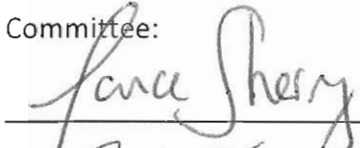


METHODS FOR UNIVERSAL BEACON CODE ASSIGNMENT

by

Vivek Kumar
A Dissertation
Submitted to the
Graduate Faculty
of
George Mason University
in Partial Fulfillment of
The Requirements for the Degree
of
Doctor of Philosophy
Systems Engineering and Operations Research

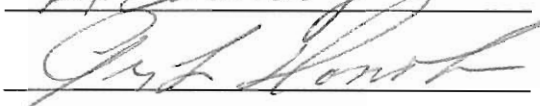
Committee:



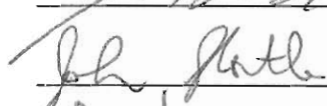
Dr. Lance Sherry, Dissertation Director



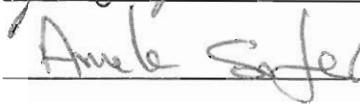
Dr. Alexander Brodsky, Committee Member



Dr. George Donohue, Committee Member



Dr. John Shortle, Committee Member



Dr. Ariela Sofer, Department Chair



Dr. Lloyd J. Griffiths, Dean, Volgenau School
of Engineering

Date:

07/28/2011

Summer Semester 2011
George Mason University
Fairfax, VA

Methods for Universal Beacon Code Assignment

A dissertation submitted in partial fulfillment of the requirements for the degree of
Doctor of Philosophy at George Mason University

By

Vivek Kumar

Master of Science in Operations Research

George Mason University, USA, 2007

Bachelor of Technology in Information Technology

Indian Institute of Information Technology, India, 2005

Director: Lance Sherry, Professor
Systems Engineering and Operations Research

Summer Semester 2011
George Mason University
Fairfax, VA

Copyright: 2011 Vivek Kumar
All Rights Reserved

Dedication

This is dedicated to my late grandfathers Dwarka Das Agarwal and Jagdish Prasad Agarwal, and also to my late uncle Sanjay Agarwal.

Acknowledgements

This dissertation would not have been possible without the help of my friends, family, faculty, and colleagues.

First, I wish to acknowledge the role of my advisor, Dr. Lance Sherry, who patiently supported and steered me over the course of this dissertation. He helped me gain invaluable technical insight into the problem through his knowledge of the domain. Also, his persistent encouragement helped me to remain focused on my work.

Dr. George Donohue, an internationally renowned expert in the field of air transportation research, provided philosophical and technical guidance on the design, implementation, and most importantly presentation of the results of this research.

Dr. Alexander Brodsky's knowledge of optimization theory and his database expertise proved very helpful in formulating and proof-reading the optimization model formulations.

About 6 years ago, Dr. John Shortle introduced me to the domain of air transportation research. As a member of my dissertation committee, he provided me with guidance on defining the scoping of the problem and suggested techniques to handle the stochastic factors in the system.

My appreciation also goes to all the faculty members in the SEOR department at GMU, especially to Dr. Karla Hoffman who kindly helped me in the formulation of the optimization algorithms.

In addition there were several others who provided technical assistance to the research: Pam Dees, Richard Jehlen, Diana Liang, Geoffrey Barker, Paul Eure, Bryan Baszczewski, John Guilding (FAA), Dr. Terrence Thompson, Dr. Arash Yousefi, Dr. Girish Sabhnani, Dr. Rafal Kicingier, Chris Ermatinger and Jason Burke (Metron Aviation Inc.). I would also like to express my appreciation for my fellow students at the Center for Air Transportation

Systems Research (CATSR) for their help with many technical and data related questions and for their friendship during the research process.

I am forever indebted to my parents and family for their constant encouragement and support in letting me pursue my Ph.D. My father has always been a constant source of inspiration to me through his selfless devotion to family and profession. I would also like to thank my friends Abhishek, Adina, Animesh, Surinder, Akshay, and Audarya for their support. I would especially like to thank Abhishek Saurav, my friend from high school for his help on my dissertation. His positive attitude towards life has always been a constant source of inspiration for me.

Last, but most of all, thank you Lilly, for all your support and sacrifice through this journey. Without your encouragement and patience, we could not have made it through this Ph.D. together.

Table of Contents

	Page
List of Tables	xii
List of Figures	xiv
List of Abbreviations	xvii
Abstract	xx
Chapter 1: Introduction	1
1.1. The Air Traffic Control (ATC) and Radars	2
1.2. Role of Beacon Codes	3
1.3. The Current Beacon Code Allocation System	4
1.4. Code Reassignment Frequency and Likelihood	9
1.5. Summary of Results	11
1.6. Unique Contributions of the Research	11
Chapter 2: Background and Literature Review	13
2.1. Description of ATC and History of ATCRBS	13
2.1.1 The ATCRBS (Air Traffic Control Radar Beacon System)	15
2.2. Description of Existing Beacon Code Allocation Method	18
2.3. Beacon Code Reassignment Scenarios	20
2.3.1. Competing Center Scenario	21
2.3.2. Overtaking Scenario	22
2.4. Research on Alternate Beacon Codes	22
2.3.1 Code Assignment by Airline (9 Airlines interaction based allocation)	23
2.3.2 Altitude Strata Code Assignment Plan (Codes assigned by Altitude, Reserved codes for Climb/Descent)	24

2.3.3 Directional Code Assignment Plan (Same codes shared by flights operating in geographically independent regions)	24
2.3.4 Master Assignment Plan (Flight Plan aware assignment of BC)	25
2.3.5 Geographic Beacon Code Allocation	25
2.5. Objectives of this Research	26
Chapter 3: Data Sources and Statistics on Beacon Code Usage and Reassignments	28
3.1 Data Sources	29
3.1.1 DoT/FAA order 7110.66D	29
3.1.2 Host Data	30
3.1.3 ETMS 4-D Trajectory Data	32
3.2 Beacon Code Usage Statistics	33
3.3 Beacon Code Reassignment Statistics	36
Mean	37
Standard Deviation	37
3.4 Beacon Code Demand	40
3.4.1 Beacon Code Life Cycle	41
3.5 Summary	44
Chapter 4: Individual Flight Code Assignment Optimization Model	46
4.1 Definition of Indices and Preparing Data for Optimization	47
4.1.1 Time Horizon Definition	47
4.1.2 Indices	47
4.1.3 Preprocessing data	47
4.2 Description of Optimization Model	48
4.2.1 Decision variables	49
4.2.2 Objective Function	49
4.2.3 Constraints	50
4.2.4 Additional Constraints	51
4.3 Results	52

4.4 Conclusion	53
Chapter 5: Space-Time Adjacency (STA) Algorithms	55
5.1 Algorithm Overview	57
5.2 STA Algorithm.....	59
5.2.1 Data Structures and Parameters.....	59
5.2.2 STA Algorithm Details	65
5.2.3 Results for Code Assignment through STA for Current Traffic (2007)	69
5.2.4 Results for Code Assignment through STA for 1.5x Traffic Projection (2032) .	77
5.3 STA-R (STA with Reassignments) Algorithm.....	83
5.3.1 Data Structures and Parameters.....	83
5.3.2 STA-R Algorithm Details.....	85
5.3.3 Results for Code Assignment through STA-R for 1.5x Traffic Projection (2032)	91
5.4 Conclusion	93
Chapter 6: Hybrid Method of Beacon Code Assignment Through Clusterization of CONUS	94
6.1 Definitions	95
6.1.1 Clusters	95
6.1.2 Intra-Cluster Flights	96
6.1.3 Inter-Cluster Flights	96
6.2 Overview of Hybrid Code-Assignment method through clusterization.....	97
6.2.1. Clusterization.....	97
6.2.2. Hybrid Code assignment method	99
6.3 Clusterization.....	100
6.3.1 Relationship between Number of Clusters and Number of Codes Used(K) ..	100
6.3.2 Goals of Clusterization.....	101
6.3.3 Challenges in Formulation as a Warehouse Location Problem.....	101
6.4 Description of Optimization Model.....	106

6.4.1 Indices and parameters	106
6.4.2 Decision variables	108
6.4.3 Objective Function.....	108
6.4.4 Constraints.....	110
6.5 Results	113
6.5.1. Results of Clusterization	113
6.5.2. Results of Code-Assignment.....	117
6.5.3 Summary of Results for Code-Assignment for all CONUS Flights	122
Chapter 7: Conclusions and Future Work.....	126
7.1 Problem Magnitude	128
7.2 Code assignment method for current traffic volume (2007).....	129
7.2.1 STA heuristic algorithm method of code assignment for current traffic volume	129
7.2.2 Hybrid method (Clusterization and STA) of code assignment for current traffic	129
7.3 Code assignment method for 1.5x Traffic (traffic volume projections of 2032) ..	131
7.4 Additional insights gained through this research	132
7.4.1 Boundary of feasibility.....	132
7.4.2 Traffic-pattern in NAS	133
7.4.3 Effect of number of clusters of CONUS on the number of Beacon Codes required	134
7.5 Future Work	137
7.5.1 STA with dynamic time-windows	137
7.5.2 Refine code-reassignment in STA-R algorithm.....	138
7.5.3 Space-Time Adjacency matrix data structure.....	139
7.5.4 Sensitivity analysis for NextGen CTOP (Collaborative Trajectory Options Program)	139
7.5.5 Distributed Transaction Management	140
Appendix A: 4DT-to-Center-Route Converter.....	141

A.1 Latitude-longitude to Cartesian coordinate system conversion	141
A.2 4-D trajectory to ARTCC network mapping algorithm.....	143
A.2.1 4-D trajectory to ARTCC network mapping algorithm details	143
A.2.2 Ray casting algorithm.....	146
Appendix B: NBCAS (National Beacon Code Allocation Plan(NBCAP) Simulator)	148
B.1 NBCAS algorithm details.....	148
References	152

List of Tables

Table	Page
Table 1: Maxim Utilization Fraction for External Primary and Secondary Codes for 153 days of 2007 (Host Data).....	35
Table 2: Max Utilization Fraction for Internal Primary and Secondary Codes for 153 days of 2007 (Host Data).....	36
Table 3: Number of Hand-Offs and Reassignments (Source: ETMS Data)	37
Table 4: Number of Beacon Code Reassignments for 17 days (Lucic, 2005)	39
Table 5: Time-Out Matrix (TOM) (3348 X 20 elements)	62
Table 6: Statistics of Traffic in the CONUS for the 5 days of 2007 used as input for STA	69
Table 7: Summary of the total number of codes required using STA for different values of "host-prediction" uncertainty (minutes).....	70
Table 8: Sensitivity of Codes-required to Change in host-prediction Uncertainty.....	73
Table 9: Statistics of 1.5x Traffic in the CONUS for 2 days of 2032 used as input for STA-R	78
Table 10: Summary of Results for Code-Assignment for 1.5x Traffic through STA algorithm.....	80
Table 11: Summary of Results for Code-Assignment for 1.5x Traffic through STA-R algorithm.....	92
Table 12: Shortest-Distance of ARTCC-pairs in the CONUS($p(i,j)$)	104
Table 13: "Exp Distance" of ARTCC-pairs in the CONUS with $X=4.(d(i,j))$	106
Table 14: The Total Number of Inter-Cluster and Intra-Cluster Flights with 3 Cluster CONUS.....	116
Table 15: Maximum Instantaneous Intra-Cluster "Active-Flights" Count/Number of Codes Required for FCFS Assignment. (Color of clusters corresponds to the coloring in Figure 43)	118
Table 16: Maximum Number of Codes Required for Code assignment to Intra-Cluster flights using STA algorithm	120
Table 17: Maximum Instantaneous Inter-Cluster "Active-Flights" Count/Number of Codes Required for FCFS Assignment	121

Table 18: Maximum Number of Codes Required for Code assignment to Inter-Cluster flights using STA algorithm	122
Table 19: Options for Code-Assignment To Intra-Cluster and Inter-Cluster Flights for 3 Cluster CONUS	123
Table 20: Maximum number of Beacon Codes required With 3 Clusters (Intra-Cluster Flights assigned Codes in FCFS order and Inter-Cluster flights assigned codes through STA Algorithm) – Code-Assignment Alternative 1	125
Table 21: Maximum number of Beacon Codes required with 3 Clusters (Code assignment through STA Algorithm for both Intra-Cluster and Inter-Cluster Flights) – Code-Assignment Alternative 2.....	125
Table 22: Variation of number of intra-cluster and inter-cluster flights and the number of codes needed for different number of clusters for code assignment using Hybrid Cluster Method. Codes assigned using STA for both Intra-cluster and Inter-cluster flights. [Day: 26 th July 2007, Host-Prediction Uncertainty: 0 minutes]	134

List of Figures

Figure	Page
Figure 1: Typical Ground Radar. SSR(Secondary Surveillance Radar) mounted on top of PSR(Primary Surveillance Radar)	3
Figure 2: A typical ATCRBS transponder (located in cockpit). Manufacturer: Honeywell International Inc.....	4
Figure 3: The 20 Centers in the CONUS (Google Earth Representation).....	5
Figure 4: Distribution of total number of Beacon Codes allocated to 20 ARTCCs in the CONUS as per the National Beacon Code Allocation Plan (DOT/FAA, 2009)	6
Figure 5: Histogram of Code-Sharing among the 20 ARTCCs in the CONUS derived from (DOT/FAA, 2009).	6
Figure 6: Flow diagram of Beacon Code Reassignment Process	7
Figure 7: Histogram of BC Reassignments for 153 days of HOST Data	10
Figure 8: ATCRBS system: Flow of Information (Bussolari, 2000)	16
Figure 9: The distinction between Mode A and Mode C interrogation pulses	17
Figure 10: Categorization of Codes into Primary, Secondary and Tertiary subsets for each ARTCC.....	19
Figure 11:A sample flight strip	20
Figure 12: Competing Center Scenario for Code Reassignment	21
Figure 13: Overtaking Scenario for Code Reassignment	23
Figure 14: Snapshot of Utilization Beacon (UB) Message from ARTCC HOST Data.....	31
Figure 15: Snapshot of Beacon Reassignment (BA) Message from ARTCC HOST Data	32
Figure 16: Maximum Code Utilization for each of the 20 ARTCCs (for 153 days of Host Data from 1 st July 2007 to 31 st December 2007)	34
Figure 17: Beacon Code Hand-Off and Reassignment Likelihood from ETMS Data.....	38
Figure 18: Count of "Active Aircraft" for each quarter-hour of the day (Source: ETMS 4-D Trajectory Data)	40
Figure 19: Departure Strip Printing Interval (DSPI) for 20 ARTCCs in the CONUS (Source: (FAA, 2007))	42

Figure 20: Count of "Beacon Code Demand" for each quarter-hour of the day (Source: ETMS 4-D Trajectory Data).....	43
Figure 21: Beacon Code Peak Demand by Individual Centers (Source: ETMS Data).	44
Figure 22:MILP Model for Individual Flight BC Assignment	49
Figure 23: Instantaneous traffic count by ARTCC for the experimental dataset.....	52
Figure 24: Assignment Frequency Chart of Codes allocated by the Optimization Model	53
Figure 25: Block diagram of STA Algorithm	57
Figure 26: Linearizing STA Matrix to Reduce Memory Requirement (Example of 4 flight case)	60
Figure 27: Boundary Crossing Time Uncertainty Buffer (Example of 15 minutes window)	65
Figure 28: Summary of the total number of codes required using STA for current traffic (2007)	71
Figure 29: Code Usage by Time-Of-Day using STA Algorithm for 3rd Jan 2007	74
Figure 30: Code Usage by Time-Of-Day using STA Algorithm for 11th April 2007	75
Figure 31: Code Usage by Time-Of-Day using STA Algorithm for 26th July 2007	76
Figure 32: Code Usage by Time-Of-Day using STA Algorithm for 21st Nov 2007.....	76
Figure 33: Code Usage by Time-Of-Day using STA Algorithm for 19th Dec 2007	77
Figure 34: Count of "Active Aircraft" for 1.5x Traffic for each quarter-hour of the day ..	78
Figure 35: Summary of the total number of codes required using STA for 1.5x schedule	81
Figure 36: Code Usage by Time-Of-Day using STA Algorithm for 3rd Jan 2032(1.5x Traffic)	81
Figure 37: Code Usage by Time-Of-Day using STA Algorithm for 11 th April 2032(1.5x Traffic)	82
Figure 38: Hybrid Model for Universal Beacon Code Allocation System using Clusterization	99
Figure 39: Pseudocode for FloydWarshall Algorithm (Source: (<i>Introduction to Algorithms</i> , 2009, pp. 558-565)).....	103
Figure 40: Graph representation of CONUS in Node-Link format	103
Figure 41: MILP Model for Clusterization of CONUS	107
Figure 42: Cluster Boundaries for Three (K=2) Cluster Case	115
Figure 43: Cluster Boundaries for Three (K=3) Cluster Case	115
Figure 44: Total Number of Intra-Cluster and Inter-Cluster Flights for 2 Clusters of CONUS.....	116

Figure 45: Total Number of Intra-Cluster and Inter-Cluster Flights for 3 Clusters of CONUS.....	117
Figure 46: Variation of number of codes needed for different number of clusters for code assignment using Hybrid Cluster Method. Codes assigned using STA for both Intra-cluster and Inter-cluster flights. [Day: 26 th July 2007, Host-Prediction Uncertainty: 0 minutes]	135
Figure 47: One-to-one mapping of Coordinate Transformation	142
Figure 48: 4-D trajectory to ARTCC Network Path Conversion Process	145
Figure 49: Ray-Casting Algorithm for Point-in-Polygon determination.....	146
Figure 50: Format of Center-crossing Data(output of 4DT-to-Center-Route Converter)	147
Figure 51: Beacon Code Allocation Flow-Chart	149

List of Abbreviations

APO	Office of Aviation Policy and Plans (FAA)
ARTCC	Air Route Traffic Control Center
ATC	Air Traffic Control
ATCRBS	Air Traffic Control Radar Beacon System
ATL	Hartsfield - Jackson Atlanta International
BC	Beacon Codes
CAA	Civil Aeronautics Administration
CONUS	Contiguous United States
CRDT	Code Reassignment Delay Time
DCA	Ronald Reagan National Airport
DFW	Dallas Fort Worth International Airport
DOT	Department of Transportation
DSPI	Departure Strip Printing Interval
ETMS	Enhanced Traffic Management System
FAA	Federal Aviation Administration
FCFS	First Come First Served
HCS	Host Computer System

JPDO	Joint Planning and Development Office
MILP	Mixed-Integer Linear Program
NAS	National Airspace System
NBCAP	National Beacon Code Allocation Plan
PSR	Primary Surveillance Radar
SSR	Secondary Surveillance Radar
TAF	Terminal Area Forecast
TB	Termination Beacon Message
TWA	Trans World Airlines
ZAB	Albuquerque Center
ZAU	Chicago Center
ZBW	Boston Center
ZDC	Washington Center
ZDV	Denver Center
ZFW	Dallas Fort Worth Center
ZHU	Houston Center
ZID	Indianapolis Center
ZJX	Jacksonville Center
ZKC	Kansas City Center
ZLA	Los Angeles Center
ZMA	Miami Center
ZME	Memphis Center
ZMP	Minneapolis Center

ZNY	New York Center
ZOA	Oakland Center
ZOB	Cleveland Center
ZSE	Seattle Center
ZTL	Atlanta Center

Abstract

METHODS FOR UNIVERSAL BEACON CODE ASSIGNMENT

Vivek Kumar, Ph.D.

George Mason University, 2011

Dissertation Director: Dr. Lance Sherry

The primary responsibility of Air Traffic Control (ATC) is to expedite the flow of traffic while maintaining safe separation. Positive identification of the primary radar returns for individual aircraft is achieved through a system of interrogation and identification known as Air Traffic Control Radar Beacon System (ATCRBS). Each flight is identified by a unique “Beacon Code” assigned by the ATC before departure.

Due to installed equipment limitations, and reservation of a few codes for special usage, only 3,348 Beacon Codes are available for use by non-military flights. ATC must “reassign” Beacon Codes to flights when they enter an ARTCC (Air Route Traffic Control Center) in which their current code is already in use. Each instance of Beacon Code reassignment requires human intervention and this process is therefore vulnerable to human-errors. An undetected error may lead to misidentification of flights which results in reduced safety margin. For this reason, Beacon Code reassignments are undesirable.

On a typical day (04/11/2007) including 48,721 flights (non-military), 62,805 handoffs occurred, when flights crossed ARTCC boundaries. With the current distributed code allocation scheme and the existing route structure, 6,730 (10.7%) code reassignments were required. The current allocation method is also subject to code shortages as the volume of air-traffic grows.

The objective of this research was to develop a detailed understanding of the problem and enumerate and evaluate alternative methods to eliminate (or minimize) code “reassignments” and potential shortages. The methods were required to be robust in the face of routing variations necessitated by weather and also to the evolution of airline networks. This dissertation describes and evaluates three new alternate methods for centralized Beacon Code assignment that assign codes by exploiting the temporal and spatial opportunities available in the NAS:

1. A Mixed-Integer Linear Program (MILP) optimization model,
2. A Space-Time Adjacency (STA) heuristic algorithm, and
3. A hybrid approach combining MILP optimization and STA heuristic algorithm.

The results of this research demonstrate the feasibility of implementing a code assignment system that eliminates need for reassignment and is scalable to future traffic growth.

Chapter 1: Introduction

The primary purpose of Air Traffic Control (ATC) is to prevent collisions between aircraft operating in the National Airspace System (NAS), organize and expedite the flow of traffic, and to provide support for National Security and Homeland Defense (Nolan, 2007). Maps, blackboards and shrimp boats were used by early controllers to track the position of aircraft. Over time, increases in volume and complexity of traffic have led to improvements in surveillance, navigation and communication capabilities.

This chapter describes the functions of ATC and the role of radar in surveillance. The secondary radar, also known as Air Traffic Control Radar Beacon System (ATCRBS) in the United States, and the role of Beacon Code as flight identifier is described. Next, the current Beacon Code allocation method (DOT/FAA, 2009) is described. A description of the drawbacks of the current allocation plan is followed by analysis of historical data to quantify the magnitude of current problem. Next, the results of the new alternate methods of Beacon Code assignment developed during this research are summarized. A list of unique contributions of this research is summarized in the last section of this chapter.

1.1. The Air Traffic Control (ATC) and Radars

The primary responsibility of the Air Traffic Control (ATC) is to expedite the flow of traffic while maintaining safe separation. The ATC supports three major functions, namely, communication, navigation and surveillance. Surveillance is primarily achieved through radar. Radar was developed during the Second World War for tracking enemy aircraft and was later adapted for civilian use for separation assurance and coordination of air-traffic. Radar technology can broadly be classified into two types:

1. Primary radar is the traditional “skin paint” radar. It operates on the principle that rotating radar transmitters broadcast electromagnetic radio pulses of which a very small portion is reflected back from aircraft that falls in the path of these pulses. The azimuth orientation of the radar antenna and the time taken for the reflected pulse to return provides the bearing and distance of the target aircraft respectively (Nolan, 2007). Primary radar is *passive* as it relies solely on the equipment (rotating antenna) on the ground.
2. Secondary radar is a system used in ATC to provide surveillance radar monitoring and separation of aircraft by transmitting aircraft ID and/or altitude from the cockpit to the ground radar station. It consists of a Secondary Surveillance Radar (SSR) which is co-located at ATC with Primary Surveillance Radar (PSR), as shown in Figure 1 and a transponder which is located in the aircraft (see Figure 2). Unlike the primary radars which are *passive*, secondary radars are *active* as they rely on the transponder which responds to interrogation from ground station by transmitting a

coded reply signal that represents either *identification* or *altitude* of the aircraft. This system is also called ATCRBS (Air Traffic Control Radar Beacon System) in the United States.



Figure 1: Typical Ground Radar. SSR(Secondary Surveillance Radar) mounted on top of PSR(Primary Surveillance Radar)

1.2. Role of Beacon Codes

The transponders located in aircraft respond to interrogations from ground stations with four digit codes, known as *Beacon Codes* (hereafter referred to as BC or codes). Each of these four digits is octal (0 to 7) resulting in a total of $4,096(8^4)$ possible combinations. The reply of transponder represents either the aircraft's identity or altitude depending on the type of interrogation.

Beacon codes are used by ATC to identify aircraft on the radar display. Every aircraft within an ARTCC boundary must have a unique code assigned to it. Out of the 4,096 total code combinations, 748 codes are assigned to the military or reserved for other special usage. This leaves 3,348 codes available for civil aviation use (DOT/FAA, 2009).

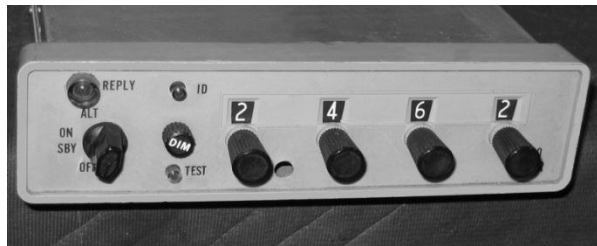


Figure 2: A typical ATRBS transponder (located in cockpit). Manufacturer: Honeywell International Inc.

1.3. The Current Beacon Code Allocation System

The Contiguous United States (CONUS) is subdivided into twenty ARTCCs as shown in Figure 3. The process of allocating Beacon Codes to flights in the National Airspace System (NAS) is owned and managed by FAA, and published in DOT/FAA orders which are revised periodically. The code allocation was last revised in November of 2009 and is published in (DOT/FAA, 2009). This order enlists the code blocks that are allocated to each of the ARTCCs in the CONUS.

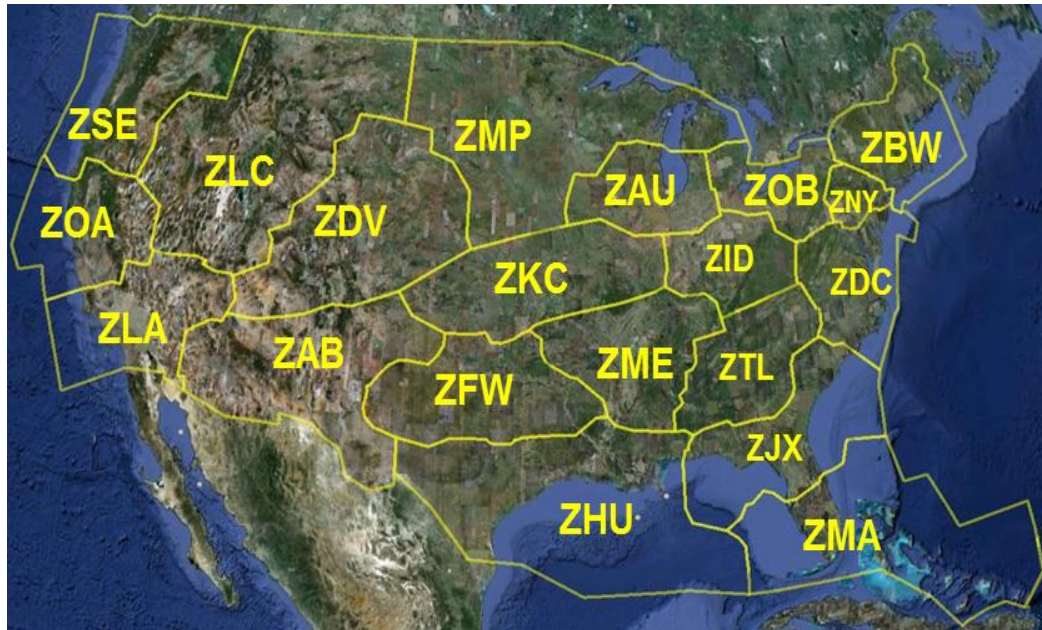


Figure 3: The 20 Centers in the CONUS (Google Earth Representation)

The current process of Beacon Code allocation is ARTCC-centric. Each of the 20 ARTCCs in the CONUS is pre-allocated a static subset of codes as per the DOT/FAA Order (DOT/FAA, 2009). The distribution of the number of codes allocated to all the ARTCCs is illustrated in Figure 4. The number of codes allocated to each ARTCC is not equal, and is dependent on the expected traffic, i.e. demand for codes. The center with the least number of codes allocated to it is ZKC (Kansas City City) center with 601 codes. The center with the most number of codes is ZMA (Miami) center with 1,559 codes.

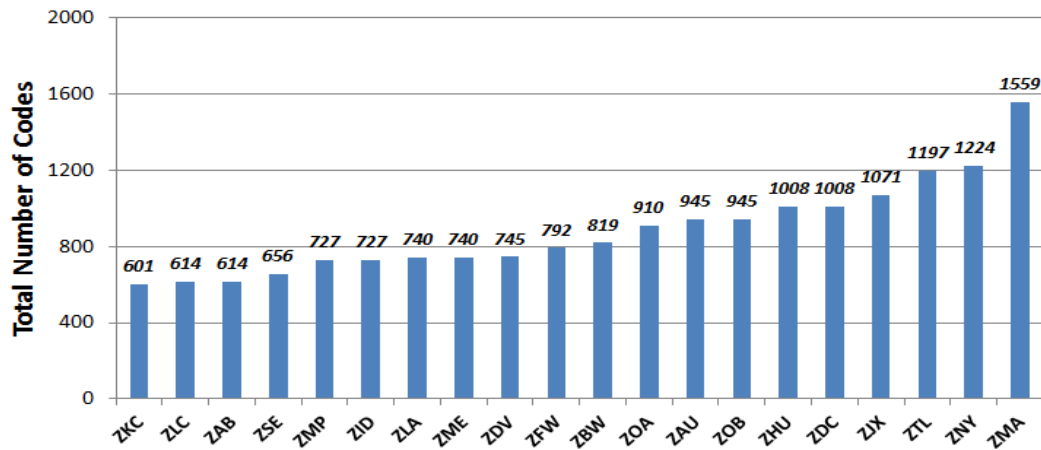


Figure 4: Distribution of total number of Beacon Codes allocated to 20 ARTCCs in the CONUS as per the National Beacon Code Allocation Plan (DOT/FAA, 2009)

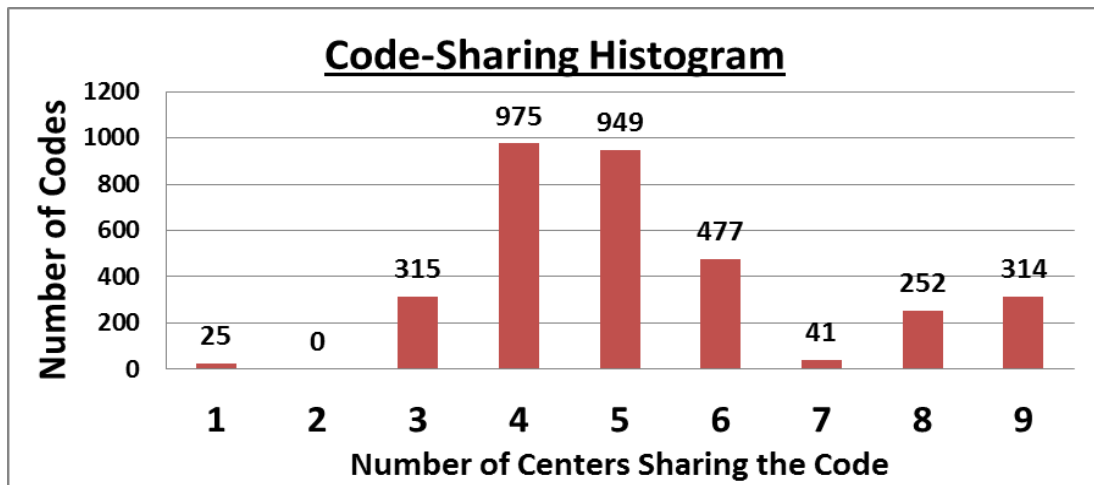


Figure 5: Histogram of Code-Sharing among the 20 ARTCCs in the CONUS derived from (DOT/FAA, 2009).

Ideally, flights could fly from their origin to destination using the same code for the entire flight duration. However, codes are limited (3,348) and as a result the code

subsets allocated to individual ARTCCs have overlapping set of codes as shown in Figure 5. For example, there are 975 codes which are shared by 4 ARTCCs. As a result of the sharing of codes among the ARTCCs it is likely that when a flight enters a new ARTCC enroute to its destination, its current code is already in use by another flight.

Whenever a flight enters an ARTCC with a code that is already in use, the Host Computer System (HCS) of the new ARTCC must assign another (non-conflicting) code to the incoming flight from its own sub-set of codes. This process is known as Beacon Code reassignment.

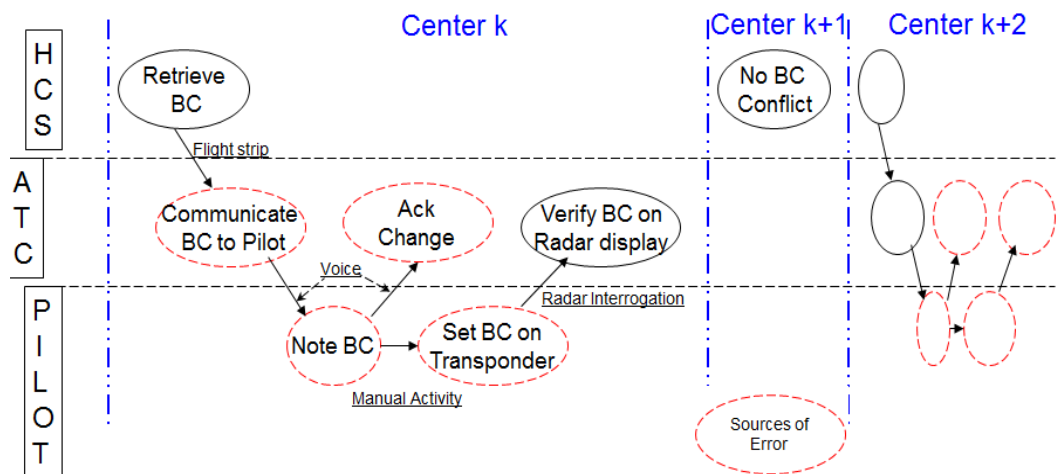


Figure 6: Flow diagram of Beacon Code Reassignment Process

Each instance of Beacon Code reassignment is achieved through a sequence of processes as shown in Figure 6. Initially, the HCS (Host Computer System) of the ARTCC

retrieves a valid BC from its subset of codes. Then the ATC communicates this BC via voice to the pilot. The pilot acknowledges the ATC's communication via voice and makes a note of the new BC. Next, the pilot manually adjusts the transponder knobs to the new BC. The ATC then verifies this change by a radar interrogation.

The voice-communications between the pilot and the ATC and also the adjustment of transponder knobs to the new BC by pilot are processes which require human intervention and are vulnerable to human-error. These human-error prone processes are shown in dotted red boxes in Figure 6. Any human-error, if undetected in the BC reassignment process would lead to flight squawking an erroneous code, which may lead to a safety hazard due to misidentification of the flight.

Eliminating reassignments also allows for more efficient Host Computer System (HCS) software improvements as the system moves to higher degree of automation. In the Next Generation Air Transportation System (NextGen) report the multi-agency Joint Planning and Development Office (JPDO) describes an expected two-to threefold increase in air traffic demand by the year 2025 and the need for new automation technology and operating procedures in the National Airspace System (Joint Planning and Development Office, 2004).

The current approach of allocating static subset of codes to individual ARTCCs is not robust to accommodate the seasonal fluctuations in code demand (because of increased number of flights) caused in certain geographic regions and may lead to

localized code shortage in the corresponding centers. For example, Miami experiences heavy traffic in winters which may lead to shortage of codes. When an ARTCC exhausts all the codes in its subset, then the ATC starts assigning non-discrete codes to flights. This process of assigning non-discrete Beacon Codes is workload intensive for the ATC and reduces the safety margin because a flight may respond to ATC communications intended for another flight on the same Beacon Code.

1.4. Code Reassignment Frequency and Likelihood

Historical data was analyzed from two “independent” data sources, namely, Host data and Enhanced Traffic Management System (ETMS) data, to quantify the number of Beacon Code reassignments and establish a baseline.

An analysis of historical ETMS data for 5 days of 2007 yielded that on an average, there are 62,111 hand-offs (Table 3) per day in NAS. The ratio of the number of code-reassignments and number of hand-offs is the likelihood that a flight crossing a center boundary gets a new Beacon Code.

The number of BC reassignments in NAS for 153 days of Host data analyzed over the period of 1st August 2007 to 31st December 2007 is shown in Figure 7. The average is 7,642 with a standard deviation of 1,451. This is equivalent to an average reassignment likelihood of 12.3 % ($7,642/62,111$).

The average number of BC reassignments for the 5 days of ETMS 4-D trajectory data analyzed is 6,208. The reassignment likelihood is in the range of 9.2%-10.7% with an average of 9.96% (See Table 3).

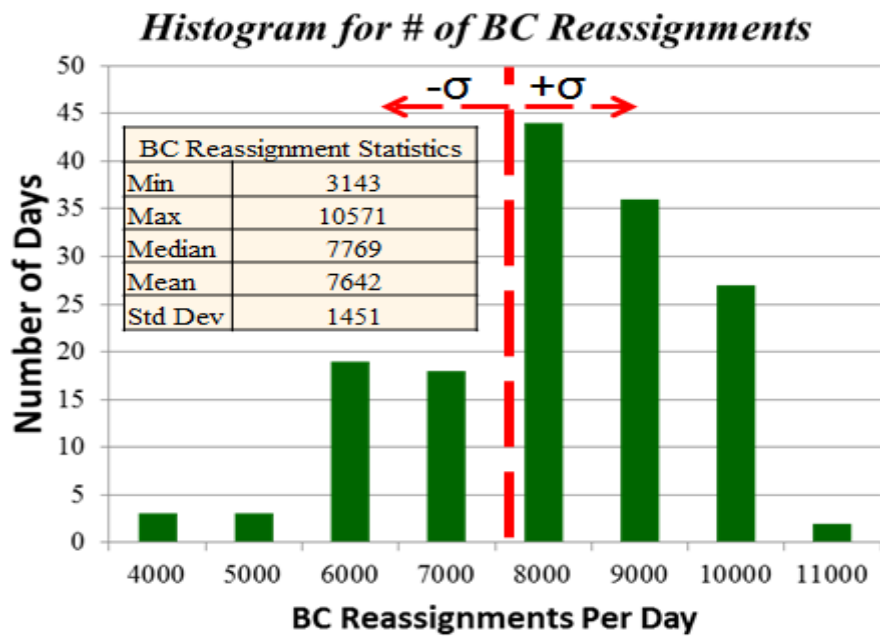


Figure 7: Histogram of BC Reassignments for 153 days of HOST Data

The BC assignment methods discussed in this dissertation are designed to eliminate (or reduce) the likelihood of code reassignments and as a result, increase safety margins, improves Host software efficiency and reduces ATC/pilot workload.

1.5. Summary of Results

The average likelihood of code reassignments from data analyzed is 9.96% (Table 3). Three alternative methods were developed for Beacon Code assignment. The Space Time Adjacency (STA) heuristic algorithm presented in the dissertation achieves a 100% improvement over the existing system by eliminating the need for code-reassignments. The Mixed Integer Linear Program (MILP) formulation proved infeasible due to computational limitations. The hybrid method derived from a combination of clusterization and STA heuristic algorithm eliminates reassignments for the current day traffic (2007 data).

The methods presented in the dissertation are also tested for 1.5x projection of current traffic volumes. Assuming the best case, where the likelihood of code-reassignments using the current system (Description of Existing Beacon Code Allocation Method) remains the same for 1.5x traffic, the STA algorithm achieves an 87% reduction (9.96% to 1.29%) in code-reassignment likelihood by reducing the likelihood of code-reassignment to 1.29%.

1.6. Unique Contributions of the Research

This research presents novel methods for Universal Beacon Code assignment that assigns codes to flights by exploiting the temporal and spatial opportunities available in the NAS so that the likelihood of code reassignments is minimized. The methods presented are Mixed Integer Linear Program (MILP) based optimization

(Chapter 4), heuristic algorithm (Chapter 5), and a hybrid method (Chapter 6) which combines MILP optimization and heuristic algorithmic approach.

Also, the robustness of the proposed Beacon Code assignment methods has been verified for different traffic pattern days including different weather scenarios and enhanced future traffic levels.

Also described is an algorithm to convert the 4-D trajectory of flights into a time-ordered sequence of ARTCCs that a flight goes through enroute to its destination along with the entry and exit time at each ARTCC along the flight's route (Appendix A). Also, a National Beacon Code Allocation Plan simulator was built that implements all the rules and procedures of Beacon Code allocation in the current system (Appendix B).

Chapter 2: Background and Literature Review

A description of the Air Traffic Control (ATC) and a brief summary of the history and functioning of ATCRBS are described in the first subsection of this chapter. In the second subsection, the current process of Beacon Code allocation is described. In the following subsection, a literature review of past relevant work on this topic is presented. In the final subsection of this chapter, the need statement for this research is stated.

2.1. Description of ATC and History of ATCRBS

The primary purpose of the Air Traffic Control (ATC) system is to prevent collisions between aircraft operating in the system, organize and expedite the flow of traffic, and to provide support for National Security and Homeland Defence (Nolan, 2007).

Maps, blackboards and shrimp boats were used by early controllers to track the position of aircraft (Nolan, 2007). Over time, increase in volume and complexity of traffic has led to improvements in surveillance, navigation and communication capabilities. In 1930, the first radio-equipped control tower was established at Cleveland Municipal Airport. Increased traffic levels created the need for extending ATC services to en-route phases of the flights. This led to the opening of the first Airway Traffic Control

Center at Newark in December 1935. In 1936, en route ATC became Federal responsibility and the Government started providing air traffic control services.

Advances in the field of flight navigation and surveillance technologies in years leading up to and during World War II led to the development of radar. Radar is a system that uses radio waves to detect distant objects. Deployment of radar enabled the controllers to see aircraft position on visual displays. This technology was eventually incorporated by Civil Aeronautics Administration (CAA) for surveillance and control of civil flights.

In 1937, the Naval Research Laboratory (NRL) developed the first U.S. radio recognition Identification Friend or Foe (IFF) system, the Model XAE, which met an urgent operational requirement to allow differentiation of friendly aircraft from enemy aircraft in World War II. The Mark X IFF was a later radar beacon system developed by NRL. It was essential to the military because it reduced fratricide when used with beyond-visual-range weapons.

By 1958, the FAA (Federal Aviation Administration) had established the Air Traffic Control Radar Beacon System (ATCRBS), which is the civil version of the Mark X. This new system required flights in certain positive control areas (high-volume air traffic areas near airports, IFR traffic under ATC guidance and ADIZ (Air Defense Identification Zone) to carry a radar beacon called a transponder. This transponder uniquely identified individual aircraft yielding improvement in radar performance and surveillance. The

International Civil Air Organization (ICAO) later adopted the ATCRBS, making the Mark X the basis of the world's air traffic control system.

2.1.1 The ATCRBS (Air Traffic Control Radar Beacon System)

ATCRBS is a system used in ATC to enhance surveillance radar monitoring and separation of aircraft by working in conjunction with primary radar to produce a synchronized surveillance. The two major components of the ATCRBS system are:

- (i) A Secondary Surveillance Radar (SSR), as shown in Figure 1, is the part of ATCRBS co-located at ATC with Primary Surveillance Radar (PSR). It transmits interrogations and listens for response.
- (ii) A transponder, as shown in Figure 2, is located in the aircraft is usually mounted in the avionics rack. Installations typically also include the altitude encoder, which is connected to the transponder and aircraft's pitot-static system to provide pressure altitude information to the transponder (for mode C interrogation)

2.1.1.1 ATCRBS Operation

The ATCRBS interrogator at the ATC facility on the ground, shown in Figure 8, periodically interrogates aircraft on a frequency of 1030 MHz using the radar's rotating antenna at the assigned Pulse Repetition Frequency (PRF)(Nolan, 2007). Typical frequency of interrogation is 450-500 per second. The interrogation travels at the speed of light in the direction of the antenna. Upon receiving an interrogation, aircraft reply with requested information (altitude or identification) at 1090 MHz after a 3 micro

second delay. The interrogator then decodes the reply and identifies the aircraft. The aircraft position is determined by the delay between interrogation and reply and antenna bearing.

The transponders typically have four operating modes: Off, Standby, On (Mode-A) and Alt (Mode-C). The only difference between the On and Alt modes is that when the transponder is in the On mode, it does not transmit any altitude information. The Standby mode allows the unit to remain powered but it inhibits any replies.

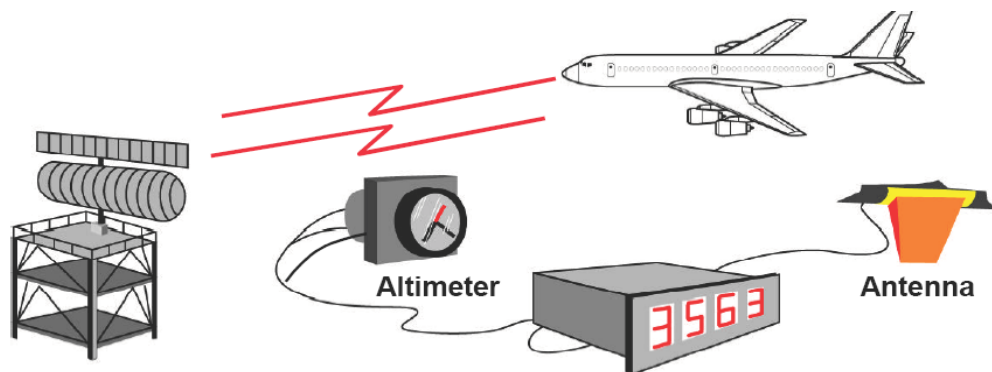


Figure 8: ATCRBS system: Flow of Information (Bussolari, 2000)

Interrogation consists of three pulses. Each of them is 0.8 microseconds in duration and is referred to as P1, P2 and P3 (Figure 9). The time interval between P1 and P3 determines the type of interrogation. P2 is used for side-lobe suppression. If P1 and P3 are separated by 8 microseconds the interrogation is of type Mode 3/A. The reply

expected from the aircraft is the beacon/squawk code. If P1 and P3 are separated by 21 microseconds, it is a mode C type interrogation, requesting aircraft pressure altitude from the transponder. There is no difference between a Mode A and Mode C reply. The decoding of the reply depends on the type of interrogation issued.

If the ground station sends a mode 3/A interrogation, the transponder replies with a string of pulses that are the squawk code only. If the interrogation is mode C, the reply is altitude only. Each altitude code has an equivalent squawk code. This means that the same data would decode as a squawk rather than an altitude. But each squawk code does not necessarily have an equivalent altitude. There are 4,096 identification codes but only 1,280 altitude codes, one for each 100 foot increment from -1,200 to 126,700 ft.

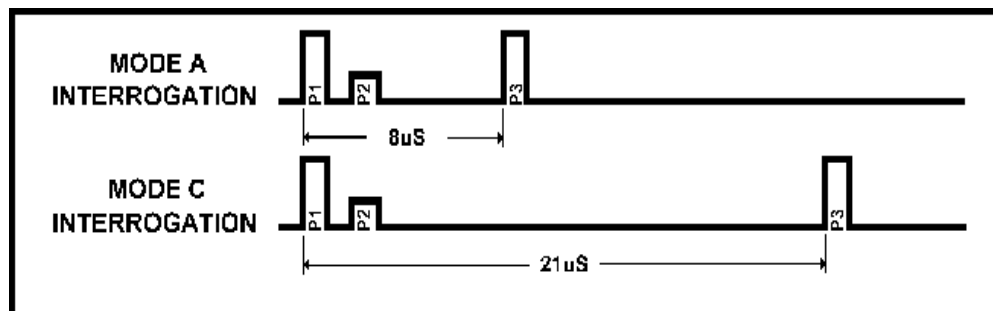


Figure 9: The distinction between Mode A and Mode C interrogation pulses

2.2. Description of Existing Beacon Code Allocation Method

The current system of Beacon Code allocation is ARTCC-centric. Each of the 20 ARTCCs in the CONUS is pre-allocated a static subset of codes as per the DOT/FAA Order (DOT/FAA, 2009). The distribution of the number of codes allocated to all the ARTCCs is illustrated in Figure 4. The number of codes allocated to each ARTCC is not equal and dependent on the expected traffic, i.e. demand for codes. The center with the least number of codes allocated to it is ZKC (Kansas City) with 601 codes. The center with the least number of codes is ZMA (Miami) with 1,559 codes. As the total number of codes is fixed (3,348), codes are shared by multiple ARTCCs (See Figure 5). For example, Beacon Code 2101 is allocated to both ZKC (Kansas City center) and ZMA (Miami center) centers.

Codes allocated to each ARTCC can be either external or internal. Internal codes are assigned to flights with flight-plans that do not cross the ARTCC boundary. For example, a flight from LAX (Los Angeles International Airport) to SFO (San Francisco International Airport) would be assigned an internal code by ZLA (Los Angeles Center). All other codes are external codes, and are to be assigned to flights that cross at least one ARTCC boundary.

The external and internal codes are further subdivided into primary, secondary and tertiary codes as shown in Figure 10. This categorization of codes represents the search order. Whenever an ARTCC needs to assign codes to a flight it looks for codes in

the primary bucket first, and then if needed in secondary and tertiary. In a given ARTCC code list, each code can only be in one of these six categories.

A code which is internal for one ARTCC may be an external code for another ARTCC. Another example is code 2677, which is an external code for ZAB (Albuquerque center) but internal code for ZTL (Atlanta center). Also, adjacent ARTCCs never share an External Code.

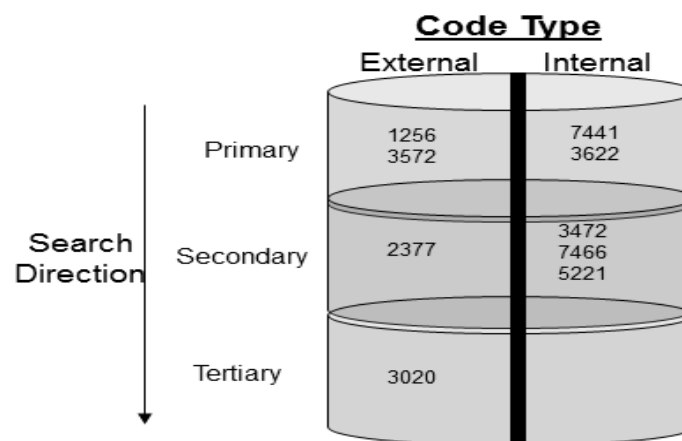


Figure 10: Categorization of Beacon Codes into Primary, Secondary and Tertiary subsets for each ARTCC

A flight is assigned its first Beacon Code by the Host Computer System (HCS) of the departure center. The HCS searches for codes in the appropriate order. Codes are allocated first from the primary bucket, and then secondary and tertiary if needed. The

primary and secondary codes are searched in a cyclic fashion, whereas the tertiary codes are searched top-down (FAA, 2007). The beacon/squawk code retrieved by the HCS is printed on the flight strip (See Figure 11) along with other information for ATC. The ATC relays the Beacon Code to pilot via VHF (Very High Frequency) communication (radio).

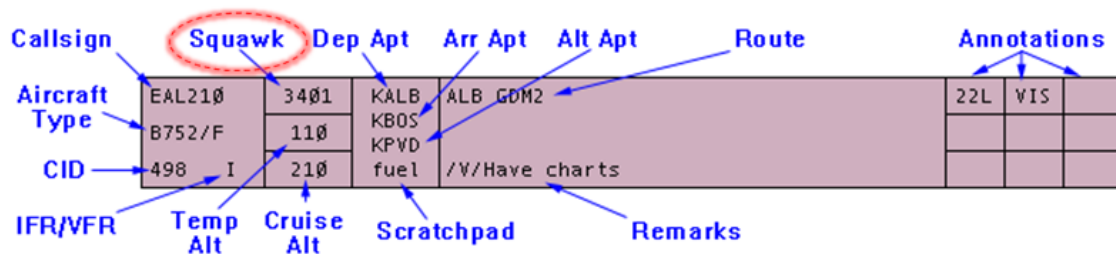


Figure 11:A sample flight strip

Before a flight crosses into a new ARTCC enroute to its destination airport, the HCS of that ARTCC checks whether the code being used by the incoming flight is also being used by any other flight in that center. If so, the HCS assigns another code from its bucket of external codes to the flight. This process of a flight getting a new Beacon Code assigned to it by an enroute center is called Beacon Code reassignment process.

2.3. Beacon Code Reassignment Scenarios

There are two scenarios in which Beacon Code reassignment occurs:

2.3.1. Competing Center Scenario

The “competing center” scenario occurs when two flights departing from airports in different centers are assigned the same code and they are in a common downstream center at the same time. In this situation, the flight which enters the downstream center second is reassigned a Beacon Code.

For example, the two flights displayed in Figure 12 start from different origin ARTCCs and head towards ZDC (Washington DC Center). Flight A (shown in red), is headed from Kansas to Washington DC and flight B (cyan), from Miami to Washington DC. If both flights happen to be assigned the same code 2101 by their origin ARTCCs, then flight B is reassigned a new Beacon Code by ZDC because flight A would already be using code 2101.



Figure 12: Competing Center Scenario for Code Reassignment

2.3.2. Overtaking Scenario

When two flights get the same Beacon Code assigned by a center initially (because they are offset in time and not in conflict in the center), but later on happen to be active in a center downstream at the same time, then the flight that enters the downstream center later gets reassigned a new code. This type of reassignment is called the “overtaking” scenario.

For example, the two flights shown in Figure 13 are starting from ZLA (Los Angeles Center). Flight A (shown in red) is headed from San Diego to Dallas Fort Worth (DFW) and flight B (shown in cyan) from Las Vegas to DFW. Both these flights happen to be assigned the same code 7201 by their origin centers; because they were not in initial conflict in departing center ZLA (B departed ZLA before A became active). When A enters ZAB (Albuquerque center) at 09:45 Hours, it has to be reassigned a new code by the center because code 7201 is already being used by B.

2.4. Research on Alternate Beacon Codes

Alternate Beacon Code allocation methods have been proposed in the past. The most notable of them are described as follows:



Figure 13: Overtaking Scenario for Code Reassignment

2.3.1 Code Assignment by Airline (9 Airlines interaction based allocation)

This method allocated blocks of codes to each airline which it in turn assigned to its own flight (Elbourn and Saunders, 1972, pp. 29-33). Airlines “whose routes do not cross or overlap” may be allocated the same codes. Non air carrier flights were assigned codes by the FAA independently. The definition of “whose routes do not cross or overlap” means flights whose routes do not enter the same ARTCC area. This research was conducted in 1971 and at that time 9 airlines were chosen. The conclusion of this research was that the airline routes were not independent of each other and therefore the scope of duplicating codes among airlines was not feasible.

2.3.2 Altitude Strata Code Assignment Plan (Codes assigned by Altitude, Reserved codes for Climb/Descent)

This method assigned codes to flights based on the flights being within certain altitude layers (Elbourn and Saunders, 1972, pp. 33-35). In addition, certain codes were reserved for climb and descent indications. The rationale was to partition the altitude into layers and the code-banks into proportional partitions for each altitude layer. The assignment plan was not considered feasible because the degree of coordination required to follow this kind of code assignment rules far outweighed the profitability of the plan.

2.3.3 Directional Code Assignment Plan (Same codes shared by flights operating in geographically independent regions)

This method proposed that flights which do not share a common center could use the same code (Elbourn and Saunders, 1972, pp. 35-36). As a result, the north/south flights on the west coast, mid-west and east could use the same codes since the flights would never run together. Based on this rationale, the same codes could be shared by flights operating in geographically independent region. To test this theory, the country was divided into geographically independent partitions so that they could all share the same codes without any interference. The conclusion of this study was that, “one can study this plan further, but the results will have a specialized applicability and the benefits will be limited.”

2.3.4 Master Assignment Plan (Flight Plan aware assignment of BC)

According to this method, codes are assigned to flights by one master control for all IFR (Instrument Flight Rules) flights in US (Elbourn and Saunders, 1972, pp. 23-28). The master center uses flight-plans to assign de-conflicted codes for each flight on a FCFS basis. This method used FCFS (First Come First Serve) rule to allocate de-conflicted codes to flights and no optimization was used. In the conclusion of this study, the authors stated that 465 codes were sufficient to allocate to 27,692 flights for peak day's IFR traffic (1970) without any code reassignment.

2.3.5 Geographic Beacon Code Allocation

This study focused on the optimization of Beacon Code allocations to reduce the number of code reassignments based on a new geographic scheme (Lucic, 2005). This method of geographic Beacon Code scheme addressed the “competing centers” (Figure 13) scenario as a major source of Beacon Code reassignments. However, this method made the problems caused by the “overtaking” scenario (Figure 13) worse since many flights flying approximately the same path were forced to share a small number of codes.

The code allocation was developed based on 17 days of ETMS data ranging from the year 2000 to 2004 as shown in Table 4. The data was initially used to estimate code demand and to determine the interference between center-regions. A destination region in this case consists of either a single center or a union of several centers. Since

the code allocation to center-regions consists of primary and secondary blocks of codes, two optimization problems were defined. The primary code allocation is a set of codes to be assigned to the traffic with the highest priority; it was determined for all center-regions first. The center-regions' primary code allocation optimization is aimed to allocate the available codes proportionally to center-regions' code demands while allowing small or no interference between center-regions sharing the code allocation. Since each center-region needed a specific number of codes to support the traffic, the difference between the required number of codes and size of primary allocation was allocated in the secondary block of codes in a way that minimized code sharing between center-regions with high interference.

The proposed allocation was tested using the Beacon Code allocation simulation. A total of 31 days of ETMS data were included in the simulation testing. The test results showed that the proposed allocation reduced the total number of reassignments by approximately 60% with standard deviation of approximately 2%. The simulation results also revealed that approximately 35% of the reassignments obtained by the proposed center-region allocation are the result of the "overtake" problem.

2.5. Objectives of this Research

The objectives of the problems being addressed in this dissertation can be summarized in the following three research questions:

RQ¹: Is there a Beacon Code reassignment problem in the currently used Beacon Code allocation system? Answering this research question formally demonstrates the existence of Beacon Code reassignment problem and establishes the primary motivation for this research. An analysis of archived data for a historically high traffic volume period (2007) is used to for identifying and quantifying the problem of reassignment in the current system (Chapter 3).

RQ²: Is there a centralized Beacon Code assignment solution that eliminates the need for reassignments? To answer this research question, alternate methods for code reassignment were developed, formulated, coded and evaluated. The results demonstrate that centralized Beacon Code assignment method exists that eliminates the need for code reassignment (Chapter 4, 5 and 6).

RQ³: Is there a centralized Beacon Code assignment solution that scales up to future traffic growth (X1.5 traffic)? The methods proposed in this research are tested using future projection of traffic to ascertain their scalability with growth in future traffic demand. It is shown that using the Space Time Adjacency (STA) algorithm developed in this research it is possible to assign codes for 1.5x traffic projections with only 1.29% likelihood of code reassignment (Chapter 5).

Analysis of historical data, and formulation and evaluation of the proposed Beacon Code assignment methods were used to answer these research questions.

Chapter 3: Data Sources and Statistics on Beacon Code Usage and Reassignments

The two primary functional problems that a Beacon Code assignment system must be capable to address are code reassignments and code shortages. This section describes the data sources used, and the analysis that was done in order to identify and quantify the code reassignment and shortage problems in the code allocation system being used currently in the National Airspace System (NAS). This system is called National Beacon Code Allocation Plan (NBCAP) and is owned and managed by Federal Aviation Administration (FAA), the Air Navigation Service Provider (ANSP) for the airspace of United States.

The goal of the analysis described in this chapter is to be able to answer the first research question Q^1 (See Section 2.5) which states: Is there a Beacon Code reassignment problem in the currently used Beacon Code allocation method? Answering this question establishes the need for this research, and also provides a baseline for comparison of the proposed Beacon Code assignment methods.

The data sources used in this research are described in the first subsection of this chapter. Next, statistics on Beacon Code usage and reassignment are presented. In the

next section of this chapter, Beacon Code demand in the NAS (National Airspace System) is discussed. The last section summarizes the results of data analysis in this chapter.

3.1 Data Sources

The three primary data sources used in this research are:

- (i) DOT/FAA Order JO 7110D
- (ii) HOST data
- (iii) Enhanced Traffic Management System (ETMS) 4-D trajectory data

Multiple data sources were used in this research for two main reasons. Firstly, there is a higher degree of confidence in the result when independent data sources are used to quantify the same metric (code reassignment) and their analysis yield similar results. Secondly, the resolution of ETMS (Enhanced Traffic Management System) data is higher than HOST data for any given day and as a result the analysis of large volumes of ETMS data is prohibitive in terms of computational space and time required. For this reason, 153 days of HOST data as opposed to 5 days of ETMS data are used for analysis.

3.1.1 DoT/FAA order 7110.66D

The process of allocation of codes to ARTCCs in the NAS is owned by FAA and published in DOT/FAA orders. The code allocation was last revised in November of 2009 and is published in (DOT/FAA, 2009). This order enlists the code blocks that are

allocated to each of the ARTCCs in the CONUS. The number of external and internal codes allocated to each of the 20 ARTCCs in the CONUS as per the order is shown in Figure 4.

3.1.2 Host Data

Host Data is recorded by the Host Computer System (HCS) for each of the 20 ARTCCs in the CONUS. HCS is the key information processing system in FAA's enroute environment. It processes radar surveillance data, processes flight plans, links filed flight plans with actual aircraft flight tracks, provides alerts of projected aircraft separation violations (i.e. conflicts), and processes weather data. The HCS along with the other hardware components also has a direct access storage subsystem which archives flight records.

The two types of Host Data used for analysis in this dissertation are Utilization Beacon (UB) and Beacon Reassignment (BA) messages. Data was extracted for a period spanning 153 days from 1st August, 2007 to 31st December 2007.

3.1.2.1 Utilization Beacon Messages

A snapshot of the Utilization Beacon (UB) message is shown in Figure 14. The HCS of each ARTCC maintains an hourly count of the number of Beacon Codes of each type (primary, secondary and tertiary) in both code categories (external and internal). Each row of the UB message for a given ARTCC represents the peak hourly count of

Beacon Codes being used in the corresponding code categories for every hour of the day.

The relevant fields that were extracted are in column (vi) through (x) of Figure 14:

- i. Column (vi): Peak Number of Internal Primary and Secondary Codes and the total number of adapted codes.
- ii. Column (vii): Peak Number of Internal Tertiary Codes and the total number of adapted codes.
- iii. Column (viii): Peak Number of External Primary and Secondary Codes and the total number of adapted codes.
- iv. Column (ix): Peak Number of External Tertiary Codes and the total number of adapted codes.
- v. Column (x): Number of Code Reassignments since midnight.

(i)	(ii)	(iii)	(iv)	(v)	(vi)	(vii)	(viii)	(ix)	(x)
zab	2007-07-18	39592	0010014081	601	0049/0189	0000/0000	0081/0315	0007/0110	0224
zab	2007-07-18	43980	0100011272	3601	0049/0189	0000/0000	0081/0315	0007/0110	0234
zab	2007-07-18	48164	0150017942	6601	0049/0189	0000/0000	0081/0315	0007/0110	0242
zab	2007-07-18	52268	0240014546	9601	0049/0189	0000/0000	0081/0315	0007/0110	0257
zab	2007-07-18	56413	0330011023	12601	0049/0189	0000/0000	0081/0315	0008/0110	0266
zab	2007-07-18	60217	0420016473	15601	0049/0189	0000/0000	0081/0315	0008/0110	0269
zab	2007-07-18	62884	0510010100	18601	0049/0189	0000/0000	0081/0315	0008/0110	0271
zab	2007-07-18	64691	0600012415	21601	0049/0189	0000/0000	0081/0315	0008/0110	0000

Figure 14: Snapshot of Utilization Beacon (UB) Message from ARTCC HOST Data

3.1.2.2 Beacon Reassignment (BA) Messages

A snapshot of the Beacon Reassignment (BA) message is shown in Figure 15.

Each row corresponds to an instance of code reassignment.

The relevant fields that were extracted are in column (vi) through (xi) of Figure 15:

- i. Column (vi): Call sign of Flight 1: The flight identifier of flight which is already using the corresponding Beacon Code.
- ii. Column (vii): Call sign of Flight 2: The flight identifier of flight whose Beacon Code needs to be reassigned due to potential conflict with Beacon Code of flight 1.
- iii. Column (viii): Computer Identifier of Flight 1.
- iv. Column (ix): Computer Identifier of Flight 2.
- v. Column (x): Beacon code of Flight 1(In use).
- vi. Column (xi): Beacon code of Flight 2(reassigned BC).

(i)	(ii)	(iii)	(iv)	(v)	(vi)	(vii)	(viii)	(ix)	(x)	(xi)	(xii)	(xiii)	(xiv)	(xv)
zab	2007-07-18	39242	0006163454	376	SWA2301	SWA2721	882	695	7334	0727	SAN	SAN	BNA	PHX
zab	2007-07-18	39982	0014174729	857	AWE369	N453AM	453	812	1546	0753	PHX	XNA	SMF	AMA
zab	2007-07-18	40596	0021055711	1265	AWE92	USA1561	326	197	4172	0721	PHX	PIT	PDX	PHX
zab	2007-07-18	41488	0031407169	1900	N390JK	UPS305	275	422	1741	2636	GBD	MMCU	AMA	KELP
zab	2007-07-18	41954	0037107941	2230	COA427	N424QS	901	929	7222	2647	SAN	SNA	DFW	GGG
zab	2007-07-18	42479	0043228820	2602	AAL2452	AAL1324	178	154	7236	2646	LAX	SNA	DFW	DFW
zab	2007-07-18	42576	0044318991	2671	AWE399	AAL2458	572	175	7371	1603	IAH	LAX	PHX	DFW
zab	2007-07-18	43084	0050069821	3006	COA541	N425AP	041	686	7370	2657	LAX	GTU	IAH	TDW

Figure 15: Snapshot of Beacon Reassignment (BA) Message from ARTCC HOST Data

3.1.3 ETMS 4-D Trajectory Data

The Enhanced Traffic Management System (ETMS) is a system developed, owned and used by FAA to manage the flow of air traffic within the NAS on a daily basis. ETMS data helps provide traffic management specialists with guidance to maintain air traffic flow in the event of changing capacities in NAS due to weather adversities. ETMS data

for five days of 2007 (a year of historical high air traffic demand) were chosen. The five days chosen span over different seasons to account for the seasonal variation in traffic demand and route structure. The days chosen for analysis were:

- (i) 3rd Jan, 2007 (Winter)
- (ii) 11th April, 2007 (Spring)
- (iii) 26th July, 2007 (Summer)
- (iv) 21st November, 2007 (Day before Thanksgiving)
- (v) 19th December, 2007 (Winter)

An algorithm was developed to convert the ETMS 4-D trajectory data to “center-crossing” data. The conversion was done by superimposing the 4-D trajectories on the center geometries and finding the entry and exit point in time and space for each center on the route of a flight. The details of this algorithm are described in Appendix A: 4DT-to-Center-Route Converter.

3.2 Beacon Code Usage Statistics

Beacon code usage statistics were derived from analysis of 153 days of HOST data for all the 20 ARTCCs in the CONUS. The ETMS data could not be used to derive code usage statistics because it is a flight centric dataset that does not have HOST specific Beacon Code information.

The maximum code utilization for all the 20 ARTCCs for 153 days of HOST-data analyzed is shown in Figure 16. The results indicate that there is no Beacon Code shortage problem in either external or internal code categories in the current code assignment system as per the current (2007) traffic load. In the internal category ZHU (Houston center) has the highest code utilization of 0.529. The center with the highest code utilization in the external category is ZLC (Salt Lake City center) with 0.389 fraction of the codes allocated to it being used.

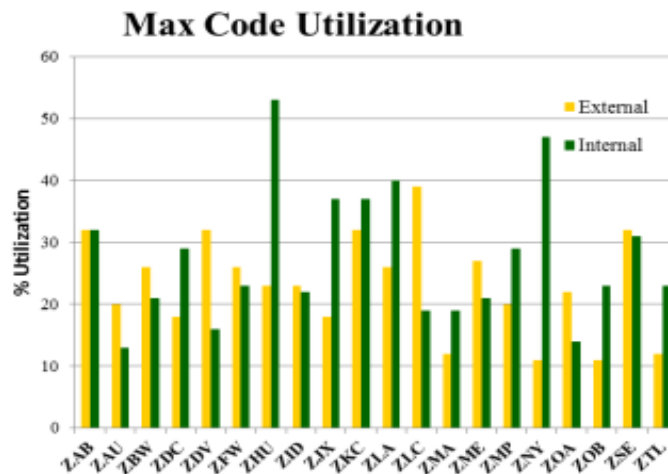


Figure 16: Maximum Code Utilization for each of the 20 ARTCCs (for 153 days of Host Data from 1st July 2007 to 31st December 2007)

A summary of the mean and median of maximum code utilization fraction per day for each of the 20 ARTCCs is shown in Table 1 and Table 2 for external and internal

Beacon Codes respectively. As the code utilization is well below 100% for all the 20 ARTCCs in the CONUS for the 153 days of 2007 data analyzed, the conclusion is that Beacon Code shortage is not a problem of the current code assignment system for current traffic load.

Table 1: Maxim Utilization Fraction for External Primary and Secondary Codes for 153 days of 2007 (Host Data)

ARTCC	Max of Max	Median	Mean
ZAB	0.321	0.289	0.289
ZAU	0.198	0.154	0.157
ZBW	0.265	0.195	0.19
ZDC	0.176	0.162	0.164
ZDV	0.317	0.263	0.26
ZFW	0.265	0.221	0.222
ZHU	0.227	0.197	0.197
ZID	0.227	0.198	0.2
ZIX	0.176	0.133	0.131
ZKC	0.317	0.282	0.283
ZLA	0.265	0.264	0.265
ZLC	0.389	0.295	0.286
ZMA	0.122	0.09	0.087
ZME	0.267	0.233	0.241
ZMP	0.204	0.167	0.164
ZNY	0.115	0.106	0.107
ZOA	0.222	0.184	0.183
ZOB	0.11	0.086	0.087
ZSE	0.317	0.253	0.25
ZTL	0.116	0.088	0.088

Table 2: Max Utilization Fraction for Internal Primary and Secondary Codes for 153 days of 2007 (Host Data)

ARTCC	Max of Max	Median	Mean
ZAB	0.317	0.219	0.233
ZAU	0.127	0.089	0.089
ZBW	0.21	0.113	0.105
ZDC	0.292	0.218	0.219
ZDV	0.163	0.117	0.119
ZFW	0.234	0.168	0.171
ZHU	0.529	0.499	0.529
ZID	0.217	0.134	0.132
ZJX	0.365	0.258	0.27
ZKC	0.37	0.262	0.286
ZLA	0.397	0.314	0.313
ZLC	0.19	0.143	0.151
ZMA	0.193	0.116	0.111
ZME	0.206	0.145	0.159
ZMP	0.286	0.225	0.238
ZNY	0.466	0.261	0.243
ZOA	0.137	0.099	0.102
ZOB	0.233	0.179	0.18
ZSE	0.313	0.237	0.246
ZTL	0.225	0.153	0.156

3.3 Beacon Code Reassignment Statistics

The number of Beacon Code reassignments was derived from both HOST Data and ETMS data. Two “independent” data sets were used to improve the confidence in the result.

An analysis of the HOST data for 153 days of 2007 from 1st August 2007 to 31st December 2007 yielded the average number of daily Beacon Code reassignment instances to be 7,642 with a standard deviation of 1,451 (Figure 7). The median number of reassignments is 7,769 with a maximum of 10,571 reassignments and a minimum of 3,143 for the 153 days of HOST data analyzed. The number of ARTCC boundary crossing instances (hereafter called hand-offs) for a typical day of NAS is 62,111 (obtained from analysis of ETMS data as shown in Table 3). The ratio of number of BC reassignment instances to the total number of hand-offs generates the likelihood of a flight getting a code reassigned when it crosses an ARTCC boundary. For the HOST data analyzed the average likelihood of code reassignment is 12.3% (7,642/62,111).

Table 3: Number of Hand-Offs and Reassignments (Source: ETMS Data)

Date(2007)	Hand-Offs (x)	BC-Reassignments (y)	BC Reassignment Likelihood (y/x*100)
3-Jan	59,797	5,491	9.2 %
11-Apr	58,529	5,645	9.6 %
26-Jul	62,805	6,730	10.7 %
21-Nov	65,076	6,779	10.4 %
19-Dec	64,348	6,394	9.9 %
Mean	62,111	6,208	9.96 %
Standard Deviation	2,849	605	0.6 %

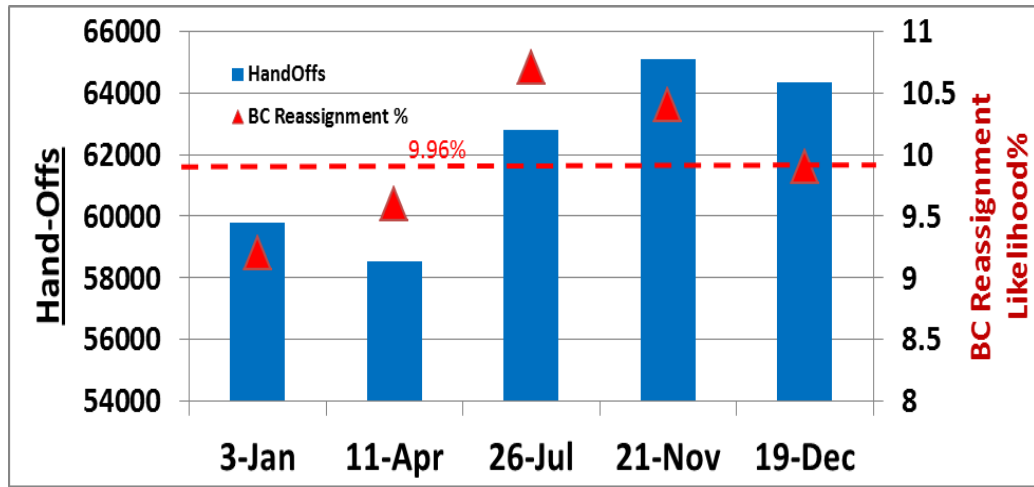


Figure 17: Beacon Code Hand-Off and Reassignment Likelihood from ETMS Data

The total number of BC reassignments was also derived from the ETMS 4-D trajectory data for all the 5 days shown in Table 3. The results are also graphically shown in Figure 17. For the 5 days of ETMS 4-D trajectory data analyzed, the likelihood of code reassignment is in the range of 9.2%-10.7% with an average of 9.96% and a standard deviation of 0.6%.

A previous study of Beacon Code reassignments by Lucic et al (Lucic, 2005) showed that the mean number of code reassignments for a period of 17 days is 8,809 (Table 4). The minimum and maximum number of Beacon Code reassignments reported was 7,014 and 9,865 respectively.

Table 4: Number of Beacon Code Reassignments for 17 days (Lucic, 2005)

Srl No.	Date	Competing Center	Overtaking	Total # of Reassignments (nCCr+nOVr)
		nCCr	nOVr	
1	11/22/2000	7392	989	8381
2	12/20/2000	6876	773	7649
3	1/24/2001	6222	792	7014
4	4/19/2001	7031	912	7943
5	5/17/2001	7059	891	7950
6	11/26/2003	8077	1256	9333
7	12/18/2003	7999	973	8972
8	1/23/2004	6638	874	7512
9	2/13/2004	8656	1209	9865
10	3/12/2004	8416	1144	9560
11	4/29/2004	8506	1220	9726
12	5/27/2004	8469	1143	9612
13	7/1/2004	8308	1164	9472
14	8/6/2004	7995	1209	9204
15	9/10/2004	7951	1090	9041
16	10/20/2004	7887	1001	8888
17	11/19/2004	8383	1245	9628
		Mean % : 88.1		Mean : 8809

The analysis of these two independent datasets yielded similar results and the range of BC reassignment likelihood was established to be 9.2% to 12.3%. This means that in the current system as per 2007 traffic load, there is one in ten chances of a flight getting its Beacon Code reassigned whenever it crosses an ARTCC boundary enroute to its destination. This range serves as the baseline for comparing any new proposed Beacon Code assignment method(s).

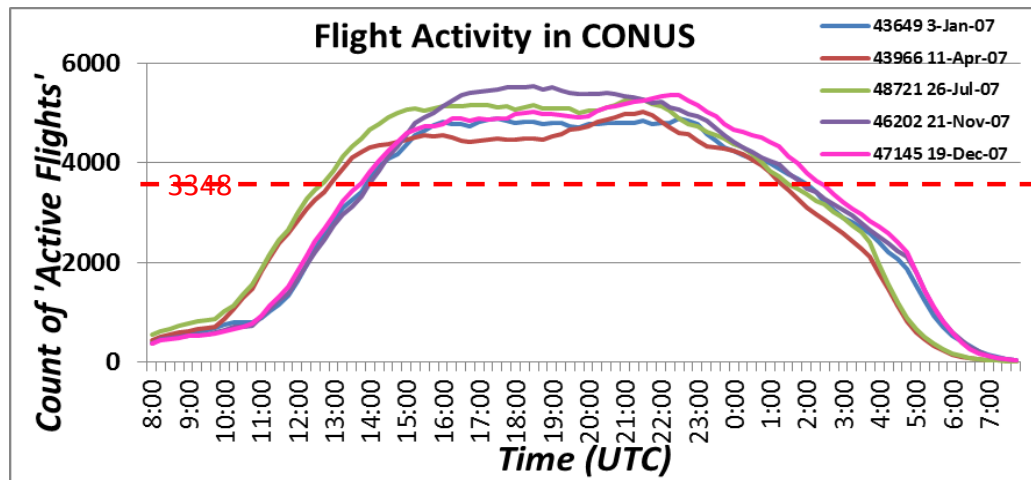


Figure 18: Count of "Active Aircraft" for each quarter-hour of the day (Source: ETMS 4-D Trajectory Data)

3.4 Beacon Code Demand

Every flight active in the NAS requires a Beacon Code so that it can be uniquely identified by ATC. As a result, the count of active flights is an indicator of the demand for Beacon Codes. The total number of flights active in each 15 minute intervals for all the five days of ETMS 4-D trajectory data analyzed is shown in Figure 18. The dotted line in the figure shows the number of codes available, i.e. 3,348. Out of all the 5 days analyzed, the maximum number of active flights observed in the NAS was 5,802 (21st Nov). The dotted line in Figure 18 shows that the number of active flights exceeds the number of codes available, i.e. 3,348. If the traffic level was lower than the number of codes available then it would be (at least theoretically in a deterministic environment)

be possible to allocate codes centrally to all flights in NAS on a First Come First Serve (FCFS) basis without any need for more complex algorithms.

3.4.1 Beacon Code Life Cycle

The imbalance between the number of Beacon Codes required and the number of codes available is further aggravated by the fact that the lifecycle of a Beacon Code is longer than the duration of a flight. A flight is assigned a Beacon Code DSPI minutes (Departure Strip Printing Interval) before its scheduled departure time. The mode of DSPI for the 20 centers in the CONUS is 30 minutes (Figure 19)(FAA, 2007).

Also, when flights cross ARTCC boundaries enroute to their destination, the Beacon Codes are not immediately available for reuse in the exiting ARTCCs. A buffer of CRDT (Code Reassignment Delay Time) minutes has to elapse before the code can be assigned to any other flight in the exiting ARTCC. The value of CRDT is typically 30 minutes (FAA, 2007).

Also, Beacon Codes are not immediately available for reuse after they are released by flights upon arrival. A buffer of SRDT (Secondary Reassignment Delay Time) minutes has to elapse before the Host Computer System (HCS) of the arrival center of the flight gets a TB (Termination Beacon) message. The duration of SRDT is typically 30 minutes.

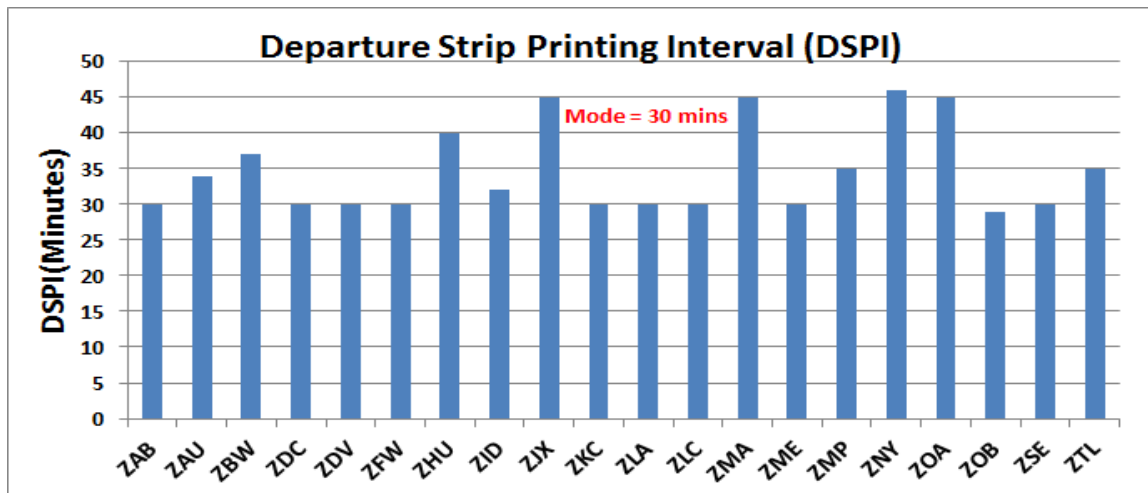


Figure 19: Departure Strip Printing Interval (DSPI) for 20 ARTCCs in the CONUS (Source: (FAA, 2007))

These buffers at the start and end of flight trajectories place additional burden on the Beacon Code allotment system. If these *start* and *end* buffers are added to each flight's duration, then the Beacon Code demand for each 15 minute interval is obtained. The demand for beacon-codes for each quarter-hour of the day is shown in Figure 20. Out of the 5 days analyzed, the highest *Beacon Code demand* observed in the NAS was 8,121 (21st Nov, 2007).

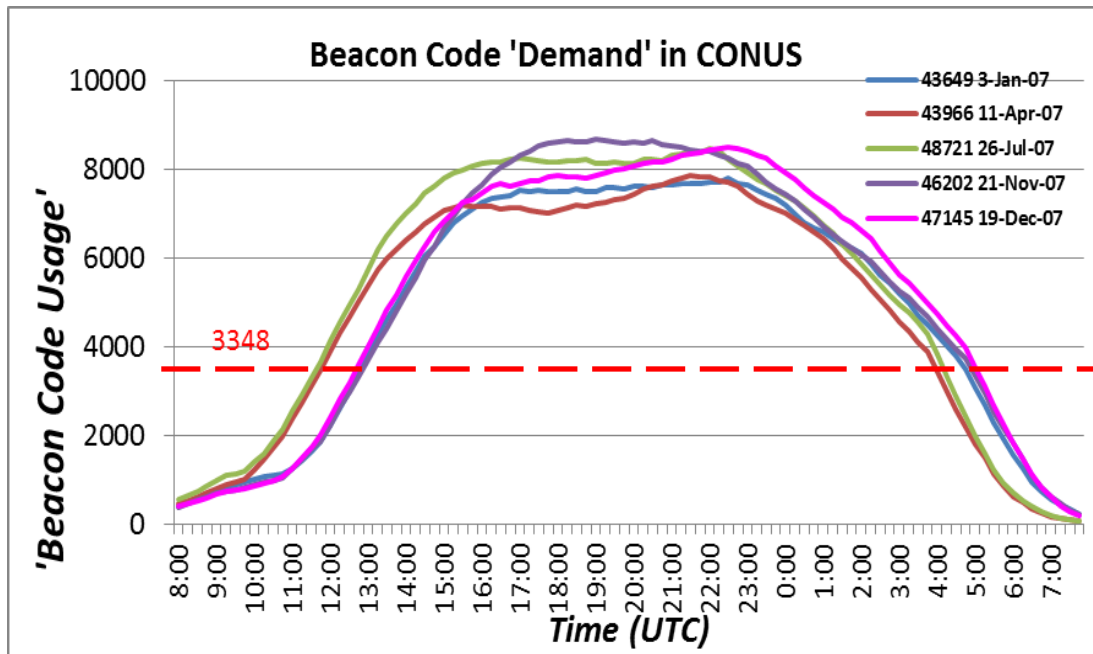


Figure 20: Count of "Beacon Code Demand" for each quarter-hour of the day (Source: ETMS 4-D Trajectory Data)

The demand for Beacon Codes by individual centers is shown in Figure 21. The data shown is sorted in descending order of code demand. It can be observed that ZDC (Washington DC) center has the highest demand on all 5 days being analyzed and is in the range of 499 to 815 with an average of 704.

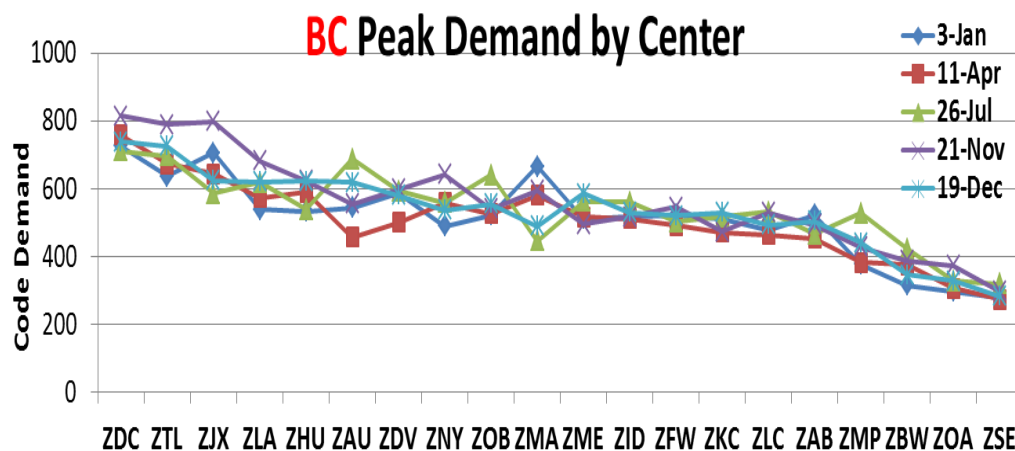


Figure 21: Beacon Code Peak Demand by Individual Centers (Source: ETMS Data).

3.5 Summary

The following is the summary of Beacon Code utilization and reassignment statistics obtained from analysis of HOST data and ETMS data:

Code Utilization (HOST Data): In the internal category ZHU (Houston center) has the highest code utilization of 0.529. The center with the highest code utilization in the external category is ZLC (Salt Lake City center) with 0.389 fraction of the codes allocated to it being used.

Code Reassignment (HOST Data): An analysis of the HOST data for 153 days of 2007 from 1st August 2007 to 31st December 2007 yielded the average number of daily Beacon Code reassignment instances to be 7,642 with a standard deviation of 1,451

(Figure 7). This is equivalent to an average code reassignment likelihood of 12.3% (7,642/62,111).

Code Reassignment (ETMS Data): The likelihood of code reassignment as derived from 5 days of ETMS 4-D trajectory data is in the range of 9.2% to 10.7% with an average of 9.96% and a standard deviation of 0.6% (Table 3).

The analysis of ETMS data and HOST data establishes the fact that there is a 9.2% to 12.3% Beacon Code reassignment problem in the current system. The result of this analysis answers the first research question Q¹ in the affirmative, i.e. there is a reassignment problem in the NAS currently used Beacon Code assignment method. This answer establishes the need for this research, and also provides a baseline for comparison of the proposed Beacon Code assignment methods.

The next three chapters describe alternate methods for code assignment that were developed during this research. The individual flight code assignment optimization model is described in chapter 4. In chapter 5, a heuristic algorithm called STA (Space-Time Adjacency) is described. Chapter 6 describes a hybrid code assignment method based on clusterization and STA heuristic algorithm.

Chapter 4: Individual Flight Code Assignment Optimization Model

For a Universal Beacon Code assignment method to be feasible, the number of codes required for allocation to flights must be less than 3,348, the total number of codes available. This section describes a Mixed-Integer Linear Program (MILP) optimization model that assigns Beacon Codes to flights using the spatial and temporal opportunities of the flights route and schedule. The objective of the optimization model is to assign Beacon Codes to an ensemble of flights using as few codes as possible. No code reassignments are permitted. By comparing the output of this model with the total number of codes available (3,348), the feasibility of Universal Beacon Code assignment to all flights in NAS is ascertained.

The goal of this chapter is to be able to answer the second (Q^2) and third (Q^3) research questions posed in Section 2.5.

This chapter is organized as follows. The first subsection of this chapter defines the indices of the MILP optimization model and also describes data preprocessing steps. The next subsection describes the MILP model including the objective function and the constraints. The results are described in the next subsection. The last section summarizes the conclusions.

4.1 Definition of Indices and Preparing Data for Optimization

4.1.1 Time Horizon Definition

- 'T' is the total duration of planning horizon, i.e. time difference between the start time of the *earliest* flight and the end-time of the *last* flight in the population.
- 't' is the duration of the individual time-steps into which the planning horizon T is divided. The lower the value of 't', the higher is the resolution of the optimization model. However, small values of 't' result in large number of variables. The default value of 't' is set to '5' minutes.
- The ratio of 'T' and 't' is the total number of time-steps, τ . For example, if 'T' is 1 hour and 't' is 5 minutes, then ' τ ' is 12(60/5). In general, $\tau = \lceil T / t \rceil$.

4.1.2 Indices

i is flight index. $i = \{1, 2, \dots, N\}$, where 'N' is the total number of flights in the planning horizon

c is the Beacon Code index. $c = \{1, 2, \dots, 3348\}$

k is the ARTCC index. $k = \{1, 2, \dots, 20\}$

m is in the index for time-steps. $m = \{1, 2, \dots, \tau\}$

4.1.3 Preprocessing data

For every flight that needs to be assigned a Beacon Code, this model requires as input, its scheduled departure time and the flight path in terms of the center(s) that the

flight is predicted to traverse along with the *host-prediction* of center boundary crossing times (if any).

In order to reduce the number of variables in the model, the flight data is converted to a center-occupancy data before being input to the model. More specifically, the flight path and the *host-prediction* of center crossing times is converted to a center-centric data, whereby, for each of the 20 centers, a list of flights predicted to be in that center for each time-step is prepared. As the result, the two data variables described below is obtained:

$\beta_{k,m}$ is the number of flights that are present during time-step 'm' in center 'k'. ($k = 1..20$ and $m=1.. \tau$).

$\alpha_{k,m,j}$ is the index of j^{th} (of $\beta_{k,m}$) flight that is in center 'k' during time-step 'm'.

4.2 Description of Optimization Model

The optimization Model is shown in Figure 22.

$$\begin{aligned}
& \text{Min } Z = \sum_c (c * y_c) \\
& \text{s.t. :} \\
& (1) \quad \sum_{j=1}^{\beta_{k,m}} x_{\alpha_{k,m,j},c} = 1, (\forall k, m, c) \\
& (2) \quad \sum_c x_{i,c} = 1, \forall i \\
& (3) \quad x_{i,c} \leq y_c, \forall i, c
\end{aligned}$$

Figure 22: MILP Model for Individual Flight BC Assignment

4.2.1 Decision variables

$x_{i,c}$ is a binary decision variable which is equal to 1, if flight 'i' is assigned Beacon Code 'c'. y_c is a dependent binary decision variable which is equal to 1, if code 'c' is assigned to any flight.

4.2.2 Objective Function

The objective function Z is a minimization of sum of ' y_c ' weighted with the corresponding code 'c'. The goal is to minimize the total number of codes needed to allocate to the flights being input to the model.

The sum of y_c 's is weighted with the corresponding code 'c' to ensure that the codes are assigned in ascending order. From an operational perspective, the order of the actual codes being assigned to a flight does not matter, i.e. there is no significance of assigning codes in ascending or descending order. However, the symmetry in problem

structure has detrimental effect on the computational time of the model. This happens because the optimization routine tries to explore all the possible combinations of the state space without being able to differentiate one code from another. By introducing weights in the objective function, codes are assigned in ascending order thereby improving computational time. As a result of codes being assigned in ascending order, the highest value of 'c' for which y_c is 1 is the maximum number of codes required for a given set of input flights.

Equation 1: Objective Function of Individual Flight Code Assignment Model

$$\text{Min } Z = \sum_c (c * y_c)$$

4.2.3 Constraints

Constraint 1 (Equation 2) ensures that all the flights that are active in a given center in the same time-period get unique codes assigned to them. The number of type 1 constraints are $K * \tau * C$. For a time horizon of 40 minutes with 5 minute time periods, τ is 8. C is the total number of codes available, i.e. 3,348. K (number of centers) is 20. In such a typical case the total number of type 1 constraints is $8 * 3348 * 20 = 535,680$.

Equation 2: Unique codes constraint

$$(1) \quad \sum_{j=1}^{\beta_{k,m}} x_{\alpha_{k,m,j},c} = 1, (\forall k, m, c)$$

Constraint 2 (Equation 3) ensures that every flight gets assigned exactly one code for all centers on its path. The number of type 2 constraints is 'n', i.e. total number of flights in the population.

Equation 3: Assign codes constraint

$$(2) \quad \sum_c x_{i,c} = 1, \quad \forall i$$

Constraint 3 (Equation 4) ensures that y_c is 1 if code 'c' is used by at least one aircraft. The number of type 3 constraints are $3348 \cdot n$.

Equation 4: Code usage constraint

$$(3) \quad x_{i,c} \leq y_c, \quad \forall i, c$$

4.2.4 Additional Constraints

If this model is implemented in rolling time-window horizons, then for flights which overlap from one time-window into next, the codes assigned to such flights are fixed for the following time-windows. (See 5.2.1.3 Planning-Window for discussion on Time-Windowing). For example, say F is the set of flights which are overlapping from the current time window to the next, and each flight 'i' in F gets a code c_i assigned to it through the optimization model. Then the initial $x_{i,c=c(i)}$ is set to 1 for all the flights in F for the next time window.

4.3 Results

Due to the large number of constraints (See 4.2.3 Constraints) with increasing N and C, this optimization model does not scale well. It was tested for a set of 683 flights which were the total number of flights that were “active” in the CONUS on 3rd April 2007, 1100 to 1120 UTC.

A total of 118 codes are needed to allocate to these 683 flights as per the optimization model. The code assignment for individual flights is shown in Figure 23. The x-axis represents the flight index and the y-axis is the Beacon Code index. The plot in Figure 24 shows the code assignment frequency for each of the 118 codes being used. For example, code 80 was allocated to 11 of the 683 flights in the population (circled in red in Figure 24).

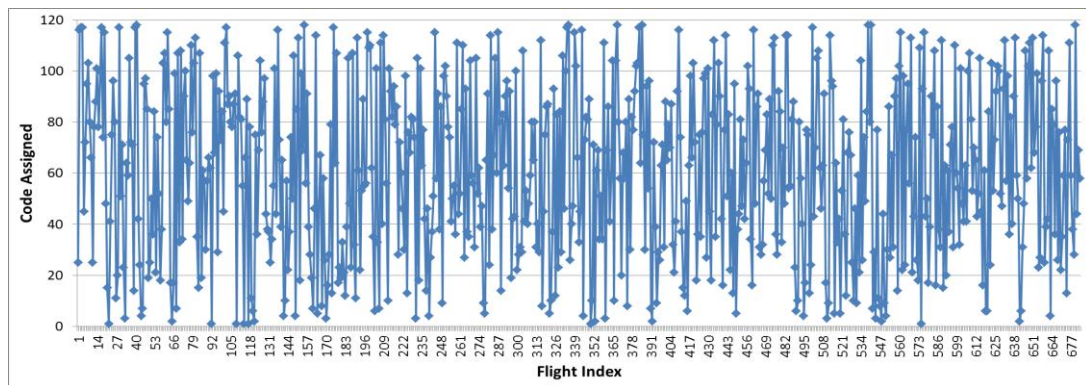


Figure 23: Instantaneous traffic count by ARTCC for the experimental dataset

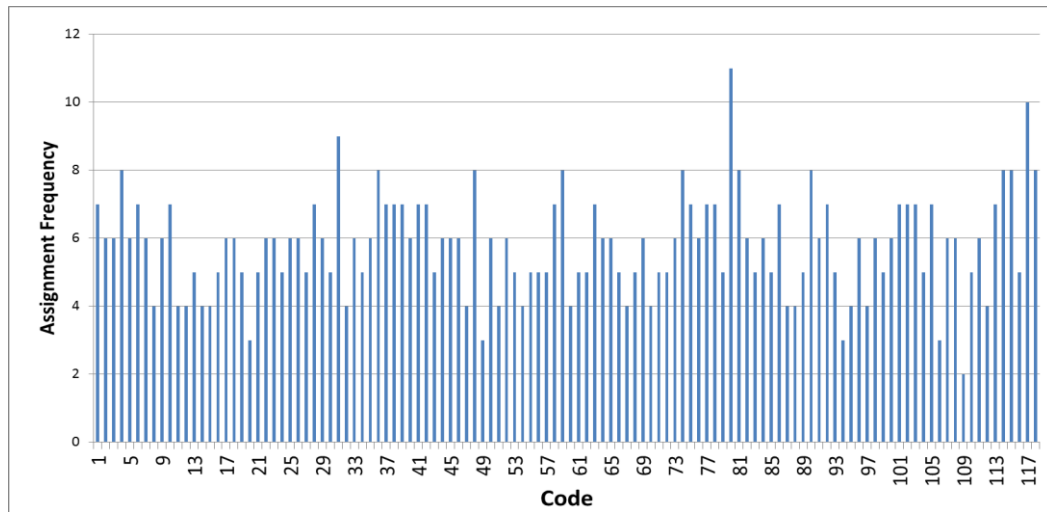


Figure 24: Assignment Frequency Chart of Codes allocated by the Optimization Model

4.4 Conclusion

The computational time of this problem depends on the structure of the input data (i.e. routes and schedules) and also on the duration of planning horizon. The Mixed Integer Linear Programming (MILP) model described above does not scale well when the total number of flights exceed 700.

During peak traffic in NAS, the number of flights is over 5,000 (See Figure 18). This optimization model is not feasible to solve the code assignment problem in practice. If the optimization model was able to scale up to the real air-traffic volume, then the output of this model would also provide us with the theoretical optimum number of codes needed for assignment to a given set of flights. This problem of code

assignment is analogous to the graph coloring problem where the nodes represent flights and the colors represent Beacon Codes. However, the graph coloring problem is known to be NP-complete. So it is not surprising that the model does not scale well with increasing number of flights and the computational time grows exponentially. Due to this scalability issue, this optimization model was adapted into a faster and more scalable algorithm called Space-Time Adjacency (STA) algorithm which is described in the next chapter.

Chapter 5: Space-Time Adjacency (STA) Algorithms

As discussed in the previous chapter, the optimization model for individual flight code assignment fails to scale up to the real air-traffic volumes. Due to this scalability issue, the Mixed Integer Linear Program (MILP) optimization model was adapted into a faster and more scalable heuristic algorithm called Space-Time Adjacency (STA) algorithm. The primary concept of STA algorithm is to assign Beacon Codes to flights by exploiting the spatial and temporal opportunities of the flight schedules and routes.

This chapter describes the STA algorithm along with its data structures, algorithmic details and results. Two versions of the STA algorithm are discussed. The first method is the basic version of STA algorithm in which code reassignments are not permitted. In the second version of the algorithm, which is referred to as STA-R (STA with reassignment), Beacon Code reassignments are permitted. The second method is required in scenarios when the total 3,348 codes available are not sufficient.

The goal of this chapter is to be able to answer the second (Q^2) and third (Q^3) research questions posed in Section 2.5. In other words, is there a Universal Beacon Code assignment method that can be used to assign codes to flights centrally

throughout NAS. Furthermore, if such a Universal Beacon Code assignment method exists, is it scalable to future growth in air-traffic (1.5x traffic).

This chapter is organized as follows. An overview of the STA algorithm is discussed in the first subsection of this chapter. Next, a detailed discussion of the STA algorithm is provided. The results of code assignment for historical days using STA algorithm is presented next. The following subsection describes the STA-R algorithm in detail along with results of code assignment for future traffic projections (1.5x traffic). The last subsection summarizes the conclusions of assigning codes using STA and STA-R heuristic algorithms.

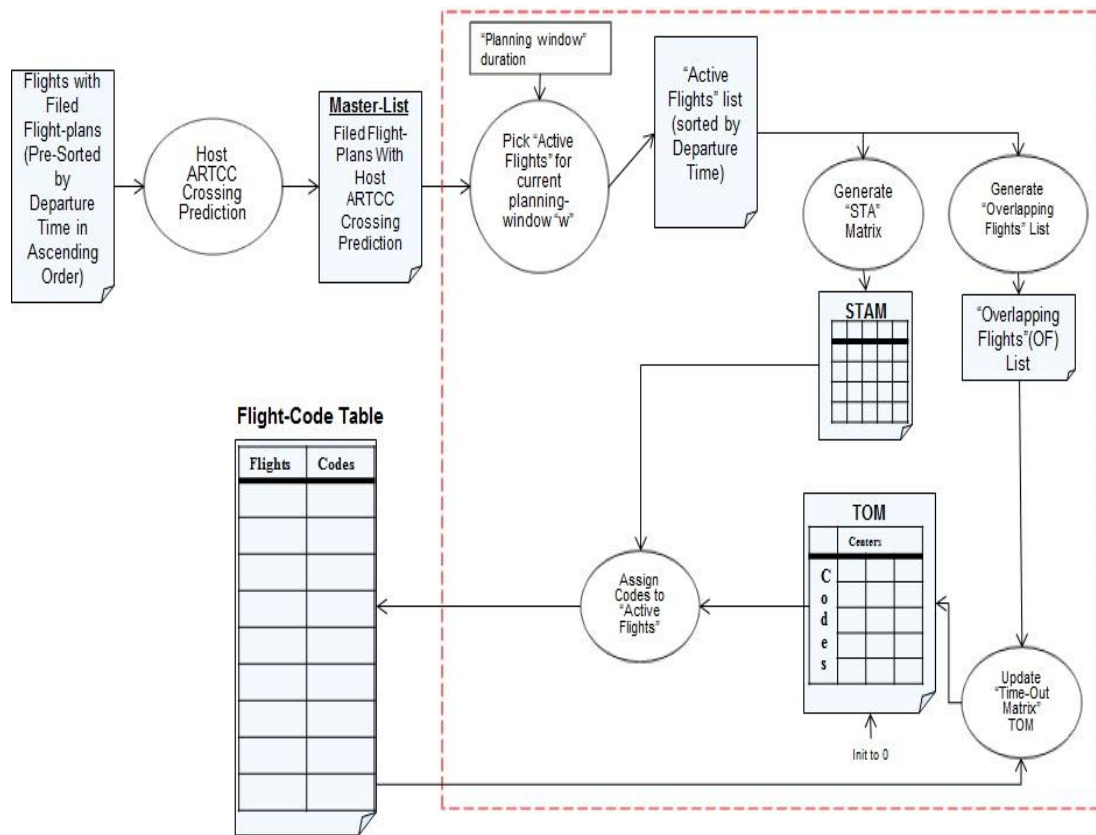


Figure 25: Block diagram of STA Algorithm

5.1 Algorithm Overview

The STA algorithm is described in Figure 25. Firstly, flights with filed flight-plans are ordered by departure time in ascending order. The ARTCC crossing times for each flight are then generated and added to the flight list. This ordered list of flights along with their predicted ARTCC crossing times is called the “Master List”.

All the flights that are *active*¹ in the current planning-window are then removed from the “Master List” and exported into a new list called “active flights” list for the current planning-window. A flight from the “Master List” is classified as *active* in a given planning-window if its schedule departure is either *before* or *no later than* DSPi (Departure Strip Printing Interval, 30) minutes after the end of the window.

The “active flights” list and their predicted ARTCC crossing times are then used to generate the Space-Time Adjacency (STA) Matrix. This matrix identifies flights that are predicted to be in the same ARTCC at the same time.

Next, the list of overlapping flights is generated. This list consists of flights that are predicted to be *active* beyond the end of the current planning window.

Based on the STA matrix and the codes timed out by overlapping flights of the previous planning-window, all the flights in the current planning window are assigned Beacon Codes.

Next, a code Time-Out Matrix (TOM) is generated for the following planning-window using the codes assigned to overlapping flights of the current window.

This process is repeated until all the flights have been assigned Beacon Codes.

¹ Flights which require Beacon Codes in a given planning-window are considered “active” in that planning-window.

5.2 STA Algorithm

The goal of STA algorithm is to be able to assign codes to all the flights in the NAS using less than 3,348 codes such that there are *no reassignment instances*. In other words, every flight is assigned a single Beacon Code for its entire flight duration.

5.2.1 Data Structures and Parameters

The following data-structures are used in the STA algorithm:

1. Space-Time Adjacency(STA) Matrix, and
2. Code Time-Out Matrix(TOM),

The STA algorithm has the following parameters:

1. Planning-window duration, and
2. Uncertainty buffer in Host-Prediction,

5.2.1.1 STA Matrix

The Space-Time Adjacency (STA) matrix is a binary matrix which is referenced for every flight-pair. If an element of STA is 1, it signifies that the flight-pair corresponding to that particular position are predicted to be in the same ARTCC at the same time for at least one instance on their trajectories. This implies that the corresponding flight-pair must be assigned different Beacon Codes. A flight-pair for which the corresponding value in the STA matrix is 0, may be assigned the same Beacon Code as they are (predicted to be) not in conflict at any point on their trajectories.

A two dimensional Space-Time Adjacency (STA) matrix of 'n' flights would require n^2 elements. The non-linear increase in memory requirement with respect to the number of flights in the planning horizon necessitates more efficient memory utilization.

Original STAM				Linearized STAM	
1	0	0	1	0	
0	1	0	1	0	
0	0	1	0	0	
1	1	0	1	1	
				1	
16 elements				0	
				6 elements	

Figure 26: Linearizing STA Matrix to Reduce Memory Requirement (Example of 4 flight case)

By definition, the space-time adjacency of a pair of flights is symmetric. If flight i is space-time adjacent to flight j, then flight j is also adjacent to flight i. As a result of this symmetry, instead of storing all the n^2 elements, only the elements below the diagonal ($i > j$) are sufficient to represent all the Space-Time Adjacency information contained in the square matrix (Figure 26). The number of elements below the diagonal in a square matrix of size n is $\frac{n*(n-1)}{2}$. This leads to savings in memory requirement in excess of 50%. Note: The diagonal elements need not be stored as a flight is always space-time adjacent to itself.

Elements stored in linearized one-dimensional matrix (row-major order).

Element $[i,j]$ in the original square matrix corresponds to $((i-1)*(i-2))/2 + j$ th element in the linearized matrix. For example, (4,2) corresponds to $(4-1)*(4-2)/2 + 2 = 5^{\text{th}}$ element.

5.2.1.2 Overlapping Flights (OF) List and Code Time-Out Matrix (TOM)

The “Overlapping Flights” (OF) list for a given planning-window is the list of flight indices of flights that are “active” (need Beacon Code) beyond the end of the planning-window.

The code “Time-Out Matrix” (TOM) is a two dimensional matrix of 3,348*20 elements. The rows and columns correspond to “Beacon Codes” and “centers” respectively. An element $[i,j]$ of TOM represents the time *until* which code ‘i’ is timed-out in center ‘j’, i.e. it can’t be assigned to any other flight in center ‘j’. For the TOM example shown in Table 5, code 3 is time-out for center 1(ZAB) until 7:00 AM. However, code 2 is available for assignment in center 1 because element $[2,1]$ is 0.

The code “Time-Out Matrix” (TOM) for the first planning-window is initialized to 0, i.e. all the elements of TOM at the start of the algorithm is set to 0.

At the *end* of the current planning-window, the codes assigned (output of current run of STA) to each of the “overlapping flights” is *timed-out* in the centers that these flights are predicted to traverse after the end of current planning-window.

At the *start* of the following planning-window, all the values in TOM that are less than the start-time of the planning-window are reset to zero. By doing so, all the codes whose time-out epoch expires before the start-time of the planning-window are made available for use.

Table 5: Time-Out Matrix (TOM) (3348 X 20 elements)

Center	1	2	3	4	5	.	.	20
Code Index	ZAB	ZAU	ZBW	ZDC	ZDV	.	.	ZTL
1	6:00 AM	0	0	0
2	0	7:00 AM	0	0
3	7:00 AM	0	0	0
4	0	7:45 AM	8:00 AM	0
5	0	0	0	0
6	8:00 AM	0	0	10:20AM
.
.
3348

5.2.1.3 Planning-Window Duration

The STA algorithm is implemented for a finite duration of time known as the planning-window. Ideally, all the flights for the entire day would be allocated codes in

one “run” of the STA algorithm. In that case the duration of planning-window, denoted by ‘T’ would be 24 hours. However, setting the ‘T’ to 24 hours is impractical due to the following two reasons.

(1) *Storage*: The number of “active flights” in each planning-window increases with ‘T’.

If the value of ‘T’ is 24 hours, then on a typical day in NAS (26th July 2007) with 48,721 flights (See 3.1.3 ETMS 4-D Trajectory Data), the number of elements required to be stored in STA matrix is 1,186,843,560 (Based on $n*(n-1)/2$). Due to the non-linear increase in memory requirement with respect to the number of flights, the planning horizon needs to be curtailed.

(2) *Weather prediction*: Also, the creation of STA matrix requires “host prediction” (See 5.2.1.4 Host Prediction Uncertainty Buffer) data for boundary crossing times of all the flights in the planning horizon. The prediction of boundary crossing times of the flights depends on accurate forecast of weather and the resulting capacity of the constrained resources in NAS. Due to randomness in weather and the resulting inaccuracy of weather forecasts, creating a plan for routes for the entire day is not realistic (Michalek, D., Balakrishnan, H., 2004). The state-of-the art in convective weather forecast is MIT Lincoln Laboratory’s Convective Weather Forecast product (Wolfson, M., et al., 2004) which provides accurate prediction of weather in the 0-2 hour range.

Due to the combination of the two factors mentioned above, the typical duration of a planning-window is set to 60 minutes. As a result, the day is divided into 24 non-

overlapping and sequential time-windows, each of 60 minutes duration. Each planning-window 'w' has a start and end time represented by α_w and Ω_w . As the windows are non-overlapping but continuous, $\Omega_w = \alpha_{w+1}$, i.e. start-time of the following window is equal to the end-time of the current window.

5.2.1.4 Host Prediction Uncertainty Buffer

The construction of Space-Time Adjacency (STA) matrix is based on the *host-prediction* of ARTCC crossing times of the flights from their filed flight-plans.

There may be uncertainty in the host-prediction of these boundary crossing times due to delays and/or changes in flight routes. This uncertainty can be represented by blocking a time-window of buffer minutes (+ and -) around each of the *predicted* boundary crossing instance. Host-prediction uncertainty buffer times were set to 0, 15, 20, 25 and 30 minutes for this analysis. A *host-prediction* uncertainty value of 0 minutes represents perfect information about the center crossing time of each flight in the system. This is ideal from the standpoint of code-assignment but impractical because of the stochastic factors that influence flight routes and time in the NAS.

For the example shown in Figure 27, flight AAL123 traveling from JFK (John F. Kennedy Airport, New York) to DCA (Ronald Reagan National Airport) is predicted to cross the ZNY-ZDC ARTCC boundary at 12 Noon. A host-prediction uncertainty of 15 minutes implies that the flight is considered *active* (for the purpose of STA matrix creation) in ZNY up to 1215 Hours and also *active* in ZDC from 1145 Hours onwards. As

expected, higher *host-prediction* uncertainty buffer values leads to higher demand for codes.

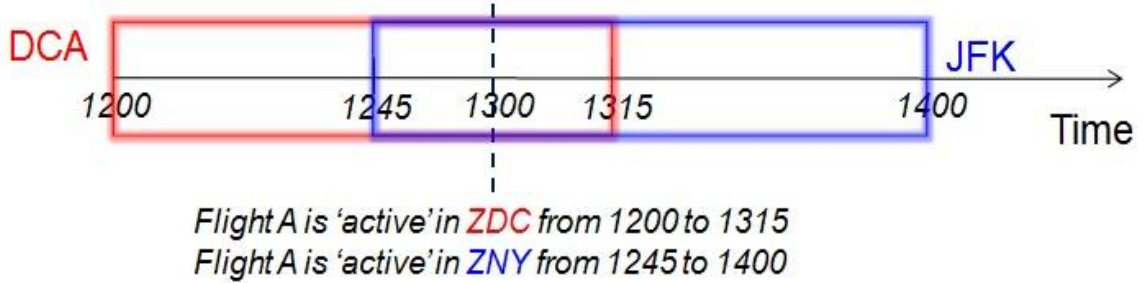


Figure 27: Boundary Crossing Time Uncertainty Buffer (Example of 15 minutes window)

5.2.2 STA Algorithm Details

The STA algorithm is composed of 6 steps.

Step1: Select list of active flights

Select the flights that are predicted to be active (need Beacon Code) in the current planning-window 'w'. For the current planning window 'w' (where $w = 1, 2, \dots, 24$), call the list of "active flights" F^w . The number of flights in F^w is denoted by 'k', i.e. $|F^w| = k$.

Step 2: Generate STA (Space-Time Adjacency) Matrix

Construct an STA matrix of length $k*(k-1)/2$ using the flight-plan and the host-prediction of boundary crossings times of all flight in F^w . The $[((i-1)*(i-2))/2+j]^{th}$ element

of STA corresponds to flight pair $i-j$, where $i > j$ and $i, j \in \{1, 2, \dots, k\}$. The value of that element is set to 1 if flight i and j are predicted to be in the same center at the same time for at least one instance, based on their flight plans and the host-prediction of their center boundary crossing times. Otherwise the element corresponding to flight $i-j$ in the STA is set to 0.

Step 3: Assign codes to “active” flights

For every flight ‘ i ’ in F^w , a candidate list of flights which may be STA (Space-Time Adjacent) to it is prepared. As code assignment within a planning-window is done in FCFS (First Come First Serve) order, the candidate list for a flight ‘ i ’ are flights 1 through $(i-1)$. Say the candidate list for flight ‘ i ’ in the planning-window ‘ w ’ is denoted by D_i^w .

For each flight ‘ j ’ in D_i^w (where $j = 1, 2, \dots, (i-1)$), check whether flights ‘ i ’ and ‘ j ’ are space-time adjacent (if $STA^w[(i-1)*(i-2)/2 + j] = 1$). If yes, then the code assigned to flight ‘ j ’ is appended to the “Conflict Codes” list of flight ‘ i ’ (CC_i^w). This process is applied to all flights in the candidate list of ‘ i ’ to complete the set CC_i^w . The mathematical representation of CC_i^w is shown in Equation 5.

Equation 5: Definition of “Conflict Codes” (CC_i^w) for a flight ‘ i ’ during planning-window ‘ w ’ of STA Algorithm

$$CC_i^w = \{C \mid \forall j < i, \forall C \in U, STA^w[(i-1)*(i-2)/2 + j] = 1\}$$

In Equation 6, U is the universal set of Beacon Code indices, i.e. 1 to 3348.

Next a list of codes which are Timed-Out in ARTCCs on flight i 's route is prepared using the Time-Out Matrix (TOM^w). Let this list of Timed-Out-Codes for flight ' i ' be denoted by TOC_i^w . The mathematical representation of CC_i^w is shown in Equation 6.

Equation 6: Definition of Timed-Out-Codes(TOC_i^w) for a flight ' i ' during planning-window ' w ' of STA Algorithm

$$TOC_i^w = \{C | \forall C \in U, \exists m \in \{1, 2, \dots, \beta_i\}, TOM^w[C, p_{i,m}] > 0\}$$

In Equation 6, U is the Universal set of Beacon Code indices and β_i denotes the total number of centers on flight i 's route. The m^{th} center of flight i 's route is represented by $p[i, m]$. The $(C, p_{i,m})$ element of Time-Out Matrix(TOM) for planning-window ' w ' is represented by $TOM^w[C, p_{i,m}]$.

The codes in sets TOC_i^w and CC_i^w are the list of codes “not permitted” to be assigned to flight i . The remaining codes in U are “permitted” to be assigned to flight i and they form a set of “Allowed Codes” AC_i^w as shown in Equation 7.

Equation 7: Definition of Allowed-Codes(AC_i^w) for flight i during planning-window ' w ' in STA Algorithm

$$AC_i^w = U - \{TOC_i^w \cup CC_i^w\}$$

As the objective is to find the minimum number of codes needed for assignment to all the flights in the population (whole day), the smallest code from set AC_i^w is assigned to flight ' i '. Say the code assigned to the i^{th} flight of planning-window ' w ' is denoted by x_i^w .

Step 4: Generate Time-Out Matrix (TOM)

The list of “Overlapping Flights” (OF^w) is constructed. This list is a subset of “active list” F^w , and consists of flights that are “active” (require Beacon Code) beyond Ω_w , the end-time of the current window ‘w’.

For every flight ‘i’ in OF^w , the code x_i^w assigned to it (in Step 3) is timed-out in all the center(s) falling on the route of flight ‘i’ beyond time Ω_w , i.e. end of current planning-window. For all such center(s), code x_i^w is timed for 30 additional minutes (CRDT- See 3.4.1 Beacon Code Life Cycle) beyond the predicted exit-time of flight ‘i’ from the corresponding center. For example, say flight AAL 123 is using code 2312 and is predicted to cross from ZNY to ZDC at 12:00 noon with a 15 minutes host-prediction uncertainty. In this scenario, code 2312 is timed out in ZNY until 12:45 (1200+15 minutes (uncertainty buffer)+30 minutes(CRDT)). In other words the [2312,16(ZNY)] element of TOM^w is set to 12:45. This implies that no other flight in ZNY can be assigned code 2312 until 12:45.

Step 5: Release Codes with Expired Time-Out epoch

Before the beginning of the following planning-window ($w+1$), the values in TOM^w for every code-center pair which are less than α_{w+1} are reset to zero. This ensures that all the codes whose time-out period expires before the start-time of the following planning-window are available for use (in the corresponding center).

Step 6: Increment w by 1 and repeat Step 1 to 5 until all flights have been assigned codes.

5.2.3 Results for Code Assignment through STA for Current Traffic (2007)

The STA algorithm for code assignment is tested using 5 high volume days of 2007: Jan 3, April 11, July 26, November 21 and December 19(See Figure 18). The traffic statistics for these days is summarized in Table 6. These days represent different seasonal traffic patterns in time-and-space.

Table 6: Statistics of Traffic in the CONUS for the 5 days of 2007 used as input for STA

Days(2007)	Total Flights	Start of Peak Quarter-Hour (UTC)	Number of Operations in Peak Quarter-Hour	Average Number of Flights per Quarter-Hour
3-Jan	43,649	17:30	4,897	3,033
11-Apr	43,966	21:30	5,019	3,013
26-Jul	48,721	21:15	5,302	3,277
21-Nov	46,202	18:30	5,541	3,228
19-Dec	47,145	22:15	5,355	3,219

The *actual* departure time for each flight is used as a proxy for its scheduled departure time. Also, the *actual* center-crossing times (obtained through Algorithm

described in Appendix A: 4DT-to-Center-Route Converter) are used as a proxy of *host-prediction* of center crossing times.

Table 7: Summary of the total number of codes required using STA for different values of “host-prediction” uncertainty (minutes)

Days (2007)	Total Flights	Uncertainty Buffer in "Host-Prediction" (Minutes)				
		0	15	20	25	30
3-Jan	43,649	1,310	1,677	1,799	1,911	2,027
11-Apr	43,966	1,286	1,668	1,782	1,896	2,012
26-Jul	48,721	1,290	1,694	1,821	1,940	2,065
21-Nov	46,202	1,501	1,947	2,096	2,244	2,362
19-Dec	47,145	1,340	1,772	1,889	2,015	2,139

The flights for each of these days are ordered by their departure time in ascending order and then code assignment is done for all flights using STA algorithm.

The duration of planning-window is set to 60 minutes.

A summary of the number of codes needed when codes are assigned using STA algorithm for each of the 5 days is shown in Table 7. The results are shown graphically in Figure 28. For 3rd January, 2007, the maximum number of Beacon Codes required during the day when exact boundary crossing time is known (“host-prediction” uncertainty of 0

minutes) is 1,310. If a 15 minute “host-prediction” uncertainty buffer is applied at each center crossing for all the 43,649 flights in the NAS on that day, then the total number of codes required is 1,677. For a 30 minute *host-prediction* uncertainty buffer, the maximum number of codes required among all the 5 days is 2,362 (Nov 21).

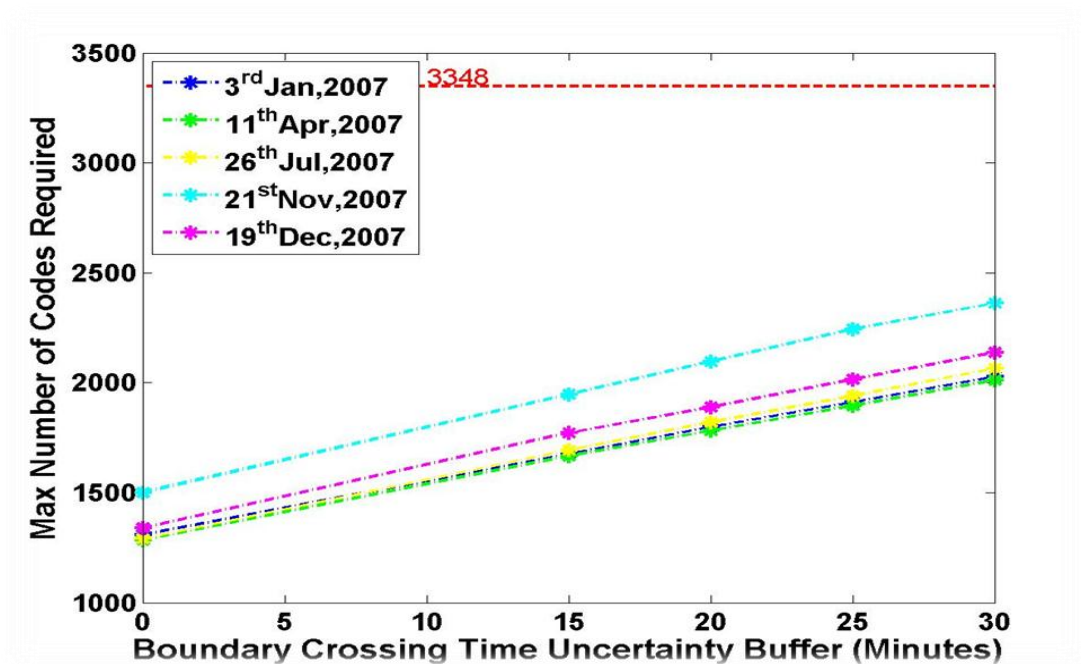


Figure 28: Summary of the total number of codes required using STA for current traffic (2007)

5.2.3.1 Linear Relationship between Uncertainty in Host-prediction and the Maximum number of Codes required

The slope of the dotted lines shown in Figure 28 represents the increase in number of codes required for every minute increase in the host-prediction uncertainty

parameter. The slopes for all the 5 days for each value of the host-prediction uncertainty parameter are shown in Table 8. For a given day, the relationship of number of codes required and host-prediction uncertainty parameter is “almost” linear (See Figure 28). The linear relationship can be established by comparing the slopes of the four ranges of host-prediction uncertainty values, i.e. 0-15, 15-20, 20-25 and 25-30 minutes. For any given day (represented by rows in Table 8) the slope of these lines are very similar; for example, for 3rd January the values of slopes for 0-15, 15-20, 20-25 and 25-30 ranges of host-prediction uncertainty parameters are 24.5, 24.4, 22.4 and 23.2 codes/minute respectively.

The number of codes required for a unit increase in uncertainty buffer varies (slope) across the 5 days and is ordered identically to the number of peak quarter-hour operations (4th column of Table 6). This implies that the higher the peak instantaneous traffic on a given day, the more sensitive it is to the duration of uncertainty in host-prediction.

Table 8: Sensitivity of Codes-required to Change in host-prediction Uncertainty

Days (2007)	Host-prediction Uncertainty Buffer Range (minutes)				Mean of Slope(Codes/Minute)	Rank of Mean Slopes
	0-15	15-20	20-25	25-30		
3-Jan	24.5	24.4	22.4	23.2	23.25	5
11-Apr	25.5	22.8	22.8	23.2	23.5	4
26-Jul	26.9	25.4	23.8	25	25.25	3
21-Nov	29.7	29.8	29.6	23.6	28.5	1
19-Dec	28.8	23.4	25.2	24.8	25.5	2

5.2.3.2 Time-of-Day Code-Usage

Code-usage by time of day is shown in Figure 29 through Figure 33. The figures include: the active flight count (solid black line), code usage for host-prediction uncertainty for 0 to 30 minutes (dotted colored lines) and maximum available codes (dotted red line parallel to x-axis).

The peak value of codes required for each value of host-prediction uncertainty is labeled at the corresponding x-y [time, number of codes being used] location. The red line parallel to the x-axis $y = 3,348$ represents the total number of codes available for assignment (DOT/FAA, 2009).

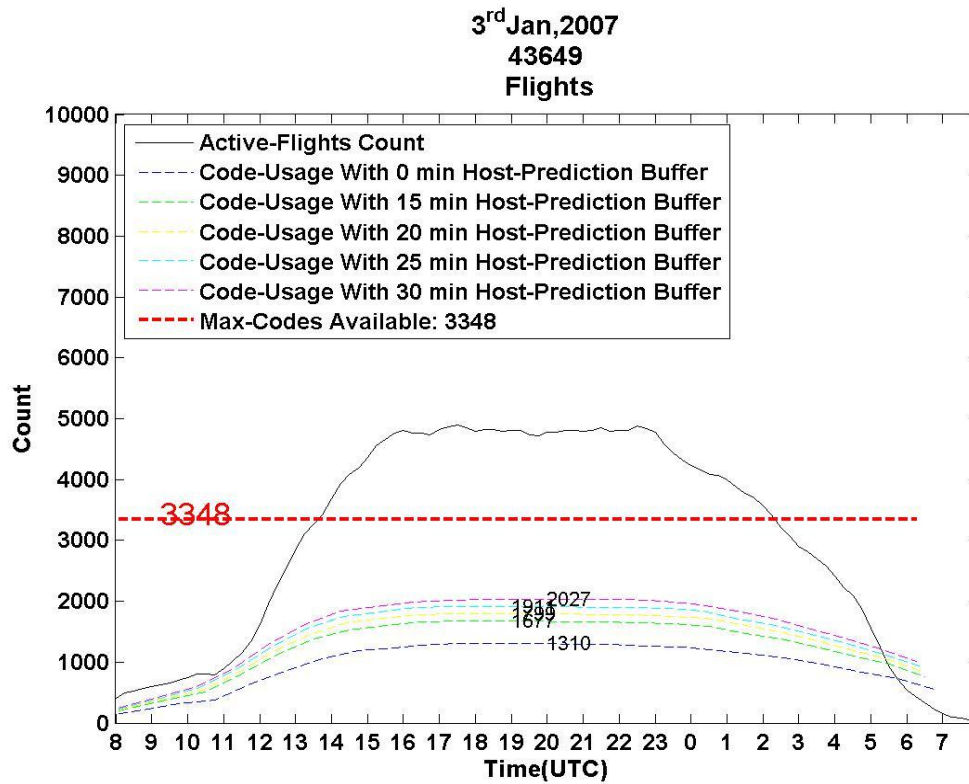


Figure 29: Code Usage by Time-Of-Day using STA Algorithm for 3rd Jan 2007

3rd Jan 2007 (Winter): The maximum number of codes required for doing code-assignment through STA algorithm is 1310, 1677, 1799, 1911 and 2027 for host-prediction uncertainty of 0,15,20,25 and 30 minutes respectively. There is an “almost linear” increase in the number of codes required for a unit increase in host-prediction buffer time for the range of 0 to 30 minutes (Table 8). The mean increase rate (slope) is 23.25 codes per minute.

The code-usage peaks between hours 20 and 21 for host-prediction uncertainty values of 0 minute and 30 minutes respectively. For host-prediction uncertainty of 15, 20 and 50 minutes, the code-usage peaks between hours 19 and 20. However, for 11th April 2007, the code usage peaks 22 and 23 for all the 5 values of host-prediction uncertainty buffer.

The number of codes required for Beacon Code assignment using STA algorithm never exceeds the total number of available codes for all the 5 high-volume days tested.

Similar charts are shown for 11th April, 26th July, 21st November and 19th December in Figure 30, Figure 31, Figure 32 and Figure 33 respectively.

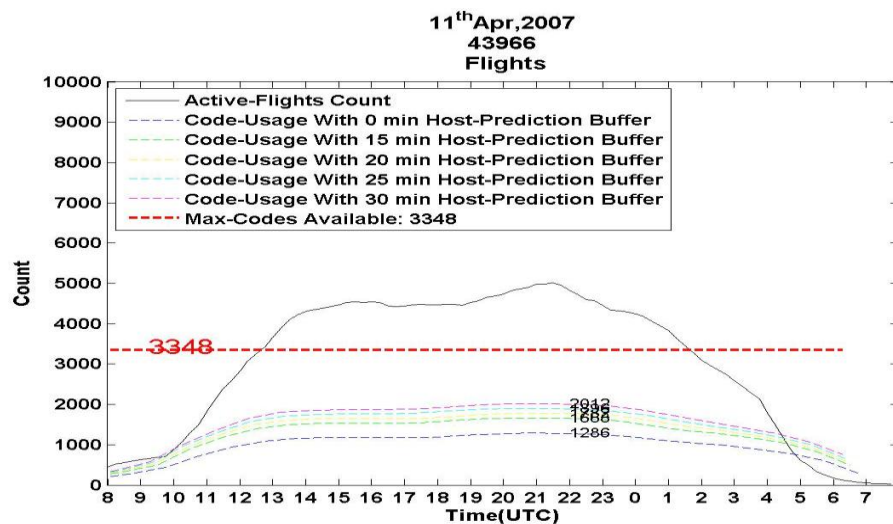


Figure 30: Code Usage by Time-Of-Day using STA Algorithm for 11th April 2007

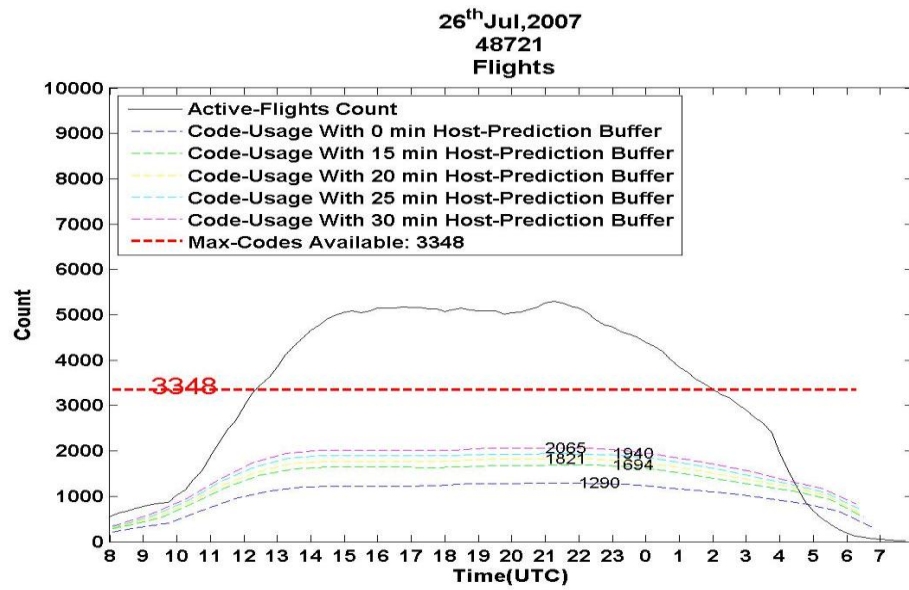


Figure 31: Code Usage by Time-Of-Day using STA Algorithm for 26th July 2007

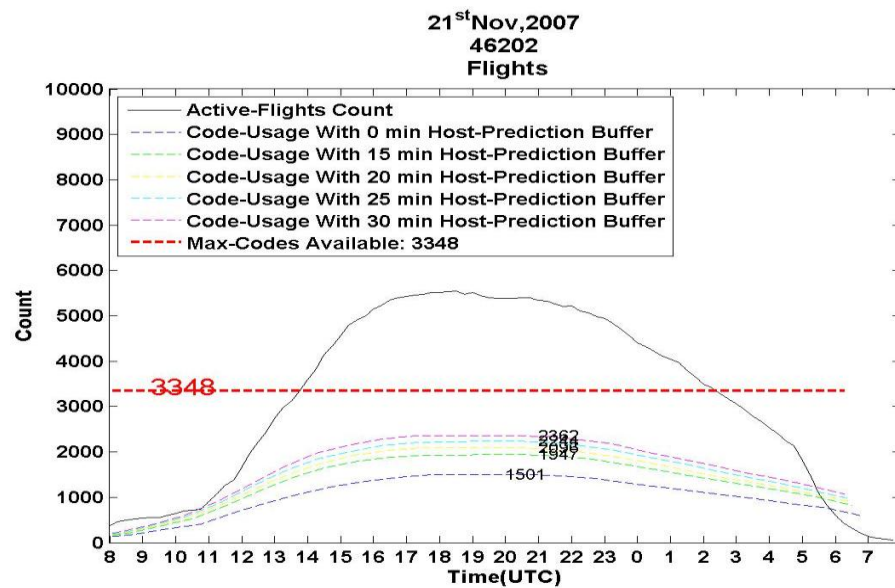


Figure 32: Code Usage by Time-Of-Day using STA Algorithm for 21st Nov 2007

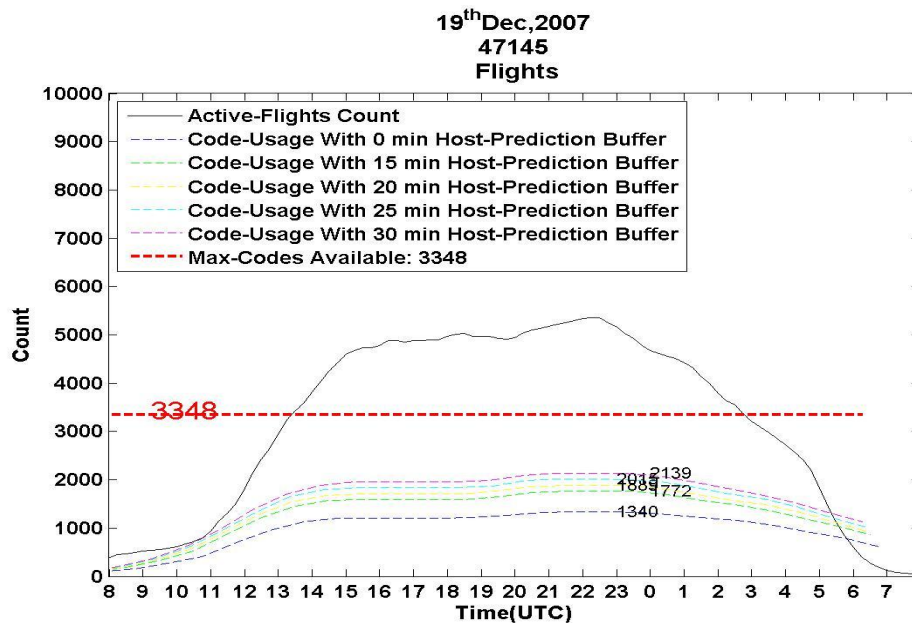


Figure 33: Code Usage by Time-Of-Day using STA Algorithm for 19th Dec 2007

5.2.4 Results for Code Assignment through STA for 1.5x Traffic Projection (2032)

The STA algorithm was tested with FAA's futuristic projection of 1.5x traffic levels. This 50% increase in traffic, according to FAA will occur by CE 2032 (FAA/APO, 2005). The objective was to test whether the STA algorithm is scalable to accommodate future traffic growth; more specifically whether it is possible to assign Beacon Codes to flights using STA algorithm without any reassignments for higher traffic level without exceeding the 3,348 codes limit.

Table 9: Statistics of 1.5x Traffic in the CONUS for 2 days of 2032 used as input for STA-R

Days(2032)	Total Flights	Start of Peak Quarter-Hour (UTC)	Number of Operations in Peak Quarter-Hour	Average Number of Flights per Quarter-Hour
3-Jan	58,125	9:30	6,208	4,755
11-Apr	58,446	7:45 AM*	6,376	4,858
		*12th April		

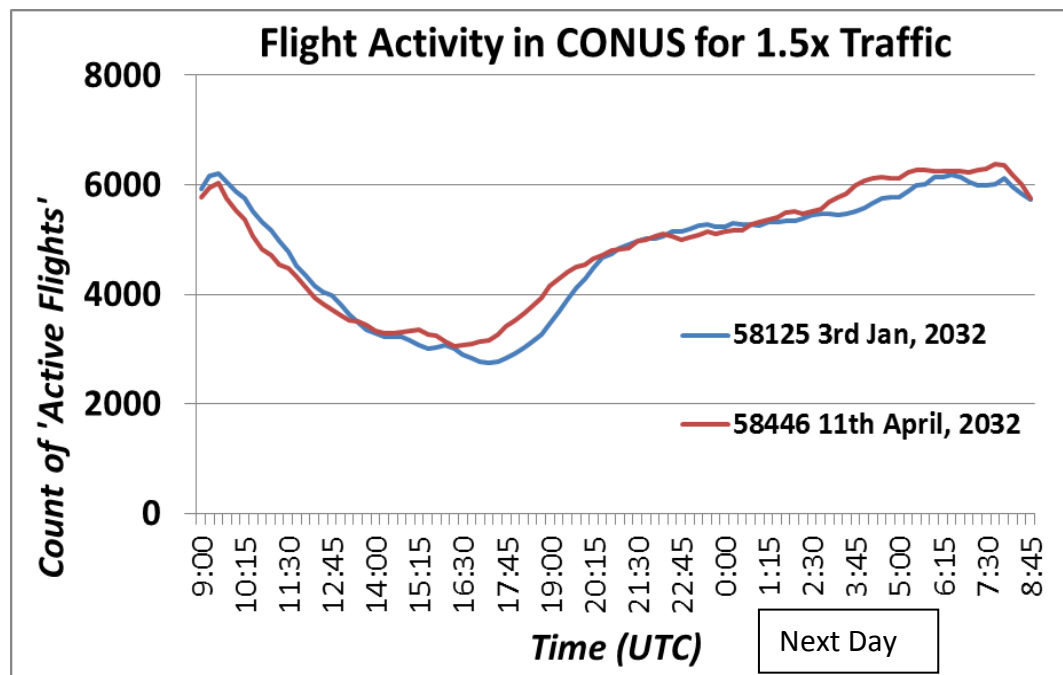


Figure 34: Count of "Active Aircraft" for 1.5x Traffic for each quarter-hour of the day

The 1.5x projection of traffic was obtained for 3rd Jan 2032 and 11th April 2032. The total number of flights on these days is projected to be 58,125 and 58,446 respectively. The traffic statistics for these days is summarized in Table 9. The number of “active flights” in the CONUS on these days as a function of time of day is shown in Figure 34.

The FAA’s projected schedule for each flight consists of its scheduled departure time and arrival time along with the departure and arrival airports. The schedule was converted to 4-D flight trajectories using NASA’s Future ATM Concepts Evaluation Tool (FACET) (NASA, 2010). Next, the 4-D flight trajectories are processed using “4DT-to-Center-Route Converter” tool. (Appendix A: 4DT-to-Center-Route Converter).

A summary of results for code assignment using STA algorithm for the two days with 1.5x traffic projection is shown in Table 10 and the values are shown graphically in Figure 36. For 3rd January, 2032, the maximum number of Beacon Codes required during the day when *exact* boundary crossing time is known (no host-prediction uncertainty) is 2,567. For 11th April, 2032, the maximum number of Beacon Codes required when *exact* boundary crossing time is known is 2,592. For both these test dates, the number of codes is less than 3,348 codes available for usage. If the uncertainty buffer in “host-prediction” of boundary crossing times is increased to 20 minutes, the number of codes required is 3,240 and 3,263 for 3rd Jan 2032 and 11th April 2032 respectively.

Table 10: Summary of Results for Code-Assignment for 1.5x Traffic through STA algorithm

Date (2032)	Tot-Flights	Uncertainty Buffer in "Host-Prediction" (Minutes)				
		0	15	20	25	30
3-Jan	58,125	2,567	3,079	3,240	3,399	3,567
11-Apr	58,446	2,592	3,094	3,263	3,432	3,595

However, when the uncertainty in host-prediction is increased to 25 minutes, the number of codes required by STA algorithm exceeds the available number of codes (3,348). For 3rd Jan 2032, 3,399 codes are required for uncertainty buffer value of 25 minutes. For 11th Apr 2032, the number of codes required is 3,432. The number of codes required is higher when the uncertainty buffer value is 30 minutes. The maximum number of codes required for different buffer values for both days are shown in Table 10.

Code-usage by time of day is shown in Figure 36 and Figure 37 for 3rd Jan 2032 and 11th April 2032 respectively.

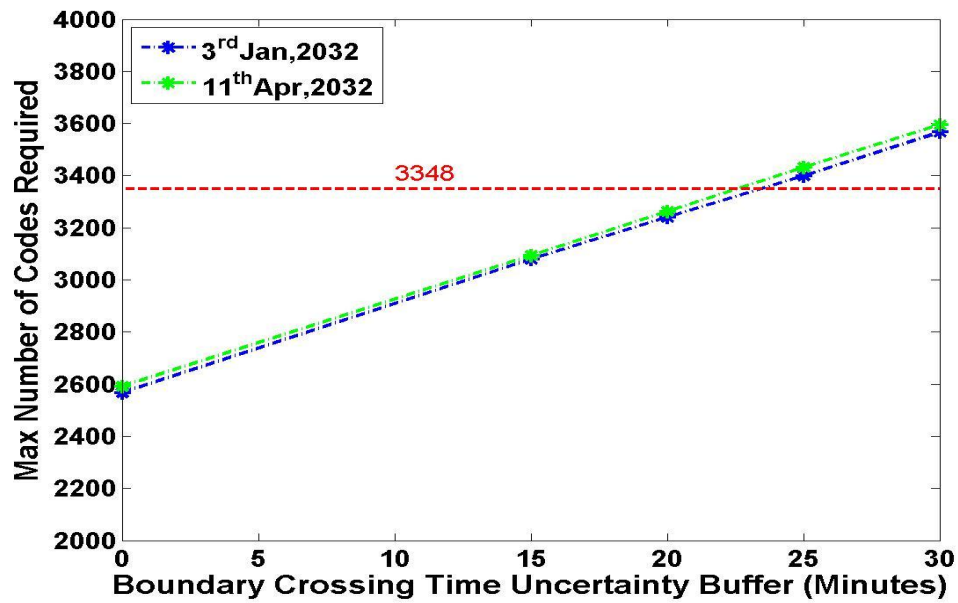


Figure 35: Summary of the total number of codes required using STA for 1.5x schedule

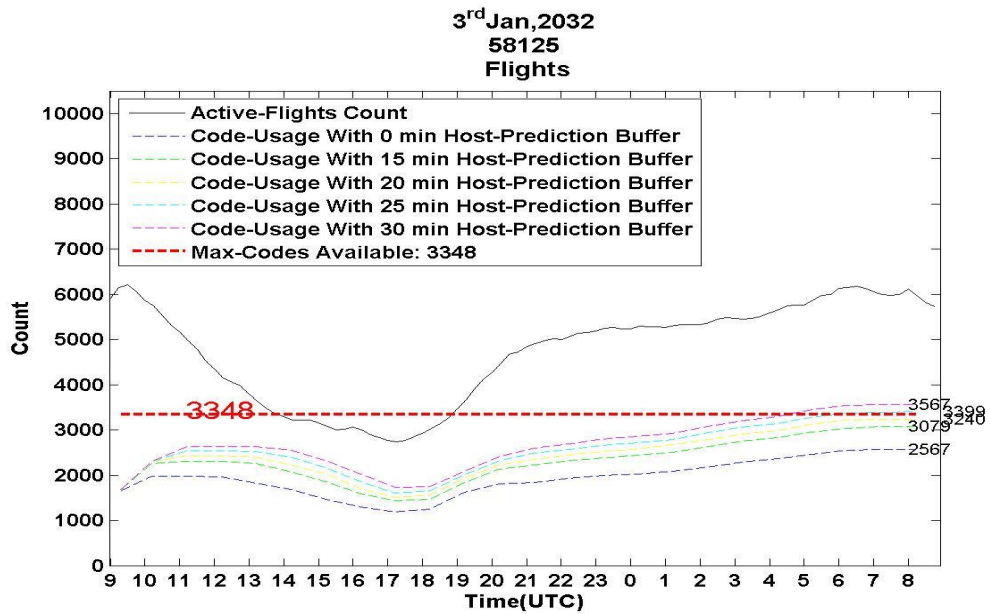


Figure 36: Code Usage by Time-Of-Day using STA Algorithm for 3rd Jan 2032(1.5x Traffic)

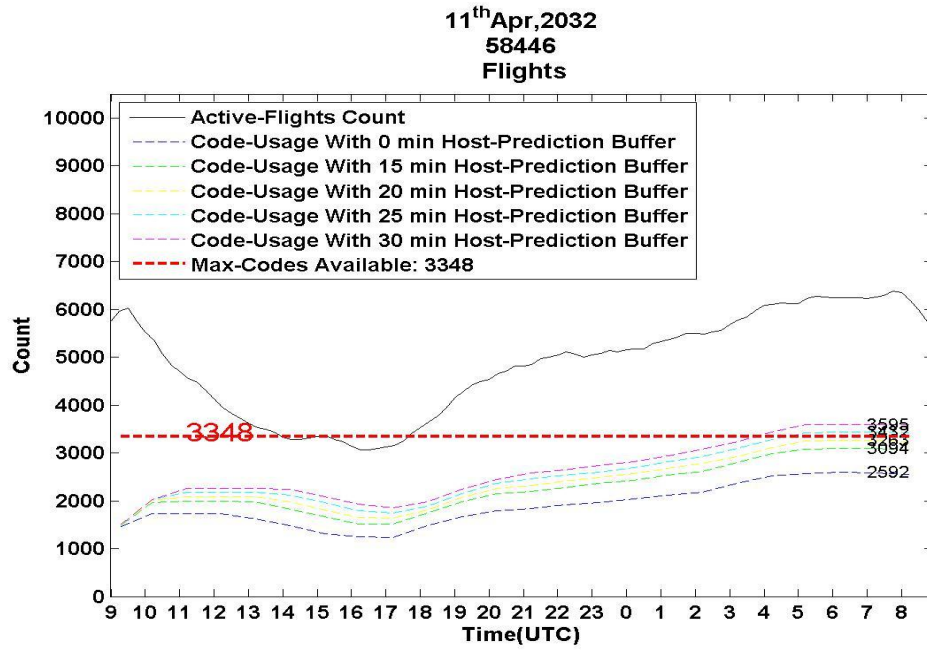


Figure 37: Code Usage by Time-Of-Day using STA Algorithm for 11th April 2032(1.5x Traffic)

The number of codes required to assign codes for the 1.5x traffic days using STA algorithm exceeds the available number of codes when host-prediction uncertainty is 25 minutes or higher. This result implies that for 1.5x traffic and a host-prediction uncertainty of 25 minutes or higher, it is not possible to assign codes to all flights in the CONUS without any *reassignment instances*. As a result, the STA algorithm was modified to *allow* (minimal) code-reassignments whenever needed (when all 3,348 codes have been used up). This algorithm is called STA-R (STA with Reassignments) and is described in the following section.

5.3 STA-R (STA with Reassignments) Algorithm

STA-R is an extension of STA algorithm that allows Beacon Code reassignments.

5.3.1 Data Structures and Parameters

The following data-structures are used in the STA-R algorithm:

1. Space-Time Adjacency(STA) Matrix, and
2. Code Time-Out Matrix (TOM).

The STA-R algorithm has the following parameters:

1. Planning-window duration,
2. Code Reassignment Threshold, and
3. Uncertainty buffer in Host-Prediction.

5.3.1.1 STA Matrix

The primary difference between STA and STA-R algorithm is the method of generation of the STA matrix through the flight routes and the host-prediction of center crossing times. In the STA algorithm, if a given pair of flights is predicted to be in the same center at *any* instance on their routes, the corresponding entry for the flights in the STA matrix is set to 1. This is done to prevent space-time adjacent flight pairs from being assigned the same code.

Unlike the STA algorithm, the STA-R algorithm allows code reassignments to occur. In other words, flights may be assigned multiple codes with different codes for

different centers on its route. In order to facilitate the code reassignment instances, the structure of the binary STA matrix needs to be modified. An element of the STA matrix in STA-R algorithm, corresponding to a given flight-pair, consists of the centers (if any) where the corresponding flight-pair are space-time adjacent. As a result of using center-specific STA matrix (instead of binary), it is possible to assign the same code, if needed, to the corresponding flight pairs in all other centers (where they are not Space-Time Adjacent).

5.3.1.2 Overlapping Flights(OF) List and Code Time-Out Matrix(TOM)

The definition of 'OF' and 'TOM' matrices for STA-R is same as STA algorithm (See 5.2.1.2 Overlapping Flights (OF) List and Code Time-Out Matrix (TOM)).

5.3.1.3 Planning window

The size of "planning-window" for STA-R is same as STA algorithm; typically set to 60 minutes.

5.3.1.4 Code-Reassignment Threshold (CRT)

The STA-R algorithm progresses exactly like the STA algorithm until a threshold of CRT codes have been used up for assignment to flights. This means that until CRT codes have been assigned to flights, all flights get a single code for all the centers on its route. This is done to reduce the computational overhead of generate center-specific STA Matrix for the flights when there are enough codes available.

The default value of CRT is set to 1500. This implies that until first 1500 codes have been assigned to flights, all the flights get a single code for their entire flight duration. After 1500 codes are in use, flights *may* be assigned center-specific codes (leading to reassignments).

5.3.1.5 Host-Prediction and Uncertainty Buffer

The STA-R algorithm is only tested for “uncertainty buffer” values of 25 minutes and 30 minutes because STA algorithm exceeded the total number of available codes (3,348) for these values of host-prediction uncertainty.

5.3.2 STA-R Algorithm Details

Step1: Select list of active flights

Select the flights that are predicted to be active (need Beacon Code) in the current planning-window ‘w’. For the current planning window ‘w’ (where $w = 1, 2, \dots, 24$), call the list of “active flights” F^w . The number of flights in F^w is denoted by ‘k’, i.e. $|F^w| = k$.

Step 2: Generate STA (Space-Time Adjacency) Matrix

The STA Matrix for the STA-R algorithm is a “matrix of arrays”, consisting of $k \cdot (k-1)/2$ arrays. The matrix is constructed using the flight-plan and the host-prediction of boundary crossings times. The $(((i-1) \cdot (i-2))/2 + j)^{\text{th}}$ element of this matrix corresponds to flight pair $i-j$, where $i > j$ and $i, j \in \{1, 2, \dots, k\}$. A non-zero element of the STA matrix implies

that the flight-pair corresponding to that element are space-time adjacent in at last one center on their routes. The element(s) at that position correspond to the center(s) where flights i and j are *predicted* to be space-time adjacent.

Step 3: Assign codes to “active” flights

For every flight ‘ i ’ in F^w , a list of “allowed codes” for all centers on its route is generated. This list of “allowed codes” for a given flight i , for the m^{th} center on its route, i.e. $p[i,m]$ (where $m=1,2..\beta_i$), is an intersection of the following two sets:

Conflict Codes ($CC_{i,m}^w$): $CC_{i,m}^w$ is a set of codes that cannot be used by flight i in planning-window ‘ w ’ for the m^{th} . It is generated by iteration through all flights ‘ $j < i$ ’ in F^w , and checking whether the element in STA corresponding to flight pair $[i,j]$ has center $p[i,m]$ in it, i.e. if flight i and j are space-time adjacent in center $p[i,m]$. If so, then the code assigned to flight ‘ j ’ is appended to the set of “Conflict Codes”, i.e. $CC_{i,m}^w$. After this process is applied to all flights $j(<i)$, the set $CC_{i,m}^w$ is completed. The mathematical representation of $CC_{i,m}^w$ is shown in Equation 8.

Equation 8: Definition of "Conflict Codes" ($CC_{i,m}^w$) for flight ‘ i ’ during planning-window ‘ w ’ of STA-R algorithm

$$CC_{i,m}^w = \{C \mid \forall j < i, \forall C \in U, p[i,m] \in \{STA^w[(i-1)*(i-2)/2 + j]\}\}, \text{where}$$

U is the Universal set of Beacon Codes available.

Timed-Out Codes(TOC_i^w): TOC_i^w is a set of codes that cannot be used by flight 'i' in planning-window 'w' for any center on its route. The TOC generation for STA-R algorithm is identical to STA and is represented mathematically in Equation 6.

The codes in sets $CC_{i,m}^w$ and TOC_i^w are the list of codes “not permitted” to be assigned to flight 'i' in the m^{th} center($m=1,2,..,\beta_i$) on its route, i.e. $p[i,m]$. The remaining codes in U are “permitted” to be assigned to flight i and they form a set of “Allowed Codes” $AC_{i,m}^w$ as shown in Equation 7.

Equation 9: Definition of Allowed-Codes($AC_{i,m}^w$) for flight i for the m^{th} center on its route during planning-window 'w' in STA-R Algorithm

$$AC_{i,m}^w = U - \{TOC_i^w \cup CC_i^w\}, \text{ where } m = \{1,2,..,\beta_i\}$$

Let $x_{i,m}^w$ denote the code assigned to the i^{th} flight of planning-window 'w' for the m^{th} center($p[i,m]$) on its route. Let X_i^w denote the set of codes that For a given set of “allowed codes” $AC_{i,m}^w$ for a flight i, it is possible to pick $x_{i,m}^w$ assign code(s) to the flight in multiple ways.²

For example, flight ($i=1$) AAL111 whose route $\{16,4,20\}$, i.e. $p[1,1]=16$, $p[1,2]=4$, $p[1,3]=20$, and $\beta_{i=1}=3$. Also, say the set of codes allowed are as follows (assume $w=1$):

$$AC_{1,1}^1 = \{2321, 5212, 3625\} \quad [\text{Codes Permitted in Center 16}]$$

² If there are no code reassignments for this flight, then $x_{i,m}^w = x_{i,m+1}^w$, for $m=1,2,.. (\beta_i-1)$. This also implies that $|X_i^w|=1$. However, the need for doing code assignments through STA-R is based on the fact that it is not possible to achieve zero code reassignments for the entire population of flights being assigned codes (See 5.2.4 Results for Code Assignment through STA for 1.5x Traffic Projection (2032)).

$AC_{1,2}^1 = \{5212\}$ [Codes Permitted in Center 4]

$AC_{1,3}^1 = \{6230, 3625\}$ [Codes Permitted in Center 20]

In general, the number of possible ways of assigning codes to flight i in time

window w is equal to $\prod_{m=1}^{\beta_i} |AC_{i,m}^w|$. In this example, the six ($3 \times 1 \times 2$) possible choices of X_1^w

are:

$\{(2321, 5212, 6230), (2321, 5212, 3625), (5212, 5212, 6230), (5212, 5212, 3625),$
 $(3625, 5212, 6230), (3625, 5212, 3625)\}$

Not all of these 6 permutations are equally “good”. There can be two measures of “goodness”, namely, total number of codes in set X_1^w , i.e. $|X_1^w|$ and the number of reassignments (lower bound $|X_1^w| - 1$).

If the objective is to find the minimum number of codes needed to “cover” all the possible sets of $AC_{i,m}^w$ for a given flight i , then this problem is analogous to the classical “set-covering problem” (Koncal and Salkin, 1973). Given a family of subsets of a universe U , the objective of “set-covering problem” is to identify the smallest number of subsets from the family whose union contains all elements in the universe. The “set-covering” problem is known to be NP-Complete (Cook, 1971).

For the purpose of Beacon Code assignment, minimizing the total number of codes required is not as important as reducing the instances of reassignments. This is

because all of the 3,348 codes are at the disposal to be used and utilizing fewer codes at the expense of causing reassignments is not desirable.

For example, say the two possible options for X^w_1 are (2,1,2,1,2) and (3,3,4,4,5). The first option uses only two codes (1 and 2) and the second option uses three codes (3, 4 and 5). However, if the first option is chosen, it leads to a code reassignment at every hand-off, i.e. a total of 4 reassignments, as compared to the second option which has fewer (3) reassignments. In such a case, the second option should be chosen.

The algorithm to select $x^w_{i,m}$ (assigned codes) given $AC^w_{i,m}$ (allowed codes, where $m=1,2,.. \beta_i$) for a flight i , is described below:

- (i) Initialize $m=1$, $B=\text{null}$.
- (ii) $B = AC^w_{i,m=z} \cap AC^w_{i,m=z+1}$
- (iii) If B is null, then $x^w_{i,m}$ is an element chosen randomly from set $AC^w_{i,m}$. Go to Step (v).
- (iv) $x^w_{i,m}$ is an element selected randomly from B .
- (v) $m = m+1$
- (vi) If $m > \beta_i$, then STOP.
- (vii) If $x^w_{i,m-1} \in AC_m$, THEN $x^w_{i,m} = x^w_{i,m-1}$. Go to Step (v)
ELSE Go to Step (ii).

The performance of this algorithm can be improved by increasing the size of the look-ahead window, i.e. instead of finding an intersection of current set with *only* the

next set, an intersection of the current set with the next 'k' sets could be found where k can be 2,3,...n-i+1. The higher the value of k, the better would be the likelihood of finding codes which leads to fewer reassignments. However, the higher the value of k, the higher is the computational time. Also, with increasing value of k, the likelihood of finding an intersecting code decreases.

However, the mean number of hand-offs($\beta - 1$) that a flight goes through in CONUS is 1.3. This implies that the mean number of ARTCCs that a flight goes through is $2.3(1.3+1)$ (See Hand-Offs section in 3.3 Beacon Code Reassignment Statistics). This implies that on an average a flight travels through 2.3 centers. For all such flights, a look-ahead window size of 1 suffices.

Step 4: Generate Time-Out Matrix (TOM)

Next, the list of "Overlapping Flights" (OF^w) is constructed. This list is a subset of "active list" F^w , consisting of flights that are "active" (require Beacon Code) beyond Ω_w , the end-time of the current window 'w'.

For every flight 'i' in OF^w , the code(s) $x_{i,m}^w$ assigned to it(in Step 3) are timed-out in the corresponding center(s) falling on its route beyond time Ω_w , i.e. end of current planning-window. For all such center(s), code $x_{i,m}^w$ is timed for 30 additional minutes(CRDT- See 3.4.1 Beacon Code Life Cycle) beyond the predicted exit-time of flight 'i' from that center $p[i,m]$. For example, say flight AAL 123 is using code 2312 in ZNY and is predicted to cross from ZNY to ZDC at 12:00 noon with a 15 minutes

uncertainty buffer. In that case, code 2312 is timed out in ZNY until 12:45 (1200+15 minutes (uncertainty buffer)+30 minutes(CRDT)). In other words $TOM^w(2312,16(ZNY))$ is set to 12:45. This implies that no other flight in ZNY can be assigned code 2312 before 12:45.

Step 5: Release Codes with Expired Time-Out epoch

This step is same as Step 5 of STA algorithm. Before the beginning of the following planning-window “w+1”, all the values in TOM^w for every [code,center] which are less than α_{w+1} are reset to zero. In other words all the codes whose time-out period expires before the start-time of the following planning-window are available for use(in the corresponding center).

Step 6: Increment w by 1. Repeat Step 1 to 5 until all flights have been assigned codes.

Say there are ‘n’ flights in the current time-step that needs to be allocated codes. These ‘n’ flights are pre-sorted by their departure time.

5.3.3 Results for Code Assignment through STA-R for 1.5x Traffic Projection (2032)

The STA algorithm is tested with increased traffic levels using FAA’s 1.5x projection of traffic (2032). The 1.5x projection of schedule was obtained for 3rd Jan 2032 and 11th April 2032. The number of “active flights” in CONUS on these days is shown in Figure 34. The traffic statistics for these days is summarized in Table 9.

Summary of results for code assignment using STA-R algorithm for two days with 1.5x traffic is shown in Table 11. The STA-R algorithm is only tested for “uncertainty buffer” values of 25 minutes and 30 minutes because STA algorithm exceeded the total number of available codes (3,348) for these values of host-prediction uncertainty.

For Jan 3, 2032, the total number of Beacon Code reassignment instances for 25 and 30 minute “host-prediction” uncertainty are 883 and 935 respectively. The total number of hand-offs that happen for 58,125 flights on this day is 74,164 (1.28 hand-offs per flight). This implies that the probability of flight getting a code reassigned when crossing an ARTCC boundary is 1.19% (883/74,164) when the uncertainty in host-prediction of center-crossing time is 25 minutes. For a 30 minute uncertainty buffer on 3rd Jan 2032, the likelihood of code-reassignment is 1.26% (935/74,164). Similarly, for 11th April, 2032, the likelihood of code reassignments when using STA-R algorithm for code assignment are 1.12% and 1.29% for “host-prediction” uncertainty of 25 and 30 minutes respectively.

Table 11: Summary of Results for Code-Assignment for 1.5x Traffic through STA-R algorithm

Date (2032)	Total Flights	Total Hand- Offs	Number of Reassignments		Likelihood of Reassignments %	
			25 min "Host- Prediction" Uncertainty	30 min "Host- Prediction" Uncertainty	25 min "Host- Prediction" Uncertainty	30 min "Host- Prediction" Uncertainty
3rd Jan	58,125	74,164	883	935	1.19	1.26
11th Apr	58,446	76,284	853	983	1.12	1.29

5.4 Conclusion

For the 5 high volume days of 2007, it was possible to allocate a code to all flights using STA algorithm without exceeding the available number of codes(3,348) such that there were *no* instances of code reassignments in the CONUS. Also, when a 30 minute host-prediction uncertainty was applied to each ARTCC crossing instance in the CONUS, it was still possible to allocate codes to all flights in the NAS using a maximum of 2,362 codes(21 November, 2007) without any code reassignments.

For 1.5x traffic of 2032, it is possible to allocate codes using STA when the host-prediction uncertainty buffer is 20 minutes or less. For “host-prediction” uncertainty values of 25 minutes and 30 minutes, STA needs more than available number of codes.

The STA-R algorithm, an extension of STA, which allows reassignments if required achieves code assignment for 1.5x traffic projection with 25 and 30 minutes host-prediction uncertainty using the available 3,348 codes and leading to less than 1.29% code reassignment likelihood.

Chapter 6: Hybrid Method of Beacon Code Assignment Through Clusterization of CONUS

This section describes a hybrid method for code assignment using clusterization of CONUS and STA algorithm. This method of code assignment is *less* sensitive to uncertainty in host-prediction of center crossing times when compared to STA algorithm. The method is based on “partitioning” the 20 centers in the CONUS into ‘K’ (<20) group of ARTCCs called clusters. The clusters formed are collectively exhaustive (of the CONUS) and pairwise mutually exclusive. Clusterization is done without redrawing of center boundaries.

Each cluster is allocated an *exactly identical* set of codes for assignment to flights that remain within its boundary. All other flights that cross cluster boundary at least once are assigned codes from a complimentary bucket of codes.

The first subsection of this chapter provides definition of terminology used in this chapter. The second subsection provides an overview of the hybrid method of code assignment using clusterization and STA. The next section describes clusterization in detail along with the description of the optimization model for clusterization. Next, the

results and sensitivity analysis of the result of assigning codes through this hybrid model for historical days are discussed. The last subsection summarizes the conclusions.

6.1 Definitions

This section defines the terminology used in this chapter.

6.1.1 Clusters

A “cluster” is defined as a subset of ARTCCs in the CONUS such that all the clusters are pairwise mutually exclusive and collectively exhaustive. C_1, C_2, \dots, C_{20} represents the 20 ARTCCs in the CONUS and L_1, L_2, \dots, L_K represents the ‘K’ clusters. Also, CONUS, the Universal set of ARTCCs is represented by U. Then for clusters to be collectively exhaustive and pairwise disjoint the following two conditions should be satisfied:

$$(i) \quad \bigcup_{m=1}^K L_m = U \text{ (Collectively Exhaustive)}$$

$$(ii) \quad L_m \cap L_n = \{\}, \forall m, n = [1, 2..K], n < m \text{ (Pairwise mutually}$$

exclusive)

This collective exhaustive property of cluster ensures that all ARTCCs in the CONUS belong to a cluster. The mutually exclusive property of clusters ensures that the clusters are geographically independent entities, i.e. they could share codes from the same bucket and assign to flights which would not cross cluster boundaries.

Besides the two conditions mentioned above, it is also ensured that all the centers belonging to a given cluster are *geographically contiguous*. If the CONUS is represented as an undirected graph, where each of the 20 centers represent nodes and all adjacent centers are connected by a link (Figure 40), then each cluster should be a “*connected graph*”. This condition is necessary because each cluster assigns codes to all intra-cluster flights from its own bucket. If all the centers belonging to a given cluster are not contiguous then the flight may not remain within that cluster’s boundary for its entire flight duration.

6.1.2 Intra-Cluster Flights

Flights that remain inside a cluster for its entire duration are defined to be *Intra-Cluster* flights for that cluster. It must be noted here that all internal flights (flight which do not cross any center boundaries) are intra-cluster flights.

6.1.3 Inter-Cluster Flights

All flights which are not Intra-Cluster are Inter-Cluster flights. These flights cross a cluster boundary at least once. Such flights are categorized as *Inter-Cluster* flights. It must be noted here that ARTCCs in CONUS are not convex (Figure 47). As a result, a flight whose origin and destination airports are in the same cluster *may* still possibly be an *Inter-Cluster* flight.

6.2 Overview of Hybrid Code-Assignment method through clusterization

This section describes the hybrid code assignment method that assigns codes to flights by classifying them as intra-cluster or intra-cluster flights. The system is shown in Figure 38. The top half of the figure shows the “clusterization” process and the bottom half of the figure shows the process of code-assignment to flights post-clusterization of CONUS.

6.2.1. Clusterization

First, ETMS 4D trajectories of flights is converted to flight trajectories with center exit and entry times. This is done using the 4DT to center route converter algorithm (Appendix A: 4DT-to-Center-Route Converter). The list of flights along with their center entry and exit times is then input to the MILP optimization model for clusterization (6.4 Description of Optimization Model).

The result of clusterization is the grouping of centers into clusters. The cluster definitions are then used to partition the universal code set (1 to 3348) into two disjoint buckets ‘B’ and ‘C’.

All “clusters” are pre-allocated an *exactly identical* copy of bucket ‘B’ for assignment to its intra-cluster flights. If B_m represents the code-bucket for cluster ‘m’ (where $m = 1, 2, \dots, K$), then $B_1 = B_2 = \dots = B_K = B$. The sharing of identical codes among clusters for assignment to its intra-cluster flights is possible because the clusters are geographically

disjoint; meaning flights which are intra-cluster for one cluster would never be in the same ARTCC with intra-cluster flights of another cluster.

The “inter-cluster” flights are assigned codes from another bucket ‘C’, which is complimentary to B. This implies that buckets B and C have no common elements, i.e. $(B \cap C = \{\})$.

As the total number of codes available is 3348, it implies that for UBCAS to successfully work, clusterization should be done such that:

$$|B| + |C| \leq 3348.$$

are assigned codes from the bucket 'C' using STA algorithm. The intra-cluster flights are assigned codes in the FCFS order from the bucket ' B_k ' of the cluster 'k' it belongs to.

This process is repeated until all the flights have been assigned Beacon Codes.

6.3 Clusterization

This section describes the relationship between number of clusters and codes required. Next, the challenges in formulating the clusterization problem as a ware-house location problem are stated.

6.3.1 Relationship between Number of Clusters and Number of Codes Used(K)

The number of clusters 'K' can range from 1(entire NAS in a single cluster) to 20(representing 20 centers in the CONUS). If the entire NAS is treated as a single cluster ($K=1$) and code assignment was done centrally based on First Come First Served method, then, on a given day the number of Beacon Codes required would be equal to the peak instantaneous aircraft count. From the analysis of 5 high volume days of 2007, the peak value for instantaneous code demand is 8,121(Figure 20). As this value exceeds the number of codes available (3,348), Universal/Central code assignment is not feasible if for a *single cluster* NAS with FCFS assignment.

As the number of clusters increases, the geographic expanse of each cluster shrinks. This results in reduction of number of intra-cluster flights for individual clusters.

However, with increasing number of clusters, the potential number of inter-cluster flights increases.

6.3.2 Goals of Clusterization

The method of Universal Beacon Code Assignment (Figure 38) partitions the CONUS into 'K' clusters such that:

1. The maximum instantaneous intra-cluster traffic (demand for codes) for each cluster is minimized. This ensures that there is maximum *code sharing* among the geographically independent clusters, and also that the number of codes to be shared by clusters is kept at a minimum.
2. All the ARTCCs belonging to a given cluster should be connected. This constraint is necessary because eventually each cluster assigns codes to all intra-cluster flights from its own bucket. If all the centers belonging to a given cluster are not contiguous then the flight may not remain within that cluster's boundary for its entire flight duration.

6.3.3 Challenges in Formulation as a Warehouse Location Problem

The Clusterization problem is formulated as a warehouse-location problem. The warehouse location problem has many variations, but a good description of the problem is given by Feldman (Feldman, Lehrer, and Ray, 1966). He states:

The warehouse location problem involves the determination of the number and sizes of service centers (warehouses) to supply a set of demand centers. The objective is to locate and size the warehouses and determine which demand centers are supplied

from which warehouses so as to minimize total distribution costs. This distribution cost is the total transportation cost, which is assumed linear, plus the cost of building and operating the warehouse.

The warehouse location problem is adapted for this dissertation. Each ARTCC is a demand center. There are total of K warehouses, where K is the number of clusters desired. Each of these warehouses is an ARTCC which “supplies” to the demand centers. Each demand center is to be “served” by exactly one warehouse.

6.3.3.1 Ensuring Cluster Contiguity: Hop-Distance Definition

In a typical warehouse location problem, “distribution cost” is defined in terms of total transportation cost. The equivalent of “distribution cost” in the formulation presented here is the “hop-distance” between a given pair of “warehouse” and “demand center” (both of which represent ARTCCs).

Floyd-Warshall algorithm, a recursive graph analysis method to find shortest paths in a weighted graph is used (*Introduction to Algorithms*, 2009, pp. 558-565). The pseudo-code for the algorithm is shown in Figure 39. Given a graph with ‘ n ’ vertices, the algorithm determines $p(i,j)$, the shortest distance from node i to node j for each pair of vertices.

```

procedure FloydWarshall ()
for k := 1 to n
  for i := 1 to n
    for j := 1 to n
      p(i,j) = min (p(i,j), p(i,k)+p(k,j))

```

Figure 39: Pseudocode for FloydWarshall Algorithm (Source: (*Introduction to Algorithms*, 2009, pp. 558-565))

The graphical representation of CONUS (Figure 40) comprises of 20 nodes and 45 links. Using Floyd-Warshall algorithm, the shortest distance of each node from all the other nodes is obtained. This is shown in Table 12.

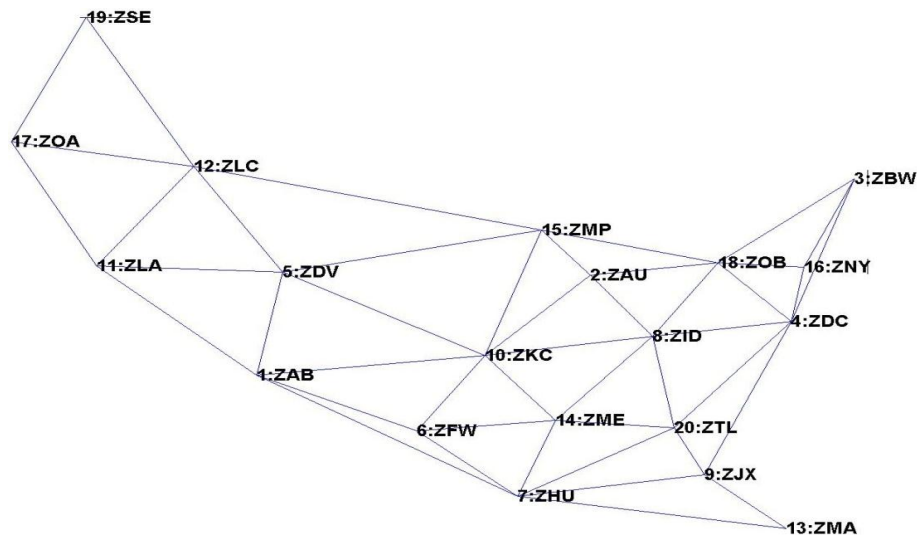


Figure 40: Graph representation of CONUS in Node-Link format

The matrix shown in Table 12 is symmetric because the CONUS graph shown in Figure 40 is undirected and the distance of a pair of nodes from each other is symmetric. An elements in position $[i,j]$ in the matrix shown in Table 12 represents the shortest-distance in terms of number of links traversed to get from node i to node j or vice-versa. For example, a value of 2 for element $[1,2]$ (ZAB, ZAU) implies that the minimum number of links needed to be traversed to get to ZAB from ZAU(or vice-versa) is 2. The shortest path in this case happens to be ZAB-ZKC and ZKC-ZAU. Another example is element $[1,3]$ (ZAB,ZBW), which has a value of 4 in the matrix. This implies that at least 4 links need to be traversed to get from ZAB to ZBW. It can be established from Figure 40 that (one of) the shortest path in this case is ZAB-ZKC-ZAU-ZOB-ZBW.

Table 12: Shortest-Distance of ARTCC-pairs in the CONUS($p(i,j)$)

	ZAB	ZAU	ZBW	ZDC	ZDV	ZFW	ZHU	ZID	ZJX	ZKC	ZLA	ZLC	ZMA	ZME	ZMP	ZNY	ZOA	ZOB	ZSE	ZTL
ZAB	1	2	4	3	1	1	1	2	2	1	1	2	2	2	2	4	2	3	3	2
ZAU	2	1	2	2	2	2	3	1	3	1	3	2	4	2	1	2	3	1	3	2
ZBW	4	2	1	1	3	4	3	2	2	3	4	3	3	3	2	1	4	1	4	2
ZDC	3	2	1	1	3	3	2	1	1	2	4	3	2	2	2	1	4	1	4	1
ZDV	1	2	3	3	1	2	2	2	3	1	1	1	3	2	1	3	2	2	2	3
ZFW	1	2	4	3	2	1	1	2	2	1	2	3	2	1	2	4	3	3	4	2
ZHU	1	3	3	2	2	1	1	2	1	2	2	3	1	1	3	3	3	3	4	1
ZID	2	1	2	1	2	2	2	1	2	1	3	3	3	1	2	2	4	1	4	1
ZJX	2	3	2	1	3	2	1	2	1	3	3	4	1	2	3	2	4	2	5	1
ZKC	1	1	3	2	1	1	2	1	3	1	2	2	3	1	1	3	3	2	3	2
ZLA	1	3	4	4	1	2	2	3	3	2	1	1	3	3	2	4	1	3	2	3
ZLC	2	2	3	3	1	3	3	3	4	2	1	1	4	3	1	3	1	2	1	4
ZMA	2	4	3	2	3	2	1	3	1	3	3	4	1	2	4	3	4	3	5	2
ZME	2	2	3	2	2	1	1	1	2	1	3	3	2	1	2	3	4	2	4	1
ZMP	2	1	2	2	1	2	3	2	3	1	2	1	4	2	1	2	2	1	2	3
ZNY	4	2	1	1	3	4	3	2	2	3	4	3	3	3	2	1	4	1	4	2
ZOA	2	3	4	4	2	3	3	4	4	3	1	1	4	4	2	4	1	3	1	4
ZOB	3	1	1	1	2	3	3	1	2	2	3	2	3	2	1	1	3	1	3	2
ZSE	3	3	4	4	2	4	4	4	5	3	2	1	5	4	2	4	1	3	1	5
ZTL	2	2	2	1	3	2	1	1	1	2	3	4	2	1	3	2	4	2	5	1

6.3.3.1.1 Exponential Hop-Distance Definition

The clusterization problem described here is analogous to the classical problem of Graph-partitioning (Lukes, 1975). Graph-partitioning is known to be a NP-Complete problem (Soundararajan and Sarkar, 2003). This problem is further aggravated by the fact that the partitions (clusters) desired should be “contiguous”.

In order to implement the contiguity of clusters, the shortest-distance obtained from Floyd-Warshall implementation for CONUS (Table 12) is modified. The rationale is to *penalize* exponentially as the distance increases between ARTCC (node) pairs. This new distance between a given pair of nodes i and j is called ‘X-exp distance’ and is denoted by $d(i,j)$.

If a given pair of nodes i and j are adjacent, then the $d(i,j)$ remains 1, otherwise $d(i,j) = X^{p(i,j)}$. The higher the value of X , higher is the degree of penalization for distance. For the purpose of this research, value of $X=4$ is chosen. The values of ‘d’ for all ARTCC pairs in CONUS is calculated and shown in Table 13.

Table 13: "Exp Distance" of ARTCC-pairs in the CONUS with $X=4 \cdot (d(i,j))$

	ZAB	ZAU	ZBW	ZDC	ZDV	ZFW	ZHU	ZID	ZJX	ZKC	ZLA	ZLC	ZMA	ZME	ZMP	ZNY	ZOA	ZOB	ZSE	ZTL
ZAB	1	16	256	64	1	1	1	16	16	1	1	16	16	16	16	256	16	64	64	16
ZAU	16	1	16	16	16	16	64	1	64	1	64	16	256	16	1	16	64	1	64	16
ZBW	256	16	1	1	64	256	64	16	16	64	256	64	64	64	16	1	256	1	256	16
ZDC	64	16	1	1	64	64	16	1	1	16	256	64	16	16	16	1	256	1	256	1
ZDV	1	16	64	64	1	16	16	16	64	1	1	1	64	16	1	64	16	16	16	64
ZFW	1	16	256	64	16	1	1	16	16	1	16	64	16	1	16	256	64	64	256	16
ZHU	1	64	64	16	16	1	1	16	1	16	16	64	1	1	64	64	64	64	256	1
ZID	16	1	16	1	16	16	16	1	16	1	64	64	64	1	16	16	256	1	256	1
ZJX	16	64	16	1	64	16	1	16	1	64	64	256	1	16	64	16	256	16	1024	1
ZKC	1	1	64	16	1	1	16	1	64	1	16	16	64	1	1	64	64	16	64	16
ZLA	1	64	256	256	1	16	16	64	64	16	1	1	64	64	16	256	1	64	16	64
ZLC	16	16	64	64	1	64	64	64	256	16	1	1	256	64	1	64	1	16	1	256
ZMA	16	256	64	16	64	16	1	64	1	64	64	256	1	16	256	64	256	64	1024	16
ZME	16	16	64	16	16	1	1	1	16	1	64	64	16	1	16	64	256	16	256	1
ZMP	16	1	16	16	1	16	64	16	64	1	16	1	256	16	1	16	16	1	16	64
ZNY	256	16	1	1	64	256	64	16	16	64	256	64	64	64	16	1	256	1	256	16
ZOA	16	64	256	256	16	64	64	256	256	64	1	1	256	256	16	256	1	64	1	256
ZOB	64	1	1	1	16	64	64	1	16	16	64	16	64	16	1	1	64	1	64	16
ZSE	64	64	256	256	16	256	256	256	1024	64	16	1	1024	256	16	256	1	64	1	1024
ZTL	16	16	16	1	64	16	1	1	1	16	64	256	16	1	64	16	256	16	1024	1

6.4 Description of Optimization Model

The warehouse-location formulation of clusterization problem is shown in Figure

41.

6.4.1 Indices and parameters

Indices 'i' and 'k' are for ARTCCs. Each of them ranges from 1 to 20.

The index 'f' is for flights.

N is the total number of flights.

Index 'h' is for *hops*(centers) along the trajectory of the aircraft. For example: If a flight travels from center 20(ZTL) to center 4(ZDC). Then *hop* 1 and 2 for that aircraft is 20 and 4 respectively

$$\text{Min } Z = [(\Omega) + (K * R) + \theta_1 * \sum_k \sum_f m_{f,k} + \theta_2 * \sum_k D_k]$$

s.t.

$$(1) \sum_{i=1}^{20} y_i = K$$

$$(2) \sum_k x_{i,k} = 1, \quad \forall i$$

$$(3) x_{i,k} \leq y_k, \quad \forall i, k$$

$$(4) m_{f,k} \geq \sum_{h=1}^{\gamma_f} x_{i=p_{f,h},k} - \gamma_f + 1, \quad \forall f, k$$

$$(5) m_{f,k} \leq y_k, \quad \forall f, k$$

$$(6) I_k = \sum_f m_{f,k}, \quad \forall k$$

$$(7) I_k \leq \Omega, \quad \forall k$$

$$(8) R = N - \sum_k \sum_f m_{f,k}, \quad \forall k, f$$

$$(9) D_k = \sum_i (x_{i,k} * d_{i,k}), \quad \forall k$$

Figure 41: MILP Model for Clusterization of CONUS

γ_f is the number of *hops*(centers) on flight f 's path.

K is a parameter for the number of clusters desired. Its typical value is 2 or 3.

$p_{f,h}$ is the h^{th} center in the path of f^{th} flight.

6.4.2 Decision variables

$x_{i,k}$ is a binary decision variable which is 1 if ARTCC 'i' belongs to cluster whose warehouse is 'k'.

y_i is a binary decision variable which is 1 if center 'i' is a warehouse.

$m_{f,k}$ is a dependent binary decision variable which is 1 if flight 'f' is an intra-cluster flight for the cluster whose warehouse is 'k'.

D_k is the sum of "Exp distances" from the warehouse of all the ARTCCs that belong to the cluster whose warehouse is 'k'.

I_k is the total number of intra-cluster flights that belong to the cluster whose warehouse is 'k'.

'R' is the total number of inter-cluster flights (dependent variable).

6.4.3 Objective Function

The objective function shown in Equation 10, is a minimization of weighted sum of four terms. The Ω in *first term* on RHS represents the maximum value of intra-cluster traffic across all clusters. By minimizing Ω , it is made sure that the maximum intra-

cluster traffic among all clusters in minimized. This is needed to ensure that there can be maximum sharing of codes across clusters by using *minimum* number of codes.

The second term on RHS of the objective function is the product of K and R . R is the total number of inter-cluster flights, which is an output of the optimization. The term R is weighted with K because one intra-cluster flight (counted by Ω) is *likely* to be ' K ' times as expensive as an intra-cluster flight in terms of beacon-codes required. In other words, if a group of flights can be classified as intra-cluster flights instead of and inter-cluster flights by moving the cluster-boundary (without adversely affecting the objective function in any other way), then on an average the group of flight requires $1/K$ times the codes it would if it were inter-cluster flights. This is because post-clusterization on CONUS, all the codes (1 to 3348) are partitioned into 2 disjoint sets and each of the K cluster used the same set of codes for assignment to its intra-cluster flights.

The *third term* on RHS is an optimization construct which, in conjunction with constraint 4, ensures that the value of a given $m_{f,k}$ is 1 only if flight ' f ' is an intra-cluster flight for the cluster whose warehouse is ' k '.

The *fourth term* on RHS is the minimization of summation of D_k 's. This term ensures that the clusters resulting from this model are contiguous, i.e. all the ARTCCs belonging to a given cluster are connected.

The weights θ_1 and θ_2 for the third and fourth term respectively of the objective function ensures that each of the objectives is weighted accordingly. The value of θ_1 should be just enough to ensure that $m_{f,k}$ is 1 only when flight 'f' is intra-cluster for cluster 'k'. The value of θ_2 should be just enough to ensure that all the ARTCCs belonging to a given cluster are contiguous. And finally, of both θ_1 and θ_2 should be scaled in accordance with the values of the first and second term of the objective function. These values are dependent on the number of flights in the planning horizon. For 11,000 flights during a peak period analysis, the typical values of θ_1 and θ_2 that achieves the four objectives are 500 and 5000 respectively.

Equation 10: Objective Function of Clusterization Model

$$\text{Min } Z = [(\Omega) + (K * R) + \theta_1 * \sum_k \sum_f m_{f,k} + \theta_2 * \sum_k D_k]$$

6.4.4 Constraints

Constraint 1 ensures that the number of clusters formed as a result of the optimization is exactly K. This is ensured by making sure that there are K warehouses (y_i is 1 only if ARTCC i is a warehouse).

Equation 11: Number of Clusters restriction

$$(1) \sum_{i=1}^{20} y_i = K$$

Constraint 2 ensures that all ARTCCs belong to exactly one cluster. This ensures that all the clusters are mutually exclusive and collectively exhaustive, i.e. they don't have common ARTCCs and cover the entire CONUS.

Equation 12: Ensure Clusters are mutually exclusive and collectively exhaustive

$$(2) \sum_k x_{i,k} = 1, \quad \forall i$$

Constraint 3 ensures that an ARTCC i , can serve another ARTCC only if it is a warehouse, i.e. $y_i = 1$.

Equation 13: Serve from a "Facility" only if it is open.

$$(3) x_{i,k} \leq y_k, \quad \forall i, k$$

Constraint 4 is used to capture the flight type into the variable $m_{f,k}$, which is 1 if flight ' f ' is an intra-cluster flight for the cluster whose warehouse is ' k '.

Equation 14: Check whether flight is Intra-Cluster

$$(4) m_{f,k} \geq \sum_{h=1}^{\gamma_f} x_{i=p_{f,h},k} - \gamma_f + 1, \quad \forall f, k$$

Constraint 5 ensures that if an ARTCC is not a warehouse, then a flight f can't be an intra-cluster flight with that center i as its warehouse.

Equation 15: Prevent non-warehouse ARTCCs from being intra-cluster center for flights

$$(5) m_{f,k} \leq y_k, \quad \forall f, k$$

Constraint 6 counts the intra-cluster traffic for all the clusters. I_k is the total number of flights that are intra-cluster for cluster whose warehouse is k .

Equation 16: Count Intra-cluster Traffic

$$(6) I_k = \sum_f m_{f,k}, \quad \forall k$$

Constraint 7 ensures that all I_k 's is less than Ω . This ensures that Ω is equal to maximum of all I_k 's.

Equation 17: Set upper limit for Intra-Cluster Traffic

$$(7) I_k \leq \Omega, \quad \forall k$$

Constraint 8 counts the total number of inter-cluster flights by subtracting the number of intra-cluster flights from the total number of flights N .

Equation 18: Count Total Inter-Cluster Traffic

$$(8) R = N - \sum_k \sum_f m_{f,k}, \quad \forall k, f$$

Constraint 9 counts, for all clusters, D_k , the total "exp-distance" of all ARTCCs belonging to that cluster from its warehouse.

Equation 19: Get total "Distance" for each each cluster

$$(9) D_k = \sum_i (x_{i,k} * d_{i,k}), \quad \forall k$$

6.5 Results

6.5.1. Results of Clusterization

The clusterization optimization model is tested for five high traffic volume days of 2007. These days represent different seasonal traffic patterns in time-and-space. The days are: Jan 3, Apr 11, July 26, Nov 21 and Dec 19(See Figure 18). The traffic statistics for these days is summarized in Table 6. The ETMS 4D trajectories for these days is converted to Center-Routes using the algorithm described in Appendix A: 4DT-to-Center-Route Converter.

The number of variables and constraints in the clusterization optimization model increases with the number of flights. The fewer the number of flights, the lesser is the computational time. In order to achieve this reduction in number of flights, the input to the clusterization model is only the peak-period traffic, instead of using flights for the entire 24 hour period of each day as input. The hypothesis here is that the non-peak period traffic does not influence the output of clusterization as much as the peak-period. This hypothesis is later tested by comparing the variation in intra-cluster traffic by time of day across clusters (Figure 29: Code Usage by Time-Of-Day using STA Algorithm for 3rd Jan 2007Figure 29 to Figure 33).

A one hour peak-traffic period is selected by first identifying the quarter-hour with maximum active-flights for the day (for each of the 5 days). If 't' is the starting time of the peak traffic quarter, then a one hour-window around 't' is chosen, ranging from

(t-30) minutes to (t+30)minutes. All the “active” flights in this window are selected.

There are three types of flights in this peak-window:

- (i) Flights which start and end within the peak-window.
- (ii) Flights which take-off in the peak-window but arrive at their destination airport outside the peak-window.
- (iii) Flights which were already flying at the starting time of the peak-window and land during the window.

For all these flight, the entire trajectory is selected to be used as an input for the clusterization optimization model.

The result of clusterization with parameter for number of clusters, K, set to 2 and 3 are shown in Figure 42 and Figure 43 respectively. All the 5 days yield exactly identical clusters. This is indicative of the network structure of traffic in CONUS. For the 2 cluster case, the cluster boundary runs north-south to the west of ZAU, ZID, ZTL, ZJX and ZMA centers.

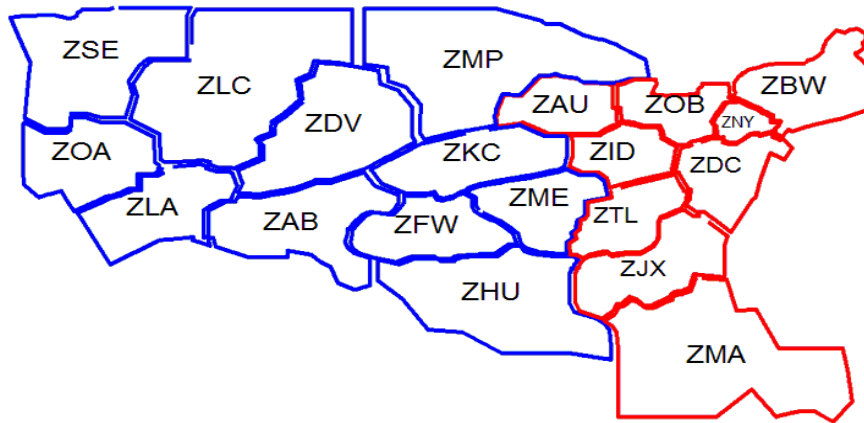


Figure 42: Cluster Boundaries for Three (K=2) Cluster Case

The number of intra-cluster and inter-cluster flights for the 5 days, for K set to 2 and 3 are shown in Table 14. These values are shown graphically in Figure 44 and Figure 45 respectively.

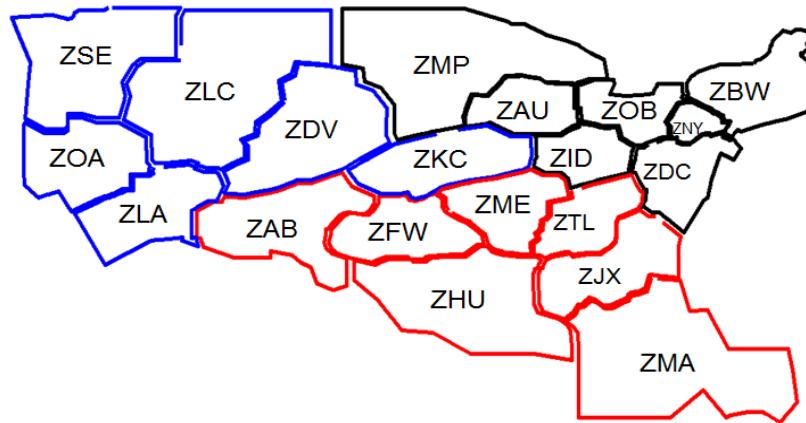


Figure 43: Cluster Boundaries for Three (K=3) Cluster Case

Table 14: The Total Number of Inter-Cluster and Intra-Cluster Flights with 3 Cluster CONUS

Day(2007)	Total Flights	2 Clusters		3 Clusters	
		Inter-Cluster Flights	Intra-Cluster Flights	Inter-Cluster Flights	Intra-Cluster Flights
3 rd Jan	43,649	8,822	34,827	14,921	28,728
11 th Apr	43,966	8,427	35,539	14,308	29,658
26 th July	48,721	9,856	38,865	15,407	33,314
21 st Nov	46,202	8,985	37,217	15,134	31,068
19 th Dec	47,145	9,450	37,695	15,512	31,633
	Mean % of Flights	20	80	33	67

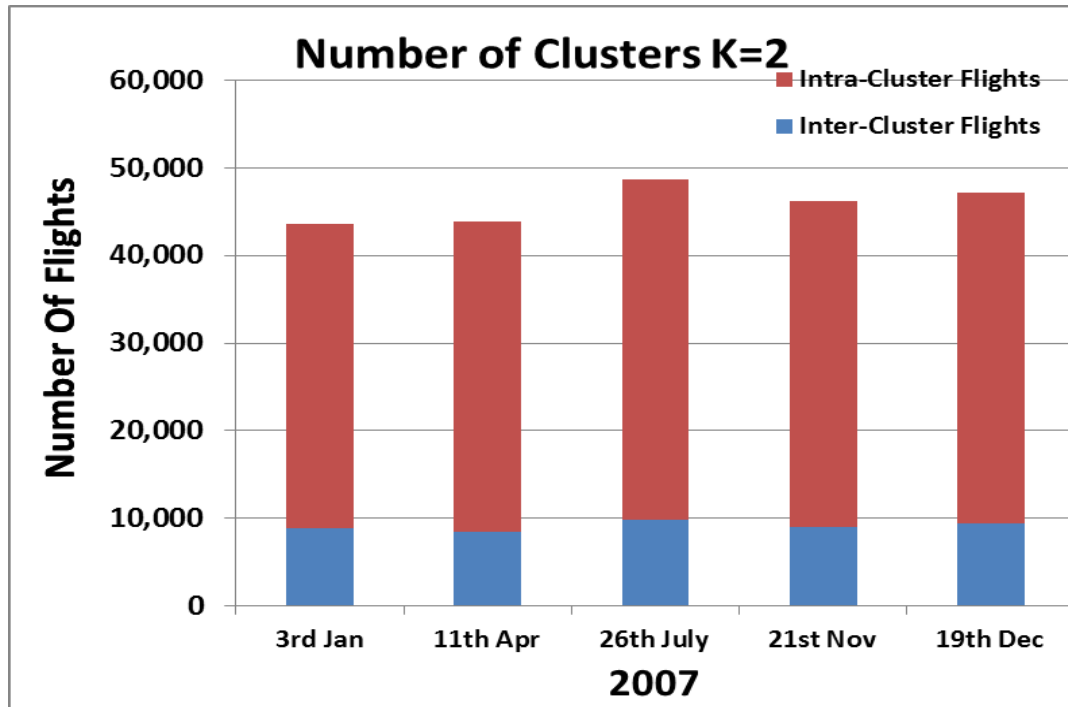


Figure 44: Total Number of Intra-Cluster and Inter-Cluster Flights for 2 Clusters of CONUS

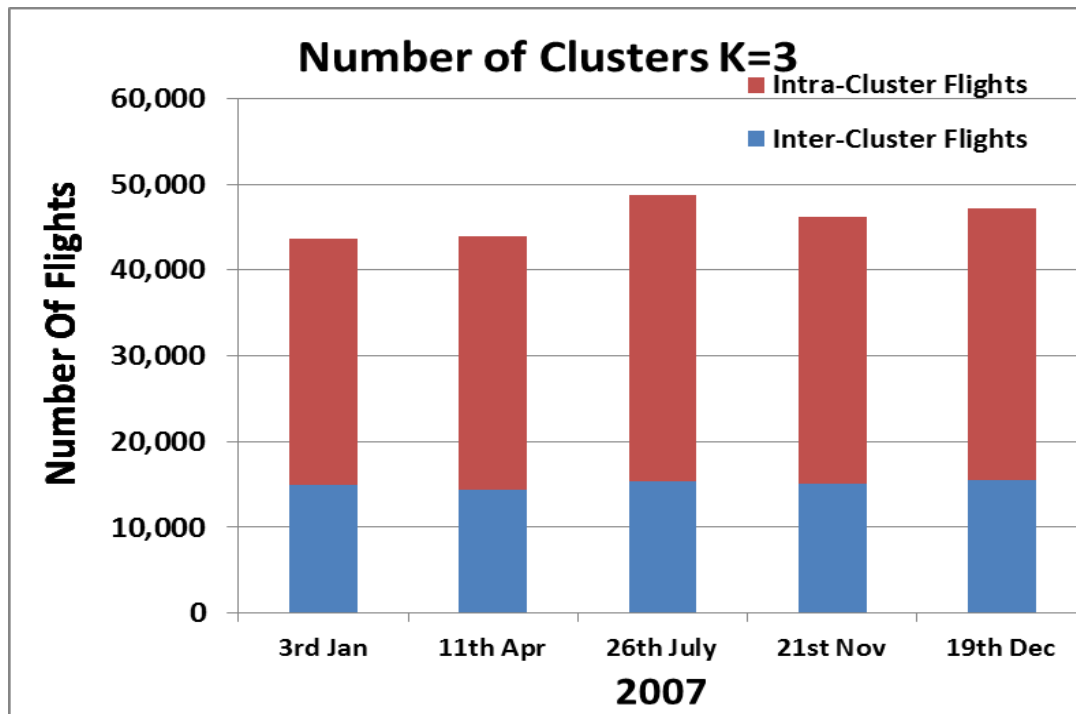


Figure 45: Total Number of Intra-Cluster and Inter-Cluster Flights for 3 Clusters of CONUS

6.5.2. Results of Code-Assignment

This section describes the result of code-assignment for flights given the cluster boundary definitions from clusterization of CONUS described in the previous section. Results are only shown for K=3 here because for K=2, the number of codes required exceeds the available number of codes.

The flights are assigned codes based on the type of flight, inter-cluster or intra-cluster. Section 6.5.2.1 and 6.5.2.2 describe results for code assignment to intra-cluster and inter-cluster flights respectively.

6.5.2.1 Intra-Cluster Flights Code-Assignment

Intra-Cluster flights can be assigned codes in the FCFS order or using STA algorithm described in previous chapter. The maximum number of codes required when assigning codes in FCFS order for intra-cluster flights is 1,300. Whereas, the maximum number of codes required when assigning codes to the intra-cluster flights using STA algorithm and for the worst case of 30 minutes uncertainty in host-prediction is 1,116 codes.

6.5.2.1.1 FCFS: Intra-Cluster Flights

The intra-cluster flights are assigned codes in the FCFS order from the bucket of the cluster they belong to. The number of codes required for code assignment to intra-cluster flights in FCFS order is shown in Table 15. The maximum intra-cluster traffic across all clusters for each day is shown in red. The maximum intra-cluster traffic for all 5 days occurs for Cluster 1 (Centers colored red in Figure 43) and its value is 2,143.

Table 15: Maximum Instantaneous Intra-Cluster "Active-Flights" Count/Number of Codes Required for FCFS Assignment. (Color of clusters corresponds to the coloring in Figure 43)

Day(2007)	Max Instantaneous Intra-Cluster Traffic			Max Intra Cluster Traffic/Codes Required
	Cluster 1 (Red)	Cluster 2 (Blue)	Cluster 3 (Black)	
3 rd Jan	1,809	1,142	1,526	1,809
11 th Apr	1,901	1,226	1,611	1,901
26 th July	1,806	1,319	2,072	2,072
21 st Nov	2,143	1,317	1,808	2,143
19 th Dec	1,999	1,261	1,780	1,999

When the intra-cluster flights are assigned codes in the FCFS order they do not need host-prediction of center crossing times. Only the schedule departure time and arrival time is needed to determine their period of code usage. The FCFS order of code assignment to intra-cluster flights also implies that the number of “active-flights” corresponds to the number of codes being used. As a result, the peak instantaneous count of intra-cluster traffic for a given cluster corresponds to the number of codes required by that cluster. The highest number of codes required for the intra-cluster flights for all these 5 days when assigning codes in FCFS order is 2,143 (Nov 21, Cluster 1).

6.5.2.1.2 STA Algorithm: Intra-Cluster Flights

The STA algorithm described in the previous chapter can also be used to assign codes to the intra-cluster flights. The algorithm uses space-time relationship of flight trajectories to assign codes.

However, when STA algorithm is used for code assignment, the host-prediction of center crossing times is needed. The number of codes required for assigning codes to the intra-cluster flights using STA algorithm for different values of host-prediction uncertainty is shown in Table 16.

Table 16: Maximum Number of Codes Required for Code assignment to Intra-Cluster flights using STA algorithm

Day(2007)	Maximum Number of Codes Required				
	Host-Prediction Uncertainty Buffer: 0 Minutes	15 Minutes	20 Minutes	25 Minutes	30 Minutes
3 rd Jan	627	774	817	870	908
11 th Apr	612	784	844	889	948
26 th July	686	911	966	1,054	1,113
21 st Nov	755	946	1,010	1,078	1,116
19 th Dec	636	822	881	935	983
Max of Column	755	946	1,010	1,078	1,116

The maximum number of codes required across all 5 days when there is *no* host-prediction uncertainty (0 minutes), is 755. However, when the uncertainty in host-prediction increases to 30 minutes, the number of codes required increases to 1,116.

6.5.2.2 Inter-Cluster Flights Code-Assignment

Inter-Cluster flights can be assigned codes in the FCFS order or using STA algorithm described in previous chapter. The maximum number of codes required when assigning codes in FCFS order is 2,655. Whereas, the maximum number of codes required when assigning codes to the intra-cluster flights using STA algorithm and for the worst case of 30 minutes uncertainty in host-prediction is 1,469 codes.

6.5.2.2.1 FCFS: Inter-Cluster Flights

The intra-cluster flights can be assigned codes in the FCFS order. In this case, the number of codes required is equal to the peak instantaneous inter-cluster traffic. The peak inter-cluster traffic for all the 5 days is shown in Table 17..

The minimum number of codes needed for assignment to inter-cluster flights if FCFS code assignment is followed is 2,655(21st Nov). This leaves only 693(3,348-2,655) codes for assignment to intra-cluster flights. This option of choosing FCFS assignment for inter-cluster flights is not feasible.

Table 17: Maximum Instantaneous Inter-Cluster "Active-Flights" Count/Number of Codes Required for FCFS Assignment

Day(2007)	Max Instantaneous Inter-Cluster Traffic/Number of Codes Required
3rd Jan	2,536
11th Apr	2,360
26th July	2,486
21st Nov	2,655
19th Dec	2,571

6.5.2.1.2 STA Algorithm: Inter-Cluster Flights

The number of codes needed if STA algorithm is used to assign codes to the Inter-cluster flights for different values of host-prediction uncertainty buffer values is

shown in Table 18. The maximum number of codes required across all 5 days when there is *no* host-prediction uncertainty (0 minutes), is 936. However, when the uncertainty in host-prediction increases to 30 minutes, the number of codes required increases to 1,469.

Table 18: Maximum Number of Codes Required for Code assignment to Inter-Cluster flights using STA algorithm

	Maximum Number of Codes Required				
Day(2007)	Host-Prediction Uncertainty Buffer: 0 Minutes	15 Minutes	20 Minutes	25 Minutes	30 Minutes
3 rd Jan	853	1,108	1,185	1,266	1,345
11 th Apr	839	1,106	1,168	1,259	1,327
26 th July	833	1,080	1,165	1,235	1,315
21 st Nov	936	1,207	1,295	1,379	1,469
19 th Dec	871	1,137	1,212	1,302	1,369
Max of Column	936	1,207	1,295	1,379	1,469

6.5.3 Summary of Results for Code-Assignment for all CONUS Flights

There are four possible combinations of code assignment strategy (FCFS or STA) and flight type (intra-cluster or intra-cluster).

Assigning codes using FCFS to inter-cluster flights is not feasible as it leaves insufficient codes for intra-cluster flights (See 6.5.2.2.1 FCFS: Inter-Cluster Flights). As a result there are two alternatives for code assignment as shown in Table 19. The values in these tables are obtained from the appropriate columns of Table 15, Table 16 and Table 18.

For example, the first value in the 3rd column of last row (755) is obtained from Table 16, which shows the number of codes required if STA algorithm is used for code-assignment for intra-cluster flights. The second value (936) is obtained from Table 18 which shown the numbers of codes required is STA algorithm is used for code-assignment inter-cluster flights.

Table 19: Options for Code-Assignment To Intra-Cluster and Inter-Cluster Flights for 3 Cluster CONUS

Alternatives for Code Assignment	Code Assignment Strategy (Intra/Inter)	0 Minutes Host-Prediction Uncertainty		30 Minutes Host-Prediction Uncertainty	
		Codes Required (Intra/Inter)	Max Total Codes Required (Feasibility)	Codes Required (Intra/Inter)	Max Total Codes Required (Feasibility)
1	FCFS/STA	2143/936	3,079	2143/1469	3,612
2	STA/STA	755/936	1,691	1116/1469	2,585

The 5th and 7th column of Table 19 shows the total number of codes required when using the corresponding code assignment strategy for host-prediction uncertainty of 0 and 30 minutes respectively. Based on these values, it is evident that if perfect information is available it is possible to use FCFS for intra-cluster flights. However, if the uncertainty in host-prediction time is closer to 30 minutes, then STA algorithm should be used to assign codes to both intra-cluster and inter-cluster flights.

The number of codes needed for both the code-assignment alternatives 1(FCFS/STA) and 2(STA/STA) for host-prediction uncertainty values of [0,15,20,25,30] minutes are shown in Table 20 and Table 21 respectively. If the number of codes required exceeds 3,348 the value is shown in red in the tables. If code assignment is done using alternative 1(FCFS/STA), then the number of codes required exceeds the available number of codes on 21st November for host-prediction uncertainty values of 15 minutes and more. Also, for 26th July and 19th December, the number of codes required exceeds the number of codes available for host-prediction uncertainty values of 30 minutes.

Table 20: Maximum number of Beacon Codes required With 3 Clusters (Intra-Cluster Flights assigned Codes in FCFS order and Inter-Cluster flights assigned codes through STA Algorithm) – Code-Assignment Alternative 1

		<u>FCFS:</u> Intra-Cluster Flights, <u>STAM:</u> Inter-Cluster Flights				
		Maximum Number of Codes Required				
Day(2007)	Total Flights	Host-Prediction Uncertainty Buffer: 0 Minutes	15 Minutes	20 Minutes	25 Minutes	30 Minutes
3-Jan	43,649	2,662	2,917	2,994	3,075	3,154
11-Apr	43,966	2,740	3,007	3,069	3,160	3,228
26-Jul	48,721	2,905	3,152	3,237	3,307	3,387
21-Nov	46,202	3,079	3,350	3,438	3,522	3,612
19-Dec	47,145	2,870	3,136	3,211	3,301	3,368

Table 21: Maximum number of Beacon Codes required with 3 Clusters (Code assignment through STA Algorithm for both Intra-Cluster and Inter-Cluster Flights) – Code-Assignment Alternative 2

		<u>STAM-</u> Intra-Cluster Flights, Inter-Cluster Flights				
		Maximum Number of Codes Required				
Day(2007)	Total Flights	Host-Prediction Uncertainty Buffer: 0 Minutes	15 Minutes	20 Minutes	25 Minutes	30 Minutes
3-Jan	43,649	1,480	1,882	2,002	2,136	2,253
11-Apr	43,966	1,451	1,890	2,012	2,148	2,275
26-Jul	48,721	1,519	1,991	2,131	2,289	2,428
21-Nov	46,202	1,691	2,153	2,305	2,457	2,585
19-Dec	47,145	1,507	1,959	2,093	2,237	2,352

Chapter 7: Conclusions and Future Work

This dissertation identified the problem of Beacon Code reassignment and potential code shortages in the existing Beacon Code assignment system. This research also helped develop an understanding of the problem through analysis of the spatial and temporal dynamics of the route structure of flights in the NAS. Based on this analysis, two feasible alternative methods for code assignment were developed.

The first method is a heuristic algorithm called STA (Space-Time Adjacency) which exploits the spatial and temporal opportunities available in the NAS to assign codes to flights such that code reassignments are minimized (Chapter 5). The STA algorithm is inspired from the individual flight code assignment MILP model which was deemed infeasible due to the computational issues (Chapter 4).

The second alternative developed is a hybrid method of code assignment using clusterization of the ARTCCs in the CONUS and the STA algorithm (Chapter 6). The method is based on “partitioning” the 20 ARTCCs (centers) in the CONUS into groups of ARTCCs called “clusters”. Based on the “cluster” definitions, flights are then grouped into intra-cluster or inter-cluster flights and are assigned codes using the STA heuristic algorithm.

The conclusions of the dissertation are presented in the form of answers to the three research questions (RQ¹ to RQ³) posed in section 2.5:

RQ¹: Is there a Beacon Code reassignment problem in the currently used Beacon Code allocation system?

RQ²: Is there a Universal Beacon Code assignment solution that eliminates the need for reassignments?

RQ³: Is there a centralized Beacon Code assignment solution that scales up to future traffic growth (X1.5 traffic)?

The first subsection of this chapter (7.1) summarizes the results of data analysis of historical data performed to quantify the magnitude of problems, namely, code reassignment and code shortages in the current system. This subsection provides an answer the first research question.

The methods of code assignment that should be used for current (2007) traffic scenario are presented in subsection 7.2. This subsection provides an answer the second research question.

The methods of code assignment that should be used when the volume of traffic increases to 1.5 times the current level of traffic are presented in 7.3. This subsection provides an answer the third research question.

Subsection 7.4 summarizes the insights that were gained during the course of this research. The final subsection 7.5 describes directions of future research.

7.1 Problem Magnitude

The analysis of two independent data sources, namely, ETMS data and HOST data establishes the fact that there is a 9.2% to 12.3% Beacon Code *reassignment problem* in the current system. The following is the summary of Beacon Code reassignment statistics obtained from analysis of HOST data and ETMS data:

- (i) *Code reassignment statistics from HOST data:* An analysis of the HOST data for 153 days of 2007 from 1st August 2007 to 31st December 2007 yielded the average number of daily Beacon Code reassignment instances to be 7,642 with a standard deviation of 1,451 (Figure 7). This is equivalent to an average code reassignment likelihood of 12.3% ($7,642/62,111$).
- (ii) *Code reassignment statistics from ETMS data:* The likelihood of code reassignment as derived from 5 days of ETMS 4-D trajectory data is in the range of 9.2% to 10.7% with an average of 9.96% and a standard deviation of 0.6% (Table 3).

Beacon Code shortage was not observed in any of the 153 days of the HOST data from 2007 that was analyzed. In fact, the highest code utilization in internal and external code category was 0.529 (ZHU) and 0.389 (ZLC) respectively.

7.2 Code assignment method for current traffic volume (2007)

Three alternative methods for code assignment were developed and evaluated in this dissertation. The individual flight code assignment MILP optimization model described in Chapter 4 was deemed infeasible due to computational issues. The other two methods, namely, the STA heuristic algorithm, and the hybrid method for code assignment using clusterization and STA, were able to achieve a 100% improvement over the existing system by eliminating the need for any code-reassignments.

The following is the summary of the result of code assignment for current traffic (2007) using STA heuristic algorithm and the hybrid method:

7.2.1 STA heuristic algorithm method of code assignment for current traffic volume

The STA algorithm facilitates assigning codes without any instances of code reassignment for current traffic volume. Based on the analysis of 5 high volume days with different seasonal space-time traffic patterns, the maximum number of codes required when assigning codes using STA algorithm is 2,362 (maximum codes available is 3,348). This occurs for the flights on 21st Nov, 2007 when a 30 minute host-prediction uncertainty is applied to all flights at every center boundary crossing.

7.2.2 Hybrid method (Clusterization and STA) of code assignment for current traffic

The hybrid method of clusterization and STA algorithm facilitates assigning codes without any instances of code reassignment for current traffic volume. In this method, the CONUS is partitioned into 'clusters' of ARTCCs. Given the number of codes available

(3,348), the optimum number of clusters for such a hybrid method of code assignment is three. Post-clusterization of CONUS and dividing the code into subsets, there are two code-assignment alternatives depending on which method is used for code assignment to inter-cluster and intra-cluster flights:

- (i) Intra-cluster flights (FCFS code assignment) and Inter-cluster flights (code assignment using STA heuristic algorithm)

When intra-cluster flights are assigned codes in the FCFS order, and the inter-cluster flights are assigned codes using STA heuristic algorithm, without any uncertainty in host-prediction, it is possible to assign codes for all the 5 days without exceeding the available number of codes (3,348). However, when the uncertainty in host-prediction is increased to 15 minutes or more, for 21st Nov the number of codes required exceeds the available number of codes. For 3rd Jan and 11th April 2007, this method works for all values ([0,15,20,25,30] minutes) of uncertainty in host-prediction. For 26th July and 19th Dec 2007, the method is able to assign codes for all values of uncertainty buffer less than 30 minutes (Table 20).

- (ii) STA algorithm is used to assign codes to both intra-cluster and inter-cluster flights

When the STA algorithm is used to assign codes to both intra-cluster and inter-cluster flights the total number of codes required for all the 5 days for all values of host-

prediction uncertainty is lesser than 3,348, the available number of codes (See Table 21).

7.3 Code assignment method for 1.5x Traffic (traffic volume projections of 2032)

The methods for code-assignment presented in the dissertation were also tested for traffic volume which is 1.5 times the current traffic volume (Table 9). Two days of 1.5x traffic scenario were used. Both these days are a representative of the FAA's Aviation Policy and Plan Office (APO) projection of traffic in the NAS in year 2032.

When the STA algorithm (Chapter 5: Space-Time Adjacency (STA) Algorithms) is used to assign codes to the flights on the 1.5x traffic days, codes were assigned with *no* code-reassignment instances when the host-prediction uncertainty in center-crossing time prediction is less than or equal to 20 minutes. When the uncertainty in host-prediction of center-crossing times is increased to 25 and 30 minutes then the number of codes required to achieve code-assignment without any reassignment exceeds the available number of codes. In this case, it is possible to assign codes using the STA-R algorithm (an extension of the STA algorithm that allows code reassignments) with a code-reassignment likelihood of less than 1.29% for both the days of 1.5x traffic data tested.

When the uncertainty in host-prediction is less than or equal to 20 minutes, then the STA algorithm eliminates the need for reassignment for 1.5x traffic scenario. For

host-prediction uncertainty in excess of 20 minutes, the code-reassignment likelihood is less than 1.29% for both the days of 1.5x traffic data that were tested. If the average likelihood of code reassignments from data analyzed from different sources is 9.96% (Table 3: Number of Hand-Offs and Reassignments (Source: ETMS Data) on an average, then assuming the best case, that the likelihood of code-reassignments remains the same even for 1.5x traffic, the STA algorithm shows a 92% reduction in code-reassignment likelihood.

7.4 Additional insights gained through this research

This section enlists some of the additional insights gained as a result of this research.

7.4.1 Boundary of feasibility

The methods of code assignment developed in this research are demonstrated to be feasible because the total number of codes required is less than the available number of codes (3,348). However, the optimality of any of these methods was not proven due to computational issues with the individual flight code assignment MILP model. An optimization model to determine the optimum number of codes required to allocate to a given set of flights was deemed non-scalable to practical traffic-levels (Chapter 4). However, the problem of code assignment to flights by ensuring no space-time adjacent flights are assigned the same code is analogous to a graph-coloring problem. The graph-coloring problem is a classic NP complete problem. However, there are some known approximation algorithms and heuristic methods which have been demonstrated to scale better than a MILP formulation. Using any of these heuristics it

may be possible to derive an optimum with an approximation ratio for the number of codes needed.

As shown in Section 4.3, when the individual flight code assignment MILP model was used to assign codes to a 683 flights data set, the number of codes required was 118. This represents the optimal allocation of codes to the flights using the minimum number of codes. In order to determine the loss of optimality when assigning codes using STA algorithm, the 683 flight dataset was assigned codes using STA algorithm. It needed 139 codes, i.e. 21 more than the optimal number of codes required.

The sub-optimality of the STA algorithm is due to the timing-out of codes assigned to flights until the end of each planning-window even though the flight may have landed towards the beginning of the planning window and “released” the code for reuse.

7.4.2 Traffic-pattern in NAS

In an effort to delineate the problem of code reassignment, the route structures of the current traffic in the NAS were analyzed for spatial and temporal patterns. An analysis of the flight routes revealed sparseness in the space-time overlap among the routes. More specifically, analysis of the trajectories of flights from ETMS data for 26th July, 2007, revealed that the sparseness is in the range of 81%-88%. This means that space-time adjacency among all the flights flying in the NAS occurred on only 12-19% (100 – sparseness %) of the time.

Table 22: Variation of number of intra-cluster and inter-cluster flights and the number of codes needed for different number of clusters for code assignment using Hybrid Cluster Method. Codes assigned using STA for both Intra-cluster and Inter-cluster flights. [Day: 26th July 2007, Host-Prediction Uncertainty: 0 minutes]

	Flights (% of Total 48,721 Flights)		Peak Count of Flights		Codes		
Number of Clusters	Intra- Cluster Flights	Inter- Cluster Flights	Intra- Cluster Flights	Inter- Cluster Flights	Intra Codes	Inter Codes	Total Codes Required
1	100	0	5302	0	1290	0	1290
2	79.8	20.2	1908	1623	971	688	1659
3	68.4	31.6	1159	2486	686	833	1519
4	59.8	40.2	994	2823	979	1125	2104
5	57.1	42.9	984	2917	764	1343	2107
..
20	29.3	70.7	106	4360	308	1450	1758

7.4.3 Effect of number of clusters of CONUS on the number of Beacon Codes required

In order to analyze the sensitivity of number of Beacon Codes required to the number of clusters, the Clusterization method described in Chapter 6 was used to clusterize the CONUS into different number of clusters, i.e. 2,3,4 and 5. The current day CONUS represents a 20 cluster case (as there are 20 ARTCCs in the CONUS). The result of clusterization on the distribution of intra-cluster and inter-cluster flights, the peak count of intra-cluster and inter-cluster flights and the number of codes required for the individual category of flights is shown in Table 22. July 26th was arbitrarily chosen

among the 5 available ETMS data days. The host-prediction uncertainty was set to 0 minutes, i.e. perfect information about ARTCC crossing time of flights was assumed to be available. Also, both intra-cluster and inter-cluster flights were assigned codes using the STA algorithm (Chapter 5). The results shown in Table 22 are also graphically represented in Figure 46.

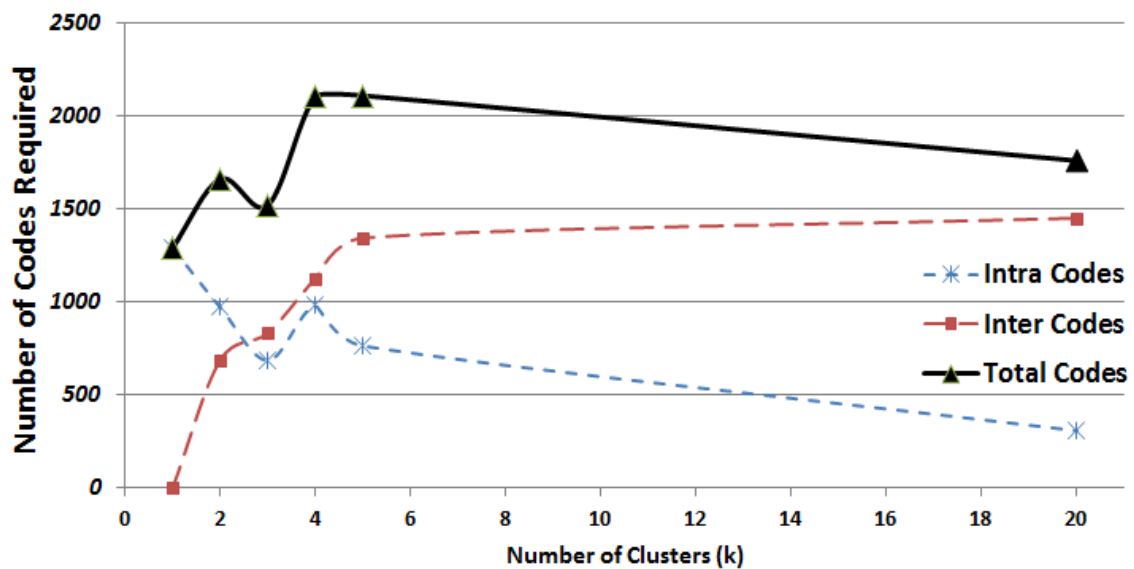


Figure 46: Variation of number of codes needed for different number of clusters for code assignment using Hybrid Cluster Method. Codes assigned using STA for both Intra-cluster and Inter-cluster flights. [Day: 26th July 2007, Host-Prediction Uncertainty: 0 minutes]

When the number of clusters is increased, the fraction of the total number of flights which are classified as intra-cluster flights decrease and the fraction of inter-cluster flights increase (column 2 and 3 of Table 22). When the entire CONUS is treated

as a single cluster, all the flights are intra-cluster flights, i.e. inter-cluster flights is 0% of the flight population. However, when number of clusters is 2, the percentage of inter-cluster flight increases to 20.2% and the percentage of flights which are intra-cluster is 78.8% (100-20.2). For a 20 cluster case, i.e. individual ARTCCs are representative of individual clusters, all the “external” flights are inter-cluster flights (70.7%) and all the “internal” flights are classified as intra-cluster flights (29.3%).

The inter-cluster and intra-cluster flights classified as a result of clusterization of the CONUS were assigned codes using the STA algorithm. The number of codes required by intra-cluster flights is the maximum number of codes required among all the clusters. The number of intra-cluster codes required is shown in Figure 46 with blue dotted line. The intra-cluster codes generally reduce as the number of clusters is increased because of decreasing number of intra-cluster flights. However, when the number of clusters increases from 3 to 4, there is an increase in the number of intra-cluster codes needed, 686 to 979. This shows that the decrease in the number of intra-cluster codes is not monotonic with increase in number of clusters.

The number of inter-cluster codes increases with the increase in number of clusters. This is shown by the red dotted line in Figure 46. For a single cluster CONUS, the number of inter-cluster codes required is 0, as there are no inter-cluster flights. For a 20 cluster CONUS, all the “external” flights are classified as inter-cluster flights and they require 1450 codes for allocation when using STA algorithm.

The black solid line represents the sum of inter-cluster and intra-cluster codes needed. The minimum number of codes required is when the number of clusters is 1. This is equivalent to allocating codes for all flights using STA algorithm. When the number of clusters is increased to 2 the number of codes required increases to 1,659. The three (3) cluster partition of CONUS represents a local minimum in terms of number of codes required, 1,519.

7.5 Future Work

This dissertation provides several opportunities for continued work.

7.5.1 STA with dynamic time-windows

The current STA algorithm is implemented with fixed time period planning windows. The default size of the planning-window is 60 minutes. Even though the size of the STA matrix is directly proportional to the length of the planning window it is in fact directly related to the number of flights. For a given duration of planning-window, the number of flights in peak-traffic may be higher than that in non-peak hours. As a result there is more variation in the processing time of STA for different time-windows because of the number of active flights. The current formulation of fixed planning-windows is more suitable for practical applications, where Beacon Code allocation would need to be done for fixed time duration rather than a fixed number of flights. However, from a theoretical viewpoint of doing sensitivity analysis for different traffic patterns and loads, it would be better to have a dynamic time-window implementation

of STA where the number of flights would be a parameter that controls the length of the planning window. For example, if the number of flights is fixed to 5,000, it may be equal to a 3 hour planning window in a non-peak period as opposed to only a 1 hour planning duration in a peak hour.

7.5.2 Refine code-reassignment in STA-R algorithm

In the current version of STA-R algorithm, the method of selecting a valid Beacon Code from a list of “allowed codes” is sub-optimal because it looks at the allowable codes in the current center and the next center on a flight’s route and picks a random code from the intersection of the “allowed codes” in those two centers and proceeds (5.3.2 STA-R Algorithm Details). This method works in practice because in the current NAS traffic structure, the average number of centers that a flight goes through is 2.3 (See 3.3 Beacon Code Reassignment Statistics). However, if due to changes in traffic structure of flights in the NAS, the average number of centers that a flight goes through increases, then the sub-optimality in the current implementation of selecting codes from “allowed list” would become more prominent. In such a scenario, this method may be modified using dynamic programming approaches in order to derive the optimal set of codes needed from a given set of “allowed code” list for a given flight that leads to minimum number of reassignments.

7.5.3 Space-Time Adjacency matrix data structure

The relationship between the size of the space-time adjacency matrix (as per the current data structure) used in the STA algorithm and the number of flight pairs 'n' is given by is $\frac{n*(n-1)}{2}$ (See 5.2.1.1 STA Matrix). By using better methods to exploit the sparseness in the composition of STA matrix, such as CSR (Compressed Sparse Row) or CSC (Compressed Sparse Column) method, the memory requirement can be reduced to a linear relationship of $(nnz+n+1)$ where "nnz" is the number of non-zero elements in the sparse matrix. Analysis of the trajectories of flights using ETMS data for 26th July, 2007 revealed that the sparseness is in the range of 81%-88%. By substituting a conservative value of "nnz" as $0.2*n$, i.e. assuming there is only 80% sparseness in the STA matrix, the relationship between the number of flight 'n' and the memory requirement of STA matrix using CSR algorithm becomes $(1.2n+1)$. This represents a significant memory saving which grows with the number of flights. For example, for a 5000 flight case, the number of elements required to be stored in the STA matrix in its current form is 12,497,500, whereas using CSR this number is reduced to 6,001.

7.5.4 Sensitivity analysis for NextGen CTOP (Collaborative Trajectory Options Program)

The capacity of airspace in NAS is affected by the presence of hazardous weather, which creates unstable regions, that forces the flights to deviate from their planned trajectories (Michalek, D., Balakrishnan, H., 2004). If the flight rerouting is such that the flight goes through new centers enroute to its destination, then it may be possible that the Beacon Code originally assigned to the flight may no longer be unique

in the centers on their modified trajectory. Conducting more experiments with STA and hybrid methods of code assignment by including the stochastic nature of flight routes will help identify the sensitivity of these methods to the variation in structure of traffic in the NAS. The stochastic nature of the flight routes can be modeled by using multiple flight plans for flights which are predicted to traverse a weather affected, capacity reduced airspace region and as a result have multiple possible (playbook) routes enroute to their destination.

7.5.5 Distributed Transaction Management

One of the primary challenges in implementation of any system modification is the ability of the users to adjust to the new protocols. Although the Universal Beacon Code assignment methods proposed in the system does not change any of the currently used procedures for both the pilots and the air traffic controllers, it does however lead to a shift in the responsibility of code assignment from the HCS of each ARTCC to a centralized authority. If required, the shift in code allocation from distributed to centralized authority can be masked by using the concept of Distributed Transaction Management. In essence, code allocation will be algorithmically determined centrally but they will still be allocated through the centers and the air traffic controllers of individual centers.

Appendix A: 4DT-to-Center-Route Converter

To test the algorithms for Universal Beacon Code Assignment for a given day in the National Airspace System (NAS), the 4-D trajectory data obtained from FAA's Enhanced Traffic Management System (ETMS) for each flight is converted to a time ordered sequence of ARTCCs that the flight goes through enroute to its destination along with the entry and exit time of each ARTCC along the flight's route. This section describes a set of algorithms that were developed to implement this conversion.

A.1 Latitude-longitude to Cartesian coordinate system conversion

Firstly, the latitude/longitude format of the ETMS data is converted to the Cartesian coordinate system using the equations listed below. This step is required because the point-in-polygon algorithms (A.2.2 Ray casting algorithm) operate in the Cartesian coordinate system. The formulae used for the conversion are as follows:

$$X = R * \cos(\text{longitude}) * \cos(\text{latitude})$$

$$Y = R * \sin(\text{longitude}) * \cos(\text{latitude})$$

, where $R = 6,378,100$ metres (Average radius of Earth).

This method of conversion is based on projection of earth on a flat surface. It must be noted here that due to the curvature of earth, such a projection does not conserve the scale of the map, i.e. the actual distance between any two points on earth may not be the same as the distance between those points on the projected map. However, for the purpose of this dissertation, we are only concerned about if and when the flights cross the ARTCC boundaries. As shown in Figure 47, the projection method being used here leads to a one-to-one mapping of each point on the earth's actual surface (curved) to the projected surface.

Similarly, the latitude/longitude boundary definition of each ARTCC is also transformed to the Cartesian coordinate system.

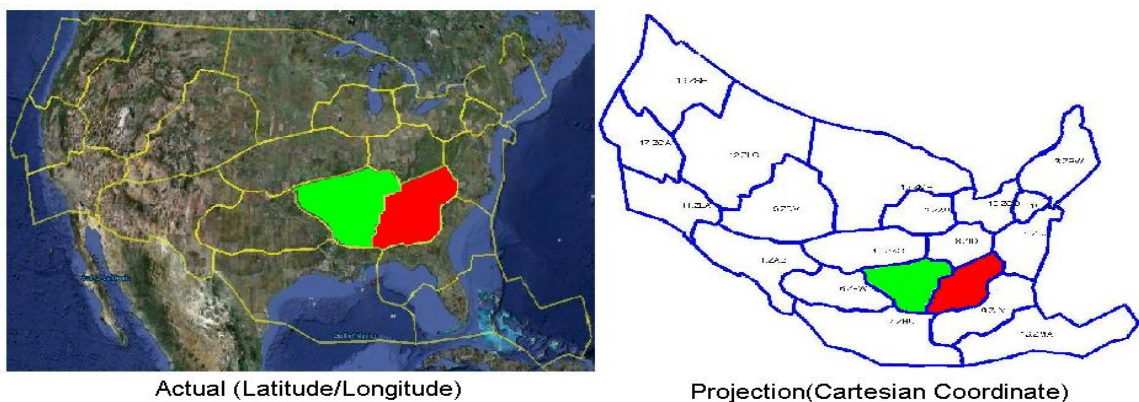


Figure 47: One-to-one mapping of Coordinate Transformation

A.2 4-D trajectory to ARTCC network mapping algorithm

After the flight trajectory data and the ARTCC boundary points have been transformed to the x-y Cartesian coordinate system using the formulae described in the previous subsection (A.1), the flight trajectories are then converted to a sequential list of ARTCC that each flight passes through. Also, ARTCC entry and exit time is determined. This mapping algorithm works as shown in Figure 48. In order to determine which ARTCC a radar hit for a particular flight belongs to, the “Ray casting algorithm” is used.

A.2.1 4-D trajectory to ARTCC network mapping algorithm details

The input to this algorithm is flight trajectories for individual flights, sorted by time and the boundary definition of the ARTCCs in Cartesian coordinate system.

Step 1: Extract next flight from the data

Extract the next flight from the dataset. Each flight’s data consists of a series of points along its trajectories stored in the form of 4-D data with x-y-z coordinates and time of radar hit. This data is presorted in ascending order with respect to time.

Step2: Initialize Variables

Initialize variables “prevCTR” and “path” to null. The variable prevCTR is used to store the “previous center” and “path” stores the entire path of the flight in terms of centers, i.e. the sequential list of ARTCCs that the flight traverses enroute to its destination.

Step 3: Extract next “radar hit” from the current flight’s dataset and run through Point-in-Polygon algorithm

The next point (radar hit) on the current flight’s trajectory is extracted. This point is run through the point-in-polygon method for detecting which ARTCC that point is in. The point-in-polygon method used is “ray casting algorithm” as described in A.2.2 Ray casting algorithm. Let “currCTR” denote the result of the “ray casting algorithm”, i.e. the ARTCC where the current point is.

Step 4: Append the current ARTCC to flight path

Check if currCTR=prevCTR. If true, it means that the flight is still in the same center as it was in the previous iteration. If so, go to Step 5. Otherwise, append “currCTR” to the “path” array.

Step 5: Check if current flight has anymore radar hits

Check if there are any more points (radar hits) for the current flight. If so, go to Step 3. Otherwise, go to Step 1. Stop when all the flights are exhausted.

After each track is run through the algorithm the resulting processed data has, for each track, the chronological sequence of ARTCC(s) the flight traveled through, and the boundary crossing times. A sample output is shown in Figure 50.

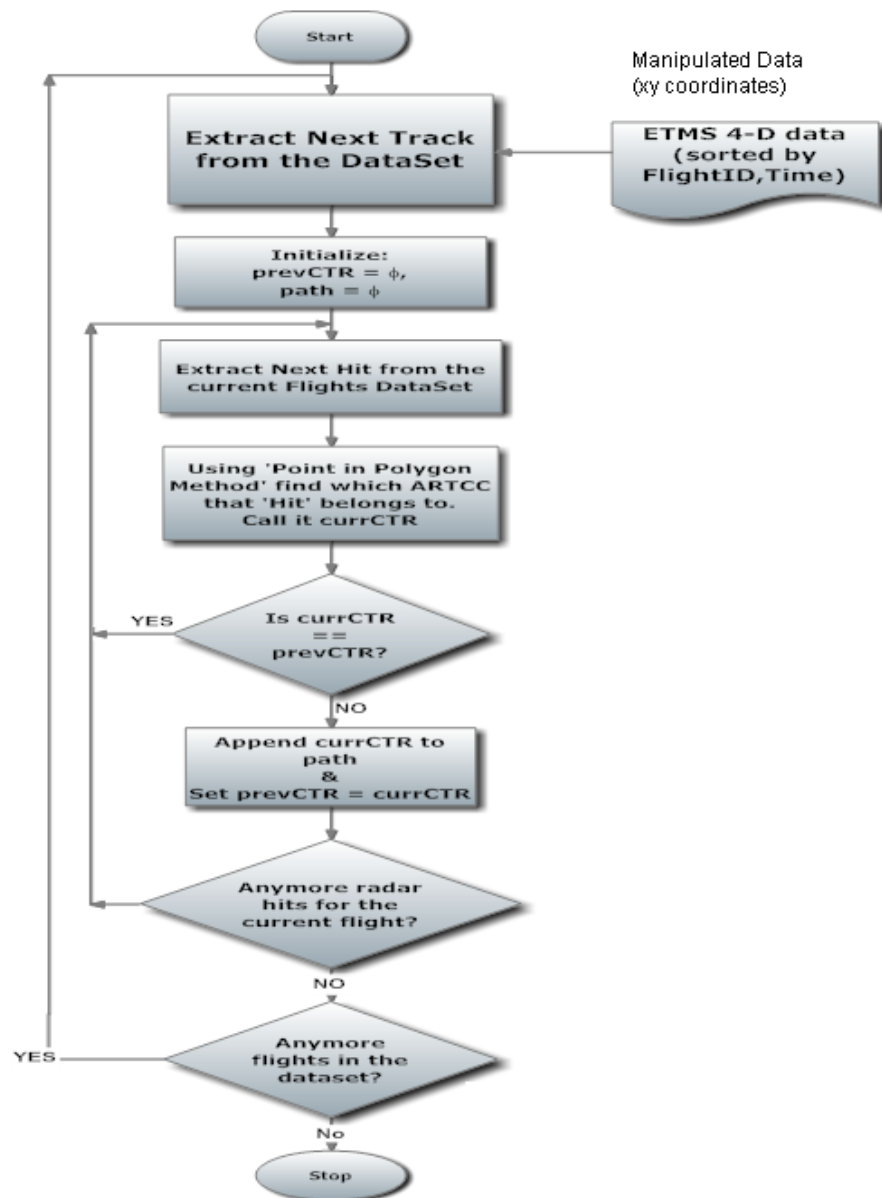


Figure 48: 4-D trajectory to ARTCC Network Path Conversion Process

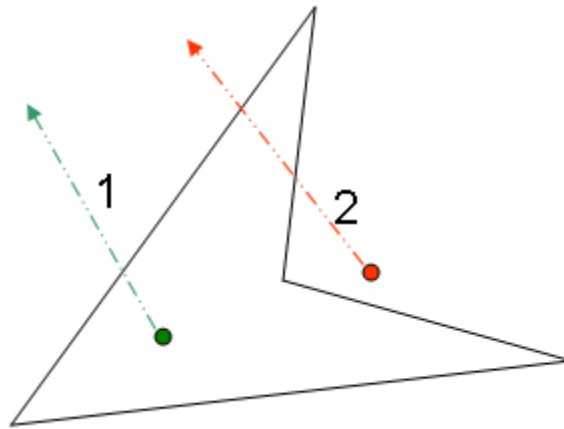


Figure 49: Ray-Casting Algorithm for Point-in-Polygon determination

A.2.2 Ray casting algorithm

To check whether any point is inside or outside an ARTCC, the number of times (say 'n') a *ray* starting from the point intersects the edges of the polygon is counted. If the point in question is not on the boundary of the polygon and 'n' is even, then the point is outside, otherwise it is inside the polygon. This concept is demonstrated in Figure 49. The ray from the green point intersects the polygon just once (odd), whereas the ray from the red point intersects the polygon twice (even). As a result it is inferred the *green* and *red* points are inside and outside the polygon respectively.

track(1)	
FID: 547244048	
centers: [6 14 8]	
BoundaryCrossingTime: [2 4 7 18]	
<hr/>	
track(2)	
FID: 548514236	
centers: 5	
BoundaryCrossingTime: [5 7]	
<hr/>	
⋮	

Figure 50: Format of Center-crossing Data(output of 4DT-to-Center-Route Converter)

Appendix B: NBCAS (National Beacon Code Allocation Plan(NBCAP) Simulator)

In the current operations of National Airspace System (NAS), the flights are assigned codes by the individual ARTCCs according to the rules published in the NBCAP (National Beacon Code Assignment Plan) Order JO 7110.66D. The distribution of codes to all ARTCC and also the rules for code allocation have been described in section 2.2(Description of Existing Beacon Code Allocation Method). A simulator was built that implements all the rules and procedures for code allocation. This simulator is called NBCAS, (NBCAP Simulator) and it assigns codes to flights according to the flowchart shown in Figure 51.

B.1 NBCAS algorithm details

The input to this algorithm is a set of flights that need to be assigned codes sorted with respect to their departure time in ascending order. Also, the code buffers, i.e. DSPI, CRDT and TB (section 2.2) are already added to the flight plans. The code assignment is done according to the following rules:

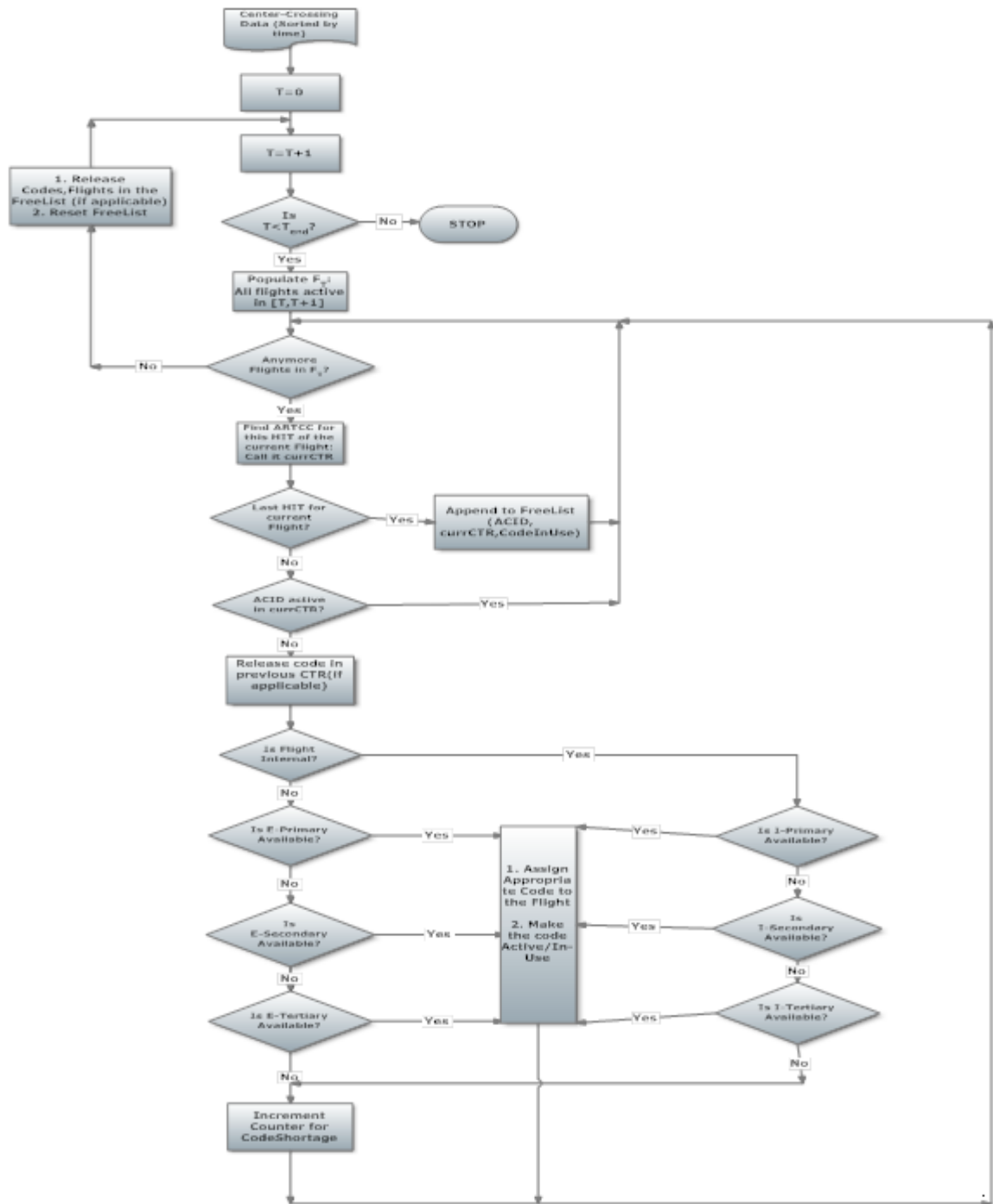


Figure 51: Beacon Code Allocation Flow-Chart

For every flight active in the NAS (active = requiring Beacon Code) during the current time period, it is first checked whether the current ARTCC for that flight is the same as the ARTCC that flight was in during the previous time step, in other words, whether or not the flight has crossed an ARTCC boundary or if it has just become active. In either case it needs to be assigned a code. If the flight is crossing into another ARTCC then the code that the flight is already assigned is checked for conflict against the flights already active in the following ARTCC. If there is no 'code conflict' the flight just keeps its old code. Otherwise, a new code is to be assigned to this flight from the code bucket of the entering ARTCC. The code assignment is done in the order primary, secondary and tertiary. The primary and secondary list is searched in cyclic order and the tertiary codes, if needed, are searched in top-down order. It must be noted here that flights which are active in the current time step but would disappear (land and no longer need a code) in the next time step are recorded in an array which is checked after each time period to release the codes corresponding to such flights at the end of current time step.

References

References

- Bussolari, S. (2000) Surveillance Tools for Distributed Air/Ground Traffic Management.
- Cook, S. A. (1971) "The complexity of theorem-proving procedures" in Proceedings of the third annual ACM symposium on Theory of computing. Shaker Heights, Ohio, United States, ACM, 151-158.
- DOT/FAA (2009) DOT/FAA Order 7110.66D, National Beacon Code Allocation Plan.
- Elbourn, R. D. and Saunders, P. B. (1972) Simulation of Air Traffic Control Radar Beacon Code Assignment Plan 3dl. [online]
<http://www.dtic.mil/srch/doc?collection=t3&id=AD0751998> (Accessed May 1, 2011).
- FAA, W. J. H. T. C. (2007) NAS-MDs, Computer Program Specification HCS – A5f1.4.
- FAA/APO (2005) FAA LONG-RANGE AEROSPACE FORECASTS FISCAL YEARS 2020, 2025 and 2030.
- Feldman, E., Lehrer, F. A., and Ray, T. L. (1966) Warehouse Location Under Continuous Economies of Scale. *Management Science*, 12(9), 670-684.
- Introduction to Algorithms* (2009) Cambridge, Mass, MIT Press.
- Joint Planning and Development Office (2004) Next Generation Air Transportation System Integrated Plan.
- Koncal, R. D. and Salkin, H. M. (1973) Set Covering by an All Integer Algorithm: Computational Experience. *J. ACM*, 20(2), 189-193.
- Lucic, P. (2005) *Geographic Beacon Code Allocation Optimization*, CSSI Inc.
- Lukes, J. A. (1975) Combinatorial Solution to the Partitioning of General Graphs.

Michalek, D., Balakrishnan, H. (2004) BUILDING A STOCHASTIC TERMINAL AIRSPACE CAPACITYFORECAST FROM CONVECTIVE WEATHER FORECASTS.

NASA (2010) FACET: Future Air traffic management Concepts Evaluation Tool.

Nolan, M. (2007) Fundamentals of Air Traffic Control.

Soundararajan, P. and Sarkar, S. (2003) An in-depth study of graph partitioning measures for perceptual organization. Pattern Analysis and Machine Intelligence, IEEE Transactions on, 25(6), 642-660.

Wolfson, M., et al. (2004) TACTICAL 0-2 HOUR CONVECTIVE WEATHER FORECASTS FOR FAA.

Curriculum Vitae

Vivek Kumar earned his Bachelor of Technology degree in Information Technology from Indian Institute of Information Technology, Allahabad, India. He then earned his Master of Science degree in Operations Research from George Mason University (GMU), USA, in 2007. His academic experience includes work as a Research Instructor at the Center for Air Transportation Systems Research (CATSR) housed at GMU where he worked on various FAA and NASA sponsored research projects involving optimization algorithms, simulation and statistical analysis.