

Performance Evaluation of Handoff in Mobile IPv6 Networks: The Case of Safety-Critical Systems with NIMBLE Platform for Mobility

Lucas Tognoli Munhoz, Daniel F. Pigatto^(✉),
and Kalinka Regina Lucas Jaquie Castelo Branco

Universidade de São Paulo (USP), Avenida Trabalhador São-carlense, 400,
São Carlos, SP 13566-590, Brazil
lucas.munhoz@usp.br, danielpigatto@gmail.com

Abstract. In real-time systems, when failures of any nature come to happen, high-value assets are put at risk and in some cases even human lives. Due to this high criticality, developing improvements to these systems in order to increase their safety becomes a topic of high importance. This research reviews handoff procedures in IPv6 networks to help identify those with lower latency and low packet loss during the operation, aiming to increase the efficiency of general communication in these systems. For this, some handoff techniques found in the literature are simulated and, from the obtained results, performance comparisons are presented following specific statistical techniques to evaluate the performance of computational systems. Finally, a discussion on the application of these techniques in critical embedded systems is presented, specially unmanned vehicles.

Keywords: Handover · Handoff · Mobile IPv6 · Safety-critical vehicles · Unmanned vehicles

1 Introduction

In the last few years there has been a growing interest in Unmanned Vehicles (UV) and Unmanned Systems, leading to the co-existence of many different types of vehicles e.g. aerial, terrestrial and aquatic vehicles. These systems should be integrated into the airspace, on public roads or on aquatic environments following specific laws and requirements of each scenario, mainly related to safety. Thus, it is essential that communications carefully meet mobility and time requirements, increasing the system overall capabilities and, consequently, allowing these vehicles to be certified and integrated into their operation space.

UVs are becoming highly connected for cooperation purposes (e.g. distributed tasks), and also to the Internet for real-time services provision (e.g. IoT and cloud-based applications). Thus, a tendency on these areas is the use of IPv6 protocol that allows an exclusive address on the Internet and provides mobility approaches suitable for UVs applications.

Therefore, the study of handoff process on IPv6 is an important research topic for applications that depend on wireless mobile networks, mainly due to the fact that it is a critical aspect regarding connection quality that is impacted by packet losses during handoff transition process. Thus, it is necessary to analyse how worth the handoff process is considering network conditions (there might exist a better network to choose to connect to) and the criticality of an ongoing operation. Moreover, when it comes to high critical applications, handoff might be a failure point to be considered.

The goal of this paper is to compare two handoff algorithms in IPv6 networks, especially investigating the impact of mobility support. Our contributions extend to embedded systems connected by IPv6 mobile networks, providing data and comparisons that can be used by developers and researchers. We analyse aspects such as run time in each step of the process and factors that influence the decision-making algorithm, such as signal strength and data transmission rate.

The overall organisation of this paper is: Sect. 2 presents the state of the art with related works that identify the gap addressed by this paper; Sect. 3 details the Internet Protocol version 6; Sect. 4 introduces a data communication architecture and emphasises the NIMBLE platform for mobility which was proposed for safety-critical unmanned systems; Sect. 5 provides an explanation about techniques used for results treatment; Sect. 6 demonstrates the simulations experimental setup; Sect. 7 presents results and discussions; Sect. 8 connects results and discussions to the UVs field; and, finally, Sect. 9 concludes the paper.

2 Background

The advantage of wireless mesh networks is seen when self-organisation is coupled with seamless handover to provide continuity of service to the users. Handover (sometimes seen as handoff) is common in cellular networks, where mobile stations frequently move out of the coverage area of one cell tower and into that of a neighbouring tower [7].

In [15], authors measured latencies of all handoff process stages. They concluded that: (i) the wireless network adapter used in both the Mobile Unit (MU) and the Access Points (APs) directly influences the handoff latency; (ii) different MUs treat sequences of messages slightly different; and (iii) the search phase takes about 90% of the handoff process time. However, authors did not explore real situations even with simulated results, which might change the perception if contextualised in some areas.

In [27], authors discussed the effects of handoff in data transmission. Similarly to [15], they concluded that the network interface affects latency, and the longest stage of the process is search. They have also noted that a MU can keep receiving data from previously connected AP even during the search phase. Finally, authors concluded that the behavior of handoff process depends not only on the hardware used (network adapter), but also on data stream, e.g., if MU is sending or receiving packets.

In [4], authors discussed latency issues in Mobile IPv6 and its derivative PMIPv6, highlighting the difficulty of using such handoff mechanisms in real-time systems. To solve this problem, authors proposed a new handoff scheme for PMIPv6 network that reduces latency and solves the loss problem.

We have identified that there is a lack of comparisons between IPv6 protocol with and without mobility on the context of unmanned aerial vehicles (UAVs). Thus, this paper will carry out a discussion on how handoff process impacts IPv6 networks from the point of view of unmanned aircraft systems (UAS) in general, which includes not only the UAV, but all the supporting systems.

Next section will address important concepts review on IPv6.

3 Internet Protocol Version 6

In 1990s, the Internet Engineering Task Force (IETF) started developing the successor to the IPv4 protocol. The motivation for that was the fact that 32-bit IP address was beginning to be used up as more devices were connected to the Internet with unique IP addresses. To respond for the need of a large IP address space, a new IP protocol was developed. The Internet Protocol version 6 (IPv6) has also improved some aspects based on accumulated operational experience with IPv4 [12]. There was considerable debate about when the IPv4 addresses would be completely allocated. Many approaches were developed since then trying to prolong the use of IPv4 as much as possible. However, we have been currently seeing major companies and institutions moving to IPv6, but the transition may still take some time to be fully completed.

According to [12], the most important changes in IPv6 datagram (Fig. 1) format are:

- **Expanded addressing capabilities.** IPv6 increases the size of the IP address from 32 to 128 bits, ensuring that there will be enough IP addresses for nowadays applications. As a comparison matter, every grain of sand on the planet can be IP-addressable.
- **A streamlined 40-byte header.** The 40-byte fixed-length header allows for faster processing of the IP datagram. A new encoding of options allows for more flexible options processing.
- **Flow labelling and priority.** IPv6 has an elusive definition of a flow e.g. audio and video transmission might likely be treated as a flow. On the contrary, the more traditional applications might not be treated as flows. Thus, the designers of IPv6 foresee the eventual need to be able to differentiate among the flows.

The following fields are defined in IPv6 [12] (see Fig. 1):

- **Version.** This 4-bit field identifies the IP version number.
- **Traffic class.** This 8-bit field is similar in spirit to the TOS in IPv4.
- **Flow label.** This 20-bit field is used to identify a flow of datagrams.

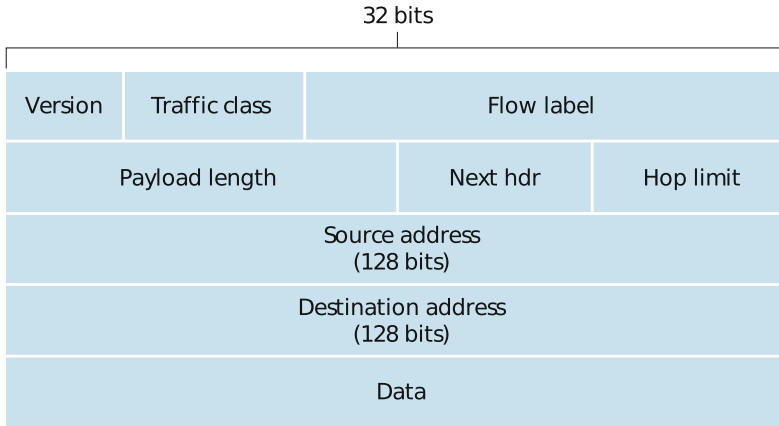


Fig. 1. IPv6 datagram. This figure was adapted from [12].

- **Payload length.** This 16-bit value is treated as an unsigned integer giving the number of bytes in the IPv6 datagram following the fixed-length, 40-byte datagram header.
- **Next header.** This field identifies the protocol to which the contents (data field) of this datagram will be delivered (for example, to TCP or UDP).
- **Hop limit.** The contents of this field are decremented by one by each router that forwards the datagram. If the hop limit count reaches zero, the datagram is discarded.
- **Source and destination addresses.** The various formats of the IPv6 128-bit address are described in RFC 4291 [8].
- **Data.** This is the payload portion of the IPv6 datagram. When the datagram reaches its destination, the payload will be removed from the IP datagram and passed on to the protocol specified in the next header field.

3.1 Mobile IPv6

Mobile IPv6 (MIPv6) describes the protocol operations needed to keep a mobile node connected to the Internet during its handover from one access router to another. These operations involve movement detection, IP address configuration, and location update (RFC 4068) [11]. Mobile IPv6 is an IETF standard that has added the roaming capabilities of mobile nodes in IPv6 network (RFC 3775) [10].

The major benefit of this standard is that the mobile nodes (as IPv6 nodes) change their point-of-attachment to the IPv6 Internet without changing their IP address [6]. Moreover, MIPv6 is an update to Mobile IP (RFC 6275) [18], designed to authenticate mobile devices using IPv6 addresses.

In traditional IP network routing, addresses represent a topology. The routing mechanisms were made under the assumption that each network node has always the same entry point to the Internet and that each IP address identifies the link

to which it is connected. MIPv6 allows a mobile node to transparently maintain connections as it moves from a network edge to another.

4 HAMSTER Architecture

The Healthy, Mobility and Security-based Data Communication Architecture (HAMSTER) is divided into three main versions according to the most common types of unmanned vehicles (UV): aerial, ground and aquatic. Moreover, two special platforms are defined: the first one is intended to control security and safety aspects under all architecture versions, and the second one aims at mobility aspects and external communications. Figure 2 presents an overview of HAMSTER hierarchical organisation.

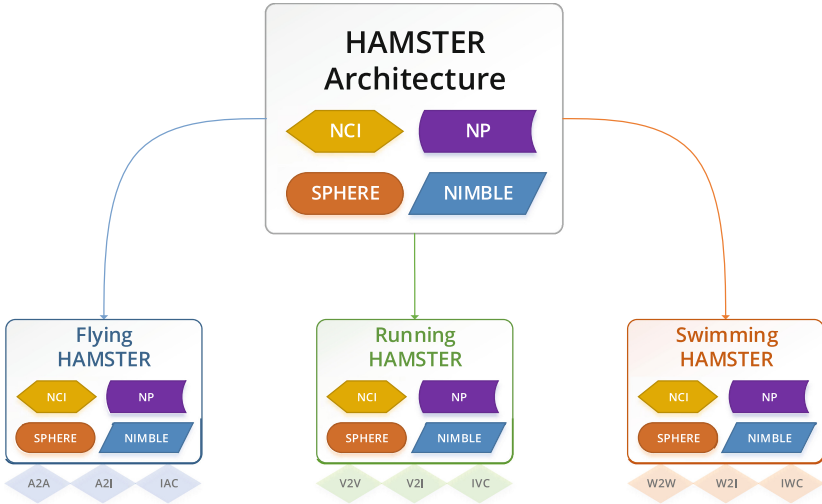


Fig. 2. HAMSTER versions and the specific modules for mobility (NIMBLE) and safety and security (SPHERE).

Flying HAMSTER deals exclusively with the aerial segment. It was defined based on specific characteristics and requirements of unmanned aerial vehicles (UAV) and unmanned aircraft systems (UAS). Flying HAMSTER deals specifically with internal airplane communication (IAC), airplane-to-airplane communication (A2A) and airplane-to-infrastructure communication (A2I).

The main applications of UAVs are related to agricultural and environmental monitoring, safety, military and civil defense. The aircraft is usually able to capture images for processing relevant information about a specific field, which may contribute to improve productivity. There are several cases where they might be applied in environmental and borders monitoring, or even applied as aerial

sensors in networks for disaster management [20] and multiple UAV applications [3, 13, 14, 24, 28].

Running HAMSTER deals specifically with vehicles on terrestrial segment. It was defined based on specific characteristics and requirements of unmanned ground vehicles (UGV) and unmanned ground systems (UGS). Running HAMSTER treats internal vehicle communication (IVC), vehicle-to-vehicle communication (V2V) and vehicle-to-infrastructure communication (V2I).

The objective of ground vehicles may vary from driver support in possible dangerous situations with the intention of preventing road accidents, to autonomous driving with no human intervention, which could be used in urban traffic, agriculture, industry and safety applications [29]. The sensor fusion technique is used for integration of multiple sensors such as cameras, digital compasses, and GPS, allowing the vehicle to become autonomous in both urban and rural areas [25].

Swimming HAMSTER was designed for vehicles that operate on aquatic environments. It was defined based on specific characteristics and requirements of unmanned surface vehicles (UGV), unmanned undersea vehicles (UUV) and unmanned water vehicles systems (UWS). Swimming HAMSTER is composed by internal water vehicle communication (IWC), water vehicle-to-water vehicle communication (W2W) and water vehicle-to-infrastructure communication (W2I).

The aquatic vehicles have been used for various tasks, especially those related to monitoring of oil exploration and maintenance of hydropower. The current challenges for these vehicles go beyond autonomy, integrating other areas with the distributed and embedded systems, such as computer networks, artificial intelligence, software engineering, electrical, mechanical and mechatronics engineering, among others. The multiple vehicles tasks are also challenging [1, 30].

Every HAMSTER version has four platforms: **NCI**, the network criticality index that measures a node's criticality within a network; **NP**, the navigation phases platform that aims at energy saving techniques; **SPHERE**, the security and safety platform; and **NIMBLE**, the mobility platform [19]. Next subsection will briefly introduce NIMBLE, the HAMSTER's platform for mobility which is the focus of this paper.

4.1 NIMBLE: Mobility Platform

Be it an infrastructured network or a mobile ad hoc network, mobility is an important requirement for UVs. In the air, mobility can be even more challenging than in roads and in the ocean, but all of them have somehow a demand for mobility. Naturally, ad hoc is the most challenging operating mode, since it does not rely on a fixed centralised access point and nodes move at high speeds and (sometimes) in unpredicted paths. Although seeming not so challenging, infrastructure-based communications may demand mobility approaches as well. That way, NIMBLE was conceived with the view to encompass mobility in M2X communications on HAMSTER architecture. It stands for Native MoBiLity platform for unmanned systEms.

Communication is a crucial aspect of, and one of the biggest challenges in, the design of multiple-vehicle systems [3,5]. In the simplest scenario, all vehicles are directly connected to a common infrastructure, and this can act as an intermediary for all communications among them. However, this strategy has several problems. Firstly, each vehicle must be equipped with expensive and complex hardware in order to perform the long-distance communication with the control station or satellite. Secondly, many factors may compromise communication reliability such as changing environmental conditions, the high mobility of vehicles, different terrain topologies, or obstacles. Finally, the typical use of a ground control station (GCS) to provide the communication infrastructure limits the mission target locations to the GCS coverage area, since beyond that vehicles disconnect from the network, and become unreachable.

The implementation of an ad hoc network connecting all vehicles is one of the most feasible alternatives to infrastructure-based communication. An ad hoc network is composed by nodes that also act as routers, forming a temporary network with no fixed topology or centralised administration [23]. This approach increases the mission target area, since communications among vehicles and the GCS can be routed through other vehicles in a series of hops. Also, even if there is no connection to a GCS, the nodes can form an ad hoc network to share information or work in cooperation.

Ad hoc networks are classified according to their implementation, utilization, communication and mission objectives. If the nodes that compose an ad hoc network are mobile, the network is classified as MANET (Mobile Ad hoc NETWORK). For vehicle-specific applications, MANETs are sub-divided into UANET (Underwater Ad hoc NETWORK) for aquatic vehicles, VANET (Vehicular Ad hoc NETWORK) for terrestrial vehicles, or FANET (Flying Ad hoc NETWORK) for aerial vehicles [2,22], as illustrated by Fig. 3.

Each type of vehicular network faces different, unique challenges: for instance, a UANET must deal with an underwater transmission medium, and VANETs often encounter unexpected road obstacles. However, it has been recognized that FANETs have to address more challenging issues than other ad hoc networks [2,22], because of the following specific characteristics:

- **Higher node mobility.** FANET nodes typically have higher mobility than those in other types of MANET. As a result, a FANET's network topology can change more frequently, which increases the overhead caused by connecting and routing operations.
- **Multiple connections.** In many applications, the nodes in FANETs collect environmental data and then retransmit it to the control station, similarly to wireless sensor networks [21]. Therefore, FANETs have to manage multiple communications between UAVs and monitoring stations, as well as providing support to peer-to-peer connections among UAVs.
- **Very low node density.** Typical distances among nodes in FANETs are usually longer than in MANETs and VANETs clapper2007unmanned; thus, the communication range in FANETs must also be greater than in other

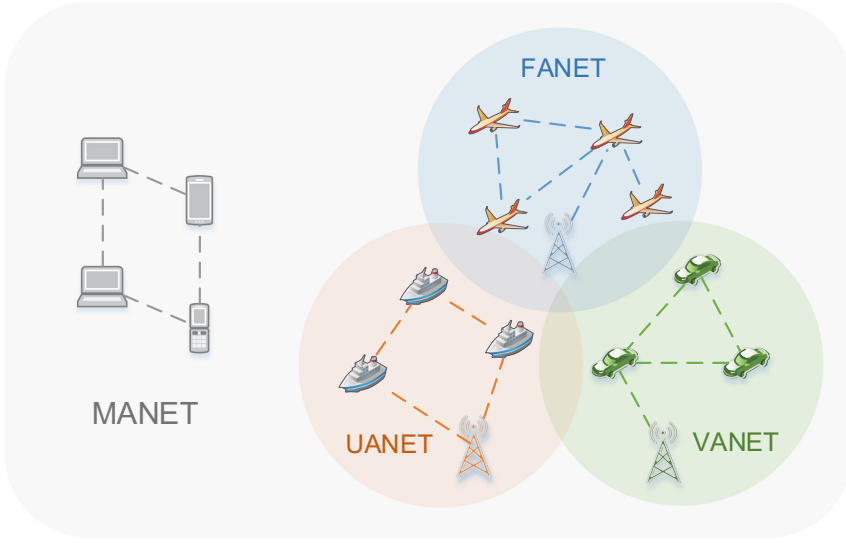


Fig. 3. Relationships between different types of mobile ad hoc networks (MANET): underwater ad hoc networks (UANET), vehicular ad hoc networks (VANET) and flying ad hoc networks (FANET).

networks. This imposes more demanding requirements for radio links and other hardware elements.

- **Heterogeneity.** UAV systems may include heterogeneous sensors, and each of them may require different strategies for data distribution.
- **Obstacles.** Due to the higher node mobility, obstacles may randomly block links among UAVs, which must be addressed in order to provide different temporary communication paths, avoiding the disconnection of nodes.

External communications on HAMSTER architecture are dealt by NIMBLE platform. Figure 4 illustrates NIMBLE’s submodules: ADHOC and INFRA.

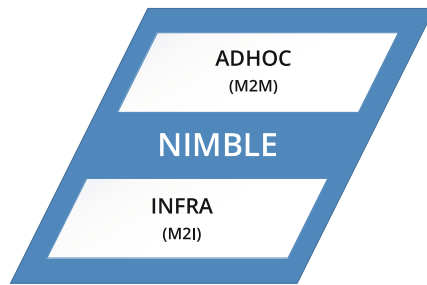


Fig. 4. NIMBLE mobility platform is composed by ADHOC and INFRA submodules for external communications.

M2M communications, including mostly MANET and derivative networks, are managed by ADHOC. On the other hand, INFRA is aimed at the management of infrastructured communications e.g. satellites and GCS (in HAMSTER case, identified as CAGE). These submodules also concentrate efforts on mobility models improvements.

The separation into two different modules provides the advantage of allowing better approaches to each external communication. For instance, while communicating with an infrastructure, INFRA submodule will most certainly need to transmit with higher signal strength due to longer distances from the UV to the infrastructured element. On the contrary, while communicating with other UVs, ADHOC should use appropriate ad hoc routing protocols for better message delivery.

This paper tests specifically NIMBLE's INFRA handoff process focused on avionics. Next section presents the methodology.

5 Methodology

This section presents the simulator used for experiments execution, and the criteria and results collection methods.

5.1 OMNeT++

OMNeT++ is an extensible, modular, component-based C++ simulation library and framework, primarily for building network simulators [17]. Due to its general structure, it can be used for analysis and study of different problems, including:

1. Modelling wired/wireless communication networks;
2. Protocols;
3. Architecture validation and distributed systems modelling;
4. Performance evaluation of complex systems;
5. Modeling and simulation of any system in which the approach to discrete event is appropriate, and entities can be conveniently mapped.

OMNeT++ provides a component architecture for models. Components (modules) are programmed in C++, then assembled into larger components and models using a high-level language (NED). Reusability of models comes for free. OMNeT++ has extensive GUI support, and due to its modular architecture, the simulation kernel (and models) can be embedded easily into applications [17].

In addition to OMNeT++, this project uses INET [9]. INET Framework is an open-source model library for the OMNeT++ simulation environment. It provides protocols, agents and other models for researchers and students working with communication networks. INET is especially useful when designing and validating new protocols, or exploring new or exotic scenarios.

INET contains models for the Internet stack (TCP, UDP, IPv4, IPv6, OSPF, BGP, etc.), wired and wireless link layer protocols (Ethernet, PPP, IEEE 802.11, etc.), support for mobility, MANET protocols, DiffServ, MPLS with LDP and RSVP-TE signalling, several application models, and many other protocols and components. Several other simulation frameworks take INET as a base, and extend it into specific directions, such as vehicular networks, overlay/peer-to-peer networks, or LTE [9].

OMNeT++ simulator was chosen to run the simulations that will be discussed in Sect. 6. We have implemented two simulations using INET Framework and OMNeT++ to compare distinct handoff processes: with and without mobility.

Nest subsection provides details on criteria and results collection.

5.2 Criteria and Results Collection

First, we have investigated a way of making simulations as similar as possible. For instance, the speed at which the MU would move, plus routers distance and coverage area must be the same in both simulations, in order not to compromise results. Criteria used for comparison between algorithms are: process time and signal strength.

The handoff run time criterion is perhaps the most complex and that best defines the difference between processes. For some algorithms, it is highly important for a handoff process to be performed as fast as possible due to the fact that the MU needs to disconnect from previous AP to start sending requests to the next one. Packet losses may occur due to disconnected time [26]. For example, if there is a real-time video transmission, packets will likely be lost causing issues.

The handoff process can be divided into two phases [16], as shown in Fig. 5. For further analysis, the run time is measured for each step separately, allowing a step by step comparison. The first phase is defined as search. The MU scans for nearby available APs. Once found, the AP in better conditions according to the algorithm criteria, is the one chosen by MU for connection establishment.

Search phase can be divided into two steps: scanning delay and classification delay. Still, in scanning delay, it is possible to scan each channel separately. In practice, the initial event of the search phase is Beacon Timeout, i.e., the MU has received at least three Beacon packets that were considered as noise, representing a connection lost state, which leads to a new search for APs. In the next event, the MU disassociates from AP1 and sends an internal command to tune the channel 0, starting a new search. The scanning delay ends in the event when the MU radio can tune channel 0, and then start the effective search.

The next step consists on scanning each channel in order to find an AP. This scan takes place as follows: the MU tunes to a certain channel and listens for incoming messages from any AP in that frequency, should it be a Beacon or a Router Advertisement. Such listening process lasts a preset time. After that, the MU is aware whether there is any AP operating in that frequency that can be include on the list of new AP candidate. If a channel has been scanned and no

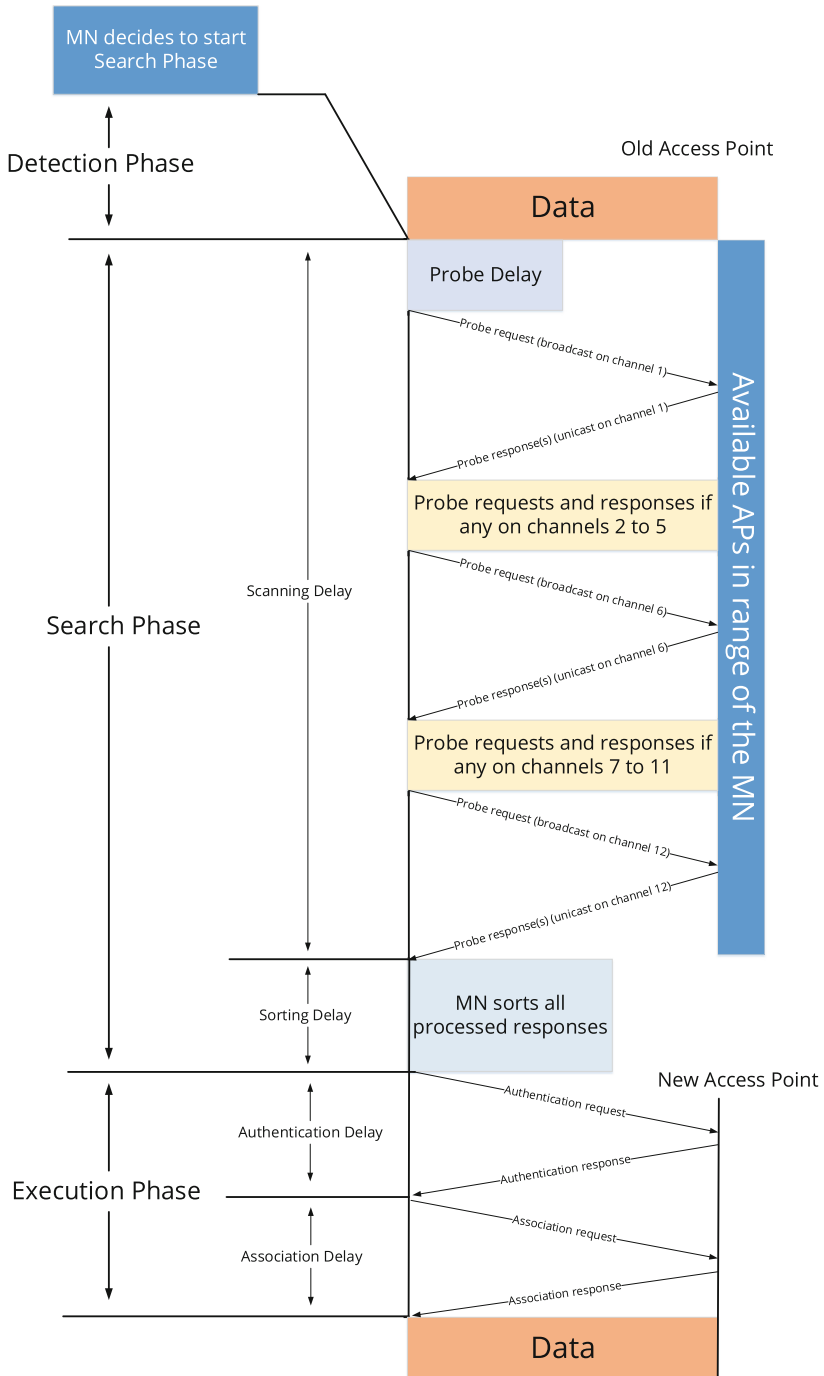


Fig. 5. Phases of handoff process. This figure was adapted from [16].

message was detected, the radio warns the management layer that the time limit is over and no AP was found. A command for channel change is also triggered.

After the minimum scanning time and the positive identification of another device operating on the same frequency, the MU sends a Probe Request message, asking for information about how the AP works, e.g. transmission type, transmission rate. The AP then responds with a Probe Response message. Later, the MU updates the known AP list and proceeds with the scanning process. This process is repeated for all four possible channels. After that, it terminates the scanning delay process. For comparison matters, search times will be measured on each channel, apart from the initial scan phase.

The last stage of the search phase is the classification delay, in which the MU classifies all found APs. A limitation imposed by our simulator is seen in this phase. Although it simulates these events as concomitant processes, they are not so in reality. Thus, the classification delay cannot be measured in details, but its overall time is included in results and does not imply problems for our analyses.

The next phase of handoff process is called Execution Phase. Here, the MU exchanges messages with the AP in order to associate with it. Some of these messages are the authentication request, followed by its response, request and association response.

The received signal strength indicator (RSSI) of an AP is one of the decisive factors of a handoff process. When making the decision of which AP the MU will connect to, there must be a consideration about the best signal, which can potentially avoid new handoffs.

Changing the decision-making policy can be a tricky task, since it is not always possible to get access to codes that implement the direct process in real-time operating systems. In our experiments, such change is not possible due to simulator restrictions. However, we will carry out tests that help checking the influence of some factors in the decision-making algorithm used by OMNeT++ simulator in this paper.

To analyse this by a different perspective, a new scenario has been proposed: the inclusion of a new AP to our simulation that addresses a scenario with no mobility. Thus, an area covered by three different APs was created. By changing parameters of any AP, it would be possible to discuss about which aspect was taken into consideration for the decision making.

The aspects we aim to analyse with this research are signal strength and data transfer rate. Basically, the positioning of all three routers forms an equilateral triangle, as shown in Fig. 6. In each test, the parameter of one of the APs were modified in order to decrease one of its features. Afterward, scenarios in Table 4 were proposed.

Next section presents the experimental setup for experiments.

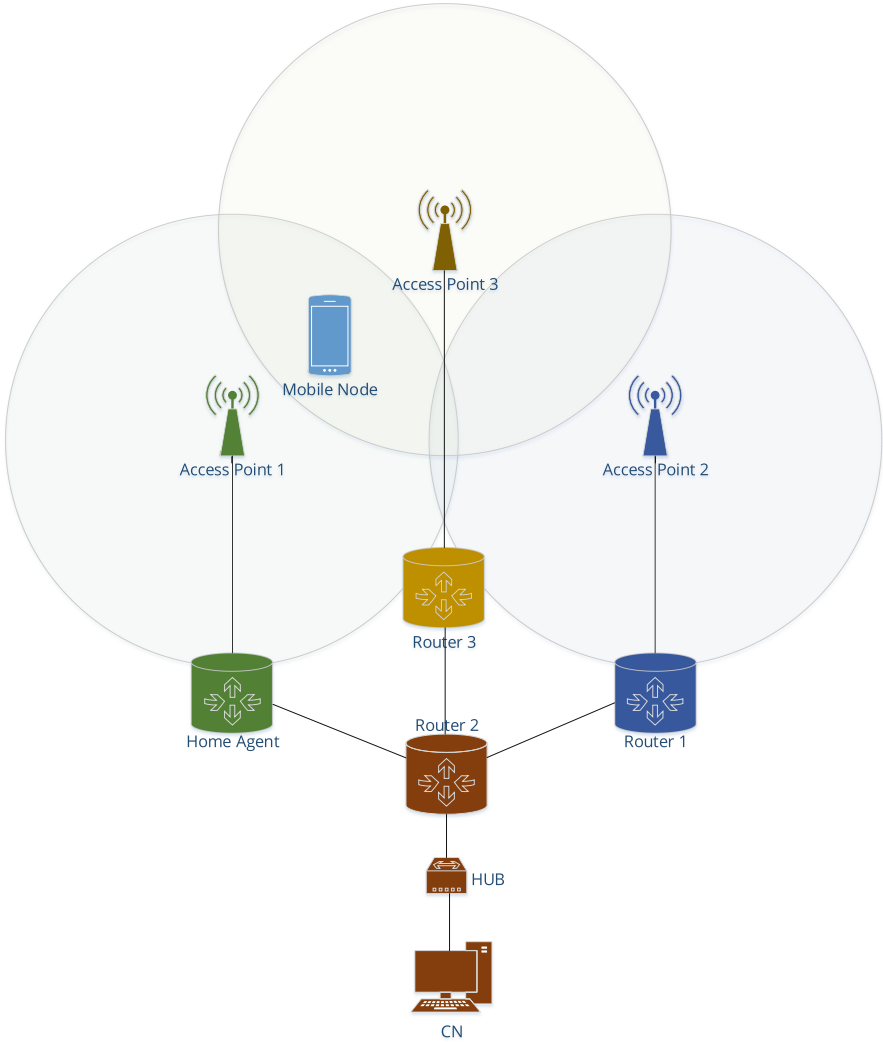


Fig. 6. Network scheme adopted for testing handoff decision making algorithm.

6 Experimental Setup

6.1 IPv6 with Mobility

The “IPv6 with Mobility” simulation (*MIPv6Network.ned*) is part of the INET simulations package [9]. This simulation allows the analysis of handoff process on an IPv6 network with mobility.

The MIPv6Network is composed by five basic elements: mobile unit, access points, routers, hub and fixed host. In addition, there are two elements in charge of configuration. Figure 7 illustrates the scenario.

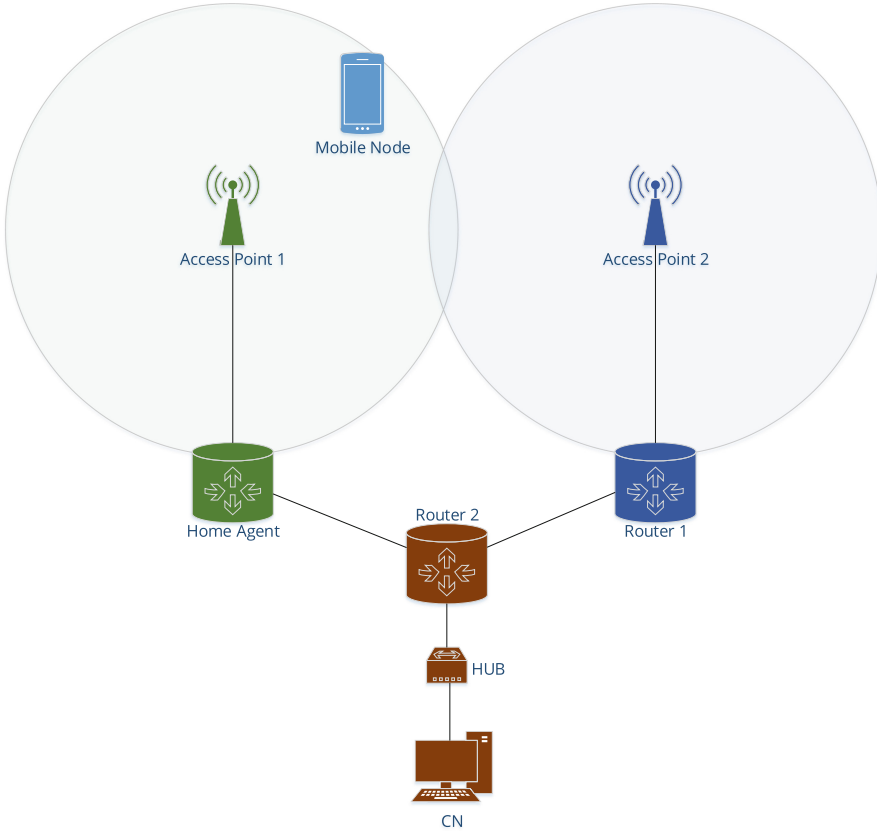


Fig. 7. Graphical representation of MIPv6Network.

The MU is composed by a *WirelessHost6* module. It is the MU, which is always connected to an AP and moves among coverage areas. The APs used in the simulation are both equal. They provide radio for routers i.e. wireless communication between MU and Fixed Host.

The first router module, identified as *Router6*, acts as Home Agent. This router captures packets for the MU should it be connected or not. The Router called *R_1* is the Foreign Agent, i.e. it distributes the packets generated by the MU while outside the original network, and acts receiving the tunnelled packets destined to MU. The other Router along with the Hub represents the Internet, providing the tunnel between Routers that provide connectivity.

6.2 IPv6 Without Mobility

We have developed the “IPv6 without Mobility” simulation (*handoff-ipv6.ned*) in order to meet the requirements we are investigating. This IPv6 network performs experiments of handoff without mobility. As the MU moves from one network to

another, considering there is no support for mobility, it completely loses its link with the native address, including a subsequent IP address change.

Figure 8 illustrates the simulation. One can point out that the network is formed by a *WirelessHost6* MU that supports MIPv6, but it has such a feature disabled. As seen in “IPv6 with Mobility” simulation, two APs are used for wireless communication between routers. They are both independent routers that provide distinct links. This is a way of guaranteeing that no mobility is used in our experiments.

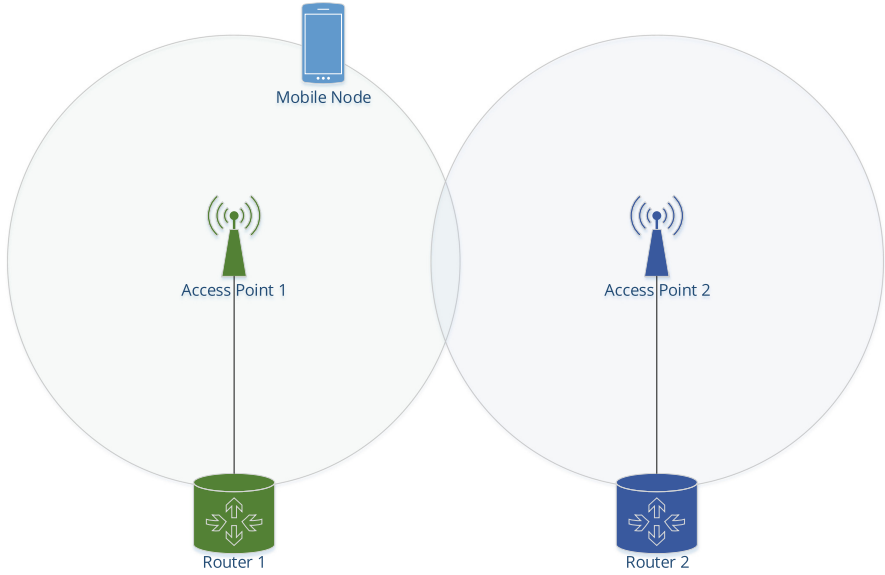


Fig. 8. Graphical representation of handoff_ipv6 network.

In addition to the network elements, two configuration modules were used: *ChannelControl* (for managing wireless communications, including distances and possible interferences), and *flatNetworkConfigurator6* (which manages addresses and routing tables).

Next section will present results and discussions.

7 Results and Discussions

7.1 Run Time Results

As mentioned in “Methodology” section, the time needed for a MU to either interpret three lost *Beacons* and decide to start a new search, cannot be calculated with our simulator. Thus, it will not be considered for comparison purposes, since it does not affect the results integrity.

Table 1 presents the measured run times of total scanning of frequency channels during handoff process in both simulations. The factor which was more decisive on run time was the number of scans that the MU had to perform before finding an available AP. This difference is noticeable in scenarios where there is intersection between the signal coverage areas (Scenarios 1 and 2) than those with no intersection (Scenarios 2 and 3). If there is no intersection, as the MU loses the first AP signal, it initiates a search on all channels. If this time is not enough for the MU to enter the area of another AP, it will have to perform other complete search. Repeating these searches is a very costly process that also compromises the efficiency of the handoff process, since each additional search takes 1.25 s. In a scenario of high transmission rates, e.g. in a video/audio streaming, a new search might fully compromise the service, as it would mean 1.25 s disconnected from the AP, meaning a loss of thousands of packets.

Table 1. Total scanning time for the tested scenarios.

	Scenarios	1st exp.	2nd exp.	3rd exp.	Scanning repetitions
IPv6 without mobility	1	1 s 400 ms	1 s 400 ms	1 s 400 ms	1
	2	1 s 400 ms	1 s 400 ms	1 s 400 ms	1
	3	2 s 650 ms	2 s 650 ms	2 s 650 ms	2
	4	6 s 400 ms	6 s 400 ms	6 s 400 ms	5
IPv6 without mobility	1	1 s 400 ms	1 s 400 ms	1 s 400 ms	1
	62	1 s 400 ms	1 s 400 ms	1 s 400 ms	1
	3	2 s 650 ms	2 s 650 ms	2 s 650 ms	2
	4	6 s 400 ms	6 s 400 ms	6 s 400 ms	3

Still, in Table 1, one can point out that the velocity of a mobile node directly impacts the handoff time on a situation in which there is no intersection of cells. As the MU moves faster through the area without signal, it needs to perform fewer scans and therefore connect faster to the new AP.

In the second phase of handoff process, it is performed authentication and association of the MU with the AP. The authentication delay times are shown in Table 2 and the association delay times are shown in Table 3.

The next handoff phase that impacts the total run time is the authentication/association process. In the case of authentication, there is a difference in microseconds among all tested scenarios. In both scenarios, the IPv6 protocol without mobility performed better than MIPv6 (Scenarios 2 and 3). On the other hand, MIPv6 performed better in the other two. In association phase, MIPv6 took advantage in three scenarios, not being the best option only in the case of travel speed and no cells intersection. This result indicates that MIPv6 can facilitate the MU association process.

Finally, Table 4 shows the total handoff process run time in all eight tested scenarios.

Table 2. Authentication delay times for the tested scenarios.

	Scenarios	1st exp.	2nd exp.	Mean time
IPv6 without mobility	1	2 ms 375 us 943 ns	2 ms 375 us 943 ns	2 ms 375 us 943 ns
	2	2 ms 376 us 59 us	2 ms 376 us 59 us	2 ms 376 us 59 us
	3	2 ms 376 us 188 us	2 ms 376 us 188 us	2ms 376 us 188 us
	4	2 ms 403 us 214 us	2 ms 403 us 214 us	2 ms 403 us 214 us
IPv6 without mobility	1	2 ms 349 us 88 us	2 ms 349 us 88 us	2 ms 349 us 88 us
	2	2 ms 403 us 226 us	2 ms 403 us 226 us	2 ms 403 us 226 us
	3	2 ms 402 us 582 us	2 ms 402 us 582 us	2ms 402 us 582 us
	4	2 ms 402 us 609 us	2 ms 402 us 609 us	2 ms 402 us 609 us

Table 3. Association delay times for the tested scenarios.

	Scenarios	1st exp.	2nd exp.	Mean time
IPv6 without mobility	1	1 ms 523 us 971 us	1 ms 523 us 971 us	1 ms 523 us 971 us
	2	1 ms 506 us 29 us	1 ms 506 us 29 us	1 ms 506 us 29 us
	3	1 ms 497 us 94 us	1 ms 497 us 94 us	1 ms 497 us 94 us
	4	1 ms 507 us 70 us	1 ms 507 us 70 us	1 ms 507 us 70 us
IPv6 without mobility	1	1 ms 479 us 44 us	1 ms 479 us 44 us	1 ms 479 us 44 us
	2	1 ms 497 us 113 us	1 ms 497 us 113 us	1 ms 497 us 113 us
	3	1 ms 496 us 805 us	1 ms 496 us 805 us	1 ms 496 us 805 us
	4	1 ms 523 us 387 us	1 ms 523 us 387 us	1 ms 523 us 387 us

One can observe in Table 4 the predominance of scan time if compared to authentication and association delays, which go from 300 times and reach up to 2000 times higher. It is important to point out that IPv6 without mobility performed better and faster in three scenarios.

As a final remark, a few microseconds (less than 100 μ s in all cases) do not considerably affect the amount of received bits. For example, in a 2Mbps

Table 4. Total handoff time for the tested scenarios.

	Scenarios	Total time
IPv6 without mobility	1	1 s 403 ms 899 us 971 us
	2	1 s 403 ms 882 us 88 us
	3	2 s 653 ms 873 us 282 us
	4	6 s 403 ms 910 us 278 us
IPv6 without mobility	1	1 s 403 ms 828 us 132 us
	2	1 s 403 ms 900 us 339 us
	3	2 s 653 ms 899 us 207 us
	4	6 s 403 ms 925 us 996 us

transmission, around 2 bits per millisecond (ms) are sent — or 0.002 bits per microsecond (μ s). Differences of 100 μ s result in the loss of 0.2 bit, which is extremely small considering the amount of transmitted information.

7.2 Decision Making

In the first scenario for decision making evaluation, both APs are configured to emit signals with the same power. However, AP1 operated with half the AP2's data transmission rate. At the end of scan, the handoff algorithm chose to connect to AP1, as shown in Fig. 9.

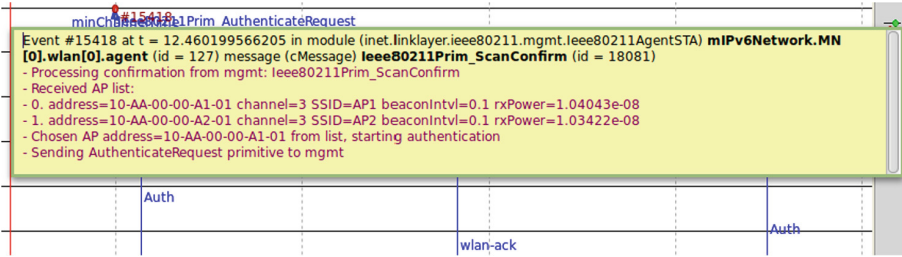


Fig. 9. Handoff classification phase on Scenario 1.

The other analysed scenario contains two APs with the same data transmission rate, however different signal strength. The algorithm has chosen to associate with AP1, as shown in Fig. 10.

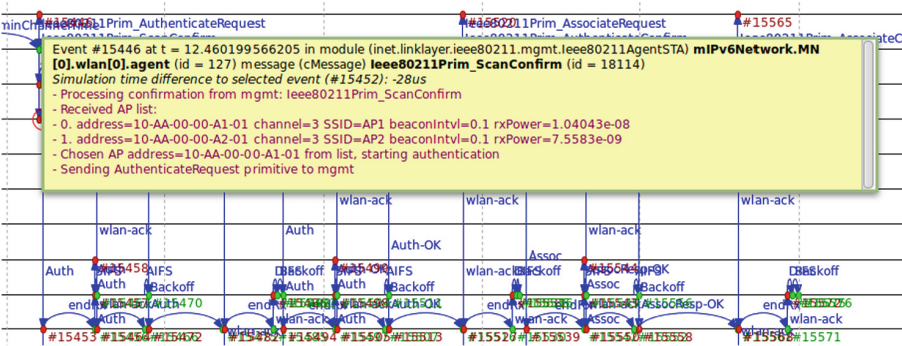


Fig. 10. Handoff classification phase on Scenario 2.

Regarding tests on Decision Making, the main conclusion is that the predominant factor in choosing an AP to connect was the signal strength received

by MU at the scanning process. In the test performed in Scenario 1, in which both APs operate with equal signal strength, and AP1 operates with half the transfer rate of AP2, AP1 was chosen. This can be explained by the difficulty in obtaining an overall symmetry of the problem: as much as the APs are equally distributed, in the meantime while receiving the packet with information about each AP signal strength, the MU has already moved. That leads to slight differences in results, as shown in Fig. 9. The predominance of signal quality explains the choice for AP1, despite the low transmission rate.

The experiment in the second scenario only confirms the conclusion aforementioned. When signal strength was far different, the chosen AP was always the one that offered a higher signal quality.

It is important to clarify that tests conducted to observe decision making were carried out in only one of the protocols mentioned in this paper, the “IPv6 with Mobility”. This choice was made due to the fact that the decision making algorithm is linked to the simulator itself, not to the protocols. Thus, for any handoff process in any OMNeT++ simulation, the decision making algorithm was the same. The purpose of these tests were to get more information about this algorithm and complement researches on handoff processes, and also to draw conclusions that could help UAVs development and research. Such a discussion will be presented in next section.

8 A Discussion on Safety-Critical Systems

As it could be seen in presented and discussed results, there is a small advantage of “IPv6 without Mobility” when compared to “IPv6 with Mobility”, specifically regarding handoff run time. Considering real-time systems, although seeming irrelevant, such a small difference can be very important. In some applications, such as aviation, critical embedded systems must have low failure rates, such as a serious failure every 10^5 até 10^9 h of operation at most. Whereas delays in communication can lead to failures, this difference becomes significant.

Taking the example of scenarios where terrestrial or aquatic vehicles operate missions in isolated areas and UAVs are used to fly and collect/provide information for terrestrial and aquatic networks, the flyovers cannot be performed more than once depending on the local access conditions. In some cases, the UAV flight range limit is very short, requiring the flight to be performed with higher speed and only once. In this case, handoff process run time for connection among UAV and terrestrial/aquatic vehicles could significantly impact the overall system.

Another practical example is related to smart cars and roads. On highways with multiple APs, handoff operations happen frequently, which can lead to inherent delays that may affect both safety and entertainment operations. Thus, the mentioned small differences in run time can be important and should be taken into account on these systems design, especially for real-time applications.

9 Conclusions

This paper carried out a comparative evaluation between two IPv6 network based handoff processes, with and without mobility. Some parameters such as run time and decision making were taken into account for performance analysis. In fact, the simulated results presented a slight advantage of handoff process without mobility. As for real-time systems, such a small difference may be very representative.

Although tests were conducted in a completely deterministic simulation and the decision making algorithm was the same in both models, this paper presented results that help researchers and developers to have insights on embedded systems behaviour using IPv6 protocol facing different handoff situations. These experiments are part of the implementation of NIMBLE platform for mobility on unmanned vehicles.

As future work we aim to perform similar experiments in real embedded systems prototypes running real-time operating systems (RTOS). These operating systems meet the needs of such systems, and might also improve the communication performance. Moreover, new scenarios of UV applications should be taken as examples for new experiments.

References

1. Abbott-McCune, S., Kobezak, P., Tront, J., Marchany, R., Wicks, A.: UGV: security analysis of subsystem control network. In: Karlsten, R.E., Gage, D.W., Shoemaker, C.M., Gerhart, G.R. (eds.) *Proceedings of SPIE - The International Society for Optical Engineering*, Baltimore, MD, vol. 8741, p. 87410Z (2013). doi:[10.1117/12.2016314](https://doi.org/10.1117/12.2016314)
2. Bekmezci, I., Sahingoz, O.K., Temel, S.: Flying ad-hoc networks (FANETs): a survey. *Ad Hoc Netw.* **11**(3), 1254–1270 (2013). doi:[10.1016/j.adhoc.2012.12.004](https://doi.org/10.1016/j.adhoc.2012.12.004)
3. Bouachir, O., Abrassart, A., Garcia, F., Larrieu, N.: A mobility model for UAV ad hoc network. In: *2014 International Conference on Unmanned Aircraft Systems (ICUAS)*, pp. 383–388. IEEE (2014). doi:[10.1109/ICUAS.2014.6842277](https://doi.org/10.1109/ICUAS.2014.6842277)
4. Chuang, M.C., Lee, J.F.: FH-PMIPv6: a fast handoff scheme in proxy mobile IPv6 networks. In: *2011 International Conference on Consumer Electronics, Communications and Networks (CECNet)*, pp. 1297–1300. IEEE (2011). doi:[10.1109/CECNET.2011.5768193](https://doi.org/10.1109/CECNET.2011.5768193)
5. Chung, H., Oh, S., Shim, D.H., Sastry, S.S.: Toward robotic sensor webs: algorithms, systems, and experiments. *Proc. IEEE* **99**(9), 1562–1586 (2011). doi:[10.1109/JPROC.2011.2158598](https://doi.org/10.1109/JPROC.2011.2158598)
6. Das, K.: Mobile IPv6: What is Mobile IPv6? (2017). <http://ipv6.com/articles/mobile/Mobile-IPv6.htm>
7. Gupta, L., Jain, R., Vaszkun, G.: Survey of important issues in UAV communication networks. *IEEE Commun. Surv. Tutor.* **18**(2), 1123–1152 (2016). doi:[10.1109/COMST.2015.2495297](https://doi.org/10.1109/COMST.2015.2495297)
8. Hinden, R., Deering, S.: RFC 4291 - IP version 6 addressing architecture. Technical report, IETF Tools (2006). <https://tools.ietf.org/html/rfc4291.html>
9. INET: Network MIPv6 Documentation (2017). <https://github.com/inet-framework/inet/blob/master/examples/mobileipv6/MIPv6Network.ned>

10. Johnson, D., Perkins, C., Arkko, J.: RFC 3775 - mobility support in IPv6. Technical report, IETF Tools (2004). <https://www.ietf.org/rfc/rfc3775.txt>
11. Koodli, R.: RFC 4068 - fast handovers for mobile IPv6. Technical report, IETF Tools (2005). <https://tools.ietf.org/html/rfc4068>
12. Kurose, J.F., Ross, K.W.: Computer Networking: A Top-Down Approach, 5th edn. Addison Wesley, Boston (2009)
13. Luo, C., Ward, P., Cameron, S., Parr, G., McClean, S.: Communication provision for a team of remotely searching UAVs: a mobile relay approach. In: 2012 IEEE Globecom Workshops, pp. 1544–1549. IEEE (2012). doi:[10.1109/GLOCOMW.2012.6477815](https://doi.org/10.1109/GLOCOMW.2012.6477815)
14. Maza, I., Caballero, F., Capitán, J., Martínez-de Dios, J.R., Ollero, A.: Experimental results in multi-UAV coordination for disaster management and civil security applications. *J. Intell. Robot. Syst.* **61**(1–4), 563–585 (2011). doi:[10.1007/s10846-010-9497-5](https://doi.org/10.1007/s10846-010-9497-5)
15. Mishra, A., Shin, M., Arbaugh, W.: An empirical analysis of the IEEE 802.11 MAC layer handoff process. *ACM SIGCOMM Comput. Commun. Rev.* **33**(2), 93 (2003). doi:[10.1145/956981.956990](https://doi.org/10.1145/956981.956990)
16. Nankani, A.: Horizontal handoffs within WLANs. Ph.D. thesis, Master thesis, Department of Microelectronics and Information Technology, Royal Institute of Technology (KTH) Sweden (2005). <ftp://ftp.it.kth.se/Reports/DEGREE-PROJECT-REPORTS/28>
17. OMNeT++ Discrete Event Simulator: What is OMNeT++? (2017). <https://omnetpp.org/intro>
18. Perkins, C., Johnson, D., Arkko, J.: RFC 6275 - mobility support in IPv6. Technical report, IETF Tools (2011). <https://tools.ietf.org/html/rfc6275>
19. Pigatto, D.F., Gonçalves, L., Roberto, G.F., Rodrigues Filho, J.F., Floro da Silva, N.B., Pinto, A.R., Lucas Jaquie Castelo Branco, K.R.: The HAMSTER data communication architecture for unmanned aerial, ground and aquatic systems. *J. Intell. Robot. Syst.* 1–19 (2016). doi:[10.1007/s10846-016-0356-x](https://doi.org/10.1007/s10846-016-0356-x)
20. Quaritsch, M., Kruggl, K., Wischounig-Struel, D., Bhattacharya, S., Shah, M., Rinner, B.: Networked UAVs as aerial sensor network for disaster management applications. *Elektrotech. Informationstechnik* **127**(3), 56–63 (2010). doi:[10.1007/s00502-010-0717-2](https://doi.org/10.1007/s00502-010-0717-2)
21. Rieke, M., Foerster, T., Broering, A.: Unmanned aerial vehicles as mobile multi-sensor platforms. In: The 14th AGILE International Conference on Geographic Information Science, 18–21 April 2011, Utrecht, Netherlands (2011)
22. Sahingoz, O.K.: Networking models in flying ad-hoc networks (FANETs): concepts and challenges. *J. Intell. Robot. Syst.* **74**(1–2), 513–527 (2014). doi:[10.1007/s10846-013-9959-7](https://doi.org/10.1007/s10846-013-9959-7)
23. Sarkar, S.K., Basavaraju, T.G., Puttamadappa, C.: Ad Hoc Mobile Wireless Networks: Principles, Protocols and Applications, 1st edn. Auerbach Publications, Boca Raton (2008)
24. Kim, S.-W., Seo, S.-W.: Cooperative unmanned autonomous vehicle control for spatially secure group communications. *IEEE J. Sel. Areas Commun.* **30**(5), 870–882 (2012). doi:[10.1109/JSAC.2012.120604](https://doi.org/10.1109/JSAC.2012.120604)
25. Sun, Z., Wang, P., Vuran, M.C., Al-Rodhaan, M.A., Al-Dhelaan, A.M., Akyildiz, I.F.: BorderSense: border patrol through advanced wireless sensor networks. *Ad Hoc Netw.* **9**(3), 468–477 (2011). doi:[10.1016/j.adhoc.2010.09.008](https://doi.org/10.1016/j.adhoc.2010.09.008)
26. Tanenbaum, A.S., Wetherall, D.: Computer Networks. Pearson Prentice Hall, Upper Saddle River (2011)

27. Vatn, J.O.: An experimental study of IEEE 802.11 b handover performance and its effect on voice traffic. Technical report, Citeseer (2003)
28. Verma, A., Fernandes, R.: Persistent unmanned airborne network support for cooperative sensors. In: Proceedings of SPIE - The International Society for Optical Engineering, Baltimore, MD, vol. 8756, p. 87560J (2013). doi:[10.1117/12.2018293](https://doi.org/10.1117/12.2018293)
29. Wong, J.Y.: Theory of Ground Vehicles. Wiley, Hoboken (2008)
30. Xiang, X., Liu, C., Lapierre, L., Jouvencel, B.: Synchronized path following control of multiple homogenous underactuated AUVs. J. Syst. Sci. Complex. **25**(1), 71–89 (2012). doi:[10.1007/s11424-012-0109-2](https://doi.org/10.1007/s11424-012-0109-2)

Communication in Critical Embedded Systems

First Workshop, WoCCES 2013, Brasília, Brazil, May, 10, 2013, Second Workshop, WoCCES 2014, Florianópolis, Brazil, May 9, 2014, Third Workshop, WoCCES 2015, Vitória, Brazil, May 22, 2015, 4th Workshop, WoCCES 2016, Salvador, Brazil, June 3, 2016, Revised Selected Papers

Branco, K.; Pinto, A.; Pigatto, D. (Eds.)

2017, XV, 131 p. 68 illus., Softcover

ISBN: 978-3-319-61402-1