FORMALIZING INTER-SATELLITE COMMUNICATION SPECIFICATION IN SMALL SATELLITE SYSTEM

Solomon Gebreyohannes [1], Radhika Radhakrishnan [1], William Edmonson [1],
Albert Esterline [1], and Fatemeh Afghah [2]

[1] North Carolina A&T State University, Greensboro, NC, USA
[1] (shgebrey, rradhakr, wwedmons, esterlin)@ncat.edu
[2] Northern Arizona University, Flagstaff, AZ, USA
[2] fatemeh.afghah@nau.edu

ABSTRACT

This paper presents a formal specification and verification of Inter-Satellite Communication (ISC) system for small satellite that uses an optimal multiple access protocol. We proposed a novel hybrid combination of Time Division Multiple Access (TDMA)/Code Division Multiple Access (CDMA) scheme for communication between satellites in a small satellite network [1]. Our protocol addresses the problem of multiple access, OSI data link layer, in heterogeneous small satellite networks and adapts to the network scale. In this paper, as a continuation on the development of Responsive and Formal Design (RFD) process for ISC, we formalize the ISC specification, utilizing the hybrid protocols and the Prototype Verification System (PVS) for verification. The RFD process is a design methodology that combines Model-Based Systems Engineering (MBSE) to manage system modeling complexity with formal methods to ensure that designs are verifiably correct against their requirements. We will show the process for proving consistency of the ISC by checking the protocol's well-formedness (no contradiction) and verifying its properties (and requirements) through a formal proof.

1 INTRODUCTION

Requirements are used to describe the operations of a system. “They are necessary attributes in a system, statements that identifies capabilities, characteristic, or quality factors of a system in order for it to have value and utility to a customer or user” [2]. Requirements are usually written in a natural language. Following the standard practice, we represent the requirements of Inter-Satellite Communication (ISC) system (at the data link layer) using a natural language [1]. This representation, however, is ambiguous and lacks a way to check the system requirements regarding functionality, consistency, and completeness. Rather, a system formulated in a formal language might be proved consistent and complete. Hence, we need to translate the system requirements to a set of logical formulae.

We proposed a design process that is specific to designing ISC based on the OSI framework using the Responsive and Formal Design (RFD) process [3]. The RFD process [4,5] represents a procedure used for designing Cyber-Physical Systems (CPS) in general and small satellites in particular. It relates a set of requirements, associated models, simulations, and the relationship between them, by integrating Model-Based Systems Engineering (MBSE) to manage system modeling complexity [6] with formal methods [7] to ensure that designs are verifiably correct against their requirements. The RFD process is integrated with the conceptual OSI framework [8] to produce reliable inter-satellite communication. It consists of a set of levels of abstraction. Each level of abstraction covers the OSI framework in varying levels of details i.e. the OSI layers will emerge as we move to lower levels of abstraction. At the highest level, we view the OSI communication framework as unpartitioned layer providing the means to communicate with other similar devices.

The various layers of the OSI framework, from Application to Physical layer, are viewed laterally. As we move to lower levels of abstraction, high level communication concepts are refined and expressed in lower level representations. Finally, at the lowest level of representation, we should end up with the different layers of the OSI stack, expressed laterally.

We propose in the paper, through the implementation of the RFD process, the formal proof of the ISC protocol by transforming it to its logical form using PVS. This will allow the proof of soundness and completeness of the ISC protocol. In particular, we will base this proof on the TDMA-centric protocol of the OSI Data Link Layer. As proposed in [1], a novel hybrid Time Division Multiple Access (TDMA)/Code Division Multiple Access (CDMA) protocol addresses the problem of multiple access in heterogeneous small satellite networks and adapting to the network scale. Two different approaches of TDMA-centric and CDMA-centric are introduced in [1] depending on the leading technique of channel access. In these schemes, users share the same frequency channel by dividing the signal into different time slots (i.e. TDMA) or assigning codes to users (i.e. CDMA). We use the Prototype Verification System (PVS) to formalize a TDMA-centric ISC system. PVS [9] is a formalism for the design and analysis of system specifications. It has a highly expressive specification language, based on higher-order logic, with a rich-type system [10]. Using a higher-order theorem-proving system (such as PVS), it is possible to reach a much higher level of confidence, compared to lighter formal methods. The formalizing process for the TDMA-centric and CDMA-centric approaches are similar, hence in this work we present the formalization for only the TDMA-centric system, which can be extended to the CDMA-centric system with minor modifications.

Some of related areas to which (semi-) formal methods are applied are wireless networked self-organized systems [11], telecommunication systems and communication protocols [12], development activities of telecommunications networks [13], the design process of telecommunication Equipment Protection Switchers (EPSs) [14], and GIS on-demand services [15]. Formalization of specifications using PVS has also been done in different application areas, e.g., airline reservation system [16], “Autopilot” specification [17], and space shuttle software requirements [18,19].

The rest of this paper is organized as follows. Section 2 introduces the RFD process and ISC. A brief introduction to PVS presented in Section 3 follows [9,10,20-22]. Section 4 presents the formal framework used to design ISC based on the RFD. Following the RFD procedure, Section 5 presents the formal specification and verification of the inter-satellite communication system using PVS. We define two PVS theories in Section 5.1. The first encodes some basic objects such as packet, small satellite, cluster, network of satellites, and frame, and then captures some notions that our formal specification will be describing. The second PVS theory defines a TDMA-centric ISC system. Section 5.2 presents the verification of the inter-satellite communication specification. In communication systems in general, one can encode properties such as security, uniqueness, or coverage of a network and verify them. For example, we verify the requirement: “In a network of small satellites, the transmitted signal carries the unique id (or code) of the receiver satellite so that the message will only be received and decoded by the intended user”. Section 6 concludes this paper.

2 BACKGROUND

2.1 The Responsive and Formal Design Process

The RFD process [4,5] is developed as a procedure used in designing CPS in general and small satellites in particular. Cyber-Physical Systems are integrations of computational and physical processes in which the computational part controls the physical entities [23]. Therefore, a paradigm shift to the CPS design philosophy that elevates the computational part, algorithms and their associated hardware/software components, to the forefront of the design process is necessary. The RFD process combines MBSE to manage system modeling complexity with formal methods to ensure that designs are verifiably correct against their requirements. The integration of formal methods throughout the design process as an integral part of requirements management provides a high-confidence system. The framework which we follow in implementing this RFD process is based on mission design flow [24], and is iterative.

Designing a system using a short and agile process relies on the ability to characterize system function at various levels of abstraction [25]. The RFD process consists of different levels of abstraction. Each level of abstraction, A_i, in an MBSE design generally represents a set of requirements and its associated models, simulations, and the relationships between them. The illustration below summarizes the essence of the RFD process.

$$\begin{tikzcd}[column sep=small, row sep=small] A_i : L^i_n \arrow[r, Leftrightarrow] & L^i_l \arrow[r, Leftrightarrow] \arrow[d] & M^i \arrow[r, Rightarrow] & S^i_p \\ & S^i_p & & \end{tikzcd} \tag{1}$$

where the design parameters $L^i{}_n$, $L^i{}_l$, M^i, $S^i{}_p$, and $S^i{}_b$ represent requirements written in natural language form, requirements written as a set of logical expression, system of interconnected models, simulations based on the parameters of M^i, and simulations based on the logical description of $L^i{}_l$, respectively. This work will prove the ISC protocols based on the relationship between $L^i{}_n$ and $L^i{}_l$ via PVS.

System requirements expressed in natural language is the starting point of the RFD process. As a model-based process, it produces two main system models representing the logical and behavioral aspects of the requirements. A simulation is successful if all constraints associated with attributes of the system are met. The objective of behavior model simulation is to describe the operations of a system and the flow of information between the different subsystems. Traditional system simulation can be described as parametric since it focuses on the parameters of a system model. The RFD process introduces formal methods and simulations of system behavior based on the formal logics used to capture the system requirements. There are different levels of system abstraction and refinement in the RFD process. Refinement and abstraction relate inversely to each other. As we go down in the levels (towards the implementation), refinements of finer granularity are obtained.

2.2 Inter-Satellite Communication using Hybrid TDMA/CDMA Protocol

Multiple satellite missions are a new trend in research in the space industry. A multi-satellite solution is highly economical and helps to provide improved spatial and temporal resolution of the target. For future space missions, a large number of heterogeneous small satellites can be deployed in space as a network and thus requiring Inter- Satellite Communications. ISC allows autonomous transfer of data analogous to terrestrial Internet with autonomous transfer of data and assists in performing advanced functions.

The current state of the art of ISC is a one-hop link between satellite and ground stations. Space agencies have developed future missions involving multiple satellites with inter-satellite communication intended to achieve mission objectives such as gravity mapping, servicing or proximity operations, etc. Examples of multiple satellite missions with inter-satellite communication are Iridium, Or-blink, Proba-3 [26], QB-50 mission [27], Teledesic [28], Edison Demonstration of Smallsat Networks (EDSN) mission [29], ESPACENET [30], and NASA's Autonomous Nano-Technology Swarm (ANTS) [31]. Much work remains, however, in order to achieve an in-depth understanding of the communication architecture needed in an absolutely autonomous and heterogeneous network of small satellites.

To facilitate ISC between small satellites, we proposed to use the OSI model as a framework to serve as a reference tool for communication among devices connected in a network. This divides the communication process into different layers. However, the performance of the entire system largely depends on the design of multiple access protocols. The MAC protocol should take into account mission specifications such as mission application, network topology, number of satellites, etc. Also, it has to consider several system constraints of small satellites, for example, limited on-board power and computing resources. There are several research projects being conducted on various multiple access methods for inter-satellite communications in small satellite systems [32 -39]. A novel hybrid TDMA/CDMA protocol for a cluster of satellites is proposed in [1] and will be formalized in this paper. We suggested two different approaches, TDMA-centric and CDMA-centric, which address the problem of multiple access in heterogeneous small satellite networks. A combination of TDMA with Direct Sequence CDMA (DS-CDMA) was investigated, where they both offer collision free transmission and their combination improve the system scalability and adaptivity.

The small satellite network can be divided into clusters with each cluster having a master satellite and several slave satellites. The proposed system model is shown in Fig.1.

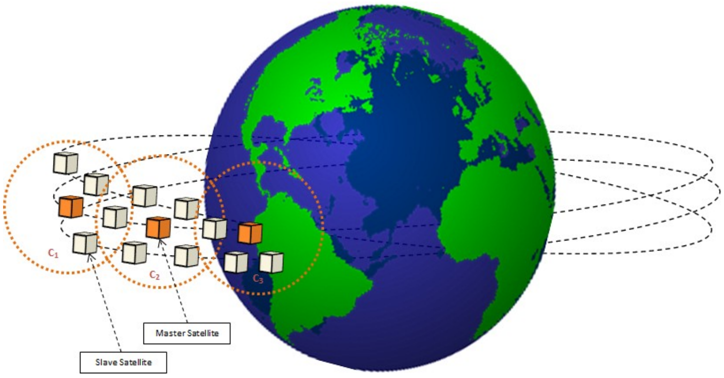


Figure 1. Overlapped cluster of small satellites

The slave satellites within a cluster communicate with the master satellite, and the master satellite forwards the data to the destination. If a member satellite needs to communicate with a satellite in another cluster, it first communicates with its own master satellite, which in turn communicates with the destination master satellite and thus forwards the data, thereby consuming much power. It is hence necessary to re-cluster the network. We propose to use a closeness centrality algorithm for the selection of a master satellite that satisfies the minimum power requirement (*threshold*, P_{th}).

The hybrid TDMA/CDMA can be implemented using two different approaches: TDMA-centric and CDMA-centric. In the TDMA-centric approach, which we will address in this paper, each cluster is assigned a unique code. Each satellite has dedicated time slots for uplink and downlink to transmit the data to and from the master satellite.

Multiple satellites from different clusters transmit in the same slot without interference using different codes. Fig. 2 shows the TDMA-centric frame structure.

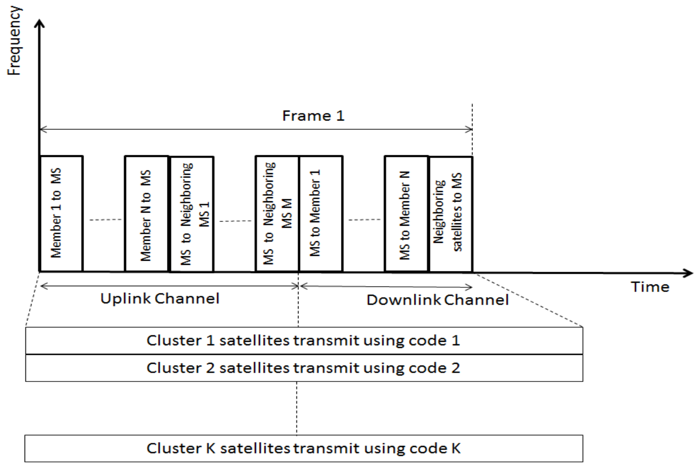

Frequency
Frame 1
Member 1 to MS
Member N to MS
MS to Neighboring MS 1
MS to Neighboring MS M
MS to Member 1
MS to Member N
Neighboring satellites to MS
Time
Uplink Channel
Downlink Channel
Cluster 1 satellites transmit using code 1
Cluster 2 satellites transmit using code 2
Cluster K satellites transmit using code K

Figure 2. The Frame structure in TDMA-Centric approach

In the CDMA-centric approach, each satellite is assigned a unique code. The member satellites can transmit data simultaneously to the master satellite in the first slot without interference using the respective orthogonal codes as shown in Fig. 3. For the master satellite, there are dedicated slots to transmit data to the neighboring satellites and downlink slots for receiving data from the neighboring satellites.

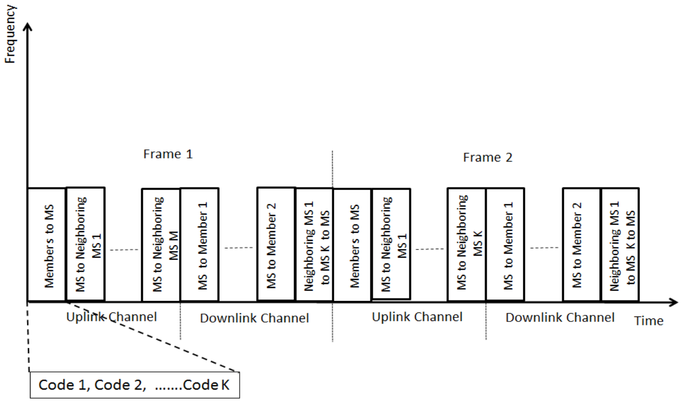


Figure 3. The Frame structure in CDMA-Centric approach

The proposed hybrid TDMA/CDMA protocol addresses the design needs of a large number of small satellites within a reconfigurable network. It allows for the simultaneous transmission of data in the allocated time slots by all satellites without interference. For a pure TDMA system, the addition of more satellites will be an issue that can be overcome using CDMA technology based on clustering, thus supporting a large scalable network. The hybrid protocol has less delay compared to other MAC protocols, thereby making it suitable for missions that require tight communication links such as servicing and proximity operations. The TDMA-centric hybrid protocol can be used in missions where the packet size varies considerably, where a variable number of slots (adaptive TDMA) are allocated depending on the size of the data packet provided there is a good control channel allocation. The cluster head must inform the members to refrain from using their slots in order to avoid collisions. The CDMA-centric system can be used when the packet size is relatively consistent and also for missions where it is required to broadcast some important information to the cluster members, for example, proximity operations. The selection of MAC protocols largely relies on the mission objectives and the number of satellites in the whole system.

3 PROTOTYPE VERIFICATION SYSTEM OVERVIEW

PVS is a verification system that consists of a specification language integrated with support tools and a theorem prover and based on higher-order logic [9,20]. It has a rich set of built-in types and type-constructors and is strongly typed. *Types* can be defined starting from base types (booleans, numbers, etc.) using the *function*, *record*, and *tuple* type constructions. The terms of the language can be constructed using function application, lambda abstraction, and record and tuple construction. Specifications are logically organized into parametrized theories and data types. A *theory* consists of type names and constants, and also axioms, definitions, and theorems associated with them. Specifications for many foundational and standard theories are preloaded into PVS as *prelude* theories. PVS also allows definition of *Abstract Datatypes (ADTs)*, from which a complete PVS theory is automatically synthesized during type checking. Details on the PVS language may be found in the *PVS Language Reference* [21].

The PVS *parser* and *typechecker* check theories for syntactic and semantic consistencies, respectively. The typechecker adds semantic information to the internal representation built by the parser. Theorem proving may be required to establish the type-consistency of a PVS specification. The theorems that are automatically generated by the PVS for type correctness check of the specification are called type-correctness conditions (TCCs). Additional theorems can be included by the user to check whether properties (or requirements) of a system are satisfied. The users should discharge the theorems using the appropriate *prover* commands. The PVS proof display consists of a list of sequent formulas as shown in Eq. 2.

$$\frac{antecedents}{consequents} \tag{2}$$

whereby the interpretation of a sequent is that the conjunction of the antecedents implies the disjunction of the consequents. Details about proofs can be found in the PVS Prover Guide [22]. There are also PVS tutorials and applications developed in [16,38,39].

4 FORMAL FRAMEWORK FOR THE DESIGN OF INTER-SATELLITE COMMUNICATION SYSTEM

We integrate the RFD process with the conceptual OSI framework, Fig. 4, to produce a reliable inter-satellite communication system [3]. The integrated framework, Fig. 5, consists of a set of levels of abstraction. Each level of abstraction covers the OSI framework in varying levels of details i.e. the OSI layers will emerge as we move to lower levels of abstraction. At the highest level, we view the OSI communication framework as unpartitioned layer providing the means to communicate with other similar devices. The various layers of the OSI framework, from application to physical layer, are viewed laterally instead of the usual vertical arrangement. As we move to lower levels of abstraction, i.e. $A_i \rightarrow A_{i+1}$ (see Eq. 1), of the representation we identify high level communication concepts are refined and expressed in lower level representations. Finally, at the lowest level of representation, we should end up with the different layers of the OSI stack, expressed laterally.

All the seven (five, for small satellites) known layers of the OSI model may not be visible at each level, especially at higher levels of abstraction. It is a common practice to have some derivatives of the OSI framework, merging layers together. However, more layers come to view as we proceed in the design process for refinement. The refinement also helps to make clear the connection with parametric considerations that are represented by $\{M, S_p\}$ in the basic RFD equation. Unlike many traditional design methods, our design process integrates high level requirements with domain specific considerations and verifies formally. As part of this process, this paper, for example, shows the formalization of ISC at the data link layer for a given level of abstraction. This is important to achieve consistency across the OSI layers. We must ensure that each layer, from application to physical layer, is consistent.

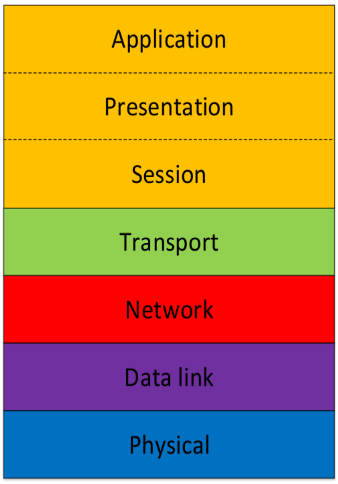


Figure 4. The OSI model

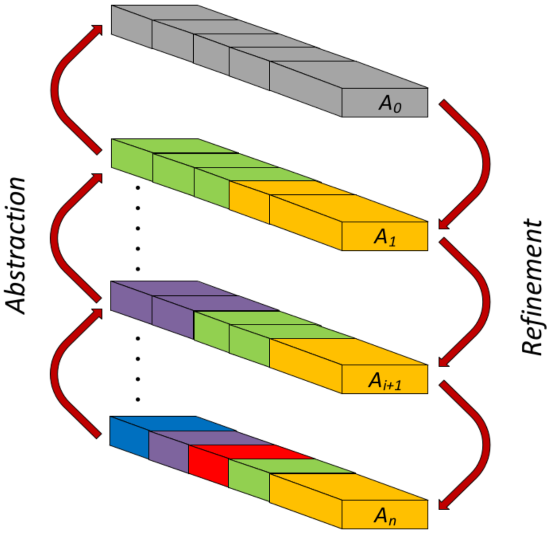


Figure 5. Integration of the RFD process with the OSI framework [3]

It is also equally important to maintain consistent information as we proceed in the design process for refinement. As levels of RFD proceed towards refinement, the design process becomes a *local* or discipline specific activity, though always with a *global* perspective. The formal methods concepts and techniques we propose for abstraction and refinement can be found in [5]. Briefly, refinement is defined as a relation between the i^{th} and the $(i+1)^{th}$ levels of representation (where $i = 0, 1, \ldots, \textit{total number of RFD representations} - 1$), which their logical properties are preserved.

5 FORMALIZATION OF INTER-SATELLITE COMMUNICATION SPECIFICATION

Our design process ensures consistency across the OSI layers and also among different layers as we proceed for a refinement. In this section, we show how to ensure consistency at the data link layer by formalizing its protocol using PVS. To fully show consistency across the OSI layers, formal specifications for the rest of the OSI layers of the communication network should also be developed.

5.1 Formal Specification

This section provides a step-by-step explanation of the development of formal specification of the ISC system.

5.1.1 Basic Definitions

We first formally define some basic objects such as packet, small satellite, cluster, network of satellites, and frame, and then capture some notions that our formal specification will be describing.

We start by defining a satellite. The small satellite type is defined as a PVS record type with *accessors* for the satellite id and a list of packets. The accessors or fields are defined as PVS uninterpreted types. They are declared as

ID, Packet : TYPE

A satellite is then formally represented as

Sat: TYPE = [# id: ID, packets: list[Packet] #]

Several satellites make up a cluster. Each cluster consists of a set of slave satellites and one cluster head. Every cluster in the network has a unique Direct Sequence (DS) code. Hence, we represent a cluster as a PVS record type with fields for the DS code, master satellite, and a set of slave satellites. We define the *code* first as uninterpreted PVS type

Code: TYPE

One may specify the code with its exact type (not in the sense of PVS data type), Walsh-Hadamard or Gold sequences, and its orthogonality property and represent with a different PVS type (and predicate). For simplicity, however, we represent it as uninterpreted type. A cluster, then, is formally defined as

Cluster: TYPE = [# code: Code, head: Sat, sats: list[Sat] #]

Different clusters of satellites make up a small satellite network. This network is defined as

SatNetwork: TYPE = setof[Cluster]

A frame structure for a TDMA-centric system is represented using a PVS record type that consists of a code and a set of time slots.

Frame: TYPE = [# code: Code, time: Time(code) #]

Code is already defined as an uninterpreted type and *Time* type is defined below.

Time(k:Code): TYPE = {id:ID|EXISTS(s:Sat): id = s`id AND EXISTS (c:Cluster):
c`code = k AND member(s,c`sats)}

Now, we will capture different notions of the system using PVS functions and predicates. We define a PVS function distance that takes two satellites *S1* and *S2*, and a given time *t* as an input and returns the distance between them as a floating point output.

distance(t:Timestamp,S1:Sat,S2:Sat): real

Before defining this function, *Timestamp* is declared as *Timestamp: TYPE = nat*. Since we will define a function that is recursive on *Timestamp* later on (in Section 5.1.2), it need to be a non-dense numeric type, for example *nat* instead of *real*.

The power of the satellites (especially that of the cluster heads) should be checked periodically.

power(t:Timestamp,s:Sat): real

If any of the master satellites does not have enough power for transmission, a new cluster head will be chosen based on centrality algorithm of network analysis [1,40]. We define a PVS function *closestToAll* that returns the central satellite of a cluster

```
closestToAll(c:Cluster,t:Timestamp): Sat
```

A PVS predicate encodes whether a satellite *s* is closest to all other satellites in a cluster *c* or not at a given time *t*.

```
closestToAll?(c:Cluster,t:Timestamp,s:Sat):bool = closestToAll(c,t) = s
```

To receive or decode the transmitted signal by a satellite, the code attached in the frame and the code of the cluster in which the satellite belongs must match. The destination cluster can be found as

```
clusterDst(f:Frame|EXISTS (c:Cluster): c`code = f`code): Cluster
```

We put these definitions together in a file, *basic_defs*.

5.1.2 Definition of ISC using TDMA-Centric Hybrid Protocol

In this section, we encode the inter-satellite communication specification for a small satellite network using PVS. We follow the TDMA-centric approach. We start by defining a PVS theory called *tdma_centric*, importing *basic_defs*, and declaring variables.

```
tdma_centric: THEORY
   BEGIN
   IMPORTING basic_defs
   c: VAR Cluster
   s,Ssrc,Sdst: VAR Sat
   t: VAR Timestamp
   f: VAR Frame
   ...
```

We know that each satellite in the network has a unique id (i.e. accessor of the *Sat* record type). We encode this idea as a PVS axiom.

```
sat_id: AXIOM FORALL (s1:Sat,s2:Sat): s1`id = s2`id IMPLIES s1=s2
```

Member satellites within a cluster communicate directly if they are within transmission range. A PVS predicate *inRange?()* checks whether two satellites (source, *Ssrc*, and destination, *Sdst*) in the same cluster *c* are in a transmission range or not at a specific time *t*. *Thdst* is the maximum distance between two small satellites for them to be within transmission range and is defined in *basic_defs*, Section 5.1.1.

```
inRange?(c,t,Ssrc,Sdst): bool= distance(t,Ssrc,Sdst) <= Thdst
```

Hence, a PVS predicate to represent the notion of direct communication between a source *Ssrc* and destination *Sdst* small satellites in a cluster *c* at time *t* will be

```
canDirectTransmit?(c,t,Ssrc,Sdst): bool= inRange?(c,t,Ssrc,Sdst)
```

The master satellite should always satisfy the minimum power requirement and will be replaced if its power is less than a threshold, *Thpow*. Choosing a new cluster head is based on the centrality algorithm. We capture this notion using PVS as follows.

```
head?(c,t)(s):RECURSIVE bool =
   IF t>to  THEN
      IF head?(c,t-1)(s) THEN
          power(t,s) >= Thpow
     ELSE
          IF power(t,epsilon) < Thpow THEN
              closestToAll?(c,t,s)
          ELSE
             FALSE
          ENDIF
     ENDIF
  ELSE
     closestToAll?(c,t,s)
  ENDIF
MEASURE t
```

Note on *epsilon* and *RECURSIVE* keywords:

- Given a predicate over the type *t*, *epsilon* produces an element of satisfying that predicate if one exists, and otherwise produces an arbitrary element of that type [41].
- Recursive definitions (using *RECURSIVE* keyword) are treated as constant declarations, except that the defining expression is required, and a measure (followed by *MEASURE* keyword) must be provided [21]. The *timestamp t* is the measure.

A satellite receives a signal if the code of the cluster it belongs to is the same as (can be decoded) the code attached to the frame and also the time slots match; otherwise, the signal will be forwarded to the next cluster. A PVS predicate can be written as

```
forwardsTo?(f,s): bool = s`id=f`time AND EXISTS (c:Cluster):
                  f`code=c`code AND member(s,c`sats)
```

We again put these together in a file to complete *tdma_centric* theory.

5.1.3 Theorem

In this section, we present PVS code to formally represent a property (or requirement) that we will be verifying the ISC specification against. The requirement:

"*Any transmitted signal is uniquely decoded in the network only by the intended user(s)*."

can be represented as

```
uniqueness: THEOREM FORALL (s1,s2:Sat,f:Frame): forwardsTo?(f,s1) AND
                                        forwardsTo?(f,s2) IMPLIES s1=s2
```

Similarly, the following requirements can also be represented using PVS and verified.

- Any satellite in the network is uniquely identified.
- A cluster in the network always has one head.

- Using TDMA-centric communication architecture, communication between any two satellites in the network is always possible (communication coverage).

5.2 Formal Verification

5.2.1 Well-formedness

PVS requires theorem proving in order to guarantee that the specification is type correct [16]. Type checking the *basic_defs* theory generates existence TCCs. Any TCC should be discharged if we are to guarantee the correctness of (no contradiction, nor false assumptions) the specification. We modify the specification by changing *Sat, ID,* and *Code* types to be non-empty (there exists at least one), *TYPE+*. In fact, that is the case in our system, i.e., the satellite (+ its id) and DS code are a non-empty set of objects (actually, several of them). Type checking the theory (after these modifications) generates no TCCs i.e. the theory is type correct. We then type check the second theory i.e. *tdma_centric*. The TCCs generated are proved automatically by PVS standard strategy (*tcc*).

At this point, all the TCCs are discharged and hence the theories are well-formed; i.e., there is no contradiction (nor false assumption) in the declaration. But we do not know yet that it satisfies any given properties (or requirements). We show this in Section 5.2.2.

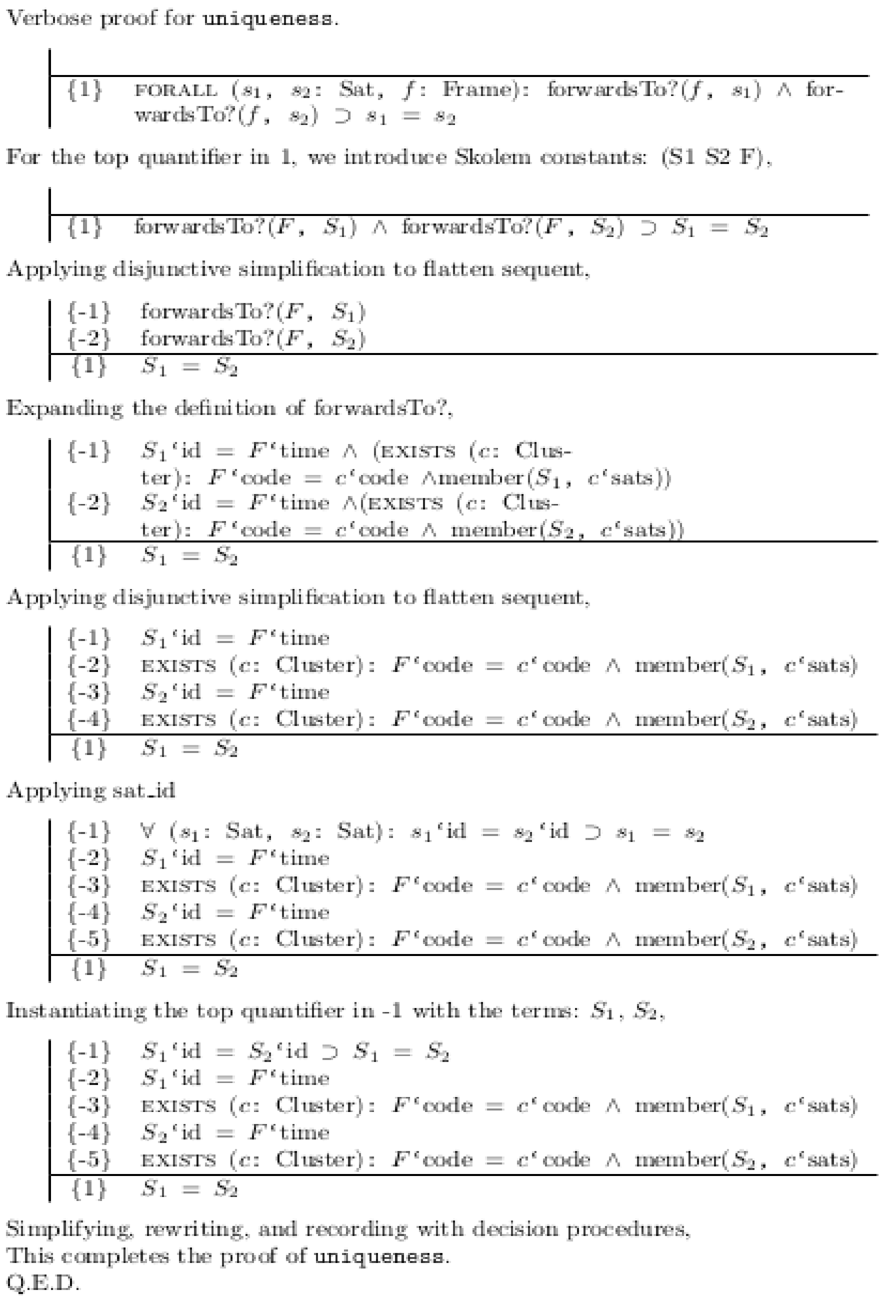
5.2.2 Requirement verification

In order to show that our specification satisfies specific properties, we prove the *uniqueness* theorem (see section 5.1.3). We use the following prover commands (with the order as listed): (*skolem 1 ("S1" "S2" "F")*) (replace universal quantification with constants), (*flatten*) (disjunctive simplification), (*expand "forwardsTo?"*) (expand definition), (*flatten*) (disjunctive simplification again), (*lemma "sat_id"*) (use axiom *sat_id*), and (*inst -1 "S1" "S2"*) (instantiation). Finally, we use an (*assert*) command that invokes the PVS decision procedures to analyze the sequent. This completes the proof of *uniqueness* theorem. Q.E.D. We include the PVS proof in the appendix.

6 CONCLUSION

In this paper, a formal specification and verification for an inter-satellite communication system is presented. It is a continuation of our work on applications of the Responsive and Formal Design process. A TDMA-centric inter-satellite communication specification is represented and verified using PVS. A theorem proving approach was followed in order to check well-formedness (no contradiction) of the specification and verify the requirements. This work can be used to verify requirements for a communication network that uses TDMA-centric architecture and to verify requirements for similar architectures with some modifications. To completely represent the specification of the whole communication system, the given formalization should be refined (add more details) and formal specifications for the rest of the OSI layers of the communication network should be developed.

7 APPENDIX - PVS PROOF



Verbose proof for uniqueness.

{1} FORALL (s_1, s_2: Sat, f: Frame): forwardsTo?(f, s_1) $\wedge$ forwardsTo?(f, s_2) $\supset s_1 = s_2$

For the top quantifier in 1, we introduce Skolem constants: (S1 S2 F),

---

{1} forwardsTo?(F, S_1) $\wedge$ forwardsTo?(F, S_2) $\supset S_1 = S_2$

Applying disjunctive simplification to flatten sequent,

{-1} forwardsTo?(F, S_1)
{-2} forwardsTo?(F, S_2)

{1} $S_1 = S_2$

Expanding the definition of forwardsTo?,

{-1} S_1‘id $=$ F‘time $\wedge$ (EXISTS (c: Cluster): F‘code $=$ c‘code $\wedge$member(S_1, c‘sats))
{-2} S_2‘id $=$ F‘time $\wedge$(EXISTS (c: Cluster): F‘code $=$ c‘code $\wedge$ member(S_2, c‘sats))

{1} $S_1 = S_2$

Applying disjunctive simplification to flatten sequent,

{-1} S_1‘id $=$ F‘time
{-2} EXISTS (c: Cluster): F‘code $=$ c‘code $\wedge$ member(S_1, c‘sats)
{-3} S_2‘id $=$ F‘time
{-4} EXISTS (c: Cluster): F‘code $=$ c‘code $\wedge$ member(S_2, c‘sats)

{1} $S_1 = S_2$

Applying sat_id

{-1} $\forall$ (s_1: Sat, s_2: Sat): s_1‘id $=$ s_2‘id $\supset s_1 = s_2$
{-2} S_1‘id $=$ F‘time
{-3} EXISTS (c: Cluster): F‘code $=$ c‘code $\wedge$ member(S_1, c‘sats)
{-4} S_2‘id $=$ F‘time
{-5} EXISTS (c: Cluster): F‘code $=$ c‘code $\wedge$ member(S_2, c‘sats)

{1} $S_1 = S_2$

Instantiating the top quantifier in -1 with the terms: S_1, S_2,

{-1} S_1‘id $=$ S_2‘id $\supset S_1 = S_2$
{-2} S_1‘id $=$ F‘time
{-3} EXISTS (c: Cluster): F‘code $=$ c‘code $\wedge$ member(S_1, c‘sats)
{-4} S_2‘id $=$ F‘time
{-5} EXISTS (c: Cluster): F‘code $=$ c‘code $\wedge$ member(S_2, c‘sats)

{1} $S_1 = S_2$

Simplifying, rewriting, and recording with decision procedures,
This completes the proof of uniqueness.
Q.E.D.

8 REFERENCES

[1] Radhakrishnan R., Edmonson W., Afghah F., Chenou J., Martinez Rodriguez-Osorio R., and Zhen Q., *Optimal multiple access protocol for inter-satellite communication in small satellite systems*, in 4S Small Satellite Systems and Services Symposium, 2014.

[2] Young R. R., *The Requirements Engineering Handbook.* Boston: Artech House, Inc, 2004.

[3] Edmonson W., Gebreyohannes S., Dillion A., Radhakrishnan R., Chenou J., Esterline A., and Afghah F., *Systems engineering of intersatellite communications for distributed systems of small satellites*, 9th Annual IEEE International Systems Conference (SysCon), pp. 705–710, April 2015.

[4] Edmonson W., Herencia-Zapana H., Neogi N., Moore W., and Ferguson S., *Highly confident reduced life-cycle design process for small satellite systems: Methodology and theory*, Complex Systems and Data Management Conference; Paris, France, 2012.

[5] Edmonson W., Chenou J., Neogi N., and Herencia-Zapana H., *Small satellite systems design methodology: A formal and agile design process*, 8th Annual IEEE International Systems Conference, pp. 518–524, 2014.

[6] *INCOSE systems engineering vision 2020*, 2007. [Online]. Available: http://www.incose.org/ProductsPubs/products/sevision2020.aspx.

[7] Wing J., *A specifiers introduction to formal methods*, IEEE Computer, vol. 23, no. 9, p. 824, 1990.

[8] Bhasin K. and Hayden J. L., *Space Internet architectures and technologies for NASA enterprises,* IEEE Proceedings in Aerospace Conference*,* Vol. 2, pp. 2-931, 2001.

[9] Owre S., Rushby J., and Shankar N., *PVS: A prototype verification system*, in Proceeding of the 11th International Conference on Automated Deduction, ser. Lecture Notes in Artificial Intelligence, D. Kapur, Ed., vol. 607. Springer, June 1992, pp. 748–752.

[10] Owre S., Shankar N., Rushby J., and Stringer-Calvert D., *PVS system guide*, SRI International. Computer Science Laboratory, vol. Version 2.4, 2001.

[11] Daoud H., Tanougast C., Belarbi M., Heil M., *Formal specification and verification of wireless networked self-organized systems on chip*, International Conference on Decision and Information Technologies (CoDIT), pp. 730–735, 2014.

[12] Ili´c D., Troubitsyna E., Laibinis L., and Leppanen S., *Formal verification of consistency in model-driven development of distributed communicating systems and communication protocols*, Second International Symposium on Leveraging Applications of Formal Methods,Verification, and Validation, pp. 425–432, 2007.

[13] Brzeziniski K., *Formalizing operator requirements for the development of telecommunications networks and services*, 8th International Conference on Telecommunications - ConTEL, Zagreb, Croatia, pp. 15–17, June 2005.

[14] Cecconi M. and Tronci E., *Requirements formalization and validation for a telecommunication equipment protection switcher*, Fifth IEEE International Symposim on High Assurance Systems Engineering (HASE), pp. 169–176, 2000.

[15] Yongxiang C. and Zhongyu H., *Research on the formalization description of user requirement in GIS demand-based services*, International Conference on Audio Language and Image Processing (ICALIP), pp. 1595–1599, 2010.

[16] Butler R., *An elementary tutorial on formal specification and verification using PVS*, NASA Technical Memorandum 108991, September 1993.

[17] Butler R., *An introduction to requirements capture using PVS: Specification of a simple autopilot*, IEEE Transactions on Software Engineering, In NASA Technical Memorandum 110255, vol. 24, May 1996.

[18] Crow J. and Di Vito B. L., *Formalizing space shuttle software requirements*, To be presented at the ACM SIGSOFT Workshop on Formal Methods in Software Practice, San Diego, CA, January 1996.

[19] Di Vito B. L., *Formalizing new navigation requirements for NASA's space shuttle*, in To appear in proceedings of Formal Methods Europe (FME'96), Oxford, England, March 1996.

[20] *PVS specification and verification system*, 2011. [Online]. Available: http://www.pvs.csl.sri.com SRI. [Sep. 15, 2015]

[21] Owre S., Shankar N., Rushby J., and Stringer-Calvert D., *PVS language reference*, SRI International. Computer Science Laboratory, vol. Version 2.4, 2001.

[22] Owre S., Shankar N., Rushby J., and Stringer-Calvert D., *PVS prover guide*, SRI International. Computer Science Laboratory, vol. Version 2.4, 2001.

[23] Edward A. Lee, *Cyber physical systems: Design challenges*, Technical Report No. UCB/EECS-2008-8, January 2008. [Online]. Available:
http://www.eecs.berkeley.edu/Pubs/TechRpts/2008/EECS-2008-8.html

[24] Wertz J. and Larson W., *Space Mission Analysis and Design*. Microcosm Press, 1999.

[25] Rasmussen J., *A framework for cognitive task analysis in systems design*, Riso National Laboratory, Denmark, 1985.

[26] Llorente J., et al. *PROBA-3: Precise Formation Flying Demonstration Mission*, Acta Astronautica, vol. 82, no. 1, pp. 38–46, Jan 2013.

[27] Gill E., Sundaramoorthy P., Bouwmeester J., Zandbergen B., and Reinhard R., *Formation Flying within a Constellation of Nanosatellites: The QB50 Mission*, Acta Astronautica, vol. 82, no. 1, pp. 110–117, Jan 2013.

[28] Kusza K. L. and Paluszek M. A., Intersatellite *Links: Lower Layer Protocols for Autonomous Constellations*, in First Joint Space Internet Workshop, NASA Goddard Space Flight Center, 2000.

[29] Cockrell J., Alena R., Mayer D., Sanchez H., Luzod T., Yost B., and Klumpar D., *EDSN:A Large Swarm of Advanced Yet Very Affordable, COTS-based NanoSats that Enable Multipoint Physics and Open Source Apps*, in 26th Annual AIAA/USU Conference on Small Satellites, 2012.

[30] Arslan T., et al. *ESPACENET: A Framework of Evolvable and Reconfigurable Sensor Networks for Aerospace? Based Monitoring and Diagnostics*, in Proceedings of the First NASA/ESA Conference on Adaptive Hardware and Systems (AHS'06), 2006.

[31] NASA Goddard Space Flight Center. [Online]. Available: http://attic.gsfc.nasa.gov/ants/, [Sep. 15, 2015].

[32] Bedon H., et al., *Preliminary Internetworking Simulation of the QB50 Cubesat Constellation, Communications (LATINCOM)*, in IEEE Latin-American Conference, 2010.

[33] Radhakishnan R., Zeng Q. A., and Edmonson W. W., *Inter-satellite Communications for Small Satellite Systems*, International Journal of Interdisciplinary Telecommunications and Networking, vol. 5, no. 3, pp. 11–24, 2013.

[34] Chen B. and Yu L., *Design and Implementation of LDMA for Low Earth Orbit Satellite Formation Network, Embedded and Ubiquitous Computing (EUC)*, in IFIP 9th International Conference, 2011.

[35] Sun R., Guo J., Gill E., and Maessen D., *Potential and Limitations of CDMA Networks for Combined Inter-satellite Communication and Relative Navigation*, International Journal on Advances in Telecommunications, vol. 5, no. 1 and 2, pp. 21–32, 2012.

[36] Sidibeh K. and Vladimirova T., *Communication in LEO Satellite Formations, Adaptive Hardware and Systems*, in AHS '08, NASA/ESA Conference, 2008.

[37] Heidari G. and Truong H., *Efficient, Flexible, Scalable Inter-satellite Networking*, in Wireless for Space and Extreme Environments (WiSEE), IEEE International Conference, Nov 2013, pp. 1–6.

[38] Crow J., Owre S., Rushby J., Shankar N., and Srivas M., *A tutorial introduction to PVS*, Workshop on Industrial-Strength Formal Specification Techniques, Boca Raton, Florida, April, Updated June 1995. [Online]. Available: WWW: http://www.csl.sri.com/sri-cslfm.html

[39] Rushby J., Owre S., and Shankar, N. *Subtypes for specifications: Predicate subtyping in PVS*, IEEE Transactions on Software Engineering, vol. 24, no. 9, pp. 709 – 720, 1998.

[40] Borgatti S.P. and Everett M.G., *A Graph-Theoretic Perspective on Centrality*, Social Networks (Elsevier) 28, 466–484, 2005.

[41] Owre S. and Shankar N., *PVS prelude library*, CSL Technical Report SRI-CSL-03-01, SRI International. Computer Science Laboratory, March 2003.