Uplink and Downlink Variation in Drone Technology for Cloud, Edge, Fog and Smart Dust Integrated IoT Architecture: Demonstrated Over WSNs

Jayantkumar Dorave¹ and Ritesh Sadiwala²

¹Research Scholar, Department of ECE, R. K. D. F. University, Bhopal, INDIA
²Asst. Prof. Department of ECE, R. K. D. F. University, Bhopal, INDIA

Email: jayant.dorve2805@gmail.com

Abstract:
IoT concepts are heavily applicable in drone communication integrated with different network architecture for optimization. Distributing the burden allows more IoT devices to execute calculations, rather than everything being done on the cloud. There are numerous IoT designs that have emerged as a result of this. By relocating calculations away from the cloud, these designs make use of the enhanced processing capacity of the devices. Based on our needs, we've limited it down to four architectures, each of which we have discussed for optimized flow useful in drone technology. We have also applied the one live dataset for the test drone using raspberry pi processor system powered with for end-to-end drone communication establishment. The analysis of downlink and uplink were studied for time analysis for IoT architecture using drone cell characteristics. New technology makes it possible to implement drone cell (DC) connectivity, which is highly flexible and cost-effective for the gathering of Internet-of-things (IoT) data when terrestrial networks are not yet accessible. DC's flight path has a substantial impact on data collecting systems.

Keywords: IOT, UAV, Drone cells, AWS

1. Introduction
The increase of the Internet of Things (IoT) [1] has introduced a significant communications paradigm change, which allows billions of devices to be connected to robotic systems from home appliances [2]. Over years, IoT devices, among others those that focuses on short-range wireless communications, such as radio, Zig-Be, NFC or Bluetooth, have been proposed with a variety of communication standards and protocols [3]. Increased numbers of IoT (Internet of Things) devices enable IoT network solutions for various cloud service providers. On-demand computer resources, typically in the form of cloud computing, is the idea of cloud computing. Instead of having to manage and keep up with hardware and infrastructure, the user is just concerned with accessing specific resources when the need arises. Microsoft Azure, Google Cloud, and Amazon Web Services have emerged as prominent cloud service providers with their own offerings. They employ continent-spanning data centers to provide its customers more resources. Due to these services being mainly pay-per-use or pay-as-you-go, businesses have been able to adjust quickly and without the need for additional equipment as they develop. Larger businesses may
support considerable computational loads while using very little of their own equipment [5]. Smaller and lighter IoT devices have evolved as processing power in the cloud is outsourcing. Due to this, a new kind of IoT industry has emerged, referred to as the Flying IoT. Computational cloud power is being used by these flying IoT gadgets. The primary goal of this research is to learn how to use a cloud service provider in the building of flying IoT. Nano-technology has given rise to smart-dust architecture which makes the IOT device more robust, small and advanced in network, memory and communication optimization [6].

1.1 Drones:
An aircraft which is controlled without a pilot is a unmanned aerial vehicle (UAV). These pens come in different sizes based on their intended usage. Autonomous drones may be operated by a pilot on the ground or by operating the drone entirely by software [10]. Drones may be utilized in many forms, including transportation of products, inspection and data collecting from agricultural sensors. Drones can be used for various uses. The control of a drone will generally vary depending on the region of the drone, the kind and the size of the drone. Thus, the ability to connect all drones with AWS is not always guaranteed. Many companies choose to use Raspberry Pi (or any other similar microcontroller) in combination with an existing control unit to address this issue.

1.2 Raspberry Pie 3:
We do not have access to real drones in this study. In our tests, we are using a Raspberry Pi 3 instead of a drone and a local node. The computer is a whole computer that is made on a tiny circuit board. All Raspberry Pi devices utilize Broadcom's system on a chip (SoC). This particular model has the ARM1176JZF-S 1 GHz CPU, which is the SoC. It is available with 512MB of RAM and Wi-Fi connectivity options (Bluetooth 4.1 and 2.4GHz). Raspbian is the operating system installed in a SD card.

1.3 Metrics:
When assessing the appropriateness of our selected IoT designs, we will utilize certain metrics we provide in this section. We decided to utilize these indicators since we believe them critical for drone management.

1.4 Network latency:
We examine and transfer the data across multiple sites on IoT networks, edge devices, datacenters and local nodes in IoT systems. Latency is the time to go from one place to another for this information. Latency in one direction is the time it takes to send a packet from one end to another and then receive the identical packet at the other end (OWL) [4]. The latency on the round trip (RTL) is the time to arrive at the finish, and then to get the acknowledgment. The most popular choice is RTL, since you don't have to synchronize two end points when measuring time. To find out whether all four architectures are capable of handling all data kinds, latency will be assessed.

1.5 Packet error rate:
Packet error rate indicates the percentage of packets that reach their destination compared to the total number of packets transmitted. IoT systems frequently exchange a lot of data, therefore unless you pay attention to the quality of the QoS settings, missing packets will not be detected. Errors, even a small number of them, may cause significant real-time data loss, which would be very difficult to recover. For our particular use case, mistakes impact drone control. Messages such as flight directions are crucial to ensure they reach their destinations.

1.6 Data use:
The quantity of data that is sent across a network is called data consumption. To evaluate the level of
dependency on the internet and what is needed for a successful internet connection, this metric takes into consideration factors including use levels, transmission requirements, and dependency on the internet. Internet of Things (IoT) networks are capable of producing huge quantities of data, and because of this, they are highly reliant on network connectivity. The IoT network is a tool that we may use to control this dependence.

1.7 IoT architectures:
Computing and logic are sometimes distributed in IoT networks in many different ways. There are distinct advantages and disadvantages of using an IoT network for air travel. We shall assess the three designs given in this part in this thesis. New communication routes should not emerge when merging the three architectures. Thus, we believe it is sufficient to examine each architecture separately.

1.8 Cloud computing:
On-demand computer resources via the internet are referred to as cloud computing. Whatever is required to keep the organization running: servers, databases, software, and analytics may all be included. By using cloud-based computing resources, it is simple to create and change software tailored to a user's requirements. IoT implies that IoT devices have minimal computational and energy requirements. They just require a network connection and some sensors or actuators to extend their field lifespan. Computational power that is accessible to the device may lead to more complicated and quicker activities being completed. However, the disadvantage is that the gadgets will become dependent on a network connection, which may not always be available [6]. Figure 1 shows the general idea of cloud computing.

1.9 Fog computing:
In 2014, a Cisco employee created the concept "Fog Computing" to construct a sort of cloud extension in a place between "things" and "cloud". In order to accomplish fog computing, computational elements and logic have to be located near to the end devices while leaving the end devices themselves as light as possible. The edge devices use a central node located near the devices, enabling communication between the devices. Because the edge devices are near to the central node, these devices may be lightweight and have fast reaction times. Fog nodes are spread worldwide, with every fog node as a lightweight cloud version [7].

![Figure 1. The idea of Cloud Computing](image-url)
The devices closest to the edge will have their resources closer to them. This capability also enables edge devices to talk to each other via fog nodes. Fog nodes, edge devices, and a network of storage devices work together to offer storage and real-time computing without the need of the cloud. For every kind of fog node, just the necessary resources are required. The possibilities are that it is storage, computation, or some other kind of network resources. It might be a PC, a drone, or any other apparatus. Fog is not meant to replace the cloud in its entirety; instead, it will extract workloads from the cloud. By reducing the usage of network bandwidth and energy usage, this will help lower the amount of traffic that flows between the cloud and edge devices. Figure 2 gives the general idea of Fog-Computing [5].

![Fog computing](image)

**Figure 2** The idea of fog computing.

1.10 Edge computing:

With the number of IoT devices in society now increasing, the pressure on cloud servers and the network that carries data is increasing. Edge computing may be utilized to help ease this issue. Moving processing and logic to the edge device is referred to as edge computing. Devices and device computations and judgements significantly minimize the quantity of data being transferred to the cloud and lessen the strain on the central servers [6][8]. Figure 3 shows the general idea of edge computing.

![Edge computing](image)

**Figure 3** The idea of Edge Computing
The design is appropriate for activities that must be completed in a timely manner. Delay may be broken down into two components: delay due to calculation, and delay due to transmission. The transmission delay is eliminated, because the edge device performs the measurements near the sensors. Using an edge architecture means you are limited to the devices you can use. Edge devices typically have modest amounts of processing power and memory, whereas cloud servers may have unlimited resources. With edge computing, it's possible that different devices will be responsible for their data, thus some devices may have a high load, while others are idle. Computations use a lot of energy; therefore, the life expectancy of the gadget may be reduced if it is operating on a battery. It is necessary to go through the device-by-device software updates if there are any modifications to be performed.

1.11 Smart dust:
Small electronics known as microelectromechanical systems (MEMS) that can sense light, temperature, vibration, magnetism, or chemicals may be used to create Smartdust. Most have wireless connection to a central computer system, and they are dispersed to various locations in order to execute duties, typically by means of RFID technology [9]. A small smart dust communication device's range can be measured in millimeters. They might be susceptible to electromagnetic disablement as well as destruction by microwave radiation if there's no antenna larger than the current models.

1.12 Radio communication:
There are several costly high-speed radio transmitters and receivers in the market. Since it's prototype and to reduce costs we have utilized the transmitter and recipient module 2.4 GHz FlySky 6 channel. The typical obstacle covers about 970 meters to 1 kilometer. (Figure 4).

![Figure 4](FlySky six channel radio transmitter and receiver)

It is a very poor signal above 1200 meters and has lost the signal altogether at 1320 meters. We utilized a vehicle to travel about and broadcast the signal from a stationary place in order to record exact data.

Technical Specifications:
- Radio: 2.4 GHz
- Length: 7.4 in (188mm)
- Height: 3.8 in (96.5mm)
- Width/Diameter: 11.6 in (294.6mm)
- Weight: 498.9 g (17.6oz)
We can control our drone flying system with the help of this gadget. Each channel controls an electronic device through which we may control the prototype movement forward, backward, right or left, in an inbuilt motor, such as brushless DC motors and servo motors.

1.13 Wi-Fi Communication:

![Figure 5](image)

Figure 5 Shows the overall drone circuit diagram

Wi-Fi connectivity for short-range video without internet connection utilized in this experiment for the IP webcam software. For video transmission, we have make use of android IP webcam software. It provides live video streams accessible through local and worldwide computers. It makes use of Wi-Fi to communicate locally. Mobile devices and on-board computers are installed on the top of the aluminum bar in the same Wi-Fi network motors and are positioned under each brushless motor with a servo motor. The center of the body incorporates all the payloads (ESC, Controller, RF receiver, battery and mobile device). All connections between all electrical components are shown in Figure 5. This graphic indicates +5V, Signal and GND correspondingly for fine-red, orange and black wires. Thick red and black are 11V and GND line connectors [2].

2. Environmental set up

Figure 6 shows the overview of the IoT data collection allowed by DC, where many DCs are available to transmit data upload from IoT devices to one BS. The DC users are defined as IoT devices whose dependable U2B connections are affected by blocking or lack of resources. BS checks the CSI for each U2B pair regularly and assigns the DC user set $\mathcal{U}$ with hacked IoT devices. With $\mathcal{U}$ and the DCs Set $\mathcal{D}$ accessible, the BS executes the design method for trajectories to identify the optimum trajectory for each DC. The data transmitting devices in the DC network are considered in all aspects as equivalent, save one: they differ in data collection time, bandwidth and power transmission. In order to avoid inter-user interference, we presume that every DC can differentiate accurately its user's signals and that the suggested DC trajectory design method provides a schedule for the transmission time of all users in one DC so as to avoid overlap between transmission and transmission intervals. $|\mathcal{U}|$ and $|\mathcal{D}|$ are the cardinalities that reflect the number of DC users and available DCs, respectively [1].
3. **D2B and U2D Channel Models:**

The wireless connections of both U2D and D2B are modelled on recent D2G channel investigations [10] [11]. According to [10], the U2D pathloss may be expressed as

\[
PL(r_{UD}, h) = 10 \log \left( \frac{16\pi^2 f_c^2 (h^2 + r_{UD}^2)}{c^2} \right) \left[ P_{\text{LoS}} \eta_{\text{LoS}} + (1 - P_{\text{LoS}}) \eta_{\text{NLoS}} \right]
\]

(1)

Where \( h \) is DC is the flying altitude, \( r_{UD} \) is the U2D horizontal distance, \( f_c \) and \( c \), respectively, are carrier frequencies and light speed in Hz and m/s. \( \eta_{\text{LoS}} \) and \( \eta_{\text{NLoS}} \) are LoS and NLoS pathloss offsets based on the environment correspondingly. \( P_{\text{LoS}} \) is the likelihood of U2D LoS, as defined.

\[
\Gamma_{\text{LoS}}(r_{UD}, h) = \frac{1}{1 + a \cdot \exp\left(-b \cdot \frac{\arctan\left(\frac{h}{r_{UD}}\right)}{\pi}\right)}
\]

(2)

Where \( a \) and \( b \) are parameters that rely on the environment. \( f_c \) should be distinct from typical IoT bands such as Wi-Fi or NB-IoT [12] to avoid interferences with communications from U2B, as well as to give additional spectrum resource for DC users. The D2B pathloss average is computed by [11]

\[
PL(r_{DB}, \theta) = 10a \log(r_{DB}) + A(\theta - \theta_0)e^{\left(\frac{E-\theta}{B}\right)} + \eta_0
\]

(3)
Where $r_{D2B}$ and $\theta$ denote the D2B horizontal distance and the DC to BS antenna vertical angle, respectively. $\alpha, A, \theta_{B}, B$ and $\eta_{d}$ these all factors are depend on the environment. Since LTE transmissions have 850 MHz band [11], (3) do not contain any carrier frequency parameter.

4. DC Trajectory Model:

For one DC $d \in D$, we suppose that it works in multiple access (TDMA) mode in the associated user set $U_{d} \subseteq U$. DC $d$ files across all their related users and serves them sequentially according to scheduling results inside a single T-trajectory period. $T$ is set to the same value as the users’ time for collecting data to allow each user to communicate the data obtained once inside a single period. The trajectory of DC $d$ may be modelled as an N-length sequence comprised of three dimensional vectors by distributing $T$ into $N$ equal time pitches within one period:

$$G_{d}[n] = [x_{d}[n], y_{d}[n], h_{d}[n]], \quad n \in \mathbb{N}$$

(4)

Here the $n$ parameter refers to the channel of frequency allocated for the drone cell characteristics for network communications.

5. Uplink and Downlink:

Satellite communications terminology defines a downlink as the transmission of data from a satellite to one or more ground stations, and an uplink as the reception of data from a ground station to a satellite. In certain cases, businesses, television stations, and other telecommunications carriers use uplink and downlink services. A business may focus on uplinks and downlinks or provide both. Equation 2, 3, and 4 are analyzed the data with the simulator and the set of data-set is generated for the calculations [13] [14]. This calculation is analyze keeping frequency as 2.4 Ghz. Figure 7 and Figure 8 represents the calculation for 10 channel and 6 channel for downlink and uplink data respectively.

**Figure 7:** Downlink timing analysis for 10 channel dataset.
6. Conclusion

The system is proposed for drone communication and Networking prospective with IoT integration of Cloud, Fog, Edge and Smartdust architecture. We have demonstrated the cloud integrated network IOT architecture on drone communication for optimization of uplink and downlink time. We can integrate the mathematical prospects for edge, fog and smartdust system. It can be seen the error chart ranged from 20.2% to 27.4% change in downlink activity and 17.6% to 23.8% change in uplink. The downlink time can be reduced by increasing the processing ability of the communication protocols or any of the four IoT architecture in micro-controller like raspberry pi, Pixhawk 4 or NODEMCU.

References

[1]. M. Mozaffari, W. Saad, M. Bennis, and M. Debbah, “Mobile Unmanned Aerial Vehicles (UAVs) for Energy-Efficient Internet of Things Communications,” IEEE Trans. Wireless Commun. Mag., vol. 16, no. 11, pp. 7574–7589, 2017.
[2]. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari, and M. Ayyash, “Internet of things: A survey on enabling technologies, protocols, and applications,” IEEE Communications Surveys Tutorials, vol. 17, no. 4, pp. 2347–2376, 2015.
[3]. J. Gubbi, R. Buyya, S. Marusic, and M. Palaniswami, “Internet of things (IoT): A vision, architectural elements, and future directions,” Future generation computer systems, vol. 29, no. 7, pp. 1645–1660, 2013.
[4]. Nguyen, Dang Tu; Song, Chengyu; Qian, Zhiyun; V. Krishnamurthy, Srikanth; J. M. Colbert, Edward; McDaniel, Patrick (2018). IoTSan: Fortifying the Safety of IoT Systems. Proc. of the 14th International Conference on emerging Networking EXperiments and Technologies (CoNEXT '18). Heraklion, Greece. arXiv:1810.09551. doi:10.1145/3281411.3281440. arXiv:1810.09551.
[5]. S. Ranger. What is cloud computing? Everything you need to know. Mar. 11, 2020. url: https://www.zdnet.com/article/what-is-cloud-computing-everything-you-need-to-know-from-public-and-privatecloud-to-software-as-a/.
[6]. Nouri, Seyed; Han, Li; Srikumar, Venugopal; Wenxia, Guo; MingYun, He; WenHong, Tian (2019). "Autonomic decentralized elasticity based on a reinforcement learning controller for cloud applications". Future Generation Computer Systems. 94: 765–780. doi: 10.1016/j.future.2018.11.049.
[7]. Arkian, Hamid Reza; Diyanat, Abolfazl; Fourkholali, Atfe (2017-03-15). "MIST: Fog-based data analytics scheme with cost-efficient resource provisioning for IoT crowdsensing applications". Journal of Network and Computer Applications. 82: 152–165. doi: 10.1016/j.jnca.2017.01.012.
[8]. Taleb, Tarik; Dutta, Sunny; Ksentini, Adlen; Iqbal, Muddasar; Flinck, Hannu (March 2017). "Mobile Edge Computing Potential in Making Cities Smarter". IEEE Communications Magazine. 55 (3): 38–43. doi:10.1109/MCOM.2017.1600249CM. S2CID 11163718. Retrieved 5 July 2019.
[9]. J.D. Dorave, Ritesh Sadiwala, “Compatibility and Integration of Smart Dust with various frequencies of electromagnetic spectrum and physical systems”, Vol.2, Issue 3, June 2019 pages 6, The Splendid Journals.
[10]. A. Al-Hourani, S. Kandeepan, and S. Lardner, “Optimal LAP altitude for maximum coverage,” IEEE Wireless Commun. Lett., vol. 3, no. 6, pp. 569–572, 2014.

[11]. A. Al-Hourani and K. Gomez, “Modeling cellular-to-UAV path-loss for suburban environments,” IEEE Wireless Commun. Lett., vol. 7, no. 1, pp. 82–85, 2018.

[12]. M. E. Soussi, P. Zand, F. Pasveer, and G. Dolmans, “Evaluating the performance of eMTC and NB-IoT for smart city applications,” in Proc. IEEE ICC, Kansas City, MO, May 2018, pp. 1–7.

[13]. ATIS committee PRQC. “network topology”. ATIS Telecom Glossary 2007. Alliance for Telecommunications Industry Solutions. Retrieved 2008-10-10.

[14]. C. E. Shannon, A mathematical theory of communication, Bell System Technical Journal, vol. 27, pp. 379–423 and 623–656, (July and October, 1948).