Analysis of Synchronization in Distributed Avionics Systems Based on Time-Triggered Ethernet

Nahman Tariq
School of Aerospace, Transport
and Manufacturing (SATM)
Cranfield University
nahman.tariq@cranfield.ac.uk

Ivan Petrunin,
School of Aerospace, Transport
and Manufacturing (SATM)
Cranfield University
i.petrunin@cranfield.ac.uk

Saba Al-Rubaye
School of Aerospace, Transport
and Manufacturing (SATM)
Cranfield University
s.alrubaye@cranfield.ac.uk

Abstract— Significant developments are made in unmanned aerial vehicles (UAVs) and in avionics, where messages sent in the network with critical Time play a vital role. Several studies in Time-Triggered Ethernet have been carried out, but these studies. Still, these improving QoS such as latency in end-to-end delays of an internally synchronized TTE network. However, no one monitors integrated modular avionics. We proposed a framework that enables TTE to be externally synchronized from a GNSS to overcome this problem. We have incorporated our proposed Algorithm in the TTE protocol based on specific parameters and multiple existing algorithms. The proposed Algorithm gives us the ability to control and synchronize the TTE network. Also, we have a developed scenario for analyzing the performance of externally synchronized end-to-end latency of TT messages in a TTE network. We simulated scenarios in our framework and analyzed QoS but, more specifically, the latency that affects the performance of time-triggered messages in externally synchronized TTE networks. The result shows that our proposed framework outperforms existing approaches.

Keywords— Time-Triggered Ethernet (TTE), Time-Triggered Traffic (TT), Integrated Modular Avionics (IMA), Distributed modular Avionics (DIMA), GNSS, External Clock, Quality of Service (QoS), Latency

I. INTRODUCTION

The communication system is an integral part of an avionics architecture. It connects the computational resources and enables sharing of data. There are several requirements for the communication system to handle the communication in an accurate model. First, the communication needs to be fault-tolerant. A Time-Triggered Ethernet (TTE) is a highly reliable network, which supports time-triggered messages to be delivered in a given global time base.

The messages exchanged between UAVs and ground stations, or messages sent within an avionics module, need this information exchange quickly and precisely. Fault-tolerant clock synchronization is the foundation of synchronous architectures such as the Time-Triggered Architecture (TTA) for dependable cyber-physical systems. Distributed Integrated Modular Avionics (DIMA) developed from Integrated Modular Avionics (IMA) and enabled distributed integration through Time-Triggered Ethernet (TTE). End-to-end delay analysis for Time-Triggered (TT) traffic with external Global Navigation Satellite System (GNSS) as the primary time source comprehensive consideration of processing and communication network is necessary to ensure the implementation of externally synchronized global Time.

In TTE based networks, several studies have been carried out to monitor and improve end-to-end delays in TT traffic performance [1] in avionics networks and other fault-tolerant networks. As our focus is integrated avionics module (IMA) or Distributed Avionics Modules (DIMA), In this paper, we look to explore methods for externally synchronizing TTE networks for fault-tolerant applications in avionics and study their impact on quality of service (QoS) focusing on end-to-end latency in the network.

The transmission of TT traffic in TTE adheres to a strict scheduling scheme. However, when TT traffic is sent in the network and misses its pre-allocated time-triggered window, the message must wait for its next scheduled round. This increases end-to-end delays in the TT network, which has crucial importance for real-time performance for TT traffic. In addition, Rate-Constrained (RC) traffic follows an event-triggered mechanism and its transmission latency at network nodes, such as End Systems (ES) and Switches (SW). Therefore, at its worst-case end-to-end delay at the network level need to be analyzed to meet the airworthiness.

Distributed Integrated Modular Avionics (DIMA) is an architectural approach consisting of distributed integrated processing resources connected by a mixed-critical communication network [2]. Integrated Modular Avionics (IMA) inherits flexibility, modularity, and interoperability; this improves coupling, barrier, hierarchical isolation, and distributed integration, providing solutions to future avionics architecture[3]. Time-Triggered Ethernet (TTE) can be applied as the communication network for DIMA systems since it possesses a time integrity guarantee mechanism, matches the processing isolation of partitions and supports mixed critical traffic, including Time-Triggered (TT) traffic, Rate-Constrained (RC) traffic and Best-Effort (BE) traffic [4].

This paper explores methods for introducing external Time from a GNSS in a TTE based network. Furthermore, we explore the effects of this external clock synchronization in the network. The main contribution of this paper is the analysis of the system latency and recommendations for improvement of the overall performance and quality of service (QoS), e.g., by simulating scenarios supporting external clock GNSS in the time-triggered

traffic in either an IMA or a DIMA as the primary time source. In this paper, we developed scenarios where we analyzed and identified QoS and concentrated on latency that affects the performance of externally synchronized TTE networks supporting fault-tolerant traffic. These parameters help us monitor and improve QoS performance in IMA, DIMA with external (by GNSS) clock synchronization. Experimental results present that our proposed system model and analysis approaches have obvious advantages compared to the traditional method.

In TTE based networks, several studies have been carried out to monitor and improve end-to-end latency delays in TTE based avionics networks and other fault-tolerant networks[5][6]. Although our focus is avionics, we look to explore methods in an externally synchronized TTE network for fault-tolerant applications in avionics to develop and study their impact. Current research focuses on utilising TTE networks based on internal clocks and discusses methods to improve delays in end-to-end traffic[1][6]. Furthermore, a considerable barrier exists when analyzing the TTE network with external clocks being synchronized for TT messages.

The simulations performed help us analyze effects on QoS such as latency whilst end-to-end monitoring delays in the network. The focus here was on how the QoS is measured in the TTE network, in IMA, and DIMA having an external clock synchronization via GNSS. With external synchronization, we are also able to control the TTE network from a ground station.

This paper is organized as follows. After a short introduction about the background knowledge in Section II, we discuss the proposed Algorithm for externally synchronizing the TTE network in Section III; respectively, we analyze the results in section IV, results and analysis are included. Finally, conclusions are presented in Section V.

II. BACKGROUND STUDY

Detail of background study is discussed in the following sections.

A. Time-Triggered Ethernet

TTEthernet extends the traditional Ethernet standard, with additional services that guarantee reliable, deterministic delivery of time-critical messages.

In the TT network, time synchronization only needs to be configured if time-triggered traffic is transmitted in the network. A TTE network, according to the SAE AS6802 (Time-Triggered Ethernet) standard, can have multiple synchronization domains and multiple synchronization priorities [4]. TTE is a time-triggered communication technology in the switched interconnection environment, consisting of End Systems (ES), Switches (SW), and full-duplex physical links. It is highly suitable for safety-critical avionics systems because it supports mixed-criticality communication, including TT, RC and BE traffic. TT traffic has the highest priority, and its transmission follows a static scheme to avoid contentions and guarantee determinism; asynchronous RC traffic has pre-defined bandwidth to ensure transmission. BE traffic has the lowest priority and supports the service of IEEE 802.3 Ethernet.

The time-triggered protocol uses a pre-defined clock which allows the entire network to synchronize with the autonomous Time. The time-triggered clock-synchronization protocol maintains the synchronized global Time [7]. The operation of any time-triggered system depends on the availability of a global time base. Any loss of the time base is tantamount to system failure. In safety-critical applications, the establishment of a fault-tolerant time-base of a priori known precision is thus a definite requirement [8].
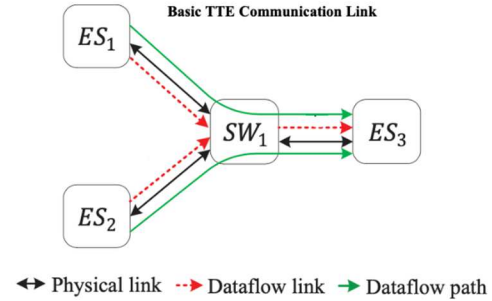

Figure 1:TTE Network Data Flow

An important aspect when dealing with flying objects is the data link QoS. Therefore, IMA and DIMA are considered reliable data transmission between the components; thus, the communication path should be available at any given time with real-time synchronization.

Time synchronization: It is possible to synchronize a TT switch and all connected components to an external timing source. This timing source can be either a self-timed (autonomous) data source or an external reference.

A TTEthernet end system implements services to synchronize its local clock with the local clocks of other end systems and switches. TT messages are used when latency, jitter, and determinism are required. All TT messages are sent at pre-defined times.

Since the TTE network supports all three traffic types, one of the key protocols used for synchronization is the frame format of these traffic classes introduce protocol control frames (PCF) used for synchronization PCFs are subject to network jitter.

B. Virtual Links (VLs)

Virtual Link (VLs) is a significant concept of the avionics network environment, and it is responsible for routing data and controlling data transmission rate. End systems (ESs), which are mutually linked with limited switches (In a TTE network, only a certain number of physical switches can be connected) and physical connections, differ from physical links. The VL is a bidirectional logic path based on the physical links; it links one ES to more ESs. VLs are rate limited whose dedicated bandwidths are decided by the two parameters, Bandwidth Allocation Gap (BAG) and the most significant frame length *Lmax*. The values of BAG provide a limited dynamical range for flow control. Furthermore, BAG is application-specific and suitable for scheduling parameters in avionics networks [1].

C. Clock Synchronization in TTE

The operation of any TT system depends on the availability of a global time base. Any loss of the time base is tantamount to system failure. If the TTE network does not send the messages based on the corrected clocks, the system is not fault-tolerant, so any loss in time base means that the TTE system is not synchronized and will perform as a fault-tolerant system. In safety-critical applications, the establishment of a fault-tolerant time-base of a priori known precision is thus a definite requirement [9]

Once the start-up protocol has initially established the synchronized time base, the clock synchronization protocol is periodically executed to resynchronize the local clocks in the end systems and switches.

The TTEthernet clock synchronization algorithm operates in two steps. In the first step, all SMs concurrently send PCFs to all CMs at the same local time.

In the second step, all the SMs and SCs use the reception times of the compressed PCFs from the CMs to derive a new reference point in Time for synchronization.

TTE specifies and standardizes several SAE AS6802 synchronization protocols that ensure close synchronization of the local clocks in the ES and switches during all network modes of operation. Designs of the protocols are in place to tolerate the failure of ES and switches in the TTE network.

D. Integrated Modular Avionics and Distributed Modular Avionics

IMA is an established architectural methodology that has been successfully applied in various aircrafts such as A380 and B787[1]. IMA considers multiple requirements and combinations of relations of several avionics applications and distributes it to desired sections.

DIMA is an approach in the scope of IMA with a necessary amendment. It combines the federated and the IMA concept [10] by physically distributing the integrated modules and connecting them with a fault-tolerant communication system. This can be seen in figure 2. The DIMA system is a distributed IMA system that interconnects and corresponds employing the time-triggered network based on an accurate global clock.

We analyze TT traffic in a TTE based DIMA and IMA system. We have created a reference network that allows the network to be simulated and correct all clocks internally. We then simulate a scenario of the same network with a GNSS connected which provides all the nodes with external timing. The TTEthernet clock will require rate corrections to compensate for differences in the reference clock rates (GPS vs TTEthernet). This can help to synchronize different local TTEthernet networks forming a worldwide distributed network(WAN).

III. PROPOSED ALGORITHM

As more and more systems rely on time-dependent networks for quick and precise delivery of messages, time-critical networks are being used. The need for fault-tolerant networks in avionics serves a critical role, with the development of TTE based on AIRNC 664 part 7 forming the basis of TTE that supports a precise and reliable fault-tolerant network [5]. The advantage of utilizing a TTE network is its use of TT messages in a TTE network relying on precise Time for the messages delivered in the network. On the other hand, TT traffic relies on internal clocks as its message delivery system in the network. This provides us with the opportunity to explore how TTE networks can utilize an external time source such as a GNSS. In a TTE network, there are three types of traffic, TT messages have the highest priority, followed by rate constrained (RC) traffic, and lastly, best effort (BE) traffic is sent in the network. A systematic analysis of correlations between TTE networks with external clock synchronization (by GNSS) for message delivery and their impact on controlling avionics networks is a considerable challenge.
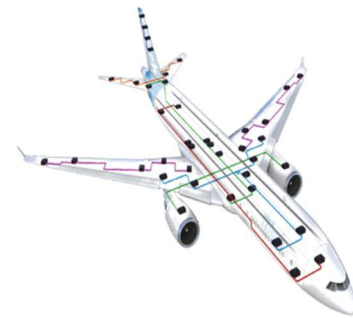

Figure 2:Deterministic Ethernet and TTE Networks for IMA & DIMA

A. Proposed Algorithm

We take full advantage of some of the built-in parameters and existing algorithms in the TTE network. The TTE network provides the ability to connect an external GNSS via means of a USB. This allows us to connect an external GNSS device to one of the ESs in the network to synchronize the entire network externally. We program the GNSS receiver to interact with the existing Algorithm of the TTE protocol. Here we propose an algorithm that builds on an existing algorithm and supports the TTE network for externally synchronized clocks for TT messages delivered in an IMA. We base this Algorithm on a wired TTE network which is also connected to a GNSS. In figure 4, we see how the Algorithm interacts with multiple ES in the network and TT traffic is sent in the network with an externally synchronized GNSS clock.

The Algorithm we propose builds on the pre-existing Algorithm for the TTE protocol. We can utilize some of the existing algorithms in the TTE protocol and build on them in the proposed Algorithm. We also had to modify specific parameters manually before getting the external Time into the TTE network. Our proposed Algorithm can pre-calculate delays once the Time is received from the GNSS and processed to the ES. Once this time delay is calculated, we allow the network to synchronize the entire network with GNSS Time. Some of the TTE protocol's algorithms are used, but some are modified to allow external

Time in the network. More on the actual protocol of TTE can be read here [9] [11] [12]and [13]

The protocols that detect failures in the synchronization are commonly known as clique detection algorithms. The Algorithm also uses a clique detection algorithm. In the synchronized states (SM_TENTATIVE_SYNC, SM_SYNC, and SM_STABLE state), TTEthernet uses clique detection algorithms to detect all blackout scenarios or loss of network reliably. Three types of clique detection algorithms are implemented that can be active in parallel [14]. . The three clique detection algorithms use synchronous clique detection algorithms, the asynchronous clique detection algorithms, and the relative clique detection algorithm.

Message exchanges do the synchronization. Specific Ethernet frames called PCF (i.e. Protocol Control Frames or "Integration Frames") are exchanged and used to perform all synchronization functions. These specific messages are exchanged every integration cycle (re-synchronization interval). Each Integration Cycle, the Synchronization Masters (SM) will send specific PCF to the Compression Masters (CM). The CMs perform the compression function based on the arrival point in Time of the PCFs sent by all SM.
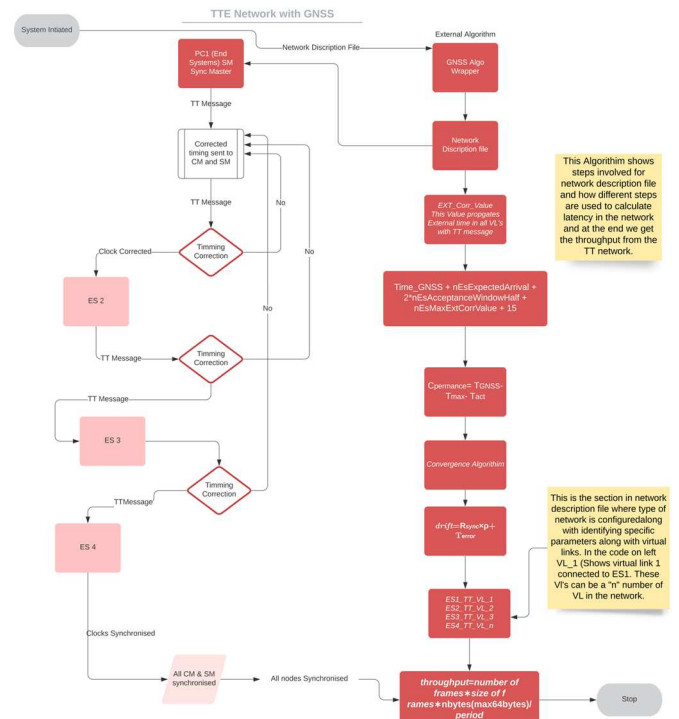

Figure 3:TTE network with proposed Algorithm.

Figure 3 depicts how utilizing the proposed Algorithm w, e achieve external clock synchronization in a TTE network used for IMA. This externally synchronized network controls both IMA and distributed integrated modular avionics (DIMA) networks. We developed a scenario based on figure 3 to utilize the proposed Algorithm, which supports both IMA and DIMA. For example, an IMA or DIMA and a similar scenario were developed for communicating with ground stations having TT traffic in the network. In figure 3, we can see the flow of TT

messages in an IMA environment; this allowed us to monitor the performance, such as QoS latency of the IMA and DIMA.

In figure 3, we look at the steps involved in the proposed Algorithm. We initially have a description file to define the network type and number of end systems being used. In the second step, we connect our GNSS to one ES, which uses the proposed Algorithm's code to connect GNSS and the network description file. In the following stages, our proposed Algorithm synchronizes the entire network with the external clock. However, one downside of this external synchronization is that if the desired TTE network is synchronized correctly and clocks are not matched, TT messages will not be sent hence a failure in the network. However, our proposed Algorithm overcomes the impact of mis-synchronization. Our proposed Algorithm is discussed more in the following steps.
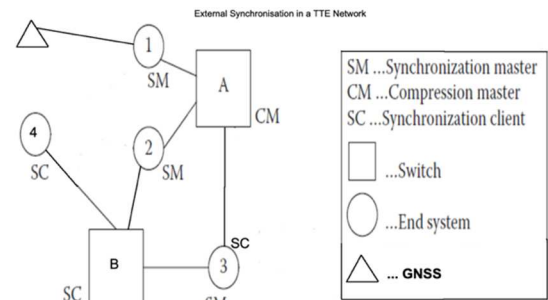

Figure 4: TTE network data flow with external GNSS.

In Figure 4, we see a brief overview of how external GNSS is connected to the TTE network. We base our scenario on figure 3 for simulation. algorithm. In figure 3, we can control a TTE network. We first have our network description file in which we describe a type of network being simulated. Next, we make sure the ES is connected to an external GNSS for synchronizing all other ES and switches in the network for TT messages. In our proposed Algorithm, we calculate initial GNSS delays in the network. In the second step, we modify some specific TTE network parameters to enable external clock synchronization. Finally, we modify and build on the existing protocol. This is discussed in more detail. These steps initially calculate delays between the GNSS receiver to the ES, ES synchronizing the TT messages with the external global time base. Then we configure the number of integration cycles in the TTE network, which allows the clocks to be corrected, as shown in figure 5. Integration cycles allow the scheduler to guarantee that TT frames are sufficiently scheduled. Once all the clocks in ES are corrected, we then calculate the drift from the GNSS clock. Once these drifts are calculated, we proceed to the final step to send the TT message in the network based on externally synchronized clocks as the primary time source. We need to calculate the delay from the GNSS to ES initially. Once this is achieved, we build on the existing Algorithm for the TTE protocol. This can also be in figure 4 as to how our proposed Algorithm works.

In (1), we start the Algorithm to calculate and connect the GNSS to ES. In this, we calculate the Time taken from a GNSS to ES calculating delays before the TTE network is synchronized with an external clock.

```
Algorithm 1 for Connecting GNSS and Getting GNSS Time
Input GNSS Time
Output GNSS Time = x
1    flag = False
2    digist_of_time = 0
3    time = []
4    time_index_previous = 0
5    print_flag = 0
6    time_index = 1
7    raw_data = ''
8    pre_data = b''
9    pre_data = pre_data + raw_data
13       pre_data_flag = pre_data[length - 7:-1]
14       if pre_data_flag == b'GNRMC': // the name of constellation being used from GNSS
15          flag = True
16          digist_of_time = 9
17       if digist_of_time == 0:
18       else:
19          time.append(pre_data[-2:-1])
20       if len(time) >= 6 and len(time) % 6 == 0
24          if time_index_previous < time_index:
27             print(time)
```
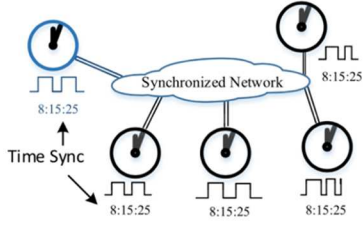——————————————————————— (1)

Figure 5: Externally Synchronised TT Network with corrected clocks [15]

TTE network has physical connections that allow the system to integrate external devices with the TTE network as the ES has standard input and output ports as a typical computer system except a PCIe Ethernet modules known as ICD ethernet modules which allow the time-triggered communication to take place in the network.

```
Algorithm 2 Connection between ES and GNSS with Delay Calcuation
Input GNSS Time
Output GNSS Time = x
1    Time_GNSS + nEsExpectedArrival + 2*nEsAcceptanceWindowHalf + nEsMaxExtCorrValue + 15
2    end
```
——————————————————————— (2)

The above Algorithm gives Time from GNSS to be imported into the TT network. The above (2) nES (N number of ES) represents the number of ES in the network. The first step is the expected arrival of Time from the GNSS. This is then added multiplied by 2 with n number of ES and their acceptance window. This is to allow for the messages to be accepted in the network. In the next phase, the maximum correction value is added to the ES in the network with the final addition of 15 micro-seconds. This helps us calculate the network's overall delay when we are sending TT messages across the network. To calculate the drift, we use nEsMaxExtCorrValue. This will limit the external clock correction value, which then can be applied per integration cycle.

We also need to calculate the permeance delay in the network; we need to calculate the difference between actual and maximum Time consumed (TC) of the network and external clock for the PCF being on a multi-hop network. To calculate this, we use (3)

```
Algorithm 3 Calculating Convergence and Permeance Delay
Input GNSS Clock
Output GNSS Delay = x
     C^permance  =  T^GNSS  −  T^max  −  T^act
1    If |SM_T^GNSSClock| = 1 then
2       CM_T^GNSSClock_j ← SM_T^GNSSClock_1
3    Else if |SM_T^GNSSClock| = 2 then
4       CM_T^GNSSClock_j ← SM_T^GNSSClock_1 + SM_T^GNSSClock_2 / 2
5       average of the (k + 1)^th largest and (k + 1)^th smallest clocks, where k is the number of faulty SMs to be tolerated
6    End if
```
(3)

We perform two more steps in the TTEthernet clock synchronization algorithm. In the first step, the SMs send PCFs to the CMs. From the arrival points in Time of these PCFs, the CMs extract the current state of the SMs external clocks. We denote these values by *SM_TGNSSClockn* (Synchronization Master with GNSS time sent to n number of nodes or ES), where $1 \leq n \leq |SM\ a\ CMj$ uses a variant of the fault-tolerant median to calculate the new "compressed" clock *CM_TGNSSClockj*.

```
Algorithm 3 Calculating Convergence and Permeance Delay
Input GNSS Clock
Output GNSS Delay = x
     C^permance  =  T^GNSS  −  T^max  −  T^act
1    If |SM_T^GNSSClock| = 1 then
2       CM_T^GNSSClock_j ← SM_T^GNSSClock_1
3    Else if |SM_T^GNSSClock| = 2 then
4       CM_T^GNSSClock_j ← SM_T^GNSSClock_1 + SM_T^GNSSClock_2 / 2
5       average of the (k + 1)^th largest and (k + 1)^th smallest clocks, where k is the number of faulty SMs to be tolerated
6    End if
```
(3)

The above Algorithm (3) takes the external Time and uses it within the primary system and sends it to CM and SMs in the network.

In (4), we calculate the PCF received in the CM. We calculate maximum and minimum bits in the PCF with external time sources.

```
Algorithm 4 Calculating PCF with Max and Min Bits in CM
Input PCF bits
Output PCF Delay = j
1    For j=1→|CM| do
2    if current_max < bits(pcf_membership_newj)
3    then
4       current_max ← bits(pcf_membership_newj)
5    end if
6    For
7    accept_threshold:−current_max−conf_faulty_SM
8    return {CM_T^GNSSClock | pcf_memberhip_new ≥ accept_threshold}
```
(4)

In the last stage of the proposed Algorithm, we look at ways to calculate the overall drift in the network we calculate the drift in the network by using the Algorithm in (5).

```
Algorithm 5 Calculating Convergence SM and Final Drift
Input GNSS Clock
Output GNSS Delay = x
1    if |select (CM_TGNSSClock)| = 1 then
2       act_corr ← SM_T^GNSSClock_1 - CM_T^GNSSClock
3       SM_T^GNSSClock - CM_T^GNSSClock_1
4    Else if
5    |select(CM_T^GNSSClock)| = 1 then
6       act_corr ← SM_T^GNSSClock_1 - CM_T^GNSSClock_1 + CM_T^GNSSClock_2 / 2
7       SM_T^GNSSClock_1 ← CM_T^GNSSClock_1 + CM_T^GNSSClock_2 / 2
8       Else
9    End if
10   drift = Rsync × ρ + Terror
```
——————————————————————— *(5)*

In (5), we represent R_{sync} as the integration cycle and ρ for the drift rate. We use the value T_{error} to summarise other factors in the clock-synchronization process (e.g., network jitter, inaccuracies from the clocks not flawlessly executing the integration cycles at the same TimtimeThese are some of the

critical steps which are involved in getting the external clock of a GNSS into a TT network.

IV. RESULTS AND ANALYSIS

To illustrate our proposed Algorithm having externally synchronized clocks in the TTE network with design and validate the effectiveness of our proposed analysis. We developed the Algorithm based on a pre-existing set of TTE algorithms. The simulations performed help us analyze effects on QoS such as latency whilst monitoring end-to-end delays in the network. The focus here is on how the QoS is measured in the TTE network, in IMA, and DIMA having an external clock synchronization via GNSS.

With the developed scenarios, we analysed and identified parameters QoS (latency) that affect the performance of externally synchronized TTE network supporting fault-tolerant critical traffic. These parameters help us compare QoS performance in IMA, DIMA with external (by GNSS) clock synchronization. Furthermore, the data gathered about fault-tolerant critical traffic in avionics from different scenarios and simulations; we identified the effects on TT messages for end-t0-end latency in an IMA or DIMA.
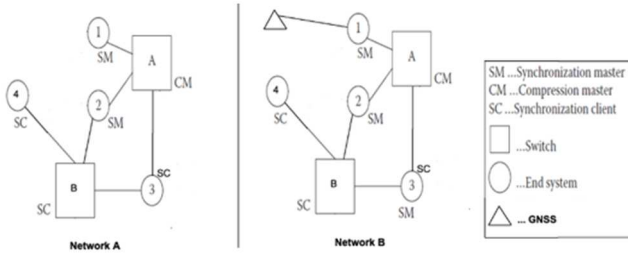

Figure 6: Network A and Network B.

In figure 6, we describe the two networks on which we based our scenario for simulation. The figure represents networks A and B. Network A is a reference TTE network with two switches and 4 ES, and all clocks are internally synchronized so that an autonomous time base corrects all nodes and their clocks. Whereas Network B has an external GNSS connected to one of the ES and the entire network is externally synchronized and the clocks of all nodes in network B are corrected with an external time base. Since the GPS time information and the TTEthernet relative time information are drifting, the TTEthernet time must be aligned to the GPS time. This is a periodic task and is executed in a control loop as described. Therefore, the external correction applied on the TTEthernet synchronized clock can only be applied within bounds.

Figure 7 presents a latency node histogram of the reference TTE network. In figure 7, two networks are shown; the networks are differentiated with green and yellow cocoloursThe yellow bar represents Network A which has TT messages based on the internal clock being scheduled. At the same time, the green bar represents Network B, an externally synchronized network utilizing a GNSS. It can be observed that in Network B, when the TT traffic is sent to the network, the end-to-end delay is quite significant. Although this is not an ideal result, all nodes are not fully synchronized in network B. Hence the delay in the network

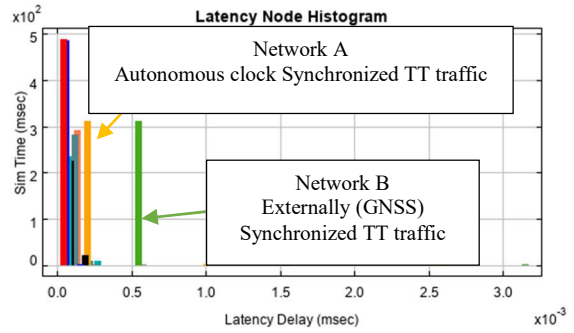is significant; this is the part focusing on how we can improve end-to-end latency.

Figure 7:TT Traffic Delay with the system clock and GNSS clock.

Figure 7 shows us a latency between both networks in the form of a stream plot for better understanding.
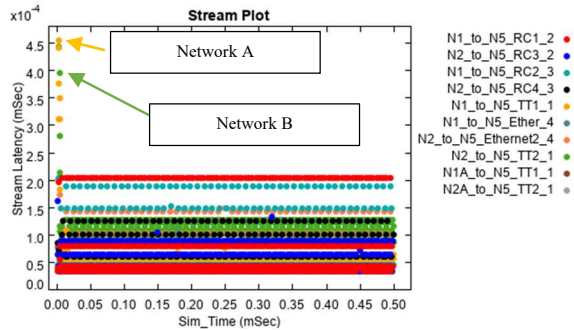

Figure 8:Stream Plolatencyatency in Network A and Network B.

In Figures 7 and 8, it can be observed that delay in delivering TT messages in network B is an externally synchronized network as the end-to-end latency is high. It means the network is not a fault-fault-tolerant to be used for delivering TT messages. But with running, however, simulation and introducing more integration cycles, we improved the QoS and reduced the end-to-end latency in the network for delivering TT messages in IMA or DIMA. The following figures will showcase the latency reduction between networks A and B. (Network A TT is represented by yellow, Network B is represented by the green bar).

We do not focus on RC and BE traffic in the network as it has given the least priority based on TT scheduling of the message [16].
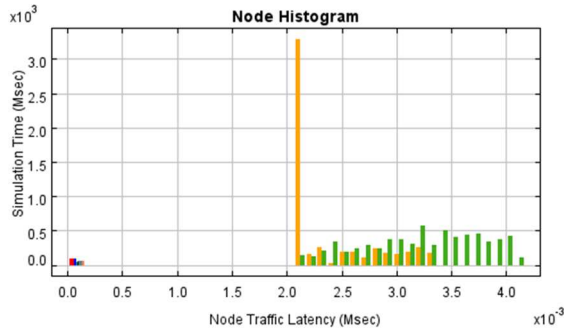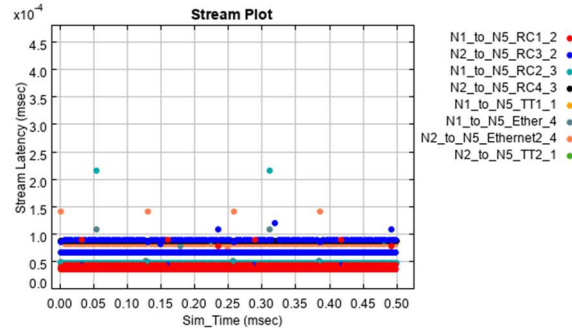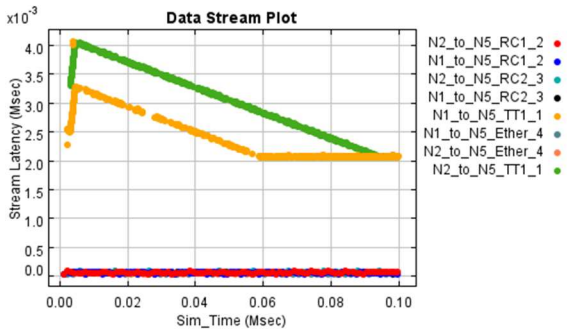
Figure 9:Latency TT traffic in Network A and B.

Figure 10:Latency Stream Plot with Increased integration cycle.

Figures 9 and 10 show that by utilizing the proposed Algorithm for external clock synchronization, we can increase the number of integration cycles. This integration cycle is built into the Algorithm such that the end-to-end latency can be decreased for TT messages when clocks are externally synchronized. However, the figures show the effects that network B with external clock synchronization latency in the network and the delay is around 2.5 msecs which is relatively high and is not an ideal outcome. One key point to consider is the data size of TT messages being sent in the network. Also, we can see that an externally synchronized network has latency when more messages are scheduled in the TT network.

Figure 10 shows the histogram of nodes connected to both internally synchronized clocks and externally synchronized clocks. Again, we can see that both network A and network B have decreased latency in the network. Again, this is because both networks are within the bound of a fault-tolerant network.
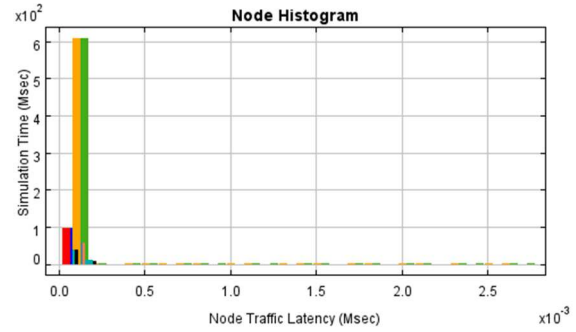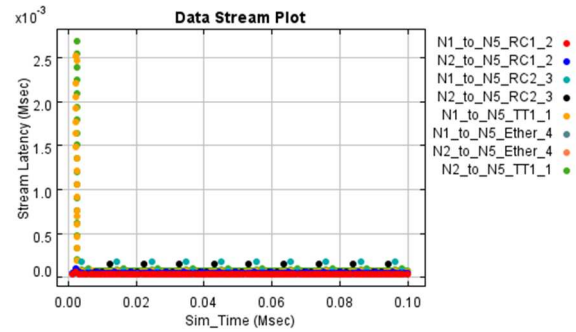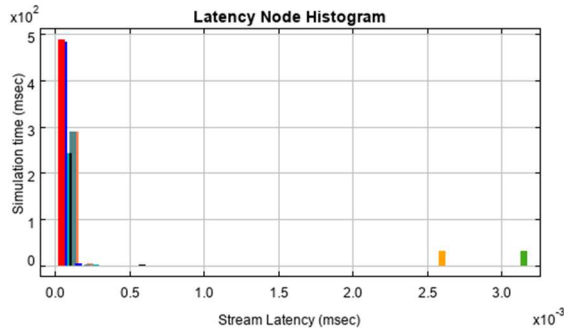

Figure 11:Latency in Network A and Network B.

Figure 12:Stream Plot for Latency in Networks.

Figure 11 and Figure 12 represent how TT messages are affected in both networks A and B. This end-to-end delay is due to system clocks not being fully synchronized.

Figure 13:Decreased Latency in Network A and B.

Figure 14:Stream Plot of Network and Network B decrease latency.

Above in both figures, we can see that using the proposed Algorithm can improve latency in Network A and Network B and bring latency of both networks to around 56.36 nanoseconds. Furthermore, as we have demonstrated, our scenarios can utilise the GNSS clock as the primary source for TT messages in the network, and that end-to-end latency has decreased whilst carrying the TT messages in the network. We can also see these results in a table (1), where Network A and Network B can be easily read.

TABLE I.   SIMULATION RESULTS IN END-TO-END LATENCY NETWORK A

Network A	End-to-End Latency Network A (Internally Syncornized Clocks)		
	Minimum Latency	Maximum Latency	Throughput Mbps
Sim 1	55.4616 us	2.6933 ms	9.5866
Sim 2	2.0494 ms	4.0638 ms	10.8744
Sim 3	2.5163 ms	2.5163ms	9.5896
Sim 4	56.3767 us	2.5253 ms	9.6084

TABLE II. SIMULATION RESULTS END-TO-END LATENCY NETWORK B (WITH GNSS CLOCKS)

Network B	End-to-End Latency Network A (Externally Syncornized Clocks)		
	Minimum Latency	Maximum Latency	Throughput Mbps
Sim 1	115.4616 us	2.6933 ms	9.5866
Sim 2	2.0482 ms	4.0478 ms	10.8744
Sim 3	3.0163 ms	3.022 ms	9.5896
Sim 4	56.3767 us	2.7774 ms	9.6084

The above Table I and Table II represent the results achieved during the simulation process of both Network A and B. proposed algorithm has a clear benefit that once the clocks are synchronised in the entire network, we can reduce the end-to-end latency of TT messages in the network. Based on our proposed Algorithm, we can see the impact it has in achieving desired results.

With the proposed algorithm, we can see that once the entire network's clocks are externally synchronized, we can improve and achieve end-to-end latency of TT messages. This furthermore gives the ability to control multiple TTE local networks and form a Wide Area Network (WAN), which can be controlled and monitored at any given time from any location.

V. CONCLUSION

In our, work we have analyzed parameters such as end-to-end latency of TT traffic in an IMA or a DIMA-based TTE network with clocks externally synchronized. The simulations results show that our proposed work decreases latency when TT messages are externally synchronized by utilizing a GNSS. These results also highlight the capability of the TTE network being controlled from a ground station allowing us to monitor the network wan with an externally synchronized network. In the future, we will extend our work supporting multiple GNSS in multiple ES.

REFERENCES

[1] X. Zhou, F. He, H. Xiong, and E. Li, "Improving End-To-End Delay Analysis for TT and RC Traffic of TTEthernet in DIMA Systems," AIAA/IEEE Digit. Avion. Syst. Conf. - Proc., vol. 2019-Septe, 2019, doi: 10.1109/DASC43569.2019.9081706.

[2] D. K. Yadav, C. S. Lamba, and S. Shukla, "A new approach of leader election in distributed system," 2012 CSI 6th Int. Conf. Softw. Eng. CONSEG 2012, pp. 1–7, 2012, doi: 10.1109/CONSEG.2012.6349512.

[3] X. ZHOU, H. XIONG, and F. HE, "Hybrid partition- and network-level scheduling design for distributed integrated modular avionics systems,"

Chinese J. Aeronaut., vol. 33, no. 1, pp. 308–323, 2020, doi: 10.1016/j.cja.2019.08.027.

[4] K. Bisson, "Sae As6802 Deterministic Ethernet Network Solution," Avion. Interface Technol., no. March, pp. 1–20, 2011, [Online]. Available: http://www.aviftech.com/.

[5] M. Jakovljevic and J. Radke, "Design and Verification for Complex Deterministic Ethernet Networks in IMA Systems," SAE Tech. Pap., vol. 2015-Septe, no. September, 2015, doi: 10.4271/2015-01-2527.

[6] J. Yao, J. Wu, Q. Liu, Z. Xiong, and G. Zhu, "System-Level Scheduling of Mixed-Criticality Traffics in Avionics Networks," IEEE Access, vol. 4, pp. 5880–5888, 2016, doi: 10.1109/ACCESS.2016.2584859.

[7] P. G. Peón, H. Kopetz, and W. Steiner, "Towards a reliable and high-speed wireless complement to TTEthernet," 19th IEEE Int. Conf. Emerg. Technol. Fact. Autom. ETFA 2014, 2014, doi: 10.1109/ETFA.2014.7005311.

[8] K. Steinhammer, P. Grillinger, A. Ademaj, and H. Kopetz, "A Time-Triggered Ethernet (TTE) switch," Proc. -Design, Autom. Test Eur. DATE, vol. 1, 2006, doi: 10.1109/date.2006.244145.

[9] H. Kopetz and P. Grillinger, "The Time-Triggered Ethernet (TTE) Design."

[10] J. Rushby, "Partitioning in Avionics : Requirements , Mechanisms , and Assurance," 2000.

[11] W. Steiner and G. Bauer, "Mixed-criticality networks for adaptive systems," AIAA/IEEE Digit. Avion. Syst. Conf. - Proc., pp. 1–10, 2010, doi: 10.1109/DASC.2010.5655319.

[12] TTTech, "TTEthernet – A Powerful Network Solution for All Purposes," pp. 1–15, 2009.

[13] W. Steiner, G. Bauer, B. Hall, and M. Paulitsch, "Time-triggered ethernet," in Time-Triggered Communication, 2011.

[14] H. Z. Csc, P. By, and P. Baxi, "Time-Triggered Ethernet," 2016.

[15] K. F. Hasan, C. Wang, Y. Feng, and Y. C. Tian, "Time synchronization in vehicular ad-hoc networks: A survey on theory and practice," Veh. Commun., vol. 14, pp. 39–51, 2018, doi: 10.1016/j.vehcom.2018.09.001.

[16] N. Tariq, I. Petrunin, S. Al-Rubaye, A. Tsourdos, and W. Zischka, "Quality of service study in synchronized time-triggered aerial networks," in AIAA/IEEE Digital Avionics Systems Conference - Proceedings, 2020, vol. 2020-October, doi: 10.1109/DASC50938.2020.9256480.