Increasing Service Visibility for Future, Softwarised Air Traffic Management Data Networks

Kyle John Sinclair White

Submitted in fulfilment of the requirements for the degree of

Doctor of Philosophy

School of Computing Science

College of Science and Engineering

University of Glasgow

March 2017

Abstract

Air Traffic Management (ATM) is at an exciting frontier. The volume of air traffic is reaching the safe limits of current infrastructure. Yet, demand for more air traffic continues. To meet capacity demands, ATM data networks are increasing in complexity with: greater infrastructure integration, higher availability and precision of services; and the introduction of unmanned systems. Official recommendations into previous disruptive outages have highlighted the need for operators to have richer monitoring capabilities and operational systems visibility, on-demand, in response to challenges. The work presented in this thesis, helps ATM operators better understand and increase visibility into the behaviour of their services and infrastructure, with the primary aim to inform decision-making to reduce service disruption. This is achieved by combining a container-based NFV framework with Software-Defined Networking (SDN). The application of SDN+NFV in this work allows lightweight, chain-able monitoring and anomaly detection functions to be deployed on-demand, and the appropriate (sub)set of network traffic routed through these virtual network functions to provide timely, context-specific information. This container-based function deployment architecture, allows for punctual in-network processing through the instantiation of custom functionality, at appropriate locations. When accidents do occur, such as the crash of a UAV, the lessons learnt should be integrated into future systems. For one such incident, the accident investigation identified a telemetry precursor an hour prior. The function deployment architecture allows operators to extend and adapt their network infrastructure, to incorporate the latest monitoring recommendations. Furthermore, this work has examined relationships in application-level information and network layer data representing individual examples of a wide range of generalisable cases including: between the cyber and physical components of surveillance data, the rate of change in telemetry to determine abnormal aircraft surface movements, and the emerging behaviour of network flooding. Each of these examples provide valuable context-specific benefits to operators and a generalised basis from which further tools can be developed to enhance their understanding of their networks.

Acknowledgements

I would like to thank my supervisor Dimitrios Pezaros for his invaluable advice, guidance, support and patience throughout the years. Thanks also to my secondary supervisors for their input and expertise: Chris Johnson and Marwan Fayed. My acknowledgements also to the Scottish Informatics and Computer Science Alliance (SICSA) who funded this work. I am grateful to have worked abroad on two separate occasions throughout this project. My gratitude to the St Andrew's Society for the State of New York for their scholarship funding, and the Carnegie Trust who administered this funding in Scotland. Thanks to Rahul Simha at The George Washington University for allowing me to work within his department and thanks to all my colleagues there, especially Pablo Frank Bolton and James Marshall for making it a brilliant experience. Many thanks to Sherry Borener and the team at the FAA for the support and opportunities. Many thanks also to Ewen Denney and the teams at NASA Ames and SGT Inc. for the opportunity to work with and learn from you. Thanks also to Matt Knudson and Ganesh Pai for your knowledge and discussions. Thanks to Alastair Sloan and the staff at NATS and other ANSPs for their insight and time.

Thanks are also due to my many fellow students who have been in the department over the years. For useful discussions and opportunities to explore ideas at length, I would like to thank Horatiu Bota, Ornela Dardha, Arnaud Prouzeau, Stefan Raue, Posco Tso and David White. Special thanks to my colleagues and officemates Richard Cziva and Simon Jouet, for their friendship and indispensable support.

Finally, I would like to thank my family and friends for their encouragement. In particular, enormous thanks to my parents for their support, advice and steely belief in education.

To my sister, who showed me the meaning of resilience.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Overview

Air Traffic Management (ATM) systems and the operational networks supporting these systems are going to encounter significant change in the near future. With greater and ever-increasing demand for air travel capacity, many ATM systems are increasingly deploying the latest technologies to help meet this demand. One example of this is wake vortex modelling as an aircraft lands. By better understanding the causes and effects of aircraft wake, aeroplanes can be scheduled to land more efficiently based on their wake characteristics. Today, a further key consideration for future ATM systems is the rapid emergence of and demand for Unmanned Aerial Vehicles (UAVs) and the need to manage these operations among the existing airspace stakeholders including: passenger transportation, freight, military operations and general aviation. For more economically developed countries, air travel is very safe. This leads to the secondary, mission critical business case of ATM operators, to manage operations with minimised delay or disruption, becoming the leading motivator for change.

More widely, many critical infrastructures are now facing an increasing number of adverse cybersecurity challenges which are threatening their resilience and ability to fulfil their critical operations. As systems' interconnectivity and interdependence increases with evermore networked devices and the rise of the Internet of Everything, critical infrastructures are fac-

ing fresh challenges which relative isolation used to ameliorate. The motivations for greater interconnectivity are clear: significant efficiency savings, systematic cooperation and information sharing. However, the trade-offs are also apparent with a range of challenges including malicious third parties, misconfigurations and large-scale incidents, having a greater ability to adversely impact core critical operations.

Global ATM systems have experienced numerous highly disruptive incidents and outages in recent years from a variety of causes, but often with common elements, such as multi-faceted consequences from an initial issue that escalated. Recent national government audits of ATM have stated that there is now an increased risk that their national ATM operators will not be able to adequately detect, contain, eradicate, or recover from incidents affecting air traffic control systems. These audits assert this because of the aforementioned contextual developments and lack of preparation, with respect to technical capabilities, awareness and scenario-based practice, in the face of these developments. This work looks to address the lacking technical capabilities. ATM safety systems and procedures are sophisticated enough to ensure that even in the event of severe disruption and outages, air traffic is safely managed. This is achieved at significantly reduced capacity using alternative, in some cases manual, methods and diverse systems. However, the impact of adverse incidents which lead to reduced capacity, clearly affects the mission critical business case for ATM services. This is leading ATM operators to increasingly seek innovative solutions to enhance their capabilities with respect to detecting, containing, eradicating and recovering from a wide range of service disrupting incidents.

Coupled with the need for innovation, airspace operations remain an under-researched field [1]. The lack of past research relative to other fields, is probably due to the previous ability to grow without significant complexity. Now with airspace becoming increasingly busy, dense and complex, and the subsequent increases in complexity in the supporting infrastructure, the need for more research within ATM to achieve greater improvements is clear. Further research is required into many areas of ATM including security, cybersecurity, infrastructure networks, information sharing, resilience, safety, separation controls, scheduling, integration

with unmanned systems, automation and energy efficiency [1–4].

It is within this broader context where this work will examine the advantages in exploiting the latest state-of-the-art future networking technologies, including Software Defined Networks (SDN) and Network Function Virtualisation (NFV). Through these technologies, future ATM data networks can be softwarised, allowing for programmable routing of appropriate traffic on-demand, as well as virtualisation of relevant network functions such as monitoring and anomaly detection. By combining these technologies through this work, operators can better visualise and understand their infrastructure and services on-demand through deployable network functions for routed (sub)sets of application traffic matrices. NFV and SDN are innovations driven by demand from large networked systems operators and suppliers. The functionality desired, and problems currently faced, by a wide-range of large-scale networks are similar to those faced by ATM data networks. The latest best practices offered through the application of these modern paradigms solve many aspects of the current and emerging near-future needs of ATM data networks. Through this work, these technologies can also facilitate the improvement of ATM operators' understanding of their infrastructure, visibility and therefore resilience.

1.2 Thesis Statement

The primary ambition of this work is to determine whether, through the softwarisation of future ATM data networks, operators are granted a richer insight and therefore gain a better understanding of their infrastructure and services. Implicitly, from this improved understanding, the intention is to make ATM systems more resilient, through more informed decision-making by operators, especially in the face of challenges or disruption to services. This hypothesis will be tested by the development and evaluation of a virtual network function deployment architecture, applying NFV and SDN technologies. The feasibility of softwarisation for ATM data networks will need to be shown. Example cases from which a greater understanding can be achieved through such softwarisation will also be examined.

1.3 Contributions

The contributions of this work are as follows:

- The design and implementation of a softwarised function deployment architecture which enables on-demand virtual network functions to be deployed with automatic OpenFlow enabled routing. The system also allows operators to configure NFs and view reported data through graphs and notifications in the user interface.
- An analysis of UAV telemetry and the definition of normal operating behaviour for specific telemetry relationships which previously have been found to be leading indicators of imminent adversity.
- An analysis of Secondary Surveillance Radar (SSR) data, and the correlations and patterns within this data which provides a tightly coupled definition of normal operating behaviour among the cyber and physical components of the ATM surveillance service.
- An analysis of surface based radar data for taxiing airport movements and the implementation of an algorithm to detect deviations of rapid change, indicative of potential disruptive events.
- The development of monitoring and anomaly detection NFs based on each of the above analyses providing ATM operators a suite of deployable tools which can be used in the aforementioned function deployment architecture on-demand in order to increase their operational awareness.

1.4 Thesis Outline

The remainder of this dissertation is structured as follows:

Chapter 2 discusses the latest developments in relevant ATM and network communication fields, reviewing a brief history and evolution of ATM systems, the guiding principles

for today's infrastructure and the forecasts and demands on future ATM. This review continues, examining the challenges and opportunities arising in meeting these future demands based on current system shortfalls. The state-of-the-art networking paradigms are reviewed including virtualisation methodologies under NFV and the logical separation of control and data planes through SDN, alongwith summaries of challenges and opportunities in network monitoring, resilience best practices, middlebox architectures, and container based virtualisation.

Chapter 3 examines past major ATM disruptive incidents and the applicable key recommendations within the scope of this work. With the significant domain knowledge from recommendations, the design considerations for future UAV requirements are then explored. Based on these findings and the latest network softwarisation technologies, the design of the new function deployment architecture is presented and the benefits discussed, including for UAV systems, in the remainder of this chapter.

Chapter 4 provides the technical details of the implementation for the design and requirements discussed in Chapter 3. The chapter begins with early proof of concept analyses and continues to present the systematic diagrams for the overall architecture, traffic routing and discussions on each of the architectural components including the controller, NFV servers, APIs and User Interface. The chapter concludes with implementation details for sample ATM network functions, highlighting the capabilities of the system.

Chapter 5 applies the work and findings of the preceding chapters, by identifying the evaluating criteria before exploring network functions for a series of applicable contexts in order to evaluate this work with respect to the thesis statement in Chapter 1. The first context is data network flooding notification and detection based on new metrics available through the latest networking technologies. This is followed by an analysis of surface movement radar data and the development of a network function to detect rapid change in this telemetry. Further analyses follow with UAV telemetry relationship modelling and secondary surveillance

radar modelling providing the basis for two further network function evaluations. Finally, a means to assist operators with the prioritisation of traffic on-demand made available through this system is presented and evaluated.

Chapter 6 gives a summary of the contributions and findings of this work, and explores directions for future work.

Chapter 2

Related Work

2.1 Overview

Air Traffic Management data networks are becoming increasingly interconnected and less isolated. This trend is being observed across numerous critical infrastructures including transportation in general, road & rail, power grid systems, water & sewage works, financial sectors, chemical operations and healthcare. The goal of this work is to examine the latest and best practices in the field of computer networking, and to tailor and apply these paradigms to the context of ATM data networks with the desire for increased resilience through greater insight and faster means of recovery in the face of challenges. The state-of-the-art in networking research broadly considers general purpose networks such as the Internet and data centres. ATM data networks have some different properties to the generic Internet which allow for further opportunities to advance the resilience of such infrastructure using these latest techniques and architectures.

Isolation has had beneficial effects with respect to systems resilience. The incentives for infrastructures to become more connected and reduce their isolation include efficiency and greater scope for information analysis through deeper insight into their overall operations or how they fit into the larger societal system of systems. Despite the beneficial aspects of isolation, the ubiquitous societal trend of greater interconnectivity seems unlikely to halt, with

future advances such as the Internet of Things in the road maps of many global influencers. Coupled with this trend is an increase in cybersecurity challenges with 60,753 incidents in U.S. fiscal year 2013 reported by federal agencies to the United States Computer Emergency Readiness Team (U.S.-CERT), a dramatic increase from the 5,503 incidents reported in 2006 [5].

To understand the context for the contributions in this work, it is important to examine the latest work in related fields. With very little published research into ATM networked systems, it is necessary to understand the broader context of ATM systems, their evolution, their future requirements, augmented with the insights of published works that are available. This chapter is therefore arranged as follows:

Section 2.2.1 examines the evolution of global Air Traffic Management systems, coupled with the need for perpetual service, with a focus on the role of interconnecting data networks for, among other things, increased air traffic capacity. The future strategic directions and roadmaps of EU and U.S. Air Navigation Service Providers (ANSPs) and the shortcomings in existing ATM networked systems are also examined with respect to the context of this work.

Section 2.3 gives details of the best practices within general networking and the motivations for applying, implementing and tailoring these state-of-the-art best practices and paradigms to the ATM context. Related work on resilience, network monitoring, anomaly detection and container-based virtualisation is also presented in Section 2.3.

Finally, Section 2.4 summarises the key findings from this chapter.

2.2 Air Traffic Management

2.2.1 Evolution of the ATM Service

2.2.1.1 Inception of ATM

Like many critical infrastructures, Air Traffic Management has evolved in the face of a need to deliver a perpetual service. This has led to ATM systems around the world often having a reactive need for change: whether from external influences, through lessons learnt from tragic incidents or in response to stakeholder demand. It is widely recognised that early modern U.S. ATM itself as it is today, was born out of a society-driven political reaction to a fatal mid-air collision over the Grand Canyon on 30th June 1956 [6, 7]. Shortly after the accident investigation, the U.S. Congress decided to take action and in 1957, the *Airways Modernization Act* was signed by U.S. President Eisenhower, requiring aircraft to have flight data recorders. The Federal Aviation Administration (FAA), the national aviation authority of the United States of America, was founded a year later on 23rd August 1958.

In general, Air Traffic Management is risk averse, since its founding purpose is to help manage and plan the manoeuvre of aircraft safely and efficiently through controlled airspace. In a risk averse context, when a process is proven safe, change is perceived as risk. However, change is inevitable as external societal influences drive ever-ambitious requirements for Air Traffic Management systems. This has been true since the inception of such systems with both increased capacity and safety demands being prevalent throughout history.

From 1959 to 1969, the number of aircraft operations at FAA's airport Air Traffic Control (ATC) towers had increased by 112% nationwide [8]. In the mid 1960's the FAA began considering automation and modernising what was essentially a manually operated system by Air Traffic Control Officers (ATCO) using radio communications, radar, and generic computers. At this time, the U.S. National Airspace System (NAS) was a series of geographically distributed and relatively isolated Air Route Traffic Control Centers (ARTCC) which provided ATM services to their surrounding airspace. Air traffic at ARTCCs from 1959 to 1969

had also increased by 110.6% [8].

In the early 1960's in the UK, around 500,000 flights per year [9] were controlled by the National Air Traffic Control Services, which is now known as NATS. Founded in December 1962 NATS preceded the UK's Civil Aviation Authority (CAA), established ten years later in 1972. In the 1970's air traffic movements were increasing by 3.5% annually[1].

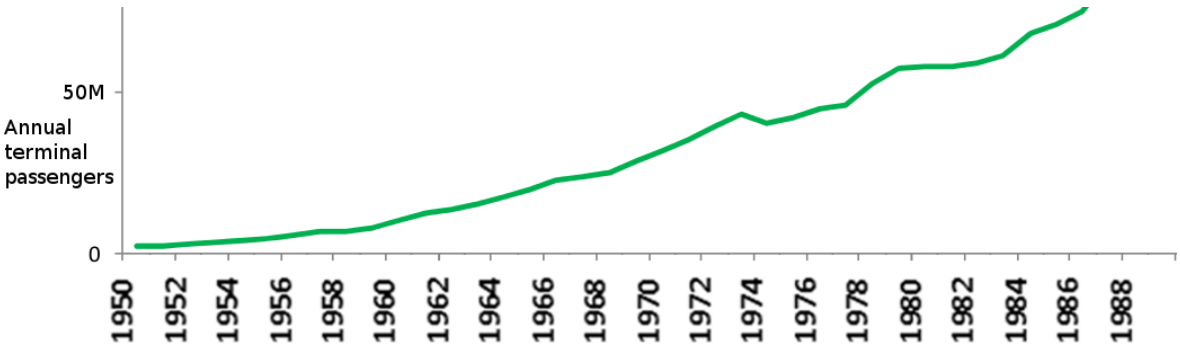


Figure 2.1: Terminal passengers at UK civil aerodromes [9]

Historical Air Traffic Statistics, Annual 1954-1980

	All Services					
	Revenue Passenger ton-miles (000)	Revenue Freightton-miles (000)	Overall Availableton-miles (000)	Revenue Passenger Originations (000)	* Revenue Passenger Originations (000)	Revenue Passenger Enplanements (000)
1954	U	U	U	U	U	35,447
1955	U	U	U	U	U	41,707
1956	U	U	U	U	U	46,004
1957	U	U	U	U	U	49,423
1958	3,168,649	676,532	7,416,772	49,165	U	53,071
1959	3,629,632	795,481	8,336,428	55,997	U	60,297
1960	3,850,303	820,778	5,024,119	57,873	U	62,258
1961	4,021,801	962,021	10,578,367	58,403	U	63,012

Figure 2.2: U.S. historical air traffic passenger growth[2], where *U* represents data unavailable

Figures 2.1 and 2.2, show the significant growth in air traffic, mirrored in the UK and in the U.S., from 1950 to 1980. There are many perspectives to explain the increase, however lower travel costs increased the affordability of flights and this was achieved in part through the U.S. *Airline Deregulation Act*, signed on 24th October 1978. The deregulation allowed

[1] `http://www.nats.aero/about-us/our-history/timeline/`; Accessed: March 2016

for the airline industry to become highly competitive. This rapid growth put increasing pressure on the ATM systems of the time with the 1961 U.S. report, *Project Beacon*, finding that weather delays were costing airlines around $70 million (USD) annually [10].

Beyond the UK and U.S., the global growth of air traffic can be inferred from the International Civil Aviation Organization (ICAO) data seen in Figure 2.3. This figure shows earlier predictions in 1999 of European air traffic increasing to more than double from 1999 by 2015 were realistic on a global scale [11].

2.2.1.2 ATM Today

Today, there are similar issues on a larger scale. Current ATM systems are forecasted to be unfit-for-purpose without significant enhancement to meet the predicted sharp rise in global demand [12, 13]. Air Traffic Management is now an international, co-operative geo-political set of services which provide the civilian air transportation industry with safe, reliable information and logistical organisation allowing for the efficient transfer and scheduling of flights, or cancellation, under all conditions. The common global structure is to divide airspace management into regions, with either federated or devolved control. Lewis [14] provides a good international comparison of the specific details of ATM delivery in six example countries. In general, management regions include different altitudes, airspace classifications, areas around airports or other restricted airspaces such as military bases or wildlife preserves.

The UK has approximately 350,000 square miles of airspace [1] which is divided into *Scottish* and *London* controlled airspace, shown in Figure 2.4, with Prestwick Area Control Centre (ACC) and Swanwick, Hampshire ACC managing each of these areas respectively. Subcontrol regions are also shown in Figure 2.4, such as Daventry Control Area. Figure 2.4 also highlights the flight corridors designated above the UK, which assist ATCOs with systematic management for high volume routes. The entire UK airspace is divided vertically

[2] `https://www.rita.dot.gov/bts/sites/rita.dot.gov.bts/files/subject_areas/airline_information/air_carrier_traffic_statistics/airtraffic/annual/1954_1980.html`; Accessed: March 2016

[3] `http://www.icao.int/sustainability/Pages/Facts-Figures_WorldEconomyData.aspx`; Accessed: March 2016

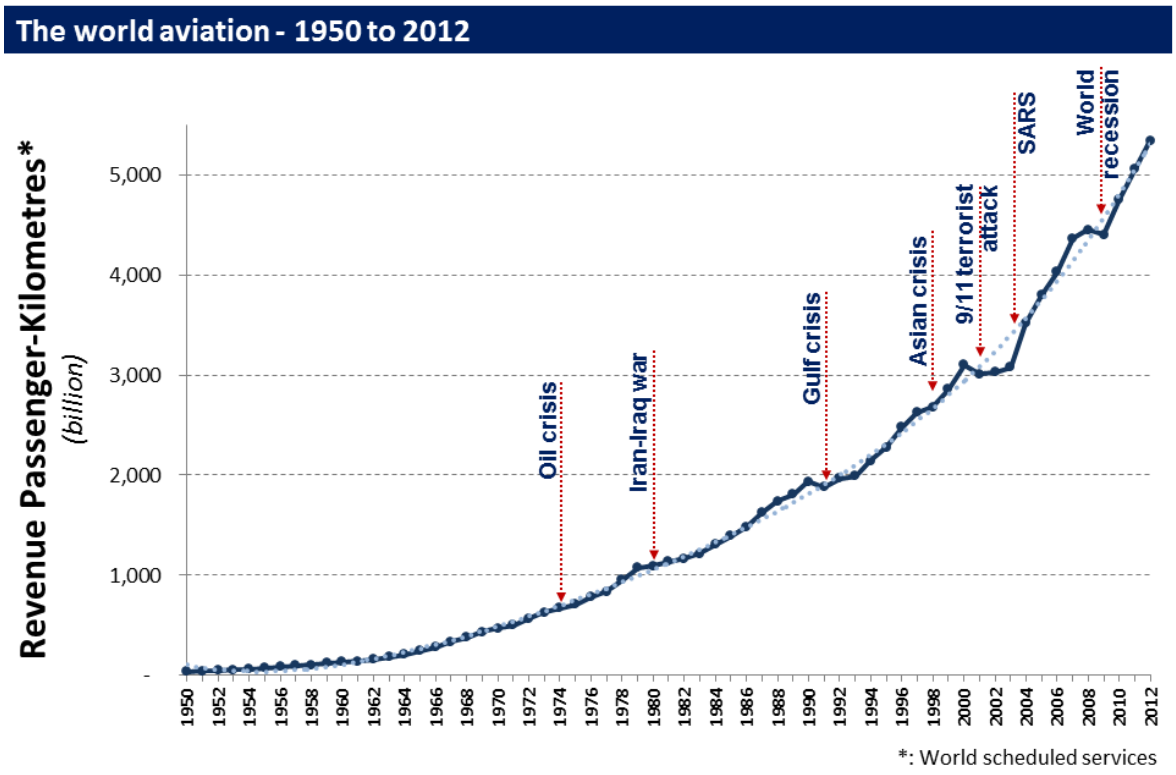


Figure 2.3: ICAO global growth of paying (revenue) passenger air traffic[3]

with a zone from the ground to 19,500ft and another for the airspace >19,500ft. Beyond these zones, there are different classifications of airspace and notions of controlled and uncontrolled airspace which each span different altitudes.

In the U.S., similar airspace segregation exists. Figure 2.5 shows the seven FAA airspace classes denoted A-G, where class A is the most controlled. Despite this level of management, infringements do occur even at current levels of air traffic. 629 infringements were reported to the CAA in 2015[4], while no collisions occurred, the cost in spent fuel and delays to many interconnected flights is considerable.

Figure 2.6 provides a systematic overview for U.S. ATM. There are 160 Terminal Radar Approach Control (TRACON) facilities which manage the airspace approximately 40 miles around an airport and above the airport tower controlled airspace. Air Route Traffic Control Centers (ARTCC) control manage the remaining geographical spread between airports, with

[4] `http://www.flyontrack.co.uk/statistics/`; Accessed: February 2016

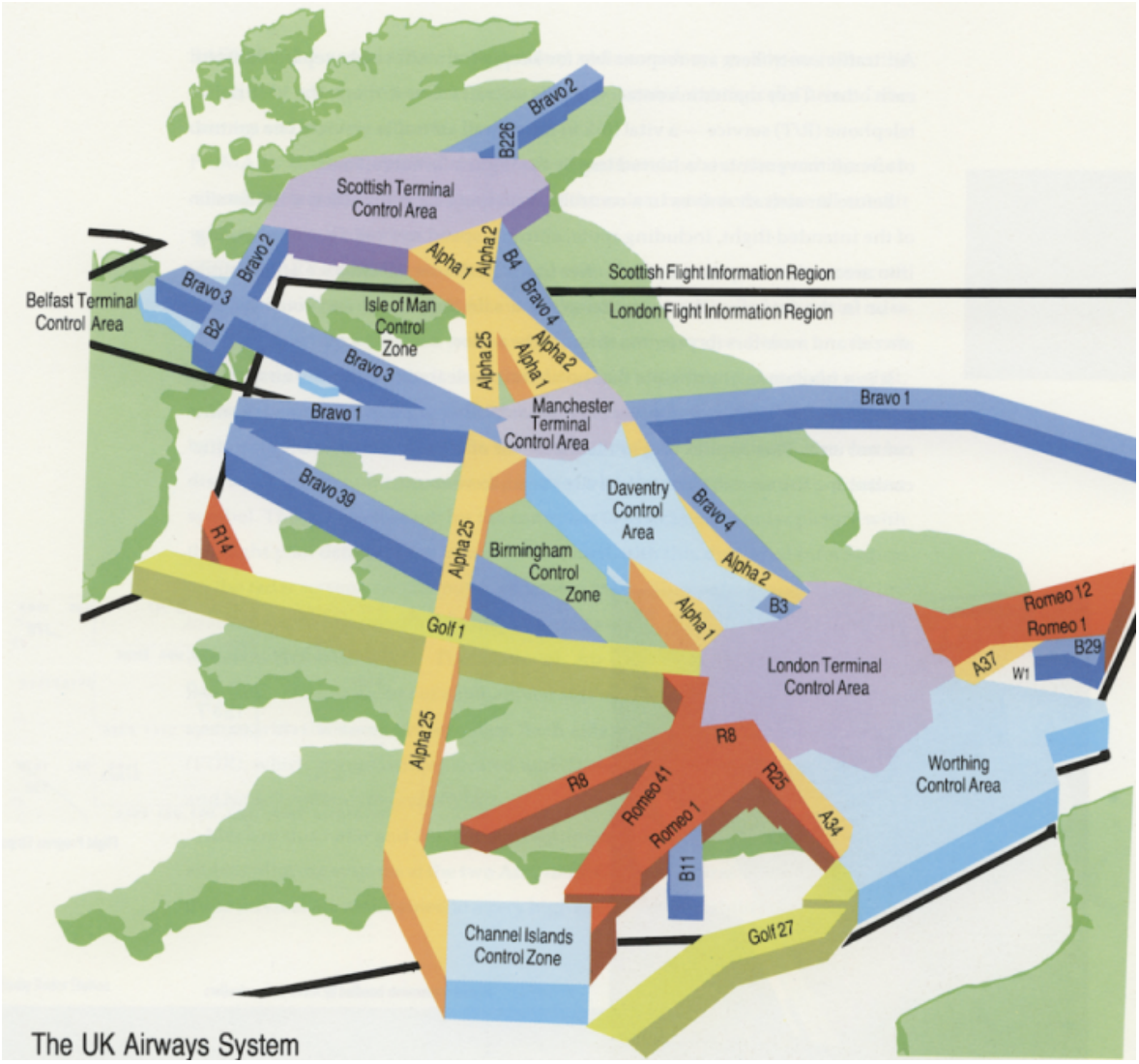


Figure 2.4: NATS Air Traffic Management flight corridors over the UK[5]

22 facilities in the U.S.. The centralised Air Traffic Control System Command Center manages the flow of air traffic nationally. When the system is under duress including from bad weather, equipment failures or closures the Command Center ATCOs look after the capacity across the NAS.

The NAS comprises 19,000 airports, ~600 ATM facilities, and ~65,000 other facilities, including radar and communications systems. Nearly 50,000 FAA staff and over 600,000 pilots manoeuvre ~230,000 aircraft with a current peak of 2,850 flights simultaneously within the NAS [16].

Services within the wider ATM system often include:

[5] `http://www.nats.aero`; Accessed: March 2016

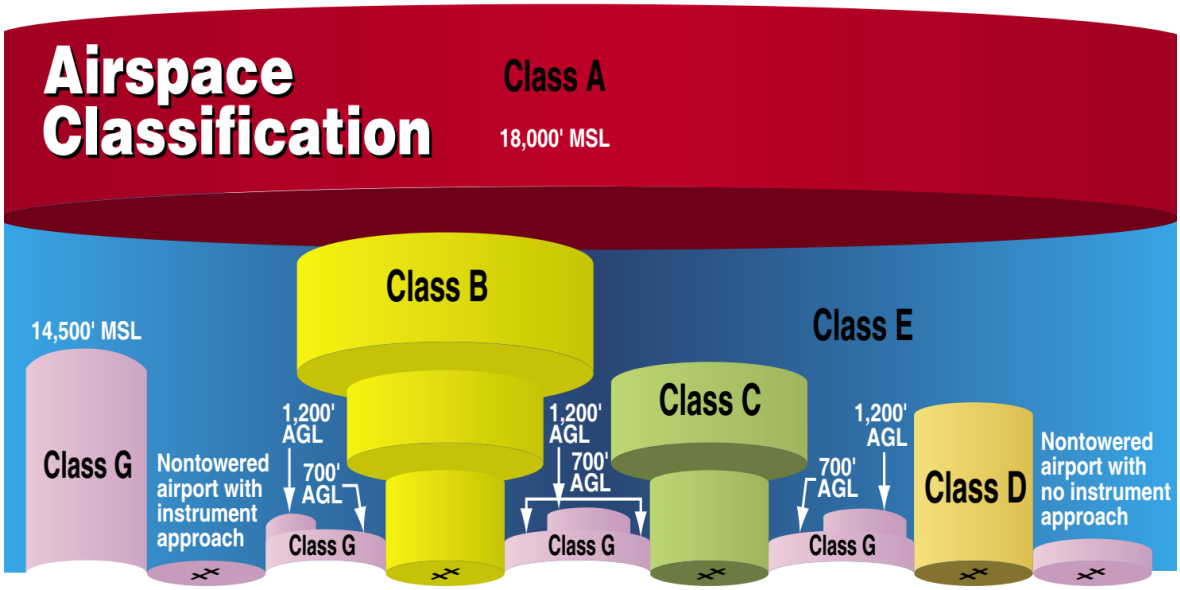


Figure 2.5: U.S. airspace classification (feet Above Ground Level (AGL) and Mean Sea Level (MSL))[15]

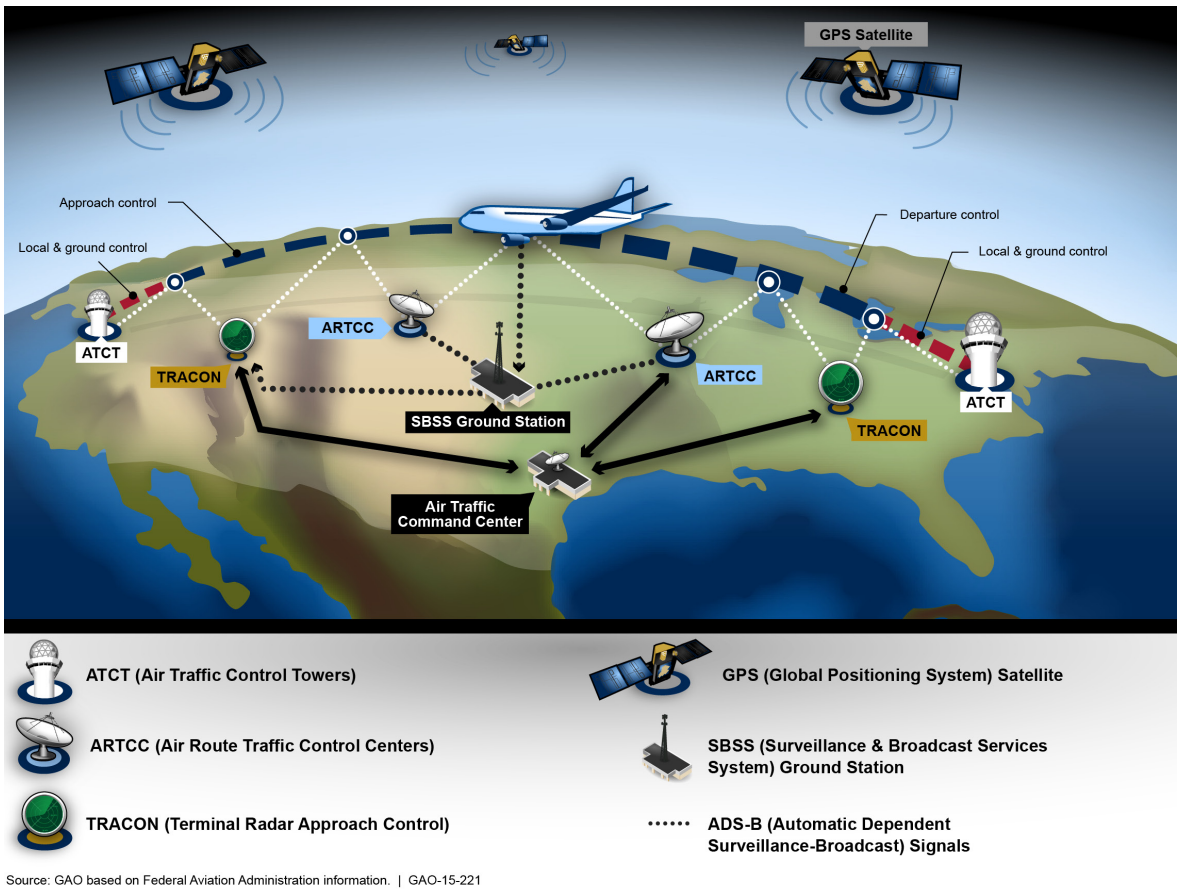


Figure 2.6: U.S. Air Traffic Management systematic overview [16]

- Primary Surveillance
- Secondary Surveillance
- Aeronautical Meteorology
- Aids to Navigation
- Flow and Capacity Management
- Terminal Surveillance

At a high level, these services can be categorised into achieving three functional goals: safe separation and routing of in-flight aircraft responding to current conditions, advanced planning and scheduling of flight routes and safe & legal movement of aircraft at lower altitudes and airports. Underlying all of these services are the physical network connections from data sources, to control centres and over backbone infrastructure between regional control centres. Figure 2.7 shows the U.S. system-wide FAA backbone; the FAA Telecommunications Infrastructure (FTI). The backbone is 10 Gbps on east coast and 2.5 Gbps on all other links. This is due to the relative density of ATM facilities on the east coast[6], compared with the more remote facilities across the rest of the country which still require high availability but typically significantly lower bandwidth, due to less air traffic volume and therefore fewer ATM facilities. The desired maximum end-to-end latency between any two nodes in the topology is $<$100 ms [17]. The FAA, similar to most ANSPs, does not own the long distance telecommunication infrastructure it uses. Instead third party vendors are procured to meet their requirements. In the UK NATS uses BT infrastructure for the backbone connecting their three major facilities, while Harris Corporation supplies the majority of the U.S. nationwide backbone.

[6] `https://www.faa.gov/air_traffic/technology/cinp/fti2/documents/media/fti_overview.pdf`; Accessed: February 2016

2.2.2 ATM Telecommunications Infrastructure: The FTI Example

2.2.2.1 Today's Infrastructure

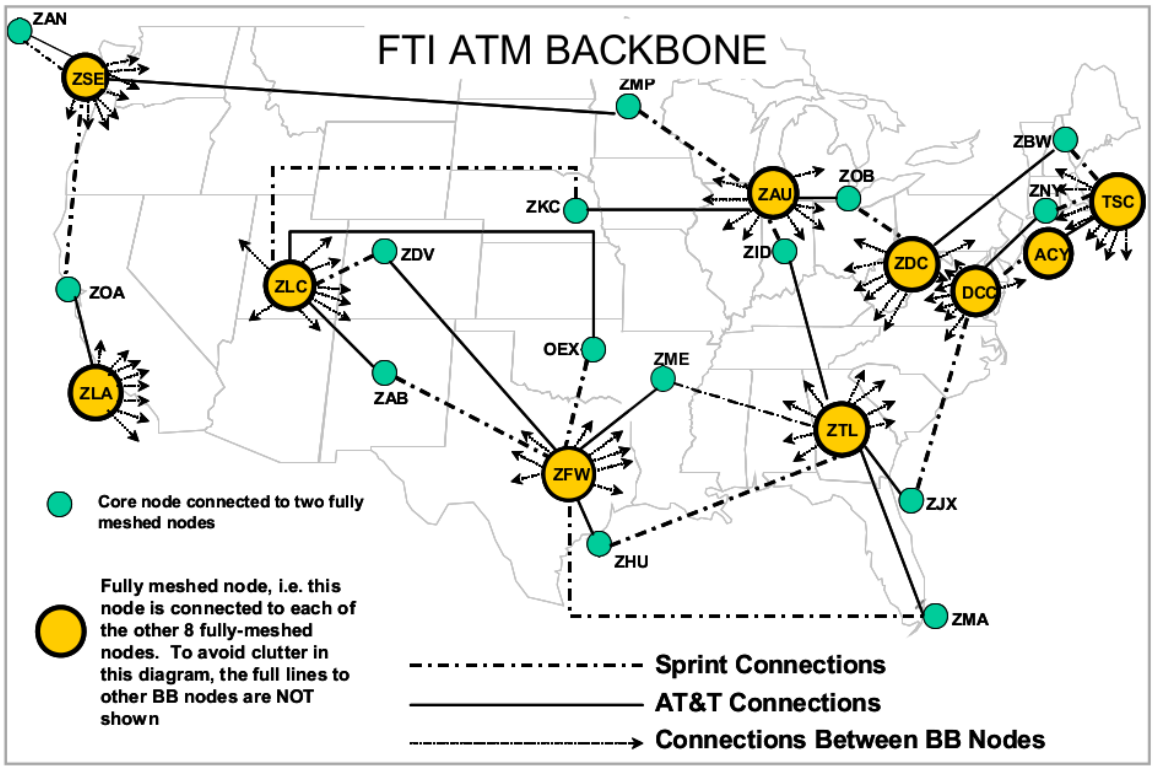


Figure 2.7: FAA Telecommunications Infrastructure fibre backbone topology [17]

To understand ATM telecommunications infrastructures worldwide, this section explores the example case of the U.S. FAA Telecommunications Infrastructure. The related work and environment reviewed is representative of other national service providers infrastructure or the aspirations for their infrastructure as air traffic increases in their regions.

The FTI comprises approximately 25,000 telecommunications services[7]. It consolidates the network infrastructure for around 5,000 ATM significant facilities across the U.S. NAS. The FTI is a complex critical infrastructure, however, it replaced a mesh of individual nation-wide networks which served different purposes and were separately managed, monitored and operated. Due to the criticality of NAS operations, the U.S. ATM requires protection from the improbable events that result in extended duration or wide-scale outages. Additional survivability is provided through a highly redundant optical backbone network with

[7] https://www.faa.gov/air_traffic/technology/cinp/fti2/; Accessed: March 2016

two independent routing domains for Operational Internet Protocol (OPIP) traffic.

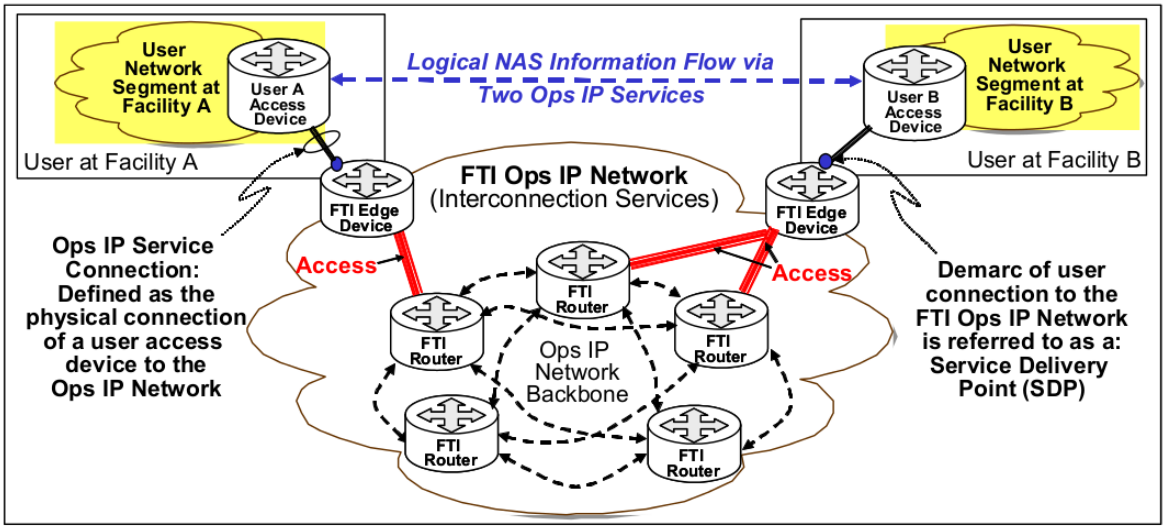


Figure 2.8: FAA OPIP abstracted network topology [18]

The FTI OPIP links Service Delivery Points (SDPs), where end user systems connect, to the Wide Area Network (WAN) for other user systems' consumption. Figure 2.8 shows the logical and abstracted topology of the FTI OPIP. At the FTI Edge Devices various security measures take place including firewalls and defence layers against Denial of Service attacks. Further connection information is documented in the publicly available, redacted, FTI User Guide [18]. OPIP services include many critical ATM processes such as voice communications. As a result there is a dual core requirement for the FTI backbone. A traffic replicator, such as seen in Figure 2.9, duplicates user packets at the ingress point and forwards the original and copied packets over the transmission and redundant isolated routing cores, as seen in Figure 2.10, respectively. Resilience against system-wide disruption outages caused by routing protocols or degradation of the transmission backbone core is therefore increased using this technique.

Due to the real-time nature of the information transmitted over ATM telecommunications, operators consider delayed packets, jitter and out-of-order delivery to be the equivalent of packet loss. In the EU, Eurocontrol, the European organisation for the safety of air navigation targets are based on the International Telecommunication Union's quality and availability targets [20].

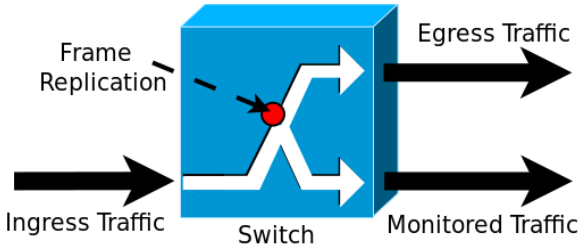


Figure 2.9: A switch with port mirroring for frame replication to facilitate uninterrupted operational traffic monitoring [19]

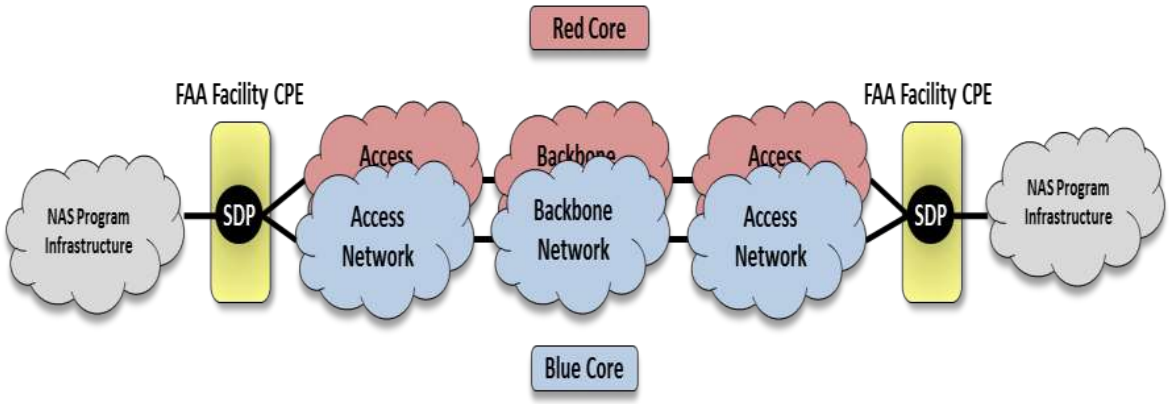


Figure 2.10: Redundant dual routing cores in FAA network topology

RMA Level	Required Performance	Max Restoration Time	Max Restoration Time of Jeopardy
1	.9999971	6 Seconds	3 Hours
2	.9999719	58.8 Seconds	3 Hours
3	.9998478	8 Minutes	3 Hours
4	.9979452	3 Hours	N/A
5	.9972603	4 Hours	N/A
7 (FTI-SAT)	.9970000	24 Hours	N/A

Figure 2.11: FAA service RMA levels and associated restoration times

ATM providers are increasingly using external providers for their ground data network infrastructure. While legacy infrastructure was owned and operated by many ANSPs, the isolation of ATM systems is reducing with some telecommunications providers offering shared access to their backbone networks with dedicated VLAN partitioning. Shared backbone infrastructure is used in the UK, while only FAA or other U.S. government services can use

the U.S. backbone network.

The third party providers of these networked systems follow Service Level Agreements (SLAs) relating to restoration times, availability, number of outages, etc. SLAs are applied to every FTI service and are the basis for service performance analyses. Standard SLAs used by the FAA, NATS and other organisations include:

- Availability
- Mean Time Between Outages
- Restoration Time

Figure 2.11 has the FTI Reliability Maintainability and Availability (RMA) levels with their respective *Maximum Restoration Times*. For RMA levels 1,2,3, those with restoration times < 10 minutes, restoration is performed by automatic switching between the redundant paths in the topology. The seven FTI RMA service levels are used for services of varying criticality with RMA-1 to RMA-3 used for high availability services with redundant paths with automated switching. RMAs 4 and 5 do not have redundant paths, while RMA 6 supports the FAAs administration network and RMA 7 is for satellite services. Internal FAA documents define over 100 service classes broken down by requirements of varying RMA, latency and physical interface, etc. A key goal of the FTI was to improve latency. Rate limiting is used to set the available bandwidth at each connection point and ensure no service can consume more network resource than expected.

NAS critical services do not necessarily correlate with the highest RMA levels. Criticality is determined by NAS availability and not necessarily the RMA availability. Availability is through combining RMA levels, switching, and alternative methods. Design choices for NAS critical services include avoidance. Avoidance or diversity, ensures that a service cannot be interrupted through one part of the infrastructure suffering an outage. The FTI satellite systems are a common isolated backup for providing avoidance however this cannot be used in every context due to the delays involved in satellite communications.

2.2.2.2 Future Infrastructure Plans

Air Traffic Management systems are undergoing significant transformations in both the EU and the U.S. through the Single European Sky ATM Research (SESAR) and the Next Generation Air Transportation System (NextGen) programmes, respectively. The transformations will see services coordinated over larger geographical areas and include integration of services such as radar and flight schedules. Figure 2.12 highlights the impact of NextGen technologies on the current U.S. NAS. One fundamental aspect of NextGen is the transition of services to use an IP-based network from Time Division Multiplexing [21]. Another key element in both programmes is the move from ground based radar to satellite surveillance and navigation systems. Key programs of modernisation within NextGen are the use of Automatic Dependent Surveillance-Broadcast (ADS-B) beacons, En Route Automation Modernization and En Route Communications Gateway (ECG). ADS-B uses GPS to ascertain an aircraft's velocity and other relevant telemetry which are passed on to ATM facilities. ERAM[8] is a more sophisticated replacement to previous ATM systems allowing ATCOs the ability to see more relevant information on their screens, improves security including backups and handles more operating modes and greater complexity of airspace configuration. Finally, ECG is a communications relay system that aggregates external data sources such as weather data and manages the flow of data into other systems e.g. ERAM.

The International Air Transport Association (IATA) predict 3.6 billion passengers in 2016, a 28.5% increase on passengers carried by airlines in 2011 [22]. Forecasts by the International Civil Aviation Organization for 2030 are shown in Figure 2.13 in Passenger-Kilometres Performed (PKP). While the largest growths by percentage increase take place outwith the U.S. and the EU, the increases in these regions are significant, with global passenger traffic expected to grow at an average rate of 4.8% per year through the year 2036 [23]. These trends remain on track in the most recent 2013 ICAO report [24] with most likely estimates revised slightly upwards with expected average annual growth of 4.9% equating to five billion PKPs per annum to more than 13 billion PKPs over the 2010-2030 period. From 2030 through

[8] `https://www.faa.gov/air_traffic/technology/eram/`; Accessed: February 2016

2040 current estimates are for continued average annual growth of 4.0%. ICAO's upper estimates predict global growth of 5.6% per year through to 2030.

In the face of such significant growth, EU Air Transportation systems are increasing their interconnectivity through an international Airport Collaborative Decision Making (ACDM) system[9]. The ACDM system, currently implemented across 16 European airports, shares Flight Update Messages and Departure Planning Information messages with centralised ATM facilities[10]. This information allows for numerous system-wide efficiencies for take-off and landing slot management through more accurate take-off information, optimised use of stands and gates, and greater predictability. As a result, the available capacity will be used more effectively. ACDM is one of numerous examples of interconnecting data networks with the goal of improved capacity and efficiency. Eurocontrol released a cost-benefit analysis on the introduction of ACDM highlighting the benefits of increased efficiency, predictability (including under adverse conditions), capacity and flow across the entire ATM system [25].

The FTI forms the basic infrastructure for NextGen in the U.S., since future growth and new services will require a lot of bandwidth. A dominant future umbrella service for data sharing within Europe and North America is the System Wide Information Management (SWIM) service. SWIM pools and shares numerous feeds of data to provide structured information encompassing flight data, weather information and airport operational statuses among others. Other significant changes within the NextGen and SESAR programmes include a move towards using more precise positioning systems to allow for a greater safe density of air traffic, wake modelling to allow for greater capacity at runways with reduced waiting time between different aircraft types, and the global demand for regulation of Unmanned Aerial Vehicles.

Unmanned Aerial Vehicles and their wider systems, Unmanned Aircraft Systems, which include Ground Control Systems (GCS) and antenna communications, are an increasingly desired aspect of the NAS and airspace use worldwide. The benefits of UAVs are clear, with

[9] https://www.eurocontrol.int/news/rome-fiumicino-cdm-implementation-gears-critical-mass-full-benefits; Accessed: April 2014

[10] http://www.eurocontrol.int/sites/default/files/publication/files/2012-airport-cdm-manual-v4.pdf; Accessed: February 2016

applications in agriculture, search and rescue, industrial inspection, and infrastructure monitoring. Their integration to date has been limited due to strict policy and safety requirements. Numerous policies also need to be revised such as the U.S. requirement for a logbook to be stored within the aircraft cockpit. Currently, UAV systems within the U.S. and EU are predominately limited to environmental work, in areas with low population densities. The future demand for UAS as part of the wider ATM global system is clear. NASA have a dedicated research program on the possible integration of UAS within the NAS[11], and their research has also led to the first FAA approved safety case for an emergency UAV mission [26].

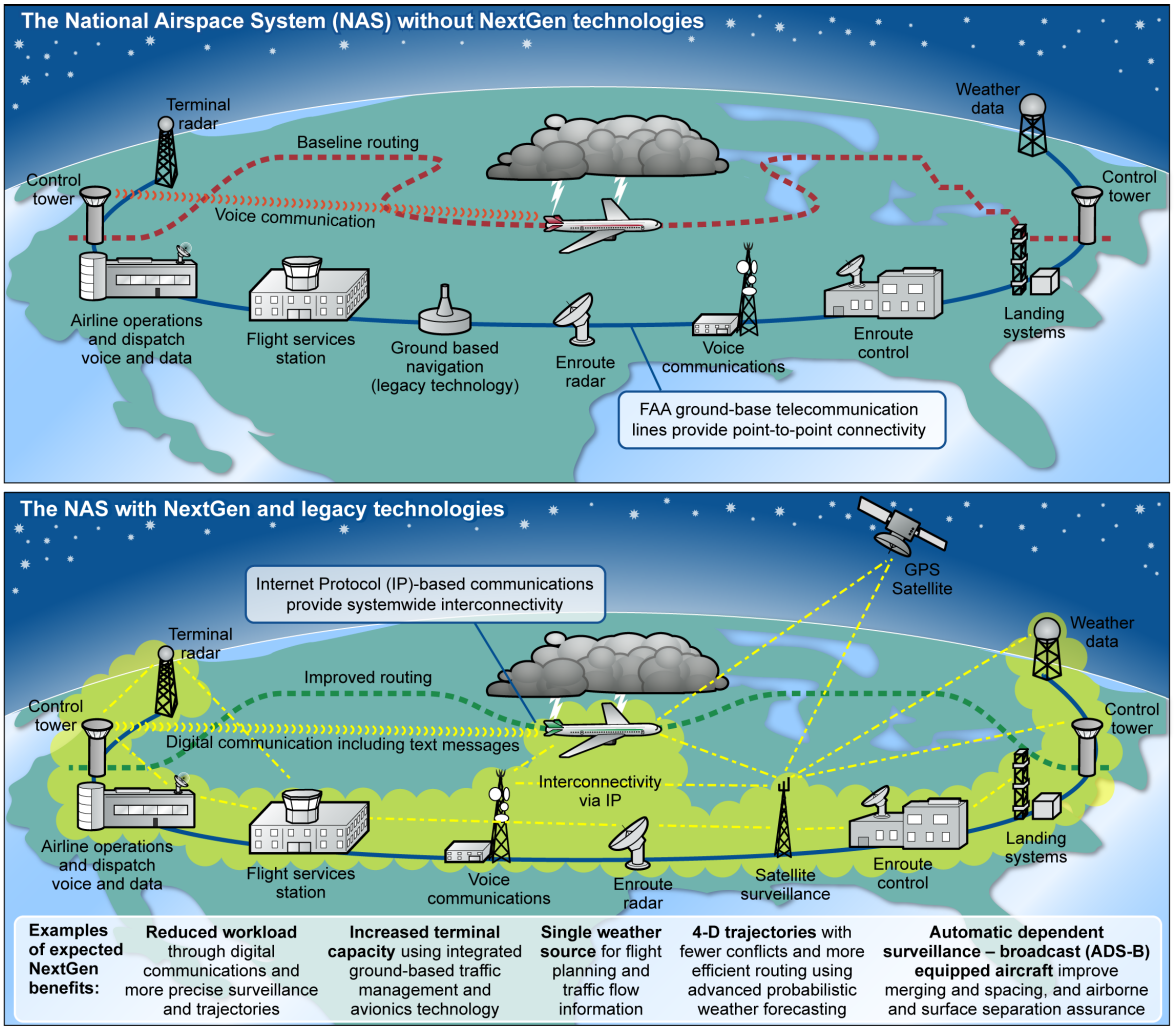
Many UAVs are small hobbyist remote control devices. The FAA distinguishes these from larger, heavier commercial aircraft which now require registration. In 2016 the forecast for non-hobbyist (>55lbs) registrations is >600,000, and current estimates are 2.7 million by 2020[12]. The FAA believe registrations will drastically increase when proposed operations with multiple UAVs controlled by a single pilot and Beyond Visual Line of Sight (BVLOS) are permitted.

2.2.3 Architectural Shortcomings

ANSP ATM data networks such as the FTI are clearly very sophisticated, with near perpetual service in the face of outages and inevitable disruption through impressive resilience techniques including redundant architectures, diversity and over-provisioning. However, there are areas in which these infrastructures are lacking. A recent MIT publication by Newell et al. [17] studied the FAA's national weather systems and presented two core areas where the FTI could be improved. The first is data traffic prioritisation and the second is with respect to bandwidth at gateways on the boundary of the FTI and external networks. Newell states that while the FTI offers high-reliability connections, there is no way at the physical network level to distinguish between high priority weather alerts, such as wind shear, from lower priority but very large, weather reflectivity data sets. Currently, this prioritisation and classi-

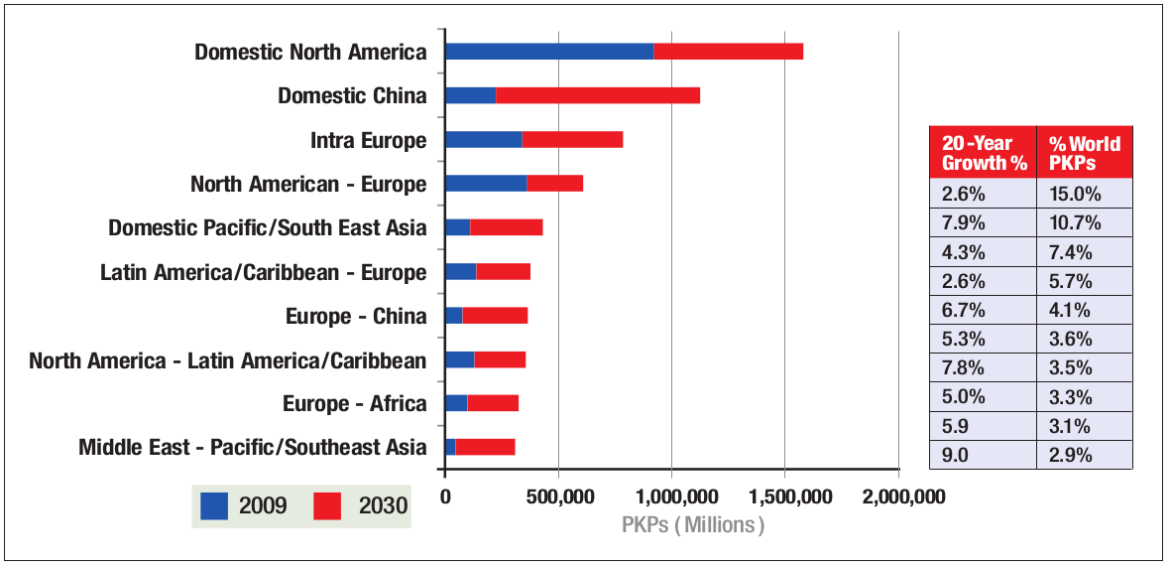
[11] http://utm.arc.nasa.gov/index.shtml

[12] `https://www.faa.gov/data_research/aviation/aerospace_forecasts/media/Unmanned_Aircraft_Systems.pdf`; Accessed: March 2016



Source: GAO. | GAO-15-370

Figure 2.12: The U.S. NAS before and after NextGen technologies



20-Year Growth %	% World PKPs
2.6%	15.0%
7.9%	10.7%
4.3%	7.4%
2.6%	5.7%
6.7%	4.1%
5.3%	3.6%
7.8%	3.5%
5.0%	3.3%
5.9	3.1%
9.0	2.9%

Figure 2.13: ICAO worldwide Passenger-Kilometres Performed forecasts [23]

fication must be handled by the application layer. The issue, Newell argues, is that physical in-network devices such as edge routers, can have their queue buffers filled with low priority traffic leaving higher priority traffic at risk of packet loss or unacceptable latency. Traffic shaping, prioritisation, and more generally Quality of Service (QoS), challenges are prevalent recurring topics in networking research [27, 28]. Traffic shaping is a component of QoS that includes techniques such as temporary traffic buffering, congestion avoidance policies e.g. dropping certain packets or imposing bandwidth limits. This is commonly used by QoS when traffic is prioritised, as different segments of traffic are given bandwidths. Recent QoS research by Hruby [29] on Voice over Internet Protocol (VoIP) QoS and backbone scalability issues suggests a backbone overlay network for time-critical services. Newell considers separate links, discounting these from a cost perspective but does not tackle the concept of overlay networks beyond a preference for prioritisation in the physical layer. Other recent works in backbone QoS also advocate the latest best practices in networking such as virtualisation [30, 31] which are explored further in Section 2.3.3.

In general, ANSP ground data networks are relatively static, highly complex networks with real-time mission critical requirements. As bandwidth demands increase with greater traffic volumes and increasingly rely on ever-more automated and interconnected systems for ATM, the costs to upgrade infrastructures to remain as over-provisioned, in a highly resilient, redundant, diverse architecture are often beyond cost benefit analysis or when planned will take years to implement and deploy [32]. Exploring necessary improvements for increased capacity within the existing infrastructure is also a major challenge.

In both the U.S. and the UK, enormous, emulative testbeds exist to test any new upgrades, equipment, protocols, and connections prior to a real-world implementation on the production infrastructure. FTI tests take place at the Harris Service Verification lab and at the FTI National Test Bed. NATS perform tests at their central control facility in Swanwick. Test beds will never be truly reflective of the production environment, due to the costs involved. Reconfiguring such networks is a costly and difficult operation requiring significant testing and planning. Therefore, any changes must have significant benefit and demand. Even when

changes are approved it can take a lot of effort to reconfigure these networks, which means when incidents and outages do occur, they can be prolonged.

Another outcome of the environment of perpetual service, numerous interconnected services, high redundancy and difficulty in reconfiguring is infrastructure patching. Patching often takes the form of in-network devices or middleboxes which perform a static translation, relay or reconfiguration. Patching the infrastructure ensures that any protocol, routing or service specific needs can be met in the face of future changes. A significant amount of infrastructure patching occurs to keep legacy systems functioning, making the wider nexus a more complex system of systems. This has very little impact on overall safety though, due to the highly resilient architectures and expectations that large amounts of the infrastructure can fail, whilst still maintaining an overall resilient service.

Earlier in this chapter some of the resilience paradigms deployed by ATM infrastructures around the world were reviewed. Despite these measures, complete system-wide outages do occur and often, due to the complexities of these infrastructures, they can be for extended periods of time. Section 3.2 reviews numerous incidents which have impacted the survivability of services and highlight the design choices in this work which will aid faster recovery and allow operators to make more informed and aware resilience choices.

While clear benefits of greater interconnectivity have been discussed, the reduction in isolation is a risk that has been identified and discussed for many years. A 1997 report [33] by the U.S. General Accounting Office (GAO) recognised that using interconnected systems brought significant efficiency savings and improvements across the FAA and wider government systems, and it also argued that threats from cybersecurity issues significantly increased. The GAO particularly highlights the infrastructure vulnerability to anonymous intruders who may gain access to information or have a greater capability to disrupt operations. This is a recurring argument today with opinions on both sides with respect to interconnectivity both weakening and strengthening cybersecurity. Those who advocate a strengthening insist that using more widely used, less isolated processes have the protection of many more users discovering, alerting, fixing and updating problems as they arise,

while isolated, potentially bespoke systems suffer from a lack of exposure leaving unknowns available to an insistent intruder. Clarke [34] provides a good summary of arguments both for and against. Ultimately, there are always trade-offs with such decisions and overriding these claims, as stated in the analysis, is the need to keep up with the high levels of quality from private sector organisations with which the public are familiar. A follow-on report in 1998 [35] specifically examining computer security with respect to Air Traffic Control was very negative with a summary stating the FAA was ineffective in all critical areas undertaken in the computer security review. The most applicable high-level outcome was that the GAO considered the FAA ineffective in managing the systems security for future ATM modernisation systems, noting that there are inconsistencies with respect to security requirements and specifications for new systems. Finally, the GAO report [35] stated there is not a well formulated system-wide security architecture, concept of operations nor security standards.

2.2.3.1 U.S. General Accounting Office Analyses

Many of these points echo today with recent GAO reports [5, 16] in 2015 stating the FAA needs to address weaknesses in Air Traffic Management systems. A variety of weaknesses are explored in these reports ranging from policy decisions, managerial structures to physical security measures, e.g., blocking physical access to networked systems. Other issues addressed highlight unauthorised access prevention and limitation weaknesses, unencrypted sensitive data and how users are authorised when accessing systems. The issues addressed most applicable to this work include insufficient systems monitoring, shortcomings in boundary protection, configuration management controls, a lack of testing security controls and timely recovery in the face of outages and disruption. Examining these in more detail with respect to this work;

- **Insufficient Systems Monitoring** Insufficient monitoring and auditing activity for the FAA systems could be considered a point for continual improvement for any system. The report [5] expands to explain that this issue refers to the regular and frequent collection, analysis and review of monitored data for abnormal activity, which implies

intrusion and anomaly detection systems also fall under this criticism. The report states that the FAA did not consistently implement sufficient controls from a wide range of best practices including logging, network and host-based intrusion detection systems e.g. packet sniffing and that the FAA did not have the capability to sufficiently monitor network traffic or ensure that security-related events were being logged across NAS systems. The GAO report [5] also states that the current deficiencies hamper efforts to ensure that remedial actions are addressed in a timely manner and that those responsible for NAS system-wide incident detection and response did not have sufficient access to network sensor data or security logs on the operational network. This second point may be a policy or access control issue, however, network monitoring across European ANSP could be substantially improved to aid engineers' understanding and assist with reducing recovery times. The author was informed anecdotally that much of the network monitoring in place at at one European ATM centre consisted of Simple Network Management Protocol (SNMP) [36] polling every 15 minutes which was predominately used as confirmation something had gone wrong. Problems were reported by ATCOs directly to network engineers long before the monitoring systems reported them. Of course, this has no impact on safety since the entire infrastructure is based upon enormous resilience and redundancy, allowing for the safe failure of individual components.

The NAS Cyber Operations (NCO) team, those responsible for many cybersecurity aspects, do not have sufficient access to notable shortcomings in this area including:

- No full network packet capture in place at network interface points or operational facilities.
- No anomaly detection capability in place at edge routers or operational facilities.
- No easy access to network flow session data
- No access to intrusion detection, packet capture or flow data at network gateways.
- Incomplete event logs

  - Poor search capability of centralised, aggregated logging system (including inability to search beyond any interruptions in the log recordings).

Numerous cybersecurity experts consulted within the report advocated a holistic continuous-monitoring program with real-time monitoring of the NAS boundaries, intrusion detection, real-time monitoring and anomaly detection. The GAO report [5] concludes that a direct consequence of inferior monitoring information meant the FAA faced an increase risk and reduction in its capability to respond to outages and malicious activities on its systems.

- **Shortcomings in Boundary Protection** - Boundary protection control issues which were highlighted centre on the connectivity between the operational NAS network and less secure systems, as explored in the topological review earlier. The boundary protection controls e.g. at FTI edge routers should, according to the GAO, be more frequently checked to determine they are operating as intended. Beyond this, the GAO state that the mechanisms in place at these edge and inter-system boundaries were insufficient at protecting and restricting connections into and out of network-connected devices.

- **Slow to implement corrective actions** - Another GAO report [16] is critical of the speed at which identified weaknesses are fixed. The report notes an example issue that was identified in the deployed intrusion detection system in 2008. The shortcoming was still unresolved in 2015. The report recommends creating milestones for the corrective actions on the discovery of weaknesses. In some legacy systems performing such updates can be non-trivial due to the lack of modern techniques such as virtualisation or decoupling middlebox software from hardware.

- **Poor recovery times in the face of outages and disruption** - The GAO report [5] concludes that the lack of monitoring is directly limiting the Federal Aviation Administration's ability to detect and respond to security incidents affecting its mission-critical systems. With insufficient monitoring capabilities poor recovery times are to

be expected. A significant overhead in systems recovery is in the diagnosis and understanding of the problems which arose to cause the disruption or outages. To implicitly improve recovery times, better informed operators are more likely to be able to understand problems more quickly. To better inform operators, better monitoring capabilities must be made available to them, so that they can investigate the change in behaviours within the network. Due to resource constraints, not all aspects of a system can be monitored or stored indefinitely. By having a mix of capabilities where some long term centralised monitoring takes place, alongwith deployable on-demand capabilities which are relevant to inform the current activities, operators can increase their awareness and be better informed, thus improving recovery times.

2.2.3.2 Main Points

Many challenges have been highlighted in this section. Each of these challenges must be addressed in order to continue to provide increasing levels of safe air travel and reduce disruption and outages. The primary concerns raised for which this work makes a contribution towards are: the inflexibility and lack of adaptive monitoring capabilities for operators; assist the speed of deployment for corrective actions; the poor recovery times in the face of outages due to lack of information from monitoring systems; and monitoring and anomaly detection capabilities and lack of on-demand adaptive actions such as data traffic prioritisation. Providing explicit and implicit contributions towards the above weaknesses, will ensure future ATM operators can be better enabled to provide highly reliable, resilient ATM services whilst incorporating future needs and demands.

2.3 Network Softwarisation Technologies

2.3.1 Technology Trends for Future Networks

The desire to flexibly adapt, control and monitor networks is not held by ATM network engineers alone. High flexibility, rapid reconfiguration via network programmability [37]

and movement away from commodity hardware with vendor lock-in have been key emerging trends in the state-of-the-art in telecommunications networking over the past few years. Challenges of complicated, expensive, incomplete test beds have also been faced by many infrastructure providers struggling to innovate.

Traditional networking approaches have become a barrier to creating new, innovative services by inhibiting easy experimentation and flexibility because they are too closed, too proprietary and too complex. These limitations stem from the complexity which arises from implementing switches, routers, middleboxes and other devices with an array of distributed protocols over closed and proprietary interfaces whilst supporting legacy services and modernisation. Given this context, it is very difficult to innovate and quantify the benefits of new approaches or services within the existing infrastructure in order to ensure reliable, realistic and thorough testing prior to becoming production-ready, especially in mission-critical environments.

In general, operators cannot easily customise or optimise their expensive infrastructure to be tailored to specific use-cases for stakeholders or evolving business needs. Modern business methods are seeking more agile ways to exploit their infrastructure. The following problems encapsulate the experiences of many operators:

- Networks, their traffic and topologies, are complicated to manage [38].
- Operators are struggling to introduce new revenue streams due to limitations and barriers to innovation [39].
- Slow lifecycle for deployment of new capabilities, with time-consuming vendor product release cycles or test-bed configuration [40].
- Difficulty customising cost-effective solutions to customer or stakeholder needs [41].

Numerous other, wider factors have led to the demand for a more flexible networking approach. These factors include the explosive growth of mobile devices, wireless access, distributed Cloud services and a general move from isolated internal services to shared, collective services. Server virtualisation and a move away from client-server computing are

also driving the trend for modern architectures. In data centres and enterprise topologies, operators are seeing an increase in *east-west* traffic, where applications behave as peers, pulling data and processing from many sources, versus traditional *north-south* traffic patterns of many clients accessing one server for data or processing applications held centrally. Increasingly, common datasets of enormous volume are also driving this traffic trend, with parallel processing taking place simultaneously across multiple interconnected servers whilst consuming a great deal of bandwidth. The need for machine-to-machine interconnectivity is fundamental to the design of national ATM data network topologies [5, 42]. Different locations are sharing the latest weather reports, take-off information, delays and scheduling in order to improve efficiency and therefore airspace capacity.

Air Traffic Management data networks experience the majority of challenges discussed above. Recurring themes for the wider networking community of complexity, inability to scale, inadequacy to evolve or respond in line with ever-increasing business and security requirements and vendor lock-in are highly applicable within the ATM context [43]. The Open Networking Foundation (ONF) produced a white paper in April 2013 [44] which highlighted these points among others. The ONF paper stated that complexity leads to stasis and that the practice of defining bespoke problem-specific protocols in isolation, without abstraction, has significantly contributed to the complex context seen today. ATM networks have a vast array of context specific protocols with application traffic from weather to surveillance information. One such protocol is the ASTERIX data format for *All purpose STructured Eurocontrol Radar Information eXchange* [45]. Another applicable key point is that for the movement or addition of any networked device, the engineers must configure or check multiple switches, routers and middleboxes. Firewalls, VLANs and Access Control Lists (ACLs) may also need low-level, device-specific modification along with any QoS or other protocol-based mechanisms. Diversity, a vital component of strong resilience [46] relies upon a variety of vendors and software as well as alternative, redundant means. Configuring aspects such as QoS across all diverse and redundant paths on a per-application basis is a significant overhead for ATM engineers and is another reason for infrequent change. As the ONF report [44] states, this is in sharp contrast to the modern dynamism seen in current server environments

where virtualisation has mitigated the problems associated with binding processes directly to hardware. Advantages of Virtual Machines (VMs) include processes and applications being host and location agnostic, migration and load-balancing, dynamic and programmable optimisation and rapid response to opportunities and challenges in general [47].

The static nature of IP networks is the underlying root cause of the issues outlined above. While being static brings some benefits, there is always a trade-off among paradigms. The current demands and expectations for lower OPEX, greater capacity across expensive infrastructure, rapid flexibility for security and recovery and ease of innovation, result in flexible and dynamic architectures being vital attributes for today's infrastructures [44].

2.3.2 Software Defined Networking

2.3.2.1 SDN Concept

Software Defined Networking (SDN) is an approach which allows for greater network programmability through the decoupling of the control and data planes, which manage logical routing and traffic forwarding, respectively.

The three core principles of the SDN paradigm as defined in the ONF architecture [48] are:

- Decoupling the control and data planes
- Logically centralised, physically distributed, control
- Network programmability

To achieve these principles, the following key architectural components are defined:

- SDN Controller
- SDN Datapath
- SDN Control to Data Plane Interface
- SDN Northbound Interfaces

The principle of decoupling the control plane from the data plane allows for modern network programmability. The SDN Controller manages the flow control for network resources. Logically-centralised control ensures that there is a single controller entity, which when implemented in a resilient context, should be physically and diversely distributed with redundancy. To achieve centralisation of control, it must be decoupled from hardware. The ONF state that managing network resources with an over-arching perspective can be done more efficiently [44]. Caveats to this assertion are stated and include vast scale with propagation latency issues and cases where tasks are best-suited to low-level management such as, e.g., link protection, where near-instantaneous switching can be performed locally with abstracted details passed to higher level control. Network programmability allows control of flows within the network, which can be dynamically updated. For example, when a flow with a previously unseen destination port and IP combination begins, the route for this flow can be updated. This enables intelligent networking [49].

Figure 2.14 highlights the layers of functionality within an SDN architecture. SDN-enabled Networking Devices can be less complex than traditional networking devices since, under SDN, they look after the data plane alone and simply need the capability to receive SDN instructions [50]. The SDN Controller relays the requirements from the SDN Application layer to the network devices. Applications communicate with the SDN Controller through an Application Programming Interface (API). Example applications are network management tools, analytics or centralised anomaly detection. The SDN APIs are known as *northbound* and *southbound* interfaces for communications between the SDN Controller and the applications and between the SDN Controller and the network devices, respectively [44].

2.3.2.2 Implementations

Protocols One of the first implementations of the SDN architecture is OpenFlow [37]. The implementation by McKeown et al. is a protocol for programmable networks which allows traffic to be managed and directed across switches and routers, independently of the vendor. OpenFlow is the most popular and widely used SDN implementation with signif-

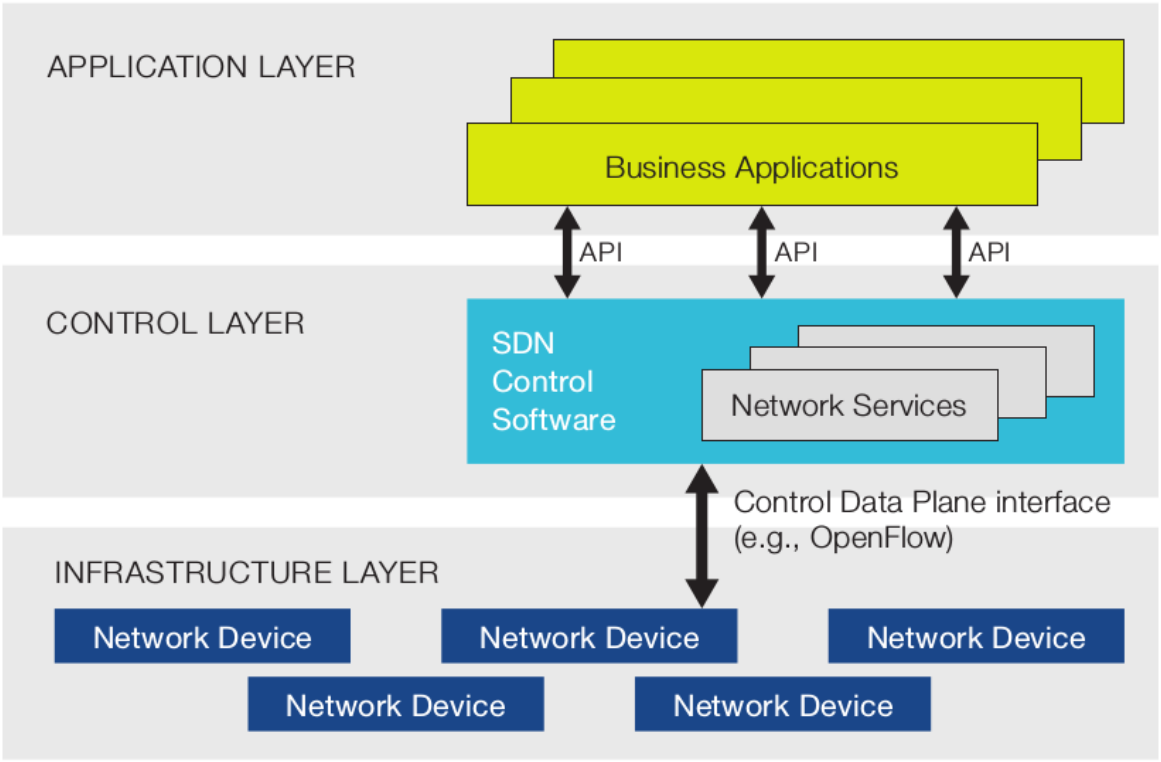


Figure 2.14: Conceptual operational layers of the SDN paradigm [44]

icant related research. There are numerous other proprietary SDN implementations. Cisco have developed their own protocol, OpFlex [51] which is open and extensible with the goal of transferring policy information in XML or JSON from the Cisco controller to devices. However, this would increase vendor lock-in issues with respect to Controller technology. Various other methods of achieving the goals of the SDN model have been proposed including: use of IETF's ForCES [52], Interface to the Routing System (I2RS) [53], a path computation element (PCE)-based architecture [54], an Extensible Messaging and Presence Protocol (XMPP) [55] based architecture or an extension to the application layer traffic optimization (ALTO) data model [56]. The merits of these various implementations are outwith the scope of this work.

Controllers There is a vast array of OpenFlow complaint SDN Controllers. As Monaco et al. state [57], the key choice of Controller is often the programming language used. An incomplete list follows: NOX [58] (C++), POX [59] (Python), OpenDayLight [60]

(Java), RYU [13] (Python), Beacon [61] (Java), Trema [14] (Ruby/C), Floodlight [15] (Java), Nettle [62](Haskell) and ONIX [63] (C++/Python/Java). Yet more controllers are built on top of other controllers such as, e.g., SNAC [16] which is built on NOX. Again, the author leaves comparison of these controllers to other studies but note that Monaco observes many controllers do not update regularly and still support only the original version of the OpenFlow protocol.

Middleboxes The emergence of virtualisation-enabling protocols such as OpenFlow is also impacting on the hardware technologies of in-network and middlebox devices. Today's enterprise and Air Traffic Management networks almost ubiquitously deploy middlebox services to improve security and performance. Although virtualisation of middleboxes is currently attracting significant attention, studies show that such implementations are still proprietary and are often deployed in a static manner at the boundaries of organisations, hindering open innovation. Enterprise networks rely on a wide spectrum of hardware-based network appliances or 'middleboxes' to transform, inspect, filter or otherwise manipulate network traffic on top of packet forwarding. Anecdotally, the author has learnt throughout this work, through conversations with senior ATM engineering teams, that ATM networks heavily utilise middleboxes. One of the most prevalent uses is interfacing across different legacy system boundaries. The focus on legacy support through in-network transformations is due to the desire for the reliability that consistent legacy systems provide.

In recent years, middleboxes have become fundamental parts of operational networks, providing approximately 45% of the network devices to enforce enterprise security (e.g., firewalls and intrusion detection) and performance (e.g., rate limiters, proxies, load-balancers) policies throughout the topology [64]. Recent studies have shown that the advent of diverse consumer devices that rely on different, network-intensive cloud services as well as the increasing need of in-network security will increase the demand for middleboxes even

[13] `https://osrg.github.io/ryu/`; Accessed: March 2016
[14] `https://trema.github.io/trema/`; Accessed: March 2016
[15] `http://www.projectfloodlight.org/floodlight`; Accessed: March 2016
[16] `http://www.openflow.org/wp/snac/`; Accessed: March 2016

further [65]. However, despite their increasing popularity, hardware-based middleboxes have significant drawbacks: they incur significant capital investment due to being provisioned and optimized for peak-demand, are cumbersome to maintain due to the expert knowledge required, and cannot typically be extended to accommodate new functionalities as operational requirements emerge. They run on proprietary software which limits innovation and creates vendor lock-in [39]. Network Function Virtualisation is a novel approach to address the above shortcomings of managing closed and proprietary appliances by decoupling network functions from their hosting hardware platform.

2.3.3 Network Function Virtualisation

Network Function Virtualisation (NFV) was proposed by a non-propriety consortium of network operators in a white paper [39], explicitly noting the independence of NFV from SDN. The white paper defines NFV as a transformation in the delivery of network functions from bespoke, specialised hardware, to network functions in software which can run on a range of industry standard hardware, which can be migrated to, or instantiated at, various locations within the network topology, on-demand. Figure 2.15 shows this transition with traditional Network Functions (NFs) on commodity hardware listed on the left and the NFV paradigm shown on the right with industry standard generic hardware. Cloud services for independent software vendors and a range of Virtual Network Functions (VNFs) which can be installed on the generic hardware, are shown at the top right of Figure 2.15.

Briefly examining the wider NFV framework, there are three core aspects:

- VNFs are software implementations of Network Functions that can be deployed on a NFV infrastructure (NFVI).
- NFVI comprises all the hardware and software components in the VNF deployment environment. The NFV infrastructure may cover multiple locations with the underlying network connecting these locations included within the NFVI.

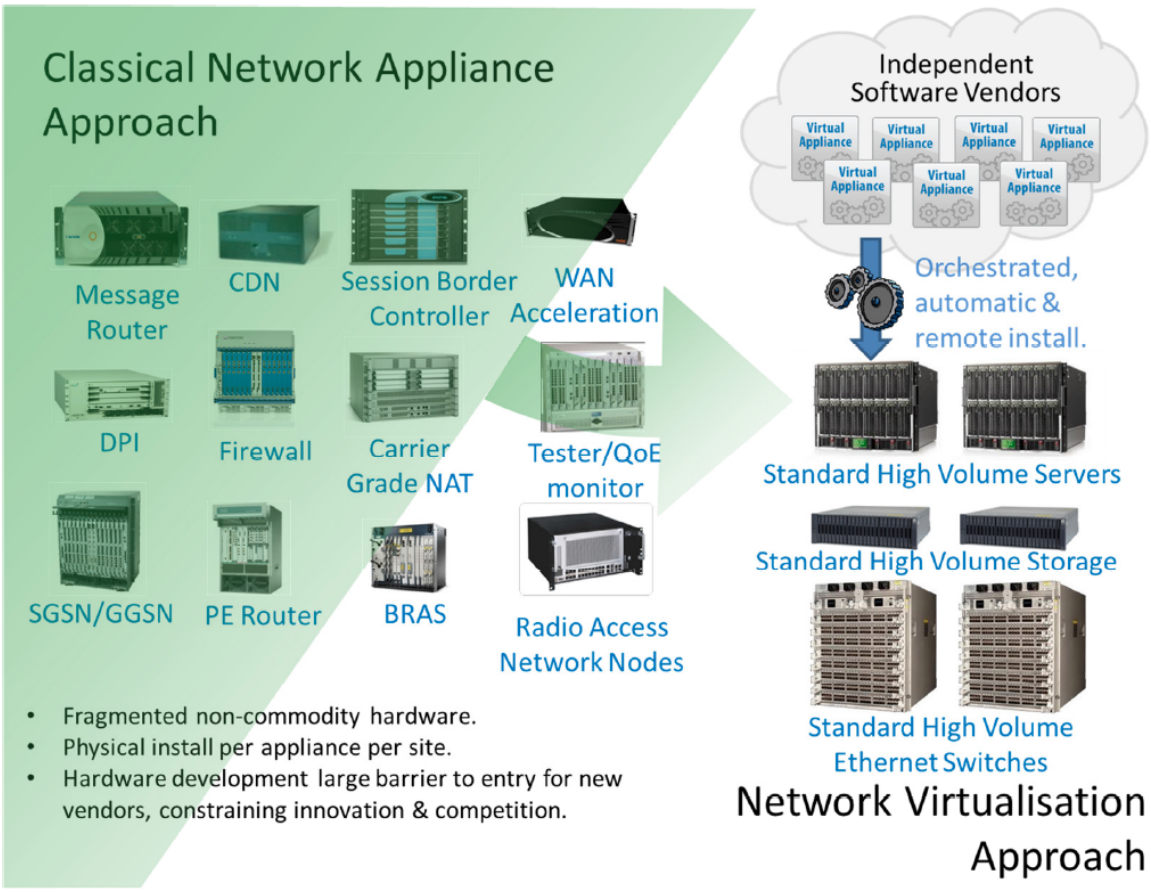


Figure 2.15: Abstracted transition from traditional approaches to Network Function Virtualisation approach [39]

- NFV Management and Orchestration framework (NFV-MANO) is the collection of all functional blocks, the data they access and their interfaces.

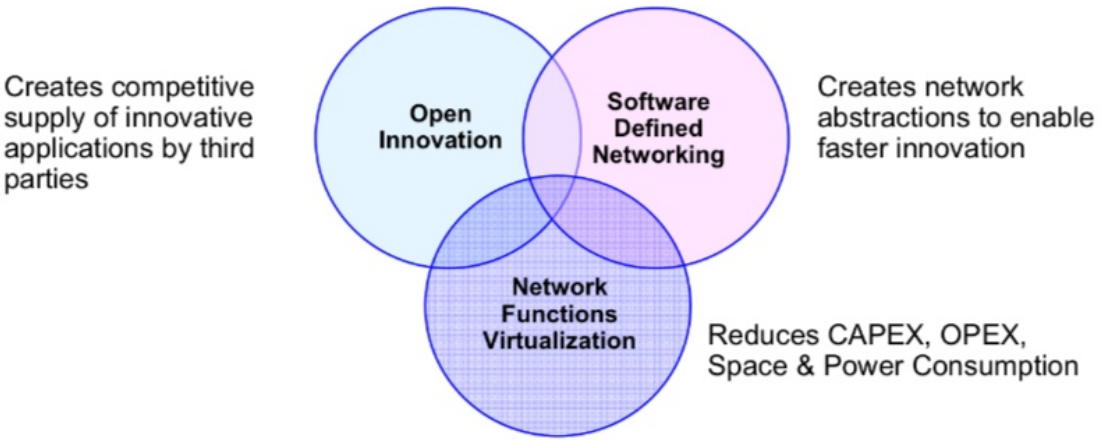


Figure 2.16: Abstraction of the distinction of NFV, SDN and open innovation [39]

Figure 2.16 highlights the distinction of the SDN and NFV architectures while portraying a level of interdependence and overlap both with each other, and importantly with open innovation. By using low-cost commodity servers, NFV can reduce CAPEX, OPEX and maximize Return on Investment (RoI) [66]. Many recent NFV deployments by large Internet Service Providers (ISPs) and enterprise network operators suffer from the statically-configured underlying routing mechanisms in place, which do not support open interfaces and result in operator and environment-specific solutions in static or semi-static environments [67]. While ATM environments can be considered more static than enterprise networks, this would also adversely affect ATM operators. For example, deploying one or more network functions requires the update of all affected switches' routing tables to redirect traffic, therefore making it impractical to deploy infrastructure-wide NFs. Consequently, NFV systems exhibit poor component reuse, and are still unable to fulfil dynamic, temporal traffic workloads in an elastic manner [68, 69]. In such environments, there is no cross-layer information exchange between the routing layer and the network functions, which results in a limited view of the network to each functional entity.

2.3.4 Latest Network Research

In this section, the state-of-the-art in related networking research is examined and critiqued where appropriate to highlight the contributions of this work. Firstly, the implementations which combine the SDN and NFV paradigms are assessed, observing that no implementations offer the flexibility and open programmability this work provides, also noting the arguments for container-based virtualisation over VMs. Then, network monitoring is briefly reviewed, focussing on the shift in this established field with the introduction of these new approaches to networking and the latest research exploiting new implementations. This work also presents contributions in context-specific anomaly detection and network monitoring, therefore similar overviews and reviews in the state-of-the-art in these fields within the ATM environment are discussed.

2.3.4.1 Research combining Software Defined Networking & Network Function Virtualisation

SDN and NFV are complementary technologies and can be functional building blocks of each other: SDN can be exploited to dynamically isolate and route traffic to specific NFs by abstracting the physical topology, while NFV can create the virtual infrastructure upon which further SDN abstractions (e.g., virtual networks) can be instantiated.

Middlebox virtualisation and the development of a NFV framework have attracted considerable research effort in recent years. ClickOS [70] focuses on the design and implementation of a Xen-based software platform optimized for middlebox processing. Although it provides high processing performance, the platform uses a specific programming environment (Click) and a modified, Xen-based hypervisor to run NFs. It would therefore require considerable effort to develop and integrate new NFs or reuse existing, well-proven software (e.g. Snort[17] or Bro[18]). Developing bespoke NFs is a significant drawback and a better architecture, such as that presented in this work, would allow generic management for NFs using container-based virtualisation, and provide a generic platform e.g. Linux for NF implementations. Experiments comparing this work to ClickOS are presented and discussed in Section 4.2.6.

CoMb (Consolidating Middleboxes) [71] is an architecture for middlebox deployments that systematically explores opportunities for consolidation. The authors present a centralised, software-centric architecture that manages middleboxes as simple processes. xOMB (the eXtensible Open MiddleBox) [72] is an extensible and consolidated framework for incrementally developing scalable middleboxes, similar to CoMb. Both of these works leverage the idea of reusable network processing pipelines for middlebox composition. However, both CoMb and xOMB lack an incorporated deployment model for SDN-enabled networks and packaging of NFs to software containers for more flexible image management. Cziva et al. [73] also present a method for reducing network-wide communication cost by migrating VMs in a SDN-enabled network which can be extended to support NF consolidation. These works can be seen as components towards the aim of an overall SDN compliant NFV

[17] `http://www.snort.org`; Accessed: March 2016
[18] `http://www.bro.org`; Accessed: March 2016

architecture, which this work achieves.

OpenNF [74] and Split/Merge [75] are two SDN+NFV based migration frameworks that aim to simplify scaling and improve efficiency of migrating middleboxes. Each propose an API that middleboxes must use to support migration benefits. Split/Merge implements an API based on a hash-table in memory. OpenNF implements get/put state calls which allows control state migration. To run a given Virtual Network Function in these frameworks requires major implementation modifications. However, migration of NFs is significantly more efficient, which helps with dynamic redistribution to achieve elastic scaling, for example. Serious concerns have been raised about both these approaches, with [76] identifying that both suffer from key safety, efficiency, and scalability problems. To mitigate the significant shortcoming of the requirement of OpenNF to make software changes, the authors offer StateAlyzr [76] which they claim can significantly reduce manual effort, however making PRADS[19] OpenNF-compliant still required ~6 man-hours of work.

Cloud4NFV [77] provides orchestration and management for NFs following the European Telecommunications Standards Institute (ETSI) guidelines [39]. It focuses on service chaining and deployment over a Cloud environment, but it provides no integration between a programmable (e.g., SDN-based) underlying routing environment and the NFV layer. Kreutz et al. [78] provide an excellent survey for further reading on the wider scope of recent SDN research which falls outwith the scope of this work.

2.3.4.1.1 Container-based virtualisation Container-based virtualisation [79] is a scalable, high-performance alternative to hypervisors where the virtualisation layer runs as an application within the operating system. In this approach, a commodity, general-purpose operating system runs on the hardware and hosts several isolated containers that share the same kernel. The main advantages of this approach is efficiency, lack of hypercalls overhead, and high container-per-host density. This work exploits containers rather than fully virtualised VMs, offering significantly better performance and faster lifecycle management for the NFs than is achieved in comparable VM-based implementations such as, e.g.,

[19] `http://prads.projects.linpro.no`; Accessed: March 2016

Cloud4NFV [77].

2.3.4.2 Network Monitoring

Network monitoring has been a central feature of network management since its inception. It is a vital component for the reliable management of critical networked infrastructures in general. ATM systems, as discussed earlier in this chapter, have been criticised in existing work with respect to their network monitoring capabilities. There have been decades of research into both passive and active measurement techniques. Active measurement involves creating stimulus traffic with certain characteristics on a network to test particular attributes of a service [80–83]. Due to the behavioural interference of additional traffic with the operational traffic, its wide-spread use is unsuitable for ATM systems. On the contrary, passive monitoring is unobtrusive and measurements are taken by observing operational traffic [84–86]. Previous work by this author, KSWatch [19], examined the use of replaying such passive monitoring, tailored to the ATM environment, to assist network engineers diagnose issues. Traditional network management techniques such as Remote network MONitoring (RMON) [87], an extension of SNMP [36] on which KSWatch is based, and Netconf [88], etc. have become less useful under SDN. This is because OpenFlow captures flow based metrics providing similar insights to what the above techniques provided.

The rise of SDN, OpenFlow, Cloud services and virtualisation such as NFV are having a profound impact on network monitoring. A significant number of tools have emerged, tailored to OpenFlow offering specific functionality including: FleXam [89], flexible sampling extension for OpenFlow; OpenTM [90], a tool for estimating traffic matrices, specifically for the NOX Controller; OpenNetMon [91], a POX implementation which monitors QoS indicators; OpenSample [92], a low-latency sampling-based measurement tool based on a modified version of sFlow[20], a high-speed switched networks monitoring tool; and FlowSense [93], which measures utilization of links in OpenFlow.

Other tools have emerged with specific deployment environments such as: PaFloMon [94],

[20] `http://www.sflow.org`; Accessed: March 2016

a passive monitoring tool suite built for FlowVisor [95], a special purpose OpenFlow controller that can divide rich slices of network resources and delegate them to other controllers to manage; BISmark [96] offers measurements of both passive and active monitoring within Procera [97], an event-driven control architecture for SDN; Distributed Collaborative Monitoring (DCM) [98], a distributed high-accuracy, low-memory traffic monitoring solution with a specific DCM controller; OpenSketch [99], a traffic measurement architecture separated from the data plane; and PayLess [100], a real-time monitoring framework which supports querying, implemented within a Floodlight controller environment.

Choi et al. [101, 102] propose an intelligent management middlebox which logically centralises the decision-making processes for managed services under NFV. Currently, this work is a proof of concept without strong performance analysis or proven scalability, however, their design offers flexible virtual network with resource auto-scaling on-demand. Another early proof of concept is the Expedition network monitoring tool [103] which aims to perform universal fault and performance management monitoring for a variety of SDN controllers and topologies managed by these controllers. The authors correctly note that an issue for future monitoring solutions is overcoming the challenge of controller dependence but their solution currently requires significant manual input by network engineers which significantly hampers its ability to scale. Finally, Yang et al. [104] virtualise a basic switch and implement a network traffic monitoring system showing that the virtual switch can act in place of a conventional managed switch, without the need for port mirroring. However, Yang's work has weaker performance than traditional hardware approaches but does offer enhanced flexibility through virtualisation.

Each of these tools offers a specific monitoring functionality, such as flow or link utilisation monitoring, or suite of functionalities as part of a wider framework. While these tools are built on open standards, they are tied to individual SDN controllers or complete implementation architectures making their widespread use and capability to be augmented and combined into an overall system much harder. The implementation and research into network monitoring solutions with the current shifts in how networks operate is still a young field. This

work contributes a function deployment architecture with the opportunity for both re-use of well-established and researched network monitoring tools and modern open innovation.

2.3.4.2.1 Anomaly Detection Anomaly detection is a mature and heavily researched field. While numerous SDN+NFV architectures have been reviewed and network functions they can support, there are very few examples of such architectures coupled with context-specific anomaly detection VNFs and none to the best of the author's knowledge in the ATM environment. Highlighting the need for revisiting well-researched ideas in the SDN+NFV architecture is the work of Braga et al. on Distributed Denial of Service (DDoS) [105] detection. Their lightweight DDoS flooding detection exploits OpenFlow's flow table management for efficient classification for anomalies within the network. In work with a similar concept of leveraging SDN, Giotis [106] presents a sketch-based anomaly detection method with OpenFlow based mitigation techniques to improve network survivability under attack. Mehdi [107] continues this trend with the implementation of four anomaly detection algorithms in an SDN controller. This NOX-based architecture can detect flooding and port scanning attacks at a centralised point. These centralisation methodologies for anomaly detection are more suitable for Small Office/Home Office (SOHO) networks than large-scale enterprise networks. This is because of the scale of enterprise networks, which make distributed anomaly detection approaches more beneficial due to the sheer volumes of information and latencies of longer distances. Highlighting the benefits available through the chaining of Virtual Network Functions, is the assertion that following many years of anomaly detection research it has become clear that different techniques can identify different forms of anomalous behaviour. By combining algorithms, better overall detection results can be achieved, giving a fuller impression of behaviours observed. For example, the ASTUTE algorithm [108] complements the Kalman filter [109] since the former examines groups of flows which simultaneously increase or decrease their traffic. The latter can better detect anomalies involving a few large flows, which ASTUTE cannot detect.

2.3.4.3 Resilience

Network resilience is the ability to provide and maintain an acceptable level of service in the face of faults and challenges to normal operation [110]. ATM systems are incredibly resilient with strong diversity, redundancy and over-provisioning across all aspects within the wider system of systems. The FTI service designs have strict diversity regulations[21] and require path switching capabilities of all FAA equipment as well as the RMAs discussed earlier. In previous work [111], the author examined the relationship between ATM power resilience and ATM network resilience observing that the challenges faced in communications resilience are greater [112].

Power supplies are source-independent, with energy needs being met from mains, back-up generators, and even fail-safe battery power if the prior sources fail. Voltage spikes or brown outs can be mitigated using standard techniques such as Diesel Rotary Uninterruptible Power Supplies (DRUPS) which convert the power to a steady, clean stream in terms of phase, harmonic distortion and consistent voltage. Figure 2.17 shows the mains power resilience measures, including DRUPS in place at the NATS Area Control Centre in Prestwick.

Communications resilience is a similarly sophisticated problem as it is time-critical for real-time applications and the content is critical. Communication payloads are a complex, non-Poisson process of traffic load and arrivals. This implies high variability and unpredictable dynamics over long timescales. Traffic peaks can also be significant in terms of utilisation, even with substantial over-provisioning. Data packets are also susceptible to corruption, loss and delay, which Muller et al. [113] describe as the three *network-safety hazards* also stating there are very few ATM network-level research publications. With these additional factors for networks there is therefore strong motivation to place a greater emphasis on communications resilience to ensure the same levels of reliability and availability as are present in power resilience. There are many more considerations for network resilience, discussed in depth by Fry [114] and Cholda [115].

[21] `http://www.faa.gov/documentLibrary/media/Order/ND/6000.36A.pdf`; Accessed: March 2016

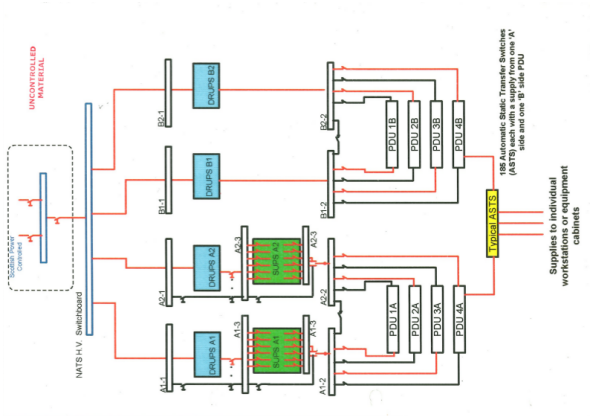


Figure 2.17: NATS Prestwick ACC Power Redundancy

The ATM approach to resilience is for the survival of the overall system function versus individual aspects. For example, if a radio breaks, there may be no need to replace the radio as there are alternative radios available and multiple backup communication links such as telephone landlines and mobile phones. At Prestwick ACC and similarly in other ATM facilities worldwide, for each networked device, there are two network connections. The networked devices such as the intelligent controller display units are themselves over-provisioned with redundant display units available should an individual piece of hardware fail. There are fully replicated data centres and the network topologies with redundant links are designed to share the service load under normal conditions but both are capable of carrying all of the bandwidth if one link fails or requires maintenance.

The ResiliNets [110] networking project has done significant prior work in network resilience including the introduction of resilience envelope graphs. These graphs show the upper and lower bounds of a specific resilience metric plotted over time. From these envelopes, iterative design choices can be explored, such as topological changes, to determine their impact on resilience. The resilience envelope concept is a component of the larger ResiliNets strategy of $D^2R^2 + DR$ (defend, detect, remediate, recover, diagnose, refine) which is discussed more comprehensively along with principles and approaches in [46, 116]. The core idea is to have real-time network resilience through successful iterations of each of these high-level concepts. Alongside the ResiliNets strategy, it is worthwhile to note the work on

the ironies of automation by Bainbridge [117]. Bainbridge highlights that automated remediation and the expectation for humans to take over from automated systems in times of severe duress is very challenging for human operators. The argument presented is that it can be less helpful for overall resilience and survivability to have such automation than remaining without automation and ensuring operators are more continually aware and involved.

2.4 Summary

This chapter examined and collated related work in the ATM and networking fields. To better understand the context of the ATM environment, an overview of the evolution of the service to date was presented along with international forecasts and trends for the future requirements of global ATM systems. The characteristics of ATM services were also discussed alongwith technical details of existing ATM data network infrastructures. With 2.7 million non-hobbyist UAVs forecast for 2020 in the U.S., integrating unmanned service support into ATM infrastructure was another key observation. Findings from official national audits of ATM data network infrastructure were also aggregated showing many challenges including the inflexibility and lack of adaptive monitoring capabilities for operators, poor recovery times in the face of outages, and insufficient monitoring and anomaly detection capabilities. This analysis was followed by a review of the latest developments in network softwarisation technologies, SDN and NFV. The motivation and high level concepts behind each of these paradigms were presented and discussed before exploring their application in a range of fields including container-based virtualisation, network monitoring, resilience and anomaly detection. This chapter has served to inform on the context and broader environment of this work as well as discussing the foundation of previous research on which this work builds.

Chapter 3

Design Considerations for Future ATM Data Networks

3.1 Overview

For increased resilience and flexibility, the architecture for modern Air Traffic Management networked systems should exploit state-of-the-art networking research. By enabling network programmability and virtualisation in particular, ATM systems can be more responsive, addressing any weaknesses faster, through software updates. With the advent of major ATM infrastructures now making the transition to IP-based networks through programs such as NextGen [118] and SESAR [119], there is an excellent opportunity to take advantage of such design choices.

With national Air Navigation Service Providers worldwide seeking to optimise their infrastructure in the face of increasing demand, and explosive growth forecasted in unmanned aerial systems, measures to maintain current and improve future levels of safety and resilience must be designed for, within the modernised networking infrastructure.

Programmability and virtualisation offer numerous design advantages and help meet many of the design goals required to achieve these aims, mitigating current bottlenecks such as time-consuming physical testbed reconfiguration, and constraints such as limited budgets

and the risk of change.

This chapter describes the key design goals and decisions in order to achieve the core aims of this research. Section 3.2 explores the design alterations suggested by recent major motivational incidents which have occurred in global ATM systems. Investigating the expert opinion and recommended outcomes from such incidents ensures this work is based upon the latest recommendations for improved resilience and safety. Section 3.3 looks at the design consideration for UAV systems and their future demands on ATM services. Section 3.4 presents the design of the architecture, with subsections examining the core design elements with respect to the requirements, operational overviews for both ATM systems and UAV systems including a Concept of Operations, before presenting the design choices of SDN and NFV. Section 3.5 summarises the chapter.

3.2 Prior ATM Networking Incidents & Recommended Outcomes

There are many examples of network systems failure and technical problems causing disruption in ATM services. Thankfully, it is rare for accidents to result from these incidents, due to other safety measures in place. However, that does not reduce the seriousness of the potential safety implications caused by these network level problems. Further motivation is evident from these failures; their often slow detection and diagnosis caused enormous disruption and therefore financial loss, including loss of revenue and the cost of engineers, in one case prolonged over seven weeks.

The following six incidents, in the sections below, highlight the severity of the impact faced when challenges do undermine the resilience of ATM data networks and therefore the safety and security of the services which rely upon them.

3.2.1 2009 FTI November Outage

The FAA Telecommunications Infrastructure experienced a five-hour outage in 2009 causing disruption for over 800 flights [120]. The investigative report found that 129 U.S. air traffic facilities experienced an outage that delayed thousands of travellers. The incident was a series of cascading events which culminated in failure. Earlier scheduled maintenance led to a FTI routing table (which directs air traffic data, such as flight plans, through the network) at the Los Angeles centre being programmed incorrectly. This was inactive until it was restarted. Independently, an automated tool that alerted engineers when a router CPU exceeded 60% utilisation over 10 minutes had been inadvertently disabled for all routers. This was due to a configuration error that was intended to suppress the alarm for a specific router at FAA's technical centre, but which caused the alarms for all routers on the FTI network to be silenced. The lack of an alarm system compounded the routing error and a significant delay occurred while network engineers manually probed the network to localise the problem and eventually determine which router was at fault. The error caused the router to send air traffic network data on the wrong paths, which blocked approximately 75% of the routes across the FTI fibre optic network. The report found that the FAA needed to be more proactive in assessing and addressing FTI network vulnerabilities. An internal FAA study was also cited, stating both the FAA and the contractor confirmed either the complete lack or inadequate proof of diversity between FTI primary and alternate network paths at several critical facilities.

The recommendations following this incident included [120]:

- Use of modelling, simulation, and network monitoring tools to examine failure mode simulation, routing configuration changes, and alarms for unexpected and significant routing changes.
- Assign staff resources to identify vulnerabilities, recommendations to improve survivability, and research into new and improved methods of building high-availability networks.

Both the FAA and the contractor stated that there is still a risk of critical outages as new NextGen services are added to the FTI.

3.2.2 2007 Los Angeles International Airport (LAX) Disruption

In 2007, the U.S. Customs and Border Protection computer systems at LAX suffered from a network outage [121]. The fault analysis concluded the single initial point of failure was caused by a malfunctioning Network Interface Card (NIC) [122]. This in turn caused data to overload the system, leading to average response times of computer systems at 2-3 minutes, two orders of magnitude greater than the usual durations of less than 5 seconds [123]. The 10-hour-long issue caused delays and congestion affecting ca. 17,000 passengers with new arrivals not being allowed to disembark and international departure disruptions. Analysis suggests this incident had flooding characteristics which propagated through the network. This case highlights the enormous impact network problems can have, the difficulty in identifying the problem and the length of time taken by manual recovery methods. The recommendations made by the Department of Homeland Security included the introduction of 'automatic error notifications' and that staff should be more effective when isolating and resolving outages. Finally, the report stated belief by staff that there is a high risk that a similar outage could occur at other U.S. ports of entry.

3.2.3 2012 FAA Technical Centre Fire

In 2012, a fire caused severe disconnectivity at a core FAA Point of Presence (PoP) at the FAA Technical Centre in Atlantic City [124]. The building was evacuated and caused some air traffic and flight planning systems in the U.S. to become temporarily unavailable. The ATM service was significantly slowed. Back-up systems relied on telephones for communications. This incident was well-managed but highlights that despite high levels of redundancy, disconnectivity is a very real threat and the challenge of continuing operations seamlessly in the face of disrupted infrastructure connectivity cannot be guaranteed. Greater

physical diversity was a key outcome from the network analysis of this incident, since many redundant services were located at the same site.

3.2.4 2008 Major European Airport ATM Shut-down

In 2008, a European airport ATM system experienced a very disruptive and prolonged networking failure. While this incident had a very similar cause to the LAX network failure, the effects had a greater impact upon ATM services. The root cause of an intermittent faulty NIC and the subsequent error in systems attempting to mitigate the issue led to anomalous manifestations including ATM losing track of aeroplanes or associated flight information at several different times. The problems persisted for seven weeks with sustained durations of normal behaviour. These ongoing problems caused delays, restrictions in air traffic such as ceasing to place aircraft in arrival holding patterns in case a problem occurred, irregularities in information presented on ATCOs' screens, and some periods of complete closure [125]. The recommendations by the commercial ATM system supplier, and the national aviation authority were 'that additional network monitoring be undertaken and that monitoring tools and a passive analyser be installed on the system to aid the early identification of similar malfunctions' and 'the possibility of other potential improvements in the network design in order to prevent a re-occurrence' [125]. Adding network monitoring was central to the detection of the faulty NIC. Overall, the recommendations from this investigation and the requirements to detect the problem were similar to the LAX fault recommendation that more network diagnostic tools should be available.

3.2.5 2014 December UK NATS Outage

On December 12th 2014, the computer system used to provide data to ATCOs to assist their decision-making for air traffic flying at high altitudes over England and Wales failed. NATS estimated that a maximum of 1,900 flights and 230,000 passengers were affected on the day of the incident with cascading impacts through cancellations causing approximately 60 aircraft and 6,000 passengers affected the following day. The official incident report [126]

states the problem was related to a latent software error that was present for over 20 years prior to the incident. The issue lay with the software's check for a maximum number of air traffic controller and air traffic controller supervisor roles. The software limit should have been for a combined maximum of military and civilian roles. However, the software check did not account for the military limit, yet included these roles in a count against the civilian limit. This total exceeded the limitation for the first time, due to the addition of new military controller roles the day before. Problems which followed on from this software error compounded the impact of this problem, leading to the widespread disruption. The report praised engineers for their rapid detection of the issue, in such a large software system. Numerous recommendations came from this incident. The most applicable to this research was that NATS should consider introducing a formal system to capture anomalous occurrences that fall below safety thresholds. With respect to the introduction of SESAR, it was recommended that NATS retain the capabilities to provide resilience in the presence of hardware and software failures and operator errors associated with configuring the system.

3.2.6 2015 New Zealand Nationwide Radar Outage

In June 2015, New Zealand's ATM systems suffered an internal network failure which resulted in the loss of radar coverage across the country's airspace for over 4 minutes [127]. An incident report into the causes is currently underway by New Zealand's Transport Accident Investigation Commission. Around 50 flights were airbourne when radar communications were lost, with 160 flights impacted by further delays. ATCOs resorted to manual paper-based methods to land all flights, with take-offs cancelled nationwide. Minimal details are available to the cause of the incident but the Chief Operating Officer for Airways, New Zealand's ANSP, stated the fault was identified within about 15 minutes and resolved by taking that particular bit of equipment out of the system. With recommendations still forthcoming, it is clear the resilience of the overall system was intact, however radar systems resilience appears to have lacked diversity.

3.2.7 Discussion

The incidents reviewed are a selection from many similar incidents worldwide. Incidents such as these, and the rapidly escalating disruption of air traffic after such outages, highlight the importance of good resilience engineering with a holistic perspective. Examining the improvements that are necessary in current implementations to meet existing service requirements in the face of evolving challenges provides a strong foundation for the design of this research. The recommendations stemming from the incidents above each tend towards common themes of better monitoring, anomaly detection and configuration management. These findings are inline with the conclusions of an internal FAA report [5], which found that monitoring of the ground stations, communication between facilities, and training related to outage response could all be improved. Overall ANSP engineers have stated:

- they lack knowledge or models of normal operating behaviour;
- there is insufficient monitoring;
- there is no way to easily adapt the monitoring or detection systems in place to probe the network for diagnosis or detection.

There was also often no standard for reporting network outages with current processes focusing on the outages but not sufficiently identifying specific operational impacts. Finally some engineers at various ANSPs reported there was no test capability that mirrors the network backbone and can simulate the application traffic mix.

In each of these incidents, Reason's widely cited Swiss Cheese failure model [128] applies since these were not single points of failure, but a series of problems which involved a technical root fault. In these cases, while network-level problems are unavoidable, aspects of network automation such as monitoring with automated analysis and notification of anomalies could stop such problems persisting and remaining unknown, and instead be detected very quickly or as emerging trends for continuing monitoring. These incidents are rare. However, with systems complexity increasing through efficiency and the introduction of UAS, it is important that currently levels of safety and security are maintained.

3.3 Unmanned Aircraft System Integration Design Considerations

With explosive growth in UAV numbers forecasted worldwide, a core concern is how to manage the ad-hoc network configuration required for mobility management for UAS. For safe and secure operations, UAV operators require real-time telemetry monitoring, alert systems, and mission payload processing as part of the Ground Control Systems. This information is especially critical for BVLOS operations. Increasingly, there is demand for tailored functionality which can assist the operator with the current task, particular environment and payloads, with monitoring specific to the payload, e.g., visual surveillance equipment, heat mapping or crop dusting tools. Since long complex operations can involve multiple tasks, environments and strains, flexibility and both reactive and pro-active adaptability of this functionality over time are also highly desirable attributes. Significant replication of standard telemetry-based monitoring functionality is prevalent, yet with isolated, independent implementations, spread across different vendor-specific ground control systems this scale cannot be beneficially exploited. Middlebox functionality for UAVs is a vital next step to increase the overall resilience of the wider UAS, necessary for integration with controlled airspace and operations over populated areas [129].

3.4 Future ATM Network Architecture Concepts

3.4.1 Core Design Elements

To achieve an architectural design that is fit for purpose, the related research has been reviewed, current architectures studied and recommendations from serious outage incidents and internal ANSP infrastructure reviews have been analysed. Findings from this primary research are central to the design of this work in order to ensure that flexible and readily available network monitoring, detection and mitigation tools are available for operators on-demand. From these findings, the following design requirements have been distilled:

1. Engineers seek greater network monitoring capabilities, with an ability to probe the network on-demand, seeking further information on specific aspects of the network infrastructure.

2. Anomaly detection tools should be readily available and running on the network infrastructure with an alerting and notification system which can be integrated with successful service monitoring systems already in place.

3. Mitigation methods should be available for engineers to take actions in the face of challenging incidents. This mitigation should refrain from use of significant automation so as to avoid compounding problems and complexity, and instead be a suite of manually deployable tools by engineers in times of need.

4. Speed and flexibility of detection, monitoring information and mitigation is imperative to the safe and secure operation of ATM, especially when under unknown duress.

5. In-network devices, middleboxes and intelligent switches (with capabilities for logic based routing decisions) are readily available within existing infrastructure implementations and can therefore be utilised in future architecture designs.

6. A reduction in test-bed configuration dependencies and alternative means to test new deployments with accurate application traffic mix would be advantageous to risk mitigation for innovation.

7. For resilience, distributed architectures with logical centralisation and physical distribution are a necessity.

Many of these aims and objectives can be met by utilising a combination of SDN and NFV technologies. Through network softwarisation, engineers can build greater, adaptable network monitoring capabilities and run these as virtual network functions on-demand. Mitigation can be achieved by exploiting network programmability to route a given (sub)set of traffic. This technique can also be used as an alternative means to test-beds. Network Function Virtualisation can also assist with mitigation, providing deployable solutions to

emerging challenges, for example via existing in-network devices such as middleboxes. The abstracted architecture in Figure 3.1, meets these high-level design goals and is the basis for the primary contribution of this work.

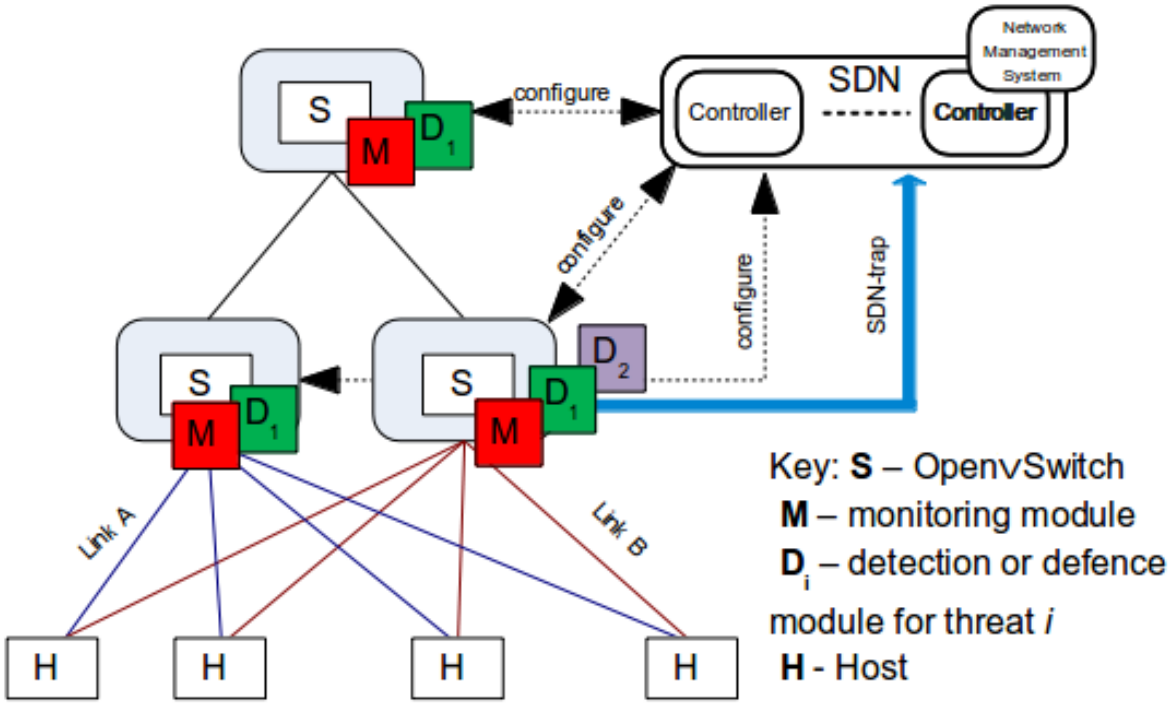


Figure 3.1: Abstracted SDN+NFV architecture design with modular programmable monitoring and detection

Figure 3.1 shows three SDN-compliant switches with in-network devices co-located where modular Network Functions can be situated. The figure represents a primitive hierarchical topology with four hosts connected via redundant A/B links, seen in red and blue. A logically centralised SDN controller connects to each switch to configure routing and the deployment of modular NFs. Routing can be configured to direct traffic through different deployed NFs (either individually, or in series), pass traffic on to another host or service, or to the SDN controller, for example, to manage traffic which does not fit current configurations or routing. Further services and integrations can interface with the SDN controller, such as existing Network Management Systems as shown. This work coins the term *SDN-traps*, which perform a similar action to predecessor *SNMP-traps*, in this case representing notifications from NFs to the SDN controller, such as anomalous behaviour. Network Functions can be Monitoring, Detection or Defence, i.e., remediation/mitigation modules. These lightweight

programmable modules can be deployed on-demand to various parts of the topology giving operators the ability to rapidly adapt and tailor their tools to the meet the needs of the current situation.

3.4.2 Operational Overview

Operationally, this architecture would provide a number of benefits in line with the previously-stated design goals. A core contribution is in the ability for operators to deploy in-network tools on-demand. With numerous in-network devices and middleboxes already situated in most ANSP topologies, these offer an excellent potential resource for hosting Virtual Network Functions. While many of these devices may be bespoke hardware such as, e.g., firewall devices, a transition from such bespoke devices to programmable in-network devices can gradually take place, with software providing the operations previously performed in hardware.

A suite of tested, well-establish deployable VNFs can be established and added to over time. ANSP engineers can share these tools and best practices for services running on the various interconnected infrastructures, such as ACDM in Europe. In Section 2.2.3, insufficient network monitoring was a major short-falling of current systems. This, coupled with the inflexibility of the current architecture that leads to poor recovery times in the face of outages and disruption is, according to accountability reports, directly limiting ANSPs' ability to detect and respond to ever-increasing security incidents affecting their mission-critical systems.

The architecture design in Figure 3.1 addresses these key issues, providing vital flexibility and the ability for engineers to deploy distributed network monitoring functionality to examine aspects of interest at different points within their live topology, without interfering with operational network traffic. This capability is also rapid due to software-based configuration and deployment, therefore without the need for engineers to individually calibrate or physically re-configure devices. With the ability to receive timely information, when challenges and disruption do occur, remediation strategies can in turn be developed with more informa-

tion and be available in shorter timescales. With a suite of pre-tested VNFs, remediation or defence modules could be deployed and their mitigation attempts monitored. This expeditious process to help detection, diagnosis and remediation is another core contribution of this work.

Developing and maintaining large realistic testbeds is another key challenge for ANSPs. Testbeds are costly and will not mirror the conditions of the production network precisely. To implement changes requires significant effort and testbed reconfiguration. With a Software Defined Networking based architecture, some innovation can be performed on the production network. By mirroring the real-world traffic application mix, and forwarding copies of live packets onto different portions of the network, realistic trials can be carried out quickly and easily without impacting the operational traffic flow. By exploiting virtualisation, different overlay networks and routing can be tested on the same hardware infrastructure, again without impacting live operations because of virtual and logical separation.

Shortcomings in boundary protection have also been highlighted as a key recurring issue [5]. Major security concerns have been raised with respect to the protection controls at the edges of ANSP mission-critical infrastructure. Security has ever-increasing requirements with evolving needs to meet the latest vulnerabilities reported. If bespoke, vendor-specific, hardware is used, software updates can take significant effort to maintain and install. The internal review also criticised the mechanisms in place at these edge and inter-system boundaries stating they were insufficient at protecting and restricting connections into and out of network-connected devices [5]. This architecture allows for boundary protection and edge switches to pass traffic through a set of easily-programmable VNFs, ensuring network traffic security measures can be kept up-to-date simply and promptly.

Configuration management and policy enforcement can also be more easily managed under an SDN-based architecture. By placing policy and configuration rules in the logically centralised SDN-controller, consistency, verification and validation of these rules can be achieved throughout the network. Updates to policy and configuration can also take place centrally, taking advantage of the automation this architecture offers, without the need to

remotely login to individual network devices and update their routing tables. The incorrect updating of routing tables, and device misconfiguration more widely, has frequently proven to be a significant contributing factor in numerous outage incidents which directly impacted overall systems and the ATM service's resilience and survivability.

It is important to recognise there are trade-offs with this architectural design with respect to complexity, security and performance. Complexity is a consideration with such a design. The increased complexity stems from introducing virtualisation, on-demand deployments, chaining network functions and a suite of different network functions. While the design adds complexity, advantages exist, such as the ability to deploy realistic tests and testbed simulations on the operation network infrastructure. Performance is degraded by using VNFs compared with hardware-based network functions. The first bottleneck is the virtual switch, as seen in Figure 3.1. The virtual switch must be able to provide sustained high-bandwidth traffic to the VNFs. By introducing the virtual switch, there is a drop in overall performance for the highest levels of throughput. The throughput degrades as the number of VNFs increase, as discussed further in Section 4.2.6.1. The second bottleneck is the VNF itself. There is a performance overhead which comes with virtualisation. However, the benefit is flexibility and the ability to adapt and change network functions on-demand.

Security is another trade-off to consider. As an example, if malicious access was achieved to the SDN Controller, firewalls could be disabled and other network functions could be deployed or disabled more easily. There is considerable research effort examining security for SDN and NFV contexts. One example is to encrypt the API traffic from the SDN Controller. Another is to have carefully designed policy management and access control management. For the context of ATM networks, there is the advantage of having the network behind edge routers and firewalls which gives additional protection to such a design.

3.4.3 Unmanned Aerial Vehicle Operations

UAV operations are becoming increasingly sophisticated and, as they transition from tasks predominately over low population density areas into controlled airspace, the need for a

better communications architecture becomes apparent. There are many interpretations of UAS environments. It is therefore important to define which UAS contexts are covered by the contributions in this work.

3.4.3.1 Current Concept of Operations (ConOps)

Figure 3.2 details the scope and environment of the Concept of Operations where the contributions of this function deployment architecture can be evaluated. The figure represents a typical military or environmental reconnaissance set up with multiple UAVs, of multiple types, operating in different environments, land, sea and mountainous regions with different payload capabilities, e.g., visual or IR cameras. There are mobile GCS on land, and at sea connecting and communicating control information to UAVs within range using radio antennas. Satellite communication links (and other, e.g., WiFi, microwave) are used to connect these mobile GCS with each other, in what can be viewed as an abstracted ad-hoc mesh network topology [130]. There is also a centralised master command and control centre located far from the operational area. Some of these links are very costly, e.g., satellite. Currently, in-depth telemetry analysis often takes place on data streamed back to a centralised control centre via these expensive links, or not at all [131]. Many of the UAV to GCS uplinks are unreliable leading to loss of data in the real-time streaming where packet latency and out-of-order delivery is also equivalent to data loss [132]. Lost link failures from the UAV to GCS are also common due to interference from particles (for example sand), out of range, etc [132]. As a result, deploying code to run on deployed UAVs over such links is a poor design choice.

In Figure 3.2, UAVs are transitioning from sea to land operations and from higher altitude mountainous regions to lower levels. During these transitions, different model-specific monitoring and anomaly detection modules can assist operators. For example, calculations at higher, colder altitudes for icing alerts. The ConOps also shows the lowest UAV transitioning from control on the leftmost GCS to the rightmost GCS. Telemetry streaming takes place from the UAV to the GCS, with a hand-over phase when a UAV migrates to the command

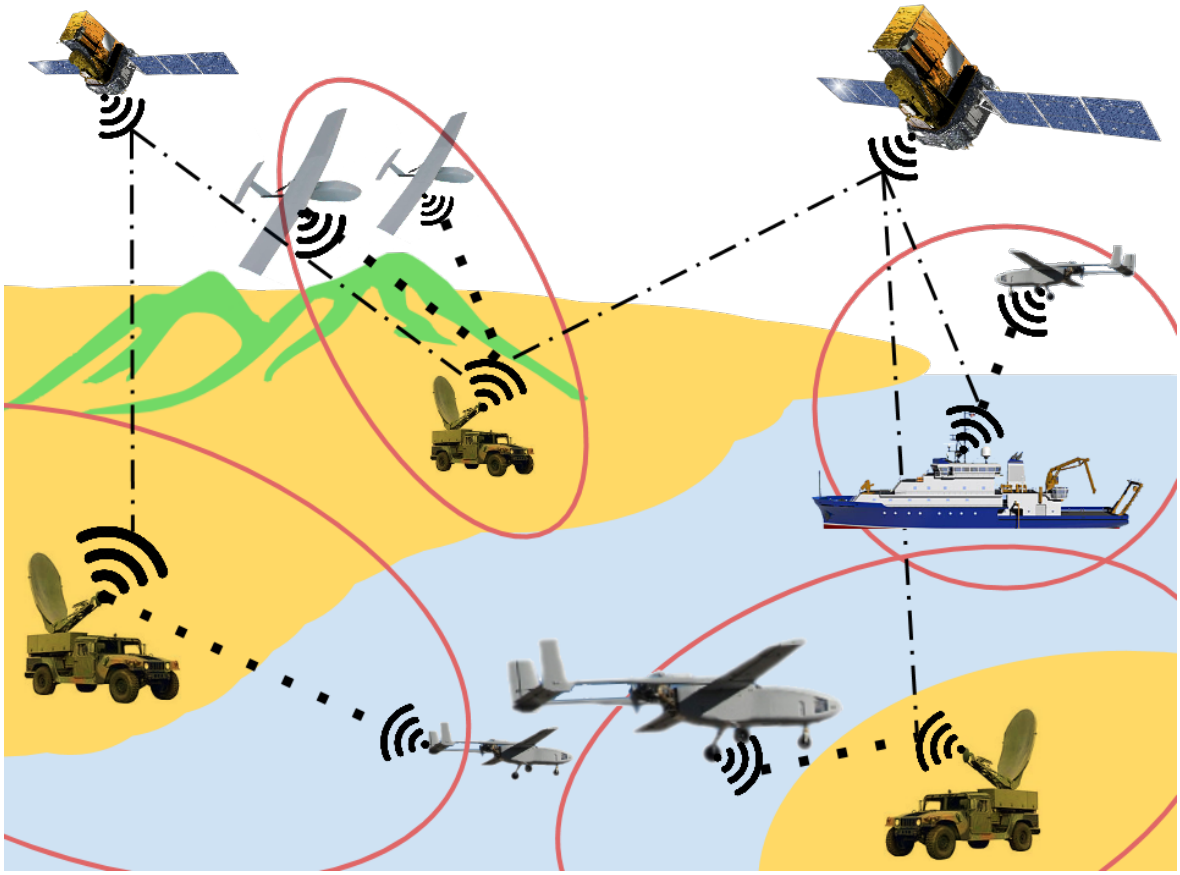


Figure 3.2: UAS ConOps for high-mobility communications infrastructure

and control of another GCS, e.g., due to a change in range or a primary GCS becoming unavailable. Autopilot systems are available to perform flight during the transition. Ground teams identify when they have resumed control and a hand-over is successful based on the telemetry they receive. In this context, the UAVs are the (migrating) hosts, the local GCS are co-located with switches and the command and control centre hosts the network controller.

The ability to integrate Unmanned Aircraft Systems with wider ATM systems is a vital next step for the design of any future Air Traffic Management architecture. As ATM moves from purely manned flight management to hybrid manned and unmanned flight management, an integrated and standarised infrastructure architecture will be a critical step in the safe path to innovation. This work carefully considers the current and future needs for the networking design of this future hybrid environment. Key design requirements for any future UAS communications architecture, and therefore a future UAS-integrated ATM communications architecture, are as follows:

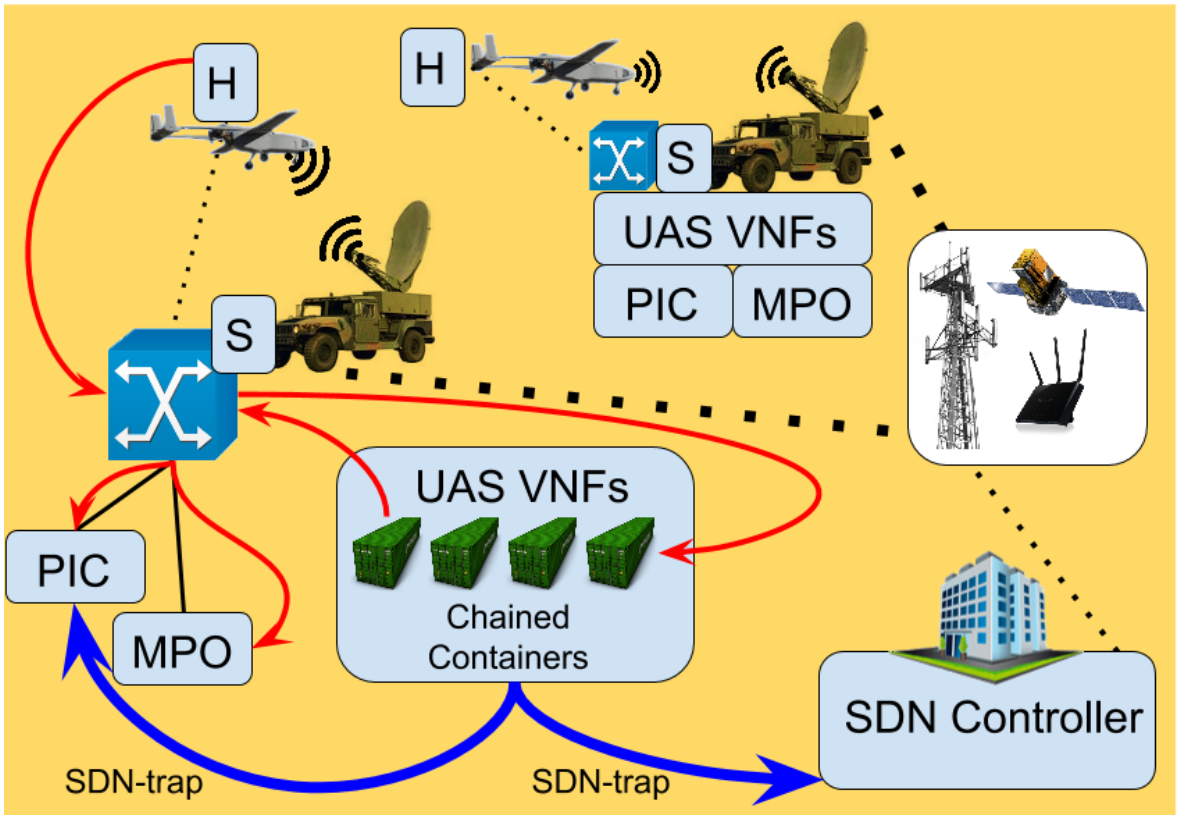


Figure 3.3: SDN+NFV architecture with programmability and virtualisation for UAS

- **High-mobility lightweight deployments** Deployment of NFs should be simple, transparent, and fast for the subscribing hosts to start a NF and redirect the traffic through it.

- **Distributed processing for lower utilisation and latency** Moving programmable, adaptable, modular processing from the command centres to the GCS reduces the traffic on the more expensive, and often strained backbone links from the remote GCS to the command centres. Streamed telemetry will no longer need to traverse these links, and processing NFs should be lightweight to migrate and deploy. Latency of anomaly detection will also be reduced. By placing detection nearer the UAV, anomalies are recognised at the GCS without first streaming over the often high latency links.

- **Increased resilience** Resilience can be improved through lower utilisation and lightweight deployments leading to less pressure on the topology and more capacity to absorb link failures, for example. Greater resilience can also be achieved through programmability and adaptability in the architecture. Operators can deploy context-specific NFs

on-demand in times of need to learn more and better understand the challenges faced to inform their decision-making.

- **Infrastructure independence** Traffic routing will be handled independently from default routing policies. This means that switching will be context-aware, allowing forwarding of traffic from hosts to Network Functions. Decoupling default routing from policy enforcement routing reduces the risk of misconfiguration of the individual network elements.

- **Open Innovation** NFs should be able to utilise the existing wealth of tools and programs available, e.g., for native Linux, without having to adapt these to work in a bespoke environment.

Figure 3.3 shows the architecture as deployed in the field. With the UAVs as hosts, the SDN-enabled switches are located at the mobile GCS vehicles. These switches route traffic from the host UAV to the Pilot in Command (PIC) and Mission Payload Operator (MPO) displays. The switches have software-defined routes to direct network traffic based on particular rules to the appropriate NF(s). If multiple UAVs of different types are operating from a single GCS, the switch should be able to be configured to route traffic from each UAV to a different set of NFs designed for the operating parameters of that model. Similarly, if NFs exist which are common to both model types, traffic should be able to be routed to the same NFs. The chained containers hosting the VNFs are situated at the GCS with the SDN controller located at the central command centre where it can be logically centralised and physically distributed for resilient oversight of the architecture. The PIC and MPO are also hosts in the network. The architecture can therefore also be configured through software rules to route traffic from the GCS hosts to the UAV to go through NFs. For example, access control or additional security measures could be deployed in sensitive environments for protection against replay or DDoS attacks via firewalls and rate limiters, for example.

Modular NFs allow for smaller functional components to act independently. For example, an NF could be running on a relatively inexperienced pilot's GCS to monitor the number

of commands sent. If this NF threw a series of anomalies, the wider GCS team or central command centre could deploy another NF configured to watch for anomalous pilot command sequences through, e.g., frequent repetition of a set or individual command. This could help diagnosis if the UAV was unresponsive or if human factors such as anxiety were involved. The ability to monitor and detect all issues simultaneously is infeasible due to the processing and storage capabilities available, especially in mobile remote environments. By chaining NFs and allowing for real-time updates, the processing and hardware available can be used to host a vast array of context and model-specific Network Functions which can inform and alert operators.

3.4.4 SDN+NFV breakdown

To better understand the contributions of this work, it is worthwhile to break the design of this SDN+NFV architecture down to see the benefits of each aspect, as well as the novelty and advantages added through their combination with one another and the aforementioned design goals. To provide an overview of how these technologies meet the requirements in Section 3.4.1, Table 3.1 shows which requirements are met by SDN, NFV and SDN+NFV.

Technology	Requirements fulfilled
SDN	5, 6, 7
NFV	2, 3
SDN+NFV	1, 4

Table 3.1: Requirements from Section 3.4.1 fulfilled by SDN, NFV and SDN+NFV

3.4.4.1 SDN benefits

The dominating outcome from the application of the SDN paradigm is the resultant programmability, and therefore flexibility, of the network. For ATM, this programmability encapsulates the flexibility of routing decisions, configuration and centralised policy control. SDN was created predominately for enterprise networks with applications in data centres yielding significant research attention. ATM networks suffer from many similar problems

that SDN can solve. Through separation of the data and control planes within the network, the hardware based engineering and overall resilience management for these mission-critical networks can also be logically decoupled. While there remains a need for the physical data plane components to function and have strong uptime, reliability and robustness, choices regarding network flow such as, e.g., alleviation of over utilised routes or changes in routing policy can each be taken at an abstracted perspective above the low-level routing details.

An architecture based solely on SDN could tackle some of the key design goals stated in this chapter, such as mitigation methods. When incidents are detected such as loss of connectivity or insufficient bandwidth, SDN routing could be used to divert traffic to alternate routes, for example, based on traffic type, providing a means of priority-based routing. This form of QoS would offer a solution to the issues raised by Newell et al. [17] regarding severe weather alerts such as wind shear notifications, being enqueued in packet buffers behind less critical bulk general weather updates. Resilience design goals would also be met with SDN providing excellent means through programmability to offer logical centralisation and physical distribution of architectural components.

The introduction of SDN can significantly assist with the infrastructure challenges of migrating UAVs in a multi-GCS context as defined in Section 3.4.3. With programmable switches able to inform logically centralised control of new packet flows from new host UAVs, the controller can install the appropriate routing rules for that UAV on-demand. This allows for the simple establishment of routing policies based on UAV hosts, with traffic flows being able to be routed to their appropriate destination, for example, based on the payload traffic.

3.4.4.2 NFV benefits

Network Function Virtualisation offers the ability to define in software, traditionally static or difficult to change network functions. These network functions, often with specialised hardware, leave little scope for adaptability or responsiveness. Deploying updates can also be a non-trivial process, waiting for vendor release cycles and ensuring hardware compatibility. NFV provides ATM network operators with the ability to deploy a series of in-network pro-

cesses on-demand. However, without an SDN component in the architecture, the locations, arrangement and order of these VNFs are fixed, limited to where the hardware middleboxes, in-network devices and intelligent switches are positioned within the topology. NFV does offer the opportunity to deploy NFs on-demand, but the impact of this capability would be severely hampered without the simple routing configuration provided by SDN.

A principal design need for UAS is numerous monitoring-specific tools for individual UAV payloads such as, e.g., cameras or sensors. Virtual Network Functionality can provide the ability to have different, easily changed, monitoring tools designed for a UAV's payload. NFV can also help towards greater standardisation. Currently, different vendor software for operators and controllers provides similar isolated functionality, such as monitoring fuel levels. NFV can be used to create a suite of VNFs which support the different vendor protocols providing a means to have standardised information displays across different UAV makes and models.

3.5 Summary

This work combines SDN+NFV principles in such a way as to exploit the benefits of each paradigm, and tailors the architecture to the ATM context. By enabling programmability and virtualisation, this work offers a feasible and powerful solution to the design challenges of both ATM today and in the future, with the integration of Unmanned Aircraft Systems. With SDN+NFV, operators will be able to deploy additional network monitoring, detection and defence capabilities, with the ability to probe the network on-demand, and have control over where the VNFs are deployed, in which order and when. The design of this architecture will ensure safe advantage is taken of automation, through automated SDN updates to routing tables on-demand, yet with the restraint to involve engineers in taking the decisions on which functionality to deploy and where. The architecture will be flexible, with lightweight processes able to be chained in series or act independently. This architecture meets the need for greater capabilities to respond to incidents, similar to and including those reviewed in this chapter, and is feasible to implement based on the hardware and characteristics of current

ATM infrastructure. These infrastructures currently make significant use of in-network devices, middleboxes, and intelligent switches and routers. Traffic patterns are also stable, and topological changes are relatively infrequent. Overall, monitoring capabilities of the ATM system can be improved through this design. This work, through its function deployment architecture, gives ANSP engineers the ability to gain greater network service and infrastructure visibility, on-demand.

Chapter 4

Implementation

4.1 Overview

Following the design outlined in the previous chapter, the architecture was implemented. This chapter shows the technical implementation details for the various aspects of the functional deployment architecture, including hardware considerations, where appropriate. The chapter gives an overview of the architecture before describing the routing, SDN Controller, APIs and User Interface. Details on the NFV Servers and sample NFs are also discussed. Without the means to develop using large-scale testbeds the implementation utilises Mininet [133]. The details of the full architecture implementation are documented in Section 4.2 and Section 4.3 explores the implementation of sample ATM NFs. Section 4.4 summarises the chapter.

4.2 System Implementation

4.2.1 Implementation Outline

The implementation of the full architectural design is presented in this section. The ability of ATM engineers to instantiate or create new network functions and deploy them in minutes

and hours instead of weeks and months provides unprecedented agility as well as increased resilience in the face of challenging events. Figure 4.1 shows a detailed overview of this implementation on a basic, abstracted topology. Three OpenFlow (OF) switches comprise the network topology in the diagram orchestrated in a simple tree. The similarities to the design presented in Figure 3.1 can be clearly seen. The implementation presented in this work relies on the collaboration between the *Router*, *Manager*, *Agent* and *User Interface (UI)* to provide a global view and control over the infrastructure. The *Controller* comprises: the *Manager*, a Python module called RYU (*Japanese for flow*), and the *Router* which provides a set of REpresentational State Transfer (REST) APIs to globally control the network functions. The *Controller* also continuously collects health status of hosts and network functions and notifications raised by NFs. The decoupling of the coordination logic (*Manager*) from the operational logic (*Agent*) allows for a more flexible orchestration of the infrastructure by delegating the responsibility for placement and routing to the *Manager*, and the network function implementation and operation to the *Agent*.

The UI is a web application that communicates with the Manager API and the northbound interface of RYU. The web display shows topological, health and status information for increased network visibility. Real-time measurements are also shown, where available, for specific Network Functions. The UI also allows operators to manage Network Functions deployed for one or more hosts. The network status is continuously updated to reflect the current state of the network and to alert engineers of new notifications such as, e.g., anomalous behaviour, raised by one of the running NFs.

The architecture is infrastructure-independent. There is no bespoke underlying hardware or topology other than an OpenFlow-enabled network ensuring that adoption across numerous political and authoritative Air Navigation Service Providers is possible. Routing does not depend on the technology used on the end-host and is applicable to bare metal machines, Virtual Machines or containers, so long as each appliance has a routable IP within the infrastructure. The network functions can be run on any Linux/Unix host or in-network device running an active server. Various existing or new technologies can be used to implement

network functions; Linux-based containers utilising common networking tools such as iptables[1] or tc[2]. Through the decoupling of hardware and software, vendor lock-in can also be alleviated while procurement and deployment costs are significantly reduced. By routing only the required traffic through a network function, operators can manage traffic at different levels of aggregation (e.g., from per-service to per-machine, such as a specific Radar site, to per-subnet) in a multi-tenant environment. The ability to route only the required subset of traffic through an NF simplifies the development and management of new network functions as well as reducing the risk associated with the deployment of new functions ensuring traffic isolation throughout the architecture. ATM systems can be considered to have multi-tenancy because of the number of different services and systems sharing the common infrastructure. Future ATM systems can accommodate the benefits of this architecture irrespective of the low-level technology choices of independent ANSPs due to the flexibility of this implementation.

This section presents the implementation of each of these architectural components in more detail with Subsection 4.2.2 detailing the traffic routing, Subsection 4.2.3 explaining the controller and APIs, and Subsections 4.2.4 and 4.2.5 reviewing the implementation details of the NFV Servers and User Interface, respectively. Throughout this section the key design aspects are referenced with respect to the implementation.

4.2.2 Traffic Routing

Typically, middlebox policies are enforced in two ways: Either place the middlebox directly in the traffic path or add dedicated routing entries to redirect traffic to the middlebox. The first allows the middleboxes to be placed on the shortest path. However, it requires infrastructural changes and can result in poor flexibility as the policy will be applied to all traffic. The second allows more flexibility as an arbitrary policy chain can be configured through custom routing entries, at the cost of a longer path and overloaded routing tables, making it hard to configure and maintain [134].

[1] `http://www.netfilter.org`; Accessed: August 2016
[2] `http://lartc.org/manpages/tc.txt`; Accessed: August 2016

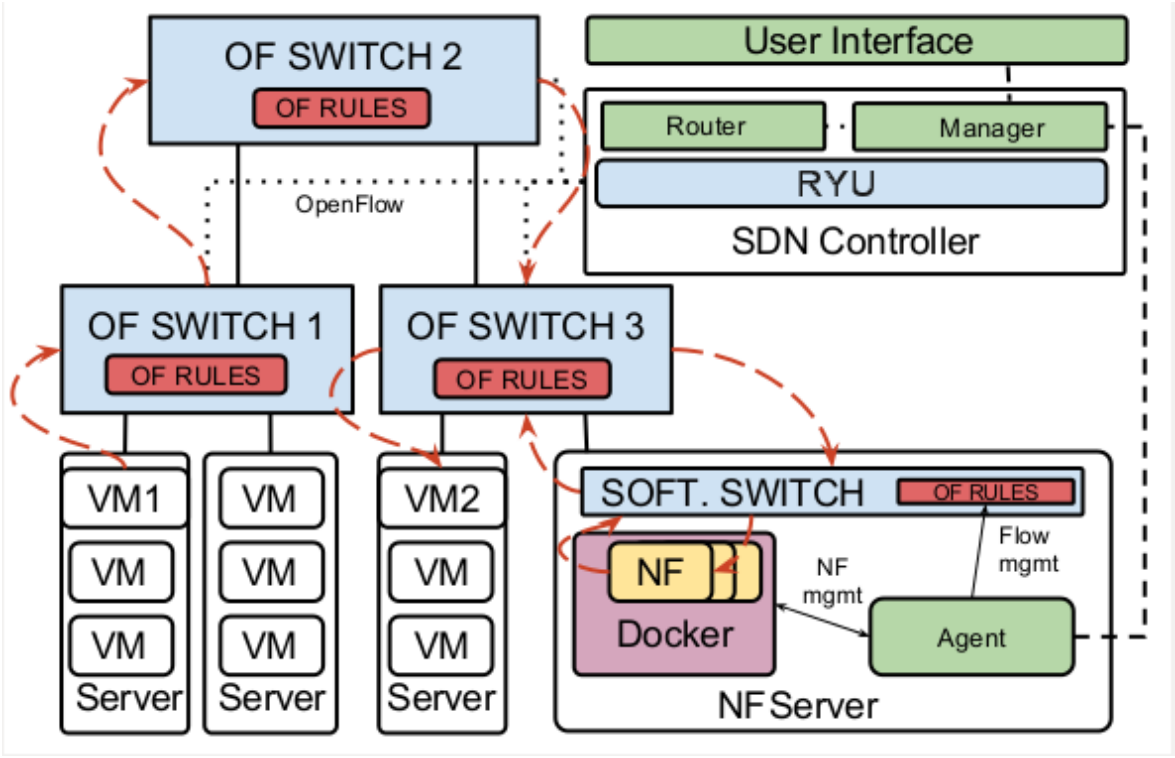


Figure 4.1: System architecture diagram highlighting key components and traffic routing shown with dashed arrows

The approach of this work, is to reroute applicable traffic to the relevant network functions, where applicable traffic and relevant network functions are determined by the configuration applied by the engineer through the UI. This approach enables dynamic placement of the NFs and uses the same hosts for compute and network functions, reducing equipment costs and machine specialisation. A consequence of traffic redirection is the potential use of non shortest path routing between source and destination possibly impacting performance. However, in ATM networks, in-network devices which could suitably host dynamic NF deployments are prevalent and typically on the shortest path for most services. Certainly, the impact is minimal as long as the number of extra hops is kept low by placing the NFs close to their associated VM [135]. Given the large distances in ATM topologies, it is important to measure closeness as both the distance as well as the number of hops.

OpenFlow is used to match and forward traffic to a NFV host. Routing is handled by matching on input port, source IP and source port, depending on the routing policy used, and forwarding matched packets to an output port on the switch. As packets are never modified,

routing through a NF is fully transparent to the end-hosts. By having a centralised control plane with a global view of the network, the input and output ports for the OpenFlow flow entries can be retrieved and potential problems of manual route (re)configuration in large-scale middlebox deployments [136] can be alleviated.

In this architecture, the *Router* is responsible for routing traffic to the *Servers* hosting the network function on in-network devices. The separation of the default routing policy from the Network Function routing, is achieved by using high flow priorities, allowing this work to be deployed on any OpenFlow-enabled infrastructure without altering normal operation or changing any flows already installed. However, to the best of the author's knowledge, no ANSP is currently exploiting the benefits of OpenFlow enabled networking on their infrastructure.

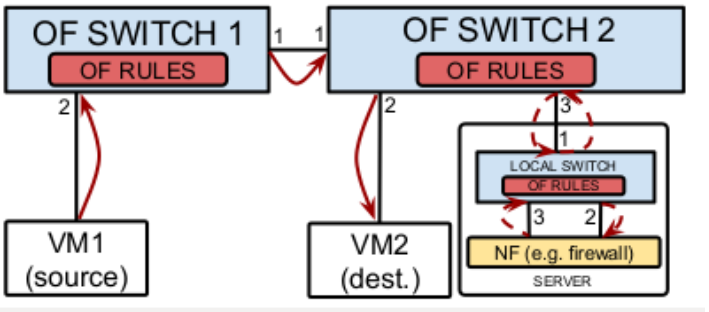


Figure 4.2: Detailed view of traffic routing from source to destination via Virtual Network Functions

Figure 4.2 shows the default traffic path using a shortest path, shown as solid arrows, going from source VM1 to the destination VM2 through two OpenFlow-enabled switches. On the use of a network function, traffic must be redirected to the NFV server and to a particular NF, depicted by dashed arrows in Figure 4.2. To achieve this, the OpenFlow rules shown in Table 4.1 are inserted to the switches. Since default routing is still in place, there is no need for extra rules to route traffic at the egress of the NFV server to the destination, from *OF SWITCH 2* to VM2. For clarity, only the forward path from source to destination is shown. The reverse path from the destination back to the source is a simple inversion of the ports in Table 4.1.

OpenFlow switches have complex trade-offs between performance and scale. This is largely

related to the hardware involved. Content Addressable Memory (CAM) enables memory lookups as a binary process, returning a match or not. Ternary Content Addressable Memory (TCAM) lookups extend CAM to match with '1', '0' or 'x', where 'x' is colloquially considered 'don't care'. This 'x' match corresponds with the OpenFlow wildcard bit '*'. TCAM can have multiple matches and determine the preferred match. The TCAM used for partial flow matching is small in size, capable of holding only a few thousand (2000-4000) flow entries [137]. It is typically collocated with a highly-specialized table holding 100,000+ entries for Media Access Control (MAC) address matching [138]. As the number of network functions increases, the growth of the flow table needs to be considered. Using only the TCAM, a single switch can redirect traffic to a maximum of 1000 NFs, as two entries are required per switch on the redirected path. However, only 2 flow entries are required for a collocated service chain, regardless of its length, or multiple hosts under the same Classless Inter-Domain Routing (CIDR) mask. Finally the flow entries shown in Table 4.1 rely on Layer 3 (Open Systems Interconnection model [139]) matching but could easily be replaced by MAC address matching using the specialised table and leaving the TCAM only for *Selective Routing* as described below.

Switch	Match	Action
1	input_port: 2, src_ip: VM1	output_port: 1
2	input_port: 1, src_ip: VM1	output_port: 3
Local	input_port: 1, src_ip: VM1	output_port: 2
Local	input_port: 3, src_ip: VM1	output_port: 1

Table 4.1: OpenFlow rules to forward packets from VM1 to VM2 through the Firewall NF

The following types of applicable routing policies, dependent on the nature of the network function, are identified below.

Exhaustive routing: All traffic from and to the host(s) goes through the NF, allowing an inspection and alteration of the entire traffic at Layer 2 or 3. Common functions requiring exhaustive routing include Intrusion Detection and Prevention (IDPS) and firewall services.

Service-based routing: A subset of services are routed through the NF, while the rest of the traffic follows the default route. This approach reduces the traffic load in the traversed NFs,

allowing better scalability and denser network function collocation. In FAA data networks, different services use different ports. NFs specific to a given service can utilise this type of routing as they operate on Layer 4. For example, on one FAA network, the Notice to Airmen (NOTAM) data service operates on port 60711 and weather data on port 2480.

Replica routing: A replica of the traffic is routed to the NF, accounting for services that only inspect but never modify data, such as, monitoring and traffic characterisation middleboxes. Replicating routing prevents performance degradation on the data path such as additional latency. Once a packet has traversed the service chain it can be discarded.

4.2.2.1 UAS traffic routing

For the UAS ConOps described in Section 3.4.3, a slightly simplified version of the architecture is sufficient to meet the design requirements. Figure 4.3 shows this variant in the architecture implementation. The key distinction from Figure 4.1 is the OpenFlow enabled switch as part of the Ground Control System, which routes traffic among the UAV, Mission Payload Operator and Pilot in Command systems. Binding the *Server* with the GCS switch alleviates any questions over the shortest path routing in this case. This simplified architecture, where the NF *Server* is located with less dynamic migration, may also prove a valuable abstraction for long distance network environments within ATM such as surveillance networks. Distance among in-network devices in surveillance networks may inhibit the advantages available through migration of *Servers* in more densely networked areas, for example, to achieve load-balancing. Table 4.2 shows the OpenFlow routing table for the UAS implementation. The routes are very similar to Table 4.1 with the key difference of a multi-port egress for traffic at the GCS switch which maps to both the PIC and MPO systems. The GCS switch is connected to the UAV host on port 1. Other UAVs can connect to new ports. PIC and MPO are connected on ports 2 and 3 respectively. The UAS VNF server connects on port 4, with another local switch routing traffic through the containerised VNFs. In this case, GCS source traffic is forwarded from port 1 to 3, and traffic egressing the NF is sent back to local switch port 3 and on to GCS port 4. Incoming traffic on port 4 is

mirrored and sent to both the PIC or MPO displays, if applicable. This routing design allows for additional NFs to be deployed without interfering with the GCS switch routing for the UAVs.

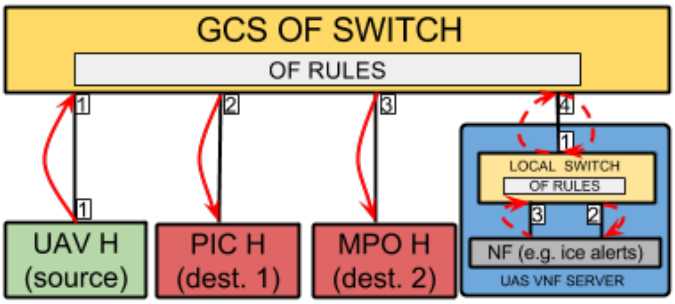


Figure 4.3: UAS VNF OpenFlow traffic routing architecture with UAV and GCS hosts

Switch	Match	Action
GCS	in_port: 1, src_ip: UAV1	out_port: 4
Local	in_port: 1, src_ip: GCS	out_port: 2
Local	in_port: 3, src_ip: NF1	out_port: 1
GCS	in_port: 4, src_ip: local_switch	out_port: 2,3

Table 4.2: OpenFlow table entries for UAS VNF management

4.2.3 Controller & APIs

4.2.3.1 RYU Controller

The RYU Controller is a Python implementation which exposes numerous northbound APIs for management and orchestration operations. The controller is logically centralised but can be physically distributed for further resilience. The controller orchestrates with the distributed, southbound Agents which hold a record of where each NF is deployed and the sequence of the chained position. For example, if a firewall and anomaly detection NF are both deployed for the same traffic routing with the firewall first filtering out unwanted traffic, this filtered traffic is then forwarded to the anomaly detection module. The same NF can be deployed at numerous instances across the topology and even chained with different deployment configurations. This is discussed further in Section 4.3. ATM engineers can add NFs, using the web UI which utilises the controller APIs.

```
actions = [parser.OFPActionOutput(ofproto.OFPP_CONTROLLER)]
instructions = [parser.OFPInstructionActions(
                      ofproto.OFPIT_APPLY_ACTIONS,actions)]

match = datapath.ofproto_parser.OFPMatch(ip_proto=17,
                     udp_dst=ANOMALY_PORT, eth_type=0x0800)
mod = datapath.ofproto_parser.OFPFlowMod(
                   datapath=datapath, match=match, cookie=0,
                   command=ofproto.OFPFC_ADD,
                   idle_timeout=0, hard_timeout=0,
                   priority=0x8010,
                   flags=ofproto.OFPFF_SEND_FLOW_REM,
                   instructions=inst)
datapath.send_msg(mod)
```

Figure 4.4: Source listing of Python RYU Controller OpenFlow rule implementation for higher priority anomaly detection notification routing

SDN-traps, representing notifications from NFs, are received by the Controller. Notifications can be used for a variety of functions such as anomalous behaviour alerts. An *SDN-trap* exploits the OpenFlow *EventOFPPacketIn* events. These events typically take place in an OpenFlow-enabled environment when a packet does not match any of the routing table entries in the OpenFlow switch, encountering an OpenFlow *table miss*. The implementation ensures that all traffic matches OpenFlow records. When a new NF is added on a switch, traffic is routed into and back out of the NF without interfering with the packet's wider routing within the network. When anomalous traffic is sent, it has to reach the Controller. This is achieved by creating and deploying a special rule using the *OFPP_CONTROLLER* as the *OutputAction* in the OpenFlow rules of the switch as seen below:

These rules for anomalous traffic are installed when the switch is first instantiated on the network and the RYU Controller sets up the basic forwarding configuration using the event handler, overwriting the default *ryu.controller.handler* methods:

```
@set_ev_cls(ofp_event.EventOFPSwitchFeatures, CONFIG_DISPATCHER)
```

The priority of the flow rule is crucial, with other flow rules installed by the Controller at the *OFP_DEFAULT_PRIORITY* (*0x8000* or 32768 in decimal). This anomalous flow entry with route directly to the Controller is set at priority *0x8010* thus ensuring it is the dominant flow

rule for packets sent out on the designated, reserved *ANOMALY_PORT*. Further matching for anomalous traffic could be added should a port number alone be an insufficiently distinct classifier.

Each NF is registered with the Controller, ensuring that the structure for anomaly messages and information is stored correctly and able to be queried by various services such as loggers, UI components and engineer's Network Monitoring Systems.

The default JavaScript Object Notation (JSON) structure of the *SDN-traps* include the Network Function deployed, and the details of the anomaly, as defined in the Network Function structure. This information along with any monitoring or other periodic data is stored by the Controller which can be accessed via the northbound API.

4.2.3.2 Northbound APIs

Northbound API calls can be broadly categorised into NFV deployment management and NFV diagnostics and monitoring management. One of the fundamental deployment management functions is adding a network function.

Add NF: The *add_nf* function takes the following arguments:

- The Network Function
- Destination *Server*/in-network device
- The configuration options for the given NF

The controller performs thorough validation of these arguments ensuring a valid NF has been requested, the configuration options are available for this NF, and the requested destination is online and available. Various error messages are returned for failures to meet these criteria. These are sent in JSON as an Hypertext Transfer Protocol (HTTP) response. Headers can be added to this response to permit Cross-Origin Resource Sharing (CORS) for localised implementations, where the controller and UI are located on the same *localhost*, which is possible in small UAS deployments or NF development environments.

Given a successful, validated, *add_nf* function call, the corresponding southbound API function call is made which takes care of the communication, deployment and instantiation messages to the switch or in-network device in the API call. Other deployment management API endpoints exposed follow a similar format and include:

- *rm_nf* - Remove a given NF from a given *Server*
- *get* - Return the NFs on a given *Server*
- *reconfig_nf* - Reconfigure a NF which is currently deployed
- *get_last* - Return the last operation

Diagnostics and monitoring management API calls are very customisable to the monitoring needs of the Network Functions developed by engineers. The basic implemented architecture allows for various *get* methods to get:

- *get_anomalies* - Returns reported anomalies
- *get_monitoring* - Returns available monitoring information
- *get_config* - Returns the configuration of a NF which is currently deployed
- *get_status* - Return the status of the *Agent*

Each function takes arguments for the given in-network device and requires a start time period, e.g., since 1 minute ago. This allows for querying systems to provide real-time asynchronous displays as explored further in Section 4.2.5.

Extending all of this underlying controller functionality is easily achieved, ensuring that ATM engineers can build on the functionality offered if desired, as the infrastructure and ATM service evolves, thus enabling them to meet the systematic demands of future needs.

4.2.3.3 Southbound OpenFlow API integration

The southbound API uses OpenFlow version 1.3 [37]. This was the latest stable version at the time of initial development. The southbound API makes use of two helper functions:

- *add_flow* - Adds a flow record to a *Server*'s OpenFlow table
- *rm_flow* - Removes a flow record from a *Server*'s OpenFlow table

These functions take the following arguments:

- The *Server*
- The *match* criteria, e.g., *in_port* number
- The OpenFlow instructions, i.e., the OpenFlow instructions for the new table entry and the *actions* to take when a *match* is made

The *match* and *OpenFlow instructions* are the same as seen in Figure 4.4. Additional options can be specified according to the OpenFlow specification [37].

There are corresponding southbound API calls for each of the northbound deployment management API calls. Adding a network function takes the same arguments of the destination host switch/in-network device, the NF, and its configuration. This function passes this information to the Agent, a daemon running in the *Server* on the in-network device as seen in Figure 4.1, and explained further in section 4.2.4.

The southbound API also has a role in the northbound monitoring and diagnostics options. The OpenFlow *PacketIn* events are used to collect incoming NF data streams, as configured by the Network Functions and store these for querying by northbound API calls.

4.2.4 NFV Servers

NFV Servers, as seen in Figure 4.1, are co-located with OpenFlow-enabled switches and can be intelligent switches, i.e., switches with processing capability and storage, or they can be

other in-network devices such as middleboxes, e.g., low-cost commodity x86 servers. The Server comprises an OpenFlow-enabled software switch, Docker containers for Network Functions and the Agent. The Agent manages the communications with the RYU Controller and sets the routing within the software switch. The routing is explained in Section 4.2.2. As new NFs are added via the southbound API calls of the Controller, the Agent must configure these containers, with appropriate configuration and set the OF forwarding rules in the switch as required.

In Section 2.3.4.1.1, the merits of container-based virtualisation were discussed. Docker[140] is an open platform for developers and system administrators to develop, ship and run distributed applications as containers. Its two core components are the Docker Engine, a portable, lightweight run-time and packaging tool, and the Docker Hub, a cloud service for sharing containers. Docker, on top of process (container) management, provides layered image management based on the Advanced multi-layered Unification FileSystem (AUFS), used in this architecture to provide dynamic management of the NFs.

Running multiple network functions on the same machine raises concerns about performance isolation and security. In Docker, resource isolation can be secured by *Linux cgroups* that group processes together and resources for every container. A recent Gartner report also showed that Docker is production-ready and using the supported security safeguards discussed in the report, the technology is mature enough to be used in public Platform as a Service (PaaS) environments [141].

To enable the instantiation and management of network functions in commodity servers, our implementation relies on the *Agent*, a single daemon running on the servers hosting network functions. The daemon is responsible for retrieving the requested network function from the repository, instantiating and running it, routing the traffic locally to the relevant container, managing service chains, and providing information on the temporal resource and status of the host.

Figure 4.5 provides a detailed schematic diagram of the Agent. It exposes a REST API to the management network, monitors the health of the machine, and communicates with Docker

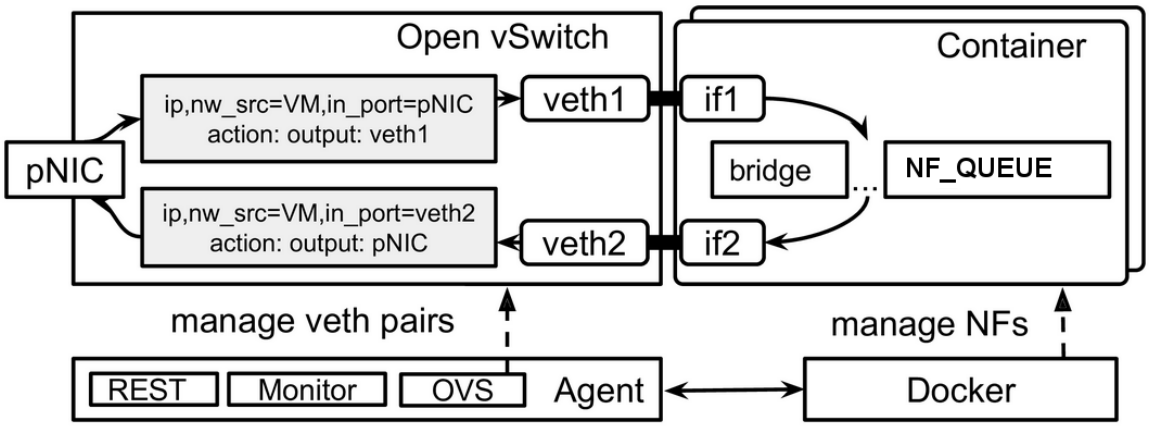


Figure 4.5: Agent's network configuration for a single container.

and OpenvSwitch to instantiate and configure the network functions. Programmable NICs (pNIC) are utilised in the Open vSwitch and virtual ethernet pairs match with bridged ingress and egress interfaces in the containers. The bridge in the containers uses NF_QUEUE[3] for receiving queued packets from the kernel and issuing verdicts and/or reinjecting altered packets to the kernel *nfnetlink_queue* subsystem.

The implementation relies on Docker containers for network functions that can be versioned, shared, shipped and tested with low resource utilisation overhead. The architecture is conceptually agnostic with respect to virtualisation techniques, e.g., LXC containers can be used if the ability to attach two virtual interfaces (egress and ingress) is provided for the Agent. Mininet hosts can also substitute containers for effective development and testing if desired.

4.2.5 User Interface

The User Interface implementation is fully functional and practical, however it is recognised that for ATM environments situational awareness and Human Factors are significant to user interface and user experience design. Such design considerations are outwith the scope of this work and that the UI presented here is highlighting the functional capabilities which could easily be adapted into a more thoroughly designed interface with e.g. ethnographic trials with engineers.

[3] `https://www.netfilter.org/projects/libnetfilter_queue/`; Accessed: August 2016

The User Interface is built in HTML with jQuery[4]. Diagnostics and monitoring management API calls from the Controller access the latest information. The real-time charts are customised using C3.js[5] which is built on top of D3.js[6]. D3.js is a JavaScript library for manipulating data driven documents. C3.js is a D3.js-based library, which facilitates making charts based on a stream of data.

Figure 4.6 shows the NF deployment UI for Switch 1. As it can be seen, Switch 1 currently has Monitoring and Telemetry NFs without any custom configuration installed. The choices for these screenshots have been limited to three basic core NFs, of course, a wider choice would likely be available in deployed environments. These NFs can be removed by clicking the 'X' button to the right. Further NFs can be added by selecting them from the 'Available' list. Configuration particular to a given NF can also be added. The UI ensures duplicate operations cannot be sent, as each operation receives a confirmation notification. A latest activity message above the panel acts like a console showing the last action performed.

Figure 4.6: UI NF deployment control panel showing Monitoring and Telemetry NFs installed

The real-time graphs for displaying monitoring information on Switch 1 are shown in Figure 4.7. The graph is updated as each API call is made to the Controller to get the latest information from the NF. The default configuration for the Monitoring NF deployed is to monitor the *InBytes* and *OutBytes*. This can be set to show a range of other statistics as

[4] `https://jquery.com/`; Accessed: August 2016
[5] `https://c3js.org/`; Accessed: August 2016
[6] `https://d3js.org/`; Accessed: August 2016

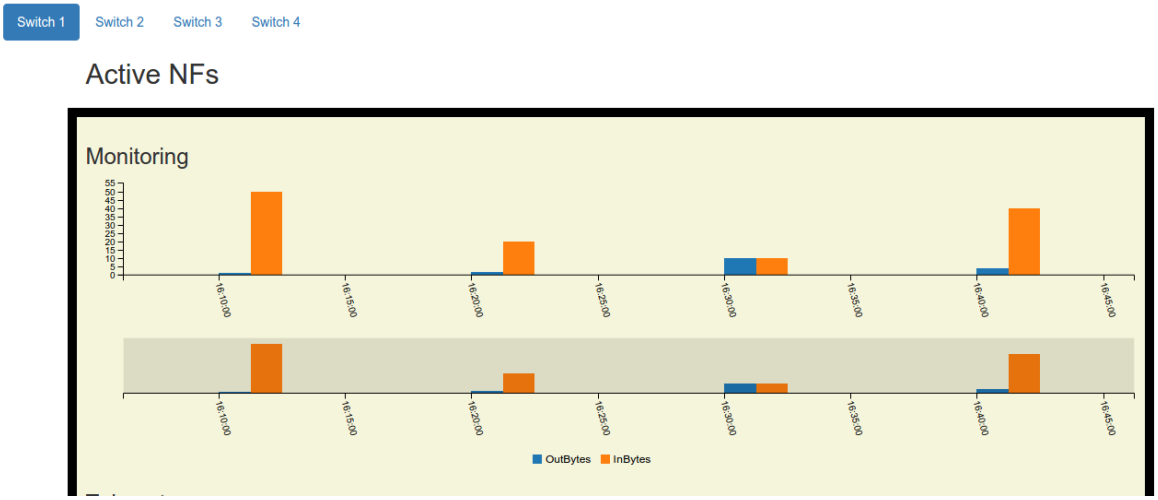


Figure 4.7: Real-time monitoring graphs using C3.js charts for NF installed on Switch 1

highlighted in section 4.3.2. The graphs shown reflect the NFs installed and are shown and removed in line with the deployment control panel. Beneath the Monitoring NF chart is the Telemetry NF chart as shown in Figure 4.8.

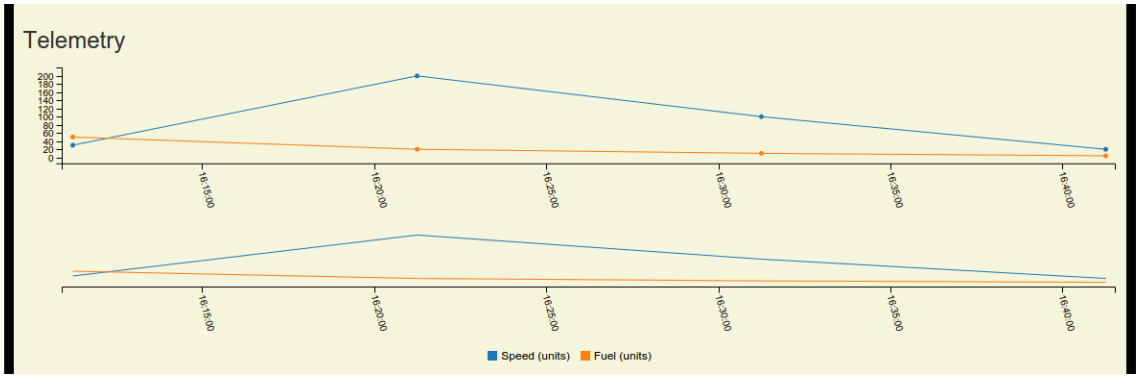


Figure 4.8: UI for telemetry dual real-time readings of Speed and Fuel

The Telemetry NF default is to detect traffic from a specific UAV model and to show the Fuel and Speed values extracted from these telemetry packets. Available telemetry varies for various aircraft and UAV models, so the NFs for telemetry would need to be programmed to read the various models and which parameters are available. This can be easily added into the existing configuration options for NFs.

Figure 4.9 shows the full page view. The top right has a button to toggle the control panel and topology view, to show purely the NFs graphs. This ensures that the operators have the opportunity to see a fuller view of the information visualisation presented to them. The four

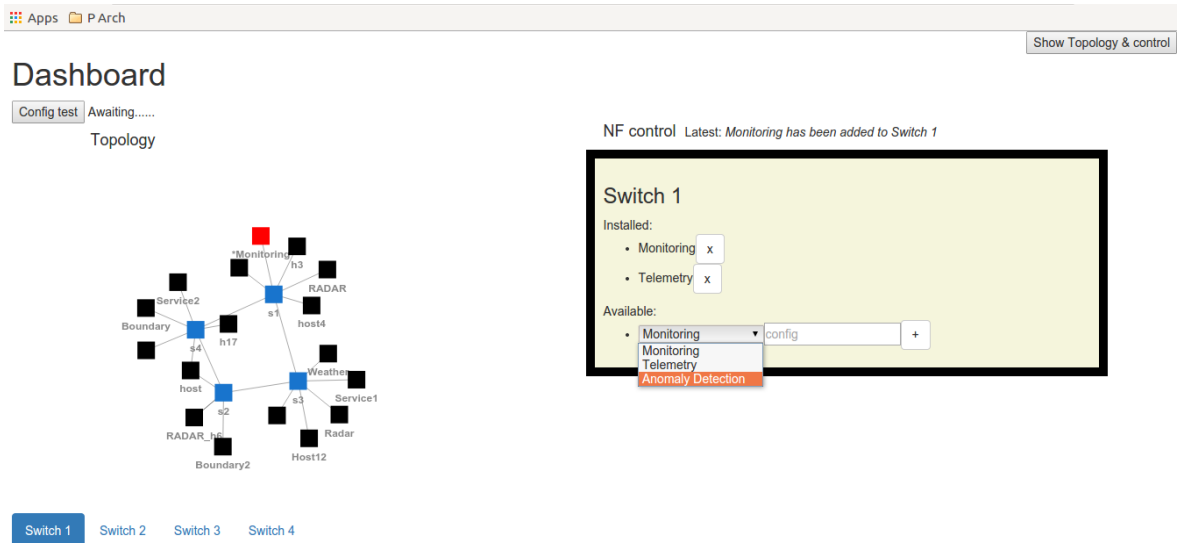


Figure 4.9: The full UI interface for configuring the architecture

switch buttons, where 'Switch 1' is highlighted as selected, move the view of the graphs and control panel among the various OpenFlow enabled switches in the topology. The topology can be seen on the left of the control panel. A basic topology can be seen which colours switches blue and hosts black, including deployed NFs. If a new anomaly detection alert arrives for a given NF, the host is coloured red in the topology and the graph for anomaly detection, throwing a more prominent notification in the information visualisation graph. Section 4.3.3 discusses an anomaly detection NF implementation. The red host can be clicked which opens a modal explaining the anomaly detected in more context as seen in Figure 4.10.

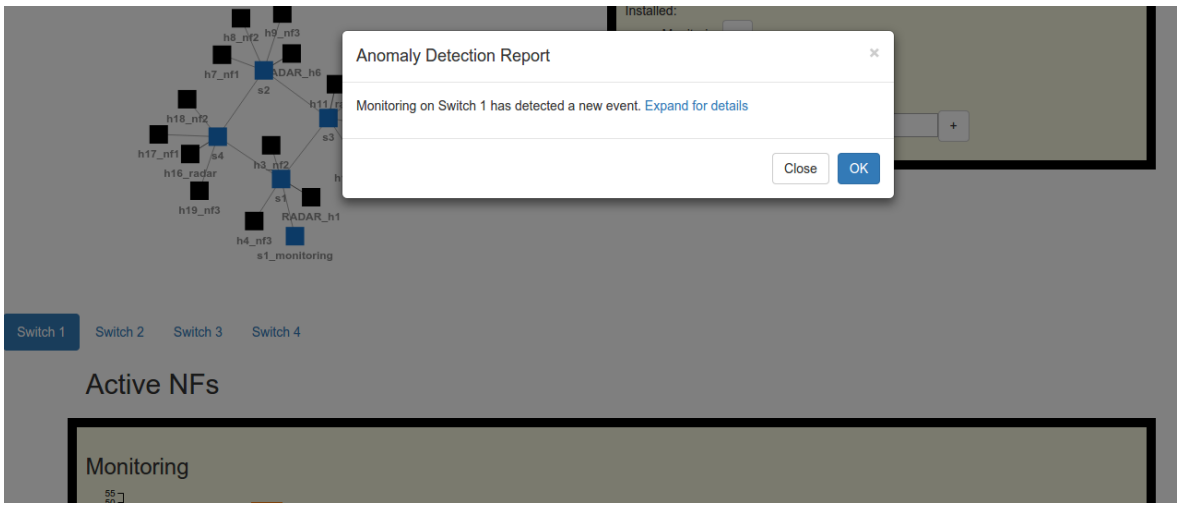


Figure 4.10: Anomaly detection topology modal from topology UI display

The UI offers operators and engineers the capability to interact with the architecture in real-

time and view information as well as to perform NF management and orchestration. Further enhancements to the UI can be made using standard web development, beyond the scope of this work.

4.2.6 Hardware Considerations

As part of the implementation, the functional deployment architecture was deployed on an experimental test bed consisting of Intel i7 servers with 16GB of memory and Gigabit OpenFlow switches. In this section, the base throughput, delay and boot time evaluation for multiple containers is provided, contrasted against other prominent function virtualization approaches, where appropriate. In Figures 4.11, 4.12 and 4.13, the functional deployment architecture implemented in this work is shown as FDA.

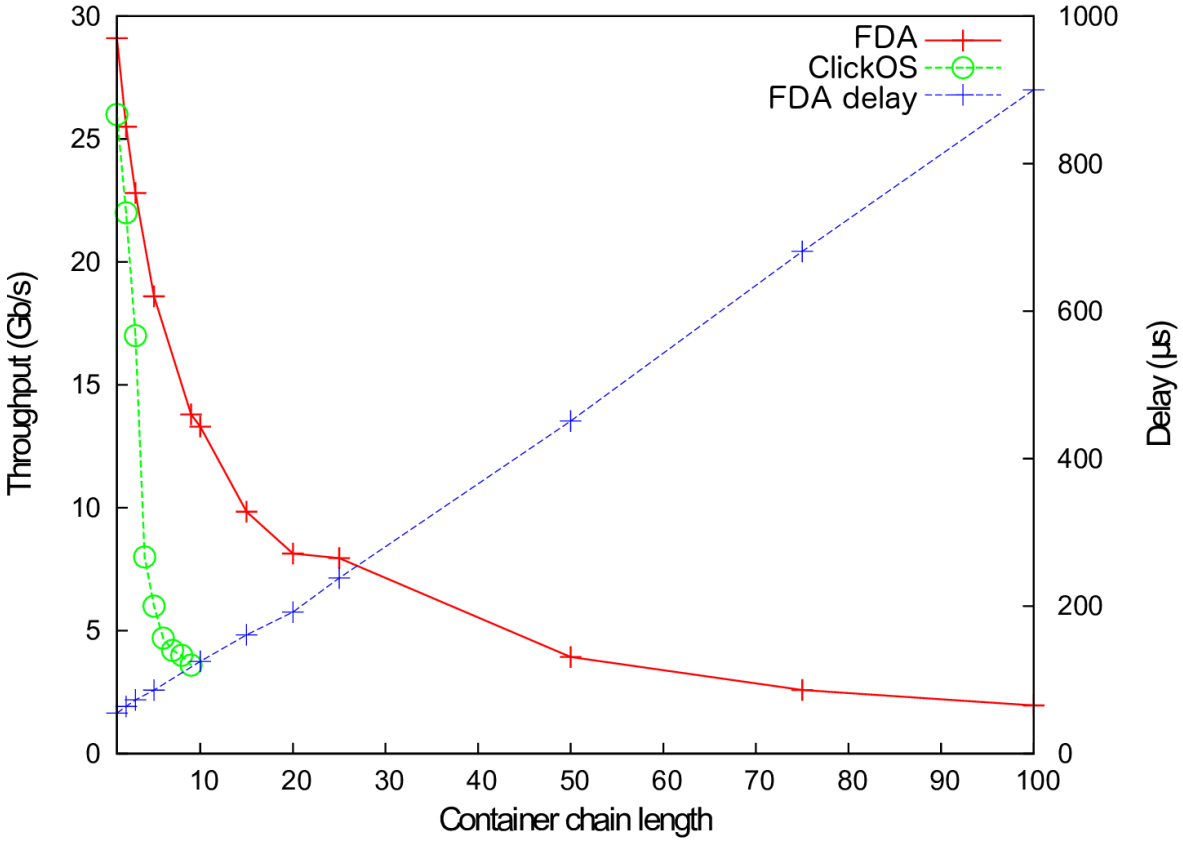


Figure 4.11: Throughput and delay of chained NFs

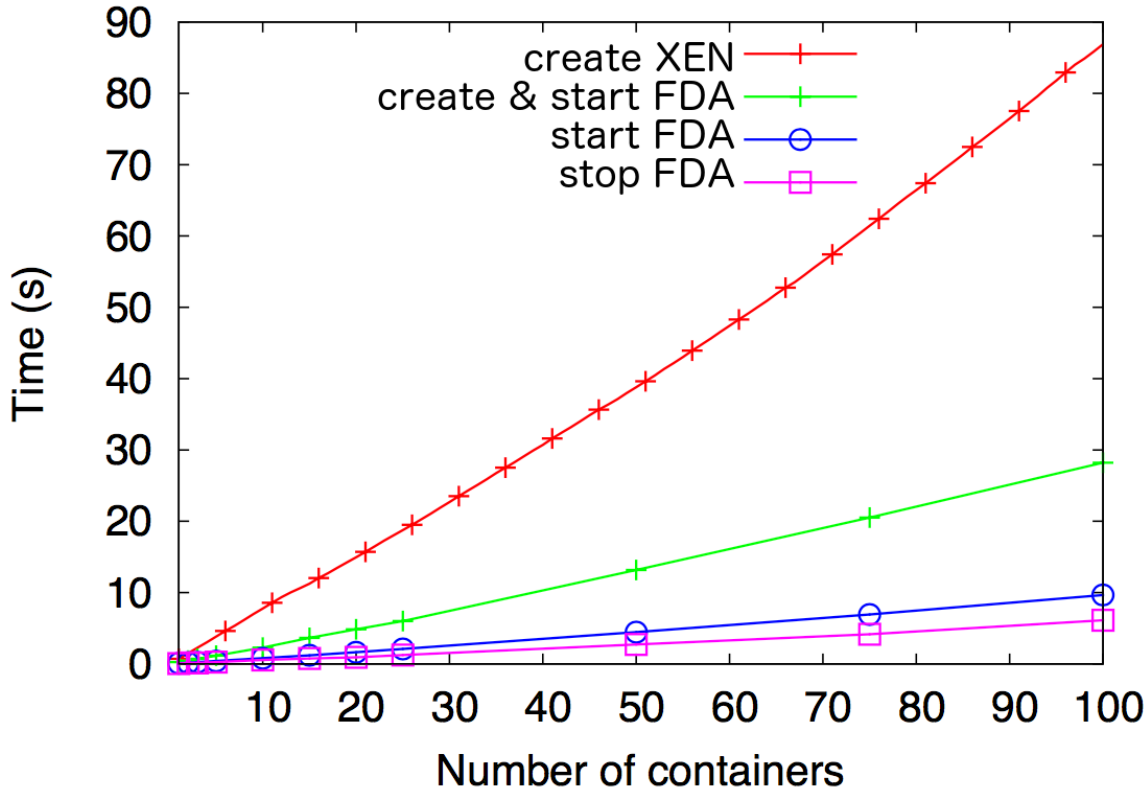


Figure 4.12: Create/Start/Stop time

4.2.6.1 Throughput

Iperf was used to measure the maximum throughput between the source and destination hosts connected via chained *wire* NFs. The wire functionality is a standard Linux bridge forwarding the traffic from the ingress to the egress ports of the NF. It is therefore the simplest form of NF and can be used to evaluate the minimum performance impact of a virtualized service. The following experiments use a single maximum segment sized TCP stream and the default networking stack configuration of Linux kernel 3.13 (TCP Cubic; initial congestion window of 10 segments; minimum retransmission timeout of 200ms; window scaling, timestamps, selective acknowledgement and server-side Explicit Congestion Notification (ECN)).

Figure 4.11 shows the packet processing throughput with NF chains hosted on a single host and is therefore not limited to the speed of the topology (e.g., 1Gbps physical switches or network cards). In this figure, we show that the container-based approach to NFs significantly outperforms ClickOS, with a single wire NF in this functional deployment architecture out-

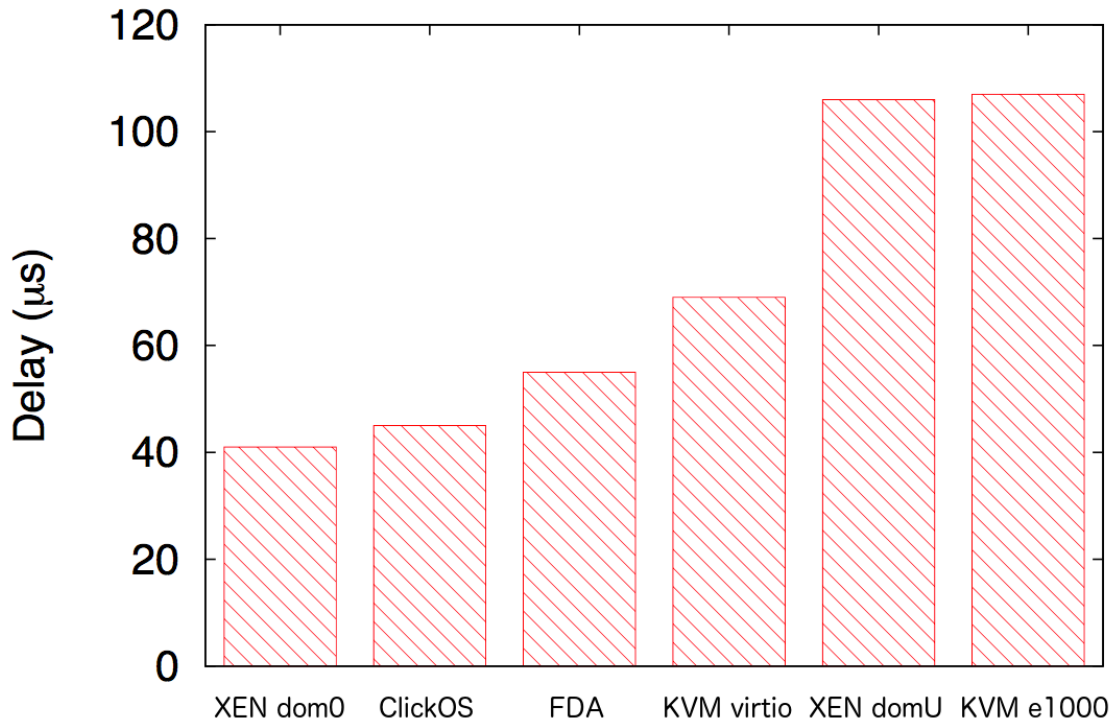


Figure 4.13: Idle ping delays

performing a ClickOS wire by more than 4Gb/s. It is also evident that this work scales better as the NF chain grows: with 9 containers chained together, packets are processed at 13.8Gb/s compared to 3.6Gb/s using ClickOS, while Gigabit speeds can be still maintained for 100 chained containers. Note that these results are limited to the wire NFs and can be explained by the fact that this implementation does not copy the packets from the kernel space to the user space as ClickOS does.

4.2.6.2 Boot times

To provide high flexibility in container placement, it is necessary to quickly manage the container's lifecycle, i.e., *create*, *start* and *stop*. With a rapid turnaround time, it is possible to enable/disable new network functions in short timescales, as well as to allow for fast migration, better consolidation and placement. In case of unexpected interruption, such as container or host failure, fast recovery can be achieved by restarting the same network function on another server and redirecting traffic.

Figure 4.12 shows the time required to create, start and stop 1 to 100 containers on the same host. In all three cases the growth is linear, allowing the infrastructure to scale when a large number of containers are instantiated. The significant difference between *create & start* and *start* shows that it is beneficial to pro-actively create frequently-used containers and only starting them when required. Containers significantly outperform the creation time of Xen, and highlight the poor and exponential cost of Xen to create more domains [142].

4.2.6.3 Delay

Middleboxes should process packets transparently, therefore keeping additional latency to a minimum in order not to compromise end-user experience.

The results shown in Figure 4.13 and 4.11 use ICMP (ping) traffic between source and destination to measure the idle delay impact of a wire NF. Figure 4.13 adds the idle latency of this implementation to the original ClickOS performance evaluation [70], and compares it to ClickOS, different Xen domains, as well as to different Kernel-based Virtual Machine (KVM) virtualized Network Interface Card (vNIC) drivers. Using a stock configuration of Ubuntu Server 14.04, this implementation performs better than KVM regardless of the vNIC driver used and Xen guest system (domU).

The ClickOS design aimed at providing high performance, low delay NF through significant modifications to the hypervisor resulting in a reduction of latency from 106μs (domU) to 45μs, 10μs faster than this work.

In order to keep each network function as a single functional block, it is necessary to be able to chain them to enforce multiple policies sequentially and at different layers of the topology. Figure 4.11 shows the maximum throughput achievable and delay induced by a chain of 1 to 100 NFs. The delay impact is linear as the number of chained containers increases. With 100 containers chained together on the same host, the functional deployment architecture provides sub-millisecond delay. As the number of chained containers increases, this work performs $3.1\times$ faster than Xen-based ClickOS with 5 containers and $3.8\times$ faster with 9 containers, and allows a much higher number of containers to be chained together.

```
FROM base
MAINTAINER Kyle White

ENTRYPOINT ifinit && \
  brinit && \
  /bin/bash
```

Figure 4.14: Source listing of Dockerfile for Wire NF

4.3 Sample ATM Network Functions

While the Network Functions required by ANSPs will vary, the core requirements for network functionality are similar to the needs of the majority of network operators including: network monitoring, anomaly detection, intrusion detection, diagnostics, and Quality of Service functions. This section details the implementations of some sample NFs across the different common network tasks. These sample NFs are a foundation for this function deployment architecture, from which operators can build more NFs for their future requirements.

4.3.1 Generic NF Set up

On instantiation, each NF has to first set up the bridge between the ingress and egress interfaces of the container. A Dockerfile is a key part of Docker which describes a sequential series of actions to build the container environment. The Dockerfile for the wire NF, or an NF which takes no action, is shown in Figure 4.14.

Figure 4.15 shows the *base* set up from which the wire NF is based *FROM*. The *base* is built on top of a standard Linux distribution. In Figure 4.15, the Dockerfile instructs the installation of various standard Linux tools required for the implementation. Scripts *ifinit* and *brinit* are added to the environment, which are for interface and bridge initialisation, respectively. They are then made executable, and a bash shell is set as the container's *entrypoint*.

Examining the bridge initialisation, Figure 4.16 shows the *brinit* bash script. It is a simple script that uses the *brctl* program to add a new bridge in the container, and adds the two

```
RUN apt-get update && apt-get install -y \
  bridge-utils \
  net-tools \
  iptables

ADD ifinit /usr/local/bin/
ADD brinit /usr/local/bin/
RUN chmod +x /usr/local/bin/ifinit
RUN chmod +x /usr/local/bin/brinit
ENTRYPOINT ifinit && /bin/bash
```

Figure 4.15: Source listing of Dockerfile for *base*

```
#!/bin/bash

brctl addbr br0
brctl addif br0 if1
brctl addif br0 if2
ifconfig br0 up
```

Figure 4.16: Source listing of *brinit* script

interfaces before bringing the bridge up.

4.3.2 Monitoring NF

Network monitoring encompasses a vast array of techniques and methods. For ATM network monitoring, *passive* monitoring is preferred, where traffic flows without any interference. The fundamental information required for network monitoring systems is the number of bytes, packets and dropped packets sent and received. Monitoring each of these parameters can easily be achieved in this architecture. A Network Function can be built which utilises the Linux kernel interface statistics[7]. This gives statistics for numerous errors and:

- collisions
- multicast

[7] `https://www.kernel.org/doc/Documentation/ABI/testing/sysfs-class-net-statistics`; Accessed: August 2016;

- rx_bytes
- rx_compressed
- rx_dropped
- rx_packets
- tx_bytes
- tx_compressed
- tx_dropped
- tx_packets

Some of these aggregations are less insightful to be monitored on an in-network device, since it is likely any dropped packets will take place on the host switch as opposed to the interfaces to the NF, but they are accessible. Packet and bytes sent and received over time can be easily obtained by measuring these values in a Python script which divides the difference between two readings by the length of delay between readings. This delay or the desired rate measurement can be configured at instantiation time using the configuration settings which are passed from the UI to the Controller and on to the Agent.

An alternative implementation is to measure the statistics held in NF_QUEUE which record similar statistics but provide values with a pre-calculated rate per second. This could be achieved using a straightforward periodic parsing of the netfilter statistics. When statistics are gathered, they can be either stored for polling by the UI on-demand, or be configured to send the real-time updates to the Controller for live network monitoring.

4.3.3 Anomaly Detection: Network Scan NF

Similar to network monitoring, network anomaly detection is a broad field. To highlight the flexibility of the architecture, the task of detecting a network scan is explained in this subsection. Network scans can include port scanning or IP address range scanning. Fontugne [143]

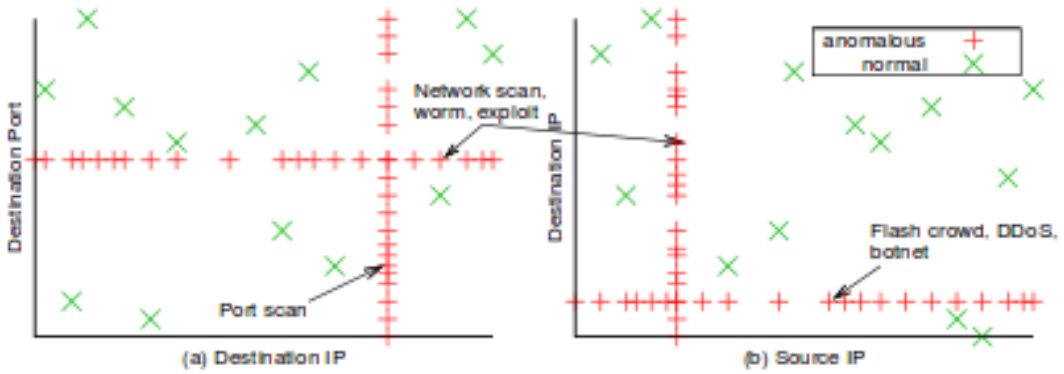


Figure 4.17: Hough transform method to detect network scans by Fontugne [143]

presents a novel technique of using the Hough-transform algorithm in order to detect such scanning attempts. This technique is chosen in order to highlight the breadth of possible implementations within NFs, provided they can be implemented using the traffic flow and the native Linux environment.

The Hough transform is a feature extraction algorithm which is predominately used in image processing analysis. In essence, the algorithm seeks shapes or features within a graphic. The idea proposed by Fontugne is to plot traffic features on a scatterplot graph, with axis pairs for four of the key flow identifiers: source IP address, destination IP address, source port and destination port. Then using thresholds of sensitivity, the Hough transform can detect straight lines, which are indicative of network scans or other anomalous behaviour. In Figure 4.17 various attacks are shown with their red coloured anomalous points. For example, a potential DDoS or flash crowd can be seen in the right graph, where the horizontal line of red points represent traffic from many source IP addresses going to the same destination IP address. A port scan can be seen as the vertical red line in the left graph with a steady destination IP address but differing across a range of destination ports.

The implementation of this method required a sensitivity threshold to determine how many points make a positive anomaly detection. Another threshold that may be required is to remove older points over time, therefore examining a time window of the previous 30 minutes, for example. This may be needed to avoid legitimate, yet very different, traffic from the

```
nfqueue = NetfilterQueue()
nfqueue.bind(1, record_and_accept)
try:
    nfqueue.run()
except KeyboardInterrupt:
    print
```

Figure 4.18: Source listing of Python Hough transform Netfilter Queue binding

```
src_ip = int(netaddr.IPAddress(parsed.src))
scaled_srcip = (src_ip % 65536)*scaled

dst_ip = int(netaddr.IPAddress(parsed.dst))
scaled_dst_ip = (dst_ip % 65536)*scaled

sports.insert(0,int(round(udp.sport*scaled)))
dports.insert(0,int(round(udp.dport*scaled)))
src_ips.insert(0,int(round(scaled_src_ip)))
dst_ips.insert(0,int(round(scaled_dst_ip)))
```

Figure 4.19: Packet features recorded for periodic scatterplots

past being compared with recent activity. These thresholds would require a domain expert to set them for a given ANSP network. Given ATM network characteristics, normal behaviour may be very tightly defined across long periods of time, meaning that a time window is not necessary.

For the purposes of this implementation, fifteen points were deemed a significant line. The algorithm was implemented in Python using NetfilterQueue and scapy[8] libraries. The first task is to establish and set a callback function with the netfilter packet queue, and call the function to record the traffic features for plotting, as seen in Figure 4.18. The netfilter queue is set up using *iptables -A FORWARD -j NFQUEUE –queue-num 1*.

The function *record_and_accept* is called when a packet enters the netfilter queue. Packets are accepted, *pkt.accept()*, and their features recorded as follows:

The IP addresses are read and converted to integers. They are then scaled, where they are multiplied by a scaling factor to ensure the plots are on a 1000px square graphic. This means

[8] `http://www.secdev.org/projects/scapy/`; Accessed: August 2016

each pixel plotted represents a range of IP addresses and ports. Depending on the traffic mix and behaviour, such scaling may need to be considered for each network deployment.

Graphs were plotted using the rpy[9] library. Axes were removed to simplify line detection. Points plotted were also a single pixel, as opposed to the crosshair icons in Figure 4.17 for the same reason.

4.3.4 Remediation NFs

ATM networks have some active network functions which modify the traffic on the network. Such NFs are necessary for many purposes including: to ensure service resilience, translation from legacy protocols and systems into modern ones, and for halting malicious traffic. A key NF for network engineers is a firewall. Enterprise networks including ANSPs have firewalls deployed ubiquitously. The FAA has numerous firewalls deployed at the boundary edge routers for their FTI network.

Using Virtual NFs for firewalls allows for far greater flexibility, including chaining firewalls which block different traffic features, rapid deployment and reconfiguration, and easy re-deployment. An example use case could be a detected malicious network device, sending attack traffic. A firewall NF can be deployed to the egress in-network devices to block traffic from this malicious source IP address.

Figure 4.20 shows a basic IP address based firewall, which accepts or drops packets based on their inclusion in *blacklisted_IPs*, a Python set. This firewall NF shows the basic foundations for virtual NFs using netfilter queue bindings and basic Python logic. More sophisticated NFs can be achieved based on these same underlying capabilities provided by this implementation of the architecture.

[9] `http://rpy2.bitbucket.org/`; Accessed: August 2016

```
def firewall(pkt):
  # firewall criteria
  if IP(pkt.get_payload()).src in blacklisted_IPs:
    pkt.set_verdict(nfqueue.NF_DROP)
  else:
    pkt.accept()

nfqueue = NetfilterQueue()
nfqueue.bind(1, firewall)
nfqueue.run()
```

Figure 4.20: Source listing of Python Firewall Netfilter Queue binding

4.3.5 Diagnostics NF

Monitoring, anomaly detection and remediation NFs are all vital capabilities for network engineers managing their infrastructure. Diagnostic capabilities are also crucial to enable engineers and operators to run tests and measure results in order to gain a better understanding of the current network behaviour.

As standard, Linux has numerous essential tools to help network engineers including: *ping*[10], *tcpdump*[11], *iperf*[12], *netstat*[13], *traceroute*[14]. Depending on the *base* of the Docker containers, some of these programs may be available by default. Any programs which are desired that are not in the *base* can be added in the Dockerfile using the appropriate *RUN apt-get install* Dockerfile commands prior to the *ENTRYPOINT* with the appropriate Linux packages/libraries. Python environments and scripts can also be made available in NFs for diagnostics using similar Dockerfile commands. An example Python diagnostic script is a deterministic anomalous traffic generator for use in combination with an anomaly detection NF. For example, running a deterministic diagnostics NF for the Network Scan NF seen in section 4.3.3 to confirm generated, expected anomalies are identified. Such a test can be used in a similar fashion to a software engineering unit-test, to prove the NFs and intermediate routing are continuing to work as they did prior to network re-configuration, for example.

[10] `http://ftp.arl.mil/~mike/ping.html`; Accessed: August 2016
[11] `http://www.tcpdump.org/`; Accessed: August 2016
[12] `https://iperf.fr/`; Accessed: August 2016
[13] `https://sourceforge.net/projects/net-tools/`; Accessed: August 2016
[14] `https://sourceforge.net/projects/traceroute/`; Accessed: August 2016

Figure 4.21 shows an abridged source listing for performing a basic deterministic port altering script to trigger the Network Scan NF seen in section 4.3.3 to throw anomalous notifications. The diagnostic NF performs the test by sending randomised data at pre-determined times, *alter_port_times* using different randomised ports. Based on the sensitivity configured for the anomaly detection NF, e.g., the number of different ports in a given time period, the operators can measure the expected time for anomalous notifications after these anomalies have been generated.

```
clock = 0

UDP_DST = (UDP_IP, UDP_PORT)

while clock < 100:

  if clock == start_time:
    sock = sock_open(port[0])

  message = json.dumps({
    'data': test_data[clock%10],
    'notification': test_notification
  })

  sock.sendto(message, UDP_DST)
  time.sleep(1)

  if clock in alter_port_times:
    sock.close()
    sock = sock_open(ports[clock])

  clock += 1

sock.close()
```

Figure 4.21: Source listing of Python deterministic diagnostic test NF for Network Scan anomaly detection NF

4.4 Summary

In this chapter, it was shown how the architecture designed in the preceding chapter was implemented. The technical choices for various aspects of the design were justified, and systematic diagrams were presented to further explain key aspects of the system. Key sections in this chapter explored the technical details with respect to traffic routing and the SDN controller, and north and southbound APIs. NFV Servers and the User Interface were also detailed before presenting sample NFs which offered a demonstration of the range of capabilities this system provides to operators. The instantiation for NFs and these implemented NFs provide the foundation for the evaluation work for this system.

This function deployment architecture is particularly suited for ATM data networks for many reasons. Firstly, ATM networks are transitioning to IP-based networks, which support a diverse set of different services. Secondly, these networks already make significant use of middleboxes and in-network devices. Such devices can be used as locations for VNFs and highlight the need for a suite of NFs offering a range of functionality from monitoring to anomaly detection and remediation actions, such as firewalls. More specifically, a core shortfalling for current ATM systems is a lack of on-demand monitoring capabilities to assist with diagnosis and defence of emerging challenges. This function deployment architecture enables operators to increase their visibility of services and infrastructure on-demand by deploying specific monitoring functionality as required. The network softwarisation technologies shown in this chapter give ATM operators the ability to easily route subsets of their traffic, for example, a specific service, through a network function on-demand.

Chapter 5

Evaluation

5.1 Overview

When evaluating this work, it is important to recognise two key factors.

Firstly, when examining past incidents, it cannot be proven that if different measures were in place, the adverse impacts would not have occurred.

Secondly, this work is based on enabling future ATM systems to have increased monitoring capabilities and resilience through providing a more flexible, responsive, and adaptive system to engineering teams and operators. Future Air Traffic Management systems will likely involve significantly increased air traffic volumes, significantly increased data traffic, the introduction of UAVs, greater interdependencies and system integrations, all while maintaining significant legacy systems and with demand for higher levels of both safe and disruption-free operation.

It is therefore important and appropriate to evaluate this work based on the enablement this virtualised flexible architecture provides. This chapter performs this evaluation by discussing the enablement achieved through a suite of NFs designed to assist in both everyday operations and in the face of emerging incidents. In summary, this work enables flexible and responsive monitoring, and facilitates a function deployment architecture which operators can use to define NFs which meet their specific, emerging requirements.

Throughout the preceding chapters and in the motivation for this work, a number of shortfalls of current ATM systems have been raised which must be tackled in order to maintain the current growth and emerging demands on the service without heightened risk of disruption. These shortfalls include: insufficient network monitoring and anomaly detection capabilities for operators; poor recovery times, often due to prolonged diagnosis periods where further network information would assist their investigations; a lack of capabilities to take remedial actions in the face of ongoing disruption or in emerging adverse conditions, such as, data traffic prioritisation for the most critical services at the given moment; and, lastly, the lack of adaptability in current monitoring capabilities. This evaluation shows how this work contributes to tackling these shortfalls and how the architecture, coupled with the characteristics of ATM networks, can be utilised to better inform and enable operators to gain increased visibility of their services in the face of emerging requirements and challenges.

The remainder of this chapter is structured as follows: Section 5.2 presents an evaluation based on the aspects of this work which facilitate rapid detection of flooding incidents, and therefore implicitly improve recovery times in the face of outages due to operators being more informed. Three evaluations based on monitoring and detecting anomalous behaviours are presented and discussed in Section 5.3. The evaluations are based on set of distinct ATM services. Surveillance is a vital aspect of ATM operations and therefore the services of surface based movements and in-flight radar data were chosen. The surface surveillance data analysis evaluation is explained in Section 5.3.1. The evaluation of secondary surveillance en-route radar in Section 5.3.2 also examines the corresponding data network traffic, which is another key capability of this work. Finally, the imminent requirement for ATM systems to integrate with unmanned systems leads an evaluation tailored to UAV telemetry modelling in Section 5.3.3.

5.2 Enabling Faster Recovery Times

5.2.1 Future Flood Detection

A recurring incident which has caused significant disruption to ATM in recent years is disruptive data network flooding. Flooding significant volumes of data causes major disruption largely due to overwhelming the network's ability to deliver a reliable service, uncertainty over the validity of data transferred and in causing significant latency and packet loss to operational traffic. ATM operators typically respond to such incidents by safely reducing their air traffic capacity or in the worst cases, grounding operations. This causes major disruption and recurring recommendations from similar incidents include the need for faster recovery times. A key aspect in achieving a faster recovery is in achieving faster detection and diagnosis. Distributing detection reporting is problematic when combating flooding, as notifications can get caught up in the disruption. Severe latency from long queues at each switch can delay notifications or they could be dropped due to link oversubscription. Centralisation of information can also assist in the diagnosis of understanding a challenge. By keeping previous data, comparisons can be made and emerging trends observed in the lead up to an event.

An increase in traffic on a given link can be detected in many existing ways, including using tools such as *netstat* inside a diagnosis NF as described in Section 4.3.5. Here, the specific example explored is where a flooding source is broadcasting or sending to a larger number of destinations than is recognised as normal behaviour at a per-switch level. Such behaviour has had a significant disruptive impact on ATM operations previously [125].

Under SDN with OpenFlow, when a switch tries to send a packet to a destination for which there is no matching flow table entry, the packet can easily be sent to the centralised SDN controller. The SDN controller would typically install a new rule on the switch enabling the packet to reach its destination. This behaviour can be exploited to meet our requirements as argued above for centralised detection with storage of past flow routes per switch which can be observed for emerging trends at the time of anomalous behaviour. Using the architecture

in this work, SDN and OpenFlow can be configured to enable human operator intervention versus autonomous actions when the number of requested new flow table entries for a given switch exceed a threshold, determined by recent behaviour. This approach is possible due to the largely predictable nature (relative to the Internet, for example) of some ATM service architectures, such as surveillance data networks, their regular traffic patterns, and single authoritative control.

Using a scaled topology based on a major European ANSP's secondary radar surveillance network, a flood and its subsequent detection is presented in the following section. To enable operators to achieve faster detection and diagnosis, the SDN controller is configured to monitor traffic flows per switch, throwing notifications of anomalous levels very quickly.

5.2.1.1 Experiment

Figure 5.1 portrays the experimental topology. The core of the network is a ring connecting switches S_1, S_2 and S_3. These switches represent the primary ATM locations throughout the country. H_1, H_2 and H_3 represent the subscribers to the surveillance data to display where aircraft are located on Air Traffic Control Officer's displays, for example. In reality, these hosts are large LANs with their own layers of redundancy, but this can be abstracted without invalidating the experiment. In Figure 5.1, the core ring network connecting switches S_1, S_2 and S_3 has a high bandwidth and shares the captured radar data from the distributed radar locations, represented here as H_4, H_5 and H_6. Each radar dish has a local switch which sends dual copies of the output data on the *Red* and *Green* links which represent the 2-fold physical redundancy in the network. Each switch is configurable through the centralised SDN controller.

To begin, standard operational traffic was initiated in the network with H_4, H_5 and H_6 sending their continuous radar data to the core ring via the *Red* and *Green* links to S_1 and S_3, respectively. The operational traffic was modelled from recordings of live packet capture trace data.

As the standard traffic began, the RYU controller established flow table entries for the

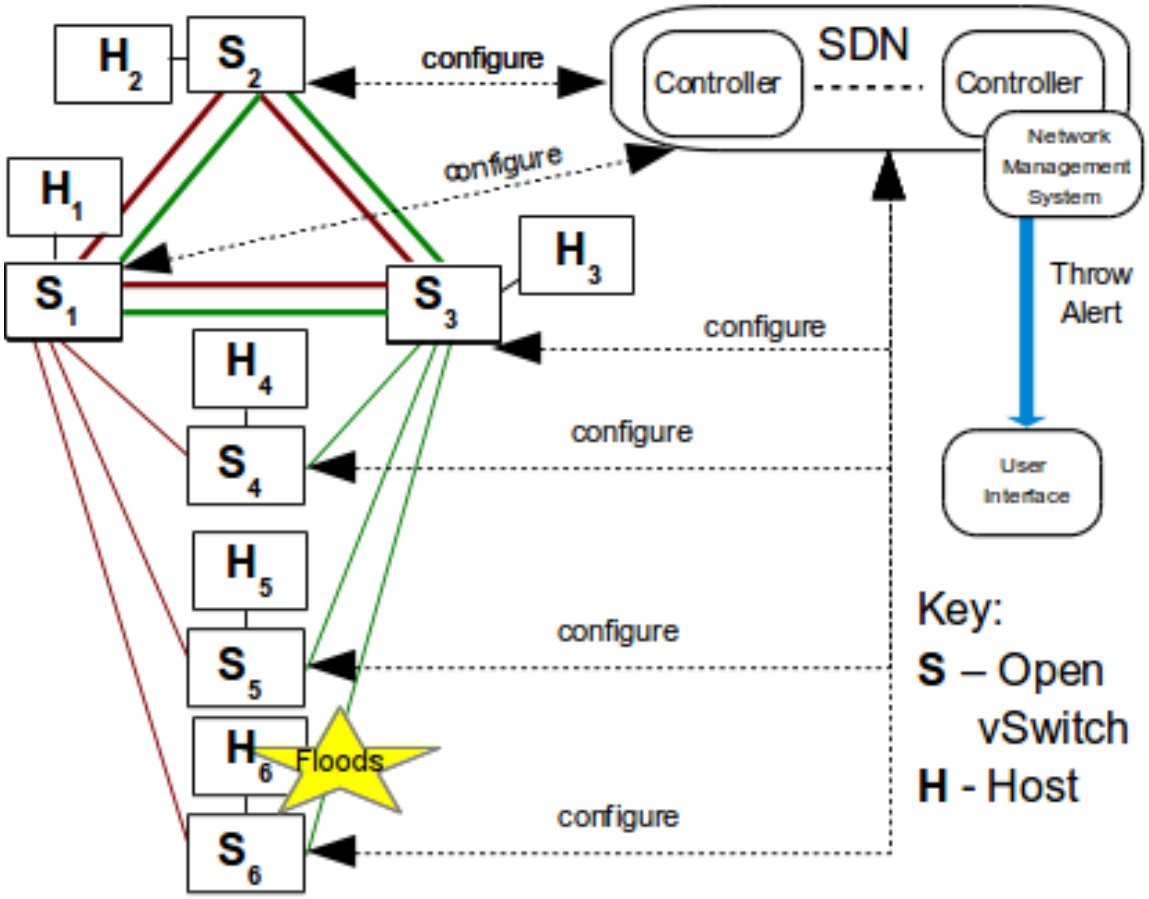


Figure 5.1: Flooding experiment on scaled ANSP radar surveillance network topology

switches on the core ring, allowing their associated hosts to directly route traffic to each other. Flow table rules were also installed on the radar switches S_4, S_5 and S_6 to allow them to send traffic from their hosts to H_1 and H_3.

In the controller, a traffic metric polling was implemented for each switch to provide the number of flows, the number of bytes, and the number of packets sent from that switch to each destination address. This polling was triggered by a timer event every five seconds. From the static characteristics common to critical infrastructure data networks, it follows that new connections between pairs of devices which had never previously exchanged data is a relatively rare event in this domain. Once the network is established and the controller has installed flow table entries for standard operational traffic patterns, new connections to send data from H_4 to H_5, for example, are unlikely. This characteristic was exploited to better detect a flooding challenge in the network. Every time a packet is sent to the controller and a new flow table entry is installed on a switch, this forwarding rule was added to a list of the

latest added routes. Each time a timer event is called, the number of new routes which have been added in the network are examined. If this number exceeds a given threshold for new connections made, this indicates abnormal behaviour within the network. Such behaviour could be representative of a number of anomalies. By then checking for a significant increase in the volume of data sent from this switch via the polled traffic metrics, it can be determined if this has the characteristics of a flooding incident. The algorithm used is:

$On\ Timer\ Event$:
for each $Switch\ in\ Latest\ Route\ Entries$:
if $number\ of\ new\ routes\ for\ this\ Switch > MAX\_NEW\_ROUTES\_THRESHOLD$:
then
  if $significant\ traffic\ volume\ increase$ **then**
    $ThrowFloodCharacteristicsAlert$
  else
    $ThrowGenericAlert$
  end if
end if

After completing the check on the latest route entries, they are archived. Any new flow table entries created in the next time period will be evaluated independently of archived results. This is based on the profile of a flooding event typically being a rapid process in which a malfunction or misconfiguration rapidly causes data to be sent from a constrained set of sources to a large set of destinations. When the algorithm throws a flood characteristic event, details of the switch, its latest routes and the traffic volumes are passed to the controller. The controller then exposes these alerts which can be collected by a network monitoring system and reviewed. Network operators could then act to block flows from this switch by deploying firewall NFs or rate limiting NFs, increase detection NFs in other parts of the network and perhaps deploy further, and/or more granular, monitoring NFs in the affected areas of the infrastructure to ensure normal behaviour is restored.

5.2.1.2 Results

With typical operational data flowing in this simulated experiment, a flooding incident was introduced from H_6 as seen in Figure 5.2. This was performed using iperf [144] in UDP mode for a prolonged 60 second burst at 5x operational traffic levels to all hosts on the network: H_1 to H_5. Experiment parameters were set with $MAX\_NEW\_ROUTES\_THRESHOLD = 2$. As the UDP flows began, the controller added new routes from H_6 to H_2, H_4 and H_5 (routes to H_1 and H_3 are already present). This took place within a five second timer event.

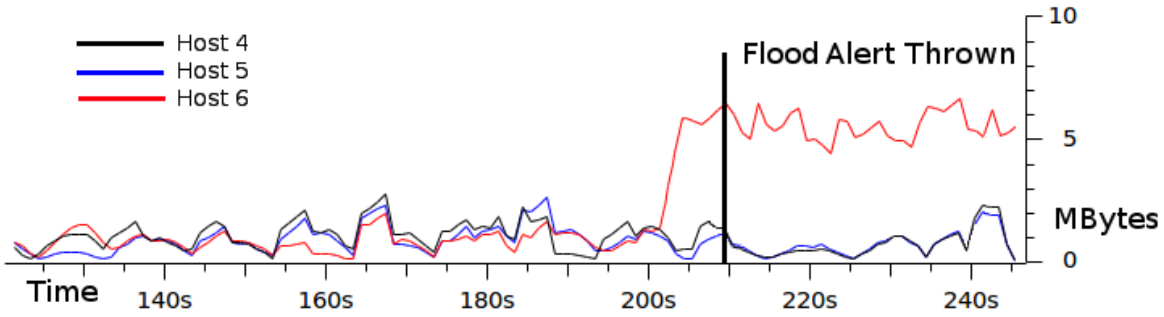


Figure 5.2: Results of flooding and detection time plotted against normal operational traffic

After the flood has initiated, the next time the timer event is triggered, a *Flood Characteristic Alert* notification is successfully thrown to the network monitoring at the controller. Considering the lengthy outages which have occurred through flooding events in the past, this experiment successfully proved the concept of such an SDN architecture in the ATM domain. Other techniques to detect flooding incidents exist. Preventive approaches such as VLAN isolation or other means of blocking access between hosts can be implemented, e.g., firewalls. However, flooding is one challenge from many which can adversely impact the ATM infrastructure. While other techniques exist, the implementation of these can involve manually distributed hard-coding of policies which can be overly restrictive and unresponsive. By exploiting the decoupled control plane approach, a flooding event can be recognised within a few seconds and presented an alert detailing the switch at the root cause. The operator then has detailed information and can act rapidly.

This experiment highlights the benefits available to ATM engineers aligned with the earlier design requirements, through the use of SDN+NFV-enabled networks, specifically the

OpenFlow implementation.

5.3 Detecting Anomalous Behaviour

As ATM providers strive to increase capacity and safety, attention is turning to detecting abnormal behaviour both at the network and at the application levels examining contents of data traffic. Using Deep Packet Inspection (DPI) to read this application level information, e.g., aircraft telemetry, anomaly detection algorithms can highlight abnormal behaviour observed in the various ATM services. Cybersecurity is also an area of real concern to all critical infrastructure providers, including ANSPs. In this section, this work is evaluated with respect to its on-demand enabling capabilities for a range of application-level anomaly detection scenarios, using data analysis performed as part of this work to determine normal behaviour patterns.

5.3.1 Monitoring Surface Aircraft Movements

5.3.1.1 Air Navigation Service Provider Needs

The FAA have identified ground movements of aircraft as an area where disruptions can be reduced through better management. When aircraft are taxiing on runways and around terminal areas, there should not be a need to rapidly accelerate, decelerate, change their heading or position as compared with their normal, steady behaviour. If there is a rapid deceleration for example, this could be indicative of sharp braking which is likely in order to avoid another vehicle. Such incidents may seem minor, but if near-misses do occur, they can increase delays. For example, aircraft may need to be checked by inspectors for any potential damage or surface debris.

The ground-based surveillance system for airport operations in the U.S. is called the Airport Surface Detection Equipment, Model X (ASDE-X) system. ASDE-X monitors real-time airport surface movements and 5 mile airport airborne operations using a variety of sources

including sensors, satellite and radar, and uses transponders to identify each aircraft and ground vehicle. ASDE-X was first deployed in the U.S. in 2003[1].

Currently, ground surface operations anomaly detection takes place offline using sophisticated algorithms which can be used for retrospective analysis of incident reporting and future scenario planning. Using the architecture presented in this work, the capability of deploying a more lightweight detection algorithm is shown, using NFs which can give operators notifications of incidents more quickly, albeit with a lower level of certainty than can be achieved through more thorough and compute-intensive offline processing. By having a better impression of incidents in real-time, ATCOs can understand emerging behaviour and take remedial or preventive actions, which could help stem further incidents occurring. No data is available to quantify to what extent this approach differs in accurate detection compared with the current offline systems. However, speed of results versus certainty is a common trade-off with each approach providing different merits.

The detection presented in this section uses the Exponentially-Weighted Moving Average (EWMA) algorithm to detect significant rates of change in normalised data. This significant rate of change could be sharp breaking or turning, for example. The significant rate of change is measured against previous recorded behaviour for a given aircraft or ground support vehicle, with and upper and lower bound trends, calculated by the EWMA algorithm.

ATCOs have real-time displays for aircraft and trajectories for their future activity. Understanding previous trends and behaviours is a task for other staff members and more sophisticated algorithms. By distributing ASDE-X significant rapid change detection modules, NFs can be kept lightweight and ensure real-time information on probable abnormal behaviour is delivered to ATCOs as quickly as possible. Distribution also divides and parallelises the workload for running the detection NF against all aircraft and vehicles on the airport surface and terminal approaches. NFs can run the detection algorithm only on the aircraft and vehicles observed in the traffic from the sensors and data gathering services which pass through the NF. Using virtualisation also ensures that these detection NFs can be more easily up-

[1] `https://www.faa.gov/news/updates/?newsId=65497`; Accessed: October 2016

dated than the infrequently altered safety-critical systems involving delivering information to ATCOs terminals. Updates will be necessary as the future systems for ATM are increasingly evolving with numerous versions of sensor systems, such as Automatic Dependent Surveillance-Broadcast (ADS-B) a satellite-based surveillance system, and increasing numbers of types of sensors to deliver more information to operators. For example, sensors to perform wake vortex calculations at runways and sensors compatible with the latest UAV models and autonomous ground vehicles. This function deployment architecture allows the chaining of multiple NFs which among many possibilities, would enable the results of two different versions of a detection NF to be compared. This alleviates some existing issues due to a lack of realistic test bed environments.

5.3.1.2 Experiment & Data Analysis

This experimental analysis used real ASDE-X data from historic U.S. airport recordings. The first task in creating the NF, was to examine the overall dataset and recognise the events that would be beneficial to detect. Through discussions with experts in this field [43], it was established that examining the rates of change in normalised acceleration, position, heading, and the number of records observed using an EWMA algorithm, would provide valuable insight of abnormal behaviour. This information was represented in the telemetry as:

- *absoluteTime*
- *heading*
- *velocity.X*
- *velocity.Y*
- *velocity.Z*
- *position.X*
- *position.Y*

- *position.Z*
- *acceleration.X*
- *acceleration.Y*
- *acceleration.Z*

Given this data set, the parameters were calculated as a change per second irrespective of arrival time periods of packets. With consistent time windows changes can be compared.

Values were also calculated for the number of records observed per second and the change in heading. Care was taken with heading data to ensure a change from 0° to 359° was represented as a change of magnitude 1. Without recognising due north, 360°, is equivalent to 0°, the rate of change would be of magnitude 359.

When processing the telemetry information it was confirmed by domain experts [43] that a combination of *Callsign*, *Timestamp* and *Aircraft_Type* was sufficient for unique identification of a vehicle. *Callsign* and *Aircraft_Type* were sent as an array of ASCII character codes. With individual vehicles identified, their readings are separated and stored. The EWMA implementation based upon description by Ye [145] takes a time series of numeric values and looks for abrupt changes. The EWMA predicts the next value and the standard deviation of the Exponentially-Weighted Mean (EWM) sets the upper and lower trends around the predicted value. If the value exceeds these trend bounds then the data point is considered anomalous.

With the ability to identify unique aircraft the EWMA algorithm could be applied to each parameter of interest for each aircraft, the NFs can work independently acting only on the information they receive. This keeps the NFs lightweight and flexible. To ensure functionality in a resource-bound NF, the storage of past information required for the EWMA algorithm, is kept in a fixed size stack. Using these fixed sized stacks with the current implementation leads to the first and last points being marked as abnormal. These are ignored when sending notifications. However, if these points still happen to be identified as anomalies when they are not the latest point in the window, then they are reported.

5.3.1.3 Results

In this section, a number of representative graphs show the type of anomalies the NF can detect across the dataset of several U.S. airports with thousands of aircraft. Similar time windows are shown on the x-axis of each graph. The results represent different aircraft in different airports, in this chosen time window across the dataset. An anomaly from each of the chosen parameters is shown. Figure 5.3 shows the high reading of 0.35. While significant jitter can be seen in the prediction line, this high value clearly exceeds the behaviour recorded in the recent time period. The change of acceleration is represented as a magnitude, so it is unknown if this is a rapid acceleration or deceleration. For the purposes of reporting a potential incident this is not necessary and could be added later if desired.

A granular view in Figure 5.4 shows a heading change of ~17° within one second. This is a significant absolute change and exceeds the recent recorded values. However, the aircraft is turning and domain expertise may state this is within an acceptable rate of turn. Figure 5.5 shows an abnormal reading shortly before *14:22:14*. The recording of this data point, skews the upper bound of the EWMA algorithm significantly allowing a greater margin of deviation from the prediction. This means that if an aircraft begins to move more quickly, subsequent actions are classified relative to recent behaviour.

The provided ASDE-X dataset had been pre-processed and therefore does not have any known measurement faults or outliers caused by sensor errors or equivalent variability. The data points identified as anomalies represent significant change in aircraft behaviours as against their recent activity and based on EWMA parameters.

5.3.1.4 Discussion

This NF highlights how this work can help operators improve future ATM systems through greater flexibility and enabling operators to deploy distributed virtualised functionality on-demand that can deliver a notification system to better improve the safety and capacity of air traffic. While the results presented do accurately show abnormal behaviour as detected by the EWMA implementation, further input by a domain expert to tune the algorithm parameters

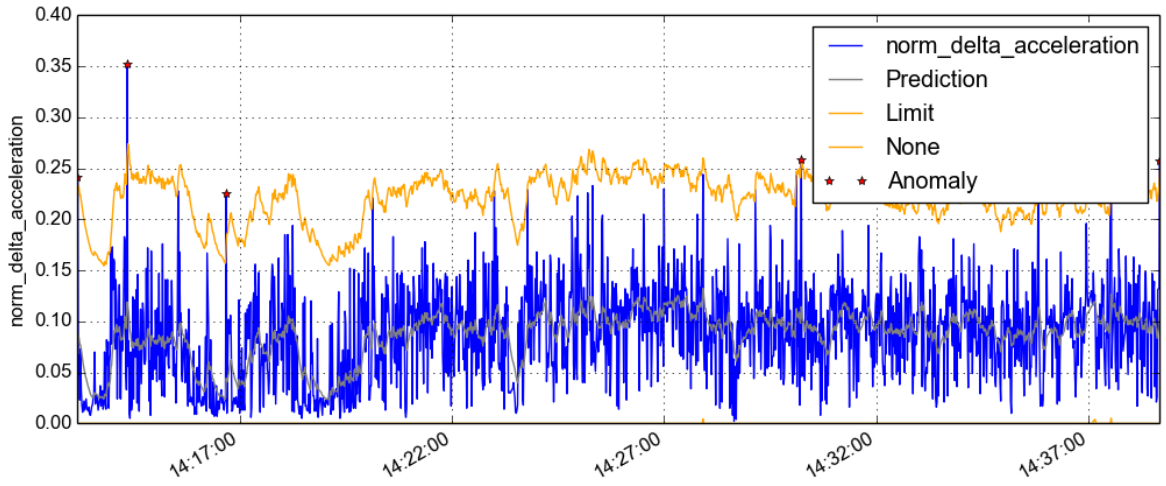


Figure 5.3: The normalised rate of change in acceleration over time for a given aircraft

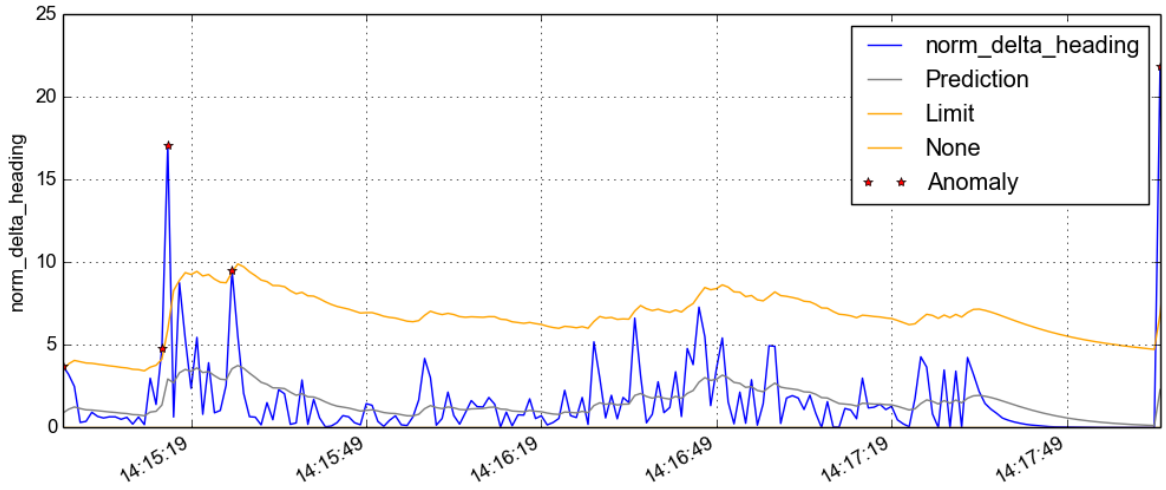


Figure 5.4: The normalised absolute change in heading over time for a given aircraft

would be advantageous in order to ensure the optimal response to given scenarios. The function deployment architecture coupled with this anomaly detection NF provides operators with the foundations from which to build more highly tuned detection NFs. Domain expertise may also help advise whether normalisation over a longer period, or, for example, in using the maximum of the readings for a given period versus the average is more appropriate.

Having an initial training window for the algorithm is necessary for the algorithm, however it is also very beneficial in ensuring that the chronological storage of readings are achieved based on the time the reading was taken, and not the packet arrival time. This means any deviations in inter-arrival times or out-of-order packets will not interfere with the rate of change observed.

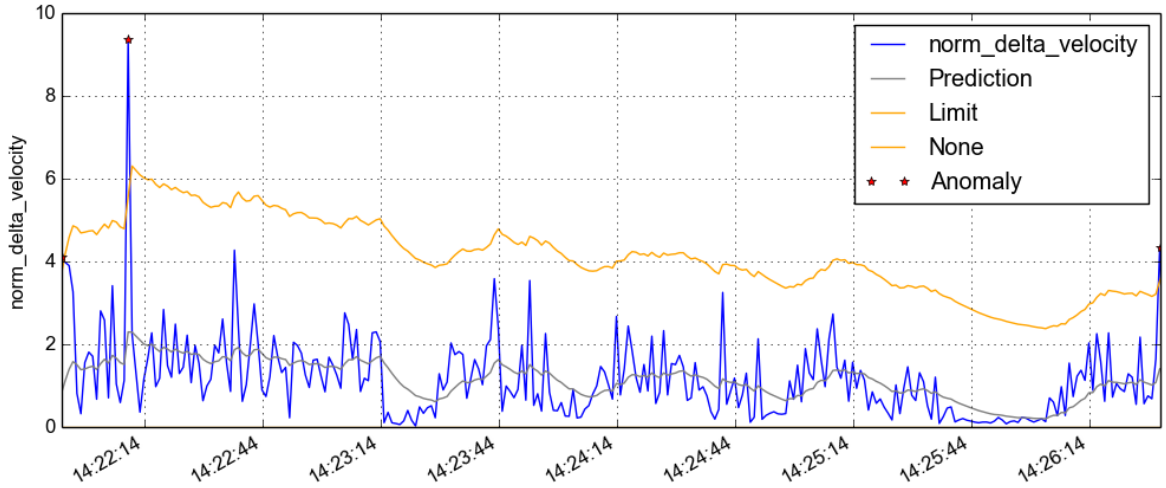


Figure 5.5: The normalised rate of change in velocity over time for a given aircraft

Lastly, it is possible this NF could also detect whether a landing was smooth or not. Since ASDE-X data records the telemetry of aircraft up to 5 miles around an airport, the rates of change when landing could determine if the landing was smooth or not. This is valuable information for ATM research in general, with a great deal of focus currently on wake vortex modelling for landing and take off, with the goal to increase airport capacity by landing aircraft closer together. By understanding the smoothness of a landing this may give yet another informative reference for operators around the activity of the aircraft as further research is conducted.

5.3.2 Combining Information Streams to Define Normal Behaviour

5.3.2.1 The Proportionality of Secondary Surveillance Networks

An early concept developed in this work was that a sensitive anomaly detection algorithm could be developed based on the relationship between the application level information transmitted and the network level data. By tightly coupling an analysis on these complementary information flows, abnormal behaviour such as unexpected data traffic could be identified very rapidly.

To better understand the context of surveillance system networks, including the highly distributed nature and long distances involved, Secondary Surveillance Radar (SSR) coverage

for the UK was mapped using publicly available information. Figure 5.6 shows the estimated coverage on Google Earth[2], with radar sites located within the yellow circles and the white circles indicating coverage. The map was created based on the publicly available NATS surveillance wind turbine interference self-assessment data[3]. From this data, and the provided information that wind farms cannot safely be built within ~14km of SSRs, the central points of these interference areas, i.e. SSR sites, can easily be deduced. Typically, these radar dishes take ~7.5 seconds per revolution, and a domain expert's estimate of 10 nautical miles range per second of revolution[4], lead to 75nmi (~140km) radius coverage areas being plotted. The results were quite accurate, as can be seen in Figure 5.7 which shows the calculation of an SSR being ~200ft from the aerial photograph of the SSR. All the locations were tested and had a similar accuracy. Overall, performing this analysis clearly showed the high levels of resilience through the diversity of SSR hardware and in the significant redundancy of coverage. It is likely airport radars have a higher rotation speed than en-route radars, which have a slower rotation rate and therefore a higher range. The higher rotation speed gives a more frequent determination for terminal approach and congested areas. Therefore, the level of coverage shown in Figure 5.6 around the London airports area is likely to have some SSRs with a shorter range but with more frequent updates for ATCOs.

Understanding the highly distributed nature of a typical SSR network gives better insight to how best to structure the placement of NFs for improved flexibility and monitoring. Using recordings from an SSR UK en-route radar and the corresponding network traces which sent this information over the network, correlations were examined. ATM surveillance data networks are built for a single purpose: to transfer the radar information over long distances to various, distributed radar processing points. The data is transferred over two mirrored links and the data traffic is combined at radar processing points and delivered to various ATM services such as the ATCOs terminals. The data is combined by merging the traffic from the different links, ensuring if packet loss has occurred on one link, it can be taken

[2] `https://earth.google.com/`; Accessed: September 2016

[3] `http://www.nats.aero/services/information/wind-farms/self-assessment-maps/`; Accessed: September 2016

[4] `http://aviation.stackexchange.com/questions/115/what-is-the-range-and-accuracy-of-atc-radar-systems`; Accessed: September 2016

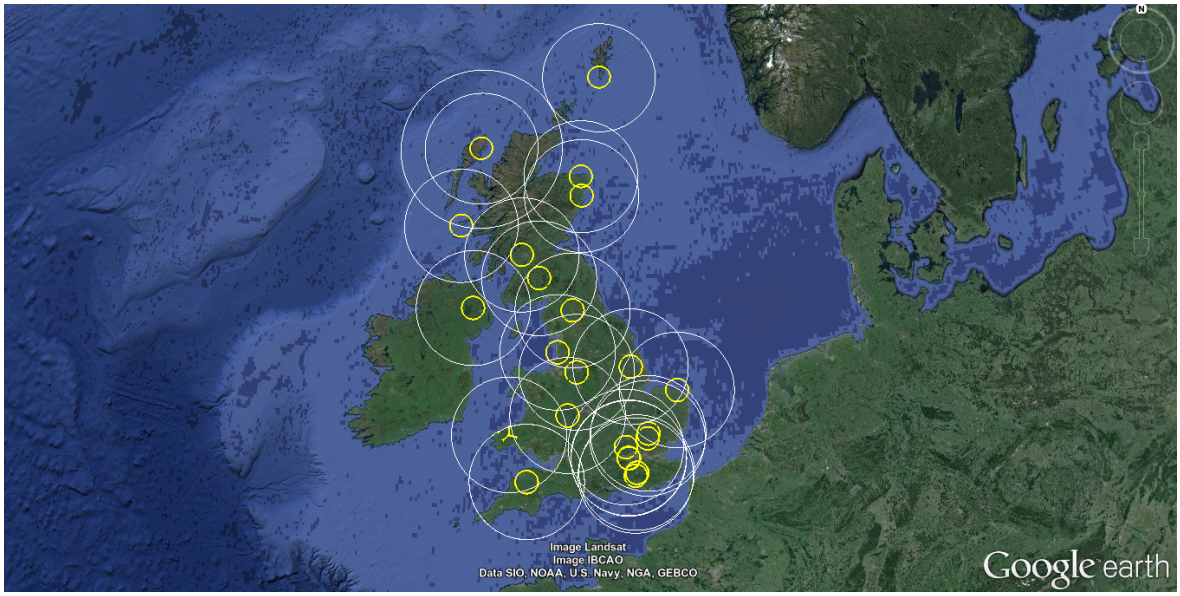


Figure 5.6: Radar Surveillance estimate coverage map of the UK showing high levels of redundancy and a highly distributed network

from the other link. The greatest levels of network data loss for SSR information are due to the long distances involved [43]. By parallelising the transfer of mirrored SSR data, the loss is drastically reduced by combining these feeds at the destination point. Once processed by the radar processing points, the information is sent over the main WAN to ATCO locations. SSRs must be so extremely geographically distributed in order to provide coverage for all of the airspace under the control of the ANSP.

To detect abnormal behaviour in a timely manner, yet without adding complexity to remote SSR locations, placing distributed NFs at the radar processing points, after the recombination of the mirrored data streams, is optimal topologically. Locating NFs at the SSRs increases the complexity of their systems, increases the risk of information loss for, e.g., anomaly notifications over the long distances, and due to the logical single path connectivity from SSR to radar processing points, will not allow for any potential gains from detection to delivery, as opposed to locating NFs at radar processing points. At processing points, NFs will have potential access to both the incoming streams and the recombined stream from each SSR, depending on their location within the information flow. This ensures anomalies can be detected rapidly, before the information is transferred over the WAN to ATCO terminals, and gives flexibility with respect to where to send notifications in the network. Placing NFs

Figure 5.7: The accuracy of the SSR locations using showing the calculated location and the photographed location

elsewhere on the WAN would be less beneficial as the immediate relevance of this real-time information would likely have already passed by the time the analysis would take place. This is because, once processed, the radar data is transferred to ATCOs. While the anomalies detected in this case are more of a cybersecurity issue and therefore more likely to be brought to the attention of the ATM systems engineers as opposed to ATCOs, as discussed earlier, ATM engineers are typically co-located with ATCOs in order to solve any problems which may arise both on site and with greater immediacy.

5.3.2.2 Data Analysis

Based upon a month's recording of NATS en-route SSR data and network trace, the correlation between the number of aeroplanes observed (uniquely identified by Callsign, Aircraft type and Timestamp) and the number of bytes transferred over the network to the radar processing points was examined. The distribution of this month of recordings can be seen in Figure 5.8. The figure clearly shows the daily peaks (~2000-3000 aeroplane observations) starting to build before the working day and ending in the evening, before periodic low numbers of observations, and therefore aeroplanes, during the night. There are also some more subtle weekly periodicities. An observation to note in Figure 5.8 is that there is some data loss from lack of recording on the sixth morning, from the previous minimum to the daily maximum.

The network trace primarily consists of UDP packets with some regular Internet Group Management Protocol (IGMP) packets of negligible size interspersed. UDP is used instead of TCP, since TCP introduces timing variability through its flow control mechanism and ATM systems need the very latest information without delay due to the real-time nature of ATM operations. The issue arises when data packets arrive out of order. For a time-critical service, such as surveillance, the effect is the same as data loss. The maximum delay from the radar site to processing is 20ms for this national ANSP [43]. Packet lengths are reasonably consistent with all packet sizes within 40-1279 bytes, 40% of which lie between 40-79 bytes. As stated, the SSR recordings were taken by a radar dish which makes one full revolution approximately every 7.5 seconds. Therefore, as an aircraft passes this radar site, it is observed numerous times as it progresses on its flight. The total number of distinct aircraft identifications observed in the month was 21,455. In Figures 5.8 and 5.9 there is a far greater number of aeroplanes represented since these figures show observations of aircraft, not distinct aircraft. Therefore, the proportionality between network traffic and air traffic is dependent on the constant factor of the rotation of the radar dish.

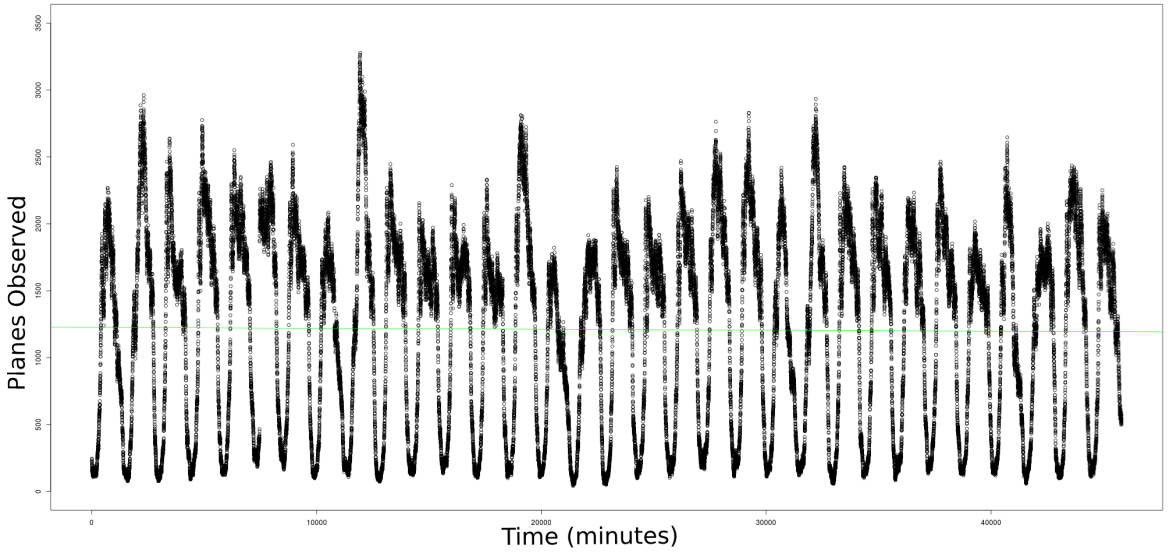


Figure 5.8: The distribution of numbers of aeroplanes observed by radar, aggregated over 60 second intervals over 1 month

5.3.2.3 Secondary Surveillance Anomaly Detection NF

Using the technique developed in observing the correlation between the aeroplanes observed and the number of bytes, a sensitive anomaly detection NF was developed. Figure 5.9 shows the correlation for the month of October, using 60 second aggregations, a granularity that gives a strong correlation, yet enough granularity to observe deviations from the perfect proportionality. The links are typically 2Mbps copper and are therefore under-utilised as expected from best practice critical infrastructure provisioning.

The upper and lower limits seen in Figure 5.9 represent the bounds as defined in the anomaly detection NF for normal behaviour. They are chosen as intentionally simplified bounds on the observed traffic. Enhancing the sensitivity of these bounds can be better achieved with further data, however they form a good approximation for the available data. They are defined as follows:

- Upper limit: $y = \frac{1}{15}x + \frac{50}{3}$
- Lower limit: $y = \frac{1}{20}x - 25$

where x is the number of *Aeroplanes Observed* and y is the *Number of KBytes*.

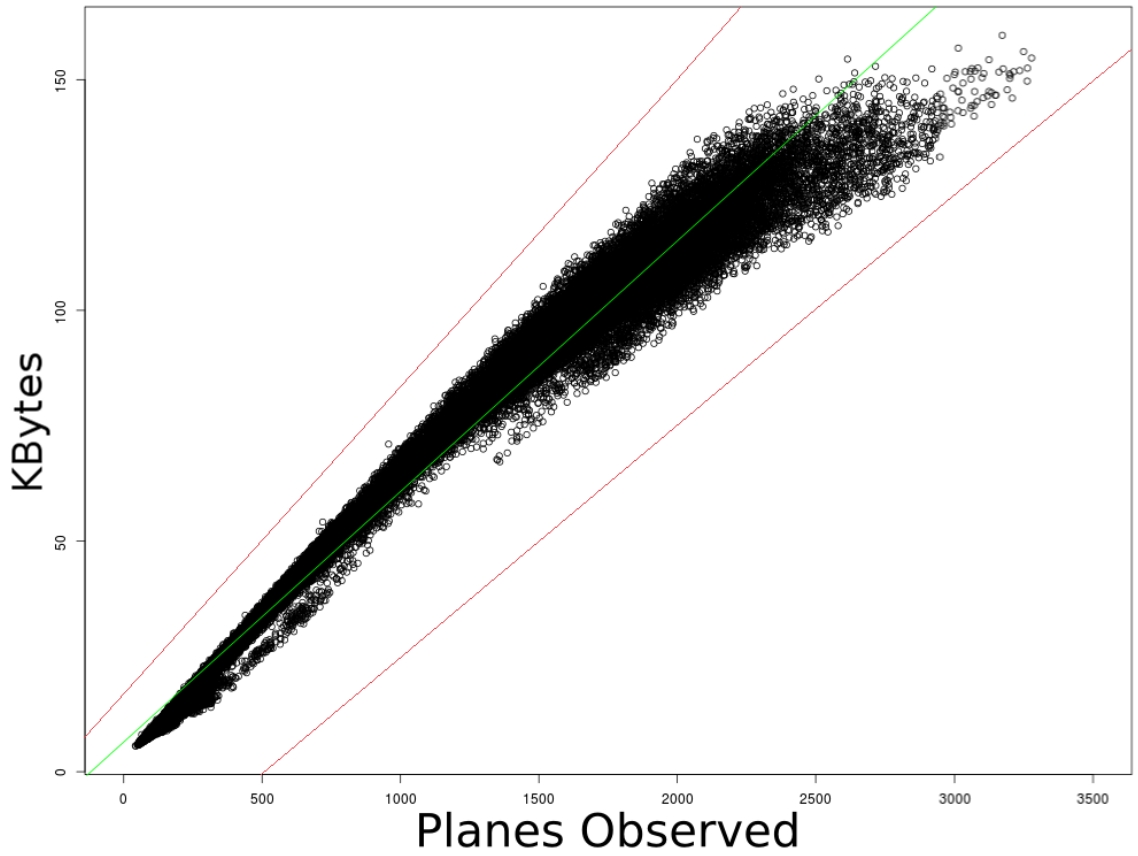


Figure 5.9: Correlation between numbers of aeroplanes observed by radar and bytes transferred, aggregated over 60 second intervals over 1 month

These limits assume the observed data in the recording was all normal behaviour, but this was confirmed in discussions with operators [43]. Intentionally, the upper and lower bounds are not tightly following the data observed, this is because this data is only for one month, October, which has seasonally low traffic. During the summer there are a greater number of recreational aircraft which may have a different correlation due to the information they send from their transponders. More widely, this work is largely in anticipation of the enormous change expected in ATM systems, not limited to the introduction of predicted high volumes of UAV traffic. Such UAV aircraft may also have different correlations, for example, they are likely to stream large volumes of telemetry to remote pilots in command. As a result, this detection NF will likely either need to add a traffic filter before analysis, or the general algorithm can be adjusted to encompass a modelling factor for these potentially different data relationships. In general, specifying the exact bounds for such detection can be performed

with greater domain expertise and will always need to evolve over time. This work presents the capability for ATM engineers to quickly and flexibly detect such behaviour, easily update their monitoring and detection systems without interfering in mission critical systems and be notified of abnormal events which may indicate emerging problems which in turn, could cause disruption, resulting in reduced air traffic capacity.

5.3.3 UAV Telemetry Anomaly Detection NFs

5.3.3.1 Domain-Specific Requirements

Earlier Sections, 3.4.3 and 4.2.2, encompass the design considerations and implementation details explicitly tailored to future needs of UAS, respectively. UAVs will become an integral part of global ATM systems in the future. Currently, there are many UAV types and common characteristics include the streaming of telemetry from the UAV to remote pilots and high mobility deployments. UAVs are typically built to be computationally lightweight, in order to reduce the complexity of systems on-board, with current, limited, monitoring capabilities taking place on the remote pilot controller systems. Similarly to ATCOs, UAV operators' primary concern is interpreting and responding to the latest situational information. Analysing past trends and archiving data can be done, if desired, at the controller systems. This is typically only used for incident investigation. As a result, monitoring capabilities supplied by NFs should also be focussed on the rapid delivery of information and therefore can in turn be more lightweight. It is likely that many future UAV systems and applications will require UAVs to migrate from one Ground Control System to another GCS pilot. Such functionality is already built-in to, e.g., the Viking 400[5] UAV, and is a feature readily used in current military and environmental UAV applications. It is therefore desirable that tailored monitoring NFs, tuned to a specific type or model of UAV, could also be migrated among the distributed GCS. Migration is made possible in this work through SDN and flexible routing using OpenFlow-enabled switches. Since the primary concern of UAV pilots is real-time, the NFs can be stateless, making migration more lightweight, unlike, e.g., VM migration retain-

[5] `https://airbornescience.nasa.gov/aircraft/Viking-400`; Accessed: October 2016

ing state. NFs would therefore process just the information passing through them, providing on-demand functionality. Being able to flexibly distribute monitoring functionality around a vastly distributed network of GCS as UAVs are carrying out their missions will greatly enable mission operators. The mission operators and individual remote pilots will have greater information available, and GCS resources can be utilised for their desired functionality at a given time. Since UAVs can take on a number of diverse tasks at different times, from crop dusting to thermal imaging, different monitoring capabilities will enable more sophisticated on-demand capabilities across missions.

Placing such functionality in the UAV controller systems would be far harder due to their inflexibility to update. UAV controller systems offer monitoring, in that they display live telemetry readings and alerts, such as low fuel warnings, but these systems are not easily updated or editable due to long rigourous safety focussed development cycles and vendor lock-in, respectively. Since the UAV controller software deals primarily in the pilot to UAV communications, any changes to these systems require significant testing and is often very highly coupled with the hardware e.g. the radio antenna. Through this work, virtualisation within the NFV framework, provides open innovation and will therefore allow for future UAV ATM integrated systems to better manage emerging UAVs through flexible monitoring solutions, thus ensuring innovation whilst retaining airspace safety.

5.3.3.2 SIERRA Incident Case Study

In 2013, NASA was running a project called the Marginal Ice Zone Observations and Processes Experiment (MIZOPEX) to better understand Arctic sea ice during the summer using UASs. The MIZOPEX reconnaissance mission involved multiple UAVs of different types. One of the UAVs deployed on this mission was the Sensor Integration Evaluation Remote Research Aircraft (SIERRA). Prior to this project, the SIERRA had been deployed on several successful missions.

On 26 July 2013, the SIERRA UAV lost engine power 4.5 hours into its 6 hour scheduled flight and crashed into the Beaufort Sea, 65 nautical miles north of Oliktok Point,

Alaska, where the controlling GCS was located. The NASA mishap accident investigation report [146] found that the only indications to the SIERRA team of the impending crash, via the Pilot In Command controller system display, a part of the GCS, were the A/C engine's Revolutions Per Minute (RPM) reading of zero RPM and the electrical bus voltage at 24V. SIERRA would have normally been at 6,000 RPM and 28 volts. The 4V lower voltage confirmed the engine had stopped turning. At this time, the report states no pilot instructions could have avoided the loss of the UAV.

Surprisingly and fortuitously for the incident investigation team, the SIERRA wreckage was successfully recovered, allowing analysis on the fuselage, fuel contents and other standard practices. The controller systems also kept a log of all telemetry received from the UAV. While not all available information is shown on the pilot display as part of the controller system, all the data is stored.

On further analysis of the telemetry prior to the engine failure, investigators discovered the throttle demand increased by over 40% and continued to rise, while the engine struggled to maintain its cruising RPM of 6,000 as much as one hour prior to the crash. This can be seen in Figure 5.10, taken from the investigation report [146]. With 6,000 RPM, 0.15 throttle and consistent 29 m/s True Air Speed (TAS), a 40% increase in Throttle and steady RPM, this was clearly abnormal behaviour. This information was not displayed to the pilot through the real-time GCS information. The RPM also plummeted at times to anomalous lows of 4,000 and the engine behaviour was described as sporadic. Figure 5.11 shows in more detail the final couple of minutes of flight. As the RPM declines, the throttle is increased to the maximum but with no corresponding increase in RPM. There is also a ~40 second period where communications, and therefore telemetry, was lost, seen prior to and around 303 minutes into the flight time. Had the team been notified of these anomalies and returned SIERRA to base, the report concludes the mishap could have been prevented.

Prior to the SIERRA crash and following the 40% increase in throttle, there were eight *ice warning* alerts in a 25 minute window, a significant increase in frequency. In discussions with domain experts, it was clear this sensor had a significant safety margin and, for operations in

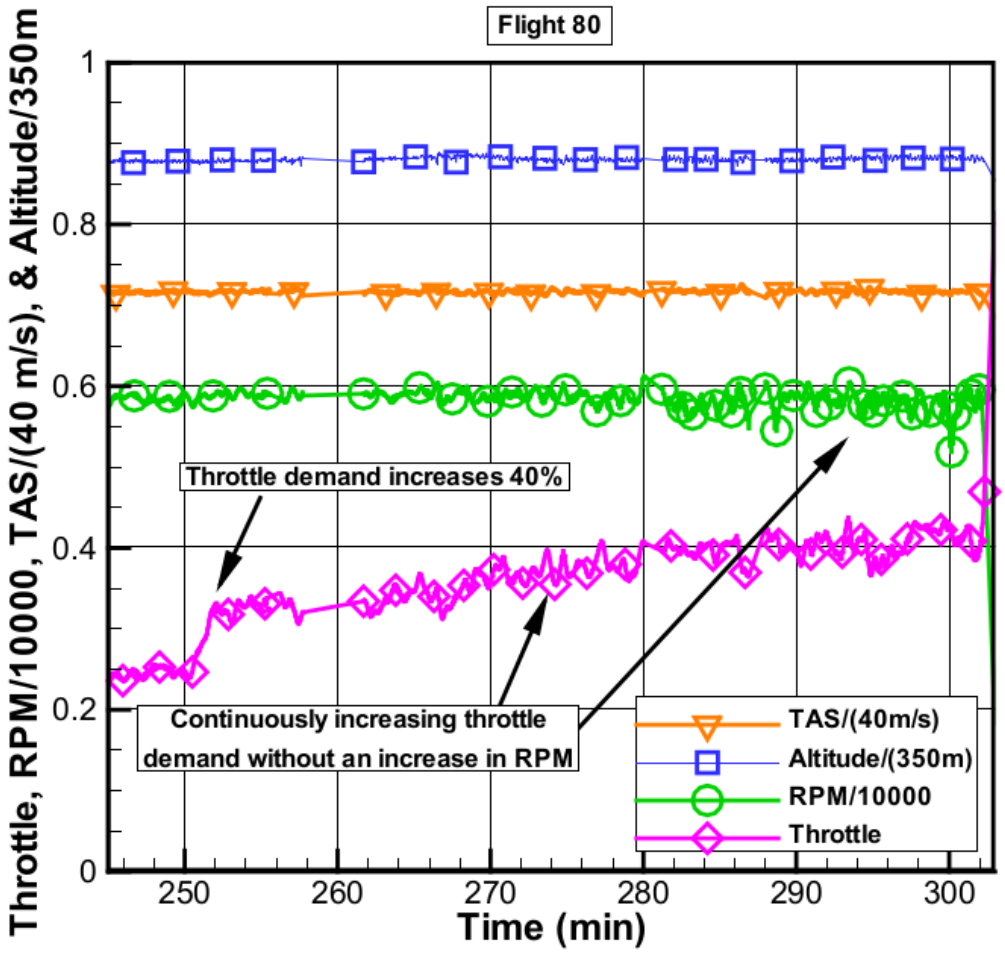


Figure 5.10: Annotated telemetry of the SIERRA UAV prior to loss of control [146]

cold environments, this had to be ignored to some extent.

An initial consideration was for an NF focused on the issue of UAVs saturating operators with alerts of current operating conditions, reducing their situational awareness. For example, a UAV with fuel reserves for 10 hours of flight, and a warning notification built into the aircraft hardware to notify the pilot every minute when fuel levels are below a threshold, e.g., $< 10\%$. Under planned or emergency circumstances where these warnings would come into effect, it is likely this notification frequency would be an unhelpful distraction to pilots. To mitigate this, an initial NF was developed which aggregated such notifications ensuring that, when under special conditions which may have demanded pushing the UAV beyond normal operating thresholds, the pilot would not be adversely distracted. The NF allowed for the setting of new thresholds in software, which were easily programmable and adaptable during live deployment, unlike those set in the UAV hardware sensor systems. In general,

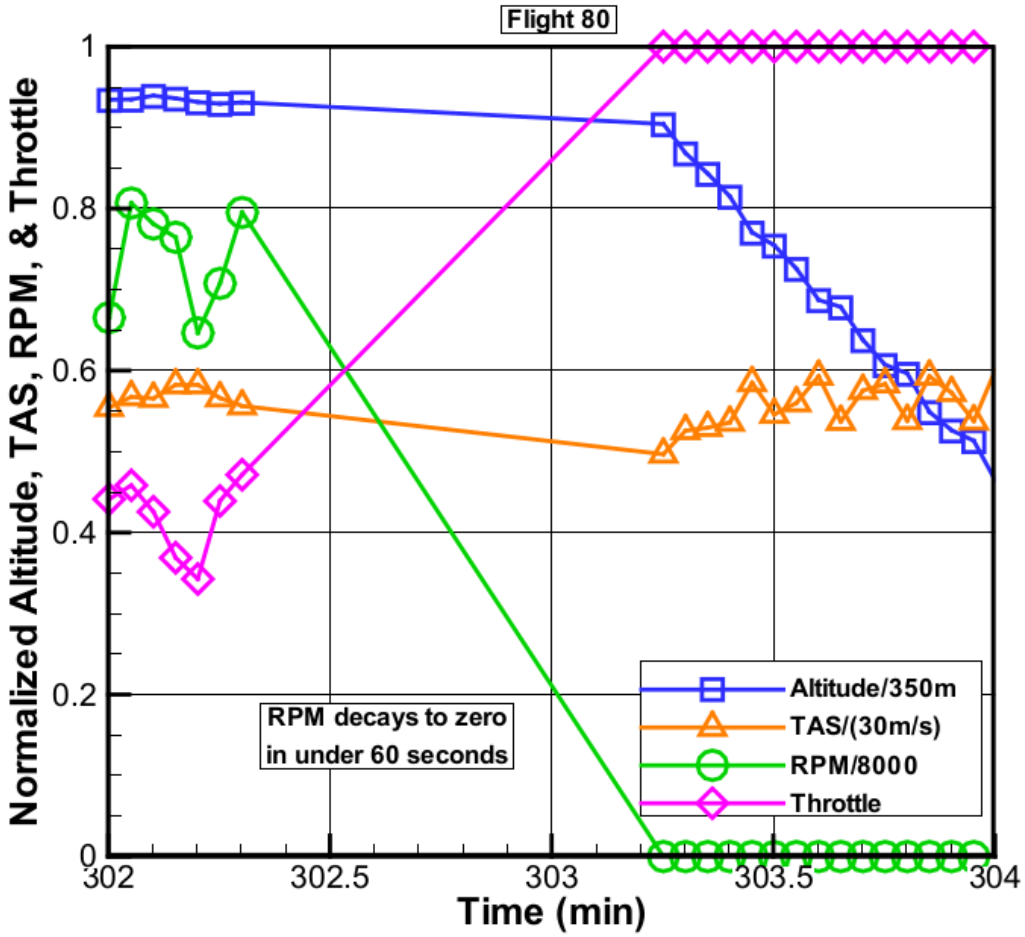


Figure 5.11: Final minutes of SIERRA flight telemetry including loss of communications link [146]

modifying the telemetry from the UAV, such as proposed in this case, e.g., filtering packets prior to reaching the controller systems, is not considered good practice and interferes with the vital task of logging all communications [146].

Overall, the SIERRA incident highlights the need for greater monitoring and anomaly detection systems. Without interfering with the communication channels, further analysis led to the design of a better NF to assist in such a set of scenarios.

5.3.3.3 Further Analysis Defining & Classifying Normal Behaviour

The telemetry of all UAVs have multi-variate inter-dependencies, with physics underlying the models of many of these such as the relationship between altitude and pressure, altitude and fuel burn rates, and outside air temperature and internal temperature readings.

These models are an excellent definition of normal operating behaviour which can be used to rapidly detect unexpected, anomalous behaviour. While operators receive real-time readings of many of these values, it is often in the relationships and trends between these parameters where problems can first be observed. Using SIERRA telemetry from more than eighty worldwide flights over a number of years, initial models of these relationship were explored. An early realisation was that telemetry relationships vary significantly across the different flight phases: *Take-Off*, *InFlight* and *Landing*.

A classification was developed with input from a domain expert, a NASA UAV pilot, to determine *InFlight* status. It was determined that basic *InFlight* classification was:

- $True\ Airspeed > 26\ m/s$
- $Revolutions\ Per\ Minute\ (RPM)\ > 2000$

Figure 5.12 shows the application of the classifier to a flight recording from the dataset. The graph shows the flight profile with points classified as *InFlight* in red and other flight phases in blue. The model is very successful with near-perfect accuracy for this flight. This is seen through the flight profile, with the vast majority of red *InFlight* points with Above Ground Level > 0 and grounded, take-off and landing phases coloured blue.

As the SIERRA incident investigation found, the deviation relationship between throttle and RPM was a leading indicator ~1 hour prior to the loss of the UAV. The analysis began by aggregating the data from historic flights and applying the flight phase classification. With *InFlight* data, the next step was to examine the data with TAS of 29 m/s. Having applied both these filters, a clear relationship between Throttle and RPM was apparent. Figure 5.13 shows the results of this analysis, with a model showing the relationship between throttle and RPM in the previous 80 SIERRA flights for 29 m/s $<$ TAS $<$ 30 m/s. TAS has a significant impact on the relationship between these parameters and forms a complex model. For the purposes of this work, the SIERRA crash incident TAS of 29 m/s is used to highlight the anomaly detection capabilities NFs can provide to UAS operators.

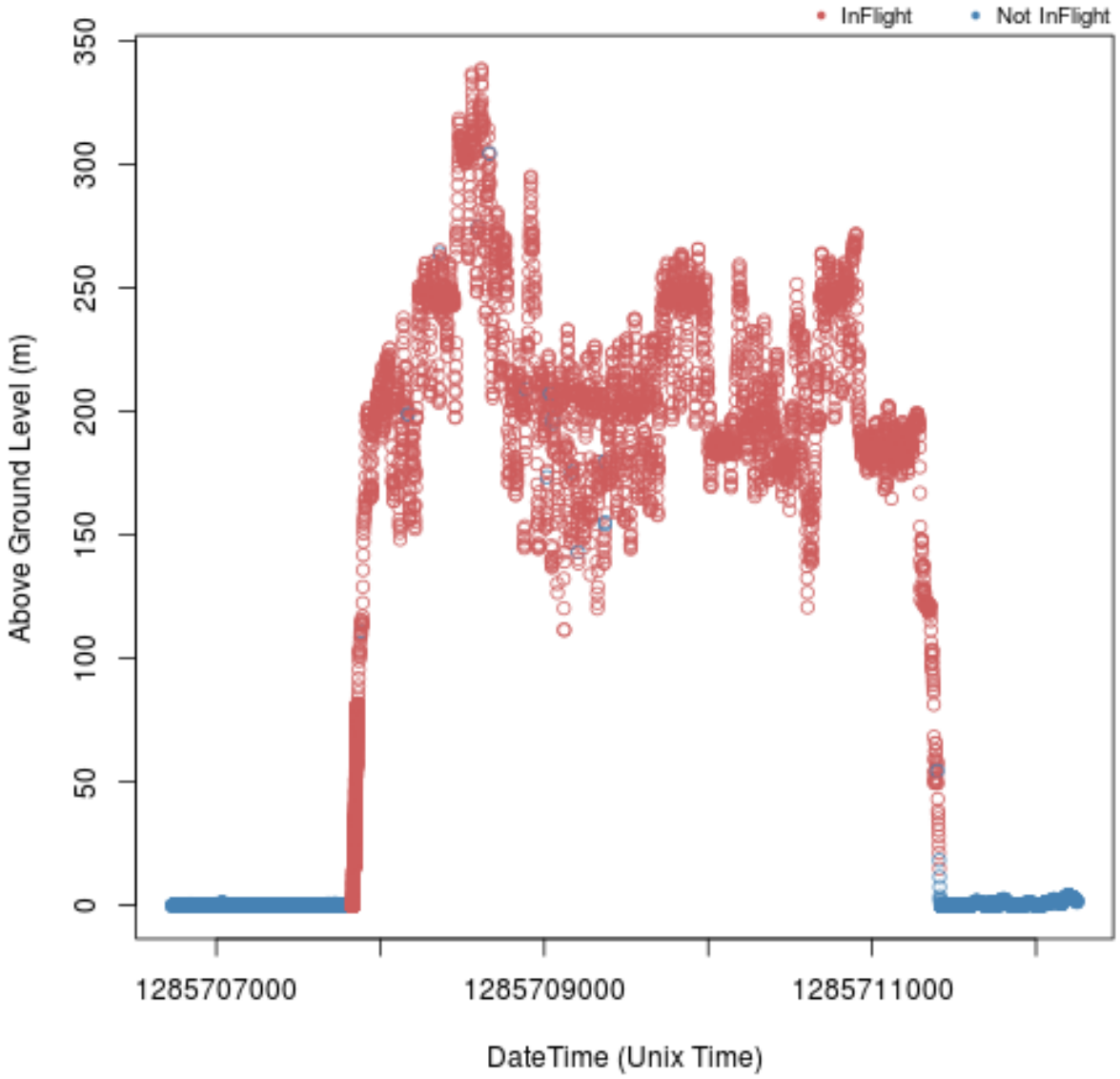


Figure 5.12: SIERRA flight classified by *InFlight* for metres Above Ground Level (AGL) over time

In Figure 5.13 the blue line shows the polynomial regression model for the relationship. The model fits quite well with the formula:

$$Throttle = 1.02811 \times 10^{-7} \times RPM^2 + (-1.199282 \times 10^{-3} \times RPM) + 3.61014$$

with a residual standard error of 0.0643 and $R^2 = 0.8948$. Figure 5.14 shows the residuals plot, approximately bell curved, and the residual versus fits scatterplot without any apparent correlations, as expected, showing that the model is encapsulating all significant variable influences. 95% confidence intervals are also plotted in green, with the two points at ~5500 RPM significantly distorting the interval, likely due to lack of data. Prediction intervals are

also shown, in blue. The anomaly detection algorithm uses the upper and lower limits of the 95% prediction interval as the definition of normal behaviour. These predictions were based upon an evenly distributed set of RPM values from the minimum RPM of 5,465 to the maximum of 8,403 RPM. The earlier classification of *InFlight* removes much of the noise seen from take-off and landing phases.

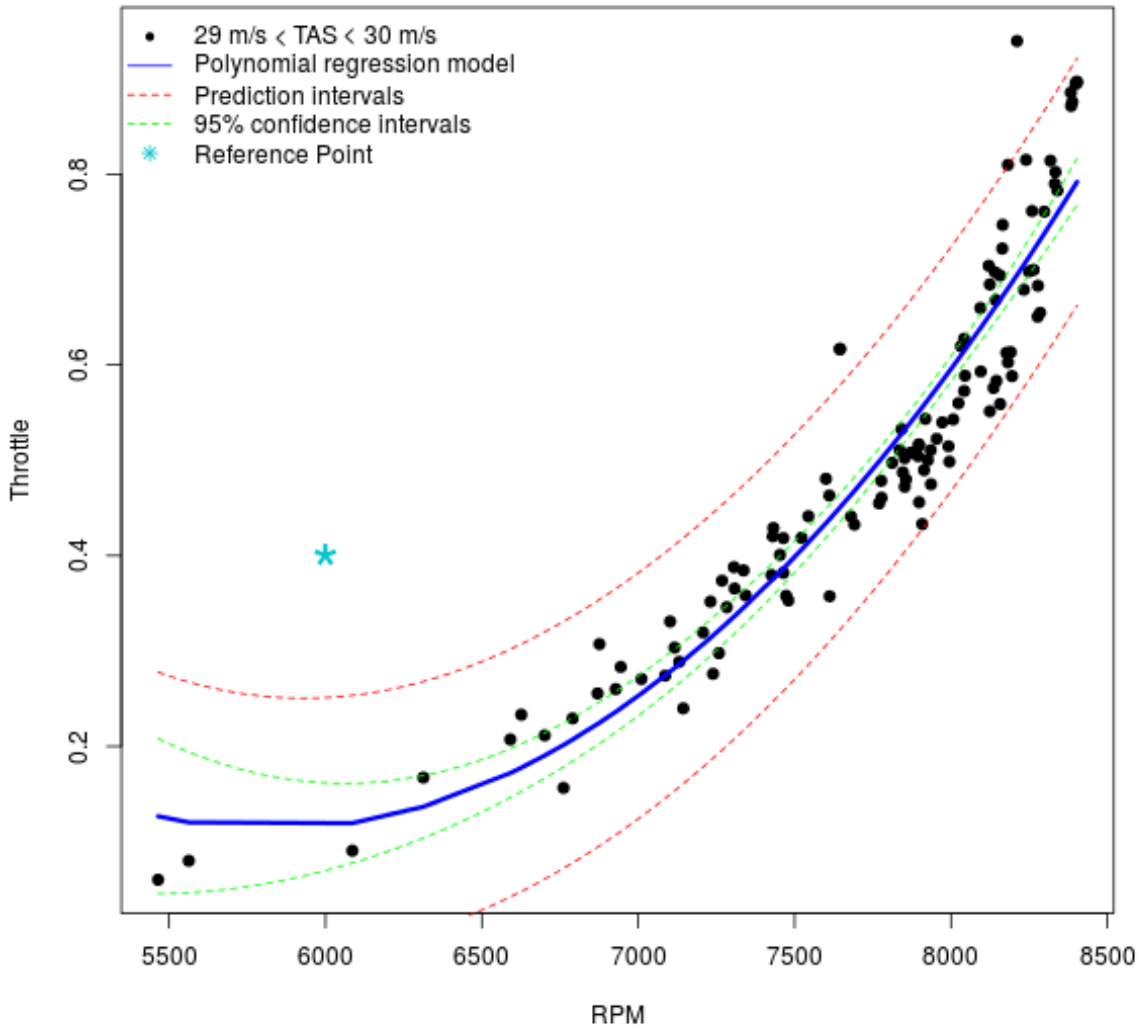


Figure 5.13: Model of normal SIERRA behaviour in the relationship between RPM and throttle telemetry with constant TAS

Figure 5.13 also includes a *Reference Point* at 6,000 RPM and 0.4 throttle. The corresponding point can be seen in Figure 5.10 where at ca. 280 minutes the throttle demand had steadily increased to reach 0.4 throttle but the RPM remained steady at 6,000 RPM. Given

the definition of normal behaviour lying within the prediction limits, this *Reference Point* is clearly anomalous. This level of deviation from the norm was continual for over 20 minutes.

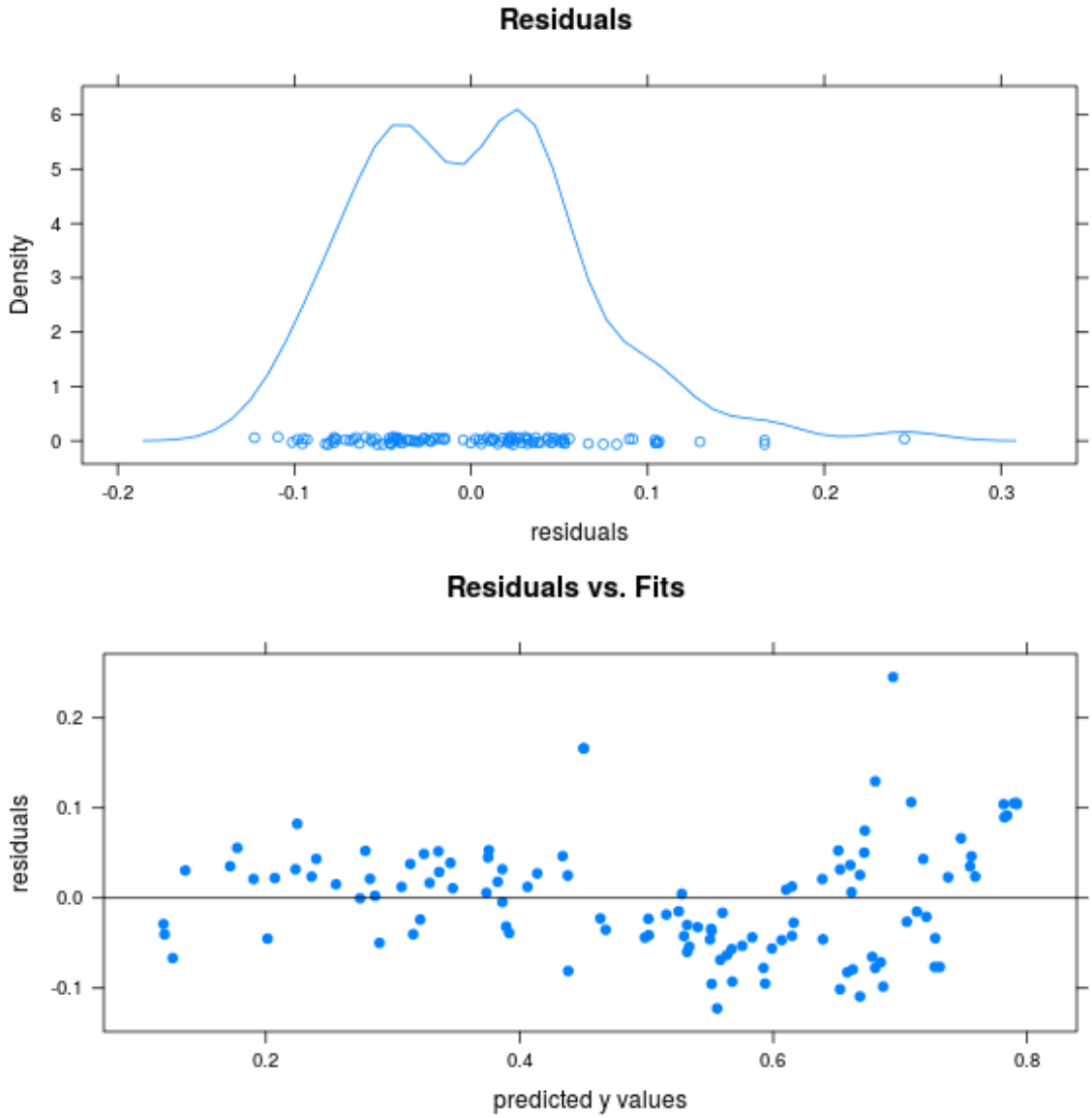


Figure 5.14: Statistical residuals and fits for the polynomial regression model in Figure 5.13, where y is Throttle

The results of this case study and the further analysis on developing deployable, virtualised NFs which can detect such abnormal behaviour in future, highlights the highly adaptable and flexible nature of systems evolution that this work enables. While the data used in this case study was calibrated to the SIERRA, the process of developing such models of normal behaviour based on previous flight data is not arduous. Once models for normal behaviour are developed among different operational parameters, NFs which provide this

additional anomaly detection capability can be deployed and offer the opportunity to avert future incidents.

5.4 Remediation Capabilities

5.4.1 Enabling On-Demand Traffic Prioritisation

When faced with network challenges, not only do engineers need to diagnose the cause and assess the impact of the challenge, but also must maintain services and systems to the best of their abilities. Often, such adverse incidents require time-critical actions to avoid compounding ATM disruption, ultimately leading to reduction in capacity of air travel or causing significant delays. Actions such as sending engineers to remote network switches to do configuration work, still common practice for some legacy ANSP ATM services, is highly impractical under such circumstances. Non-physical remote access is far more practical. SDN improves remote management capabilities further still. SDN enables engineers to configure routes in the network via the logically-centralised controller. With OpenFlow enabled switches, flow table rules can direct traffic based on its characteristics.

Without a form of Quality of Service in the network, when links become saturated such as, e.g., in the event of a network flood, the impact such as high jitter and likely increased packet loss, is spread indiscriminately among traffic. For operators, certain traffic may be more crucial at the given moment. As a result, operators may wish to prioritise certain traffic on the network, on-demand. Such prioritisation becomes possible with distributed SDN routing.

As discussed in Section 2.2.3, Newell observes the lack of QoS prioritisation capabilities in the FTI network. As Newell discusses, bulk weather data can cause queuing at switches, potentially delaying much smaller yet important weather notification traffic such as wind shear alerts as they are queued behind large bulk weather data transmissions. Clearly, for such an example, a general static prioritisation choice could be made for perpetual prioritisation for

such traffic. However, with dynamic prioritisation and routing capabilities, engineers are enabled to take strategic remedial actions in the face of disruption. For example, during storms and severe weather disruption which causes the ATM service considerable challenges, network traffic prioritisation could be deployed on-demand to send higher priority traffic over cabled infrastructure as opposed to non-cabled links, such as satellite links, which would have a greater level of interference in bad weather.

Severe weather and network flooding are two examples where on-demand prioritisation, achievable through this work, can give engineers significant assistance in the face of maintaining service under adverse conditions.

5.4.2 Experiment Design

To highlight the impact of this on-demand remediation, a basic oversubscribed topology to represent flooding, was designed. Figure 5.15 shows the topology. The topology had hosts H_1, H_2, H_3 and H_4 each connected to switch S_1, with links of bandwidth 10 Mbps each. H_1 and H_2 are used for generating TCP traffic, sent to H_4, the destination. H_3 is used to send ICMP (ping) traffic to H_4, representing the smaller, more important traffic. The experiment will run in three parts: running with idle links between both H_1 and H_2 to S_1; running with H_1 to H_4 at capacity (10Mbps) and no traffic from H_2; and at oversubscription with H_1 and H_2 sending at 10Mbps to H_4, therefore oversubscribing the link from S_1 to H_4 by ca. 2:1, taking into account the ICMP traffic. A remote Controller, logically commands and controls the topology with the web-based UI for operator instruction.

The different protocols were selected as a clear means to distinguish characteristics in the traffic, used in the flow rules in S_1's flow tables. The flow rules installed are based on OpenFlow's queuing and priority specifications. When prioritisation is desired, operators use the UI to deploy the QoS on-demand to a specific switch. The first action the system takes is to $create\ a\ new\ qos$ entry in the OpenVSwitch. The $qos\ entry$ specifies the queues with indexes and labels. The queue index is used to assign the traffic to a queue in the OpenFlow tables action list. For each queue in queues, these queues must then be created

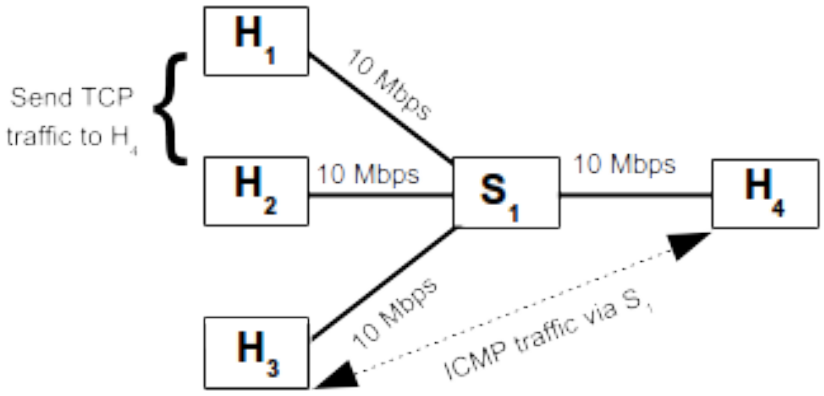


Figure 5.15: Basic oversubscribed topology to represent flooding

and added to the OpenVSwitch. For the purposes of this experiment, the only configuration specified for the queues is the prioritisation. However, maximum and minimum rates are also applicable to the wider context of this work, enabling remedial capabilities on-demand. With the $queues$ installed and the $qos\ prioritisation$ configured, the Controller's last task is to install the OpenFlow rules. The rules take the match criteria the operator specifies in the UI, in this case a match on TCP or ICMP traffic, respectively. The OpenFlow rule then assigns the ICMP traffic to have $actions = set\ queue : 0, output : 4$, where queue index 0 is the higher priority queue in the OpenVSwitch QoS entry and output 4 is the egress port to reach the packet destination (H_4).

5.4.3 Results & Discussion

The simulations were run in Mininet. As a result, the simulations have very low latency on the idle links between the virtualised hosts. The first phase of the experiment with and without $QoS\ prioritisation$ gives similar latency for idle traffic utilisation of the links from H_1 and H_2 to H_4, as seen in Figure 5.16. In the figure, two separate simulations are shown. The orange graph represents ICMP traffic with a lower priority. The black graph represents ICMP traffic with a higher priority. No TCP traffic was being sent in this experiment, making the prioritisation redundant. This resulted in the two simulations performing very similarly, with low latency, as expected.

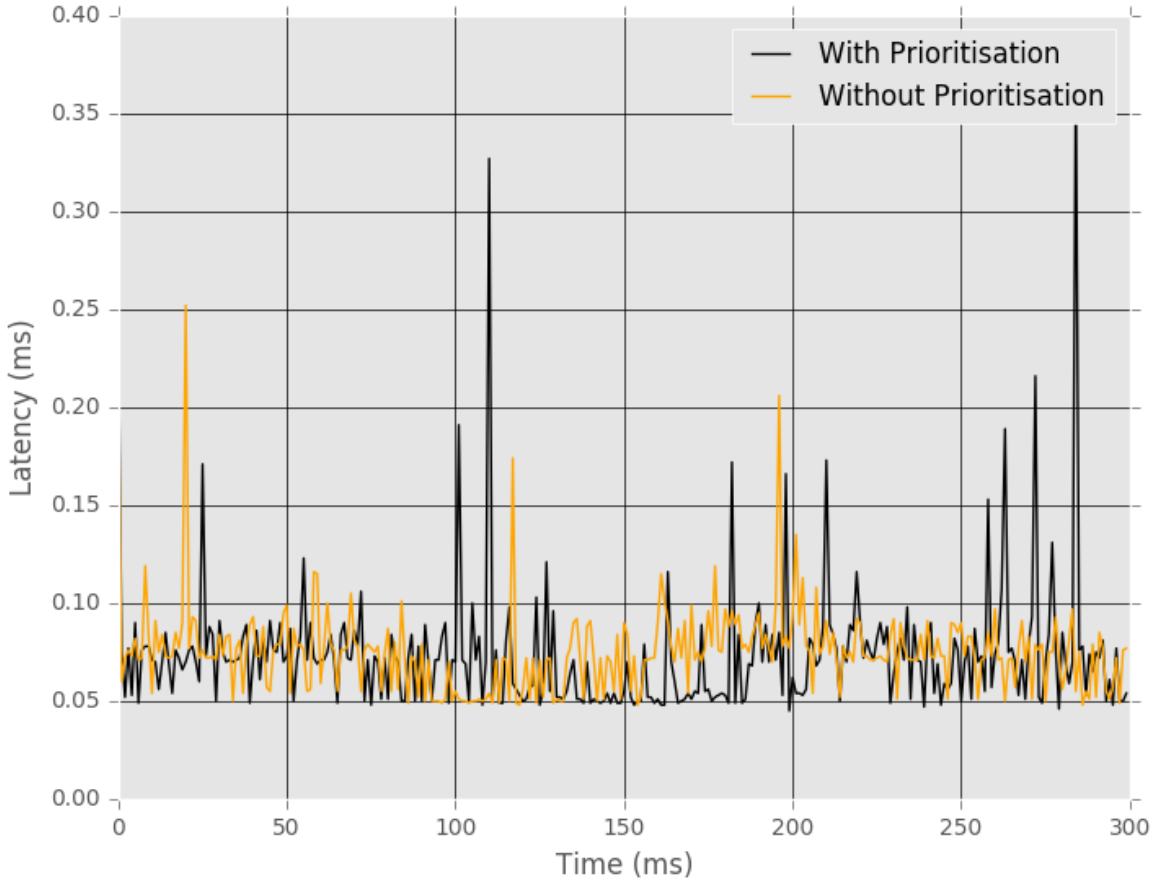


Figure 5.16: Latency for higher and lower priority simulations of traffic with idle links

Figure 5.17 shows the impact on the higher priority traffic as compared with the traffic without prioritisation when the TCP traffic (not shown) has increased to be matching the capacity of the link. Clearly, with priority, seen in black, ICMP traffic has similar behaviour as in Figure 5.16. However, the ICMP simulation without prioritisation experiences significant jitter and much higher relative latency.

Figure 5.18 has the same experimental set up as above, however, TCP traffic is now being sent from H_1 and H_2 to H_4, oversubscribing the link from S_1 to H_4. The simulation with prioritised traffic, in black, is now at such a low scale it is obscured on the graph, remaining at the same low levels, unaffected by the lower priority oversubscribed traffic. The simulation where the ICMP traffic in orange is at the same prioritisation level as the oversubscribed TCP traffic shows vastly greater latencies. Dropped packets, shown as red circles at 650ms for readability, also arise. The impact of the TCP traffic congestion controls can be observed in the ICMP latencies.

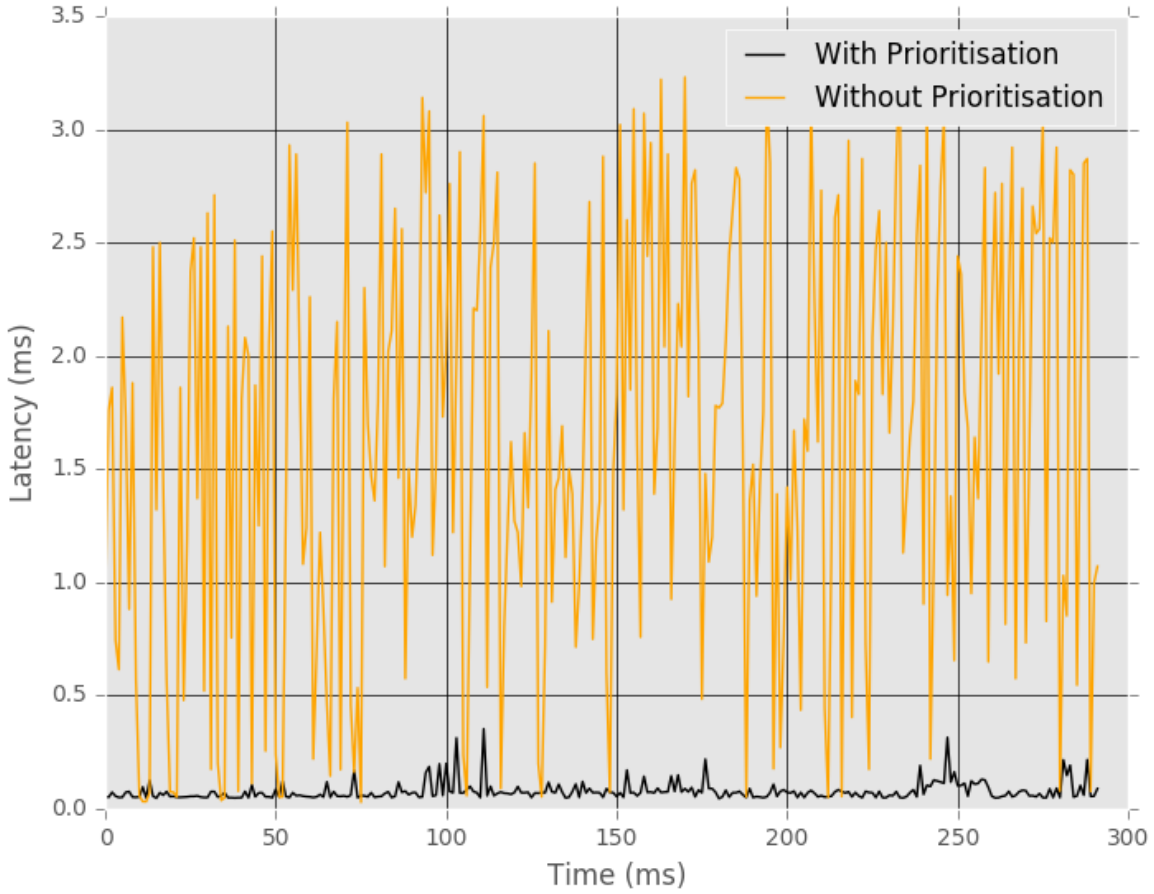


Figure 5.17: Traffic with and without prioritisation latencies for fully utilised link

The on-demand prioritisation clearly provides significant improvement to the prioritised traffic latencies as the overall levels of traffic in the network approach full utilisation. With lower levels of traffic, there is no adverse impact from using prioritisation. With this on-demand capability, engineers and operators are enabled to manage the critical data traffic services relevant to the adverse impacts faced at the given moment.

5.5 Summary

In this chapter a thorough evaluation of the capabilities of the system presented in this work has been provided. The evaluation gives strong justification that the system meets the aims of this work by enabling operators to have, among other beneficial aspects, a greater understanding of their networks through deployable, on-demand monitoring and detection NFs.

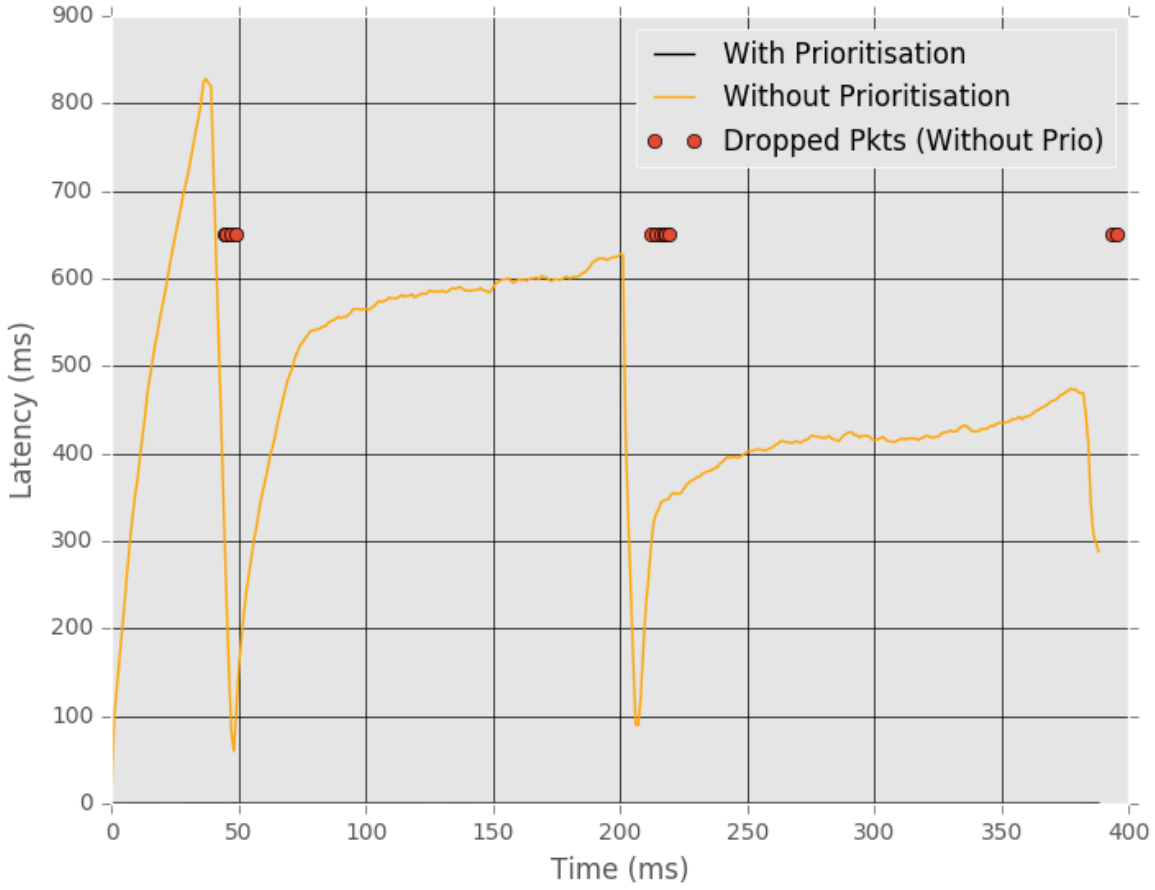


Figure 5.18: High latency ICMP traffic including dropped packets without prioritisation when using oversubscribed link

Throughout this chapter the shortfalls in current ATM infrastructure have been referenced, providing context for the contributions presented. In the first instance, a rapid flooding detection method was presented which uses the capabilities available through the technologies chosen within this system.

A suite of NFs tackling on-demand anomaly detection was then presented, each of which also contributes to an analysis of normal behaviour for a different service within the overall ATM operation.

Each of the NFs presented a core functional offering to help operators monitor their networks better and detect adverse events. These NFs implicitly improve monitoring, enable sufficient monitoring capabilities, and assist with the reduction of recovery times through better informing operators about their network. Finally, this work gives operators the ability to deploy further NFs to investigate their current network behaviour and increase their aware-

ness. Therefore, the system presented in this work meets the evaluation criteria, providing a significant series of key contributions in this field.

Chapter 6

Conclusions & Future Work

6.1 Overview

This chapter summarises and concludes this work, highlighting the contributions made, revisiting the original thesis statement, before discussing directions for further work. Overall, this work has successfully met the objectives set forth in the opening chapter with numerous contributions stemming from this research. The remainder of this chapter is structured as follows: Section 6.2 details the contributions made throughout this work, Section 6.3 revisits the original thesis statement and presents how the assertion has been met. The chapter concludes with a discussion on directions for future work in Section 6.4, from which more research can follow and concluding remarks in Section 6.5.

6.2 Contributions

This work has applied the latest networking softwarisation paradigms of SDN and NFV tailored to the needs of the ATM context in order to provide operators with richer insight into their networks. Increasing ATM service visibility through such insights is necessary for operators faced with increasing complexity, demand and a need to have full capacity operations with high availability and rapid recovery from any disruption. The generated function

deployment architecture design meets current and emerging requirements of global ATM systems, as identified in the literature. To meet future needs, being able to integrate UAV systems was crucial. Most prominently, this work was required to meet the need for sufficient monitoring capabilities which have been shown to be lacking in existing systems with respect to both increasing capacity and reducing disruption. The implemented system provides numerous contributions from which ANSPs can architect their future ATM operational networks:

1. ATM systematic strengths and weaknesses were assessed and reviewed from numerous sources, including a collection of major incident reports and recommendations from the past decade.

2. A function deployment architecture focussed on improved on-demand deployable monitoring capabilities for ATM operators was designed with the ability to deploy modular components for monitoring, including anomaly detection and remediation.

3. The arguments in favour of the technical implementation decisions to achieve the designed architecture, including Linux-based containers for these modular components, were provided. A modular, on-demand system with deployable Virtual Network Functions, tailored and configurable to operator information requirements at a given time, was developed.

4. In-depth analyses on real-world ATM datasets were performed, including the definitions of normal behaviour with domain expertise guidance, for UAV telemetry relationships, surface vehicle ground movements surveillance data and secondary aerial en-route surveillance data. For each model and definition of normal operating behaviour, network functions were developed. Justifications were provided for the different components of the architecture which were used for each of the functions.

5. The abilities for operators to deploy a range of functionality distributed throughout their networks on-demand were highlighted through example NFs. Moreover, nu-

merous NFs were developed highlighting the capabilities of the function deployment architecture with real-world examples.

6.3 Thesis Statement Revisited

In this section, the thesis statement is repeated from Section 1.2, and the remainder of this section indicates how it has been addressed. The thesis statement is restated as follows:

> The primary ambition of this work is to determine whether, through the softwarisation of future ATM data networks, operators are granted a richer insight and therefore gain a better understanding of their infrastructure and services. Implicitly, from this improved understanding, the intention is to make ATM systems more resilient, through more informed decision-making by operators, especially in the face of challenges or disruption to services. This hypothesis will be tested by the development and evaluation of a virtual network function deployment architecture, applying NFV and SDN technologies. The feasibility of softwarisation for ATM data networks will need to be shown. Example cases from which a greater understanding can be achieved through such softwarisation will also be examined.

A new ATM data function deployment architecture has been designed and developed, which exploits the latest best practices of softwarisation through SDN and NFV in order to facilitate greater monitoring capabilities on-demand for ATM operators. The overarching hypothesis has been proven, with clear evidence that the softwarisation of ATM data networks is meaningful and through the developed function deployment architecture, infrastructure and service visibility has been increased for ATM operators. The implementation in this work allows for generalised monitoring, anomaly detection and remediation functions to be deployed on-demand, facilitating significant visibility for operators through a suite of tools. Based on an analysis of previous incidents, this work asserts proof of this hypothesis through a number of example network functions. Network flooding detection functions, implemented in this work, show alert notifications thrown after 10 seconds of abnormal activity. This com-

pares with significantly slower polling techniques widely used today, with alerts triggering after 5 minutes. Flooding incidents have caused hours of disruption and reduced capacity for ATM in previous incidents. Therefore, more rapid detection can help to better inform operators and implicitly reduce recovery times, in particular with respect to incident diagnosis. Similarly, visibility is increased for UAV operations with anomaly detection network functions developed in this work, providing immediate notifications for deviations outwith prediction intervals of previously acceptable operating parameters in the telemetry relationships between the engine throttle and RPM of the UAV. Official reports state that had such knowledge and leading indicators been available to operators and alerted them, a previous incident with a 40% deviation from normal operating conditions could have been avoided an hour prior to the incident [146]. Remediation capabilities have also been developed for the function deployment architecture as part of this work, allowing operators to enable on-demand traffic prioritisation. Such prioritisation gives operators the means to recover from challenges such as flooding or service issues, and ensure systematic resilience with other service traffic keeping bandwidth. An anomaly detection VNF for secondary surveillance observes deviations in the relationship between the number of physical aeroplanes and the number of bytes transferred in the ATM data network. Typical values observed in this work show 1000 aircraft equates to ~50KB. This network function would alert operators if 1000 aircraft were identified, but the number of bytes fell outwith the range of 25KB to ~83KB. This deployable function can be used as a means to detect unexpected traffic, and due to the tightly coupled cyber-physical relationship, provides an effective way to monitor for abnormal behaviour. Finally, ground based surface movements were analysed and a network function utilising EWMA was developed allowing for anomaly detection based on a deviation from recent past acceleration, heading and velocity. This capability helps operators observe rapid changes such as sharp braking during which can be indicative of a near-miss. Small collisions and near-misses during taxiing and on runways are a major disruption for ATM as visual inspections of aeroplanes and the surface must be carried out. By identifying such anomalies operators can gain a greater insight into the trends around near-misses, as well as more accurately determining safe operating limits, when faced with ever-increasing

capacity demand.

Each of the specific examples outlined above shows a proven increased visibility to operators, achievable through this work. Each example can be generalised, and moreover, these examples provide a foundation for operators to develop further enhanced network functions based on more of their service data.

In general, the state-of-the-art softwarised function deployment architecture presented meets some of the key current and emerging future needs for ATM services, as defined by the international forecasts and trends for greater capacity, interconnectivity and, crucially, the independent recommendations for the imminent need for greater monitoring capabilities, faster recovery times and greater adaptability for responsiveness under adverse conditions. The requirements collated lead to a recognition for elements of ATM infrastructure needing greater flexibility, open innovation and responsiveness. Increasing automation is advantageous but still a need remains for control to reside with human operators to ensure the overall safety of air travel, which is paramount, over and above increased efficiency and capacity. Through examples, it was shown that deployable virtual network functions assist operators' understanding of their ATM networks, particularly with respect to notification of abnormal behaviour across numerous aspects of ATM services. Moreover, it was explained how each architectural component aligns with ATM data characteristics, in particular SDN and NFV. Crucial characteristics of ATM networks exploited for this design were the relatively static nature (with respect to the Internet, for example) of the topology, the largely interpretable traffic matrices, the single authoritative domain owner, the ANSP, and the highly distributed nature of multiple services. Analysing real-world data sets taken from live ATM systems and services, provided confidence in this work through a thorough evaluation, where analyses were performed to define normal behaviour. These models were then used in combination with the capabilities exploited in the new architecture such as, network functions and controller based notifications, to allow operators to better understand their networks and be informed based on their on-demand needs. The definitions of normal operating behaviours for a variety of services and relationships within the ATM service are clear contributions which can, in

their own right, be taken forward in future work, alongwith the application of the novel networked system comprising virtualisation and on-demand functionality, presented throughout this work.

6.4 Future Work

The work in this dissertation has demonstrated how ATM networked systems can be improved to facilitate better understanding for operators through enabling on-demand monitoring and remediation capabilities which provide greater insight and resilience as required, leading to more informed decision-making to mitigate disruptive incidents. The work presented in the preceding chapters provides a number of opportunities for future research, which are outlined below.

6.4.1 ATM Service Deployment

The ATM services of en-route surveillance, surface movements surveillance, UAV telemetry processing and core backbone network monitoring and remediation provisions have been focussed upon throughout this work. Similar relationships and definitions of normal behaviour will be available in all aspects of ATM network traffic, including VoIP, flight data planning and meteorological data. In total, over eighty services contribute to the UK ATM infrastructure. Analysis for each of these, and any interactions, as part of further research would result in a comprehensive understanding of the ATM data networks. Based on further understanding of these services, this architecture can be utilised, through various modular network functions, to provide in-depth monitoring and remediation capabilities tuned to the individual service characteristics present in the overall infrastructure.

6.4.2 Modelling of longer periods

The models of normal behaviour for ATM services presented in this work are based on generous, yet limited data. With longer periods of data, longer term patterns can become apparent, e.g., en-route surveillance is likely to exhibit yearly periodicity with inherent seasonal differences which could be incorporated into the models and definitions of normal behaviour to increase their precision. From the further ATM services mentioned previously, interdependencies may exist similar to those highlighted in this work between the number of aeroplanes observed by surveillance and the bytes transferred on the wire, for example, or the relationship of throttle and RPM for UAVs. Exploiting more of these highly coupled variable sets will lead to more precise models which can more easily detect deviances from the norm.

6.4.3 Large-Scale Deployment

To fully experience the impact of this work, a full-scale deployment using large-scale ATM testbeds will be a vital next stage. If an ATM operator who manages a large-scale testbed wished to deploy this system, they could do so. Such a deployment would also provide scope for further enhancements and future work as the characteristics of the specific testbed could be utilised for increased performance.

6.5 Concluding Remarks

This work set out to prove the hypothesis that, through softwarisation of future ATM data networks, operators can gain richer insight into their infrastructure and services. By designing, developing, and evaluating a functional deployment architecture using SDN and container-based NFV, this work has shown through sample NFs that service visibility is increased. This work demonstrates advanced monitoring functionality which is configurable and operates as part of the wider ATM data network.

Bibliography

[1] Lucy Budd. “Air craft: producing UK airspace”. In: Routledge (Taylor & Francis Group), 2008, pp. 115–134.

[2] Cheng-Lung Wu and Robert E. Caves. “Research review of air traffic management”. In: *Transport Reviews* 22 (2002), pp. 115–132.

[3] Francisco Javier Saez Nieto. “The long journey toward a higher level of automation in ATM as safety critical, sociotechnical and multi-Agent system”. In: *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering* 230.9 (2016), pp. 1533–1547.

[4] Fabrizio d’Amore Tommaso De Zan and Federica Di Camillo. “The Defence of Civilian Air Traffic Systems from Cyber Threats”. In: *Istituto Affari Internazionali* (2016).

[5] United States General Accounting Office GAO. *AIR TRAFFIC CONTROL: FAA Needs a More Comprehensive Approach to Address Cybersecurity As Agency Transitions to NextGen*. Tech. rep. Apr. 2015.

[6] National Park Service United States Department of the Interior. *NATIONAL HISTORIC LANDMARK NOMINATION: 1956 GRAND CANYON TWA-UNITED AIRLINES AVIATION ACCIDENT SITE*. 2011. URL: `http://www.nps.gov/nhl/news/LC/spring2011/GrandCanyonREDACTED.pdf`.

[7] Stanley J Levy. “The Expanding Responsibility of the Government Air Traffic Controller”. In: *Fordham Law Review* 36.3 (1968), p. 401.

[8] United States Department of Transportation: Federal Aviation Administration. *FAA HISTORICAL CHRONOLOGY, 1926-1996*. 2008. URL: https://www.faa.gov/about/media/b-chron.pdf.

[9] United Kingdom: House of Commons Library: Tom Rutherford. *Air transport statistics*. 2011. URL: http://researchbriefings.files.parliament.uk/documents/SN03760/SN03760.pdf.

[10] Thomas A Heppenheimer. *Turbulent Skies: The History of Commercial Aviation*. J. Wiley & Sons, 1995.

[11] Air Traffic Action Group (ATAG). *European Air Traffic Forecast 1985–2015*. International Air Transport Association (IATA), 1999.

[12] Federal Aviation Administration. *FAA's NextGen Implementation Plan 2011*. Tech. rep. Mar. 2011.

[13] James H Williams and TL Signore. "National Airspace System Security Cyber Architecture". In: *Journal of Air Traffic Control* 53.1 (2011), p. 38.

[14] Ira Lewis. "Analysis of alternative institutional arrangements for reform of US air traffic control". In: *International Public Management Journal* 7.3 (2004), p. 385.

[15] Federal Aviation Administration US Department of Transportation. *Pilot's Handbook of Aeronautical Knowledge*. 2008.

[16] United States General Accounting Office GAO. *INFORMATION SECURITY: FAA Needs to Address Weaknesses in Air Traffic Control Systems*. Tech. rep. Jan. 2015.

[17] OJ Newell, MM Wolfson, and ER Ducot. *NextGen Weather Processor Architecture Study*. Tech. rep. DTIC Document, 2010.

[18] FEDERAL AVIATION ADMINISTRATION. *FAA TELECOMMUNICATIONS INFRASTRUCTURE OPERATIONAL NETWORK IP USERS' GUIDE*. 2010.

[19] Kyle JS White, Dimitrios P Pezaros, and Christopher W Johnson. “Increasing resilience of ATM networks using traffic monitoring and automated anomaly analysis”. In: *Proceedings of the 2nd International Conference on Application and Theory of Automation in Command and Control Systems*. IRIT Press. 2012, pp. 82–92.

[20] TELECOMMUNICATION STANDARDIZATION SECTOR OF INTERNATIONAL TELECOMMUNICATION UNION. *SERIES G: TRANSMISSION SYSTEMS AND MEDIA, DIGITAL SYSTEMS AND NETWORKS. Digital networks – Quality and availability targets*. 2001.

[21] Harris Corporation Mark Graham. *Impact to NextGen of the Telecommunications Industry Evolution from Time Division Multiplexing (TDM) Technology to Internet Protocol (IP) Technology*. Tech. rep. 2014.

[22] International Air Transport Association (IATA). *Press Release: Airlines to Welcome 3.6 Billion Passengers in 2016*. 2012.

[23] International Civil Aviation Organization. *ICAO Environment Report 2010: Aviation Outlook Overview*. 2010.

[24] International Civil Aviation Organization. *ICAO Environment Report 2013: Aviation and Climate Change*. 2013.

[25] Eurocontrol: EUROPEAN AIR TRAFFIC MANAGEMENT PROGRAMME. *Airport CDM Cost Benefit Analysis*. 2008.

[26] Ewen Denney et al. “Assuring ground-based detect and avoid for UAS operations”. In: *2014 IEEE/AIAA 33rd Digital Avionics Systems Conference (DASC)*. IEEE. 2014, 6A1–1.

[27] Ram Ramanathan and Jason Redi. “A brief overview of ad hoc networks: challenges and directions”. In: *IEEE communications Magazine* 40.5 (2002), pp. 20–22.

[28] Xavier Masip-Bruin et al. “Research challenges in QoS routing”. In: *Computer communications* 29.5 (2006), pp. 563–581.

[29] Martin Hruby, Michal Olsovsky, and Margareta Kotocova. “IAENG Transactions on Engineering Technologies: Special Volume of the World Congress on Engineering 2012”. In: 2013. Chap. Solving VoIP QoS and Scalability Issues in Backbone Networks.

[30] B. Amarasekara, A. Nirmalathas, and R. J. Evans. “Analysis of ip-based communication backbone over shared wide area-network for Smart Grid applications”. In: *Wireless Personal Multimedia Communications (WPMC), 2014 International Symposium on*. Sept. 2014, pp. 601–606.

[31] S. Tomovic, M. Radonjic, and I. Radusinovic. “Bandwidth-delay constrained routing algorithms for backbone SDN networks”. In: *Telecommunication in Modern Satellite, Cable and Broadcasting Services (TELSIKS), 2015 12th International Conference on*. Oct. 2015, pp. 227–230.

[32] Federal Aviation Administration. *TDM-to-IP Migration*. Tech. rep. Oct. 2015.

[33] United States General Accounting Office GAO. *High-Risk Series: Information Management and Technology*. Tech. rep. Feb. 1997.

[34] Russell Clarke, David Dorwin, and Rob Nash. “Is open source software more secure?” In: *Homeland Security/Cyber Security* (2009).

[35] United States General Accounting Office GAO. *Report to the Committee on Governmental Affairs, U.S. Senate: AIR TRAFFIC CONTROL Weak Computer Security Practices Jeopardize Flight Safety*. Tech. rep. May 1998.

[36] Jeffery Case et al. *A simple network management protocol (SNMP)*. 1989.

[37] Nick McKeown et al. “OpenFlow: enabling innovation in campus networks”. In: *ACM SIGCOMM Computer Communication Review* 38.2 (2008), pp. 69–74.

[38] Martin Casado et al. “Fabric: a retrospective on evolving SDN”. In: *Proceedings of the first workshop on Hot topics in software defined networks*. ACM. 2012, pp. 85–90.

[39] E. T. S. Institute. *2012 Network Functions Virtualisation, White Paper*. URL: http://portal.etsi.org/NFV/NFV%20White%20Paper.pdf.

[40] Gabriel Cuba et al. "PUCPLight: A SDN/OpenFlow controller for an academic campus network". In: *ANDESCON, 2016 IEEE*. IEEE. 2016, pp. 1–4.

[41] C. Chappel. *Unlocking Network value: service Innovation in the Era of SDN, White paper.* 2013. URL: http://www.cisco.com/web/solutions/trends/open_network_environment/docs/hr_service_innovation.pdf.

[42] O. Atia. "Future trends for IP services over FAA telecommunications infrastructure". In: *2008 Integrated Communications, Navigation and Surveillance Conference*. May 2008, pp. 1–8.

[43] *Private Communications*. 2016.

[44] Open Networking Foundation White Paper. *Software-Defined Networking: The New Norm for Networks*. Tech. rep. Apr. 2012.

[45] EUROCONTROL Specification. *Surveillance Data Exchange - Part 1 All Purpose Structured EUROCONTROL Surveillance Information Exchange (ASTERIX)*. Tech. rep. Apr. 2012.

[46] Paul Smith et al. "Network resilience: a systematic approach". In: *Communications Magazine, IEEE* 49.7 (2011), pp. 88–97.

[47] Peter M Chen and Brian D Noble. "When virtual is better than real [operating system relocation to virtual machines]". In: *Hot Topics in Operating Systems, 2001. Proceedings of the Eighth Workshop on*. IEEE. 2001, pp. 133–138.

[48] Open Networking Foundation. *SDN Architecture*. Tech. rep. Feb. 2016.

[49] Miyoung Kang et al. "Formal modeling and verification of SDN-OpenFlow". In: *Software Testing, Verification and Validation (ICST), 2013 IEEE Sixth International Conference on*. IEEE. 2013, pp. 481–482.

[50] S. Hares and R. White. "Software-Defined Networks and the Interface to the Routing System (I2RS)". In: *IEEE Internet Computing* 17.4 (July 2013), pp. 84–88.

[51] M Smith et al. *OpFlex control protocol, Internet-Draft*. Tech. rep. IETF, Apr, 2014.

[52] Evangelos Haleplidis. *Forwarding and Control Element Separation (ForCES) Model Extension*. RFC 7408. Oct. 2015. DOI: 10.17487/rfc7408. URL: https://rfc-editor.org/rfc/rfc7408.txt.

[53] A Atlas et al. *An architecture for the interface to the routing system. Work in progress Draft IETF I2RS Architecture*. Sept. 2015.

[54] Adrian Farrel, Jean-Philippe Vasseur, and Jerry Ash. *A path computation element (PCE)-based architecture*. Tech. rep. RFC 4655, August, 2006.

[55] Peter Saint-Andre. *Extensible Messaging and Presence Protocol (XMPP): Address Format*. Tech. rep. RFC 7622, September, 2015.

[56] Tina Tsou et al. *Use Cases for ALTO with Software Defined Networks*. Tech. rep. 2012.

[57] Matthew Monaco, Oliver Michel, and Eric Keller. "Applying Operating System Principles to SDN Controller Design". In: *Proceedings of the Twelfth ACM Workshop on Hot Topics in Networks*. HotNets-XII. ACM, 2013, pp. 21–27.

[58] Natasha Gude et al. "NOX: towards an operating system for networks". In: *ACM SIGCOMM Computer Communication Review* 38.3 (2008), pp. 105–110.

[59] J Mccauley. *Pox: A python-based openflow controller*. 2014.

[60] Jan Medved et al. "Opendaylight: Towards a model-driven sdn controller architecture". In: *2014 IEEE 15th International Symposium on*. IEEE. 2014, pp. 1–6.

[61] David Erickson. "The Beacon Openflow Controller". In: *Proceedings of the Second ACM SIGCOMM Workshop on Hot Topics in Software Defined Networking*. HotSDN '13. ACM, 2013, pp. 13–18.

[62] Andreas Voellmy and Paul Hudak. "Nettle: Taking the Sting Out of Programming Network Routers". In: *PADL*. 2011, pp. 235–249.

[63] Teemu Koponen et al. "Onix: A Distributed Control Platform for Large-scale Production Networks." In: *OSDI*. Vol. 10. 2010, pp. 1–6.

[64] Justine Sherry et al. “Making middleboxes someone else’s problem: network processing as a cloud service”. In: *ACM SIGCOMM Computer Communication Review* 42.4 (2012), pp. 13–24.

[65] ABI Research. *World enterprise network and data security markets*. URL: https://www.abiresearch.com/market-research/product/1006059-world-enterprise-network-and-data-security/.

[66] Jorge Carapinha et al. “Network virtualization-opportunities and challenges for operators”. In: *Future Internet-FIS 2010*. Springer, 2010, pp. 138–147.

[67] Daniel King and Chris Ford. “A critical survey of Network Functions Virtualization (NFV)”. In: *iPOP: IP Over Optical*. 2013, pp. 1–21.

[68] Nicolai Leymann and Deutsche Telekom AG. “Flexible Service Chaining. Requirements and Architectures.” In: *EWSDN. Presentation* (2013).

[69] Justine Sherry and Sylvia Ratnasamy. “A survey of enterprise middlebox deployments”. In: *University of California at Berkeley. Technical Report No. UCB/EECS-2012–24* (2012).

[70] Joao Martins et al. “ClickOS and the art of network function virtualization”. In: *Proceedings of the 11th USENIX Conference on Networked Systems Design and Implementation*. USENIX Association. 2014, pp. 459–473.

[71] Vyas Sekar et al. “Design and implementation of a consolidated middlebox architecture”. In: *Presented as part of the 9th USENIX Symposium on Networked Systems Design and Implementation (NSDI 12)*. 2012, pp. 323–336.

[72] James W Anderson et al. “xOMB: extensible open middleboxes with commodity servers”. In: *Proceedings of the eighth ACM/IEEE symposium on Architectures for networking and communications systems*. ACM. 2012, pp. 49–60.

[73] Richard Cziva et al. “SDN-based virtual machine management for cloud data centers”. In: *Cloud Networking (CloudNet), 2014 IEEE 3rd International Conference on*. IEEE. 2014, pp. 388–394.

[74] Aaron Gember-Jacobson et al. “OpenNF: Enabling innovation in network function control”. In: *ACM SIGCOMM Computer Communication Review* 44.4 (2015), pp. 163–174.

[75] Shriram Rajagopalan et al. “Split/merge: System support for elastic execution in virtual middleboxes”. In: *Presented as part of the 10th USENIX Symposium on Networked Systems Design and Implementation (NSDI 13)*. 2013, pp. 227–240.

[76] Aaron Gember-Jacobson and Aditya Akella. “Improving the Safety, Scalability, and Efficiency of Network Function State Transfers”. In: *Proceedings of the 2015 ACM SIGCOMM Workshop on Hot Topics in Middleboxes and Network Function Virtualization*. ACM. 2015, pp. 43–48.

[77] Joao Soares et al. “Cloud4nfv: A platform for virtual network functions”. In: *Cloud Networking (CloudNet), 2014 IEEE 3rd International Conference on*. IEEE. 2014, pp. 288–293.

[78] Diego Kreutz et al. “Software-defined networking: A comprehensive survey”. In: *Proceedings of the IEEE* 103.1 (2015), pp. 14–76.

[79] Stephen Soltesz et al. “Container-based operating system virtualization: a scalable, high-performance alternative to hypervisors”. In: *ACM SIGOPS Operating Systems Review*. Vol. 41. 3. ACM. 2007, pp. 275–287.

[80] Warren Matthews and Les Cottrell. “The PingER project: active Internet performance monitoring for the HENP community”. In: *Communications Magazine, IEEE* 38.5 (2000), pp. 130–136.

[81] Sunil Kalidindi and Matthew J Zekauskas. “Surveyor: An infrastructure for internet performance measurements”. In: *INET’99*. Citeseer. 1999.

[82] Fotis Georgatos et al. “Providing active measurements as a regular service for ISPs”. In: *PAM*. 2001.

[83] Vijay Kumar Adhikari et al. “Vivisecting youtube: An active measurement study”. In: *INFOCOM, 2012 Proceedings IEEE*. IEEE. 2012, pp. 2521–2525.

[84] Joel Apisdorf et al. “OC3MON: Flexible, Affordable, High Performance Staistics Collection.” In: *LISA*. Vol. 96. 1996, pp. 97–112.

[85] Chuck Fraleigh et al. “Design and deployment of a passive monitoring infrastructure”. In: *Evolutionary Trends of the Internet*. Springer, 2001, pp. 556–575.

[86] Anja Feldmann et al. “Deriving traffic demands for operational IP networks: Methodology and experience”. In: *IEEE/ACM Transactions on Networking (ToN)* 9.3 (2001), pp. 265–280.

[87] S Waldbusser. *Remote Network Monitoring Management Information Base”, RFC 1757*. 1995.

[88] Rob Enns, Martin Bjorklund, and Juergen Schoenwaelder. *NETCONF configuration protocol*. 2011.

[89] Sajad Shirali-Shahreza and Yashar Ganjali. “Flexam: Flexible sampling extension for monitoring and security applications in openflow”. In: *Proceedings of the second ACM SIGCOMM workshop on Hot topics in software defined networking*. ACM. 2013, pp. 167–168.

[90] Amin Tootoonchian, Monia Ghobadi, and Yashar Ganjali. “OpenTM: traffic matrix estimator for OpenFlow networks”. In: *Passive and active measurement*. Springer. 2010, pp. 201–210.

[91] Niels LM Van Adrichem, Christian Doerr, and Fernando A Kuipers. “Opennetmon: Network monitoring in openflow software-defined networks”. In: *Network Operations and Management Symposium (NOMS), 2014 IEEE*. IEEE. 2014, pp. 1–8.

[92] Junho Suh et al. “OpenSample: A low-latency, sampling-based measurement platform for commodity SDN”. In: *Distributed Computing Systems (ICDCS), 2014 IEEE 34th International Conference on*. IEEE. 2014, pp. 228–237.

[93] Curtis Yu et al. “Flowsense: Monitoring network utilization with zero measurement cost”. In: *Passive and Active Measurement*. Springer. 2013, pp. 31–41.

[94] Christos Argyropoulos et al. “PaFloMon–A Slice Aware Passive Flow Monitoring Framework for OpenFlow Enabled Experimental Facilities”. In: *Software Defined Networking (EWSDN), 2012 European Workshop on*. IEEE. 2012, pp. 97–102.

[95] Rob Sherwood et al. “Flowvisor: A network virtualization layer”. In: *OpenFlow Switch Consortium, Tech. Rep* (2009), pp. 1–13.

[96] Hyojoon Kim and Nick Feamster. “Improving network management with software defined networking”. In: *Communications Magazine, IEEE* 51.2 (2013), pp. 114–119.

[97] Andreas Voellmy, Hyojoon Kim, and Nick Feamster. “Procera: A Language for High-level Reactive Network Control”. In: *Proceedings of the First Workshop on Hot Topics in Software Defined Networks*. HotSDN ’12. ACM, 2012, pp. 43–48.

[98] Ye Yu, Chen Qian, and Xin Li. “Distributed and collaborative traffic monitoring in software defined networks”. In: *Proceedings of the third workshop on Hot topics in software defined networking*. ACM. 2014, pp. 85–90.

[99] Minlan Yu, Lavanya Jose, and Rui Miao. “Software Defined Traffic Measurement with OpenSketch”. In: *Presented as part of the 10th USENIX Symposium on Networked Systems Design and Implementation (NSDI 13)*. 2013, pp. 29–42.

[100] Shubhajit Roy Chowdhury et al. “Payless: A low cost network monitoring framework for software defined networks”. In: *Network Operations and Management Symposium (NOMS), 2014 IEEE*. IEEE. 2014, pp. 1–9.

[101] Taesang Choi et al. “Virtualized traffic monitoring function and resource auto-scaling in software-defined networks”. In: *Network Operations and Management Symposium (APNOMS), 2015 17th Asia-Pacific*. IEEE. 2015, pp. 546–549.

[102] Taesang Choi et al. “SuVMF: software-defined unified virtual monitoring function for SDN-based large-scale networks”. In: *Proceedings of The Ninth International Conference on Future Internet Technologies*. ACM. 2014, p. 4.

[103] Ashwin Joshi Darshan Maiya Gaurav Chheda and Vamsikrishna Nethi. *Expedition: An Open Source Network Monitoring Tool for Software Defined Networks*. 2015.

[104] Yao-Yu Yang et al. "The Implementation of Real-Time Network Traffic Monitoring Service with Network Functions Virtualization". In: *2015 International Conference on Cloud Computing and Big Data (CCBD)*. IEEE. 2015, pp. 279–286.

[105] Rodrigo Braga, Edjard Mota, and Alexandre Passito. "Lightweight DDoS flooding attack detection using NOX/OpenFlow". In: *Local Computer Networks (LCN), 2010 IEEE 35th Conference on*. IEEE. 2010, pp. 408–415.

[106] Kostas Giotis, Georgios Androulidakis, and Vasilis Maglaris. "Leveraging SDN for efficient anomaly detection and mitigation on legacy networks". In: *Software Defined Networks (EWSDN), 2014 Third European Workshop on*. IEEE. 2014, pp. 85–90.

[107] Syed Akbar Mehdi, Junaid Khalid, and Syed Ali Khayam. "Revisiting traffic anomaly detection using software defined networking". In: *Recent Advances in Intrusion Detection*. Springer. 2011, pp. 161–180.

[108] Fernando Silveira et al. "ASTUTE: Detecting a different class of traffic anomalies". In: *ACM SIGCOMM Computer Communication Review* 40.4 (2010), pp. 267–278.

[109] Augustin Soule, Kavé Salamatian, and Nina Taft. "Combining filtering and statistical methods for anomaly detection". In: *Proceedings of the 5th ACM SIGCOMM conference on Internet Measurement*. USENIX Association. 2005, pp. 31–31.

[110] James PG Sterbenz and David Hutchison. *Resilinets: Multilevel resilient and survivable networking initiative wiki*. 2008.

[111] Kyle JS White, Dimitrios P Pezaros, and Chris W Johnson. "Using programmable data networks to detect critical infrastructure challenges". In: *International Conference on Critical Information Infrastructures Security*. Springer. 2014, pp. 209–221.

[112] Kyle J. S. White, Dimitrios P. Pezaros, and Christopher W. Johnson. "Principles for increased resilience in critical networked infrastructures". In: *6th International Conference on Research in Air Transportation (ICRAT 2014)*. May 2014.

[113] Johannes Muller et al. “What does safety mean for networks?” In: *Systems Safety 2009. Incorporating the SaRS Annual Conference, 4th IET International Conference on*. IET. 2009, pp. 1–6.

[114] Michael Fry et al. “Challenge identification for network resilience”. In: *Next Generation Internet (NGI), 2010 6th EURO-NF Conference on*. IEEE. 2010, pp. 1–8.

[115] Piotr Cholda et al. “A survey of resilience differentiation frameworks in communication networks.” In: *IEEE Communications Surveys and Tutorials* 9.1-4 (2007), pp. 32–55.

[116] James PG Sterbenz et al. “Resilience and survivability in communication networks: Strategies, principles, and survey of disciplines”. In: *Computer Networks* 54.8 (2010), pp. 1245–1265.

[117] Lisanne Bainbridge. “Ironies of automation”. In: *Automatica* 19.6 (1983), pp. 775–779.

[118] Federal Aviation Administration. *NextGen Implementation Plan*. Tech. rep. 2016.

[119] European Union and Eurocontrol. *SESAR JU, European ATM Master Plan - The Roadmap for Delivering High Performing Aviation for Europe*. Tech. rep. 2015.

[120] FAA FTI Review Panel. *Report on November 19, 2009 Outage*. 2010.

[121] Los Angeles Times. *LAX outage is blamed on 1 computer*. 2007.

[122] Office of Inspector General Department of Homeland Security. *Lessons Learned from the August 11, 2007, Network Outage at Los Angeles International Airport (Redacted)*. 2007.

[123] Chief Customs Los Angeles Times Quote by Jennifer Connors and Border Protection Officer. *Customs acts fast to shore up systems*. 2007.

[124] Press of Atlantic City. *Fire at Hughes Technical Center caused $2.2M in damage*. 2007.

[125] Irish Aviation Authority. *Report of the Irish Aviation Authority into the ATM System Malfunction at Dublin Airport*. 2008.

[126] Robert Walmsley. *Independent Enquiry, NATS System Failure 12 December 2014 – Final Report*. 2015.

[127] The Dominion Post. *Directing traffic in the dark: when flight control crashes*. 2015.

[128] James Reason. *Human Error*. Cambridge University Press, 1990.

[129] K. J. S. White et al. "A Programmable SDN+NFV-based Architecture for UAV Telemetry Monitoring". In: *14th IEEE Consumer Communications and Networking Conference (CCNC)*. 2017.

[130] Hüseyin Okcu. "Operational Requirements of Unmanned Aircraft Systems Data Link and Communication Systems". In: *Journal of Advances in Computer Networks* 4.1 (2016).

[131] H. Skinnemoen. "UAV amp; satellite communications live mission-critical visual data". In: *2014 IEEE International Conference on Aerospace Electronics and Remote Sensing Technology*. Nov. 2014, pp. 12–19.

[132] Spenser D Lee. *Routing UAVs to Co-Optimize Mission Effectiveness and Network Performance with Dynamic Programming*. Tech. rep. DTIC Document, 2011.

[133] Bob Lantz, Brandon Heller, and Nick McKeown. "A network in a laptop: rapid prototyping for software-defined networks". In: *Proceedings of the 9th ACM SIGCOMM Workshop on Hot Topics in Networks*. ACM. 2010, p. 19.

[134] Dilip Joseph, Arsalan Tavakoli, and Ion Stoica. "A Policy-aware Switching Layer for Data Centers". In: *SIGCOMM*. ACM, Aug. 2008, pp. 51–62.

[135] Fung Po Tso and Dimitrios P. Pezaros. "Baatdaat: Measurement-based flow scheduling for cloud data centers". In: *2013 IEEE Symposium on Computers and Communications, ISCC 2013, Split, Croatia, 7-10 July, 2013*. 2013, pp. 765–770.

[136] Justine Sherry and Sylvia Ratnasamy. *A Survey of Enterprise Middlebox Deployments*. Tech. rep. EECS Department, University of California, Berkeley, Feb. 2012.

[137] Brent Stephens et al. "PAST: Scalable Ethernet for Data Centers". In: CoNEXT '12. ACM, 2012, pp. 49–60.

[138] Alex X. Liu, Chad R. Meiners, and Eric Torng. “TCAM Razor: A Systematic Approach Towards Minimizing Packet Classifiers in TCAMs”. In: *IEEE/ACM Trans. Netw.* (2010), pp. 490–500.

[139] John D Day and Hubert Zimmermann. “The OSI reference model”. In: *Proceedings of the IEEE* 71.12 (1983), pp. 1334–1340.

[140] Dirk Merkel. “Docker: lightweight linux containers for consistent development and deployment”. In: *Linux Journal* 2014.239 (2014), p. 2.

[141] Gartner. *Security Properties of Containers Managed by Docker*. Jan. 2015. URL: `https://www.gartner.com/doc/2956826/`.

[142] Paul Harvey and Joseph Sventek. “Wireless sensor network simulation with Xen”. In: *Proceedings of the 46th Annual Simulation Symposium*. Society for Computer Simulation International. 2013, p. 4.

[143] Romain Fontugne and Kensuke Fukuda. “A Hough-transform-based anomaly detector with an adaptive time interval”. In: *ACM SIGAPP Applied Computing Review* 11.3 (2011), pp. 41–51.

[144] Ajay Tirumala et al. *Iperf: The TCP/UDP bandwidth measurement tool*. 2005.

[145] Nong Ye, Connie Borror, and Yebin Zhang. “EWMA techniques for computer intrusion detection through anomalous changes in event intensity”. In: *Quality and Reliability Engineering International* 18.6 (2002), pp. 443–451.

[146] NASA. *SIERRA UAS Mishap Classification: Type C, IRIS Case Number: S-2013-208-00001*. Oct. 2013.

List of Publications

The work reported in this dissertation has led to the following publications:

- *"A Programmable SDN+NFV-based Architecture for UAV Telemetry Monitoring."*
  K. J. S. White, E. Denney, M. D. Knudson, A. K. Marnerides, D. P. Pezaros
  14th IEEE Consumer Communications and Networking Conference (CCNC), 2017.

- *"Container-based Network Function Virtualization for Software-Defined Networks."*
  R. Cziva, S. Jouet, K. J. S. White, D. P. Pezaros
  20th IEEE Symposium on Computers and Communication (ISCC), 2015.

- *"Using Programmable Data Networks to Detect Critical Infrastructure Challenges."*
  K. J. S. White, D. P. Pezaros, C. W. Johnson
  9th International Conference on Critical Information Infrastructures Security (CRITIS), 2014.

- *"Principles for Increased Resilience in Critical Networked Infrastructures."*
  K. J. S. White, D. P. Pezaros, C. W. Johnson
  6th International Conference for Air Transport Research (ICRAT), 2014.

- *"Increasing Resilience of ATM Networks using Traffic Monitoring and Automated Anomaly Analysis."*
  K. J. S. White, D. P. Pezaros, C. W. Johnson
  2nd International Conference on Application and Theory of Automation in Command and Control Systems (ATACCS), 2012.