Formal Guarantees of Timely Progress for Distributed Knowledge Propagation

Saswata Paul
Stacy Patterson
Carlos Varela

DEPARTMENT OF COMPUTER SCIENCE
RENSSELAER POLYTECHNIC INSTITUTE

The 3rd Workshop on
Formal Methods for Autonomous Systems
Oct 21-22, 2021

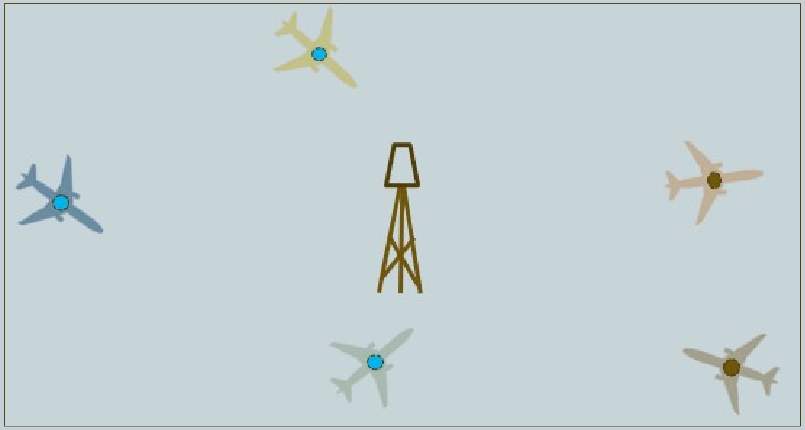
Introduction

- **Distributed knowledge**
  - **The knowledge possessed by a set of distributed agents**
  - **Can have various states**
    - *Both Alice and Omar know that it is raining*
    - *Alice knows that it is raining, but Omar does not*

- **Knowledge propagation**
  - Propagating a fact ø through a network of distributed agents to attain a desired ***state of knowledge***

Introduction

- **Motivation - *Decentralized Air Traffic Management***
  - ***Uncrewed Aircraft Systems (UAS)*** will navigate **highly congested** airspaces in ***Urban Air Mobility (UAM)*** scenarios
    - ***Centralized infrastructure*** such as dedicated ground stations for UAM can become single points of failure
  - Need to ***autonomously*** maintain ***safe separation*** for avoiding collisions
    - UAS must ***coordinate in a decentralized*** manner



Centralized Coordination

Decentralized Coordination

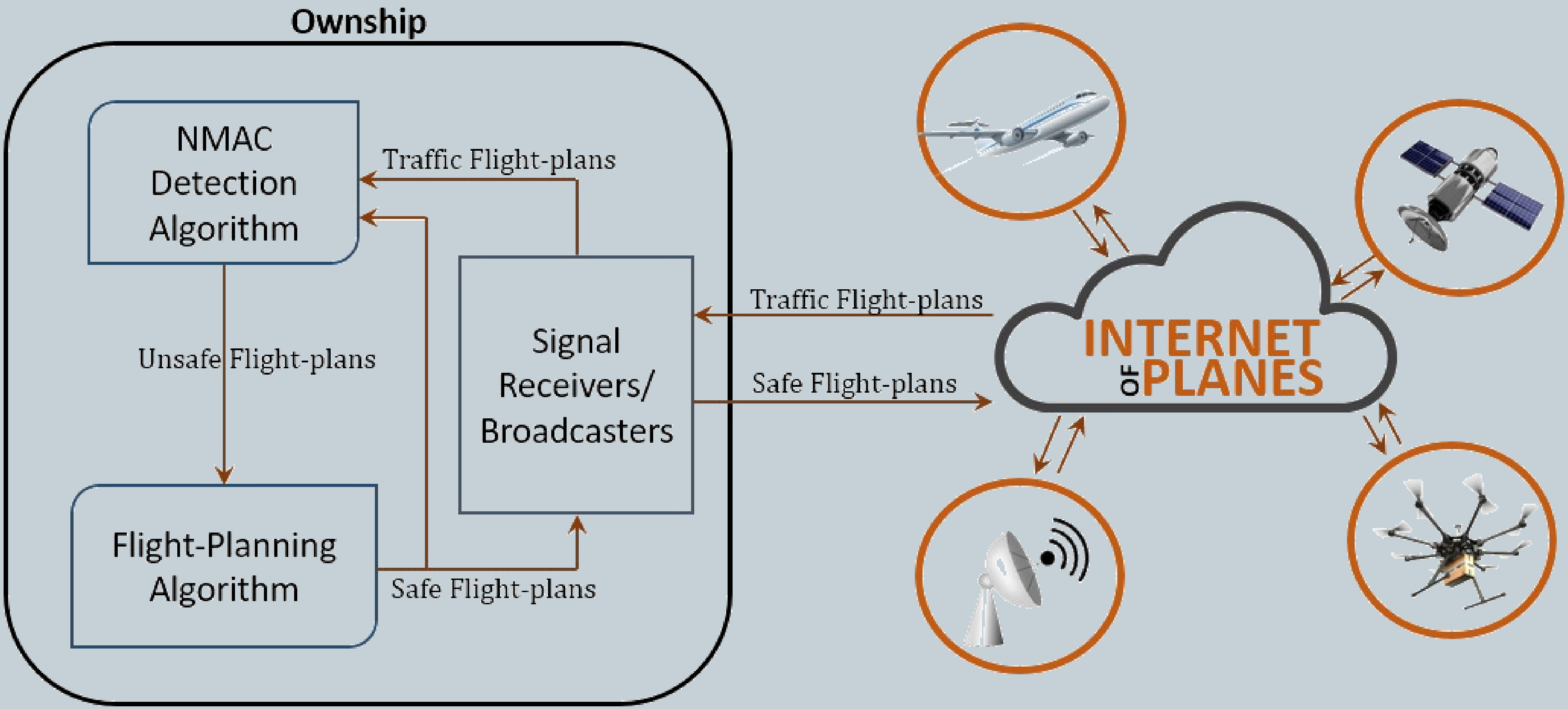
Introduction

- Advantages of ***Decentralized Air Traffic Management***:
  - Allows completely autonomous operation
    - Free from human errors
    - Can be **formally verified** for correctness
  - More fault tolerant than centralized approaches

- UAS must use ***distributed coordination protocols***
  - ***Strategic coordination***
  - *E.g.*, deciding the order for passing through a shared intersection

Introduction

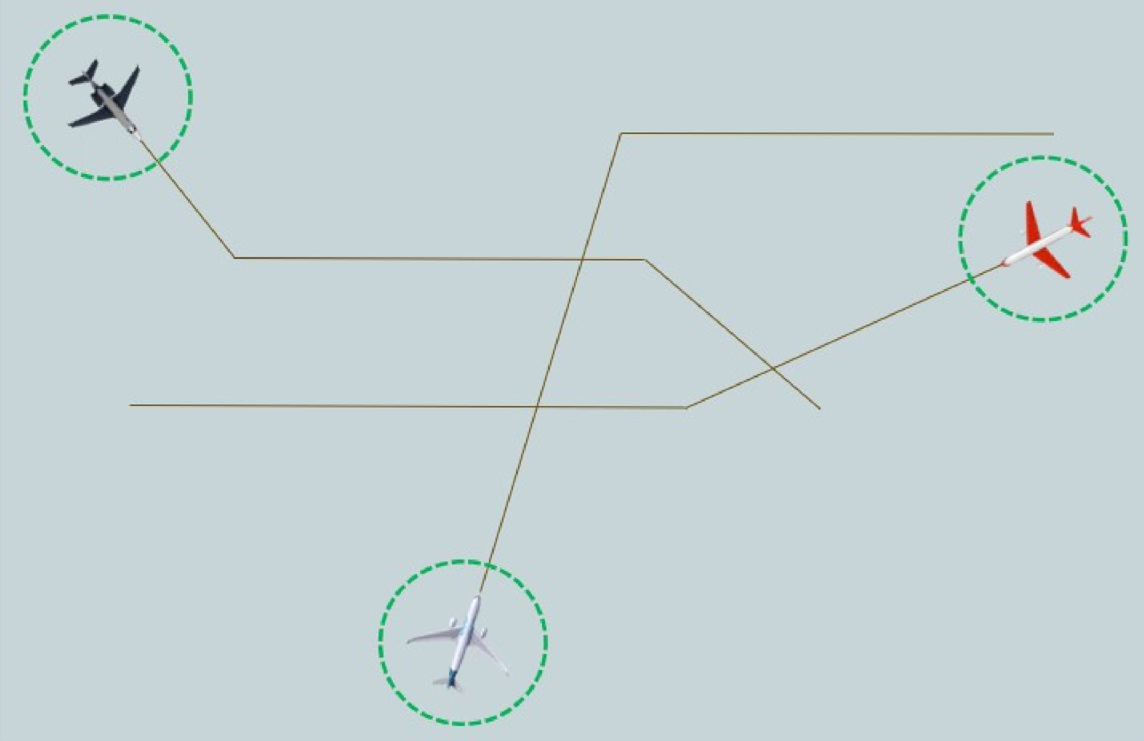
- ***Internet of Planes (IoP)*** - an asynchronous network over which air traffic data can be directly shared among aircraft *[Paul et al., DDDAS 2020]*



Introduction

Conflict-Aware Flight Planning: *[Paul et al., DASC 2019]*

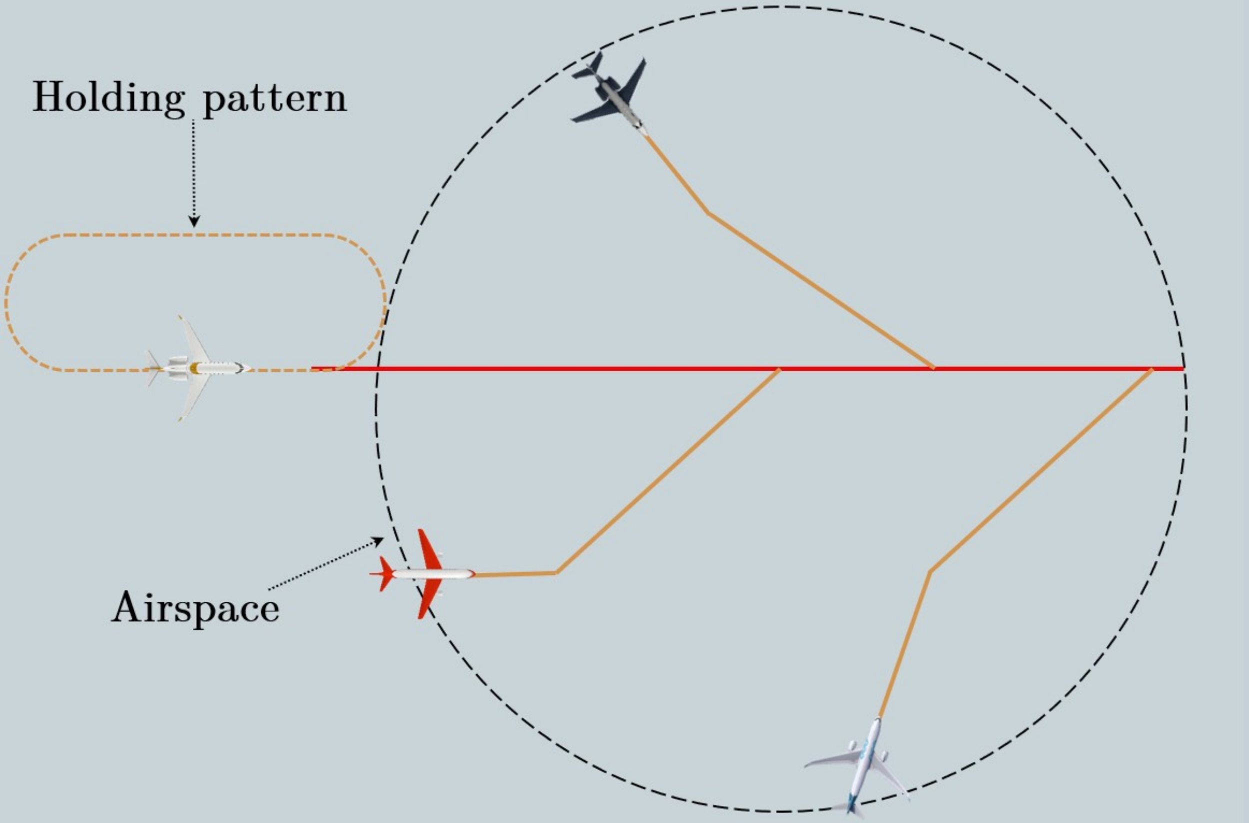
- Aircraft can detect future conflicts in multi-segment flight plans and compute ***4D conflict-aware flight plans*** to avoid those conflicts



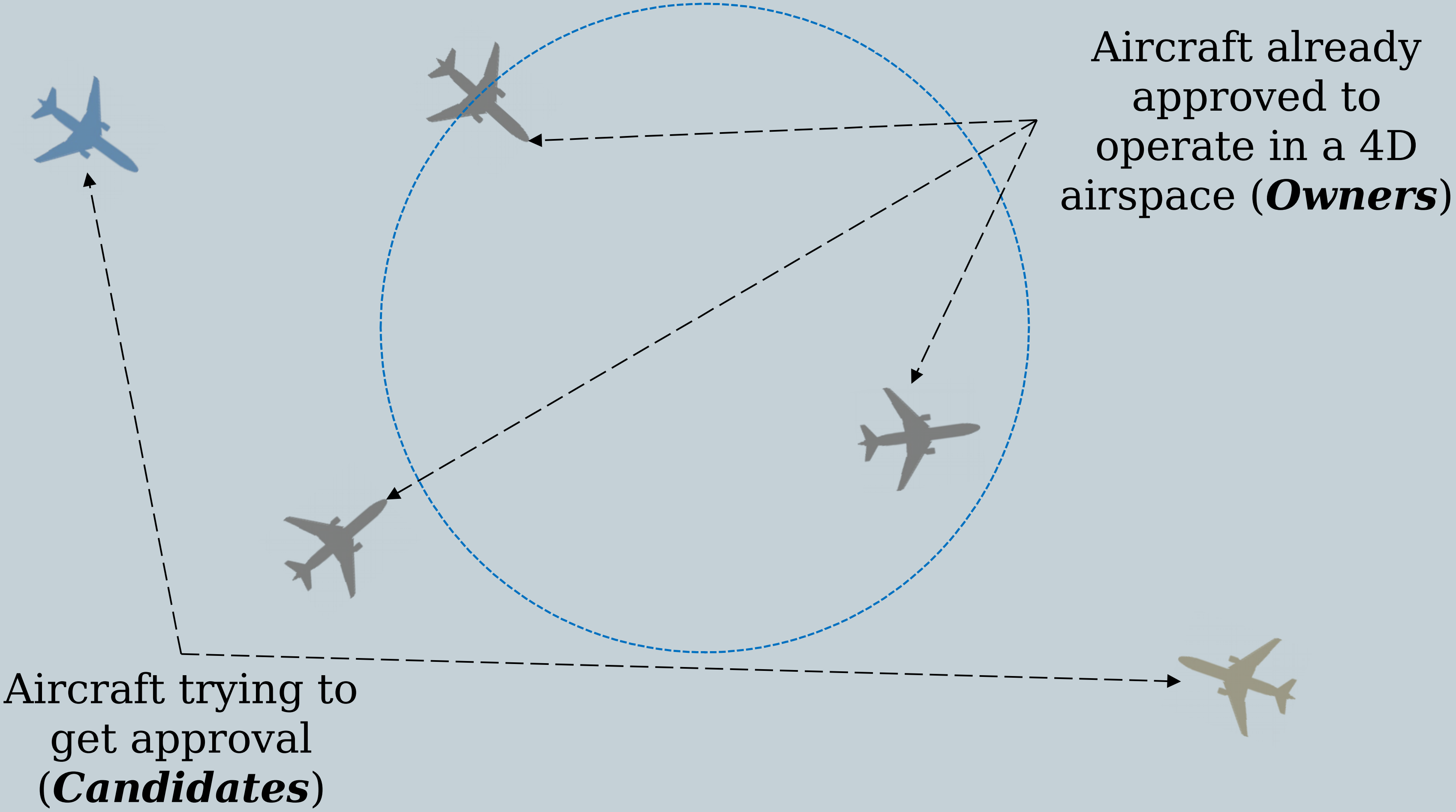
Introduction

Decentralized Admission Control (DAC) [Paul et al., NFM 2021]

- Conflict-aware flight-plans can be used for admission control into *well-defined **4D airspaces***



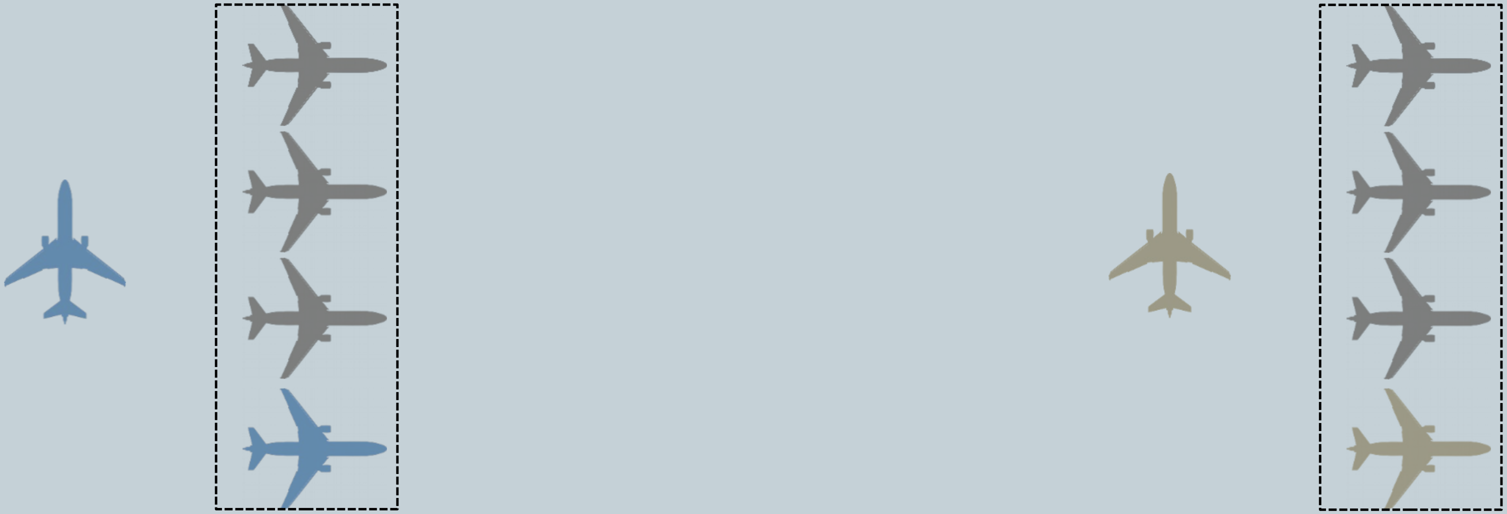
Introduction



Aircraft already approved to operate in a 4D airspace (***Owners***)

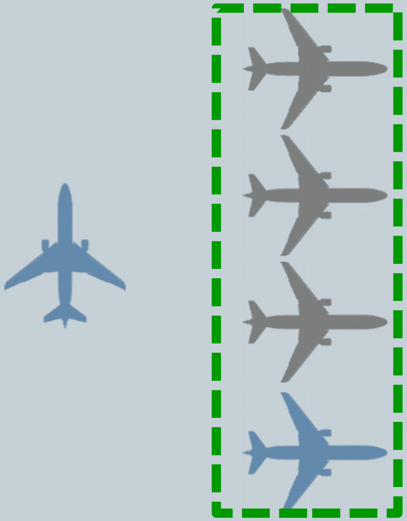
Aircraft trying to get approval (***Candidates***)

Introduction

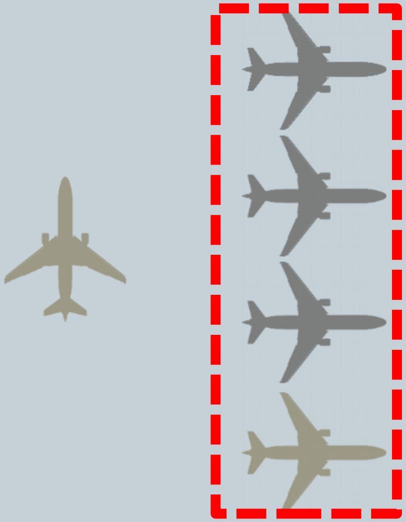


Each Candidate Creates a Conflict-Aware Set of Safe Flight-plans With the Current Owners and Tries to Get the Owners to Agree on That Set

Introduction



Set of Owners Changes if the Owners Agree on One of The Proposed Sets



Any Other Proposed Set **Becomes Invalid**

Only one aircraft can be granted access at a time - **sequential admission**

Introduction

- Decentralized Admission Control (DAC) involves two steps
  - First, a ***consensus protocol*** is needed to agree on a single proposal
    - Guarantees that a *sufficiently large subset* of the aircraft will *agree* on a single candidate's proposal
    - Does not guarantee that all aircraft *relevant to the airspace* will learn about the agreement
      - Relevant aircraft include the new set of owners and all expected future candidates
  - Next, a ***knowledge propagation protocol*** is needed to propagate the knowledge of the agreed upon proposal
    - Propagates the knowledge of the agreement among all *relevant aircraft*
    - Allows the system attain a *safe state* of knowledge with respect to the agreed upon value

Introduction

- UAM applications are ***time-critical***
  - Air-traffic data has a short useful lifetime
  - Aircraft have limited time to remain airborne

- ***Progress*** in distributed protocols
  - The protocol will achieve its intended goal
  - Two types of guarantees:
    - ***Eventual progress*** - *Eventually*, the protocol will succeed
    - ***Timely progress*** - Associated with some definite time

- Previously, we have presented a ***Two-Phase Acknowledge Propagation (TAP) protocol*** for DAC and verified its *eventual progress property* ***[Paul et al., DASC 2020]***

Introduction

- **Challenges with formally guaranteeing timely progress in the Two-Phase Acknowledge Protocol (TAP)**
  - Time for progress is dependent on multiple factors
    - Number of messages involved
    - Message transmission delays
    - Message processing delays
  - Asynchronous communication in the Internet of Planes (IoP)
    - No known bounds on message processing and transmission delays
    - Impossible to provide deterministic guarantees of timely progress
  - Mobility of nodes in airborne communication
    - Will necessitate the use of *ad-hoc* networks
    - Formal guarantees will need to be based on theories appropriate for ***Vehicular Ad-Hoc Networks*** **(VANET)**

Contributions

- **Contributions of this work**:
  - Formalize theory of the ***Multicopy Two-Hop Relay Protocol (MTR)*** to model message transmission delays in airborne networks
  - Formalize theory of the ***M/M/1 queue system*** to model message processing delays in airborne networks
  - Develop a formal ***timely progress guarantee*** for our ***Two-Phase Acknowledge Protocol (TAP)*** using theories of MTR protocol and M/M/1 queue system
  - Develop a ***formal library*** tailored towards reasoning about distributed protocols in airborne networks in the ***Athena proof assistant***

The Two-Phase Acknowledge Protocol

- We have previously proposed a ***safe*** state of distributed knowledge for DAC denoted by $E^2\phi$ [Paul et al., DASC 2020]
  - ϕ denotes the new set of owners after consensus
  - $E\phi$ implies that every aircraft knows ϕ
  - $E^2\phi$ implies that every aircraft knows that every aircraft knows ϕ

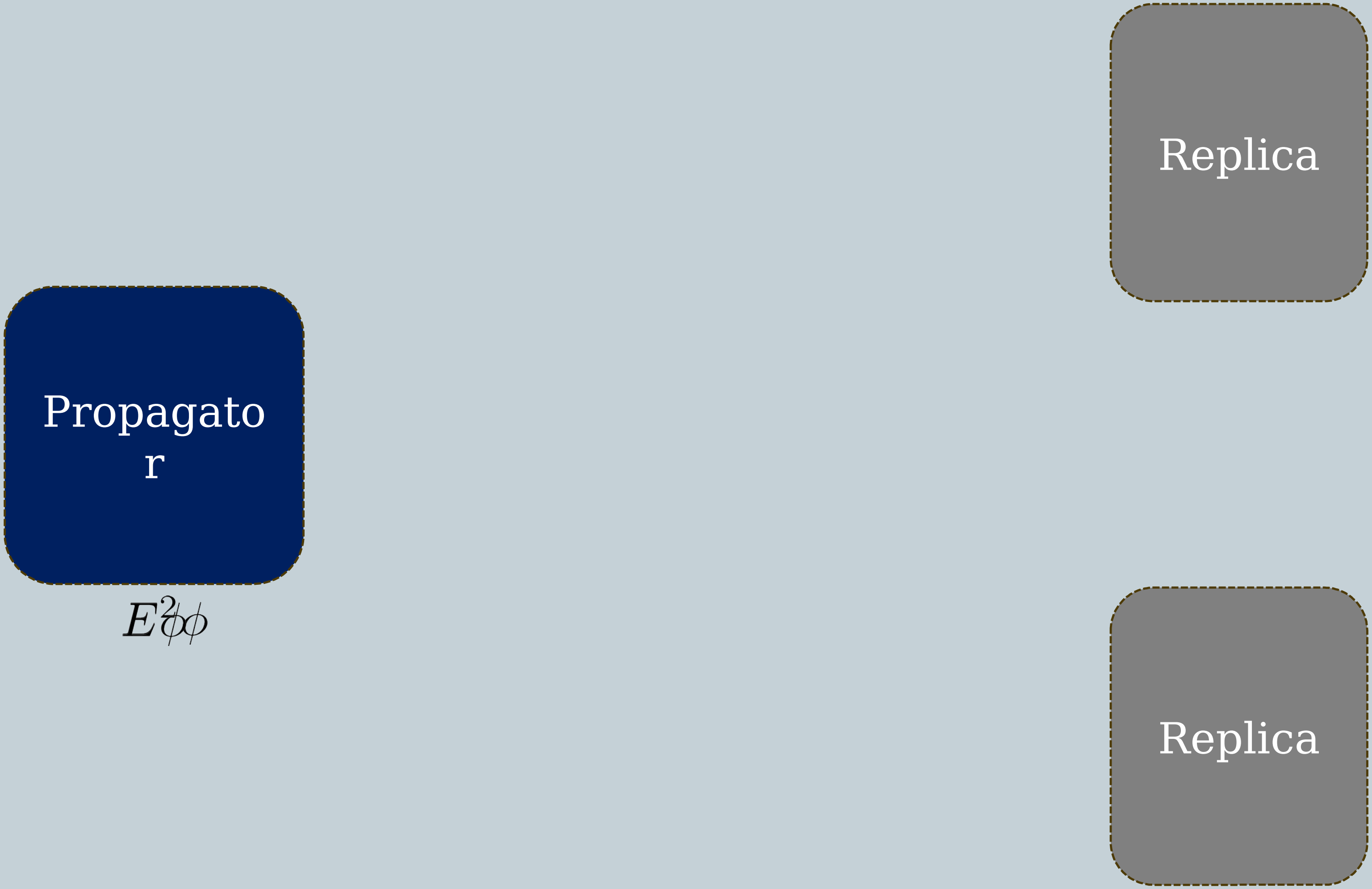
- The Problem Statement for knowledge propagation in TAP
  - ***K*** - the set of aircraft which know the agreed upon proposal ϕ
  - ***S*** - the set of all aircraft relevant to a given 4D airspace
  - *Each member of K should try to propagate ϕ to attain $E^2\phi$ with respect to S and at least one member of K should eventually learn that $E^2\phi$ has been attained*

The Two-Phase Acknowledge Protocol

- Two logically separate non-empty sets of agents
  - *Replicas*
    - Can learn a value
  - *Propagators*
    - Each knows the value ϕ
    - Each knows the membership of the set of replicas

- For Decentralized Admission Control (DAC)
  - Propagators are the aircraft which know about the agreement
  - Replicas include all aircraft relevant to an airspace

The Knowledge Propagation Protocol



Replica

Propagato
r

$E^2\phi\phi$

Replica

Probabilistic Guarantees of Timely Progress

- Not possible to deterministically predict message delays under asynchronous conditions
  - Not possible to provide deterministic guarantees of progress

- Guarantees of eventual progress are not sufficient for time-critical UAM applications

- ***Probabilistic guarantees*** can be provided
  - *E.g.*, A fact ø will be propagated to attain $E^2 \phi$ within 5 seconds with 98% probability
  - Candidates can decide to compute plans that will start after 5 seconds in order to best ensure that they will remain valid

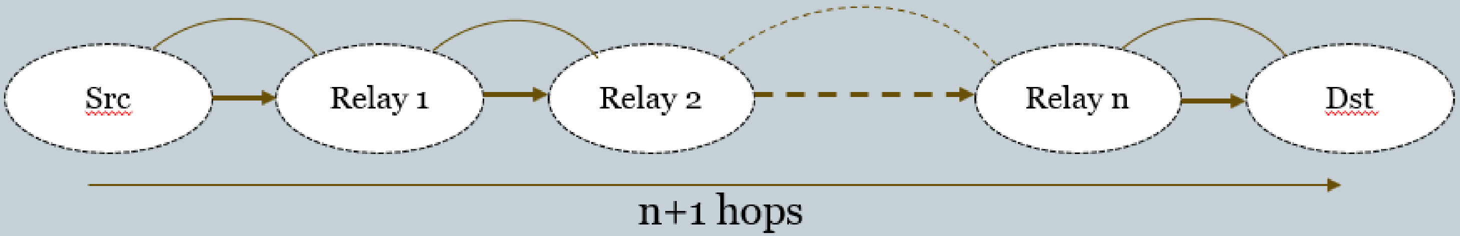
Probabilistic Guarantees of Timely Progress

- Probabilistic guarantees of progress can be developed by ***stochastically modeling*** the operational environment of Two-Phase Acknowledge Protocol (TAP)
  - Theories used should be reasonable for modeling communication in **airborne networks**
    - Airborne networks are *dynamic* in nature as the aircraft are mobile
    - Message transmission may require *relaying* when the source and the destination are not directly connected
    - Received messages must be placed in queues until they are processed

- To stochastically model message delays in TAP we use theories from ***relaying protocols*** and ***queueing systems***

The Multicopy Two-Hop Relay Protocol

- **Relaying** - a message travels from a *source* to a *destination* via one or more *relay nodes*



  - ***Two-hop relaying*** has been shown to be an effective mode of communication in VANETs *[Grossglauser & Tse, 2002], [Wang & Zhao, 2006]*
    - A message is passed via a maximum of one relay node
    - Suitable for modeling airborne communication for UAM

- We have formalized some useful analytical properties of the ***Multicopy Two-Hop Relaying (MTR)*** protocol proposed by *[Al Hanbali et al., 2007]*

The Multicopy Two-Hop Relay Protocol

- **MTR specifications**
  - Nodes are said to *meet* when they are within transmission range
  - A message is instantaneously transmitted when nodes meet
  - *Inter-meeting times* are i.i.d. with the common cdf $G(t)$
  - Each relay node can hold only one copy of a message
  - The source transmits a copy of the message to multiple relay nodes (multicopy)
  - Each message has an associated time-to-live (TTL) after which it has to be dropped by a relay node

- The message transmission time T_{D_m} for MTR is dependent on the TTLs, and the mobility model of the aircraft

The Multicopy Two-Hop Relay Protocol

- **To derive a probabilistic bound on** T_{D_m}, we assume:
  - Unrestricted time to live (TTL) for all messages
  - The inter-meeting times of the nodes are exponentially distributed with a *rate parameter* λ_{MTR}
  - A source aircraft transmits a message to all relay aircraft

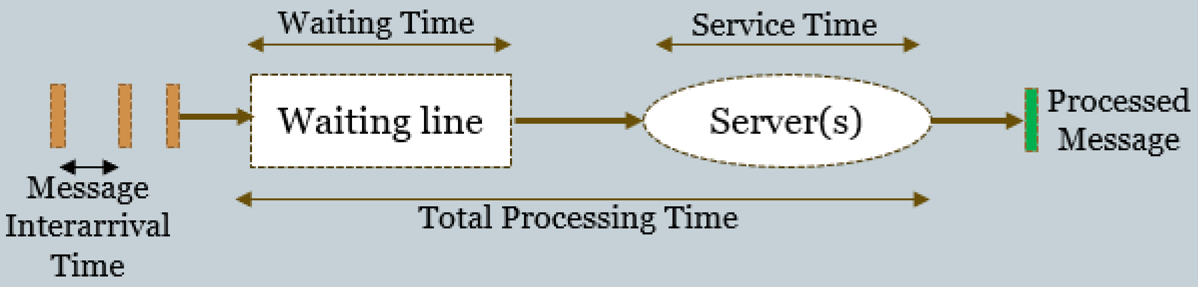
- Probabilistic bound on T_{D_m}
  - $P\left(T_{D_m} \leq t\right) = 1 - e^{-\lambda_{MTR}Mt}$
    - M is the number of relay nodes + 1

M is the number of relay nodes + 1

- T_{D_m} is exponentially distributed with rate parameter $\lambda_{D_m} = \lambda_{MTR}M$

The M/M/1 Queue System

- Queueing theory has been extensively used for modeling throughput in VANETs *[Wang et al, 2012], [Wen-jie et al, 2011], [Yin & Lin, 2004]*



- ***The M/M/1 queue***
  - Consists of a *single server* that processes all messages
  - Can be used to model message processing at each aircraft as the received messages need to be processed sequentially for TAP to work

The M/M/1 Queue System

- **M/M/1 has some analytical properties that enable reasoning about** the message processing delays $\boldsymbol{T_{P_m}}$
  - T_{P_m} is exponentially distributed
  - Interarrival times of messages are exponentially distributed with a rate parameter λ_a
  - Service time of messages (processing time - waiting time) is exponentially distributed with mean $1/\mu_s$

- We have proven that
  - $\lambda_{P_m} = \mu_s - \lambda_a$
    - Where λ_{P_m} is the rate parameter of T_{P_m}

Timely Progress in TAP

- Dependent on the rate parameters λ_{D_m} and λ_{P_m} of the message transmission and processing delays
- We make the following assumptions
  - All aircraft use the MTR protocol for message transmission
  - Each aircraft independently implements an M/M/1 queue to process the messages it receives
  - The transmission delays of the messages are independent and identically distributed
  - The processing delays of the messages are independent and identically distributed
  - The transmission delays are independent of the processing delays

- Dependent on the rate parameters and of the message transmission and processing delays
- We make the following assumptions
  - All aircraft use the MTR protocol for message transmission
  - Each aircraft independently implements an M/M/1 queue to process the messages it receives
  - The transmission delays of the messages are independent and identically distributed
  - The processing delays of the messages are independent and identically distributed
  - The transmission delays are independent of the processing delays

Timely Progress in TAP

- In the Two-Phase Acknowledge Protocol, a deterministic number of messages are involved in progress with respect to a propagator
  - The propagator sends a *learn* and an *all-know* message to each replica
  - A replica sends a *learnt* and an *acknowledgement* message to the propagator
  - 4 messages are involved per replica
  - 4 x R messages for R replicas

- Total time for progress
  - $T_S = T_D + T_P = \sum_{m=1}^{N_M} T_{D_m} + \sum_{m=1}^{N_M} T_{P_m}$
  - Where $N_M = 4R$ is the deterministic number of messages

Timely Progress in TAP

- $T_S = T_D + T_P = \sum_{m=1}^{N_M} T_{D_m} + \sum_{m=1}^{N_M} T_{P_m}$

- We have derived an expression for the bound in terms of the known properties of MTR protocol and the M/M/1 queue system

We have derived an expression for the bound in terms of the known properties of MTR protocol and the M/M/1 queue system

  - $P\left(T_S \leq (x + y)\right) \geq F_{ER}(x, N_M, \lambda_{MTR} M) \times F_{ER}(y, N_M, \mu_s - \lambda_a)$
    - Where gives the cdf of an Erlang distribution with shape parameter and rate parameter with respect to a real
    - Where $F_{ER}(t, k, \lambda)$ gives the cdf of an Erlang distribution with shape parameter k and rate parameter λ with respect to a real t

A Formal Library in Athena

- We have used the **Athena proof assistant** to verify our guarantee of timely progress for TAP
- Athena *[Arkoudas & Musser, 2017]*
  - Based on ***many-sorted first order logic***
  - Uses ***natural deduction*** style of proofs
  - Provides an environment for interactive theorem proving
  - Provides ***modules*** for grouping related theories and importing them in a proof context
  - ***Soundness guarantee***
    - Any theorem that is proven is a logical consequence of axioms and proven lemmas in the *assumption base*

A Formal Library in Athena

- Mechanically verifying higher-level properties of complex systems requires access to lower-level formalizations
  - Domain-specific formal constructs are needed to express properties and specifications
    - *E.g.*, expressing the node mobility model for a relaying protocol
    - *E.g.*, expressing the specification of a distributed protocol like TAP
  - Challenging to verify using interactive theorem proving techniques
    - Requires domain knowledge of all aspects of the system to be modeled
    - Requires knowledge of formal logic and reasoning techniques
    - Requires familiarity with a machine-checkable language
    - The formalizations require significant time and effort to develop

A Formal Library in Athena

- Beneficial to have a library of pre-developed formalizations that can be used as a foundation to develop higher-level specifications and proofs
  - Domain-specific formal constructs for specifying domain-specific properties
  - Mathematical theories necessary for reasoning about common properties
  - Well-organized theories that can be used in different contexts during proof development

- We have developed a formal library in Athena that is tailored towards reasoning about distributed protocols in airborne networks

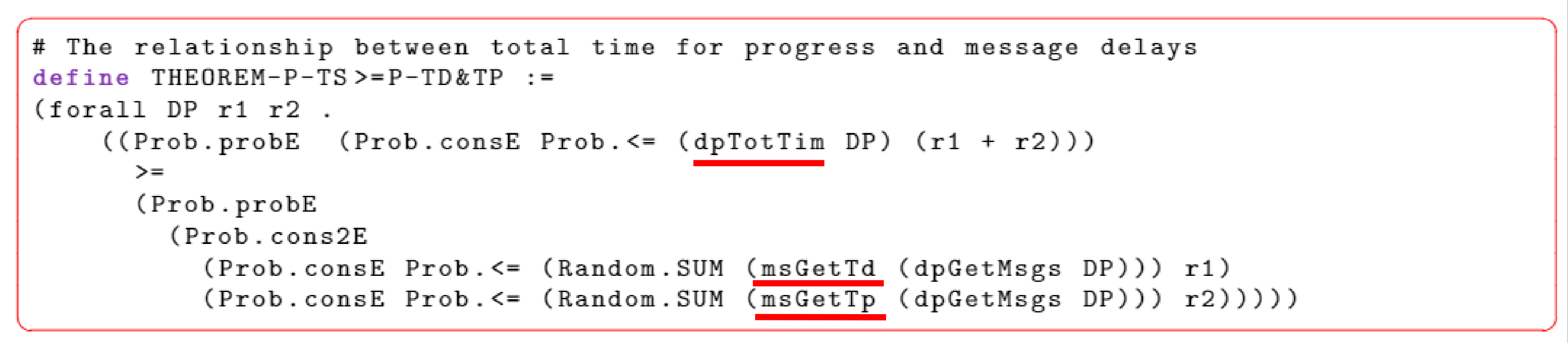
A Formal Library in Athena

- For verifying timely progress for the Two-Phase Acknowledge Protocol (TAP) we formalized theories from
  - Probability
  - Distributions
  - Random variables
  - Queues
  - Aircraft mobility models
  - Relaying protocols
  - Distributed protocols

- We adopted a ***top-down*** approach of proof development
  - Identified the higher-level progress properties of interest
  - Created the lower-level theories necessary for formal reasoning

A Formal Library in Athena

- $P(T_S \le (x+y)) \ge P((T_D \le x) \wedge (T_P \le y))$



```
# 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)))))
```

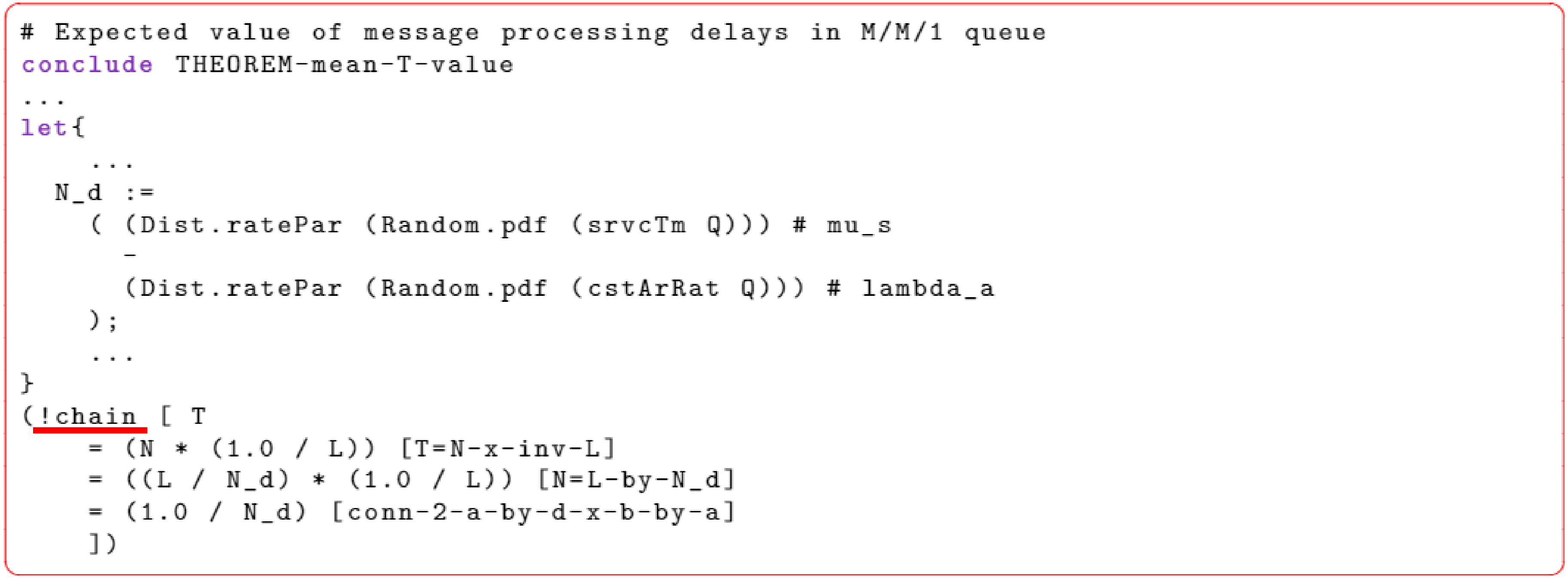
- dpTotTim, msGetTp and msgetTd, represent the total time, message processing times, and message transmission delays for the domain of distributed protocols called DistProt in our Athena formalization

dpTotTim, msGetTp and msgetTd, represent the total time, message processing times, and message transmission delays for the domain of distributed protocols called DistProt in our Athena formalization

A Formal Library in Athena

- We have used Athena's built-in proof tactics to interactively develop the proofs
  - E.g., the proof of $T_P = \frac{1}{\mu_s - \lambda_a}$ for M/M/1 queue system using the Athena's *equality chaining* tactic

E.g., the proof of for M/M/1 queue system using the Athena's *equality chaining* tactic



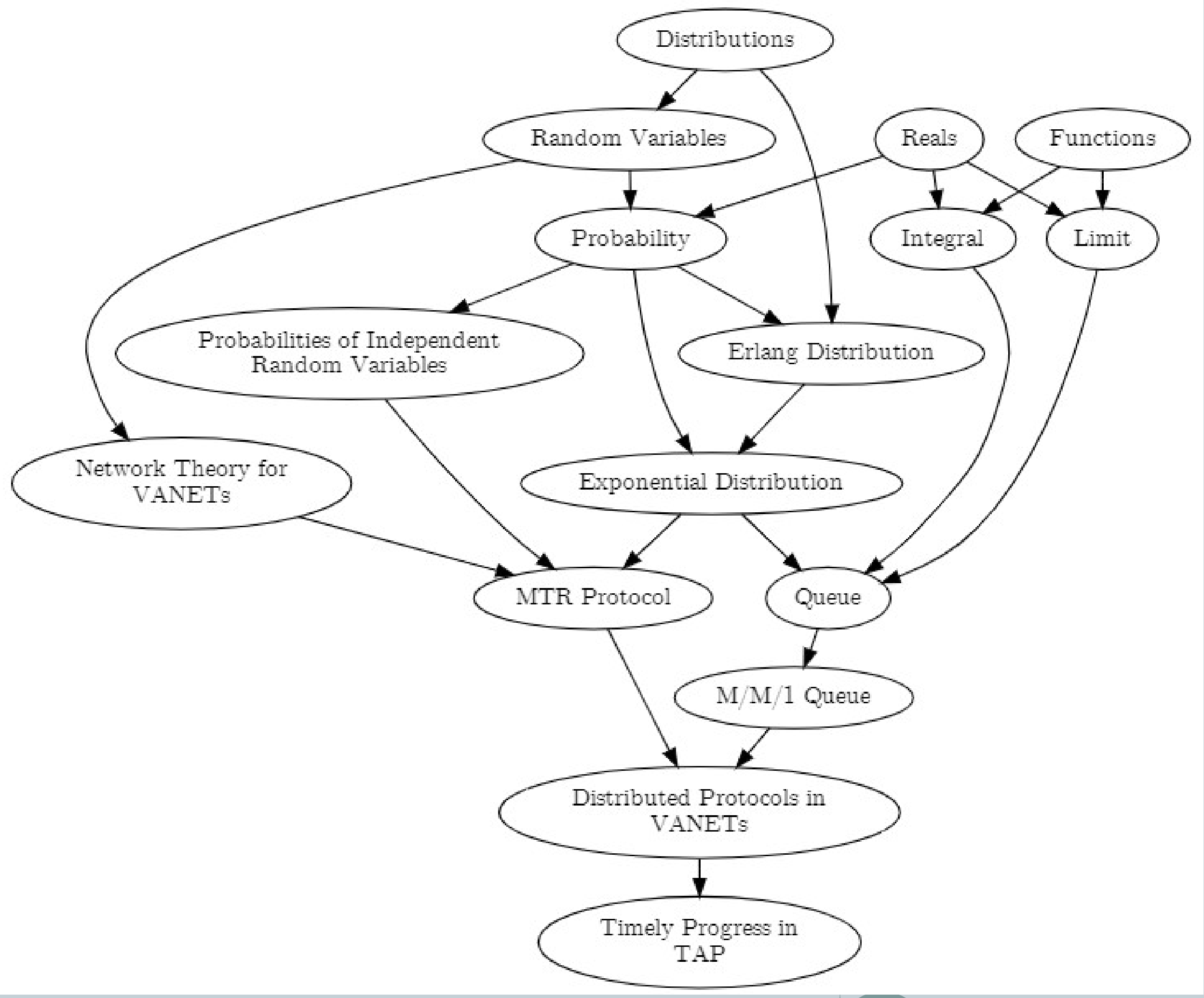
```
# Expected value of message processing delays in M/M/1 queue
conclude THEOREM-mean-T-value
...
let{
    ...
  N_d :=
    ( (Dist.ratePar (Random.pdf (srvcTm Q))) # mu_s
      -
      (Dist.ratePar (Random.pdf (cstArRat Q))) # lambda_a
    );
    ...
}
(!chain [ T
    = (N * (1.0 / L)) [T=N-x-inv-L]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

A Formal Library in Athena

- Currently, our formalizations contain some conjectures which are unproven
  - All such conjectures used are well-established results from the domains
  - Our library currently lacks the formalizations of lower-level theory necessary to verify them
  - As the goal of the paper was to verify high-level progress properties for TAP that are interesting for UAM, the proofs are left for future iterations of the library

- The library currently contains 27 main theorems from the different domains

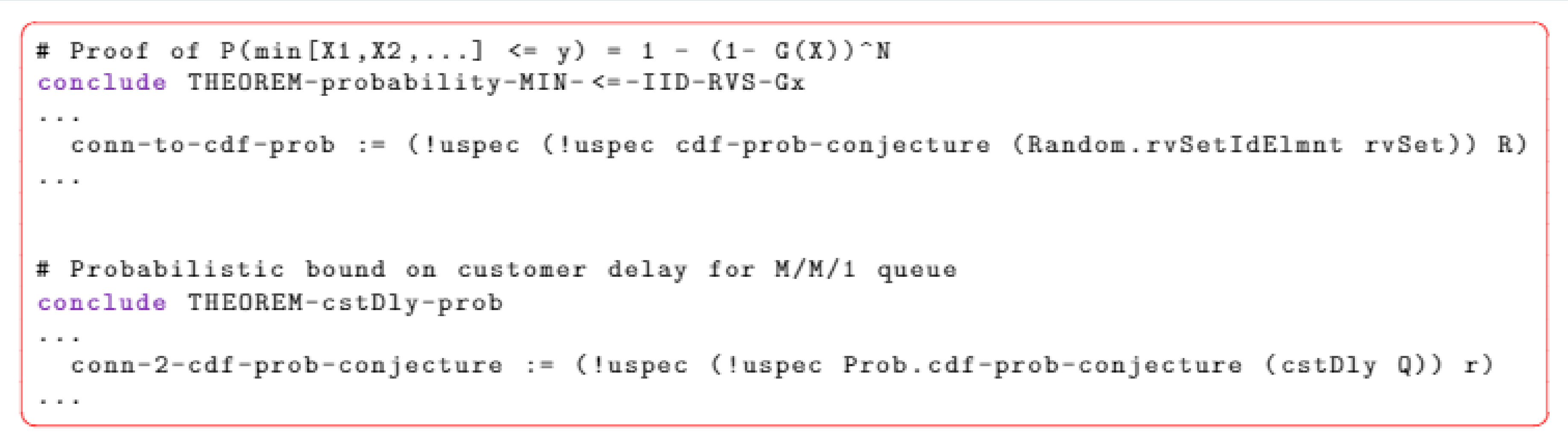
A Formal Library in Athena



- The theories have been grouped into well-defined modules
- Can be imported easily

A Formal Library in Athena

- We have tried to make our Athena formalizations reusable and generic
  - Can be easily *plugged into* different contexts during proof development
  - *E.g.*, use of the `cdf-prob-conjecture,` our Athena formalization of the relationship between cdf and probability, in different contexts



```
# 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)
...
```

Conclusion

- **Contributions**
  - Probabilistic guarantees of timely progress for the Two-Phase Acknowledge Protocol (TAP)
    - Useful for time-critical multi-agent distributed coordination for UAM
    - Based on well-established fundamental theories of airborne communication
  - A proof library tailored for reasoning about distributed coordination in VANETs
    - Can be used as a foundation for verifying properties of other distributed protocols for autonomous multi-agent UAM operations

- **Limitations**
  - Some conjectures have been left unverified
  - The guarantees will become invalid if the assumed operating conditions do not hold at runtime
  - No support yet for protocols involving non-deterministic number of messages

Future Work

- Formalize the lower-level theory required to prove all conjectures
- Model other possible operating conditions
  - *E.g.*, non-i.i.d. message delays
  - Can provide different guarantees based on runtime conditions
    - If one guarantee doesn't hold at runtime, another guarantee may hold
- Model other relaying protocols designed specifically for VANETs
  - *E.g.*, *AeroRP* *[Jabbar et al., 2008]*
- Support for distributed protocols with non-deterministic number of messages
  - *E.g.*, the Synod consensus protocol
  - Will need theory of transition sequences, probabilistic fairness for real-world airborne networks, etc.

Questions?

This work was supported by the **Air Force Office of Scientific Research (AFOSR)** and the **National Science Foundation (NSF)**