Routing and Internet Gateway Selection in Aeronautical Ad Hoc Networks

vorgelegt von
Diplom-Ingenieur
Felix Hoffmann
geb. in Köln

von der Fakultät IV – Elektrotechnik und Informatik
der Technischen Universität Berlin
zur Erlangung des akademischen Grades

Doktor der Ingenieurwissenschaften
– Dr.-Ing. –

genehmigte Dissertation

Promotionsausschuss:

Vorsitzender:	Prof. Dr.-Ing. Hans-Joachim Grallert
Berichter:	Prof. Dr.-Ing. Adam Wolisz
Berichter:	Prof. Dr. Hannes Hartenstein
Berichter:	Prof. Dr.-Ing. Matthias Hollick

Tag der wissenschaftlichen Aussprache: 3. Dezember 2014

Berlin 2014

Acknowledgements

First of all, I would like to thank my advisor, Prof. Adam Wolisz of the Telecommunication Networks Group at TU Berlin, for supporting my work on this thesis. The open discussions with him and his comments on my papers and this thesis were very helpful.

I am also grateful to Michael Schnell and Uwe-Carsten Fiebig at the Institute of Communications and Navigation at the German Aerospace Center (DLR) in Oberpfaffenhofen for giving me the opportunity to work on the field of aeronautical ad hoc networks. Many thanks also to my colleagues Daniel Medina, Serkan Ayaz, and Christian Bauer at DLR for the many discussions that we had.

Finally, I would like thank my wife and my two sons for the support that they have given me over the last years.

Zusammenfassung

In Verkehrsflugzeugen wird ein Internet-Zugang für die Passagiere typischerweise durch eine Verbindung über geostationäre Satelliten ermöglicht. In manchen Regionen kann das Flugzeug auch eine Verbindung mit einem eigens dafür aufgebauten zellulären Mobilfunknetz am Boden herstellen. Allerdings ist der Internet-Zugang über eine Satellitenverbindung relativ teuer und leidet unter den extrem langen Signallaufzeiten zu den geostationären Satelliten. Ein Mobilfunknetz kann nur genutzt werden, solange sich das Flugzeug über Land befindet, und in entlegenen ländlichen Regionen lohnt sich der Aufbau eines solchen Netzes aus wirtschaftlichen Gründen häufig nicht. Wegen dieser Nachteile der bestehenden Möglichkeiten sind ad hoc Netze als mögliche Alternative für entlegene Regionen mit einer ausreichend hohen Verkehrsdichte, etwa der Nordatlantik, vorgeschlagen worden. Durch drahtlose Flugzeug-zu-Flugzeug Verbindungen können Datenpakete von einem Flugzeug zu einem anderen durch das Netz weitergeleitet werden. Flugzeuge, die über eine direkte Internetverbindung verfügen – entweder über einen Satelliten oder eine Bodenstation – können diese Verbindung somit anderen Flugzeugen im Netz zur Verfügung stellen und diesen als sog. "Internet Gateway" dienen. Jedes Datenpaket erreicht oder verlässt das ad hoc Netz also durch einen dieser Internet Gateways. Die Anbindungen der Internet Gateways an das Internet können sich hinsichtlich ihrer Übertragungskapazität oder Latenzzeit jedoch stark unterscheiden. Damit spielt die Zuordnung, welche Flugzeuge welche Internet Gateways verwenden, eine kritische Rolle für die Leistungsfähigkeit des gesamten Netzes. Diese Zuordnung zwischen Flugzeugen und Gateways ist das wesentliche Problem, das in dieser Arbeit behandelt wird. Allerdings hängt diese Zuordnung auch stark von der Qualität des Pfades ab, den die Pakete durch das ad hoc Netz zwischen dem Gateway und dem Flugzeug nehmen. Und die Qualität dieser Verbindung wiederum hängt stark von der Verfügbarkeit des drahtlosen Übertragungskanals ab, den sich die Flugzeuge in dem ad hoc Netz miteinander teilen müssen. In dieser Arbeit werden somit die Zuteilung von Internet Gateways, das Routing von Paketen durch das Netz und die Zuweisung des Funkkanals an einzelne Flugzeuge als gemeinsames Optimierungsproblem behandelt. Dazu werden zunächst die Flugbewegungen im Nordatlantik untersucht, um die Eigenschaften des ad hoc Netzes zu modellieren. Danach wird die Zuordnung von Internet Gateways, das Routing und die Ressourcenzuteilung als mathematisches Optimierungsproblem formuliert, das zum Ziel hat, die durchschnittliche Latenzzeit von Datenpaketen in dem ad hoc Netz zu minimieren. Als heuristisches Lösungsverfahren für dieses Optimierungsproblem wird ein genetischer Algorithmus vorgeschlagen. Durch Simulationen kann gezeigt werden, dass die Qualität der Lösung, die durch den genetischen Algorithmus erzielt wird, in kleinen Netzen vergleichbar ist mit der Lösung des mathematischen Optimierungsproblems. Außerdem lässt sich der genetische Algorithmus wesentlich besser auf größere Netze anwenden, in denen die klassischen Optimierungsverfahren nicht mehr anwendbar sind. Nach diesen beiden zentralisierten Lösungsverfahren wird das sog. "Minimum Downstream Delay" Verfahren als verteilter Lösungsansatz entwickelt. Die Leistungsfähigkeit dieser Lösung wird anhand von Simulationen untersucht, die den Flugverkehr sowie den durch die Passagiere erzeugten Datenverkehr über dem Nordatlantik realistisch nachbilden.

Abstract

In-flight Internet access for passengers in commercial aircraft is currently typically provided by geostationary satellites. Recently, connectivity in some areas is also provided by means of a cellular network of base stations on the ground. However, satellite based Internet access is relatively expensive, and suffers from large propagation delay due to the extremely large distance of the geostationary satellites. Cellular networks can only be deployed over land areas, but not in oceanic regions, and their deployment on land in remote regions may not be economically feasible. Due to these drawbacks of the existing solutions for in flight Internet access, ad hoc networks formed by air to air links between the aircraft have recently been proposed as an alternative possibility, especially for oceanic regions with a sufficiently high amount of air traffic, such as the North Atlantic corridor. Aircraft that have a direct connection to the Internet, either through a ground station or a satellite link, may act as Internet Gateways for other aircraft in the network. All data packets that are generated at or destined for one of the aircraft in the network must pass through one of these Internet gateways. At the same time, the gateways' connections to the Internet may exhibit significantly different characteristics regarding their average packet delay or their available capacity. Therefore, the allocation of traffic flows to Internet gateways is crucial for the overall network performance. This problem of Internet gateway selection, with the objective of minimizing average packet delay, is the core problem that is addressed in this thesis. The allocation of flows to gateways is closely related to the quality of the path through the network between the aircraft and the gateway. Obviously, this path quality is also closely linked to the availability of wireless channel resources along the path. Therefore, we consider the joint problem of gateway selection, routing, and channel access in aeronautical ad hoc networks. We first analyze the typical air traffic characteristics in the North Atlantic region and show that such a network would indeed be feasible. Then, we formulate the joint gateway selection, routing, and scheduling problem as a mathematical optimization problem with the goal of minimizing the average packet delay in the network. As a less computationally complex alternative, we define a Genetic Algorithm to solve the optimization problem in a heuristic manner. By means of simulations, it is shown that the performance of the Genetic Algorithm approach comes close to the mathematical programming approach in small networks and the algorithm scales well to larger networks. After these centralized approaches to solving the gateway selection and routing problem, the Minimum Downstream Delay algorithm is defined as a distributed approach to the problem and its integration in a protocol within the IPv6 protocol stack is addressed. The performance of this protocol is then simulated in a realistic environment that models the air traffic situation in the North Atlantic and generates realistic data traffic mimicking the Internet usage of passengers on board the aircraft.

Contents

List of Figures

List of Tables

1. Introduction

1.1. Motivation and Problem Statement

During the last years, ubiquitous Internet access has become commonplace. People have become accustomed to high speed Internet access wherever they go — whether through DSL connections at home and at work, through third or fourth generation cellular networks for mobile users, or by means of local WiFi hotspots in public places. One domain that has fallen behind in this continuous advance is the domain of civil aviation. In-flight Internet access to airline passengers has traditionally been provided by means of geostationary satellites. However, satellite based Internet access suffers from high costs [4] and large propagation delay. This has recently motivated the search for alternative ways of providing Internet connectivity to airline passengers. Since 2008, Aircell [5] has been offering direct air to ground Internet connectivity through a network of dedicated base stations in North America. But such cellular networks require a large number of base stations and can only be deployed over land. When aircraft fly in oceanic or very remote areas, which is often the case on long distance flights where demand for Internet access is highest, other solutions must be found. Recently, wireless ad hoc networks have been proposed for connecting aircraft to the Internet (see e.g. [6], [7], [8]). In a wireless ad hoc network, nodes that are not within direct communications range of each other can still communicate if intermediate nodes relay the data packets until they arrive at the intended destination. Here, this destination can be either an aircraft in the ad hoc network, or one of potentially many ground stations that are located on land, close to the shore e.g. in Great Britain or northeastern Canada. The paths along which packets are forwarded must be determined by a routing protocol that is executed by the nodes in the network. A subset of the aircraft in the ad hoc network may have a connection to the Internet via a ground station or a satellite link. Such *Internet gateways* can provide Internet access to those aircraft in the network that do not have a direct connection to the Internet.

This makes ad hoc networks an attractive solution for Internet access in regions where base stations cannot be deployed at all, such as over the oceans, or in remote areas where the costs of deploying a network of base stations and connecting them to the Internet would be prohibitive. In this work, we will focus on an aeronautical ad hoc network formed by airliners flying over the North Atlantic. We have analyzed the traffic patterns in this region in previous work [9] in order to determine the feasibility of such an ad hoc network. The North Atlantic is one of the most dense oceanic airspaces worldwide, and the traffic characteristics there appear to be particularly favorable.

On any given day, there are two distinct waves of aircraft flying over the North Atlantic:

one wave of eastbound aircraft and another wave of westbound aircraft. Typically, the westbound aircraft depart from Europe around the middle of the day and arrive in North America in the evening (local time). The eastbound aircraft depart from North America in the late afternoon or evening, fly during the night, and arrive in Europe early in the morning. These two waves of aircraft practically do not overlap, i.e. the North Atlantic can be thought of as a "one way street" with alternating directions, depending on the time of day. Similar to the lanes of a highway, aircraft fly on predefined routes, the so-called North Atlantic Tracks [10]. These air routes are calculated each day based on the current weather conditions, with the goal of making the air traffic as efficient and as safe as possible. When entering the North Atlantic airspace, aircraft are assigned to one of these tracks by an air traffic controller either in Canada or in Great Britain or Ireland and are then required to stay on that track at the assigned altitude and fly with a well defined velocity, in order to maintain sufficient separation between aircraft. Therefore, there is little relative movement between the aircraft themselves once they are over the ocean. From a networking point of view, this is good news, since it means that stable communications links between aircraft could be set up and are likely to exist for practically the entire duration of the transatlantic flight.

Aircraft near the coastline can connect to base stations on the ground and thus assume the role of Internet gateways by connecting the ad hoc network to the Internet. In addition, some aircraft may be equipped with satellite transponders that allow them to access the Internet directly, over a satellite connection, without the need for relaying by other aircraft. These aircraft could also act as Internet gateways. This may be useful when the total traffic load in the network is so high that the base stations on the ground would be overloaded, or as a fallback solution in case the aircraft density is so low that aircraft over the ocean cannot establish a multihop path to an Internet gateway connecting to a ground station. Thus, we are dealing with a heterogeneously connected ad hoc network, in which the Internet gateways have significantly different characteristics. Here, we will use the terms *terrestrial* gateways and *satellite* gateways to distinguish between these two classes of gateways.

Since we envisage that this ad hoc network will primarily provide Internet access to the passengers, practically all data traffic will need to pass through an Internet gateway. Direct traffic from one aircraft to another aircraft can be neglected. Since the quality of service that can be provided by the Internet gateways will vary, depending e.g. on the current radio channel conditions, traffic load, etc., and given the inherently different delay characteristics of terrestrial and satellite gateways, the choice of which Internet gateway to use for a certain traffic flow at a certain aircraft will have a significant effect on the overall network performance, and thereby the users' satisfaction. Avoiding satellite links was one of our motivations for considering ad hoc networks in the first place, but there may be situations in which a satellite gateway is preferable to a terrestrial gateway.

This problem of choosing the appropriate Internet gateway is referred to as the *Internet gateway selection* problem. Of course, the quality of service that a gateway can provide to an aircraft depends not only on the properties of the gateway itself, but also on the properties of the path between the aircraft and the gateway through the ad hoc network.

The distance to the gateway, measured in terms of hops can be different, or the traffic load along the path between the aircraft and the gateway may vary significantly. Thus, the problem of Internet gateway selection is closely tied to the problem of *routing* in the ad hoc network. In wireless networks such as our aeronautical ad hoc network, all nodes must share the common radio channel. That is, radio resources must be allocated among the nodes, e.g. by time or frequency division multiplexing, or by separating transmissions spatially by means of directional antennas. To a certain degree, users are inherently separated due to the large geographic extent of the network – a transmission by an aircraft on one side of the Atlantic cannot be received by an aircraft on the other side of the Atlantic regardless of the transmit power, because it is beyond the transmitter's radio horizon, which is determined by the curvature of the Earth's surface, as well as the altitudes of the transmitter and receiver. This limitation of the communication range can be seen as a benefit, since it also reduces the scope of the interference that is generated by a given transmission. However, the radio horizon can still be very large, on the order of hundreds of kilometers. This means that each aircraft would receive a very large number of transmissions that are actually intended for another aircraft, and is not able to receive data intended for itself during this time. Therefore, measures to reduce the amount of interference must also be considered. For example, directional antennas could increase the degree of spatial reuse of the wireless channel.

If the channel is divided up between nodes in the time domain, then the allocation of resources to nodes or links is referred to as the *scheduling* problem (see e.g. [11] or [12]). Since the scheduling determines what resources are available along a path or at the gateway, the gateway selection, routing, and scheduling problems must be considered jointly in order to utilize the ad hoc network as efficiently as possible. This is the core problem that will be addressed in this thesis.

In previous work, these three subproblems have typically been treated separately. A gateway selection algorithm would first determine to which gateway a packet should be sent. This decision could be based on a metric such as the distance of the gateway in terms of hops [13], its current traffic load [14], or some combination of different metrics [15]. The address of this gateway would then be given to the routing algorithm, which would try to find a suitable route to this destination. Finally, when packets are routed along this path, the scheduling algorithm would react to the changed traffic demand and try to allocate sufficient resources to the links along the path.

However, we have seen that these three steps are obviously very closely linked to each other, so that addressing each of them separately is clearly suboptimal. In this thesis, we consider the joint gateway selection, routing and scheduling problem in aeronautical ad hoc networks. We are interested in distributing traffic in the network as efficiently as possible, such that a global design goal is achieved, e.g. minimizing the average delay of packets within the wireless network. In addition, the different characteristics of the Internet gateways can be exploited to distribute traffic flows of different service classes among the gateways such that the quality of service targets of the more demanding flows can still be fulfilled.

Finally, since we are considering ad hoc networks as a means of providing Internet access for passengers, we must also address the question of how this ad hoc network can be integrated into the TCP/IP protocol suite of the Internet. In this thesis, we will focus on IPv6 rather than the currently still predominant IPv4. Although similar problems have been addressed e.g. in the areas of car to car communications (see e.g. [16], [17]), we must find a suitable protocol architecture for the aeronautical ad hoc network that is able to accommodate the proposed concepts for routing and Internet gateway selection.

1.2. Structure and Approach of This Thesis

As described in the previous section, we address the joint Internet gateway selection, routing, and scheduling problem in this thesis. In this section, we will describe the approach that was taken and the structure of this thesis.

In **Chapter 2**, background information relevant for the work in this thesis is given. This includes the development of in-flight Internet access, an overview of the current state of the art in aeronautical ad hoc networks, the air to air radio channel, and the role of Internet gateways in ad hoc networks.

Chapter 3 presents a set of requirements that must be fulfilled by the IPv6 protocol architecture of the aeronautical ad hoc network, as well as a set of evaluation criteria for selecting the most suitable architecture. Different protocol architectures are then discussed, the most promising architecture is selected, and a functional routing architecture is defined, based on this architecture.

In **Chapter 4**, we perform an analysis of the network topology of the envisaged aeronautical ad hoc network in the North Atlantic, based on actual flight data. In particular, the gateway connectivity of the ad hoc network and the node degree of the aircraft are examined in detail.

Chapter 5 defines the communication system model that will be used in the remainder of the thesis. In particular, the model for the physical layer and link layer of the air to air and air to ground communication systems are defined, and a simple model for the general network structure is defined.

We first look at the gateway selection, routing, and scheduling problem from a centralized point of view. That is, we assume that a centralized entity has full knowledge of all relevant network parameters such as the traffic demands and the network topology. In **Chapter 6**, we formulate the core problem of this thesis in a mathematical form: minimizing the average packet delay in the network by jointly optimizing the gateway allocation, routing, and scheduling. Since this problem is non-linear and exhibits a large number of binary variables, a sub-optimum approach to solving this problem is proposed, which consists of decomposing the problem into two sub-problems. Even though this decomposition no longer finds a solution to all components of the problem – gateway selection, routing, scheduling – in a single step, it is still superior to solving the three components independently. The first step of the decomposition includes constraints that ensure the feasibility of the second step. If gateway selection were done completely independently of the other two subproblems, it could result in a solution for which no feasible routing and scheduling exists.

However, even with this simplification, the mathematical optimization approach cannot be applied to networks whose size is of practical interest. Therefore, in **Chapter 7**, a heuristic approach to solving the delay minimization problem that was formulated in the previous chapter is proposed. This approach uses a Genetic Algorithm with the average packet delay as its cost function. This algorithm can be applied to both static and dynamic networks. Its performance is assessed in a small test network by means of comparison with the mathematical programming approach of Chapter 6, and it is seen that the performance of the Genetic Algorithm approach comes close to the mathematical optimization.

Centralized algorithms as those proposed in Chapters 6 and 7 are difficult to implement in practice, since they require up to date knowledge of the network parameters at a central location, and require that the calculated solution be conveyed back to the network nodes before the solution has become outdated. Therefore, the rest of this thesis focuses on a distributed solution that does not rely on such a central entity.

Chapter 8 builds on the functional architecture that was defined in Chapter 3 and defines a routing and gateway selection protocol that aims to minimize packet delay. In contrast to the algorithmic approach of Chapters 6 and 7, this protocol is designed to run in a distributed manner without a central optimization entity. The performance of this protocol is analyzed by a comparison with the Genetic Algorithm approach of Chapter 7, and it is shown that the performance of the distributed algorithm comes close to the performance of the centralized one.

In **Chapter 9**, simulations are carried out to assess the performance of the protocol defined in Chapter 8 in an environment that realistically models the envisaged aeronautical ad hoc network in the North Atlantic region. This includes realistic air traffic patterns and models of the data traffic that is generated by the airline passengers.

In **Chapter 10**, the main contributions of this thesis are summarized, and potential directions for future research are suggested.

Finally, the Appendix contains detailed definitions of the control messages that are used in the routing and gateway selection protocol defined in Chapter 8, as well as detailed descriptions of the user traffic models used in the simulations of Chapter 9.

2. Background

In this chapter, relevant background information for the work in this thesis is given. This includes an overview of the development of in flight Internet access over the last years, the idea of aeronautical ad hoc networks, the underlying network architecture, as well as the role of Internet gateways in ad hoc networks.

2.1. In Flight Internet

In aeronautical communications, three different service domains are typically distinguished [18]: Air Traffic Control (ATC), Airline Operational Communications (AOC), and Aeronautical Passenger Communications (APC). ATC services are related to the safe and secure operation of flight, and comprise the communication between the cockpit crew and the air traffic controllers on the ground. A typical example for ATC services are flight clearances instructing a pilot to take a new heading. AOC services enable airlines to operate their fleets more efficiently by exchanging business related information, such as catering information or the status of connecting flights, between the cockpit or cabin crew and the airline operations center on the ground, but are generally not safety related. Finally, APC services are intended for the entertainment of passengers during the flight. In this work, we will focus on APC services only.

Traditionally, in-flight Internet access for passengers has been provided by means of geostationary satellites, with a WiFi hotspot inside the cabin to allow passengers to access the network from their private laptop computers or smartphones. Most of the remainder of this section has been summarized from an article by Glenn Fleishman, which is available online [19]. The first such system in operational use was the Connexion By Boeing (CBB) system [20], which started service in 2004. However, the CBB service was already discontinued by Boeing in 2006, reportedly because of limited acceptance by airlines [21]. This was due in part to the costs for installing the CBB hardware onboard an aircraft, which were reported to be as high as $500,000 per aircraft [19]. Many airlines shunned the high costs of installing an expensive new system on their long haul fleets. After CBB was canceled, Lufthansa, which had been one of the main users of the system, began searching for an alternative solution in order to continue offering in-flight Internet to its passengers on long distance flights. In 2011, Lufthansa presented the FlyNet system, which was developed together with Panasonic Aviation [22]. Similar to CBB, Internet access is provided by geostationary satellites. The American company Row44 [23] has also developed an Internet access solution based on geostationary satellites. Its coverage was initially limited to North American airspace only, in order to reduce the costs of leasing satellite transponder capacity [19]. Currently, though, Row44 is advertising global coverage [23].

In parallel to these developments, other possibilities for Internet access have also been investigated, motivated by the high costs and high latency associated with satellite links. With a geostationary satellite located in an orbit 36,000 km above the Earth, the propagation delay of the signal from a terminal to the satellite and back down to a ground station already amounts to roughly 240 ms. Additional delays in the ground network further increase the end to end latency. In 2008, AirCell commenced operations of its Gogo Inflight system [24] in North America, which provides Internet access through a cellular network of terrestrial base stations based on a modified version of the EV-DO Rev A standard [25], which is a data optimized version of the CDMA2000 family of standards. By 2010, Gogo reportedly had coverage across the entire contiguous United States as well as Alaska. Gogo has been adopted by a number of airlines, including major commitments by Delta and American Airlines. As of 2011, Gogo was reportedly planning to extend its network by upgrading to EV-DO Rev. B and directional antenna technology [19]. However, such a cellular system is inherently limited to continental areas, and the costs of installing and operating base stations may not be justified in remote regions with a low flight density.

2.2. Aeronautical Ad Hoc Networks

The drawbacks of cellular and satellite solutions have motivated investigations into Aeronautical Ad Hoc Networks (AAHNs) as an alternate means of providing in flight Internet access. By relaying packets from aircraft to aircraft via an air to air data link, the connectivity from terrestrial base stations can be extended into remote or oceanic regions. In 2007, the AeroSat company performed flight tests demonstrating the feasibility of relaying a videoconference from a ground station via an intermediate aircraft to a second aircraft over the Atlantic ocean in a multihop fashion and has been examining the economical potential of this technology [7]. In the last years, the concept of aeronautical ad hoc networks formed by commercial airliners has also been attracting a growing amount of attention from the research community (see e.g. [6] [8] [26]). Fig. 2.1 shows a picture of the ad hoc network that could potentially be formed by commercial airliners over the North Atlantic.[1] This image has been simulated by us based on a data base of current airline schedules and flight routes provided by the company Innovata LLC [28].

We can classify the research activities in the area of aeronautical ad hoc networks into two main fields: AAHNs for purposes of Air Traffic Control (ATC) or Airline Operational Communications (AOC), and AAHNs as a means of providing in-flight Internet access to passengers. Whereas the ATC and AOC applications mainly rely on short, infrequent messages and may be able to tolerate delay in the order of seconds, in-flight Internet access for passengers is a much more demanding application. Passengers will likely expect onboard Internet access to be comparable in their experience to the case of Internet access at home, in the office, or in a public WiFi hotspot. This will lead to high requirements regarding bandwidth and delay.

[1] The picture was created using Google Earth, which uses satellite imagery from TerraMetrics [27].

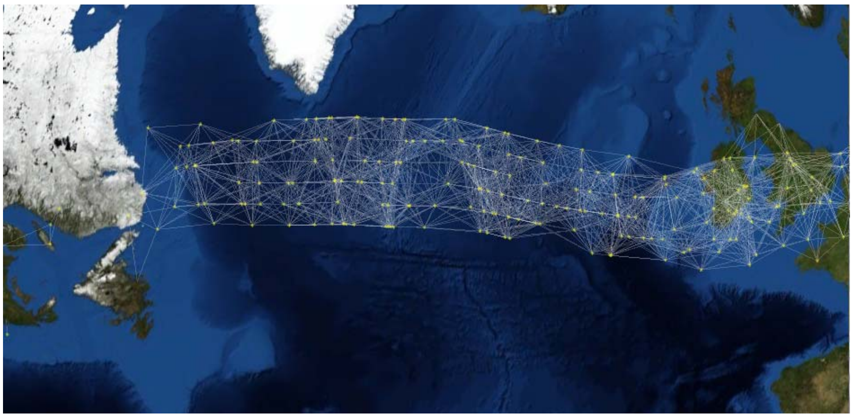


Figure 2.1.: Envisaged aeronautical ad hoc network over the North Atlantic.

So far, it appears that more work has been performed regarding the ATC/AOC services: A number of general considerations regarding AAHNs for such operational services can be found in [8], including a brief survey of potential air to air data links. VHF radios with omnidirectional antennas, Ka-band radios with phased array antennas, and optical links are considered. It is noted that VHF links only provide a limited data rate. The Ka-band link offers higher data rates and can change their transmit and receive directions very quickly. Optical links offer still higher data rates, but need to be mechanically steered and thus do not offer the flexibility of the Ka-band links. In [29], Tu and Shimamoto address the scenario of aircraft flying in oceanic regions sending periodic position reports to ground stations. They propose a channel access scheme assigning each aircraft a dedicated time slot for transmission and a simple position based routing scheme. In [30], Besse *et al.* investigate the feasibility of an AAHN in high density continental airspace over France. They show that the expected maximum throughput of the network is higher than what can be provided by the current air/ground (A/G) data link, VDL Mode 2 [31].

Sakhaee *et al.* have presented their vision of the "Global In-Flight Internet" [6] for passenger communications. They identified that traffic will concentrate around the Internet gateways connecting the AAHN to the ground network and have proposed a scheme for caching web content on board aircraft in order to reduce the amount of traffic that must be handled by these gateways. Medina *et al.* have investigated the feasibility of an AAHN over the North Atlantic [9] and over continental European airspace [32] using real flight data and have come to the conclusion that especially the North Atlantic region is very attractive for an AAHN due to the specific characteristics of the air traffic patterns there. A similar analysis for the North Atlantic has been performed by Kingsbury in [26], estimating the data rate that can be provided to each aircraft by approximating the capacity of air to air links by means of the Shannon capacity and applying a max-min fair algorithm to assign resources among the aircraft. Kingsbury reports that, if all aircraft flying across the North Atlantic participate in the AAHN, a theoretical median downlink capacity of 2.83 Mbps can be provided to an aircraft located at the middle of the ocean. Those aircraft that are closer to the coast will receive a higher data rate. In this work, we also consider satellite gateways in addition to ground stations near the coast and therefore expect a significantly higher data rate.

All of these previous works have in common that they consider an AAHN connected to the ground network only by means of terrestrial base stations, typically at the edges of the network. Heterogeneous connectivity by both air/ground and satellite links is not considered, nor is gateway selection addressed.

Several authors have addressed routing in AAHNs, a problem that applies equally to both goals of providing ATC/AOC services and passenger Internet access. Most authors identify the high mobility of the aircraft as a key factor and focus on position based routing solutions in order to deal with the rapidly changing topology (e.g. [29], [33], [34]). Medina *et al.* have proposed an opportunistic geographic forwarding and Internet gateway assignment algorithm designed specifically for aeronautical networks in [35], which attempts to minimize congestion and maximize overall throughput in the network.

Typically, security is not considered in the previous work on aeronautical ad hoc networks. Security is a much more critical topic for ATC and AOC applications than for passenger communications. Some general security considerations regarding the introduction of digital communications technology and Internet protocol based networking into the aeronautical domain are given in [36] and [37]. One of the main results is that ATC/AOC services should be handled physically separately from APC services to guarantee the security of ATC/AOC services.

The economical aspects and potential business cases for an AAHN have also been investigated in several recent publications ([38], [39], [40]). Interestingly, all of these focus on the North Atlantic region due to its high traffic density and lack of ground infrastructure and see this area as the most promising one for an AAHN to be successful.

Bhadouria [38] concludes that an AAHN over the North Atlantic may be feasible for in flight Internet access, but further work, both technological and economical, is needed. Campos [39] mainly sees potential for such a network to improve the efficiency of airline operations. Watkins [40] concludes that an ad hoc network would not increase the number of passengers using in-flight Internet services. However, it would offer airlines the potential for cost reductions in comparison to satellite access.

2.3. Relation to Wireless Mesh Networks

In many ways, aeronautical ad hoc networks are closely related to Wireless Mesh Networks (WMNs) [41], which have recently attracted a growing amount of interest as a means of providing Internet access especially in rural communities or campuses. WMNs are organized into two hierarchical layers: mesh clients and mesh routers. Mesh clients may be mobile, nomadic, or static nodes that attach to a mesh router for Internet connectivity. The mesh routers form the infrastructure backbone of the WMN and are typically static infrastructure nodes. Some mesh routers have a connection to the Internet, while others may not. To allow all mesh clients access to the Internet, mesh routers run a routing protocol in order to forward packets through the backbone, potentially over multiple hops, until they reach a mesh router with a connection to the Internet.

The aeronautical networks considered here exhibit a similar hierarchical structure: The mesh clients correspond to the devices used by the aircraft passengers, i.e. smart phones, laptops, or seatback devices. The mesh routers correspond to the routers on board the aircraft themselves. Both WMNs and the AAHN share the same purpose: to allow users to access the Internet. Traffic between two users who are both within the WMN or AAHN is not ruled out, but is certainly not the typical use case. In contrast to some ad hoc networks, power constraints are typically not an issue for the mesh routers or for the airborne routers, thereby allowing more complicated processing and proactive signaling.

Despite these similarities, a number of important differences exist between WMNs and AAHNs as well. In WMNs, clients may roam between mesh routers. Obviously, this is not possible in an AAHN, since a passenger stays on board the aircraft for the entire flight. On the other hand, whereas mesh routers in WMNs are typically static, the routers in an AAHN are very mobile. Although the relative positions of the mobile routers, or aircraft, in the AAHN are very stable, the connectivity of the AAHN to the terrestrial Internet gateways is constantly changing, as the cloud of aircraft flies past the ground stations.

However, the major difference between WMNs and AAHNs is that a good deal of network optimization can be done before a WMN is deployed [42]. Given certain assumptions on the expected network traffic, the optimal number and location of mesh routers can be determined such that satisfactory network performance can be assured. This is referred to as the Gateway Placement Problem (GPP). However, the possibilities for such planning in AAHNs are very limited. Air/ground base stations can only be placed in a small number of locations at the coast, and the location of routers within the network is a topic for flight planning, not for network planning. Thus, since the deployment of routers in an AAHN cannot be optimized to fit to the traffic, it is all the more important that the network can achieve the best possible performance given a certain topology and set of traffic demands. This can be done by distributing the traffic in the network in the most efficient way.

2.4. Relation to Vehicular Networks

A large amount of work has also been performed in the area of vehicular, or car to car, networks. Notable research initiatives include the Car to Car Communications Consortium (C2C-CC) [43], consortium of major automobile manufacturers, suppliers, and research institutions, or the German FleetNet project [16]. In many regards, such car to car networks are quite similar to the envisaged aeronautical ad hoc networks. In both cases, multi-hop communication in a network of moving nodes is considered. The network nodes may be mobile networks, not just mobile hosts. However, there are again a number of significant differences between car to car and aeronautical networks. The main focus of the former is communication between vehicles inside the ad hoc network, for traffic safety applications such as warnings about road hazards or for collision avoidance [44]. Internet access is also considered, but is not the primary goal of vehicular networks. In contrast, Internet access is the primary purpose of the aeronautical networks considered here. The network architecture proposed by the C2C-CC is described in [17]. For routing inside the vehicular ad hoc network, the C2C-CC proposes to add an intermediate layer between the Medium Access Control (MAC) layer and the network (IP) layer. For the forwarding of packets, the C2C-CC suggests position based forwarding. This reflects the nature of the services that are targeted for vehicular communications. Much of the information distributed by these traffic safety applications is only relevant in a certain geographic area. For example, a warning about an oil spill on the road may only need to be distributed within a radius of a few hundred meters around the site of the oil spill. Position based forwarding facilitates concepts such as geocasting of information to all, or only some, vehicles within a defined geographic region [17]. An additional advantage of geographical forwarding is its stateless nature, i.e. packets are forwarded based only on current position information, and without searching for an end to end path in the network [45]. This is particularly useful in networks with a very dynamic topology, which is encountered e.g. on roads with two-way traffic, or around intersections. However, in this work, we are interested in classical unicast services, in which the geographical position of the host does not matter. Also, we will see later in Chapter 4 that the topology of the aeronautical ad hoc network investigated here is quite stable. Therefore, the main advantages of position based forwarding do not apply to our case.

Some research has also been performed on Internet access for passengers on high speed trains. For an overview, see the survey by Fokum and Frost [46]. However, ad hoc communication from train to train is not considered for this purpose, since it is more effective to install infrastructure alongside the tracks. Inside the train, passengers connect to a WiFi hotspot, as is also assumed in our network architecture. The problem of IP mobility for mobile networks is also faced by passenger trains, and is discussed e.g. in [47] by Pareit *et al.* The authors discuss the usage of the NEMO protocol in a railway environment. Direct train to train communications has been considered for purposes of railway collision avoidance by Mayer zu Hörste *et al.* in [48]. The authors consider the periodic transmission of position information by trains. However, these beacons are not forwarded over multiple hops, nor are Internet access or general IP issues addressed. To our knowledge, multi-hop communication in railway ad hoc networks has not been considered.

2.5. Internet Gateways

The generic structure of an aeronautical ad hoc network is shown in Fig. 2.2. The ad hoc network is connected to the Internet by means of A/G links and satellite links. Those aircraft that have a direct A/G or satellite connection and make this connection available to other aircraft are referred to as *Internet Gateways* (IGWs). The Mobile Router (MR) on board the aircraft relays IP packets to an Access Router (AR) that is

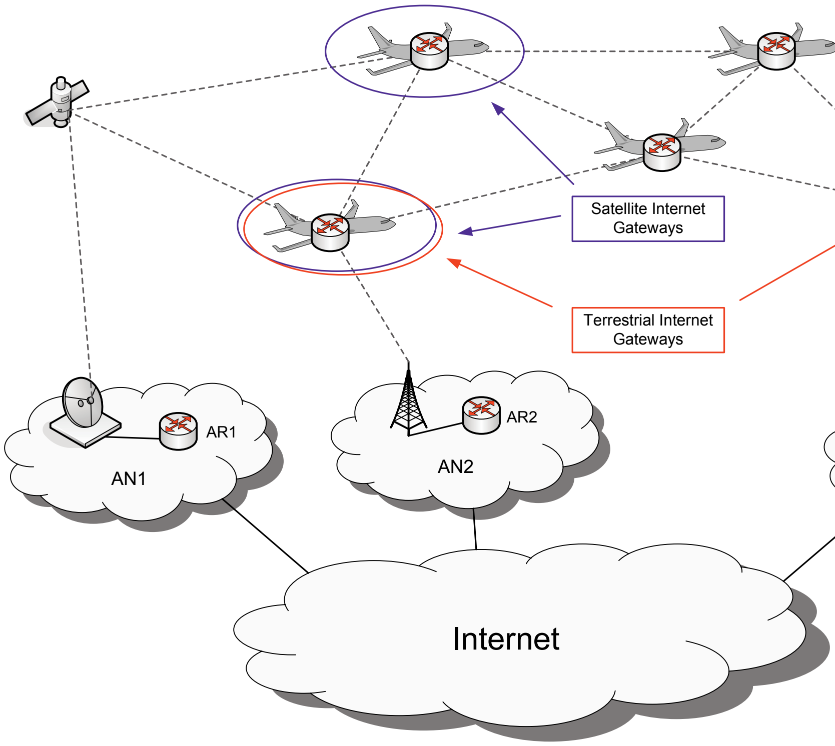


Figure 2.2.: Schematic view of aeronautical ad hoc network.

located in the ground network, or access network (AN). The link between the MR and the AR may include a terrestrial ground station or a satellite Earth station and satellite. However, these wireless links are outside the scope of the ad hoc network. In the rest of this thesis, we distinguish between two kinds of Internet gateways: *terrestrial* Internet gateways, which are aircraft that connect to ground stations, and *satellite* Internet gateways, which are Internet gateways that connect to the Internet via a satellite link. Of course, as shown in Fig. 2.2, an aircraft may be a terrestrial gateway and a satellite gateway at the same time. Satellite gateways may be located anywhere in the ad hoc network. On the other hand, connections to terrestrial base stations are typically made at the geographic edges of the network, at least if ad hoc networks in oceanic regions are considered.

Since the purpose of the ad hoc network considered here is to provide Internet connectivity to the aircraft, practically all network traffic will flow through these gateways. This makes Internet gateways more likely to suffer from congestion than other nodes in the network. Therefore, the decision, which gateway is to be used by which aircraft, has a large impact on the overall network performance.

Sometimes, the term IGW is also used to refer to the node that has both a wired interface towards the ground network and a wireless interface to the ad hoc network

[13]. Here, this would mean that the role of IGW would be located within the network of the service provider operating the network of base stations in the case of A/G links and within the ground network of the satellite service provider in the case of satellite links. It is reasonable to assume that the the base stations will be dedicated base stations for air/ground communications, as in the Aircell network discussed in Section 2.1. However, the operator of the satellite network is providing services to many different customers, most of whom have nothing to do with the aeronautical ad hoc network. It is unlikely that the satellite operator will support any functionality that is specific to the AAHN, such as an AAHN routing protocol or mechanisms for Internet gateway discovery or selection. The role of the satellite operator is assumed to be limited to leasing transponder bandwidth to the operator of the AAHN. Therefore, we do not follow this classification, and use the term Internet Gateways to refer to those aircraft that connect to a ground station or satellite. This has the additional advantage that the ad hoc network itself is agnostic of the access links that are used by the Internet Gateways to connect to the Internet, facilitating the integration of heterogeneous access link technologies.

The view of mobile nodes as Internet Gateways as adopted here has been previously proposed by Cabrera *et al.* in [49]. The motivation of Cabrera *et al.* was to allow mobile nodes to access the Internet through a WiFi Access Point over a Mobile Ad Hoc Network (MANET) although the Access Point has not been configured to support a MANET. The operation of these gateways will be described in more detail below, after a brief introduction to the concepts of Internet Gateway discovery and Internet Gateway selection, which are essential for Internet access in an ad hoc network.

2.5.1. Internet Gateway Discovery

In case the decision, which gateway to use, is made locally by each mobile node of the ad hoc network, and not by some entity within the terrestrial network, the mobile node must first become aware of at least one Internet gateway that it can reach. This process of finding available Internet gateways is known as *Internet gateway discovery.* Three different methods of gateway discovery are distinguished in the literature: proactive (e.g. [13]), reactive [50], and hybrid [51] discovery. In proactive discovery, all gateways periodically broadcast Internet Gateway Advertisement (IGWADV) messages, which are forwarded by the nodes in the ad hoc network. They can be flooded through the entire ad hoc network, or only a limited part of the network in order to reduce the overhead. However, it should be guaranteed that every node receives IGWADVs from at least one gateway. In reactive gateway discovery, IGWs do not transmit periodic advertisements. Rather, a node that requires Internet connectivity generates an Internet Gateway Solicitation (IGWSOL) message that is then flooded through the network. A gateway receiving an IGWSOL responds with an IGWADV that is unicast back to the soliciting node. In general, proactive gateway discovery leads to a higher overhead due to the periodic flooding of advertisements. Reactive discovery generally incurs much lower overhead, but suffers from longer delays before Internet connectivity can be established, since the node only triggers an IGWSOL upon demand and must then wait until it receives a response from at least one gateway. Due to the lower overhead, reactive discovery is often preferred in energy constrained networks. However, it has been reported in [52] that in networks with a large number of active users requiring an Internet connection, the overhead of reactive discovery can even be higher than for proactive discovery. Finally, hybrid gateway discovery combines proactive and reactive discovery. The IGW generates periodic IGWADVs, but these are only flooded in a limited region close to the gateway itself, in order to limit the total overhead. Nodes that are located outside this region must reactively solicit a gateway. The region in which proactive flooding takes place is typically defined by the maximum number of hops, i.e. the number of times that an IGWADV may be forwarded before it is discarded by the receiver.

Since the purpose of an AAHN is to provide Internet access to all aircraft, all aircraft in the network will require a route to the Internet at all times. Also, energy consumption of the communications equipment of the aircraft is not a major issue. In this case, proactive flooding of advertisements is the most attractive gateway discovery mechanism.

2.5.2. Internet Gateway Selection

Whenever there is more than one Internet gateway in the ad hoc network, nodes or traffic flows must somehow be assigned to these gateways. This task of *Internet gateway selection* can either be performed by a central entity within the ground network (network based gateway selection), or by the wireless nodes themselves (node based gateway selection). Node based selection is the much more common case. An example for network based gateway selection is proposed by Galvez *et al.* in [53]. The central controller monitors the traffic at all gateways and attempts to balance the number of flows handled by each gateway.

In an aeronautical environment, it is likely that each airline will wish to implement its own gateway selection scheme based on business policy, or the kind of service that they want to provide to their customers. In this case, the airlines will prefer to stay in control of the decision which access networks to use under which conditions, and will likely not delegate the task of gateway selection to a central entity in the ground network, which is responsible for all airlines. Thus, we consider node based gateway selection to be the more interesting solution for a practical deployment. However, centralized solutions also have their advantages: If the central entity is able to collect current information regarding the network topology, traffic demands, and link capacities, it would be able to calculate a gateway selection solution that minimizes some global cost function. A distributed, node based selection scheme can try to approximate, or achieve, such a centralized solution. It is well known that distributed routing algorithms such as the Bellman-Ford algorithm (see e.g. [54]) can run in a distributed manner and achieve the true shortest path solution.

It is conceivable that a different gateway is used by a flow in the upstream direction to send packets to the Internet than in the downstream direction to receive packets from the Internet. However, this can lead to a large asymmetry between the uplink and downlink path, which is generally considered harmful, especially for the performance of TCP [55]. Issues related to the handling of packets by the Internet Protocol (IP) can also arise if the upstream and downstream gateways are different. Therefore, we will assume that the same gateway is used in both directions.

In the last years, a number of different Internet gateway selection algorithms have been proposed for wireless ad hoc networks in the literature. IGWADVs typically carry some kind of metric that allows the nodes to select the best IGWADV to use. The simplest and most robust metric is the distance of the Internet gateway from the node in terms of hops [13]. This requires every node forwarding the IGWADV to increment the hop count of the message by one. The obvious drawback of hop count based gateway selection is that the traffic load is not considered. The random distribution of nodes in the network, or random traffic patterns, may result in some gateways being overloaded, while other gateways are idle, leading to unnecessarily high packet delay or even packet loss.

Therefore, more advanced metrics attempt to balance the traffic load of the gateways. This can be done by advertising the capacity and current traffic load of the IGW directly [56], allowing nodes to select the gateway with the lowest utilization, or ratio of load to capacity. However, the actual capacity of a gateway is not easy to define. Due to interference on the wireless channel, or the channel access scheme itself, the actual capacity may be far from the nominal data rate of the physical layer, thereby reducing the effectiveness of the load metric. For example in a TDMA network in which the number of slots assigned to a link depends on its traffic load, higher traffic load on a link may also result in higher capacity.

Even if the gateway's capacity can be estimated reliably, both the hop count metric as well as the gateway load are one sided: the hop count metric only considers the topological distance from the gateway, but not the traffic load, either at the gateway or in the network. On the other hand, the gateway load metric only considers the load of the gateway, but not its distance. In some cases, it may be beneficial to select a gateway with slightly higher load, because it is significantly closer than another gateway with slightly lower load. This has led to the introduction of compound metrics that are computed as a weighted sum of multiple criteria such as hop count, load, or remaining energy in case of battery powered nodes [57], [15]. This allows properties of both the gateway and the path between the node and the gateway to be considered in a single metric. Still, an open question is how the weights of the different criteria should be selected. In fact, it can be seen from the results in [57] or in our previous work in [58], that it would be necessary to constantly adapt the weights to the current network conditions in order to achieve optimal performance at all times. When the traffic load in the network is very low, it is best to focus on hop count as a selection criterion. But when the traffic load increases, the capacity of the gateways and along the path becomes more and more important. To our knowledge, no solutions have been proposed so far for such a dynamic tuning of the weights.

Brännström *et al.* have proposed the Running Variance Metric (RVM) [59] for networks using proactive gateway discovery to estimate the congestion of the IGW and the path to the node in a single metric. All nodes in the network measure the variance of the time that elapses between the reception of two subsequent IGWADVs from the same gateway. The RVM is then computed as an exponentially weighted running average of these inter-arrival times. The basic idea is that higher traffic load, either at the gateway or at an intermediate node somewhere along the path, will lead to a higher variance in the inter-arrival times of the IGWADVs, mainly due to the queueing of packets and contention for the wireless channel. Each node then selects the gateway with the lowest RVM as its gateway. This metric has the advantage that it combines the effects of the path length and the traffic load at the gateway as well as along the path, without the need to choose any weighting coefficients. In addition, it does not incur any additional overhead, since the selection metric is not carried explicitly by the advertisements but is based on measurements at each node. However, it is closely tied to the underlying link layer (IEEE 802.11 WiFi was assumed in [59]), and assumes that the data packets will experience a similar treatment by the network as the advertisements. This may not always be the case, though, as advertisements are typically broadcast, whereas data packets are unicast traffic, and advertisements are typically much smaller than data packets.

All of these approaches have in common that they separate the problems of routing and gateway selection, i.e. a gateway selection process selects the gateway that will be used, and the routing protocol is then responsible for getting the packets there. In the case of the Running Variance Metric, the path along which the packets are routed can be different than the path taken by the IGWADVs, upon which the decision to use this gateway was based.

2.5.3. Opportunistic Internet Gateways

In [49], Cabrera *et al.* have proposed an algorithm that allows nodes in an ad hoc network to decide if they should act as Internet gateway for the other nodes in the ad hoc network. The underlying assumption is that the access router is unaware of the ad hoc network and is only configured to provide connectivity to nodes that are directly connected to the access router itself. This assumption is similar to the assumption that we have made about the ground and satellite networks and thus could be adopted for our scenario as well. This solution will be described briefly below.

Of all nodes within range of the access router, one node acts as default gateway and generates gateway advertisements which are forwarded through the ad hoc network. These nodes are referred to as *opportunistic* Internet gateways. When this node leaves the coverage area of the access router, it stops acting as gateway and no longer sends advertisements. The other nodes will subsequently detect that there is no longer an active Internet gateway. All nodes within range of the access router then start random timers such that they will configure themselves as gateways upon expiration of the timer and commence sending IGWADVs. If a node receives an IGWADV from another node before its timer has expired, it recognizes that another node has taken over the role of IGW and cancels its timer. This solution allows ad hoc networks to be attached to wireless access routers without the need for the access router to be aware of the ad hoc network. In our AAHN, this approach could be applied to the satellite gateways.

However, the scheme proposed by Cabrera *et al.* has a number of drawbacks. To limit the overhead due to the IGWADVs, only one opportunistic IGW is allowed at each access router, and all traffic destined to or from the access router into the wireless network must flow through this IGW, thereby creating a potential bottleneck. Also, the process for configuration of nodes as opportunistic gateways relies on random timers and does not consider any kind of performance criteria, such as link quality or the expected duration for which the mobile node will be able to act as IGW. Therefore, we are interested in a more effective way of integrating opportunistic Internet gateways into the ad hoc network that does consider such performance criteria. The definition of such a policy for the selection of opportunistic Internet gateways is addressed in this thesis.

3. IPv6 Network Architecture

Although the main focus of this thesis will be on algorithmic aspects of gateway selection and routing, it must be kept in mind that the purpose of the envisaged ad hoc network is to provide Internet access to the passengers on board the aircraft. Therefore, the proposed algorithms must be compatible with the current Internet protocol stack in order to be able to provide such end to end IP connectivity. In this chapter, we design a protocol architecture to allow the integration of the ad hoc network into the end to end IP environment.

Due to the increasing importance of IPv6 in the future, the discussion here is limited to IPv6 and does not address the currently predominant IPv4. If backwards compatibility to older IPv4 devices used by passengers is required, this can be achieved by letting the Mobile Router run both IPv4 and IPv6 on the mobile network in dual stack mode. IPv4 packets between the onboard users and the Internet can then be tunneled over IPv6.

3.1. Network Architecture

The development of aeronautical communication networks has historically been driven by the operational requirements of ATC and AOC services. A recent trend is the transition from traditional voice based ATC services to data based services, as described in the Communications Operating Concept and Requirements for the Future Radio System (COCR) [60], a report which has been jointly published by the FAA and EUROCONTROL as the air traffic safety organizations of the United States and Europe, respectively. This trend has prompted the International Civil Aviation Organization (ICAO) to standardize an Aeronautical Telecommunications Network (ATN) based on the Internet Protocol suite [61]. This so-called ATN/IPS is intended to replace the existing ATN, which is based on the ISO/OSI protocol stack [62]. All of these activities have focused on IPv6 and have considered IPv4 only as a legacy technology that will need to be supported for purposes of backwards compatibility with already existing services.

Recent research activities in the field of aeronautical networking, such as the European projects NEWSKY [63], [64] and SANDRA [65], have investigated design options and demonstrated the feasibility of IP based networking for aviation. Although they focused mainly on operational services, i.e. ATC and AOC, they have also looked at the possibility of integrating passenger services (APC) in the same network. Results from these projects partially contributed to the ICAO ATN/IPS standardization activities. Although certainly possible from a technical perspective, the integration of ATC/AOC and APC in a single network is seen to pose a high security risk [37]. Therefore, a so-called *constrained* network architecture has been proposed by the NEWSKY project

[18], [66], in which APC services are handled by a network that is physically separated from the operational network for ATC/AOC services. A so-called *unconstrained*, or *integrated* network architecture has been proposed as a long term solution. In the integrated architecture, operational and non-operational services are handled by the same physical network, and appropriate Quality of Service and security measures are deployed to ensure that the non-operational service do not interfere with the operational services. In this work, we will use the constrained APC network architecture described in [18], which is depicted in Fig. 3.1. It is assumed that an airline has a contract with a primary global Internet Service Provider (ISP). This service provider may have peering agreements with other network operators, such as the satellite network operator in Fig. 3.1, thereby allowing the aircraft to access the Internet through other providers than its own ISP. The networks providing air/ground communications are denoted Access Networks (ANs). A router in the AN that has a wireless air/ground interface is denoted an Access Router (AR).

From a functional point of view, the unconstrained architecture is similar for APC, the major difference being that the same network elements also handle ATC and AOC traffic in parallel. The passenger devices are termed Mobile Network Nodes (MNNs). The key network elements on the aircraft's onboard network are the Security Access Gateway (SAG) and the Mobile Router (MR). The SAG performs encryption of all data in order to preserve the confidentiality of passengers' data, and the MR performs the mobility related signalling. On the ground, the key elements of the NEWSKY network architecture are the Home Agent (HA) and the peer SAG in the ground network. The Correspondent Node (CN) is the ground node with which the MNN is communicating. The role of the MR and HA within the Network Mobility Basic Support protocol (NEMO BS) [67] will be described in detail below, in Section 3.1.2. The ground SAG decrypts traffic coming from the aircraft and encrypts data flowing towards the aircraft. The airborne SAG can also act as a TCP Performance Enhancing Proxy (PEP). The purpose of the PEP is to improve the performance of the Transmission Control Protocol (TCP) over wireless links, as the performance of standard TCP according to RFC 793 [68] is well-known to suffer in case of packet losses over a wireless link or when a link exhibits high delay [69]. The end-to-end TCP connection between a MNN and a CN is broken up into three segments: one segment between the MNN and the PEP onboard the aircraft, a second segment between the airborne PEP and a PEP in the ground network, and a third segment between the ground PEP and the CN. This allows the middle segment to operate with TCP parameters that have been adapted specifically to the characteristics of the wireless link. This TCP splitting is transparent to the end nodes, but it works only if the end nodes do not apply network layer encryption that prevents the PEPs from accessing the TCP headers. For the best performance improvement, the PEPs should be located as close as possible to the endpoints of the wireless link. However, in order to be able to read the TCP headers, the PEP needs to be placed outside of the end points of the security and mobility tunnels, which are terminated by the SAGs and the MR or HA. For a detailed discussion of PEPs in the NEWSKY architecture, see [70].

The core problems addressed by the IPv6 based NEWSKY network architecture as summarized in [18] are Quality of Service and IP mobility. Quality of Service measures are required on the one hand to distinguish the different service domains, i.e. ATC, AOC, and APC in case of the unconstrained network architecture. Obviously, ATC services, which are flight critical, must always have a higher priority than passenger entertainment services. On the other hand, different services within the same service domain may have different priorities. For example, an ATC clearance sent from an air traffic controller to a pilot must have a higher priority than weather information, which may also belong to the ATC domain. In the NEWSKY architecture, QoS is handled by the DiffServ framework [71], [72], which allows different forwarding behavior to be defined for different service classes. These service classes are distinguished by the DiffServ Code Point (DSCP) field in the header of the IP packets. It is assumed that the Security Access Gateway and Mobile Router perform this DiffServ tagging, and not the passengers' devices. In this way, the handling of different service classes remains under control of the network operator.

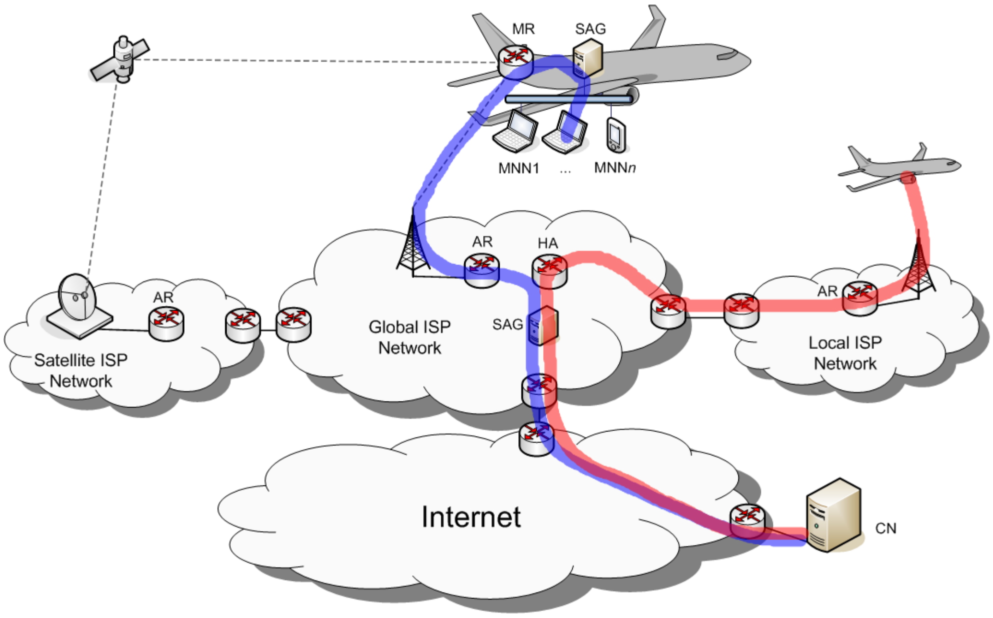


Figure 3.1.: Newsky APC network architecture.

The Connexion By Boeing (CBB) system relied on a different network architecture, based on IPv4, in which the mobility problem was solved by using the Border Gateway Protocol (BGP) [73]. BGP is the dominant inter-domain routing protocol in the Internet. This approach requires no custom protocols to handle mobility. Instead, the onboard network is a unique Autonomous System (AS) whose network prefix does not change during the flight. The router on board the aircraft is running external BGP to constantly announce the reachability of this AS to the rest of the Internet. This guarantees that the aircraft is always reachable from the Internet through an optimal path, in contrast to a NEMO based mobility solution, as will be seen below. However, since every aircraft is an AS, the number of BGP routing updates increases significantly [74]. It has been reported that CBB alone was responsible for 20% of the BGP routing churn during the short time that it was in use [75].

3.1.1. IPv6 Address Configuration

Before any host on the aircraft can exchange data with an Internet node, the Mobile Router must connect to an Access Network and configure a valid IP address from this AN.

The Access Routers (ARs) of an IPv6 Access Network periodically transmit Router Advertisement (RA) messages advertising the network prefix that has been assigned to the router, thereby allowing hosts to configure a topologically correct IP address from this prefix. Using Stateless Autoconfiguration [76], nodes can construct an IPv6 address by appending their EUI-64 identifier, which is based on the MAC address of the interface, to the advertised prefix. Since MAC addresses are not guaranteed to be unique and can be modified by the user, the host must in general perform Duplicate Address Detection (DAD) [77] in order to verify that the address that it has constructed is not currently in use by any other node on this network. Only when DAD has been performed and no addressing conflict has been detected is the node allowed to use the address to send traffic. In an aeronautical network, it is possible to accelerate the process of address autoconfiguration, since each aircraft is assigned a unique 24 bit identifier by ICAO, the International Civil Aviation Organization. This identifier is also used in transponders to allow unique identification of an aircraft for purposes of air traffic control. Here, it can also be used to construct a globally unique IPv6 address for the aircraft by means of stateless autoconfiguration, instead of using the EUI-64 identifier. Since ICAO identifiers are assigned by a central authority and are guaranteed to be unique, it is possible to deactivate the Duplicate Address Detection mechanism, thereby reducing the latency of the attachment to a new access router by about one second [78]. This mechanism of stateless autoconfiguration has been designed for Ethernet-like networks in which all nodes are on the same network as the router and all nodes on the network are able to hear all other nodes on the network. In a wireless ad hoc network, the situation is more complex, since not all nodes may be within direct range of each other, even though they are on the same IP network. The IETF formed the AUTOCONF Working Group [79] to design a solution for IPv6 address autoconfiguration in ad hoc networks. However, this working group was closed due to a lack of agreement before any solution could be standardized. Therefore, IP address allocation in an ad hoc network is still an open issue.

3.1.2. Overview of NEMO Basic Support Protocol

Whenever a mobile host connects to a new Access Network due to its movement, it will configure a new IP address from this network, according to the procedure of Stateless

Autoconfiguration as described in the previous section. Since TCP sessions are identified in part by the IP addresses of the communicating hosts, this change of the IP address would break the end to end TCP session. If end to end TCP sessions can be maintained despite such a handover to a new Access Network and a change in the node's IP address, session continuity is said to be fulfilled. Global reachability is achieved if the mobile host can always be reached from the Internet even though its current IP address may change.

The IETF has standardized Mobile IPv6 in RFC 3775 [80] as a solution to provide session continuity and global reachability despite a host's mobility between different access networks. Mobile IPv6 provides a mobile host with a special IP address, denoted the host's Home Address (HoA) from its home network, which remains constant as the host moves and under which the host is always globally reachable. In every new Access Network, the host configures a new Care of Address (CoA) using the address autoconfiguration mechanism described above. It registers this CoA with its Home Agent (HA), which is located in its home network. Whenever a Correspondent Node wishes to send data to the mobile host, it addresses this data to the HoA, since the mobile host's current CoA is not known to the CN. The HA intercepts the packets when they arrive in the home network and tunnels them to the mobile host's current CoA. RFC 3775 also provides a mechanism for route optimization, allowing the mobile node and the CN to exchange packets directly, instead of having to send them via the HA. This reduces the packet latency, since packets no longer need to take the detour via the Home Agent, and also reduces the risk of packet loss due to failure of the HA.

However, as explained in Sec. 3.1, each aircraft in the AAHN is not just a mobile host, but an entire mobile network, consisting of a Mobile Router and a potentially significant number of Mobile Network Nodes (MNNs), which are connected to the onboard network of the MR. Therefore, Mobile IPv6 is not directly applicable to our scenario.

To address such cases, the IETF has also standardized the NEtwork MObility Basic Support protocol (NEMO BS) in RFC 3963 [67]. Essentially, NEMO BS is an extension of Mobile IPv6 allowing it to handle not just mobile hosts, but entire mobile networks. A rationale for choosing NEMO to address the mobility issue in aeronautical networks is given in [81]. This protocol is common to the ICAO ATN/IPS and the Newsky network architecture, which is assumed here. Therefore, we will provide a short overview of the functionality of NEMO BS in this section.

In NEMO BS, the MR takes care of all mobility signaling on behalf of the MNNs. This signaling is transparent to the MNNs, who are completely unaware that they are part of a mobile network. The MR is configured with a network prefix, the Mobile Network Prefix (MNP), from its Home Network. The MNP can be assigned either statically or dynamically, e.g. via DHCPv6 [82]. This prefix is advertised on the onboard network, e.g. a WiFi hotspot inside the aircraft cabin. The MNNs that are attached to the MR obtain an IP address from this prefix, regardless of whether the MR is at home or not, thereby hiding the mobility from the hosts. Apart from this difference, the operation of NEMO is very similar to Mobile IPv6: As the MR moves away from its Home Network to a different Access Network, it configures a CoA from the network prefix that is advertised by the new Access Router. The MR registers this CoA with its HA, which is located in the MR's Home Network, by sending a Binding Update (BU) message to the HA. The HA responds to a BU with a Binding Acknowledgement (BA) message, and a bidirectional tunnel is set up between the MR and the HA. Afterwards, the MR must send a new BU to the HA periodically, at least every seven minutes, in order to indicate that it is still reachable under the current CoA. Whenever an MNN sends an IP packet to the Internet, it is encapsulated by the MR and tunneled from the MR to the HA. The HA then decapsulates the packet, and it is routed to its destination as if it had originated from the Home Network. Conversely, if a Correspondent Node (CN) wishes to send a packet to a MNN, the packet is routed to the Home Network, where it is intercepted by the HA and encapsulated and tunneled to the MR, which then forwards the inner packet to the MNN. Due to the binding of the MR's CoA at the Home Agent, the MNNs are always reachable under the address that they have configured from the MNP. In contrast to MobileIPv6, however, NEMO BS does not provide a mechanism for route optimization.

Fig. 3.1 shows the paths taken by data packets exchanged between two different aircraft and a single Correspondent Node, which is reached over the Internet. As can be seen, all traffic is routed via the Home Agent, which is located within the network of the aircraft's Global ISP.

To comply with the NEWSKY APC architecture, the MR must read the DSCP field from the header of all packets that will be tunneled to the HA and write the value of this field into the DSCP field of the outer IP header, as the inner header will no longer be readable after encapsulation for the MR-HA tunnel.

3.2. Protocol Architecture Goals

The necessary background for the discussion of our aeronautical ad hoc network's protocol architecture has been laid down in Sec. 3.1. As stated there, the basis for our network architecture is the NEMO BS Protocol, since this has been selected as the solution to the IP mobility problem in IP-based aeronautical networks. However, NEMO was not designed with ad hoc networks in mind, but is primarily intended to operate in an environment where the MR attaches directly to an Access Network's AR. Therefore, extensions to the baseline network architecture of Sec. 3.1 are required.

In the following subsection, we formulate a set of requirements that such a solution should fulfill in order to integrate seamlessly with the baseline NEMO architecture and to allow the dynamic routing of flows over different gateways depending on some performance criterion, such as packet delay in the previous chapters. Then, we formulate evaluation criteria that allow us to select the most appropriate protocol architecture solution out of the set of candidates. These different solutions are then discussed in detail and evaluated with respect to these criteria. We have previously published parts of this work in [83]. Based on this analysis, a functional routing architecture for the Aeronautical Ad Hoc Network is defined. This functional architecture will serve as the baseline for the definition of a routing and gateway selection protocol for the AAHN in the following chapter.

3.2.1. Protocol Requirements

The following requirements must be satisfied by the AAHN protocol architecture:

- **Transparency** to the end user: A passenger on board the aircraft must be unaware of the ad hoc network that is connecting him to the Internet. The connection should exhibit similar behavior as what the passenger is accustomed to when using the Internet at home, in his office, or in a public hot spot.

- **No modifications to end user equipment**: The solution must not rely on any modifications to the protocol stack of the end users' equipment on board the aircraft. These devices may belong to the passenger, and it cannot be expected that they will modify their system in order to attain Internet access.

- **No modifications to Correspondent Nodes (CNs)**: The solution must not rely on any modifications to the protocol stack of the Correspondent Nodes, i.e. the communication end nodes in the Internet. These can be arbitrary nodes in the Internet and are completely outside the scope of the AAHN.

- **Session Continuity**: As the aircraft moves and connects to different access networks, open sessions (e.g. TCP sessions) must not be interrupted.

- **Global Reachability**: The solution should allow for global reachability of nodes on board the aircraft network from the Internet. Although this is typically not required by services currently used in the Internet, the solution should not preclude such a scenario in the future.

- **Multihoming** support: The solution should allow for multihoming of the Mobile Router, i.e. for the Mobile Router to attach to and receive an IP address from different access networks simultaneously. For example, the aircraft could connect to the satellite network and the A/G network, e.g. for purposes of redundancy, load sharing, or policy based routing.

- **Dynamic GW Selection** on a flow by flow basis: The aircraft should be able to decide for each flow which access network to use. For example, delay sensitive flows could be routed over the A/G network, whereas less delay sensitive traffic is sent over the satellite.

- **Routing Symmetry**: For each traffic flow, the solution should make use of the same GW in upstream and downstream directions. TCP flows may not behave properly if the upstream and downstream paths have significantly different characteristics regarding packet loss or delay [55]. To prevent this from happening, upstream and downstream traffic of the same flow should be sent over the same Internet gateway.

Most of these requirements were already considered when the NEMO protocol was designed, as is evident from the Network Mobility requirements document [84], which was published by the IETF Network Working Group. The first five of the above requirements are inherently supported by NEMO. The required extensions for support of multihoming by NEMO, i.e. multiple CoA registration [85] and flow bindings [86], have also been standardized as RFCs by the IETF. However, relying only on NEMO to handle the routing within the ad hoc network has serious drawbacks, as will be seen in the discussion of potential protocol architectures for the network in Section 3.3.

Dynamic gateway selection can be implemented by means of *Policy Based Routing* (see e.g. [87] for an overview) and will need to be considered in the definition of the functional architecture. Whether routing symmetry is achieved depends on how the routing of packets inside the ad hoc network is implemented and will also need to be addressed in the functional architecture.

3.2.2. Evaluation Criteria

Before discussing in detail the individual candidate solutions, we formulate a set of evaluation criteria which will be used for the assessment of the candidate protocol architectures:

- **Route Optimality**: An optimal route is one that takes the direct path between the CN and MR. Unnecessary detours through the ground network between the AR and the CN, e.g. via the HA, reduce the optimality of the route. In addition, the most suitable route through the wireless network should be chosen between the AR and the MR, based on some quality metric.

- **Signaling Overhead**: A handover from one Access Router to another leads to signaling overhead due to the need to update the binding of the CoA at the HA. This criterion considers the amount of signaling triggered by a handover as well as the frequency of such handovers. In addition, signaling information may be exchanged between aircraft periodically or on demand for the maintenance of routing tables within the ad hoc network.

- **Data Overhead**: This criterion considers overhead that is encountered with every data transmission, e.g. larger packet sizes due to the encapsulation of data packets.

- **Robustness**: Wireless networks may experience sudden degradations in link quality. Also, single points of failure on the ground may lead to service disruptions. This criterion serves to assess the degree to which the candidate solutions may be affected by such unforeseen events.

- **Protocol Maturity**: Standardized solutions that are in general use outside the aeronautical domain offer several advantages to custom built solutions. They are likely less expensive, can be provided by a larger number of vendors, and may be technically more mature due to more widespread use.

3.3. Candidate Architectures

In the following, four different candidate protocol architectures that have been previously proposed in the literature are analyzed: The Nested NEMO architecture relies only on the NEMO BS protocol. The three remaining solutions make use of an ad hoc network routing protocol within the wireless network. In the IP MANET solution, the ad hoc routing takes place at the IP layer. The Sub-IP MANET moves this task below the IP layer. Finally, the protocol translation solution relies on the Internet gateways to translate the TCP/IP protocol stack to a custom set of protocols that are specifically designed for use within the ad hoc network.

3.3.1. Nested NEMO Architecture

The Nested NEMO architecture is already described in the original NEMO standard [67]. The protocol stacks running on the involved network nodes are shown in Fig. 3.2.

Functional Overview

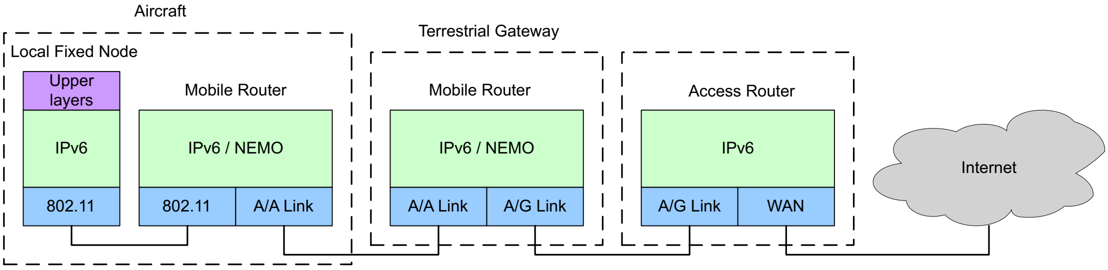


Figure 3.2.: Protocol stack of Nested NEMO architecture.

The NEMO BS protocol allows the mobile network, to which a MR is providing Internet connectivity, to consist not only of hosts, but of additional routers as well. These routers may themselves be Mobile Routers with an attached network. Instead of configuring a CoA from the Access Network, such a nested MR configures a CoA from the MNP of the MR to which it is connected, and registers this CoA with its Home Agent. This creates a so-called *Nested NEMO* structure. An example for this configuration is shown in Fig. 3.3. MR1 and MR4 are directly connected to the Access Router AR, which is advertising the network prefix ANP::. Therefore, MR1 and MR2 configure their CoAs from this prefix. MR2 and MR3 are part of the mobile network of MR1, and configure their CoAs from MR1's MNP. MR1 is referred to as the *root* MR of this Nested NEMO tree. Only Home Agents HA1 and HA2 of MR1 and MR2 are shown. Further levels of nesting are possible if additional MRs connect to MR2 or MR4. With the exception of multihoming, NEMO BS intrinsically supports the basic requirements of our application scenario.

As an example for the Nested NEMO configuration, consider an aircraft flying in the middle of the Atlantic Ocean, connected to an Access Router on the Irish coast via a multihop path. Given a distance of 1500 km to the coast and a maximum transmission range of 370 km for an air/air link and 185 km for an air/ground link, this path will require at least five hops. This implies five levels of IP encapsulation, adding a total of 200 Bytes of overhead to each IP packet generated by the MNN. Considering the maximum packet size imposed by the Ethernet MTU of 1500 Bytes, this overhead is quite significant. In addition, a total of five Home Agents must be traversed by each packet. If the HAs and the CN are alternatingly located in Europe and North America, this could result in an additional 450 ms of delay within the ground network. Here, we have assumed a latency of 90 ms for a transatlantic undersea cable. This value is based on the NTT Global IP Network Service Level Agreement [88]. This significant additional delay would likely eliminate any advantages of the ad hoc network in terms of delay compared to satellite access.

Evaluation of Nested NEMO Architecture

- Route Optimality:

  Nested NEMO networks have a number of well-known drawbacks (an overview is given in e.g. [89] or [90]). Packets that are exchanged between an MNN in a nested network and a CN must travel via the Home Agents of all MRs that lie between the MNN and the AR. The path of a packet from CN to an MNN connected to MR2 is shown in Fig. 3.3. The packet is sent from the CN to the MNN's Home Address, where it is intercepted by HA2. HA2 tunnels the packet to the current CoA reported by MR2, which is configured from the MNP of MR1 (blue line: tunnel MR2-HA2). Hence, the packet is sent to the Home Network of MR1, where it is intercepted by HA1 and tunneled to MR1 (orange line: tunnel MR1-HA1). MR1 decapsulates the packet and forwards it to MR2. This problem is referred to as pinball, or dog-legged, routing. In aeronautical networks, this problem can be particularly serious, since the HAs may be distributed all over the globe, potentially leading to extremely long paths. Thus, Nested NEMO performs quite poorly with respect to the criterion of route optimality.

- Data Overhead:

  In addition to the large delay that is caused by the highly sub-optimal routing, each MR-HA tunnel also leads to an additional encapsulation of the packet, generating a potentially significant amount of overhead. This overhead is not only undesirable because of the increased load on the wireless channel, but also because each additional Byte in the header increases the probability that a packet will need to be fragmented somewhere in the network. Thus, the amount of overhead per data transmission is also a significant weakness of this approach.

- Signaling Overhead:

In contrast to the overhead incurred per data packet, the amount of overhead due to signaling traffic is quite low, since it is limited to the Binding Updates and Acknowledgements exchanged between each MR and HA. Although these exchanges are periodic, their frequency can be as low as once every seven minutes. In [91], a method has been proposed to handle both of these drawbacks by allowing a MR to determine the CoA of the root MR. The MR then notifies its HA of the root MR's CoA, allowing the HA to tunnel packets directly to the root MR, bypassing all other HAs that might lie between the MR and the root MR. The packet is decapsulated at the root MR and must then be routed towards the correct MR inside the Nested NEMO tree. This requires the existence of an additional routing protocol for forwarding packets in the wireless network. The use of OLSR, a link state routing protocol for wireless ad hoc networks, for routing inside a Nested NEMO has been proposed in [92]. From the point of view of data transmission, this turns the Nested NEMO architecture into the IP MANET architecture, which is discussed in the following section. However, the Nested NEMO structure still remains in place for the configuration of IP addresses by the MRs.

- Robustness:

  In our aeronautical scenario, there is little relative movement between two aircraft, i.e. MRs, since all aircraft fly largely in parallel with comparable velocities. Therefore, the tree constructed by Nested NEMO and the CoAs of the MRs will not change often. However, the movement of the entire cloud of aircraft will lead to steadily changing points of attachment at the Access Routers. An aircraft will be in range of a base station for at most ca. 28 min[1]. When the root MR loses connectivity to the base station, all MRs 'down the tree' will also lose their connection temporarily. In Fig. 3.3, assume that MR1 loses its direct connection to AR. MR1 will detect this loss, and configure a CoA from MR4, which has come within radio range. The addresses of the MRs down the tree from MR1 will not be directly affected by this change, since their CoAs are derived from MR1's MNP, which remains unchanged. However, packets may be lost, depending on the time that is required by MR1 to complete its handover from AR to MR4, which may take several seconds. This problem could be handled by multihoming. If MR1 is aware that the link to AR is about to break, it can configure a second CoA from MR4 and switch to using this CoA before the link to the AR breaks.

  The tree structure makes the Nested NEMO approach quite sensitive to failures of any of the routers involved. When traffic flows through multiple HAs, the failure of any one of the HAs will lead to a service disruption. In addition, traffic cannot be routed dynamically inside the wireless network. Instead, it must follow the tree structure of the Nested NEMO. If any router or link of this tree is overloaded, traffic cannot simply be routed along an alternate path to distribute the load more equally. If a router or link of the tree fails entirely, the routers down the tree

[1] Based on an assumed air/ground communications range of 185 km and an aircraft velocity of 800 km/h

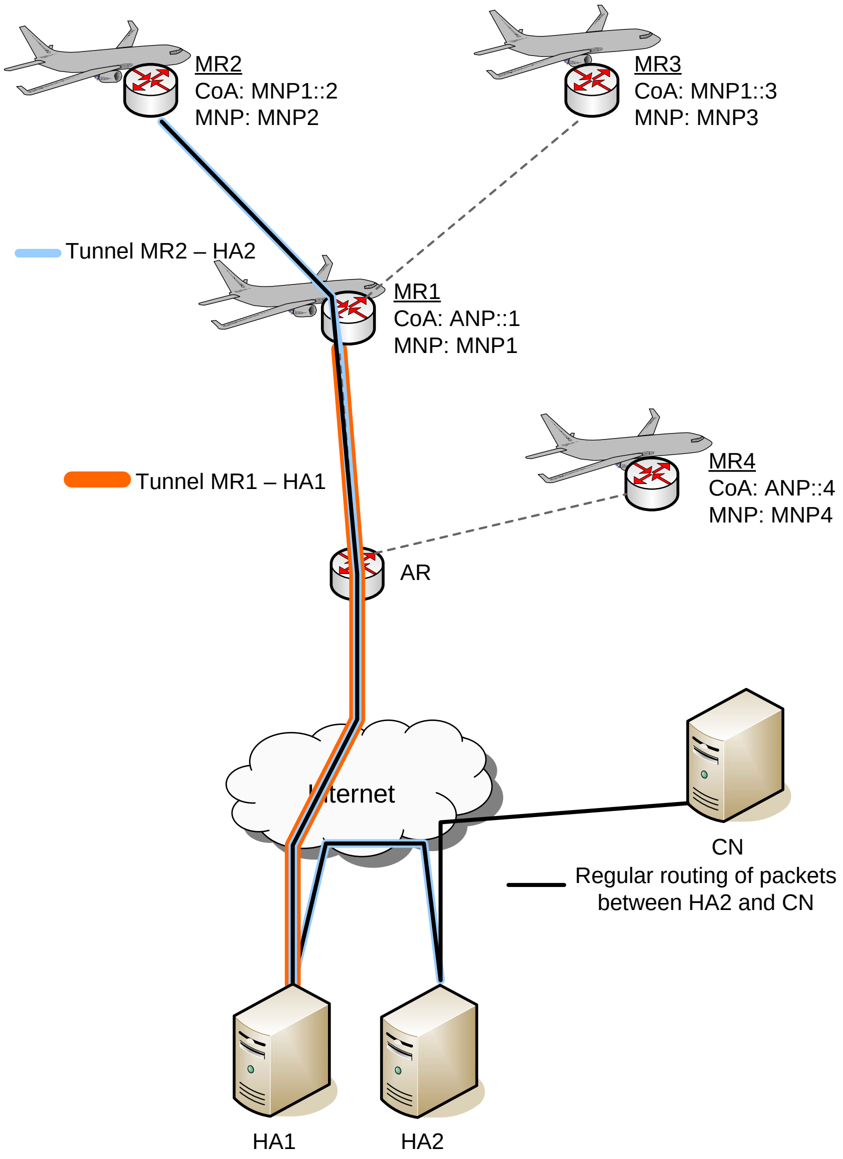


Figure 3.3.: Example network using the Nested NEMO approach.

from the failure must first recognize that they are no longer receiving RAs from the failed router and then perform a handover to a new MR. This leads to an interruption of the Internet connection of all MRs down the tree from the failed router or link. Thus, one weakness of the Nested NEMO is its robustness to link or router failures.

- Protocol Maturity:

  The major strength of the Nested NEMO approach is that it relies only on protocols that are already quite mature. NEMO BS itself has been implemented and tested in vehicular environments. An overview of existing NEMO implementations and related publications is given by the Nautilus6 working group[2].

3.3.2. IP MANET Architecture

In a so-called MANET centric architecture, as has been defined e.g. in [17] for vehicular ad hoc networks, each Mobile Router is still running the NEMO protocol in order to take care of the required mobility signaling as it attaches to different Access Routers. However, the Mobile Routers do not form the hierarchical Nested NEMO structure described in the previous section. Instead, the forwarding of packets between aircraft is now controlled by a MANET routing protocol. This allows for greater flexibility in the choice of a route within the wireless network, since packets are no longer bound to follow the tree structure that is imposed by Nested NEMO. Here, we discuss two variants of MANET centric architectures. In this section, we consider a MANET formed at the IP layer. This combination of a MANET routing protocol and NEMO is sometimes referred to as MANEMO, and has been introduced in [93]. In the next section, a MANET formed below IP will considered. The network topology shown is applicable to both cases.

Functional Overview

As can be seen in Fig. 3.5, all Mobile Routers generate a CoA from the network prefix that is advertised by the Access Router. When a MNN connected to MR2 sends a packet to CN, this packet is encapsulated by MR2 and tunneled directly to HA2. The packet is then routed normally from HA2 to CN. The blue line in Fig. 3.5 denotes the tunnel between MR2 and HA2.

Examples for MANET routing protocols standardized by the IETF's MANET Working Group are DYMO [94] as a reactive protocol and OLSR [95] as a proactive protocol. OLSR is currently being updated to OLSRv2 [96]. The protocol stacks of the network nodes in this approach are shown in Fig. 3.4. With Nested NEMO, MRs configured their CoA from the MNP of the MR that is 'up the tree' towards the root MR. However, such a tree structure does not exist in the MANET centric case, so that a new mechanism for the configuration of IP addresses in an ad hoc network from the prefix advertised by the Access Router is required. In principle, the Router Advertisements generated by the Access Routers will need to be flooded through the wireless network, so that every

[2] http://www.nautilus6.org

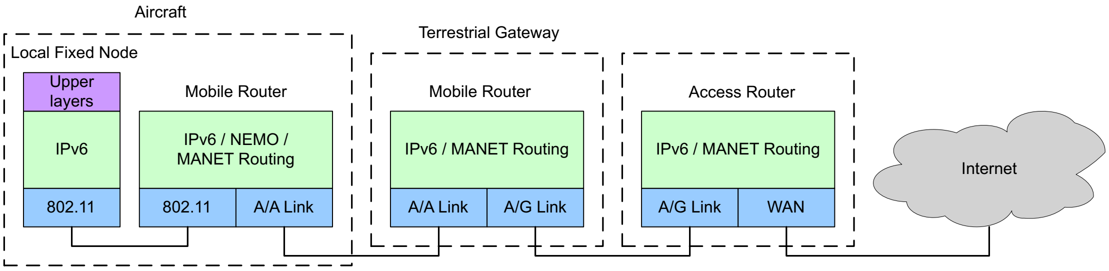


Figure 3.4.: Protocol stack of IP MANET architecture.

MR can configure a valid CoA. This is in contrast to regular IPv6 operations, where the scope of a RA is limited to a single IP hop.

In the case of a multihomed MR, the MR needs to be able to decide which Access Router shall be used for which traffic flows, based e.g. on QoS considerations. Then, it would set the source address of each IP packet accordingly. However, it must then also be guaranteed that a packet is in fact routed via the AR that was intended by the MR. Typically, routing protocols only inspect the destination address of a packet in order to determine the next hop for forwarding. However, ingress filtering at an AR might result in packets being dropped if they are not sent to the AR that was assumed by the MR. Also, it is desirable that the same AR is used in upstream and downstream directions due to the detrimental effects of asymmetric routing on the performance of TCP. The problem of routing an upstream packet towards the correct AR in the MANET can be solved by introducing policy based routing at the Mobile Routers. Each MR may maintain separate routing tables for each AR. The correct next hop for a packet is determined by first selecting the correct routing table based on the source address of the packet and then looking up the destination in this table. Since the number of ARs in the network is limited, the number of routing tables that a MR will need to maintain will remain manageable. For downstream traffic, only a single routing table is needed. Other means of routing a packet through the desired AR include source routing or the tunneling of packets from the MR to the AR by IP encapsulation. However, this would introduce additional overhead, due to the additional header in case of encapsulation, or the source routing option in the header.

Evaluation of the IP MANET Architecture

- Route Optimality:

  In contrast to the Nested NEMO approach, the operation of a routing protocol in the wireless network allows packets to be routed along the optimum route in the MANET. Since only a single HA is involved in each packet transmission, the path in the ground network is also much shorter than with Nested NEMO.

- Data Overhead:

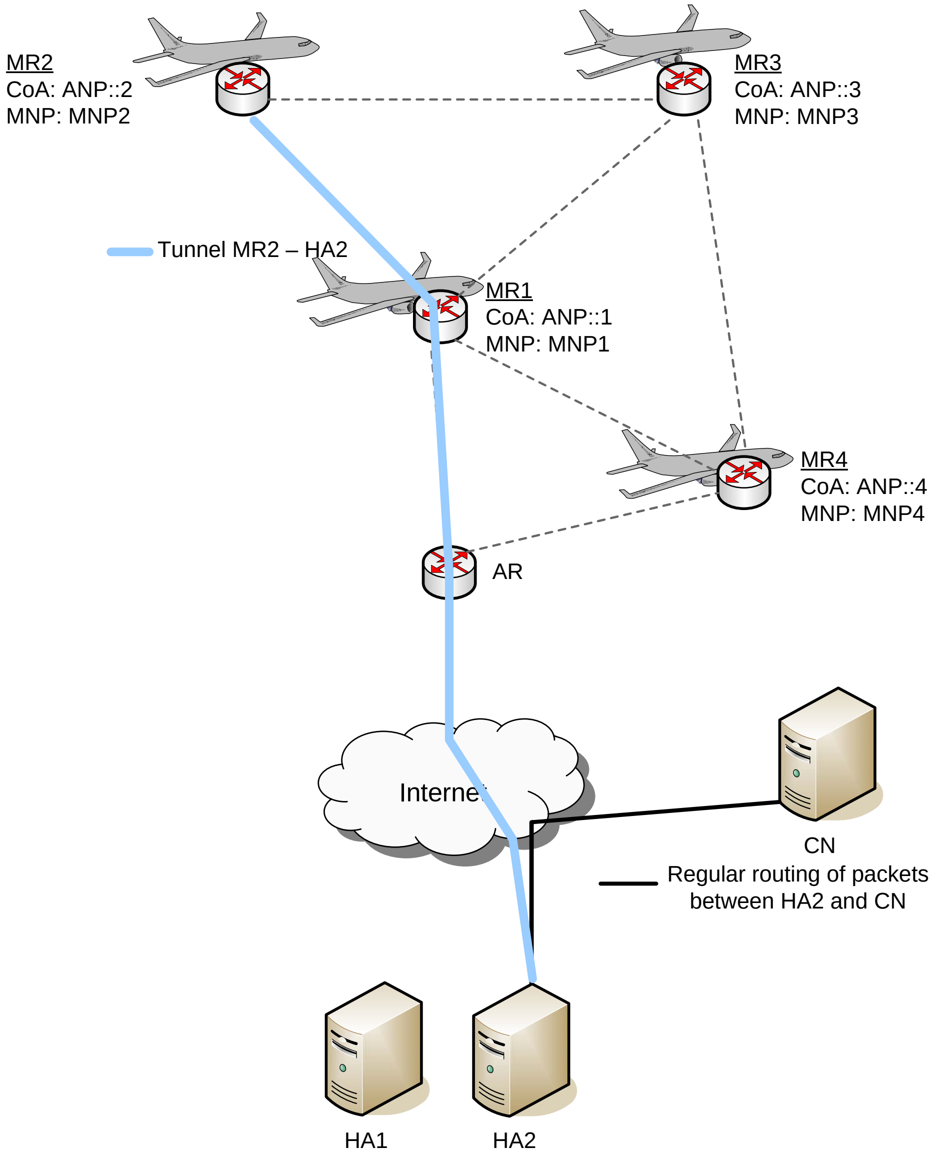


Figure 3.5.: Example network using MANET routing and address autoconfiguration.

Since only a single HA is involved in each packet transmission, only a single level of encapsulation is required as well. This can significantly reduce the amount of overhead that is encountered with each data transmission, compared to the Nested NEMO solution.

- Signaling Overhead:

  Of course, running a routing protocol in the MANET leads to a constant overhead of signaling traffic. Given the rather static nature of our network, though, this overhead can likely be kept to a minimum.

- Robustness:

  Another advantage of the MANET centric approach is its improved robustness. If any router or link fails, traffic can quickly be directed around this failure by the ad hoc network routing protocol. No MR in the network will need to configure a new CoA because another MR has failed. With appropriate routing metrics, traffic can also be distributed such that congestion in the ad hoc network is avoided.

- Protocol Maturity:

  The maturity of the IP MANET approach is not as high as the Nested NEMO approach, since less practical experience with the combination of MANETs with NEMO has been collected so far. However, the MANET routing protocols themselves are either standardized, such as OLSR, or are on the standards track, as DYMO is. The large community working with these protocols ensures that they are well tested. Existing real-life implementations of OLSR include OOLSR at INRIA [97] and the implementation b the Naval Research Laboratory [98]. Both of these implementations are available for download in the Internet.

3.3.3. Sub-IP MANET Architecture

A possible approach to overcoming the issues of IP address autoconfiguration in an ad hoc network is to hide the entire ad hoc network below the IP layer by introducing an additional network layer between IP and the link layer, as shown in Fig. 3.6. An example application of this network architecture can be found in the Car 2 Car Communications Consortium, where the so-called C2C-CC NET layer is inserted below the IP layer [99]. In our case, this custom network layer will be referred to as the Aeronautical Ad Hoc Network Layer (AAHNet).

Functional Overview

In this Sub-IP MANET architecture, IP packets are forwarded through the ad hoc network transparently by the AAHNet layer. Thus, every MR is only a single IP hop away from the AR, and NEMO BS is able to operate practically unaltered on every aircraft. The process by which the MRs perform address configuration is essentially the same as on a wired link. The AAHNet layer must ensure that Router Advertisements

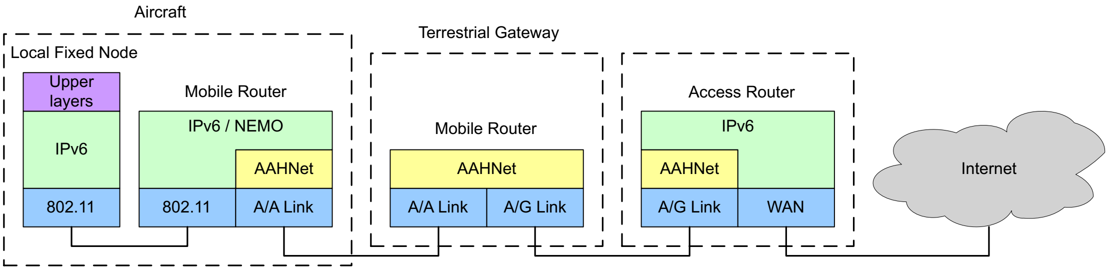


Figure 3.6.: Protocol stack of Sub-IP MANET architecture.

are distributed through the ad hoc network. Optimal routes may be used for packet forwarding within the MANET, and packets only need to pass through a single HA. However, compared to the MANET-centric IP solution, the additional protocol layer also leads to a certain additional amount of overhead that is incurred with every packet transmission in form of the AAHNet packet header. Whenever an MR or AR intends to transmit an IP packet over the ad hoc network, this packet must be encapsulated in an AAHNet packet, and the node must translate the destination IP address into a destination AAHNet address. MRs can learn the AAHNet address of the AR from the RAs that they receive. Similarly, the AR can create a table mapping IP to AAHNet addresses by listening to the traffic that it forwards: After configuring a CoA from an AR, the first packet that a MR will send via this AR is a Binding Update to register the new CoA with its HA. When forwarding this packet into the Internet, the AR can extract the AAHNet address from the AAHNet packet and the IP address from the encapsulated IP packet in order to create a mapping between IP address and AAHNet address. The AR is thus able to forward packets from the Internet to the MR as well, as long as the MANET routing protocol provides a next hop AAHNet address for the AAHNet destination. Since MRs must periodically refresh their bindings at the HA, this mapping can time out automatically after the maximum allowed time between two BUs. The advantages of a MANET centric approach regarding resiliency against router or link failures that were mentioned in the previous section apply equally to a sub-IP MANET. The problem of guaranteeing that a packet is routed via the correct AR, as described in the previous section, does not apply to this case, since the MR encapsulates an IP packet inside an AAHNet packet, which carries the address of the AR as its destination address. As stated above, such a sub-IP approach has been adopted for vehicular communications by the Car-to-Car Communications Consortium. However, the main driver behind vehicular communications is the introduction of traffic safety applications. Many of these applications do not require multi-hop forwarding of packets, others require flooding in a certain geographic region. These constraints, in addition to the more dynamic network topology than in our case, have led the C2C Consortium to adopt geographic forwarding below IP within the MANET. It is argued in [99] that this functionality cannot be provided if routing is based on IP addresses. In contrast to traffic safety applications, passenger infotainment applications in the C2C architecture are not required to make use of the C2C-CC NET layer.

Evaluation of Sub-IP MANET Architecture

- Route Optimality:

  Similar to the IP MANET architecture, the routing protocol in the ad hoc network assures that optimal routes are used for the forwarding of packets within the wireless network. Similarly, each data packet only traversed one HA. Therefore, routes in the ground network are shorter than those of the Nested NEMO solution.

- Data Overhead:

  As in the IP MANET architecture, only one HA is traversed by each data packet, and thus only one level of IP encapsulation is required. However, the sub-IP routing protocol will add its own header to each data packet. Since this header will likely be more compact than a full IPv6 header, the data overhead is similar to the data overhead of the IP MANET solution.

- Signaling Overhead:

  Similar to the IP MANET case, this solution requires the periodic dissemination of routing information through the ad hoc network. Therefore, the signaling overhead is also expected to be similar to the overhead of the IP MANET solution, and higher than the Nested NEMO solution.

- Robustness:

  The robustness of the Sub-IP MANET solution is similar to the IP MANET solution.

- Protocol Maturity:

  In ad hoc networking research, an abundance of routing protocols have been proposed, although these are typically not as mature or as thoroughly tested as those developed by the IETF MANET Working Group for use in IP MANETs.

3.3.4. Protocol Translation

In [100] and [101], Rohrer *et al.* and Çetinkaya *et al.* propose a network architecture for airborne telemetry networks that relies on the concept of protocol translation. Custom network layer and transport layer protocols (denoted AeroNP and AeroTP, respectively) are defined to operate within the AAHN. The gateway connecting the AAHN to the ground network takes on the additional role of translating between the regular TCP/IP stack used in the ground network and the AeroTP/AeroNP stack used in the wireless network. In our case, the same conversion would also need to be performed by the MR in order to allow end users to use the regular TCP/IP stack installed on their systems. The protocol stacks of the nodes in this approach are shown in Fig. 3.7. This architecture

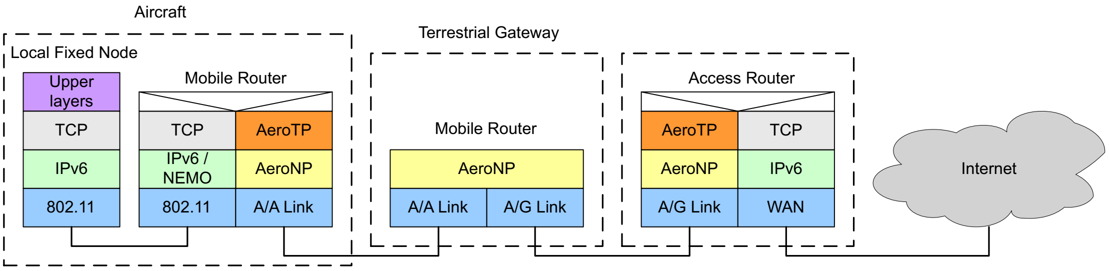


Figure 3.7.: Protocol stack of Protocol Translation architecture.

allows the protocols used inside the AAHN to be perfectly adapted to the characteristics of the wireless network.

However, the problems of address configuration and mobility also need to be solved again, since the solutions existing for IPv6 cannot be used directly. The gateways that perform the task of protocol translation will need to maintain a certain amount of state for each traffic flow, thereby complicating the handover of a flow to a different gateway.

If a user is applying encryption at the IP layer, e.g. using IPSec for a VPN connection, the translation gateway will not be able to read the TCP header and thus will not be able to perform the TCP/AeroTP translation. This violates the requirement of transparency to the end user. Therefore, we do not consider the concept of protocol translation to be applicable to our scenario.

3.4. Discussion of Candidate Architectures

Table 3.1 summarizes the behavior of the candidate protocol architectures that were discussed in the previous sections with respect to the five evaluation criteria specified in Section 3.2.2. The protocol translation architecture is not shown in this table, since it does not fulfill the original requirements.

Criterion	Nested NEMO	IP MANET	Sub-IP MANET
Route optimality	-	+	+
Signaling overhead	+	o	o
Data overhead	-	+	+
Protocol maturity	+	+/o	o
Robustness	-	+	+

Table 3.1.: Summary of evaluation criteria for candidate architectures.

In general, the advantages and disadvantages of the MANET-centric approaches complement those of the Nested NEMO approach. The advantages of Nested NEMO lie in the low signaling overhead and its high protocol maturity. On the other hand, the

MANET centric approaches are able to provide much more efficient routes, both in the wireless part and in the ground networks. They require less IP in IP encapsulations for user data and are much more robust against link failures in the ad hoc network. Overall, the Nested NEMO solution does not appear attractive when compared to the MANET approaches. The differences between the IP MANET and Sub-IP MANET architectures are relatively small. IP MANETs can be considered slightly more mature due to the working code developed within the IETF MANET Working Group. However, a Sub-IP MANET simplifies IPv6 address autoconfiguration, because the Mobile Router has the impression that it is on the same link as the Access Router. Also, the Sub-IP approach easily handles the problem of ensuring that packets inside the ad hoc network are routed towards the correct Internet gateway by routing them according to the 24 bit ICAO identifiers. This is more efficient than possible solutions at the IP layer, which would be based on the 128 bit IPv6 addresses of the MRs.

Based on this discussion, we will assume that the ad hoc network routing functionality is located at Layer 2, below IP. However, the gateway selection must be performed at the IP layer, in order to set the source address of upstream packets correctly, so that they are not dropped due to ingress filtering in the Access Network. A functional routing architecture to enable this concept is discussed in the following section.

3.5. Proposed Routing Architecture

Based on the discussion of the candidate architecture analysis in the previous section, we now propose a functional routing architecture for the Mobile Routers of the AAHN that performs gateway selection at the IP layer but performs the routing inside the AAHN below the IP layer. A schematic diagram of the proposed architecture is shown in Fig. 3.8.

The goal of the routing architecture is to enable the appropriate Internet gateway to be selected for each flow, based on its Quality of Service requirements. In principle, any of the gateways in the network can be selected for a flow by a Mobile Router. This decision is left to the MR based on the gateway advertisements that it has received and the information that is provided by the ad hoc routing algorithm.

We assume that a set of Classes of Service have been defined, each with associated QoS targets. A Decision Engine is responsible for the allocation of each of these CoS to an Internet gateway. To achieve this, the Decision Engine relies on path quality information that is provided by the Routing Daemon. The Routing Daemon runs a link state routing protocol, which continuously calculates the best paths to all known Internet gateways and reports the metric of each path to a gateway to the Decision Engine. In addition to the path costs, the Decision Engine can also take into account a cost metric of the Internet gateways themselves and combine this metric with the metric of the path to the gateway.

Based on these metrics and the QoS targets that have been defined for the different service classes, the Decision Engine decides which CoS shall be allocated to which gateway. The Decision Engine then also populates the IP forwarding tables. Multiple routing tables are required, one for each source address that is used by the MR. However, the next hop IP address in the upstream direction is always the address of the corresponding Access Router. In addition to reporting the path costs to the Decision Engine, the Routing Daemon also performs its core task, i.e. populating the Layer 2 routing tables with the next hop information to reach a given destination in the AAHN. In a situation such as this, where all nodes in the network constantly require knowledge of the best paths to a large number of other nodes, i.e. the gateways, proactive routing protocols require less signaling than reactive ones.

To explain the functionality of the routing architecture, we will examine the processing of an IP packet that is generated by one of the passengers' devices. Whenever an IP packet arrives from one of the Mobile Network Nodes via the MR's Ethernet interface to be sent towards the Internet, it must first be classified, i.e. assigned to one of the CoS. Next, when this packet is encapsulated inside another IP packet for the MR-HA tunnel, the IP source address of the outer packet must be set correctly. That is, it must be set to the CoA that the MR has received from the AN that this packet will traverse. The destination IP address of this outer IP packet is the address of the Home Agent. This information – the source and destination IP addresses of the outer packet – are queried from the Decision Engine, providing the packet's CoS as a parameter.

The IP Forwarding block then looks up the next IP hop toward which the packet is forwarded as well as the corresponding interface of the MR from the system's IP forwarding table. Here, several IP routing tables must exist, one for each possible source address of a packet. Although different packets may be bound for the same CN, they may belong to different classes of service, and therefore have different source IP addresses. Thus, they may need to be routed through different access networks and therefore need to be forwarded differently. Since the routing in the ad hoc network is done at Layer 2, the next IP hop of an upstream packet will be the Access Router, regardless of the number of intermediate hops in the AAHN. The packet can also be sent directly over the MR's satellite interface. In this case, MR itself is the Internet gateway, and the next IP hop is the Access Router within the satellite ground network.

If the packet is handed down to the AAHN layer, an AAHN header must be added to the IP packet. Finally, the packet is forwarded over the AAHN interface to the next hop according to the sub-IP routing table. For the AAHNet forwarding, the next IP hop that was inserted previously by the IP forwarding is taken as the AAHNet packet's destination address. Packets being relayed in the AAHN at Layer 2 are only processed by the L2 Packet Forwarding block of Fig. 3.8.

The processing of packets in the downstream direction is simpler than the upstream case. The L2 and IP Packet Forwarding blocks must determine that the packet's destination address is the address of the Mobile Router, decapsulate the packet (first removing the L2 header, then the IP header of the HA-MR IP tunnel), and forward the inner IP packet onto the aircraft's onboard network.

The description in this chapter was a purely functional one. The details of the proposed algorithm that shall be used for the gateway selection in the Decision Engine and for the routing of data through the AAHN by the Routing Daemon will be given in the

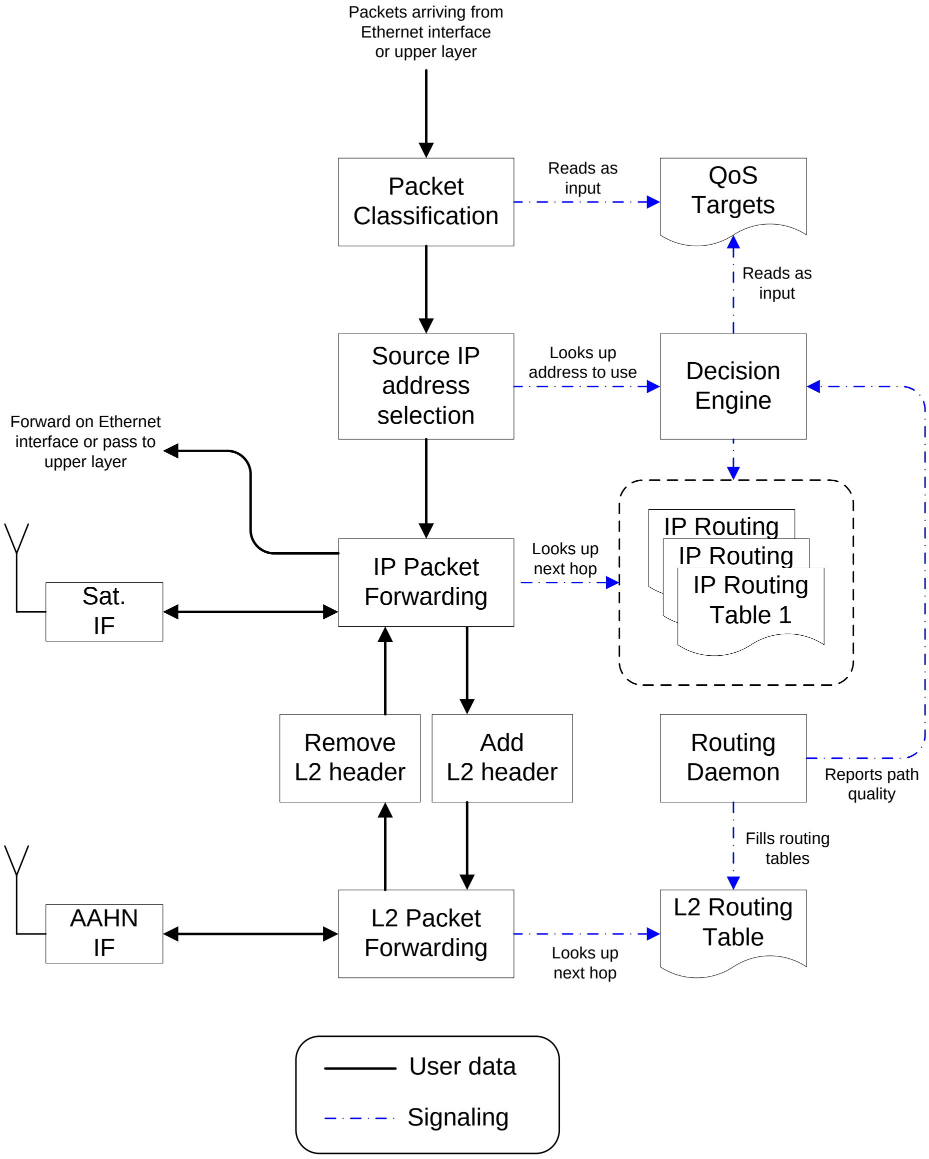


Figure 3.8.: Functional routing architecture of the Mobile Router.

following chapter.

4. Topology Analysis

In this chapter, we analyze the topological characteristics of the envisaged aeronautical ad hoc network. This includes properties of the wireless air to air channel between two aircraft, as well as properties of the network topology that is formed by the cloud of aircraft in the North Atlantic. The characterization of the network topology is based on simulations that we have performed using actual flight schedules.

4.1. Air to Air Channel

This section discusses the characteristics of the wireless communications channel between two aircraft or between an aircraft and a ground station. The aeronautical radio channel in all phases of flight has been thoroughly described in [102]. Here, we focus on those aspects that are relevant for aeronautical ad hoc networks, while the aircraft are in the en route flight phase. The information in this chapter is mostly summarized from [102] and the monograph on wireless channels by Parsons [103].

Since only aircraft during the en route flight phase in oceanic or remote areas are considered here, it can safely be assumed that there are no obstacles between two stations, i.e. we assume that a line of sight path always exists. The maximum distance at which line of sight between two nodes is possible is then determined by their elevation above the surface of the Earth as well as the curvature of the Earth's surface, as shown in Figure 4.1. In this case, the maximum distance at which line of sight between aircraft A and aircraft B is possible occurs when the line segment $\overline{AB}$ is tangent to the Earth's surface. This distance is called the radio horizon of an aircraft. Then, the maximum line of sight distance between aircraft A and B is given by

$$\begin{aligned} d &= d_A + d_B \\ &= \sqrt{h_A^2 + 2h_A r_E} + \sqrt{h_B^2 + 2h_B r_E}, \end{aligned} \tag{4.1}$$

where r_E is the Earth's radius and h_A and h_B are the altitudes of the two nodes. In this case, the length of the curved segment along the ground between the two points on the Earth's surface below the aircraft is given by

$$\tilde{d} = r_E \left(\cos^{-1} \left(\frac{r_\mathrm{E}}{r_\mathrm{E} + h_A} \right) + \cos^{-1} \left(\frac{r_\mathrm{E}}{r_\mathrm{E} + h_B} \right) \right). \tag{4.2}$$

However, this simple geometric model ignores the effect of the atmosphere on the propagation of electromagnetic waves. The refractive index of the atmosphere decreases

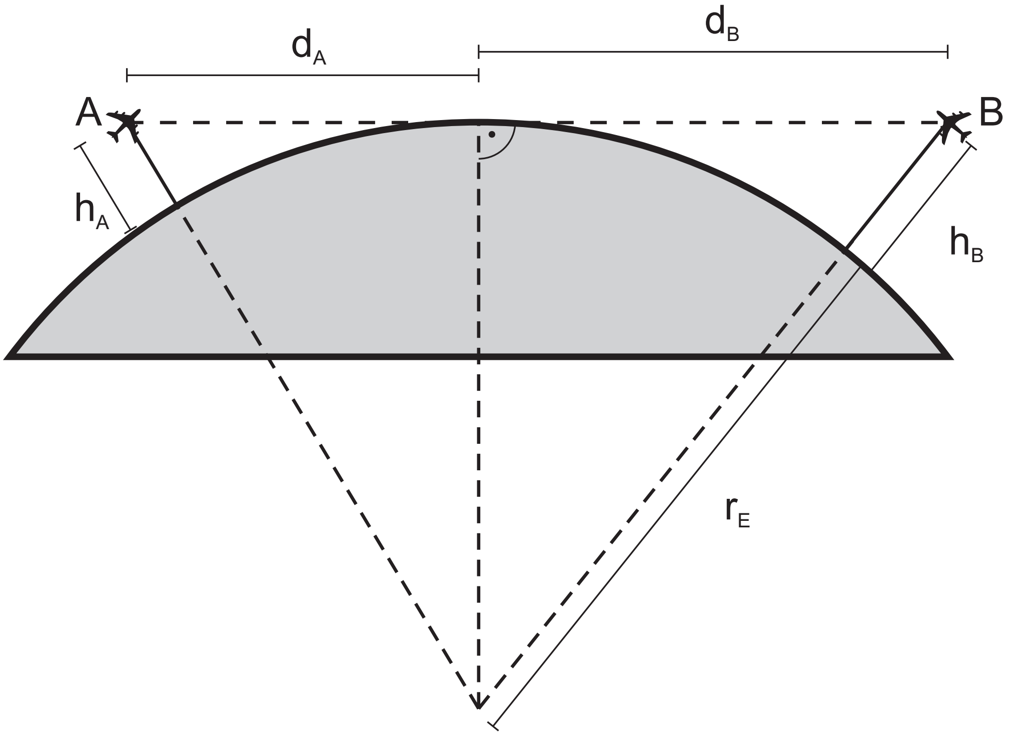


Figure 4.1.: Maximum line of sight distance between aircraft A and B.

exponentially with the altitude due to the decreasing density of the atmosphere [103]. Regardless of their frequency, electromagnetic waves are bent towards the ground, thereby enabling line of sight communications at a larger distance than the distance given by Eq. 4.1, as shown in Fig. 4.2. This effect is commonly dealt with by scaling the Earth's radius appropriately, such that Eq. 4.1 is used with an equivalent radius $r'_{\mathrm{E}} = \frac{4}{3} r_{\mathrm{E}}$ [103].

As noted in [103] the refractive index in reality does not always strictly decrease with the altitude. Small fluctuations of the refractive index may lead to another effect called ducting, whereby a radio wave is "trapped" in a duct between two layers with a higher refractive index, in which it can be reflected back and forth, much like in a waveguide. This allows waves to propagate much further than normal. However, this phenomenon only occurs relatively close to the ground, up to heights of ca. 1,500 m. Thus, ducting has no impact on air to air communications between aircraft at cruise altitude.

At distances up to the radio horizon d_{horizon}, which depends on the altitude of the transmitter and receiver, free space propagation of the electromagnetic signal can be assumed. This has been confirmed by measurements of the air to air channel between Unmanned Aerial Vehicles (UAVs) by Allred *et al.* [104], who determined a path loss coefficient very close to two, even for relatively low flying UAVs. Measurements between aircraft performed by Walter *et al.* [105] came to similar results for small aircraft flying

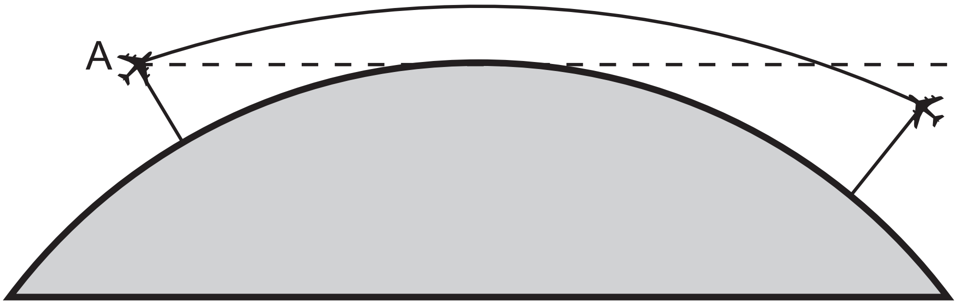


Figure 4.2.: Maximum line of sight distance between aircraft A and B, showing effect of atmospheric refraction.

at altitudes between 600 m and 2,600 m. At distances larger than d_{horizon}, the received signal strength decays much more rapidly due to the lack of a line of sight path as well as effects caused by ground reflections.

The aeronautical channel model recommended by the International Telecommunication Union (ITU) [106] makes use of the scaling factor of $\frac{4}{3}$ for the Earth's radius, shows good agreement with free space loss within the radio horizon, and exhibits much faster degradation of signal power beyond the radio horizon.

For $h_A = h_B = 10$ km, which is a typical cruise altitude on long distance flights, and an Earth radius of 6,371 km, scaled by a factor $\frac{4}{3}$ to account for refraction, Eq. 4.1 provides us a maximum line of sight distance of 824.6 km (and a distance $\tilde{d}$ along the ground of 824.0 km according to Eq. 4.2). As we can see in Fig. 2.1, the cloud of aircraft can span across the entire Atlantic ocean, which is approx. 3000 km wide at this point. With a such a high line of sight range in any direction, an aircraft in the middle of the ocean would be able to cover more than half of this distance with its transmissions. Thus, most of the aircraft in the North Atlantic region do indeed share a common physical radio channel.

In summary, a very good communications channel between aircraft flying at cruise altitude exists up to the distance d_{horizon}, whereas interference from concurrent transmissions farther away is inherently limited by the curvature of the Earth's surface.

4.2. Network Topology

For the characterization of the network topology that can be expected in an AAHN over the North Atlantic, we will use the current air traffic patterns and number of aircraft as a baseline. The flight routes that are in use on a given day can be retrieved e.g. from [1], and the number of aircraft is provided by a flight database from Innovata LLC [28].

The expected increase of the amount of air traffic in the future is not considered here.

According to estimates by STATFOR, the Statistical Forecast Group of EUROCONTROL, the European air safety organization, air traffic worldwide is expected to double between the years 2008 and 2030 [107]. In order to handle this increase, it is likely that aircraft will fly more flexible routes that are less restrictive than the currently used airways, and an increased number of flights will take place in the early morning or late evening to prevent airports from becoming overloaded. Since the analysis in this work is based on current flight data, it presents a conservative estimate of the potential network size and node density within the network. To obtain an ad hoc network of this size in the future, it would be sufficient to equip only a fraction of the aircraft with ad hoc networking capability.

To provide Internet access to the aircraft, we assume that several ground stations are deployed along the coast of the North Atlantic. Via access routers attached to these ground stations, aircraft can connect to the Internet. One possible deployment scenario is shown in Fig. 4.3: Two ground stations are located in northeastern Canada, two in Ireland and Scotland, and one each in Greenland and Iceland. These locations have been selected by us as plausible sites for such ground stations, and are all co-located with existing air traffic control centers or airports.

The circles drawn around the stations have a radius of 200 nmi (370.4 km), and are intended to visualize their potential maximum communication range.[1] As can be seen, the possibilities for the placement of further ground stations are very limited. Additional stations might increase the total capacity of the network, but not its connectivity, since flights across the North Atlantic follow certain routes, as described in the following section.

4.2.1. North Atlantic Tracks

The airspace in the North Atlantic is also referred to as the North Atlantic Corridor (NAC). When flying in the NAC, aircraft typically travel along the so-called North Atlantic Tracks (NATs). The flight procedures that are in use on these tracks are defined in [10]. These tracks are flight routes that are calculated each day, taking into consideration the current weather situation over the ocean, especially the position of the jet stream. The eastern half of the North Atlantic airspace is controlled by Shanwick Air Traffic Control (ATC) in Great Britain and Ireland, and the western half is controlled by Gander ATC in Newfoundland, Canada. The controllers in these ATC centers assign aircraft heading for the Atlantic to one of the current tracks, depending on the aircraft's destination, aircraft type, etc. To reduce the risk of collisions, aircraft are assigned a velocity that must be maintained on the track, and are subject to a number of vertical and lateral separation rules. Each track is separated by one degree of latitude (i.e. 60 nmi) from the next track to the South and to the North. Aircraft may fly at different altitudes on the same track. Typically, six to nine flight levels are used, all separated by 1,000 ft vertically. In addition, aircraft on the same track and altitude are separated

[1] For a ground station located at sea level, and an aircraft flying at 10 km, the radio horizon is slightly larger, at 223 nmi, or 412 km, according to Eq. 4.1.

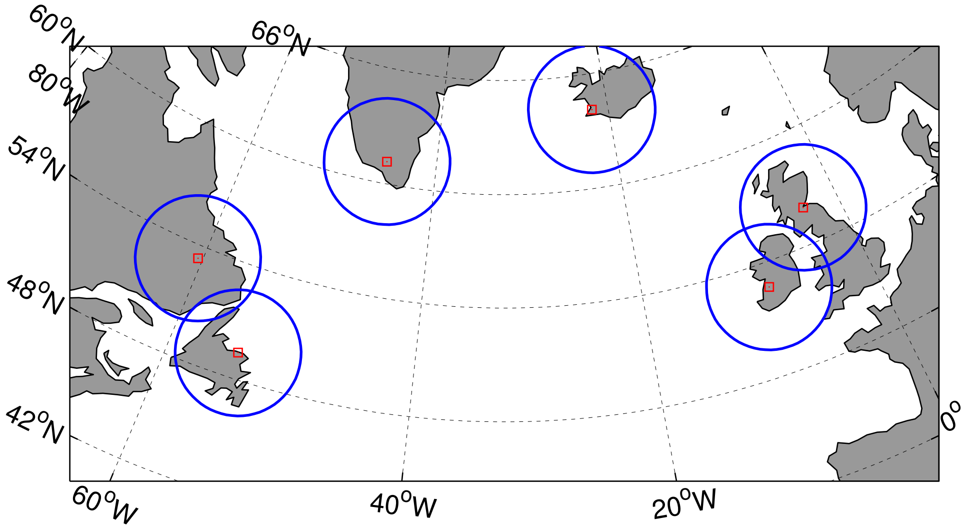


Figure 4.3.: Potential positions of ground stations in North Atlantic region.

by a gap of ten minutes. In the future, more precise navigation techniques, e.g. due to satellite navigation, may allow a further reduction of these separation constraints, increasing the capacity of the North Atlantic Tracks. The tracks can be considered as "one way streets". Only eastbound or westbound tracks are active at any point in time. This results in one wave of eastbound aircraft and one wave of westbound aircraft every 24 hours. Flights typically depart from Europe around late morning or noon, arriving in North America in the afternoon (local time). In the other direction, flights leave North America in the evening, arriving in Europe the next morning. Eastbound flights profit from the tailwind provided by the jet stream, leading to a shorter flight duration. The jet stream typically flows further to the South than the Great Circle route between Europe and North America, so that Eastbound tracks are also further to the South than Northbound tracks. This can be seen in Fig. 4.4, which shows the actual North Atlantic Tracks that were used on August 5th, 2009, according to [1]. The eastbound tracks are shown in blue, westbound tracks in green. The yellow triangles are navigational waypoints.

In the future, as navigation becomes more exact and the air traffic volume increases, the strict separation rules on the North Atlantic Tracks may be loosened in order to accommodate more flights in the North Atlantic region [10]. This may give pilots more freedom in their choice of routes. However, the NATs are currently already calculated with the goal of reducing the total fuel consumption during the flight. Since this will likely remain an important goal in the future, we expect that aircraft will most likely continue to fly along similar routes as they do now, but the total number of flights is expected to increase.

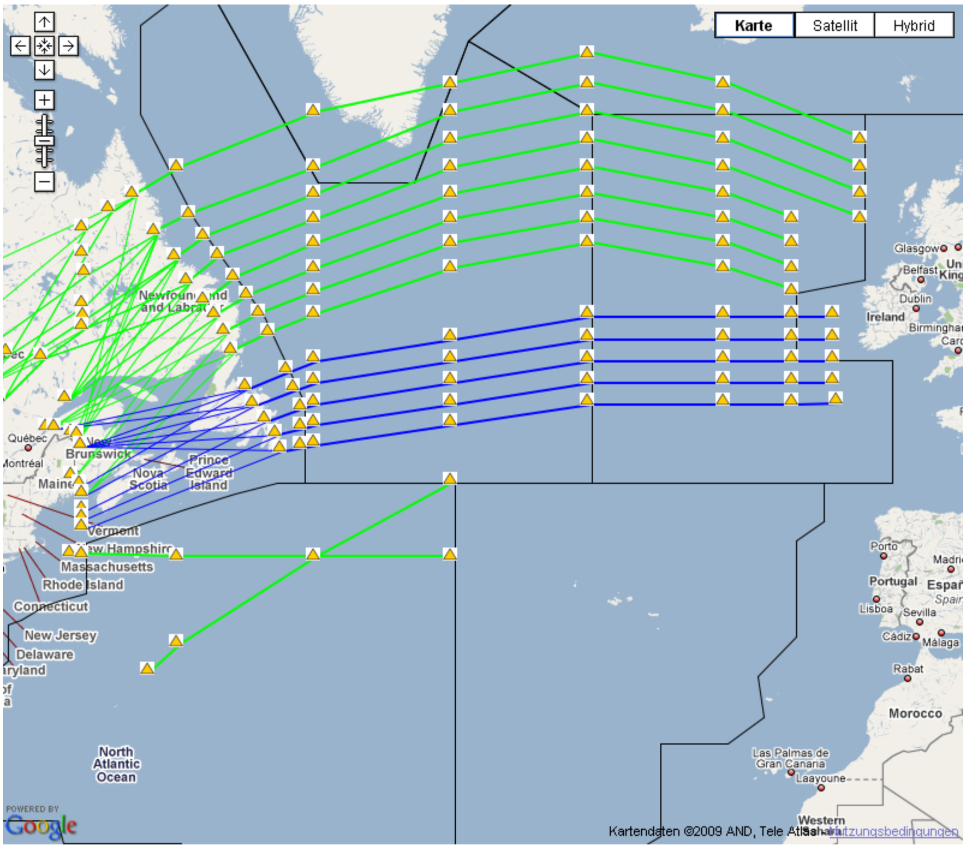


Figure 4.4.: North Atlantic Tracks on Aug. 5th, 2009. The figure is a screenshot taken from [1], visited on Aug. 5th, 2009.

4.2.2. Gateway Connectivity

The use of the North Atlantic Tracks leads to a very well defined, stable "cloud" of aircraft over the North Atlantic, as shown in Fig. 2.1. In previous work, we have investigated the topology of a potential aeronautical ad hoc network over the North Atlantic in [9], based on a database of scheduled flights for May 21-22, 2007. These days were chosen as "typical" days that do not have a particularly high or low number of flights. The flight database is a product of the company Innovata LLC [28] and is produced in cooperation with the International Air Transport Association (IATA), an association of commercial airlines worldwide. The analysis in [9] was performed without considering the North Atlantic Tracks. Instead, 'Great Circle' routes were assumed between the airports at the endpoints of each flight. Great Circle routes are defined as the shortest connection between two points on the surface of a sphere. For the topology analysis in this section and the following one, we have repeated simulations similar to those in [9], based on the same flight data, but here we do make use of the NATs according to [1], as opposed to the Great Circle routes of our previous work. Therefore,

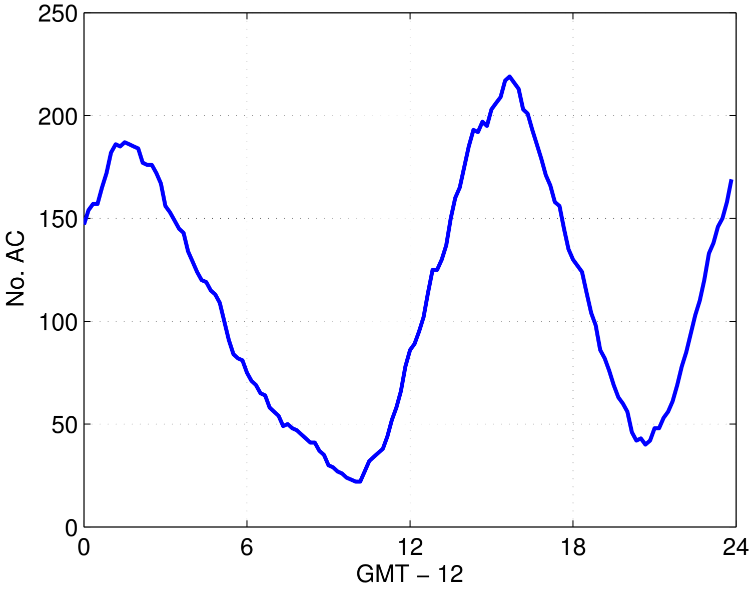


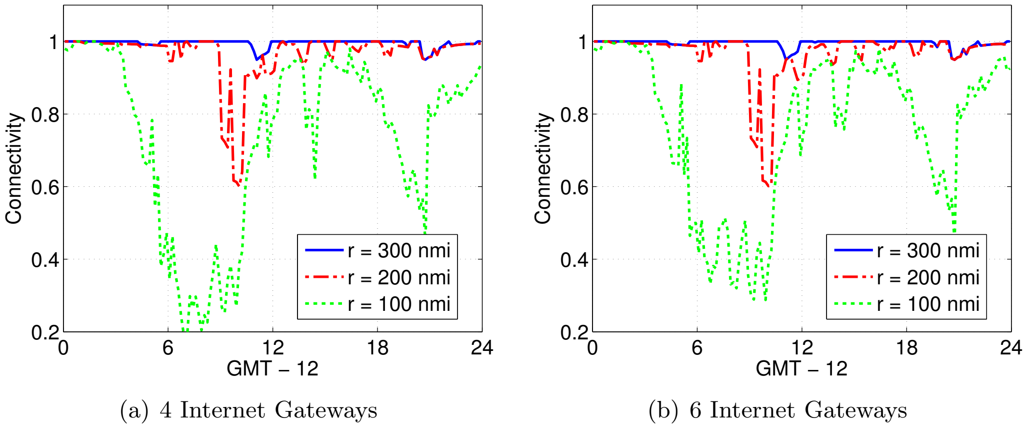
Figure 4.5.: Number of aircraft in North Atlantic Corridor within 24 hour period, based on our analysis of the INNOVATA flight database.

the analysis presented here is more realistic than our previous analysis in [9]. This analysis of the network topology is necessary in order to answer the question, whether such an ad hoc network is feasible, or if the cloud of network is too sparse to allow reliable connections.

Between Ireland and the coast of Newfoundland, the Atlantic Ocean is approximately 3000 km wide. Currently, the number of aircraft flying in this region at any point in time varies between approx. 25 and 225. The number of aircraft in the North Atlantic Corridor, defined as the region between 45° N and 90° N and 10° W and 60° W during a 24 hour period, is shown in Fig. 4.5. The first peak corresponds to the wave of aircraft flying from North America to Europe, whereas the second peak corresponds to the aircraft flying in the other direction (note the shift by 12 hours with respect to Greenwich Mean Time (GMT) on the x-axis). At its largest extent, this cloud of aircraft spans across the entire NAC, touching both the European as well as the North American coastline. Thus, a wave of aircraft traveling from Europe to North America will always make contact with the North American side before contact with the European side is lost. The same is also true for the other direction.

A cloud of aircraft traveling from Europe to North America is typically connected to the European coast only for approx. 4.5–5 hours. Then, it makes contact with the North American coastline, and is connected to both sides for about 5–7 hours. After the last aircraft has left the European coast, the cloud is only connected to the North American coast for another 4.5–5 hours.

Due to the jet streams, which are strong winds flowing from West to East in the



(a) 4 Internet Gateways

(b) 6 Internet Gateways

Figure 4.6.: Degree of gateway connectivity over 24 hour period for 100, 200, and 300 nmi transmit range.

Northern hemisphere, flights from North America to Europe take less time. When the first aircraft of an eastbound cloud leaves the North American coast, it takes approx. 3.5–4 hours to reach the European coast near Ireland. For approx. 4 hours, the cloud has contact with both sides of the Atlantic. Finally, it is only connected to the European side for 3.5–4 hours again.

In general, these numbers depend on the weather and may change as the NATs are shifted further to the North or South. The duration that the cloud is only connected on one side of the ocean depends only on the speed of the aircraft at the front of the cloud, whereas the duration that the cloud is connected on both sides depends on the size of the cloud, i.e. the amount of air traffic as well as the speed of the aircraft on both ends.

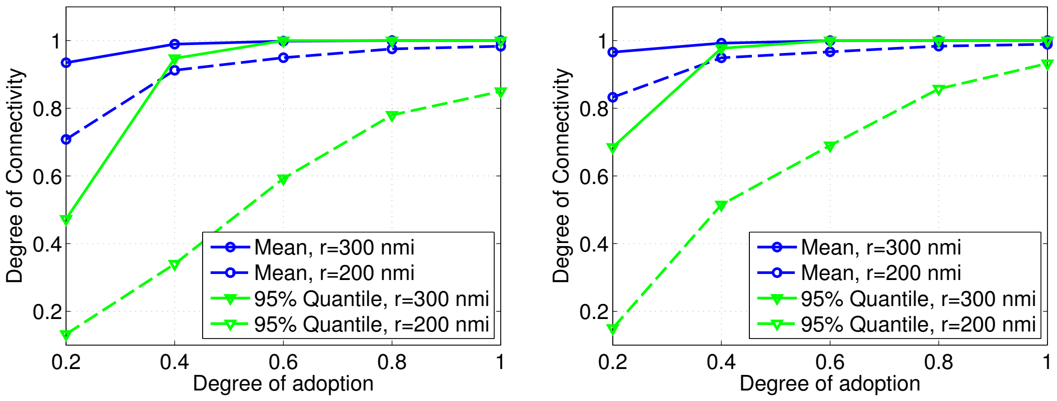
The most important aspect of these considerations is that the cloud of aircraft comprising the ad hoc network can always connect to a terrestrial gateway on at least one side of the Atlantic, and can often connect to terrestrial gateways on both sides. Using the database of scheduled flights, we have performed simulations with two different ground station deployment scenarios: one scenario assuming the deployment of six stations as shown in Fig. 4.3, and one scenario as in Fig. 4.3, but without the two ground stations in Greenland and Iceland. We have selected these locations for ground stations based on the current locations of ATC centers or large airports or military airfields. For the moment, satellite gateways are not yet considered. The aircraft in the simulation flew along the NATs as shown in Fig. 4.4 and were assigned to these tracks in a round robin manner, ensuring that the separation constraints of [10] were fulfilled.

Now, we take a look at the *gateway connectivity* of the envisaged network. Here, we define the term ’gateway connectivity’ at a certain time as the fraction of all aircraft in the North Atlantic that can reach at least one gateway, either because the aircraft is itself an Internet gateway, or over a multi-hop path through the ad hoc network. Thus, the connectivity will always be a value between zero and one. Since the NATs confine the aircraft to a relatively compact region, the results presented here indicate a slightly higher degree of connectivity than the results of [9]. When aircraft fly on great circle routes, as was assumed in [9], the cloud of aircraft becomes more spread out, and individual aircraft are more likely to become separated from the rest of the ad hoc network.

Fig. 4.6 shows the degree of gateway connectivity for the two different deployment scenarios over a period of 24 hours. The range of the air to air link, here denoted r, is varied between 100, 200, and 300 nmi. In the characterization of the radio channel for air to air communications in Section 4.1, it was shown that a transmission range of more than 800 km (432 nmi) can theoretically be achieved. However, this range will likely be limited by transmit power constraints or the interference that would be generated for other simultaneous transmissions. The range of the air to ground links between ground stations and aircraft is kept constant at 200 nmi.

It can be seen that in this case, the two stations in Greenland and Iceland contribute very little to the gateway connectivity of the network. A slight effect can only be seen for the air to air range of 100 nmi. Here, it is likely that some aircraft in the middle of the ocean have lost contact to the other aircraft in the cloud, which would provide a connection to gateways near Canada or Europe, but are still able to connect to one of the other gateways near Greenland. Even though these two ground stations do not provide much additional connectivity, they may still be of great importance when the distribution of data traffic within the network is considered, as they may prevent the gateways at the edges of the network from becoming overloaded. Aside from the ground station locations considered here, the geography of the North Atlantic region offers no further possibilities for the placement of ground stations on land. The significant drops in connectivity, especially for 100 nmi and 200 nmi range, occur during the times of the day when there are only very few aircraft flying over the North Atlantic and the network becomes too sparse to maintain connectivity. Although the connectivity drops as low as 20% for four ground stations and r=100 nmi, there are only approx. 40 aircraft over the North Atlantic during this time. During the times of good connectivity, there are over 200 aircraft in the North Atlantic, of which the vast majority is connected.

Since it cannot be expected that all aircraft will be equipped to participate in the ad hoc network, and during the initial deployment phase the number of aircraft will only increase gradually, it is interesting to look at how the connectivity of the network depends on the number of aircraft that have been equipped to participate in the ad hoc network. This ”degree of adoption” by the aircraft can be varied as a parameter between zero (i.e. no aircraft flying over the North Atlantic have adopted the ad hoc network) to one (all aircraft have been equipped). Fig. 4.7 shows how the connectivity behaves as a function of this degree of adoption. Here, the connectivity is shown as the fraction of its flight time over the North Atlantic that an aircraft is connected. Both the mean and the 95% quantile of this value are shown for two different values of the maximum transmission range (200 nmi and 300 nmi). In addition to the case of six ground stations located according to Fig. 4.3, these connectivity results are also shown for the case that one tenth of the aircraft have a satellite link and are able to offer Internet connectivity



(a) No satellite gateways.

(b) 0.1 satellite gateway probability.

Figure 4.7.: Fraction of time in NAC which aircraft have gateway connectivity, depending on fraction of aircraft participating in the ad hoc network, for 200, and 300 nmi transmit range.

to other aircraft through this link by acting as satellite gateways.

Fig. 4.7(a) shows the degree of connectivity when there are no satellite gateways. For a degree of adoption of 60%, practically all aircraft can be connected for the entire time that they spend over the ocean when the range is 300 nmi. For 200 nmi though, only 70% of the aircraft are connected for more than 95% of their flight. As shown in Fig. 4.7(b), these rates are improved somewhat when one tenth of all aircraft are able to act as satellite gateways. However, further increasing the number of satellite gateways to more than one in ten only leads to small improvements in the connectivity. This is due to aircraft at the front or rear end of the cloud flying in isolation from other aircraft. These aircraft only become connected if they themselves have a satellite link. Even if additional satellite gateways may not lead to significantly higher gateway connectivity, they do play an important role when data traffic is considered. Whereas terrestrial Internet gateways are only located at the edge of the aircraft cloud, satellite gateways may be located in the middle of the network, thereby potentially alleviating congestion in the network and increasing the possible throughput considerably.

4.2.3. Node Degree

Due to the separation constraints that are imposed on the aircraft in the North Atlantic Track system, as well as the joint movement of all aircraft in the same direction, the network topology is quite stable. If an aircraft has a link to the aircraft flying in front of it and behind it, these links will likely exist for the entire time that the aircraft is flying over the Atlantic. Aircraft on other tracks might be flying at a higher or lower speed, but the relative velocity will be very low in comparison to the transmission range. When one aircraft is passing another aircraft at a different flight level on the same track,

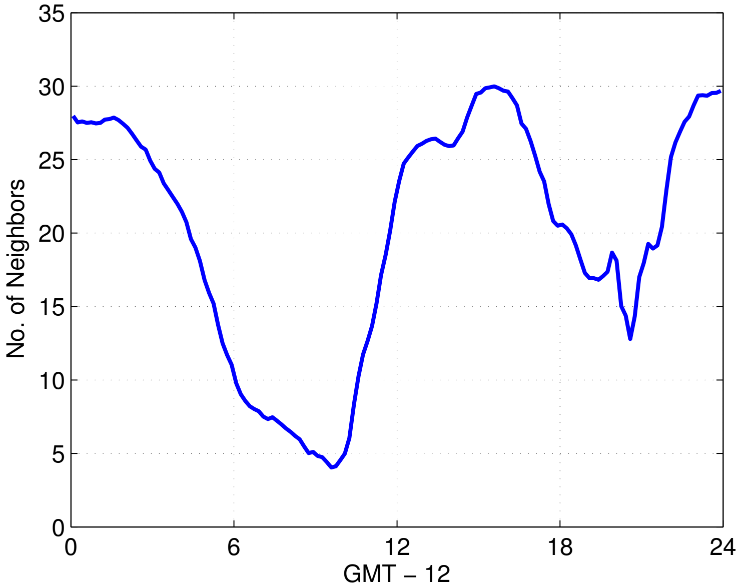


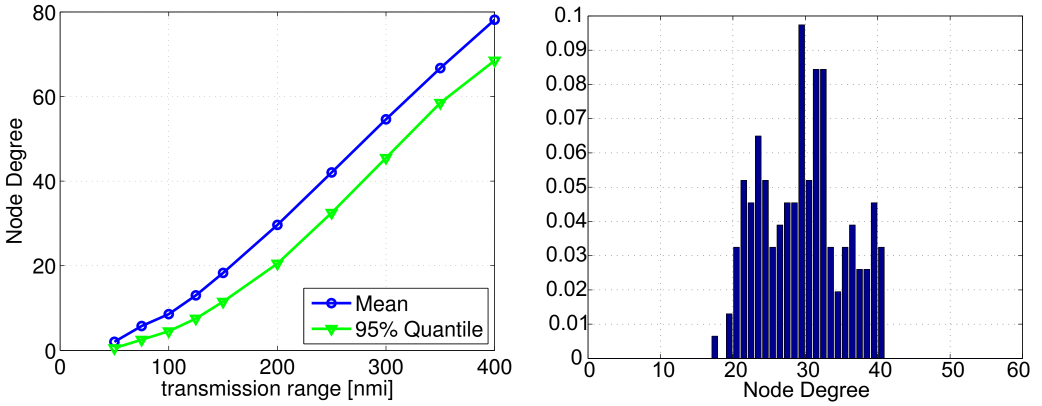
Figure 4.8.: Average node degree of aircraft in NAC within 24 hour period.

assuming a relative velocity of 200 km/h, and a transmission range of 800 km, the link between the two aircraft will still exist for eight hours, which is longer than it takes for them to cross the Atlantic. Since only east- or westbound flights are active at any given time, the situation that two aircraft are flying towards each other, resulting in a very high relative velocity of approx. 1800 km/h, is not likely to occur. Essentially, the network can be considered as a ”quasi-static” network locally. Only the connectivity to the terrestrial Internet gateways is somewhat more dynamic, as the cloud of aircraft flies past the terrestrial Internet gateways. However, the duration of these links is also typically on the order of tens of minutes.

The number of other aircraft that are within the communications range of a given aircraft is referred to as that aircraft's *node degree* [108]. The node degree can be used to quantify the density of the network. We define the *average node degree* at a certain instant in time as the average of the node degrees of all aircraft that are in the North Atlantic Corridor at that time instant. Note that this average is defined as an average over all aircraft and not over time.

Of course, the average node degree also varies over time, since it depends on the number of aircraft over the North Atlantic. This variation of the average node degree of all aircraft in the North Atlantic over a time period of 24 hours is shown in Fig. 4.8 for an air to air transmission range of 200 nmi. At peak times, an average aircraft in the network has about 30 neighboring aircraft. Note that the node degree is roughly proportional to the number of aircraft in the NAC as shown in Fig. 4.5. An average aircraft can reach approx. one sixth of the other aircraft in the network directly.

In addition, the expected node degree depends on the position of the aircraft in the



(a) Average node degree as a function of transmit range.

(b) Average node degree distribution for 200 nmi range.

Figure 4.9.: Average node degree when the cloud of aircraft spans across the entire NAC.

network. Obviously, an aircraft on one of the outer tracks will have less neighbors than an aircraft in the middle of the network. For a snapshot of the topology, when the cloud of aircraft spans across the entire NAC, the average node degree as a function of the transmission range and the distribution of the aircraft's node degrees for a range of 200 nmi are shown in Fig. 4.9. It is important to note that the node degree increases linearly with the transmission range. This is due to the long and narrow rectangular shape of the cloud of aircraft. If the nodes were randomly distributed in an area much larger than the transmission range, the node degree would increase quadratically with the range, corresponding to the quadratic increase of the area that is within the aircraft's transmission range. The relatively sharp edges in the distribution in 4.9(b) are also due to the track system. At this time instant, the node degrees of all aircraft in the North Atlantic are between 18 and 41. Larger variations due not occur, because the aircraft are constrained to the tracks.

The high average node degree (approx. 30 neighbors at a moderate range of 200 nmi) implies that a single aircraft's transmission would generate a significant amount of interference to neighboring nodes. Therefore, measures to reduce the interference are required in order to reach a high utilization of the wireless channel. These measures could include directional antennas or transmit power control, for example.

5. Communication System Model

Following the background information given in the previous chapters, we now define a system model that allows us to address the problem of routing and Internet gateway selection in aeronautical ad hoc networks. This also includes assumptions that must be made for the air to air communications link due to the lack of a currently operational system.

We assume that each aircraft participating in the ad hoc network is equipped with two different air to air communication systems. A high bandwidth, directional link will be used for transmitting user data, whereas a low bandwidth, omnidirectional link will be used for transmitting control traffic. This control link could be the air to air mode of the L-band datalink that is currently under development by Eurocontrol, the European air safety organization [109]. In the following sections, we define our model for the high bandwidth link that is used for transmitting user data. Then, the network model, including aspects of the onboard network and the ground connectivity, is defined.

5.1. The Physical Layer

Currently, there is no data link technology available in the civilian domain for high bandwidth air to air communications. In this section, we define our model of the physical layer of the air to air communications system.

5.1.1. Antenna

It is assumed that all aircraft as well as ground stations are equipped with Uniform Circular Arrays of antennas (UCA), i.e. the antennas are equally spaced along a circle. The deployment of large scale antenna arrays on board passenger aircraft is a realistic assumption, as arrays have previously been used in the Connexion By Boeing system [110]. By weighting the signal with a different complex factor at each antenna, the UCA can maximize the antenna gain in the direction of the link to be activated. Thus, the transmitter increases the signal power towards the intended receiver while reducing the amount of interference generated towards other concurrent transmissions, and the receiver maximizes the amount of power that it receives from its transmitter while suppressing interference from other ongoing transmissions. In this manner, the array pattern can be steered electronically, and the direction of the beam can be changed on a per-packet basis. Expressions for the antenna weights that maximize the gain in a desired direction, as well as the resulting gain as a function of the azimuth for a UCA can be found in e.g. [2]. In Fig. 5.1 we have plotted the antenna gain for different numbers

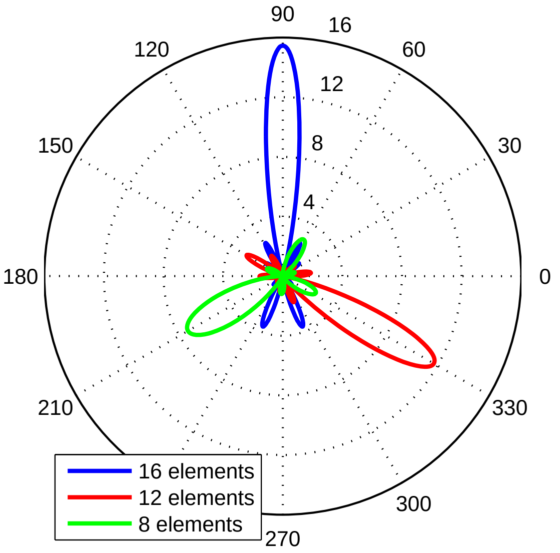


Figure 5.1.: Beampatterns of UCA for 8, 12, and 16 antenna elements; figure generated according to the UCA beampattern expression in [2].

of array elements. It can be seen that the gain of the main lobe is roughly equal to the number of antenna elements, and the number of nulls in the antenna pattern is equal to the number of elements minus two. More advanced beamforming algorithms can also steer the position of these nulls in the antenna pattern in order to reduce the interference generated to other nearby receivers by a transmitter, or to zero out interference from other transmitters at a receiver [111].

The size of such an antenna array increases with the number of elements. With an antenna spacing of one half of the wavelength and a carrier frequency of 1 GHz[1], the radius of a UCA composed of 16 antenna elements is 38.4 cm [2]. For comparison, the dimensions of the antenna used by Connexion by Boeing were approx. 46×61 cm, not including the fairing [110]. Therefore, the installation of a UCA in the 1GHz range on an aircraft appears to be realistic. In the remainder of this work, a UCA with 16 elements will be assumed at each aircraft.

5.1.2. Channel Model

Background information concerning the aeronautical radio channel was given in Section 4.1. For the remainder of this work, we will assume free space propagation within the

[1] The value of 1GHz was chosen because it lies within the aeronautical L-band, extending from 960 MHz to 1164 MHz.

radio horizon, i.e. a path loss exponent of two. We assume that beyond the radio horizon, nothing can be received.

5.1.3. Link Capacity

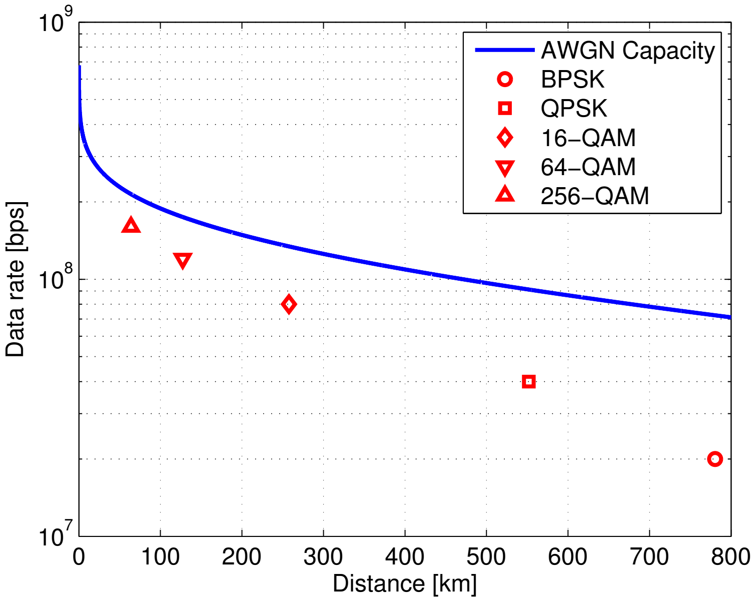


Figure 5.2.: AWGN channel capacity and achievable data rates for digital modulation.

In Fig. 5.2 we have simulated the maximum distance at which different digital modulation schemes from Binary Phase Shift Keying (BPSK) up to Quadrature Amplitude Modulation (QAM) with a symbol alphabet of size 256 are able to achieve a target bit error rate (BER) of $1 \cdot 10^{-6}$. The results rely on the expressions for the bit error probabilities given in [112]. For example, QPSK modulation requires a Signal to Noise Ratio (SNR) greater than 13.54 dB in order to achieve a BER of less than $1 \cdot 10^{-6}$. This target can be reached at distances up to approximately 550 km. The parameters assumed for this figure are summarized in Table 5.1. Fig. 5.2 shows these maximum distances and the nominal data rate of each these modulation schemes. As an upper bound, the Shannon capacity of an Additive White Gaussian Noise (AWGN) channel is also plotted as a function of the distance. A symbol rate equal to the Nyquist rate, and free space path loss are assumed. The effects of channel coding are not considered. Channel coding would improve the performance of the communications system, shifting all red markers to higher distances towards the right. Of course, other simultaneous transmissions on the wireless channel would create interference in addition to the thermal noise, thereby reducing the achievable data rates.

Spectrum in the range 960–1164 MHz, the so-called aeronautical L-band, has been allocated for aeronautical communications by the World Radiocommunication Conference

Parameter	Value
System bandwidth	20 MHz
Receiver temperature	300 K
Transmit power	10 W
Carrier frequency	1 GHz
Tx Antenna Gain	10 dB
Rx Antenna Gain	10 dB
Channel	AWGN

Table 5.1.: Radio parameters used for calculations in Fig. 5.2.

(WRC) 2007 [113]. It is planned to use this allocation for air to ground communications in continental areas [109]. We expect that this bandwidth can be reused for air to air communications in remote and oceanic regions where no ground infrastructure is present. The transmit power of 10 W in Table 5.1 is comparable to what is planned for future air to ground datalinks in the L-band [109]. As seen in the previous section, an antenna gain of 10 dB can easily be realized with a circular antenna array mounted on the aircraft.

5.2. The Link Layer

In the following sections, the model of the data link layer assumed for the high data rate air to air link that will be used in this work is presented. This includes the channel access scheme and considerations on the per link packet delay.

5.2.1. Channel Access

The channel access scheme, or Medium Access Control (MAC), determines how the nodes in the network share the wireless channel. Channel access schemes are typically divided into contention based and contention free schemes [12]. The most common contention based scheme is Carrier Sense Multiple Access (CSMA), which is used in the IEEE 802.11 (WiFi) standard [114]. In CSMA, nodes listen on the channel, to sense if it is currently being used by another node. If the channel is free, the node may transmit. If the channel is busy, the node schedules a backoff timer and senses the channel again when the timer expires. Due to its simplicity, widespread use in WiFi, and the lack of need for coordination between nodes, CSMA is the most commonly used MAC scheme in wireless ad hoc networks. However, CSMA has a number of shortcomings. The well known hidden terminal problem [115] can lead to packet collisions at the receiver, whereas the exposed terminal problem [116] causes a node to defer from transmitting, although the receiver would have been able to receive the packet correctly. Long signal propagation times resulting from the very large distances between nodes in an aeronautical network also increase the probability of collisions: Assume that some node A begins to transmit.

A second node B, which is very far away from A, may wish to transmit to the same receiver after A, but before it can hear A's transmission. It will sense the channel as being free, and begin to transmit. This will lead to a collision at the receiver, and the probability of such an event increases with the propagation times encountered in the network.

The spectral efficiency of CSMA can be increased by employing directional antennas (see e.g. [117]). This allows nodes to transmit simultaneously, because the desired signal power is increased and the interference power is reduced by the directional antennas. This is referred to as *spatial reuse*. However, the use of directional antennas also increases the complexity of the MAC algorithms and generally introduces a need for coordination between nodes.

On the other hand, contention free MAC schemes allocate resources of the wireless channel to nodes, so that collisions can be avoided. Depending on how the channel is divided up between users, we distinguish between Time, Frequency, or Code Division Multiple Access. Contention free channel access schemes have the advantage that the channel can be fully utilized, since collisions are avoided. However, the allocation of resources among the nodes can only be achieved at the cost of a higher need for coordination between the nodes, which makes the use of contention free MAC protocols difficult in highly dynamic networks. Rapid changes in the network topology would create a need for frequent changes in the allocation of resources. This would lead to a large amount of signaling. In the worst case, the MAC protocol would no longer be able to adapt fast enough to changes in the topology, and collisions would again occur due to the outdated resource allocations.

However, we have seen in Section 4.2 that the topology of an AAHN in the North Atlantic is in fact very stable. Therefore, resources that are allocated to a node can remain valid for a very long time, and the overhead that is required for coordination between the nodes is not a serious issue. The traffic demands may change more dynamically than the topology. However, due to the aggregation of many users on board a single aircraft, an averaging effect will occur. The aggregated traffic flow to or from an aircraft will fluctuate over time, but will not start or stop abruptly.

In this work, we will assume a Time Division Multiple Access (TDMA) scheme. Time is divided into TDMA frames, which are in turn composed of a number of TDMA slots. Each of these slots can be allocated to one or more links. Which links are activated in which time slot is referred to as the TDMA schedule. Due to the unicast nature of the transmissions in our network, and the assumption of directional antennas at the nodes, it is generally possible to activate multiple links simultaneously, thereby increasing the throughput of the network. Because users are not only separated in time, but also in space, the term Spatial TDMA (STDMA) was introduced for this kind of network by Nelson and Kleinrock in 1985 [11]. TDMA requires tight timing synchronization between nodes. However, this is not a major problem on board large commercial airliners. It is reasonable to assume that passenger aircraft are equipped with satellite navigation receivers, e.g. GPS, which are able to provide a sufficiently precise common timing reference for all aircraft in the ad hoc network.

An example for a small STDMA network is shown in Fig. 5.3. The network consists of four nodes, labelled A, B, C, and D. Nodes A and B are communicating with each other, and nodes C and D are also communicating with each other. All nodes are equipped with directional antennas, allowing the two transmissions to be activated simultaneously, as shown in the figure. If omnidirectional antennas were used, node B would likely not be able to receive the transmission from A, since the interfering node C is much closer to B than node A is. The exemplary STDMA schedule shown at the bottom of the figure shows that the transmission from A to B and the transmission from C to D are scheduled in the same slot. Since such slot allocations are associated with a directional link, the transmissions from B to A and from D to C must be allocated their own slot.

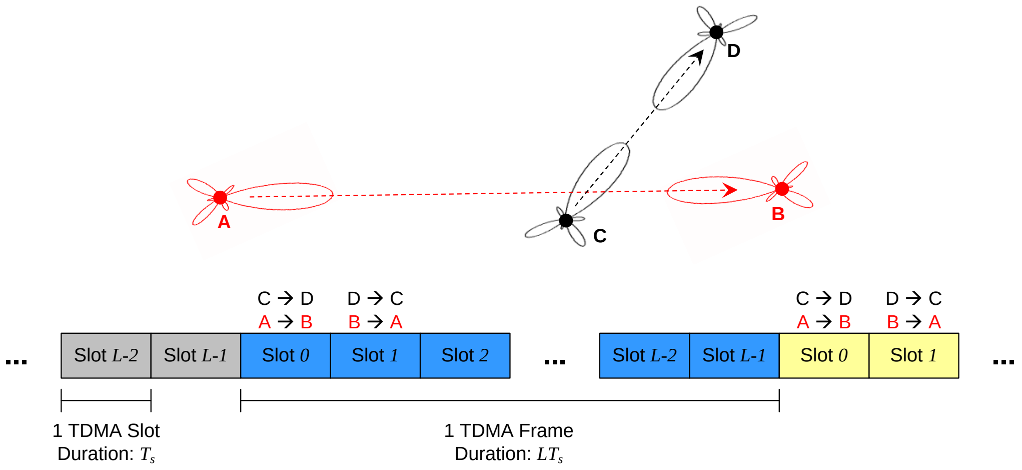


Figure 5.3.: Example of STDMA network with four nodes and two parallel transmissions.

5.2.2. Scheduling Algorithm

To construct a TDMA schedule such as the example schedule shown in Fig. 5.3, a scheduling algorithm must be run. In the literature, a number of distributed algorithms have been proposed for the scheduling of links in MANETs in a TDMA environment (see e.g. [11], [118], [119], [120], or the survey in [121]). The common goal of these algorithms is to define what information must be exchanged by network nodes and how this information is processed in order for the nodes to come up with a common view of the network's TDMA schedule. In this work, we adopt the STDMA scheduling algorithm proposed by Grönkvist [118] as the baseline scheduling algorithm. The basic idea of the algorithm proposed by Grönkvist is described below.

Each node continuously monitors the arrival rate of packets in the transmit queue of each of its outgoing links. The priority of a link is then defined as the ratio of its estimated packet arrival rate to the number of slots per frame that are currently assigned to that link. Links periodically exchange their priorities and their current view of the TDMA schedule. At each update of the schedule, typically once per frame, every link checks if it has the highest priority among all links within its local neighborhood. This local neighborhood is defined as the union of the two hop neighborhoods of the transmitter and the receiver. If a link has the highest priority within its local neighborhood, it is allowed to allocate itself an additional time slot. To do so, it must ensure that the Signal to Interference and Noise (SINR) at its own receiver will be above the SINR threshold required for reliable communication and that the additional interference generated by its own transmission will not lead to any other active links SINR falling below its communication threshold. According to [118], this check can be based on interference measurements or on information gathered from the schedule. Obviously, duplex constraints preventing a node from transmitting and receiving simultaneously must also be respected when searching for a valid time slot. Depending on the antenna model, simultaneous transmissions by a node to multiple receivers, or simultaneous reception from several transmitters, can be allowed. Here, we will assume that a node can only transmit or receive on one link at a time. Due to these SINR and duplex constraints, a link will potentially need to check several TDMA slots before finding one in which it can add itself.

If the link with the highest priority is not able to find any slot in which it can add itself to the schedule, it is allowed to steal slot allocations from other links with lower priority. That is, it removes a slot allocation for another link in order to free up the schedule, and then assigns itself a slot allocation in the slot that has become free. Note that a link is only allowed to steal from a 'victim' link with lower priority than itself. Since the victim link loses a slot allocation due to the stealing, its priority will increase, and the priority of the stealing link will decrease due to the additionally allocated slot. For stability, a link is only allowed to steal from another link, if the priority of the stealing link *after* stealing is still higher than the priority of the victim link after stealing. If still no slot can be found, the link enters a "sleep" state in which it no longer attempts to assign itself a slot, in order to give other links with lower priority the opportunity to receive more slots, increasing the spatial reuse in the network. The link wakes up from the sleep state again after a certain duration and then continues to participate in the contention for slot allocations. Since the traffic patterns or the interference from other links may have changed in the meantime, it may now be able to assign itself additional slots again, or its priority may have dropped, so that it no longer requires additional slots. All changes in the schedule, such as slot allocations or the stealing of slots from other links, are communicated to the other links when the TDMA schedules are again exchanged periodically. If any link detects an inconsistency between its view of the schedule and the schedule communicated from another link, it must restrain from transmitting in any inconsistent slots in order to prevent potential packet collisions.

The full details of the algorithm are given in [118]. The most important characteristic of this algorithm is that it is traffic sensitive, i.e. links with a higher traffic load will have a higher priority, and thus will be able to allocate themselves more slots than other links with less load. Ideally, the algorithm converges to a state in which all links have the same priority, i.e. the allocation of slots is proportional to the links' traffic load.

5.2.3. Link Delay

In order to assess the performance of the network in terms of delay, it is important to find an appropriate model for the delay of packets on a link. It is possible to model each wireless link as a queueing system. In [12], a distinction is made between regular TDMA systems, where each user is only assigned a single time slot, and so-called Generalized TDMA systems, where each user may be assigned an arbitrary number of slots. The Spatial TDMA system with Grönkvist's scheduling algorithm that is assumed here is essentially a Generalized TDMA scheme. Each link can be seen as a user, since slots are assigned to links, not to nodes. The delay of a packet on a link consists of three distinct components: queueing delay, offset delay, and transmission delay.

Queueing delay refers to the amount of time that a packet must wait in the queue until there are no more packets in front of it. The average queueing delay depends on the link's packet arrival rate and its service rate, which is typically defined by the number of slots assigned to the link.

Offset delay is the amount of time that the packet must wait until its transmission can begin, i.e. a time slot assigned to the link comes up in the schedule, after it has come to the front of the queue. This results from the "phase shifted" operation of the incoming and outgoing links at a node. While it is clear that the outgoing links must provide enough capacity to relay the total incoming traffic (minus traffic consumed by the node) and traffic generated at the node, the beginning of the incoming slot can be shifted in time relative to the beginning of the next outgoing slot, leading to a certain offset delay for a packet. In the ideal case, this delay can be zero, if the outgoing slot immediately follows the incoming slot. In the worst case, the offset delay can be the entire frame length, minus one slot. For the packets in a given flow, this offset delay at a node may appear deterministic since the same schedule is used in all frames, and the routing of the flow is also constant (over a sufficiently short observation interval). However, the average offset delay over all flows in the network depends on the number of time slots assigned to each link, as well as the temporal distribution of slots within the frame. The average offset delay is minimized when the slots are uniformly distributed in the TDMA frame and maximized when the slots allocated to the link form one contiguous region within the frame.

Finally, transmission delay is the time required to actually send the packet over the channel. We assume here that the transmission delay is constant and corresponds to one slot duration per packet.

The service rate $\mu_{i,j}$ of a link (i,j) from node i to node j, expressed in terms of packets per second, is

$$\mu_{i,j} = \frac{h_{i,j}}{LT_s}, \tag{5.1}$$

where $h_{i,j}$ is the number of slots assigned to link (i,j) per frame, L is the length of the frame in slots, and T_s is the slot duration in seconds. The utilization $\rho_{i,j}$ of link (i,j) is defined as the ratio of the packet arrival rate $\lambda_{i,j}$ to the service rate

$$\rho_{i,j} = \frac{\lambda_{i,j}}{\mu_{i,j}} = \frac{\lambda_{i,j} L T_s}{h_{i,j}}. \tag{5.2}$$

A utilization greater than one indicates that the link is overloaded. In this case, packets would build up in the transmission queue. Eventually, the queue would become full and further incoming packets would need to be discarded.

In case the ratio $\tilde{L} = \frac{L}{h_{i,j}}$ is integer, and the slots assigned to a link are spaced at uniform intervals of $\tilde{L}$ slots, the generalized TDMA system simplifies to a regular TDMA system with with only one slot per user and frame. In this case, the average delay of a packet on the link is given by [12]

$$\delta_{i,j} = T_s \left(1 + \frac{\tilde{L}}{2(1 - \rho_{i,j})} \right), \tag{5.3}$$

if the arrival of packets in the queue can be modeled as a Poisson process.

The scheduling algorithm proposed by Grönkvist attempts to spread the slots assigned to a link as evenly across the frame as possible, in order to minimize the offset delay. In Fig. 5.4, we show the distribution of the distance in slots between two successive slots assigned to a link. These values were gathered from a simulation of Grönkvist's scheduling algorithm that we have performed in a network of 50 nodes, four ground stations, and 80 slots per frame. Traffic flows in downstream direction from the gateways to the network nodes. Each of the 50 nodes is the sink of a traffic flow originating from one of the ground stations.

Since the distribution of the distance in slots between two successive slots assigned to a link depends on the number of slots assigned to a link, results for links with $h_{i,j}$ =2, 4, 8, and 16 slots are shown. As can be seen, the distributions always exhibit very sharp peaks at $\tilde{L} = \frac{L}{h_{i,j}}$, corresponding to equal inter-slot distances. This was to be expected, since the algorithm tries to distribute the slots equally in the frame. Less prominent peaks can be observed at $2\tilde{L}$. These peaks are caused by to slot theft. If a link with high priority attempts to assign a time slot to itself, but cannot find a slot anywhere in the frame in which its addition would not violate any SINR constraints, it may "steal" a slot from a link with lower priority, i.e. remove the other link from the schedule, and schedule itself in that link's time slot. If the slots of the link from whom the slot was stolen were previously all spaced at equal distances, two slots will now be spaced at twice the original distance, since the slot in between has been stolen. This effect creates the peaks at $2\tilde{L}$. For comparison, the red lines in the figure show the expected distribution of the inter-slot distances in case the slots are randomly assigned, resulting in an exponential distribution. It can be seen that the assumption of equally spaced slots is well fulfilled by the scheduling algorithm assumed for our network.

In Fig. 5.5, the per link delay as a function of the link utilization is shown for the queuing model, both for equidistant slot spacings according to Eq. 5.3 and for random slot spacings. In addition, the per link delay in a simulated network running Grönkvist's

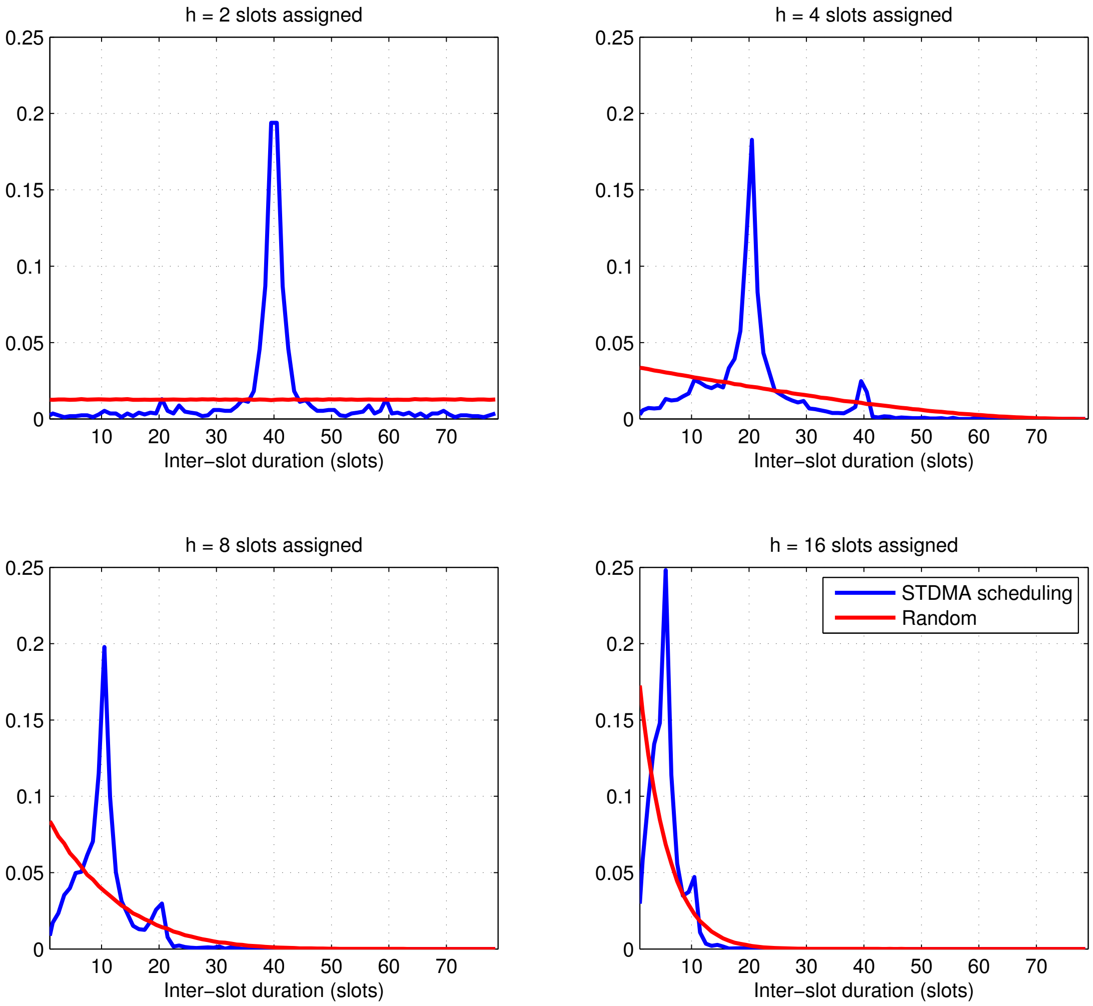


Figure 5.4.: Distribution of duration (in slots) between two successive slots assigned to a link for Grönkvist's scheduling algorithm (blue lines) and random allocation of slots (red lines) for h = 2, 4, 8, 16.

scheduling algorithm is shown. As in the previous figure, the network consists of 50 aircraft and four ground stations. Each TDMA frame consists of 80 slots of duration 10 ms. It is assumed that exactly one packet can be transmitted per time slot. At low traffic load, the delay of the system is dominated by the offset time, so here the difference between uniform and random slot distributions is greatest. At higher traffic load, queuing delay becomes more and more important in comparison to offset delay, and the difference between the two systems decreases. Note also that as the utilization goes to zero, the delay of both systems floors out at different levels. The regular TDMA system with equidistant slot allocations achieves lower average delay than the GTDMA system with a random slot distribution.

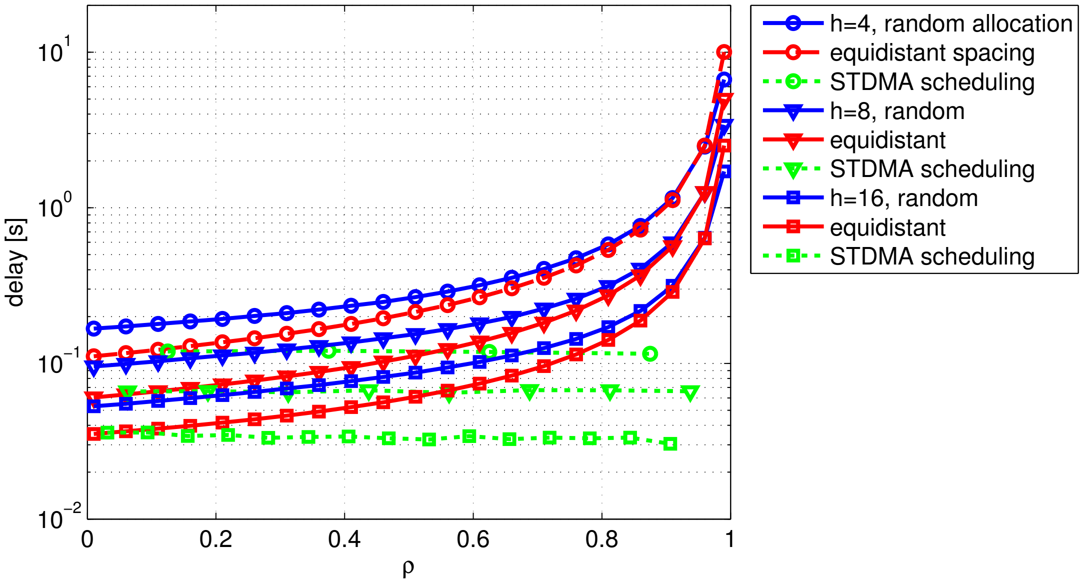


Figure 5.5.: Average per link delay according to queueing model assuming random slot allocations (solid lines), equidistant spacing of slots (dashed lines), as well as delay measured in simulations of Grönkvist's scheduling algorithm (dotted lines).

However, as seen in Fig. 5.5, the actual per link delay measured in the network does not significantly depend on the utilization, as long as the utilization remains below one, i.e. as long as the link is not overloaded. Rather, it depends only on the absolute number of slots that have been allocated to a link. Only when the utilization becomes greater than one and a link's queue begins to build up indefinitely does the delay become unbounded. This indicates that the delay model derived above, which is given in Eq. 5.3, is not appropriate for our traffic sensitive TDMA network. The reason for this is that the packet arrivals on a link do not follow the Poisson distribution that was assumed for the queueing model. The rate at which packets can arrive is limited by the number and distribution of the time slots that are allocated to the incoming links at the head of the link under consideration. Effectively, this regulation of the incoming traffic prevents large fluctuations in the queue length and the larger delay values that would result from such fluctuations. If we model the delay experienced by a packet as the time required to transmit that packet (i.e. one slot duration) plus the average offset delay until the next available time slot, the average delay can be expressed as

$$\delta_{i,j} = T_s \left(1 + \frac{L}{2h_{i,j}}\right), \tag{5.4}$$

again assuming that the time slots allocated to a link are spaced at regular intervals. The average delay according to Eq. 5.4 is plotted together with the measured delay from our network simulations in Fig. 5.6. Obviously, this expression models the link delay much better than Eq. 5.3 and will be used for the modeling of the network in the remainder of this work. Again, it is important to point out that our delay model only considers offset delay and transmission delay, but not queueing delay. This model is justified by simulations of the behavior of the STDMA link layer.

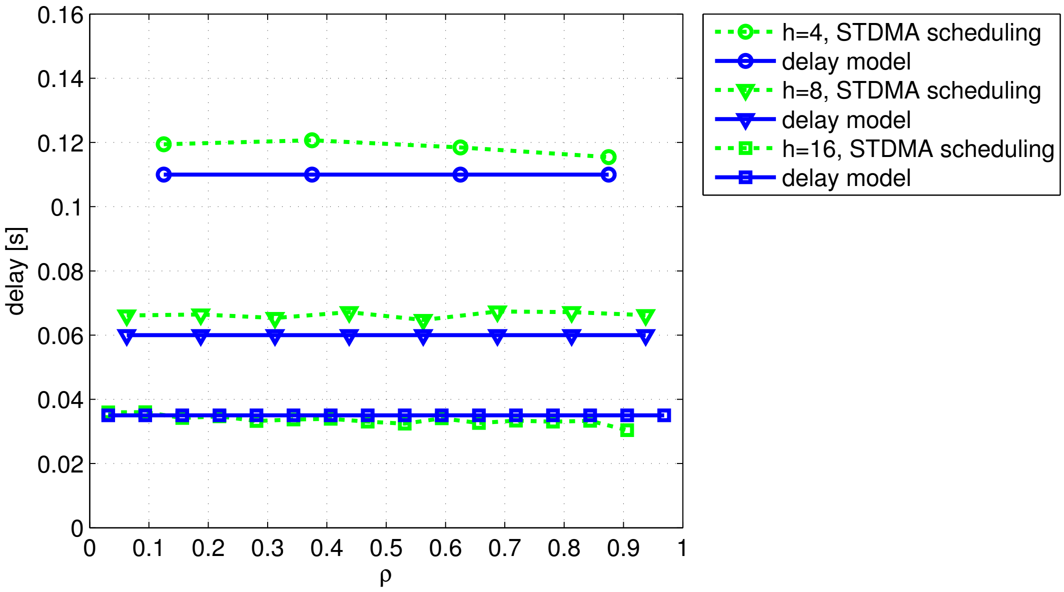


Figure 5.6.: Average per link delay according to Eq. 5.4, as well as delay measured in simulations of Grönkvist's scheduling algorithm (dotted lines).

Aiming to achieve an equal distribution of the slots in the frame attempts to minimize the average per link delay. However, it is also possible to devise a schedule that further reduces the end to end delay of packets along the entire path. In [122], Djukic *et al.* propose a centralized algorithm that reduces the delay on multi-hop paths by finding a schedule in which the outgoing links of a node are assigned slots shortly after the incoming links, so that the offset delay of packets is minimized. However, [122] does not present a distributed algorithm capable of solving this task in an operational network.

As seen above, the average packet delay depends on the duration of a time slot, T_s. Shorter time slots result in a lower per link delay. However, time slots cannot be made arbitrarily short. Packets transmitted by two different aircraft in the same time slot will arrive at a third aircraft with different propagation delay. This can lead to collisions at a receiver if a packet received in time slot n overlaps into the following time slot $n+1$. In cellular radio systems, this problem is typically solved by the base station sending timing correction commands to the mobile terminals, so that all packets transmitted by the mobile terminals arrive at the base station exactly inside their allocated time slot [123]. However, such timing correction is not possible in an ad hoc network due to the lack of a common point of reference. Therefore, guard intervals must be introduced between time slots in order to compensate for these differences in propagation delay. These guard intervals must be long enough to ensure that a packet transmitted by an aircraft at the maximum possible communications distance away from a receiver has been fully received before the next time slot begins. If we wish to compensate all propagation delays within the maximum radio horizon of 952 km, we would need guard intervals of 3.2 ms. Limiting the length of guard intervals to a smaller value also limits the maximum distance allowed for links, because longer propagation times can no longer be compensated. Of course, longer guard intervals also result in a lower utilization of the wireless channel, since no data can be transmitted during the guard intervals.

In order to maximize channel utilization, it is desirable to keep the TDMA slots as long as possible. On the other hand, longer slots lead to higher packet delay. Therefore, the choice of a slot length must always be a tradeoff between these two objectives.

5.3. Network Model

In Chapter 3, we introduced the network architecture that is assumed for this work. For the algorithm design of the following chapters, we use a simplified network model that captures only those components that are directly relevant for the behavior of the algorithms. This network model is shown in Fig. 5.7. The entire mobile network of each aircraft is collapsed into a single node which aggregates the traffic of all nodes onboard the aircraft. For each service class, each aircraft is seen as the starting point of a single upstream traffic flow towards the Internet and the endpoint of a single downstream traffic flow. Likewise, the Internet is represented by a single node, which is the source of all downstream traffic flows. For each service class, a single downstream flow is generated towards each aircraft. All ground stations are connected directly to this ”Internet node” via a link with delay equal to zero. Satellite gateways also have a direct connection to the ground node with a delay value corresponding to the satellite link propagation delay.

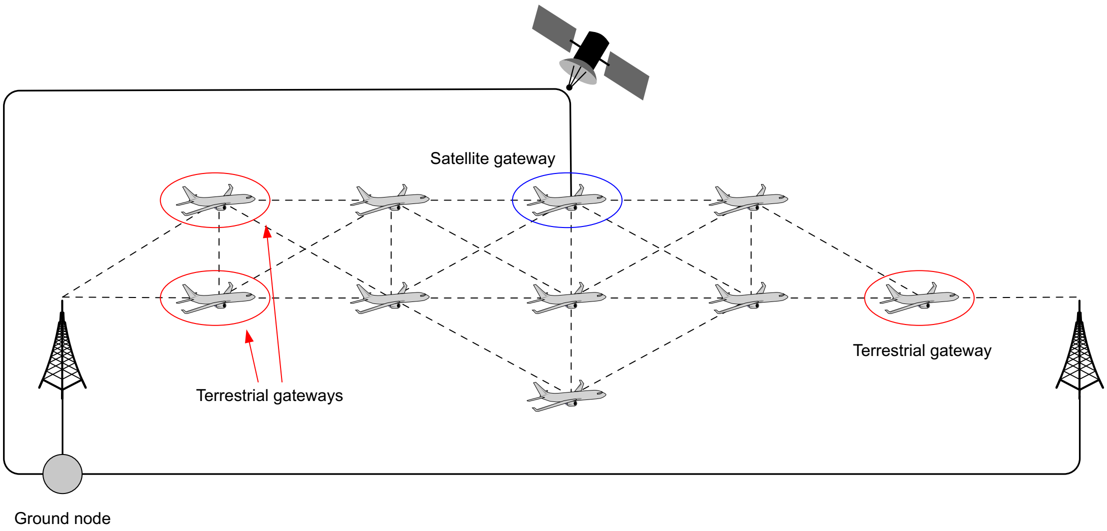


Figure 5.7.: Network model showing connectivity to ground networks as well as structure of onboard network.

6. The Gateway Selection and Routing Problem

In this chapter, a mathematical formulation of the joint gateway selection, routing, and scheduling problem is given. It is shown that a straightforward minimization of the average packet delay in the network is a nonconvex problem, and an alternate suboptimal, but computationally more efficient, approach is proposed. Large parts of the work presented in this chapter have been previously published by us in [124].

6.1. Network Model

First, we will define the terminology that will be used in formulating the optimization problem. The notation used is summarized in Table 6.1. The network is represented as a directed graph $\mathcal{G}(\mathcal{U}, \mathcal{V})$, where $\mathcal{U}$ is the set of all nodes in the network and $\mathcal{V}$ is the set of all directional links connecting these nodes. The subset of aircraft nodes is denoted $\mathcal{U}_{\mathrm{AC}}$, and the subset of those aircraft that are acting as Internet gateways is denoted $\mathcal{U}_{\mathrm{GW}}$. In our model, the ground network is represented by a single node, referred to as node 0. All Internet gateways are directly connected to this node.

Each aircraft in $\mathcal{U}_{\mathrm{AC}}$ is considered to be an active user in the network, generating an upstream flow of packets towards node 0, and consuming a downstream flow of packets from node 0. These ordered pairs (p, q) of source node p and sink node q comprise the set of all flows $\mathcal{F}$, each with a known traffic demand $R_{p,q}$, $(p, q) \in \mathcal{F}$, which is given in bits per second. Since the endpoints of each flow are given by the aircraft generating or consuming the flow and node 0, the problem of allocating a gateway to a flow is implicitly contained in the routing problem. If the route between node 0 and the aircraft is known, the gateway is the node on the path that is adjacent to node 0.

For communication to be feasible between any two nodes, we require that the link between the nodes fulfill a minimum Signal to Interference and Noise Ratio (SINR) γ_0. Assuming a maximum transmit power P_{tx} and free space signal propagation allows us to determine the maximum distance at which communication is possible in the absence of interference, denoted $d_{\max}$:

$$d_{\max} = \frac{\lambda}{4\pi}\sqrt{\frac{P_{\mathrm{tx}}}{k_B T \gamma_0}}, \tag{6.1}$$

where λ is the wavelength of the signal, k_B is the Boltzmann constant, and T is the receiver temperature. Thus, a link (i, j) from node i to node j exists, if the distance $d_{i,j}$

between the nodes satisfies

$$d_{i,j} < \min(d_{\text{horizon}}, d_{\max}), \tag{6.2}$$

where d_{horizon} is the radio horizon as defined in Eq. 4.1. Then, the set of all wireless ad hoc links in the network graph (i.e. not including the links between A/G gateways and node 0, or between satellite gateways and node 0) is given by

$$\mathcal{V}_{\text{air}} = \{(i,j)|d_{i,j} < \min(d_{\text{horizon}}, d_{\max}),\ i,j \in \mathcal{U},\ i \neq j\}\,. \tag{6.3}$$

The neighbors $\mathcal{N}_i$ of node i are those nodes to which i can establish a communications link:

$$\mathcal{N}_i = \{j \in \mathcal{U}|(i,j) \in \mathcal{V}\} \tag{6.4}$$

The set of links $(p,q) \in \mathcal{V}_{air}$ that can cause interference to a link $(i,j) \in \mathcal{V}_{air}$ are those links whose transmitter p is within the radio horizon of node j. According to the channel model in Section 5.1, interference from transmitters beyond the radio horizon is neglected. This set of links interfering with receiver j is denoted $\mathcal{I}_j$:

$$\mathcal{I}_j = \{(p,q) \in \mathcal{V}_{\text{air}}|d_{p,j} \leq d_{\text{horizon}}\}\,. \tag{6.5}$$

Note that this set of interfering links does not consider the nodes' antenna patterns. All interference contributions, no matter how small, are considered, as long as they are inside the radio horizon.

To model the STDMA protocol regulating the users' access to the channel, we introduce the set of binary variables $\{u_{i,j}[n]\}$:

$$u_{i,j}[n] = \begin{cases} 1 & \text{if link } (i,j) \text{ has been allocated slot } n, \\ 0 & \text{else.} \end{cases}$$

With this notation, SINR $\gamma_{i,j}[n]$ of link (i,j) in slot n is

$$\gamma_{i,j}[n] = \frac{u_{i,j}[n]G_{i,j}[n]P_i d_{i,j}^{-2}}{n_j + \sum_{(p,q)\in\mathcal{I}_j} u_{p,q}[n]G_{p,j}[n]P_p d_{p,j}^{-2}}, \tag{6.6}$$

where n_j is the noise at the receiver[1] and P_i is the transmit power of node i. $G_{i,j}[n]$ denotes the combined transmit antenna gain of transmitter i in the direction of receiver j and receive antenna gain of j in the direction of i in slot n. This model allows for arbitrary antenna patterns. In order for a link to be used for transmitting data, we require that its SINR according to Eq. 6.6 be greater than some threshold γ_0. In this case, the link is able to support a corresponding nominal data rate of R_0 bits per second.

We define the routing variables $\ell_{(i,j)(p,q)} \in [0,1]$ to denote the fraction of traffic demand of the flow from node p to node q that is sent over link (i,j). To force the solution to use only single-path routing, these variables can be restricted to be binary: $\ell_{(i,j)(p,q)} \in \{0,1\}$. However, it is also possible to consider the more general case of continuous variables and multipath routing.

[1] Actually, n_j is the thermal noise, scaled by the factor $\frac{\lambda^2}{(4\pi)^2}$, where λ is the wavelength of the carrier.

Variable	**Description**
$\mathcal{G}$	connectivity graph
$\mathcal{U}$	set of all network nodes
$\mathcal{U}_{\mathrm{GW}}$	set of gateway nodes
$\mathcal{U}_{\mathrm{AC}}$	set of aircraft nodes
$\mathcal{V}$	set of all links
$\mathcal{V}_{\mathrm{air}}$	set of all wireless links within the ad hoc network
$\mathcal{N}_i$	set of all neighbor nodes of node i
$\mathcal{I}_i$	set of all links potentially causing interference at node i
$\mathcal{F}$	set of all flows in the network
$R_{p,q}$	target data rate of flow (p, q)
$\ell_{(i,j)(k,l)}$	variable indicating whether link (i, j) is used for flow (k, l)
$u_{i,j}[n]$	variable indicating whether link (i, j) is activated in slot n
$\gamma_{i,j}[n]$	SINR of link (i, j) in slot n
γ_0	min. required SINR of a link
R_0	data rate provided by a link
η_j	thermal noise at node j
P_i	transmit power of node i
$d_{i,j}$	distance between nodes i and j
$G_{i,j}[n]$	combined antenna gains of transmitter i towards receiver j and of j towards i in slot n
L	number of time slots per frame
T_s	time slot duration
n	time slot index
$\delta_{i,j}$	mean delay of packets on link (i, j)
D	mean flow delay
$h_{i,j}$	number of slots allocated to link (i, j) per frame
$w_{i,j}$	weight assigned to link (i, j)

Table 6.1.: Summary of notation used for the joint routing, scheduling, and gateway selection problem.

6.2. Problem Formulation

Our overall goal is to balance traffic in the wireless network such that the passengers on the aircraft are provided with a satisfactory quality of service. At the same time, we wish to avoid the satellite links whenever possible, due to their high cost and delay. This leads us to define the average packet delay in the network as the performance metric that we attempt to minimize. The packet delay is directly related to the users' perception of the network QoS. At the same time, including the per link delay of packets in the cost function will naturally shift traffic away from the high delay satellite links under lightly loaded traffic conditions. Only when the traffic demand increases, leading to higher delay or congestion on the A/G links (or the A/A links in the vicinity of the terrestrial gateways), or when an aircraft is so far away from the nearest terrestrial gateway in terms of hops that the direct satellite link becomes more attractive, will an increasing amount of traffic be routed over the satellite links. Finally, the round trip time between a node in the wireless network and its correspondent node in the ground network has a direct influence on the throughput that can be achieved by a TCP flow between these two end nodes when the network is lossy [125]. Thus, even bulky data transfers that are not delay sensitive will benefit from increased throughput when the packet delay is minimized.

Of course, objective functions other than the average delay are also possible. A common optimization goal in wireless networks is to maximize the throughput directly. However, in our scenario, we assume that the traffic load is given and must be handled in the most efficient way. Maximization of the throughput would lead to the solution that all gateways, including the satellite gateways, are transmitting data into the network at the maximum possible rate. Instead, we are interested in using the satellite gateways only when required in order to fulfill the users' demand. We assume that every flow (p, q) in the network is associated with a target rate $R_{p,q}$, which is given in bits per second.

Summing over all flows in $\mathcal{F}$, and all links used by each flow, the average flow delay D can be written as

$$D = \frac{1}{|\mathcal{F}|} \sum_{(p,q)\in\mathcal{F}} \sum_{(i,j)\in\mathcal{V}} \ell_{(i,j)(p,q)}\delta_{i,j}, \tag{6.7}$$

where $\delta_{i,j}$ is the average link delay as defined in Eq. 5.4 for all A/A and A/G links. This is the expression that we are interested in minimizing. The number of slots assigned to link (i, j), which is needed for the average link delay, is given by

$$h_{i,j} = \sum_{n=1}^{L} u_{i,j}[n], \tag{6.8}$$

where L is the number of slots per frame. The delay of links between gateways and node 0 is assumed to be constant, but may be different for different gateways.

In addition to the objective function itself, a large number of constraints is required to ensure that the problem provides a valid routing and scheduling solution. In particular, it must be ensured that the paths found are connected from source to destination, and preferably free of loops. In addition, each link must be assigned sufficient capacity in terms of slots to carry the traffic load that it has been allocated, and for every link, the SINR at the receiver must be above the specified SINR threshold in every slot in which the link is active. The joint gateway allocation, routing, and scheduling problem for delay minimization is summarized below:

$$\min_{\{u_{i,j}\},\{\ell_{(i,j)(p,q)}\}} \quad D \tag{6.9}$$

$$\text{s.t.} \sum_{j\in\mathcal{N}_p} \ell_{(p,j)(p,q)} = 1 \quad \forall (p,q) \in \mathcal{F} \tag{6.10}$$

$$\sum_{i\in\mathcal{N}_q} \ell_{(i,q)(p,q)} = 1 \quad \forall (p,q) \in \mathcal{F} \tag{6.11}$$

$$\sum_{i\in\mathcal{N}_k} \ell_{(i,k)(p,q)} = \sum_{j\in\mathcal{N}_k} \ell_{(k,j)(p,q)} \ \forall k \in \mathcal{U}\backslash\{0\}, (p,q) \in \mathcal{F}, \ k \neq p,q \tag{6.12}$$

$$\sum_{j\in\mathcal{N}_i} (u_{i,j}[n] + u_{j,i}[n]) \leq 1 \ \forall i \in \mathcal{U}\backslash\{0\}, 1 \leq n \leq L, \tag{6.13}$$

$$R_0 \sum_{n=1}^{L} u_{i,j}[n] \geq \sum_{(p,q)\in\mathcal{F}} \ell_{(i,j)(p,q)} R_{p,q} \quad \forall (i,j) \in \mathcal{V}_{\text{air}}, \tag{6.14}$$

$$\gamma_{i,j}[n] \geq \gamma_0 u_{i,j}[n] \quad \forall (i,j) \in \mathcal{V}_{\text{air}}, \ 1 \leq n \leq L. \tag{6.15}$$

Constraints 6.10 and 6.11 assure that flows begin at their source and are terminated at their destination, respectively. Flow conservation at relay nodes is guaranteed by 6.12. Due to physical limitations, a node's wireless interface cannot transmit and receive at the same time, nor can it transmit to or receive from two different nodes simultaneously. These duplex constraints are covered by Constraint 6.13. Constraint 6.14 ensures that the total capacity allocated to a link is at least equal to the total traffic demand on that link. Finally, Constraint 6.15 ensures that the SINR at the receiver of any link active in slot n is above the minimum required SINR threshold γ_0. This constraint is a highly nonlinear function, as can be seen from the definition of the SINR in Eq. 6.6. However, it has been shown in [126] that this constraint can be rewritten as an equivalent linear constraint by taking advantage of the fact that the link activation variables $\{u_{i,j}\}$ are binary. The linearized form of Constraint 6.15 is then written as

$$G_{i,j} P_i d_{i,j}^{-2} \geq \gamma_0 \left(\eta_j + \left(\sum_{(p,q)\in\mathcal{I}_j} u_{p,q}[n] G_{p,j} P_p d_{p,j}^{-2} \right) + \psi u_{i,j}[n] - \psi \right), \tag{6.16}$$

where ψ is a sufficiently large constant. When $u_{i,j}[n]$ is equal to one, the last two terms on the right hand side of the inequality cancel out, and the constraint is equal to the original constraint. When $u_{i,j}[n]$ is equal to zero, link (i,j) is inactive and the

SINR constraint does not need to be enforced. In this case, the term in the parentheses reduces to $\eta_j + \sum_{(p,q)\in\mathcal{I}_j}(u_{p,q}[n]g_{p,j}P_p d_{p,j}^{-2}) - \psi$. If ψ is chosen to be large enough, this term becomes negative, and the inequality is automatically fulfilled. In practice, ψ should not be made much larger than absolutely necessary, though, in order to avoid numerical difficulties. This linearization does not work if the variables continuous, as could be the case if power control is also added to the scope of the problem. However, it is possible to include discrete power control by allowing multiple links between the same pair of nodes, each assigned a distinct power level and capacity. This would come at the expense of a larger number of optimization variables.

Note that the routing constraints do not explicitly force paths to be free from routing loops. However, the objective function itself prevents such loops. Since any unneeded links forming a loop contribute to the total delay, they will be eliminated in the optimization process without violating any of the other constraints. In case the objective function is modified, it may become necessary to add a constraint preventing routing loops.

6.3. Related Work

An efficient approach to solving the joint scheduling, routing, power control and rate adaptation problem in Wireless Mesh Networks based on a column generation technique has been presented by Luo *et al.* in [127]. The optimization goal in [127] is to maximize the minimum throughput among all flows. The scheduling problem is modeled as the scheduling of maximal Independent Sets (ISs) in the network's conflict graph, i.e. sets of links that can be active at the same time. In contrast, we assume a TDMA schedule with slots of fixed duration. The problem of IS scheduling can be solved more efficiently, since it is formulated as a continuous problem of assigning a fraction of the total time to each maximal IS, whereas our $u_{i,j}[n]$ are discrete variables. However, the maximal ISs must be precomputed before the actual optimization run. Our optimization goal is quite different from [127] in that we wish to handle a given demand with the lowest delay possible, rather than maximizing the throughput.

Another approach to solving the joint optimization of routing and scheduling in mesh networks using directional antennas has been proposed in [128] by Capone *et al.*, with the goal of minimizing the total number of time slots needed to satisfy the given demand. This is analogous to maximizing the throughput, as the shorter schedule can be repeated more often in the same period of time. Packet delay, and the allocation of nodes to gateways were both not considered in this work. Livingstone *et al.* address the problem of gateway allocation and routing and formulate a suboptimum solution to maximizing the network throughput in [129].

When the number of potential satellite gateways is large, the decision which gateways should be used is similar to the Gateway Placement Problem (GPP) [130] encountered in mesh network planning. In the GPP, potential sites for gateways are known, and a smaller number of gateways are assigned to a part of these sites. Typical objectives of the GPP are to minimize the number of gateways or to minimize the total cost of deploying gateways while fulfilling some performance target, e.g. minimum throughput.

In [131], Papadaki *et al.* address the problem of planning wireless mesh networks. They formulate a capacitated and an uncapacitated version of an optimization problem to select which mesh routers should act as gateways and determine the routing solution that minimizes the total cost of gateway usage. Again, delay is not addressed in this formulation, nor is the interference between the wireless links considered. The combination of GPP and routing is also addressed by Targon *et al.* in [132] with the goal of minimizing the cost of the gateway deployment.

6.4. Complexity Considerations

Unfortunately, the joint gateway allocation, routing, and scheduling problem formulated above in the previous section is a nonconvex problem. In general, the global optimum of nonconvex problems cannot be found. Convex functions must be continuous, whereas our problem contains a large number of integer variables. Even if the integer constraints are relaxed, the delay minimization objective function remains nonconvex[2]. Even without the nonconvex objective function, the scheduling problem alone has been shown to be NP-complete [134] and therefore can only be solved in reasonable time for very small networks. These practical limitations lead us to search for an alternate formulation of the problem that can be solved more efficiently.

6.5. Two-Step Decomposition

Since the original delay minimization problem is non-convex and thus cannot be optimized globally, the problem is divided into two subproblems. First we optimize the routing in the network by minimizing a weighted hop count. Subsequently, we optimize the scheduling for the previously computed routing solution.

Such a decomposition of a routing and scheduling problem into subproblems is also proposed by Livingstone *et al.* in [129]. However, the authors in [129] considerably simplify the routing subproblem by not considering any scheduling constraints. Their goal in the first step is to minimize the maximum node utilization (i.e. the ratio of the traffic handled by a node to its nominal data rate) in the network. However, such an approach has a number of drawbacks. Since interference and scheduling constraints are not accounted for, this may in fact result in a routing solution that cannot be scheduled, although a feasible routing and scheduling solution does exist. Second, the minimization of the maximum node utilization does not consider the length of paths through the network and may select arbitrarily long paths, as long as the load of the most highly loaded node is not increased. Third, the minimization of the maximum node utilization will in general lead to a solution that spreads traffic out over as many nodes as possible. From a scheduling point of view, such a solution may be undesirable, since it leads to many links with relatively low load competing for access to the wireless channel.

[2] A multivariate function is convex if and only if its Hesse matrix is positive semidefinite [133].

Therefore, we consider the scheduling and interference constraints already in the first step of the problem, in order to ensure that a feasible schedule for the resulting routing exists.

First, we aim to find a reasonable routing and gateway allocation scheme. To do so, we assign weights $w_{i,j}$ to each link, allowing e.g. satellite links to be assigned a higher cost than air/ground links, thereby avoiding the costly satellites. We then find a routing solution that minimizes this weighted hop count. The constraints of this subproblem are the same as those in the original delay minimization problem formulated in Section 6.2. That is, we require a feasible schedule to exist for the routing and gateway allocation solution that is produced by this step. The first step of our two-step approach, the minimization of the Weighted Hop Count (**mWHC**) is defined as follows:

$$\min_{\{u_{i,j}\},\{\ell_{(i,j)(p,q)}\}} \sum_{(i,j)\in\mathcal{V}} w_{i,j} \sum_{(p,q)\in\mathcal{F}} R_{p,q}\ell_{(i,j)(p,q)}, \tag{6.17}$$

$$\text{s.t.} \sum_{(i,j)\in\mathcal{V}} \ell_{(p,j)(p,q)} = 1 \quad \forall (p,q)\in\mathcal{F} \tag{6.18}$$

$$\sum_{(i,j)\in\mathcal{V}} \ell_{(i,q)(p,q)} = 1 \quad \forall (p,q)\in\mathcal{F} \tag{6.19}$$

$$\sum_{i\in\mathcal{N}_k} \ell_{(i,k)(p,q)} = \sum_{j\in\mathcal{N}_k} \ell_{(k,j)(p,q)} \; \forall k\in\mathcal{U}\backslash\{0\}, (p,q)\in\mathcal{F},\; k\neq p,q \tag{6.20}$$

$$\sum_{j\in\mathcal{N}_i} (u_{i,j}[n]+u_{j,i}[n]) \leq 1 \; \forall i\in\mathcal{U}\backslash\{0\}, 1\leq n\leq L, \tag{6.21}$$

$$R_0\sum_{n=1}^{L} u_{i,j}[n] \geq \sum_{(p,q)\in\mathcal{F}} \ell_{(i,j)(p,q)}R_{p,q} \quad \forall (i,j)\in\mathcal{V}_{\text{air}}, \tag{6.22}$$

$$\gamma_{i,j}[n] \geq \gamma_0 u_{i,j}[n] \quad \forall (i,j)\in\mathcal{V}_{\text{air}},\; 1\leq n\leq L. \tag{6.23}$$

In a second step, we take the routing and gateway allocation solution that is provided by the first step in the form of the $\{\ell_{(i,j)(p,q)}\}$ variables and optimize the schedule for this traffic pattern. In principle, the scheduling variables $\{u_{i,j}[n]\}$ produced by the first step are discarded, but they can also be provided to the solver of the second step as an initial solution, to reduce the number of solver iterations that are required. Step 2 of the problem, the minimization of the Average Flow Delay (**mAFD**), is defined as follows:

$$\min_{\{u_{i,j}\}} \sum_{(p,q)\in\mathcal{F}} \sum_{(i,j)\in\mathcal{V}} \ell_{(i,j),(p,q)}\delta_{i,j}, \tag{6.24}$$

$$\text{s.t.} \sum_{j\in\mathcal{N}_i} (u_{i,j}[n] + u_{j,i}[n]) \leq 1 \;\; \forall i \in \mathcal{U}\backslash\{0\}, 1 \leq n \leq L, \tag{6.25}$$

$$R_0 \sum_{n=1}^{L} u_{i,j}[n] \geq \sum_{(p,q)\in\mathcal{F}} \ell_{(i,j)(p,q)} R_{p,q} \;\; \forall (i,j) \in \mathcal{V}_{\text{air}}, \tag{6.26}$$

$$\gamma_{i,j}[n] \geq \gamma_0 u_{i,j}[n] \;\; \forall (i,j) \in \mathcal{V}_{\text{air}},\; 1 \leq n \leq L. \tag{6.27}$$

The mAFD problem minimizes the average flow delay in the network, while respecting the duplex constraints 6.25, rate constraints 6.26, and minimum SINR constraints 6.27. The flow conservation constraints of the original problem are no longer needed here, since the routing has already been decided.

The output of these two problems are the variables $\{\ell_{(i,j)(p,q)}\}$ from step 1, telling us the exact route used for the traffic of each flow, as well as the variables $\{u_{i,j}[n]\}$ from step 2, telling us the STDMA schedule, i.e. which links are activated in which time slot. Due to the separation of the routing and scheduling problems, this formulation does not provide the globally optimum solution to the original problem of Section 6.2. However, decomposing the problem into two steps allows it to be solved much more efficiently. The mWHC problem can be solved particularly efficiently, since it is a pure linear mixed integer program, which can be solved by standard branch and bound routines [135]. Unfortunately, the mAFD problem is still NP-complete, thus limiting the applicability of this problem to small networks. However, the number of variables in the mAFD problem is significantly lower than the number of variables in the original joint routing, gateway selection, and scheduling problem, since the routing variables have been fixed. This allows the decomposition approach to be applied to larger problems than the original problem. Also, the continuous relaxation of the mAFD problem is convex. This property is important for the solver, since it allows bounds on the objective function to be calculated more efficiently.

Simulation results to assess the performance of the proposed mWHC/mAFD optimization problem will be presented in the following chapter in Section 7.5.

7. Genetic Algorithm Approach to Gateway Selection and Routing

Due to the computational complexity of the joint routing, gateway selection, and scheduling problem, the mathematical programming approach presented in the previous chapter can only be applied to very small networks. This is especially true for the original problem defined in Eq. 6.9 – Eq. 6.15, but also applies to the decomposed mWHC/mAFD problem. In this chapter, we present an optimization approach based on Genetic Algorithms, which has the advantage that it can be applied not only to static networks, but to mobile networks as well. For a small test network, the performance of this approach is compared to the performance of the mathematical programming approach that was presented in the previous chapter.

7.1. Introduction to Genetic Algorithms

Genetic Algorithms (GAs) have become increasingly popular in the last years as a simple but effective means of solving optimization problems that are difficult or practically impossible to solve using more traditional optimization methods such as integer or nonlinear programming. A thorough introduction to GAs can be found in [136]. The idea behind GAs is to consider a large number, or population, of potential solutions to the problem in parallel. Each of these solutions is referred to as an individual of the population, and the manner in which the solution to the problem is encoded in each individual is referred to as the genome. For example, the genome for a network scheduling problem could be a binary vector of zeroes and ones, where each entry specifies whether a certain link is active in a certain slot or not. Through a series of operations acting on the population, the fitness of the population is improved step by step until, after some time, the algorithm has converged and no further improvement can be reached. Staying in line with the biologically inspired terminology of Genetic Algorithms, each iteration is referred to as a generation.

In each generation of the algorithm, the performance of each individual is determined according to a fitness function. In the scheduling problem, the fitness of an individual could be defined as the throughput that is achieved by the schedule that is encoded by that individual's genome. Based on their fitness, a subset of individuals is selected to remain in the population in the next generation, and some less fit individuals are discarded. To replace these individuals, the genomes of some individuals of the population, termed parents, are combined with each other, thereby creating new genomes, or children. This operation is called recombination or crossover. In the scheduling problem example, this could mean that the first half of the newly created child schedule is copied from one of the parents' schedule, and the second half of the schedule is copied from the other parent's schedule. In addition to the recombination of the parents' genomes, random changes. or mutations, may also be introduced in the genome of the children. For example, entries of the scheduling vector could be arbitrarily flipped from a zero to a one, or vice versa, with a certain probability.

This whole process, consisting of the steps of recombination between genomes, random mutations, and selection of individuals based on a fitness function, mimics the natural process of evolution. The offspring may have either a higher or lower fitness than their parents. However, the selection of individuals based on their fitness assures that those offspring with lower fitness will slowly disappear from the population, whereas the fitter individuals are retained, thereby gradually improving the overall fitness of the whole population.

The main advantage of Genetic Algorithms is their inherent parallelism. Since an entire population of potential solutions to a problem is considered simultaneously, the GA is less likely to become stuck at a local optimum than a gradient search method. Also, the requirements on the cost function are relatively weak. For example, it does not need to be differentiable. Both of these points are important arguments for our application of Genetic Algorithms to the routing and scheduling problem. Due to the integer variables in the average delay function Eq. 6.9, the cost function is not differentiable. Also, the continuous relaxation of the cost function is not convex, indicating the presence of local optima.

On the other hand, the main weakness of Genetic Algorithms is that there is no guarantee that the algorithm will converge to the global optimum, or even to a local optimum. In fact, the algorithm can converge to an arbitrary point in the search space. However, the risk of this occurring can be reduced by increasing the probability of mutations. Also, since the fitness function must be evaluated for all individuals in the population in every generation, the overall complexity of the GA can be quite high if the fitness function is complex.

Genetic Algorithms have been proposed for use in the optimization of wireless networks in a number of papers. Recently, Lee *et al.* [137] have presented a GA approach to the allocation of wireless users to access points in a WiFi network. However, their scenario is limited to a single-hop environment, and does not address radio resource scheduling. GAs have been previously applied to the problem of joint routing and scheduling in Wireless Mesh Networks by Badia *et al.* [126] and by Pries *et al.* [138]. Badia *et al.* demonstrate the general suitability of GAs for solving such routing and scheduling problems. However, the authors do not apply an objective function related to the network performance, and focus mainly on finding a feasible solution. Pries *et al.* also address the joint routing and channel assignment problem in WMNs. Each node is assigned a radio channel on which it is allowed to transmit. Although the allocation of channels is similar to the allocation of TDMA slots, our problem formulation is more comprehensive, in that each link may be allocated an arbitrary number of TDMA slots. Therefore, the resource scheduling can be better adapted to the actual traffic load in the network.

7.2. Proposed Genetic Algorithm for Static Networks

An overview of the population update performed by the GA in each generation is given in Fig. 7.1, showing the major steps of selection, recombination, and mutation. The details of each step of the proposed GA will be discussed in the following sections. The goal of this algorithm is to solve the delay minimization problem subject to the routing and scheduling constraints defined in Eq. 6.9 – Eq. 6.15 for a given network topology and flow demands.

7.2.1. Genome Encoding

A solution to the routing and scheduling problem must be encoded within each genome of the population. For each flow, a path is encoded as a list of nodes in the order that they are visited by a packet from a source to its destination. During the operation of the GA, nodes may be inserted, removed, or replaced in this list. The TDMA schedule of each genome is encoded in a separate table containing the set of active links in each time slot. A slot that is allocated to a link is not tied to any dedicated flow. During the operation of the GA, links may be added to or removed from the schedule of a genome, or rescheduled from one slot to another. In general, each node in the network will have one upstream (US) flow of traffic from the node to the ground network and one downstream (DS) flow in the other direction. The traffic load of these two flows can be different.

7.2.2. Cost Function

As in the problem formulated in Section 6.2, the goal of our optimization is to minimize the average packet delay in the network. Therefore, the cost function according to which an individual g of the population is ranked must reflect the packet delay. To penalize infeasible solutions, i.e. genomes in which at least one link is overloaded, we also add a penalty term that is proportional to the amount of unfulfilled demand in the network:

$$penalty(g) = \sum_{(i,j)\in\mathcal{V}} \max\left(load_{(i,j)}(g) - \sum_{n=1}^{L} u_{i,j}[n], 0 \right), \tag{7.1}$$

where g is a genome of the population and $load_{(i,j)}(g)$ is the total traffic load of link (i, j), as it results from the routing of flows in the network according to genome g. This penalty term takes over the role of the capacity constraints in Chapter 6. Using the notation of Chapter 6, we can write

$$load_{(i,j)}(g) = \sum_{(p,q)\in\mathcal{F}} \ell_{(i,j)(p,q)} R_{p,q}. \tag{7.2}$$

The max operator in Eq. 7.1 ensures that a link with overprovisioned capacity does not affect the penalty function. Thus, the cost function is written as

$$c(g) = delay(g) + penalty(g), \tag{7.3}$$

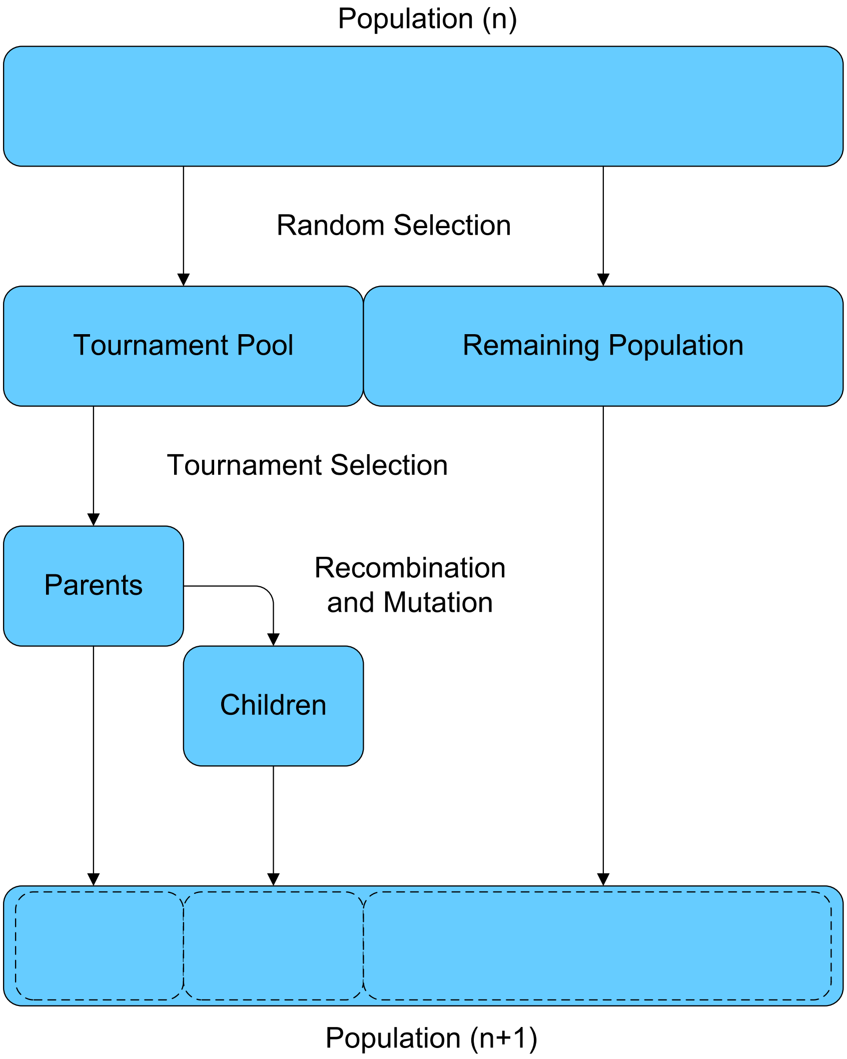


Figure 7.1.: Update mechanism of proposed Genetic Algorithm from generation n to generation $n + 1$.

where the average delay of the genome, $delay(g)$, is calculated according to Eq. 6.7.

The term *fitness* is more commonly used for the ranking of individuals. However, our problem is formulated in such a manner that we wish to minimize the cost instead of maximizing the fitness of the population. In some cases, it may be possible that no feasible solution is found by the algorithm before it converges. In this case, the individual with the lowest cost is the one that is best able to fulfill the traffic demand in the network.

Note that the partitioned mWHC/mAFD optimization problem formulated in Section 6.5 did not attempt to minimize the average delay directly, due to the nonconvex nature of this problem. However, GAs can be applied to nonconvex problems quite successfully, as in [139] for cooperative spectrum sensing in cognitive radio networks. This is due in part to the random initialization of the population across the search space, and partly to the random modifications caused by mutations and crossover, which allow the GA population to overcome local optima.

7.2.3. Initialization

When a GA is applied to a problem, an initial population must be generated before the process of recombination, mutation, and selection can begin. A common technique is to initialize the genomes randomly. In principle, this is also possible in our case. However, our joint routing and scheduling problem possesses a large number of constraints that separate feasible solutions from infeasible solutions, and almost all randomly chosen initial solutions would be infeasible. Although the penalty term in the cost function would direct the algorithm towards feasible solutions, we dedicate some additional effort to initializing the population with solutions that are likely to be close to feasibility, in order to accelerate the convergence of the algorithm. More precisely, the genomes are initialized with a valid path to a gateway for every flow, and all subsequent operations on the genome ensure that these paths remain valid. A valid path is defined as a list of nodes beginning with the flow's source and ending with the flow's destination in which each node is within transmission range of its preceding and following nodes, and each node is allowed to appear at most once.

For this purpose, we initially calculate the shortest path between each node and each gateway. When a new individual is created, a path to a random gateway is selected for each flow and time slots are scheduled for each flow. A link is allowed to be scheduled in a certain slot if the SINR of all links already active in that slot as well as the SINR of the new link remain above the required SINR threshold after the new link is added. It is possible that some links may not receive a sufficient number of slots initially, since links cannot be scheduled due to the interference constraints. However, this can be fixed by the following evolutionary process, or the genome will eventually be removed from the population because the overloaded links contribute to the penalty term in the cost function. Due to the selection of a random path for each flow, the genetic diversity of the population is still quite high. Subsequent mutations will lead to deviations from these precomputed paths that may use the wireless channel more efficiently by allowing for more spatial reuse in the network.

7.2.4. Selection

Several different mechanisms have been proposed for selecting the subset of a population that will generate offspring for the next generation (see e.g. [140]). The most simple form of selection is based on *ranking* all members of the population according to their fitness. The fittest members are allowed to recombine in order to create new members for the next generation, whereas the least fit individuals are removed from the population.

A second approach, which is used here, is referred to as *tournament selection* [141]. Here, a pool of individuals is randomly chosen as potential parents for the next generation. From this pool, two individuals are selected at random and the fitter of the two is kept as a parent, whereas the less fit individual is dropped from the population. The major advantage of this approach compared to ranking selection is that less fit individuals also have a chance of creating offspring. They must only be selected for a tournament in which the second participant is less fit. This aspect is quite important, since modifications of a genome, either due to recombination or mutation, may not immediately result in a higher fitness. For example, if a node on a path is replaced with another one, the fitness may initially be worse, since only few time slots have been reserved for the new links. However, over the course of several generations, additional time slots may be allocated to these new links, until the fitness of the genome has become higher than it was before the switching of the node. Thus, a selection mechanism that does not immediately remove individuals with lower fitness from the population is well suited to our problem.

7.2.5. Crossover

A crossover is the creation of new individuals from a set of parents. To ensure that the paths remain valid, only complete paths are exchanged between two genomes. When two genomes are recombined, each of their paths for a flow may be exchanged with a certain probability. However, this exchange may create conflicts in the schedule. Therefore, it is first checked, whether the links of the new path can be scheduled in the same time slots as they were assigned in the parent. If this is not possible for a link, the algorithm attempts to schedule the link in a different time slot. If it is not possible to schedule one or more of the links, this will again be reflected by the penalty term in the cost function.

7.2.6. Mutation Operators

The purpose of mutation operators is to introduce small random variations in the genomes of some individuals, in the hope that these will improve the fitness of the individual. In fact, most mutations will degrade the fitness, and may lead to the individual being removed from the population. However, the new information that is brought into the population by a mutation may occasionally lead to an improvement that would not have been achievable by means of recombination alone. Whenever a new individual is created by a crossover between parent individuals, the new individual may be affected by mutations. Each mutation operator is activated with a certain probability, which can

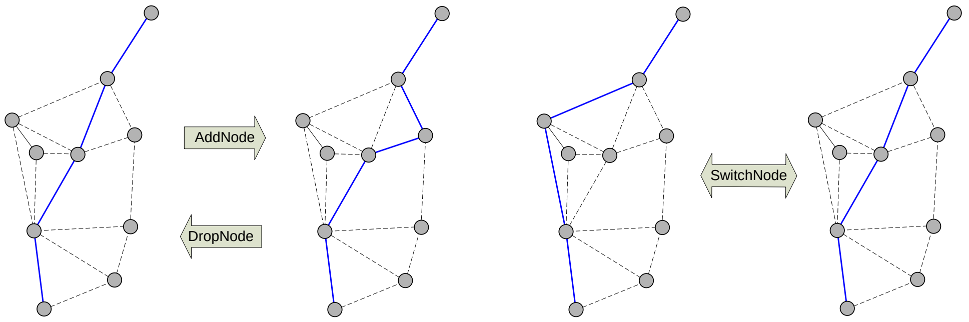


Figure 7.2.: Mutation operators acting on routing.

be adjusted as a parameter of the GA.

The problem that we are faced with when choosing appropriate mutation operators is that a random change in the routing or the scheduling will in most cases lead to an infeasible solution. Therefore, instead of randomly modifying parts of the genome, we propose the following operations that consider the constraints of the network optimization problem. These mutations operate at three distinct levels of granularity. The first three mutation operators modify only the allocation of slots to links. The next three operators act on the routing, by modifying the path that is taken between any two nodes. However, the path remains connected and free of loops. Finally, the last mutation operator acts at the flow level, replacing a flow's entire path with a path ending at another gateway.

- *Slot insertion*: This operation attempts to schedule an additional slot for a link that is already active in some other slot, thereby reducing the delay of packets on this link.

- *Slot removal*: This operation attempts to remove a time slot that has been allocated to a certain link from the schedule. However, it only does so if the time slots that remain allocated to the link are still sufficient to carry its allocated load. The benefit of this operator is that it frees up resources in the schedule, which may later be consumed by other links.

- *Slot exchange*: This operation attempts to remove a time slot that has been allocated to a certain link from the schedule and replace it with another time slot. Thus, the link's capacity, and the genome's cost function, are not affected. However, the resulting schedule may allow links that previously were not able to receive further slots to do so now.

- *Node insertion*: This operation attempts to insert an additional node k into a path by breaking up an existing link between nodes i and j into two links (i, k) and (k, j). The insertion is only carried out if the two new links can be scheduled successfully, such that the entire traffic load on the original link can be supported by the new links as well.

- *Node removal*: This operation attempts to remove a node j from a path by merging the incoming and outgoing links (i, j) and (j, k) into the single link (i, k). The merging is only performed if the new link can be scheduled successfully, supporting all of the traffic of the original two links.

- *Node exchange*: This operation attempts to reroute traffic by replacing a node j by another node ℓ. The incoming and outgoing links (i, j) and (j, k) must accordingly be replaced by the links (i, ℓ) and (ℓ, k), respectively. Again, this operation is only carried out if both new links can be scheduled successfully.

- *Path exchange*: This operation replaces an flow's entire path, similar to the manner in which the crossover operator works. Here however, paths are not exchanged between two genomes. Rather, a random path from the initial set of shortest paths is selected to replace an existing path of the genome. This can bring new information into the population and significantly alter the interference situation. Note that this operator has a unique function in that it is the only mutation operator that allows a node to change its Internet gateway. This has two notable effects: First, the newly selected path may not have been represented in the initial population, thereby bringing significant new information into the population. Second, the slots that were previously allocated to the existing path are no longer needed. These slots are freed, and slots are scheduled for the links along the new path. Thus, links on or near the original path may profit from the reduced level of interference.

Fig. 7.2 shows how the mutation operators can modify the path used to route a flow. The node insertion, removal, and exchange operators serve to modify the path between a node and its gateway. To prevent paths from becoming longer and longer, it is reasonable to assign a slightly higher probability to the node removal operator. The node exchange can be seen as a combination of node insertion and removal into a single step. The slot insertion operator reduces the cost of a genome by reducing the delay of some link in the network. As more and more links in the network are scheduled, it will become more difficult to find a free slot to allocate to a link. The slot removal operator frees up unused capacity on some link, thereby potentially allowing an additional slot to be allocated to link with higher load. A slot exchange does not affect the genome's cost directly, but other links may profit from the modified schedule.

7.2.7. Convergence Criterion

As in any iterative algorithm, a convergence criterion must be defined for the GA [136] in order to halt the execution of the algorithm when it is unlikely that the algorithm will find a solution that is better than the current best solution. In genetic algorithms, convergence is typically declared when the maximum fitness of the population no longer improves significantly over a duration of several generations. The actual threshold that is used to define what is considered to be a significant improvement, as well as the number of generations over which the improvement is considered, depend on the actual problem at hand. However, convergence can also be defined using other metrics, such as when the mean fitness of all individuals in the population no longer improves significantly, or when the diversity in the population has become less than some threshold [142]. Of course, static criteria such as a maximum number of generations or a maximum execution time can also be defined.

When the convergence criterion is fulfilled, the fittest individual of the population is taken as the solution to the problem. If the convergence criterion is too loose, there is a higher probability that the algorithm will stop prematurely before it has found the optimum solution. On the other hand, an excessively strict criterion will cause the algorithm to continue running, although no substantially better solution is found.

Here, convergence of the algorithm was declared if, for ten subsequent generations, the relative performance improvement of the genome with the lowest cost over the last 100 generations was less than $5 \cdot 10^{-5}$. These values have been found to be reasonable values empirically, based on simulations of the proposed GA.

It is necessary to look at the change in the cost function over such a large number of generations due to the discrete nature of the cost function, especially when no feasible solution has been found. In this case, many generations may pass without any change in the cost function. Therefore, looking at the difference between two subsequent generations is not a good indication of the algorithm's convergence. In any case, we allow a maximum of 5000 generations before halting the algorithm.

In practice, GAs perform well on many different types of problems. However, in general it is not possible to prove that a GA will converge to a global, or even a local optimum of a problem. This is due in part to the probabilistic nature of the algorithm, and partly to the complicated internal structure in which several different mechanisms – selection, recombination, and mutation influence the convergence of the algorithm. The main theoretical result regarding optimality of GAs is the so-called *schema theorem*, which was originally formulated by Holland in 1975 [143]. This theorem states that the frequency of schemata, or substrings of the genome, that are relatively short and have above average fitness will increase exponentially from generation to generation, subject to some assumptions on the structure of the GA. It has been shown in [143] that this behavior is very close to optimal.

7.3. Tracking of Mobile Networks

The algorithm defined in the previous section is applied to static networks. That is, it is given a certain network topology and flow demands as input, and attempts to minimize the average packet delay for this static scenario. However, the aeronautical ad hoc network that we are interested in is not static. Even if the aircraft move only a small amount, the relative positions of the aircraft do change, causing the interference levels that are seen at the receivers to change. This may render a previously feasible schedule suddenly infeasible. As the aircraft move even more, existing links may break up, or new links may be created. This can cause not just the schedules, but also the routing solutions to become infeasible.

With the mathematical programming approach of the previous chapter, or the GA defined in the previous section, it would be necessary to restart the algorithm and compute a new solution whenever any aircraft positions change. Clearly, this would be very inefficient. Therefore, in this section, we define extensions to the GA for static networks that allow it to be applied to dynamic networks as well. The changes in the network topology are incorporated into the population of the GA as they occur, allowing the GA to track the mobile network continuously.

Changes in the network topology, such as the creation or deletion of links or nodes can easily be incorporated into the population of the GA on the fly. In case link (i, j) breaks, the algorithm checks whether any genome was using this link to carry traffic. If this is the case, it tries to repair the path by finding an intermediate node k such that the links (i, k) and (k, j) can be activated instead. Likewise, if an aircraft node is deleted, the algorithm checks all genomes if this node was used to relay traffic for any other node. If so, it again tries to reroute the flow around the deleted node. If it is not possible to reroute the flow in either case, the algorithm tries to replace the entire path with a path to a different gateway. If a new link is created, no special actions are taken. However, subsequent mutations may introduce this link into the existing genomes. If a new aircraft node is created, the corresponding up- and downstream flows are introduced into every genome. In addition, the movement of the nodes continuously changes the interference levels at each receiver. Therefore, each time slot of the TDMA schedule must be checked in each generation in order to ensure that the set of simultaneously activated links are still compatible, i.e. each receiver's SINR is above the threshold γ_0. In case of an SINR violation, one or more of the links need to be rescheduled or deallocated in order to return to a valid schedule. Of course, this leads to a significant increase in the algorithm's complexity per step.

After these changes have been made to all individuals in the population, the algorithm continues to run with the modified population. Obviously, this is much more efficient than beginning a new optimization run with every change in the topology. One danger in this approach is that the population converges during the operation of the GA. When a change in the network topology occurs, it may become difficult for the GA to react if the population has become too homogeneous. This effect can be prevented by running the algorithm with a relatively high mutation rate. To prevent these frequent mutations from degrading the performance of the best individual in the population, which will be used to derive the routing and scheduling information for the nodes in the network, it is useful to introduce the concept of elitism [144]. In Genetic Algorithms, this means that the fittest individuals of the population, referred to as the *elite set*, are automatically copied into the population of the next generation and cannot be lost due to selection or mutations. Fig. 7.3 shows the update mechanism from one generation to the next. In contrast to the GA for static networks, this diagram includes the elite set. To keep the population size constant, the new population is filled up with individuals that are randomly selected from that part of the previous population that is not included in the elite set or tournament pool.

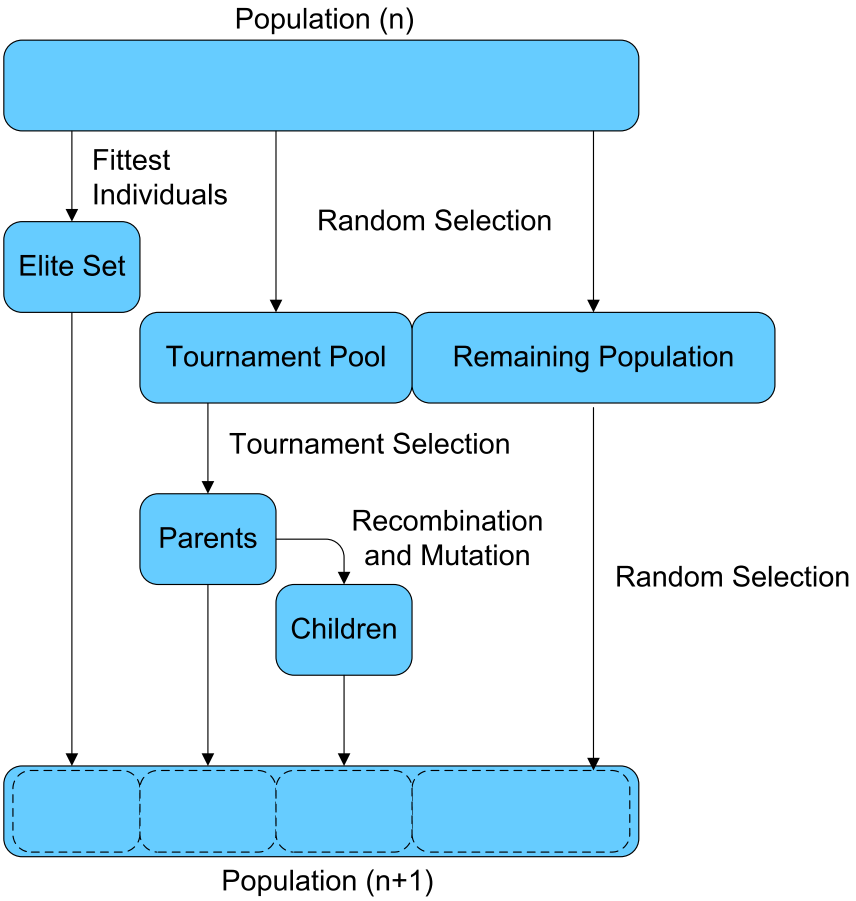


Figure 7.3.: Update mechanism of proposed Genetic Algorithm for tracking of mobile networks.

When tracking a mobile network on the fly, the GA does not check for convergence, as in the static case. Rather, it continues to run indefinitely. In principle, the algorithm can be run with a constant number of generations per simulated second. However, it is more effective to increase the algorithm's update rate whenever significant changes have occurred in the topology and slow the algorithm down when the topology is more static, or when the GA's solution seems to have converged. This improves the algorithm's ability to react to changes quickly and prevents unnecessary effort when the algorithm has already converged to a solution.

7.4. Multiple Service Classes

In practice, it may be desirable to offer different Classes of Service (CoS) to users in the network, allowing e.g. delay sensitive traffic such as Voice over IP (VoIP) or real time streaming of audio or video to be assigned a guaranteed Quality of Service (QoS) in terms of bandwidth or delay, whereas other traffic, such as web browsing or file transfer are treated according to a best effort policy. In this section, we define modifications to our proposed GA that allow it to handle multiple service classes with different QoS targets in terms of delay. Each CoSi is associated with a delay target, denoted $delayTarget(i)$. The Classes of Service are sorted according to their delay targets, so that for $i < j$, $delayTarget(i) < delayTarget(j)$.

In principle, the encoding of the routing and scheduling information in the genomes remains the same. However, each aircraft in the network now has one pair of US and DS flows for each service class. Let $(p, q)_x \in \mathcal{F}$ denote the flow of CoSx from node p to node q, and let $R^x_{p,q}$ be its associated data rate. Although US and DS flows of the same CoS should still use the same GW, different CoS may use different GWs. The major change with respect to the GA defined in Section 7.2 lies in the way that the fitness of the genomes is determined. In Section 7.2, the cost function of Eq. 7.3 consisted of the penalty term accounting for capacity violations on the links, and the average packet delay. Here, two different penalty terms for the capacity violations and the violations of the delay target are used. The penalty for capacity violations is split between the different service classes:

$$penaltyDemand_{\mathrm{CoS}X}(g) = \sum_{(i,j)\in\mathcal{V}} \max\left(\sum_{x=1}^{X} load_{(i,j),\mathrm{CoS}x}(g) - \sum_{n=1}^{L} u_{i,j}[n],\ 0\right). \tag{7.4}$$

This definition of the penalty for unfulfilled demand differs from the original definition in Eq. 7.1 for a single CoS. The demand for CoS1 is assumed to be fulfilled if the capacity of the link is greater than or equal to the demand of CoS1. In general, the load of a link due to CoSx, denoted $load_{(i,j),\mathrm{CoS}x}(g)$, is defined as

$$load_{(i,j),\mathrm{CoS}x}(g) = \sum_{(p,q)_x\in\mathcal{F}} \ell_{(i,j)(p,q)} R^x_{p,q}, \tag{7.5}$$

The demand for CoS2 is assumed to be fulfilled only if the capacity of the link is greater than or equal to the combined demand of CoS1 and CoS2 on that link, and so on. The delay penalty term of CoSX is then defined as

$$penaltyDelay_{\mathrm{CoS}X}(g) = \sum_{(p,q)_X\in\mathcal{F}} \max\left(delay_{(p,q)_X}(g) - delayTarget(X),\ 0\right), \tag{7.6}$$

where $delay_{(p,q)_X}(g)$ is the delay of flow $(p, q)_X$ as was defined in Eq. 6.7 and was used in the cost function of the GA for a single CoS, $delayTarget(X)$ refers to the delay target of CoSX. The summation is performed over all flows of the CoSX.

In summary, service classes are sorted according to their priority. Each CoS is associated with a predefined delay target. The GA first attempts to ensure that the demand of the CoS with the highest priority is met on every link. Then, the GA attempts to ensure that the delay target of all flows of this CoS are met. When the delay target has also been met, the GA proceeds to the next service class. Finally, if the demand and delay targets of all service classes are fulfilled, the GA tries to further reduce the average packet delay over all flows. This behavior is reflected by the algorithm by which the cost of two genomes $g1$ and $g2$ is compared is summarized in Table 7.4. This algorithm returns the genome with the lower cost.

The crossover and mutation operators remain unchanged with respect to those defined in Section 7.2 for a single service class.

Algorithm Genome cost comparison for multiple CoS

```
for all CoS$i$ do
  if $penaltyDemand_{\mathrm{CoS}i}(g1) < penaltyDemand_{\mathrm{CoS}i}(g2)$ then
    return $g1$
  else if $penaltyDemand_{\mathrm{CoS}i}(g2) < penaltyDemand_{\mathrm{CoS}i}(g1)$ then
    return $g2$
  end if
  //Both genomes have the same demand penalty for CoS$i$
  //(Typically, this will be 0.)
  if $penaltyDelay_{\mathrm{CoS}i}(g1) < penaltyDelay_{\mathrm{CoS}i}(g2)$ then
    return $g1$
  else if $penaltyDelay_{\mathrm{CoS}i}(g2) < penaltyDelay_{\mathrm{CoS}i}(g1)$ then
    return $g2$
  end if
  //Both genomes have the same delay penalty for CoS$i$
  //(Again, this will typically be 0.)
end for
if $delay(g1) < delay(g2)$ then
  return $g1$
else
  return $g2$
end if
```

Table 7.1.: Genome cost comparison for multiple service classes.

7.5. Performance Assessment

In this section, we assess the performance of the proposed Genetic Algorithm and the mWHC/mAFD mathematical programming approach of Sec. 6.5. This includes both the behavior of the optimization algorithms, e.g. the evolution of the GA population and the convergence speed of the GA, or the number of solver iterations required for the mWHC/mAFD solution. In addition, the quality of the solutions provided by the algorithms are assessed in terms of packet loss and packet delay. The mWHC/mAFD optimization problem has been solved with the Lingo optimization software [145]. The Genetic Algorithm implementation as well as the simulations to determine the network performance have been performed using the OMNeT++ network simulator [146]. After the routing and gateway selection optimization has been performed for a given topology and set of traffic demands using the GA and mWHC/mAFD approaches, packet level simulations are performed for the different solutions using OMNeT++. Packets are randomly generated by the source nodes in the network and then forwarded through the network according to the gateway selection and routing solutions that have been previously calculated. As performance metrics, the packet loss rates and the average packet delay are recorded for the different solutions.

The performance of the proposed Genetic Algorithm has been assessed in small scale and large scale scenarios. In the small scale scenario, the GA is compared to the mWHC/mAFD approach and to a simple hop count based routing and gateway selection scheme. In the large scale scenario, the mathematical programming approach can no longer be applied due to the computational complexity of the problem, and the GA is only compared to the hop count solution.

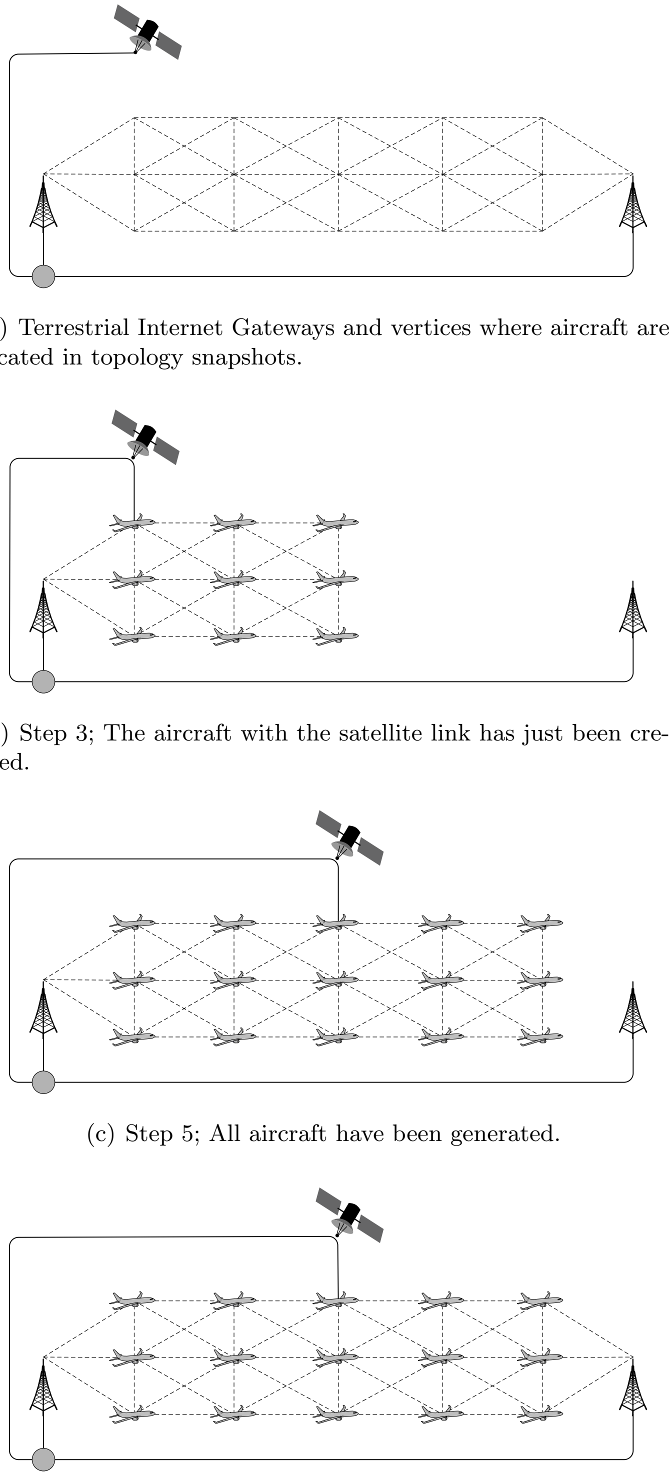
7.5.1. Small Scale Scenario

Simulation Scenario

For the small scale performance assessment, we consider a simulation topology with aircraft flying from left to right between two air/ground gateways, as shown in Fig. 7.4(a). At first, only three aircraft are located near the left gateway. Shortly before these aircraft lose their direct links to the gateway, three new aircraft are generated at the left. This process repeats until the first three aircraft have come within range of the right gateway. At this moment, the network consists of a total of 15 aircraft, as shown in 7.4(d). One aircraft is equipped with a satellite link and can also act as Internet gateway.

However, the mWHC/mAFD optimization can only be performed for static topologies, so that we also consider six discrete snapshots of this mobile network. In the first step, only three aircraft are present, occupying the three leftmost vertices. In each subsequent step, all aircraft move one position to the right, and three additional aircraft are introduced at the left side, until in step 5, all 15 vertices of the grid are occupied by aircraft, but the rightmost aircraft do not yet have connectivity to the air/ground gateway on the right. This step corresponds to Fig. 7.4(c). These three air/ground links are introduced in step 6 (Fig. 7.4(d)), without any further movement by the aircraft. The mWHC/mAFD optimization and the static GA have been carried out for each of these six discrete steps.

As described in Chapter 5, both the aircraft and ground stations are equipped with Uniform Circular Arrays consisting of 16 antenna elements, whose beam patterns are calculated so as to maximize the gain along the direction of the link. The frame length



(a) Terrestrial Internet Gateways and vertices where aircraft are located in topology snapshots.

(b) Step 3; The aircraft with the satellite link has just been created.

(c) Step 5; All aircraft have been generated.

(d) Step 6; The cloud of aircraft has reached the Internet gateway on the right.

Figure 7.4.: Small scale topology at different points in time.

Parameter	Value, static case	Value, mobile case
Population size	600	600
Selection mechanism	Tournament selection	Tournament selection
Pool size	260	260
Elite set size	0	14
p(slot insertion)	0.02	0.02
p(slot removal)	0.02	0.02
p(slot exchange)	0.02	0.03
p(node insertion)	0.01	0.02
p(node removal)	0.02	0.02
p(node exchange)	0.02	0.04
p(path exchange)	0.02	0.04
p(crossover)	0.1	0.2

Table 7.2.: Summary of GA parameters used for simulations.

is set to eight slots, with one time slot lasting 0.01 s. A minimum SINR of 10 dB is required for a link to be usable for communication. In this case, one packet per slot may be transmitted over a link. For simplicity, only downstream traffic is considered in this scenario. The ground node generates packets for each aircraft according to a Poisson process at a rate of one packet per aircraft per frame. The average generation rate is provided to the optimization algorithms. Due to the random nature of the packet generation process, packets may need to be queued at intermediate nodes, leading to additional delay. Due to the finite queue lengths, packets may also be lost.

Parameters specific to the Genetic Algorithm are shown in Table 7.2. With Genetic Algorithms, a good deal of experimentation is typically required until the set of parameters providing the best results has been found, and the best parameters may vary according to the network parameters. Here, we have used one set of parameters providing reasonably good performance for all static GA runs and another set for all mobile GA runs.

The GA tracking algorithm and the hop count based routing and gateway allocation methods were run in parallel to the movement of the aircraft. In addition, for each of the static snapshots, the weighted hop count and delay minimization problem according to Section 6.5 and the static GA according to 7.2 have been executed. The GA tracking should not result in worse performance than the static GA algorithm, as this would be a sign that the population is not able to adapt properly to the changes in the network topology.

Behavior of the Optimization Algorithms

The convergence behavior of the GA in step six of the small scale test topology is shown in Fig. 7.5. Fig. 7.5 shows how the average cost (in terms of delay) of the entire population develops over the course of 60 iterations of the GA, as well as the cost of the fittest individual in the population. Several observations can be made from this figure: Whereas the cost of the fittest individual decreases steadily, the average cost of the population can also increase, especially at the beginning of the run. The cost of the fittest individual decreases in discrete steps, caused by discrete changes in the routing or scheduling. In most iterations, the cost of the fittest individual does not change at all. This is important for the convergence criterion of the GA. Obviously, halting the algorithm when the change in the cost of the fittest individual from one generation to the next is less than some threshold is not a reliable measure of convergence. Rather, we observe that the change in the average fitness could be considered, or the improvement of the maximum fitness over a larger number of generations. Finally, it can be seen that after approx. 53 generations, the population has become so homogeneous that the average fitness is practically equal to the fitness of the fittest individual. The behavior of the entire population is shown in Fig. 7.6. Fig. 7.6(a) shows the distribution of the population's fitness values after 20, 40, and 60 iterations. Note how the distribution becomes narrower and shifts towards lower delay as the number of iterations increases. The cumulative distribution is shown in Fig. 7.6(b). The steep slope of the cumulative distribution after 50 generations confirms the convergence of the overall population after about 53 generations, which was observed in Fig. 7.5. It can be seen that the cumulative distribution does not saturate at the value one, as might have been expected. This is due to individuals representing unfeasible solutions, which have not been included in this plot.

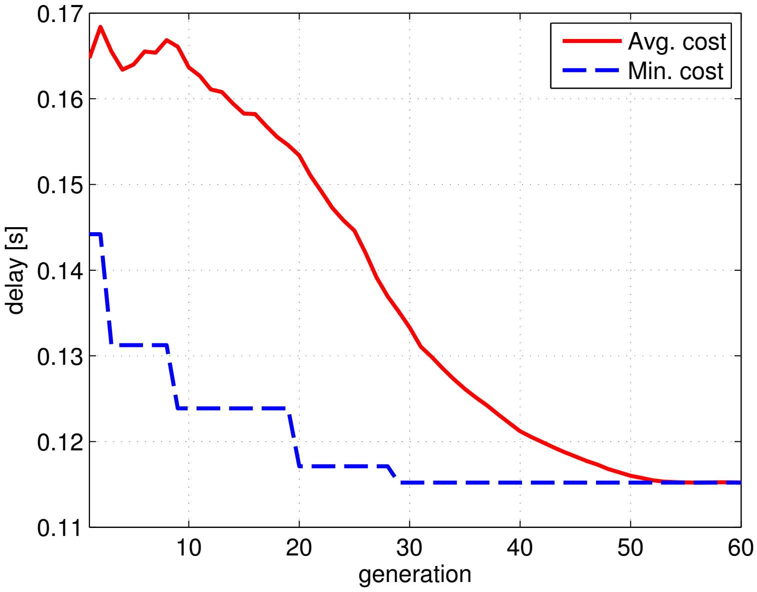
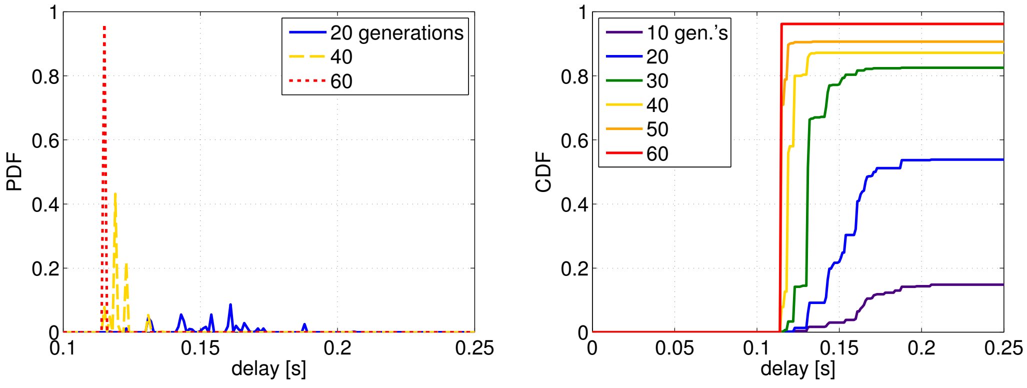


Figure 7.5.: Average and minimum cost of all genomes in the population for a run of the GA on step 6 of the small scale topology.



(a)

(b)

Figure 7.6.: Distribution (left) and cumulative distribution (right) of the population's cost values after different number of iterations.

A comparison of the computational complexity of the mWHC/mAFD approach and the GA is shown in Fig. 7.7. The number of solver iterations required by Lingo to solve the mWHC and mAFD problems, as well as the number of generations of the GA until convergence are shown for each of the six snapshots of the small scale topology. For the GA, the convergence criterion defined in Section 7.2.7 was used. Lingo was run using the default parameters. It can be seen that the complexity of the mathematical programming approach increases dramatically with the problem dimensions. However, the number of iterations depends not only on the size, but also on the "hardness" of the problem. In step 6, the problem is larger than in step 5, since more links and an additional gateway have been added. But do to the additional connectivity of the second terrestrial gateway, the problem has become easier to solve, and Lingo requires fewer iterations than in step 5. Note that the mAFD scheduling problem is much more difficult to solve than the mWHC routing problem, which contents itself with finding a feasible schedule. Compared to the mathematical optimization, the number of generations required by the GA increase much more moderately. However, one weakness of the GA is that it converges prematurely in step 5, hence the low number of generations, because the problem has become too difficult. The resulting degradation in performance with respect to the mWHC/mAFD solution can be seen e.g. in 7.9(b), which shows the average delay, and thus reflects the cost function of the GA. The performance of the GA could be increased here by a larger population size or by a stricter convergence criterion.

Fig. 7.8 shows the value of the cost function of the mAFD step after solving the mWHC step with different link weights for the link from the satellite gateway to the ground network. The weights of all other links are kept at one. It can be seen that the cost, corresponding to the average flow delay, decreases as the satellite link weight increases and traffic is shifted away from the satellite as. Setting w_{sat} to 5 does not result in a further reduction of the cost.

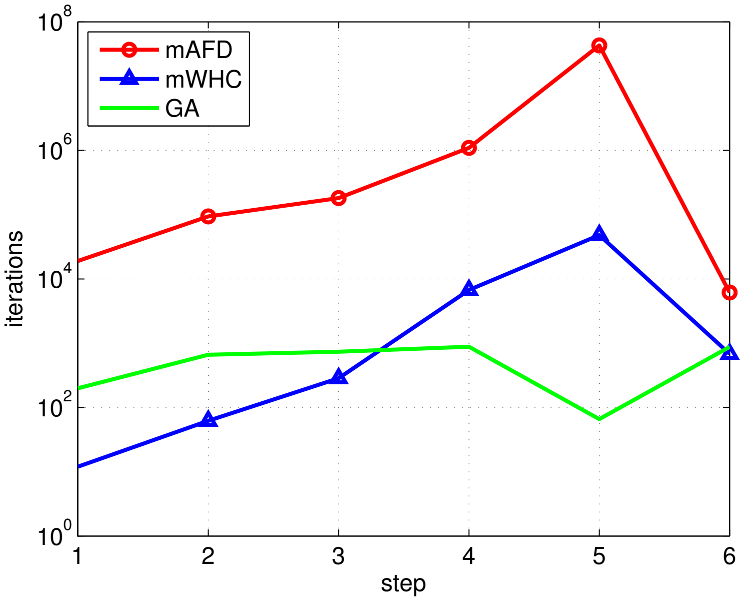


Figure 7.7.: Number of solver iterations (mWHC/mAFD) and GA generations required for each of the six small scale topology snapshots.

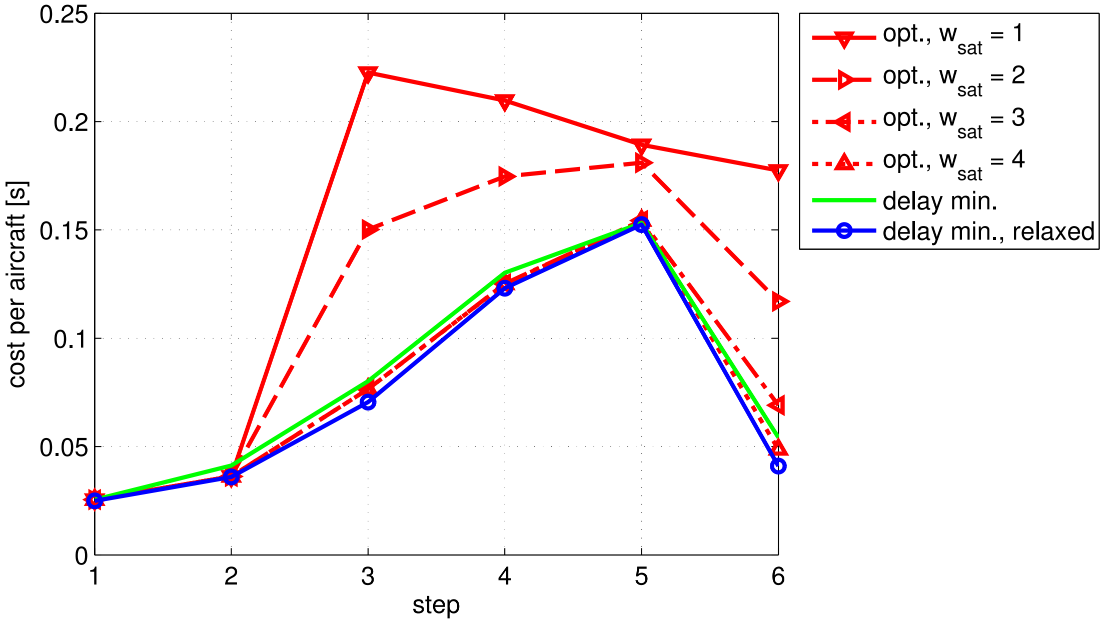


Figure 7.8.: Cost per aircraft after optimization for each of the six small scale topology snapshots.

In addition, the cost resulting from trying to solve the original problem in Eq. 6.9 – Eq. 6.15 directly, i.e. in a single step, is also shown. Since this is a nonconvex problem, it is not possible to determine the true global optimum. However, the problem has been solved by Lingo using a nonlinear solver with multiple starting points. Using multiple starting points significantly increases the computational cost, but reduces the risk of converging to a local optimum. Although Lingo also provides a global solver for nonlinear problems, the computational cost is prohibitive even for the smallest problem, step 1, consisting of only three aircraft and one gateway. Here, we show the cost for the direct delay minimization (Eq. 6.9) in case all variables have been relaxed to continuous values between zero and one (blue line), and in case all variables are constrained to be binary (green line). As could be expected, the cost of the relaxed problem is less than the cost of the binary variable problem. Surprisingly, though, the cost for the mAFD problem with $w_{\text{sat}} = 4$ is actually slightly less than the cost when attempting to solve the delay minimization problem directly. This can be attributed to convergence of the direct delay minimization problem to a non-global optimum.

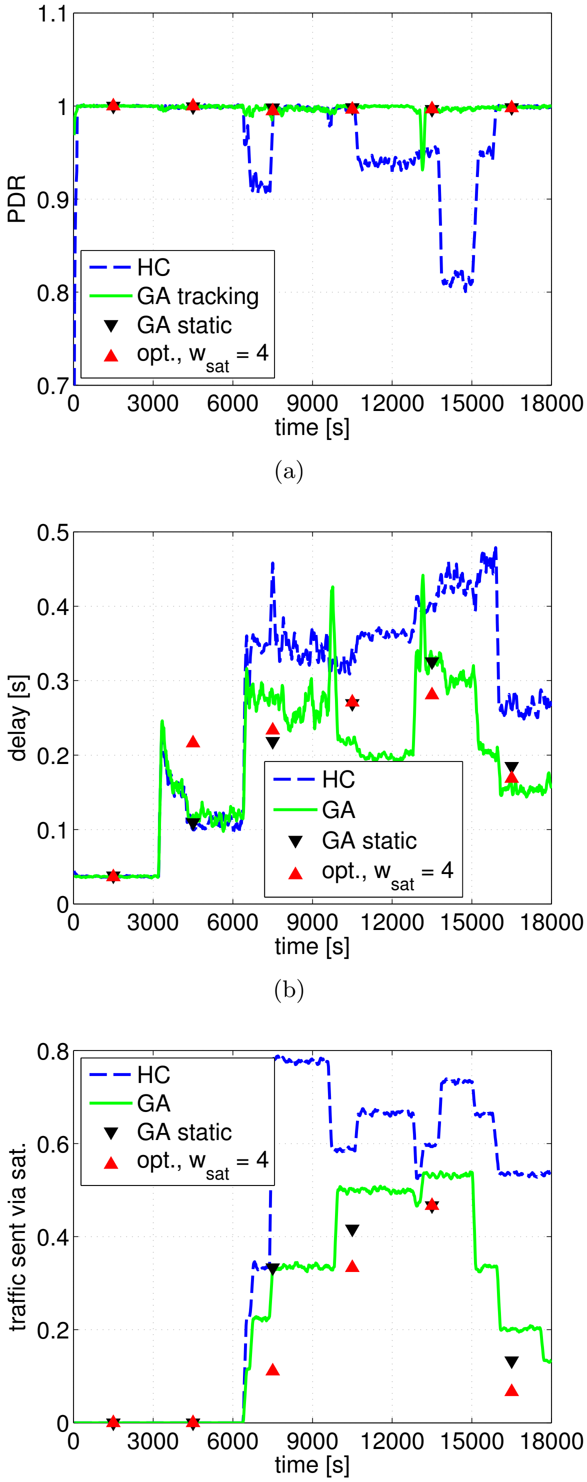
This plot confirms that decomposing the original delay minimization problem into two steps does not lead to a significant performance degradation if the link weight for the satellite links is chosen correctly. For $w_{\text{sat}} = 4$, we are able to outperform the global solver of Lingo, applied to the original problem. This may be due to the nonconvex nature of the original problem. Allowing the nonlinear solver of Lingo to use a larger number of starting points should allow it to find a better solution as well.

However, lower values of w_{sat} lead to significantly higher delay. In larger networks, setting w_{sat} to unnecessarily high values will lead to very long paths being used, potentially resulting in a larger delay than if a satellite gateway at a smaller distance were used instead. We conclude that the performance of the mWHC/mAFD approach is highly sensitive to the choice of w_{sat}.

Network Performance

The routing and scheduling information resulting from the optimization was then used as input for network simulations in order to determine the Packet Delivery Ratio (PDR) and average packet delay achieved by each of these solutions. These results are shown in Fig. 7.9 for a packet generation rate of 12.5 packets per second per aircraft at the ground node, corresponding to one packet per aircraft per frame. The fraction of the aircraft that are using the satellite gateway instead of the terrestrial gateways is also given.

It can be seen that the hop count solution in all steps sends more traffic over the satellite than the other solutions. This is due to the prominent position of the aircraft offering the satellite link in the middle of the ad hoc network. In step 3, when the satellite gateway has just been created, it practically "eclipses" the terrestrial gateway at the left side, since only one aircraft is closer to the terrestrial gateway than to the satellite gateway. The GA performs nearly as well as the mathematical programming solution. The hop count based solution loses almost 20% of the packets just before the cloud of aircraft establishes a link to the gateway on the right side, whereas the GA and

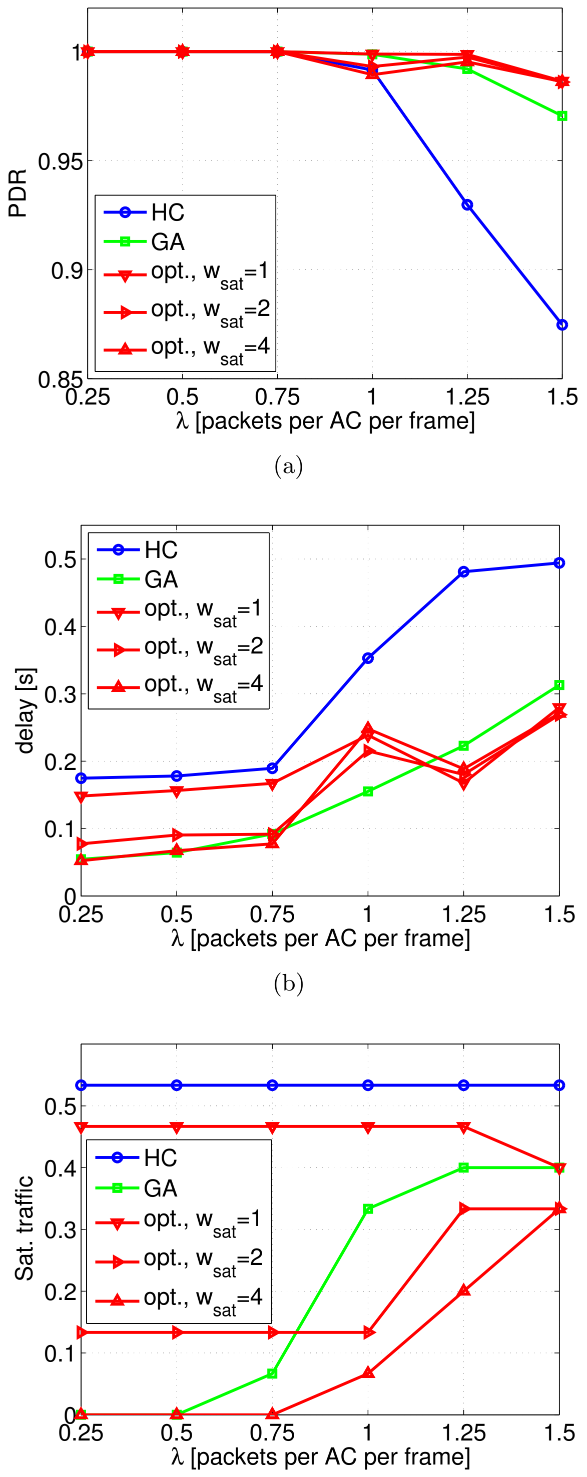


(a)

(b)

(c)

Figure 7.9.: PDR, average delay, and percentage of traffic sent over satellite as a function of time for the small scale test topology.



(a)

(b)

(c)

Figure 7.10.: PDR, average delay, and percentage of traffic sent over satellite for increasing values of packet generation rate λ in Step 6 of the small scale sample topology.

mWHC/mAFD solutions deliver practically all packets over the entire simulation. The average packet delay of the tracking GA is comparable to the static GA and mWHC/-mAFD solutions, and significantly lower than the delay of the hop count solution. It is important to note in Fig. 7.9(c) that, when a connection is made to the right hand gateway, the GA solution succeeds in switching most traffic away from the satellite link. This shows that the population is still able to react efficiently to significant changes in the topology.

In Fig. 7.10, a closer look is taken at the network performance in a static case. Here, we consider the network of 15 aircraft just before a connection is made to the right hand terrestrial gateway, for varying values of the packet generation rate. Again, the PDR, average packet delay, and the fraction of traffic sent over the satellite gateway, are considered. The mWHC/mAFD optimization is performed with several different values for the weight of the satellite link. In general, higher values of w will lead the algorithm to avoid the satellite link longer. In our small network, a value of $w = 4$ would be sufficient to force all traffic to go over the terrestrial gateway. However, this is prevented by the interference and capacity constraints. Thus, $w = 4$ shows us the minimum amount of traffic that must be sent over the satellite in order for the solution to remain feasible. It is interesting to note that until $\lambda = 1$ packet per aircraft per frame, the routing solution found by mWHC/mAFD is not affected by the interference and capacity constraints. However, at $\lambda = 1$, the network has become quite congested, packet delay is increasing, and some packets are being lost due to random fluctuations in the queue lengths. In the next step, the mWHC/mAFD solution has adapted its routing to this congestion, actually leading to less packet delay and packet loss, although the amount of traffic has increased. This behavior is caused by the two-step nature of the approach, where the first step only tries to minimize the weighted hop count, restricted only in that its solution must be feasible. Here, the GA has an advantage, since it attempts to minimize the packet delay directly, and gracefully shifts traffic from the terrestrial gateway to the satellite gateway as the load increases. In our network, the best performance is achieved by the mWHC/mAFD with $w = 4$. However, it would be difficult to predict the optimum value for this parameter for arbitrary topologies.

These results confirm that the GA approach in static networks, as well as the tracking of mobile wireless networks with a GA on the fly, are useful tools in network optimization, since the resulting performance is very similar to what can be achieved by classical mathematical programming methods.

Sensitivity Analysis

In the previous section, it was shown that the centralized solutions outperform the hop count based approach to routing and gateway selection. However, both the mWHC/-mAFD and GA optimizations require a significant amount of knowledge about the network, such as the positions of all nodes, their traffic demands, and their antenna characteristics. In this section, we take a look at the performance of the solutions in case the assumptions made about these properties before running the optimization are violated when the solution is applied to the network. In other words, the information that was available to the optimization problems was not fully correct, and there is a mismatch between the assumed and the real demands. First, we look at the impact of traffic demands that are lower or higher than what was originally assumed on the network performance. In practice, the future traffic load can never be predicted completely reliably, and traffic loads that are higher than anticipated might lead to packet loss.

The dependency of the PDR is shown in Fig. 7.11 for the three different algorithms in four different traffic load situations: the nominal traffic load for which the GA and mWHC/mAFD optimization was performed, as well as 20% less than then nominal load, 20% more, and 50% more. From Fig. 7.11, it can be seen that all three algorithms are quite sensitive to variations in the traffic load. The HC solution was already losing packets at the nominal load, and the PDR naturally decreases further as the load is increased. Surprisingly, the reduction of the traffic load by 20% does not result in a higher PDR. The GA and mWHC/mAFD solutions were able to deliver all packets at the nominal rate and begin to lose packets when the rate is increased. It appears that the GA solution is slightly more robust. It is important to note that feasible solutions exist for a 50% increase in traffic in all steps except step 4 and step 5. Therefore, the lost traffic in all other steps can be attributed to the mismatch between the assumptions made for the optimization and the actual traffic load.

Next, we analyze the effect of incorrect information about the aircrafts' positions on the network performance. For this purpose, a random offset is added to the position of each aircraft when simulating the performance of the precomputed routing, gateway allocation, and scheduling solutions. Small variations will lead to changes in the interference experienced by links. Some transmissions can potentially no longer be scheduled, since the interference is higher than what was assumed during the optimization phase. On the other hand, some links may profit from lower interference. Larger variations can even lead to links breaking up, potentially rendering a precalculated routing solution relying on that link useless. Other links might be created but cannot be used.

Fig. 7.12 shows the effect of inaccuracies in the position information on the PDR achieved by precalculated routing and gateway allocation solutions. After the optimization, a random offset is added to the position of each aircraft. This offset is assumed to be normally distributed with zero mean and variance σ^2_{pos}. For each value of σ_{pos}, 50 different topologies have been simulated.

For comparison, the hop count scheme is also applied to each of the 50 randomly modified topologies. The average PDR of this hop count solution is denoted 'HC' in the legend. The precomputed, static HC solution is denoted 'HC st.' The fact that the PDR of the HC solution for the modified network topology is always close to one demonstrates that a feasible solution still exists, despite the random changes in the topology. Therefore, the degradation in PDR of the three precalculated solutions can be attributed to the inaccuracies in the position information.

The three precalculated solutions react differently to position inaccuracies. Since the paths through the network are shorter with hop count based routing and gateway selection, the probability that a link along the path will break is lower than for the mWHC/mAFD and GA solutions. This makes the HC solution more robust to changes

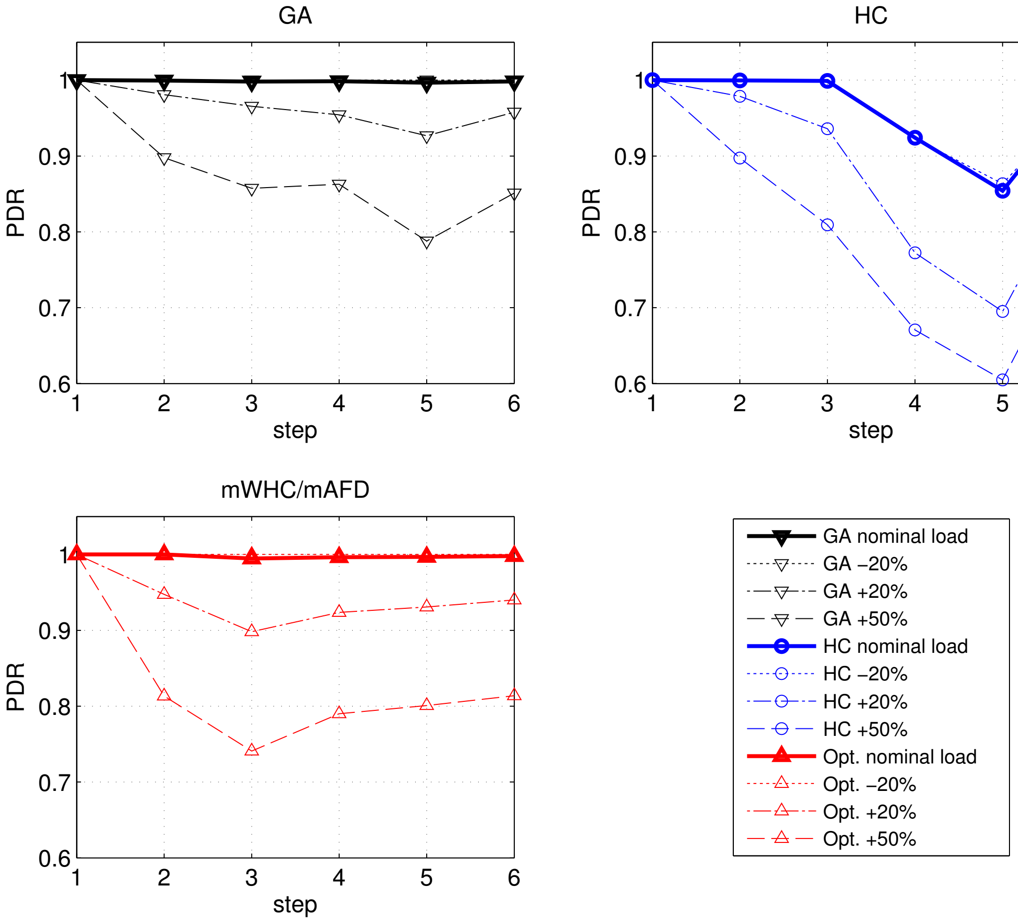


Figure 7.11.: Dependency of the PDR on the traffic load.

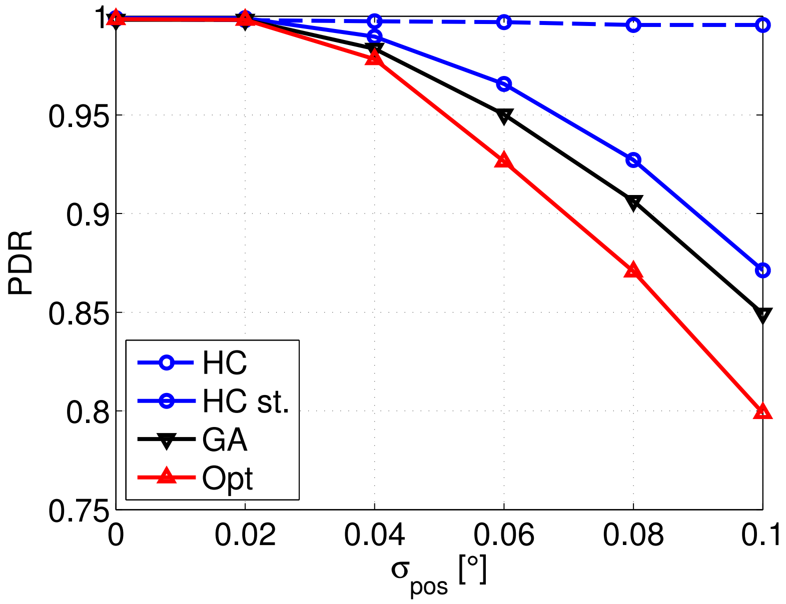


Figure 7.12.: Dependency of the PDR on the standard deviation of the aircraft position inaccuracy in step 6 of the small scale test topology.

in the topology and results in a higher PDR than the GA and mWHC/mAFD solutions.

The standard deviation of 0.1° latitude translates to approx. 12 km. This is less than the Required Navigation Performance (RNP) on the North Atlantic Tracks of 10 nautical miles (18.52 km). Thus, navigational errors that are well within the maximum allowed range already appear to create significant problems for any precomputed routing and scheduling scheme.

Fig. 7.13 shows the dependency of the GA's convergence behavior on the population size. Here, the population size is varied from 100 to 1000 individuals. The pool size of the tournament selection is also varied such that the ratio of pool size to population size is always approximately one third. Fig. 7.13(a) shows the cost of the fittest individual when the algorithm converges. As could be expected, a higher population size results in better performance. However, the improvement flattens out at a population size of approximately 500 individuals. It is worthwhile to compare this to the effort spent in terms of computation time, shown in Fig. 7.13(b).[1] The computation time increases significantly with the population size. Therefore, a reasonable tradeoff must be found between performance and efficiency. The number of iterations required by the GA until convergence is not shown. However, the GA required around 550 iterations (+/- approx.

[1] The processing time has been measured on a notebook running Microsoft Windows XP using the clock() function of C/C++. This function does not return the CPU time of a process, but the total elapsed time. Therefore, the results may be influenced by other processes running on the computer, although these were kept to a minimum. Obviously, the absolute elapsed time would also be different on any other hardware configuration.

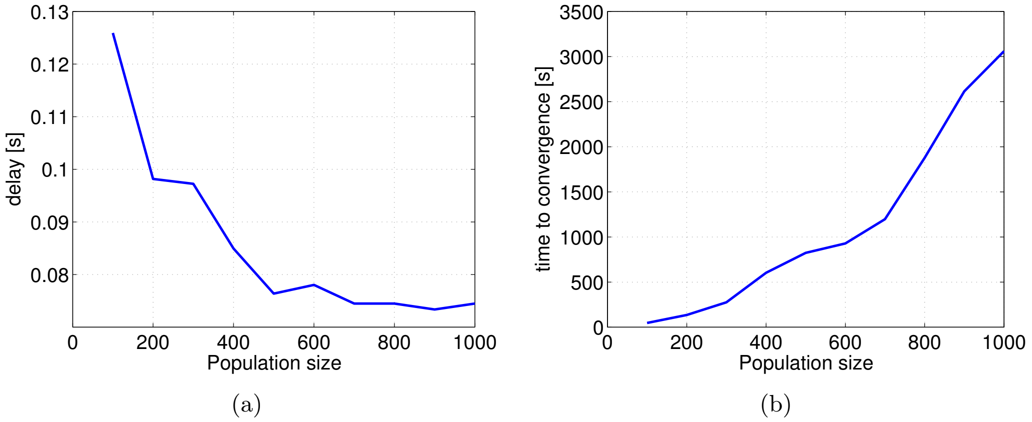


Figure 7.13.: Dependency of the GA on the population size: cost at convergence (left) and time elapsed until convergence (right).

100), regardless of the population size. The algorithm requires more time to converge as the population size is increased because each iteration requires more time. A direct dependency of the number of iterations required until convergence on the population size is not apparent.

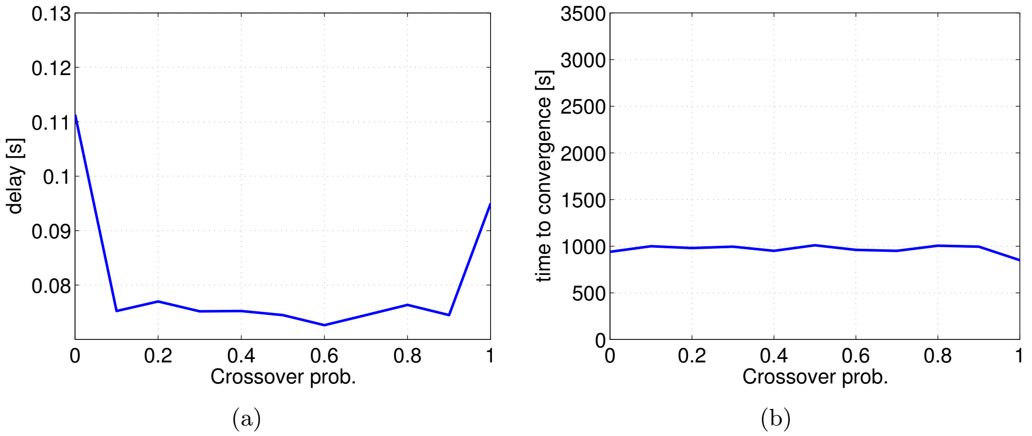


Figure 7.14.: Dependency of the GA on the recombination probability: cost at convergence (left) and time elapsed until convergence (right).

Fig. 7.14 shows the dependency of the GA's convergence behavior on the recombination, or crossover, probability when the population size is held constant at 600 individuals. Again, the cost of the fittest individual at convergence, as well as the time elapsed until convergence are shown. Interestingly, the GA is relatively insensitive to changes in the recombination probability, as long as the probability is neither zero (meaning that recombination does not take place at all) nor one (meaning that recombination always takes place). In these two cases, the solution provided by the GA is significantly worse than for all other values of the recombination probability. Also, the recombination step apparently only accounts for a small amount of the total computational effort. As can be seen in Fig. 7.14(b), the total time required does not depend on the recombination probability.

		mean [s]	$\sigma[s]$
initial	**avg. cost**	0.1756	$8.80 \cdot 10^{-3}$
	min. cost	0.1349	$12.49 \cdot 10^{-3}$
final	**avg. cost**	0.0801	$6.14 \cdot 10^{-3}$
	min. cost	0.0800	$6.16 \cdot 10^{-3}$

Table 7.3.: Dependency of GA convergence on initial population.

Table 7.3 shows the dependency of the GA on the initial population, again for a population size of 600. For this purpose, the GA was executed 20 times, with a different initialization of the population each time. The table shows the mean and standard deviation across these 20 runs of the cost of the fittest individual and the average cost of the entire population, both after initialization and after convergence. On average, the relative improvement from the fittest individual of the initial population to the fittest individual at convergence is 40.2% with a standard deviation of 7.0%.

7.5.2. Large Scale Scenario

In the previous section, the performance of the GA was assessed in a small scale test topology, allowing a comparison with the mWHC/mAFD programming approach. Although the simulation results are promising for the GA, it is important to know how these results scale to larger networks. Therefore, we present further simulation results of a larger, more realistic network topology in the section below. This scenario is no longer tractable for the mWHC/mAFD approach, so no results for the mathematical programming approach are presented in this subsection. The results of the GA are compared only to the HC scheme.

Simulation Scenario

In this section, the performance of the GA approach is assessed by means of simulations intended to reflect the aeronautical environment in the North Atlantic that we are considering. We consider a rectangular area 3000 by 440 km in size. Terrestrial Internet gateways are placed in each of the four corners of this field. A fifth terrestrial gateway is placed at the middle of the Northern edge of this rectangle, corresponding to a gateway in Greenland or Iceland. Again, we consider a dynamic topology, with aircraft being created at the left side of the field and being deleted when they have reached the right side. The generation rate has been modeled according to the actual rate of aircraft entering the North Atlantic corridor in a twelve hour time period. Information about the actual air traffic characteristics has been extracted from a flight database of scheduled airline flights that was also used for the analysis in 4.2. At the beginning and end of the cloud, the generation rate is lower than during the peak period. The generation rate in the simulations was set to one fourth of the true rate derived from the schedule, since not all aircraft or airlines may participate in such an ad hoc network. The variation over time of the number of aircraft in our simulated network resulting from this model is shown in Fig. 7.15. Whenever an aircraft is created, it is chosen with probability 0.5 to be equipped with a satellite link and be able to act as Internet gateway for other aircraft.

In the North Atlantic, aircraft typically fly on predefined tracks, the so called North Atlantic Tracks [10], that ensure separation between aircraft in the uncontrolled airspace. To model these tracks, aircraft in our simulation are placed on a system of four parallel tracks, each separated by one degree of latitude and with a minimum spacing of 145 km between two aircraft on the same track. Again, these values have been chosen to fit the North Atlantic scenario. A screenshot of a typical network topology is shown in Fig. 7.16. The transmission ranges of the terrestrial gateways are indicated by blue circles.

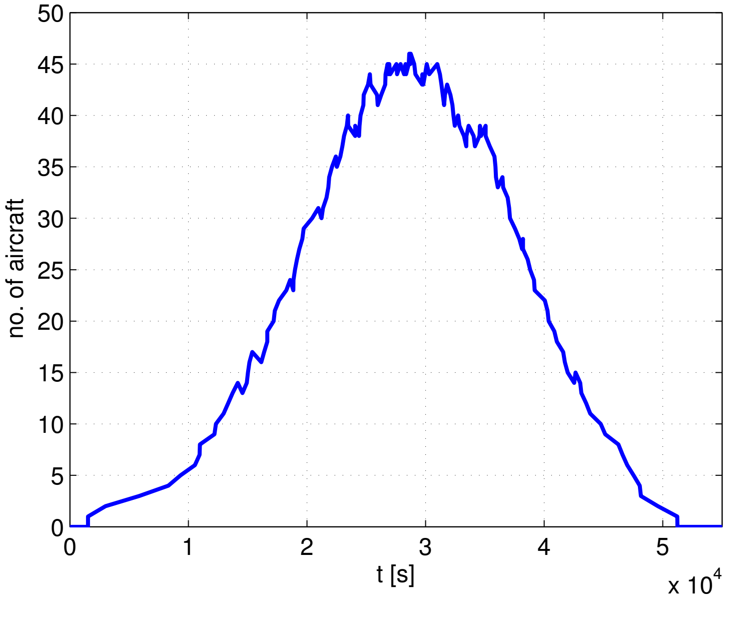


Figure 7.15.: Number of aircraft in the network over time.

A summary of the key network parameters used for the simulations is given in Table 7.4, a summary of the parameters used for the GA, especially the probabilities with which each of the GA operators is invoked, is given in Table 7.5.

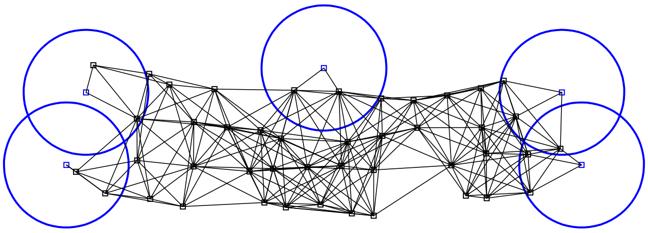


Figure 7.16.: Example topology at time t=30000 s.

Parameter	Value
Playground size	3000 × 440 km
No. terrestrial gateways	5
Probabability of acting as sat. GW	0.5
Max. range d_{horizon}	824 km
Min. required SINR γ_0	10 dB
TDMA slot duration	10 ms
TDMA frame length	80 slots
Traffic asymmetry (DS:US)	4:1
Traffic load (high)	4 packets per frame per node DS 1 packet per frame per node US
Traffic load (low)	2 packets per frame per node DS 0.5 packets per frame per node US

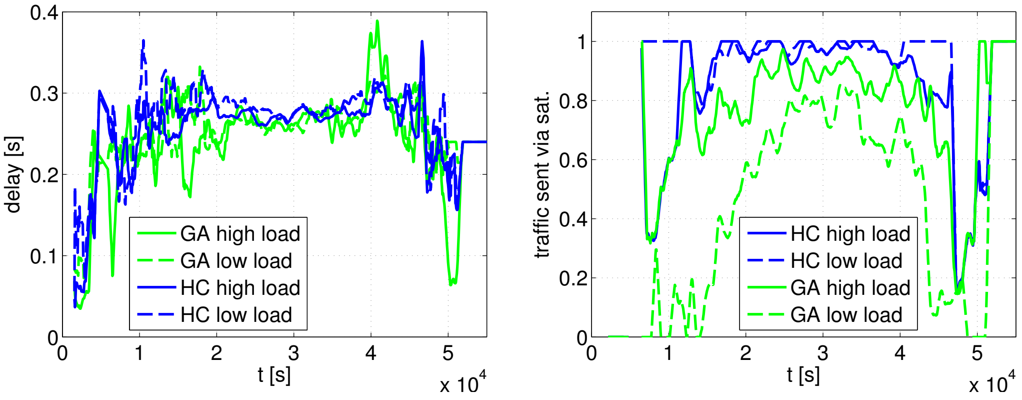
Table 7.4.: Summary of network parameters used for large scale scenario.

Parameter	Value
Population size	300
Selection mechanism	Tournament selection
Pool size	140
p(slot insertion)	0.02
p(slot removal)	0.006
p(slot exchange)	0.1
p(node insertion)	0.06
p(node removal)	0.1
p(node exchange)	0.1
p(path exchange)	0.2
p(crossover)	0.5

Table 7.5.: Summary of GA parameters used for large scale scenario.

Simulation Results

As in the small scale scenario, the hop count based gateway selection and routing again serves as a reference. Simulations have been performed with a high traffic load (4 downstream packets per aircraft per frame) and a low traffic load (2 downstream packets per aircraft per frame). Simulation results of the average packet delay and the fraction of the overall traffic that is sent via a satellite link rather than over an air/ground link are shown in Fig. 7.17(a) and 7.17(b). The packet loss rate in this scenario is negligible due to the high number of satellite gateways. It can be seen that neither the traffic load nor the gateway selection and routing scheme has a large effect on the average delay. However, the true value of the GA approach here is to increase the utilization of the terrestrial gateways, as seen in 7.17(b). Both the GA and the hop count approach rely less on satellite gateways at the beginning and end of the simulation, when the cloud of aircraft is close to one side of the simulated area, but the GA is able to reduce the amount of traffic sent over satellites considerable over the whole duration of the simulation. This is true especially for low traffic load, when there is more room in the network to send traffic over longer paths to one of the terrestrial gateways.

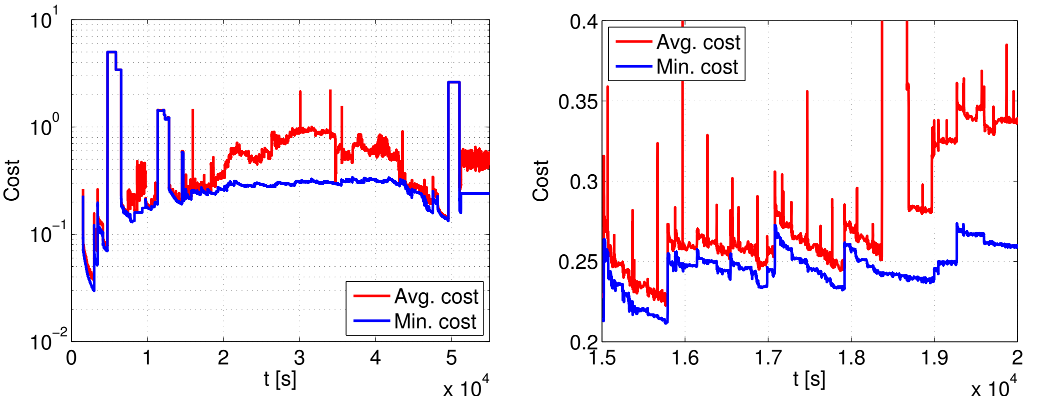


(a) Average packet delay.

(b) Fraction of traffic sent via satellite.

Figure 7.17.: Performance of GA and HC approach in large scale scenario.

Fig. 7.18 shows the behavior of the GA over the duration of the simulation. Due to the high mutation rate, the average fitness is always quite far away from the fittest individual. This effect is desired, since it implies that the population does not lose its diversity, which is necessary to react to changes in the network. Subfig. 7.18(b) shows a smaller time window of 5,000 s. Here, it can be seen how changes in the topology lead to a sharp increase in the GA cost. But then, the GA again converges to a lower cost that is similar to the value before the sudden increase. This indicates that the GA is able to track the dynamic topology successfully. Note in Subfig. 7.18(a) that there are three situations in the simulation where the cost function is exceptionally high (around approx. 6,000 s, 12,000 s, and 50,000 s). Here, the network is only weakly connected, and the GA is unable to fulfill all flows' demand targets. Although the GA is aware of the flows' demands, it is unable to find a feasible solution satisfying these demands. Therefore, the penalty term in the genome cost function (Eq. 7.3) dominates the delay term. At all other times in the simulation, the demands are fulfilled, and the cost function directly reflects the packet delay.



(a) Cost over the duration of the entire simulation.

(b) Closer view of the time window from 15,000 s to 20,000 s.

Figure 7.18.: Cost of the GA in the large scale mobile network.

7.6. Conclusion

In this chapter, we have presented a Genetic Algorithm approach to the joint routing, gateway selection and scheduling problem. The performance of the proposed algorithm was compared to the performance of the mWHC/mAFD optimization in static networks, and the GA was extended to mobile networks. This allows the GA to run in real time, reacting to changes in the network topology by dynamically rerouting traffic flows. Due to the quasi-static nature of the air traffic in the North Atlantic, the GA is able to converge faster than the topology changes.

If a centralized network optimization approach using the proposed GA is to be implemented in reality, it would require the relevant network parameters, such as the aircraft positions and traffic demands to be collected and continuously supplied to the entity running the GA. Likewise, the output of the GA would constantly need to be distributed to the aircraft. In principle, this could be efficiently accomplished by a satellite link, due to the ability to broadcast information within a large region. This additional control traffic can be expected to be much less than the large amount of user data transmitted through the ad hoc network.

However, the main weakness of the GA is its dependency on a central entity. We have seen that inaccuracies in the information which is used as input to the optimization, such as the aircraft positions, can result in a significant degradation in performance. Therefore, the following chapters will focus on a fully distributed solution that does not require such a central, omniscient optimization entity.

8. Minimum Downstream Delay Routing

The routing and gateway selection solutions that we proposed in Chapter 6 and Chapter 7 both rely on a central entity to perform the optimization task. This entity requires complete knowledge of the network topology, the nodes' traffic demands, etc. in order to carry out this optimization. To be useful, this information needs to be accurate and up to date. In a real network, this is obviously a significant limitation. Even if such information can be gathered or estimated, the resulting routing, gateway selection and scheduling assignments need to be provided to the network nodes after the optimization has been performed. While such an approach is certainly useful in order to determine the performance that can be provided by the network, it is clearly difficult to implement in practice. Therefore, we propose a distributed algorithm in this chapter that runs locally on all nodes and is much better suited to real life implementation. We assume that the task of link scheduling is performed by the scheduling algorithm that was proposed by Grönkvist in [118] and which was summarized here in Section 5.2.2, which also runs in a distributed manner. The performance of the proposed distributed algorithm will be compared to the performance of the centralized Genetic Algorithm approach which was defined in Chapter 7, for both static and mobile networks.

At the end of Chapter 3, a functional routing architecture for the AAHN was defined, based on IPv6 and relying on a sub-IP routing protocol. However, the details of the gateway selection and routing protocol were left open. In this chapter, we define the Minimum Downstream Delay (MDD) routing and gateway selection protocol for use in this functional architecture. The reasons for choosing this name will become clear in the following discussion. In general, any definition of a routing protocol must address the following three questions:

1. How is the routing information disseminated through the network?
2. What metric is used in the calculation of paths?
3. Given the information about the network from the previous two steps, how does a node determine the best paths to use, i.e. what is the actual routing *algorithm*?

Each of these three aspects will be discussed for our proposed solution in the following sections.

8.1. Routing Information Dissemination

Due to the large number of users per aircraft, a large amount of traffic is created and consumed by each aircraft. If the aircraft's traffic flow is routed through an Internet gateway with unsatisfactory quality of service, a large number of users will become dissatisfied. Clearly, this situation is to be avoided. Also, the Mobile Router should be able to change its gateway as soon as it becomes aware of a more suitable one. Therefore, all aircraft will need to constantly determine the most suitable Internet gateways and paths for their traffic flows. In such a situation, a proactive routing protocol is more efficient in terms of signaling overhead than a reactive one [52]. The Optimized Link State Routing protocol (OLSR) [95] is a successful proactive ad hoc routing protocol that has been developed by the IETF MANET WG and is currently being updated to OLSRv2 [96]. Much of the signaling used by OLSRv2 has been defined in separate documents, allowing for easy reuse, modifications, and extensions. In this paper, we will make use of the signaling framework of OLSRv2. However, we define some extensions to control messages, define our own metric for purposes of routing and gateway selection, and propose a significantly different algorithm that combines the two tasks of routing and gateway selection. The proposed gateway selection algorithm allows different Classes of Service to be assigned to different gateways, based on their QoS requirements.

In OLSRv2, each MANET router periodically broadcasts HELLO messages in order to build up a database of neighboring nodes. These messages are defined in the MANET Neighborhood Discovery Protocol (NHDP) [147]. HELLO messages contain a list of the sender's neighbors, as determined from the HELLO messages that the node has received from its neighbors. Thus, every node is informed of all other nodes within its 2-hop neighborhood. In our network, this information is also required by the TDMA link scheduling algorithm. Additional fields can easily be added to HELLO messages according to a Type-Length-Value (TLV) structure. We extend the HELLO messages by a TLV for the link priority, which is required by the scheduling algorithm to decide which link in a 2-hop neighborhood will be allowed to allocate itself a slot at the next assignment opportunity. In addition, each node transmits its current TDMA schedule, i.e. in which slots it is transmitting to or receiving from which other node. A TLV for this schedule is also defined. Thus, HELLO messages can be efficiently reused by the scheduling algorithm as well. The TLV structures that are defined for HELLO messages for the operation of our proposed protocol are listed below. The exact format of the messages used by the proposed routing algorithm, and their respective message sizes, are defined in Appendix A.

- LINK_PRIO: This TLV structure carries the link priorities, which are required by the scheduling algorithm.

- SCHED_COMP: This structure carries the complete schedule information known to a node. For every slot in the frame, all links (transmitter and receiver) in the 2-hop neighborhood that are assigned to this slot are encoded. This information must be exchanged periodically in order to ensure that all nodes have a consistent view of the schedule.

- SCHED_INCOMP: This structure only carries information about those entries in a node's schedule in which the node generating the HELLO message is either the transmitter or the receiver. This incomplete schedule information is exchanged more frequently than the complete information.

- POSITION: This structure notifies the node's neighbors of the node's position. This information is needed by the scheduling algorithm.
- NEIGHBOR_POS: This structure carries the positions of the node's neighbors. The positions of all nodes in the 2-hop neighborhood is needed by the scheduling algorithm.

In addition to HELLO messages, which are not forwarded through the network, each OLSRv2 node also periodically creates Topology Control (TC) messages, that contain the link metrics of all outgoing links of the generating node. These messages are flooded through the network, allowing each node to build a database of the complete network topology. The information in this database is used by a link state routing algorithm to calculate the best paths through the network. TC messages can also be created on-demand, in response to significant changes in the topology, e.g. link up or link down events. In OLSR, Internet gateways advertise their connectivity to the Internet with an Attached Network field in the TC message. In principle, each gateway can advertise a default route (::/0 in IPv6) that will be selected for traffic whose destination address does not match any other longer advertised prefix. A metric for the link to this external network is also included. We define the following TLV structures for TC messages for the operation of our routing and gateway selection protocol:

- GATEWAY: This structure indicates that a node is acting as an Internet gateway and conveys the metric of the link to the access network.
- AR_ADDR: This structure is only present if the GATEWAY field is present. It carries the IP address of the Access Router. Since the gateway is always an aircraft and the Access Router is always in the ground network, these nodes will not be the same.
- GW_BLOCKINGS: This structure indicates if any gateways have been blocked for traffic of a certain service class. These gateway blockings are described below in Section 8.3, where the gateway selection algorithm is defined.

Overall, the additional overhead due to the delay metric information carried in TC messages is quite low, as is seen in Appendix A. We adopt the basic operation of the OLSRv2 protocol to exchange routing information, including flooding reduction by using Multipoint Relays, for our scheme.

8.2. Delay Metric

Similar to our centralized routing and gateway selection algorithms, we propose to use an estimate of the average packet delay as the link metric for attached networks as well as for the regular wireless links between aircraft. However, the metric that we define here could also be used as a component of any other ad hoc routing protocol that supports link metrics.

Each node constantly monitors the delay of packets in the transmit queue of each of its outgoing links. It timestamps a packet when it is placed in a queue, and calculates the elapsed queueing and transmission delay when the packet is removed from the queue to be sent. The link metric is then defined as the exponentially weighted moving average of the individual packet delay measurements:

$$\hat{\delta}_{n+1} = (1 - \alpha)\hat{\delta}_n + \alpha\tau_n, \tag{8.1}$$

where $\hat{\delta}_n$ is the link metric after transmission of the nth packet, τ_n is the delay of the nth packet, and $\alpha \in [0, 1]$ is a parameter that controls how much weight is given to the new measurement value. Larger values of α will lead to a faster response to changes in the delay, but may cause stability problems for the routing algorithm. As stated above, a node advertises the metrics of its outgoing links to all other nodes in the network by means of the TC messages.

For links carrying no data traffic, such measurements obviously cannot be performed, and a default metric must be defined. According to Grönkvist's scheduling algorithm, a link that has packets waiting in its queue to be sent, but has not been assigned any time slots, has infinite priority, and is thus guaranteed to receive at least one slot. At low traffic load, when queueing delay due to other packets in the transmission queue is negligible, the delay is given by the time that the packet must wait until the next transmission opportunity comes up, plus the duration of the slot required for transmission itself. At one slot per frame, the average link delay according to Eq. 5.4, is

$$\delta = T_s \left(1 + \frac{1}{2L}\right), \tag{8.2}$$

where L is the frame length in slots and T_s is the slot duration in seconds. This value will be used as the default link metric for newly created links that have not yet carried any traffic. Likewise, when a link is no longer used, its metric will periodically be updated with the default metric, so that it slowly returns to the default value. That is, the default metric according to Eq. 8.2 is periodically inserted as τ_n into the link metric update equation Eq. 8.1. This is necessary, since the arrival rate estimate used by the TDMA scheduling will go to zero, and the time slots that have been allocated to the link will eventually be stolen by other links that still are carrying traffic.

Internet gateways must include the delay metric of their connection to the Internet in the GATEWAY field of the TC messages. For example, a satellite gateway would specify the queueing and transmission delay of its satellite transmit queue, plus an additional 250 ms propagation delay. We assume that this information about the propagation delay can be configured statically for satellite gateways. Terrestrial gateways would specify the queueing and transmission delay of the air/ground link.

The use of a delay metric in routing protocols is not a novel idea. A delay metric was temporarily used in the ARPANET, which was the predecessor of today's Internet, in the 1980's [148]. More recently, the Expected Transmission Time (ETT) metric [149] has been proposed for wireless ad hoc networks. The ETT metric relies on small probe packets that are sent over each link in order to measure the packet loss rate of the link. The expected transmission time for a data packet is then calculated from its packet length, the link's nominal data rate, and the measured packet loss rate. In contrast to our proposed metric, ETT does not consider queueing delay at the transmitter, and it assumes that the packet loss rate experienced by the small probe packets accurately predicts the packet loss rate that will be experienced by larger data packets. Our metric does not rely on any such probe packets, but estimates the delay only from information that is available at the transmitter itself – when a packet enters the queue and when it is transmitted. Due to the TDMA based MAC, collisions at the receiver that lead to packet retransmissions are very unlikely. Therefore, the local information alone is sufficient to characterize the link.

8.3. Minimum Downstream Delay Routing and Gateway Selection Algorithm

As stated previously, one goal of our routing and gateway selection algorithm is to be able to assign different Internet gateways to different service classes, based on their QoS requirements. We assume that the QoS requirements are formulated in terms of a target delay that should be met by packets of that service class. In this way, delay sensitive traffic can be sent over a terrestrial gateway, whereas more delay tolerant traffic is sent over a satellite gateway, or another terrestrial gateway with a higher traffic load. We use the delay metric which was defined in the previous section as the gateway selection metric. Since most of the traffic is flowing in the downstream direction, i.e. from the Internet to a user on board an aircraft, we base the gateway selection on the quality of the downstream paths from the various gateways to the aircraft. Hence, we call the proposed algorithm *Minimum Downstream Delay* (MDD) gateway selection. The MDD algorithm is tightly linked to the ad hoc network routing protocol.

The TC messages that are generated by a node carry the delay metrics of the links to all neighbors. If the node is acting as Internet gateway, it will also specify the delay metric of the link to the Access Network. This information is used by the routing function at a node to calculate the expected downstream delay from every known gateway to to the node. This delay includes the delay of the links along the path through the ad hoc network, as well as the delay of the link to the access network, which is especially important in the case of a satellite link.

In principle, it would be possible for every node to always select the gateway with the lowest associated downstream delay. However, to prevent one aircraft from congesting a gateway with delay tolerant traffic when another aircraft would like to transfer delay sensitive traffic over the same gateway, we introduce the concept of gateway *blocking*. An aircraft is allowed to block a gateway for use by certain service classes in order to achieve better performance for all service classes with higher priority. There are two levels of blocking, each associated with a certain CoS. A gateway that is *blocked*1 for

CoS i must not be used by any aircraft for traffic in CoS i or higher[1]. This blocking can reduce the amount of traffic being sent over a certain gateway. On the other hand, a gateway that is $blocked2$ for CoS i may not be assigned any *additional* traffic of CoS i or higher. That is, if gateway k is $blocked2$ for CoS i, an aircraft that is currently not using gateway k for CoS i or higher is not allowed to begin doing so. However, aircraft that are already using gateway k for CoS i or higher may continue to do so. This serves to prevent the load of a certain gateway from increasing further, but does not require that the load be reduced. Also, aircraft may still assign traffic of CoS j, $j < i$ to a gateway that is $blocked2$ for CoS i. If not renewed within a certain predefined time, these blockings time out automatically, and the gateway can again be used for all service classes. Information about the blockings of gateways is carried in the GW_BLOCKINGS field of the TC messages.

Name	**Description**
$GW(j)$	GW assigned to CoS j
$target(j)$	delay target of CoS j
$k.adoptTime(j)$	time at which GW k was adopted for CoS j
$k.blocked1$	CoS for which GW k is $blocked1$
$k.blocked2$	CoS for which GW k is $blocked2$
$k.blocked1Time$	time at which $k.blocked1$ was set
$k.blocked2Time$	time at which $k.blocked2$ was set
$k.cost$	cost of GW k

Table 8.1.: Nomenclature used in the definition of the MDD algorithm.

The proposed gateway selection algorithm is summarized in pseudocode in Algorithm 1. The nomenclature that is used in this description is defined in Table 8.1. This algorithm is invoked whenever a new TC message is received by an aircraft i and an entry in the node's topology database changes, or when node i itself detects a change in the status of one of its outgoing links. The aircraft first calculates the shortest downstream paths from all known gateways to itself. The cost of a gateway is then given by the cumulative delay metric of all links on the path as well as the delay metric offset advertised by the gateway itself. Aircraft i sorts the gateways according to this cost. Now, i assigns a GW to each CoS, beginning with the CoS that has the highest priority, or, equivalently, the lowest delay target. It always tries to assign the gateway with the lowest associated delay metric that is not blocked for this CoS. If the GW that is currently being used for CoS j has been in use for less than the minimum duration $gwHoldTime$, the algorithm continues with the next service class. This prevents frequent fluctuations of gateway allocations, and gives the delay metrics and TDMA schedule sufficient time to adapt to changes in the gateway allocation.

If the current gateway has been $blocked1$ for this service class, it can no longer be

[1] Higher CoS implies lower priority, and higher delay target.

used, and the algorithm sets the current gateway used by CoS j to -1 (i.e., an invalid value). Then, the algorithm loops over all known gateways, beginning with the one with the lowest cost. If the gateway is *blocked*1 for the CoS being considered, it is skipped. If the gateway is *blocked*2, and is currently not in use by node i, it is also skipped. On the other hand, if the gateway is *blocked*2, but already *is* being used by node i, it may continue to use this gateway. The purpose of this blocking level is to prevent the load on a gateway from being increased further. If the gateway is not skipped due to one of these two conditions, it is assigned to CoS j, and the current time current_time is remembered in order to be able to enforce the minimum gateway hold time. Before continuing on to the next CoS, the algorithm checks if the selected gateway fulfills the delay target of the current CoS. If the cost of the gateway is more than α times the delay target (with $0 < \alpha \leq 1$), the gateway is *blocked*1 for all following service classes. If the cost is between α and β times the target (with $0 < \beta < \alpha$), the gateway is *blocked*2 for all following service classes in order to prevent the load from increasing further. If the gateway is blocked, the current time is recorded, allowing the blocking to expire automatically if it is not renewed within *gwBlockTime*. (This is not shown in Algorithm 1.) Since gateway blockings are also communicated to other nodes by Topology Control messages, these blocking values and times can also be changed outside of the MDD algorithm, upon receipt of a TC message from another node. When a gateway has been assigned to a CoS, the algorithm goes directly on to the next CoS without considering the remaining gateways, as these all have a higher cost than the gateway that was just assigned.

Although the gateway selection is done according to the delay metric, it is important to note that the calculation of shortest paths in the MDD algorithm is based on the hop count. There are two reasons for this: First, we have seen in our previous work [150] that when the gateway selection is done according to a delay metric, also optimizing the routing according to this metric brings very little additional gain in terms of delay. Second, the hop count metric is very robust, whereas the delay metric can lead to unstable routing in some situations. We have not observed such stability problems when the delay metric is used for gateway selection only.

With the algorithm as described above and summarized in Algorithm 1, it can occur in some cases that an aircraft does not allocate a gateway for a certain CoS. This can happen if all reachable gateways have already been blocked by other service classes with higher priority, and would result in a service outage for the affected CoS. In general, though, it might be more desirable to always assign a gateway to each service class, even if this means that the more stringent QoS targets cannot be fulfilled. For example, the algorithm could assign all following service classes to the last gateway that could be assigned successfully.

In Algorithm 1, a node always immediately selects a gateway for CoS i if this gateway is the best non-blocked gateway for CoS i, and if the node has been using the current gateway for at least *gatewayHoldTime*. However, the stability of the algorithm can be improved by adding some constraints: A node does not switch immediately, but starts a counter when it realizes that its current gateway is no longer the one with the lowest cost. The counter begins at zero, and every time the MDD algorithm is executed, and this potential new gateway is still at a lower cost than the current gateway, the counter is incremented by one. Only when the counter reaches the value $gwSwitchCounterMax$ does the node actually switch to the new gateway. If any other gateway becomes the best non-blocked gateway for CoS i before $gwSwitchCounterMax$ is reached, $gwSwitchCounterMax$ is reset to zero, and a corresponding counter is started for the new gateway.

8.4. Performance Assessment

To evaluate the performance of the proposed MDD protocol, we again make use of the 15-aircraft test topology that was defined in Section 7.5.1 and perform simulations with two service classes. CoS1 is considered to be delay sensitive and has a delay target of 0.25 s. CoS2 is considered to be best effort traffic and does not have any delay target. The proposed MDD algorithm is compared to the Genetic Algorithm for multiple service classes, which was defined in Section 7.4, the standard hop count approach in which routing and gateway selection are based on hop count (HC), regardless of the service class, as well as a modified hop count approach (mod. HC). In this modified hop count approach, CoS1 is always assigned to a terrestrial Internet gateway, in the hope that this will result in lower delay, and CoS2 is always assigned to a satellite gateway. Finally, gateway selection based only on the delay metric of Section 8.2 is also included in the assessment. Here, a node always selects the gateway with the lowest downstream delay for all service classes. Thus, the major difference to the MDD algorithm is the lack of gateway blockings.

A summary of the relevant simulation parameters is given in Table 8.2. The network parameters, such as frame length, slot duration, etc. are identical to those of the small scale simulations in Section 7.5.1.

Parameter	Value
α	0.75
β	0.5
$gwBlockTime$	60 s
$gwHoldTime$	600 s
$gwSwitchCounterMax$	10
HELLO interval	10 s
TC interval	10 s

Table 8.2.: Simulation parameters in assessment of MDD algorithm.

First, we consider the static network in Step 6, i.e. when the network is connected to both terrestrial gateways, and the satellite gateway is in the middle of the network. Only downstream traffic is simulated, which is split between two service classes. The packet generation rate for CoS1 is kept constant at 1.5625 packets per second per aircraft

Algorithm 1 MDD GW Selection Algorithm

```
Calculate DS paths from all GWs to $i$
Sort all known GWs according to cost
$\Rightarrow$ provides $sortedGWs$

for all CoS $j$ do

  if current_time $- GW(j).adoptTime(j) < gwHoldTime$ then
    // CoS $j$ must use gateway GW($j$) for at least $gwHoldTime$
    continue
  end if
  if $GW(j).blocked1 \leq j$ then
    // This gateway has been $blocked1$ and must no longer be used by CoS $j$
    $GW(j) = -1$
  end if

  // Now try to assign a gateway to CoS $j$
  for all $k \in sortedGWs$ do
    if $k.blocked1 \leq i$ then
      // This gateway is $blocked1$ and must not be used by CoS $j$
      continue
    else if $GW(j).blocked2 \leq j \bigwedge k \neq GW(j)$ then
      // This gateway is $blocked2$ and currently not used by CoS $j$,
      // therefore, it must not be used by CoS $j$
      continue
    end if

    // Assign gateway $k$ to CoS $j$
    $GW(j) = k$
    if $k \neq GW(j)$ then
      // Remember when $i$ starts using gateway $k$ for CoS $j$
      $k.adoptTime(j) =$ current_time
    end if

    if $cost(k) > \alpha \cdot target(j)$ then
      // Cost of gateway is too high: need to reduce traffic
      $k.blocked1 = j + 1$
      $k.blocked1Time =$ current_time
    else if $cost(k) > \beta \cdot target(j)$ then
      // Cost of gateway is relatively high: don't allow further traffic
      $k.blocked2 = j + 1$
      $k.blocked2Time =$ current_time
    end if

    // A gateway has been assigned - proceed to next CoS
    break

  end for

end for
```

(corresponding to a packet every 0.64 s on average). The packet generation rate for CoS2 is increased from the same value of 1.5625 packets per second per aircraft in steps of 0.78125 packets per second up to 10.9375 packets per second per aircraft.

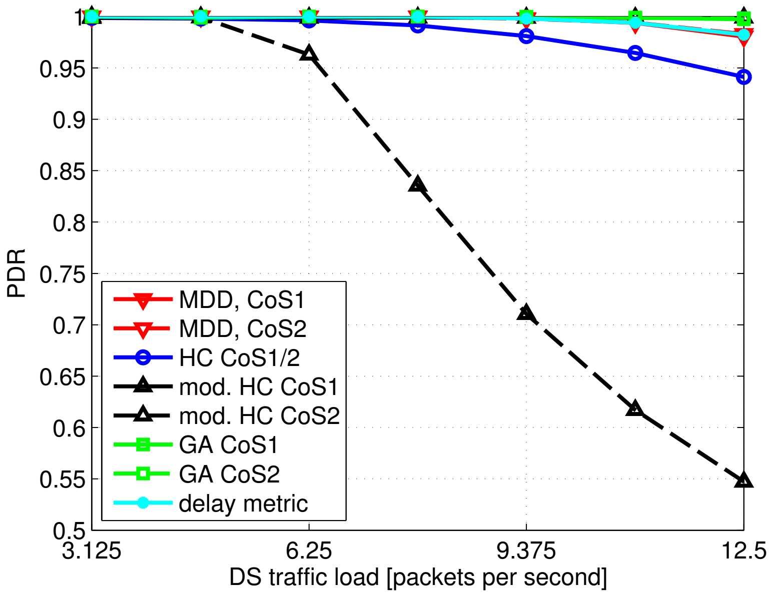


Figure 8.1.: PDR as a function of the traffic load.

The average PDR for both service classes is shown in Fig. 8.1 as a function of the downstream traffic load in packets per aircraft per second; This includes both CoS1 and CoS2. The average packet delay is depicted in Fig. 8.2. It is apparent that the modified hop count scheme with the static allocation of service classes to gateways does not provide satisfactory performance. CoS1 has perfect PDR and very low delay. However, the single satellite gateway is soon completely overloaded with traffic of CoS2, resulting in high packet loss and delay. The simple hop count scheme also does not perform well. At high traffic load, it no longer meets the delay target of CoS1 and begins to drop packets. On the other hand, the Genetic Algorithm has no problems finding a solution that delivers all packets and fulfills the delay target of CoS1 in all situations. The average delay of CoS2 is only slightly higher than that of CoS1. The performance of the proposed MDD algorithm is slightly worse than that of the GA. It loses approx. 2% of the packets at high traffic load (compared to 6% for HC). The delay of CoS1 is kept below the delay of CoS2, but does not meet the delay target at the highest traffic load value. For gateway selection based on the delay metric, the average packet delay and PDR is similar to the values of the MDD algorithm, but identical for both service classes. Here, it can be seen that the gateway blocking functionality of MDD is able to improve the delay of CoS1 at the expense of a higher delay in CoS2.

The PDR and delay results indicate that the MDD solution outperforms all of the other distributed routing and gateway selection schemes. The Genetic Algorithm, which

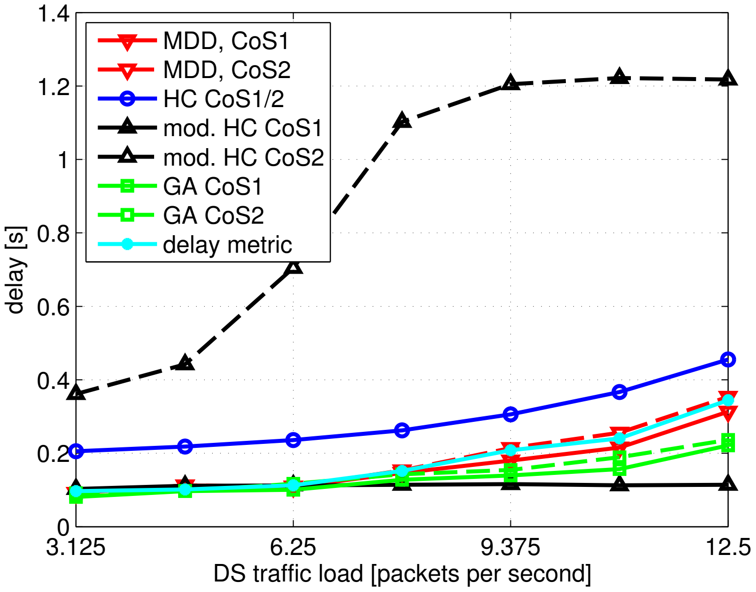


Figure 8.2.: Delay as a function of the traffic load.

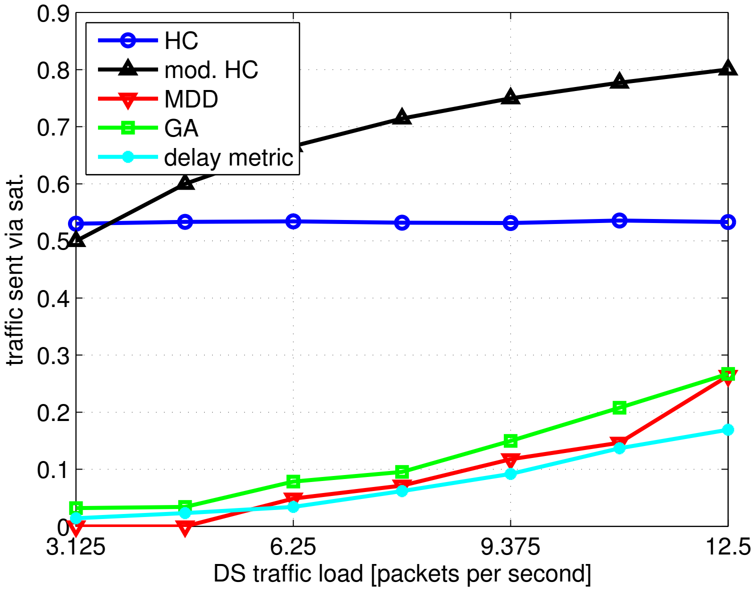


Figure 8.3.: Fraction of traffic sent via satellite as a function of the traffic load.

relies on global knowledge of the network topology and traffic demands is able to perform only slightly better.

The fraction of traffic sent via the satellite gateway is shown in Fig. 8.3. The gateway allocation of the HC scheme remains constant at approx. 53%. As the traffic load in CoS2 increases, so does the fraction of traffic sent over the satellite gateway in the mod. HC scheme. The GA and MDD schemes only slowly shift traffic to the satellite, the MDD algorithm being slightly more reluctant to use the satellite. However, the rate at which the satellite traffic increases is almost the same for these two algorithms. Since the terrestrial gateways are not blocked for CoS2 when the gateway selection is based only on the delay metric, this approach results in less traffic being sent over the satellite gateway than with the MDD algorithm.

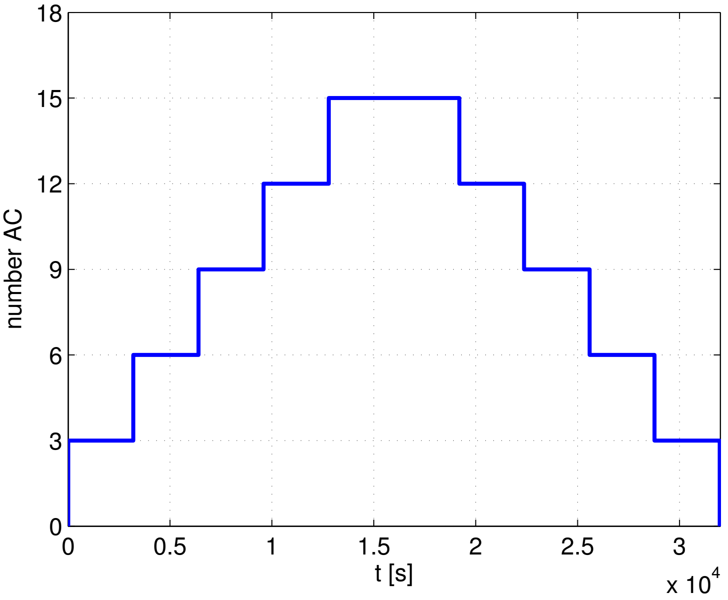
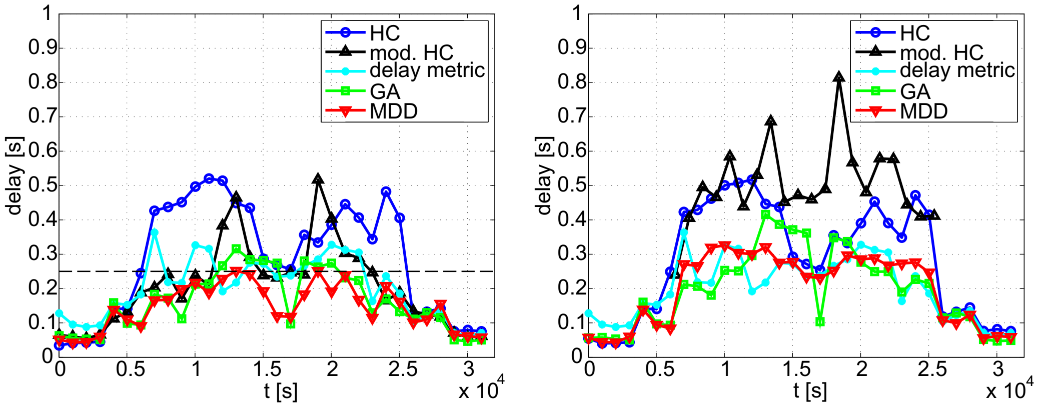


Figure 8.4.: Number of aircraft nodes in the simulated network.

Next, we look at the performance of the MDD algorithm in a mobile network. Again, we consider the small scale scenario defined in Section 7.5.1. The variation of the number of aircraft in the network during the course of the simulation is shown in Fig. 8.4. The parameters of Table 8.2 also apply to these simulations, unless explicitly stated otherwise. The traffic load is kept constant at 3 packets per second per aircraft and service class. The parameters of the GA for this scenario are the same as those for the mobile case in Table 7.2.

The average packet delay is shown in Fig. 8.5. The delay target of 0.25 s is drawn as a dashed black line. The delay of all algorithms exhibits the same general behavior: as the number of aircraft increases and the aircraft move farther from the first terrestrial gateway, the delay increases. After a while, the aircraft reach the second terrestrial gateway, and the average delay decreases. The aircraft continue flying, and when the connection



(a) CoS1.

(b) CoS2.

Figure 8.5.: Delay over time for the different algorithms simulated.

to the first gateway is lost again, the delay again increases. Then, as the aircraft come closer to the second terrestrial gateway and the number of aircraft decreases, the delay also decreases again. Note that the delay of CoS1 in Fig. 8.5(a) and CoS2 in Fig. 8.5(b) is the same for both the HC and delay metric schemes. For mod. HC, GA, and MDD, the delay of CoS1 is consistently lower than the delay of CoS2. Similar to the static case above, the delay of CoS1 for the mod. HC scheme is quite low, but CoS2 exhibits very high delay. Surprisingly, the MDD algorithm performs better than the GA in meeting the CoS1 delay target. This could be attributed to the fact that MDD uses actual delay measurements for its gateway selection decision, whereas the GA uses the delay model according to Eq. 5.4.

Plots of the PDR are omitted here, since practically all packets are delivered at all times for all aircraft, regardless of the routing and gateway selection scheme.

Fig. 8.6 shows the cumulative flight time for which an aircraft's delay for CoS1 exceeds the value $(1 + \alpha)\cdot$ 0.25 s, for $0 < \alpha < 1$. For sake of comparison, each aircraft in the simulation flies for approx. 18,000 s, leading to a total flight time of approx. 270,000 s. This corresponds to the area underneath the curve in Fig. 8.4. The MDD algorithm is not able to keep the delay of CoS1 below the target (i.e. $\alpha = 0$) for all aircraft at all times. However, the total time for which the target is violated is only one third that of the HC algorithm, and about half of the mod. HC and delay metric based gateway selection algorithms. Also, higher delay values ($\alpha > 0$) are also much less frequent with MDD than with the HC or mod. HC schemes. Surprisingly, the MDD algorithm leads to less violations than the GA. This is most likely due to the fact that MDD makes its decisions based on actual delay estimates in the network, whereas the GA relies on the delay model of Chapter 5. When there are differences in the link delay model and the actual link delay resulting from the operation of the distributed scheduling algorithm, the MDD algorithm will take the actual delay estimates from the network into account. The solution that was calculated by the GA was based on slightly incorrect inputs and thus does not behave exactly as expected.

The fraction of traffic sent over the satellite gateway is shown in Fig. 8.7. As expected, the mod. HC solution sends exactly half of its traffic (i.e. all CoS2 traffic) over the satellite, as long as a satellite gateway exists in the network. At the beginning and end of the simulation, no satellite gateway exists and the mod. HC scheme is forced to send all traffic over the terrestrial gateways. Due to the prominent position of the satellite gateway in the middle of the network, the HC scheme sends the most traffic over the satellite. MDD uses the satellite slightly more than the GA. The delay metric scheme does not appear to provide a stable gateway allocation, as the fraction of satellite traffic fluctuates significantly. Although this could probably be improved by adjusting the parameters of the gateway selection, particularly $gatewayHoldTime$, it is clear that MDD with the delay metric provides better performance than the delay metric alone.

The average values for packet delay, PDR and fraction of traffic sent via the satellite gateway over the course of the simulation are given in Table 8.3. It can be seen that MDD keeps the delay of CoS1 significantly lower than CoS2, but achieves this at the expense of a slightly larger packet loss for CoS2, due to the concentration of packets around the satellite gateway.

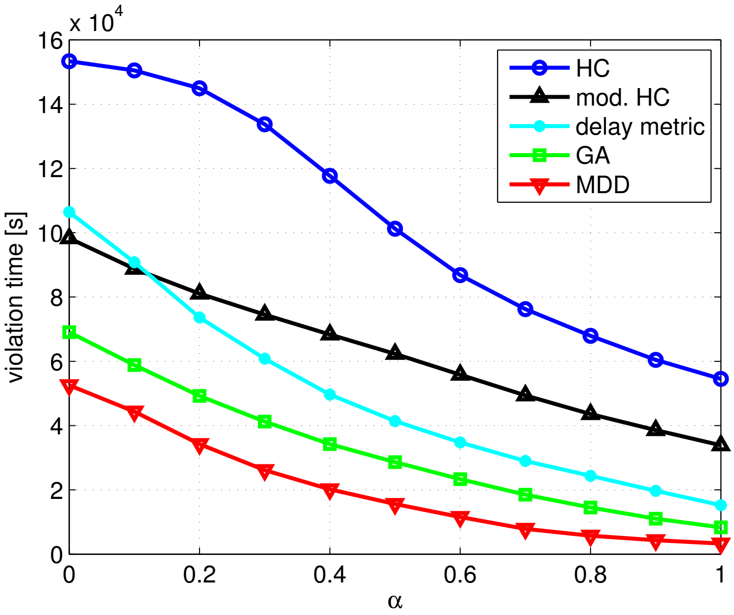


Figure 8.6.: Cumulative delay violations for CoS1.

8.5. Signaling Overhead

Next, we will take a look at the effects of the rate at which HELLO and Topology Control messages are generated by the nodes. In general, the routing overhead increases

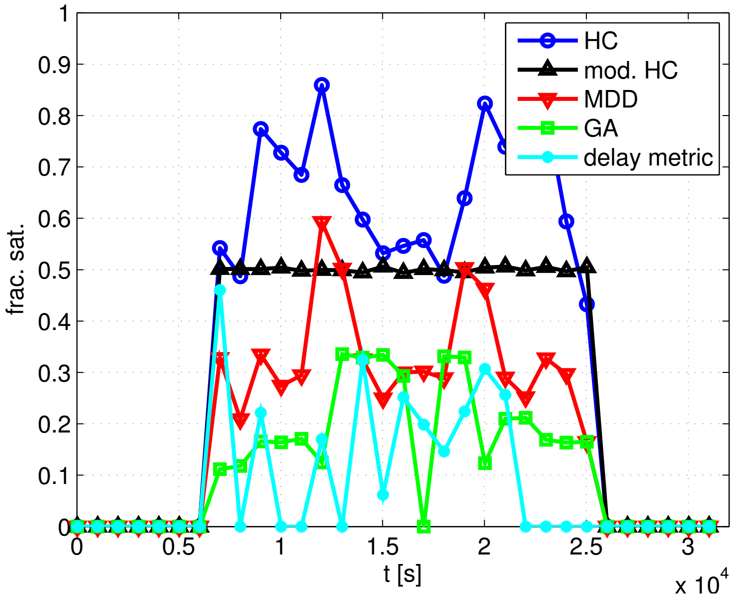


Figure 8.7.: Fraction of traffic sent via satellite.

	delay [s]		**PDR**		**frac. sat.**
	CoS1	**CoS2**	**CoS1**	**CoS2**	
HC	0.331	0.331	0.991	0.991	0.518
mod. HC	0.259	0.541	0.999	0.998	0.444
delay metric	0.243	0.243	0.982	0.982	0.122
GA	0.216	0.248	0.986	0.988	0.1557
MDD	0.185	0.261	0.999	0.974	0.272

Table 8.3.: Mean delay, PDR, and fraction of satellite traffic over entire simulation duration.

with the number of aircraft and the update rate of the control messages, so that lower update rates are desirable in order to reduce the amount of overhead. On the other hand, outdated information in the nodes' topology database could lead to bad decisions in the routing and gateway selection. Therefore, the optimum update rate is a compromise between these two goals.

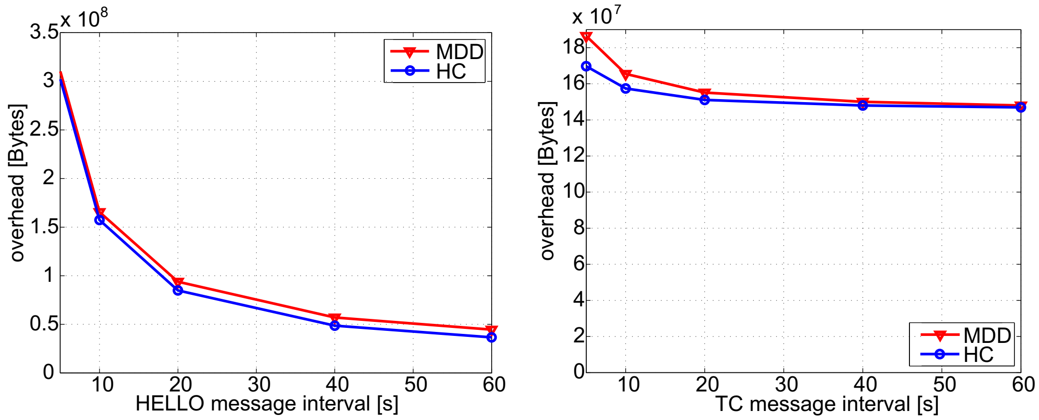
For the calculation of the control message overhead of the MDD protocol, the definitions of the HELLO and TC messages as given in Appendix A were used. For the HC scheme, the link metrics and gateway blockings were omitted from the TC messages. However, HELLO messages still carry the information about the TDMA schedule and node positions, and the TC messages contain the GATEWAY and AR_ADDR fields, when applicable.

First, we will take a look at the effect of varying update rates for the HELLO and TC messages on the total overhead. Fig. 8.8 shows the total overhead in terms of Bytes that is produced by the 15 aircraft over the course of the simulation. In Fig. 8.8(a), the interval between HELLO messages sent by a node is varied between 5 s and 60 s, while the TC interval is kept constant at 10 s. In Fig. 8.8(b), the interval between TC messages is varied in the same range, while the HELLO interval is kept at 10 s. Note that HELLO and TC messages that are created on demand, e.g. when a link is created or breaks, or a neighbor appears or when times out, are not affected by this varying message interval.

Obviously, the HELLO messages account for a much larger share of the total overhead than the TC messages. Doubling the HELLO interval from 5 s to 10 s almost reduces the total overhead by half. The difference in overhead between the HC and MDD protocols is very low.

The effect of varying the control message transmit intervals on the average packet delay is shown in Fig. 8.9. The message intervals of HELLO and TC messages are varied in the same range as previously. As could be expected, the delay of the MDD algorithm increases as the time between updates increases from 10 s to 60 s. However, The delay actually drops when the interval is increased from 5 s to 10 s. The same behavior can be observed for the TC intervals. Too frequent updates can apparently lead to unstable behavior. Frequent changes in the routing or gateway selection do not give the scheduling enough time to react, resulting in higher delay values.

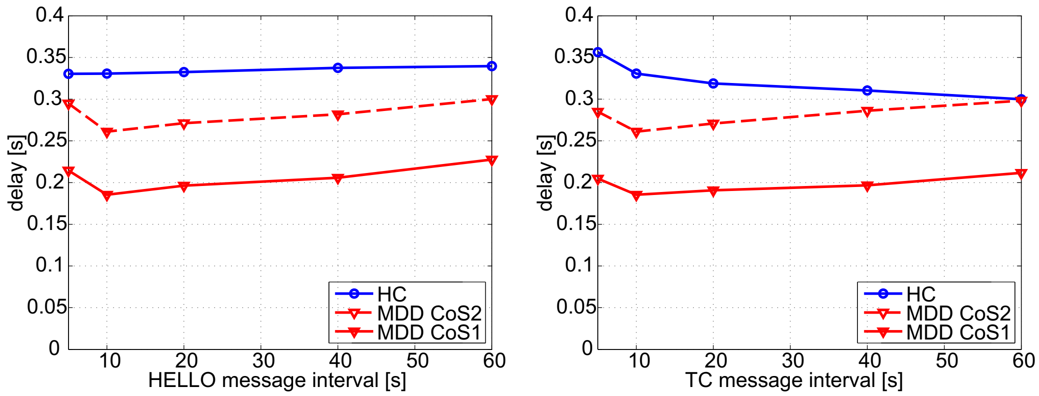
An unexpected effect is the decrease of the average packet delay for the HC scheme as the TC interval is increased in Fig. 8.9(b). A possible cause for this is a higher route stability. When hop count is used as the routing metric, many routes of equal cost exist between a source-sink pair. Likewise, two gateways may be an equal number of hops away from a node. In such a case, the HC algorithm randomly selects one of these alternatives, subject to a constraint on the minimum gateway holding time. If TC messages are generated less frequently, the nodes will perform fewer routing updates, leading to a higher stability of the routing and gateway selection in the network. This is beneficial for the network performance, since the scheduling algorithm always requires some time to adapt to the new situation. During this time, packets will experience longer delays and queues may build up temporarily, leading to packet loss. Since only the TC messages and not the HELLO messages contribute to the routing database, this effect is not present in Fig. 8.9(a). Due to the fundamentally more dynamic nature of the MDD algorithm, and its dependency on up do date knowledge of the link delay, this effect cannot be observed in the results for MDD.



(a) Varying HELLO interval.

(b) Varying TC interval.

Figure 8.8.: Dependency of routing overhead on control message intervals.

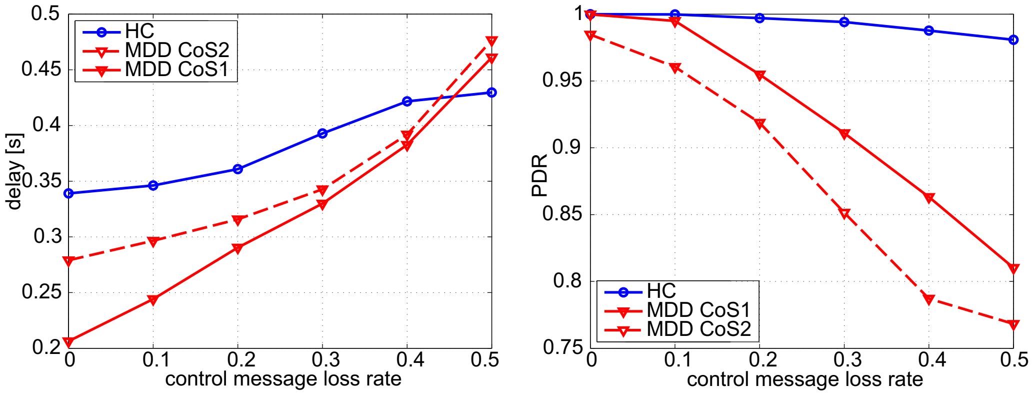


(a) Varying HELLO interval.

(b) Varying TC interval.

Figure 8.9.: Dependency of packet delay on control message intervals.

Up to now, it was assumed that HELLO and TC messages are delivered reliably. However, control messages can be lost on the wireless channel, due to noise or collisions. Therefore, we also simulate the MDD and HC algorithms with varying packet loss probabilities for control messages. Note that this packet loss does not directly affect the transmission of user data, since a separate radio interface is assumed for user data and control data. The effect on user data is only indirect, via the routing and gateway allocation decisions. The packet delay and PDR over the control message loss rate are shown in Fig. 8.10. TC and HELLO messages are transmitted every ten seconds. A node assumes that a neighbor is no longer reachable if it has missed three subsequent HELLO messages. Likewise, a node is removed from the routing topology database after three missed subsequent TC messages. It is obvious that the MDD algorithm is much more sensitive to loss of control messages than the HC scheme. As the message loss rate increases, the MDD algorithm is no longer able to fulfill the CoS1 delay target, and at high loss rates, CoS1 and CoS2 experience almost the same delay. For MDD, even a small packet loss rate for control messages already leads to a significant degradation of the PDR. The PDR of the HC scheme is much less affected by the loss of control messages. Clearly, the communications system that is used for the control messages should be made as reliable as possible for MDD to function properly.



(a) Average packet delay.

(b) PDR.

Figure 8.10.: Network performance as function of control message loss rate.

9. Realistic Performance Assessment

In this chapter, we analyze the performance of the proposed MDD routing and gateway selection scheme in a realistic simulation environment. We simulate air traffic over the North Atlantic, based on information from a database of actual scheduled commercial flights. Realistic data traffic is generated by modeling typical Internet services. As a reference for the performance assessment of the MDD scheme, we again simulate the hop count (HC) based gateway selection and routing scheme.

As in the rest of this thesis, all simulations are performed with the OMNeT++ network simulator. Where necessary, we have extended the functionality of the OMNeT++ simulator with functionality specific to the aeronautical environment. These extensions were partially within the scope of the project FACTS - Future Aeronautical Communications Traffic Simulator at the German Aerospace Center (DLR) [151].

9.1. Simulation Network Topology

The possibilities for deploying air/ground base stations in the North Atlantic region are limited. Sites should to be close to settlements, and should have a backbone connection to the Internet capable of handling large volumes of data with low latency. Based on these considerations, we have selected eight locations in the North Atlantic region as sites for ground stations in our simulations. These sites are:

- Prestwick, Scotland (55.51°N, 4.59°W)
  Prestwick is an import site for North Atlantic air traffic, since an Air Traffic Control (ATC) center there currently provides ATC services for half of the eastern North Atlantic airspace.

- Shannon, Ireland (52.70°N, 8.92°W)
  Together with Prestwick, Shannon currently provides ATC services in the eastern North Atlantic.

- Sligo, Ireland (54.28°N, 8.60°W)
  Sligo is the site of an airport on the Atlantic coast of Ireland.

- Gander, Newfoundland, Canada (48.96°N, 54.61°W)
  The ATC center in Gander provides ATC services for one half of the western North Atlantic.

- Goose Bay, Labrador, Canada (53.32°N, 60.43°W)
  Together with Gander, Goose Bay provides ATC services for the western North

Atlantic. Also, Goose Bay is the location of a major Royal Canadian Air Force airfield.

- Deer Lake, Newfoundland, Canada (49.21°N, 57.40°W)
  Deer Lake is the site of an airport serving the city of Corner Brook in western Newfoundland.

- Qaqortoq, Greenland (60.72°N, 46.03°W)
  Qaqortoq is the location of a small airport, and the landing site of the *Greenland Connect* undersea cable connecting Greenland to Canada and, via Iceland, Denmark [152].

- Nanortalik (60.15°N, 45.22°W)
  Nanortalik is a small town in southern Greenland, and the site of a heliport.

The locations of these eight ground stations are depicted as small blue circles in Fig. 9.1. Their communication ranges are indicated by larger black circles.[1] The North Atlantic Tracks used in our simulations do not bring the aircraft within range of the coastline of Iceland, and routes leading aircraft to fly this far the the North are in general used very seldom. Therefore, we do not consider any additional ground stations located in Iceland.

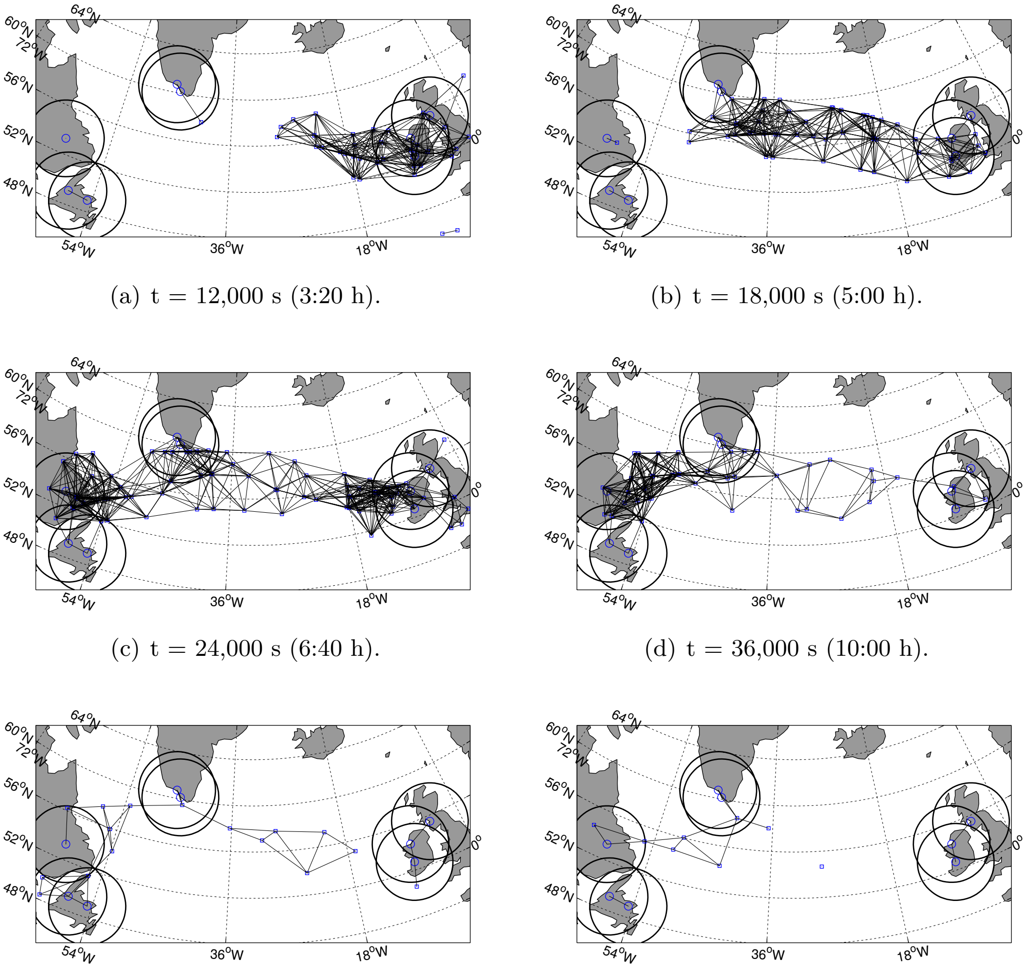
For our simulations, we have used the North Atlantic Tracks that were active on 14 March 2012 as provided in the Innovata flight database [28]. We extracted all westbound transatlantic flights from Europe to North America within the middle 24h interval from the database of flights on 21-22 May 2007. The reason for considering only westbound flights is that these flights take place during daytime, and passengers will be awake for most of the flight's duration. Eastbound flights take place at night, and many passengers will try to sleep. Therefore, the data traffic load on westbound flights is expected to be significantly higher. We focus on this case since it is the more challenging of the two.

Forty percent of the flights contained in the database are created in the simulations. These flights are then assigned to tracks in a round-robin fashion, keeping their departure and arrival times fixed. The route segments from the starting airport to the first waypoint of the track, between waypoints, and from the last waypoint to the destination airport were interpolated along great circle routes. The speed of each aircraft was adjusted such that the arrival time at the destination airport as given in the database is met. All aircraft are assumed to fly at a typical cruise altitude of 10,000 m. The resulting network topology at different time instants is shown in Fig. 9.1. As stated above, the range of the ground stations is indicated by circles. Links are drawn between any two nodes that are within each others' radio horizon according to Eq. 4.1. Due to the aircraft's elevation above the ground, the maximum range of air to air links is twice the maximum range of an air to ground link. Note that the six snapshots are not taken at regularly spaced intervals. The instantaneous number of aircraft in the simulation is plotted in Fig. 9.2.

[1] The range is defined such that an aircraft flying at an altitude of 10,000 m is exactly on the radio horizon of the ground station, which is assumed to be located at sea level.

The peak aircraft count is 86, and is encountered between t=8:20h and t=9:36h. The total number of aircraft involved in the simulation is 96.

Whenever an aircraft is generated in the simulation, it is chosen to act as a satellite gateway with probability one half. This relatively high resulting number of Internet gateways ensures that the Internet connectivity of the ad hoc network is sufficient to handle the traffic demand of all aircraft, and helps to prevent the network, or parts of it, from becoming disconnected from the Internet.



(a) t = 12,000 s (3:20 h).

(b) t = 18,000 s (5:00 h).

(c) t = 24,000 s (6:40 h).

(d) t = 36,000 s (10:00 h).

(e) t = 46,000 s (12:47 h).

(f) t = 52,000 s (14:27 h).

Figure 9.1.: Snapshots of the simulated network topology in the North Atlantic.

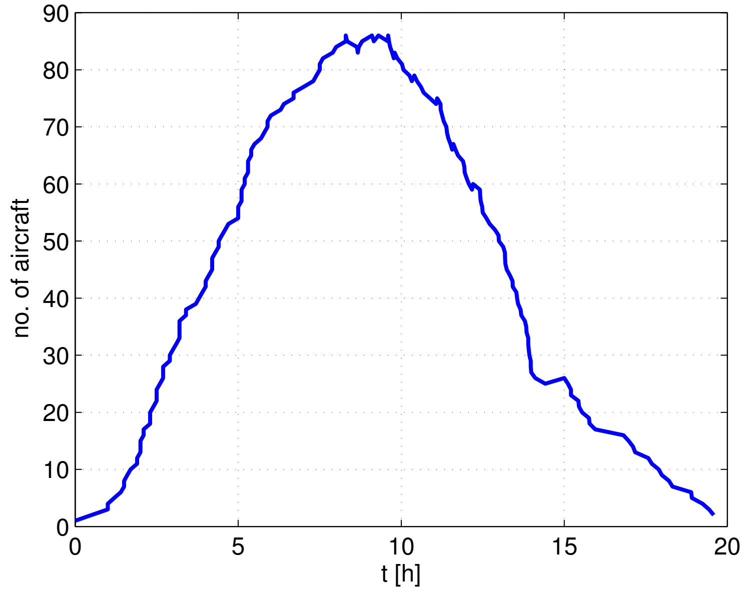


Figure 9.2.: Number of aircraft in the simulation.

9.2. Simulation Parameters

In the previous chapters, it was always assumed that one packet could be transmitted per slot, regardless of the transmission distance or the interference. Here, we apply a more realistic model for the capacity of a link. The data rate of the wireless A/A and A/G links is adapted to the SINR conditions of each TDMA slot. The SINR of a link can vary from slot to slot. It depends on the signal and thermal noise powers, antenna patterns, and the allocations of other links to the same time slot, and is calculated according to Eq. 6.6. With the same link parameters as in Table 5.1, we use the following rate regions:

SINR			Modulation Scheme	data rate [Mbps]
	<	10.00 dB	-	0
10.00 dB	–	13.54 dB	BPSK	20
13.54 dB	–	20.15 dB	QPSK	40
20.15 dB	–	26.28 dB	16-QAM	80
26.28 dB	–	32.26 dB	64-QAM	120
	>	32.26 dB	256-QAM	160

Table 9.1.: Rate regions for large scale simulations, depending on the SINR.

These regions were calculated such that larger modulation alphabets (from BPSK to 256-QAM) are used as the SINR increases and the bit error rate is always kept below

$1 \cdot 10^{-6}$, assuming the bit error rate expressions given in [112]. The data rates given in the table are the gross data rates and do not consider the effect of the guard intervals which are required between each TDMA slot. Also, these data rates are the burst rates that are only achieved during the short times that a slot is actually allocated to a link. Since a link will not be able to transmit in every slot, the actual data rate achievable on a link in the simulations will be significantly less than the rates in Table 9.1.

The satellite link is assumed to have a data rate of 10 Mbps in the downlink (i.e. to the aircraft) and 2 Mbps in the uplink direction. This corresponds to twice the rates that were provided by the Connexion By Boeing service per transponder [110]. However, it was possible to equip each aircraft with more than one such transponder in order to increase the capacity. The delay of a packet sent over a satellite link consists of the queuing delay, if applicable, as well as transmission delay and 240 ms propagation time. As with the other wireless links, the length of the transmission queue is limited. When the queue is full, newly arriving packets are dropped.

In the simulations in the preceding chapters, data traffic was simply modeled by a Poisson process generating packets of fixed size, with the destination being randomly selected on a packet by packet basis. Here, we will use a much more realistic traffic model intended to reflect the properties of real Internet use by the passengers. Models of HTTP web browsing, audio/video streaming, eMail, and file transfer are considered. These models provide the rate at which packets are generated, as well as the size of each packet in Bytes. In any time slot, the number of Bytes that can be sent over a link is determined according to the link's SINR as described above. Thus, multiple packets can be sent per slot, but packets can also be fragmented if they are too large to be transmitted in a single slot. How these applications are modeled on a per user basis and how the traffic is scaled to an entire aircraft is described in detail in Appendix B.

Each of the types of traffic is assigned a Class of Service and a corresponding delay target for use in the MDD algorithm. The values assigned to the four traffic types are given in Table 9.2. The delay target for eMail traffic is very generous, at 10 s. Effectively, such large delay values are not encountered in the simulations, and this amounts to not setting a delay target at all. The relevant parameters of the MDD algorithm are summarized in Table 9.3, remaining system parameters are collected in Table 9.4. Unless explicitly stated otherwise, these parameters are used for all following simulations.

CoS	**Traffic type**	**Delay Target**
1	Audio/video streaming	250 ms
2	HTTP	500 ms
3	File transfer	1 s
4	EMail	10 s

Table 9.2.: Mapping of traffic types to Classes of Service and delay targets.

Parameter	Value
α	0.75
β	0.5
$gwBlockTime$	60 s
$gwHoldTime$	600 s
$gwSwitchCounterMax$	10
HELLO interval	10 s
TC interval	10 s

Table 9.3.: Simulation parameters of MDD algorithm.

Parameter	Value
Antenna elements	16
max. transmit queue length	30 packets
TDMA frame length	100 slots
TDMA slot duration	10 ms
Active users per aircraft	40

Table 9.4.: General simulation parameters.

9.3. Simulation Results

9.3.1. Choice of Time Slot Duration

As seen in the expression for the average per link packet delay that was given in Eq. 5.4, the TDMA time slot duration T_s directly affects the average delay. The shorter a time slot can be made, the lower the resulting packet delay. However, each time slot must contain a guard interval to compensate for the long signal propagation times. Otherwise, transmissions in subsequent time slots could overlap at the receiving node, leading to collisions and packet loss. Timing correction is typically used in terrestrial cellular radio systems such as GSM or LTE to align all receptions properly at the base station [123]. However, such a correction is not possible in ad hoc networks due to the lack of a central point of reference such as the cellular base stations. The length of this guard interval is determined by the maximum time shift due to propagation delay. Here, we will assume a constant guard interval of 2 ms in each slot. If the time slot duration is reduced in order to reduce the packet delay, a larger fraction of the total time is lost to the guard intervals, and the utilization of the channel is reduced. Therefore, choosing a value for the time slot duration T_s is a compromise between minimizing delay and maximizing capacity. The relationship between T_s and the channel utilization η is shown in Fig. 9.3. We have simulated the ad hoc network topology with the multiservice traffic model for values of the TDMA time slot duration T_s varying between 3 ms and 10 ms and will now take a look at the effect that the time slot duration has on the network performance.

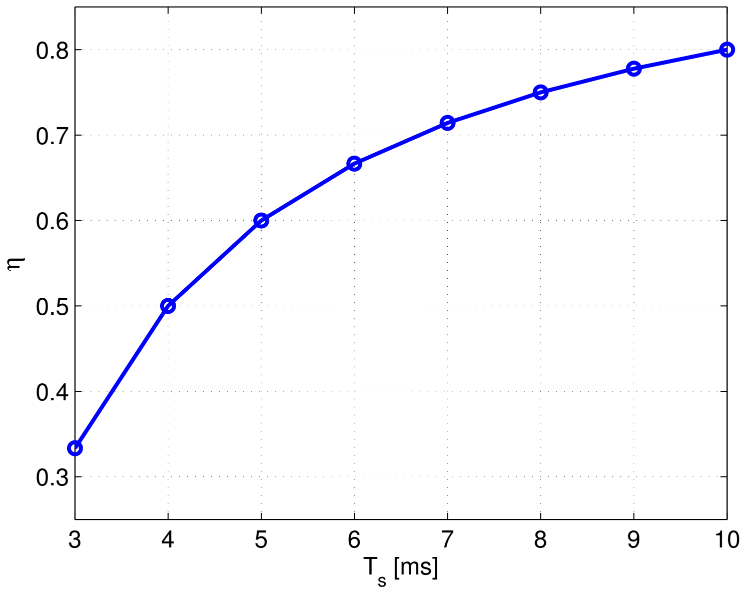


Figure 9.3.: Channel utilization vs. time slot duration T_s.

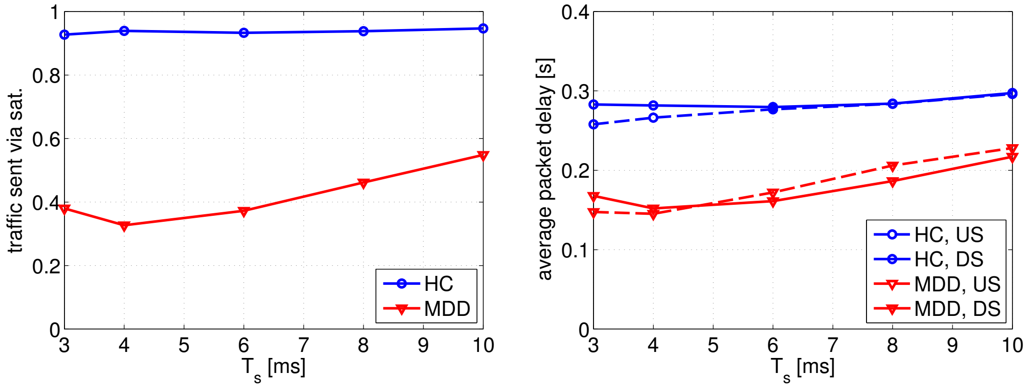
According to the link delay equation Eq. 5.4, the link delay decreases as T_s decreases, making longer paths towards the terrestrial Internet gateways more attractive. However, at some point, the resulting channel utilization will be so low that only very little data can be transmitted over these links, congestion may occur, and the satellite gateways may become more attractive.

This expected behavior is confirmed by the plots of the fraction of traffic sent via the satellite gateways in Fig. 9.4(a) and the average packet delay in Fig. 9.4(b). Of course, the HC scheme sends most traffic over satellite gateways for all values of T_s, as it considers only the hop count. On the other hand, the fraction of satellite traffic in the case of MDD depends strongly on T_s. Obviously, the average packet delay is very tightly linked to the amount of traffic that is sent over the satellite gateways. For all values of T_s, the delay of the MDD scheme is significantly lower than for the HC scheme due to the lower use of satellite gateways. Therefore, as T_s increases, the delay of MDD rises faster than the delay of HC, since proportionally more packets are being affected by the longer time slot duration.

In Fig. 9.4(b), the delay in downstream direction is drawn as a solid line, whereas the upstream delay is drawn as a dashed line. The upstream delay is higher than the downstream delay for MDD. This is due to the STDMA scheduling algorithm, which allocates time slots to links based on their traffic demand. Since the amount of upstream traffic is significantly less than downstream traffic, the links towards the gateways are assigned fewer slots, resulting in a higher delay. Since HC relies much less on the STDMA links, this effect on the delay cannot be observed there.

For MDD, these two general trends – higher delay due to longer slots, and higher delay in upstream direction – are no longer valid for T_s=3 ms. This extremely short slot duration wastes two thirds of the channel capacity and leads to congestion on the wireless links. This is reflected by the increase in satellite traffic at T_s=3 ms for MDD. For both HC and MDD, this is also the only point where the upstream delay is less than the downstream delay. Since there is less traffic in upstream direction, the upstream links are not yet congested and the packets continue to benefit from the shorter time slots. In the other direction, though, the queue lengths are building up, leading to longer delays.

Since the lowest delay is achieved by MDD for T_s=4 ms, we will use this value for all following simulations.



(a) Fraction of traffic sent via satellite.

(b) Average packet delay.

Figure 9.4.: Dependency of the fraction of satellite traffic and average packet delay and on the TDMA slot duration T_s.

9.3.2. Routing Message Update Rate

The rate at which nodes in the wireless network generate routing control messages, i.e. the HELLO and Topology Control messages is an important parameter of the MDD protocol. The delay based routing and gateway selection scheme can only function properly if reliable and up to date estimates of the link delays are available at each node in the network. On the other hand, these messages are overhead, which should be kept to a minimum. In this section, we analyze the network performance for different values of the HELLO and TC message generation rates.

Fig. 9.6 shows the dependency between the network performance in terms of delay and PDR and the rate at which HELLO messages are generated. Here, the TC message interval is kept constant at 10 s. In both cases, the performance of the MDD algorithm degrades much faster than the HC algorithm as the interval between subsequent HELLO messages is increased.

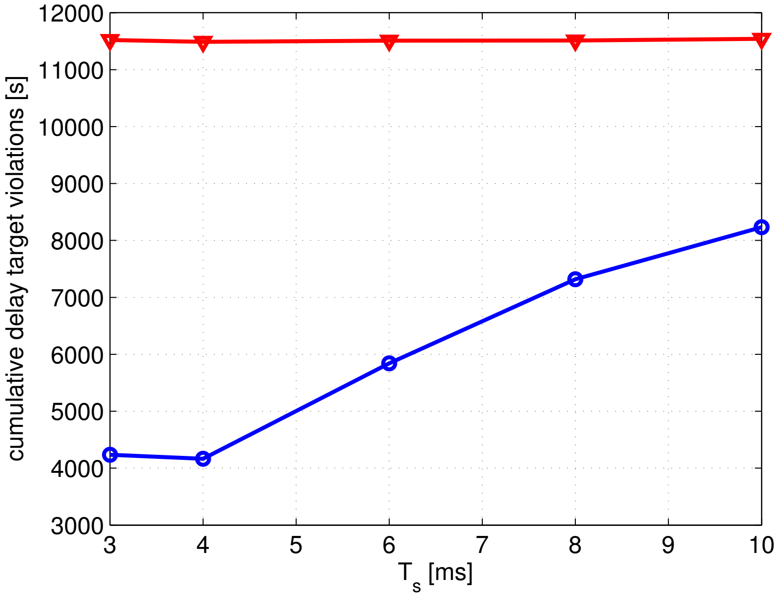


Figure 9.5.: CoS1 delay target violations vs. time slot duration T_s.

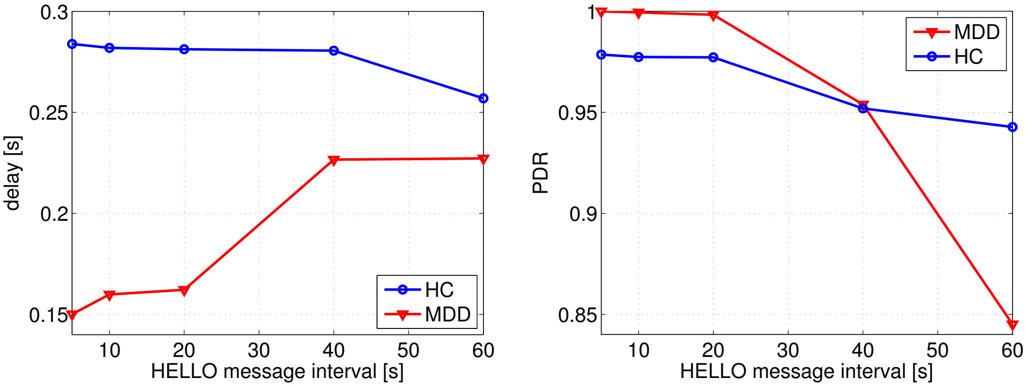
On the other hand, the sensitivity with respect to the TC messages is much lower, as seen in Fig. 9.7. Here, the HELLO message interval is kept constant at 10 s. The delay is hardly affected by longer intervals between TC messages. The PDR of the MDD algorithm does decrease as the message interval is increased, but the PDR is still very close to one.

The total amount of routing overhead generated by the nodes in the network during the course of the simulation is shown in Fig. 9.8, again for varying intervals between HELLO and TC messages. Depending on the parameter values, the overhead generated by the MDD protocol can be up to twice the overhead of the HC scheme. The contribution of the TC messages to the total overhead is higher than the contribution of the HELLO messages.

In the simulations of the following sections, the transmit interval of TC and HELLO messages will be kept at 10 s. This value is small enough to prevent a significant performance degradation due to outdated topology information.

9.3.3. Network Performance

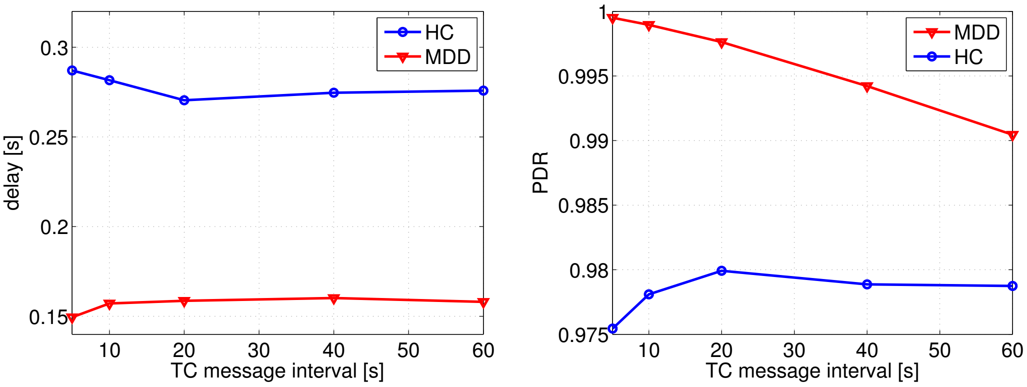
With the choice of the TDMA time slot duration of 10 ms and the interval between routing messages set to 10 s, we now take a closer look at the network performance. Fig. 9.9 shows how the average packet delay changes during the course of the simulation. The HC scheme does not distinguish between service classes. Therefore, only one line is plotted in this case. On the other hand, in the case of MDD, the routing and gateway allocation of packets does depend on the service classes and their delay targets. There-



(a)

(b)

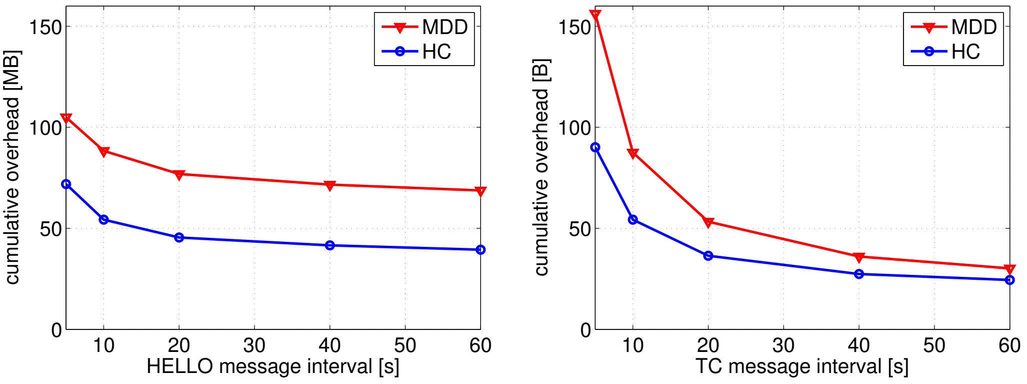
Figure 9.6.: Dependency of average packet delay and PDR on the interval between HELLO messages.



(a)

(b)

Figure 9.7.: Dependency of average packet delay and PDR on the interval between TC messages.



(a)

(b)

Figure 9.8.: Dependency of the cumulative signaling overhead on the frequency of HELLO and TC messages.

fore, the average delay of each service class is plotted separately. It can be seen that lower delay targets do indeed result in lower delay being experienced by the packets. However, when using MDD, even packets of CoS4 still experience lower delay than they would with the HC scheme.

The behavior of the delay is strongly correlated with the way in which the different service classes are allocated to gateways. The fraction of traffic sent over the satellite gateways is shown in Fig. 9.10. Again, only one line is plotted for HC, whereas MDD is split between service classes. As expected, the lower a service class' delay target, the lower the probability that it will be sent via satellite.

The curves of the MDD scheme in Fig. 9.9 and Fig. 9.10 exhibit a strong downward trend between t=5 h and t=17 h. This is due to the particular deployment of the ground stations. As the cloud of aircraft shifts towards the western Atlantic shore, a larger fraction of the traffic can be handled by the ground stations in Greenland. After the cloud loses connectivity to the ground stations in Great Britain around 12:30 h, the delay continues to decrease, because the paths to the ground stations on the western shore now become shorter and shorter. The HC scheme does not send a significant amount of traffic over the terrestrial gateways anyway, and hence does not profit from the better connectivity in the western Atlantic.

The number of aircraft in the network that are unable to fulfill the delay targets of the four service classes is shown in Fig. 9.11 for both HC and MDD. The MDD algorithm is able to significantly reduce the number of violations of the delay target of CoS1. On the other hand, the service classes with higher delay targets experience a slightly larger amount of violations. This is due to the unfairness that is incurred by giving preferential treatment to the service classes with lower delay targets.

The average values for DS packet delay, cumulative delay target violations, the fraction of satellite traffic, and the PDR over the course of the simulation are shown in Table

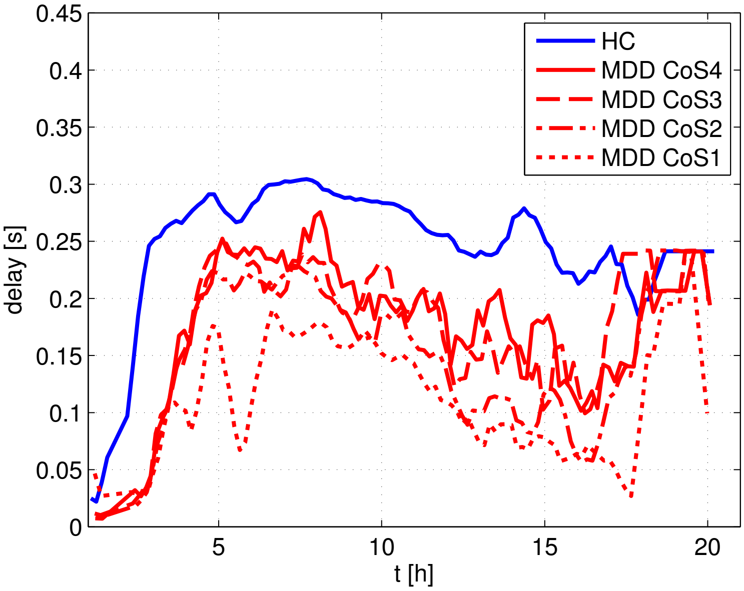


Figure 9.9.: Average packet delay for HC and MDD for the different service classes.

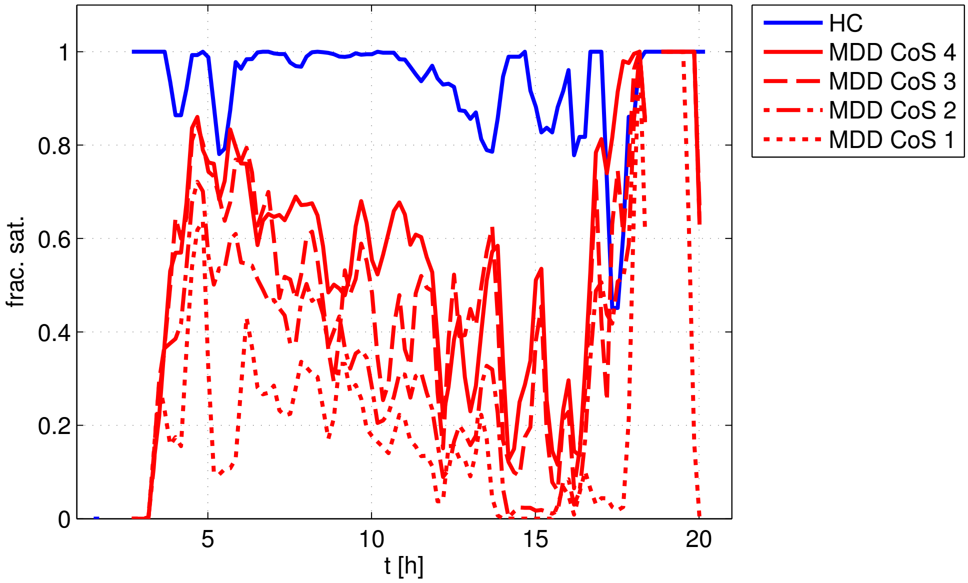
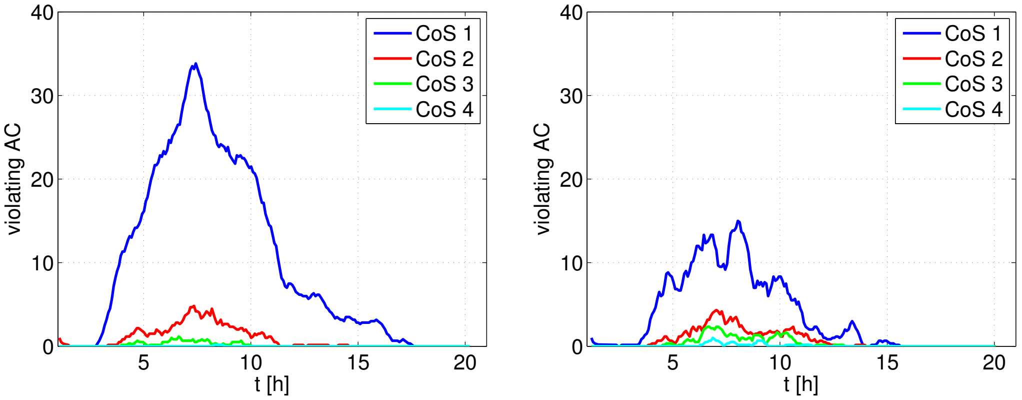


Figure 9.10.: Average fraction of satellite traffic for HC and MDD for the different service classes.



(a)

(b)

Figure 9.11.: Delay target violations for HC (left) and MDD (right) for the different service classes.

9.5. For MDD, these values are also split up among the four service classes. This table shows how the MDD algorithm provides lower delay to those service classes with lower delay targets. This is also reflected by the fraction of traffic that is sent over the satellite gateways. The average delay of CoS4 in the MDD case is still less than the average delay of the HC scheme. Also, MDD provides very good PDR, with HC losing six times as many packets.

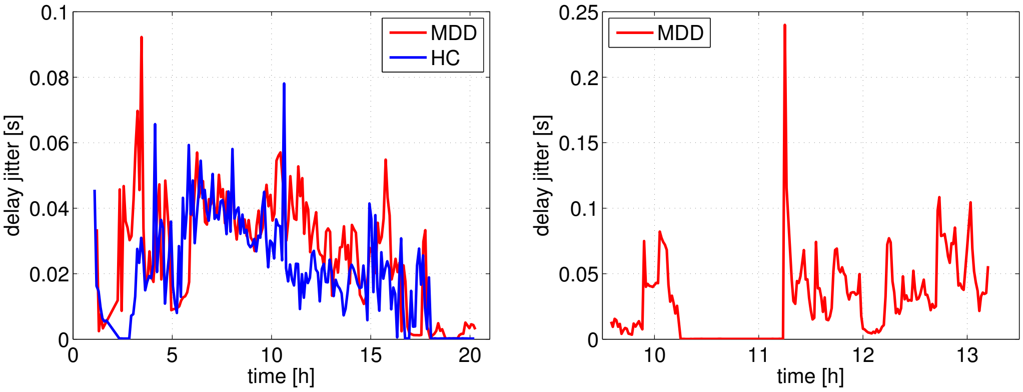
	MDD					**HC**
	CoS1	CoS2	CoS3	CoS4	all CoS	all CoS
delay [s]	0.1259	0.1878	0.2101	0.2020	0.1519	0.2817
cum. delay target violations [s]	4,166	763	335	64	5,328	12,714
sat. fraction	0.2168	0.3976	0.5132	0.5960	0.3269	0.9392
PDR	0.9997	0.9838	0.9936	0.9997	0.9926	0.9751

Table 9.5.: Time averages of network performance values over the duration of the simulation.

Another important network performance criterion is delay jitter. Table 9.6 shows how delay jitter affects the perceived quality of service for both a video streaming service and a VoIP connection. These values have been taken from the documentation of the MyConnection Server software, which can be used to analyze the quality of network connections [3]. It can be seen that VoIP is much more sensitive to delay jitter than video streaming. The main method of dealing with jitter is to use a buffer at the receiver in order to compensate the effects of the jitter. However, this leads to additional delay. Whereas this delay can be tolerated in video streaming, the interactive nature of a VoIP connection prohibits any additional delay due to such large buffers. Therefore, the jitter requirements of VoIP are much stricter. Although we do not consider a VoIP service for our performance assessment, it is interesting to see if the network would be able to support this service.

Video Streaming		**VoIP**	
1 ms:	high quality	1 ms:	radio quality
10 ms:	fair quality	5 ms:	standard quality
100 ms:	poor quality	14 ms:	broken sound
1000 ms:	unsupported	40 ms:	unsupported

Table 9.6.: Effect of delay jitter on perceived QoS, according to MyConnection Server software documentation [3].



(a) Average delay jitter.

(b) Delay jitter of single aircraft.

Figure 9.12.: Behavior of the delay jitter over time.

To answer this question, Fig. 9.12 shows the behavior of the delay jitter experienced by packets of the streaming service (CoS 1) over the course of the simulation. The average jitter of all aircraft in the simulation is shown in Fig. 9.12(a), whereas Fig. 9.12(b) shows the jitter that is experienced by a single aircraft during the course of its flight. From Fig. 9.12(a), it can be seen that the average delay jitter of the HC scheme is lower than the jitter of the MDD scheme. This is to be expected, since MDD prefers the terrestrial gateways, although they may be further away in terms of hops, and each hop along the path contributes to the jitter. Fig. 9.12(b) shows that the jitter experienced by a single aircraft can exhibit sharp peaks. These are caused by the time that the scheduling algorithm requires to adapt to changes in the routing or the gateway allocation. The average jitter rarely exceeds 50 ms. According to Table 9.6, this value is small enough to allow a video streaming application to function. However, it is much too high for VoIP. If VoIP services should be supported by an aeronautical ad hoc network, it will likely be necessary to add QoS measures to the link layer, allowing delay or jitter intolerant traffic to be prioritized by the scheduler.

9.3.4. Network Performance With Alternate Ground Station Deployment

The routes that are used by aircraft flying in the North Atlantic corridor depend on the current weather conditions. Sometimes, the routes may be shifted further to the North or South than in the network topology that was used for the preceding simulations (cf. Fig. 9.1). In these cases, the shape of the network would still be very similar. However, the connectivity of the network to the Internet could change. If the routes are very far to the South, the network may no longer be able to connect to the ground stations that are located in Greenland, thereby degrading especially the performance of the MDD scheme. On the other hand, the impact on the HC scheme will likely be very small, since the reliance on ground stations is much lower.

To quantify this effect, we present simulation results in this section, where the two ground stations located in Greenland have been removed. We are now left with only six ground stations instead of the previous eight. Again, we take a look at the average packet delay, the fraction of traffic that is sent over the satellite gateways, and the delay target violations.

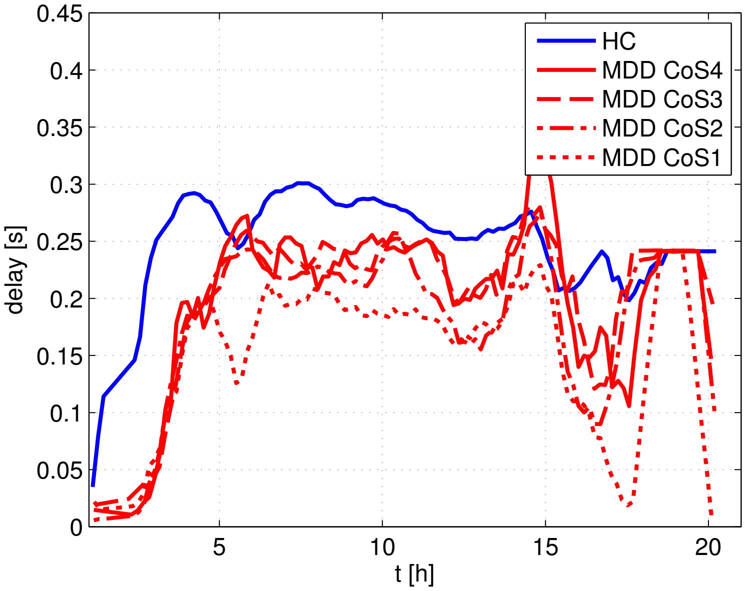


Figure 9.13.: Average packet delay for HC and MDD for the different service classes without ground stations in Greenland.

Fig. 9.13 shows the average packet delay. The HC scheme is not noticeably affected by the different ground station deployment scenarios. However, in contrast to the previous results (cf. Fig. 9.9), we see that the average delay of the MDD scheme no longer

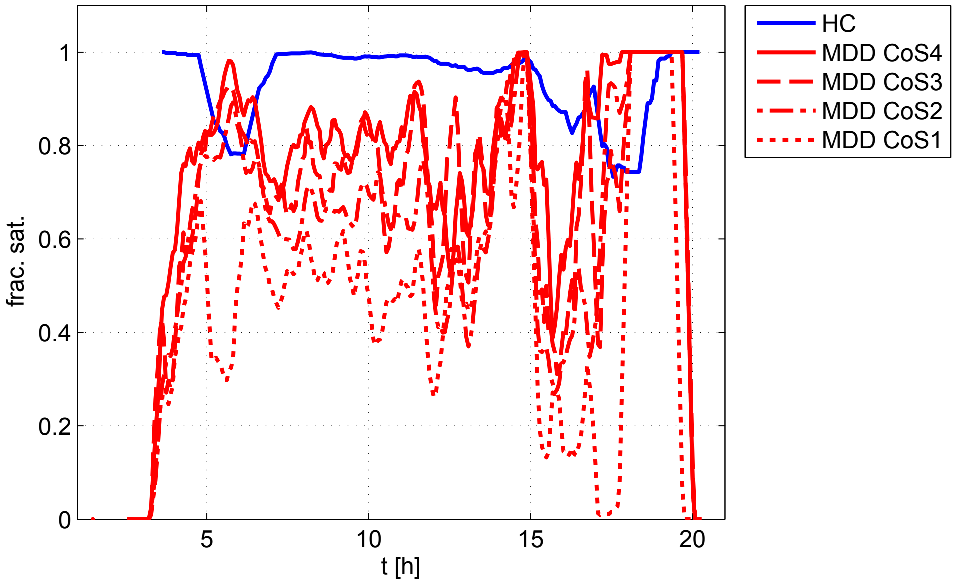


Figure 9.14.: Average fraction of satellite traffic for HC and MDD for the different service classes without ground stations in Greenland.

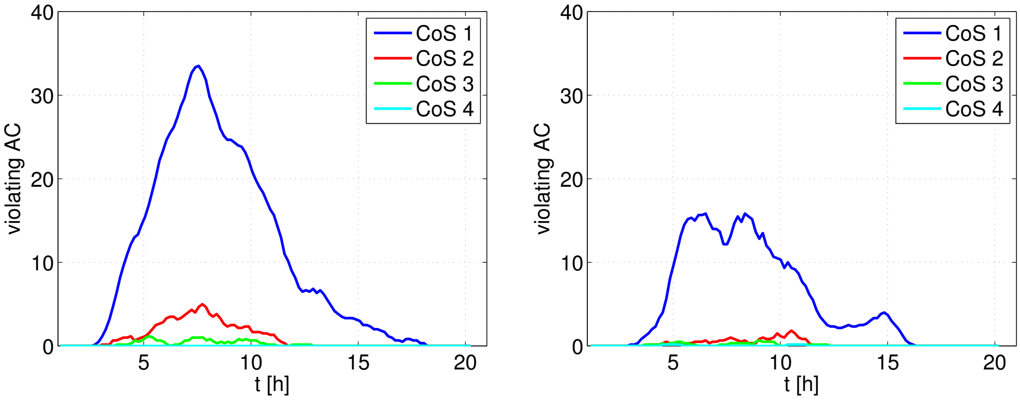
decreases between t=5 h and t=17 h. Correspondingly, the amount of satellite traffic remains relatively high during this time, as can be seen in Fig. 9.14.

In the case of the delay target violations, the HC scheme again is largely unaffected, as shown in Fig. 9.15(a). For MDD, we see in Fig. 9.15(b) that the number of violations in CoS1 increases slightly. Interestingly, there are fewer delay target violations for the remaining service classes. Here, MDD is forced to send more traffic over the satellite gateways. Obviously, this affects all four service classes. Since the delay of the satellite links is less than the target of CoS2-3, this may result in fewer violations.

The time averages of the performance parameters are given in Table. 9.7. The values of the HC scheme are almost the same as those in Table 9.5, which included the ground stations in Greenland. However, for MDD, the fraction of satellite traffic has almost doubled, and the delay has increased by about 30%. However, the PDR is still much better than that of HC.

9.4. Conclusion

In this chapter, we have performed a performance assessment of the proposed MDD gateway selection and routing protocol in a realistic simulation setting intended to capture the relevant aspects of an envisaged aeronautical ad hoc network in the North Atlantic region. It was shown that the MDD protocol is able to adapt its routing and gateway allocations to the delay targets that have been defined for the different service classes. MDD leads to significantly fewer violations of the delay target of the most sensitive



(a)

(b)

Figure 9.15.: Delay target violations for HC (left) and MDD (right) for the different service classes without ground stations in Greenland.

	MDD					**HC**
	CoS1	CoS2	CoS3	CoS4	all CoS	all CoS
delay [s]	0.1697	0.2199	0.2345	0.2315	0.1936	0.2794
cum. delay target violations [s]	5,960	345	183	47	6,535	13,049
sat. fraction	0.4706	0.6559	0.7103	0.7825	0.5892	0.9381
PDR	0.9996	0.9904	0.9968	0.9998	0.9967	0.9748

Table 9.7.: Time averages of network performance values over the duration of the simulation for alternate ground station configuration.

service class than the HC scheme.

However, the MDD protocol incurs higher overhead than a HC based routing and gateway selection scheme, and is more sensitive to stale information in the routing database. Also, the ability of the MDD protocol to support applications that would require low delay jitter, such as VoIP, is limited. Further QoS mechanisms at the link layer could reduce the jitter to tolerable levels.

10. Conclusions and Outlook

10.1. Conclusions

Ad hoc networks have recently been proposed as a means of providing in-flight Internet access to airline passengers in remote or oceanic regions. The ad hoc network is connected to the Internet by Internet gateways. These are aircraft that are connected either to a ground station via an air/ground link, or have a satellite link to the ground network. Existing gateway selection schemes fail to address the different characteristics of these two classes of gateways, and the traffic conditions along the path between the node and the gateway, adequately.

In this thesis, we consider the gateway selection, routing, and scheduling problem in such aeronautical ad hoc networks. Our goal is to minimize the average packet delay in the network, whether it is caused at the gateways or within the ad hoc network. We first consider the integration of such a joint routing and gateway selection scheme into the predominantly IPv6 based aeronautical network environment. Requirements for the protocol architecture of an aeronautical ad hoc network are formulated, and it is found that the most suitable solution is to perform the routing of packets through the ad hoc network below the IP layer. However, gateway selection must be done at the IP layer in order to comply with constraints of the IP protocol suite. We define a functional routing architecture for the mobile routers that performs the task of gateway selection at the IP layer, based on information provided by the underlying sub-IP ad hoc network routing protocol.

We then approach the gateway selection, routing, and scheduling problem from an algorithmic perspective and formulate a nonlinear binary integer program to minimize the average packet delay. Due to the mathematical complexity of this approach, we decompose the problem into two steps. The first step minimizes the weighted sum hop count of all flows in the network, subject to constraints that require the resulting gateway allocation and routing solution to be feasible, i.e. a valid schedule must exist for this solution. The second step then minimizes the average packet delay by optimizing the TDMA schedule for the gateway allocation and routing solution that was found in the first step. This decomposition greatly reduces the complexity, but comes at the cost that the solution is no longer guaranteed to be optimum. As an alternative, we define a Genetic Algorithm whose cost function is the average packet delay in the network. This algorithm can easily be extended to multiple service classes as well as mobile networks. This is a significant advantage over the mathematical programming approach, which requires the complete optimization to be performed again for every incremental change of the network topology or traffic demands.

Simulations show that solving the gateway selection, routing, and scheduling problems jointly yields significant performance gains over the more common approach of treating these problems separately. In heterogeneously connected networks with both terrestrial and satellite Internet gateways, the delay minimization criterion is able to split traffic elegantly between the two gateway types based on the current traffic load. The performance of the Genetic Algorithm is similar to the performance of the mathematical optimization, but at a much lower computational cost.

The Minimum Downstream Delay (MDD) routing and gateway selection protocol is then defined to operate within this functional architecture. It uses measurements of the average packet delay on each link as its metric for the task of gateway selection, analog to the centralized mathematical programming and genetic algorithm approaches. The concept of blocking gateways for certain service classes serves to protect more delay sensitive traffic from congestion caused by less delay sensitive traffic. The overhead due to control messages is limited by reusing messages for the scheduling and routing tasks. In small scale network topologies, it is verified that the performance of the MDD solution is close to that of the centralized solutions, and performs significantly better than other distributed solutions performing routing and gateway selection separately, based on e.g. the hop count metric or the delay metric, but without the gateway blocking functionality.

The performance of the proposed MDD protocol is then assessed by means of realistic simulations that are intended to capture the relevant characteristics of airline traffic in the North Atlantic region. We use flight data taken from a database of actual scheduled flights, along with actual routes flown by aircraft in the North Atlantic. Models of web browsing, media streaming, file transfer, and eMail services provide a realistic data traffic load.

Comparisons with a hop count based routing and scheduling solution in this realistic simulation environment show that MDD on average leads to lower packet delay, is better able to fulfill the service classes' delay targets by considering the delay that can be provided by each gateway, and assigns service classes to gateways accordingly. However, the control overhead of the MDD protocol is about twice the overhead of a pure hop count based solution, mainly due to the need to carry link metrics in the Topology Control messages. Also, MDD is more sensitive to outdated network topology information.

10.2. Outlook

In this work, we have addressed quality of service in terms of packet delay at the network layer, i.e. the tasks of routing and gateway selection. For the link layer scheduling in the distributed case, we have adopted the algorithm proposed by Grönkvist. Apart from this traffic load aware scheduling, we have not considered any particular quality of service mechanisms at the link layer. Packets to be transmitted over a link are simply placed into a first in / first out queue. When the queue is full, all newly arriving packets are dropped. We have seen that the delay jitter of packets in the ad hoc network is too high to support VoIP services. Although we do not believe that VoIP would be one of the primary applications for such a network, there may be a demand for such services in the future. Alternatively, other interactive services may require low jitter. The delay jitter could potentially be reduced by adding appropriate quality of service mechanisms at the link layer. For example, a packet scheduler could determine the order in which packets are transmitted over a link.

The stability of the topology could also permit resource reservations along the paths used to route traffic. Due to the near deterministic channel access behavior of the TDMA MAC, this would provide very reliable end to end quality of service guarantees inside the ad hoc network and go one step further in combining the routing and gateway selection problems with the scheduling.

Finally, a very interesting direction for future work would be the practical implementation of the proposed MDD protocol in a real-life testbed. Although simulations are certainly important, much more insight could be gained with experiments on real hardware.

A. MDD Message Formats

A.1. HELLO Message

The format of HELLO messages has been defined by the IETF MANET Working Group in the MANET Neighborhood Discovery Protocol (NHDP) [147] for the purpose of neighborhood discovery in ad hoc networks. The message definitions in [147] are in turn based on the Generalized MANET Packet Format as specified in [153]. Here, we adopt the definition of HELLO messages from [147], but add some extensions that are necessary for the use of HELLO messages in the MDD protocol described in Section 8.3, operating together with an STDMA scheduling algorithm at the MAC layer.

To be able to reuse these messages for the TDMA scheduling algorithm, we define a Type Length Value (TLV) structure to carry the link priorities, and the SCHED_COMP and SCHED_INCOMP TLVs to carry either complete or incomplete information about the TDMA schedule, respectively. NHDP specifies that HELLO messages are not forwarded. They are only relevant for a node's direct neighbors. In our case, though, we require these messages to be forwarded exactly once in order to cover the entire two hop neighborhood of the message originator, since all nodes within the two hop neighborhood must coordinate with each other in order to arrive at a valid TDMA schedule.

The beginning of an MDD HELLO message contains the typical information from [153]: Message Type, Message Flags (MF), Message Address Length (MAL), Message Length and Sequence Number fields. HELLO messages in our case do not need to carry an originator address, since the node is identified by the 24 bit ICAO identifier which can be extracted from both the IPv6 and link layer addresses. The Message TLV Block contains the Validity Time and Interval Time TLVs, a POSITION TLV, and the Address Block Flags (ABF) field.

The POSITION TLV is used to notify all neighboring nodes of the sender's position. Ten Bytes are used to encode the geographical position information, consisting of four Bytes each for latitude and longitude and two Bytes for altitude. Four Bytes allow for a precision of approx. $10 \cdot 10^{-6}$ degrees of latitude, corresponding to approx. 1 m. Precision in longitudinal direction will be higher, because all lines of equal longitude converge at the poles. The two Bytes for the altitude would allow for a precision of 1 ft in the range 0–65535 ft. This is certainly sufficient, given that the typical flight altitude is around 10,000 m, or 32,808 ft.

The address block specifies the 24 bit ICAO identifiers of all neighbors of the aircraft generating the HELLO message. The Generalized MANET Packet Format allows an efficient representation of addresses by defining address *head*, *mid*, and *tail* fields, which can be common to more than one neighbor. For example, if the HELLO is carrying IP addresses, many neighbors might share the same *head* if they are configured from the same IP network prefix. However, the use of ICAO identifiers in our case will likely not profit from this representation, because they are not easily aggregatable. Therefore, each address is represented completely in a 24 bit long *mid* part without a *head* or *tail*. This is indicated by the value of zero in the ABF field. The beginning of an MDD HELLO message is shown below in Fig. A.1. In all figures in this section, those data structures that have been defined in this work for use specifically by the MDD protocol are highlighted in yellow. Those parts that are defined by one of the MANET WG documents but are also used by MDD are kept in white.

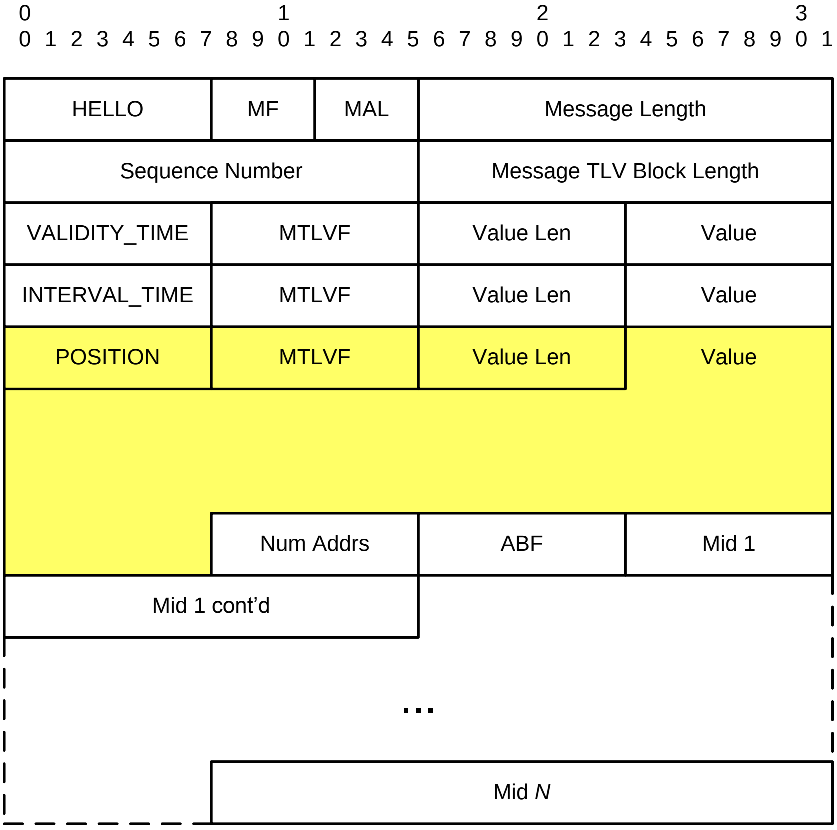


Figure A.1.: Beginning of an MDD HELLO message.

After this initial part, we add a message TLV that carries the node's current view of the TDMA schedule. Each HELLO message carries either a SCHED_COMP TLV or a SCHED_INCOMP TLV, depending on whether the complete TDMA schedule is contained, or only those parts of the schedule that are relevant to the message originator. The full schedule is encoded as a series of *(SlotID, SenderID, ReceiverID)* triplets of length 7 Bytes. The slot ID is encoded using 8 bits, the sender and receiver IDs are the 24 bit ICAO identifiers. Due to spatial reuse, there may be more than one entry for a slot.

An incomplete schedule only contains information about those slots which the originating node has been allocated either as transmitter or receiver of a link, and can be encoded more efficiently than the complete schedule. The slot ID is encoded in 7 bits, a single bit indicates whether the message originator is sending (1) or receiving (0), and 24 bits contain the id of the node on the far end of the link, which can be either transmitting or receiving, depending on the value of the preceding bit. Thus, each entry is 4 Bytes long. The total length of the SCHED_INCOMP TLV depends on the number of slots which have been allocated to the message originator.

In a lossless environment, the incomplete schedule information alone would be sufficient to provide every node with a consistent and complete picture of the schedule. However, losses of HELLO messages might lead to inconsistencies between the nodes' schedules. Therefore, the complete schedule information needs to be transmitted occasionally in order to ensure that all nodes have the same view of the schedule and to allow conflicts to be resolved if this is not the case.

The formats of the SCHED_INCOMP and SCHED_COMP TLVs are shown below in Fig. A.2 and Fig. A.3, respectively.

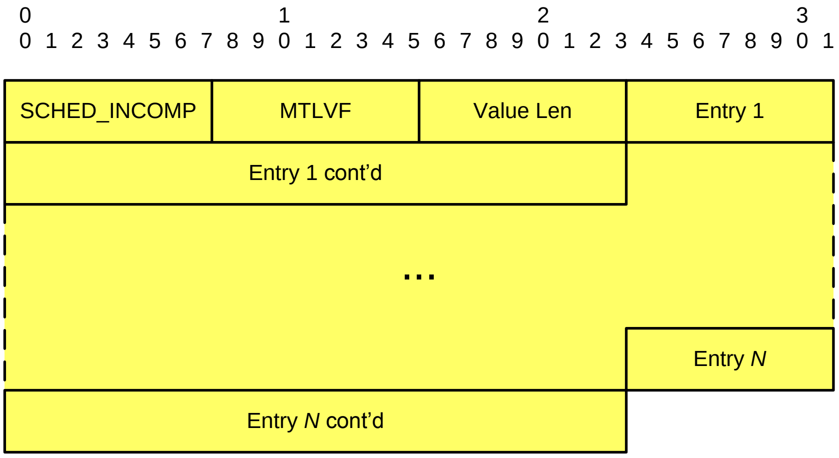


Figure A.2.: MDD SCHED_INCOMP TLV.

The Address TLV Block first contains the standard NHDP Link Status information (i.e. LOST, SYMMETRIC, or HEARD) for the link to each neighboring node. Then, the LINK_PRIO TLV, which is shown in Fig. A.4, contains the link priorities that are used by the TDMA scheduling algorithm. A single Byte is sufficient to represent the link priority value. Because the information in these TLVs is tied to neighbor addresses,

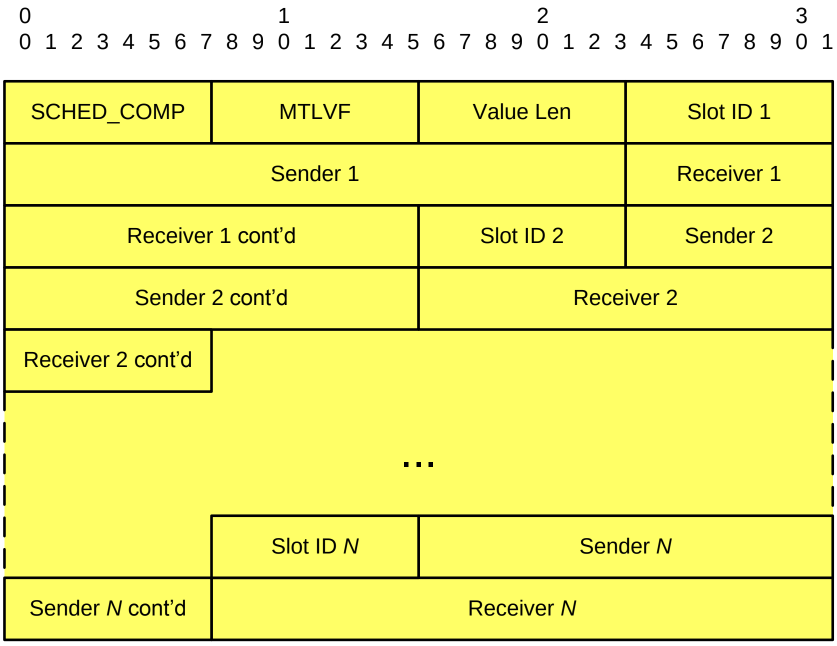
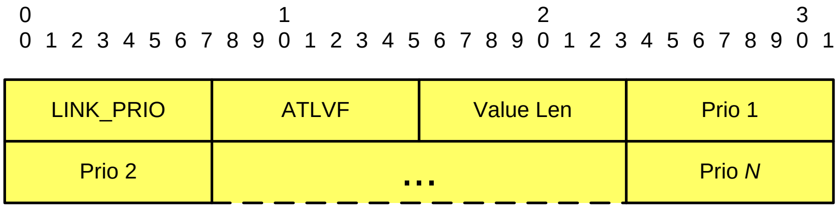


Figure A.3.: MDD SCHED_COMP TLV.

they are carried in the Address TLV Block. Finally, the positions of the neighboring nodes are contained in a NEIGHBOR_POS TLV, shown in Fig. A.5. This information is used by the scheduling algorithm. Again, ten Bytes are needed for each neighbor's position.



LINK_PRIO	ATLVF	Value Len	Prio 1
Prio 2	...		Prio *N*

Figure A.4.: MDD LINK_PRIO TLV.

The total length of a HELLO message depends on the number of neighbors and the number of slots reported in the schedule TLV. Assume that the message originator has N neighbors, the entire schedule contains $SLOTS\_TOTAL$ entries, and MY_SLOTS slots have been assigned to the message originator, either as transmitter or receiver. $SLOTS\_TOTAL$ may be larger than the number of slots per frame, since spatial reuse allows links to be activated simultaneously, as long as the signal quality at the receivers is sufficient. In this case, the message length is $(75+15\times N+7\times SLOTS\_TOTAL)$ Bytes for a HELLO carrying the complete schedule and $(75+15\times N+4\times MY\_SLOTS)$ Bytes for a HELLO carrying an incomplete schedule. In addition, a HELLO can be generated on demand, in response to a change in the node's neighborhood, e.g. when a link is created or breaks up. In this case, the HELLO does not need to carry any schedule information, and may carry information only about the neighbor that has changed. Then, the length of the HELLO is only 61 Bytes.

If a node has 20 neighbors, $SLOTS\_TOTAL$ is 200, and MY_SLOTS is 20, a HELLO carrying the complete schedule information will be 1775 Bytes long, whereas the length of a HELLO with incomplete schedule information will be only 455 Bytes.

A.2. Topology Control Message

The Topology Control (TC) messages are used by the OLSRv2 routing protocol [96] to build the topology database which is used to calculate the routing tables. They are used by nodes mainly to inform other nodes about the link metrics to its neighbors and possible attached networks. Since OLSR is a link state protocol, TC messages are flooded through the entire network. However, OLSR provides means to reduce the overhead due to flooding by introducing so-called Multipoint Relays (MPR). These are a subset of all nodes in the network, and only MPRs actually forward TC messages.

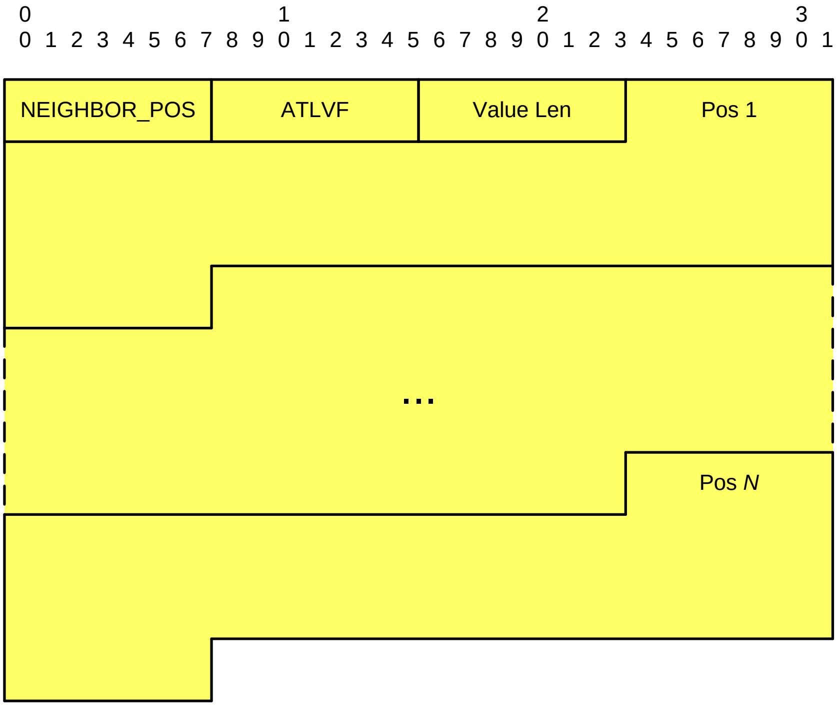


Figure A.5.: MDD NEIGHBOR_POS TLV.

The MPR selection process guarantees that all nodes in the network still receive all messages. Since MDD is also a link state algorithm, the TC format, as well as the MPR functionality, can be largely reused. Like the HELLO messages, TC messages are also based on the generalized MANET packet format defined in [153].

In our case, we extend the TC message definition to include information about blocking levels of gateways (cf. Sec. 8.3) and whether a node is acting as gateway. In this latter case, the node also reports the IPv6 address of the Access Router to which it is connected, since this is required for packets in the upstream direction as the next hop address at the IP layer. The routing within the AAHN is based only on the ICAO identifiers.

The beginning of a Topology Control message follows the definition in [96]. It contains the message type identifier (TC), Message Flags (MF), Message Address Length (MAL), Message Length, Hop Limit, Hop Count, and Message Sequence Number fields. This is followed by a Message TLV Block containing several TLVs. The VALIDITY_TIME TLV specifies how long the information in the TC message is valid, in order to prevent nodes from using outdated information. The INTERVAL_TIME TLV specifies how often the TC message is sent. CONT_SEQ_NUM is a sequence number for the message. MPR_WILLING indicates if the node is willing to act as a Multipoint Relay (MPR) for flooding reduction in OLSR.

The next TLV, the GATEWAY field, has been added for the MDD protocol. Its presence indicates that the node is acting as Internet gateway. The value corresponds to the cost metric of the gateway's link to the ground network. When the GATEWAY TLV is present, the AR_ADDR TLV must also be present. This TLV informs other nodes of the IPv6 address of the Access Router through which the gateway is connected to the ground network. The message can contain several instances of the GATEWAY and AR_ADDR TLVs if the node is directly connected to more than one access network. The format of the beginning of a TC message is shown below in Fig. A.6.

The Message TLV Block is followed by an Address TLV Block specifying the 24 bit ICAO identifiers of the node's neighbors. The LINK_METRIC TLV specifies the routing metrics of the directional links to neighboring nodes. This part is identical to what is used in OLSRv2. The structure is shown in Fig. A.7.

Finally, a second address block, whose structure is shown in Fig. A.8, allows nodes to specify their gateway blocking status, as defined in the MDD protocol. The address block contains the list of gateway addresses. The GW_BLOCKINGS TLV specifies the CoS for which this gateway is blocked.

The total length of a TC message depends on a number of parameters: whether the message originator is a gateway, the number of neighbors that it has, and the number of gateways that are blocked for a certain CoS. Again assume that the message originator has N neighbors, and reports the blocking status of M gateways. Then, the length of a TC message sent by an aircraft that is not acting as gateway is $(40 + 9 \times N + 4 \times M)$ Bytes. For an aircraft also acting as satellite gateway, the GATEWAY and AR_ADDR TLVs must be included, leading to a length of $(68 + 9 \times N + 4 \times M)$ Bytes. For a terrestrial gateway, the length is $(61 + 9 \times N)$ Bytes. For N=M=20, the TC messages generated by an aircraft not acting as gateway are 300 B long. If the aircraft is also a

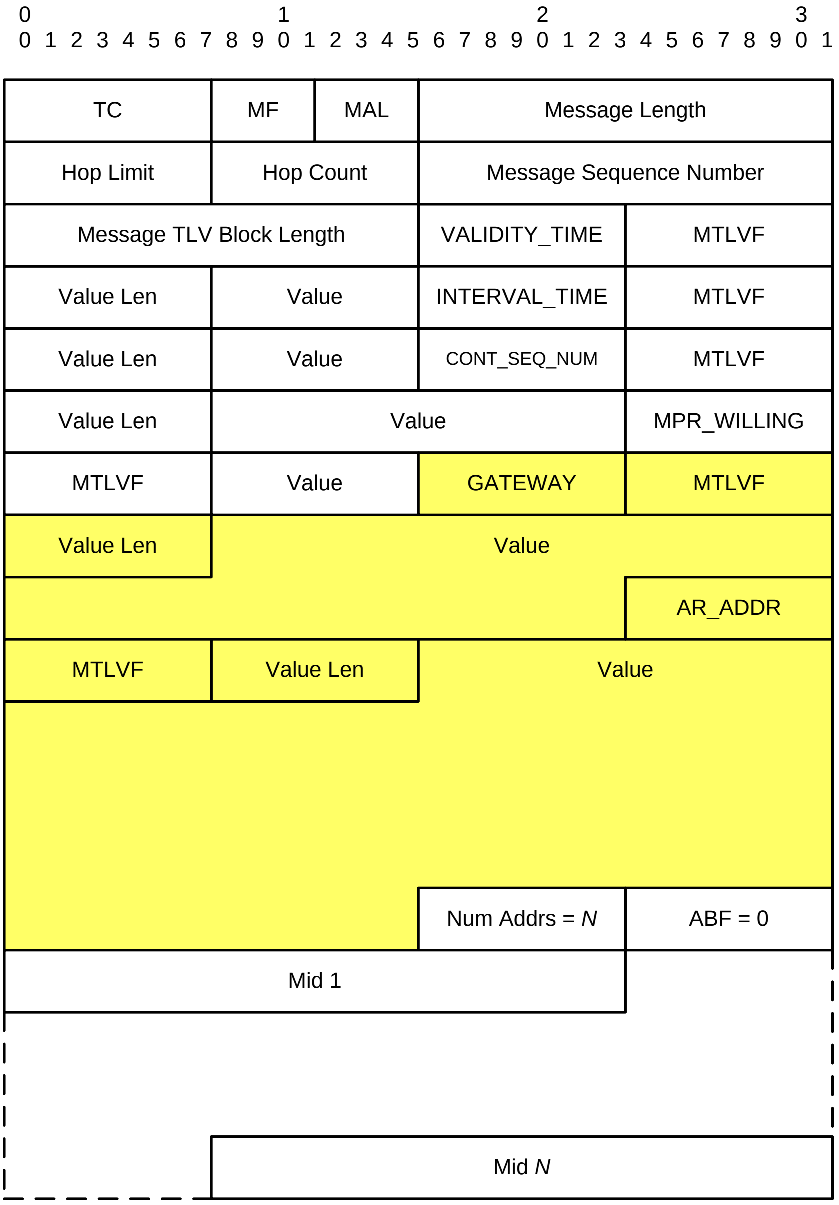


Figure A.6.: MDD TC Message.

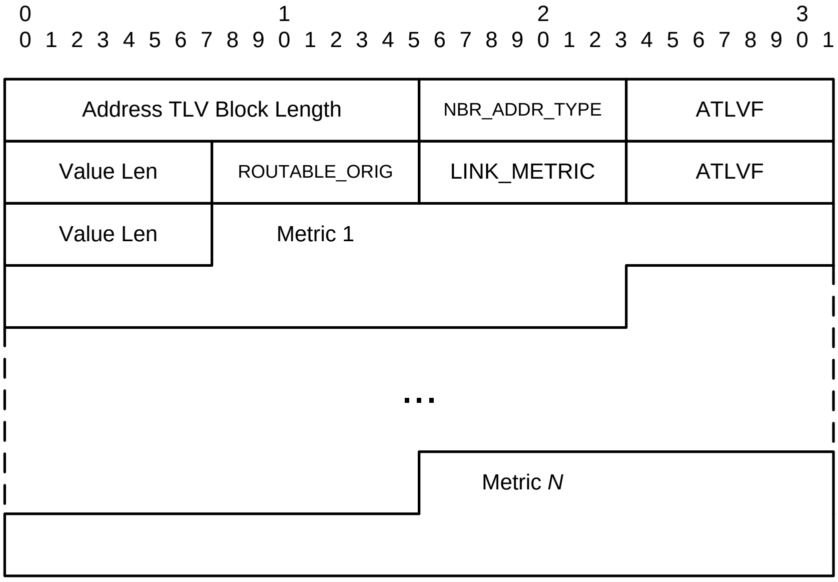


Figure A.7.: MDD TC Address TLV block with link metrics.

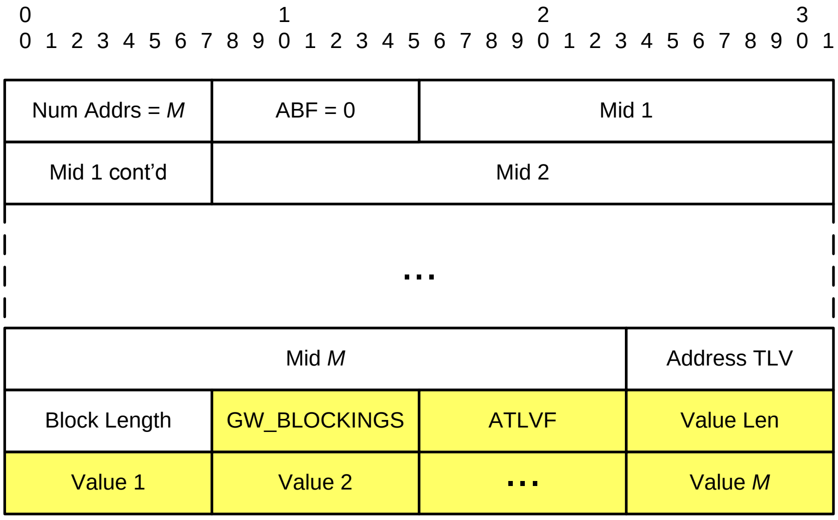


Figure A.8.: MDD TC address block with GW blockings.

gateway, the message size increases to 328 B.

B. Data Traffic Models

In this section, we will define in detail the data traffic models that are used for the realistic performance assessment of the MDD protocol in Chapter 9. It is assumed that airline passengers will typically use the same applications that they would use when accessing the Internet while they are on the ground, either at home, at work, or in a public WiFI hot spot. Here, we consider four different types of applications: HTTP web browsing, file transfer, audio/video streaming, and eMail. We explicitly choose not to consider Voice over IP (VoIP) traffic, since telephone conversations by passengers are likely to be considered a nuisance by other passengers nearby. Nor do we consider interactive online gaming, since this places a very high demand on latency times, or peer to peer file sharing, since this application could produce a significant amount of traffic and would likely be blocked by the network service provider. In the following section, the traffic models for the four relevant applications are discussed in more detail.

B.1. Application Types

The traffic models presented in this section rely on traffic models that have been previously published in research papers or in technical reports of standardization working groups. In particular, the IEEE 802.19 Wireless Coexistence Working Group has published a data traffic model for use in simulations [154]. Similarly, the IEEE 802.16 Wireless Broadband Access WG has published an Evaluation Methodology Document [155] containing traffic models to be used in simulations. Internet use by airline passengers in particular has been characterized by Unger in 2003 [156] and within the ESA project DVB-RM [157]. A model of in flight Internet web browsing on North Atlantic flights has previously been performed by Unger *et al.* in [158], but in less detail than the web browsing models contained in the other documents.

B.1.1. HTTP Web Browsing

HTTP based web browsing is the predominant use of the Internet. The HTTP model is characterized by a number of hierarchies: At the highest level, a user starts a web session, which consists of several web requests for web pages. Between page requests, the user spends some time reading a page. In response to each page request, the web server on the ground answers with one main HTTP object, followed by zero or more embedded objects. The parameters for the modeling of each of these layers are adopted from [155], and are summarized in Table B.1. However, [155] does not specify the number of page requests per session, or the session arrival rates or durations. We use a geometric distribution for the number of pages per session with a mean of 10, as specified in [157].

Note that the mean object size is significantly larger than the Ethernet MTU of 1,500 B. Whenever an application layer object is created that is larger than the MTU, it is fragmented in order to adhere to the MTU restriction. This applies to the file transfer, streaming, and eMail applications which will be defined in the following sections as well.

Parameter	**Model**

Main object size	truncated lognormal distribution
	Mean = 10,710 B,
	σ = 25,032 B,
	Min. = 100 B,
	Max. = 2 MB,

Embedded object size	truncated lognormal distribution
	Mean = 7,758 B,
	σ = 126,168 B,
	Min. = 50 B,
	Max. = 2 MB,

Number of embedded objects per page	truncated Pareto distribution
	Mean = 5.64
	Max. = 53

Pages per session	geometric distribution Mean = 10
Reading time	exponential distribution Mean = 30 s
Session rate	0.00111 s^{-1}

Table B.1.: HTTP traffic model (σ: Standard Deviation).

The page requests sent by the mobile user are modeled as packets with a constant size of 318 Bytes, which is the mean size reported by [159], based on an analysis of web server log files.

HTTP relies on the use of TCP at the transport layer. Therefore, an additional 40 B overhead due to the IPv6 header and 20 B overhead due to the TCP header are added to each packet that is sent. Although we do not include an implementation of TCP in the simulations, the acknowledgements in the upstream direction are modeled as small packets of 60 B, corresponding to the size of the IP and TCP headers. If the ground node does not receive an acknowledgement for a transmitted object within 10 s, the object is retransmitted. After three subsequent timeouts, the entire session is aborted.

Web browsing is an interactive application in which the user expects a reaction to his input within a reasonable amount of time. According to Chen *et al.* [160], the packet delay for web browsing should be kept below 400 ms, although higher values may still be acceptable. Here, we will assume a delay target of 500 ms for HTTP traffic.

B.1.2. File Transfer

The downloading of large files, typically via FTP, is another important application because of the large amount of traffic that is generated for a limited time. We adopt the File Transfer Protocol Model given in [155]. The parameters are summarized in Table B.2. A file is transmitted by the ground node in segments of 1,500 Bytes (including 40 B IPv6 header and 20 B TCP header), to account for the Ethernet Maximum Transmission Unit (MTU).

As HTTP, FTP also uses TCP as its transport layer protocol. Again, we model acknowledgements as packets of size 60 Bytes. As before, if the ground node does not receive an acknowledgement for a transmitted packet within 10 s, it is retransmitted. After three subsequent timeouts, the entire download is aborted.

File transfer applications typically run in the background and do not interact with the user. According to [160], higher delay can be tolerated than for HTTP traffic. We will assume a delay target of 1 s for file transfer packets.

Parameter	**Model**

File size	truncated lognormal distribution
	Mean = 2 MB
	$\sigma = 0.722$ MB
	Max. = 5 MB

Download rate	exponential distribution Mean = $2.78 \cdot 10^{-4}\text{s}^{-1}$

Table B.2.: File transfer traffic model.

B.1.3. Audio / Video Streaming

The importance of audio and video streaming has grown significantly in the last years. It has been reported that the web site YouTube, the currently most popular video sharing web site, alone accounts for about 10% of the total web traffic [161] [162]. Actually, YouTube does not send traffic in a constant unacknowledged stream, but rather sends traffic using HTTP and TCP. To model the size of individual packets, we follow the approach in [157]. The parameters in [157] lead to an average data rate of 192 kbps for a video stream. However, both [161] and [162] report that the average data rate of a YouTube video is around 300–400 kbps. Therefore, we keep the frame rate constant but adapt the parameters of the gamma distribution to provide larger video frames, leading to an average data rate of 384 kbps.

For audio streaming, we use the original packet size distribution of [157]. The average data rate for an audio stream is 128 kbps. The parameters of audio and video streaming sessions are summarized in Table B.3.

The same model of acknowledgements and timeouts applies to the streaming service as to the web browsing and file transfer services.

Parameter	Model
Packet size video	gamma distribution Mean = 1,600 B $\sigma = 80,000$ B
Packet rate video	constant 30 packets / s
Session rate video	exponential distribution Mean = $1.11 \cdot 10^{-4} s^{-1}$
Session duration video	exponential distribution Mean = 300 s
Packet size audio	gamma distribution Mean = 800 B σ = 200 B
Packet rate audio	constant 20 packets / s
Session rate audio	exponential distribution Mean = $4.44 \cdot 10^{-4} s^{-1}$
Session duration audio	exponential distribution Mean = 300 s

Table B.3.: Streaming traffic model.

B.1.4. eMail

We model both the sending and receiving of eMails by passengers. A detailed model for eMail traffic is given in [155]. It is noted that 20% of eMails are sent with attachments, and different distributions for the eMail size are given for the cases with and without attachments. In [157], these two cases are combined into one distribution with the same average mean and variance as in the more detailed model from [155]. Here, we adopt the simplified model of [157]. The parameters of the eMail application are summarized in Table B.4. Note that eMails are received four times more frequently than they are sent. This accounts for the typical user behavior of sending one eMail to multiple recipients. The same model of acknowledgements and timeouts applies to the streaming service as to the web browsing and file transfer services.

We assume that eMail clients are running in the background, and the process of sending or receiving mails does not require direct interaction with the user. We assume a delay target of 10 s for the eMail service, indicating that this service is relatively insensitive to delay.

B.1.5. Ping

In addition to the typical web applications defined in the previous sections, we also consider a simple ”ping-like” application to measure the quality of an aircraft's Internet connection. Each aircraft periodically sends a small packet to the ground node. The ground node immediately responds with a packet of the same size. This allows the

Parameter	Model
Mail size	300 B + lognormal distribution Mean = 60,300 B σ = 185,000 B
Send rate	exponential distribution Mean = 0.0001 s^{-1}
Receive rate	exponential distribution Mean = 0.0004 s^{-1}

Table B.4.: eMail traffic model.

round trip time to be measured and the packet loss rate in downstream and upstream directions to be estimated. The volume of traffic generated by this application is very low, compared to the other applications.

The rate at which ping packets are sent is kept as a simulation parameter. The size of a ping packet is 56 B, including IP overhead. A transport layer protocol is not used. Ping packets are neither acknowledged nor retransmitted in case of loss.

B.2. Aircraft Level Model

The application models in the previous section were from the point of view of a single instance of this application. However, an aircraft aggregates hundreds of potential users. In our simulations, we consider the aircraft as a single node and generate the aggregated traffic from all users on board at this node. The session generation processes of all four application types are modeled as Poisson processes. Thus, the aggregate process of an aircraft with N active users on board is also a Poisson process whose generation rate is simply N times the generation rate of a single user.

Of course, not all passengers on the aircraft will use the Internet during the flight. The number of passengers making use of in flight Internet access has been estimated in [158] for different aircraft types. The analysis was based on the number of seats in first, business, and economy class, and a pricing plan of 35 EUR per 5 MB of data. This results in somewhere between 37 active users for an Airbus A340 and 81 users for an Airbus A380. Since we do not have access to the aircraft type in our flight database, we assume that the number of users on an aircraft follows a Gaussian distribution, whose mean value will be kept as a parameter in our simulations.

For the most part, we do not consider any correlations between the different applications. For example, the HTTP traffic model is not affected by simultaneous eMail traffic. This is realistic, since users typically perform several tasks at the same time and may read and send eMails while browsing the web. The single exception regards the audio and video streaming applications. Each user is assumed to have at most one active audio or video streaming session active at any given time. Thus, the streaming session generation rate for an aircraft is not simply N times the session generation rate of a single user. Rather, it is proportional to N minus the number of currently active streaming sessions.

In general, the amount of traffic generated by the passengers on board an aircraft will also depend on other factors, such as the phase of flight, the time of day, whether meals are being served, or films are being shown. During takeoff and landing phases, the use of electronic devices is prohibited entirely. Here, we focus only on flights in the North Atlantic region. As seen in Fig. 4.5, there are two distinct waves of aircraft within each period of 24 hours. Eastbound aircraft from North America to Europe typically depart in the evening, fly over the Atlantic during the night, and arrive in Europe early in the morning. A majority of passengers on these flights will try to sleep, and data traffic will be relatively low, with peaks expected close to the coast on either side of the ocean. On the other hand, westbound flights from Europe to North America typically depart in the morning, fly during the daytime and arrive in the afternoon (local time). Passengers will be awake most of the time, and use of Internet services will be much more spread out over the duration of the flight. Therefore, these westbound flights are more demanding for the ad hoc network. In the simulations, we will focus on this scenario, and assume the behavior of passengers to remain stable for the entire time that the aircraft is over the North Atlantic.

Fig. B.1 shows the temporal variation of the traffic that is generated by a single aircraft flying over the North Atlantic, according to the traffic model of the previous section. Here, only ten passengers are using the in-flight Internet access. For each minute of flight, the average traffic rate in kbps is plotted. It can be seen that the fluctuations in the downstream direction are much greater than in the uplink. This is caused by the large variation in the size of files and objects downloaded from the Internet, whereas a large part of the upstream traffic is caused by acknowledgements, which do not vary significantly in size. The ratio of downstream to upstream traffic is approximately 10:1, which is significantly higher than the ratio of ca. 5.67:1 reported in [163] for residential traffic. It is reported in [164], that file sharing accounts for over 68% of all upstream traffic in Europe. However, we do not consider file sharing to be a relevant application for a passenger using a mobile device on board an aircraft. Voice over IP telephony is another contributor to upstream traffic in fixed networks, which is not considered here. Therefore, the high asymmetry of our traffic mix appears to be justified.

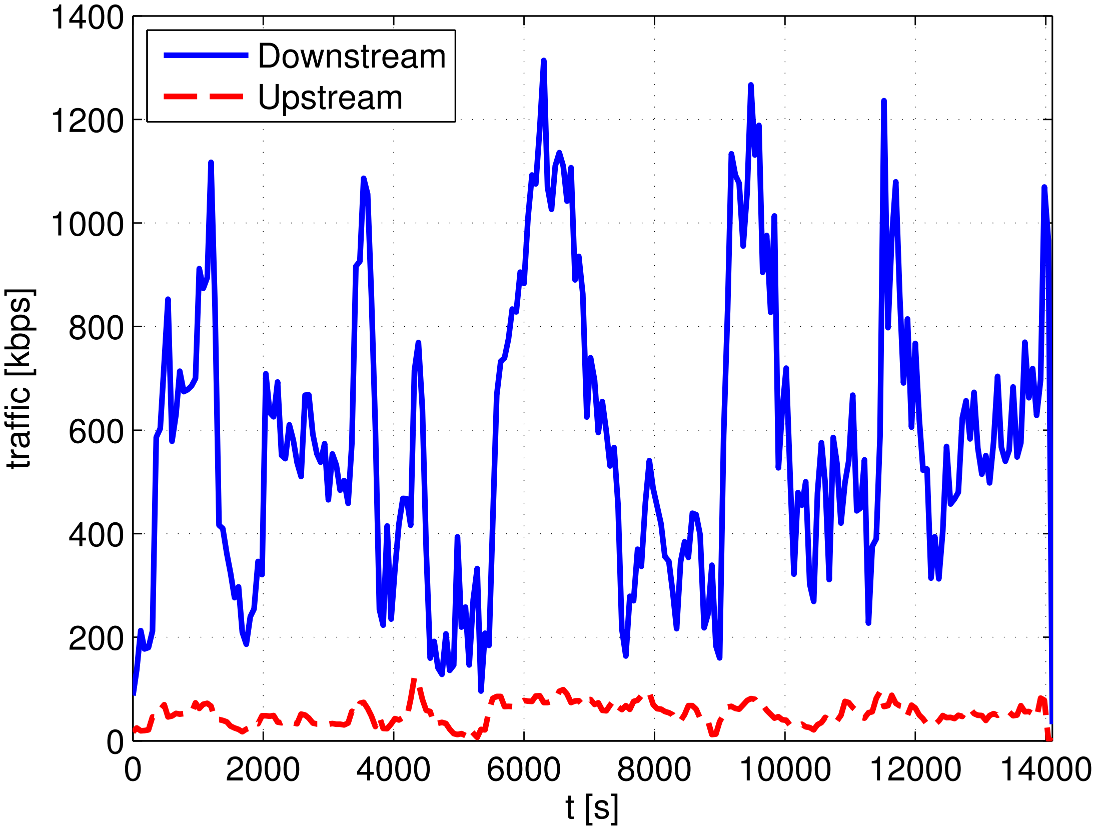


Figure B.1.: Downstream and upstream traffic generated by a single aircraft, $N = 10$.

List of Acronyms

A/A	Air to Air
A/G	Air to Ground
AAHN	Aeronautical Ad Hoc Network
ABF	Address Block Flags
AN	Access Network
ANP	Access Network Prefix
AOC	Airline Operational Communications
APC	Aeronautical Passenger Communications
AR	Access Router
AS	Autonomous System
ATLV	Address Type Length Value
ATLVF	Address Type Length Value Flags
ATN	Aeronautical Telecommunications Network
ATS	Air Traffic Services
AWGN	Additive White Gaussian Noise
B	Byte
BA	Binding Acknowledgement
BGP	Border Gateway Protocol
BPSK	Binary Phase Shift Keying
BU	Binding Update
CBB	Connexion By Boeing
CN	Correspondent Node
CoA	Care of Address
CoS	Class of Service
CSMA	Carrier Sense Multiple Access
DAD	Duplicate Address Detection
DHCPv6	Dynamic Host Configuration Protocol, version 6
DS	Downstream
DSCP	DiffServ Code Point
DSL	Digital Subscriber Line
EUI	Extended Unique Identifier
ESA	European Space Agency
FTP	File Transfer Protocol
GA	Genetic Algorithm
GMT	Greenwich Mean Time
GPP	Gateway Placement Problem
GPS	Global Positioning System

GW	Gateway
HA	Home Agent
HC	Hop Count
HoA	Home Address
HTTP	Hypertext Transfer Protocol
ICAO	International Civil Aviation Organization
IETF	Internet Engineering Task Force
IF	Interface
IGW	Internet Gateway
IGWADV	Internet Gateway Advertisement
IGWSOL	Internet Gateway Solicitation
IP	Internet Protocol
IPv6	Internet Protocol version 6
ISO	International Organization for Standardization
ISP	Internet Service Provider
ITU	International Telecommunications Union
LAN	Local Area Network
LOS	Line of Sight
MAC	Medium Access Control
mAFD	minimize Average Flow Delay
MAL	Message Address Length
MANET	Mobile Ad Hoc Network
MB	Megabyte
MDD	Minimum Downstream Delay
MF	Message Flags
MINLP	Mixed Integer Nonlinear Program
MILP	Mixed Integer Linear Program
MIPv6	Mobile IPv6
MNN	Mobile Network Node
MNP	Mobile Network Prefix
MR	Mobile Router
MTLVF	Message Type Length Value Flags
mWHC	minimize Weighted Hop Count
NAC	North Atlantic Corridor
NAT	North Atlantic Tracks
NEMO	Network Mobility
NEMO BS	Network Mobility Basic Support
nmi	Nautical Mile
OLSRv2	Optimized Link State Routing Protocol Version 2
OIGW	Opportunistic Internet Gateway
OSI	Open Systems Interconnection (network reference model)
OTS	Organized Track System
PDR	Packet Delivery Ratio

PEP	Performance Enhancing Proxy
QAM	Quadrature Amplitude Modulation
QoS	Quality of Service
QPSK	Quaternary Phase Shift Keying
RA	Router Advertisement
SAG	Security Access Gateway
SIR	Signal to Interference Ratio
SINR	Signal to Interference and Noise Ratio
SLA	Service Level Agreement
STDMA	Spatial Time Division Multiple Access
TC	Topology Control
TCP	Transmission Control Protocol
TLV	Type Length Value
TDMA	Time Division Multiple Access
UAV	Unmanned Aerial Vehicle
UCA	Uniform Circular Array
UDP	User Datagram Protocol
US	Upstream
VDL	VHF Digital Link
VHF	Very High Frequency
VoIP	Voice over Internet Protocol
WAN	Wide Area Network
WiFi	IEEE 802.11 Wireless LAN
WMN	Wireless Mesh Network

Bibliography

[1] NAT Tracks. Online. http://jetvision.de/nattracks.shtml.

[2] C. Moser, “Ad Hoc Networking With Beamforming Antennas: Modeling, Visualization, and Connectivity,” Diploma Thesis, Technische Universität München, 2004.

[3] Visualware myconnection server. Online. http://www.myconnectionserver.com.

[4] T. Farrar, “A Bumpy Take-Off? The Future of Connexion-By-Boeing,” TMF Associates, Tech. Rep., June 2005, available online at http://www.tmfassociates.com.

[5] Aircell. Online. http://www.aircell.com.

[6] E. Sakhaee and A. Jamalipour, “The Global In-Flight Internet,” *Selected Areas in Communications, IEEE Journal on*, vol. 24, no. 9, pp. 1748–1757, September 2006.

[7] W. McNary, “Transformational Aircraft Communication Using a Broadband Mesh Network.” Presented at 6th Integrated Communications, Navigation and Surveillance Conference (ICNS), May 2006.

[8] K. Karras, T. Kyritsis, M. Amirfeiz, and S. Baiotti, “Aeronautical Mobile Ad Hoc Networks,” in *Proc. European Wireless (EW)*, June 2008, Prague, Czech Republic.

[9] D. Medina, S. Ayaz, and F. Hoffmann, “Feasibility of an Aeronautical Mobile Ad Hoc Network Over the North Atlantic Corridor,” in *Proc. Fifth IEEE Conference on Sensor, Mesh, and Ad Hoc Communications and Networks (SECON)*, June 2008, San Francisco, USA.

[10] *North Atlantic MNPS Airspace Operations Manual*, ICAO European and North Atlantic Office, August 2008.

[11] R. Nelson and L. Kleinrock, “Spatial TDMA: A collision free multihop channel access protocol,” *IEEE Transactions on Communications*, vol. 33, no. 9, pp. 934–944, September 1985.

[12] R. Rom and M. Sidi, *Multiple Access Protocols: Analysis and Performance.* Springer Verlag, 1990.

[13] Y. Sun, E. M. Belding-Royer, and C. E. Perkins, “Internet Connectivity for Ad Hoc Mobile Networks,” *International Journal of Wireless Information Networks*, vol. 9, no. 2, April 2002.

[14] C. Huang, H. Lee, and Y. Tseng, “A Two-Tier Heterogeneous Mobile Ad hoc Network Architecture and Its Load Balance Routing Problem,” *ACM Mobile Networks and Applications*, vol. 9, no. 4, pp. 379–391, August 2004.

[15] S. Bouk, I. Sasase, S. Ahmed, and N. Javaid, “Gateway Discovery Algorithm Based on Multiple QoS Path Parameters Between Gateway Node and Mobile Node,” *IEEE Journal of Communications Networks*, vol. 14, no. 4, pp. 434–442, August 2012.

[16] A. Festag, H. Füssler, H. Hartenstein, A. Sarma, and R. Schmitz, “Fleetnet: Bringing Car to Car Communication to the Real World,” in *Proc. 11th World Congress on ITS*, October 2004, Nagoya, Japan.

[17] R. Baldessari, A. Festag, W. Zhang, and L. Le, “A MANET-Centric Solution for the Application of NEMO in VANET Using Geographic Routing,” in *Proc. 1st Workshop on Experimental Evaluation and Deployment Experiences on Vehicular Networks (WEEDEV 2008)*, March 2008, Innsbruck, Austria.

[18] S. Ayaz, C. Bauer, C. Kissling, F. Schreckenbach, F. Arnal, C. Baudoin, K. Leconte, M. Ehammer, and T. Gräupl, “Architecture of an IP-based Aeronautical Network,” in *Proc. 9th Integrated Communications, Navigation and Surveillance Conference (ICNS)*, May 2009, Arlington, USA.

[19] G. Fleishman, “In-flight internet: the view from 35,000 feet and three years,” June 2011, Available online at http://arstechnica.com/business/news/2011/06/in-flight-internet-the-view-from-35000-feet-and-three-years.ars, accessed on July 29th, 2011.

[20] A. L. Dul, “Global IP Network Mobility using Border Gateway Protocol (BGP),” The Boeing Company, Tech. Rep., 2006.

[21] T. Farrar, “Lessons from the failure of Connexion-By-Boeing,” TMF Associates, Tech. Rep., November 2006, available online at http://www.tmfassociates.com.

[22] M. Franz, “FlyNet: Details zum Lufthansa-Internet auf Flugreisen,” December 2010, Available online at http://www.netzwelt.de/news/84936-flynet-details-lufthansa-internet-flugreisen.html, accessed on April 27th, 2013.

[23] Row44. Online. http://new.row44.com.

[24] Gogo llc. Online. http://www.gogoair.com/gogo/splash.do.

[25] “cdma2000 High Rate Packet Data Air Interface Specification,” June 2006, 3GPP2 C.S0024-B v1.0.

[26] R. Kingsbury, “Mobile Ad Hoc Networks for Oceanic Aircraft Communications,” Master’s thesis, Massachusetts Institute of Technology, 2009.

[27] TerraMetrics, Inc. Online. http://www.truearth.com/.

[28] Innovata LLC. Online. http://www.innovatallc.com.

[29] H. D. Tu and S. Shimamoto, "A Proposal for High Air-Traffic Oceanic Flight Routes Employing Ad-hoc Networks," in *Proc. IEEE Wireless Communications and Networking Conference (WCNC)*, April 2009, Budapest.

[30] F. Besse, A. Pirovano, F. Garcia, and J. Radzik, "Aeronautical Ad Hoc Networks: a new Datalink for ATM," in *Proc. 9th Eurocontrol Innovative Research Workshop and Exhibition (INO)*, December 2010, Bretigny-sur-Orge, France.

[31] ICAO, "Manual on VHF Digital Link (VDL) Mode 2," ICAO Doc. 9776-AN/970, 2001.

[32] D. Medina, F. Hoffmann, S. Ayaz, and C.-H. Rokitansky, "Topology Characterization of High Density Airspace Aeronautical Ad Hoc Networks," in *Proc. Fifth IEEE International Conference on Mobile Ad Hoc and Sensor Systems (MASS)*, September 2008, Atlanta, USA.

[33] 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 *Proc. 5th International Symposium on Communications Systems, Networks and Digital Signal Processing (CSNDSP)*, July 2006, Patras, Greece.

[34] A. Tiwari, A. Ganguli, A. Sampath, D. S. Anderson, B.-H. Shen, N. Krishnamurthi, J. Yadegar, M. Gerla, and D. Krzysiak, "Mobility Aware Routing for the Airborne Network Backbone," in *Proc. Military Communications Conference (Milcom)*, November 2008, San Diego, USA.

[35] D. Medina, F. Hoffmann, F. Rossetto, and C.-H. Rokitansky, "A Geographic Routing Strategy for North Atlantic In Flight Internet Access Via Airborne Mesh Networking," *IEEE/ACM Transactions on Networking*, vol. 20, no. 4, pp. 1231–1244, August 2012.

[36] K. Sampigethaya, R. Poovendran, and L. Bushnell, "Security of Future eEnabled Aircraft Ad Hoc Networks," in *Proc. AIAA Aviation Technology, Integration and Operations (ATIO)*, September 2008, Anchorage, USA.

[37] M. Ehammer, T. Gräupl, and C.-H. Rokitansky, "Security Considerations for IP based Aeronautical Networks," in *Proc. 27th Digital Avionics Systems Conference (DASC)*, October 2008, St. Paul, USA.

[38] A. Bhadouria, "Airborne Internet - Market & Opportunity," Master Thesis, Massachusetts Institute of Technology, 2007.

[39] N. Campos, "Encouraging Technology Transition Through Value Creation, Capture and Delivery Strategies: The Case of Data Link in the North Atlantic Airspace," Master Thesis, Massachusetts Institute of Technology, 2008.

[40] C. Watkins and C. Dagli, "Agent-Based Model of Aerial Ad Hoc Network Market Potential," in *Proc. 30th Digital Avionics Systems Conference (DASC)*, October 2011, Seattle, USA.

[41] I. F. Akyildiz, X. Wang, and W. Wang, "Wireless Mesh Networks: A Survey," *Computer Networks*, vol. 47, no. 4, pp. 445–487, 2005.

[42] P. Pathak and R. Dutta, "A Survey of Network Design Problems and Joint Design Approaches in wireless Mesh Networks," *IEEE Communication Surveys and Tutorials*, vol. 13, no. 3, pp. 396–428, 2011.

[43] CAR 2 CAR Communication Consortium. Online. http://www.car-to-car.org.

[44] (2007, August) CAR 2 CAR Communication Consortium manifesto, version 1.1. Online. http://www.car-to-car.org.

[45] B. Karp and H. T. Kung, "Greedy Perimeter Stateless Routing for Wireless Networks," in *Proc. Sixth Annual ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom)*, no. 9, August 2000, pp. 243–254, boston.

[46] D. Fokum and V. Frost, "A Survey on Methods for Broadband Internet Access in Trains," *IEEE Communications Surveys and Tutorials*, vol. 12, no. 2, pp. 171–185, April 2010.

[47] D. Pareit, E. V. de Velde, D. Naudts, J. Bergs, J. Keymeulen, I. D. Baere, W. V. Brussel, C. Vangeneugden, P. Hauspie, G. D. Vos, I. Moerman, and C. Blondia, "A Novel Network Architecture for Train-to-Wayside Communication With Quality of Service Over Heterogeneous Wireless Networks," *EURASIP Journal on Wireless Communications and Networking*, 2012, available online at http://dx.doi.org/10.1186/1687-1499-2012-114.

[48] M. M. zu Hörste, T. Strang, and X. Gu, "A Railway Collision Avoidance System exploiting Ad-hoc Inter-Vehicle Communications and GALILEO," in *Proc. 13th World Congress and Exhibition on Intelligent Transportation Systems and Services (ITS 2006)*, October 2006, London, UK.

[49] A. Trivino-Cabrera, S. Singh, E. Casilari-Perez, and F. J. Gonzalez-Canete, "Integration of Mobile Ad Hoc Networks into the Internet without Dedicated Gateways," in *Proc. International Conference on Wireless and Mobile Communications (ICWMC)*, July 2006, Bucharest, Romania.

[50] R. Wakikawa, J. Malinen, C. Perkins, A. Nilsson, and A. Tuominen, "Global Connectivity for IPv6 Mobile Ad Hoc Networks," draft-wakikawa-manet-globalv6-02.txt, November 2002.

[51] P. Ratanchandani and R. Kravets, "A Hybrid Approach to Internet Connectivity for Mobile Ad Hoc Networks," in *Proc. IEEE Wireless Communications and Networking Conference (WCNC)*, March 2003, New Orleans, USA.

[52] P. Ruiz and A. Gomez-Skarmeta, "Adaptive Gateway Discovery Mechanisms to Enhance Internet Connectivity for Mobile Ad Hoc Networks," *Ad Hoc & Sensor Wireless Networks*, vol. 1, no. 1-2, pp. 159–177, March 2005.

[53] J. J. Galvez, P. M. Ruiz, and A. F. Gomez-Skarmeta, "Responsive on-line gateway load balancing for wireless mesh networks," *Ad Hoc Networks*, vol. 10, no. 1, pp. 46–61, 2012.

[54] D. Medhi and K. Ramasamy, *Network Routing: Algorithms, Protocols, and Architectures.* Elsevier, 2007.

[55] H. Balakrishnan, V. N. Padmanabhan, G. Fairhurst, and M. Sooriyabandara, "TCP Performance Implications of Network Path Asymmetry," IETF RFC 3449, Best Current Practice BCP 69, December 2002.

[56] C. Huang, H. Lee, and Y. Tseng, "A Two-Tier Heterogeneous Mobile Ad hoc Network Architecture and Its Load Balance Routing Problem," in *Proc. IEEE Vehicular Technology Conference (VTC Fall)*, October 2003, Orlando, USA.

[57] F. P. Setiawan, S. H. Bouk, and I. Sasase, "An Optimum Multiple Metrics Gateway Selection Mechanism in MANET and Infrastructured Networks Integration," in *Proc. IEEE Wireless Communications and Networking Conference (WCNC)*, April 2008, Las Vegas, USA.

[58] F. Hoffmann and D. Medina, "Optimum Internet Gateway Selection in Ad Hoc Networks," in *Proc. IEEE International Conference on Communications (ICC)*, Dresden, Germany, June 2009.

[59] R. Brännström, C. Ahlund, and A. Zaslavsky, "Maintaining Gateway Connectivity in Multi-hop Ad hoc Networks," in *Proc. IEEE Conference on Local Computer Networks (LCN)*, November 2005, Sydney, Australia.

[60] FAA/EUROCONTROL, "Communications Operating Concept and Requirements for the Future Radio System, version 2," FAA/EUROCONTROL Future Communications Study Report, 2007.

[61] ICAO, "Manual on Aeronautical Telecommunication Network (ATN) using Internet Protocol Suite (IPS) Standards and Protocols," ICAO Doc. 9896, 2010.

[62] ——, "Manual on Detailed Technical Specifications for the Aeronautical Telecommunication Network (ATN) using ISO/OSI Standards and Protocols," ICAO Doc. 9880, 2010.

[63] M. Schnell and S. Scalise, "NEWSKY - NEtWorking the SKY Concept for Civil Aviation," *IEEE Aerospace and Electronic Systems Magazine*, vol. 22, no. 5, 2007.

[64] NEWKSY, a European Commission FP6 Project. Online. http://www.newsky-fp6.eu.

[65] SANDRA - Seamless Aeronautical Networking through integration of Data links, Radios, and Antennas, a European Commission FP7 Project. Online. http://www.sandra.aero.

[66] "NEWSKY Design Document," July 2009, project Deliverable D11, NEWSKY - Networking the Sky, available online at http://www.newsky-fp6.eu.

[67] V. Devarapalli, R. Wakikawa, A. Petrescu, and P. Thubert, "Network Mobility (NEMO) Basic Support Protocol," IETF RFC 3963, January 2005.

[68] "Transmission Control Protocol," IETF RFC 793, September 1981.

[69] H. Balakrishnan, V. Padmanabhan, S. Seshan, and R. Katz, "A Comparison of Methods for Improving TCP Performance Over Wireless Links," *IEEE/ACM Transactions on Networking*, vol. 5, no. 6, pp. 756–769, December 1997.

[70] "Efficient Resource Management Techniques," October 2009, project Deliverable D12, NEWSKY - Networking the Sky, available online at http://www.newsky-fp6.eu.

[71] K. Nichols, S. Blake, F. Baker, and D. Black, "Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers," IETF RFC 2474, December 1998.

[72] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, and W. Weiss, "An Architecture for Differentiated Services," IETF RFC 2475, December 1998.

[73] Y. Rekhter, T. Lee, and S. Hares, "A Border Gateway Protocol 4 (BGP-4)," IETF RFC 4271, January 2006.

[74] W. Eddy, W. Ivancic, and T. Davis, "Network Mobility Route Optimization Requirements for Operational Use in Aeronautics and Space Exploration Mobile Networks," IETF RFC 2255, October 2009.

[75] T. Davis, IETF CIN mailing list, Nov. 6 2012, http://www.ietf.org/mail-archive/web/cin/current/msg00060.html.

[76] S. Thomson and T. Narten, "IPv6 stateless Address Autoconfiguration," IETF RFC 2462, December 1998.

[77] T. Narten, E. Nordmark, and W. Simpson, "Neighbor Discovery for IP Version 6," IETF RFC 2461, December 1998.

[78] S. Ayaz, F. Hoffmann, U. Epple, R. German, and F. Dressler, "Performance Evaluation of Network Mobility Handover over Future Aeronautical Data Link," *Elsevier Computer Communications*, vol. 35, no. 3, pp. 334–343, February 2012.

[79] IETF AUTOCONF WG, Ad-Hoc Network Autoconfiguration (AUTOCONF) Charter. Online. http://www.ietf.org/html.charters/autoconf-charter.html.

[80] D. Johnson, C. Perkins, and J. Arkko, "Mobility Support in IPv6," IETF RFC 3775, June 2004.

[81] C. Bauer and M. Zitterbart, "A Survey of Protocols to Support IP Mobility in Aeronautical Communications," *IEEE Communications Surveys and Tutorials*, vol. 13, no. 4, pp. 642–657, 2011.

[82] R. Droms, J. Bound, B. Volz, T. Lemon, C. Perkins, and M. Carney, "Dynamic Host Configuration Protocol for IPv6 (DHCPv6)," IETF RFC 3315, July 2003.

[83] F. Hoffmann, D. Medina, and A. Wolisz, "Protocol Architecture Analysis for Internet Connectivity in Aeronautical Ad Hoc Networks," in *Proc. 29th Digital Avionics Systems Conference (DASC)*, October 2010, Salt Lake City, USA.

[84] "Network Mobility Support Goals and Requirements," IETF RFC 4886, July 2007.

[85] R. Wakikawa, V. Devarapalli, G. Tsirtsis, T. Ernst, and K. Nagami, "Multiple Care-of Addresses Registration," IETF RFC 5648, October 2009.

[86] G. Tsirtsis, H. Soliman, N. Montavont, G. Giaretta, and K. Kuladinithi, "Flow Bindings in Mobile IPv6 and Network Mobility NEMO Basic Support," IETF RFC 6089, January 2011.

[87] O. Younis and S. Fahmy, "Constraint-Based Routing in the Internet: Basic Principles and Recent Research," *IEEE Communications Surveys and Tutorials*, vol. 5, no. 1, pp. 2–13, 2003.

[88] Nippon Telegraph and Telephone Corporation (NTT), "SLA of Global IP Network," Available online at http://www.eu.ntt.com/en/products/global-network/transit/sla-of-global-ip-network.html, accessed on May 4th, 2013.

[89] A. Shahriar, M. Atiquzzaman, and W. Ivancic, "Route Optimization in Network Mobility: Solutions, Classification, Comparison, and Future Research Directions," *IEEE Communications Surveys and Tutorials*, vol. 12, no. 1, pp. 24–38, January 2010.

[90] B. McCarthy, M. Jakeman, C. Edwards, and P. Thubert, "Protocols to Efficiently Support Nested NEMO (NEMO+)," in *(Proc. ACM MobiArch 2008)*, August 2008, Seattle, USA.

[91] M.-S. Jeong and J.-T. Park, "Hierarchical Mobile Network Routing: Route Optimization and Micro-mobility Support for NEMO," in *Proc. International Conference on Embedded and Ubiquitous Computing (ICEUC)*, August 2004, aizu-wakamatsu, Japan.

[92] T. Clausen, E. Baccelli, and R. Wakikawa, "Route Optimization in Nested Mobile Networks NEMO Using OLSR," in *Proc. IASTED International Conference on Networks and Communication Systems (NCS)*, April 2005, Krabi, Thailand.

[93] R. Wakikawa, P. Thubert, T. Boot, J. Bound, and B. McCarthy, "Problem Statement and Requirements for MANEMO," draft-wakikawa-manemo-problem-statement-01, expired, July 2007.

[94] I. Chakeres and C. Perkins, "Dynamic MANET On-demand (DYMO) Routing, draft-ietf-manet-dymo-10," IETF Internet Draft, work in progress, July 2007.

[95] T. Clausen and P. Jacquet, "Optimized Link State Routing Protocol (OLSR)," IETF RFC 3626, October 2003.

[96] T. Clausen, C. Dearlove, and P. Jacquet, "The Optimized Link State Routing Protocol version 2," IETF Internet Draft draft-ietf-manet-olsrv2-13, work in progress, October 2011.

[97] INRIA, "OOLSR," Available online at http://hipercom.inria.fr/OOLSR/, accessed on May 4th, 2013.

[98] Naval Research Laboratory, "The NRL OLSR Routing Protocol Implementation," Available online at http://cs.itd.nrl.navy.mil/work/olsr/, accessed on May 4th, 2013.

[99] R. Baldessari, A. Festag, and J. Abeille, "NEMO Meets VANET: A Deployability Analysis of Network Mobility in Vehicular Communication," in *Proc. International Conference on ITS Telecommunications (ITST)*, June 2007, Sophia Antipolis, France.

[100] J. Rohrer, A. Jabbar, E. Perrins, and J. Sterbenz, "Cross-Layer Architectural Framework For Highly-Mobile Multihop Airborne Telemetry Networks," in *Proc. Military Communications Conference (MILCOM)*, San Diego, USA, November 2008.

[101] E. Cetinkaya and J. Sterbenz, "Aeronautical Gateways: Supporting TCP/IP-based Devices and Applications over Modern Telemetry Networks," in *Proc. International Telemetering Conference (ITC)*, Las Vegas, USA, October 2009.

[102] E. Haas, "Aeronautical Channel Modeling," *IEEE Transactions on Vehicular Technology*, vol. 51, no. 2, pp. 254–264, 2002.

[103] J. D. Parsons, *The Mobile Radio Propagation Channel.* John Wiley & Sons, 2000.

[104] J. Allred, A. B. Hasan, S. Panichsakul, W. Pisano, P. Gray, J. Huang, R. Han, D. Lawrence, and K. Mohseni, "SensorFlock: An Airborne Wireless Sensor Network of Micro-Air Vehicles," in *Proc. ACM Conference on Embedded Networked Sensor Systems (SenSys)*, November 2007, Sydney, Australia.

[105] M. Walter, S. Gligorevic, T. Detert, and M. Schnell, "UHF/VHF Air-to-Air Propagation Measurements," in *4th European Conference on Antennas and Propagation (EuCAP)*, April 2010, Barcelona, Spain.

[106] "Propagation Curves for Aeronautical Mobile and Radionavigation Services Using the VHF, UHF and SHF Bands," Tech. Rep., 1986, recommendation ITU-R P.528-2.

[107] *Eurocontrol Long Term Forecast: Flight Movements 2008 - 2030*, EUROCONTROL STATFOR, November 2008.

[108] C. Bettstetter, "On the Minimum Node Degree and Connectivity of a Wireless Multihop Network," in *Proc. 3rd ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc)*, June 2002, Lausanne, Switzerland.

[109] *L-DACS1 System Definition Proposal*, EUROCONTROL, 2009.

[110] "Analysis of Requirements and Technologies," April 2006, project Deliverable D4.1, ANASTASIA - Airborne New and Advanced Satellite Techniques and Technologies in a System Integrated Approach.

[111] M. Joham, W. Utschick, and J. Nossek, "Linear Transmit Processing in MIMO Communications Systems," *IEEE Transactions on Signal Processing*, vol. 53, no. 8, pp. 2700–2712, 2005.

[112] J. G. Proakis, *Digital Communications*, 4th ed. McGraw Hill International Edition, 2001.

[113] International Telecommunication Union (ITU), "Final Acts - WRC-07," 2007, Geneva, Switzerland, 2007.

[114] IEEE Computer Society, "IEEE Standard for Information technology - Telecommunications and information exchange between systems - Local and metropolitan area networks - Specific requirements: Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications," March 2012, IEEE Std 802.11-2012.

[115] F. Tobagi and L. Kleinrock, "Packet Switching in Radio Channels: Part II - The Hidden Terminal Problem in Carrier Sense Multiple Access Modes and the Busy Tone Solution," *IEEE Transactions on Communications*, vol. 23, no. 12, pp. 1417–1433, 1975.

[116] A. Jayasuriya, S. Perreau, A. Dadej, and S. Gordon, “Hidden vs. exposed terminal problem in ad hoc networks,” in *Proc. Australian Telecommunication Networks & Applications Conference (ATNAC)*, December 2004, Sydney, Australia.

[117] R. Ramanathan, J. Redi, C. Santivanez, D. Wiggins, and S. Polit, “Ad Hoc Networking With Directional Antennas: A Complete System Solution,” *IEEE Journal on Selected Areas in Communications*, vol. 23, no. 3, pp. 496–506, 2005.

[118] J. Grönkvist, “Interference-Based Scheduling in Spatial Reuse TDMA,” Ph.D. dissertation, KTH Stockholm, 2005.

[119] L. Bao and J. J. Garcia-Luna-Aceves, “Receiver-Oriented Multiple Access in Ad Hoc Networks with Directional Antennas,” *Wireless Networks*, vol. 11, no. 1-2, pp. 67–79, 2005.

[120] J. B. Cain, T. Billhartz, L. Foore, E. Althouse, and J. Schlorff, “A Link Scheduling and Ad Hoc Networking Approach Using Directional Antennas,” in *Proc. Military Communications Conference (Milcom)*, October 2003, Monterey, USA.

[121] A. D. Gore and A. Karandikar, “Link Scheduling Algorithms fir Wireless Mesh Networks,” *IEEE Comm. Surveys and Tutorials*, vol. 13, no. 2, pp. 258–273, 2011.

[122] P. Djukic and S. Valaee, “Delay Aware Link Scheduling for Multi-Hop TDMA Wireless Networks,” *IEEE/ACM Transactions on Networking*, vol. 17, no. 3, pp. 870–883, June 2009.

[123] T. Halonen, J. Romero, and J. Melero, *GSM, GPRS and EDGE Performance: Evolution Towards 3G/UMTS.* John Wiley & Sons, 2003.

[124] F. Hoffmann, D. Medina, and A. Wolisz, “Joint Routing and Scheduling in Mobile Aeronautical Ad Hoc Networks,” *IEEE Transactions on Vehicular Technology*, vol. 62, no. 6, pp. 2700–2712, 2013.

[125] J. Padhye, V. Firoiu, D. Towsley, and J. Kurose, “Modeling TCP Throughput: A Simple Model and its Empirical Validation,” in *Proc. ACM SIGCOMM*, September 1998, Vancouver, Canada.

[126] L. Badia, A. Botta, and L. Lenzini, “A Genetic Approach to Joint Routing and Link Scheduling for Wireless Mesh Networks,” *Ad Hoc Networks*, vol. 7, no. 4, pp. 654–664, June 2009.

[127] J. Luo, C. Rosenberg, and A. Girard, “Engineering Wireless Mesh Networks: Joint Scheduling, Routing, Power Control, and Rate Adaptation,” *Networking, IEEE/ACM Transactions on*, vol. 18, no. 5, pp. 1387–1400, October 2010.

[128] A. Capone, I. Filippini, and F. Martignon, “Joint Routing and Scheduling Optimization in Wireless Mesh Networks with Directional Antennas,” in *Proc. IEEE International Conference on Communications (ICC)*, May 2008, pp. 2951–2957, Beijing, China.

[129] H. Livingstone, H. Nakayama, T. Matsuda, X. Shen, and N. Kato, "Gateway Selection in Multi-Hop Wireless Networks Using Route and Link Optimization," in *Proc. IEEE Global Communications Conference (Globecom)*, Miami, December 2010.

[130] Y. I. B. Aoun, R. Boutaba and G. Kenward, "Gateway Placement Optimization in Wireless Mesh Networks With QoS Constraints," *IEEE Journal of Selected Areas in Communications*, vol. 24, no. 11, pp. 2127–2136, 2006.

[131] K. Papadaki and V. Friderikos, "Gateway Selection and Routing in Wireless Mesh Networks," *Computer Networks*, vol. 54, no. 2, pp. 319–329, February 2010.

[132] V. Targon, B. Sanso, and A. Capone, "The Joint Gateway Placement and Spatial Reuse Problem in Wireless Mesh Networks," *Computer Networks*, vol. 54, no. 2, pp. 231–240, February 2010.

[133] S. Boyd and L. Vandenberghe, *Convex Optimization.* Cambridge University Press, 2004.

[134] O. Goussevskaia, Y. Oswald, and R. Wattenhofer, "Complexity in Geometric SINR," in *MobiHoc '07: Proceedings of the 8th ACM international symposium on Mobile ad hoc networking and computing*, Montreal, Canada, September 2007.

[135] M. Pioro and D. Medhi, *Routing, Flow, and Capacity Design in Communication and Computer Networks.* Elsevier, 2004.

[136] X. Yu and M. Gen, *Introduction to Evolutionary Algorithms.* Springer Verlag, 2010.

[137] J.-H. Lee, B.-J. Han, H.-J. Lim, Y.-D. Kim, N. Saxena, and T.-M. Chung, "Optimizing of Access Point Allocation Using Genetic Algorithmic Approach for Smart Home Environments," *The Computer Journal*, vol. 52, no. 8, pp. 938–949, 2009.

[138] R. Pries, D. Staehle, B. Staehle, and P. Tran-Gia, "On Optimization of Wireless Mesh Networks using Genetic Algorithms," *International Journal of Advances in Internet Technology*, vol. 3, no. 1, pp. 13–28, July 2010.

[139] M. Sanna and M. Muroni, "Optimization of Non-Convex Multiband Cooperative Sensing With Genetic Algorithms," *IEEE Journal of Selected Topics in Signal Processing*, vol. 5, no. 1, pp. 87–96, 2011.

[140] B. Miller and D. Goldberg, "Genetic Algorithms, Selection Schemes, and the Varying Effects of Noise," *Evolutionary Computation*, vol. 4, no. 2, pp. 113–131, 1996.

[141] ——, "Genetic Algorithms, Tournament Selection, and the Effects of Noise," *Complex Systems*, vol. 9, pp. 193–212, 1995.

[142] K. Zielinski, P. Weitkemper, R. Laur, and K.-D. Kammeyer, "Examination of Stopping Criteria for Differential Evolution based on a Power Allocation Problem," in *10th International Conference on Optimization of Electrical and Electronic Equipment*, May 2006, Brasov, Romania.

[143] J. H. Holland, *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control and Artificial Intelligence.* Cambridge, MA, USA: MIT Press, 1992.

[144] M. G. M. Musnjak, "Using a Set of Elite Inidviduals in a Genetic Algorithm," in *Proc. 26th International Conference on Information Technology Interfaces)*, June 2004, Cavtat, Croatia.

[145] *Lingo User's Manual*, Lindo Systems, Chicago.

[146] OMNeT++ Discrete Event Simulation System. Online. Available online at http://www.omnetpp.org.

[147] T. Clausen, C. Dearlove, and J. Dean, "Mobile Ad Hoc Network (MANET) Neighborhood Discovery Protocol (NHDP)," IETF RFC 6130, April 2011.

[148] J. McQuillen, I. Richer, and E. Rosen, "The New Routing Algorithm for the ARPANET," *IEEE Transactions on Communications*, vol. 28, no. 5, pp. 711–719, 1980.

[149] R. Draves, J. Padhye, and B. Zill, "Routing in Multi-Radio, Multi-Hop Wireless Mesh Networks," in *Proc. ACM MobiCom*, Philadelphia, USA, September 2004.

[150] F. Hoffmann, D. Medina, and A. Wolisz, "Optimization of Routing and Gateway Allocation in Aeronautical Ad Hoc Networks Using Genetic Algorithms," in *Proc. 7th International Wireless Communications and Mobile Computing Conference (IWCMC)*, Istanbul, Turkey, July 2011.

[151] F. Hoffmann, D. Medina, C. Bauer, and S. Ayaz, "FACTS – An OMNeT++ Based Simulator for Aeronautical Communications," in *Proc. 1st International OMNeT++ Workshop at the ACM SIMUTools Conference*, March 2008, Marseille, France.

[152] "Greenland Connect Submarine Cable," available online at http://www.telepost.gl/en-US/GreenlandConnect/Soekablet/Sider/Forside.aspx.

[153] T. Clausen, C. Dearlove, J. Dean, and C. Adjih, "Generalized Mobile Ad Hoc Network (MANET) Packet / Message Format," IETF RFC 5444, February 2009.

[154] "Data traffic model for wireless coexistance in us 3.65 ghz band simulation, document ieee p802.19-08/0001," January 2008.

[155] "Ieee 802.16m evaluation methodology document (emd), document ieee 802.16m-08/004r5," January 2009.

[156] P. Unger, “Multiservice traffic analysis for global aeronautical communications over broadband satellite systems,” Diplomarbeit, TU Ilmenau, September 2003.

[157] “Tn2: System scenario definition and traffic characterization,” November 2005, project Deliverable, ESA Resource Management Using Adaptive Fade Mitigation Techniques in DVB-RCS Multi-Beam Systems.

[158] P. Unger, L. Battaglia, M. Werner, and M. Holzbock, “Dynamic Behavior of Air-Com Internet Users on Long-Haul North Atlantic Flights,” in *Proc. IEEE International Conference on Communications (ICC)*, June 2004, Paris, France.

[159] J. J. Lee and M. Gupta, “A new traffic model for current user web browsing behavior,” Intel Corp., Tech. Rep., 2007.

[160] Y. Chen, T. Farley, and N. Ye, “QoS Requirements of Network Applications on the Internet,” *Information Knowledge Systems Management*, vol. 4, no. 1, pp. 55–76, January 2004.

[161] P. Gill, M. Arlitt, Z. Li, and A. Mahanti, “YouTube Traffic Characterization: A View From the Edge,” in *Proc. ACM SIGCOMM Conference on Internet Measurement (IMC)*, October 2007, San Diego, USA.

[162] L. Plissonneau, T. En-Najjary, and G. Urvoy-Keller, “Revisiting Web Traffic From a DSL Provider Perspective: the Case of YouTube,” in *Proc. 19th ITC Specialist Seminar on Network Usage and Traffic*, October 2008, Berlin, Germany.

[163] G. Maier, A. Feldmann, V. Paxson, and M. Allman, “On Dominant Characteristics of Residential Broadband Internet Traffic,” in *Proc. ACM SIGCOMM Conference on Internet Measurement (IMC)*, November 2009, Chicago, USA.

[164] “Global Internet Phenomena Spotlight - Europe, Fixed Access, Spring 2011,” May 2011, Sandvine Inc. ULC.