Formal Guarantees of Timely Progress for Distributed Knowledge Propagation

Saswata Paul Stacy Patterson Carlos Varela
Rensselaer Polytechnic Institute, Troy, New York, 12180, USA
pauls4@rpi.edu {sep,cvarela}@cs.rpi.edu

Autonomous air traffic management (ATM) operations for urban air mobility (UAM) will necessitate the use of distributed protocols for decentralized coordination between aircraft. As UAM operations are time-critical, it will be imperative to have formal guarantees of progress for the distributed protocols used in ATM. Under asynchronous settings, message transmission and processing delays are unbounded, making it impossible to provide deterministic bounds on the time required to make progress. We present an approach for formally guaranteeing timely progress in a Two-Phase Acknowledge distributed knowledge propagation protocol by probabilistically modeling the delays using theories of the Multicopy Two-Hop Relay protocol and the M/M/1 queue system. The guarantee states a probabilistic upper bound to the time for progress as a function of the probabilities of the total transmission and processing delays being less than two given values. We also showcase the development of a library of formal theories, that is tailored towards reasoning about timely progress in distributed protocols deployed in airborne networks, in the Athena proof assistant.

1 Introduction

The integration of *uncrewed aircraft systems* (UAS) in the *National Airspace System* (NAS) for *urban air mobility* (UAM) operations will pose significant challenges for urban air traffic management in the near future. The UAS will need to operate in highly congested uncontrolled urban airspaces to perform tasks ranging from package delivery to passenger transportation. Under such circumstances, it will be a complex task to ensure *safe separation* [10] among the aircraft. The loss of safe separation can have catastrophic consequences such as *near mid-air collisions* (NMAC) [34] and *wake-vortex induced rolls* [40]. Centralized techniques for air traffic management, that require either human *air traffic controllers* or dedicated *ground stations* for coordinating the safe operation of aircraft, will be insufficient as they are prone to human errors [58] and failures, and are economically infeasible [7]. Therefore, there is a need for developing decentralized *UAS traffic management* (UTM) protocols and standards for future UAM operations that will allow aircraft to autonomously coordinate and maintain safe separation. Such autonomous techniques will be amenable to *formal verification*, making them suitable for safety-critical UAM applications.

Formal methods [59] can be used for the rigorous verification of critical properties of autonomous UTM applications by using techniques like *model checking* and *theorem-proving*. However, the operational conditions presented by UAM are expected to be highly dynamic in nature, making it infeasible to exhaustively model them. Under such circumstances, it is possible to probabilistically reason about these conditions to provide valuable non-deterministic guarantees that can be used by the participating aircraft to take important real-time operational deci-

Submitted to:
FMAS 2021

© S. Paul, S. Patterson, & C. Varela
This work is licensed under the
Creative Commons Attribution License.

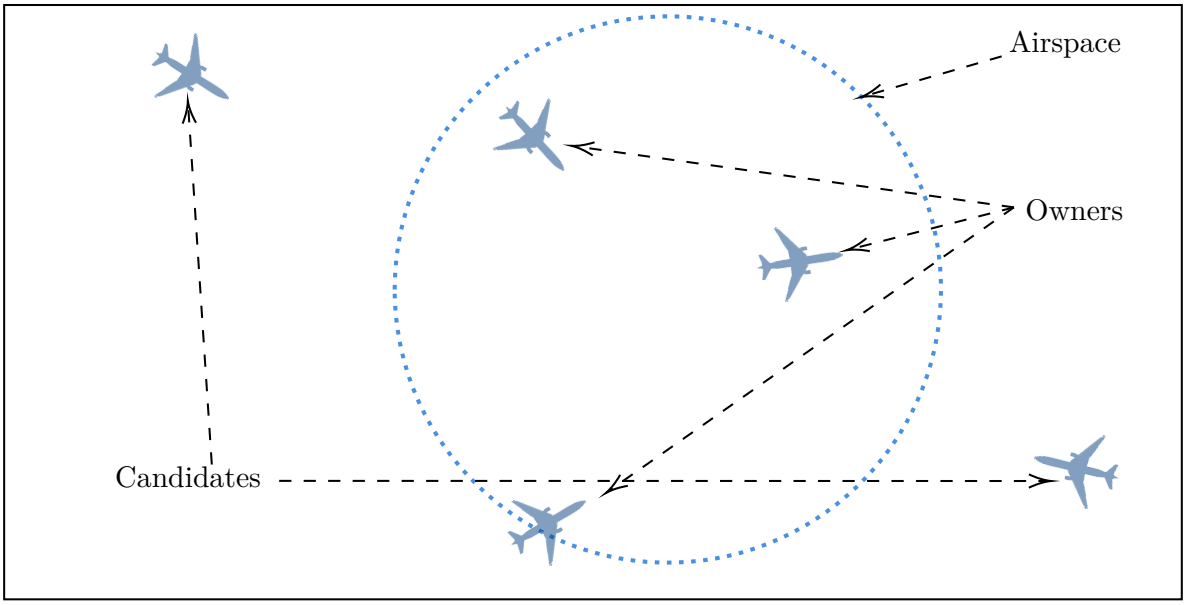

Figure 1: Aircraft trying to coordinate flight through a common airspace using DAC (top view).

sions [51]. These guarantees can be supported by pre-developed formal proofs that have been mechanically verified for correctness under some reasonable assumptions about the conditions.

In [50] we have presented an autonomous protocol for UTM called *Decentralized Admission Control* (DAC) in which aircraft connected through an asynchronous *vehicle-2-vehicle* (V2V) [47] network called the *Internet-of-Planes* (IoP) [51] can coordinate their use of a shared four-dimensional airspace. In DAC, an airspace has a set of *owner* aircraft which already have the authorization to carry out a fixed set of *flight plans* through the airspace (Fig. 1). The set of flight plans corresponding to the owners is *safe*, *i.e.*, there is no possibility of NMACs between any two flight plans in the set. One or more *candidate* aircraft can use the knowledge of this set to compute a *conflict-aware flight plan* [52] that is compatible with all the owners. A candidate can then propose its conflict-aware flight plan to the owners to request authorization to use the airspace. Since the candidates do not consider each other in their conflict-free proposals, the owners can only authorize a single candidate at a time. When a candidate is authorized, the set of owners changes to include this candidate, and all aircraft *relevant* to the airspace must be informed about the new set of owners so that they can learn the new value. Aircraft relevant to an airspace comprises of the set of new owners and the set of candidates expected to try to enter the airspace in the future (this includes the candidates who may have been previously denied authorization). Therefore, after a candidate is authorized, a *knowledge propagation protocol* is needed to propagate the knowledge of the new set of owners to all relevant aircraft.

In [53] we have presented a *Two-phase Acknowledge* knowledge propagation protocol (TAP) that can be used for *sufficiently propagating* the knowledge of the new set of owners ϕ of an airspace to attain a *safe state of knowledge* [18] for DAC. For TAP, we have formally proven *consistency*, which implies that TAP will correctly propagate the knowledge, and *eventual progress*, which implies that eventually, the safe state will be attained. Although a guarantee of eventual progress *alone* is insufficient for time-critical UAM applications, as it does not state any bound on the time for successful propagation, it is a necessary precondition for providing formal

guarantees of *timely progress*, which can provide some useful time bounds. The total *clock time* that is required for successful propagation is dependent on three factors—message transmission delays, message processing delays, and the number of messages involved. In asynchronous settings message delays are unbounded, making it impossible to provide deterministic bounds on the total time that may be required for successful propagation.

In the absence of deterministic time bounds, it is possible to provide probabilistic bounds for the timely progress of distributed protocols by modeling the factors affecting total time as stochastic processes. In this paper, we formulate *probabilistic timely progress guarantees* for TAP (assuming eventual progress) by using theories apposite to *low-altitude platforms* (LAP) [11] of airborne communication such as *vehicular ad hoc networks* (VANET) [23]. Particularly, we formalize the theory of the *Multicopy Two-Hop Relay* (MTR) protocol [39] to model message transmission delays in VANETs and the *M/M/1 queue system* [37] to model message processing delays. We then use the probabilistic bounds on these delays to provide sufficiently high-level guarantees of timely progress that are useful for UAM applications. *E.g.,* if a timely progress guarantee states that *"propagation will take at most 5 seconds with a probability 98%"*, then an aircraft can choose to only compute flight plans that start after 5 seconds to ensure that the plans don't become obsolete by the time that they are successfully propagated.

We have formalized our theory for timely progress in TAP using the *Athena proof assistant* [2, 4]. Athena provides a specification and proof language along with an interactive proof development environment. It is based on *many-sorted first-order logic* [42] and it uses a *natural deduction* [3] style of proofs. Athena is *sound*, *i.e.*, any method that successfully executes produces a theorem that is guaranteed to be a logical consequence of the *assumption base*. Athena also provides some helpful mechanisms to modularize and organize theory in a *reusable* manner. Using Athena, we have developed a formal theory library that is tailored towards reasoning about timely progress properties of distributed protocols deployed in airborne networks. This library can be used for reasoning about time-critical decentralized UAM applications.

Another important motivation behind developing a tailored library was to lay the foundation for a *parameterized proof library* that can be used in tandem with the *dynamic data-driven applications systems* (DDDAS) [14] paradigm for the runtime verification of autonomous distributed applications [51]. Autonomous UAM applications like DAC are expected to operate in unpredictable dynamic environments which cannot be completely modeled. Therefore, formal proofs that are developed in the pre-deployment stages may cease to be useful at runtime if the operational conditions assumed for the proofs do not correspond to the actual operational conditions. In the DDDAS-assisted approach, formal proofs can be parameterized by runtime-observable parameters about the operational conditions of distributed protocols and it is possible to augment the proofs in real-time using a *formal DDDAS feedback loop* [51]. A parameterized proof library will make it possible to provide formal guarantees that can be augmented with appropriate real-time parameters while retaining the formal rigor. This will allow the development of highly adaptive autonomous distributed UAM applications that will be capable of adapting to the formal guarantees that are actually valid at runtime.

The main contributions of this paper are:

- we formalize the theory of the Multicopy Two-Hop Relay (MTR) protocol to model message transmission delays between aircraft,
- we formalize the theory of the M/M/1 queue system to model message processing delays in each aircraft,

- we use the theory of MTR protocol and M/M/1 queue to provide machine-checkable probabilistic guarantees of timely progress for TAP, and
- we showcase the development of a formal proof library in Athena tailored towards timely progress guarantees of distributed protocols in VANETs.

The paper is structured as follows: Section 2 describes TAP; Section 3 presents our timely progress guarantee for TAP; Section 4 showcases the development of the proof library in Athena; Section 5 presents a discussion on our approach towards developing the formal guarantees; Section 6 presents prior work related to progress analysis and formalization of mathematical theories; and Section 7 concludes the paper with a discussion about future directions of work.

2 The Two-Phase Acknowledge Protocol

For decentralized UAM operations, aircraft will need to have *heightened knowledge* about an airspace to operate with a *sense of safety* [53]. Fagin *et al.* [18] succinctly explain this concern with the following example – *"..even if all drivers in a society know the rules for following traffic lights and follow them, that is not enough to make a driver "feel safe". This is because unless a driver knows that everyone else knows the rules and will follow them, then the driver may consider it possible that some other driver, who does not know the rules, may run a red light."*. The knowledge propagation protocol (TAP) for DAC ensures a state of knowledge in which every aircraft can *feel safe* to operate. In *knowledge logic* [18], the expression $k_i\phi$ represents that *an agent i knows the fact* ϕ, and $E\phi$ represents that *all agents in the system know* ϕ. The expression $EE\phi$ (or $E^2\phi$) represents a higher state of knowledge than $E\phi$ and implies that every agent knows two facts—ϕ and $E\phi$. In the context of DAC, ϕ represents the set of new owners after a candidate has been admitted and $E^2\phi$ represents the required safe state of knowledge which implies that all aircraft relevant to the airspace will know that all other aircraft know about the same new set of owners. Aircraft relevant to an airspace comprises of the set of new owners and any known candidate expected to try to enter the airspace in the future. This allows any future candidate to use the knowledge of ϕ to compute a correct conflict-aware plan.

2.1 The Protocol

TAP considers an asynchronous, non-Byzantine system model in which agents may fail temporarily and message delivery is reliable [20]. There is a non-empty set of *propagators*, and a logically separate non-empty set of *replicas*. All propagators know the set of replicas and the same value ϕ. $E^2\phi$, in the context of our protocol, implies that *all replicas know that all other replicas know* ϕ. The goal of every propagator is to propagate ϕ among all the replicas and eventually learn that $E^2\phi$ has been achieved. Propagators and replicas are logical abstractions and may be functionally implemented by the same physical node (*e.g.,* an aircraft) simultaneously. For DAC, the set of aircraft which know about the new set of owners acts as the set of propagators and the set of all aircraft relevant to the airspace acts as the set of replicas.

From the perspective of a propagator, TAP proceeds in two consecutive phases:

- **Phase 1**
  (a) The propagator sends a *learn* message with a value ϕ to each replica.

(b) A replica learns a value ϕ if and only if it receives a *learn* message with the value ϕ from the propagator and it replies to the propagator with a *learnt* message if and only if it has learned a value ϕ.

- **Phase 2**
  (a) The propagator sends an *all-know* message to each replica if and only if it has received *learnt* messages from all replicas.
  (b) A replica learns that all replicas know the value ϕ if and only if it receives an *all-know* message from the propagator and it replies to the propagator with an *acknowledgment* message if and only if it has learned that all replicas know the value ϕ.
  (c) The propagator learns $E^2\phi$ has been achieved if and only if it receives *acknowledgement* messages from all replicas.

2.2 Progress in TAP

We have formally proven in [53] that under our system model, TAP will make eventual progress under some suitable conditions. If eventual progress is guaranteed for TAP, then the total time taken for successful propagation is dependent on the following:

1. message transmission delays,
2. message processing delays, and
3. the total number of messages involved.

From the perspective of a propagator, successful propagation involves a deterministic number of messages. For each replica, 4 messages are required, so for R replicas, the total number of messages required for successful propagation is $4 \times R$. However, in asynchronous settings, transmission and processing delays are unbounded. Therefore, even with a deterministic number of messages, it is impossible to provide a deterministic bound on the time for progress.

3 Probabilistic Guarantees of Timely Progress

In the absence of deterministic solutions, it is possible to employ a probabilistic approach for providing formal guarantees of timely progress. This can be done by probabilistically reasoning about the message delays using either theoretical models or data-driven models. In contrast to theoretical models that are based on analytical results about system characteristics, data-driven results are usually based on real-time or historical statistical observations about the system. This paper focuses on using theoretical models for reasoning about timely progress.

For UAM applications, it is desirable to choose theoretical models that are appropriate for airborne networks like the IoP. Through the IoP, aircraft should be able to not only communicate their own information but also relay received information from other aircraft using a *multi-hop ad hoc* approach [56]. Moreover, an aircraft can be considered to be a single *server* where messages arrive, wait for some time until they are processed, and take some time to be processed. This makes it appropriate to use *queueing theory* [37] to reason about message processing delays. The M/M/1 system is an elementary queue system that consists of a single server that processes all messages which are received, making it a reasonable choice for modeling message processing with respect to a single aircraft. M/M/1 system also provides some useful analytical properties that make it possible to formally reason about the message processing delays. Therefore, to formally derive timely progress guarantees for TAP, we assume the following:

- all aircraft use the MTR protocol for message transmission between each other,
- each aircraft independently implements an M/M/1 queue to process the messages that it receives,
- the transmission delays of the messages are *independent and identically distributed* (i.i.d.),
- the processing delays of the messages are i.i.d., and
- the transmission delays are independent of the processing delays.

From the perspective of a propagator, if eventual progress is guaranteed, then in the worst-case, when there is no concurrency in message transmission and processing, the total time (T_S) taken for successful propagation can be calculated by using the total number of messages that are involved (N_M), and the transmission delay (T_{D_m}) and the processing delay (T_{P_m}) of each message m. Since we know that N_M has a deterministic value ($4 \times R$) for TAP, the total time for the worst-case scenario can, therefore, be obtained using Eq. 1.

$$T_S = \sum_{m=1}^{N_M} T_{D_m} + \sum_{m=1}^{N_M} T_{P_m} \tag{1}$$

Now, to derive a probabilistic bound on T_S, we first need to probabilistically model T_{D_m} and T_{P_m} for every message m involved in progress.

3.1 Modeling the Message Transmission Delays

Two-hop relaying, for data transmission between a *source* and a *destination* when the two nodes are not within *transmission range*, has been proposed to be an efficient mode of communication in *mobile ad hoc networks* (MANET) [22]. Relaying has also been considered to be appropriate for airborne networks like the IoP where two communicating aircraft may not always be within direct *transmission range* [56]. Therefore, to ensure that our assumptions about message transmission delays in airborne networks are consistent with prior work in the literature, we use the theory of the multicopy two-hop relay (MTR) protocol. For this, we base our assumptions on the description of the MTR protocol presented by Al Hanbali *et al.* [1].

In the basic two-hop relay protocol, there are a set of $M+1$ mobile nodes whose trajectories are i.i.d. [22]. If a *source* node wants to transfer a message to a *destination* node, it can either transmit it directly to the destination, if the destination is within its transmission range, or, it can transmit copies of the message to one or more *relay* nodes. A relay node can transmit a copy of a message directly to the destination node if the two are within the transmission range of the relay node. A relay node, however, cannot transmit a copy of a message to another relay node. Each message, therefore, makes a maximum of two hops—it is either transmitted directly from the source to the destination, or it is transmitted through one intermediate relay node.

In the MTR protocol [1], transmission delay T_{D_m} is the time taken for the destination to receive a message m or a copy of m for the first time. Two nodes *meet* when they come within the transmission range of one another and *inter-meeting time* is the time interval between two consecutive meetings of a given pair of nodes. The inter-meeting times of all pairs of nodes are i.i.d. with the common *cumulative distribution function* (CDF) $G(t)$. The source can only transmit a message to a relay that does not already hold a copy. Message transmission between two nodes within range is instantaneous. Each copy of a message has a *time-to-live* (TTL), which is the time after which a relay has to drop an untransmitted copy.

To derive a probabilistic bound on the time taken to transmit a message using MTR, we make the following assumptions:

- the TTLs for all messages are unrestricted, *i.e.*, a message can be held by a relay node until it can transmit it,
- since delivery is instantaneous when the nodes meet and TTLs are unrestricted, the inter-meeting time between the source and the destination (T_{sd}) or between a relay i and the destination (T_{r_id}) also represents the time taken by the source or the relay to directly deliver a message to the destination,

$$T_{D_m} = \min(\{T_{sd}, T_{r_1d}, ..., T_{r_{M-1}d}\}) \tag{2}$$

- the source node transmits the copy of a message to all $M-1$ relay nodes. A message (or a copy) can, therefore, be delivered to the destination by the source or any of the relay nodes. Hence, the actual time taken to deliver the message will be the minimum of $\{T_{sd}, T_{r_1d}, ..., T_{r_{M-1}d}\}$ (Eq. 2), and
- the inter-meeting times of the mobile nodes are *exponentially distributed* with the *rate parameter* λ_{MTR}.

By using the above assumptions and necessary theories from probability, random variables, and exponential distributions, we have formally derived the following probabilistic bound on the time taken to deliver a message m using MTR:

$$P\left(T_{D_m} \leq t\right) = 1 - \left(1 - \left(1 - e^{-\lambda_{MTR}t}\right)\right)^M \tag{3}$$

Now, the derivative of the above expression conforms to the *probability density function* (PDF) of an exponential distribution with rate parameter $\lambda_{MTR}M$. Therefore, we can conclude that T_{D_m} is exponentially distributed with a rate parameter $\lambda_{D_m} = \lambda_{MTR}M$.

3.2 Modeling the Message Processing Delays

Queueing theory [9] has been extensively used in the literature for modeling throughput in MANETs [32, 55, 57, 60]. An aircraft communicating using the IoP can be considered to be a single *server* where messages arrive, wait some time in a queue until they are processed, and take some time to be processed. Therefore, for modeling the message processing delays in the aircraft, we use theory of queues, particularly the *M/M/1* queue system [37]. We choose the M/M/1 queue system because it provides elegant analytical properties for computing the total time T_{P_m} required for a message m to be processed. This processing time includes the time spent in the queue and the time spent in actually processing the message. An important property of T_{P_m} in the M/M/1 queue system is that T_{P_m} is exponentially distributed [9].

The M/M/1 queue system consists of a single server where customers arrive according to a *Poisson process* [33]. It has the following characteristics [9]:

- the *interarrival times* of messages in the queue are exponentially distributed with a rate parameter λ_a,
- the *service time* of messages, which is the time spent in actually processing a message, is exponentially distributed with a mean $1/\mu_s$,
- the queue is managed by a single server, and

- the number of message arrivals in an interval of length τ follows a *Poisson distribution* [36] with a parameter $\lambda_a \tau$.

To probabilistically model T_{P_m}, we use *Little's theorem* [38], which is an important fundamental result in queueing theory. If N_q and T_p represent the *average number of messages* and the *mean processing delay* (queueing delay + service delay) respectively[1], then Little's Theorem states the useful relationship conveyed in Eq. 4.

$$N_q = \lambda_a T_p \tag{4}$$

A proof of Little's Theorem has been presented in [9], which we have formalized[2]. The proof uses $N(\tau)$, $\alpha(\tau)$, and $T(i)$ to represent the number of messages in the system at time τ, the number of messages that arrived in the interval $[0, \tau]$, and the average time spent in the system by message i respectively. The average arrival rate over the time period $[0, t]$ is then given by:

$$\lambda_t = \frac{\alpha(t)}{t} \tag{5}$$

Similarly, the average processing delay of messages that arrived in the interval $[0, t]$ is given by:

$$T_t = \sum_{i=1}^{\alpha(t)} \frac{T(i)}{\alpha(t)} \tag{6}$$

The average number of messages in the system in the interval $[0, t]$ is given by:

$$N_t = \frac{1}{t} \int_0^t N(\tau) d\tau \tag{7}$$

Now, the graphical analysis of a FIFO system reveals the following [9]:

$$\frac{1}{t} \int_0^t N(\tau) d\tau = \frac{1}{t} \sum_{i=1}^{\alpha(t)} T(i) \tag{8}$$

Using Eq. 5 to Eq. 8, it can be shown that:

$$N_t = \lambda_t T_t \tag{9}$$

Assuming that the limits $N_q = \lim_{t \to \infty} N_t$, $\lambda_a = \lim_{t \to \infty} \lambda_t$, and $T_p = \lim_{t \to \infty} T_t$ exist, we can now obtain Eq. 4 to prove Little's Theorem.

Now, using the properties of the Poisson distribution, it has been shown in [9] that:

$$N_q = \frac{\lambda_a}{\mu_s - \lambda_a} \tag{10}$$

Again, from Eq. 10 and Eq. 4, we can obtain the following relationship:

$$T_p = \frac{1}{\mu_s - \lambda_a} \tag{11}$$

Finally, since T_{P_m} is exponentially distributed, its rate parameter λ_{P_m} can be obtained as[3]:

$$\lambda_{P_m} = \frac{1}{\mathbf{E}[T_{P_m}]} = \frac{1}{T_p} = \mu_s - \lambda_a \tag{12}$$

[1] Note that $T_p = \mathbf{E}[T_{P_m}]$ is the *mean* or the *expected value* of T_{P_m}.

[2] We present the detailed proof described in [9] to clearly highlight the current deficiencies in our formalization.

[3] For an exponentially distributed random variable, the rate parameter is the reciprocal of the mean.

3.3 Probabilistic Bound on the Worst-Case Time

Using the rate parameters λ_{D_m} and λ_{P_m} for the exponentially distributed message transmission and processing delays, we can now provide a probabilistic bound on the worst-case time for progress T_S. For that, let

$$T_D = \sum_{m=1}^{N_M} T_{D_m} \qquad \& \qquad T_P = \sum_{m=1}^{N_M} T_{P_m} \tag{13}$$

From Eq. 1 and Eq. 13, for any two real numbers x and y, we can state:

$$((T_D \leq x) \wedge (T_P \leq y)) \implies (T_S \leq (x+y)) \tag{14}$$

Now, for two events, A and B, the following statement holds [44]:

$$(A \implies B) \implies (P(B) \geq P(A)) \tag{15}$$

Therefore, from Eq. 14 and Eq. 15, we get:

$$P(T_S \leq (x+y)) \geq P((T_D \leq x) \wedge (T_P \leq y)) \tag{16}$$

Since the processing delays are independent of the transmission delays, by the *product rule* of independent events [19] we have[4]:

$$P((T_D \leq x) \wedge (T_P \leq y)) = P(T_D \leq x) \times P(T_P \leq y) \tag{17}$$

From Eq. 16 and Eq. 17 we have:

$$P(T_S \leq (x+y)) \geq P(T_D \leq x) \times P(T_P \leq y) \tag{18}$$

Now, both T_D and T_P are sums of i.i.d. exponential random variables. Therefore, T_D and T_P follow two different *Erlang distributions* [36], each of which are parameterized by a *shape parameter* (which is N_M in both cases) and a rate parameter (λ_{D_m} for T_D and λ_{P_m} for T_P). The CDF of an Erlang distribution with shape parameter k and rate parameter λ is given by:

$$F_{\text{Er}}(t,k,\lambda) = 1 - \sum_{n=0}^{k-1} \frac{1}{n!} e^{-\lambda t} (\lambda t)^n \tag{19}$$

This can be used to compute the delay probabilities as follows:

$$P(T_D \leq x) = 1 - \sum_{n=0}^{N_M-1} \frac{1}{n!} e^{-\lambda_{D_m} x} (\lambda_{D_m} x)^n = F_{\text{Er}}(x, N_M, \lambda_{D_m}) \tag{20}$$

$$P(T_P \leq y) = 1 - \sum_{n=0}^{N_M-1} \frac{1}{n!} e^{-\lambda_{P_m} y} (\lambda_{P_m} y)^n = F_{\text{Er}}(y, N_M, \lambda_{P_m}) \tag{21}$$

From Eq. 18, Eq. 20, and Eq. 21, we can state the required bound on T_S:

$$P(T_S \leq (x+y)) \geq F_{\text{Er}}(x, N_M, \lambda_{D_m}) \times F_{\text{Er}}(y, N_M, \lambda_{P_m}) \tag{22}$$

[4]The product rule states that for independent events A and B, $P(A \wedge B) = P(A) \times P(B)$.

where $N_M = 4 \times R$, $\lambda_{D_m} = \lambda_{MTR} M$ and $\lambda_{P_m} = \mu_s - \lambda_a$.

We have mechanically verified all the equations described in this section in Athena by using theory from limits, distributions, real numbers, and other domains. However, currently, we have not formalized the theory required to reason about Poisson distributions and graphical analysis, and, therefore, have taken Eq. 8 and Eq. 10 as conjectures for our formal proofs. Both these equations are established results from queuing theory that have been borrowed from Bertsekas *et al*. [9] and we plan to prove them in future work.

4 A Library of Reusable Formal Theories in Athena

For verifying the correctness of distributed protocols that can be used in autonomous UTM applications, it is oftentimes necessary to prove complex high-level properties about the protocols. Formally proving the correctness of a high-level property using an interactive proof assistant like Athena requires access to a formalization of all the fundamental theories on which the high-level property is based. *E.g.,* for proving the correctness of Eq. 22, it is necessary to formalize theories from a wide range of domains such as probability, MTR, and queues. With the increasing complexity of the high-level properties, it becomes very arduous and time-consuming to formalize such fundamental theories from scratch, thereby making the task of verification intractable. For this reason, we are developing a formal library in Athena that is tailored for reasoning about high-level progress properties of distributed protocols in airborne networks[5].

Athena provides a language for "proof programming" which relies on a fundamental technique called *method call*. Method calls in Athena represent logical *inference rules* that can accept arguments of arbitrary types and are higher-order [48]. Successful evaluation of a method call results in a *theorem*, which is then added to the *assumption base*. The assumption base is a set of sentences that have either been proven to be correct by using inference rules or asserted to be correct. Athena provides the *soundness* guarantee that any theorem that is proven will be a direct consequence of sentences already in the assumption base.

Athena allows the use of *modules* that can be used to organize, partition, and encapsulate theories into multiple logically separate *namespaces*. Modules are dynamic in the sense that they can be extended at any time and additional content can be added to them using the `extend-module` command (Fig. 2). This feature of Athena allows theories, which are logically separate from one another, to be categorized and structured for brevity. It also makes it easy to generalize theories for use in different contexts by importing them as required.

```
# Module Prob being extended by module PrbIndRv
load "Athena_LibDDDAS/math/Prob/Prob.ath"
extend-module Prob {
  module PrbIndRv {
    # contents of module PrbIndRv
    ...
    }
```

Figure 2: Extending modules in Athena.

We have used our Athena library to formalize and verify all the high-level properties and theorems mentioned in Section 3. For this purpose, we have formalized theories necessary to

[5]Complete Athena code available at http://wcl.cs.rpi.edu/pilots/fvcafp

reason about distributed protocols in airborne networks by adding constructs that can be used to specify the properties of interest. *E.g.*, we have created a domain of distributed protocols `DistProt` that has properties which represent the characteristics of the operational environment such as `dpTotTim` (the total time for progress), `msGetTd` (the message transmission delays) and `msGetTp` (the message processing delays). The Athena code for Eq. 16, which uses these constructs to define the probabilistic relationship between the total processing time, the message transmission delays, and the message processing delays, is given in Fig. 3.

```
# The relationship between total time for progress and message delays
define THEOREM-P-TS>=P-TD&TP :=
(forall DP r1 r2 .
    ((Prob.probE  (Prob.consE Prob.<= (dpTotTim DP) (r1 + r2)))
      >=
      (Prob.probE
        (Prob.cons2E
          (Prob.consE Prob.<= (Random.SUM (msGetTd (dpGetMsgs DP))) r1)
          (Prob.consE Prob.<= (Random.SUM (msGetTp (dpGetMsgs DP))) r2)))))
```

Figure 3: The representation of Eq. 16 in Athena.

Similarly to `DistProt`, we have specified a domain of networks called `Network`, a domain of network protocols called `PrtType` that can represent protocols like MTR, a domain of queues called `Queue` to represent queuing models like M/M/1, and other specialized theories to reason about distributed protocols implemented over ad hoc airborne networks.

```
# Expected value of message processing delays in M/M/1 queue
conclude THEOREM-mean-T-value
...
let{
    ...
  N_d :=
    ( (Dist.ratePar (Random.pdf (srvcTm Q))) # mu_s
      -
      (Dist.ratePar (Random.pdf (cstArRat Q))) # lambda_a
    );
    ...
}
(!chain [ T
    = (N * (1.0 / L)) [T=N-x-inv-L]
    = ((L / N_d) * (1.0 / L)) [N=L-by-N_d]
    = (1.0 / N_d) [conn-2-a-by-d-x-b-by-a]
    ])
```

Figure 4: A snippet from the proof of Eq. 11 in Athena.

Athena supports the use of *tactics* for interactively guiding the proof development of higher-level proof obligations from lower-level facts in the assumption base. There are built-in methods in Athena that can be used to implement tactics for both *forward* and *backward* reasoning. *E.g.,* Fig. 4 shows the use of the built-in `chain` method for deriving the proof of Eq. 11 using forward reasoning, where `T` represents T_p and `N_d` represents $\mu_s - \lambda_a$. In the future, we aim to define specialized methods that can be used as tactics for reasoning about distributed protocols.

One of our goals while developing the library in Athena was reusability. As the proofs of different independent high-level properties are often dependent on common fundamental theories,

```
# The relationship between cdf and probability
define cdf-prob-conjecture :=
(forall x R . ((Random.cdf x R) = (probE (consE <= x R))))
```

Figure 5: The relationship between CDF and probability defined as a conjecture in Athena.

it becomes beneficial to design the common theories in a manner that makes them reusable in different contexts. *E.g.,* the statement `cdf-prob-conjecture` (Fig. 5), which defines that the CDF $G_X(x)$ of a random variable X is the probability that $X \leq x$ for all reals x, has been used in various contexts throughout our formalization of the results presented in Section 3 (Fig. 6).

```
# Proof of P(min[X1,X2,...] <= y) = 1 - (1- G(X))^N
conclude THEOREM-probability-MIN-<=-IID-RVS-Gx
...
  conn-to-cdf-prob := (!uspec (!uspec cdf-prob-conjecture (Random.rvSetIdElmnt rvSet)) R)
...


# Probabilistic bound on customer delay for M/M/1 queue
conclude THEOREM-cstDly-prob
...
  conn-2-cdf-prob-conjecture := (!uspec (!uspec Prob.cdf-prob-conjecture (cstDly Q)) r)
...
```

Figure 6: Some instances where the `cdf-prob-conjecture` has been reused.

At present, our extension to the Athena library consists of 71 axioms and 27 theorems, including the high-level properties of distributed protocols and other domains described in this paper. It is composed of theories from three main domains—distributed systems, mathematics, and network theory. Within each domain, there are theories from various related sub-domains. *E.g.,* under mathematics, we have theories from probability, queues, random variables, functions, and distributions. Fig. 7 depicts the current logical hierarchy of our extension to the Athena library and the dependencies between the various modules representing the different logical domains and sub-domains. As our primary goal was to develop high-level probabilistic progress properties of TAP that can be directly useful for UAM purposes, we adopted a *top-down* approach of proof development where we only developed the lower-level theories that were required to support the higher-level properties of interest to us. In most of the cases, we have tried to use only fundamental facts from the various domains as axioms, but there are instances in our formulation where we have currently taken some well-established results from the domains as conjectures that we aim to prove in future work (*e.g.,* Eq. 10, `cdf-prob-conjecture`).

5 Discussion

The availability of probabilistic guarantees of timely progress for autonomous distributed coordination protocols will allow the participating agents to make educated operational decisions. *E.g.*, if candidates for DAC know that it will take at most x seconds to propagate the knowledge of their accepted plans with 99.99% probability, then they can safely decide to compute plans that start after x seconds, to ensure that the plans do not become stale. Since the guarantees

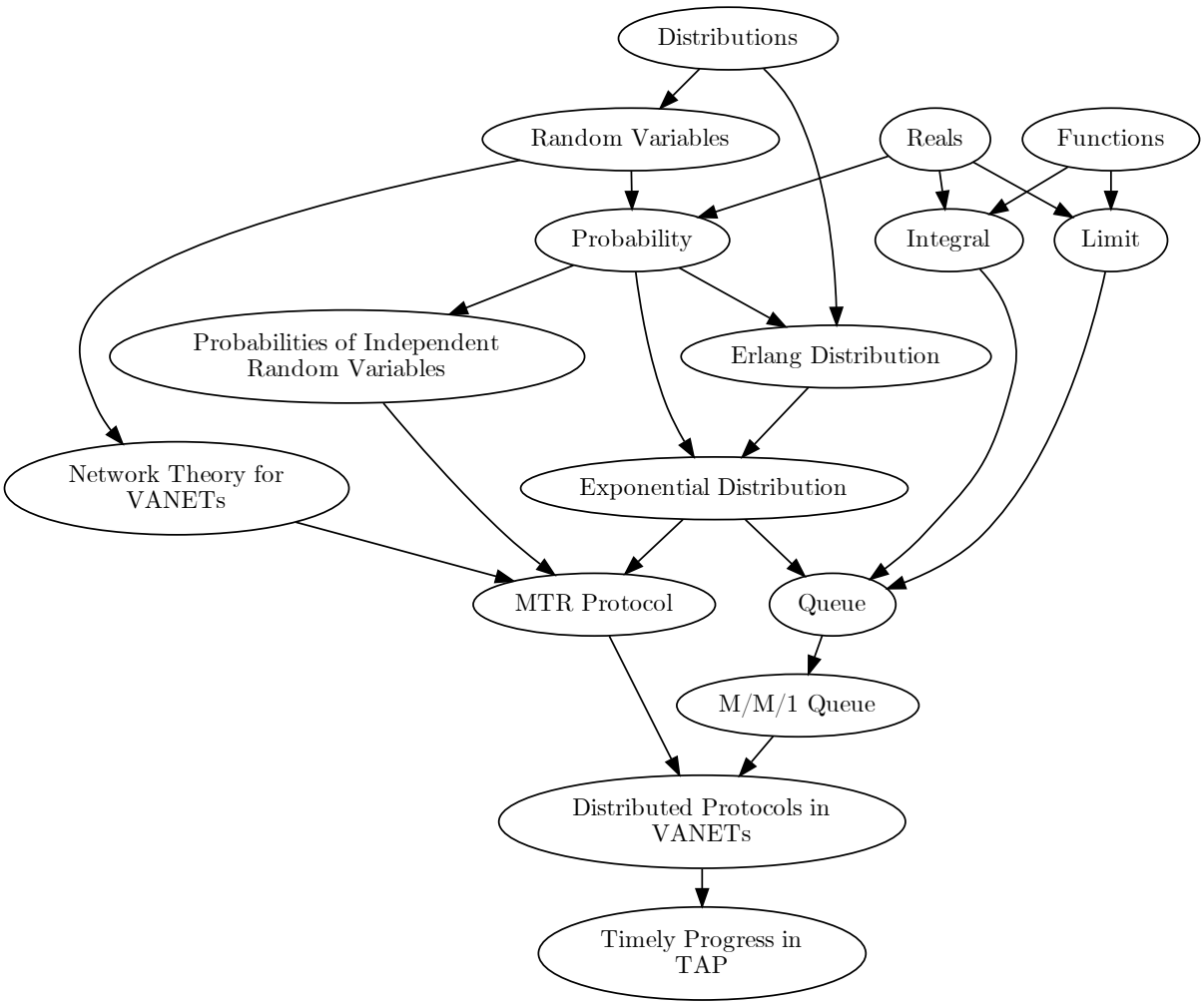

Figure 7: Current hierarchy of our extension to Athena's library for reasoning about distributed protocols in VANETs (does not depict Athena's built-in library).

we have presented in Section 3 are based on well-established theoretical models of VANETs, it makes them suitable for safety-critical airborne applications like DAC, where aircraft will need to coordinate over ad hoc networks in a decentralized manner. Two-hop relaying has been proposed to be an efficient mode of communication in VANETs [22] and since each aircraft will be independently responsible for processing all the messages sent to it, we believe that the proposed combination of the MTR protocol and the M/M/1 queue system is a reasonable approach for formally modeling communication between aircraft over the IoP.

Autonomous decentralized operations of mobile agents are complex in nature. Such operations involve many interlinked aspects such as the mobility of the agents, the characteristics of the communication network, the internal computational capabilities of the agents, and the interaction between the agents. Therefore, to formally reason about such operations, it is necessary to holistically consider all these important aspects. Achieving this in a machine-verifiable manner requires access to formal constructs, that are sufficiently expressive to correctly and

completely specify such aspects, in some machine-readable formal language. Developing such expressive formal constructs in a machine-readable language is a challenging task since it requires domain knowledge of all aspects of the system that need to be specified, strong knowledge of formal logic and reasoning techniques, experience and familiarity with the language in which the constructs are to be specified, and a significant amount of time and effort.

In the absence of a pre-existing formal library that provides the necessary formal constructs to express some system, every time high-level properties of the system need to be verified, engineers have to develop the required formalizations from the ground up, often making the task of verification intractable. *E.g.*, when we set out to specify timely progress properties for TAP using the theory of MTR protocol and M/M/1 queue system, we required theories in Athena to express constructs such as networks, mobility models, communication models, inter-agent interactions, queues, and distributed protocols, which were necessary to express the concepts presented in Section 3. For that reason, we developed most of the symbols, domains, and relationships necessary to comprehensibly express all the assumptions and specifications using the capabilities of Athena's many-sorted first-order logic. We have tried to make the specifications reusable to different contexts so that further expansion of the library can be aided by using the existing specifications as building blocks rather than requiring to redevelop them for use in other contexts. In our experience, efficiently designing reusable formal structures to express interconnected theories requires several iterations where the specifications need to be improved to be more general as new contexts for application are identified. Another challenge is the meaningful modularization of the developed theories into appropriate categories to make it easy to import them independently in different contexts. This is important because when theories are imported during proof development, they are automatically added to the assumption base. Well-defined modules allow theories to be imported only as needed without making the assumption base unnecessarily large. This is an evolving project and we are certain that as more theory is added to the library, many of the formalizations in our current library will need to be redesigned in order to make them more general and better-organized.

During the development of our high-level formal guarantees of timely progress for TAP in Athena, we have relied on some theories which we have not formally verified in Athena, but have used them as conjectures for the proofs. Nevertheless, we believe that this does not invalidate the results we have presented in the paper or diminish their importance in any way. This is because all the conjectures we have used are well-established facts in the literature of the respective domains and have been informally proven beyond a reasonable doubt. Since we were more interested in verifying the novel higher-level properties of timely progress for TAP that we have presented in Section 3, in the interest of time, we decided to take some of these well-established theories as foundations for developing our higher-level guarantees since our library currently lacks the necessary formal theory to mechanically verify them. However, if time is not a concern, it should be possible to develop the proofs of these theories in Athena from just the fundamental axioms of the corresponding domains.

To the best of our knowledge, at the time of publication, our extension to the Athena library is the only library that provides a tailored formal foundation for reasoning about autonomous distributed coordination in airborne networks based on the theory of VANETs and queuing systems. Therefore, we believe that our contribution will make the task of reasoning about properties of other distributed coordination protocols, which are implemented using MTR and the M/M/1 queue, in Athena easier in the future as a significant amount of reusable formal constructs to express such properties have already been developed. Nonetheless, additional

theories will be required to reason about distributed protocols involving a non-deterministic number of messages (*e.g.*, the *Synod consensus protocol* [50]). Our eventual goal is to expand the library by adding additional theories from distributions, queues, airborne networks, message sequences, and other necessary domains to support reasoning about a variety of distributed protocols that can be employed for autonomous decentralized UAM applications.

6 Related Work

There is existing work in the literature on the informal analysis of timely progress of consensus. Attiya *et al*. [6] provided lower-bounds on the time for progress in *round-based* [16] consensus protocols. Attiya *et al*. [5] analyzed the time complexity of solving distributed decision problems. Berman *et al*. [8] studied the number of message rounds involved in various consensus protocols.

Existing work on machine-verified guarantees has mainly analyzed eventual progress. McMillan *et al.* [43] have proven eventual progress of *Stoppable Paxos* [41] in Ivy [49]. Dragoi *et al*. [17] and Debrat *et al*. [15] have proven eventual progress in *LastVoting* [13]. Hawblitzel *et al.* [30,31] have proven eventual progress for a *Multi-Paxos* implementation using Dafny [35]. In [50], we have verified eventual progress in the Synod consensus protocol using Athena.

Formalization of mathematical theories has also been presented in the literature. Hasan [24] presented formalizations of statistical properties of discrete random variables in the *HOL Theorem Prover* [21]. Hasan *et al.* have formalized properties of the standard uniform random variable [25], discrete and continuous random variables [26, 28], tail distribution bounds [27], and conditional probability [29]. Mhamdi *et al.* [45, 46] have extended Hasan *et al*.'s work by formalizing *measure theory* and the *Lebesque integral* in HOL. HOL formalizations of the Poisson process, *continuous chain Markov process*, and M/M/1 queue have been presented in [12]. Qasim [54] has used Mhamdi *et al*.'s work to formalize the *standard normal variable*.

In contrast to the existing work, we have developed formal theories tailored towards machine-verifiable timely progress guarantees for distributed protocols using models appropriate for airborne networks and have presented the timely progress guarantee of a knowledge propagation protocol that can be used for autonomous decentralized UAM applications.

7 Conclusion and Future Work

In this paper, we have presented a formal probabilistic guarantee of timely progress for the Two-phase Acknowledge knowledge propagation protocol by using theories from the Multi-copy Two-Hop Relay protocol and the M/M/1 queue system to reason about the non-deterministic message delays. Since the progress guarantee provides useful probabilistic bounds on the time that may be required to make progress, it is more useful than a guarantee of eventual progress for time-critical UAM applications of the knowledge propagation protocol. We have also showcased the development of a formal library in Athena for mechanically verifying the timely progress guarantee. In the future, autonomous UAM applications will require participating aircraft to employ different types of distributed protocols for achieving different operational goals such as maintaining safe separation, autonomously coordinating operations through a shared airspace, etc. Our proof library has, therefore, been tailored to be reusable for reasoning about the timely progress of a variety of distributed protocols that can be used for such autonomous applications.

Currently, our formalizations have some high-level conjectures since our current library lacks the theories required to formally verify them. In the future, we plan to formalize the necessary fundamental theories required to prove these conjectures as theorems. The library also does not support the complete formalization of timely progress if the number of messages is non-deterministic. Therefore, another potential direction of future work is to add additional theories to the library to allow reasoning about distributed protocols that involve a non-deterministic number of messages, such as the Synod consensus protocol. In the future, we would also like to model other routing protocols suitable for VANETs in addition to MTR so that appropriate guarantees can be provided depending on actual implementations.

Acknowledgment: This research was partially supported by the National Science Foundation (NSF), Grant No. – CNS-1816307 and the Air Force Office of Scientific Research (AFOSR), DDDAS Grant No. – FA9550-19-1-0054.

References

[1] Ahmad Al Hanbali, Arzad A Kherani & Philippe Nain (2007): *Simple Models for the Performance Evaluation of a Class of Two-Hop Relay Protocols*. In: *International Conference on Research in Networking*, Springer, pp. 191–202, doi:10.1007/978-3-540-72606-7_17.

[2] Konstantine Arkoudas: *Athena*. Available at `http://proofcentral.org/athena`.

[3] Konstantine Arkoudas (2005): *Simplifying Proofs in Fitch-Style Natural Deduction Systems*. *Journal of Automated Reasoning* 34(3), pp. 239–294, doi:10.1007/s10817-005-9000-3.

[4] Konstantine Arkoudas & David Musser (2017): *Fundamental Proof Methods in Computer Science: A Computer-Based Approach*. MIT Press, doi:10.1017/s1471068420000071.

[5] Hagit Attiya & Taly Djerassi-Shintel (2001): *Time Bounds for Decision Problems in the Presence of Timing Uncertainty and Failures*. *Journal of Parallel and Distributed Computing* 61(8), pp. 1096–1109, doi:10.1006/jpdc.2001.1730.

[6] Hagit Attiya, Cynthia Dwork, Nancy Lynch & Larry Stockmeyer (1994): *Bounds on the Time to Reach Agreement in the Presence of Timing Uncertainty*. *Journal of the ACM (JACM)* 41(1), pp. 122–152, doi:10.21236/ada229766.

[7] Swee Balachandran, Christopher Manderino, César Muñoz & María Consiglio (2020): *A Decentralized Framework to Support UAS Merging and Spacing Operations in Urban Canyons*. In: *2020 International Conference on Unmanned Aircraft Systems (ICUAS)*, IEEE, pp. 204–210, doi:10.1109/icuas48674.2020.9213973.

[8] Piotr Berman, Juan A Garay, Kenneth J Perry et al. (1989): *Towards Optimal Distributed Consensus*. In: *FOCS*, 89, pp. 410–415, doi:10.1109/sfcs.1989.63511.

[9] Dimitri P Bertsekas, Robert G Gallager & Pierre Humblet (1992): *Data Networks*. 2, Prentice-Hall International New Jersey.

[10] Marc Brittain & Peng Wei (2018): *Autonomous Aircraft Sequencing and Separation with Hierarchical Deep Reinforcement Learning*. In: *Proceedings of the International Conference for Research in Air Transportation*.

[11] Xianbin Cao, Peng Yang, Mohamed Alzenad, Xing Xi, Dapeng Wu & Halim Yanikomeroglu (2018): *Airborne Communication Networks: A Survey*. *IEEE Journal on Selected Areas in Communications* 36(9), pp. 1907–1926, doi:10.1109/jsac.2018.2864423.

[12] Donia Chaouch (2015): *Formalization of Continuous Time Markov Chains with Applications in Queueing Theory*. Master's thesis, Concordia University.

[13] Bernadette Charron-Bost & André Schiper (2009): *The Heard-Of Model: Computing in Distributed Systems With Benign Faults*. *Distributed Computing* 22(1), pp. 49–71, doi:10.1007/s00446-009-0084-6.

[14] Frederica Darema (2004): *Dynamic Data-Driven Application Systems: A New Paradigm for Application Simulations and Measurements*. In: *Computational Science-ICCS 2004*, Springer, pp. 662–669, doi:10.1007/978-3-540-24688-6_86.

[15] Henri Debrat & Stephan Merz (2012): *Verifying Fault-Tolerant Distributed Algorithms in the Heard-Of Model*. *Archive of Formal Proofs* 2012.

[16] Carole Delporte-Gallet, Hugues Fauconnier, Rachid Guerraoui & Bastian Pochon (2007): *The Perfectly Synchronized Round-Based Model of Distributed Computing*. *Information and Computation* 205(5), pp. 783–815, doi:10.1016/j.ic.2006.11.003.

[17] Cezara Drăgoi, Thomas A Henzinger & Damien Zufferey (2016): *PSync: A Partially Synchronous Language for Fault-Tolerant Distributed Algorithms*. In: *ACM SIGPLAN Notices*, 51, ACM, pp. 400–415, doi:10.1145/2837614.2837650.

[18] Ronald Fagin, Joseph Y Halpern, Yoram Moses & Moshe Vardi (2004): *Reasoning About Knowledge*. MIT press, doi:10.7551/mitpress/5803.001.0001.

[19] Robert G Gallager (2013): *Stochastic Processes: Theory for Applications*. Cambridge University Press, doi:10.1017/cbo9781139626514.

[20] László Gönczy, Máté Kovács & Dániel Varró (2007): *Modeling and Verification of Reliable Messaging by Graph Transformation Systems*. *Electronic Notes in Theoretical Computer Science* 175(4), pp. 37–50, doi:10.1016/j.entcs.2007.04.015.

[21] Michael JC Gordon (1989): *Mechanizing Programming Logics in Higher Order Logic*. In: *Current trends in hardware verification and automated theorem proving*, Springer, pp. 387–439, doi:10.1007/978-1-4612-3658-0_10.

[22] Matthias Grossglauser & David NC Tse (2002): *Mobility Increases the Capacity of Ad Hoc Wireless Networks*. *IEEE/ACM transactions on networking* 10(4), pp. 477–486, doi:10.1109/tnet.2002.801403.

[23] Mustafa Maad Hamdi, Lukman Audah, Sami Abduljabbar Rashid, Alaa Hamid Mohammed, Sameer Alani & Ahmed Shamil Mustafa (2020): *A Review of Applications, Characteristics and Challenges in Vehicular Ad-Hoc Networks (VANETs)*. In: *2020 International Congress on Human-Computer Interaction, Optimization and Robotic Applications (HORA)*, IEEE, pp. 1–7, doi:10.1109/hora49412.2020.9152928.

[24] Osman Hasan (2008): *Probabilistic Analysis Using Theorem Proving*. In: *21 st International Conference on Theorem Proving in Higher Order Logics*, Citeseer, p. 21.

[25] Osman Hasan & Sofiène Tahar (2007): *Formalization of the Standard Uniform random variable*. *Theoretical Computer Science* 382, pp. 71–83, doi:10.1016/j.tcs.2007.05.009.

[26] Osman Hasan & Sofiène Tahar (2008): *Using Theorem Proving to Verify Expectation and Variance for Discrete Random Variables*. *Journal of Automated Reasoning* 41(3-4), pp. 295–323, doi:10.1007/s10817-008-9113-6.

[27] Osman Hasan & Sofiene Tahar (2009): *Formal Verification of Tail Distribution Bounds in the HOL Theorem Prover*. *Mathematical Methods in the Applied Sciences* 32(4), pp. 480–504, doi:10.1002/mma.1055.

[28] Osman Hasan & Sofiene Tahar (2010): *Formal Probabilistic Analysis: A Higher-Order Logic Based Approach*. In: *International Conference on Abstract State Machines, Alloy, B and Z*, Springer, pp. 2–19, doi:10.1007/978-3-642-11811-1_2.

[29] Osman Hasan & Sofiène Tahar (2011): *Reasoning About Conditional Probabilities in a Higher-Order-Logic Theorem Prover*. *Journal of Applied Logic* 9(1), pp. 23–40, doi:10.1016/j.jal.2011.01.001.

[30] Chris Hawblitzel, Jon Howell, Manos Kapritsos, Jacob R Lorch, Bryan Parno, Michael L Roberts, Srinath Setty & Brian Zill (2015): *IronFleet: Proving Practical Distributed Systems Correct*. In: *Proceedings of the 25th Symposium on Operating Systems Principles*, ACM, pp. 1–17, doi:10.1145/2815400.2815428.

[31] Chris Hawblitzel, Jon Howell, Manos Kapritsos, Jacob R Lorch, Bryan Parno, Michael L Roberts, Srinath Setty & Brian Zill (2017): *IronFleet: Proving Safety and Liveness of Practical Distributed Systems*. *Communications of the ACM* 60(7), pp. 83–92, doi:10.1145/3068608.

[32] Rashmi Kushwah, Shashikala Tapaswi & Ajay Kumar (2020): *Multipath Delay Analysis Using Queuing Theory for Gateway Selection in Hybrid MANET*. *Wireless Personal Communications* 111(1), pp. 9–32, doi:10.1007/s11277-019-06842-9.

[33] Günter Last & Mathew Penrose (2017): *Lectures on the Poisson Process*. 7, Cambridge University Press, doi:10.1017/9781316104477.007.

[34] Seung Man Lee, Chunki Park, Marcus A Johnson & Eric R Mueller (2013): *Investigating Effects of Well Clear Definitions on UAS Sense-And-Avoid Operations in Enroute and Transition Airspace*. In: *2013 Aviation Technology, Integration, and Operations Conference*, doi:10.2514/6.2013-4308.

[35] K. Rustan M. Leino (2010): *Dafny: An Automatic Program Verifier for Functional Correctness*. In: *International Conference on Logic for Programming Artificial Intelligence and Reasoning*, Springer, pp. 348–370, doi:10.1007/978-3-642-17511-4_20.

[36] Alberto Leon-Garcia (1994): *Probability and Random Processes for Electrical Engineering*, 2nd edition. Addison-Wesley, Reading, MA.

[37] Lester Lipsky (2009): *M/M/1 Queue*. *Queueing Theory: A Linear Algebraic Approach*, pp. 33–75, doi:10.1007/978-0-387-49706-8_2.

[38] John DC Little (1961): *A Proof for the Queuing Formula: L= λ W*. *Operations research* 9(3), pp. 383–387, doi:10.1287/opre.9.3.383.

[39] Jiajia Liu, Xiaohong Jiang, Hiroki Nishiyama & Nei Kato (2013): *On the Delivery Probability of Two-Hop Relay MANETs with Erasure Coding*. *IEEE Transactions on Communications* 61(4), pp. 1314–1326, doi:10.1109/tcomm.2013.020413.120198.

[40] Robert Luckner, Gordon Höhne & Michael Fuhrmann (2004): *Hazard Criteria for Wake Vortex Encounters During Approach*. *Aerospace Science and Technology* 8(8), pp. 673–687, doi:10.1016/j.ast.2004.06.008.

[41] Dahlia Malkhi, Leslie Lamport & Lidong Zhou (2008): *Stoppable Paxos*. Technical Report, Microsoft Research.

[42] Maria Manzano (1996): *Extensions of First-Order Logic*. Cambridge University Press.

[43] Kenneth L McMillan & Oded Padon (2018): *Deductive Verification in Decidable Fragments with Ivy*. In: *Static Analysis*, Springer, pp. 43–55, doi:10.1007/978-3-319-99725-4_4.

[44] Ronald Meester & Klaas Slooten (2021): *Some Philosophy of Probability, Statistics, and Forensic Science*, p. 1–29. Cambridge University Press, doi:10.1017/9781108596176.002.

[45] Tarek Mhamdi, Osman Hasan & Sofiène Tahar (2010): *On the Formalization of the Lebesgue Integration Theory in HOL*. In: *International Conference on Interactive Theorem Proving*, Springer, pp. 387–402, doi:10.1007/978-3-642-14052-5_27.

[46] Tarek Mhamdi, Osman Hasan & Sofiène Tahar (2013): *Formalization of Measure Theory and Lebesgue Integration for Probabilistic Analysis in HOL*. *ACM Transactions on Embedded Computing Systems* 12(1), doi:10.1145/2406336.2406349.

[47] Andreas F Molisch, Fredrik Tufvesson, Johan Karedal & Christoph F Mecklenbrauker (2009): *A Survey on Vehicle-to-Vehicle Propagation Channels*. *IEEE Wireless Communications* 16(6), pp. 12–22, doi:10.1109/mwc.2009.5361174.

[48] David R. Musser & Carlos A. Varela (2013): *Structured Reasoning About Actor Systems*. In: *Proceedings of the 2013 Workshop on Programming Based on Actors, Agents, and Decentralized Control*, AGERE!, ACM, New York, NY, USA, pp. 37–48, doi:10.1145/2541329.2541334.

[49] Oded Padon, Kenneth L McMillan, Aurojit Panda, Mooly Sagiv & Sharon Shoham (2016): *Ivy: Safety Verification by Interactive Generalization*. *ACM SIGPLAN Notices* 51(6), pp. 614–630, doi:10.1145/2980983.2908118.

[50] Saswata Paul, Gul A. Agha, Stacy Patterson & Carlos A. Varela (2021): *Verification of Eventual Consensus in Synod using a Failure-Aware Actor Model*. In: *Proceedings of the 13th NASA Formal Methods Symposium (NFM 2021)*, pp. 249–267, doi:10.1007/978-3-030-76384-8_16.

[51] Saswata Paul, Fotis Kopsaftopoulos, Stacy Patterson & Carlos A. Varela (2020): *Dynamic Data-Driven Formal Progress Envelopes for Distributed Algorithms*. In: *Dynamic Data-Driven Application Systems*, pp. 245–252, doi:10.1007/978-3-030-61725-7_29.

[52] Saswata Paul, Stacy Patterson & Carlos A. Varela (2019): *Conflict-Aware Flight Planning for Avoiding Near Mid-Air Collisions*. In: *The 38th AIAA/IEEE Digital Avionics Systems Conference*, San Diego, CA, pp. 1–10, doi:10.1109/dasc43569.2019.9081658.

[53] Saswata Paul, Stacy Patterson & Carlos A. Varela (2020): *Collaborative Situational Awareness for Conflict-Aware Flight Planning*. In: *The 39th IEEE/AIAA Digital Avionics Systems Conference*, pp. 1–10, doi:10.1109/dasc50938.2020.9256620.

[54] Muhammad Qasim (2016): *Formalization of Normal Random Variables*. Master's thesis, Concordia University, doi:10.1016/j.entcs.2013.09.001.

[55] Xinbing Wang, Qiuyu Peng & Yingzhe Li (2012): *Cooperation Achieves Optimal Multicast Capacity-Delay Scaling in MANET*. *IEEE Transactions on Communications* 60(10), pp. 3023–3031, doi:10.1109/tcomm.2012.081512.110535.

[56] Yang Wang & Yiyuan J Zhao (2006): *Fundamental Issues in Systematic Design of Airborne Networks for Aviation*. In: *2006 IEEE Aerospace Conference*, IEEE, pp. 8–pp, doi:10.1109/aero.2006.1655882.

[57] Xia Wen-jie, Yan Han & Liu Feng-yu (2011): *The Analysis of M/M/1 Queue Model with N Policy for Damaged Nodes in MANET*. In: *2011 IEEE International Conference on Computer Science and Automation Engineering*, 1, IEEE, pp. 289–294, doi:10.1109/csae.2011.5953224.

[58] David Wing & William Cotton (2011): *For Spacious Skies: Self-Separation with "Autonomous Flight Rules" in US Domestic Airspace*. doi:10.2514/6.2011-6865.

[59] Jim Woodcock, Peter Gorm Larsen, Juan Bicarregui & John Fitzgerald (2009): *Formal Methods: Practice and Experience*. *ACM Computing Surveys* 41(4), pp. 1–36, doi:10.5555/3065491.3065655.

[60] Shouyi Yin & Xiaokang Lin (2004): *MALB: MANET Adaptive Load Balancing*. In: *IEEE 60th Vehicular Technology Conference, 2004*, 4, IEEE, pp. 2843–2847, doi:10.1109/vetecf.2004.1400578.