Development of an IEEE 1588 Simulator and Analysis of UAVs' Synchronization in a FANET environment

Eiras F. C. S., Zucchi W. L

***Abstract*—**The application of IEEE 1588 standard is becoming quite common in distribution of synchronization information in packet networks. Studies of synchronization accuracy that can be reached with PTP protocol, defined in IEEE 1588 are essential for application design, both in telecommunications and in sensors industry. This article presents a new strategy for simulation of distribution synchronization which is more scalable and where simulation time is better controlled. We propose a strategy to use this simulation technique in a FANET environment where the synchronization information exchange is impaired by nodes movement, besides traffic bottlenecks in the network.

***Keywords*—** FANET; Time Synchronization; IEEE 1588 PTP Protocol; Computer Communication Network; Network Protocols; Mobile AdHoc Networks; Medium Access Control.

I. Introduction

Time synchronization for distributed elements is a common requirement for many types of applications [1], among which we can cite applications in sensor and telecommunication networks. The synchronization in wireless sensor networks is essential to facilitate the development of collaborative applications that perform various operations such as location, data aggregation, distributed sampling, etc. [2] To this end, several synchronization protocols for wireless sensor network applications have been studied, such as Reference-Broadcast Synchronization (RBS) [3], Timing-sync Protocol for Sensor Networks (TPSN) [4], Flooding Time Synchronization Protocol (FTSP) [5].

With the increasing use of asynchronous packet transmission networks in telecommunication and sensor applications, it emerged the interest about the possibility of information synchronization transmission through this type of network.

Ethernet networks have been increasingly used in backhaul of telecommunications networks replacing TDM/SDH links. The rationale for this trend is the lower cost associated with the Ethernet standard that, in some instances, represents 1/6 of the amount for SDH links. However, SDH links can provide a clock signal of high precision adequate for base stations operation in cellular telephony and other applications [6].

Additionally, TDM networks provide a hierarchical system for synchronism signals generation where a more accurate clock that provides an adjustment of the signal for stations with less accurate clocks.

On the other hand, the technology evolution in the last years in fields like embedded technology, size and capacity of processors and memories and in telecommunications among others, have allowed the quick Unmanned Aerial Vehicle (UAV) growth [7].

The UAVs normally are used in military missions, however civil applications have grown quickly in recent years [8]. In civil missions one can mention logistic applications, geographical information survey, hazardous land mapping, capture of meteorological data, agricultural mapping and disaster monitoring. In military applications one can mention its use for surveillance mission battlefield recognition, sensing troop positioning, critical infrastructure mapping such as nuclear power plant mapping.

There has been an increase in the use of a composition of multiple UAVs to accomplish both civil and military missions because this type of composition allows the capture of data in parallel mode, reducing the mission time, increasing system availability - that is, the task can be completed even if there is a hardware failure - and increasing sensing capability because the UAVs may be fitted with different types of sensors. Figure 1 show a possible composition of UAVs.

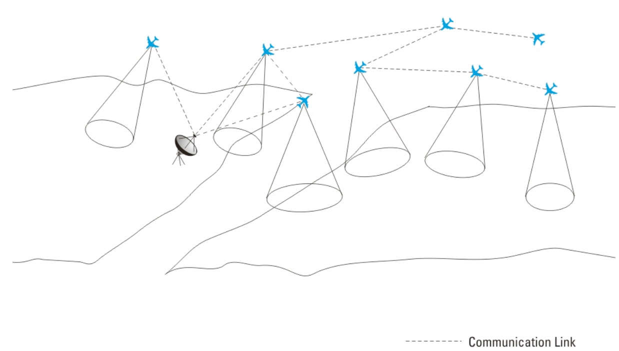

Fig. 1 – UAVs Scalability Scenario [7]

This type of UAVs' composition requires the establishment of a data communication network capable of sending information to one or more UAVs, and a voice communication network capable of establishing communication between the

UAV pilot located in a Ground Control Station (GCS) and the Air Traffic Controller when the UAV is inserted within a controlled air space context.

The communication network used for the command and control of UAVs is essentially a Mobile Adhoc Network (MANET). However, due to the specific characteristics of mobile nodes, in this case of UAVs, such as speed, mobility, quality of service requirements, discovering of other nodes and the delivery capacity of the data captured, this network became known as Flying AdHoc Network (FANET) [9].

There may be unexpected situations during the execution of a mission that can change the characteristics of UAVs' composition: weather conditions, electromagnetic interferences, a UAV shot down, hardware failure and geographical relief can generate a loss of communication between GCS and UAVs. Therefore, the FANET must be able to allow a new composition to maintain the capacity of communication.

To provide this capability, communication, routing and synchronism protocols used in other types of mobile applications need to be adapted and improved.

For the interoperation between all nodes, in order to maintain the integrity of captured and sent information, the integrity of voice communication (where applicable), the correct aggregation of sensed information in cluster (many UAVs capture parts of information and then sent to the GCS which makes aggregation and consolidation of information) and the establishment of precise geographical location of the UAV a highly accurate synchronization mechanism is required. This mechanism must be hierarchical and have the ability to adapt throughout the mission if necessary.

The precision of a GPS receiver is not sufficient. The estimate of position error defined as the distance between real position and estimated one, vary from zero up to a few hundred meters depending on the GPS receiver manufacturer [10].

One possibility to achieve this synchronization is to use the PTP protocol- defined into IEEE 1588 standard - since the communication network has similar characteristics of a packet network. It is of interest that this mechanism will be simulated on a FANET environment in order to verify and validate its operation and precision. Figure 2 shows an UAVs communication.

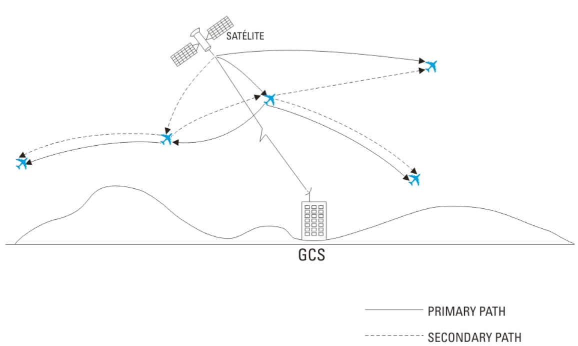

Fig. 2 – UAV's communication – FANET

A Primary Reference Clock (PRC) generates a clock signal with a precision level of 10^{-11} which amounts to an error of a second over 3172 years. Other levels have less intrinsic precision clocks, but this error is limited by a periodic comparison with PRC. The first two levels of synchronization hierarchy are generally implemented with a specific clock generation equipment containing oscillators of Cesium or Rubidium. Network equipment, like switches and routers, implement the third level.

The exchange of clock signals between different nodes is implemented by coding the time information over the bit stream sent by the reference node. Ethernet synchronous [11] [12] uses the same method between neighbor nodes.

In packet networks, reference clocks are still used, but the asynchronous characteristic of packet transmission requires a new distribution method because time information must be included in the packet fields. The standard known as IEEE 1588 [13] addresses this need.

IEEE 1588 standard uses discrete timestamps included in special time packets. The accuracy of this method depends on time packets frequency and delay uniformity. In packet networks queueing mechanisms introduces random delays and returning packets can travel by different routes. Owing to this synchronization accuracy is limited.

Since application performance depends on time precision, it follows that to be able to foresee limits on synchronization accuracy is of tantamount importance. Nevertheless, few methods to do that were seen in the literature until now.

The aim of this article is to propose to use a new simulation technique which is described in this paper and discuss how to adapt this simulation technique to study a FANET environment. Sections 2 and 3 present the concepts of synchronization and PTP protocol defined in Standard IEEE 1588 and its application in a wireless mesh network, of which FANETs are a special case. Section 4 presents the model developed and the results obtained and also the FANET model to be used and lastly section 5 shows the conclusions of the work.

II. Precise Time Synchronization

In general, two phenomena are said to be synchronized when they occur simultaneously. When one compare two clocks there can be a difference between them. This time difference can be of two kinds:

a) the difference is constant between the two clocks. In that case, there is only a time or phase difference. For example, the clock A displays 15:00 while clock B displays 15:05 and, after a time interval t, the time on clocks A and B will be displayed as 15:00+t and 15:05+t, respectively.

b) the difference varies along the time itself and there is a frequency difference. Thus, the time difference between clocks A and B changes along the time t. For example, at 15:00 the difference between clocks is 2 time units, at 15:00+t the difference will be 3 time units, at 15:00+3t the difference will be 4 time units, and so on. This means that the rate of measuring time between the two clocks is different.

When there is a rate difference between the two clocks, it is implicit that one of them is more precise than the other, and therefore, the less precise clock must be synchronized considering the more precise as reference. Time differences above a certain level induce errors in sensors applications and increase the bit error rate in data transmission.

Analytical methods to analyze delay have already been proposed in SDH networks. However such methods can not be applied in packet networks due to the difficulties in characterizing the traffic delay in this type of network.

A. Simulink as a Time Measurement Tool

This section presents some considerations on the use of Simulink for delay analysis and measuring time differences.

The model represented in figure 3 shows two clocks implemented by pulse generators and counters. The number of pulses counted is a measure of time in each clock and the model calculates and shows the difference between the two clocks.

If there is no frequency difference, but only time (or phase) difference between pulse generators (generator 1 generates a pulse in the instant 0.1 with period 1 and generator 2 generates a pulse in instant 0.3 with the same period), the value of the difference is always zero or one.

If there is a frequency difference between the two clocks, the time difference increases with time. This difference can jeopardize application performance and, therefore, must be controlled.

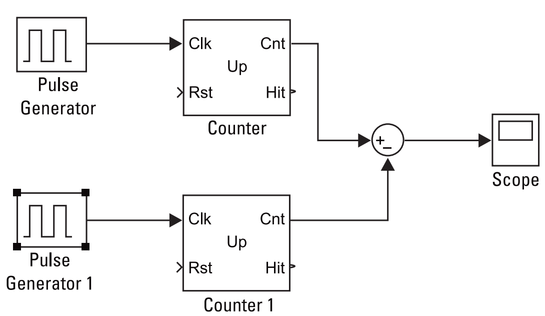

Fig. 3 - Model Comparing Pulses

A clock composed of a variable frequency pulse generator is used to adjust frequency. Figure 4 shows a Simulink model with automatic frequency adjustment feature.

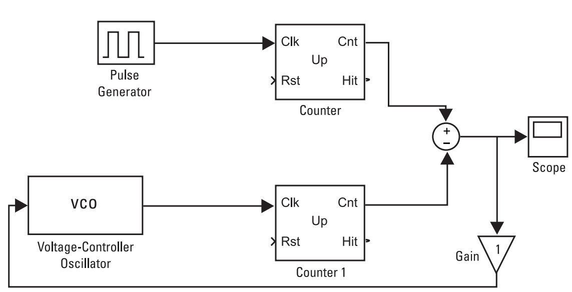

Fig. 4 - Model with Automatic Frequency Adjustment

Although there is a frequency difference between the two clocks, the time difference is limited. In order to make this useful in a network environment, it is required that the model represents the communication system between the stations.

These ideas are represented in figure 5 which is the basis for our simulator design.

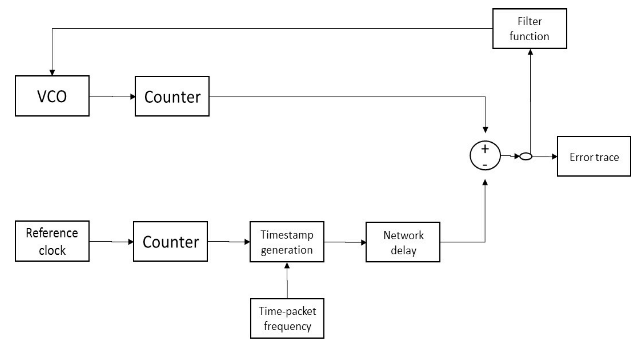

Fig. 5 - High Level Simulator

A precise network delay model depends on several factors and it is not easy to build. However, a worst-case analysis is enough for application validation.

Network Time Protocol (NTP) [14] and IEEE 1588 standards use messages and can transmit them between remote points of the network; however the transmission time between these points varies according to the traffic in the network at the time when this transmission occurs.

The receiver can calculate message delay and clock offset using information exchanged through protocol messages. This process is explained on next section.

B. Synchronization in a wireless mesh network

A Wireless Mesh Network (WMN) involves direct communication between end stations, in many cases without involving a fixed station.

In a WMN, after nodes are deployed they need to discover their neighbors. Knowledge of its neighbors is essential for almost all routing protocols, medium-access control protocols and several other topology-control algorithms [15].

IEEE 802.11 defines a good model for such kind of environment. Nearby stations can set relationships and exchange data between them, but the relationships are dynamics. They can be set or release according with the local movement of the stations.

Every station in a wireless mesh network can be the source, the destination or a forwarding element in the communication path. The interconnection between wireless mesh stations and the outside world is done by a “mesh AP”.

In a typical scenario, the mesh AP receives precise clock information from an external source and distributes it inside the wireless domain. Stations inside the wireless domain can be UAVs that demand synchronization information.

The delay for information transmission is highly variable, because routes inside the wireless area are changing quickly. Besides that, the residence times are different between stations because one can be busier than the other can.

IEEE 1588v1 admits a symmetrical delay in both communications directions. This hypothesis can be reasonable in many kinds of wireline networks, but it is hardly acceptable for a wireless mesh network. In that case, the reverse communication direction is often different from opposite direction and delay differences can influence clock synchronization.

Fortunately, IEEE 1588 v2 has adopt some mechanisms to cope with asymmetrical delays. They are called transparent clock. These mechanism was developed to deal with different queueing sizes in network switches, but the idea can be adapted for a mesh wireless network.

In the next section we develop the simulation ideas for a delay symmetrical network and discuss an model evolution for an asymmetrical case.

III. Packet Networks Synchronization

The standard proposed by IEEE for the time synchronization in packet networks defines a two-way synchronism protocol. This denomination occurs because not only the master (equipment with the more precise clock) sends event packets to a slave (equipment with less precise clock), but also the slave sends event packets to the master. However, the synchronism flow is always established from the master to the slave.

In 2008, IEEE 1588 v2 was defined to standardize the mechanism for the synchronization of clocks through a packet network. This synchronism is obtained through Precision Time Protocol (PTP) that is more precise than other protocols, as for instance NTP in addition to implementing tolerance to failures, message losses and receipt of out of order messages.

The operation of this protocol is based on the knowledge of transmission and receipt times of each event packet generated by nodes. The master node observes the time t_1, in which each event packet is transmitted, in its local clock, conventionally more precise.

This value must be sent to the slave node in its own packet, or, if not at all possible, it must be sent in a subsequent packet. When the packet is received by the slave node, the instance of time t_2 in which the packet arrived is observed.

The difference between t_2 and t_1 can be calculated, originating a series with the differences in values. These differences are filtered and can be used to adjust the slave node clock.

The process is complete with the slave node sending a message back to the master node. In this case the slave node measures and marks in the packet the instant of transmission of the message measured in its own clock and the master records the instant in which the packet is received. These instants in time are called t_3 and t_4, respectively.

Figure 6 shows the exchange of synchronization messages between the master node and the slave node, established by PTP in the standard IEEE 1588.

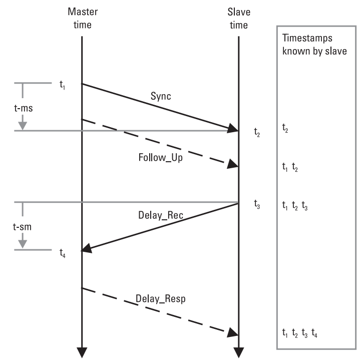

Fig. 6 - IEEE 1588 Synchronism Mechanism

Once t_1, t_2, t_3 and t_4 times are known, one can establish the value of Offset and Delay which are obtained from the equations (1) and (2) as follows:

$$t_2 - t_1 = Offset + Delay \qquad (1)$$

$$t_4 - t_3 = Delay - Offset \qquad (2)$$

Considering $(t_2 - t_1) - (t_4 - t_3)$ the Offset value is obtained by equation (3):

$$Offset = [(t_2 - t_1) - (t_4 - t_3)] / 2 \qquad (3)$$

Considering $(t_2 - t_1) + (t_4 - t_3)$ one obtains the Delay value, given by equation (4):

$$Delay = [(t_2 - t_1) + (t_4 - t_3)] / 2 \qquad (4)$$

The main simplification of this model is to consider symmetrical delay is in both ways. However, in a real network this characteristic is improbable and the delay can vary from one moment to another due to the traffic in the network, topology alterations and other factors. This can generate imprecision in the times used for correction. This imprecision can be corrected if, in addition to the average delay, the delay value in one of the directions is also known.

The main source of variable delay occurs inside the network equipment, such as switches and routers, as the residence delay of the messages depends on the queues in each transmission direction. The compensation of this effect can be obtained with equipment that measure this time in residence. IEEE 1588 standard specifies this equipment as transparent clock.

Some simulations of IEEE 1588 try to represent the behavior of stations involved. Some results were reported using OMNET++ [16] simulator, but simulation time increases exponentially with the rates of network transmission.

In this work we propose a new approach based in the representation of significant events of synchronization network. In fact slave clock synchronization accuracy depends on time message frequency and network delay value, but does not depends on the specific way this knowledge is obtained.

This way our approach does not represent network data traffic and can be useful even in high speed networks. Next section describes the simulator in bigger detail.

A. Model extension for asymmetrical delay

Figure 7 shows an example of transparent clock utilization to compensate for asymmetrical delays. We suppose there is an intermediate node between source and destination and the delays in each direction are different as shown.

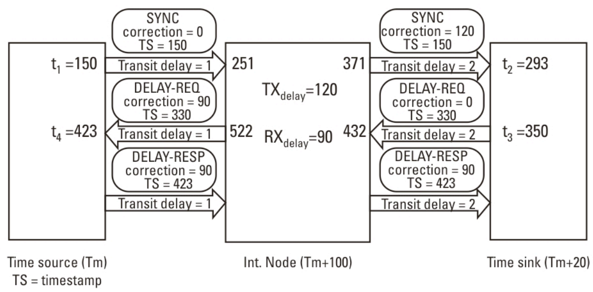

Fig 7 – Example of transparent clock with asymmetrical delays

In figure 7, t_1, t_2, t_3 and t_4 are the same time events described before. We suppose the time source is the clock reference, and there is a offset difference both in the intermediate node and in the destination node (time sink).

Message names are those defined by IEEE 1588 standard.

According with the standard a correction information is included in some messages to account for the asymmetrical delay in the intermediate node. We call these corrections TX_{delay} in the forwarding direction and RX_{delay} in the reverse directions.

The time sink can calculate the delay and the offset using the expressions:

$$Delay = ((t_2 - t_1)+(t_4 - t_3)-TX_{delay} - RX_{delay})/2 \quad (5)$$

$$Offset = t_2 - t_1 - TX_{delay} - Delay \quad (6)$$

Using those expressions in the figure 5 example one can easily show that:

$$Delay = 3;$$
$$Offset = 20;$$

As predicted. Note that with this technique Delay only include transit delay, which depends on physical propagation in the wireless medium, but does not include stations internal delays.

The transparent clock behavior will be included in the Simulink model to represent a wireless mesh network. We discuss this inclusion later in section 4.3 of this paper.

IV. SIMULATION

The simulation model built depicts the relevant features of the IEEE 1588 standard and combines a realistic representation of clock adjustment mechanism plus a functional simulation of network behavior, in order to obtain acceptable duration simulations. Simulink [6] tool was used for the creation and development of above model.

A. Model Description

The simulation model built, as shown in Figure 8, is composed of 3 main components: *Master, Network Delay* and *Slave.* Each component has a specific function into the model operation and may contain one or more blocks.

The *Master* component is composed of *Master_CLK* and *Master_Message* blocks. This component's main function is the generation of the system reference clock from a pulse generator and to simulate the reference clock (PRC) from which the entire system should be synchronized.

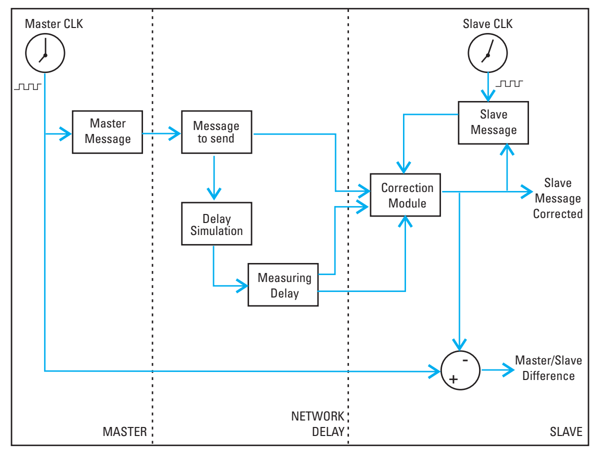

Fig. 8 – Model Block Diagram

The *Network Delay* is composed of *Message_to_Send, Delay_Simulation* and *Measuring_Delay* blocks. It's not made any attempt to emulate the behavior of switches or communication channels. This component main function is to calculate the delay that the packet experiences during transmission and update this information to the Slave. This is done through *Delay_Simulation* and *Measuring_Delay* blocks that insert random delays simulating a packet network behavior. *Message_to_Send, Delay_Simulation* and *Measuring_Delay* blocks are shown in more details in Figure 9.

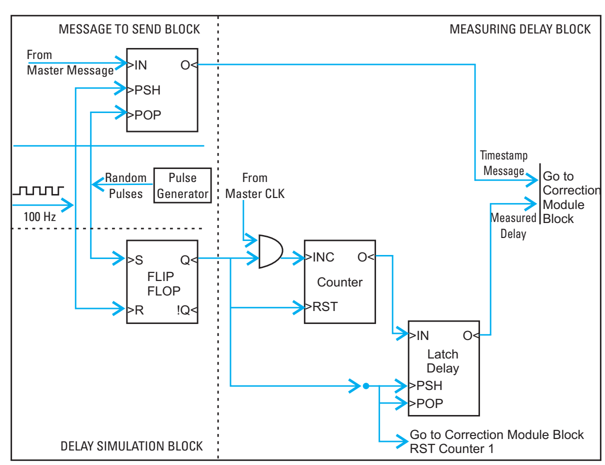

Fig. 9 - Network Delay

Message_to_Send and *Delay_Simulation* blocks simulate the delay in a packet network. This is done via a pulse generator component with random intervals, using different probability distributions.

The issue in this approach is that model's time scale is given by a reference pulse generator and not by the simulation step. Consequently, the generated random value must be converted into a time interval in the range of the reference clock and not be used as a number of simulation steps. Figure 10 shows the model used for timescales conversion. At the end of the random interval, a pulse is generated to load the package in the *Slave*.

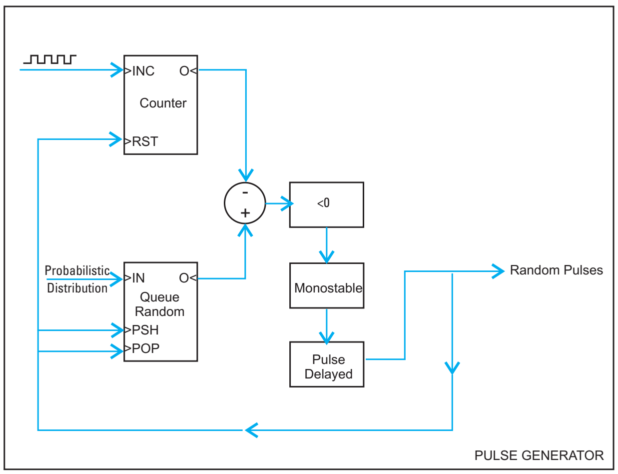

Fig. 10 - Random generation of pulses in the time scale of the model

The *Slave* component is composed of *Slave_CLK, Slave_Message* and *Correction_Module* blocks. This component main function is to keep *Slave_Message* block set from the reference clock generated by the *Master*. Figure 11 displays the *Correction_Module* block.

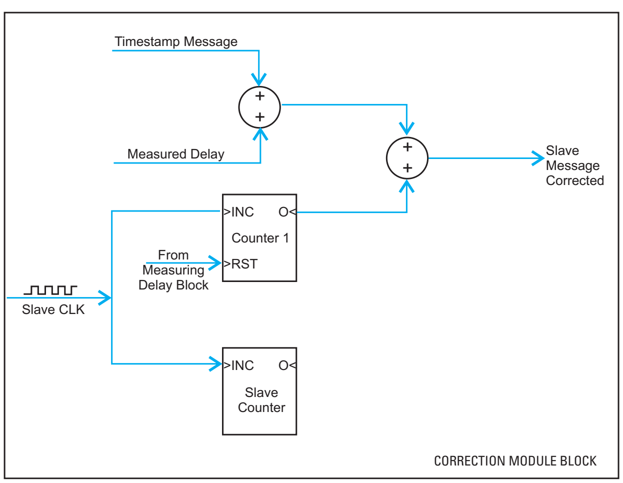

Fig. 11 - Correction Module Block

B. Model Operation

Master_CLK and *Slave_CLK* blocks start their internal counter based on independent signals (oscillators). At the same time the random pulse generator is triggered.

The *Slave_CLK* oscillator is adjusted in a frequency that is slightly lower than the frequency of the *Master_CLK* oscillator, in order to obtain a difference between *Master_CLK* and *Slave_CLK*.

The information generated in *Master_CLK* block is read periodically and inserted in *Message_to_Send* block queue at a rate of 100 Hz. This rate represents the operation rate defined in the IEEE 1588 standard that is up to 128 messages per second.

The message inserted into *Message_to_Send* block queue is stored in its buffer until the time that a random pulse is generated. The generation of control pulses is shown in Figure 7. The random pulse generates the remove signal from the queue of the block *Message_to_Send* causing the message (*Timestamp_Message*) stored in the queue to be sent to the *Correction_Module* block. The duration of this pulse is adjusted to allow a new measuring process to begin keeping the delay with desired probabilistic distribution.

The random generated pulse also triggers a counter whose function is to measure the delay generated in the time scale of the model. This delay is the difference between the input and output of the message queue of the block *Message_to_Send* measured in pulses of the reference clock. When the message stored in *Message_to_Send* block is transmitted, the delay value is stored in the *Latch_Delay* of *Measuring_Delay* block.

As a result of algorithm, a *Timestamp_Message* generated by *Message_to_Send* block and the *Measured_Delay* message generated by *Measuring_Delay* block are both sent to *Correction_Module* block.

The *Correction_Module* block performs a sum of the amounts received through *Timestamp_Message* and *Measured_Delay* messages and then sum this result with the counter value *Counter_1*. This counter begins operation with a slightly different frequency from the frequency of *Master_CLK*, but this counter is reset whenever the *Latch_Delay Measuring_Delay* block provides the *Measured_Delay* message. With that it is possible to determine the correct value of *Slave_CLK* and thus correct it with the received values, thus obtaining the *Slave_Corrected_Message*. This message is compared with the value of *Master_CLK* and thus the correction established by the system is obtained.

From this moment the slave increments its clock based on its local oscillator to the instant of time in which a new message containing the reference to the *Master_CLK* is received and the correction algorithm is executed again.

The *Correction_Module* block is still composed of the *Slave_Counter* whose function is to allow comparison with *Master_CLK* only, thus enabling to make the reading of *Master_CLK* and *Slave_CLK* values without correction.

C. FANET Model

The clock distribution model described in previous section is presently being extended to be useful in the study of FANET environments.

In a FANET the clock distribution path are variable because neighborhood relationships are dynamically set according to spatial distribution of nodes.

A UAV can be in a position where there is no viable route from it to the mesh gate, which receives clock information from the outside world and spreads it inside the FANET. The characteristics times of those communications blackout can be orders of magnitude larger than statistics of delay variation owing to traffic bottlenecks.

Beside that it is necessary to complete the model with transparent clock behavior. Otherwise the wander of slave clocks will be uncontrollable.

Figure 12 shows a high level block model of the FANET simulator.

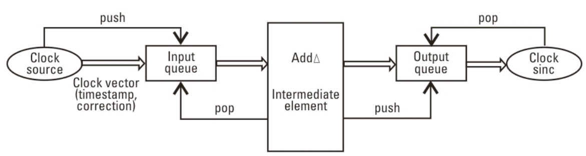

Fig. 12 - FANET Model

Only the forward path is shown in figure 13. As far as a symmetrical delay hypothesis is acceptable this is enough because the reverse correction will be the same as the forward correction. Nevertheless the model developed allows for two-way transmissions.

The correction random variable, Δ, cannot be represented by a continuous density probability function. In fact the model includes a bimodal distribution implemented as a Markov Modulated Poisson Process (figure 13).

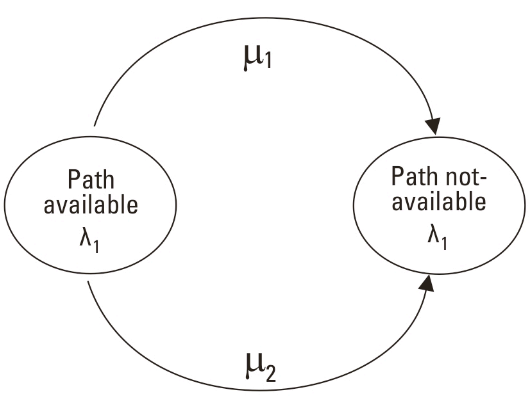

Fig.13 – Delay Estimation Model

λ_1 and λ_2 are characteristics parameters of the time delay distribution in each state. In path not-available state there is no communication so $\lambda 2 = 0$.

μ_1 and μ_2 are related to residence time of each state. They depend on the size of FANET interest area, the number of UAV inside the area and the transmitter reach. This problem has been evaluated in a previous work [17].

The first step to obtain useful results from this model is to adjust the model to fit IEEE 1588 behavior. This will be done in the next section. Results generated by FANET model will be discussed in a future paper.

D. Experiments and Measurements

Uniform and exponential probability distributions were used to carry out the simulations for obtaining message transmission delays in the underlying network which enables the possibility to simulate a bottleneck during transmission over a packet network.

The simulation from a uniform probability distribution aims to verify the behavior of the modeled system in an environment where the packet network has controlled traffic as, for example, a distributed sensor network system in a factory where it is possible to determine and / or control the timing of each transmission as well as the size of each message sent.

The distribution parameters of the random generated number depend on the average delay that is intended to simulate and on the frequency generation of desired messages. The measuring period is given by the product of the average delay (N) by the period of generation of random numbers (T) that is:

$$N.T = \text{Measuring Period} \quad (7)$$

The conducted simulations with uniform probability distribution adopted a measuring period of 0.01 and average delay of 50μs for a unit time simulation equal to 1uS, resulting in T = 0.0002. This ensures that no sampled value by the system is lost, i.e., is not stored in the *Message_to_Send* block queue.

The simulation with an exponential probability distribution aims to study an uncontrolled environment where the delay that each packet suffers has no dependence on the previous delay and therefore cannot be controlled by the sender or receiver of the message. Parameters included are the average N = 0.1 and measuring period of 0.01 resulting in an operating frequency of 10 KHz. Thus the average generation of random numbers is lesser than the operational rate of the system, thus ensuring that all sampled values are inserted in the *Message_to_Send* Block queue.

Figure 14 shows the difference between the master and slave clocks both without correction and with correction obtained in the simulations with uniform and exponential network delay probability distribution. As noted, the correction maintains the difference between the clocks, in both cases, limited.

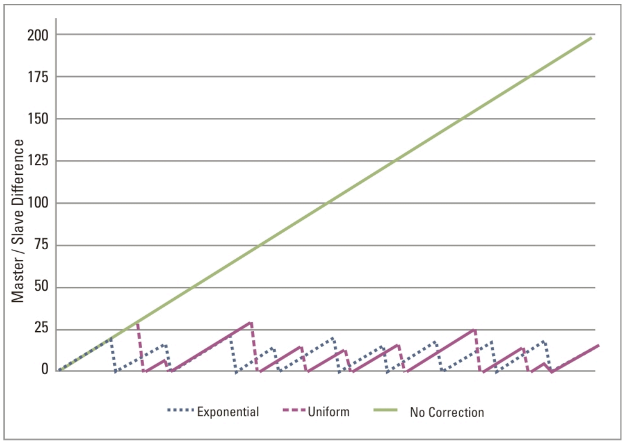

Fig. 14 – Delay Correction – Master / Slave

From obtained results, it is possible to calculate the Maximum Time Interval Error (MTIE) parameter. MTIE is defined as the maximum peak-to-peak delay variation of a given timing signal relative to the ideal signal in a given time interval ($\tau = n\tau_0$) for all observed time within the measured time period [11] [18].The observation time is defined as $\tau = n\tau_0$, being:

τ_0 = *the time-error sampling interval;*

τ = *the integration time*;

n = the number of sampling intervals within the integration time τ.

Equation (8) shows the mathematical definition of the MTIE [7].

$$MTIE\ (T) = \max_{1 \leq k \leq N-n} \left[\max_{k \leq i \leq k+n} x_i - \min_{k \leq i \leq k+n} x_i \right] \quad n=1,2,.....,N-1; \quad (8)$$

$$T = n\tau_0$$

For MTIE parameter calculation it was considered τ_0 = 5 and n = {1,100}. Figure 15 graphically displays the results obtained for the uniform and exponential probability distribution used in the model.

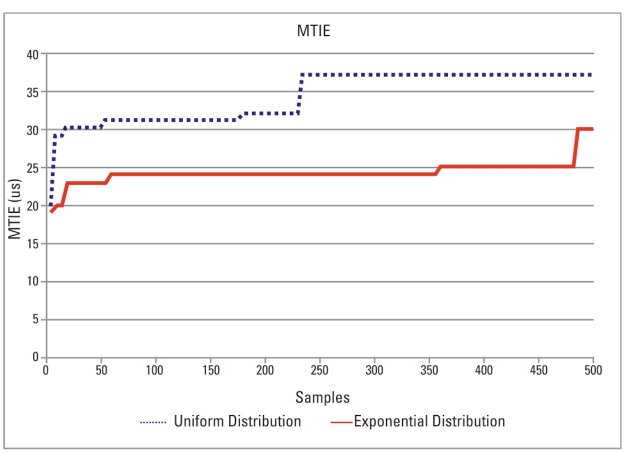

Fig. 15 - MTIE

V. Conclusion

The results of this study demonstrate the feasibility and practicality of the use of discrete-event simulation in the analysis of discrete-time synchronization distribution over packet networked systems. This is a vital step to combine the synchronization distribution model with the proposed FANET model.

The presented quantitative results allow the conclusion that a synchronization system based on the transmission of timestamps in protocol packets and the slave clock adjustment in function of these timestamps can be used for synchronization of sensors networks. Such area is in the goals of the IEEE 1588 standard.

This feasibility is maintained even if the network packet transit presents variable delays in the studied conditions.

More than feasibility of an application, it was possible to analyze the system behavior in times of the order of an hour, which is feasible in this type of system.

FANET environments require higher precision levels that indeed the IEEE 1588v1 standard can reach. Authors' intentions are to improve the simulation model developed to incorporate modules capable of measuring the differential delay of each communication direction and send this information by the messages exchanged between nodes. The influence of node movement and environment conditions is also being evaluated with a two stage Markov Chain where analytical results can be obtained for comparison.

The results already obtained allows one to design simple FANET environments, but the use of discrete event simulation with embedded time – as proposed in this paper – can also yield results for more complex situations.

References

[1] H. C. H. Cho, J. J. J. Jung, B. C. B. Cho, Y. J. Y. Jin, S.-W. L. S.-W. Lee, and Y. B. Y. Baek, "Precision Time Synchronization Using IEEE 1588 for Wireless Sensor Networks," *2009 International Conference on Computational Science and Engineering*, vol. 2, 2009.

[2] J. del Río, D. Toma, S. Shariat-Panahi, A. Mànuel, and H. G. Ramos, "Precision timing in ocean sensor systems," *Measurement Science and Technology*, vol. 23, no. 2, p. 025801, Feb. 2012.

[3] J. Elson, L. Girod, and D. Estrin, "Fine-grained network time synchronization using reference broadcasts," *ACM SIGOPS Operating Systems Review*, vol. 36, no. SI, p. 147, Dec. 2002.

[4] S. Ganeriwal, R. Kumar, and M. B. Srivastava, "Timing-sync protocol for sensor networks," *Proceedings of the first international conference on Embedded networked sensor systems - SenSys '03*, p. 138, 2003.

[5] M. Maróti, B. Kusy, G. Simon, and Á. Lédeczi, "The Flooding Time Synchronization Protocol," in *ACM International Conference on Embedded Networked Sensor Systems*, 2004, pp. 39–49.

[6] M. Anyaegbu, C. Wang, and W. Berrie, "Dealing with Packet Delay Variation in IEEE 1588 Synchronization Using a Sample-Mode Filter," *IEEE Intelligent Transportation Systems Magazine*, pp. 20–27, 2013.

[7] İ. Bekmezci, O. K. Sahingoz, and Ş. Temel, “Flying Ad-Hoc Networks (FANETs): A survey,” *Ad Hoc Networks*, vol. 11, no. 3, pp. 1254–1270, May 2013.

[8] A. Sanna and B. Pralio, “Simulation and Control of Mini UAVs,” in *Proceedings of the 5th WSEAS Int Conf. on Simulation, Modeling and Optimization*, 2005, vol. 2005, pp. 129–135.

[9] O. K. Sahingoz, “Networking Models in Flying Ad-Hoc Networks (FANETs): Concepts and Challenges,” *Journal of Intelligent & Robotic Systems*, vol. 74, no. 1–2, pp. 513–527, Oct. 2013.

[10] N. Aleksander, K. Kamil, D. Krzysztof, and K. Roman, “Differential navigation for UAV platforms with mobile reference station,” in *Selected Topics in Applied Computer Science*, H. Fujita and J. Sasaki, Eds. Japan: WSEAS Press, 2010, pp. 465–471.

[11] N. Networks, “ITU-T G.8261/Y.1361 Timing and Synchronization Aspects in Packet Networks,” 2008.

[12] ITU-T, “ITU-T G.8262/Y.1362 - Timing Characteristics of a Synchronous Ethernet Equipment Slave Clock,” 2010.

[13] T. Committee, T. Tc-, and M. Society, “IEEE Std 1588-2008, IEEE Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems,” 2008.

[14] J. Martin and W. Kasch, “RFC 5905 - Network Time Protocol V4 - Protocol and Algorityhms Specification,” 2010.

[15] G. Astudillo and M. Kadoch, “A Location-aware Routing Scheme for Wireless Mesh Networks using Directional Antennas,” in *Recent Advances in Computer, Communications and Information Technology*, J. Music, Ed. Tenerife: WSEAS Press, 2014, pp. 57–62.

[16] Y. Liu and C. Yang, “OMNeT++ based modeling and simulation of the IEEE 1588 PTP clock,” *2011 International Conference on Electrical and Control Engineering*, pp. 4602–4605, Sep. 2011.

[17] F. C. da S. Eiras, “Benefits of IEEE 802.11e Standard for real-time traffic in WLAN ad hoc networks.,” Biblioteca Digital de Teses e Dissertações da USP, 2009.

[18] ITU-T, “ITU-T G.810 Definitions and temrinology for synchronization networks,” 1996.