’s location. The last privacy issue is the protection of a user’s usage pattern of some generated data by IoT devices, such as in the smart grid. Many privacy-preserving schemes have been proposed in different IoT applications such as smart grids, healthcare systems, and vehicle ad hoc networks. IoT devices limit the techniques that can be used to deliver efficient and effective privacy-preserving schemes.

**e. Intrusion Detection**

Intrusion detection techniques detect misbehavior or malicious IoT devices and notify others in the network to take appropriate actions. Most of the existing techniques in the IoT target a few attacks with low efficiency. Additionally, the complicated design of intrusion detection techniques that meets the limited resources in the IoT is another challenging task.

**REFRENCES:**

[6]Bonomi, Flavio, Rodolfo Milito, Jiang Zhu, and Sateesh Addepalli. "Fog computing and its role in the internet of things." In *Proceedings of the first edition of the MCC workshop on Mobile cloud computing*, pp. 13-16. 2012.

[7] Puliafito, Carlo, Enzo Mingozzi, and Giuseppe Anastasi. "Fog computing for the internet of mobile things: issues and challenges." In *2017 IEEE International Conference on Smart Computing (SMARTCOMP)*, pp. 1-6. IEEE, 2017.

[8] Huang, Cheng, Rongxing Lu, and Kim-Kwang Raymond Choo. "Vehicular fog computing: architecture, use case, and security and forensic challenges." *IEEE Communications Magazine* 55, no. 11 (2017): 105-111.

[9] Yousefpour, Ashkan, Genya Ishigaki, and Jason P. Jue. "Fog computing: Towards minimizing delay in the internet of things." In *2017 IEEE international conference on edge computing (EDGE)*, pp. 17-24. IEEE, 2017.

[10] Alrawais, Arwa, Abdulrahman Alhothaily, Chunqiang Hu, and Xiuzhen Cheng. "Fog computing for the internet of things: Security and privacy issues." *IEEE Internet Computing* 21, no. 2 (2017): 34-42.