# DSCK: Drone movement for better signal quality

Aastha Tripathi
University of Texas at Austin
aastha_tripathi@utexas.edu

Amanda Austin
University of Texas at Austin
ajaustin@cs.utexas.edu

## Abstract

*The Internet is becoming an invaluable resource that can connect many people around the world. However, holes still remain in various locations around the world in Internet connectivity. Drones that are capable of transmitting 802.11 wireless signals can be used to fill in these holes. Drones are attractive in these scenarios because they are able to move around in their environment to provide better connectivity for their clients. We present DSCK, a system that tries to maintain a good quality signal at a client connected to a drone by allowing the drone to move around its environment. We implement and test a prototype of DSCK on a Parrot AR.Drone 2.0 and show that our protocol is able to maintain a good quality signal with a connected client.*

### Keywords

*Drone Controller, Received Signal Strength, Wireless Communication, Access points*

## 1. Introduction

More and more, the Internet is becoming an invaluable resource with the potential to positively transform many aspects of humanity. The Internet has helped facilitate many advances in health, education, financial services, transportation, agriculture, energy, and more. However, over half of the world's population still remains disconnected from the Internet. Many reasons exist for these gaps in Internet access. Many people lack Internet access because they live in remote and difficult-to-reach areas. Other people live in areas without access to basic infrastructure such as electricity or transportation. Connecting these disconnected areas to the Internet at a low cost remains a challenge. [1]

Drones are an attractive option for addressing this challenge. For example, Facebook over the past few years has been working on bringing Internet connectivity to rural areas using large carbon-fiber drones. These drones will be put into flight over disconnected rural areas and, using a system of lasers, will help bring wireless connectivity to the Internet to these areas. This concept is not new, and other technology companies like Google are exploring solutions to the same problem. [2]

Drones can also be used as an extender for cell towers. Traffic over outdoor cellular networks is on the rise. As more and more applications operate over cellular networks, more and more pressure is being put on the bandwidth of these networks. Experts predict a 1000X increase in wireless data demand by 2020. While advances in wireless network technologies such as MIMO, beam-forming, spectrum sensing, and others have been helpful in coping with the increased pressure put on wireless networks, other approaches are needed to fill in the gaps that still remain. Despite these recent advancements in wireless network technologies, users can still experience spatial or temporal gaps in connectivity or degradation of signal quality. For example, tall buildings can cause a wireless shadowing effect, which can significantly degrade the signal quality in certain areas. Additionally, large crowds put a large amount of pressure on the network resources due to the sudden traffic spikes. Natural disasters can destroy network infrastructure, disconnecting an entire area from the Internet. Addressing these problems in a cost-effective and scalable way remains a challenge. Drones can help address these challenges. [3]

Drones are currently being used in solving the issues in Internet connectivity all around the world. Drones have came to be a powerful tool; much current research in the field of wireless networking is focused around drones. However, drone availability alone does not ensure the quality of the signal. The mobility of the drone is a major factor that can affect the signal quality received by clients connected to the Internet through the drone. Our paper takes a step forward to ensure the signal quality received from the drone at the client remains good. Here we present a system that we call DSCK (Drone Signal Check) that continuously monitors the received signal strength from the drone at a connected client and, if the signal quality drops below a certain threshold, commands the drone to move around the environment in order to find a location that gives a better signal at the client.

The rest of our paper is organized as follows: Section 2 discusses our design of DSCK, Section 3 discusses the implementation of DSCK using a Parrot AR.Drone 2.0, Section 4 discusses our experimental evaluation of DSCK, Section 5 discusses the limitations of our approach and future work, and Section 6 concludes.

## 2. Design

Many factors can affect the quality of the signal received at a client. Wireless signals degrade with distance. Other factors can degrade a wireless signal, such as traveling through an object. Interference from other RF signals or electrical interference can also degrade the quality of the signal. Finally, environmental factors such as weather can affect the quality of the received signal.[4]

Previous work by Wang and Gutierrez [5] shows that signal strength at the receiver correlates fairly well with the distance from the receiver. Using the same drone on which we implement our prototype of DSCK, Wang and Gutierrez monitored the signal strength at the receiver as the drone moved back and forth over the receiver at various speeds. They showed that the received signal strength followed the free space model at the distances they were working with. That is, the received signal strength was inversely proportional to the square of the distance from the transmitter. Building on this previous work, DSCK monitors the received signal strength. If DSCK decides that the received signal strength is poor, moving the drone closer should achieve a better signal at the receiver.

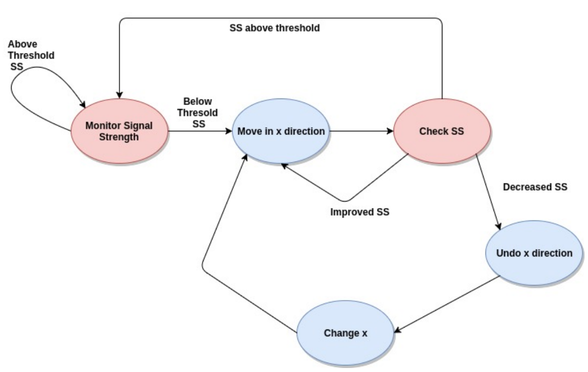


Figure 1: State Machine for drone movement decision

Figure 1 shows the state machine for DSCK. At the client, DSCK monitors the signal strength received from the drone. In order to smooth out any noise in the received signal strength, an average signal strength is computed over a defined time interval, which is ten seconds in our prototype. If the average signal strength over this time period is above a specified threshold signal strength, which in our prototype is -61 dBm, DSCK continues to monitor the signal strength received from the drone. If, on the other hand, the received signal strength is less than the specified threshold, DSCK must attempt to move the drone closer to the client in hopes of improving the signal quality at the client.

In order to move the drone closer to the client, DSCK tells the drone to explore in four directions: forward, backward, left, and right, in that order. After making the drone move, DSCK monitors the new received signal strength for a shorter period than the above interval, which is five seconds in our prototype. DSCK uses a shorter time interval when monitoring the signal strength after making a move to benefit responsiveness in the event that the move actually makes the signal strength at the client worse.

If the direction the drone moved made the signal strength at the client better, the drone will continue to move in that direction until the signal strength returns to the above defined threshold. If, on the other hand, the signal strength after moving is worse than the signal strength before making the move, DSCK determines that the drone must have moved in the wrong direction away from the client. DSCK tells the drone to move in the opposite direction in order to return the drone to its original position. After this, the drone will explore in a different direction and the above process repeats itself.

In this way, the drone explores the area, using changes in signal strength to deduce where the client is relative to itself. Since the client may move around, every time the signal strength drops below the defined threshold, the drone starts its exploration stage over again.

## 3. Implementation

We implement a prototype of DSCK for a Parrot AR.Drone 2.0, seen in figure 2. This drone contains four motors to control movement. The drone has an ARM Cortex A8 1 GHz 32-bit processor running the Linux kernel version 2.6.32. Users can connect to the drone using 802.11 WiFi. The drone also has various sensors, including a gyroscope, accelerometer, magnetometer, pressure sensor, and an altitude ultrasound sensor.

We implement our prototype of DSCK as an extension to the *tum_ardrone* package [6]. This package contains the framework for controlling an AR.Drone 2.0. Once installed

Figure 2: Falcon AR.Drone 2.0

on a client's device, the client can launch the package in order to control the drone. The drone can be controlled via one of three ways.

- **Autopilot**: A predefined flight plan is sent to the drone to follow
- **Keyboard**: Certain keys on the client's device are mapped to control the movement of the drone
- **Joystick**: A video game controller can be used to control the movement of the drone

We extend the Keyboard control method to control the drone. Upon connecting to the drone and launching the package, the client presses his or her escape key on the keyboard in order to enable keyboard control of the drone. This launches a background thread that continuously monitors the signal received from the drone via calls to iwconfig. When the average signal strength over the monitor period drops below a predefined threshold signal strength, keyboard commands are programmatically sent to the drone from the background thread to cause the drone to move. Upon receiving a keyboard command, the tum_ardrone package automatically calculates the required motor speeds in order to move the drone in the specified direction.

After sending the move command, the background thread continues to monitor the received signal strength from the drone. If the signal quality improved, the same keyboard command is sent to the drone until the received signal strength improves to above the given threshold. If, on the other hand, the signal strength got worse, then the keyboard command that undoes the last move is sent to the drone, and a new keyboard command is sent in order to make the drone explore a different direction. This process repeats until the signal strength improves to above the predefined threshold, at which point the background thread returns to monitoring the received signal strength from the drone.

Figure 3 shows the pseudocode of the implemented algorithm:

```
moves={FORWARD, BACKWARD, LEFT, RIGHT}
undoMoves={BACKWARD, FORWARD, RIGHT, LEFT}

while True:
  signal = monitorSignal(MONITOR_PERIOD)
  if signal < THRESHOLD:
    for move in moves:
      sendMoveCommand(move)
      newSignal = monitorSignal(MOVEMENT_PERIOD)
      if newSignal < signal:
        undoMove = undoMoves[move]
        sendMoveCommand(undoMove)
      else if newSignal < THRESHOLD:
        while newSignal < THRESHOLD:
          sendMoveCommand(move)
          newSignal = monitorSignal(MOVEMENT_PERIOD)
        break
      else:
        break
```

Figure 3: Psuedocode

# 4. Evaluation

## 4.1. Experimental Setup

We evaluated our prototype implementation in multiple experiments. We connect a Dell laptop with an Intel(R) Core(TM) i5-7200U CPU running Linux kernel version 4.10.0.33 to the drone and start our program. We run our experiments inside a classroom building on the campus of the University of Texas at Austin whose floor plan is given in Figure 4. In each run of the experiment, the drone is placed on the ground at the red dot. The client starts at the blue labeled dot. The drone controller program is started and after a period of time, the client begins to move to various areas of the room. The client moves backwards into the hallway, stands there for a few seconds, then moves forward and then towards the right, pausing in various places in between. We log the computed average signal strengths received from the drone throughout the experiments.

## 4.2. Results

Figure 5 shows the received signal strength over time for various runs of the experiments. The monitor period when

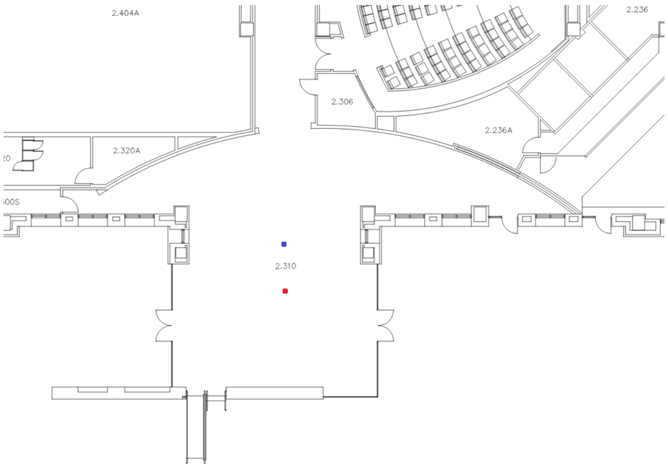


Figure 4: Floor plan of our experimental setup. On each run, the drone starts off at the red labeled dot. The client starts off at the blue labeled dot. After a period of time, the client moves around the room and the room

the signal strength is strong at the receiver is ten seconds while the monitor period after the drone has moved is five seconds. The threshold signal strength is -61 dBm.

Because the client initially starts in a position that is close to the drone, the initial signal strengths are above the threshold. However, as the client moves around, the received signal strength starts to fluctuate. As the client moves further from the drone, the received signal strength drops to below the threshold. At this point, the drone begins to explore the environment according to its exploration pattern until it notices that the signal strength has improved. The exploration further causes the received signal strength at the client to fluctuate.

In the first two experiments, shown in the first row of Figure 5, the client moved very slowly around the environment. This behavior manifests in the smoother fluctuations in signal strength over time. In the second two experiments, shown in the second row of Figure 5, the client was moving more quickly through around the environment. This behavior produced the sharper curves in the plots as the signal strength fluctuated more rapidly. In the final experiment, shown at the bottom of Figure 5, the client moved quickly around the environment and then remained stationary in a fixed position until the end of the experiment. Once the client has stopped moving, the drone is able to find a position that gives a good received signal strength at the client, significantly above the given threshold. Because the client is no longer moving around and because the signal strength at the client is good, the drone no longer moves around and hovers in its position

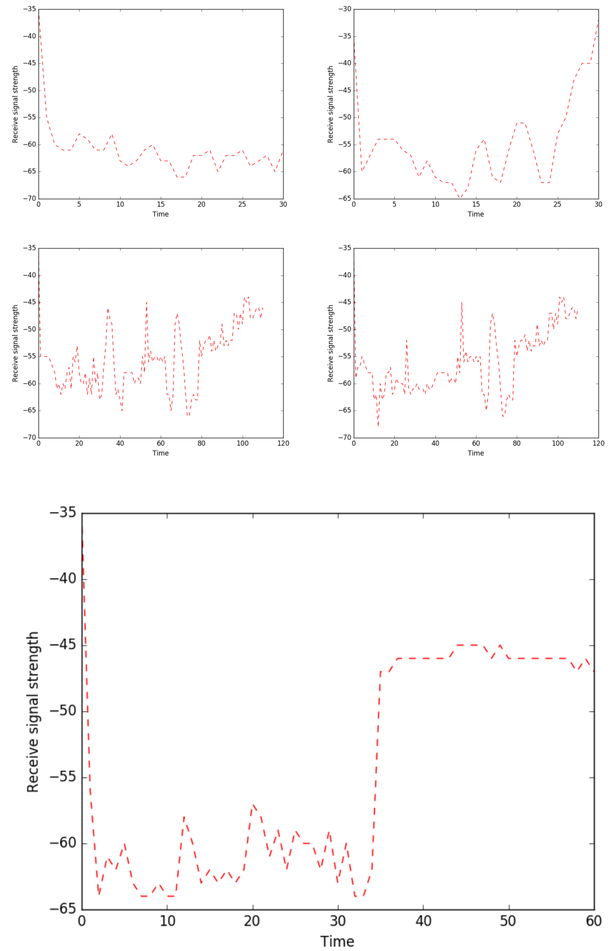


Figure 5: Plots of received signal strength from the drone over time for various experiments. Drops in signal strength happen when the client moves away from the drone. Fluctuations in signal strength are a result of the drone exploring the environment to figure out where the client is based on signal strength improvements

close to the client.

## 4.3. Discussion

Our results show that the exploration pattern of DSCK allows the drone to find new positions that allow the signal strength at the client to improve. As the client moves around, the drone is able to explore the environment until the signal strength has improved. In many cases, the new position of the drone gives a much better received signal strength than the threshold strength. Finally, as the client stops moving, the drone is able to find a position that gives a strong signal to the client and remain hovering in its signal monitoring state.

## 5. Limitations and Future Work

Although DSCK is able to achieve good performance when exploring the environment for locations that provide a better signal strength, the protocol could still be improved. We leave the following as future work:

- Single User: In the current implementation, the movement of the drone is controlled according to the received signal strength at a single client. DSCK could be expanded in the future to handle multiple clients. This would involve moving the implementation of DSCK from the client to the drone.

- Exploration directions: In the current design, the drone explores only four directions: forwards, backwards, left, and right. For better performance and robustness, the drone could also explore up and down as well as rotation for diagonal movement or movement along random vectors.

- Received signal strength: The movement of the drone is controlled on the basis of received signal strength at the client which may not always correlate well with the actual channel quality. The current implementation of DSCK ignores factors such as multi-path propagation that could affect signal strength. Further improvements of DSCK could focus on monitoring multiple factors in order to decide when the drone should be moved.

- Threshold and time parameters: The current implementation of DSCK uses a constant signal strength threshold when deciding whether the drone should move. Due to differences in network conditions in different environments as well as differences in network cards in various devices, the threshold in the current implementation may not be appropriate for all environments or clients. Rather than use a predefined constant for the signal strength threshold, DSCK could monitor the signal strength for a period of time and attempt to infer a suitable threshold value. Additionally, the time parameters used for the monitor periods of DSCK are constant. These values could be varied depending on how responsive the client needs the drone to be to changes in signal strength.

## 6. Conclusion

DSCK is a system for monitoring and maintaining signal quality at a client connected to a drone via 802.11 wireless signals. Received signal strength at the client correlates well with the distance from the drone. Therefore, when the signal quality drops below a predefined threshold, DSCK allows the drone to explore the environment to find a location that gives a better signal at the client. Our evaluation shows that DSCK is able to respond to changes in signal strength caused by client mobility and can help maintain a good signal strength at the client.

## References


[1] I. Philbeck, "Connecting the unconnected working together to achieve connect 2020 agenda targets." [Online]. Available: http://broadbandcommission.org/Documents/ITU_discussion-paper_Davos2017.pdf

[2] "Facebooks internet drone: Connecting the unconnected." [Online]. Available: https://sites.psu.edu/dsalonickcas272/2017/04/14/facebooks-internet-drone-connecting-the-unconnected

[3] A. Dhekne, G. Mahanth, and R. R. Choudhury, "Extending cell tower coverage through drones," *Proceedings of the 18th International Workshop on Mobile Computing Systems and Applications, ACM.*, 2017. [Online]. Available: http://synrg.csl.illinois.edu/papers/dronenet_hotmobile2017.pdf

[4] J. Schiller, "Mobile communications, second edition," pp. 31–41, 2003.

[5] R. A. Gutierrez and C. Wang, "Mobility-aware signal strength prediction for drone communication."

[6] "tum_ardrone." [Online]. Available: https://github.com/tum-vision/tum_ardrone