Model-driven approach to design a secure routing protocol for UAV Adhoc networks

Jean-Aimé Maxa

Congrès des Doctorants 2015
EDSYS (École Doctorale Systèmes)
19-20 mai, Toulouse

Model Driven Approach to design a Secure Routing Protocol for UAV Ad Hoc Networks

Jean Aimé MAXA (maxa@recherche.enac.fr)
TELECOM/Resco/ENAC, Toulouse
Ecole Nationale de l'Aviation Civile, Univ de Toulouse, France
Thesis supervised by: Mohamed Slim Ben Mahmoud et Nicolas Larrieu (ENAC/TELECOM/RESCO)

Abstract

UAANET (UAV Ad Hoc Network) is a wireless network that is able to organize itself without a pre-existing infrastructure. It consists of forming an ad hoc network with multiple UAVs and the Ground Control Station (GCS). In order to route data packets between nodes, a routing protocol is required. This routing protocol must not only satisfy the UAANET network requirements but also the validation requirements with a formal method to verify the routing protocol conformance. This will eventually contribute to the certification of the UAANET communication network (e.g to be compatible with the specification document DO 178C). In this paper, we propose a validation model of a secure routing ad hoc protocol for UAANET based on the well known AODV algorithm. The MDD (Model Driven Development) approach is used and provides a formal specification capabilities to ensure reusability, modularity, and conformance of the AODV implementation.

Introduction

An Unmanned Aerial Vehicle (UAV) is a pilotless aerial vehicle that can be either controlled by an on-board computer or remotely piloted by a distant operator. As almost all of them are equipped with a wireless communication system, UAVs can be used to create a self-organizing and multi-hop network called UAV Ad-hoc NETwork (UAANET) as shown in figure 1. Similar to MANETs (Mobile Ad hoc Networks), the UAANET communication architecture is infrastructure-less and self-configuring network of several nodes forwarding data packets. Nevertheless, it also has some specific features (cf. section A1) that brings challenges on network connectivity. Consequently, an adapted routing protocol is needed to exchange data packets within UAANETs.

With this in mind, in our previous work (detailed in section B), we evaluated different MANET routing protocols perfomances with hybrid experimental system (combining simulation and emulation paradigm) under realistic UAANET scenarios. We found that AODV [16] was the best starting point for our UAANET routing protocol design. AODV is a simple and efficient reactive routing protocol allowing the network to be completely self-organizing and self-configuring without the need of administration.

Following that, from a network security point of view, both data traffic and the routing protocol itself need to be strengthened. Indeed, in a wireless environment, attacks are likely to occur: control traffic needs to be authenticated in order to give the right orders to flying UAVs. Also, attacks such as wormhole [8] or packet forwarding attack [19] are easy to conduct if the routing protocol is not protected against them. It appears then that an efficient and secure routing protocol is the main objective to be achieved for UAANETs.

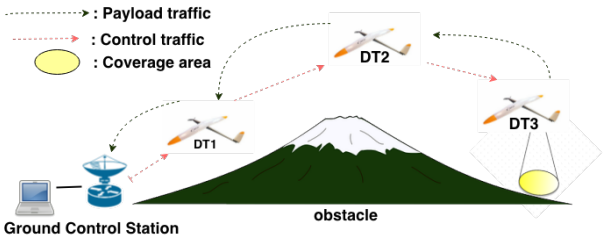

Figure 1: UAANET topology illustration

Furthermore, it is important to underline that, UAANETs have to be certified in the short future in order to enable integration of UAV into civil airspace. This certification includes different parts of the UAV System (UAS) such as the autopilot, the operating system and the communication system. In order to be compliant with the French Civil Aviation Administration [1] airspace certification process as suggested by the specification document DO-178C [15], we use a Model Driven Development ap-

[1] DGAC: Direction Générale de l'Aviation Civile which is equivalent to FAA (Federal Aviation Administration) but at a French scale.

proach.

MDD [10] is a design methodology that enables cost-effective and fast development of complex systems. The main idea is to begin the design with models rather than software algorithms. The model is an executable specification that can continuously be modified during the development step. Also, MDD allows to generate code for hardware implementation requirements by minimizing programmation errors. These steps help not only for modularity and reusability of the final code but also contribute for the certification of the UAS communication system. For the formal specification of the AODV protocol, we use the Matlab Simulink and Matlab stateflow. We would like to stress out that this work is included into the SUANET[2] (Secure Uav Ad hoc NETwork) project.

In this paper, we propose to use the MDD methodology to design a secure UAANET routing protocol. In the first section, we will present the UAANET state of the art. In the second section, we will detail the MDD methodology used for UAANET with its benefits. In section III, we will briefly present the security component of our routing protocol to secure the traffic exchanged between the UAV and the ground station. Finally, in section IV, we conclude by giving few directions for future work.

A. UAANET state of the art

A.1. UAANET Specifications

Similar to MANETs, the UAANETs architecture is an infrastructure-less network which uses several nodes to forward data packets. It shares some common features with standard mobile ad hoc networks such as self-organized pattern, self-managed information in a distributed fashion and communication between nodes without a centralized authority. However, UAANETs have also some specific features that can be listed as [3]:

- **Network connectivity**: depending on the mobility degree, the connectivity between two UAVs could be lost while they are transmitting critical information (control/command traffic). In addition, UAV platform failures can also affect the network topology. Indeed, when a UAV fails, the links with an UAV have been involved also fails, and it results in a topology update. Another factor that affects the connectivity are the connection outages. Because of the UAV movements and variations of distances, link quality fluctuates and may cause loss of connectivity and performance degenerations.
- **Sufficient energy and storage**: depending on their sizes. UAVs are usually assumed to have more energy and computing power than nodes in MANETs. This can be explained by the fact that the energy needed to move the UAV is much greater than the energy needed to compute data.
- **Mobility**: UAV mobility patterns are a lot different from any other vehicle. An UAV movement is above all 3D based. This brings a whole set of challenges on the physical layer, the antenna behaviors and the security aspect(e.g, misbehavior detection). Furthermore, UAVs are used for specific missions that can include several different mobility patterns like area scanning, reaching a way-point, staying at a position or even patrol around a circuit. The diversity of UAV moves leads to very varied connectivity patterns.
- **Propagation model**: in MANETs networks, nodes usually move close to the ground (like in VANETs or sensors networks). UAANETs are rather different as it is composed of flying node moving in large free space. Consequently, the free-space path loss model is often used to model the physical layer. Nevertheless it is advisable to take into account factors like large obstacles, ground reflections or weather conditions which can affect connectivity between UAVs.
- **Strict delay constraints**: generally, UAANETs are used for real-time applications, such as, aerial photography and video capture in case of remote monitoring and environmental measurements. In addition, the control/command traffic should also arrive on time and computed by the UAV in order to avoid loss of control.

A.2. Routing protocols for UAANETs

Several dynamic routing mechanisms are available for UAANETs. They can be classified into four main categories such as proactive, reactive, hybrid and geographical routing.

First, a proactive routing mechanism tries to establish a route from one node to another before it is needed. Each node in the ad-hoc network sends control messages at a fixed rate. They usually contain the node routing table and relayed information from other nodes. Step by step, routing information is relayed from the destination node to the source node, and a route can be established. As UAANET proactive routing protocols we can cite DOLSR [1] or Predictive-OLSR [17].

Secondly, reactive protocols establish a route when it is necessary. When a node wants to send a packet to a destination node, it first sends a route

[2]It is a specific French research project whose objectives are to secure the integration of a UAV swarm into the French civilian airspace

request packet which will be flooded through the whole network. When a node receives the route request packet, it adds its address to the list of the nodes that the packet went through. When one (or several) route request packets reach the destination node, a route response packet is sent back to the source using the shortest route discovered. The source uses this route to reach the destination. As UAANET reactive protocols we can cite Time-slotted on-demand routing [5] or UEDSR (UAV Energy Dynamic Source Routing Protocol) [11].

As far as geographical routing concern, it uses the nodes positions to find the best route from a source to a destination. Usually, it uses two distinct mechanisms: greedy forwarding and a backup mechanism in case where the former failed. The greedy forwarding consist of selecting as a next hop the closest node from the source node position. Alternatively, in case where no node within range closer to the destination is found, a backup mechanism is automatically launched. Several geographical routing protocols have been proposed, we can cite: GPMOR [13] or USMP [12].

Lastly, hybrid routing is a generic term referring to a combination of two routing mechanisms. As an example, we could cite RGR [18], which is a reactive protocol using greedy forwarding as a backup mechanism.

B. Emulation-based performance evaluation of routing protocols for UAANET

In order to design a UAANET routing protocol for the SUANET project, it is necessary to compare the existing MANETs routing protocol and select one that most fits UAANET requirements. Note that some studies have already approached the comparison of routing protocols for UAANETs, but they are based on simulation which does not consider the linux kernel networking stack and also a realistic mobility model. In other words, they hide several important parameters (e.g, protocol implementations, background traffic, real time execution) due to the lack of OS-based implementations. These limitations could induce significant differences between simulations and real test-bed results. Accordingly, we decided to create an innovative tool combining the low cost of a simulation with the accuracy of a real protocol stack [14]. This tools includes an hypervisor to run the virtual machines, a measurement tools and a framework to allow virtual machines to communicate through a virtual wireless medium. The traces used to generate UAVs mobility pattern were extracted from real traces so that physical related factors could be as realistic as possible. An illustration of this system is on Figure 2.

This tool available for community allowed us to compare AODV, DSR and OLSR while considering UAANET realistic scenarios. As can be seen in Table 1, we found AODV as our starting point since it outperformed DSR and OLSR in terms of overhead, end-to-end delay and finally the time for connectivity retrieval after route loss.

	AODV	**DSR**	**OLSR**
Delay	5.32 ms	10.15 ms	5.91 ms
Overhead	501 kB	759.99kB	438kB
Connectivity	90.65 %	58.2 %	24.1 %

Table 1: Evaluation performance results

It is important to underline that we decided not to evaluate any geographical protocols because their mechanisms are usually based on how to implement an additional mechanism to exchange the different node positions through the Ad-Hoc Network (otherwise we would have to use an other communication medium, which is in our case not available). As future works, it could be interesting to analyse the different position sharing mechanisms available in the literature and verify if one shows off as the most efficient to enhance the comparison study with an additional geographic routing protocol.

Now that AODV has been identified, it is necessary to re-design it with MDD approach in order to ensure modularity, reusability, conformance and also to contribute for certification. This is detailed in the next section along with the MDD methodology.

C. Model Driven Development for UAANET routing protocol

In this section, we will present the methodology used for UAANET routing protocol design to respond to a certification requirements. This methodology is mainly based on MDD approach and divided into 6 steps as it is shown in the figure 4.

C.1. Model Driven Development Approaches

As stated previously, MDD is a design methodology that claims the use of models as primary artifacts in the development process. The main idea is to focus on creating models rather than software codes. Indeed, a system model is the focus of the development process, from requirements specification, development through model design, simulation testing and integration. Addtionnaly, MDD allows to generate high level code which allows to verify to test the coverage of the model and therefore the conformance compared to the requirements documents. It also allows to create unitary test and execute simulation for system verification. MDD therefore allows to improve productivy, quality and efficiency of the software product by:

- linking designs directly to requirements;
- generating embedded software code;
- generating documentation;

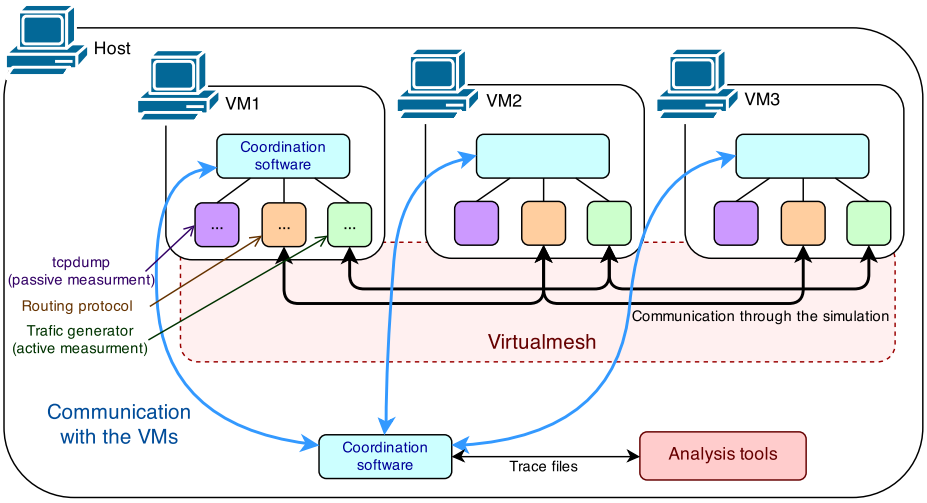

Figure 2: Testbed implementation

- reusing the model and add other functionnalities easily;
- performing code coverage;
- performing conformance verification based on requirements documents.

Note that, it is important to underline that it takes time and effort to develop models. However, reusing them by generating code for implementation and test is a good way to gain from MDD. More importantly, for our work, the underlying motivation for MDD is twofold [9]: to improve productivity (modularity and reusability) from software development effort and to contribute for future UAS Communication System certification (validation model, conformance formal verification).

C.2. Model-Based Design for DO-178C with Qualified Tools

As stated in [10], methodology design with automatic code generation is an important advantage for developing aeronautical software. It allows early verification, validation and test of models. DO-178 is a commercial aviation standard used to certify aeronautical systems. It define objectives for software lifecycle process activities combining software requirements, software design, coding, integration and configuration management. For each objective, outputs that need to be specified are created and verified. DO-178C is the latest version of DO-178 standard. Compared to its predecessor, it contains two additonnal important features: the Model-Based Development along with Verification supplement and the Formal Methods supplement. In other words, the focus is placed on MDD approaches that can automatically generate code by taking as inputs a high level models. The DO-178C allows to validate such a methodology by using formal verification.

C.3. AODV MDD validation model

In this part of the document, we will run through the methodology used to modelize AODV. Many steps are required to design our protocol and to execute it. Figure 4 shows these steps.

Step 1: Requirements specification: the AODV model requirements is mainly based on the AODV RFC Specifications. The main functionnalities have been modelized expect for few ones which were too complicated to modelize. We would rather prefer to directly add them during the glue code phase.

Step 2 : AODV modelization: after defining AODV model requirements based on RFC specifications (https://www.ietf.org/rfc/rfc3561.txt), Simulink and Stateflow allow to pass from high-level requirements to a design pattern. Moreover, in order to get full benefits of MDD through Matlab Simulink and Stateflow, we employ verification measures with *Simulink Design Verifier*. It allows us to check the compatibility of the model, traceability and certification with respect to the reference document DO-178C used in aeronautical design of embedded systems. So at the end of this step, we get a model verified and validated.

An illustration of AODV model is shown in figure 3. Our purpose is to verify the importance of the use of the MDD approach for the UAV application. The methodology consist then to verify that the automatic code generation provides an executable source code conforming to RFC recommendations and the performance of the standard code of AODV protocol. In the following, the basic concepts of route discovery process through MDD is explained.

Packet passes through the first block "Packet_Mngnt" in which, we get the package in the global variable "currentPacket" in order to ma-

nipulate its fields. The type field of the packet is tested to determine if it is an RREQ or RREP (= 1 = 2 respectively). Four outputs are possible in this block.

The first ouput is the "init" event which is mandatory and executed regardless of the packet entry type, in order to initialize the configuration of the routing table and network interfaces. Note that this configuration is an external file that is passed to the program as input. Similarly, whatever the type of package, it will go through the last exit "input_packet" which is used to count the number of incoming packet. This is for sake of simulation (unitary test) within matlab environment to verify that each entry is pointing at a given output.

Afterwards, the program chooses between the two outputs "rreq" and "rrep" depending on the result type field that we have already identified. If the packet is a RREQ packet, the program process proceeds to the block "RREQ Mngnt" through the event "goRouting". Three outcomes are possible after completion of the block:

- Whether this node is the receiver and then it goes to "ProcessLocal_RREP";
- Or it is an intermediate node and must broadcasts the message. In that case, it follows the process "forwardPacket";
- Or it does not find a path represented by "noRouteFound."

Figure 5 shows the transition states of the bloc "RREQ_Mngnt." In this block, we get the IP source (IPsrc) and destination (IPdest) of the incoming packet and we test if the number of requests has not exceeded the maximum number "MAX_ROUTE_ENTRY" which is 16 in our case. If this value exceeds, the program sends out a noRouteFound variable output.

Then, we get the IP address of the local node in the routing table of AODV in the variable "myLLAddr". Following that, we verify whether or not it is equivalent with IPdest. In the event of a tie, this node is the destination and he goes to the process "processLocalRREP" to send a RREP message to the source. Otherwise the node broadcasts the message to its neighbors by changing the destination address to broadcast address and executes a number of update (Increment_OrigSN, etc) in accordance with the AODV RFC recommendations.

Then, the block "SendThePacket" is used to retrieve the package "currentPacket" after updates and send it in the network. The block "RREP_Mngnt" allows the destination node to create a RREP packet and send it to the source while the "fwrd_rrep_packet" block take care to update the fields (TTL, hopCount, destination IP) of RREP received packet and send it to the next node until it achieves the source.

Finally, The block "Statistics_Mgnt" receives the variables of the different blocks that make statistics on whether the packet is received, sent or there is errors.

As a result, a simulation of the model is executed in order to get the generated code. Note that, the code generation leads to the creation of several C language files. The code of the individual blocks of the model is generated in the "AODV.c" file and fichier "AODV.h". The definition of types is created in the "AODV_types.h" file. On the other hand, the "ert_main" generated file used to add C code to the initial program and integrate it into the model.

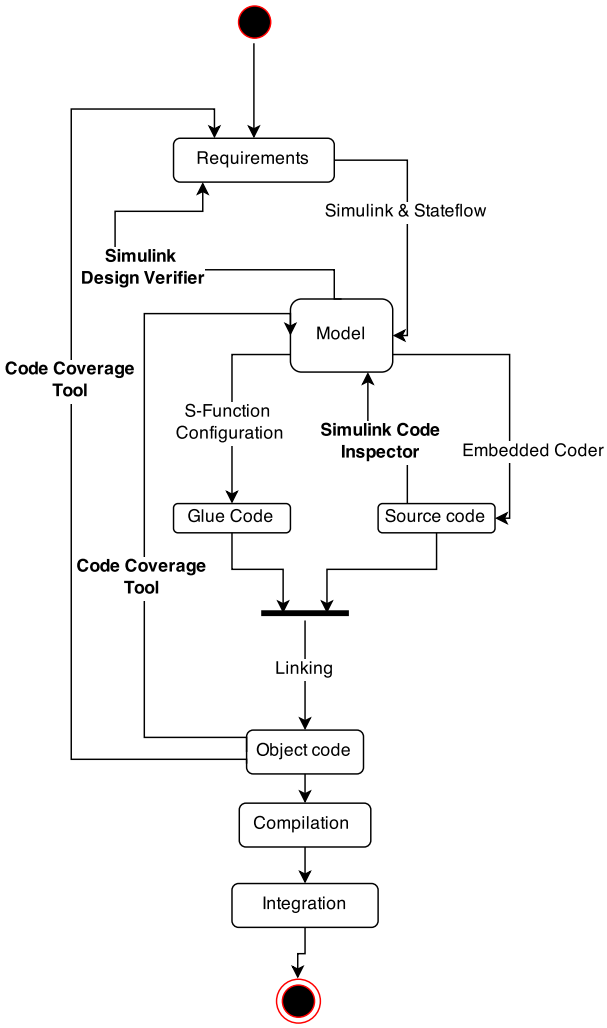

Figure 4: MDD Methodology

Step 3: Coding process: the next step corresponds to the automatic generation of code. This step consists on installing the "Embedded Coder" tool offered by Matlab. The verification of the source code, whether automatically or manually, requires a code review as part of the DO-178C process. With Mathworks, this operation is run automatically with *Simulink Code Inspector*. It compares generated code with its source model to test specification conformance. The code inspector systematically examines blocks, state diagrams, parameters,

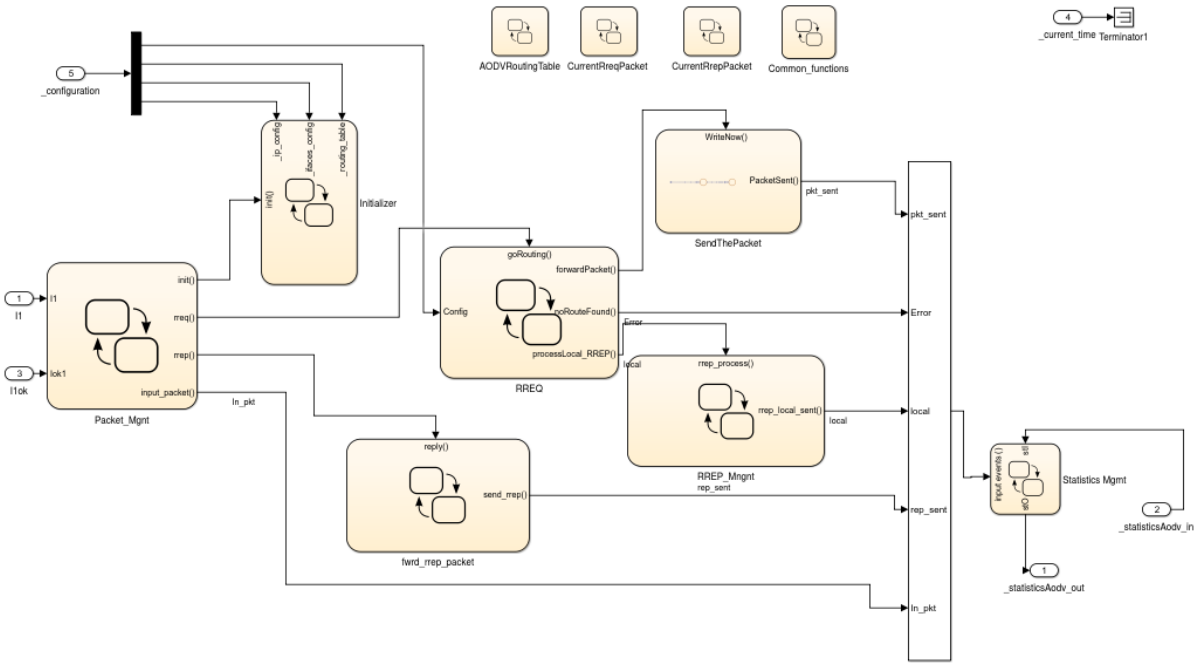

Figure 3: AODV Design

and settings in the model to determine whether they are structurally equivalent to operations, operators, and data in the generated code. It generates traceability documentation that can be used for certification application to certification authorities.

Step 4: Glue code process: the previous generated code is a high abstract and level code and as such, there is a need to create a link to the operating system. The glue code consist of specifying how the generated code will exchange inputs and outputs with the linux kernel by calling the requested library functions. It also specify how the source code exchanges with the different hooks of the netfilter. This kernel communication interface can be manually written or automatically generated with a predefined tool. We decided to use *S-Function* to complete this task. The S-function option is a computer language description of a Simulink block and can encompass C code. Afterwards, the S-function blocks are compiled as MEX files using the mex compiler. Finally, the S-functions are dynamically linked to the model which allow to run unitary test along with the multiple model blocks. It also possible to directly generate code embedded in S-Function along with code from Real-Time Workshop Embedded Coder.

Step 5: Object code process: the third step is the creation of the object code. This step is automated with Matlab code generation with an appropriate compiler in the program preferences. In our work, we decided to use the well-known gcc compiler. However, when integrating the object into a specific embedded systems (For instance, a phytec ARM BOARD [2]), it is necessary to cross-compile with a cross-compiler and linker tools. With Matlab, it is also possible to test the object code and achieve structural coverage compared to the model and to the requirements. Such task can be executed automatically by the *Matlab model coverage tool*.

Step 6: Compilation process: the compilation phase consist of merging the generated code and the glue code. In SUANET project, the compilation is two-fold, first, we have to test it into the emulation tools, and verify if it works properly. The second step is the cross-compilation with an ARM-BOARD which is the final end system of the SUANET project.

Step 7: Integration: the last step consist of executing into the embedded target the binary image of the routing protocol in order to ascertain the conformance of the routing protocol to the RFC documentation. This integration consist of testing the protocol into the ARM-BOARD and perform a communication relay as illustrated in figure 1 .

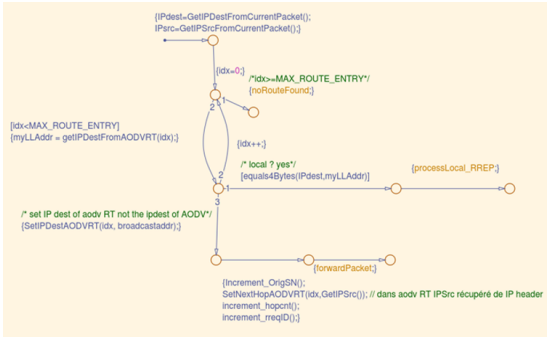

Figure 5: Transition states of rreq mngnt

D. Secure Routing Protocol for UAVs (SRPU)

In this section, we will briefly detail the security extension of our prototocl.

D.1. Security for UAANET

UAANET security is necessary to verify that nodes are trustworthy before exchanging data packets. It is also needed to ensure that any malicious entity cannot disturb the transmission of data messages. The specific features of UAANETs mentioned above represent a challenge from a security point of view. In addition, many existing security solutions for MANETs are inefficient and not suitable to UAANETs. Consequently, security should be taken into account at the early stage of UAV network mechanism design.

Furthermore, attacks against the routing process can be divided into two categories: route discovery attacks and data forwarding attacks. The first category refers to the attacks which could harm the control and command traffic, while the second category is relevant to the payload traffic (image and video traffic). We consider that a corrupted configuration settings packets could have a significant impact on the entire UAS since it corrupts directly the route finding process. Therefore, the first class is considered to be more critical. Likewise, the second category of attacks has essentially an impact on network performances. These potential attacks happen only after the route finding process, which can therefore be detected by a security mechanism. For instance, with hash chain mechanism which allow each node to verify the integrity of a message hop count, each node is able to detect forged message and recognize if the message is originated or forwarded from an untrusted node.

As shown in the figure 1, our application scenarios involves three UAVs. We assume that the Ground Control Station (GCS) has to send a traffic control to DT3. This message is then transferred through DT1 and DT2. In this specific case, if any attackers between DT1 and DT3 happens to capture and modify all of the control packet, it will lead to a misconfiguration of DT3 and thus, induce complete failures of the UAS. Consequently, we need to prioritize the route discovery atttacks to ensure data transmissions by an efficient and secured route finding mechanism.

D.2. Secure routing protocol features

In this part, we will summarize the features of the chosen routing protocol for the SUANET research collaboration. As stated previously, this protocol is essentially based on AODV, and will add some AODV-SEC [4] properties. It is therefore a reactive based protocol which only searches a route when there is a data traffic to send within the UAANET. By choosing AODV protocol, the end to end delay value and the retrieval time value in case of route loss will be optimized. As regards the security criteria, the AODV-SEC algorithm will be implemented to ensure the authentification and the integrity security services. The confidentiality of both the data and the control traffic will be added through the addition of MSAODV [6] protocol. This protocol is an extension of AODV-SEC and bring confidentiality feature by using hybrid cryptographic mechanisms. Furthermore, note that, AODV-SEC does not have a countermeasure against wormhole attacks. This attack is especially sophisticated since it involves two attackers who perform a colluding attack. They records packets at a particular location and replay them at another node by using a high-speed private network. To tackle this specific attack, the Packet Leashes [7] mechanism will be added, which offers the possibility to calculate the expiration time and the distance travelled of the packets, and includes this information into the packets, so that the other UAVs can infer whether or not the packet has been altered.

E. Conclusion and future work

In this paper, we present a validation model for a secure routing protocol for UAANETs. It includes a formal specification of AODV routing protocol, a methodology based on model driven development and a set of tools for unitary test and simulation model. The methodology presented in this paper presents several advantages. First and probably the most important reason is the contribution for a certification of the UAS communication system based on the DO-178C specification document. Secondly, one important benefit is also the design of a formal specification from which functionnality code and code verification are automatically generated. It can indeed ensure conformance of the routing protocol based on the RFC requirement while eliminating design errors and inconstancy. Another advantage is also the productivity. Indeed, MDD allows modularity of the design techniques and the reusability of the software algorithms.

Moreover, it is important to stress out that MDD does not solve all the design issues. Programma-

tion skills is still necessary to complete the final implementation. In our future work, we are currently working on the last point of the methodology (step 7: Integration). We are looking forward to test the software solution in real environment with three UAVs form DELAIR-TECH company[3]. Also, we are currently working on how to ameliorate the routing performance by considering the link quality as a routing metric. Lastly, our perspective would be to strengthen the security part of our routing protocol.

References

[1] Abdel Ilah Alshabtat, Liang Dong, J Li, and F Yang. Low latency routing algorithm for unmanned aerial vehicles ad-hoc networks. *International Journal of Electrical and Computer Engineering*, 6(1):48–54, 2010.

[2] Alexander Bauer. Realtime capabilities of low-end powerpc and arm boards for embedded systems. In *Real-Time Linux Workshop*, 2007.

[3] Ilker Bekmezci, Ozgur Koray Sahingoz, and Şamil Temel. Flying ad-hoc networks (fanets): a survey. *Ad Hoc Networks*, 11(3):1254–1270, 2013.

[4] Stephan Eichler and Christian Roman. Challenges of secure routing in manets: A simulative approach using aodv-sec. In *Mobile Adhoc and Sensor Systems (MASS), 2006 IEEE International Conference on*, pages 481–484. IEEE, 2006.

[5] J Hope Forsmann, Robert E Hiromoto, and John Svoboda. A time-slotted on-demand routing protocol for mobile ad hoc unmanned vehicle systems. In *Defense and Security Symposium*, pages 65611P–65611P. International Society for Optics and Photonics, 2007.

[6] Ayman A Hanafy, Sherif Hazem Noureldin, and Marianne A Azer. Immunizing the saodv protocol against routing information disclosure. In *Internet Technology and Secured Transactions (ICITST), 2011 International Conference for*, pages 330–334. IEEE, 2011.

[7] Yih-Chun Hu, Adrian Perrig, and David B Johnson. Packet leashes: a defense against wormhole attacks in wireless networks. In *INFOCOM 2003. Twenty-Second Annual Joint Conference of the IEEE Computer and Communications. IEEE Societies*, volume 3, pages 1976–1986. IEEE, 2003.

[8] Yih-Chun Hu, Adrian Perrig, and David B Johnson. Wormhole attacks in wireless networks. *Selected Areas in Communications, IEEE Journal on*, 24(2):370–380, 2006.

[9] Nicolas Larrieu. How can model driven development approaches improve the certification process for uas? In *Unmanned Aircraft Systems (ICUAS), 2014 International Conference on*, pages 253–260. IEEE, 2014.

[10] Nicolas Larrieu and Antoine Varet. Methodology for rapid prototyping avionic software. *Rapid Prototyping of Software for Avionics Systems*, pages 23–60.

[11] Juan Li, Xiao Cheng Liu, Yue Feng Pang, and Wei Wei Zhu. A novel dsr-based protocol for small reconnaissance uav ad hoc network. In *Applied Mechanics and Materials*, volume 568, pages 1272–1277. Trans Tech Publ, 2014.

[12] Robert L Lidowski, Barry E Mullins, and Rusty O Baldwin. A novel communications protocol using geographic routing for swarming uavs performing a search mission. In *Pervasive Computing and Communications, 2009. PerCom 2009. IEEE International Conference on*, pages 1–7. IEEE, 2009.

[13] Lin Lin, Qibo Sun, Jinglin Li, and Fangchun Yang. A novel geographic position mobility oriented routing strategy for uavs. *Journal of Computational Information Systems*, 8(2):709–716, 2012.

[14] Jean-Aimé Maxa, Roudierea Gilles, and Larrieu Nicolas. Emulation-based performance evaluation of routing protocols for uaanets. In *Communication Technologies for Vehicles*, pages 227–240. Springer, 2015.

[15] Yannick Moy, Emmanuel Ledinot, Hervé Delseny, Vi Wiels, and Benjamin Monate. Testing or formal verification: Do-178c alternatives and industrial experience. *Software, IEEE*, 30(3):50–57, 2013.

[16] Charles E Perkins and Elizabeth M Royer. Ad-hoc on-demand distance vector routing. In *Mobile Computing Systems and Applications, 1999. Proceedings. WMCSA'99. Second IEEE Workshop on*, pages 90–100. IEEE, 1999.

[17] Stefano Rosati, Karol Kruzelecki, and Louis Traynard. Speed-aware routing for uav ad-hoc networks. In *Globecom Workshops (GC Wkshps), 2013 IEEE*, pages 1367–1373. IEEE, 2013.

[18] Rostam Shirani, Marc St-Hilaire, Thomas Kunz, Yifeng Zhou, Jun Li, and Louise Lamont. Combined reactive-geographic routing for unmanned aeronautical ad-hoc networks. In *Wireless Communications and Mobile Computing Conference (IWCMC), 2012 8th International*, pages 820–826. IEEE, 2012.

[19] Hao Yang, Haiyun Luo, Fan Ye, Songwu Lu, and Lixia Zhang. Security in mobile ad hoc networks: challenges and solutions. *Wireless Communications, IEEE*, 11(1):38–47, 2004.

[3] For more information: http://www.delair-tech.com/