Universitat Autònoma de Barcelona

Departament d'Enginyeria de la Informació i de les Comunicacions

Architectures for Aeronautical Opportunistic Networking

Submitted to Universitat Autònoma de Barcelona
in partial fulfillment of the requirements for the
degree of Doctor of Philosophy in Computer Science

by Rubén Martínez-Vidal
Bellaterra, September 2015

Advisor: Dr. Joan Borrell
Professor Universitat Autònoma de Barcelona

I certify that I have read this thesis entitled "Architectures for Aeronautical Opportunistic Networking" and that in my opinion it is fully adequate, in scope and in quality, as a dissertation for the degree of Doctor of Philosophy.

Bellaterra, September 2015

Dr. Joan Borrell
(Advisor)

Committee:
Juan-Carlos Cano Escribá (Univ. Politècnica de València)
Jordi Herrera Joancomartí (Univ. Autònoma de Barcelona)
Yves Mahéo (Univ. de Bretagne-Sud)
Substitutes:
Pietro Manzoni (Univ. Politècnica de València)
Guillermo Navarro Arribas (Univ. Autònoma de Barcelona)
Frédéric Guidec (Univ. de Bretagne-Sud)

Program: *Doctorat en Informàtica.*
Department: Departament d'Enginyeria de la Informació i de les Comunicacions.

To my loved ones.

Abstract

IN recent years, the aeronautical communications field has experienced increased demand for data exchanges between terrestrial, aerial and satellite platforms. Conventional aeronautical communications have proven to fall short on handling this growing demand. As a result, aeronautical systems have moved towards the use of satellite-based communications, and while effective, their deployment and later use poses an economical challenge. To deal with this problem, academic work on the topic has proposed the use of networking alternatives in the form of Aeronautical Ad-hoc Networks. These networks use large radio communication ranges and work on the premises of Mobile Ad-Hoc Networks (MANET), requiring end-to-end connectivity to function properly. Aeronautical networks have an extremely varying topology coupled with a frequently unreliable communication channel. With this in mind, we think that assuming full network connectivity is a problematic choice. Therefore, in this thesis we will approach those problems using a different method. Specifically, we will strive to establish a network of aircraft assuming no guarantees for connectivity, limited radio ranges, and relying solely on the sporadic encounters between nodes to perform data exchanges. This opportunistic approach offers a cheaper solution than those based on long-range radio links or satellite communications. The foundation of this proposal lies in opportunistic networks, and eventually merges with the current communication trend based on satellite links, to provide a new network infrastructure that greatly minimizes communication costs and equipment expenditure. The main contributions of this thesis are as follows. On one hand, the use of an accurate mobility model describing aeronautical patterns by considering on flight route and scheduling information of real flights. Additionally, this work provided accurate modeling of the network capabilities of each node including the complete representation of all layers of the network stack. These realistic features are of severe importance to ensure successful deployment in real networks. On the other hand, this thesis provides Quality of Service assurances that are hard to achieve in opportunistic networking. To the best of our knowledge, this thesis represents the first in-depth analysis of a realistic large-scale aeronautical opportunistic network.

Resum

En els últims anys, el camp de les comunicacions aeronàutiques ha experimentat un increment en la demanda d'intercanvis de dades entre plataformes terrestres, aèries i satelitals. Els mètodes de comunicació aeronàutics convencionals han demostrat no ser suficients per manejar aquesta demanda creixent. Com a resultat, els sistemes aeronàutics han tendit cap a l'ús de comunicacions via satèl·lit, que encara que efectives, suposen un repte econòmic. En resposta a aquest problema, el treball acadèmic en aquesta àrea ha proposat l'ús d'alternatives basades en xarxes aeronàutiques ad-hoc. Aquestes xarxes usen abastos de comunicació immensos i treballen sota la premissa de les Xarxes Ad-Hoc Mòbils (MANETs) requerint l'existència de connectivitat extrem a extrem per al seu correcte funcionament. Les xarxes aeronàutiques tenen una topologia extremadament variable amb enllaços de comunicació poc fiables. Amb aquestes propietats en ment, creiem que assumir connectivitat de xarxa és una elecció desafortunada. Per tant, en aquesta tesi intentem solucionar aquests reptes amb un enfocament diferent. Específicament, intentarem establir una xarxa d'avions sense assumir garanties de connectivitat, enllaços de radi curts i utilitzant tan sol les trobades esporàdiques entre nodes per realitzar intercanvi de dades. Aquest enfocament oportunista ofereix una alternativa més barata que aquelles basades en enllaços de radi de llarg abast o en comunicacions via satèl·lit. Aquesta proposta està basada en les xarxes oportunistes, i eventualment s'uneix amb la tendència actual d'utilitzar enllaços via satèl·lit, per oferir una nova arquitectura de xarxa que redueix àmpliament costos de comunicació i equipament. Les principals contribucions d'aquesta tesi són les següents. D'una banda, l'ús d'un model de mobilitat precís descrivint patrons aeronàutics basats en informació i horaris de vol reals. A més, aquest treball utilitza un model minuciós per descriure les capacitats de xarxa dels nodes incloent la representació completa de totes les capes de xarxa. Aquestes característiques realistes són importants per assegurar el correcte desplegament en xarxes reals. D'altra banda, aquesta tesi ofereix una sèrie de garanties en qualitat de servei que són difícils d'obtenir en xarxes oportunistes. Fins a on sabem aquesta tesi representa el primer estudi detallat d'una xarxa aeronàutica a gran escala utilitzant comunicacions oportunistes.

Acknowledgements

THE journey leading to the culmination of this thesis has been a long and arduous one. I would like to spend some lines thanking all those I have met along the way and whose help and support have contributed to making it possible.

First of all, I would like to thank my family, and particularly my parents, for their trust and support during all my life and education.

I would like to express my thanks to all the people from the Mobile and Internet Systems Laboratory for their warm welcome during my stay in Cork. I'm especially thankful to Prof. Cormac Sreenan, my host supervisor, for all his help and guidance during those months.

Multiple thanks to all the people I have encountered around the world during multiple conferences and symposiums. A special mention to Juan-Carlos Cano for the great conversations and enlightening ideas shared during our meals in Sydney at the LCN 2013. Another mention, to the Darmstadt PhD students with whom I "explored" the Edmonton pedestrian subway at LCN 2014.

I would also like to thank Thomas Henderson for his careful mentoring during my Google Summer of Code at the ns-3 project, and for his later cooperation in our WNS3 paper. I have to say that I learned a lot from that experience, and I'm really grateful. An extra thanks to the whole ns-3 developer community for their continuous effort in improving and expanding such a magnificent network simulator.

Additional thanks go to the colleagues from my home department dEIC for the working and non-working moments together. I specially loved being part of the annual "Sortides de Departament" in those hiking trips around Catalonia. I would like to thank Sergi Robles for giving me the chance to join the department as a "Becari de Col·laboració" and later as a support technician working on the PROSES project. I would like to thank Ramon Martí for his corrections and

remarks. I would also like to thank Adrian Antunez for his work during his final undergraduate project that contributed to our social graph analysis. Another word of gratitude goes to my office mates (and friends) for being there at the good and bad times.

And without a doubt, I would like to thank my thesis supervisor, Joan Borrell, for his help, advice, dedication and cheering during all these years. Thank you very much!

And lastly, a very special word of thanks goes to that one person that has been by my side with words of encouragement for the past few months of dissertation writing.

Works contained in this thesis have been partially supported by the following institutions and organizations: Universitat Autònoma de Barcelona with Grant UAB PIF 472-01-1/E2010. Catalan AGAUR under projects 2009SGR-1224, and 2014SGR-691. Spanish Ministry of Science and Innovation under projects AVANZA TSI-020100-2009-115, TIN2010-15764, and TIN2014-55243-P. Google Inc. under the 2014 Google Summer of Code program.

An additional recognition to the IEEE LCN for their funding under LCN-2013 and LCN-2014 Student Participation Grants.

Rubén Martínez Vidal
Bellaterra, September 2015

Contents

List of Figures

List of Tables

List of Algorithms

Part I
Preliminaries

The passenger jet was, and still is, one of the wonders of the world, a world whose other wonders the jet made accessible. Along with the personal computer, it ranks as the greatest technological innovation of the second half of the twentieth century. The computer turned your lowly desk into a cross between Harvard and Hollywood. The jet turned you into an adventurer. It freed you from the shackles of that desk and set you free to roam the world, to become a Lindbergh, an Earhart, a James Bond, and, if you had enough money and time, a jet-setter.

Willian Stadiem

Jet Set: The People, The Planes, The Glamour, and The Romance in Aviation's Glory Years. 2014.

1 Introduction

IT can be said that the airplane and the computer may be considered the greatest technological innovations of the 20th century. Conversely, the 21st century has marked the dawn of the Digital Era. Computers proliferate, devices are growing smaller and smaller, inherently mobile, and capable of interconnection. These devices form networks, creating intermittently connected scenarios that become more and more common as mobile devices grow more ubiquitous in our society. Nowadays, wireless networks have become part of our everyday lives, up to the point where it has become difficult to conceive situations where our devices are isolated. Flying around the world while sitting in a passenger cabin was one of these rare occasions. But apparently, the connectivity wave has finally reached the aviation domain as more and more commercial flights are starting to offer In-Flight Internet services.

Current airline deployments provide passenger connectivity using solutions either based on ground infrastructure [1] or through the use of expensive dedicated satellite links [2, 3]. In regards to communications between aircraft or communications with traffic control, the current communication capabilities are usually more limited. Aircraft surveillance is performed using radar-based

surveillance or using a system known as ADS-C (Automatic Dependent Surveillance - Contract). This system periodically sends aircraft identification, position, altitude, speed, and other relevant tracking data to a ground station via a CPDLC datalink (Controller Pilot Data Link Communications). This datalink has a very limited bandwidth of barely a few tens of bytes per second. Any other control information is sent over traditional voice-based radio links. The limitations in data communication become more evident when looking for the causes of recent aerial accidents. To respond to this problem the European and North American ATM (Air Traffic Management) agencies have invested on the improvement and unification of the ATM control systems by providing new levels of interconnectivity between aircraft. The US Next Generation Air Transportation System (NextGen) and the Single European Sky ATM Research (SESAR) are the key initiatives on this line. One of the key aspects of these initiatives is the substitution of conventional data links with satellite-based ones. The most relevant example is the mandatory implantation of ADS-B (Automatic Dependent Surveillance - Broadcast) in all aircraft by the end of 2020. This system broadcasts surveillance information using a satellite-based communication systems.

The deployment of a computer network between several aircraft can contribute to a significant improvement in aeronautical communications, surveillance, and passenger security. All of these topics are critical concerns for general aviation. When considering connectivity between aircraft the first concept that comes to mind is that of a Vehicular Ad-hoc Network, or VANET. There has been some previous academic work on this line; these proposals rely on the use of radio communication systems with large coverages (from 200 to 800 km). As a result, airplanes form a network with guaranteed end-to-end connectivity. This sort of aerial networks have been designated as Aeronautical Ad-hoc Networks (AANETs) [4, 5].

AANETs are characterized by a highly varying topology of high-velocity nodes coupled with a frequently unreliable wireless communication channel. Taking these facts into account, we think that assuming full connectivity for the network is a problematic choice. Depending on the considered scenario, for example in continental areas with a high density of flights, it may be achievable. For less connected scenarios, such as trans-oceanic flights, it becomes challenging. We believe that for these sparse scenarios, assuming no end-to-end connectivity simplifies the problem, requires smaller radio ranges, and removes the need to keep up with the constant topological changes. Therefore, we consider

that opportunistic networking, also called Delay-Tolerant Networking (DTN), is more suited for this scenario.

The starting point and assumptions of this thesis originated from the results of the Spanish Science Foundation PROSES (Protocols for the Single European Sky) project [6, 7, 8]. This project offered a proof of concept, including physical deployment and field experimentation, of a small-scale aeronautical Delay-Tolerant Network, and ultimately assessed the feasibility of performing opportunistic communications on aerial vehicles. Final results and field deployment details can be seen on Appendix A

In this thesis, we are going one step ahead and expand the scale of the aeronautical network to that of a real scenario. We will tackle the problems posed by scarcely populated air scenarios. Specifically, we will strive to establish a network of aircraft in the North Atlantic ocean. We will assume no guarantees for connectivity, limited radio ranges, and rely solely on the sporadic encounters between nodes to perform data exchanges. This opportunistic approach offers a cheaper solution than those based on long-range radio links or satellite communications.

One of the main focuses of our study will be to provide realistic results. Most studies on opportunistic networks usually use theoretical or synthetic models, both for the scenario and its mobility patterns. However, theoretical models derived from synthetic mobility rarely hold when applied to real world networks. Despite that, evaluation of real large-scale opportunistic networks is usually uncommon, a handful of examples can be found in [9, 10]. In our case, we disregard synthetic models and create our own scenario based on flight route and scheduling information of real transatlantic flights.

1.1 Objective

The objective of this thesis is to provide a reliable and efficient communication alternative to conventional aeronautical communications on trans-oceanic scenarios. We achieve this purpose by providing a cost-effective network architecture whose foundation lies in opportunistic communications and eventually merges with the current communication trend based on satellite links. This unification provides a new network architecture that greatly minimizes communication costs and equipment expenditure. This alternative network architecture is achieved after careful analysis of all the aspects of the network scenario,

including mobility, topology and data propagation patterns.

1.2 Contributions

The contributions of this thesis are the following:

- The design of trans-oceanic aeronautical scenarios and mobility models based on realistic data.
- A characterization and parametrization of the mobility and topological characteristics of the studied scenario.
- An analysis of information propagation and a performance evaluation of the network scenario using opportunistic based communications.
- A design and evaluation of architectural alternatives based on opportunistic communications working together with satellite links.
- A measurement of the quality of service of the proposed network architecture and the enumeration of service guarantees achievable.
- A proof of concept and field testing of a small scale aeronautical opportunistic network.
- A design and implementation of reliable DTN poin-to-point transmission protocols.

1.3 List of publications

Positive results of this research have been published in national and international conferences and journals:

- R. Martínez-Vidal, M. de Toro, R. Martí, J. Borrell. *A Delay and Disruption Tolerant Key Distribution Scheme for Aeronautical Environments* (In Spanish). In Proceedings of XII Reunión Española de Criptología y Seguridad de la Información (RECSI 2012) Mondragón Unibertsitatea, (September 2012), pp. 181-186. [11]

- R. Martínez-Vidal, S. Castillo, S. Robles, A. Sánchez, J. Borrell, M. Cordero, A. Viguria, N.Giuditta. *Mobile-Agent Based Delay-Tolerant Network Architecture for Non-Critical Aeronautical Data Communications*. In Proceedings of the 10th International Symposium on Distributed Computing and Artificial Intelligence. Springer, vol. 217, (May 2013), pp. 513-520. [8].

- R. Martínez-Vidal, R. Martí, J. Borrell. *Characterization of a transoceanic aircraft delay tolerant network*. In Proceedings of the 38th Annual IEEE Conference on Local Computer Networks. IEEE Computer Society, (October 2013), pp. 590-597. [12]

- R. Martínez-Vidal, R. Martí, J. Borrell.*Analyzing Information Propagation in a Transoceanic Aircraft Delay Tolerant Network*. In Proceedings of the 39th IEEE Conference on Local Computer Networks. IEEE Computer Society Press, (October 2014), pp. 116-123. [13]

- R. Martínez-Vidal, R.Martí, C. J. Sreenan, J. Borrell. *Methodological evaluation of architectural alternatives for an aeronautical delay tolerant network*. International Journal on Pervasive and Mobile Computing, (July 2015), DOI: 10.1016/j.pmcj.2015.06.012. [14]

- R. Martínez-Vidal, T. R. Henderson, J. Borrell. *Implementation and Evaluation of Licklider Transmission Protocol (LTP) in ns-3*. In Proceedings of the 2015 Workshop on ns-3. ACM Press, (May 2015), pp. 75-82. [15]

1.4 Document Layout

This thesis is divided into five parts. The first part, *Preliminaries*, introduces the key aspects of the work performed in this thesis, and describes the several concepts required to understand the rest of the document. This part is divided in two chapters: Chapter 1, this **Introduction**, and following, Chapter 2: **State of the Art**, an overview of the several technologies relevant to aeronautical networks and their state prior to the publication of this thesis.

The second part, *Network Characterization*, models the network scenario and analyzes all of its relevant characteristics. This part is divided into four chapters. Chapter 3, **Scenario Description and Mobility Model**, describes the scenario and the design of its mobility model. Chapter 4, **Network Model**,

describes the network stack, including all the applications and protocols that may be used in different sections of this thesis. Chapter 5, **Network Mobility Characterization**, analyzes the way mobility influences the network and quantifies it using several metrics. Lastly, Chapter 6, **Network Topology Characterization**, analyses the topological structure of the network scenario.

In part three, *Network Evaluation*, is where we evaluate our proposals. This part is divided into three chapters. In Chapter 7, **Opportunistic Network Performance Evaluation**, we analyze the feasibility and performance of the scenario using opportunistic communications. Chapter 8, **Evaluation of Hybrid Networking Alternatives**, proposes and evaluates a novel hierarchical architecture that mixes opportunistic networking with satellite communications. Chapter 9, **Quality of Service**, properly quantifies and validates the degree of service the can be offered by our proposed architecture.

In part four, *Conclusions*, exposes the conclusions obtained from this thesis and describes possible future lines of work.

Lastly, in part five, *Appendices*, we offer relevant work related to opportunistic and aeronautical networks, but not directly linked to the evaluation of our transoceanic network architecture. Appendix A, describes field deployment and experimentation of our initial small scale aeronautical DTN. Appendix B, describes design and implementation of simulation models for reliable DTN transmission protocols.

"In the present state of the art this is all that can be done"

Henry Harrison Suplee

Gas Turbine: progress in the design and construction of turbines operated by gases of combustion. 1910.
First recorded use of the term "state of the art"

2 State of the Art

THIS chapter introduces the reader to the technologies, architectures, and protocols commonly used in Aeronautical Ad-hoc Networks, or AANETs. We also describe the key technologies employed in this thesis, namely Opportunistic Networks, and review their application in the aeronautical field.

This chapter is divided into four main sections. The first section introduces the concept of AANETs, describes their characteristics, and their most challenging features. It concludes by reviewing the standard routing approaches that proposals found in the literature used to cope with those problems. The second section introduces the concept of opportunistic networking highlighting its mains differences with conventional mobile ad-hoc networks. This section continues by introducing the most common modeling approaches and routing protocols used in this kind of network. The third section defines our approach to aeronautical systems (opportunistic communications) and reviews related work on this topic. The last section provides a summary of the most relevant aspects and highlights the distinct contribution that this thesis offers to the field.

2.1 Aeronautical Ad-Hoc NETworks (AANETs)

An ad-hoc network is a multihop wireless network composed of a set of nodes connected via point-to-point wireless links. This network is created spontaneously and does not require existing infrastructure. The term Mobile Ad hoc NETwork (MANET), is commonly applied to a wireless ad-hoc network whose nodes are mobile. Another interesting case is the term VANET (Vehicular Ad hoc Network) which can be considered as a particular type of MANET where the links are established between nearby vehicles. The node mobility in a MANET changes the network topology by continually creating and destroying the paths linking the nodes of the network. Despite that, end-to-end connectivity between any two nodes of the network needs to be always guaranteed. In recent years, there has been an increased interest in this kind of network along with several examples of small to medium network deployments.

Aeronautical Ad-hoc Networks (AANETs) imports the concepts of land-based VANETs to aeronautical environments. The notion of ad hoc networking between aircraft was first introduced in [4]. This work addressed critical issues regarding connectivity and how aeronautical mobility patterns affect the paths connecting nodes in the network and offered several measures in path distance and duration. This work concluded with the proposal of a specialized MANET routing protocol for aeronautical environments. This work focused on an intra-continental flight scenario and used radio technologies with a communications range between 200 and 600 km. A follow-up work of the authors [5], further expanded the network boundaries to a worldwide context, creating the concept of densely connected continental clusters of nodes, and including ground and satellite infrastructure into the network architecture. Later proposals like [16], delve into the requirements of AANET network architectures, and proposes and evaluates concrete network protocols and applications based on a TCP/IP stack.

2.1.1 Communication Links

The communication links in AANETs are based on Radio Frequency (RF) technologies and are assumed to be capable of only Line of Sight (LOS) communications. Common proposed technologies include VHF Data Link (VDL) [17], UHF, or microwave [18]. Antennas are considered as omni-directional, radio ranges between 200 to 800 km [4, 19], and frequencies in the reserved aeronautical band of 1GHz [20]. Due to the high speed of the nodes, many of the models

often consider Doppler shift[1] effects [17, 4]. More recent proposals like [21] have offered an alternative to RF communications, and proposed the usage of Free Space Optics (FSO). This data link can provide higher data-rates (on the order of several Gbps) with ranges up to 200 km.

2.1.2 Connectivity

The connectivity issues and the highly dynamic topology is a common topic of concern which is addressed by several works such as [22],[17],[23],[19], and [18]. First, [22] offers a connectivity analysis using a probabilistic theoretical model based on the density of nodes in flight-paths. Contrarily, [23],[17],[19],[18] offer connectivity analysis based on simulations and parameterized by density of aircraft and communication range.

The first work of this second type [23], measures connectivity based on two performance criteria: a ratio based on coverage area, and the number of aircraft each aircraft can connect to (node degree). Several flight traffic patterns are evaluated and their connectivity is compared using simulations.

In [17], the authors provide a theoretical network model composed of a set of aircraft flying at a constant speed and spaced according to a Poisson distribution while operating over several lanes representing flight-levels. Connectivity is evaluated using a mathematical model that describes aircraft isolation probability based on radio range and scenario characteristics.

In [19], the authors analyze connectivity and link stability over a realistic aeronautical scenario. The purpose of this paper was to determine the possibility of deploying an aircraft MANET over the North-Atlantic corridor. The radio model used in this work has a range of 225 to 450nmi (∼370-833 km). The conclusions were positive achieving a stable network with link durations greater than half an hour. The authors present related work in [20] where a similar analysis is offered focusing on European continental flights.

Similarly [18], analyses the feasibility of deploying a MANET in oceanic areas. This document estimates connectivity based on measures of the number of connected aircraft and the duration of those connections.

[1] Doppler Shift is the difference between observed and emitted frequency of a wave for an observer moving relative to the source of the waves. For nodes moving at high speeds (such as aircraft) the Doppler shift of radio communications becomes fairly apparent and can be easily measured.

2.1.3 AANET Routing

The principal concern of MANET is to keep up with the dynamically changing network topologies. MANET routing protocols work by announcing its node presence and by listening for announcements made by its neighbors. Commonly used MANET routing protocols can be divided into two broad categories:

- **Proactive Routing Protocols:** this kind of routing protocol maintains up-to-date routing tables by periodically exchanging information with the nodes in the network. Optimized Link State Routing Protocol (OLSR) [24] and Destination Sequence Distance Vector (DSDV) [25] are the best-known examples.

- **Reactive Routing Protocols:** this type of routing protocol only finds a specific routes when nodes require them. Ad hoc On-demand Distance Vector(AODV) [26] and Dynamic Source Routing (DSR) [27] are the most widely known examples.

AANETs have the distinguishing feature of being composed by nodes moving at high speeds resulting in highly dynamic network topologies. This trait has a significant impact on the performance of routing protocols, and as a consequence, providing reliable communications in such environments is a great challenge. Most proposals agree with the necessity of a multi-hop based approach for the successful delivery of data within the network. It is also commonly remarked the inability of traditional MANET routing protocols to properly deliver data when used in AANET environments. There have been a few studies on the field of routing in aeronautical networks [28], [29], [30], [31], [32] each one presenting their efficient way to select optimal paths.

An example of such protocols is Ad-hoc Routing Protocol for Aeronautical Mobile Ad hoc Networks (ARPAM)[28]. This protocol is a customized version of AODV for use in AANETs. The main additions of ARPAM consist in the exploitation of geolocalization information obtained from aeronautical applications, the use of proactive functions and improved route maintenance.

Another of such proposals is the geographical-based routing protocol AeroRP [29]. This routing protocol takes decisions based in a TTI (Time To Intercept) metric that gives an estimation of how soon a neighbor node will be within transmission range of the destination. Some studies regarding this routing protocol ([33, 34, 35]) offer a performance evaluation under different traffic conditions.

These works compare AeroRP against other reactive (AODV and DSR) and proactive (OLSR and DSDV) routing protocols commonly used in MANETs. Finally, they prove that their protocol outperforms them.

Another example of such a protocol is DASR (Delay Aware Routing Protocol) [30], a reactive routing protocol that keeps a periodically updated list of neighbors. This protocol uses a decision function to select the best hop from its neighbor list. This function takes decisions based on a node mobility and expected delay metric. The authors use metrics related to traffic demand and network stability to compare their proposal against other AANET routing protocols.

The routing protocol MUDOR (Multipath Doppler Routing Protocol) [31] is another fine example of specialized AANET routing. MUDOR is a reactive routing protocol that uses a decision function based on the relative speed of nodes. The relative speed is obtained by accounting for the Doppler shift of reply packets (hence the protocol name). This protocol aims to select stable paths within the network and to guarantee link durations long enough to forward data successfully.

Lastly, the Mobility Aware Routing/Mobility Dissemination Protocol, or (MARP/MDP) protocol suite [32]. This protocol exploits unique features of AANETS, namely predefined flight plans, to find optimal routes, and to reduce latency and overhead. MARP utilizes the knowledge of expected node trajectories to predict the network topology beforehand. However, in such networks many reasons force the nodes to change their predefined schedules (weather, departure delays, etc.). In such cases, MDP is used to inform all nodes in the network of the possible discrepancies between the original scheduled topology and runtime changes.

As a final note, it is worth to remark that MUDOR [31] holds particular relevance because its performance assessment used a realistic scenario and mobility model. The evaluation scenario used in this work is based on continental flights, with realistic proportions, density of nodes and node trajectories. Other proposals validate their results using small-scale synthetic scenarios based on both random node distributions and random mobility patterns (Gauss-Markov mobility models). However, the general problem is that theoretical models inferred from synthetic mobility rarely hold when implemented on concrete real world networks.

2.1.4 Disruption Tolerance

An important aspect of AANETs is that they are highly mobile MANETs, with frequent link failures produced by reduced contact times and highly changing topologies. However, this is an often neglected topic, and fault tolerance is not typically considered or supplied. Nevertheless, there are a few studies that are aware of this problem [36],[37] and offer custom mechanisms to guarantee reliable services over this kind of networks. Concretely, these works propose and evaluate the use of a new domain-specific transport protocol called AeroTP. AeroTP performs reliable end-to-end data transfers between airborne nodes with support for TCP/UDP splicing in gateways exiting the aeronautical network.

2.2 Opportunistic Networks

Opportunistic networks are a variety of MANETS characterized by low node densities and short communication ranges, resulting in network topologies that are disconnected most of the time. Naturally, the nodes in an opportunistic network spend most of its time in isolation. Therefore, nodes in these networks are usually equipped with buffers capable of storing data for long periods. Sporadically another node may come into range due to node mobility. Those occasions are called contacts (or *communication opportunities*). During a contact, nodes have a chance to forward some of their data to other nodes. This communication paradigm is called *store (carry) and forward*, and it allows the eventual delivery of messages when one of the nodes carrying a copy of the message comes across the destination. As opposed to MANETs where there was always a direct path linking any two nodes in the network, in opportunistic networks there is no guarantee for end-to-end connectivity. It is straightforward to see that this delivery process can take a long time. Therefore, opportunistic networking is inherently tolerant to long transmission delays, and it is also commonly called Delay (and Disruption) Tolerant Networks (DTN).

The notion of DTNs [38, 39] originated from importing the concepts developed for the Interplanetary Internet to more mundane land-based networks. The DTN paradigm has been since used to cope with the problems intrinsic to opportunistic and challenged networks. The DTN architecture [40] defines an intermediate layer between the application and transport layer, which overcomes the limitations of TCP/IP, and provides the foundation for the use of

the *store and forward* paradigm. Two protocols compose this layer: the Bundle Protocol [41] and the Licklider Transmission Protocol (LTP) [42]. Bundle focuses on providing end-to-end delay tolerance. LTP is a specialized point-to-point transmission protocol that provides reliability over links characterized by long round-trip times (RTTs) and regular disruptions.

2.2.1 Mobility Characterization

Node mobility is an essential aspect of opportunistic networks because mobility is what generates communication opportunities. As opposed to other networks, mobility is the means through which communications are possible, and not a characteristic that hinders them. Therefore, careful analysis and modeling of mobility patterns is a crucial aspect in opportunistic network research. The focus of these studies is to analyze the mobility patterns of the network to unveil relationships between the underlying mobility model and the derived network characteristics.

The scenarios found in opportunistic networks are quite diverse. Opportunistic networks are defined by a set topological characteristics or specific challenging conditions. Therefore, the boundaries between opportunistic networking and other wireless networks (such as Vehicular Networks (VNs), Wireless Sensor Networks (WSNs), Underwater Networks, etc.) may be a little blurry. As a result, the types of encountered nodes and mobility patterns vary widely (*e.g.*, humans, animals, vehicles, etc.)

In this section, we review the previous work in network mobility characterization of opportunistic networks. Works on this line are divided into two categories: the ones based on synthetic or theoretical mobility models (like Random Waypoint (RWP)[43] or Random Walk (RW) [44]) and those created using realistic mobility traces.

Synthetic Mobility

An example of synthetic mobility based studies for opportunistic networks can be seen in [45]. In this document, the authors analyze the mobility characteristics of the RWP and RW mobility models to offer numerical estimates of duration of contact opportunities and elapsed time between contacts (inter-contact time). Later, in [46] the same authors extend their analytical results through simulation studies.

In [47] we can see another example. In this work, the authors examine the expected times between contacts for the RWP and RD mobility models and infer mathematical expressions for all the scenario parameters involved.

Further analysis on this topic can be found in [48]. In this work, the authors assume inter-contact-times spaced according to an exponential distribution and prove the validity of their assumption through experimental results.

Synthetic mobility models have the advantage of being general solutions, capable of modeling several aspects of real mobility with simplicity. However, a significant drawback is that synthetic models usually fail to catch particular mobility aspects in many scenarios. As a result, network characterizations based on synthetic mobility models do not usually hold when applied to real world networks.

Realistic Mobility

The usual approach used to tackle the problems posed by synthetic models is to use mobility traces extracted from real world scenarios. These traces allow to define mobility models that represent the behavior of the nodes accurately. The most common scenarios found in the literature are based on human or vehicular mobility.

We can find an example of human mobility in [49] which is based in traces of wireless network activity through an university campus. Other examples are [50] and [51], which are based in attendee mobility in conference environments. Both of these contributions proved that inter-contact time distributions shown by theoretical models were not accurate.

Another sort of mobility patterns are those found in vehicular networks. In these scenarios, the mobility models are reasonably different than those found in human mobility. Higher node speeds contribute to intermittent connectivity, and mobility constraints (roads, streets) along with points of interest (stations, airports, etc.) influence the mobility patterns. Most studies use small scale models of such networks, and in fact, few works actually study large-scale vehicular networks using real world data.

One example of real large-scale networks is found in [9]. In this work, the authors base their study in the public transportation network of Seattle and use traces of roughly 1,200 buses. This study offers accurate values on spatiotemporal characteristics of the network, contact, inter-contact times and node clustering coefficients. Another example is found in [10]. This study is based in

mobility traces of roughly 12,000 taxis in Beijing gathered during a period of one week. In this work, the authors analyze the distributions of active vehicles, node degree, and inter-contact times.

We can find more examples that use realistic mobility traces in [52], [53], and [54]. Those works focus on routing performance analysis of DTNs using realistic characterizations of scenario mobility. The works from [52],[53] base their model on real mobility traces from buses inside the cities of San Francisco and Amherst (Massachusetts) respectively. In the case of [54], the authors use a model based on urban vehicular mobility in the city of Milan.

2.2.2 Opportunistic Routing

Due to the characteristics of opportunistic networks and their *store, carry, and forward* paradigm, the nature of their routing protocols differs significantly from those used in other types of networks. One choice that routing protocols in opportunistic networks always need to make is whether they replicate data in a contact opportunity or not. Therefore, opportunistic routing protocols may be divided into two large groups based on the number of copies they generate(this taxonomy was introduced in [55]):

- **Forwarding based** (or single-copy): Only a single copy of the message can be present in the network at any instant of time. These protocols minimize network load but in exchange of reduced delivery ratios and higher delays.
- **Replication based** (or multi-copy): Allow the existence of multiple copies of the same message through the network. These protocols improve delivery ratios and reduce delivery times but in exchange they create message overhead.

Forwarding based Protocols

In these schemes, the source node initially holds a message, in every contact opportunity a decision must be taken, either forward the packet or keep it. These decisions are usually handled by metrics that measure the likelihood to reach the destination. These metrics use a wide variety of means to perform this task. Two common examples of forwarding based routing protocols are *Seek and Focus* [56] and *SimBet*[57].

Seek and Focus [56] alternates between different phases: During the *Seek phase* performs randomized forwarding. Data is forwarded to any encountered node with probability $p \in (0, 1)$. During the *Focus phase*, the protocol performs data forwarding according to a utility function U. This utility function is computed using elapsed times between encountered nodes. Conversely, *Simbet* [57] exploits the social component of encountered nodes, and uses graph centrality metrics to predict the likelihood of encounters.

Replication based Protocols

In these schemes, multiple copies of a packet can be created to increase the chances of successful data delivery at the cost of higher network load. Therefore, the primary concern for this kind of protocols is to what extent is replication allowed. According to this concept, replication based routing protocols can be divided into: *controlled* or *uncontrolled*, depending on whether there is some control over packet replication or not.

The most fundamental *uncontrolled* replication based protocol is *Epidemic Routing* [58]. In this scheme, nodes replicate and forward packets to any encountered node that does not already own a copy of the packet. The packets spread like an epidemic disease (hence the name of the protocol). Several *controlled* variations of this protocol can be found where some constraints are added to limit the number of packet replications.

Spray and Wait[59], and *BubleRap*[60] are examples of *controlled* replication based protocols. *Spray and Wait*[59] alternates between two phases. A *Spray phase* where it forwards copies of the message to any encountered node up to a maximum number L. A *Wait Phase* where the node remains idle and only forwards the packet if the encountered node is the destination. On the other hand, *BubbleRap* [60] exploits the social component of encountered nodes. This protocol defines a ranking of social communities, and limits the replication to nodes socially connected to the destination.

2.2.3 Quality of Service

Achieving Quality of service (QoS) in an opportunistic network is an often untreated topic. The unpredictable nature and the long delays make offering any particular degree of service challenging. The most common research efforts are

usually focused either on improving the routing protocols (usually through efficient buffer management policies), defining traffic priority classes or congestion control mechanisms.

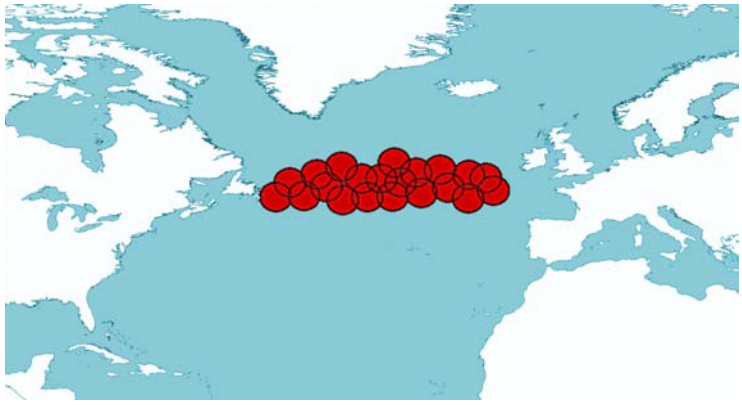
Buffer management policies are examined in several works [61], [62, 63]. In [61], the authors offer a comparative review of existing buffer management policies implemented in several well-known routing protocols. This comparison is performed using delivery probability and message overhead. As a result, an optimal algorithm for buffer management is proposed. In [62, 63, 64], the authors suggest a complementary mechanism for routing protocols. This system uses admission control policies to accept or reject traffic demands according to the current available resources. They demonstrate that their approach improves performance based on delivery probability and fariness resource allocation metrics.

Traffic priority classes are provided by the standard DTN architecture [40]. Specifically, the Bundle Protocol [41] defines packet header priority classes with three well-defined priority levels. Additionally, the Bundle Protocol [41] provides congestion control mechanisms as packet delivery options. These options are a set of signals indicating several events that may help manage QoS related processes, such as congestion control or routing. In [65] we can see another example of congestion control mechanisms. The authors propose a local and autonomous rule based congestion control mechanism using a financial model of buffer space management. As a result, this proposal does not add communication overhead and provides tolerance to failures due to loss of connectivity.

2.3 Opportunistic Aeronautical Networks

In our conception of opportunistic aeronautical networking, we assume that each airplane is equipped with a simple communication system with a communication link of just a few tens of kilometers. As opposed to a fully connected AANET (see Figure 2.1a), where vast radio ranges allow the formation of a fully interconnected network. This limited radio range leads to a sparsely connected network (see Figure 2.1b) where nodes are usually isolated, and communications are performed on sporadic occasions. These characteristics are fully in range with those displayed in opportunistic networks.

In the known literature, works assuming similar connectivity premises as ours are scarce. On one side, we can find the previously mentioned protocol suite AeroRP[29]/AeroTP[36] wich provides disruption tolerance for highly



(a) Aeronautical Ad-hoc Network

(b) Opportunistic Aeronautical Network

Figure 2.1: Aeronautical network graphical concept

changing airborne networks. However, their assumptions fall under the domain of connected MANETs. On the other side, we can find a handful of works that operate under the assumptions of DTNs. As far as we know they are limited to [66],[67],[68], and [69].

In [66], the authors propose the use of a DTN architecture for commercial aircraft on scheduled routes. This work explores aeronautical DTN concepts, requirements and application scenarios. This document is a general proposal

and does not include sound empirical or theoretical analysis. Follow-up works by the same authors [62, 63, 64] explore the problem of QoS provisioning and resource allocation in aeronautical DTN networks.

On the same line of work, [68] provides an overview of the requirements of airborne networks and details how the DTN architecture can be applied to enable reliable IP-based communications.

Another case is [67], the authors assume an airborne network with DTN characteristics, and define a synthetic small scale (200 nodes) scenario. The authors analyze the performance of common opportunistic routing protocols (*Epidemic*[58], *Prophet[70]*), versus their proposal: History of Encounters Probabilistic Routing Algorithm (HEPRA). Finally, in [69] the authors propose and evaluate a DTN replication based routing protocol for sparsely connected aeronautical networks.

2.4 Summary and Conclusions

In this chapter, we reviewed the current trends in ANEETs and opportunistic networking. We now highlight some of the key ideas presented in this chapter. First, MANET routing protocols are usually not applicable to AANETs due to the increased rate of topology changes. Second, the fundamental principles of MANET and opportunistic routing are entirely different. Therefore, common AANET protocols are not applicable to opportunistic aeronautical networks. Third, proper mobility characterization is a critical aspect of opportunistic network analysis and protocol design. Additionally, the use of realistic mobility is of great importance during the design of any network architecture or protocol, to ensure adequate deployment in real networks. The use of opportunistic approaches in the aeronautical field is relatively unexplored and lacks the extensive analytical component found in other kinds of opportunistic networks.

Lastly, we have reviewed several works that offer proposals to improve QoS in opportunistic networks. Despite that, at the moment of writing of this dissertation we have not been able to find existing work in the literature capable of guaranteeing specific levels of service or offering a Service Level Agreement (SLA) [2].

[2] A service level agreement (SLA) is a contract between a service provider and the end-user. This contract defines the level of service expected from the service provider according to several performance metrics.

To the best of our knowledge, this thesis represents the first in-depth analysis of a realistic large-scale aeronautical opportunistic network. Our work provides careful analysis of topological and mobility aspects as well as routing performance evaluation, and QoS considerations.

Part II
Network Characterization

Oh well, I suppose lots of people will do it now.

Arthur Whitten Brown to
Capt. John Alcock

At Cliften, Ireland, after completing the first transatlantic flight. 1919.

3

Scenario Description and Mobility Model

In this chapter, we will introduce the scenario used to evaluate the proposals made in this thesis. We will describe the several aspects of our scenario, the mobility of the nodes, and methodology used to define our mobility model including its multiple characteristics. This model was implemented as a modular extension of the ns-3 [71] network simulator. We conclude the chapter with the design and implementation details, and describe each of the parts comprising the module that manages the mobility of the nodes in the scenario.

3.1 Scenario

Our network scenario represents an aeronautical ad-hoc network composed of nodes (aircraft) moving at high speed. We seek to analyze the behavior of such a network in a disconnected environment and thus, we focus in transoceanic flight scenarios which are characterized by the lack of ground infrastructure and long intervals of disconnection. The nodes in this scenario use wireless

radio technologies with limited communication ranges. Therefore, the lack of infrastructure and their limited communication range forces them to rely on sporadic encounters to exchange and forward data in an opportunistic manner.

The scenario is focused on the North Atlantic oceanic area, due to the fact that this is the ocean with the highest concentration of flights. We define a geographically bounded area (see Figure 3.1) where the planes are disconnected. This area will be referred as **disconnected zone**. We will also refer to the time interval in which each aircraft remains in this disconnected area as the aircraft's **disconnected flight duration**. Our study focuses on the intervals when the aircraft are operating inside of this disconnected area, and the nodes operating in this are will referred as **active nodes**. When the nodes leave the disconnected area, we assume that they are capable of delivering their data using existing ground infrastructure, and thus their opportunistic capabilities are inactive.

Figure 3.1: Scenario representation: geographical bounds of the disconnected zone

The scenario consists in a set of aircraft traversing the previously defined oceanic area in both directions according to a set of predefined flight routes. These routes and their schedules are based on real data gathered from the flights departing on the same concrete day (Thursday, 03/14/2013, all time-zones included). Departure times are converted to UTC time format in order to provide a standardized time for the whole network. Simulation starting time is synchronized with the instant 0:00 AM - UTC time zone. The details on the

definition of routes and schedules are provided in the following section.

3.1.1 Communication Models

In this scenario, we can define three types of communication based on the communication's purpose, the origin and the destination of the data:

- **Air-To-Ground:** This kind of communication seeks to deliver data generated on-flight to the ground, such as passenger data or aircraft status notifications.
- **Air-To-Air:** This kind of communication seeks to deliver data generated on-flight from one node to another. One example of this may be the exchange of flight-route information between planes (such as the meteorological conditions of a previously traversed area).
- **Ground-To-Air** This kind of communication tries to deliver data generated on-ground to a node which is currently on-flight. Examples of such application are key distribution or any kind of two-way communication.

We can see that the pattern of node mobility ensure that all the nodes reach connected areas at the end of their journey. Therefore, any data being carried by a node will always be delivered (assuming no data losses), and the end of the flight will mark the latest possible delivery opportunity. Thorough this thesis, we will focus on **Air-To-Ground** communications. Our main objective will be to provide cost-effective, reliable communications for air to ground data delivery while minimizing communication delays.

3.2 Mobility Model

In this section, we describe the design and characteristics of the mobility model. The characteristics that define this mobility model are summarized as follows:

3.2.1 Node Trajectory

Nodes follow geodesic patterns crossing over the Atlantic ocean. Routes depart from Europe to America and vice versa. Trajectories are constructed using datasets from [72]: a user maintained public domain database based on the

latest DAFIF[1] public release. This database contains a total of 59,036 routes between 3,209 airports from 531 airlines. We use the subset of routes that cross the North Atlantic ocean (up to 239 routes) which are shown in Figure 3.2.

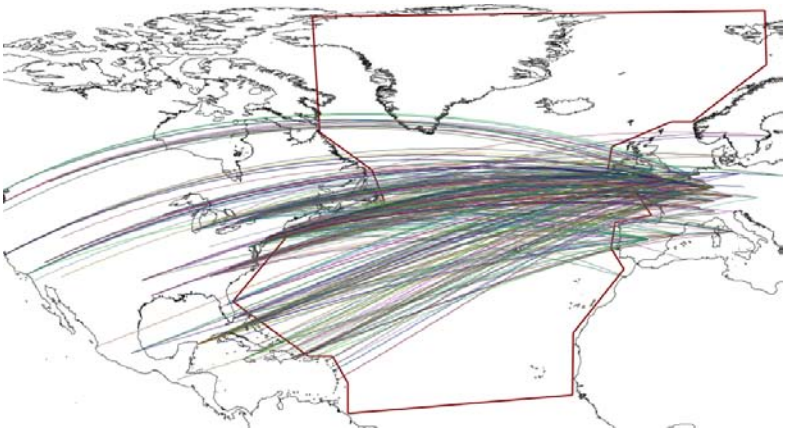


Figure 3.2: Mobility Model: geodesic representation of transatlantic routes over the North Atlantic oceanic area.

Trajectories are computed for each route using the geographic position of the source and destination airports as their starting and ending point. We then compute the geodesic trajectories followed by the planes using GIS (Geographic Information Systems) software, concretely the ArcGIS [73] and OpenJump[74] software applications. The trajectories are then stored as a set of intermediate points using a 2D projection with coordinates in the standard WGS 84[2] coordinate system.

3.2.2 Node Departure Scheduling

Each of the previously defined routes is traversed by one or multiple planes, these planes are scheduled using real world-data. We gather information on the concrete number of planes traversing each route and their corresponding

[1]Digital Aeronautical Flight Information File: A complete database of up-to-date aeronautical data managed by the National Geospatial-Intelligence Agency.

[2]World Geodetic System 84: the reference coordinate system used by the Global Positioning System.

departure times obtained from several flight status providers like [75] and [76]. This data was collected using web scraping techniques through the use of a custom web crawler implemented in Java and the HtmlUnit[77] library. We include all the flights with a scheduled departure in the same day (Thursday, 03/14/2013), in all time zones, and use a unique simulation node for every departing flight (up to 2,878 nodes).

Inclusion of Real-World Departure Delays

Daily air traffic commonly suffers deviations from scheduled departure times. In this section we consider those changes and analyze the impact they have in our network model. To include these delays in our model, we compared scheduled departure times (gathered one day before departure) against the actual departure times. This data was gathered from all departing flights during a time lapse of 5 days. The histogram of the gathered samples is shown in Figure 3.3, which displays a bell-shaped distribution centered around a positive delay of 980 seconds. As there is no good fit using commonly known probability distributions, we use kernel density estimation [78] to describe the probability density function of the gathered samples, which is provided by the expression:

$$\hat{f}_h(x) = \frac{1}{nh}\sum_{i=1}^{n} K\Big(\frac{x - x_i}{h}\Big) \text{ for } h = 174.491$$

$$\text{and Gaussian kernel } K(u) = \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}u^2}$$

We compute the cumulative distribution function under the range of observed delays and use the inverse-transform sampling method [79] to generate pseudo-random numbers according to this distribution. These values are added to the scheduling procedure as a random parameter that describes the deviation of each plane from its initially scheduled flight plan.

3.2.3 Node Mobility

Nodes move at a constant speed of Mach 0.84 (285.852 m/s), which is (according to specifications) the standard cruise velocity for most commercial aircraft models. Node altitude only changes during take-off or landing. All nodes flight at the same altitude, and we do not consider multiple flight levels. The flight

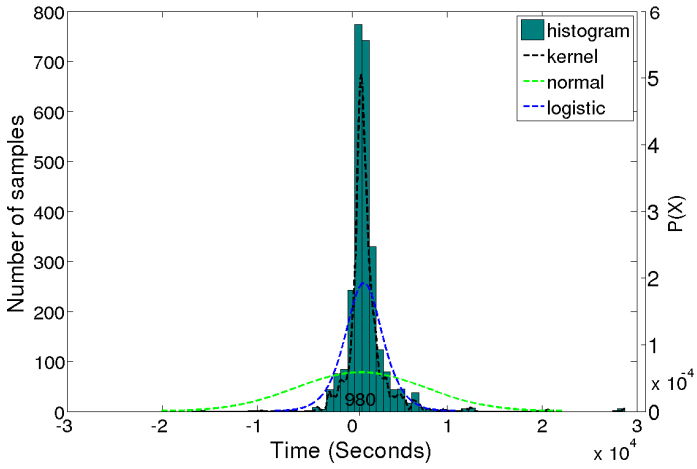


Figure 3.3: Histogram of real-world flight departure delays with fittings to common probability distributions.

level has little relevance due to the use of omni-directional antennas with a range of several kilometers (details given in the next section).

3.2.4 Node Separation

Our model, uses a simplistic separation ATM (Air Traffic Management) model which only takes into account same course longitudinal separation guidelines. Upon scheduling a new flight our procedure checks it against previously recorded flights, in case that two aircraft come within a 10 min flying time of each other, the new flight departure is postponed. Also, note that the unique set of guidelines OTS (Organized track system) used in the North Atlantic ocean is not being applied.

3.2.5 Simulation Model: Design and Implementation

Our model was implemented as new mobility model for the ns-3 simulation framework. It was designed as an extension of a waypoint based mobility model whose initialization is performed using precomputed mobility traces and schedules. In this section we will describe the structure of this precomputed data and

the design details of our ns-3 implementation.

Precomputed Data

Geographic data files are stored in *wkt* (well-known text)[80] file format and represent several geographical aspects of the scenario using 2D primitives in the WGS 84 coordinate system (Longitude, Latitude):

- Airport Locations: are represented as single points with coordinates.
- Aicraft Route Trajectories: are precomputed geodesic aircraft trajectories stored as 2D projections. They are represented as a set of connected points corresponding to the routes from Figure 3.2.
- Scenario Bounds: are represented as a set of connected points forming a polygon. The area shown in Figure 3.1 is one example.

Aircraft departure schedules files are stored in *csv* [81] (comma separated values) file format. Each route has a corresponding scheduling file containing information on all departing flights. The stored information includes: departure and destination airport codes, flight number, airline identifier, and departure times.

Software Design

In this section, we will describe the design of our implementation. We offer the UML class diagram in Figure 3.4. Our implementation includes three new types of nodes, a mobility model and several auxiliar classes responsible of mobility scheduling and management. From a functional perspective the several classes perform three clearly distinguished tasks:

Import Procedure

These classes are responsible of importing the pre-computed data files for use in the simulator mobility model.

The first step, is reading files from disk and interpreting its contents, this is performed by parser classes. We use an *Interpreter pattern* and provide a pure abstract class **Parser**, this class defines the virtual method *interpret* responsible of parsing a string. Inherited classes are responsible of implementing this method following the structure of the file format they are intended to parse. We

provide the classes **WktParser** and **ScheduleParser** which are responsible of parsing *wkt* and *cvs* files respectively.

The **WktParser** is used by classes representing geographical concepts. The main class of this kind is the base class **GeoDataFile** which represents the contents of an *wkt* file. There are three specifications of this class: **AirportLocation**, **Route**, and **ContactLocation** classes. Each one represents one case of the previously defined geographic data files: airport locations, aircraft route trajectories, and scenario bounds.

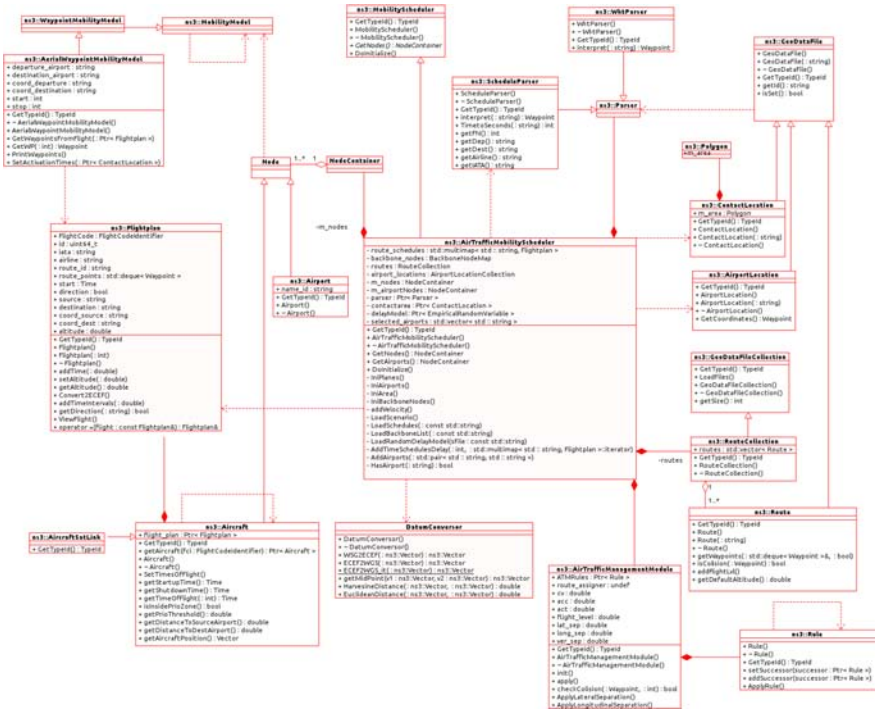


Figure 3.4: UML class diagram of the mobility model.

Mobility Model

Our mobility model is a specification of the standard ns-3 **WaypointMobilityModel**. A waypoint is a structure with two components: a position vector

(x,y,z) and a time variable *t*. A waypoint mobility model computes course changes based on a list of waypoints which defines the position of the node at each specific instant of time. We provide the mobility model **AerialWaypointMobilityModel**, this class extends functionality providing additional methods that allow initialization using a **Flightplan** object. We offer two child classes of the **Node** class:

- The **Aircraft** class: represents a mobile node holding a **Flighplan** object.
- The **Airport** class: represents a non-mobile node with some additional attributes.

These classes extend the functionality of the **Node** class and provide methods to acquire flight related information. The creation, initialization and association of the previously described classes is performed by *Scheduler* objects. Additionally, the **Aicraft** class is further subclassed on the **AircraftSatLink** class. This class represents an aircraft with a dedicated satellite link. The implications of this type of node are properly discussed in Chapter 8.

Mobility Scheduling

The scheduler classes are responsible of importing the external precomputed data and use it to create nodes with an initialized mobility model. We provide an interface **MobilityScheduler** and its implementation **AirTrafficMobilityScheduler**. This scheduler, provides several *Loader* methods which are responsible of importing geographical data using the **GeoDataFile** child classes. Specifically, the trajectories stored in **Route** classes are represented using only two dimensions (latitude and longitude), the flight altitude is then added to obtain a third dimension followed by datum conversion into ECEF[3], for compatibility with the Cartesian coordinate system used by the ns-3 *MobilityModel* class. Conversion between coordinate systems is provided by the **DatumConversor** class. Another *Load* method is responsible of importing the aircraft departure schedules using the **ScheduleParser**.

The other type of methods are the *Initializers*, these methods invoke the *Loaders*, and use the obtained information to generate simulation nodes. The most relevant is the method responsible of creating the aircrafts. The pseudocode of this method is shown in Algorithm 1. First, the geographical routes

[3]Earth-Centered, Earth-Fixed is a Cartesian coordinate system which takes the center of mass of the Earth as point (0,0,0).

Algorithm 1 High Level Mobility Aircraft Initialization Procedure: Each node is assigned a mobility model defined as a list of waypoints.

```
nodes ← []
LoadScenario()
▷ Initilize Random Delay model.
delayModel ← LoadRandomDelayModel()
routes ← LoadRoutes()
for all route in routes do
    flights ← LoadSchedules(route.name)
    for all flight in flights do
        node ← []
        ▷ Add Random Delay to scheduled flight.
        flight ← delayModel.GetDelay()
        ▷ Create Waypoint list initializing the position component.
        flight.waypoints ← route.getWayPoints()
        node.flight ← flight
        ▷ Assign the Waypoints' time component assuming constant speed.
        node.SetTimesOfFlight()
        ▷ Enforce ATM separation regulations.
        AirTrafficMangamentModule :: Apply(node.flight)
        nodes ← nodes + node
    end for
end for
```

are loaded, next the corresponding aircraft schedules for that route are also imported. Finally, a **FlightPlan** object is initialized using this data. At this point, the departure time of the Flight is modified according to a **Delay Model** provided by the output of an *Empirical Random Variable* following the distribution from Figure 3.3. The flightplan is then modified again to account for Air Traffic Management Regulations. This task is performed by the class **AirTrafficManagementModule** which acts as handler for the class **Rule**, this class is implemented as a *Chain-of-responsibility pattern*, rules are expressed as a chain of ordered objects that are applied sequentially.

The final tasks of the *Initializer* procedure deal with the assignation of the mobility model to its corresponding node. First, a new instance of the **AerialWaypointMobilityModel** is created, the mobility model initialization method

is then invoked using the thoroughly prepared **Flightplan** as its parameter. This method takes care of creating the way-point list. Finally, the mobility model is assigned to a new **Aircraft** object. Finally, this object is added to a **NodeContainer** that stores all the nodes in the simulation.

3.3 Summary and Conclusions

In this chapter, we have introduced the basic features and characteristics of the scenario followed by the description of its mobility model. We described the mobility patterns and provided a detailed enumeration of all the parameters that define the model. Lastly, we described the corresponding simulation module and offered design and implementation details. In a way, this chapter provides a description of how the nodes move. In contrast, the next chapter is going to describe their networking characteristics and capabilities.

The Quote Of The Day (QOTD) service is a member of the Internet protocol suite, defined in RFC-865.

J. Postel

May 1983

4 Network Model

IN this chapter, we will outline node networking capability for the scenario previously introduced in Chapter 3. This chapter will define the network model used to evaluate the proposals made in this thesis. We describe all the layers of the network stack, including network protocols and specific configuration details. This model was implemented using the features of the ns-3 [71] network simulator. We conclude each section with the design and implementation details of each simulation module.

4.1 Physical Layer

In this section, we analyze the requirements of our physical layer. The physical layer defines the basic networking hardware transmission technology, the means of transmitting raw bits over a transmission medium, including the shapes and properties of electrical components, and the frequency and modulation of the signal itself. We start off by describing the specific characteristics of our selected model from a theoretical perspective. We close the section offering the several

modeling aspects of our implementation.

4.1.1 Communication Link

As a communication link, we propose a cost-effective system based on IEEE 802.11b/g[82] with a high gain and high sensitivity (low noise) transceivers, directly connected to omni-directional antennas. The communication range should be of several dozen kilometers in order to allow contact opportunities considering the size of the scenario and the speed of the nodes. In the following sections we estimate the several parameters required to attain a communication link with the described properties.

Transmission Environment and Communication Path

We now consider the transmission path between nodes in the scenario. In the proposed aeronautical scenario, nodes fly at several thousand feet high, no obstacles can be assumed inside the first Fresnel zone[1]. Therefore, free-space communications and free-space path loss can be considered.

Estimating Theoretical Radio-Range and Transfer Rate

We estimate the theoretical maximum range of the communication link. First, we consider the signal power at the receiver (P_{RX}) which is calculated with the following parameters:

- The transmission power (P_{TX}).
- The transmitter and receiver antenna gains (G_{TX} and G_{RX}, in dBi).
- The free-space path loss (L_{FS}, in dB).
- Miscellaneous losses ($L_M = 6dB$). This parameter would include rain and fog fade which for 2,400 MHz would is minimal (up to 0.05 dB/km) .

$$P_{RX} = P_{TX} + G_{TX} - L_{FS} - L_M + G_{RX}$$

[1] Fresnel zones are concentric ellipses centered around the direct transmission path of the radio beam of two communicating parties. The first fresnel zone must be mostly kept free from obstacles in order to guarantee radio reception. Usually, the maximum obstruction allowable is 40%, but the recommended obstruction is 20% or less.

The maximum free-space path loss allowed (L_{FSmax}) will be when the P_{RX} equals the sensitivity of the receiver (S_{RX}).

$$L_{FSmax} = P_{TX} + G_{TX} - L_M + G_{RX} - S_{RX}$$

Moreover L_{FS} (in dB) can also be calculated from the frequency of the signal (f, in Mhz), and the distance of the link (d, in km).

$$L_{FS} = 32.44 + 20\log_{10}(f) + 20\log_{10}(d)$$

Taking the two previous formulae, the maximum range for a usable link (d_{max}) will be obtained for L_{FSmax}.

$$d_{max} = 10^{(L_{FSmax} - 32.44 - 20\log_{10}(f))/20}$$

Table 4.1 shows the maximum range of the given communications link for all the IEEE 802.11b/g transfer rates (TR), including all the values used to calculate it, for the proposed link. It can be seen that a 49.864 km maximum communications link can be obtained (for a 1 Mbps transfer rate).

Although the transmission power needs to be increased to 36 dBm which is the maximum allowed limit for antennas with 6 dBi or lower gain. Finally, it must be taken into account that the transfer rates provided by IEEE 802.11b/g are the link transfer rates and the theoretical maximum. Usually, real achievable transfer rates are around 55-60% of those maximum values.

Equipment Selection

This sort of configuration is easily achievable with low-cost commercial solutions. As an example, Bullet 2HP transceivers [83] offer really similar parameters. For 1/54 Mbps transfer rates provide 29-23 dBm transmission power with a -97/-72 dBm sensitivity, could be used, together with 6 dBi omni-directional antennas in order to have a low directivity in the elevation (vertical) plane.

On a side note, this is not an exclusive proposal. As long as cost can be kept to a minimum, the desired data rates are achievable, and the radio ranges can be reached, other physical alternatives can be used. A few examples of technological alternatives can be: Optical Wireless Communications (OWC)[84] or 802.16 technologies (WiMax)[85].

TR (Mbps)	P_{TX} (dBm)	G_{TX} (dBi)	G_{RX} (dBi)	S_{RX} (dBm)	L_M (dB)	L_{FSmax} (dB)	f (Mhz)	d (km)
1 (b)	30	6	6	-98	6	134	2,400	49.864
1 (b)	29	3	3	-97	6	126	2,400	19.851
2 (b)	29	3	3	-96	6	125	2,400	17.692
5.5 (b)	29	3	3	-95	6	124	2,400	15.768
6 (g)	28	3	3	-94	6	122	2,400	12.525
9 (g)	28	3	3	-93	6	121	2,400	11.163
11 (b)	29	3	3	-92	6	121	2,400	11.163
12 (g)	28	3	3	-91	6	119	2,400	8.867
18 (g)	28	3	3	-90	6	118	2,400	7.903
24 (g)	28	3	3	-86	6	114	2,400	4.986
36 (g)	26	3	3	-83	6	109	2,400	2.804
48 (g)	24	3	3	-77	6	101	2,400	1.116
54 (g)	23	3	3	-74	6	97	2,400	0.704

Table 4.1: Maximum range of the communication link for all the IEEE 802.11b/g transfer rates

4.1.2 Simulation Model

The ns-3 simulator contains all the necessary features to describe the physical layer accurately. In this section, we will list the several models used and the configurations chosen.

In order to represent the communication channel the simulator provides the **ns3::YansWifiPhy** object. This object has multiple configurable parameters, we have chosen the configuration that allows us to represent our theoretical 49.864 km Wi-Fi communication link.

- *TxGain:* Corresponding to the transmission antenna gain (6 dBi).
- *RxGain:* Corresponding to the receiver antenna gain (6 dBi).
- *EnergyDetectionThreshold:* Corresponding to the sensitivity (-98db).
- *TxPowerStart:* Corresponding to the transmission power (30 dbm.)

The physical channel is also modeled by a class called **YansWifiChannel**. The channel depends on two main attributes: the channel loss and the delay models which are provided by the classes **ns3::PropagationLossModel**

and **ns3::PropagationDelayModel**. We assume the air as the transmission medium and therefore use a constant propagation delay, this is described by the **ConstantDelayPropagationModel**. The propagation loss model provides an estimate of the maximum transmission distance between two stations before the signal fades too much for communication to be possible. We use the **RangePropagationLossModel** with a configuration of 50 km. This is a simplistic propagation loss model that relies on the physical distance between nodes in order to check whether communications are possible or not. Finally, channel link bandwidth is set to 1Mbps and with Direct-Sequence Spread Spectrum (DSSS) modulation.

4.2 Data Link Layer

As the datalink layer we use IEEE 802.11b working in ad-hoc mode. This is a necessity due to the lack of infrastructure and therefore is required to achieve opportunistic communications. The use of ad-hoc mode has no impact on the MAC layer, all carrier sensing and frame types are the same regardless of the mode used. The only effect is the absence of architectural infrastructure or distinctions between access points or stations.

4.2.1 Simulation Model

In a physical computer system the Network Interface Card (NIC) handles the physical and data-link layer. Akin to how a computer can contain separate interface cards for differentiated data-links, nodes in the simulator can contain multiple *NetDevice* objects to represent separate data-links. In our scenario, each node is equipped with a single *NetDevice* of type **WifiNetDevice** which is responsible of modeling 802.11 based technologies. This model supports the following functions or features:

- **Wi-fi Topological Elements:** Access Points (AP), Non-AP Stations (STA) and STA with IBBS (Independent Basic Service Set) independence (commonly known as Ad-hoc).
- Rentrasmission, Fragmentation, Association and Beaconing.
- **Low-level MAC transactions:** RTS/CTS/DATA/ACK.

- **Media Access Control:** DCF (Distributed Coordination function) and EDCAF (Enhanced Distributed Channel Access Function)
- Multiple **Rate Control Algorithms**.

Each topological element is implemented as a high-level mac sublayer which implements specific functions. In our case, we use the **ns3::AdhocWifiMac** which is a simplistic implementation that does not perform any kind of beacon generation, probing, or association. We also use all media access control and QoS-based mechanisms by default. Finally, we use the **ConstantRateWifiManager** configured to 1Mbps data rate. This ensures the same transmission mode for every packet.

4.3 Network Layer

In this section, we describe the configuration of the network layer. The purpose of the network layer is to perform packet forwarding across networks, as such its protocols defined network addressing and routing protocols.

4.3.1 Internet Protocol

As our network layer we use the Internet Protocol version 4 (IPv4). We deemed that using IPv4 was enough for our experimentation purposes, as most of the improvements and special features provided by IPv6 were not required.

In our scenario, we define a single network which encompasses all of the nodes. We use the the private IP address 10.0.0.0 as our network address. Additionally, and taking into account that this scenario is composed of thousand nodes, we define our network subnet mask as 255.255.0.0, this provides a total of 65536 available addresses.

4.3.2 Routing

Routing is a part of the process of forwarding network traffic among networks. Specifically, it is the process of selecting the best path that data should follow in order to reach its destination. This task is performed through the use routing tables which are lists that map network addresses to specific network interfaces. These tables can be created dynamically by exchanging routing information

between routers, the algorithms and criteria used to exchange this information and decide which is the best path to forward data are called routing protocols.

In our scenario, we use three differentiated routing protocols. The use of a specific protocol depends on the experimental purpose, and the motivation is provided in future chapters. In this section, we will describe the routing protocols from a theoretical and simulation modeling perspective.

Static Routing

This is the most simple routing, it relies on manually configured routing entries. This routing does not generate any additional traffic, since routers do not exchange any sort of routing information. Static routing is not exclusive and can work together with a dynamic routing protocol, in these cases static entries usually act as a failsafe system.

Epidemic Routing

This protocol is a flooding based scheme. These schemes work by flooding the network with broadcasted messages, these messages are then stored and forwarded by all receiving parties. This kind of approach allows to maximize the delivery ratio and improves delivery time by exploring all possible paths available. On the other hand, these schemes increase message overhead and the increased data traffic usually leads to network congestion. Epidemic routing[58], was originally conceived for use in mobile wireless ad-hoc networks, under the assumption that any arbitrary node can act as an intermediate router two other nodes not into immediate radio range. We will now summarize the key aspects of the protocol.

In epidemic routing each node maintains a fixed length buffer. This buffer stores messages originating from this node as well as messages buffered on behalf of other nodes. Messages are stored for prolonged intervals of time until the data can be transmitted or upon expiration after a predefined time interval. For efficiency, each hosts maintains a *hash table* indexing the buffered messages, and a bit vector indicating which entries of the table are set, this vector is known as the *summary vector*.

Possible transmission opportunities are found by periodically broadcasting beacons. When a contact is produced, a procedure known as *anti-entropy* session is initiated. At the start of the session, both nodes exchange their summary vectors and determine which messages stored in the remote node have not been

seen by the local node. Immediately afterwards, each host issues a request of all unseen packets owned by the remote host. Finally, the communicating nodes exchange a copy of every requested packet. As an additional feature, a cache of of previously contacted hosts is maintained, this avoids the establishment of redundant anti-entropy sessions with hosts that have already been contacted.

This protocol has several drawbacks: it does not offer control over data forwarding, produces data overhead, and generates a great volume of network traffic which eventually leads to a reduced delivery ratio due to network congestion. Although, the protocol specification offers several methods and configurable parameters that can be toggled in order to reduce these drawbacks, it is usually very costly when compared to other routing protocols.

Constrained Epidemic Routing based on Landing Time Estimation

As a way to cope with the congestion drawback inherent to the Epidemic Routing protocol, we implement a new version of this protocol with specific transmission constraints. This new protocol uses a decision function based on a custom metric called Expected Landing Time (ELT). This metric provides is obtained thanks to the deterministic nature of the mobility model, nodes have access to all detailed information about their routes and landing times.

Our routing protocol uses a limited flooding approach, its inner processes are mostly analogous to those of epidemic routing protocols. The main differences are related to the *anti-entropy session*. Our protocol adds an additional intermediate step where nodes exchange the value of their ELT metric together with their summary vectors. The respective values of the metricx are compared, and only the node with a higher ELT value proceeds to make a request for messages.

In this way, we reduce the number of transmissions and data overhead in the network. As we will see in a future chapter, this is an adequate optimization due to characteristics of the scenario and several aspects of the contact opportunities.

4.3.3 Simulation Model

In the ns-3 simulator, the basic model provided for the IPv4 protocol is implemented in class **Ipv4L3Protocol**, additionally all routing and forwarding are delegated to a class called **Ipv4RoutingProtocol**. In order to define new routing protocols this class needs to be derived and its methods must be implemented. Incoming packets are handled by the method *Ipv4L3ProtocolRe-*

ceive() which delegates into *Ipv4RoutingProtocol::RouteInput()* to decide how the packet should be handled. The packet is then processed by calling one of the four possible callbacks: *LocalDeliver*, *UnicastForward*, *MulticastForward*, and *Error*.

The ns-3 implementation allows multiple routing protocol working independently in what is called *ListRouting*. The IP protocol queries each of these protocols sequentially following a priority order until a route is found. In our scenario, we use this option with the several routing protocol combinations:

- **Ipv4StaticRouting**: Is set as our lowest priority routing protocol. This static routing implementation is part of the standard ns-3 build.

- **EpidemicRouting**: Is set as the first priority routing. We used the implementation from [86] which is not currently a part of the standard simulator.

- **EpidemicELTRouting**: Is set as an alternative priority routing that substitutes the standard *EpidemicRouting*. In order to implement this model, we used [86] as a base and extended it to provide the following additional features: flight information gathering, ELT metric evaluation, and extended *anti-entropy* session.

Our scenario has a specific requirement that needs to be considered by routing protocols. All packets being carried by a node need to be immediately delivered to the destination upon exiting the bound of the oceanic area 3.1. Since there is no dedicated link is available to perform a direct delivery, we use a workaround to emulate this behaviour. This was achieved by performing several changes to the **Ipv4L3Protocol** and **EpidemicRouting** classes. First, the packet header is marked with a special type of flag. Secondly, a new method called *DeliveryOnDisable* is added to the routing protocol. This method gets called when an aircraft becomes inactive (leaves the oceanic area), their method perform a *LocalDeliver* callback to deliver the packet. Finally, upon receiving the marked packet the instance of the IP protocol treats the delivery as if it was actually delivered to the destination.

The family of epidemic based routing protocols have several attributes that influence the behavior and performance of the protocol. These attributes act as configurable parameters and are described below:

- *BeaconInterval*: Defines the time interval between successive broadcasts of beacon messages.
- *HopCount*: Defines the maximum number of allowed hops for a forwarded packet.
- *BufferSize*: Defines the maximum number of packets the buffer can hold.
- *EntryExpireTime*: Defines the maximum time a packet can remain in the buffer before it gets dropped due to expiration.

4.4 Transport Layer

In this section, we describe the configuration of the transport layer. The transport layer is in charge of providing end-to-end communications. Protocols in this layer usually prove connection-oriented data transmission, reliability and flow-control services. Despite of that, under the conditions found on DTN scenarios, intermittent contacts between nodes, and highly variable link performance, the existing Internet transport protocols become unsuitable.

In our scenario, we use the UDP protocol (User Datagram Protocol). This protocol uses a simple connectionless transmission mechanism with minimum protocol algorithm, it is stateless, and does not offer reliability (leaving this task to the upper layer protocol). This is adequate for our DTN scenario, we use multiple application depending on the experimental purpose, some need to be connectionless, and do not require reliability. Others handle the reliability themselves, or a intermediate DTN specific reliable protocol may be used (such is the case of LTP (Licklider Transmission Protocol) protocol described in Appendix B).

4.4.1 Simulation Model

In the simulator, each transport protocol implementation is provided as a socket factory. Applications may use the socket API and specify the type of socket in order to communicate using a specific Layer 4 protocol. In real system sockets the application needs to read the socket in order to obtain the data (this process usually implies blocking I/O). In ns-3 this is implemented as an asynchronous model based on a callback system. Applications defining a socket specify a

callback to be invoked once data is received. The transport protocol then invokes this callback when the data is available.

4.5 Application Layer

The application layer specifies shared services between programs running in different computer systems. In our scenario, we use two types of models corresponding to this layer:

4.5.1 Application Models

Application models are implementations that reproduce the behaviour of concrete applications in great detail. Such models may include internal logic, internal data structures, and the all of the steps in communication exchanges.

Neighbor Discovery Service

This is an application layer service used to asses network connectivity by registering and keeping track of active neighbors. The service uses multicast over UDP to periodically broadcast beacon messages. Simultaneously, the application keeps track of incoming messages originated by other nodes and registers them as contacts. Information about contacted nodes is then stored in a cache of active neighbors. Cached entries have a fixed duration and are automatically removed upon expiration. This application provides a way to gather neighbor information such as: contact patterns, frequency, or locations. The service is designed as a sub-layer and may be used below other applications. Interested applications can register within the service in order to receive notifications upon detection of new neighbors.

Simulation Model

In Figure 4.1, we offer the UML representation of our Network Discovery Service implementation. The model is implemented in class **NeighborDiscovery** a subclass of the **ns3::Application**. The base class provides two virtual methods that must be implemented *StartApplication* and *StopApplication*. We use the *StartApplication* method to initialize the listening and receiver sockets using a *ns3::UdpSocketFactory*. A callback function is assigned to the listening socket

to handle incoming data. An event is scheduled to send data periodically using the sender socket in order to represent the beaconing system. Cache entries are represented with the class **NeighborStats**. The cache is represented as a map of *NeighborStat* entries indexed by node IP addresses. The model has several attributes acting as configurable parameters:

- *BeaconInterval*: Represents the time interval between successive broadcasted beacons.
- *CacheExpiration*: Defines the duration of cache entries.
- *ListenerPort*: Defines the UDP port used to listen for incoming beacons.
- *PacketSize*: Defines the size of the broadcasted packets.

Lastly, we describe the several subscribed applications to the *Neighbor Discovery Service*: First, the **NeighborDiscoveryMonitor** class which is used to compute contact statistics and to evaluate several metrics regarding encounter properties. Secondly, the **GraphBuilder** class which is capable of building network topology graphs based on information about pair-wise node contacts.

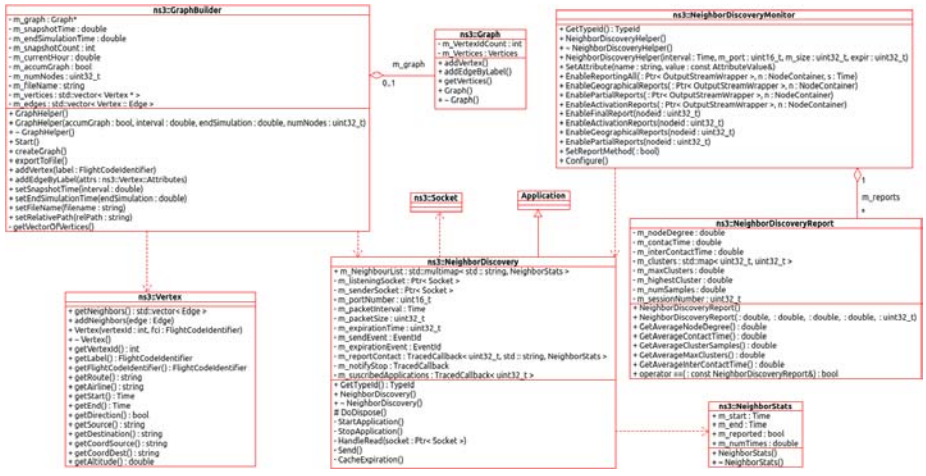


Figure 4.1: UML class diagram of the Neighbor Discovery Service

4.5.2 Traffic Models

These are generic theoretical models that emulate the traffic patterns generated by real applications. Details on each model depend on the purpose of the experimental setup, it will be provided at the start of each section.

Constant Bit Rate On-Off Traffic Model

The Constant Bit Rate (CBR) model is a simple traffic model that generates network traffic by sending periodic packets at a constant rate. The properties of this traffic are mainly defined by two parameters: the link data rate and the size of the packet.

Simulation Model

The ns-3 simulator provides a CBR traffic generator known as **OnOffApplication**. This application generates CBR traffic according to an *On/Off* pattern. The application has two states (On/Off) which periodically alternate. A random variable determines the duration of each state. During the *Off* state, no traffic is generated. Inversely, during the *On* state CBR traffic is generated by periodically transmitting packets. The model has several attributes acting as configurable parameters:

- *DataRate:* The data rate during the *On* state.
- *PacketSize:* The size of packets sent during the *On* state
- *MaxBytes:* The total number of bytes to send.
- *OnTime* and *OffTime*: Random variables used to select the duration of the *On* and *Off* states respectively.

Four Interrupted Poisson Processes (4IPP) Traffic Model

The 4IPP[87] is a general traffic model originally designed for performance testing of PHY/MAC on IEEE-802.16 fixed WAN applications. This model accurately generates self-similar traffic found in Ethernets and on the Internet. The main application of this model is to simulate aggregated HTTP and FTP traffic. A characteristic of this model is that simulated traffic is one-directional, to simulate both outgoing and incoming traffic two independent 4IPP instances are required.

	c1 probability rate (transitions/sec)	c2 probability rate (transitions/sec)	λ (pkts/sec)
IPP#1	4.571e-01	3.429e-01	1.1480
IPP#2	1.445e-02	1.084e-02	0.7278
IPP#3	4.571e-04	3.429e-04	0.5949
IPP#4	4.571e-06	3.429e-06	0.5289
		4IPP Average Rate: (packets/unit-of-time)	3

Table 4.2: Basic 4IPP traffic model parameters.

The model represents the network traffic observed between a Router and a LAN with multiple computers. The model describes two general states: the **ON** state where generated traffic is destined for the router, and the **OFF** state where generated traffic is exchanged between the internal nodes of the LAN.

This traffic model is constructed using a superposition of four Interrupted Poisson Process (IPP). Each IPP is a process with two internal states ON/OFF. During the ON state a flow of packets is generated at a specific rate λ, during the *OFF* state there is not packet generation. The average size for generated packets is 1536 bits (192 bytes). Transitions between both states happen with probabilities c1 (*ON* to *OFF*) and c2 (*OFF* to *ON*). The resulting network traffic can be perceived as periodic bursts of packets spaced by intervals of inactivity.

This model has been proved to simulate Internet traffic accurately [87] and closely resembles aggregated traffic measurements. The values corresponding to standard model parameters can be seen in Table 4.2, in rows 1-4 we represent the parameters for each IPP. In row 5 we show the aggregated of all four processes, offering the average generation ratio of the traffic model.

Simulation Model

An IPP is implemented as two **ExponentialRandomVariables** and one sender socket. The random variables are used to define the duration of the On/Off intervals. The socket is used to generate traffic by sending data periodically during the *On* state. On the other hand, during the *Off* state, no traffic is generated. The 4IPP is constructed by simply using four independent IPP

simultaneously (each one with their socket). The model has several attributes acting as configurable parameters:

- *Lambda#* (where # may be 1 to 4): represents the data rate during the *On* state for one IPP.
- *PacketSize:* The size of packets sent during the *On* state
- *MaxBytes:* The maximum number of bytes to send.
- *IPP#C1* and *IPP#C2* (where # may be 1 to 4): specifies the mean value of the exponential variable for one IPP.

The model by default uses the values shown in Table 4.2. When the model needs to be used for experimentation, all the parameters must be scaled to give suitable data rates.

4.6 Summary and Conclusions

In this chapter, we outlined the networking capabilities of the nodes in the scenario. We described all the layers of the network stack, including description, and implementation details. We conclude each section with a description of the configuration parameters that characterize each protocol. In future chapters, we will include a section describing the chapter's experimental set-up. In that section, we will reference the protocols described in this chapter and provide the specific values for each configuration parameter.

In opportunistic networks mobility is indeed the only possible communication means: without mobility, network-wide communication would not be possible at all. This is a very distinguishing feature of opportunistic networks with respect to other types of next generation wireless networks, which also explains why so many efforts in the research community have been devoted to understanding and modeling mobility in opportunistic networks.

Paolo Santi

Mobility Models for Next Generation Wireless Networks: Adhoc, Vehicular and Mesh Networks. 2012.

5 Network Mobility Characterization

IN opportunistic networks such as the one in our scenario, mobility represents the key aspect in the process of delivering information from source to destination. It is then obvious the necessity to provide and in-depth analysis of the impact on the scenario, as well as the properties of mobility itself. In this chapter, we analyze how mobility influences the network topology and the properties of the encounters between nodes. This chapter introduces several metrics that allows us to determine and quantify the several network properties. We offer two differentiated experimental models and conclude the chapter with a detailed discussion of the experimental results.

5.1 Experimental Models

In this section, we seek to analyze multiple aspects of the scenario. The impact of the mobility patterns on the overall network topology is constrained by several external parameters. In order to properly analyze and quantify the relevance of these parameters we use two differentiated experimental models.

5.1.1 Incremental Partial Model

This model studies the influence of the number of nodes and the radio coverage on the network topological characteristics using our mobility model. This model is used to identify the minimum network properties that allow opportunities of communication. For these experiments we have used a subset of nodes that range from 100 to 1,500, this subset is chosen uniformly at random from all the nodes of the scenario (2,878). The wireless coverage uses ranges from 10 km to 20 km. Results from this model are used to recognize patterns and relationships between mobility metrics and specific scenario characteristics. It is also an interesting model that can be used as a first approximation to other less populated oceans, since the velocities and overall mobility patterns are equivalent.

5.1.2 Complete Model

This model uses the whole set of 2,878 nodes and wireless ranges at either 50 km or (if specified) oscillating from 20 to 50 km. This model represents an approximation to a real-world trace. Therefore, we use it to analyze the key properties of the encounters between nodes. Results from this model are useful to: estimate effective data transfer rates per encounter, fine-tuning configuration parameters for network protocols and applications, and to define scenario specific optimal routing protocols.

5.2 Experimental Configuration

In this section, we describe the experimental set-up used to perform the several measurements and experiments. It is composed of the following protocols and models:

- Mobility Model described in chapter 3.
- Network Layers 1 to 4 described in chapter 4.
- *Static Routing* with predefined routes (described in chapter 4.3.2).
- *Neighbor Discovery Service* application service (described in chapter 4.5.1)
- Experimental models: Incremental Partial Model and Complete Model.

Simulation Parameter	**Value**
Radio Communication Range	10-50 km
Simulated Time	48 hours
Number of Nodes	100-2,878
Neighbor Discovery Service	
Beacon Interval	10 Seconds
Cache Expiration	250 Seconds
Packet Size	100 bytes
Subscribed Applications	**Neighbor Discovery Monitor**

Table 5.1: Mobility characterization experimental set-up: general simulation parameters and *Neighbor Discovery Service* configuration parameters.

The configuration parameters of the *Neighbor Discovery Service* are shown in Table 5.1. The frequency of the *Beacon Interval* indicates the precision of contact detection. Lower values allow to detect contacts with short duration, higher values may miss some contacts but ensure that the detected ones will will allows proper data transfers. After performing empirical experimentation with changing beacon intervals, we selected a 10 second beacon interval. This value was precise enough to detect most contacts. The *Cache Expiration* defines how how often we receive a value of the network status, we have used a 250 seconds which considering the long duration of the simulation provides several hundred samples which is fairly sufficient. The *Packet size* of the beacons is not really relevant as long as it is small, we use 100 byte packets. As our subscribed application we use the **Neighbor Discovery Monitor**, this application uses the periodic notifications from the *Neighbor Discovery Service* to compute the network metrics described in the following sections.

5.3 Mobility Induced Topological Changes

In this section, we enumerate the several metrics and network characteristics that will be used to analyze the influence of mobility in network topology. Since we are working with a mobile network the changes of topology over time must be taken into account, therefore the values of these metrics are either offered as an aggregate or as a function over time with a selected granularity defining how often is the network sampled to measure the value of the corresponding metric.

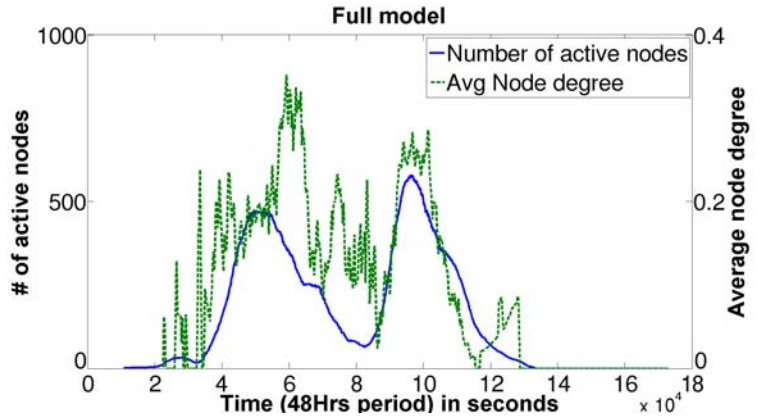


Figure 5.1: Active Node Distribution and superposed Average Node Degree over simulation time (48 hours).

- **Active nodes distribution:** This measures the number of active nodes (nodes inside the disconnected zone) over different intervals of time.
- **Geographical location of encounters:** Provides insight on the geographical distribution of the contacts.
- **Average Node Degree:** Node degree is a measure of the number of neighbors a node has at any time.
- **Number of Clusters:** This is a metric aimed at measuring the degree to which nodes tend to clump together, each cluster can be defined as a smaller network inside the whole network. In a fully connected network all the nodes form a single cluster, in a fully disconnected network, there will be as many clusters as nodes. The average number of clusters provides a rough estimation of how fragmented the network is.

5.3.1 Active Node Distribution

In Figure 5.1, we show the distribution of active nodes over a period of 48 hours according to the Complete model (2,788 nodes). The nodes show two main peaks of activation which is consistent with the usual trend of North Atlantic flights: airplanes start flying west (from Europe to America) during daytime

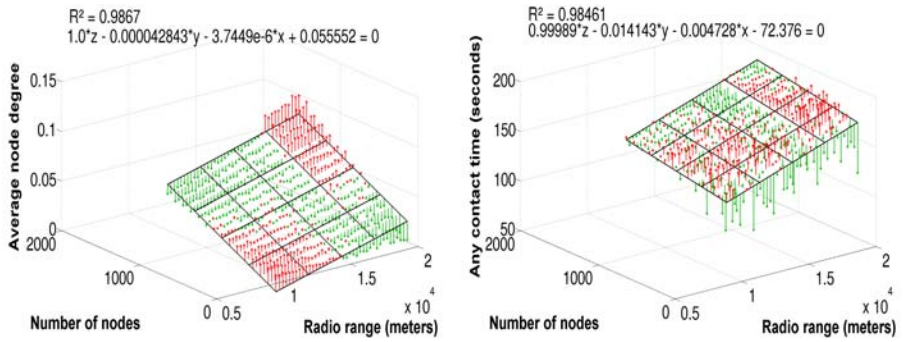
(∼ 06:00 UTC), and east (from America to Europe) during the night (∼ 22:00 UTC). This is a periodic pattern repeated daily, but for our analysis we will focus on a single period, and thus only two peaks of activation can be observed. Another interesting point is that the intervals of activation overlap, an thus there are always some active nodes during the whole period. Another fact that may favor this is that we have noticed some cases of heavily crowded airports, which seem to have scheduled flights for most of the day as opposed to the usual trend.

5.3.2 Average Node Degree

The average node degree gives us an estimation of the degree of connectivity within the network. Following up from the previous section, we relate the active node distribution and the average node degree. In Figure 5.1, we can observe the superimposed distribution of the average node degree over the active node distribution at different time intervals during 48 hours of simulation. We can appreciate that the two main peaks of node activation are being reflected in the node degree. As a result, the node degree increases for intervals with a high number of active nodes. Surprisingly, the central section that has a reduced number of active nodes still shows some degree of connectivity which seems counter-intuitive. The reason for this matter is provided in the next section as a direct result of analyzing the geographical location of aircraft encounters.

The next step of our analysis focuses on determining which external parameters influence the average node degree and to what extent. The most evident observable parameters are the radio range and the number of nodes. In Figure 5.2a, we show the aggregated node degree for the whole 48 hours simulation using the incremental partial model. Each sample point is the average value resulting from 10 simulation runs. The first observable characteristic is that there is a direct dependence between the average node degree and each of the variables. An increase of any of the two variables produces an increase of the average node degree. We fit the data using a plane, the goodness of the fit is $R^2 = 0.98$[1]. Observed samples suggest that for radio ranges the progression follows a linear distribution. In the case number of nodes, the progression seems to be linear for network configurations close to the ones we use, although for

[1] R-squared is a statistical measure used on regression analysis. This measure describes how close the data samples are to the fitted regression line. R-squared value is always between 0 and 1.



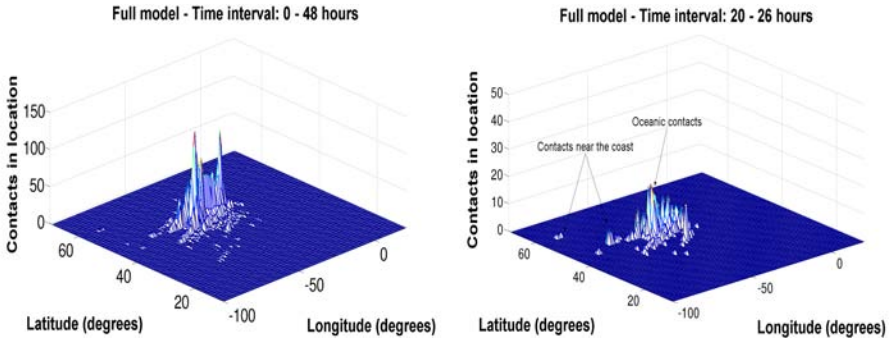
(a) Average Node Degree (b) Any Contact Time

Figure 5.2: Incremental Partial Model: using radio ranges from 10 to 20 km and number of nodes from 100 to 1,500.

higher values (farther away from the ones at our disposal) values show increasingly higher separation from the fitted plane. This seems to indicate a non-linear growth possibly pointing towards an exponential pattern.

These distributions can be appreciated more easily in Figure 5.6, this figure displays the relationship of the average node degree with each variable separately. This is performed for both the number of nodes (Figure 5.6a) and the radio coverage (Figure 5.6b). For the variable that is not being considered at the moment, we use its aggregate value. Lastly, it is worth the mention that the fitted plane equation can be used as a theoretical model to estimate average node degree in function depending on both variables.

Lastly, in Figure 5.9 we show the averaged node degree for the whole 48 hours simulation. This figure uses the Complete Model with radio ranges from 20 to 50 km. Figure 5.9 reveals that the average node degree only reaches one unit (1.07) when the maximum achievable radio ranges are used. This value suggests that the nodes display encounters on a periodic basis, although not with multiple aircraft simultaneously. Additionally, this figure also confirms the previously observed linear relationship between the average node degree and the radio range.



(a) Time interval 0 to 48 hours.

(b) Time interval 20 to 26 hours.

Figure 5.3: Geographical distribution of encounters.

5.3.3 Geographical location of encounters

The specific location of contacts is another key aspect that needs to be analyzed to properly understand the several implications of node mobility. In Figure 5.3a, we consider the geographical distribution of contacts through the whole simulation. This figure shows that most contacts are produced around the area close to (40°N, 50°W), the middle of the Atlantic ocean but slightly towards the American continent.

In the previous section, we pointed the oddity shown by the central section of Figure 5.1. This figure displayed a low number of active nodes but contained a fairly high degree of connectivity. We will now observe this specific time interval (see Figure 5.3b). This figure shows that there are contacts produced both in the oceanic area and near the coast. When the direction of the flight is considered, we can notice contacts between planes flying eastbound and planes flying westbound.

Contacts produced between aircraft flying in opposite directions points towards possible encounters between the last stragglers from Europe and the first planes departing from America. Note that the encounters displayed on Figure 5.3, are only produced on the American side of the ocean. This situation occurs because the simulation does not contain the batch of aircraft departing from Europe on the second day. In Figure 5.4, we consider a longer time interval

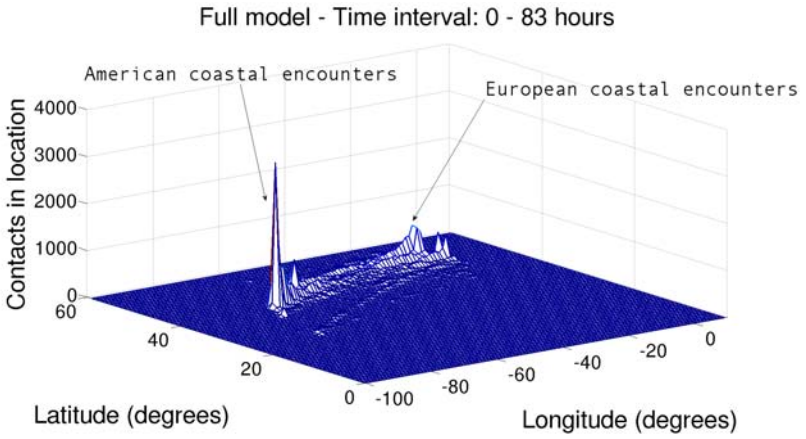


Figure 5.4: Geographical distribution of encounters (time interval 0 to 72 hours). Includes a second wave of departing flights.

and include this second wave of departing flights. We can now observe that the same kind of contacts is also produced on the European coastal side.

5.3.4 Number of Clusters

In this section, we observe how the aircraft interact with each other and group together into communities. We start by analyzing how the network becomes more inter-connected as the number of nodes and the wireless range increases.

In Figure 5.5, we show the aggregated number of clusters for the whole 48 hours simulation using the incremental partial model. As seen in previous sections, the number of active nodes varies through the simulation (and thus the maximum number of possible clusters). A simple aggregate count of the number of clusters would not be a representative measure of network fragmentation. Therefore, the plotted data is normalized using the expression 5.1 which represents the number of clusters as an interval between 0 and 1. The closer the values to 1 the more fragmented the network would be (independently of the number of active nodes).

$$Norm.\#clusters = (\frac{\#clusters}{max.clusters}) \tag{5.1}$$

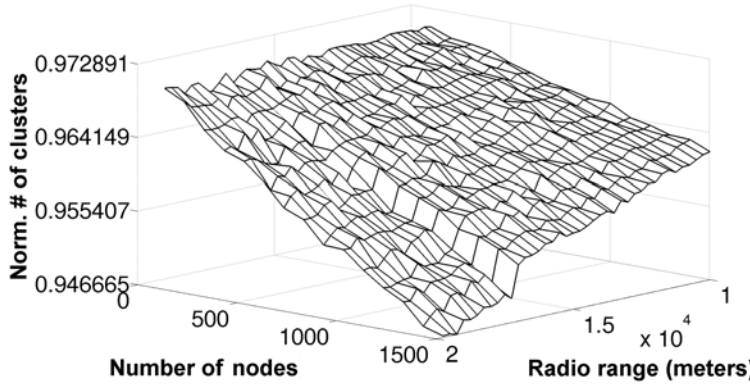


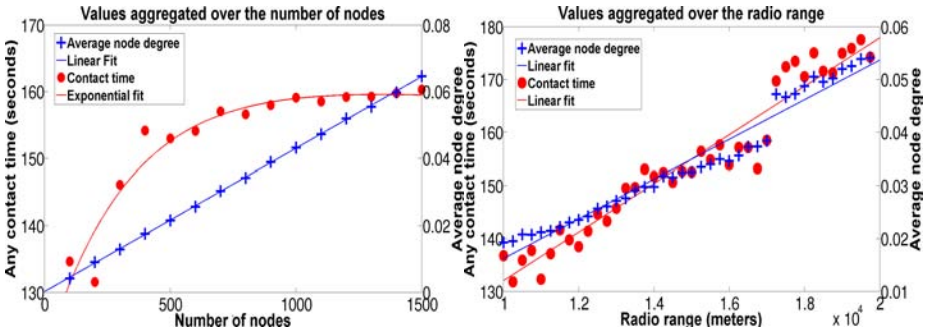
Figure 5.5: Normalized number of clusters: Incremental Partial Model with radio ranges from 10 to 20 km and number of nodes from 100 to 1,500.

It can be observed that the number of clusters decreases linearly as the wireless range or the number of nodes increases. Increasing any of the two variables improves network connectivity by reducing the number of isolated nodes that become part of a bigger community of nodes. These results coincide with those displayed by the node degree as both metrics offer similar information.

Analyzing the complete model shows that the average number of clusters through the simulation is 185.399 of a total average of 203.9. Therefore, roughly 10% of active nodes are connected between them to some degree, while the rest remains isolated. Further analysis of cluster characteristics, composition, and quantity is provided in the topological analysis from Chapter 6.

5.4 Mobile Contact Properties

In this section, we enumerate the several metrics that describe the properties of encounters between nodes. Most of the metrics regarding mobile contacts are based on the event of a concrete pair of nodes A, B encountering each other in repeated occasions. In our network scenario, the mobility patterns of the nodes turn repeated encounters into a highly unlikely event. Therefore, we will consider some of the alternate metrics presented in [51], which are based on encounters with *any* pair of nodes.



(a) Aggregation over the number of nodes (b) Aggregation over the radio range

Figure 5.6: Incremental Partial Model: Average Node Degree and ACT aggregated values

- ***Any* Contact Time:** This is the expected duration of a meeting between any two nodes of the network. We compute it as the elapsed time between the first and the last consecutive messages received during a contact situation. This metric is an interesting measure because it allows to measure how much data can be transferred in each encounter.
- **Inter *Any* Contact Time:** This metric measures the elapsed time between two consecutive meetings of any pair of nodes. As such, it offers insight on the frequency of communication opportunities.

5.4.1 Any Contact Time (ACT)

The Any Contact Time (ACT) gives us an estimation of the duration of contact opportunities. We can intuitively guess that this duration will be directly influenced by the radio range, longer ranges will guarantee longer intervals of connectivity. We start our analysis by determining to what extent this parameter influences the ACT. Similarly, we also evaluate how the number of nodes impacts this metric.

In Figure 5.2b, we show the aggregated any contact time for the whole 48 hour simulation using the incremental partial model. Each sample point is the average value resulting from 10 simulation runs. We can observe that ACT

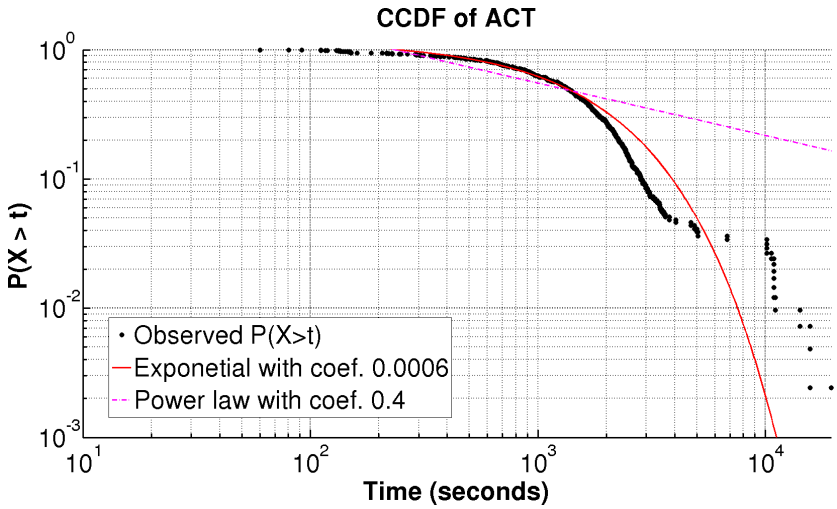


Figure 5.7: CCDF of the Any Contact Time

increases linearly along with the radio range. It is appreciable that for small radio ranges the ACT is non-existing or has a very low value. This event happens either because the nodes are not close enough to produce a contact, or because its duration is below the detection threshold (10 seconds, see section 5.2). It can also be seen that a higher concentration of nodes attenuates this issue by providing more chances for contact opportunities. Overall we can observe that the number of nodes also increases the ACT linearly.

We study this issue further in Figure 5.6, this figure displays the relationship between the ACT and each variable separately. As can be seen in Fig. 5.6a the node density offers an exponential tendency, and for very low number of nodes the contact duration is nonexistent. Those reduced samples of nodes may be geographically dispersed, and their routes may not let them approach each other resulting in a lack of contacts. There is a rapid grow as the number of nodes increases, and the higher density generates some contacts. Finally, it stabilizes and shows a constant tendency. Therefore, we can say that the number of nodes has little or no effect on the duration of the communications. In Fig. 5.6b we can observe the impact of the radio range. The distribution of this variable increases linearly, from which we can say that the radio range directly influences contact

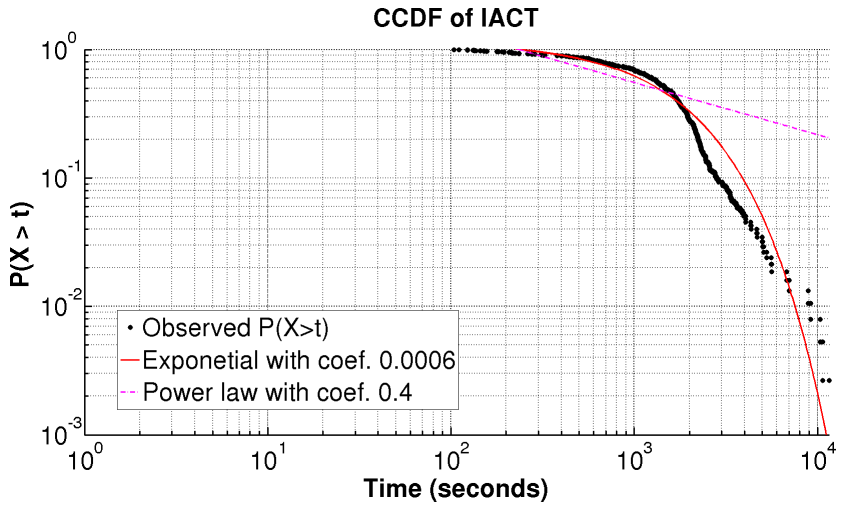


Figure 5.8: CCDF of the Inter Any Contact Time.

duration.

In Figure 5.7 we show the complementary cumulative distribution function (CCDF) of the any contact time for the Complete model. 50% of contacts last less than 1,000 s, about 30% last more than 2,000 s, and 10% of the samples last more than 3,000 s. The average ACT of the samples is roughly 1,400 s. We compare the distribution using both a power-law and an exponential. The distribution is not heavy tailed and is closer to the exponential distribution. We get an approximation using a expression of the form $P(X > t) = e^{-\lambda t}$ with coefficient 0.0006.

Lastly, In Figure 5.9 we show the ACT for the whole 48 hours simulation. This figure uses the Complete Model and analyzes radio ranges from 20 to 50 km. Figure 5.9 reveals that the ACT increases along with the radio range up to ∼1,400 seconds for the maximum achievable radio ranges. This figure additionally confirms the linear relationship between radio range and the ACT.

5.4.2 Inter Any Contact Time (IACT)

The Inter Any Contact Time (IACT) provides insight on the frequency between contact opportunities. Analysis of the distribution of IACT between mobile

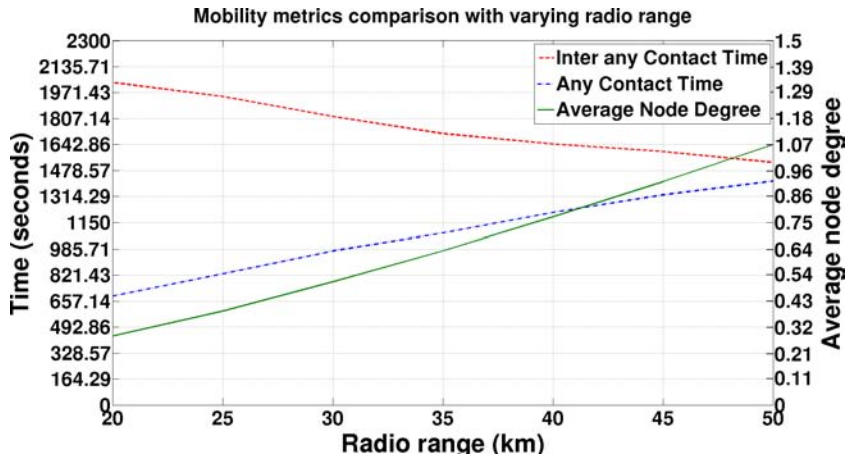


Figure 5.9: Mobility Characterization: Aggregated metric measurements using the Complete Model with radio ranges from 20 to 50 km.

devices is one of the key aspects of the design of synthetic mobility models and routing protocols for opportunistic networks.

In Figure 5.8 we show the CCDF of the inter any contact time for the Complete model. The average inter-any-contact time for the full model is roughly 1,500 seconds. We can see the model offers large inter-contact-times, all of them are bigger than 100 seconds, 70% exceed 1,000 seconds, and 10% exceed 3,000 seconds. The CCDF shows a clear exponential decay with a suggested power-law behavior up to a characteristic time of 1,000 seconds. The tail of the distribution is approximated using an expression of the form $P(X > t) = e^{-\lambda t}$ with coefficient 0.0006.

Finally, In Figure 5.9 we show the IACT for the whole 48 hours simulation using the Complete model with radio ranges from 20 to 50 km. This figure shows that there is an inversely proportional dependency between radio range and inter-contact-time. This is a fairly intuitive result since the radio range increases the ACT linearly. Therefore, as long as the mobility pattern is unchanged, if the duration of the contacts increases the time between those encounters should decrease. Therefore, the IACT decreases linearly and eventually becomes lower than the ACT.

5.5 Summary and Conclusions

In this chapter, we analyzed multiple aspects related to network mobility. We found that the number of active nodes during the simulation changes over time and displays two peaks of high activity. The network is sparsely connected displaying average node degree below one node for all configurations but the ones with the highest radio ranges. The encounters between nodes are produced deep into the ocean and also in some coastal, those encounters are usually produced by planes flying in the same direction but also in opposite directions for the coastal areas. This kind of movement patterns may favor the use of forwarding protocols based on some time-to-destination measure (such as the expected landing time (ELT)). Such protocols should focus on delivering data to nodes that will end their route sooner than others.

The analysis of the ACT allows us to estimate the quantity of data that can be transferred during each encounter. The ACT is mostly dependent on the radio range, and different configurations show values that range from 400 seconds to a thousand seconds. Such values allow the transmission of a few hundred Megabytes during each encounter.

The IACT is widely used as a metric for the prediction of encounters in routing protocols. Our results show that our network IACT displays a heavy-tailed distribution with exponential decay. This is a common trait found in multiple opportunistic and vehicular networks. This trait is known as the *power-law exponential dichotomy*, it was first observed in [88]. The fact that the IACT is exponentially bounded is a beneficial property that allows the design of protocols assuming no infinite delays.

Additionally, we can use the equation of the fitted plane from Figure 5.2a as a theoretical model for computing the average node degree for any desired set of parameters. In this way, we estimate that a fully connected network can only be attained using radio ranges of several hundred kilometers. This result is consistent with studies found in the literature and described in Chapter 2.

From the results of our analysis, we conclude that even with the use of an opportunistic approach it is very challenging to achieve successful communications in this kind of environment. Therefore, the most adequate configuration is the one with the highest achievable radio range 50 km (as defined in Chapter 4.1). The several implications and results of this mobility characterization will be used in the deployment of opportunistic communications and configuration of opportunistic routing protocols described in Chapter 7.

A child['s]... first geometrical discoveries are topological... If you ask him to copy a square or a triangle, he draws a closed circle.

Jean Piaget

6

Network Topology Characterization

IN this chapter, we will analyze the topological structure of our network scenario. In a mobile opportunistic network such as ours, nodes create the network by interacting with each other and forming changing communities. Some of the complex characteristics of the mobility patterns may be difficult to understand by only analyzing individual pairwise contacts. A clear example of these issues may be found in some specific types of opportunistic network, such as *pocket-switched networks* and non-scheduled vehicular networks (cars, cabs, etc.). In these networks, mobility is driven by the social activities of human beings, therefore the pattern is usually not obvious. An alternative analytical method has been widely to solve these issues, the *social network graph*. This tool is used to characterize the structure of the graph, locate the most relevant nodes, or to identify communities within the whole network.

This chapter applies analysis of the *social network graph* to perform a topological characterization of the network. Several metrics are presented to measure the structural aspects of the network, the relevant nodes, and its communities. We conclude the chapter with a detailed discussion of the experimental results.

Simulation Parameter	**Value**
Radio Communication Range	50 km
Simulated Time	48 hours
Number of Nodes	2,878
Neighbor Discovery Service	
Beacon Interval	10 Seconds
Cache Expiration	250 Seconds
Packet Size	100 bytes
Subscribed Applications	**Contact Graph Builder**

Table 6.1: Topology characterization experimental set-up : general simulation parameters and *Neighbor Discovery Service* configuration parameters.

6.1 Experimental Configuration

In this section, we describe the experimental set-up used to perform the several measurements and experiments. It is composed of the following protocols and models:

- Mobility Model described in chapter 3.
- Network Layers 1 to 4 described in chapter 4.
- *Static Routing* with predefined routes (described in chapter 4.3.2).
- *Neighbor Discovery Service* application service (described in chapter 4.5.1)

The configuration parameters of the *Neighbor Discovery Service* are shown in Table 6.1. The configuration and rationale for the values of: *Beacon Interval*, *Cache Expiration*, and *Packet size* is analogous to that of chapter 5. As our subscribed application we use the **Contact Graph Builder**, this application uses the periodic notifications from the *Neighbor Discovery Service* to build a contact based social graph whose characteristics are described in the following section.

6.2 Graph-based Network Representation

In this section, we define the analytical method used to analyze the opportunistic network topology. This method establishes a social graph using the contacts

of the network. This graph is called a *Contact Graph* and it displays the relationships between nodes and provides detailed insight on the roles of each node within the network. Applying several graph topological analysis techniques and metrics to the contact graph provides information about the structure of the network. This sort of study is often called *Social Network Analysis*.

6.2.1 Building the Contact Graph

A contact-based social graph is a graph-based representation of the opportunistic network, where nodes correspond to the network entities, and edges represent their relationships, in our case a contact.

The contact graph is defined as: an undirected labeled dynamic graph $G(N, E)$. Each of the aircraft i is a node of the graph, $n_i \in N$, and the edge $e_{i,j}$ represents that nodes i and j had a contact between them. Contacts between the same pair of nodes happen only once, therefore, $e_{i,j}$ has no assigned weight. Each $n_i \in N$ has a limited lifetime represented by a label $n_i^{(start,end)}$ depicting the time interval in which this node is part of the graph (active node). Each $e_{i,j}$ is also labeled with the time offset of the contact. Additional labels are included to provide geographical and trajectory information. The inclusion of lifetime information provides a way to analyze the network topology at specific instants of time. Figure 6.1 shows the aggregated contact graph considering all the contacts produced during the full time interval (0 to 48 hours).

6.2.2 Metrics and Concepts

We study the properties of the contact graph and examine the relationship between the nodes. For this study, we consider the following concepts and metrics :

- **Network Structure:** are a set of measures that describe characteristics of the network from a general perspective.
  - **Connected Components:** of an undirected graph is a subgraph in which any two nodes are connected by a path, and has no connection to nodes in the supergraph.
  - **Network Diameter** and **Average Path Length**: represent the longest and average (respectively) of all the shortest paths between nodes in a graph.

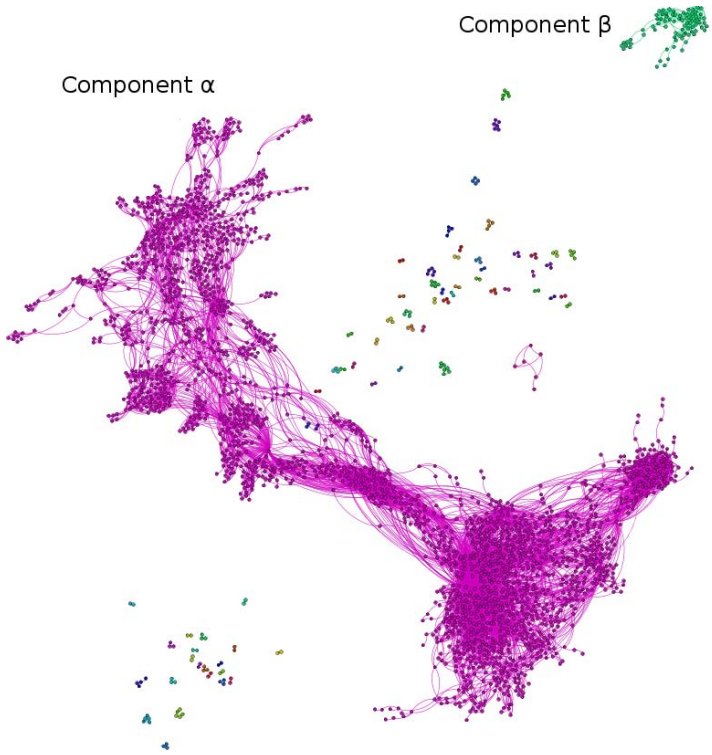


Figure 6.1: Contact Graph: nodes are grouped into connected components with uniquely identifying colors.

  - **Modularity [89]:** is a standard quality function that quantifies the quality of a given division for nodes within the network. Networks displaying high modularity have dense connections between nodes in the same module but sparse connections with nodes in different modules.

- **Centrality metrics [57]:** are a set of metrics that measure the structural importance of a node in the network.

  - **Degree centrality :** is measured as the number of direct edges that reach a given node.

  - **Betweenness centrality [90]:** is equal to the number of shortest paths linking other nodes that pass through the node being measured.
  - **Closeness:** is based on the length of the shortest path linking to all other nodes.

- **Brownian distance covariance [91]** (or distance correlation): is a measure of statistical dependence between two random variables. A value of zero implies statistical independence.

6.3 Network Structure

In this section, we analyze some structural characteristics of the contact graph. For this structural analysis, we use the static version of the contact graph. This graph is constructed by disregarding the time component of the dynamic graph, and aggregating all nodes and edges belonging to the graph at any interval of time.

6.3.1 Connected Components

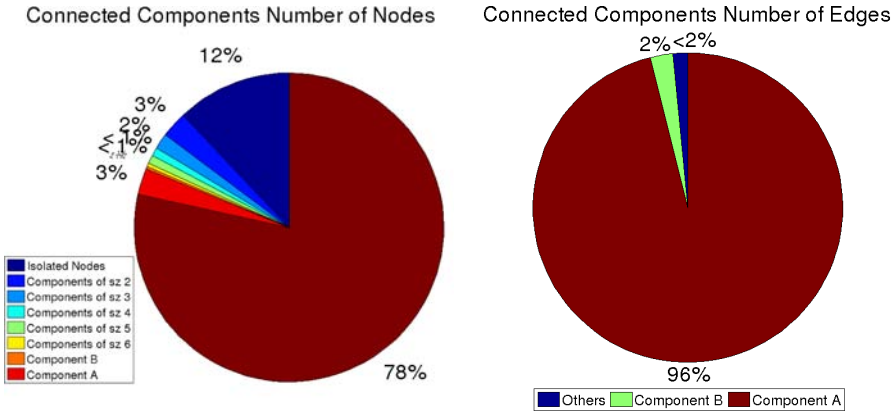
The number and composition of the connected components is an important characteristic of the contact graph. Analysis of this metric provides an estimation of how well connected is the network. Furthermore, it can be used to locate locate nodes with limited or inexistent communication capabilities.

We start our analysis by determining the number of connected components of the graph. In Figure 6.1, we show the graph representation of the network. The graph has 2,878 nodes and 9,530 edges representing the contacts between nodes. This Figure shows the distribution of nodes into connected components. Connected components appear as groups of nodes with a defining unique color. There is a total of 428 connected components, and 358 of those correspond to isolated nodes without any contact opportunity. The most distinguishing features are the two biggest connected components: one composed of 2,249 nodes (Component α) and another of 78 nodes (Component β).

We will now analyze the internal structure of the connected components. In Figure 6.2, we show the composition of the percentages of nodes and edges corresponding to each type of component. We can observe that that the isolated nodes represent the 12% of the nodes of the network. The connected components of size smaller than 10 represent a 7.7 % of the network, but only a 1.43 % of

the edges (contacts). Component β represents the 3% of the network and 2% of the edges. Finally, Component α represents most of the network with a 78% of nodes and 96% of edges.

Nodes pertaining to independent connected components are not able to communicate with each other. The most extreme case being the *one-node* components corresponding to isolated nodes, these nodes are completely incapable of performing any sort of opportunistic communications. Additionally, connected components with few edges have little relevance due to contacts being scarce. This turns Components α and β into the most relevant components both by the percentage of nodes they contain and the amount of communication opportunities they provide. Therefore, they will become our focus for further study.



(a) Distribution of nodes

(b) Distribution of edges

Figure 6.2: Circle graph displaying proportion of nodes and edges pertaining to connected components of specific sizes.

6.3.2 Network Diameter and Average Path Length

In this section, we analyze two statistic measures referring to the shortest paths between any pair of nodes in a connected component. First, the network diameter, this statistic will be used to determine the maximum number of hops required to communicate any two nodes. Secondly, the average path length

Component Size	Component count	Network Diameter	Avg. path Len.
2	39	1	1
3	16	1.777	1.255
4	6	2.118	1.406
5	5	2.5	1.5
6	3	2.571	1.598
78 (β)	1	9	3.531
2249 (α)	1	18	6.024
Full Network	71 (+358 isolated)	18	6.021

Table 6.2: Network Structure: averaged metrics for each connected component class (defined by size).

which instead will provide an averaged estimate of the number of hops.

In Table 6.2, we display the number components classified by class (number of nodes or size), the second column (*Component Count*) represents the number of connected components of that size. Lastly, the values displayed in *Network Diameter* and *Average Path Length* are averages. For component α, we observe a network diameter of 18 and average path length of 6.024. For Component β, a diameter of 9 and average path length of 3.531. For components of other sizes network diameter oscillates between 1 and 2.571, while the average path length ranges from 1 to 1.598. Analyzing the statistics considering the whole network displays nearly identical results than those of component α, it is clear that this component is the most influential. Therefore, future sections will mostly focus on analyzing this component.

6.3.3 Modularity

In this section, we study the internal structure of each connected component. In a connected component, all nodes are connected to each other by nodes in a seemingly unique uniform structure. Despite of that, it is still possible to further infer new substructures based on how strongly are nodes connected with each other. This property is called *Modularity*.

Modularity can be inferred using the Louvain ([89]) community detection algorithm which is based on *modularity* maximization. This method uses an iterative approach to optimize local communities until the modularity value of the global graph can no longer be improved. The main advantage of this method is the low computational complexity and good performance times in respect to

other algorithms (as shown in benchmark ([92]).

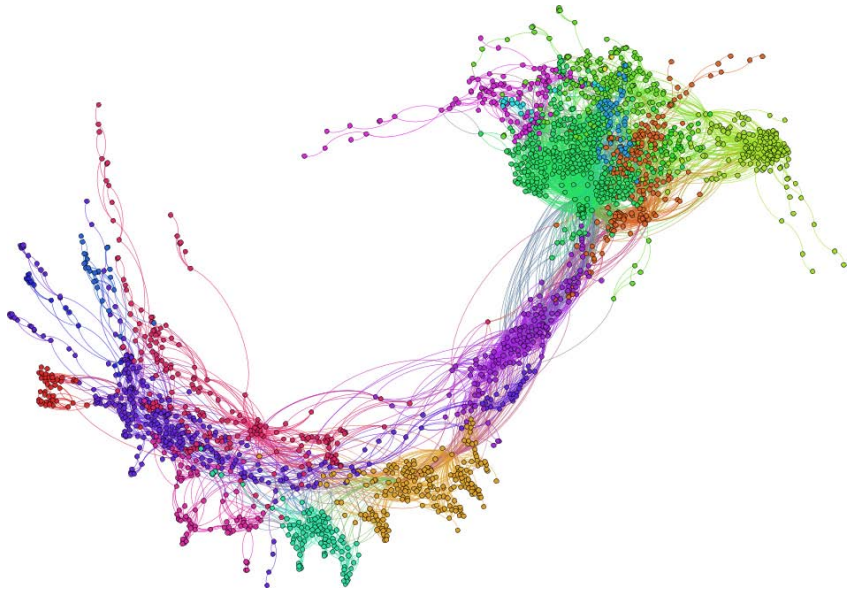
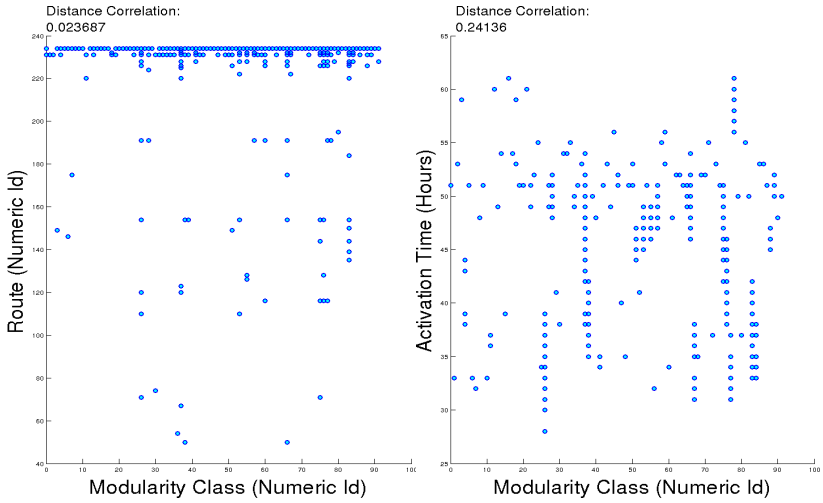


Figure 6.3: Contact Graph: Component α divided into clusters with uniquely identifying colors.

In Figure 6.3, we show component α divided into communities based on modularity (clusters). We can observe that the nodes inside component α are grouped into 19 distinct clusters (each one represented by a different color). The maximized modularity value achieved by this classification is 0.75, this is a high modularity value indicating a clear fragmentation of the network into smaller communities. Nodes in the same modularity class have good chances of successful communication due to the higher number of interconnecting paths between them.

Consequently, we now proceed to analyze which characteristics cause one node to become part of a specific community. In Figure 6.4, we analyze the node distribution between clusters to infer possible relationships. The graph is built from contacts, contacts are influenced by mobility, and mobility is defined by two key aspects: geographical position and time. Figure 6.4 shows the distribution of clusters based on node geographical routes and activation times,



(a) Distribution by flight route (b) Distribution by nodes activation time

Figure 6.4: Modularity: distribution of nodes with specific properties among clusters.

we use a Brownian distance correlation metric to infer non-linear relationships.

First, in Figure 6.4a we observe that the distribution of routes between clusters is fairly homogeneous as there are planes from the same route distributed among many clusters, this is also pointed by the low distance correlation coefficient (0.02). In contrast, in Figure 6.4b we can observe a strong relationship (distance correlation of 0.24) between the assigned cluster and the activation time of the node, this is an intuitive result as the graph is constructed from contacts, and only planes with overlapping activation times will have the chance to contact each other.

6.4 Network Centrality

In this section, we locate the most influential nodes within a network graph, and for this purpose we use the centrality metrics defined in section 6.2.2. Central nodes display interesting characteristics that make them good candidates to

perform special roles within the network.

6.4.1 Degree Centrality

In this section, we locate the most influential nodes according to degree centrality. This metric is conceptually simple as it just measures the number of incident edges upon a node. Degree centrality for a given node n_i is measured as the expression 6.1 where $e(n_i, n_k) = 1$ if a direct edge exists between n_i and n_k and i $\neq$ k.

$$C_D(n_i) = \sum_{k=1}^{N} e(n_i, n_k) \tag{6.1}$$

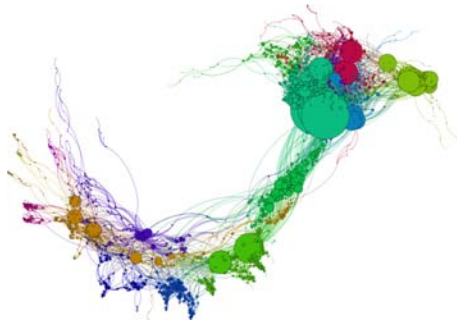
In Figure 6.5a we show the nodes of component α with different sizes. Node sizes follows a scale from 1 to 200, and represents values from 1 to 214. This figure gives an overview of the most influential nodes within the network. The structural position of these nodes maximizes the chances of acquiring any data flowing through the network. Nodes are distributed according to the following proportions: 45% of the nodes have a degree between the range [1-5], 35.97% [6-10], 19.03% [11-100], and 0.31% [100-214].

6.4.2 Closeness Centrality

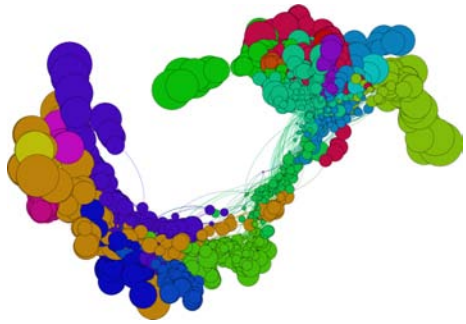
In this section, we locate the most influential nodes according to closeness centrality. This metric measures the distance of a node to all other nodes. The distance between nodes i and k is defined as $d(n_i, n_k)$ and it corresponds to the shortest path linking them. Closeness centrality for a given node is the reciprocal of its mean distance to all other nodes, it is computed as in expression 6.2 where N is the number of nodes in the network and i $\neq$ k.

$$C_C(n_i) = \frac{N-1}{\sum_{k=1}^{N} d(n_i, n_k)} \tag{6.2}$$

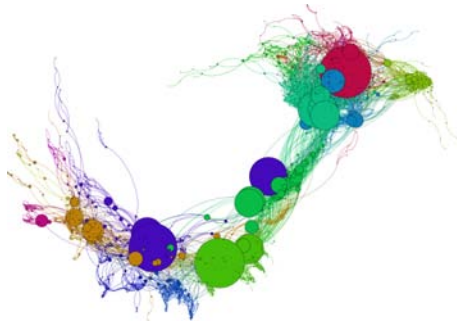
In Figure 6.5b we show the nodes of component α with different sizes. Node sizes follows a scale from 1 to 200, and represents values from 4.08 to 10.7. This figure gives an overview of the most influential nodes within the network. The structural position of these nodes maximizes the spread of information originating from them. Nodes are distributed according to the following proportions:



(a) Degree Centrality



(b) Closeness Centrality



(c) Betweenness Centrality

Figure 6.5: Contact Graph: Component α depicting nodes with variable size depending on their relevance according to different centrality measures.

55.45% of the nodes have a closeness between the range [4-6), 39.97% (6-8), and 4.58% [8-10.7].

6.4.3 Betweenness Centrality

In this section, we locate the most influential nodes according to betweenness centrality. This metric measures the number of shortest paths passing through this node. Betweenness centrality for a certain node is computed using the expression 6.3 where g_{jk} is the total number of shortest paths linking nodes j and k, and $g_{jk}(n_i)$ is the number of paths that include node i.

$$C_B(n_i) = \sum_{j=1}^{N} \sum_{k=1}^{j-1} \frac{g_{jk}(n_i)}{g_{jk}} \tag{6.3}$$

In Figure 6.5c we show the nodes of component α with different sizes. Node sizes follows a scale from 1 to 200, and represents values from 0 to 377744. This figure gives an overview of the most influential nodes within the network. The structural position of these nodes can improve interaction between the nodes that it links together. Nodes are distributed according to the following proportions: 54.2% of the nodes have a closeness between the range [0-1500), 25.7% (1500-5000), 16.67% (5000-25000), and 3.42% [25000-377744].

6.5 Summary and Conclusions

In this chapter, we analyzed multiple topological aspects of our network scenario. The network is represented graph-based representation constructed from the aggregation of pair-wise contacts. Multiple common techniques common on social network analysis are then applied to extract structural information, and infer the reason behind such structure. We will now proceed to provide the conclusions and implications resulting from analyzing the results of each applied metric.

The classification into connected components provides insight on the structure and communication capacities of the network, and more importantly on the number of isolated nodes. Isolated nodes require separated treatment due to their complete inability to perform opportunistic communications. Additionally, it also uncovers the inability of communication between nodes in separate components. The network diameter provides insight on the maximum number

of hops of the network, this information can be used for optimal configuration of routing protocols.

The study of modularity inside of connected components allows provides more information on the nature of the contacts. The homogeneous distribution of aircraft following different routes among the clusters favors that any improvement on the network connectivity of a given cluster will affect a wide geographical area. On the other hand, the temporal dependency uncovers the necessity to take into account the lifetime of nodes for the design of possible delay tolerant applications or protocols where data survivability is a necessity.

The study of centrality allows to locate nodes with interesting characteristics that make them good candidates to perform special roles within the network. Some examples of such roles could be: nodes with extended buffer capacity acting as caching services or nodes with dedicated data links. As a result, nodes with special characteristics can be used as a mean to improve network connectivity and performance.

The several implications of this topological analysis are extensively used in the design of a specialized network architecture which will be described in Chapter 8.

Part III
Network Evaluation

Never underestimate the bandwidth of a station wagon full of tapes hurtling down the highway.

Andrew Tanenbaum.

1989
Computer Networks.
New Jersey: Prentice-Hall. p. 57.

7

Opportunistic Networking Performance Evaluation

IN this chapter, we analyze how data propagates inside the proposed transoceanic aircraft network scenario using opportunistic communications. We evaluate the network performance and the feasibility of using multi-hop opportunistic communications as a reliable way to deliver on-flight generated data (either from passengers or company) to the ground. One of the key aspects of successful data delivery relies on the appropriate selection of a forwarding strategy. In this chapter, we use two specialized opportunistic routing protocols and analyze their impact on the network. We study the information flow resulting from these routing strategies and analyze its properties through several well-known network performance metrics. Additionally, we compare the performance of both routing strategies to determine which one provides better results. We close the chapter with a discussion of the obtained results and enumerate the main limitations encountered.

7.1 Motivation

In this chapter, we analyze how the network behaves using multi-hop opportunistic communications. The previous characterization of the network properties revealed that this a sparsely connected network composed of several independent node communities (see Chapters 5 and 6). As a result, we will now analyze the impact of this network structure, and check whether opportunistic communications are capable of overcoming these challenges and provide a reliable alternative for data transmission.

The scarce connectivity of the network makes the use of every contact opportunity a priority. Therefore, we have chosen flooding-based schemes as our routing protocols. This sort of routing will allow to maximize the delivery ratio and minimize delivery time by exploring all the paths available. The main aspects we seek to evaluate are related to the flow of data thorough the nodes. Concretely, we want to check whether the data can reach the ground, and estimate the time required and the amount of data lost in the process.

7.2 Experimental Configuration

In this section, we describe the experimental set-up used to perform the several measurements and experiments. It is composed of the following protocols and models:

- Mobility Model described in chapter 3.
- Network Layers 1 to 4 described in chapter 4.
- We use two different routing protocols (described in chapter 4.3.2):
  - *Epidemic Routing.*
  - *Constrained Epidemic Routing based on ELT estimation.*
- *Constant Bit Rate* traffic generators (described in chapter 4.5.1)

The configuration parameters of the routing protocols are shown in Table 7.1. Both versions of the routing protocol share the same attributes and configuration parameters. Most parameters are fine-tuned using information about the network characteristics to properly fit the purpose of the experiments.

Simulation Parameter	**Value**
Radio Communication Range	20-50 km
Simulated Time	48 hours
Number of Nodes	2,878
Routing Protocols	Epidemic/ELT
Beacon Interval	400 seconds
Buffer Capacity	20-100 packets
Max. Number of hops	40 hops
Expiration Time	14,440 seconds
Traffic Generators	CBR
Packet Size	512 bytes
Traffic per node	10-35 kbytes
Bit Rate	256 bytes/second

Table 7.1: Information propagation experimental set-up: general simulation, routing protocols, and CBR configuration parameters.

The frequency of the *Beacon Interval* indicates the precision of contact detection. Lower values allow to detect contacts with short duration, higher values may miss some contacts. In Chapter 5, we analyzed the distribution of the IACT of the nodes of the scenario. This value describes the time lapse between contacts of any two nodes on the scenario, and the beacon interval needs to be smaller than the IACT to ensure that no contact opportunity is missed. As a rule of thumb, we set the *Beacon Interval* to roughly a third of the average IACT, which for a radio range of 50 km is approximately 400 seconds.

In opportunistic networks such as this one, the delivery delays can span to several hours. We need to guarantee that packets are stored long enough to be delivered at least by some of the nodes carrying a copy of them. As such, our next step is to study how long does a node remain in the disconnected area. This value is known as the **disconnected flight duration** (previously defined in Chapter 3). This value gives us the higher bound for the worst-case data delivery, which is when the aircraft has to perform direct delivery upon leaving the disconnected area. In Figure 7.1, we show the cumulative distribution function (CDF) of the disconnected flight duration for all the nodes of the scenario. We use its mean value, approximately 14,400 seconds, as the packet *Expiration time*. This ensures an appropriate packet expiration drop

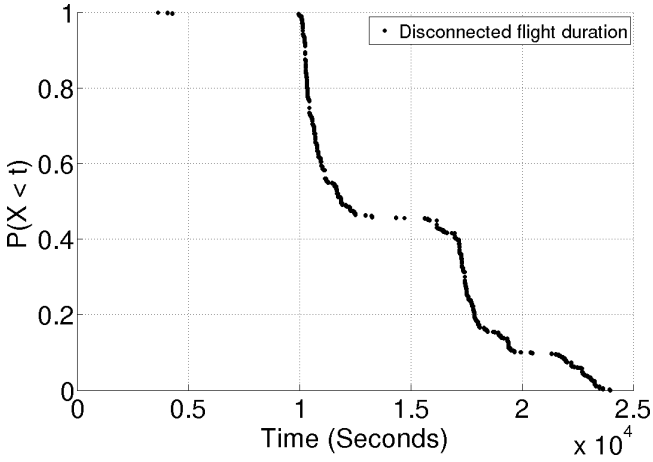


Figure 7.1: CCDF of the disconnected flight duration

ratio for the characteristics of this scenario.

The maximum *Number of Hops* is set to 40, which is the highest packet replication value observed using a model without buffer size or expiration limitations (this is shown in Figure 7.2). The rationale for this value follows a rather conservative approach which assumes that the packet is only replicated once in every contact (usually lower values would suffice).

The characteristics of the On/Off CBR traffic generator are defined by the *Packet Size* which is set to 512 bytes and the *Bit Rate* which is set to 256 bytes/second. The traffic generators become active as soon as the airplane enters the **disconnected zone** and they are disabled when the data generated reaches a maximum of 10 KBytes. This pattern ensures a small burst of data coming from each plane upon entering the disconnected zone. The burst contains enough packet redundancy to ensure that some will reach the ground.

Finally, the *Buffer Size* of the routing protocol is set accordingly 100 packets which guarantee the capability to carry data originating from 10 other nodes.

The analytical objectives of this chapter can be achieved using low volumes of data traffic, which is a favorable feature considering the nature of the routing protocols. As a result, traffic generators are configured to generate small amounts of data per node. The low traffic volume allows to reduce the storage requirements of the nodes and reduces simulation overhead.

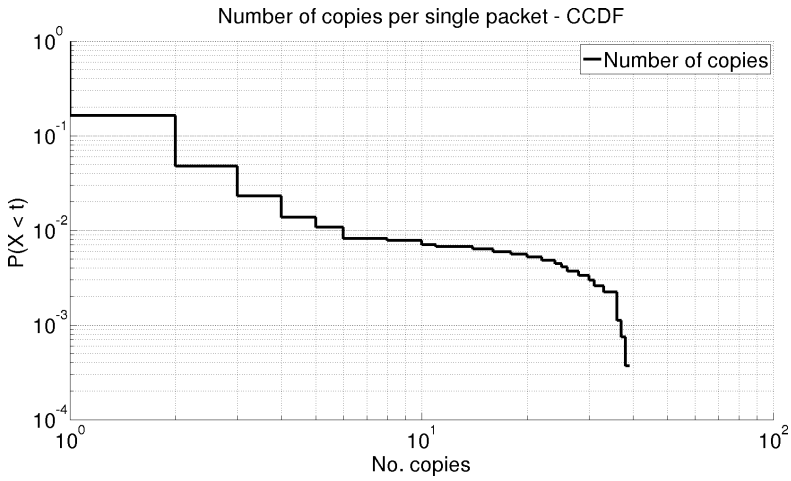


Figure 7.2: CCDF of the number of copies per packet (model with no buffer limitations).

7.3 Network Performance

In this section, we study the network capabilities and performance of the network scenario using opportunistic communications. For this study, we consider three well-known performance metrics:

- **Delivery ratio:** This is the ratio of successfully delivered packets to the total number of packets generated in the network at the end of the simulation. The delivery ratio may be lower than one because of packet drops caused by either insufficient buffer capacity, packet expiration or by exceeding the allowed number of hops.

- **End-to-end delay:** This is defined as the time required for a packet to reach its destination. It is calculated by accumulating the delay of each hop. We perform this computation at network layer and only for packets successfully delivered.

- **Network traffic per packet:** This refers to the average network traffic per packet, and it is calculated by dividing the number of forwarded packets by the number of generated packets.

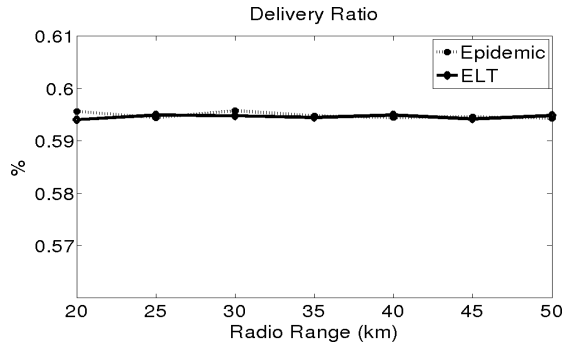
In the following sections, we describe the evaluation of the performance of both routing protocols under changing conditions. Concretely, we modified specific configuration parameters (such as: radio range, traffic demand, and buffer size) and observed their influence in network performance.

7.3.1 Delivery Ratio

The delivery ratio gives us an estimation of the amount of generated packets that reach the ground. We analyze the several parameters that can affect the delivery ratio. First, the most obvious one is the size of the buffers. In Figure 7.3c, we show the progression of the delivery ratio for varying buffer sizes, ranging from 20 to 70 packets. We can observe that the delivery ratio increases along with the size of the buffer, which is a straightforward result. For small buffers, the number of drops due to congestion is higher, and as long as the size is increased, the number of discarded packets decrease.

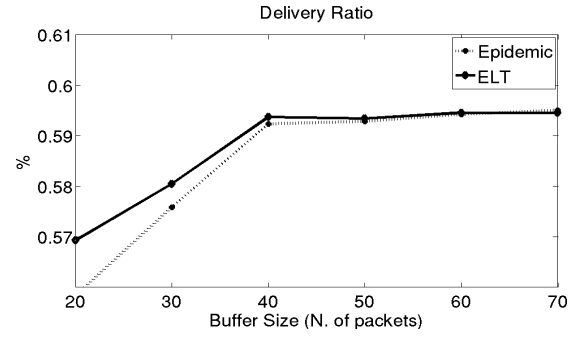
The next parameter we analyze is the influence of the volume of traffic generated by each node. In Figure 7.3b, we show the delivery ratio for changing values of traffic demand, ranging from 2 to 35 KB. It is observable that higher values of traffic decrease the delivery ratio. For fixed buffer sizes, an increase of traffic volume eventually leads to packet drops due to filled buffers. This fact is shown in Figure 7.3b, where we can see mostly constant values up to 25 KB, at which point the sudden drop in delivery ratio becomes clearly visible.

Finally, we analyze the impact of the radio range of each node. In Figure Fig. 7.3a, we show the distribution of the delivery ratio for variable radio ranges 20 to 50 km. We can observe a mostly constant tendency for the considered values. This trend shows that increasing the radio range does not offer any noticeable improvement as it does not provide new paths for data delivery. In this case, it is worth to mention that the network connectivity is still very low even on the highest configuration (50 km). The contacts are heavily influenced by the trajectories of the aircraft, and are usually produced between aircraft that have intersecting trajectories or sporadic approximations. Our conclusion, is that although higher radio ranges may provide new contacts, they are produced with planes with similar routes and landing times. Therefore, drops by packet expiration will be produced at similar times, implying no useful improvements in delivery ratio.



(a) Varying radio range

(b) Varying traffic per node



(c) Varying node buffer size

Figure 7.3: Delivery Ratio: evaluation of two routing protocols (Epidemic and ELT) under changing conditions.

7.3.2 End-to-End Delay

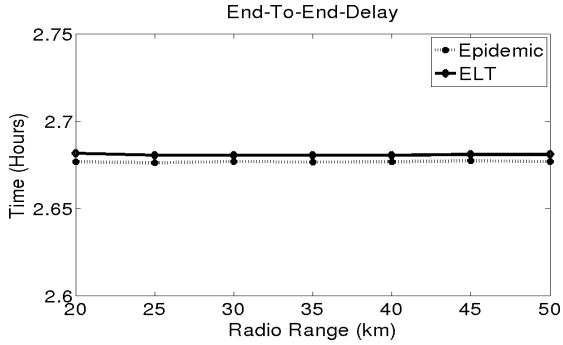
The end-to-end delay provides an estimation of how long does the data require to reach the ground. Intuitively the most influential parameter should be the radio range, as new it produces new contacts and thus, alternative and perhaps faster ways to deliver data. In Figure 7.4a, we show the average end-to-end delay for changing radio ranges from 20 to 50 km. We can observe that it remains mostly stable despite the variation in radio range. We conclude that the increase in radio range is not enough to create contacts with planes from other routes, as such there are no new paths from which the data can be delivered faster to the destination.

Subsequently, we analyze the impact of increasing the traffic demand. In Figure 7.4b, we display the averaged end-to-end delay for changing amount of traffic, from 2 to 35 KB. We can observe little variations, the values of this metric are mostly influenced by the mobility of nodes and the paths available to the destination. Therefore, the increase in packet drops has very little impact in the end-to-end delay. In Figure 7.4c, we can observe end-to-end delay under changing buffer sizes (from 20 to 70 packets). Similarly to the variation of traffic demand, we can see observe a mostly constant tendency, this is because the impact of both parameters usually just has an effect on the number of packet drops. Therefore, as it has been seen before, this has little impact on the end-to-end-delay.

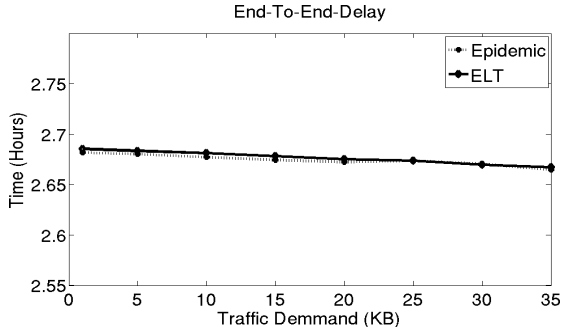
7.3.3 Network Traffic per Packet

The network traffic per packet (or replication cost) gives us an estimate of the overhead generated by each transmitted packet. We start by analyzing the influence of the radio range. In Figure 7.5a, we show the average values of network traffic per packet under different configurations (20 to 50 km). We can observe that higher values of radio range increase the replication cost. This observation is easily explained by the fact that increasing the radio range improves network connectivity, so there is both a higher frequency of encounters and longer contact durations. This improvement results in an increased number of chances to forward data, and the opportunity to exchange a higher number of queued packets. In the end, the network traffic per packet increases as a result of a higher degree of replication.

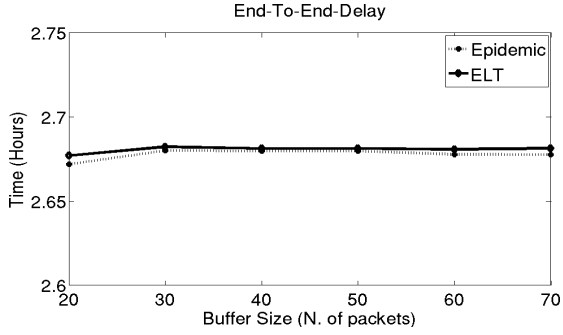
In Figure 7.5b, we show the changes in the network traffic per packet for



(a) Varying radio range



(b) Varying traffic per node



(c) Varying node buffer size

Figure 7.4: End-to-End Delay: evaluation of two routing protocols (Epidemic and ELT) under changing conditions.

changing values of traffic volume. We can see that increasing traffic demand reduces the network traffic per packet. This trend is explained by the fact that for fixed buffer sizes and relatively high expiration times, the packets will eventually accumulate. Therefore, a high traffic demand results in a higher number of nodes with filled buffers, which results in new incoming packets not being able to replicate.

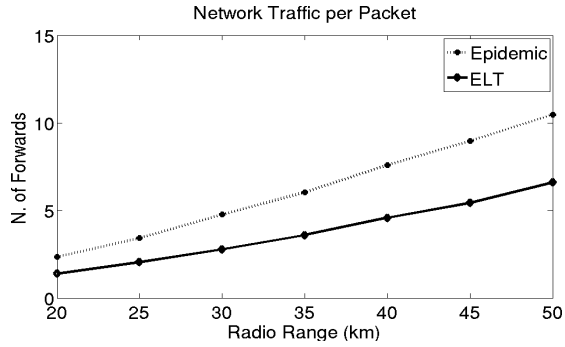
Finally, we analyze the impact of changing the buffer size. In Figure 7.5c, we show the distribution of the replication cost for changing buffer sizes. We can see that it increases slightly along with the buffer size until reaching a maximum value (at size 40), and then becomes stable. For small buffer sizes, the buffer fills faster causing the forwarded packets to be dropped. Increasing the buffer size allows more packets to be successfully replicated, and the higher bound at which the values stabilize is defined by the amount of data being generated by the network.

7.3.4 Routing Protocol Comparison

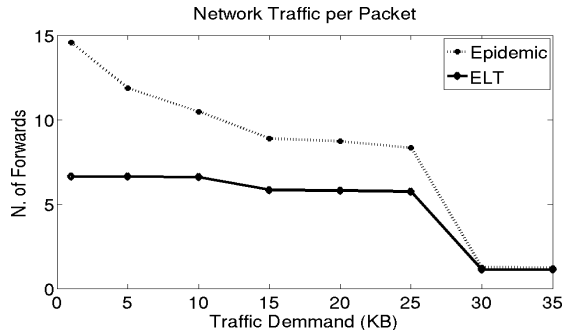
In this section, we will analyze the results from the previous sections, and highlight the main differences in network performed displayed by the two routing protocols. The variations on performance displayed by the selected routing protocols are mostly equivalent. Since both protocols are based on the same principle, they are influenced by the changing network parameters similarly, differentiated only by the scale of the impact.

In Table 7.2, we show a comparison of the two protocols using fixed parameters: radio range (50 km), traffic per node (10 KB), and buffer size (100 packets). As expected, the ELT constrained protocol offers much lower network traffic at the cost of slightly decreased delivery ratios. Even though we are using a flooding approach (to maximize successful deliveries) we get a relatively low delivery ratio of $\sim 60\%$, mostly due to expiration drops. The network traffic displayed by the epidemic routing is ~ 10.5 on average, but if we take into account the number of nodes of the network (2,878) this represents a very small percentage. We have also noticed that the configuration parameters have little influence on the end-to-end delay, as it usually spans several hours and is mostly influenced by mobility patterns.

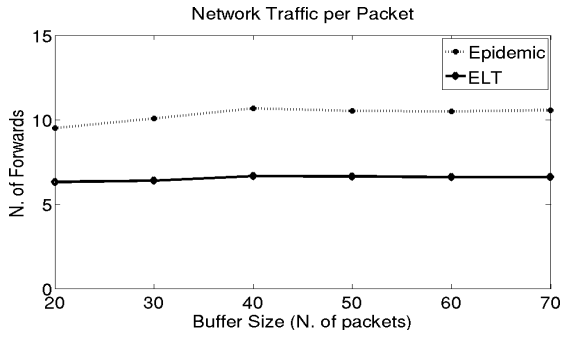
Regarding the results obtained when varying parameters, we can say that the epidemic routing protocol offers slightly better performance under loose configuration constraints. Otherwise, the ELT is preferable because it achieves



(a) Varying radio range



(b) Varying traffic per node



(c) Varying node buffer size

Figure 7.5: Network traffic per packet: evaluation of two routing protocols (Epidemic and ELT) under changing conditions.

Traffic per node	10 KB	
Buffer Size	100 packets	
Radio Range	50 km	
Protocol	**Epidemic**	**ELT**
Min. End-to-End Delay (hours)	2.6770778	2.6810722
Max. Delivery Ratio (%)	0.594838	0.594317
Network Traffic (hops)	10.4784	6.59046

Table 7.2: Network performance comparison of the proposed routing protocols (Epidemic and ELT).

similar results with a great reduction in network traffic. It is also interesting to note that the limited spread of packets makes the ELT a good decision function for this scenario. In most cases, the packets replicate very few times and remain in nodes that traverse similar routes, which makes selecting the one with the smallest time of arrival the optimal choice.

7.4 Communication Delay Optimization

In this section, we evaluate the feasibility of using opportunistic communications as a way to speed-up the delivery of data in this disconnected scenario. To this end, we define a specific custom metric, based on delivery times, which provides a way to quantify the degree of improvement.

- **Delivery time reduction:** Is the gain in seconds obtained by forwarding the packet through the network instead of performing a direct delivery upon leaving the disconnected area.

7.4.1 Delivery Time Reduction

This metric is based on the following concepts. Assume that an aircraft gets disconnected at time t_d, and regains connectivity at time t_c. During this interval a packet is generated, say at time t_g, then the time required for its delivery would be $t_c - t_g$. Now the aircraft, instead of waiting for t_c, uses the sporadic encounters with other aircraft, and forwards the packet. When the packet finally

gets delivered by a different aircraft, at time t_f, then the time for delivery will be $t_f - t_g$. In the case that t_f is smaller than t_c there is a reduction in delivery time ($t_c - t_f$) and we call this improvement *delivery time reduction*.

In Figure 7.6 we show the CCDF of the delivery time reduction in linear scale. Configuration parameters for this Figure are fixed to: radio range (50 km), traffic per node (10 KB), and buffer size (100 packets). As it can be seen, the network displays moderate improvement against direct delivery. First, roughly half of the samples offer an improvement of less than 185 seconds, and a significantly better delay (1,000 seconds) is only displayed in 23% of the cases. Finally, less than 10% of the samples show improvements of 5,000 seconds or more. This value represents one-third of the disconnected flight duration (14,400 seconds), we consider this value a significant improvement and use it as a baseline indicator of good performance. Despite that, the percentage of flights in which we can find this improvement is small.

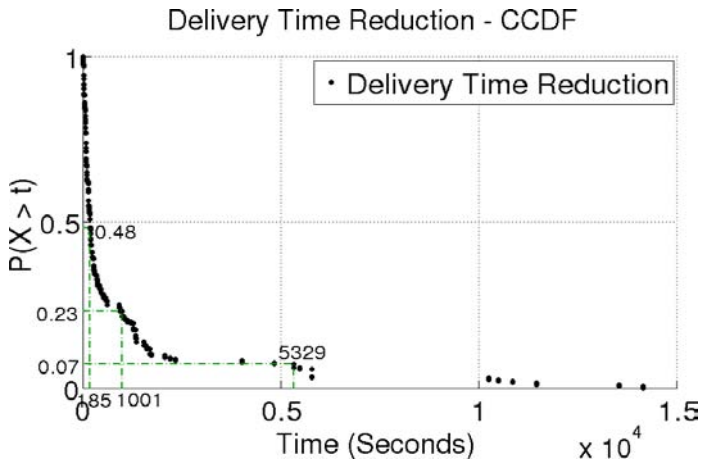


Figure 7.6: CCDF of the Delivery Time Reduction

These results show that almost half of the samples have almost no improvement (less than 185 seconds). We can explain this result by two facts. First, the contacts between planes are not homogeneously distributed: some aircraft display a high number of encounters while others remain isolated for most of the flight (see Chapter 6). An implication of this disparity is shown in Figure 7.2 which displays a low degree of packet replication. Figure 7.2 shows the CCDF of

the number of replicated packets. We can observe that less than 20% ($2 \cdot 10^{-1}$) of the samples achieve a degree of replication higher than 2. The second factor is related to the geographic location of the encounters (see Chapter 5). We observed that the contact locations are closely related to the aircraft trajectory and that encounters are produced between aircraft following similar routes. As a result, planes encountering each other leave the disconnected area at similar times, and packets lack the chance to spread further away from its source.

7.5 Summary and Conclusions

In this chapter, we evaluated the feasibility of using opportunistic communications as a way to perform reliable air-to-ground communications. We evaluated the performance of the network using two differentiated routing protocols, comparing both of them and pointing which was the preferred routing strategy. Finally, we quantified the degree of improvement provided by opportunistic communications towards minimizing delivery times.

We conclude that achieving reliable air-to-ground communications with this network is challenging due to several reasons. First, the delivery ratio is relatively low, although this could be improved by using sufficiently large buffers with high expiration timers. Nevertheless, the main drawback is the limited dissemination of information due to the apparition of node communities isolated from each other. A direct consequence of this is that more than half of the nodes show no delivery time reduction when forwarding data. Nevertheless, we would like to point that there is a small percentage (7%) that offers fairly good results (regarding delivery time). These results show that opportunistic communications can achieve the initial objective at least in some cases. In fact, this is limited to aircraft that have a high number of contacts with other nodes heading to geographically distant destinations.

We believe that it is possible to change the network architecture in such a way that those results can be mimicked in other less connected areas of the network. To this end, we propose to retrofit a handpicked set of selected nodes with specialized satellite links. As a result, we obtain a two-level hierarchical network architecture. This architecture relies on opportunistic communications in tandem with a backbone of nodes with satellite links to improve connectivity of key areas of the network. The details and evaluation of this proposal will be provided in Chapter 8.

To some degree, there's an insatiable demand among people for connectivity, and particularly the business traveler who wants to get some work done while they're flying

Michael Small. Gogo InFlight Chief Executive. TV Interview.

People have an expectation, and it's grown over the past few years, that Wi-Fi should be free.

Tim Farrar. TFM associates Satcom Analyst. Commenting on Wi-Fi on-flight services.

8

Evaluation of Hybrid Networking Alternatives

IN this chapter, we propose a novel network architecture and quantify the improvement that provides towards minimizing delivery times. This architecture defines a two-level hierarchical network composed of two differentiated classes of nodes. Nodes using short-range radio technologies and opportunistic communications operate together with nodes owning a dedicated satellite link. With this new architecture, we seek to remedy the shortcomings of using a purely opportunistic approach that were enumerated in Chapter 7. With the inclusion of these specialized nodes, a new issue appears which is the economical cost. Therefore, our deployment strategy aims to provide an optimal distribution of nodes that reduces costs and effectively minimizes the transmission delays while improving the degree of connectivity of the network. To this end, we heavily rely on the information provided by our network topological analysis (see Chapter 6). Discussion of the obtained results, followed by a cost evaluation and a comparison with existing alternatives, is provided at the end of this chapter.

8.1 Hierarchical Network Architecture

Our previous architecture focused on using a purely opportunistic approach. Unfortunately, the performance evaluation of the network offered limited results and unveiled several inherent deficiencies (see chapter 7). Therefore, in this section we propose a new strategy that expands the network architecture to a hierarchical one with two differentiated node categories. We propose to combine opportunistic communications with the usage of dedicated satellite links. Satellite Communications are the current state of the art aircraft air-to-ground communication system for disconnected environments [93, 3, 1, 2]). The main drawbacks for aeronautical satellite communications are the economic cost (both in usage and deployment) and their dependence on deployed infrastructure (satellite constellations). This dependence adds constraints to the service depending on the coverage offered by those satellites.

This novel architecture is a combination of both technologies in which one approach is used to mitigate the drawbacks of the other. In one side, we improve the connectivity and delivery times of the opportunistic network, and on the other side we reduce both the costs and the infrastructure dependence of satellite communications. Our architecture seeks to become a balanced, cost-effective solution that minimizes the requirement for expensive satellite links with the use of almost cost-negligible opportunistic communications.

In Figure 8.1, we show a graphical concept of our two level hierarchical architecture. The lower level is composed of nodes equipped with a radio-link with limited range (depicted by a red circle). The upper level is composed of nodes with a dedicated satellite link in addition to the radio link. Nodes in both levels can communicate with each other using their radio links. In this way, the upper level can extend its satellite communication capabilities to the lower level nodes that come into contact. In turn, lower level nodes extend this service by forwarding data originated from other nodes using opportunistic communications. We now describe the characteristics of the two node types that define the levels of our architecture:

- **Basic nodes:** This sort of node is analogous to the ones considered in previous chapters. They are only able to use short-range radio communications based on 802.11 b/g and rely on opportunistic encounters to forward their data. These nodes have negligible cost and will constitute the majority of the nodes in the network.

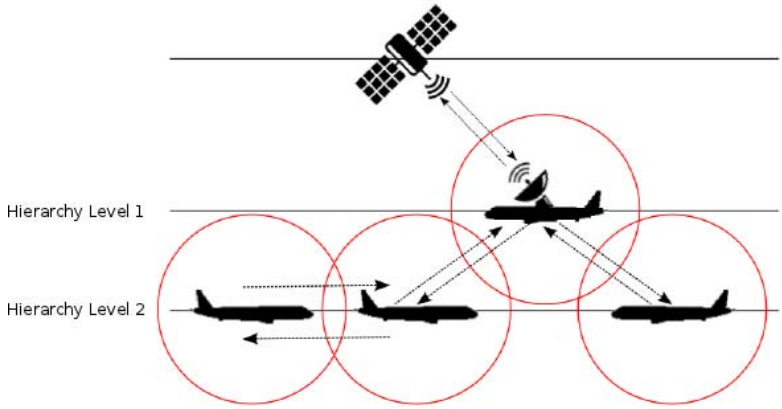


Figure 8.1: A depiction of the two-level Hierarchical Aircraft Network Architecture. *Satellite forwarder nodes* represent the upper-level hierarchy while *basic nodes* using opportunistic communications represent the lower-level.

- **Satellite forwarder nodes:** This sort of node has the characteristics of a basic node plus the capability of using satellite communications. These nodes will provide fast delivery of packets during flight time, providing an alternative way for data delivery. Data generated from basic nodes can now be delivered during flight time instead of waiting for the carrying aircraft to land. The dedicated satellite link of these nodes is costly, therefore we try to minimize their number and deploy them only on key locations.

The inclusion of this new kind of node provides additional exit points for delivery inside the network, which will allow faster air-to-ground communications and result in higher delivery time reductions. The extent of this improvement is determined by the number of **satellite forwarder nodes** that can be deployed. The main priority of our deployment strategy is to minimize this number (and thus reduce the economical cost) while maximizing delivery time reduction.

To achieve this objective, selecting the nodes more suited to perform this task is critical. The key aspect to consider is that **basic nodes** in the network still rely on opportunistic communications to transmit their data. Therefore, the selection of forwarder nodes must be based on factors such as the number,

Simulation Parameter	**Value**
Radio Communication Range	50 km
Simulated Time	48 hours
Number of Nodes	2,249
Routing Protocols	Epidemic Routing
Beacon Interval	400 seconds
Buffer Capacity	100 packets
Max. Number of hops	18 hops
Expiration Time	14,440 seconds
Traffic Generators	CBR
Packet Size	512 bytes
Traffic per node	10 kbytes
Bit Rate	256 bytes/second

Table 8.1: Hierarchical Architecture experimental set-up: general simulation, routing protocols, and CBR configuration parameters.

frequency and timing of contacts between those nodes. We achieve this optimal selection by basing our deployment strategy on the analysis of the contact graph representation of the network (see Chapter 6) and the use of network centrality metrics. This method provides a way to determine the nodes most capable of efficiently performing as **satellite forwarder nodes**.

8.2 Experimental Configuration

In this section, we describe the experimental set-up used to perform the several measurements and experiments. It is composed of the following protocols and models:

- Mobility Model described in chapter 3.
- Network Layers 1 to 4 described in chapter 4.
- *Epidemic Routing* as our routing protocol (described in chapter 4.3.2).
- *Constant Bit Rate* traffic generators (described in chapter 4.5.2).

The configuration parameters for this chapter are shown in Table 8.1. Many configuration parameters heavily rely on the results from the topological characterizations of the network (see Chapter 6). This study revealed that the network is grouped into several independent node communities. The most populated and heavily connected component was designated as α, representing the 78% of the nodes in the network. As a result, from this point onward our study will focus on evaluating component α which is composed of a total of 2249 nodes. The other components are of little interest due to the scarcity of nodes and the inability to communicate with nodes in different components.

In regards to **satellite forwarder nodes**, we treat them as data sinks and mark any packet that reaches them as successfully delivered. We do not account for possible delays in the use of satellite links nor include a packet loss model. We consider delay and losses for this link negligible in comparison to the ones displayed by the nodes using opportunistic communications. On the other hand, **normal nodes** will be equipped with CBR traffic generators with identical configuration as the one used in Chapter 7.

The insertion of a new communication model inside of the network architecture changes some of the basic assumptions for the scenario. The first relates to how data is delivered to the destination. Data delivery can now be performed by nodes exiting the disconnected area or by nodes capable of satellite communications. As a result, the use of the airplane expected landing time (ELT) is no longer an appropriate metric for routing. Therefore, for the experimentation in this Chapter we will only consider *Epidemic Routing*. The configuration and rationale for the parameters: *Beacon Interval*, *Buffer Capacity* and *Expiration Time* is identical as that of Chapter 7. However, the higher-bound value for the *Max Number of Hops* is finely tuned to 18, which corresponds to the network diameter of component α (see Table 6.2).

8.3 Deployment Strategy

In this section, we describe the methodology used to select the most suitable nodes to perform as satellite forwarders. For this purpose, we search for notorious nodes inside the network through the use of the centrality metrics defined in chapter section 6.4.

8.3.1 Inferring Node Notoriety

We define the set of nodes N which represents the total of nodes (2,249). Afterwards, we compute the value of each centrality metric for each node of this set. Lastly, we create three list of nodes ordered by its metric value in decreasing order. A configurable percentage of the nodes in each list is picked in top-down order to define the following subsets:

- $D \subseteq N$ (based on degree centrality): nodes in this set are popular nodes with a high number of contacts with others, their structural position favors the role of conduits of information.
- $B \subseteq N$ (based on betweenness centrality): nodes in this set display a structural position that allows control over the information flowing through the network.
- $C \subseteq N$ (based on closeness centrality): nodes in this set are display a structural position allows a fast spread of information originating from them.

Nodes in these sets display interesting characteristics that make them good candidates to perform special roles within the network and, therefore, have a high chance of improving delivery times. Using these sets as our criteria for satellite deployment we will try to achieve several goals. First, determine the extent of the performance improvement achieved by a hierarchical architecture over a purely opportunistic approach. Secondly, evaluate the efficiency of using centrality metrics as the criteria to choose the adequate nodes. Lastly, determine the most effective centrality metric for choosing nodes that need to prioritize data delivery in air-to-ground communications.

8.3.2 Communication Delay Optimization

In this section, we evaluate the extent of the performance improvement achieved by a hierarchical architecture over a purely opportunistic approach. To this end, we use the network metric defined in Section 7.4.

- **Delivery time reduction:** Is the gain in seconds obtained by forwarding the packet through the network instead of performing a direct delivery upon leaving the disconnected area.

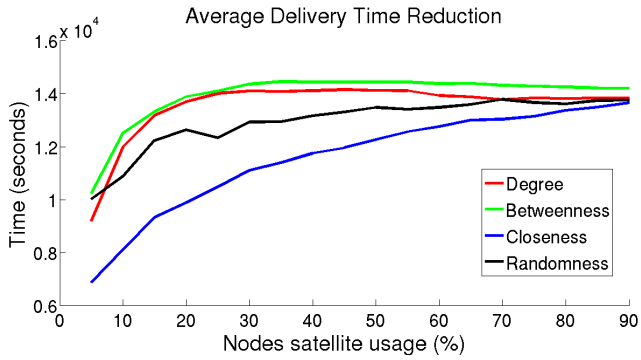


Figure 8.2: Average Delivery Time Reduction for sets of variable size, ranging from 5% to 90% of the total of nodes.

In the experimental evaluation of this section, we will deploy a satellite link on any node contained in a centrality set. Additionally, we define a fourth set U that will contain nodes chosen uniformly at random and will be used as a baseline for comparison purposes. The sets B, C, D, and U will have varying sizes ranging from 5% to 90% of the total number of nodes.

In Figure 8.2, we show the average delivery time reduction corresponding to multiples sizes of the four sets, represented by percentages of the total of the nodes from 5% to 90%. In Figures 8.3, 8.4, 8.5 we show the CCDF in linear scale corresponding to the sets of size $5\%, 15\%$ and 40% of the total of nodes (respectively). Each one represents a relevant point of the exponential growth displayed by the values shown on Figure 8.2.

In Figure 8.3, we can see that all four selection criteria provide a fair improvement when compared with a pure opportunistic approach (Figure 7.6). In this case, more than half of the samples offer an improvement higher than 5,000 seconds (one-third of the average disconnected flight duration). The next interesting fact is that using closeness offers very poor results, even lower than selecting nodes at random. Conversely, using betweenness and degree proves to be a good selection criteria as they offer far better results than using a random selection. From these results, we conclude that delivery time reduction is mostly influenced by nodes acting as hubs for information flow (betweenness) or those with a high quantity of contacts themselves (degree). As opposed to closeness, which evaluates how fast the information may spread from a concrete node but

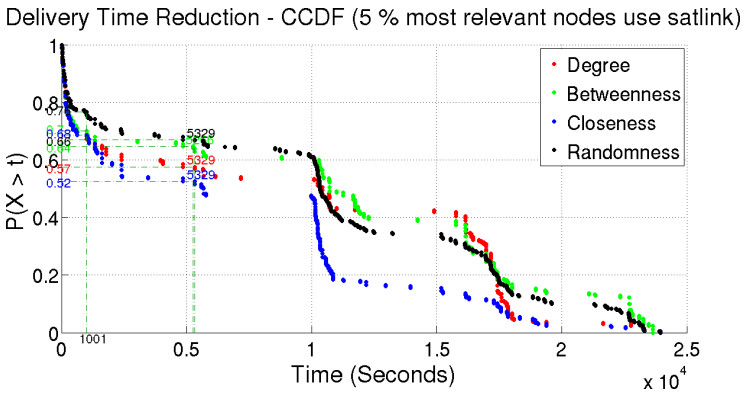


Figure 8.3: CCDF of the Delivery Time Reduction using 5% of total nodes selected from all sets

it does not imply that the node itself handles a big amount of information.

In Figure 8.4, we can observe a higher percentage of samples with increased values for delivery time reduction than those of the previous figure. In the case of betweenness, we can see that more than 88% of the samples achieve a delivery time reduction of 5,000 seconds or higher. In the case of degree, this value is about 82%. In Figure 8.5 these values are further increased. The baseline of 5000 seconds is achieved in 95% of the cases by the samples corresponding to betweenness and 90% by the samples corresponding to degree.

As the number of nodes using satellite links nears 100%, the sample distribution tends to stabilize. All four distributions converge, and the shape of the distribution is analogous to that of the disconnected flight duration seen in Figure 7.1. We can see that the disconnected flight duration defines the optimal upper limit. This is a logical result because the delivery time reduction can not increase any further than the duration of the flight itself.

Lastly, in Figure 8.2, we can observe that the delivery time reduction increases exponentially based on the number of nodes capable equipped with a satellite communications link. First, we can see that both betweenness and degree increase exponentially reaching a maximum and stabilizing at around 30% of the nodes. The random set does not follow a clear distribution, and several oscillations can be observed. The samples based on closeness have a much more limited growth and are significantly lower than the rest. Finally, we can see that

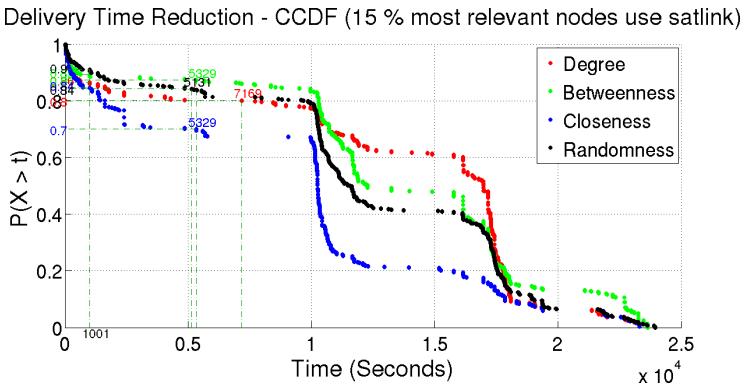


Figure 8.4: CCDF of the Delivery Time Reduction using 15% of total nodes selected from all sets

all four curves eventually tend to similar values. A theoretical limit close to the average disconnected flight duration as observed before. We can also derive the actual delivery time for each configuration by subtracting the average time reduction to the average delivery time of the network with no satellites (∼14,900 seconds).

8.3.3 Deployment Cost Optimization

In this section, we analyze the temporal relationships between nodes. To this end, we will consider the dynamic graph. As a result of this temporal analysis, we obtain a method to minimize the costs of deployment by further reducing the number of satellite links required.

Ensuring Temporal Continuity

In this section, we describe the methodology of our temporal analysis of the dynamic graph. There are several facts to account for when the time of the contacts is considered. First, if an edge connects two nodes, it is ensured that they have a contact at some point in time and also that they will be able to exchange information. Despite that, if the path linking two nodes has a distance greater than one hop, there is no guarantee that the information can be exchanged between these two nodes. In other words, if we assume that

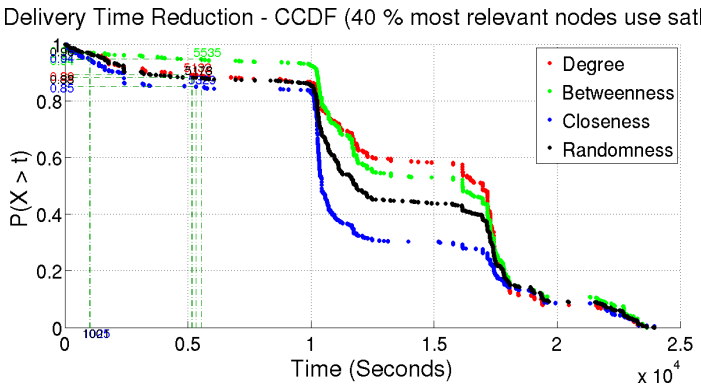


Figure 8.5: CCDF of the Delivery Time Reduction using 40% of total nodes selected from all sets

nodes i and j are linked by a n-hop path, node i is only capable of delivering data to node j if each contact along the path is produced at a later time than the previous one. Following this temporal assumption, in this section we will determine which paths can be effectively used to communicate any given two nodes of the network.

To perform this analysis, we will require the two following subsets:

- $R \subset N$: stands for relevant nodes and may be equal to either B, D or C.
- $I \subset N$: stands for irrelevant nodes and is defined as the relative complement of R in N ($I = N \setminus R$).

First, we represent the network as a directed dynamic graph with edges directed towards nodes in the set R. With this representation, we study to what extent nodes in the set I can reach a node in the set R. To this end, we use the **temporal path** [94] between each node in I towards each node in R. The temporal path between two nodes i and j is defined as a sequence of n hops through a distinct node $n_k^{t_k}$ at time t_k: $p_{ij} = (i, n_1^{t_1}, ..., n_k^{t_k}, j)$ where $t_{k-1} < t_k$. If a **temporal path** from i to j does not exist, (i,j) is a **temporarily disconnected node pair**. Lastly, we compute the number of **available temporal paths** for each node in R using to the following procedure:

- Compute all the combinations of node pairs (i,j) where $i \in I$ and $j \in R$.

- For each node pair, find the shortest path between nodes i, j.
- Evaluate if this path is a **temporal path**.
- If the conditions are meet, then the value of **available temporal path** for node j is increased by 1.

Once all the paths are explored, we define the following set:

- $R_{tmp} \subseteq R$: is an ordered set composed of nodes from R but re-ordered decreasingly according to the **available temporal paths** value of each node.

Finally, we compute the new R subset for each of the notorious subsets (D, B, C) and analyze the similarities with the original sets. We apply Spearman rank correlation to compare R and R_{tmp}, the average of the correlation coefficient for the sets D ($\rho = 0.0067$ with p-value of 0.44661), B ($\rho = 0.0012$ with p-value of 0.4583), and C ($\rho = 0.0158$ with p-value of 0.5158). The correlation coefficients show that there is little relation between centrality and temporal orders. This result indicates that occupying a significant position within the graph does not necessarily ensure good communication with all the other nodes when a temporal factor is taken into account.

We estimate that notorious nodes with a reduced value of **available temporal paths** should have a low impact on delivery time reduction because there are no temporal paths linking to them. Based on this concept, we strive to further optimize satellite link usage by discarding nodes located on the lower part of the R_{tmp} subset.

Evaluation and Comparison

In this section, we estimate if using temporal criteria to reduce the number of deployed satellite links is capable of offering comparable performance to that of basic static centrality metrics. To evaluate network performance we use the delay based metric **delivery time reduction** as defined in section 8.3.2. We compare the performance of deploying satellite links based on both R and R_{tmp}. As R, we use the subset B, since it is the one that offers the best performance in the static analysis of the contact graph (see section 8.3.2). After constructing B_{tmp} we discard the 10% of its lower ranked nodes, effectively removing the nodes with a small number of temporal paths. After this procedure, B_{tmp} has

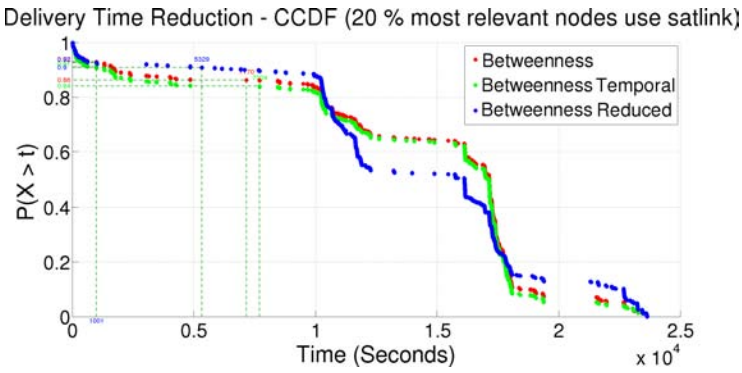


Figure 8.6: Comparison of the performance of centrality and temporal ordering. The figure uses betweenness centrality and sets with a size of 20% of the network nodes.

fewer nodes than B. Therefore, in the following experiments we will compare three sets of nodes:

- The set B with a size of 20% of the total number of network nodes.
- The set B_{tmp} computed from B using the procedure from section 8.3.3, and discarding the 10% of its lower ranked nodes.
- The reference set B_r which is used as a comparison reference (r stands for reduced). We construct this set from B discarding the 10% of its lower-ranked nodes, directly and without performing temporal path exploration and reordering. B_r and B_{tmp} have the same size.

In Figure 8.6 we compare the delivery time reduction obtained using the nodes of the sets B, B_{tmp}, and B_r as the satellite link deployment criterion. As we can see, the samples from B_{tmp} display slightly lower performance than those of B, but a noticeable improvement if compared to the samples of B_r.

In Figure 8.7, we show the average delivery time reduction corresponding to multiples sizes of the three sets, represented by percentages of the total of the nodes from 5% to 90%. As we can see, B and B_{tmp} display very similar results but always slightly better than B_r. All three samples show an exponential curve. The differences in performance between the three subsets can be mostly

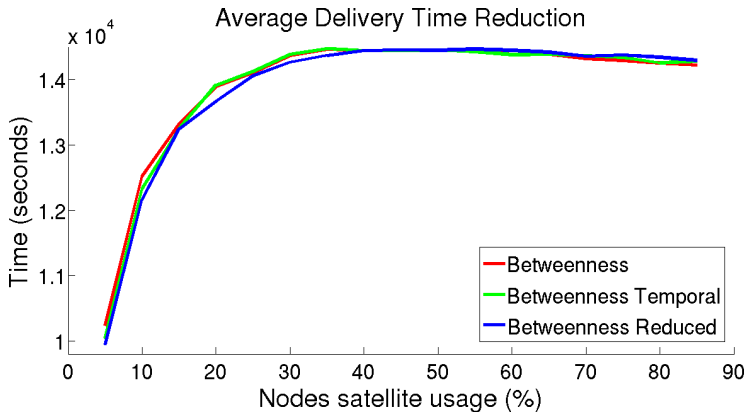


Figure 8.7: Comparison between basic and temporal ordering. Average Delivery Time Reduction for sets of variable size (5% to 90% of the total nodes)

appreciated in the first stages of the exponential growth, at higher percentages of nodes all three curves stabilize and converge to similar values.

These results show that B_{tmp} is an optimized version of B that requires fewer satellite links to achieve comparable performance. On the other hand, it is also appreciated that temporal ordering is a preferable criterion since B_{tmp} achieves better results than using a B set of the same size (B_r). From these results, we can confirm that the new order imposed by the temporal restrictions allows us to successfully discard the nodes that despite being structurally relevant have a poor contribution to network performance.

8.4 Summary and Conclusions

In this chapter, we provided a detailed evaluation of a novel hierarchical aeronautical network architecture which merges the concept of opportunistic communications with satellite communication systems. We provided an off-line deployment strategy based on the analysis of the graph representation of the contacts between nodes. Finally, we evaluated the network performance resulting of deploying satellite links according to this deployment strategy. The performance is vastly superior to that shown by a purely opportunistic approach, and successfully solves the inherent problems highlighted in Chapter 7.

Evaluation results show that using betweenness centrality to equip a (∼35%) of nodes with satellite communication systems, provides performance on par with those architectures that equip satellite systems on all nodes. We further reduced the number of nodes required by proposing a methodology that considers the temporal relations between nodes and discards those that have little impact on network performance. Those contributions can be used to greatly reduce the deployment cost of satellite-based architectures in this kind of scenarios.

This chapter offers a new approach that takes the currently deployed technology (satellite links) and offers an alternate extension at negligible cost (opportunistic communications) effectively reducing the expenditure in satellite links. This approach is a relevant contribution in this area as nowadays airlines operate on rather thin profit margins, and one of the main concerns is optimization and reduction of costs.

Considering the economic cost, some estimations like [95] estate that equipping a single aircraft with satellite-based equipment for air-to-ground Wi-Fi capabilities has a minimum cost of $100,000. Retrofitting unused equipment from one plane to another is also a very costly and time-consuming task. It is also stated that some airlines are currently offering onboard Wi-Fi at a loss in part due to the high investment required for equipment. In regards to usage costs, satellite providers operating in the K_u band[1] (such as [2] and [3]) are the most economic offers available on the market. According to [96], the estimated cost of those services is 20 cents per megabyte, which is about $200 per gigabyte. If we compare it to the rates of similar services such as 4G LTE (usually around $10 per gigabyte) it is obviously a very costly service.

Our approach provides a way to equip satellite links efficiently just to the nodes in key locations and obtain delivery-times fairly close to those that would be obtained by installing a satellite link per aircraft. From the results obtained we conclude that the optimal selection criterion is to use betweenness, as seen in Figure 8.2, which is able to reach optimal values with approximately just using a 35% of total nodes. It is straightforward to affirm that this approach reduces the cost of fitting an airline's fleet to almost one-third of its price, and this base cost may be further reduced by applying optimization based on temporal relations.

[1] 'K-under' is the 12–18 GHz portion of the electromagnetic spectrum assigned to satellite communications. It is commonly used for fixed and broadcast services. Compared to other bands, it allows increased signal power and smaller dish sizes making it generally cheaper for the end-user.

Achieving the required Quality of Service (QoS) by managing the delay, delay variation (jitter), bandwidth, and packet loss parameters on a network becomes the secret to a successful end-to-end business solution.

CISCO

Cisco IOS Technologies - Documentation

9 Measure of the Quality of Service

IN this chapter, we evaluate the behavior of the hierarchical network architecture proposed in Chapter 8 when Internet-like application traffic is deployed into the network. We evaluate the network in regards to Quality of Service (QoS) and measure several related aspects such as the delivery time, delivery ratio, and the degree of fairness on bandwidth allocation. Promoting QoS in opportunistic networks is usually really hard due to their long delays and scarce resources. Our hierarchical network architecture represents a prospect to provide an improved degree of service and, therefore, QoS needs to be properly quantified and validated. We close the chapter with a discussion of the obtained results and present the list of service guarantees that can be provided.

9.1 Experimental Model

In this section, we define the experimental scenario used for the evaluation of QoS metrics. The Complete Model (defined in chapter 5.1) gives accurate insight on the behavior of the network and has been consistently used through

previous Chapters (5, 6, 7, 8) as the standard evaluation model. However, the fact that nodes are numbered in the thousands makes it unpractical for the study of applications generating a high traffic volume as simulation time grows exponentially. Therefore, in this section we propose a method for generating statistically equivalent (in regards to connectivity) reduced versions of the scenario. The new model, more suited for this kind of studies, is described in the following section.

9.1.1 Reduced Model

This experimental model is based on a small selection of nodes chosen from the total used in Chapter 8. As our base, we use the nodes from component α (see Chapter 6). This component contains a total of 2,249 nodes, and 35% of them carry a satellite communication link.

We define a selection algorithm that focuses on providing a connected network that maintains the same proportion between satellite links and standard nodes. The algorithm takes as input the original set of nodes (2,249) and the size of the new scenario. The output will be a scenario of the requested size that maintains the same connectivity properties of the original. This algorithm works as follows:

- Use the size of the reduced scenario (n), and a graph representing all nodes of the complete model (G) as input. Initialize a void set $\mathbf{N}$ of maximum length n that will be the output.
- Compute the number of required satellite communication links for the new scenario as: $s = 0.35 \times n$.
- Choose s non-repeated satellite nodes uniformly at random from the input scenario (G) and define a list $\mathbf{S}$ containing them.
- Iterate through $\mathbf{S}$ applying procedure BreadthFirstSearch (see Algorithm 2). As a result, we obtain a list of neighbors ordered in BFS exploration order for each chosen satellite node.
- Iterate among the first element of each list. Check whether it is a satellite node or not. If negative add it to $\mathbf{N}$, otherwise move to the next list. Once all the lists have been visited move on to the second element of each list, then third, and so on.
- Repeat until $\mathbf{N}$ has been filled.

Algorithm 2 Simplified functions used during the scenario generation procedure.

```
function BreadthFirstSearch(node,G,depth)
    ▷ Explore G in BFS order starting from node up to depth
    ▷ Return list (L) of nodes in exploration order.
    return L
end function
function GenerateScenario(S,N,G,n,s)
    NeighbourLists[s][ ] ← {}                    ▷ Define container of lists of size s
    i ← 0                                        ▷ Index for NeighborLists
    depth ← 0
    TotalNeighbors ← 0
    N ← N + S
    do
        depth ← depth + 1
        for all SatNode in S do
            NeighbourLists[i] ← BreadthFirstSearch(SatNode, G, depth)
            TotalNeighbors ← TotalNeighbors + SizeOf(NeighbourLists[i])
        end for
    while TotalNeighbors < (n − s)
    do
        pos ← pos + 1
        for i = 0 ; i < s ; i++ do
            Node ← NeighbourLists[i][pos]
            if NotIsSat(Node) AND NotIn(N, Node) then
                N ← N + Node
            end if
        end for
    while SizeOf(N) < n
    return N
end function
```

The resulting scenario is a network that starts with a set of satellite nodes and grows outwards including neighboring nodes of increasing depth until reaching the specified maximum size. We set up our algorithm to $n = 200$. This generates a scenario with 200 nodes, 130 of which are basic nodes and 70 satellite nodes.

9.2 Experimental Configuration

In this section, we describe the experimental set-up used to perform the several measurements and experiments. It is composed of the following protocols and models:

- Mobility Model described in chapter 3.
- Network Layers 1 to 4 described in chapter 4.
- *Epidemic Routing* as our routing protocol (described in chapter 4.3.2).
- *4IPP* traffic generators (described in chapter 4.5.2).

The configuration parameters for this chapter are shown in Table 9.1. In this chapter, we will use the reduced model described in the previous section, and thus the scenario is composed of 200 nodes (130 basic nodes and 70 satellite forwarders). The scenario is generated by using Algorithm 2. As this algorithm includes randomness each run provides different results. In the next sections, results of each experiment will be the average of ten runs using different generated scenarios.

In this chapter, we use more complex traffic patterns, that resemble traffic commonly found on the Internet. As a result, more complex traffic generators have been used along with a more precise configuration of the routing protocols. Therefore, in the following sections we will provide a detailed description of the configuration parameters of both traffic generators and routing protocols.

9.2.1 Traffic Model Configuration

The basic nodes are equipped with a traffic generator and are capable of data forwarding. Alternatively, the satellite forwarder nodes are equipped with data-sinks and do not generate data. Each aircraft will only generate traffic during the whole duration of the disconnected flight. The outgoing traffic of each node represents all the user and company traffic generated inside of the aircraft. Lastly, we assume that each aircraft represents an independent LAN with a Router acting as its exit point. Therefore, the outgoing traffic is modeled using one instance of the 4IPP traffic model (defined in 4.5.2). We set up a single instance of the traffic model per aircraft because we are only interested in analyzing outgoing traffic.

Simulation Parameter	**Value**
Radio Communication Range	50 km
Simulated Time	48 hours
Number of Nodes	200
Routing Protocols	Epidemic Routing
Beacon Interval	400 seconds
Buffer Capacity	250,000 packets
Max. Number of hops	18 hops
Expiration Time	14,500 - 20,000 seconds
Traffic Generators	4IPP
Packet Size	192 bytes

Table 9.1: Evaluation of the Quality of Service of the aeronautical architecture. Experimental set-up: general simulation, routing protocols, and basic traffic configuration parameters.

Note that a standard 4IPP model can not be used out of the box. In order to provide data rates fitting our aeronautic scenario, the traffic model needs to be properly dimensioned using an estimation of the output bandwidth of the Router. Our estimation of the output bandwidth is based on the values found in aircraft communication Internet services (as shown by [97]). This work provides bandwidth usage statistics of the six most operated aircraft models for North Atlantic passenger flights. We chose to use the values corresponding to the Airbus 380, the model with the highest number of passengers (550), which displays outgoing data traffic of 8 kbps. The 4IPP model is scaled using a factor of 1.7361 (computed as in expression 9.1). Finally, in Table 9.2.1 we show the parameters for the scaled traffic model.

$$\begin{aligned} \textbf{Packet rate} &= \textbf{Data rate} \;/\; \textbf{Packet size} \\ 8000\ \textit{bps} \;/\; 1536\ \textit{bits} &= 5.2083\ \textit{packets/sec} \\ \textbf{Scaling factor} &= \textbf{Desired Packet Rate} \;/\; \textbf{Default 4IPP Packet Rate} \\ 5.2083\ \textit{packets/sec} \;/\; 3\ \textit{packets/sec} &= \mathbf{1.7361} \end{aligned} \tag{9.1}$$

	c1 probability rate (transitions/sec)	c2 probability rate (transitions/sec)	(pkts/sec)
IPP#1	7.936e-01	5.953e-01	1.9930
IPP#2	2.509e-02	1.881e-02	1,2635
IPP#3	7.936e-04	5.953e-04	1.0328
IPP#4	7.936e-06	5.953e-06	0.9182
		4IPP Average Rate: (packets/unit-of-time)	5.2083

Table 9.2: 4IPP traffic model parameters scaled to 8 kbps

9.2.2 Routing Protocol Configuration

As in previous chapters, we rely in Epidemic routing to maximize delivery ratio and minimize delivery time by exploring all the possible routing paths. The configuration of routing protocol parameters has a big impact on network performance. Our main concern is to ensure a good delivery ratio, since this is a critical metric to ensure quality of service. The configuration and rationale for the parameters *Beacon Interval*, *Expiration Time* , and *Number of Hops* is identical as that of Chapter 8. However, the increased traffic volume requires a redefinition of the optimal *Buffer Capacity*. Additionally, the values for the parameter *Expiration Time* may change depending on the experiment performed.

To assign an appropriate *Buffer Capacity* we need to consider several aspects. Mainly, we need to guarantee that the buffers can hold the amount of data being generated by a node plus the additional data generated by encountered nodes. Those nodes in turn may even be holding data from others).

First, we consider the data generated by the local node. To this end, we need to evaluate the data generation ratio (GR) and the duration of the flight (DF). Secondly, we account for the data from other encountered aircraft. In each communication opportunity, the aircraft will forward some packets from its buffer but in exchange will store data from other aircraft. We need to relate the number of contact opportunities that happen during the flight and the amount of data generated between encounters (inter-contact packets, CP). For this purpose, we relate the data generation rate (GR), the time interval between data exchanges (given by the $IACT$), and the average number of contact opportunities (CO).

Lastly, we need to account for aircraft carrying data from sources other than

Number of Copies per Packet	c	N/A
Inter Any Contact Time	$IACT$	1,500 seconds
Disconnected Flight Duration	DF	14,400 seconds
Generation Ratio	GR	5.2083 pkts/second
Inter Contact Packets	$CP = GR \times IACT$	7813 packets
Contact Opportunities	$CO = DFD \ / \ IACT$	10
Packet Replication Percentage (based on Figure 7.2)	PR_c	$80\% < c = 2$ $20\% < c = 3$

Table 9.3: Relationship between relevant parameters used in the computation of the optimal buffer size.

their own. We know some facts from our previous study (see Chapters 5,7). First, the average node degree is approximately one unit. Second, the values of packet replication (see Figure 7.2, show that only 20% samples are replicated more than twice. We include this information into our calculation by defining the packet replication percentage (pr_c). For simplification purposes, we only consider the percentage of packets that are replicated twice or less (80%) and the rest (20%) from which we assume a replication of 3. Please note that any desired degree of precision may be achieved from the values from the CCDF shown on Figure 7.2.

We represent the previously described parameters, their values and the relationship between them in Table 9.3. Finally, we relate all of those parameters and define the optimal buffer size in expression 9.2.

$$\textbf{Buffer Size} = DF \cdot GR + \sum_{c=2}^{n} c \cdot PR_c \cdot CP \cdot CO \tag{9.2}$$

Expression 9.2 is evaluated up to three replications ($n = 3$). As a result, we obtain a buffer size of ∼250,000 packets. Afterward, we compute the storage requirements for the nodes using the packet size (192 bytes), which results in approximately 45 MB. In a scenario such as this one, where nodes are aircraft and the limitations on computer equipment are mostly negligible, we consider that this value is fairly affordable.

9.3 Quality of Service

In this section, we introduce the concept of Quality of Service in an opportunistic network and apply it to our hierarchical architecture. We describe the several metrics used to measure QoS, along with the parameters and characteristics of the performed experiments.

The term Quality of Service describes the overall quality of the network perceived by its users. QoS measurements are usually based on characteristics of the link such as bandwidth, transmission delay, or throughput. Other important aspects are the delivery ratio of packets and the fair allocation of resources.

Opportunistic networks (and DTNs) are challenged networks that require delay tolerance and as a result cannot guarantee fixed delivery times. Therefore, factors such as the end-to-end delivery time or throughput cannot be used as a good measure of QoS on this kind of networks. On the other hand, this kind of network should be robust. So a good degree of delivery ratio should be guaranteed. Other common aspects of such networks is the scarcity of resources. In our scenario, the critical resource would be the access to satellite links and the limited bandwidth offered by that service. A fair distribution of this bandwidth among the several basic nodes of the network should be assured. To this end, we evaluate the QoS of our scenario using the following metrics:

- **Delivery time:** Is defined as the time required for a packet to reach its destination.
- **Delivery Ratio:** Is defined as the ratio of successfully delivered packets to the total number of generated packets at the end of the simulation.
- **Bandwidth Allocation Fairness:** Quantifies the fairness in the allocation of resources among nodes. We use the Jain's fairness index [98] which offers an estimation on the fairness of network resource allocation based on the throughput of the nodes.

9.3.1 Delivery Time

In this section, we study the delivery time. In opportunistic networks, this metric is characterized by being long and variable. As such, it is usually not possible to provide a guaranteed delivery time, and service is provided in a best-effort manner. In our scenario, there are two factors that allow us to achieve

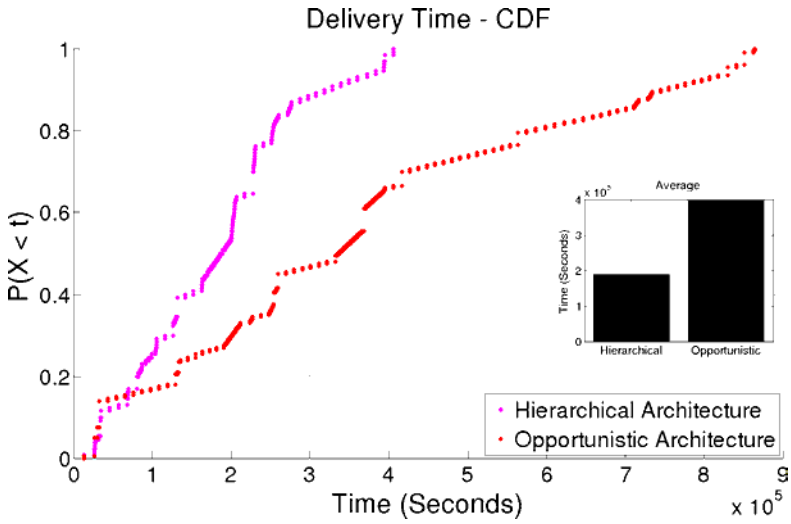


Figure 9.1: CDF of the Delivery Time

a compromise, first the predictable contact pattern of our nodes and secondly the availability of the distributed satellite nodes. For this experiment, we use the default parameters from table 9.1 with an *Expiration Time* of 14,500 seconds. We compare the hierarchical architecture against a purely opportunistic approach using the exact same nodes but without any satellite link available.

In Figure 9.1, we show the CDF of the delivery time. The abscissa represents the time component. The ordinate represents the percentage of packets delivered at a time less than or equal to the one shown in the abscissa. We can see that the opportunistic network samples show long delays with samples almost reaching a delay of ∼90,000 seconds. Additionally, the average of the samples is ∼40,000 seconds. Fortunately, 50% of the samples is smaller than ∼40,000 seconds. On the other hand, the hierarchical architecture shows a clearly reduced delivery time, with 100% of the samples being below ∼40,000 seconds and an average delay of ∼10,000 seconds. This fact clearly shows that the presence of the distributed satellite nodes helps to improve the delivery time, and concretely it roughly doubles the performance regarding this metric.

9.3.2 Delivery Ratio

In this section, we study the delivery ratio. The delivery ratio is an important metric to infer the reliability of the network. Several parameters influence this metric, but a proper configuration of the routing protocol is essential. In the following experiments, we will evaluate how the hierarchical architecture provides a significant improvement on delivery ratio compared to a pure opportunistic approach.

The two main routing parameters that affect delivery ratio are the *Expiration Time* and the *Buffer Capacity*. The *Buffer Capacity* has been heavily analyzed theoretically resulting in expression 9.2 and its value has been fine tuned to maximize delivery ratio. On the other hand, in this section we will study the impact of the *Expiration Time* by means of empirical experimentation.

In Figure 9.2, we show the average delivery ratio (expressed in a ratio between 0 and 1) under varying expiration times. We start evaluating our results using the default expiration value (14,500 seconds) and progressively increase it up to 20,000 seconds. This figure displays two lines; one corresponds to the hierarchical architecture and the other to a purely opportunistic approach. As we can see, a purely opportunistic approach displays fairly low delivery ratios (0.65) for expiration values below 17,500 seconds. On the other hand, the hierarchical architecture always displays delivery ratios higher than 0.8. The opportunistic approach requires an expiration time of 18,500 seconds to achieve the same delivery ratio displayed by an expiration time of 14,500 seconds of the hierarchical architecture.

The results from Figure 9.2 show that the insertion of satellite nodes has a high impact on the delivery ratio. The reason is that the presence of nodes capable of delivering data while still on flight reduces the need for holding data in the buffer for extended periods of time. As a result, the number of packets dropped by expiration decreases. Additionally, the buffers are less congested which also reduces the number of dropped due to lack of space.

Subsequently, we study how the inclusion of satellite nodes speeds up the delivery of messages. Before proceeding, we recall a specific scenario characteristic from our mobility characterization (see Chapter 5). Nodes in this scenario, display two peaks of activation. These peaks represent the usual trend of North Atlantic flights: airplanes start flying west (from Europe to America) during the day and east (from America to Europe) during the night. In Figure 9.3, we show the percentage of packets delivered (with respect to the total) at a

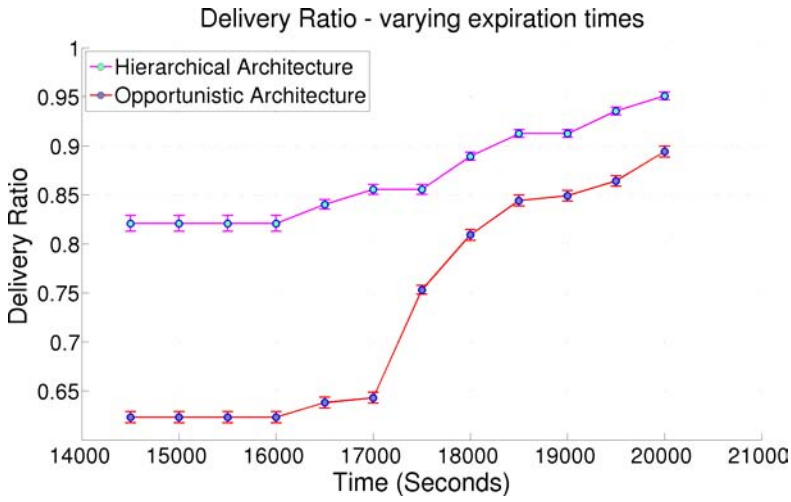
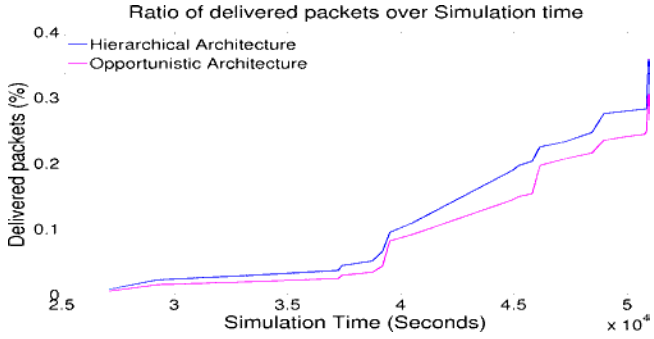


Figure 9.2: Average of the Delivery Ratio - 10 simulation runs with confidence intervals.

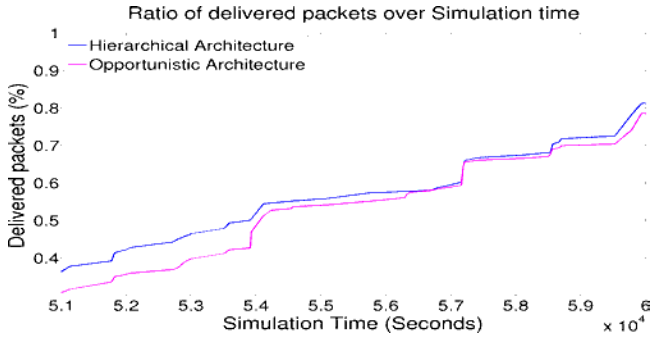
specific time. The figure is split into three intervals, which correspond to the two peaks of node activation and to the median interval where planes flying in both directions converge.

In Figure 9.3, we can appreciate that the hierarchical architecture always displays a slightly faster delivery ratio, mainly because the presence of the satellite links. This can be easily appreciated in Figures 9.3a and 9.3c. Inversely, Figure 9.3b displays a time interval that has many contacts between nodes flying in opposite directions. This characteristic offers opportunities for fast delivery in an opportunistic manner. As a result, both architectures show relatively similar results.

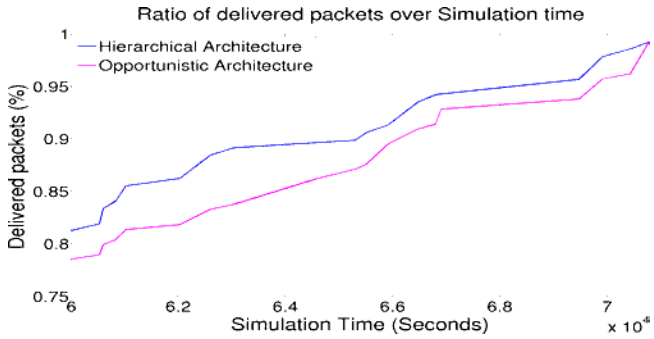
In both cases, full delivery of all packets is only achieved at around ∼75,000 seconds. This late delivery is caused by packets being generated during the whole flight and the contacts with satellite nodes being limited. As a result, some packets can not be delivered using the a satellite link. Therefore, they will only be delivered using opportunistic communications or upon landing of the aircraft.



(a) First peak of node activation.



(b) Intermediate interval with high number of contacts.



(c) Second peak of node activation.

Figure 9.3: Coefficient (0-1) of total packets delivered over simulation time.

9.3.3 Bandwidth Allocation Fairness

In our basic scenario, the bandwidth available to perform air-to-ground communications depended primarily on mobility factors. Factors such as the number of encounters and the how fast can data spread to reach the ground. The addition of nodes equipped with satellite communication provided a new alternative and greatly increased the available bandwidth. As a result, the access to these special nodes became an important resource that needs to be equally distributed.

To assess the fair distribution of resources we use the Jain fairness index [98] which is described by the mathematical expression 9.3. This index rates the fairness of a set of resources for n users where x_i is the throughput for the i-th connection. Values range from $\frac{1}{n}$ (worst case) to 1 (best case) which signals equality in resource allocation. To compute this index, we consider the aircraft (as if it was a user) and its outgoing throughput.

$$\mathcal{J}(x_1, x_2, \ldots, x_n) = \frac{(\sum_{i=1}^{n} x_i)^2}{n \cdot \sum_{i=1}^{n} {x_i}^2} \tag{9.3}$$

In Figure 9.4, we show the CDF of the throughput displayed by each aircraft for both architectures. It is appreciable that the hierarchical architecture has samples displaying higher throughput than the purely opportunistic approach. First, 100% of the samples from the opportunistic set have throughput below 0.002 kbps. In the case of the hierarchical architecture, around 40% of the samples have values higher than 0.002 kbps, and usually reach 0.004 kbps, a small percentage can achieve up to 0.1 kbps. The average throughput for each case is 0.0023 (hierarchical) and 0.0005 (opportunistic). Note that since the network delays are long the throughput per unit of time is fairly low. These values show that the presence of the satellite nodes improves the throughput by four times. We compute the fairness index for both cases, being 0.3817 for the opportunistic network and 0.4196 for the hierarchical architecture. This result indicates that the inclusion of satellite nodes can improve the fairness of the network.

We now analyze the relationship between network fairness and the distribution of nodes with a satellite link. In this experiment, we use different distributions of satellite links. We obtain each configuration through a run of the scenario generation algorithm (see algorithm 2). Each configuration has a different proportion of normal nodes and nodes with a satellite link. In Figure 9.5, we show a bar plot displaying the fairness index resulting from different network

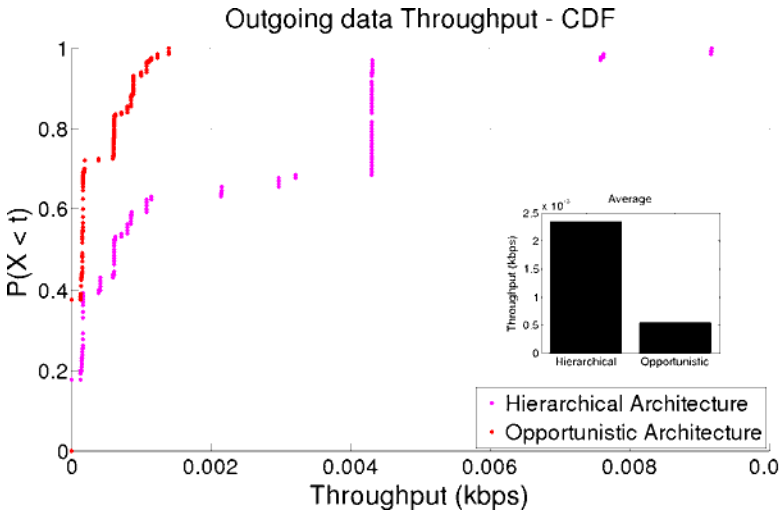


Figure 9.4: CCDF of the Throughput per aircraft

configurations, corresponding to (% of normal nodes, % of satellite nodes). We consider the cases: **(100/0)**, **(85/15)(75/25)**,**(65/35)**, **(50/50)**, **(35/65)**.

In Figure 9.5, we can observe that the use of satellite nodes can help improve the fairness to some degree, but not in all cases. It depends heavily on the distribution and number of satellite nodes. The network architecture without satellites has a fairness index of 0.38. When we add small numbers of satellite nodes to the network (15%), the fairness decreases due to a reduced number of nodes having much higher bandwidth than the rest. As the number of satellite nodes increases the availability to the higher bandwidth becomes more widespread and thus the fairness increases, achieving a maximum fairness index of 0.41. Again, if the number of satellite nodes increases too much it generates disparities between nodes with high bandwidth nodes and nodes with limited access to the resource, thus the fairness index decreases.

9.4 Summary and Conclusions

In this chapter, we provided an evaluation of the QoS of the hierarchical network architecture proposed in Chapter 8. To this end, we provided an experimental

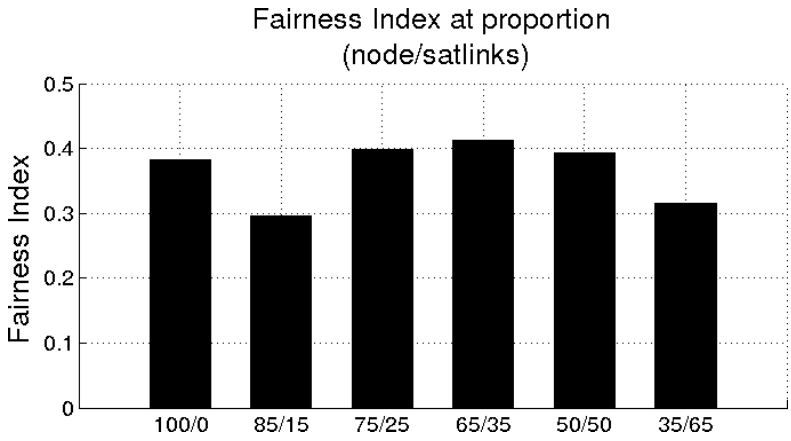


Figure 9.5: Fairness Index representation: 200 nodes with different proportions of normal nodes and nodes with satellite links

model of the transoceanic network scenario for use in simulations with high traffic volumes. Using this model, we measured several QoS related aspects such as the delivery ratio, the delivery time and the fairness in the distribution of network resources. Finally, we compared the improvements in service obtained with this new network architecture compared to that of a simple network using opportunistic communications. Obtained results show that the delivery ratio is increased by almost a 30% margin, delivery times are cut to one-third, and the fairness can be slightly increased. Furthermore, the configuration with 35% of satellite links seems to be the optimal distribution.

These improvements allow us to offer certain guarantees in terms of delivery and time of the service, which is quite unusual on the field of opportunistic networking and DTNs. To define these guarantees, we first need to take into account the available resources of the scenario. In this regard, there are no limitations concerning buffer sizes and data storage due to the nodes being aircraft. On the other hand, the access to satellite links is costly. Therefore, their usage should be limited. To sum up, with the default hierarchical network architecture composed of a 35% of nodes with satellite links we can guarantee:

- Delivery ratio higher than 90%.
- Delivery times below 10,000 seconds.

- A bandwidth fairness index of 0.41.

Achieving those results requires expanding the packet expiration to values higher than 18,000 seconds. These changes influence the buffer of the node which are cost negligible. Nevertheless, even without this increase, we can guarantee delivery ratios higher than 80%. On the other hand, the increase in bandwidth fairness is more discrete. Further improvement would require deploying a specific resource management policy for satellite distribution.

Part IV
Conclusions and Future Work

The phrase that is guaranteed to wake up an audience: "And in conclusion."

Author Unknown

10 Conclusions and Future Work

IN Chapter 1 we presented the objective of this thesis: to provide a reliable and efficient communication alternative to conventional aeronautical communications based on opportunistic networking. In this chapter, we review how we have fulfilled this objective and present the future research lines available on this topic. some

10.1 Conclusions

We started this endeavor using the initial results obtained from our previous work on small-scale DTN aeronautical networks (see Appendix A). In this thesis, we aimed at bringing this research one step further. We attempted to obtain a general network architecture based on opportunistic communications, and capable of tackling the communication problems of large-scale aeronautical systems.

Our first step towards this goal was to locate the primary networking obstacles found in aeronautical scenarios. This was followed by a careful study of

general knowledge found in prior large-scale opportunistic network studies (see Chapter 2). As a result, we established the fundamental matters that need to be satisfactorily addressed before starting any proper architectural design. Node mobility is a critical aspect of opportunistic networks and, therefore, needs to be represented with accuracy. Furthermore, to ensure successful deployment in real networks, the use of realistic mobility is of severe importance.

In consideration of those facts, we moved onto designing a complete mobility model (see Chapter 3). We modeled the mobility patterns of hundreds of aeronautical nodes, including real flight schedules and examined the implications of departure delays. Our objective with this model was to set the foundation for an in-depth analysis that would lead to the design of a working network structure. In our task to provide as much reality as possible to the evaluation scenario, we then progress onto describing the full network stack for the nodes (see Chapter 4). We took special consideration into correctly measuring the achievable radio ranges, as in opportunistic networks the other critical aspect apart from movement is the communication range of the nodes.

Now with an adequate model to represent our network scenario, we progressed onto analyzing the properties of mobility itself (see Chapter 5). As asserted by many previous works in opportunistic networking, knowing information about facts such as frequency of contacts, or identifying predictable node patterns can lead to developing highly successful network schemes. Similarly, this study inferred essential characteristics of the network that proved to be instrumental in the later design of the architecture, the configuration of protocols, and to understand seemingly nonsensical occurrences in data propagation.

With this detailed knowledge, we moved onto analyzing how data propagates inside our transoceanic aircraft scenario using opportunistic communications (see Chapter 7). Our first objective was to examine the problem of air-to-ground communications, which primarily meant to deliver data successfully outside the opportunistic network. To this end, we sought to maximize network delivery performance, and after considering the sparse connectivity of the network, disregarded network load on the process. We chose to use a commonly known opportunistic routing protocol, Epidemic routing, an *uncontrolled* replication based routing protocol. To attenuate the problem of the message overhead we additionally designed our own *controlled* adaptation using a scenario specific metric.

Unfortunately, we obtained relatively small delivery ratios, and delivery times were not satisfactory for more than half of the nodes, although a small

percentage (7%) that displayed reasonably good results. These issues were seemingly produced by the limited dissemination of information due to the apparition of node communities isolated from each other.

Going back to the drawing board, we decided to analyze the topological structure of our network scenario (see Chapter 6). In this analysis, we used a *social network graph* constructed from the contacts between nodes. This tool allowed us to characterize the structure of the network, helping us to locate the most relevant nodes, and to identify the several communities within the whole network. As a result, we designed an entirely new architecture capable of solving the performance deficiencies of a pure opportunistic approach (see Chapter 8). This new architecture defined a hierarchical network composed of two distinct types of nodes. In this architecture, nodes using short-range radio technologies with opportunistic communications operate together with nodes owning a dedicated satellite link. Due to these new nodes, the economic cost became a new concern to factor in our studies. To this end, we provided an optimal distribution method of satellite links based on graph centrality. Fortunately, this new strategy successfully solved the problems regarding delivery time performance and maintained costs to the minimum.

Finally, with a working scheme that meets our delivery standards, we moved toward measuring Quality of Service (QoS) when real Internet-like traffic is deployed (see Chapter 9). Our results showed a fair degree of improvement in regards to a pure opportunistic network. Additionally, the characteristics of our new architecture allowed us to offer certain guarantees in terms of service, which is quite unusual in the field of opportunistic networking and DTNs.

We can say that our proposal is an efficient network architecture successfully tested for data delivery in air-to-ground communications. The immediate use of this architecture revolves around applications that do not require data round-trip. We can provide some examples based on the final user:

- Passenger data: examples of such applications are the delivery of mail or twitter messages.

- Airline data: one example of such an application is early notification of faulty components prior to landing for compliance with MEL (Minimum Equipment Lists). These lists are safety regulations files that define which items (instrument, equipment or systems) are required so the aircraft can be allowed to depart after an intermediate stop. These lists can include non-critical components such as cabin lighting items, and a departure may

suffer delays due to the time required for finding a spare and replacing such element. For this reason, early notification of any faulty equipment during the flight can allow for a faster response after landing.

Some degree of service can also be provided in the field of air-to-air communications. Applications like meteorological forecast, on-route event information, or notification systems can be used between aircraft in a one-hop fashion.

10.2 Future Work

Our results have provided a practical architecture with a fair with of potential. In this section, we will enumerate the several areas that remain unstudied or can be improved,

The first area that has room for improvement is providing support for successful air-to-air communications. This type of communication poses several challenges. First, as opposed to air-to-ground where our only priority was existing the disconnected area (since all paths lead to the ground), in air-to-air we need to locate a particular aircraft. Due to the isolated nature of the network, nodes can not rely on external information such as continuous routing table updates like those of proactive routing protocols used in MANETs. An adequate approach to this problem is the use of a probabilistic routing protocol such as Prophet[70]. Another option would be a protocol based on Markov Chains describing node ICT patterns (found in some works on Vehicular Networks [99]). Additionally, the distribution of satellite nodes through the network allows the possibility of taking a "short-cut" by using two satellite nodes as relays to reach a particular node faster. This kind of routing would be somewhat complex and nodes would have to be provided with knowledge about the network topology.

Another area is the study of ground-to-air communications. If the destination is a particular node, this would be a subclass of air-to-air communications. Alternatively, if we are broadcasting information from the ground to all the nodes in the network, satellite nodes can be used to speed-up the spreading process by broadcasting simultaneously from several areas of the network. This method has some interesting applications in the field of network security, and it could be used as part of a key distribution scheme. We had some work in this area [11] where we used Identity Based Encryption (IBC) as an alternative to Public Key Infrastructure (PKI) in disconnected aeronautical environments.

The inclusion and evaluation of a full DTN stack is another topic of interest. In the last couple of years, there has been a substantial effort towards modeling the standard DTN protocols accurately in the ns-3 simulator. Implementations of the Bundle Protocol [100] and the Licklider Transmission Protocol [15] are readily available but there is a lack of support for end-to-end multi-hop communications. This fact makes our evaluation not possible at the moment.

Another topic for future work is the implementation of non-architectural improvements to QoS. We believe that similar approaches as those found in the literature may be able to improve the fairness of resource allocation. The introduction of traffic priority levels would help distribute satellite link access more equitably (Bundle priority classes could be used). And as a result, it would be possible to implement specific buffer management policies.

Part V
Appendices

"Ok, it is gone ... wait a second ... and ... back"
On succesful round-trip transmission.
"Saturn is down. I can see it, it is definitely in range... but there is no ping response."
Upon hardware failure of the RC Helicopter node.

PROSES field experiment transcript quotes

A

Aeronautical DTNs: A Proof of Concept

This chapter describes the results of the Spanish Science Foundation PROSES (Protocols for the Single European Sky) project [6, 7, 8]. This project offered a proof of concept, including physical deployment and field experimentation, of a small-scale aeronautical Delay-Tolerant Network. This chapter is a re-elaborated extension of the paper:

- R. Martínez-Vidal, S. Castillo, S. Robles, A. Sánchez, J. Borrell, M. Cordero, A. Viguria, N.Giuditta. *Mobile-Agent Based Delay-Tolerant Network Architecture for Non-Critical Aeronautical Data Communications.* In Proceedings of the 10th International Symposium on Distributed Computing and Artificial Intelligence. Springer, vol. 217, (May 2013), pp. 513-520. [8].

A.1 Introduction

The year 2020 will mark a turning point in the field of air traffic management (ATM) and control, as the next evolution in ATM is expected to become fully

operational and deployed both in Europe and in North America. The Single European Sky (SES) initiative will unify the heterogeneous air traffic control models used by each country, transforming the European airspace into a single integrated air management scenario.

The correct deployment of these initiatives will depend on network connectivity able to support multiple applications at the same time. However, the mobility of the network elements and their heterogeneity (ranging from commercial passenger planes to small, general aviation users) introduce several obstacles in terms of maintaining constant connection, such as interoperability, disruption, security, delay and so on. This renders the majority of current-day communication protocols invalid for their inclusion in such scenarios.

Hence, PROSES [6, 7] was conceived to tackle these issues and offer a stable and reliable communication solution for the future ATM framework. The starting point of our research has been Delay Tolerant Networks, a scheme originally created [40, 101] to answer the difficulties of deep space communications, and currently being applied in other areas thanks to its possibilities. Our objective is to transfer this concept to ATM to design a protocol oriented to every possible participant.

PROSES introduces a new architecture, moving from prevailing store-carry-forward technologies used in previous DTN applications, to focus instead on the advantages offered by mobile agents (both data and routing algorithm carried together). This new architecture (see section A.2) represents a low-cost solution suitable for diverse aeronautical applications which need diverse routing schemes (see section A.3). A set of virtual emulations and field tests using radio-controlled aircraft is presented in sections A.4-A.5. These experiments show the practical effects of the new architecture under different conditions, meant to reflect application scenarios derived from the future implementation of the new ATM framework.

A.2 Mobile-agent based DTN architecture

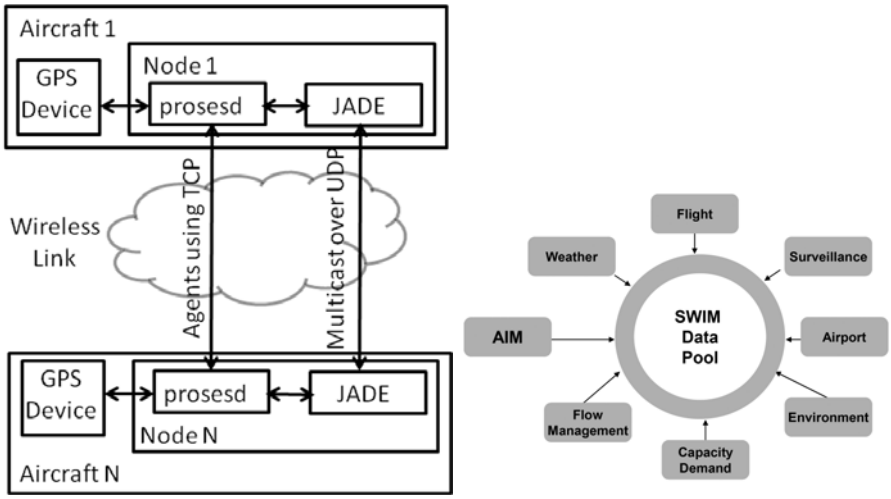
Traditionally, computer networks have been designed using protocol stack schemes organised in several layers, with the theoretical OSI model being the most representative standard. The first ideas on DTN still use this scheme, posing it as an additional "overlay" layer providing the application layer with the support

needed to withstand the problems derived from disrupted connectivity and delays in data exchange. However, the current proposals of DTN models, such as Bundle Protocol[41] or LTP [42], do not effectively answer a number of issues, mainly regarding the routing problem and its underlying processes, preventing the formation of a distinct DTN standard.

PROSES introduces a layer-less DTN conception, doing away with the hierarchical model of tiers and instead proposing a model based on several modules interacting without rigid interactions. This novel communication model for DTN networks requires a basic communication service beneath, the so called Network Abstraction. This abstraction defines a set of minimum specifications in order to maintain an effective connection between the nodes.

Taking into account our previous assertion, the mobile code - and the mobile agents in particular - is a clear way to provide a solution of a multi-routing protocol scenario like in the DTN networks. PROSES is sustained through the use of mobile agents, software entities moving from host to host in a network, and able to carry data and code which can be autonomously run, such as routing schemes. In this sense, our proposal for the PROSES environment is to use the JADE framework [102] as the basis of our infrastructure. In fact, a modified implementation of the Inter-Platform Mobility Service (IPMS) module [103] of the JADE framework has been made in order to overcome the main problems of the DTN routing. This gives us the chance of introducing as many routing protocols as applications we want to deploy.

The PROSES architecture (see figure A.1a) proposes the use of JADE mobile agents as carriers of data messages. Mobile agents allow the construction of active messages composed by data and routing algorithms. PROSES agents implement forward decision algorithms which use information about the network nodes to decide which node will be the next hop in the path towards the final destination of the data message. This routing information is gathered by an independent software daemon called "*prosesd*" running on each node and is stored in a memory array accessible by the JADE platform running in that node. Each *prosesd* daemon basically reads the current node's position from a GPS device attached to the node and sends multicast messages announcing its presence and position information to the neighbor nodes, which will store the received information in the routing arrays so it can be accessed by the PROSES agents.



(a) PROSES Architecture (b) Information sources present in SWIM

Figure A.1: Mobile-agent based DTN architecture.

A.3 Application scenarios

One of the main features that characterize SESAR (SES ATM Research) is a massive increase in data communications between the participants in European ATM. The system that will provide the new networking functionalities is called SWIM (System Wide Information Management [104]). This infrastructure will integrate distributed and/or geographically sparse services, like third party services (e.g. weather information services), monitoring systems, by means of the different communication links (air-ground datalinks, satellite, grounded networks...), acting as an "intranet" for ATM users and services (see figure A.1b [105]).

PROSES has identified a number of scenarios where it is possible to improve the connectivity and interoperability of SWIM using the DTN approach, as an intermediate step towards its full deployment or as a complementary system to augment its planned capabilities. In order to select the PROSES scenarios, the project team has focused on the situations where connectivity is not fully assured and the introduction of a new system able to withstand interruptions

would be useful. Another research line is to act as a secondary system in a SWIM environment, offering certain low-cost services with non-critical requirements to ATM activities. This study has resulted in the selection of two main scenarios:

Total connectivity: Nowadays, there are certain airspace sectors, mainly during transatlantic flights, where the surveillance radar systems provide no coverage and the pilot must periodically contact the ground ATC facilities by voice communication to update the aircraft position. However, the update rate is quite low and only very basic information can be exchanged this way. PROSES will provide mechanisms to automatically update the aircraft position in this scenario increasing the update rate using other airplanes as relays to send the information to the ground ATC facilities.

Non-critical data exchange: PROSES is aimed to support the interconnected scenario provided in SESAR by offering users a set of basic data communication services (e.g. e-mail, SMS) using air-ground communication capabilities without the need of costly, additional equipment onboard.

A.4 Emulation and field experimentation

A.4.1 Emulation

The implementation of the proposed architecture was first tested and validated in a software emulated network environment. This step proved to be instrumental for the design and validation of the implementation prior to the field work. A commercial software network emulation and simulation called EXata[106] was used for this purpose. It can operate in two different modes: simulation and emulation. In emulation mode, real applications are run in virtual emulated networks that operate in real time. The emulated network behaves exactly as a real network allowing SIL (Software-in-the-Loop) tests to be done. This mode was used for testing real nodes running JADE platforms and *prosesd* software daemons over a emulated wireless network. The SIL tests carried out using the network emulation environment have several objectives: Firstly, the evaluation of the feasibility and suitability of using mobile agents as carriers of user data in DTN scenarios. Secondly, to perform a performance evaluation of the proposed communication architecture in the application scenarios.

Emulation Setup

In the proposed communication architecture, each node is formed by a computer, a wireless transceptor (e.g. WiFi) and a GPS receiver. Figure A.2 shows the emulation environment used for the validation and assessment of the proposed communication architecture. A powerful workstation has been used for running the whole emulation environment. Virtualbox virtual machines running Ubuntu 10.04 operating system have been used as network nodes, each running a JADE platform and a *prosesd* daemon. EXata runs on the native Windows 7 operating system of the workstation (i.e., the emulation host machine).

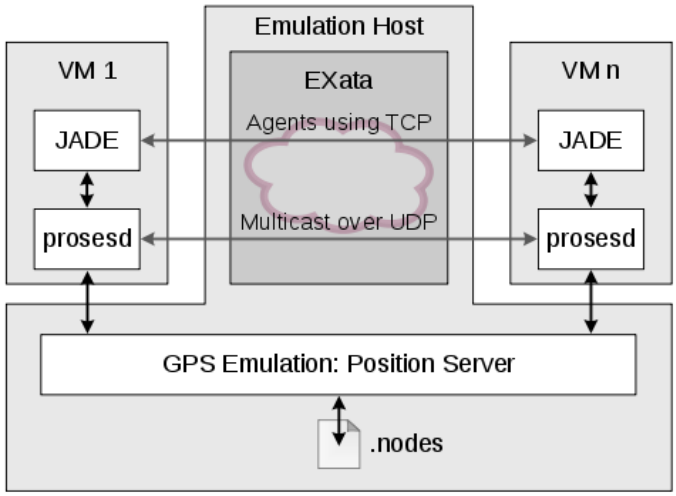


Figure A.2: PROSES Emulation Environment

The emulated network scenarios were defined to meet the main features of the proposed application scenarios. These scenarios include a ground node as well as one or two aerial nodes (depending on the experiment), all of them running a JADE platform and a *prosesd* daemon. The nodes' trajectories emulated

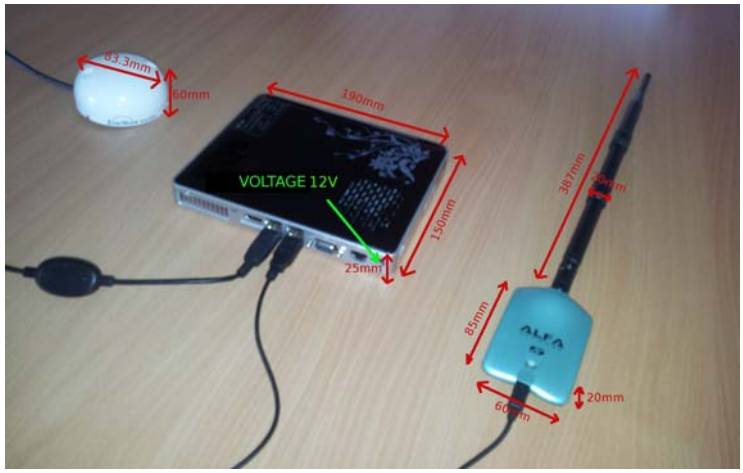


Figure A.3: PROSES node equipment: mini computer, GPS receiver, and WiFi transmitter

in EXata were defined to resemble the connectivity scenarios proposed. A positioning server defines these trajectories. This server uses previously stored files containing a set of waypoints for each node with their position and attitude at a particular time instant (linear interpolation obtains positions and attitudes between waypoints).

A.4.2 Field Experimentation

Following the emulation tests a set of field tests was performed. The field tests pursued a two-fold objective. Firstly, to validate some of the figures found out in the emulations. Secondly, to check the feasibility of deploying the architecture in a real scenario. These field tests evaluate if both the paradigm and the equipment can perform their task with an acceptable level of performance.

Experimental Setup

The field work was performed in an aerodrome near Seville at the end of 2011, and it took one day for equipment integration and two days for flying tests. The basic equipment for nodes was a mini computer, a GPS receiver, and a WiFi

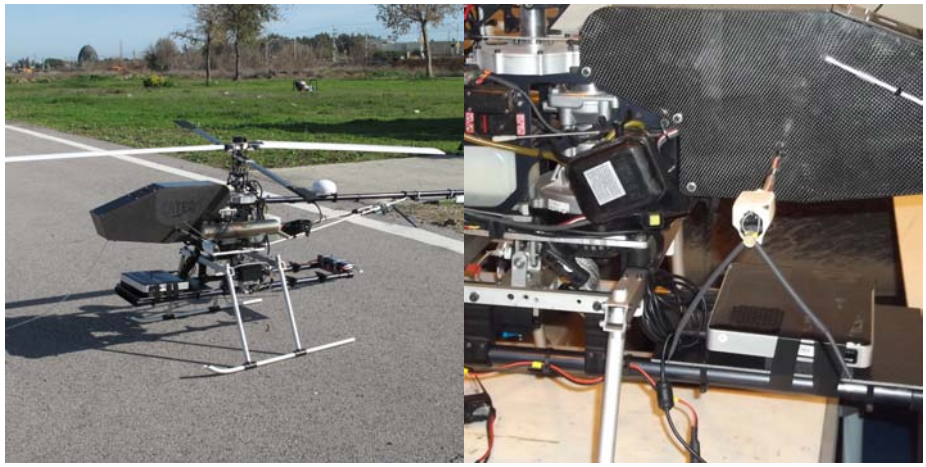
(a) Side View (b) Equipment close-up

Figure A.4: PROSES Aerial Node: RC fixed-wing plane

transmitter (shown in Figure A.3). The communication scenario was composed by three nodes: one ground node installed inside a van (see figure A.6) and two aerial nodes installed onboard an RC fixed-wing plane (see Figure A.4) and a RC helicopter (shown in figure A.5). Agents originated from from the static node located inside the van which additionally performed as the monitoring node.

Incidences

During the experiments, several non-planed issues happened. Firstly, meteorological problems caused the suspension of the first day of experiments halfway. Secondly, technical problems arose on the second day of experiments. The batteries of the aerial vehicles could be used for around 15 minutes, after this time, the level of charge dropped to levels where the computer systems started to malfunction. One of the hard-drives failed, and some experimental data were lost. Another technical issue appeared in the system being carried by the helicopter. The great vibrations made by the helices caused a breakdown of the Wi-Fi adapter attached to the tail. This fact created unplanned disconnections



(a) Side View (b) Equipment close-up

Figure A.5: PROSES Aerial Node: RC Helicopter

of the system from the network, the whole adapter and its connectors had to be replaced to finish the experiments.

A.4.3 Experimental Description

The experiments are divided into several categories based on node connectivity and routing criteria. The emulation experiments were to be similar to those of the field work. Only three nodes have been used in both cases as only two aerial vehicles, and a ground station were available.

In each experiment all nodes are initially located close to each other in the center of the aerodrome runway. At the start, the aerial vehicles take off in opposite directions until they reach a separation of 300 meters. Then, they make a turnaround and go back to the center of the runway, then again they separate themselves 300 meters in the opposite direction. In each experiment, the two aerial nodes moved using different mobility patterns but always keeping this motion of meeting and separating. For simplification purposes, from now on the nodes will be called *A, B,* and *C*. *A* corresponds to the static ground

(a) Monitoring center (b) Experimental Equipment

Figure A.6: PROSES Ground Node: traffic source and monitoring center

station, B and C are the mobile nodes (helicopter and aircraft, respectively).

1. Intermittent Connectivity with routing constrained to predefined itinerary.

A single agent is launched from the static node with a predefined route (A-B-C-A). Hence, if the node that is the next hop in the route is not in range, that agent will wait until the node is close enough for the agent to jump. The path has been defined in a way that assures that the agent will traverse all the nodes in the network and then come back to the origin. Due to the routing criteria used, the agent may have to wait in several occasions (while waiting for node reconnection).

2. Intermittent Connectivity with Dynamic Routing.

Two simultaneous agents are used with a routing decision function that varies its criterion depending on the situation. This experiment seeks to analyze the coexistence of multiple routing schemes in a single network, by offering the same

options to multiple agents and observe them taking different routing decisions.

3. Intermittent Connectivity with Indiscriminate Routing.

Only two nodes (the static one and a mobile node) have been used. Several agents are launched simultaneously from the static node. The routing decision function makes the agent jump to whichever node is in range. This experiment seeks to observe the performance when the routing has no restrictions.

A.5 Results

This section contains the main conclusions made from the results obtained for each one of the experiments. The behavioral data has been represented using chronograms. The chronograms show the state of connectivity and the actions being performed by the several nodes of the experiment. The abscissae show time, and the state or action is represented in the ordinates. Possible states are numbered this way: **[0]** disconnected, **[1]** connected, **[2]** performing neighbor discovery, **[3]** executing routing code, **[4]** performing migration.

In the connected state, the several neighbors that are connected are listed. A crossed circle shows the location of the agent; the dotted arrows depict exchange of neighbor discovery messages between platforms; and finally, the solid lines show the exchange of migration data. For each chronogram, we picked the most common behavior or the one that better shows the purpose of the experiment. Then we proceed to give a detailed description of such concrete case.

A.5.1 First Experiment: Predefined Itinerary

This experiment uses three nodes: *A*,*B*, and *C*. The agent starts its execution in node *A*, its routing function has been set to follow the predefined itinerary *A*-*B*-*C*-*A*. When the experiment starts, *A* performs neighbor discovery, and locates node *C* but not *B*. The routing code is executed, and the agent decides to stay in the platform because *B* is not available. *A* periodically performs neighbor discovery until the mobility of nodes brings *B* in range. Then the routing code is executed again, *B* is chosen as the destination and the agent migrates. Once in *B* a neighbor discovery is performed again, both *A* and *C* are in range, so the agent immediately jumps to *C*. Once in *C* the discovery is repeated and *A* is in range, so the agent jumps there, finally it returns to *A* and the agent finishes

its execution. This process can be seen in figure A.7a, which is a chronogram that follows the conventions explained at the beginning of this section.

A.5.2 Second Experiment: Dynamic Routing

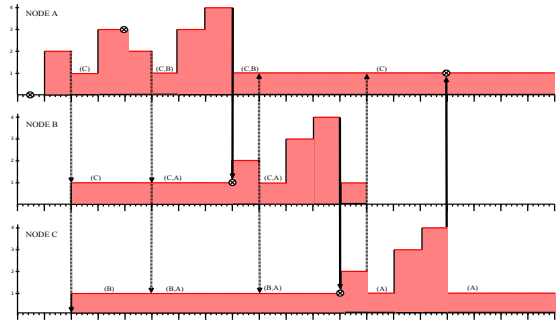
This scenario uses three nodes: *A,B,C*. *A* remains static, *B* and *C* are mobile nodes. Two agents are used, using dynamic routing criteria; they start their execution in node *A*. First, *A* performs neighbor discovery, it locates *B* and *C*. Both agents are notified, the routing code for each agent is executed, the first one chooses to stay in the platform while the other one chooses to migrate to *C*. Platform *A* performs neighbor discovery once again and notifies the agent. This time, it migrates to *B*. Simultaneously, the agent residing in node *C* is executed and selects to migrate to *A*. Node *B* becomes disconnected from the other two carrying one of the agents. The other two nodes exchange the remaining agent with some intermittent disconnections between themselves. Those agents do not have a predefined final destination, so the exchange continues until the experiment finishes. This process can be seen in figure A.7b.

A.5.3 Third Experiment: Indiscriminate Routing

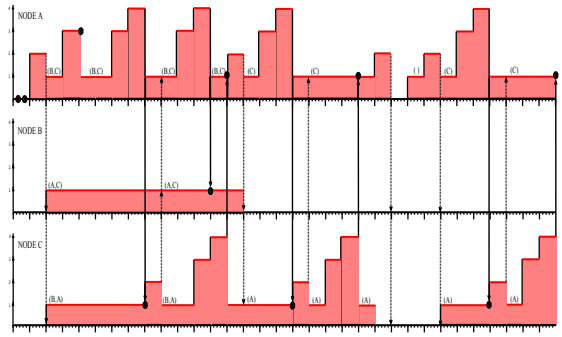
This experiment uses two nodes *A,B*, *A* remains static, *B* has mobility. We use several agents with a routing policy that decides to migrate every time it has the chance. Several agents have been consecutively sent in this experiment. The agents start their execution in platform *A*, they decide to migrate as soon as platform *B* is discovered. After this, the agents keep going back and forth between the two platforms, several interruptions occur due to loss of connectivity when the mobile node goes out of range. The mutual exchange continues until the experiment is finished, see figure A.7c.

A.6 Conclusions

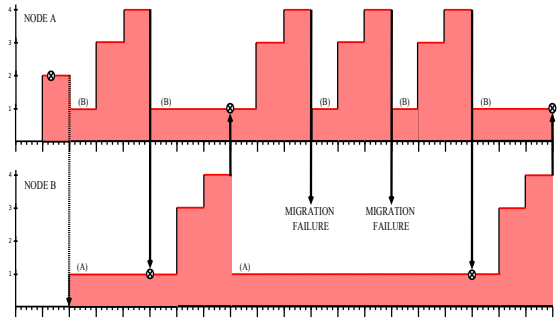
This chapter introduces a novel network architecture based on a new approach to Delay Tolerant Networking, using mobile agents as dynamic messages. The resulting communication network sought to take advantage of the new communication landscape and represents an evolution of current networks in order to adapt to challenged scenarios.



(a) Predefined Itinerary



(b) Dynamic Routing



(c) Indiscriminate Routing

Figure A.7: PROSES field experimentation: Behavior Chronograms.

The presented communication model is quite different from traditional communication networks, for delay, bandwidth, and throughput are second to data delivery. In scenarios were there is not a contemporaneous path between the communicating ends, most network solutions are just outstripped. Other networks can deal with these constraints, but the novelty of the proposal in this chapter dwells in the possibility of simultaneously having different messages using different routing algorithms by using dynamical processes. Furthermore, the scheme also allows to introduce new routing algorithms, or removing them, without uploading new information to node-routers. It is the message itself which carries its own routing algorithm.

The system was successfully designed and implemented. Emulations were run in laboratory to verify the feasibility of the proposal. The system was subsequently uploaded aboard unmanned drones to perform test flights to validate the results and check feasibility. Albeit the proposed network has a number of advantages over other networks, such as being resistant to long connectivity disruptions or supporting several applications simultaneously and yet having the best expediency routing algorithm for each, it is not a general replacement for current networks. The cost of having all these pluses is an overhead that might not be worth when connections are otherwise possible. There are some scenarios, though, for which the PROSES architecture is the only possibility.

There is a project that's underway called the interplanetary Internet. It's in operation between Earth and Mars. It's operating on the International Space Station. It's part of the spacecraft that's in orbit around the Sun that's rendezvoused with two planets.

Vint Cerf

B

Transmission Reliability in Challenged Network Environments

THIS chapter represents a work started in the Google Summer of Code 2014 and describes the implementation of RFC-5326 (Licklider Transmission Protocol) as a simulation model for the ns-3 simulator. This work contributed towards improving support for DTN simulations in the ns-3 simulator.

The implications of this work towards the thesis were in part of technical nature. The knowledge acquired allowed performance improvements on several of the aeronautical models we had originally designed. On the other part, the generic model described herein can provide reliability to any scenario. Therefore, if point-to-point transmission reliability is required, this model can be used as part of the network stack described in Chapter 4. This chapter is a re-elaborated extension of the paper:

- R. Martínez-Vidal, T. R. Henderson, J. Borrell. *Implementation and Evaluation of Licklider Transmission Protocol (LTP) in ns-3.* In Proceedings of the 2015 Workshop on ns-3. ACM Press, (May 2015), pp. 75-82. [15]

B.1 Introduction

The current Internet architecture and protocols have proven to be successful for a wide range of communication environments, provided that a relatively stable link is provided between the communicating parties. However, environments with long round-trip times, intermittent connectivity or high transmission error rates severely limit the operation of protocols like the Transmission Control Protocol (TCP); this can result in a severe performance degradation [107].

As a result, a new framework for Delay- and Disruption-Tolerant Networking (DTN) was developed [40]. Among the new standards is a protocol called the Licklider Transmission Protocol (LTP), optimized for reliable data transport in DTN environments.

This chapter will present the design and implementation of LTP in the ns-3 network simulator. The main contribution is the verification (via interoperability testing, unit testing, and packet-level trace inspection) and validation (via comparison of simulation results with previously published results) of the LTP model. Our results should be fully reproducible using freely available software and the methodology outlined in the main body of this chapter.

The rest of this chapter is organized as follows: Section 2 provides an overview of DTNs and LTP, and a review of related work. Sections 3 describes the design and implementation of the LTP ns-3 module. Section 4 presents the testing approach and a simulation-based performance analysis. Section 5 presents results from interoperability testing with an existing implementation. Lastly, section 6 offers the conclusions.

B.2 Background

B.2.1 Delay Tolerant Networks

Delay Tolerant Networks are challenged networks where contacts among nodes are intermittent, and link performance is highly variable. Under these conditions, the existing Internet protocols are unsuitable. In those scenarios, an alternative DTN architecture [40] is used. The DTN architecture overcomes these difficulties by several means: It provides the Bundle Protocol [41] which uses a variable length data unit called "bundle" as a communication abstraction, an improved naming syntax, and a persistent store-and-forward function to allow long-term packet storage.

The basic design of the DTN architecture tried to cope with the issues found in the Interplanetary Internet (deep-space communication in high-delay links). However, its applicability can be extended to a wide variety of other scenarios such as military/tactical, underwater, or disaster response scenarios. The difficulty of deploying experiments in this kind of environments makes the use of simulation tools a necessity for the study of DTNs.

B.2.2 Licklider Transmission Protocol

The Licklider Transmission Protocol (LTP) [42] provides reliability over links characterized by long round-trip times (RTTs) and common connectivity losses. LTP is designed to be a convergence layer for the Bundle Protocol and acts as the reliable transport protocol in the DTN-protocol stack.

LTP is a transport protocol for long-delay or long interruption sessions. Under those conditions, the performance of other reliable transport protocols (such as TCP) degrades greatly and offers poor results. In these environments, communication opportunities are usually brief. For this reason, the protocol does not rely on handshakes as they might not complete before connectivity is lost. Instead, LTP achieves reliability by performing Automatic Repeat reQuest (ARQ) of data transmissions by requesting selective-acknowledgment of reception reports. In addition, it does not include any flow control or congestion control mechanism as the impact on performance in a long-delay environment may be huge.

LTP considers that each transmission session is composed of two parts: a red part, which requires acknowledgment of reception and is subject to a retransmission otherwise. Followed by a green part, whose transmission is attempted but not guaranteed. This distinction allows the protocol to support the conveyance of reliable and unacknowledged blocks of data within the same session.

In order to optimize the link usage, LTP relies on operating system "link state cues", which provide information about the state of the link (including the availability of the peer). Additionally, it uses coarse round-trip time (RTT) estimates to synchronize its retransmission mechanism. Another aspect in which LTP differs from other protocols is that transmission sessions are half-duplex. This feature derives from the fact that deep-space links are often unidirectional.

B.2.3 Related Work

Although DTN is a heavily researched topic, we were not able to find many previous works focusing on LTP performance evaluation or LTP simulation models. Much of the DTN published research focuses on layers above the DTN transport layer.

We found only one other simulation model for LTP, written for the OMNeT++ simulator. The model, written by Stefan Fisches, was uploaded to GitHub in 2009 [108], but does not seem to have been maintained since, and we could not find a corresponding publication. DTN simulation models for ns-3 include a Bundle Protocol implementation by Dizhi Zhou [100] and models for both ns-2 and ns-3 by Jani Lakkakorpi to study Epidemic Routing, Spray and Wait, and DTN Congestion Control [109], but neither project covers LTP. The Opportunistic Network Environment (ONE) simulator [110] is a frequently cited Java-based DTN simulator that focuses on routing and application protocols but does not appear to have an LTP model.

Researchers conducting performance evaluation of LTP have generally relied on operating a DTN implementation with LTP on real or virtual machines connected by simulated links. Examples of this type of study include the work by Amondaray and Pascual [111], using the DTN Reference Implementation within User Mode Linux (UML) virtualization, and the work of the N4C project that developed a DTN simulator combining LXC containers, an ns-3 simulated physical layer, and ns-3 PyViz visualization [112].

An analytical model of LTP developed by Wang et al. [107] is validated in that paper against experiments using the Interplanetary Overlay Network (ION) implementation of LTP [113] operating over simulated channels. The work by Wang et al [107] focuses on Bundle Protocol performance over different DTN convergence layers, using Linux PCs running the NASA ION implementation of LTP, and a space link simulator.

In Section 4 of this chapter, we compare our simulation results to those of [107], and in Section 5 of this chapter, we use a different LTP implementation for validation of our model.

B.3 Design and Implementation

In this section, we describe the major design decisions made for the development of the LTP model, along with the several challenges that we encountered during

its implementation. For details in class design, please refer to the Doxygen documentation of the module cited in section B.6.

B.3.1 Design Choices

Lower-Layer Interaction

LTP may be deployed over UDP or directly over the datalink layer. We choose to implement it over UDP because we did not have a DTN-oriented link layer model, and running over UDP seemed to give greatest flexibility to combine it with other models in ns-3.

DTN protocols such as Bundle or LTP require the services of a Convergence Layer Adapter (CLA) to interact with a lower layer link or protocol. In our model, we provide a CLA for UDP (*'LtpUdpConvergenceLayerAdapter'*) which offers the following services:

- Maps outgoing LTP segments into UDP socket send() operations.
- Maps incoming LTP segments into receive() operations using the IANA-assigned listening port of 1113 (deep-space).
- Allows the LTP implementation to determine the MTU that will avoid fragmentation.
- Provides link-state notifications.

Link state cues offer information about the state of the link (up/down) and notify certain events of the transmission session, such as the transmission of specific control segments. Each LTP protocol instance is connected to one or more CLAs that provide the link state-cue services. Link up and down information is provided as part of the API of the CLA. Control segment transmission notifications are provided through the use of the callback system.

LTP endpoints are identified by their LTP Engine Ids, this requires the design of an address resolution system. We implement it as an static table (*'LtpIpResolutionTable'*) which contains direct mappings between LTP Engine IDs and IPv4/IPv6 addresses. This class is used by the UDP CLA during the transmission procedure to perform the conversion transparently for the LTP engine. The mappings are predefined during the installation procedure of the system.

Higher-Layer Interaction

The upper-level service that uses the LTP engine to transfer data is referred as a Client Service Instance (CSI). Communication between the LTP engine and the CSI can happen in two ways. The client service makes requests to the LTP engine (start or cancel transmission) and the LTP engine issues back notifications (to report certain events or to hand over received data). The design of these communications is a local implementation matter, in our ns-3 module:

Requests are provided as methods of the *LtpProtocol* object that needs to be instantiated and invoked by the client service instance. Notifications are provided through the use of callback functions; they are usually used on the receiving end of the communication session. Client service instances must register with a corresponding LTP engine and implement a specific function prototype used by LTP to report status notifications and deliver incoming data.

B.3.2 Modeling Assumptions

The ns-3 LTP model has been simplified in several aspects. First, there are some functionalities that have been modeled with low fidelity; this is the case for the provided link state cues. LTP deployment scenario assumes that it will be used by nodes that have scheduled losses of connectivity due to specific mobility patterns (such is the case for satellites). The LTP specification assumes that both the local and the remote peer's link operation schedules are known by the operating system, this knowledge is offered through the link state-cues which allows maximum optimization of the link-usage. In the case of our model, link state-cues only provide information on the local link state. We consider the other functionality to be a specific use case of the deployment scenario and thus being outside of the scope of a general model. A second aspect we simplified was the control system mechanism, LTP provides two types of control segments: reports and cancellations. The cancellation control sequence has not been modeled as it was not critical for the data transmission procedure. Finally, another aspect in which this model is lacking is that it only supports a single concurrent transmission session per destination. The ns-3 LTP model can be seen in Figure B.1, this figure depicts the protocol components and its interaction with each other.

B.3.3 Implementation Issues

We addressed an ns-3 implementation challenge due to the lack of native support for path MTU discovery in the UDP socket API. In real implementations, an application running over UDP can find the MTU by opening a test socket to the system, connecting it to a destination address, and running the getsockopt(IP_MTU) on the connected socket. We are interested in finding the effective path MTU that avoids fragmentation. Since this method was not supported, we modified the existing GetTxAvailable() method of the UDP socket to support an approximation. Specifically, we changed this method to iterate over the available interfaces to find the smallest device (not path) MTU, and return the minimum one found (as a conservative estimate).

Another issue that we encountered was that the version of ns-3 used in this study does not easily support manual insertion of IP neighbor table caches or entries. As a result, when experimenting in ns-3 with large propagation delays, UDP packets can be lost if the LTP protocol instance sends more than three packets (the default size of the ARP cache pending queue size) while MAC address resolution is taking place. To support experiments with small round trip delays such as on the order of hundreds of milliseconds (Section 5), we started the experiment by sending a ping to seed the ARP cache, and attached a callback to the ping application's trace source. The callback, when triggered, launched our LTP sending process. This method does not work as well if the propagation delays grow above 1 second round trip time, since the ns-3 ping application does not permit one to extend the wait timeout. For experiments with very large transmission delays, unless the IP interface object is provided with a pre-built neighbor cache, one may need to build in enough start up time to ensure that packets will not be blocked by address resolution, and the value of the ArpCache::WaitReplyTimeout attribute must be raised beyond 1 second.

B.4 Model Evaluation

The implementation of the protocol was tested on a simple two-node topology, because this is sufficient to exercise the retransmission mechanisms. The network uses point to point links, a MTU of 1500 bytes, and a channel configured to emulate the characteristics of deep space communications. We use long channel delays and attach an error model to the *NetDevice* corresponding to each node.We selected a 1500 byte MTU because it is likely the MTU that

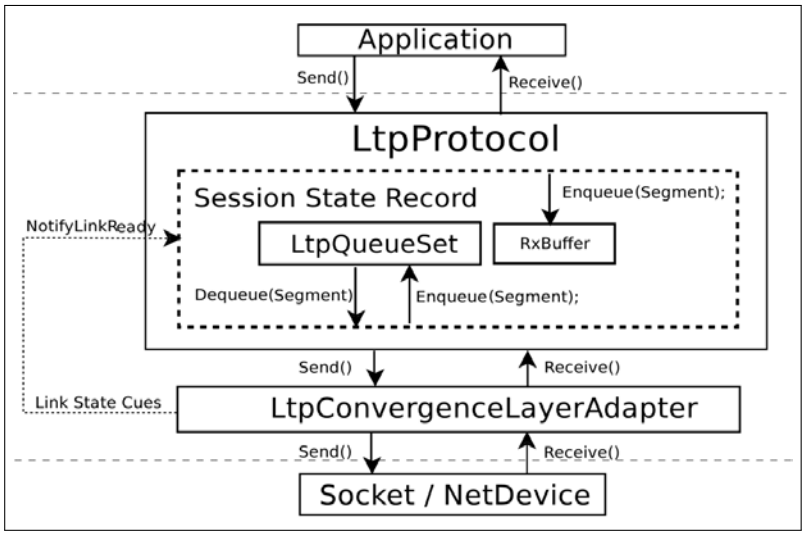


Figure B.1: LTP module interaction diagram.

was used in previous LTP evaluation studies that used virtual machines over simulated links; the actual MTUs used in previous studies do not seem to have been documented. Performance trends shouldn't be very sensitive to this parameter, although the MTU choice affects the packet error ratio when bit errors are injected to the channel.

B.4.1 Testing Approach

Now we proceed to describe the tests performed and the purpose of each test. The LTP model provides two main *test-suites*:

- The **ltp-protocol** test suite contains extensive unit tests for the different auxiliary data structures and objects provided in the model.
- The **ltp-channel-loss** test suite, seeks to guarantee the correct behavior of the protocol implementation by forcing certain conditions that require data retransmission.

Retransmission Tests

In these tests, the LTP protocol is used to transfer a data block of 5000 bytes between the two nodes. In order to transfer this block, the protocol is forced to split the block into several segments (of size smaller than the MTU) which are then re-assembled by the instance on the receiving side.

Additionally, it is possible to specify which part of that block should be transmitted reliably, this implies that the parts to be sent unreliably are not subject to data retransmission. We study three main cases: First, the mixed case in which some segments are to be transmitted reliably while others not. Second, the full red block case in which all segments need to be transmitted reliably, and last, the full green block when all segment may be lost without retransmissions.

In Figure B.2 we can see a sequence diagram showing the pattern of transmitted segments for the basic case, transmission of a 5000 byte block containing both red and green data.This represents the protocol behavior under lossless operating conditions. We set-up a *ReceiverListErrorModel* in both nodes to force controlled segment losses. Each test-case contains a list of segments to be lost, including both data and control segments, we test for include losses of single segments, multiple segments and a variety of possible combinations, seeking to test all the possible eventualities that may occur.

These segment losses force the protocol to perform retransmissions in order to successfully deliver all the parts of the block marked as red data. A test is successfully passed by asserting that the re-assembled data at the end of the transmission session corresponds to the amount of data that was requested to be sent reliably.

Table B.1 shows the different test combinations. The first column defines the number of bytes that need to be sent reliably. The second and third columns contain the ID of the packets that are lost, and each ID follows the same ordering as the segments shown in Figure B.2. The last column displays the number of successfully received bytes at application level. Packet label coloring defines whether they were red, green, or control (black) segments. In the event of a loss of a segment (data or control) pertaining to the reliable transmission (rows 1-6), the appropriate control mechanism kicks in to guarantee the transmission of all mandatory data. In the case, that the lost segment corresponds to the green section of the block no retransmission is triggered, and the total amount of received bytes decreases (rows 7-8).Combinations of both cases in which only

	Red Data Length	Lost packet ID (receiver)	Lost report segment (sender)	Rx Bytes
1	1500			5000
2	1500	0		5000
3	1500	1		5000
4	1500	0 , 1		5000
5	1500		✓	5000
6	1500	5	✓	5000
7	1500	2		3539
8	1500	2 , 3		2078
9	1500	0 , 3		3539
10	1500	1 , 3		3539
11	1500	0 , 1 , 2 , 3		2078
12	1500	2	✓	3539
13	1500	4	✓	4422
14	5000			5000
15	5000	0 , 1		5000
16	0			5000
17	0	1 , 2		2078

Table B.1: Retransmission tests using a 5000 bytes data block

red segments are subject to retransmissions can be seen in rows 9 to 13. Additionally, the individual cases in which the whole block is entirely red or green can be seen in rows 14 to 17.

B.4.2 Transmission Scenarios

In this section, we review the examples provided by this model and summarize the main functionalities shown by them. The model includes two examples:

Basic transmission: Is a transmission scenario provided by the script *ltp-protocol-example*. This script shows how to use provided helper in order to set up the basic two-node topology and transfer a 5000 bytes data block between two nodes using the LTP protocol.

Long haul transmission: Is a transmission scenario provided by the script *ltp-long-transmission-example* which represents a common application scenario where the LTP protocol is used to transfer 10 MB of data between the Earth and Mars. It defines a channel with high transmission delay (750 seconds, based on the average distance between both celestial bodies), data rate of 500kbps, and common packet losses using a *RateErrorModel*.

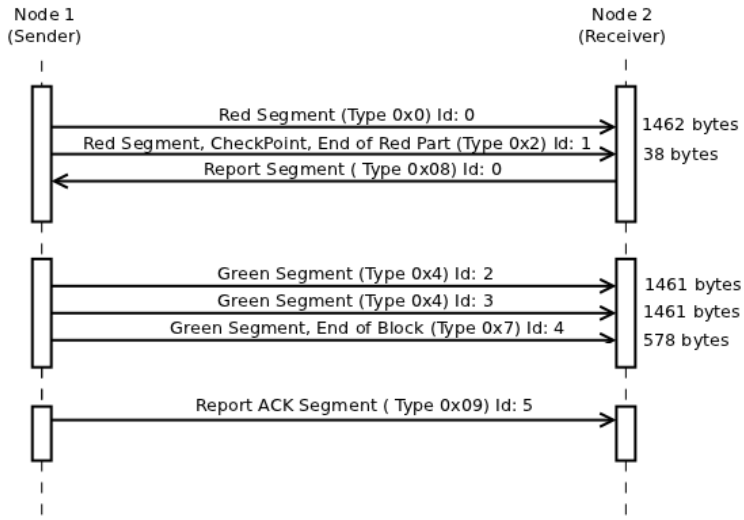


Figure B.2: LTP transmission sequence under lossless operating conditions (part of the block is to be sent unreliably as green segments)

As described in section B.3 the LTP is used by a Client Service Instance, which is a software entity, usually an application or a higher-layer protocol (commonly the Bundle protocol). This example defines a class *ClientServiceInstance* inherited from *ns3::Application*, which is installed on the nodes and implements the required methods to send and receive requests from its corresponding LTP engine.

The ClientServiceInstance successively performs requests to transfer 100 KB data blocks to the LTP engine. For each request, the full block is demanded to be sent reliably, upon successfully transmitting the demanded block the LTP engine notifies the Client Service Instance, which then issues the next block until all 10 MB have been transmitted.

B.4.3 Performance Evaluation

In this section, we discuss the experimental results of the performance evaluation of the model. Our performance analysis is based on similar experiments as those performed in real evaluations of LTP such as [107].

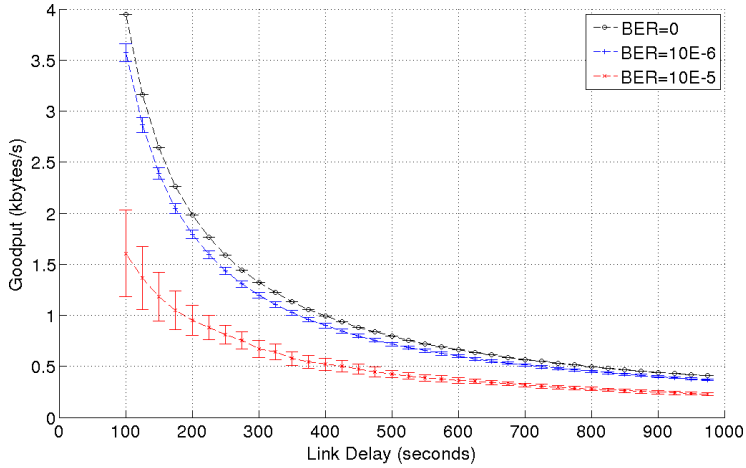


Figure B.3: Impact of increased channel noise on goodput performance with varying link delays

For this performance analysis we use the previously described long haul transmission scenario and study it under changing channel conditions. We analyze the network performance in function of the measured goodput at application level, which is defined as the rate of useful bytes delivered per unit of time, computed between the transmissions of the first transmitted byte and the last received byte.

Impact of BER on performance

In this section, we study the impact of BER on performance for long channel-delays. We consider one-way channel delays that range from 100 to 1000 seconds, and three byte error rates - 0, $10^{-6}, 10^{-5}$ - to represent: an error-free channel, a moderate error rate (common space link conditions [114]), and the maximum acceptable error rate.

In Figure B.3, we display the goodput for each channel delay and plot the values achieved by each of the analyzed error rates. Each point shows the mean value (with an error bar representing the standard deviation) over a sample of 10 simulation runs. The decrease in goodput follows a smooth curve. In the case of low channel delays, there is a clear decrease but for higher ones it

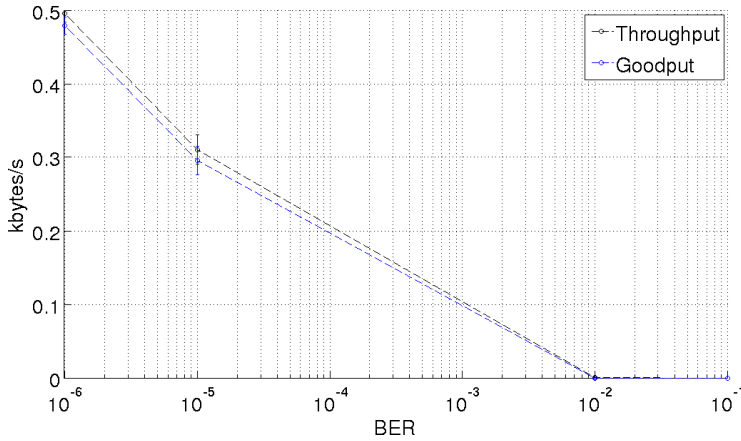


Figure B.4: Throughput and Goodput comparison with varying channel noise and fixed link delay (750s)

starts to flatten. This behaviour shows that the increase of channel delay has a direct impact in goodput as it directly affects the duration of the transmission. We can also observe that the three error rates follow similar distributions but at different goodput levels. Another fact is that for high channel-delays the goodput of different error-rates converges to similar values.

Figure B.4 depicts a comparison between throughput and goodput in varying channel noise while keeping a fixed link delay. As we can see a higher channel noise triggers additional retransmissions resulting in more protocol overhead and a decreased goodput. It is also observable that the difference between throughput and goodput widens as the error rate increases (10e-5, 10e-4, 10e-3) but it becomes closer again for higher error rates (10e-2, 0.1), as the number of losses becomes bigger their impact can also be observed in the throughput.

Impact of inaccurate RTT estimation

In this experiment, we analyze how an imprecise RTT estimation affects the goodput. As mentioned before, LTP relies in coarse RTT estimation to time retransmission of unacknowledged segments. This RTT estimation consists on doubling the known one-way delay and adding an arbitrary margin (corresponding to local delays and other factors). In these experiments, we use multiples of

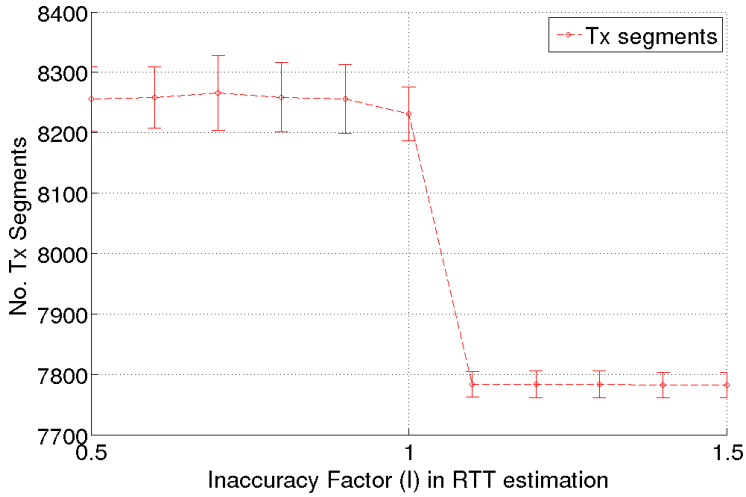


Figure B.5: Impact of inaccurate round-trip estimation

the one-way delay (defined by a factor I) to cause inaccuracy on the estimation of the RTT.

$$RTT = 2 * OneWayDelay * I + (ProcessingDelay + Margin)$$

Now, we observe the resulting performance impact of these changes. In Figure B.5 we show the number of packets transmitted for each RTT estimation. RTT underestimation triggers redundant retransmissions; this can be seen in the increased number of transmitted segments. However, the impact is not as high as would be expected as only RP and CP segments (a reduced fraction of all the segments) are subject to this kind of retransmission. On the other hand, overestimation of the RTT assures that there will be no spurious retransmissions. Consequently, the number of retransmissions decreases but latency grows, and the data transfer takes longer.

LTP/TCP performance comparison

This experiment reproduces the configuration used in [107]: channel delay from 1.5 to 5 seconds, three error rates (0, 10e-6, and 10e-5), data rate of 115Kbps, and 1MB of transferred data. Some simplifications were made; originally the

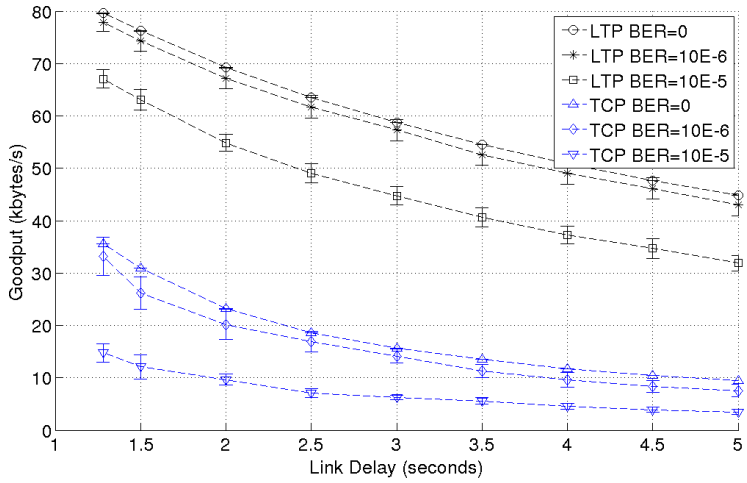


Figure B.6: Goodput comparison between LTP and TCP with varying link delay

transfer was performed between two end-points via an intermediate relay node, first over BP/LTPCL/LTP/UDP and then over BP/TCPCL/TCP. In our case, we perform a direct transfer between the two end-points over LTP/UDP and TCP. The reason being that interaction between BP and LTP in ns-3 is not yet supported.

In Figure B.6, we show the goodput performance comparison between LTP and TCP. This figure shows the several error rates and how the goodput is influenced by the increase of channel delay. The TCP transmission rate degrades significantly for the provided channel delays, and we can see how LTP shows a clear performance advantage for all link delay levels studied. LTP roughly doubles the goodput of TCP for delays higher than 1500 ms in an error-free channel and this trend becomes even more apparent for higher error rates as TCP degrades even further.

We obtain higher goodput levels for both protocols, and the impact of error rate in our TCP implementation is not as significant as that shown by [107]. These differences are probably the result of the simplifications made to the experiment, and to differences in the TCP models. Nevertheless, the main trends are consistent with those shown by the previous work, illustrating that

LTP is more tolerant of transmission errors and link delay than is TCP.

B.5 Experimental Validation

In this section we validate the ns-3 LTP model against an existing implementation of the protocol. First, in Section B.5.1, we summarize available implementations and describe the one selected for experiments in this chapter. Section B.5.2 describes the methodology used for validation, and present the results of interoperability testing, looking at basic loss-free operation of a single file transfer, operation of different combinations of file sizes and different red and green data portions, and operation in the presence of packet losses. The results in this section prove that the ns-3 LTP implementation can perform basic data block transfers over lossy channels, interoperate with an independent implementation, and provide comparable performance to that implementation.

B.5.1 LTP Implementations

We found three available open-source implementations of LTP, with each one providing convergence layer adapters for different underlying layers:

- LTPlib [115] (Trinity College, Dublin) works over UDP.
- LTP-RI [116] (Ohio University) works over UDP, SerialPort, and Bluetooth
- ION-DTN [113] (NASA's Jet Propulsion Lab) works over UDP, DCCP, and AOS.

We evaluate our model against LTPlib operating over UDP. This provides a similar approach as the one used in our model tests. We selected LTPlib because we found it to be easier to configure than the other options. LTPlib was written by Steven Farrell; the implementation is in C++, and options exist to make it interoperate with the ION-DTN implementation.

LTPlib provides a number of command-line options, and some are mandatory to set the role of the implementation to client or server (or router), and to configure the addresses in use. One option that has bearing on our experiments is the "wait time" that the client will wait for the red part transfer to succeed before terminating the daemon. We extended this value to 30 seconds from the

default of 10 seconds. There appears to be no option, however, to configure the round trip time, and inspection of the source code confirmed that LTPlib assumes a 200 ms round trip time (used in determining the need to retransmit data). To avoid possible spurious retransmissions due to additional small delays other than propagation delays, we configured the round trip delay with a value 10% smaller than 200 ms (i.e., 180 ms).

B.5.2 Methodology and Results

This section provides an overview of the emulation methodology; more details for reproducibility are described in the Appendix. This section also provides emulation results.

In order to validate the correctness and interoperability of our model, we use the Common Open Research Emulator (CORE) [117] to set up a network scenario where nodes exchange data using a real implementation of LTP and our ns-3 model. CORE is a framework for network emulation experiments, providing an intuitive canvas-based graphical user interface for topology creation and run-time introspection of nodes. The backend of CORE coordinates the instantiation and configuration of Linux network namespace containers for the network nodes, and the GUI provides access to the containers at runtime. The links between nodes are instantiated by Linux bridges with link effects such as packet loss and delay provided by the *netem* tool. We used both the GUI mode and the Python scripting (non-GUI) mode of CORE to perform the experiments.

In our experiments, the network topology consists of two nodes connected by a point-to-point link, with the nodes assigned the addresses 10.0.0.1/24 and 10.0.0.2/24 respectively. The nodes were interconnected by a simple link with configurable delay and packet error ratio (all link effects are applied to both directions of the link). A reference scenario is shown in Fig. B.7.

In this scenario, node 2 performs the role of a server and runs the real implementation LTPLib. Node acts as the client and will alternatively run LTPlib and ns-3 in emulation mode (using the file descriptor NetDevice and the real-time simulator).

Interoperability has been tested assuming two different working conditions: error-free channel and lossy channel. Each case follows a different testing approach that is explained in the next sections.

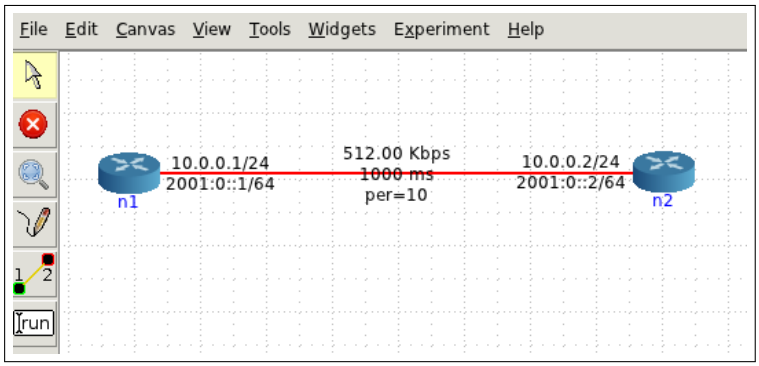


Figure B.7: CORE GUI canvas environment. Network topology of the validation scenario.

Error free operation

We qualitatively compared the operation of the two configurations (ns-3 client to LTPlib server, and LTPlib client to LTPlib server) by repeating several file transfers, performing packet captures using the tcpdump utility, and inspecting the captures using the Wireshark tool that is able to parse the LTP headers.

In both cases, we sent a text file of 102400 bytes from client to server and observed the packet exchanges. Both implementations performed similarly, breaking the block of data into roughly 70 UDP datagrams, and marking the last segment as the end of red data, with checkpoint needed. This last data segment from the client elicited a Report Segment from the server confirming the receptions, which generated a Report Acknowledgment from the client to the server. Finally, the server would send a Cancel Segment about 1.5 seconds after receiving the Report Acknowledgment, and no further segments were sent. In cases in which segments are lost, the Report Segment reports on the missing red data and the client will retransmit the missing data.

The transfers differed in the following aspects.

- ns-3 used full-sized IP datagrams (1500 bytes) while LTPlib typically used slightly smaller datagrams (1485 bytes);
- in the ns-3 experiments, the Session ID of the originator, and the Client Service ID, were both set to zero, while these were populated with non-zero values by LTPlib;

Block size	Red size	LTPlib (segment type)	ns-3 (segment type)	Rx
500	500	3, RS, RAS	3, RS, RAS	✓
5000	5000	0, 0, 0, 3, RS, RAS	0, 0, 0, 3, RS, RAS	✓
500	200	3, RS, RAS	2, 7, RS, RAS	✓
5000	2000	0, 2, 4, 7, RS, RAS	0, 2, 4, 4, 7, RS, RAS	✓
500	0	7, CS	7	✓
5000	0	4,4,4,7, CS	4,4,4,7	✓

Table B.2: Interoperability tests over error-free channel

- the pacing of segments by the clients differed slightly. ns-3 regularly spaced segments by 10 ms intervals, resulting in roughly 0.7 seconds to send the 70 full-sized segments in the block. The pacing is controlled by an ns-3 attribute "LocalProcessingDelays" in class LtpProtocol. LTPlib did not send according to as regular of a schedule, but took about 0.5 seconds longer to send its 71 segments of data;

- LTPlib used a different ephemeral UDP source port for each segment, while ns-3 used the same ephemeral port consistently. Both used port 1113 as the destination port.

Protocol nominal operating conditions

In this section, we present test results on the interoperability of different data exchanges over an error-free channel in which all communication steps are successfully completed. To perform these tests we again used both LTPlib and ns-3 as clients communicating with an LTPlib-based server. We evaluated the interoperability of both implementations by checking that the server is capable of reassembling the data and by comparing the segments generated by each implementation.

We performed tests using multiple configurations: First, we used two different block sizes, 500 bytes and 5000 bytes, the latter being bigger than the MTU and requiring segmentation at the LTP layer into multiple UDP datagrams. Second, we varied the reliability of the transmission to study three cases: full-red block , full-green block, and a block mixed with red- and green-data.

In Table B.2, we show the several testing combinations performed; the comparison is performed by showing the number and type of packets generated by

each application (packet types are listed in section B.2). In all cases, the server received and reassembled the block successfully. However, we observed some slight differences between the behavior of the implementations. First, in the event of a mixed block (rows 3-4), LTPLib uses fewer segments to perform the same transmission. This behavior is due to LTPlib taking data corresponding to the green part data and merging it together with a red segment (that otherwise would be of lower size than the MTU). Second, in the case of a full green part (rows 5-6), LTPLib sends an additional segment to cancel the transmission session on the server side (CS). The LTP specification (RFC 5326) describes the first case as a possible bandwidth optimization; the second one seems to be an implementation choice that does not interfere with the operation of the protocol.

Protocol retransmission capabilities

In this section, we present interoperability testing results over a lossy channel in which some segments may be lost, triggering the retransmission mechanisms of the protocol.

We performed multiple tests with varying error rates (from 0 to 10%). We again used two client configurations, LTPlib and ns-3, and repeated each experiment in each configuration ten times. We first confirmed that all transmissions eventually succeeded despite retransmissions, and then we compiled statistics on the overall data transfer delay for each trial. From the vantage point of the channel (a packet trace conducted on the bridge device), we defined the transfer delay as the difference in timestamp between the last segment sent from client to server (the final Report Acknowledgment) and the first segment sent from client to server.

In Table B.3, we list the sample mean and sample standard deviation of the ten trials for each configuration. For error-free operation (the first row), it can be observed that the delay for the ns-3 client was roughly half a second less, on average, than the delay for the LTPlib client. As discussed above, this difference is due to the difference between implementations in how they each pace their transmissions. Taking this half second advantage for ns-3 into account, the two implementations otherwise perform roughly the same until the 7.5% and 10% packet error ratio cases, for which the variability in performance was generally larger. We observed that despite being configured for a low one-way delay of 100ms, the LTPlib was very conservative (up to 10 seconds) in retransmitting

data, leading to larger latencies when a red segment with checkpoint was lost.

We then reversed the configuration to operate ns-3 as the LTP server, and LTPlib as the LTP client. This configuration was used mainly to confirm that ns-3 could successfully interoperate also as a server, but we also gathered data similar to that shown in Table B.3, for packet error ratios of 0, 5%, and 10%. In the error-free case, the performance was about the same as the LTPlib client to LTPlib server case (about 1.5 seconds), which is expected since the LTPlib retransmission timeouts dominate the performance results. The results with error rates of 5% (4.03/4.72 sec) and 10% (6.41/8.54 sec) also show a high variability and are within a standard deviation of the results collected with the LTPlib server. We also test robustness to mobility; we ran a simulation by disconnecting the link for a long period and then reconnecting. We observed a successful data transfer, although some data redundancy was present and not yet debugged.

B.6 Reproducibility

This section provides additional details about reproducing the experiments described herein.

The version of ns-3 used in the tests was changeset 10879 of the repository found at `http://code.nsnam.org/rmartinez/ns-3-dev-ltp`. This was based on the ns-3.19 release, with modifications outside of LTP in two places. First, the GetTxAvailable () method of class UdpSocket was modified to return the outbound MTU (see issue 2064 of the ns-3 bug tracker). Second, a bug in a length calculation of the FdNetDevice (see issue 2063 of the ns-3 bug tracker) was patched. Another required feature is the Self-Delimiting Numeric Values (SDNV) encoding support, borrowed and finalized from the Bundle Protocol implementation [100].

B.6.1 Simulations

The performance evaluation of the protocol was performed using the script found in *src/ltp-protocol/ltp-long-transmi ssion-example.cc*. This script may be configured using com mand-line arguments to select configurations for each experiment. The results displayed in Figures B.3, and B.6 may be reproduced specifying the arguments: –BER, –channelDelay, –totalBytes, and –dataRate, and setting them to values analogous to those shown on the axis of each figure.

Additionally, for Fig B.3 the argument totalBytes must be set to 10485760 and dataRate to 500Kbps. In the case of Fig. B.6 the argument totalBytes must be set to 1048576 and dataRate to 115200bps. The TCP data used for comparison is obtained using *src/test/ns3tcp/ns3tcp-state-test-suite.cc* as a base. The network topology was replaced by two nodes connected by a point-to-point link. The properties of the link were set to the same configuration as that of its LTP counterpart. Finally, the writing size used by the sender application was equal to the size of the LTP blocks (100 KB). Finally, the data in Table B.1 is a representation of the test vectors found in *src/ltp-protocol/tests/ltp-channel-loss.cc* and may be replicated executing the test-suite.

B.6.2 Emulations

Emulations were performed using the CORE emulator, on different machines. Experiments concerning performance over a lossy channel were conducted manually using version 4.7 of the CORE tool on a Fedora Core 20 Linux server. Experiments with different block sizes and different combinations of red and green data were automated by a Python program that used the CORE Python API but with similar topology configuration, and were conducted using version 4.3 of the CORE tool on an Ubuntu 11.04 server. Packet captures were performed on the underlying bridge device created by the CORE emulation.

When using CORE, we set the `eth0` interface to promiscuous mode, and also disabled generic segmentation offload (GS0), and transmit and receive checksum offloading, on the interfaces, as follows:

```
ethtool -K eth0 gso off
ethtool -offload eth0 rx off tx off
```

In addition, when running ns-3 within a container in emulation mode, the IP address on the container virtual interface was deleted to avoid having the container respond to address resolution requests.

The version of LTPlib was the version fetched from the website as of early 2015. The command for invoking the server was `ltpd -m S -L 10.0.0.2:1113 -S 10.0.0.1:1113 -o ltpd.server.out` and for the client `ltpd -v -w 20 -i a.txt -m C -D 10.0.0.2:1113 -S 10.0.0.1:1113` where the `txt` file contained the character 'A' repeated in 102400 bytes of data.

The script found in `src/ltp-protocol/emu/ltp-emu.cc` was used for ns-3, and the program was run with a command-line argument to select different test numbers corresponding to the Table B.2 configurations. To reproduce these

tests with the ns-3 client, the argument `-testNum` with values ranging from 1 to 6 will reproduce the six rows of the table, respectively. To reproduce the same tests with LTPlib, the basic client command shown above is used and modified with an extra argument `-R` whenever the data must be split between red and green parts, and modified with a different sized input file named `ltpd.in` (500 or 5000 bytes) as follows. Tests 1, 3, and 5, used an input file of size 500 bytes, while the others used a file of 5000 bytes. Test 3 passed the argument `-R 200` to the `ltpd` program, while Test 4 passed `-R 2000`, and Tests 5 and 6 both passed `-R 0`.

Finally, the data in Table 3 was gathered as follows. The ns-3 client program `src/ltp-protocol/emu/ltp-emu.cc` was run with no arguments, and the LTPlib client and server commands were as listed above. For each configuration, ten trials were conducted and the time difference for each trial between the sending of the first segment and the receipt of the Report Segment confirming successful delivery was measured. The sample mean and standard deviation computed on this time interval across all trials. The CORE emulation provided the random error model, so each run faced a different (non-repeatable) set of lost packets. To configure the experiment in the reverse direction with ns-3 as the server and LTPlib as the client, the ns-3 program was run on virtual node 2 with the same program, but with the `-server=true` argument passed to the program.

B.7 Conclusion

This chapter offers a ns-3 model of the LTP protocol; our implementation has been verified by means of extensive tests and interoperability with an existing implementation, and performance has been validated by comparing results with previously published work. The most important aspects of LTP that our implementation tried to model with high fidelity were the structure of the data packets, and the transmission/retransmission procedure sequence, which is the main functionality of this reliable transport protocol.

Overall, the inclusion of this new model contributes greatly to improve ns-3 support for DTN simulations. However, there are still several issues that require attention before ns-3 can be used as a serious alternative for DTN simulations. First, we need to offer a fully operational DTN stack by integrating the Bundle and LTP protocols together. Secondly, there is the need to provide models of

PER	LTPlib client(mean/std)	ns-3 client(mean/std)
0	1.44/0.025 (sec)	0.92/0.0056 (sec)
1%	1.58/0.14 (sec)	1.03/0.11 (sec)
2%	1.61/0.12 (sec)	1.39/0.99 (sec)
5%	2.63/2.85 (sec)	1.84/1.35 (sec)
7.5%	1.78/0.11 (sec)	2.14/1.52 (sec)
10%	8.64/4.59 (sec)	3.21/2.56 (sec)

Table B.3: Comparison between an LTPlib client and ns-3 client of block transfer times over lossy channels. The table lists sample mean and standard deviation of ten 100KB transfers to an LTPlib server over a channel with a varying packet error ratio and a 180ms round trip time.

the common baseline routing protocols used in DTNs.

The implementation of the LTP cancellation control segments has to be considered, adding multiple concurrent transmission capabilities, and running more validation tests involving high-delay transmissions and high-error rates. Additionally, follow-up work has to include the integration of LTP with the existing ns-3 Bundle protocol implementation, for inclusion in an upcoming DTN module.

Bibliography

[1] Gogo-Inflight, "Gogo inflight business aviation: broadwand and voice," 2015. "http://www.gogoair.com/".

[2] GEE, "Global eagle entertainment: Fast inflight broadband for the modern traveler," 2015. http://www.geemedia.com/.

[3] Inmarsat, "Inmarsat business aviation: passenger, operational, and safety communications," 2015. "http://www.inmarsat.com/aviation/business-aviation/".

[4] E. Sakhaee, A. Jamalipour, and N. Kato, "Aeronautical ad hoc networks," in *Wireless Communications and Networking Conference, 2006. WCNC 2006. IEEE*, vol. 1, pp. 246–251, 2006.

[5] E. Sakhaee and A. Jamalipour, "The global in-flight internet," *Selected Areas in Communications, IEEE Journal on*, vol. 24, pp. 1748–1757, Sept 2006.

[6] "Proses, network protocols for the single european sky," tech. rep., Spanish Ministerio de Indústria, Turismo y Comercio, 2009.

[7] N. Giuditta, S. Robles, A. Viguria, S. Castillo, M. Cordero, and L. Fernández., "PROSES - Network Communications for the Future European ATM system," in *Proceedings of 1st International Conference on Application and Theory of Automation in Command and Control Systems, pp. 87-90*, 2011.

[8] R. Martínez-Vidal, S. Castillo-Pérez, S. Robles, A. Sánchez-Carmona, J. Borrell, M. Cordero, A. Viguria, and N. Giuditta, "Mobile-agent based

delay-tolerant network architecture for non-critical aeronautical data communications," in *10th International Symposium on Distributed Computing and Artificial Intelligence*, pp. 513–520, 2013.

[9] S. Ahmed and S. K. Salil, "Characterization of a large-scale delay tolerant network," in *Proceedings of the 2010 IEEE 35th Conference on Local Computer Networks*, LCN '10, (Washington, DC, USA), pp. 56–63, IEEE Computer Society, 2010.

[10] C. Xia, D. Liang, H. Wang, M. Luo, and W. Lv, "Characterization and modeling in large-scale urban dtns," in *Local Computer Networks (LCN), 2012 IEEE 37th Conference on*, pp. 352–359, 2012.

[11] R. Martínez-Vidal, M. de Toro, R. Martí, and J. Borrell, "Esquema de gestión de claves criptográficas tolerante a retrasos e interrupciones en entornos aeronáuticos," in *XII Reunión Española de Criptología y Seguridad de la Información*, RECSI 2012, pp. 181–186, Mondragon Unibertsitatea, 2012.

[12] R. Martínez-Vidal, R. Martí, and J. Borrell, "Characterization of a transoceanic aircraft delay tolerant network," in *Proceedings of the 38th Annual IEEE Conference on Local Computer Networks*, pp. 590–597, IEEE Computer Society, 2013.

[13] R. Martínez-Vidal, R. Martí, and J. Borrell, "Analyzing information propagation in a transoceanic aircraft delay tolerant network," in *Proceedings of the 39th Annual IEEE Conference on Local Computer Networks*, pp. 116–123, IEEE Computer Society, 2014.

[14] R. Martínez-Vidal, R. Martí, C. J. Sreenan, and J. Borrell, "Methodological evaluation of architectural alternatives for an aeronautical delay tolerant network," *Pervasive and Mobile Computing*, 2015.

[15] R. Martínez-Vidal, T. R. Henderson, and J. Borrell, "Implementation and evaluation of licklider transmission protocol (ltp) in ns-3," in *Proceedings of the 2015 Workshop on Ns-3*, WNS3 '15, (New York, NY, USA), pp. 75–82, ACM, 2015.

[16] K. Karras, T. Kyritsis, M. Amirfeiz, and S. Baiotti, "Aeronautical mobile ad hoc networks," in *Wireless Conference, 2008. EW 2008. 14th European*, pp. 1–6, June 2008.

[17] H. Li, B. Yang, C. Chen, and X. Guan, "Connectivity of aeronautical ad hoc networks," in *GLOBECOM Workshops (GC Wkshps), 2010 IEEE*, pp. 1788–1792, 2010.

[18] R. Kingsbury, "Mobile ad hoc networks for oceanic aircraft communications," Master's thesis, Massachusetts Institute of Technology. Dept. of Aeronautics and Astronautics, 2009.

[19] D. Medina, F. Hoffmann, S. Ayaz, and C.-H. Rokitansky, "Feasibility of an aeronautical mobile ad hoc network over the north atlantic corridor," in *Sensor, Mesh and Ad Hoc Communications and Networks, 2008. SECON '08. 5th Annual IEEE Communications Society Conference on*, pp. 109–116, 2008.

[20] D. Medina, F. Hoffmann, S. Ayaz, and C.-H. Rokitansky, "Topology characterization of high density airspace aeronautical ad hoc networks," in *Mobile Ad Hoc and Sensor Systems, 2008. MASS 2008. 5th IEEE International Conference on*, pp. 295–304, Sept 2008.

[21] B. Newton, J. Aikat, and K. Jeffay, "Analysis of topology algorithms for commercial airborne networks," in *Network Protocols (ICNP), 2014 IEEE 22nd International Conference on*, pp. 368–373, Oct 2014.

[22] N. T. X. My, Y. Miyanaga, and C. Saivichit, "Connectivity analytical modelling for a single flight path ad hoc aeronautical network," in *Electrical Engineering/Electronics Computer Telecommunications and Information Technology (ECTI-CON), 2010 International Conference on*, pp. 51–55, 2010.

[23] M. X. Cheng and Y. J. Zhao, "Connectivity of ad hoc networks for advanced air traffic management," *Journal of Aerospace Computing, Information, and Communication*, vol. 1, pp. 225–238, 2004.

[24] T. Clausen and P. Jacquet, "Optimized Link State Routing Protocol (OLSR)." RFC 3626 (Experimental), Oct. 2003.

[25] C. E. Perkins and P. Bhagwat, "Highly dynamic destination-sequenced distance-vector routing (dsdv) for mobile computers," *SIGCOMM Comput. Commun. Rev.*, vol. 24, pp. 234–244, Oct. 1994.

[26] C. Perkins, E. Belding-Royer, and S. Das, "Ad hoc On-Demand Distance Vector (AODV) Routing." RFC 3561 (Experimental), July 2003.

[27] D. Johnson, Y. Hu, and D. Maltz, "The Dynamic Source Routing Protocol (DSR) for Mobile Ad Hoc Networks for IPv4." RFC 4728 (Experimental), Feb. 2007.

[28] M. Iordanakis, D. Yannis, K. Karras, G. Bogdos, G. Dilintas, M. Amirfeiz, G. Colangelo, and S. Baiotti, "Ad-hoc routing protocol for aeronautical mobile ad-hoc networks," in *Fifth International Symposium on Communication Systems, Networks and Digital Signal Processing (CSNDSP)*, July 2006.

[29] A. Jabbar and J. P. Sterbenz, "AeroRP: A geolocation assisted aeronautical routing protocol for highly dynamic telemetry environments," in *International Telemetering Conference (ITC) 2009*, (Las Vegas, NV), October 2009.

[30] W. Gu, J. Li, F. He, F. Cai, and F. Yang, "Delay-aware stable routing protocol for aeronautical ad hoc networks," *Journal on Information & Computational Science*, 2012.

[31] E. Sakhaee, A. Jamalipour, and N. Kato, "Multipath doppler routing with QoS support in pseudo-linear highly mobile ad hoc networks," in *Communications, 2006. ICC '06. IEEE International Conference on*, vol. 8, pp. 3566–3571, June 2006.

[32] A. Tiwari, A. Ganguli, A. Sampath, D. Anderson, B.-H. Shen, N. Krishnamurthi, J. Yadegar, M. Gerla, and D. Krzysiak, "Mobility aware routing for the airborne network backbone," in *Military Communications Conference, 2008. MILCOM 2008. IEEE*, pp. 1–7, Nov 2008.

[33] K. Peters, A. Jabbar, E. K. Çetinkaya, and J. P. Sterbenz, "A geographical routing protocol for highly-dynamic aeronautical networks," in *Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC)*, (Cancun, Mexico), pp. 492–497, March 2011.

[34] K. Peters, E. K. Çetinkaya, A. Jabbar, and J. P. G. Sterbenz, "Analysis of a geolocation-assisted routing protocol for airborne telemetry networks," in *Proceedings of the International Telemetering Conference (ITC)*, (San Diego, CA), October 2010.

[35] Y. Cheng, E. K. Çetinkaya, and J. P. Sterbenz, "Performance comparison of routing protocols for transactional traffic over aeronautical networks," in *Proceedings of the International Telemetering Conference (ITC)*, (Las Vegas, NV), October 2011.

[36] J. P. Rohrer and J. P. Sterbenz, "Performance and disruption tolerance of transport protocols for airborne telemetry networks," in *International Telemetering Conference (ITC) 2009*, (Las Vegas, NV), October 2009.

[37] J. Rohrer, K. Pathapati, T. Nguyen, and J. Sterbenz, "Opportunistic transport for disrupted airborne networks," in *MILITARY COMMUNICATIONS CONFERENCE, 2012 - MILCOM 2012*, pp. 1–9, Oct 2012.

[38] K. Fall, "A delay-tolerant network architecture for challenged internets," in *Proceedings of the 2003 conference on Applications, technologies, architectures, and protocols for computer communications*, SIGCOMM '03, (New York, NY, USA), pp. 27–34, ACM, 2003.

[39] S. Farrell and V. Cahill, *Delay and Disruption Tolerant Networking*. Artech House, 2006.

[40] V.Cerf, S. Burleigh, A. Hooke, L. Torgerson, R. Durst, K. Scott, K. Fall, and H. Weiss, "Delay tolerant network architecture," *IETF RFC 4838*, April 2007.

[41] K. Scott and S. Burleigh, "Bundle Protocol Specification." RFC 5050 (Experimental), Nov. 2007.

[42] M. Ramadas, S. Burleigh, and S. Farrell, "Licklider Transmission Protocol - Specification." RFC 5326 (Experimental), Sept. 2008.

[43] D. B. Johnson and D. A. Maltz, "Dynamic source routing in ad hoc wireless networks," in *Mobile Computing*, pp. 153–181, Kluwer Academic Publishers, 1996.

[44] K. Pearson, "The Problem of the Random Walk," *Nature*, vol. 72, no. 1865, p. 294, 1905.

[45] M. Abdulla and R. Simon, "Characteristics of common mobility models for opportunistic networks," in *Proceedings of the 2nd ACM workshop on Performance monitoring and measurement of heterogeneous wireless and*

wired networks, PM2HW2N '07, (New York, NY, USA), pp. 105–109, ACM, 2007.

[46] M. Abdulla and R. Simon, "A simulation study of common mobility models for opportunistic networks," in *Simulation Symposium, 2008. ANSS 2008. 41st Annual*, pp. 43–50, 2008.

[47] T. Spyropoulos, K. Psounis, and C. S. Raghavendra, "Performance analysis of mobility-assisted routing," in *Proceedings of the 7th ACM international symposium on Mobile ad hoc networking and computing*, MobiHoc '06, (New York, NY, USA), pp. 49–60, ACM, 2006.

[48] R. Groenevelt, P. Nain, and G. Koole, "The message delay in mobile ad hoc networks," *Perform. Eval.*, vol. 62, pp. 210–228, Oct. 2005.

[49] M. McNett and G. M. Voelker, "Access and mobility of wireless pda users," *SIGMOBILE Mob. Comput. Commun. Rev.*, vol. 9, pp. 40–55, Apr. 2005.

[50] P. Hui, A. Chaintreau, J. Scott, R. Gass, J. Crowcroft, and C. Diot, "Pocket switched networks and human mobility in conference environments," in *Proceedings of the 2005 ACM SIGCOMM workshop on Delay-tolerant networking*, WDTN '05, (New York, NY, USA), pp. 244–251, ACM, 2005.

[51] A. Chaintreau, P. Hui, J. Crowcroft, C. Diot, R. Gass, and J. Scott, "Impact of human mobility on opportunistic forwarding algorithms," *Mobile Computing, IEEE Transactions on*, vol. 6, no. 6, pp. 606–620, June 2007.

[52] S. Jain, K. Fall, and R. Patra, "Routing in a delay tolerant network," *SIGCOMM Comput. Commun. Rev.*, vol. 34, pp. 145–158, Aug. 2004.

[53] X. Zhang, J. Kurose, B. N. Levine, D. Towsley, and H. Zhang, "Study of a bus-based disruption-tolerant network: Mobility modeling and impact on routing," in *Proceedings of the 13th Annual ACM International Conference on Mobile Computing and Networking*, MobiCom '07, (New York, NY, USA), pp. 195–206, ACM, 2007.

[54] S. M. Tornell, C. T. Calafate, J.-C. Cano, and P. Manzoni, "Assessing the effectiveness of dtn techniques under realistic urban environments," in *Proceedings of the 2013 IEEE 38th Conference on Local Computer Networks*, pp. 598–605, IEEE Computer Society, 2013.

[55] A. Balasubramanian, B. Levine, and A. Venkataramani, "Dtn routing as a resource allocation problem," *SIGCOMM Comput. Commun. Rev.*, vol. 37, pp. 373–384, Aug. 2007.

[56] T. Spyropoulos, K. Psounis, and C. Raghavendra, "Efficient routing in intermittently connected mobile networks: The single-copy case," *Networking, IEEE/ACM Transactions on*, vol. 16, pp. 63–76, Feb 2008.

[57] E. M. Daly and M. Haahr, "Social network analysis for routing in disconnected delay-tolerant manets," in *Proceedings of the 8th ACM International Symposium on Mobile Ad Hoc Networking and Computing*, MobiHoc '07, (New York, NY, USA), pp. 32–40, ACM, 2007.

[58] A. Vahdat and D. Becker, "Epidemic Routing for Partially Connected Ad Hoc Networks," tech. rep., July 2000.

[59] T. Spyropoulos, K. Psounis, and C. S. Raghavendra, "Efficient routing in intermittently connected mobile networks: The multiple-copy case," *IEEE/ACM Trans. Netw.*, vol. 16, pp. 77–90, Feb. 2008.

[60] P. Hui, J. Crowcroft, and E. Yoneki, "Bubble rap: Social-based forwarding in delay-tolerant networks," *Mobile Computing, IEEE Transactions on*, vol. 10, pp. 1576–1589, Nov 2011.

[61] A. Krifa, C. Barakat, and T. Spyropoulos, "Optimal buffer management policies for delay tolerant networks," in *Sensor, Mesh and Ad Hoc Communications and Networks, 2008. SECON '08. 5th Annual IEEE Communications Society Conference on*, pp. 260–268, June 2008.

[62] M. Al-Siyabi, H. Cruickshank, and Z. Sun, "Dtn qos metrics and fair resources management model," in *Electrical and Computer Engineering (CCECE), 2011 24th Canadian Conference on*, pp. 000704–000707, May 2011.

[63] M. Al-Siyabi, H. Cruickshank, and Z. Sun, "Quality of service provisioning for delay tolerant network by implementing admission control model for aircrafts bundles data transmission," in *Proceedings of the 6th International Wireless Communications and Mobile Computing Conference*, IWCMC '10, (New York, NY, USA), pp. 706–710, ACM, 2010.

[64] M. Al-Siyabi, H. Cruickshank, Z. Sun, and G. Ansa, "Fairness and satisfaction model for dtn applications using various transportation means," in *Proceedings of the 9th IFIP TC 6 International Conference on Wired/Wireless Internet Communications*, WWIC'11, (Berlin, Heidelberg), pp. 446–457, Springer-Verlag, 2011.

[65] S. Burleigh, E. Jennings, and J. Schoolcraft, "Autonomous congestion control in delay-tolerant networks," in *SpaceOps Conference*, june 2006.

[66] M. Al-Siyabi, H. Cruickshank, and Z. Sun, "Delay/disruption tolerant network architecture for aircrafts datalink on scheduled routes," in *Personal Satellite Services* (K. Sithamparanathan, M. Marchese, M. Ruggieri, and I. Bisio, eds.), vol. 43 of *Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering*, pp. 235–248, Springer Berlin Heidelberg, 2010.

[67] "A delay tolerant networking architecture for airborne networking," Tech. Rep. AFRK-RI-RS-TR-2010-095, Air Force Research Laboratory, 2010.

[68] T. Jonson, J. Pezeshki, V. Chao, K. Smith, and J. Fazio, "Application of delay tolerant networking (dtn) in airborne networks," in *Military Communications Conference, 2008. MILCOM 2008. IEEE*, pp. 1–7, Nov 2008.

[69] S. Khan and K.-I. Kim, "A new dtn routing protocol for sparse airborne networks," in *Mobile Ad-Hoc and Sensor Systems (MASS), 2013 IEEE 10th International Conference on*, pp. 417–418, Oct 2013.

[70] A. Lindgren, A. Doria, E. Davies, and S. Grasic, "Probabilistic Routing Protocol for Intermittently Connected Networks." RFC 6693 (Experimental), Aug. 2012.

[71] ns-3, Network simulator 3, "Discrete-event network simulator for internet systems," 2015. http://www.nsnam.org.

[72] OpenFlights.org, "Flight logging, mapping, stats and sharing. airport, airline and route data," 2012. http://openflights.org/data.html.

[73] "ArcGIS platform ESRI, a Geographic information system software," *http://www.esri.com/*, 2012.

[74] OpenJump, "Open source geographic information system (gis)," 2015. http://www.openjump.org.

[75] Flightview, "Global flight tracker: Real time flight tracker & airport delays," 2013. http://www.flightview.com/TravelTools/.

[76] Flightaware, "Flightaware, flight tracker , flight status , flight tracking," 2013. http://es.flightaware.com/.

[77] M. Bowler, "Htmlunit: the gui-less browser for java programs," 2015. http://htmlunit.sourceforge.net.

[78] E. Parzen, "On estimation of a probability density function and mode," *Ann. Math. Statist.*, vol. 33, no. 3, pp. 1065–1076, 1962.

[79] L. Devroye, *Non-Uniform Random Variate Generation*. Springer-Verlag, 1986.

[80] ISO/IEC 13249-3:2011, 2011. https://www.iso.org/obp/ui/#iso:std:iso-iec:13249:-3:ed-4:v1:en.

[81] Y. Shafranovich, "Common format and mime type for comma-separated values (csv) files." RFC 4180 (Informational), Oct. 2005. Updated by RFC 7111.

[82] Institute of Electrical and Electronics Engineers, "Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications High-speed Physical Layer in the 5 GHz Band," 2003.

[83] Ubiquiti networks, "Bullet: Revolutionary outdoor radio device," 2013. http://www.ubnt.com/bullet.

[84] S. Arnon, J. Barry, G. Karagiannidis, R. Schober, and M. Uysal, *Advanced Optical Wireless Communication Systems*. New York, NY, USA: Cambridge University Press, 1st ed., 2012.

[85] Institute of Electrical and Electronics Engineers, "IEEE standard for local and metropolitan area networks — part 16: Air interface for broadband wireless access systems," May 2009.

[86] M. J. Alenazi, Y. Cheng, D. Zhang, and J. P. Sterbenz, "Epidemic routing protocol implementation in ns-3," in *Proceedings of the 2015 Workshop on Ns-3*, WNS3 '15, pp. 83–90, 2015.

[87] C. R. Baugh and J. Huang, "Traffic model for 802.16 tg3 mac/phy simulations," tech. rep., IEEE 802.16 Broadband Wireless Access Working Group, 2001.

[88] T. Karagiannis, J.-Y. L. Boudec, and M. Vojnovic, "Power-law and exponential decay of inter-contact times between mobile devices," in *Proc. of ACM Mobicom 2007*, pp. 183 – 194, Association for Computing Machinery, Inc., September 2007.

[89] V. D. Blondel, J.-L. Guillaume, R. Lambiotte, and E. Lefebvre, "Fast unfolding of communities in large networks," *Journal of Statistical Mechanics: Theory and Experiment*, vol. 2008, no. 10, p. P10008, 2008.

[90] L. C. Freeman, "A Set of Measures of Centrality Based on Betweenness," *Sociometry*, vol. 40, pp. 35–41, Mar. 1977.

[91] G. J. Székely and M. L. Rizzo, "Brownian distance covariance," *Annals of Applied Statistics*, vol. 3, no. 4, pp. 1233–1303, 2009.

[92] S. Fortunato and A. Lancichinetti, "Community detection algorithms: A comparative analysis: Invited presentation, extended abstract," in *Proceedings of the Fourth International ICST Conference on Performance Evaluation Methodologies and Tools*, VALUETOOLS '09, pp. 27:1–27:2, ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 2009.

[93] J. Mitchell, "Aircraft satellite communications system for distributing internet service from direct broadcast satellites," Mar. 4 2003. US Patent 6,529,706.

[94] J. Tang, M. Musolesi, C. Mascolo, V. Latora, and V. Nicosia, "Analysing information flows and key mediators through temporal centrality metrics," in *Proceedings of the 3rd Workshop on Social Network Systems*, SNS '10, (New York, NY, USA), pp. 3:1–3:6, ACM, 2010.

[95] J. Schawalder, "The future of inflight entertainment in europe, according to passenger expectations: Why airlines should embrace consumer technology," Master's thesis, Grenoble Ecole de Management, 2014.

[96] G. Karp, "Sky-high wifi nearly ready to widely fly: As airlines equip and battery life expands, more passengers expected to connect in the clouds," 2012. http://articles.chicagotribune.com/.

[97] L. Battaglia, P. Unger, M. Werner, and M. Holzbock, "Satellite capacity dimensioning for in- flight internet services in the north atlantic region," in *2nd AIAA International Communications Satellite Systems Conference & Exhibit 2004 (ICSSC)*, 2014.

[98] R. Jain, D. Chiu, and W. Hawe, "A quantitative measure of fairness and discrimination for resource allocation in shared computer systems," tech. rep., DEC Research Report TR-301, 1984.

[99] H. Zhu, S. Chang, M. Li, K. Naik, and S. Shen, "Exploiting temporal dependency for opportunistic forwarding in urban vehicular networks," in *INFOCOM, 2011 Proceedings IEEE*, pp. 2192–2200, April 2011.

[100] "Bundle protocol model for ns-3." http://code.nsnam.org/dizhizhou/ns-3-dev-socis2013/summary.

[101] F. Warthman, *Delay-Tolerant Networks (DTNs) - A Tutorial*. Warthman Associates, 2012.

[102] F. Bellifemine, A. Poggi, and G. Rimassa, "JADE - a FIPA-compliant agent framework," in *Proceedings of the Practical Applications of Intelligent Agents*, 1999.

[103] J. Ametller-Esquerra, J. Cucurull-Juan, R. Martí, G. Navarro, and S. Robles, "Enabling mobile agents interoperability through FIPA standards," *Klusch, M., Rovatsos, M., Payne, T.R. (eds.) CIA 2006. LNCS (LNAI), vol. 4149, pp. 388-401. Springer, Heidelberg*, 2006.

[104] S. J. Undertaking, "Wp 14 - swim technical architecture, statement of work," tech. rep., 2008.

[105] S. Wilson, "Information and service models - swim technical briefing," tech. rep., 2011.

[106] "EXata," *http://www.scalable-networks.com/exata/*, 2012.

[107] R. Wang, S. C. Burleigh, P. Parikh, C.-J. Lin, and B. Sun, "Licklider transmission protocol (LTP)-based DTN for cislunar communications," *IEEE/ACM Trans. Netw.*, vol. 19, pp. 359–368, Apr. 2011.

[108] "Ltp model for omnet++." https://github.com/Phisches/omnet_ltp.

[109] "Dtn routing for ns-3." http://www.netlab.tkk.fi/tutkimus/dtn/ns/.

[110] A. Keränen, J. Ott, and T. Kärkkäinen, "The ONE Simulator for DTN Protocol Evaluation," in *SIMUTools '09: Proceedings of the 2nd International Conference on Simulation Tools and Techniques*, (New York, NY, USA), ICST, 2009.

[111] L. Romero Amondaray and J. Seoane Pascual, "Delay tolerant network simulation with vnuml," in *Proc. of CHANTS '08*, pp. 109–112, ACM, 2008.

[112] W. Hołubowicz, Ł. Kiedrowski, and K. Romanowski, "DTN simulation tool for N4C integration support," vol. 6994 of *LNCS*, pp. 327–328, Springer, 2011.

[113] "The interplanetary overlay network (ION): Delay-tolerant networking suitable for use in spacecraft." http://ion-dtn.sourceforge.net/.

[114] "Space networks user's guide (SNUG)," Tech. Rep. Rev. 9, Goddard Space Flight Center, 2007.

[115] "LTPlib documentation and test." http://dtn.dsg.cs.tcd.ie/sft/ltplib/.

[116] "LTP-RI: A ltp java implementation distributed under the terms of the ocp-license.." http://irg.cs.ohiou.edu/ocp/ltp.html.

[117] "Common open research emulator (CORE): A tool for emulating networks on one or more machines. developed developed by boeing research and technology division.." http://www.nrl.navy.mil/itd/ncs/products/core.

Rubén Martínez-Vidal
Bellaterra, September 2015