Integrating Drone System with SDN Framework

Project Workbook

By

Aakash Shah (aakash.shah@sjsu.edu)
Rutvij Shah (rutvij.shah@sjsu.edu)
Satyam Arun Sheth (satyamarun.sheth@gmail.com)

Project Advisor

Prof. Younghee Park

October 9, 2016

Index

Chapter 1. Literature Search, State of the Art and References

1.1 Literature Search

Significant improvement in technology has brought about a revolution and has impacted each and every aspect of human life. A plane flying in the air, a dream decade ago, has become a reality. Further advancements in aeronautical technology gave rise to Unmanned aerial vehicles (UAVs) initially used for Military purposes, Drones in particular. Realizing the vast functionality and the usability of these drones, they are now made available for the public domain use by the commercial industry. As a result, the airspace has been flooded with drones and a centralised governing and controlling system for monitoring multiple drones has become a necessity. Integrating Software Defined Network (SDN) to the drone system would provide a centralised decision taking entity that would monitor and manage the entire system. For integrating SDN with drone there is a need of collision detection module, centralised controller taking the necessary decisions, security measures to secure the system and a communication module between the SDN controller and the drones.

It is very essential to manage and measure the data packets processed by the SDN controller. This work related to Traffic Engineering is well researched and documented by Z. Shu et al [1] where PayLess, an SDN flow measurement framework, maintains an abstract view of network information for applications and provides a unified programming interfaces for a variety of network applications. Hedera is a dynamic and scalable traffic management system which schedules packet flow adaptively to effectively utilize network resources, and tries to realize equal bandwidths.

Software defined radio communication is an important aspect for communication between the SDN framework and the drones. The work carried out at Cottbus, Germany [2] suggests a GNU Radio based IEEE testbed where a trans receiver module with a layered structure is developed for rapid prototyping. The Network Stack in the GNU testbed provided interoperability from the physical to network and application layer. Another finding at 5th IEEE Vehicular Networking Conference (VNC 2013) [3] suggested a Full SDR based Trans receiver, where the core of the framework is a modular Orthogonal Frequency Division Multiplexing (OFDM) trans receiver. On the hardware side, The Ettus Research USRP N210 with a CBX daughter board is used, allowing the user to operate on the Intelligent Transportation Systems (ITS) frequency band around 5.9 GHz.

Collision Avoidance mechanism is an essential aspect for drone management. Many papers have proposed different techniques and protocols to improve collision detection and avoidance in Wireless Sensor Networks. CSMA/CN: Collision avoidance in WSN is possible using CSMA/CN as per the study conducted by Souvik Sen [9]. The author has proposed this technique to notify the presence of an interference once it is detected. Here, the transmitter has two channels- one for transmission and one for the notification handling. The receiver will detect the interference and will notify the transmitter. However, the receiver would only be having one channel to receive transmission as well as transmit the notification.

Another paper proposes spin protocol for collision avoidance and detection. There will be one node which will gather all the data and then will transmit through the destination station [12]. Negotiation mechanism ensures elimination of redundant information. Modified spin protocol provides a fast response to provide real-time collision detection and avoidance.

Need of security in any system is very crucial. In our Drone system integrated with SDN framework, security needs to be provided at each layer. When the sensors on drone senses the real world data and send this data set to the SDN controller, the system needs an intrusion detection and prevention system between them so that packet filtering can be implemented. This avoids the Denial of Service and flooding attacks. Authors Nhu-Ngoc Dao, Junho Park, Minho Park and Sungrae Cho have very thoroughly researched this topic and they have provided the solution in their research paper on how to avoid DoS attack by providing IDS [5] [7].

There is also a need to provide Software Defined Security module in SDN controller to define the security policy measures and Hierarchical flow tables. This implementation provides user a flexibility to decide its own policies as per the development of the system. Also there are security framework such as OpenSec and Software Defined Security in which these policies have been implemented with user flexibility. Only an integration should have been done. OpenSec related study [6] have defined OpenSec framework which can be tested on GENI testbed and stated how this security framework is efficient and reliable. The deep analysis has been provided in this paper. Software defined based smart grid architecture presented at Security Technology International Carnahan Conference [8] has provision for privacy preservation and integrity protection.

1.2 State of the Art

Following are the State of the art solutions for different technologies and techniques used for integrating SDN with the Drone system for managing and monitoring the entire system.

GNSS-SDR: An Open Source tool

This is a wireless communication tool which uses an open source Global Navigation Satellite System Software-defined-receiver. The receiver provides an interface to different suitable Radio Frequency fronts -ends and provides a navigation solution by implementing the receiver chain. This tool provides for effective communication module that fulfils the goals of efficiency, modularity, flexibility and interoperability as demanded by the user [4].

Packet Trace-back: A Traffic Engineering tool

Packet Traceback can determine how a packet reached its current location, including the path through the network and any modifications of the packet en-route. The tool provides a precomputation of a compact symbolic representation of the back policy which can quickly produce all possible predecessors for any input packet [5].

Hierarchical flow table

Hierarchical Flow Tables (HFT) is a framework for specifying and managing Hierarchical policies in SDN. These policies are mainly implemented above the controller to provide the solution for conflicts. These policies allow particular nodes with IP addresses to be in software defined networks. HFT policies are organized as trees, where each component of the tree can independently determine the action to be taken on each packet. When independent parts of the tree arrive at conflicting decisions, HFT resolves conflicts with user-defined conflict-resolution operators, which exist at each node of the tree. These Hierarchical policies can be implemented in our system in SDN controller to provide security policies.

OpenSec

OpenSec consists of a software layer running on top of the SDN controller and multiple external devices that perform security services such as firewall, encryption, spam detection, deep packet inspection (DPI) and report the results to the controller. OpenSec allows network operators to describe security policies for specific flows. The policies include a description of the flow, a list of security services that apply to the flow and how to react in case malicious content is found. So in our system the user can write its own policies and then OpenSec framework can implement these policies on the packets it receives.

Snort

Snort is an open source network intrusion prevention system (IPS) capable of performing real-time traffic analysis and packet-logging on IP networks. It can perform protocol analysis, content searching & matching and can be used to detect a variety of attacks and probes, such as buffer overflows, stealth port scans, CGI attacks, SMB probes, OS fingerprinting attempts and more.

1.3 References

1. Z. Shu et al., "Traffic engineering in software-defined networking: Measurement and management," in IEEE Access, vol. 4, no., pp. 3246-3256, 2016
   from http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=7496952

   The research paper proposed a reference framework for Traffic Engineering (TE) in the SDN, which consists of two parts, traffic measurement and traffic management.

2. Bastian Bloessl, Christoph Leitner, Falko Dressler and Christoph Sommer, "A GNU Radio-based IEEE 802.15.4 Testbed", Proceedings of 12. GI/ITG KuVS Fachgespräch Drahtlose Sensornetze (FGSN 2013), Cottbus, Germany, September 2013, pp. 37-40
   from http://www.ccs-labs.org/bib/bloessl2013gnu/bloessl2013gnu.pdf

   The paper provides information on developing a trans receiver testbed for GNU radio with interoperability between different layers.

3. Bastian Bloessl, Michele Segata, Christoph Sommer and Falko Dressler, "Towards an Open Source IEEE 802.11p Stack: A Full SDR-based Transceiver in GNU Radio", Proceedings of 5th IEEE Vehicular Networking Conference (VNC 2013), Boston, MA, December 2013.
   from http://www.ccs-labs.org/bib/bloessl2013towards/bloessl2013towards.pdf

   The research paper focused on developing a Full SDR-based trans receiver as a wireless mode of communication. It helped in understanding the basic building blocks of an trans receiver for GNU radio.

4. GNSS-SDR: An Open Source Tool
   from http://gnss-sdr.org/

   The website was helpful for learning how GNSS-SDR provides for an effective SDR based communication.

   Packet Traceback for Software-Defined Networks
   from ftp://ftp.cs.princeton.edu/techreports/2015/978.pdf

   This article guided on state of the art tools and technologies trending in the SDN Traffic Management field.

5. Nhu-Ngoc Dao, Junho Park, Minho Park and Sungrae Cho, "A feasible method to combat against DDoS attack in SDN network", Information Networking (ICOIN) 2015 International Conference on, pp. 309-311, 2015.
   from https://www.computer.org/csdl/proceedings/icoin/2015/8342/00/07057902-abs.html

This paper describes the feasible method to void attacks like Dos and DDos by developing and implementing Intrusion detection and prevention system.

6. Adrian Lara and Byrav Ramamurthy, "OpenSec: Policy-Based Security Using Software-Defined Networking", Network and Service Management IEEE Transactions on, vol. 13, pp. 30-42, 2016, ISSN 1932-4537.
   from http://ieeexplore.ieee.org/document/7378982/

   This paper mainly focused on OpenSec, an OpenFlow-based network security framework that allows operators to implement security policies across the network is security framework in which the flexibility to user

7. Pin-Jui Chen and Yen-Wen Chen, "Implementation of SDN based network intrusion detection and prevention system", Security Technology (ICCST) 2015 International Carnahan Conference on, pp. 141-146, 2015, ISSN 2153-0742.
   from https://www.casra.ch/en/research-consulting/publications.html

   This paper has described different Intrusion detection and prevention system for software defined network.

8. Yaser Jararweh, Ala' Darabseh, Mahmoud Al-Ayyoub, Abdelkader Bousselham and Elhadj Benkhelifa, "Software Defined based smart grid architecture", Computer Systems and Applications (AICCSA) 2015 IEEE/ACS 12th International Conference of pp. 1-7, 2015, ISSN 2161-5330.
   from http://dblp.uni-trier.de/pers/hd/d/Darabseh:Ala

   The paper has described a smart grid architecture which is an intact efficient and reliable for SDN by dividing the whole network into grids.

9. Sen Souvik, et al. "Moving away from collision avoidance: Towards collision detection in wireless networks." ACM HOTNETS, 2009.
   From http://synrg.csl.illinois.edu/papers/csma-cn.pdf

   The paper proposes a new approach of providing collision avoidance in wireless networks by adding a new channel on transmitter side which will receive notification and accordingly avoid collision.

Chapter 2: Project Justification

In recent time there has been a lot of work carried out in the field of Internet of Things (IoT) as it collects and analyses data acquired by the devices connected to an internet network. A network of multiple drones is one such IoT environment where multiple drones capture, analyse and communicate information.

The use of drones for commercial and personal purposes has been on increase with the multi-functionality it provides. Thus the airspace has been flooded with this drones and a technique for monitoring and controlling drones is a necessity. Traditional drones are controlled by individual operators who do not have the location information of other drones controlled by their respective operators in the vicinity, hence there is a high probability that these drones might collide with each other. To avoid this, more refined drone system that is controlled by a single centralised entity capable of taking real time decisions, defining flight paths and communicating the drone information over a secured single network and communication mode is required that would add to the efficiency and safety of these drone systems.

Current work in this field have come up with some solutions that avoid the drone inter-collision, but these solutions are not scalable, efficient and cost effective enough. Most of the systems rely on proximity sensors assembled over individual drones that set an alarm when another objects are in the vicinity. Hence these drones do consider other stationary objects as obstacle in the vicinity and set false alarm. Due to non – centralised control of multiple drones, more number of resources are required, as a decision making controller is required on each drone. This adds to the overall cost of the drone. Thus, commercially available solutions are not optimal and there is a need for more efficient and reliable system for managing and monitoring multiple drones.

In this project, we propose a SDN framework to control and monitor a large number of concurrently connected drones. The SDN framework would gather the sensor data from the individual drones to decide their flight path avoiding their intercollision. In addition, SDN would perform all resource intensive calculations, remote configuration would allow ease to add new drones to the system and programmability at fly would provide added utilisation and efficiency to the system. In addition, intrusion detection and prevention techniques to avoid leakage or corruption of data in the drone system can be facilitated by SDN based security modules. As SDN is a centrally organised governing system, a central controller radio system is required, A GNU Radio can be used to develop software – defined radios as same hardware can be used create many kinds of radios to communicate wirelessly between the SDN controller and individual drones. By integrating SDN over the drone system we plan to implement a flight determination, information computation and an efficient wireless radio communication module to improve and add efficiency to the overall management of the drone system.

Chapter 3: Project Requirements

3.1 Functional Requirements

Essential Requirements

- As a business user, we want the dashboard with login so that application users can access analysed data of the drone system.
- The application must indicate the user of possible drone collision in real time and with an alarm system to buzz.
- As an implementation of floodlight controller, the controller should have different modules for security purpose, communication purpose and modules to provide automation in drone system so that the drones can detect and avoid collision.
- The Drone system should have sensors which can sense the real time data, capture images and send this information over the Gnu radio.
- To manage the traffic and avoid the attacks caused by broadcast storms, the Intrusion detection and prevention system should be implemented.

Desired Requirements

- As the administrators, we want to create security policies which can be understood by my security modules implemented in SDN framework.
- We need the GNU Radio communication to be wireless. So software defined radio should be developed which can eventually send packets to the controller on wireless basis.
- Web application should be able to handle all the incoming data sets as well as control the drones.

3.2 Non-Functional requirements

Essential Requirements

- Reliability – Reliability depends on how accurately the entire system would provide accurate results. To provide usability over a period of time our systems contains open source tools and software. Also the entire system is majorly implemented in software which can be modified with changing scenarios.
- Accuracy – The controller should accurately analyse the possible collision and process the collision prevention algorithm to avoid the collision.
- Efficiency- The entire drone management and monitoring system depends on how efficiently the SDN controller handles the data packets in real time and analyses the possible collision and provides an alert to the user. Efficient working of the GNU Radio is also a main aspect.

Desired requirements:

- Scalable - It should be able to handle large number of packets.
- Performance – As this project is highly dependent on the data sensed in real time by drone's sensors, performance measure is an important task to look upon. If the communication overhead in the network is higher than it will decrease the performance and that can also affect the system and delay the transmission providing delay in alerting the user of a possible collision.

3.3 Required

- Integration of SDN framework with drone system
- Implementation of Security modules and IDS.
- Communication on wireless basis with the help of Software Defined Radio
- Collision detection and avoidance mechanism

Chapter 4: Dependencies and Deliverables

4.1 Dependencies

- Third party Navigation system: The project heavily relies on the GPS location of the drone system which needs to be provided in real time by third party navigation system. The location of the individual drones is very crucial for collision detection and avoidance.

- Drone Sensors: The sensors on the drone senses real world data such as images, proximity to external objects which are to be provided through the application dashboard to the user. A failure in sensor functionality and data would provide incorrect results.

- SDN Controller: SDN controller plays a heavy role in monitoring and managing the drone system as it takes the crucial decisions of controlling the drone behaviour. Failure in SDN controller functioning would halt the entire system.

- Data set: Integration of SDN Framework with drone system can be checked by the simulation of this data set. As the network of drones get bigger, resources of physical devices will prove inadequate and we will need data from a large number of physical devices to provide traffic as well as to check behaviour of the system.

- GNU radio medium: GNU radio is used for wireless communication between the Drones and the SDN controller. The radio communication depends on the frequency range which should match at both communication ends. Packets may get lost if the medium is interfered externally.

4.2 Deliverables

- Dashboard: Dashboard presents a unified view of the analysis performed by the SDN controller. Statistical analysis of the data set of the drone system will be presented to the user via a dashboard. Also it will provide user control to entire system like launching a particular drone.

- Wireless Communication module: We will develop a wireless GNU based radio system that is software defined and can be programmed to perform signal processing in software instead of using dedicated hardware.

- Security Measures: To make the entire system secure, we will implement security measures to avoid any intended attempt to cause abnormal behaviour in the drone system. OpenSec would provide privacy and integrity to the data packets processed by the SDN controller. The Intrusion detection and prevention system such as Snort would filter the packets send and received with the drone system.

- Conference: On an academic front, we would like to conduct a conference to present our work to our peer colleagues and provide a demo of our SDN controlled drone system.

Chapter 5: Project Architecture

Architecture diagram for the proposed SDN controlled drone system:

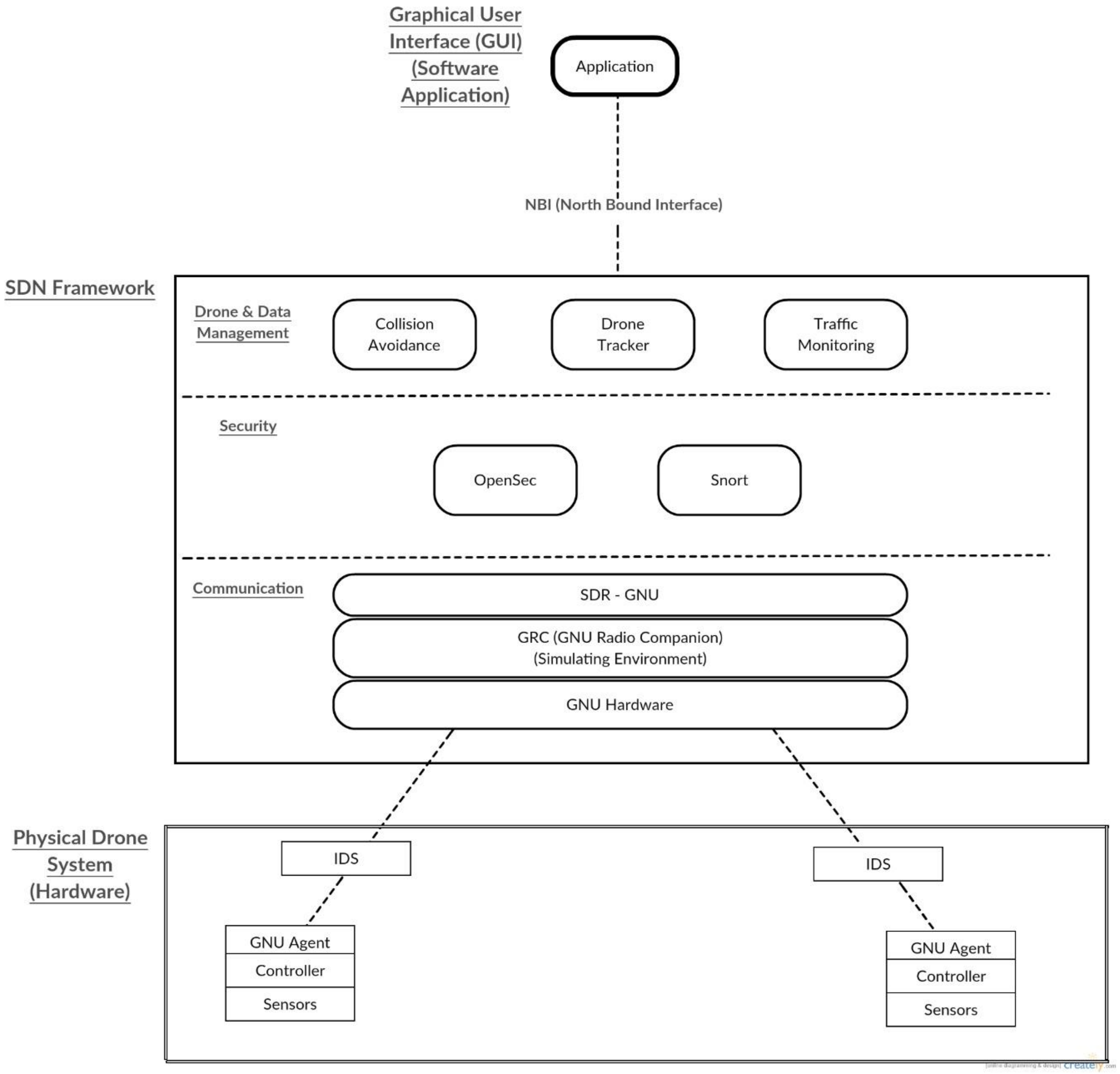


Figure 1: Project Architecture

5.1 Elaboration of architecture:

- **Application**

  Application is the GUI for the user which will enable the user to provide basic instructions to the individual drones. The application will indicate the user about a possible collision with another drone as analysed by the SDN controller. The application will be implemented on Amazon web services EC2.

- **SDN Framework**

SDN Framework is divided into three categories that have different functions.

1. **Drone and data management:**

   Collision detection and avoidance: In SDN framework, it is critical and important to implement a collision avoidance module. This module will improve drone tracking. Each drone will have its own location. A threshold range (safe radius) for each drone will be defined. If any other drone comes in this safe range, the algorithm for collision avoidance will be implemented and it will avoid collision.

   Traffic monitoring: Traffic monitoring is an essential aspect for our project as we need to prioritize the traffic related to drones so that we can efficiently manage and monitor the drone system. We will implement network traffic management, with an intent of providing reasonable traffic scheduling strategies according to network parameters. Traffic load balancing, Security and QoS-guarantee for data related to drones would be the main functions of the traffic monitoring module.

   Drone tracker: This module saves the log of location of drones. The module works in real time system as the drone's location will be changed every second. So the drone tracker keeps track of drones with the tags and location which can eventually help the Collision detection and avoidance module to keep track of two or more drones simultaneously and avoid collision. This module would also help to identify each individual drone.

2. **Security:**

   Snort: It is an intrusion detection and prevention system which detects the abnormal behaviour of nodes. This IDS and IPS system will filter the packets and avoid the Denial of Service attacks as well as flood attacks. This system will also compare the security policies decided by the user with incoming packets and on that basis it will discard or accept the packets.

   OpenSec: It is an OpenFlow-based network security framework that allows users to implement security policies across the network. Because OpenSec provides an abstraction of the network, the users can focus on designing simple and human-readable security policies, instead of on configuring all the devices to achieve the desired security. We will implement OpenSec for security measures and will implement this framework with GENI testbed to evaluate the flexibility, accuracy and scalability. Also this security module will provide privacy and integrity of data aggregation between drones.

3. **Communication module:**

   SDR – GNU Radio: SDN controller and Drones are two different physical entities with controller being a software environment and drones being a movable physical device. To control the drone, the SDN controller would require to send and receive data packets to the drone for collision avoidance on wireless basis. To do so we are using Software defined radios (SDR) as a mean of communication module. We are using SDR – GNU radio which allows us to process signals (data packets) in software and provides filters, decoders, demodulators, equalizers and other elements that are part of a radio system.

   GRC: GNU Radio Companion (GRC) is the simulating environment for creating GNU radio. It is the graphical tool for designing signal processing flow graphs. We are using GRC to create a radio transmitter and receiver at both ends i.e at SDN controller end and in drones. GRC provides graphical blocks which are interconnected to design a transreceiver. When this flow graph is executed, a python code is generated that implements the radio communication.

   SDR hardware: GRC is just a simulation environment which can be used to process the data to be send to the drones. We require SDR- GNU Radio compatible hardware that consists of antenna and USB module to send and receive the communication signal and connect to personal computers. Hack-RF is the hardware module that we are going to use as it operates over radio signals from 1MHz to 6GHz covering a vast range and has a USB peripheral.

- **Physical Drone System:**

  Drones are the physical devices which are primarily unmanned aerial vehicles which collect sensor information from various sensors integrated over the drone hardware. This information is processed by the controller mounted over the drone and is communicated to the SDN controller via the GNU module.

  The third party navigation system would provide exact location information of the drone to the drone controller. This information would be then processed to match the transport medium requirements in the GRC and then emitted over the space by the GNU hardware. The same GRC and GNU hardware is used to receive signals from the SDN framework. For our project we would use the drones commercially available with built-in navigation facility.

Chapter 8: Implementation plan and progress

8.1 Programming and execution environment setup

- **Progress status: Completed**

- Each member will setup following programming and execution environment:

  1. Application: Amazon Web Services EC2
  2. IDE for Programming Languages: Java, Python, C++
  3. Controller: Floodlight Controller
  4. Intrusion Detection and Prevention System: Snort
  5. Security module: OpenSec, GENI Testbed
  6. Development Tools: Eclipse
  7. Communication: GNU Radio Companion (GRC)

8.2 Acquiring development tools

- **Progress status: Completed**

- As all the tools are Open source it can be acquired online only.

- For Controller, Floodlight controller is a Java based OpenFlow Controller which is acquired from Open Networking Foundation.

- For programming, Integration of Java and Python will be provided to integrate floodlight controller.

- For Security, Snort is an open source IDS and IPS. OpenSec framework can be acquired from Software Defined Security modules.

- For Communication between drone, SDR simulation and development environment is obtained from open source SDR support.

8.3 Simulation environment

- **Progress status: In progress**

- As the dataset will be received on an application, this data can be simulated on cloud based application or Network Simulator NS2.

- GRC a simulation environment for GNU radio is used to develop the required flow graphs for designing the transreceiver.

8.4 Understanding/analysing/testing example programs or hardware simulations

- **Progress status: In progress**
- Analysing the floodlight controller modules written in Java.
- Understanding the security modules implemented in Java and Python.
- Research on Software Defined Radio for communication written in Python.
- Testing available GRC programs for understanding the GNU Radio receiver and transmitter designing.

8.5 Implementing the Essential feature set

- **Progress status: In progress**
- Following are the essential features for developing SDN integrated drone system, extensive research about individual systems should be done:

  1. **Integration:** Implementing an SDN framework over the drone system is main essential feature that our project is focused on.

  2. **Collision detection:** Collision can be detected by tracking Drone’s location and comparing them. We will set the threshold value of their distance. If the distance is less than the threshold value, the collision can be detected.

  3. **Communication:** Drones will communicate with controller with the help of GNU Radio. GNU Radio is a Software defined radio which is the essential communication module for our project.

  4. **Security:** Security will be provided in terms of two things. Snort will provide Intrusion detection and prevention system in which the traffic management can be done. It will in turn can avoid Sybil attack, DoS attack and flood attack. Another part is OpenSec framework. This Open Security framework will be implemented in SDN framework which provides Security policies defined by users and act accordingly.

8.6 Implementation of Desired and Optional Features

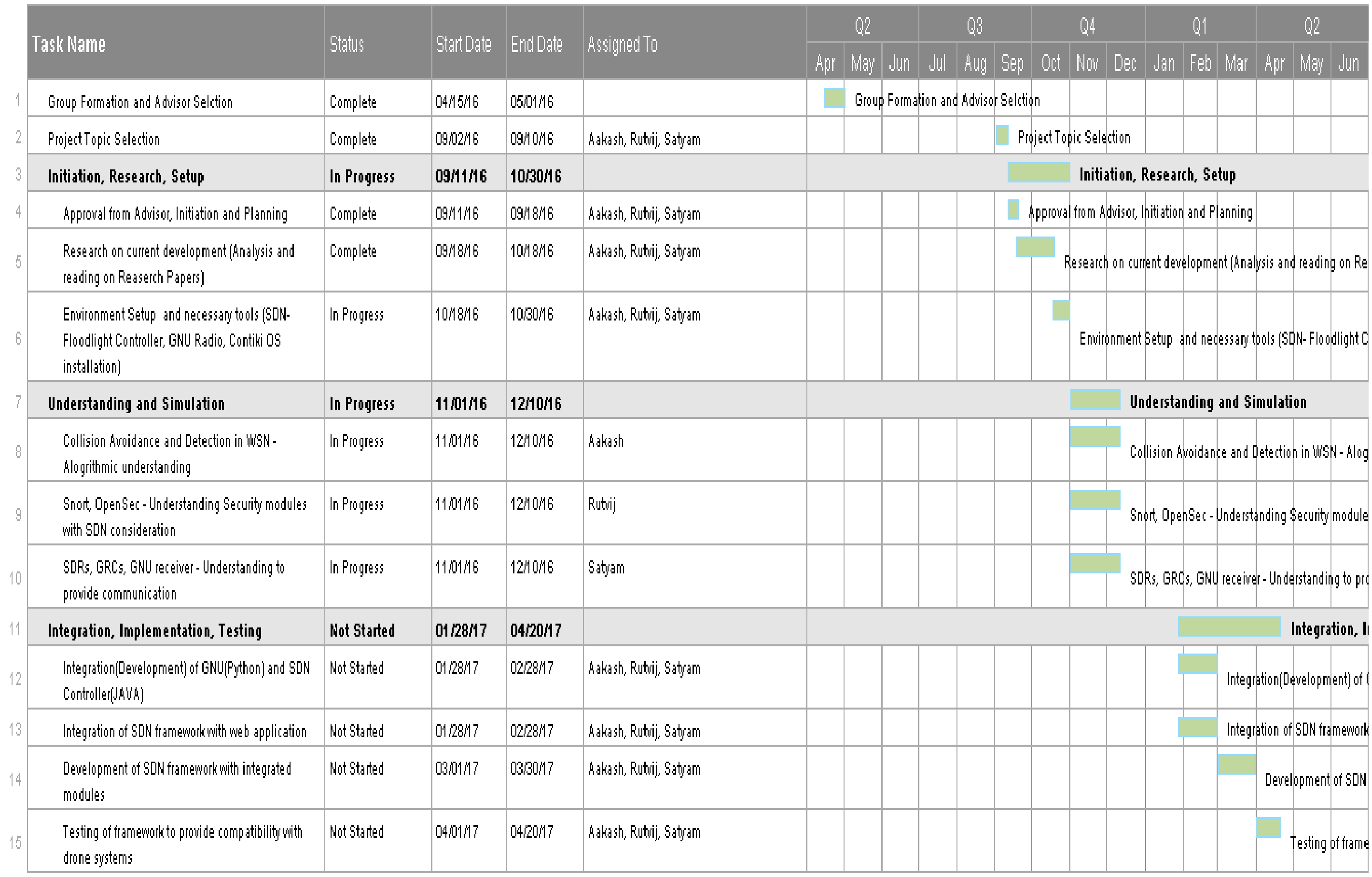
- **Progress status: To be started**
- There will be many optional features that will be implemented in our project:
  1. Automated Collision avoidance algorithms to resolve real world problems faced will be provided in which the drones can eventually learn to avoid the collision depending on the scenario.

2. The application is very basic application to just control and manage the drones. Cloud based web application can also be provided.

3. Wi-Fi module for communication between drone and SDN controller can be implemented as an optional communicating module.

Chapter 9: Project Schedule

9.1 Gantt Chart



	Task Name	Status	Start Date	End Date	Assigned To
1	Group Formation and Advisor Selction	Complete	04/15/16	05/01/16	
2	Project Topic Selection	Complete	09/02/16	09/10/16	Aakash, Rutvij, Satyam
3	**Initiation, Research, Setup**	**In Progress**	**09/11/16**	**10/30/16**	
4	Approval from Advisor, Initiation and Planning	Complete	09/11/16	09/18/16	Aakash, Rutvij, Satyam
5	Research on current development (Analysis and reading on Reaserch Papers)	Complete	09/18/16	10/18/16	Aakash, Rutvij, Satyam
6	Environment Setup and necessary tools (SDN-Floodlight Controller, GNU Radio, Contiki OS installation)	In Progress	10/18/16	10/30/16	Aakash, Rutvij, Satyam
7	**Understanding and Simulation**	**In Progress**	**11/01/16**	**12/10/16**	
8	Collision Avoidance and Detection in WSN - Alogrithmic understanding	In Progress	11/01/16	12/10/16	Aakash
9	Snort, OpenSec - Understanding Security modules with SDN consideration	In Progress	11/01/16	12/10/16	Rutvij
10	SDRs, GRCs, GNU receiver - Understanding to provide communication	In Progress	11/01/16	12/10/16	Satyam
11	**Integration, Implementation, Testing**	**Not Started**	**01/28/17**	**04/20/17**	
12	Integration(Development) of GNU(Python) and SDN Controller(JAVA)	Not Started	01/28/17	02/28/17	Aakash, Rutvij, Satyam
13	Integration of SDN framework with web application	Not Started	01/28/17	02/28/17	Aakash, Rutvij, Satyam
14	Development of SDN framework with integrated modules	Not Started	03/01/17	03/30/17	Aakash, Rutvij, Satyam
15	Testing of framework to provide compatibility with drone systems	Not Started	04/01/17	04/20/17	Aakash, Rutvij, Satyam

Q2
Q3
Q4
Q1
Q2
Apr
May
Jun
Jul
Aug
Sep
Oct
Nov
Dec
Jan
Feb
Mar
Apr
May
Jun
Group Formation and Advisor Selction
Project Topic Selection
Initiation, Research, Setup
Approval from Advisor, Initiation and Planning
Research on current development (Analysis and reading on Re
Environment Setup and necessary tools (SDN- Floodlight C
Understanding and Simulation
Collision Avoidance and Detection in WSN - Alog
Snort, OpenSec - Understanding Security module
SDRs, GRCs, GNU receiver - Understanding to pro
Integration, I
Integration(Development) of (
Integration of SDN framework
Development of SDN
Testing of frame

Figure 2: Gantt Chart showing task assignments and schedule

9.2 Pert Chart

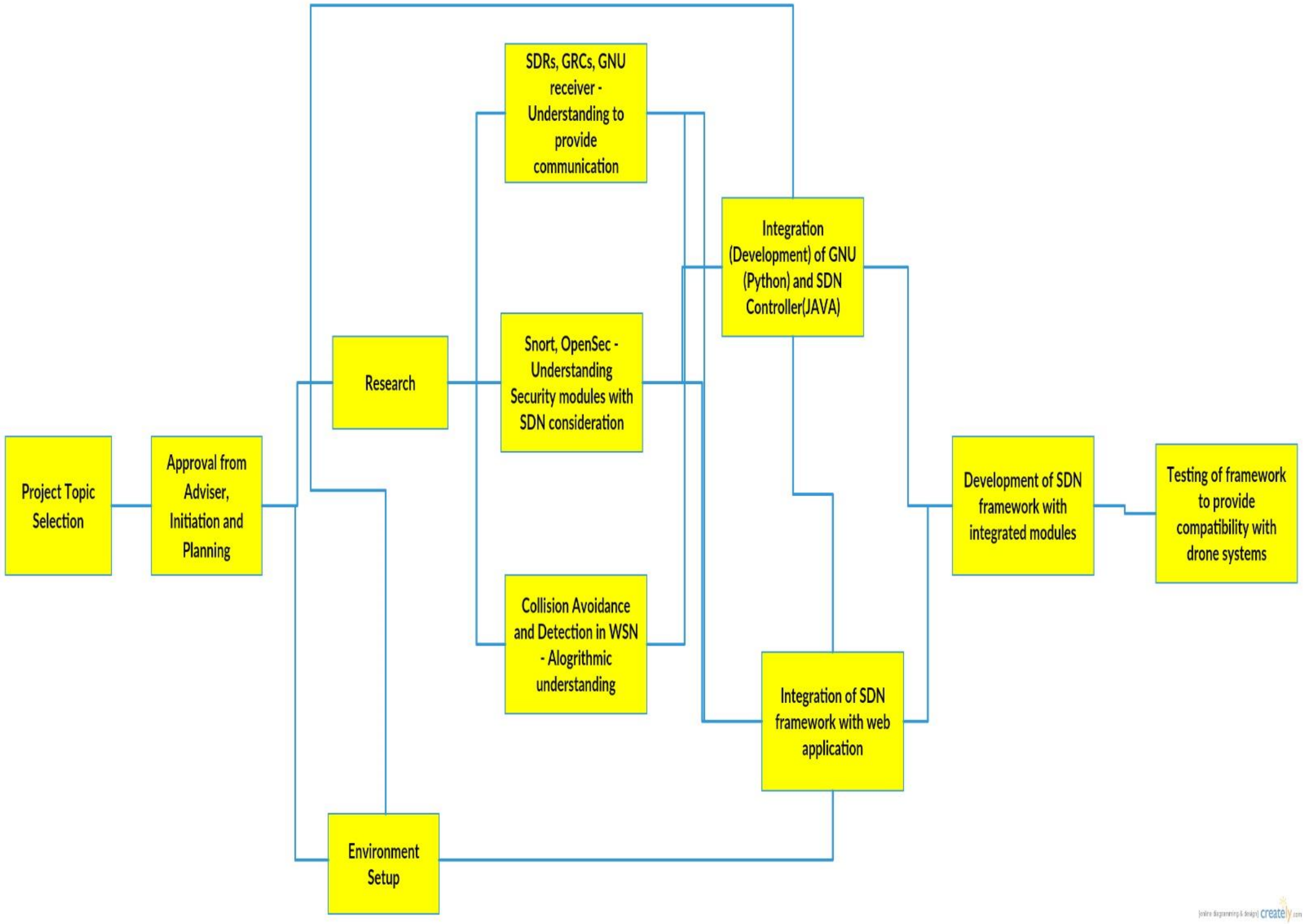

Project Topic Selection
Approval from Adviser, Initiation and Planning
Research
Environment Setup
SDRs, GRCs, GNU receiver - Understanding to provide communication
Snort, OpenSec - Understanding Security modules with SDN consideration
Collision Avoidance and Detection in WSN - Alogrithmic understanding
Integration (Development) of GNU (Python) and SDN Controller(JAVA)
Integration of SDN framework with web application
Development of SDN framework with integrated modules
Testing of framework to provide compatibility with drone systems

Figure 3: Pert Chart

Chapter 10: Contribution in Document

Chapter	Name	Contribution
1.	Literature Search, State of the Art and References	Aakash, Rutvij, Satyam
2.	Project Justification	Satyam
3.	Project Requirements	Aakash, Rutvij, Satyam
4.	Dependencies and Deliverables	Aakash, Rutvij, Satyam
5.	Project Architecture	Aakash, Rutvij, Satyam
8.	Implementation Plan and Progress	Rutvij
9.	Project Schedule	Aakash