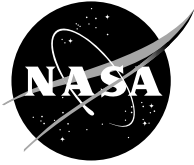


NASA/TP—2008-214813



# Space Telecommunications Radio System Software Architecture Concepts and Analysis

*Louis M. Handler*  
*Glenn Research Center, Cleveland, Ohio*

*Charles S. Hall*  
*Analex Corporation, Brook Park, Ohio*

*Janette C. Briones and Tammy M. Blaser*  
*Glenn Research Center, Cleveland, Ohio*

## NASA STI Program . . . in Profile

Since its founding, NASA has been dedicated to the advancement of aeronautics and space science. The NASA Scientific and Technical Information (STI) program plays a key part in helping NASA maintain this important role.

The NASA STI Program operates under the auspices of the Agency Chief Information Officer. It collects, organizes, provides for archiving, and disseminates NASA's STI. The NASA STI program provides access to the NASA Aeronautics and Space Database and its public interface, the NASA Technical Reports Server, thus providing one of the largest collections of aeronautical and space science STI in the world. Results are published in both non-NASA channels and by NASA in the NASA STI Report Series, which includes the following report types:

- **TECHNICAL PUBLICATION.** Reports of completed research or a major significant phase of research that present the results of NASA programs and include extensive data or theoretical analysis. Includes compilations of significant scientific and technical data and information deemed to be of continuing reference value. NASA counterpart of peer-reviewed formal professional papers but has less stringent limitations on manuscript length and extent of graphic presentations.
- **TECHNICAL MEMORANDUM.** Scientific and technical findings that are preliminary or of specialized interest, e.g., quick release reports, working papers, and bibliographies that contain minimal annotation. Does not contain extensive analysis.
- **CONTRACTOR REPORT.** Scientific and technical findings by NASA-sponsored contractors and grantees.

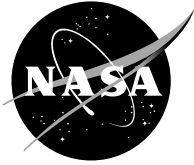
- **CONFERENCE PUBLICATION.** Collected papers from scientific and technical conferences, symposia, seminars, or other meetings sponsored or cosponsored by NASA.
- **SPECIAL PUBLICATION.** Scientific, technical, or historical information from NASA programs, projects, and missions, often concerned with subjects having substantial public interest.
- **TECHNICAL TRANSLATION.** English-language translations of foreign scientific and technical material pertinent to NASA's mission.

Specialized services also include creating custom thesauri, building customized databases, organizing and publishing research results.

For more information about the NASA STI program, see the following:

- Access the NASA STI program home page at <http://www.sti.nasa.gov>
- E-mail your question via the Internet to [help@sti.nasa.gov](mailto:help@sti.nasa.gov)
- Fax your question to the NASA STI Help Desk at 301-621-0134
- Telephone the NASA STI Help Desk at 301-621-0390
- Write to:  
NASA Center for AeroSpace Information (CASI)  
7115 Standard Drive  
Hanover, MD 21076-1320

NASA/TP—2008-214813



# Space Telecommunications Radio System Software Architecture Concepts and Analysis

*Louis M. Handler*  
*Glenn Research Center, Cleveland, Ohio*

*Charles S. Hall*  
*Analex Corporation, Brook Park, Ohio*

*Janette C. Briones and Tammy M. Blaser*  
*Glenn Research Center, Cleveland, Ohio*

National Aeronautics and  
Space Administration

Glenn Research Center  
Cleveland, Ohio 44135

---

June 2008

*Level of Review:* This material has been technically reviewed by an expert single reviewer.

Available from

NASA Center for Aerospace Information  
7115 Standard Drive  
Hanover, MD 21076-1320

National Technical Information Service  
5285 Port Royal Road  
Springfield, VA 22161

Available electronically at <http://gltrs.grc.nasa.gov>

## Contents

List of Figures .....	iv
Summary .....	1
1.0 Introduction .....	1
2.0 Documents .....	3
3.0 Use Case Process .....	4
Appendix A—Key Terms in Use Cases .....	7
Appendix B—Acronyms .....	9
Appendix C—State Diagrams .....	11
C.1 Radio State Diagram .....	11
C.2 Waveform State Diagram .....	12
Appendix D—STRS Use Cases .....	13
D.1 Power On .....	13
D.2 Power Off .....	15
D.3 Waveform Upload .....	17
D.4 STRS OE Upload .....	19
D.5 Waveform Instantiation .....	21
D.6 Waveform Start .....	23
D.7 Processor Resource Sharing .....	24
D.8 Set Waveform Parameter .....	27
D.9 Get Waveform Parameter .....	29
D.10 Transmit a Packet .....	31
D.11 Receive a Packet .....	32
D.12 Waveform Stop .....	34
D.13 Waveform Deallocate .....	37
D.14 Waveform Abort .....	39
D.15 Waveform Remove .....	41
D.16 Fault Management .....	43
D.17 Commanded Built-In Test .....	45



# Space Telecommunications Radio System Software Architecture Concepts and Analysis

Louis M. Handler  
National Aeronautics and Space Administration  
Glenn Research Center  
Cleveland, Ohio 44135

Charles S. Hall  
Analex Corporation  
Brook Park, Ohio 44142

Janette C. Briones and Tammy M. Blaser  
National Aeronautics and Space Administration  
Glenn Research Center  
Cleveland, Ohio 44135

## Summary

The Space Telecommunications Radio System (STRS) project investigated various software-defined radio (SDR) architectures for space. An STRS architecture has been selected that separates the STRS operating environment from its various waveforms and also abstracts any specialized hardware to limit its effect on the operating environment. The design supports software evolution where new functionality is incorporated into the radio. Radio hardware functionality has been moving from hardware-based application specific integrated circuits (ASICs) into firmware- and software-based processors such as field programmable gate arrays (FPGAs), digital signal processors, (DSPs) and general purpose processors (GPPs).

Use cases capture the requirements of a system by describing how the system should interact with the users or other systems (the actors) to achieve a specific goal. The unified modeling language (UML) is used to illustrate the use cases in a variety of ways. The top-level use case diagram shows groupings of the use cases and how the actors are involved. The state diagrams depict the various states that a system or object may be in and the transitions between those states. The sequence diagrams show the main flow of activity as described in the use cases.

## 1.0 Introduction

The Space Telecommunications Radio System (STRS) project investigated various software-defined radio (SDR) architectures suitable for NASA's missions, which are subject to the constraints of the space environment. The STRS architecture (fig. 1) separates the STRS operating environment (OE) from its various waveform applications and also abstracts any specialized hardware to limit its effect on the operating environment. The STRS OE enables the startup, operation, and teardown of the waveform applications. The waveform applications are implemented in firmware and software to transform information to or from signals that are transmitted over the air. Separating functionality within the radio promotes waveform portability and reuse.

Use cases capture the requirements of a system by describing how the system should interact with the users or other systems (the actors) to achieve a specific goal. The unified modeling language (UML) is used to illustrate the use cases in a variety of ways. The top-level use case diagram shows groupings of the use cases and how the actors are involved. The use case actors are distinguished by whether they are external or internal to the spacecraft and whether they are external to the radio in question. The actors' categories determine their allowed roles in interacting with the radio.

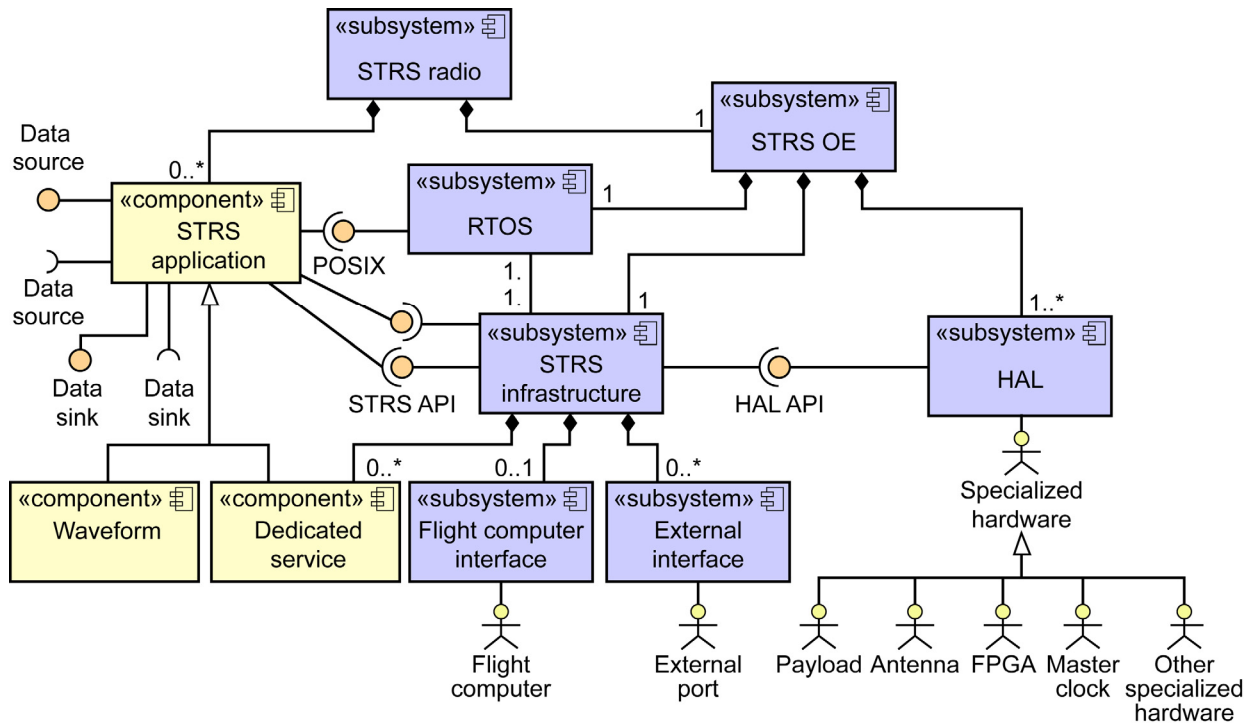


Figure 1.—STRS radio architecture.

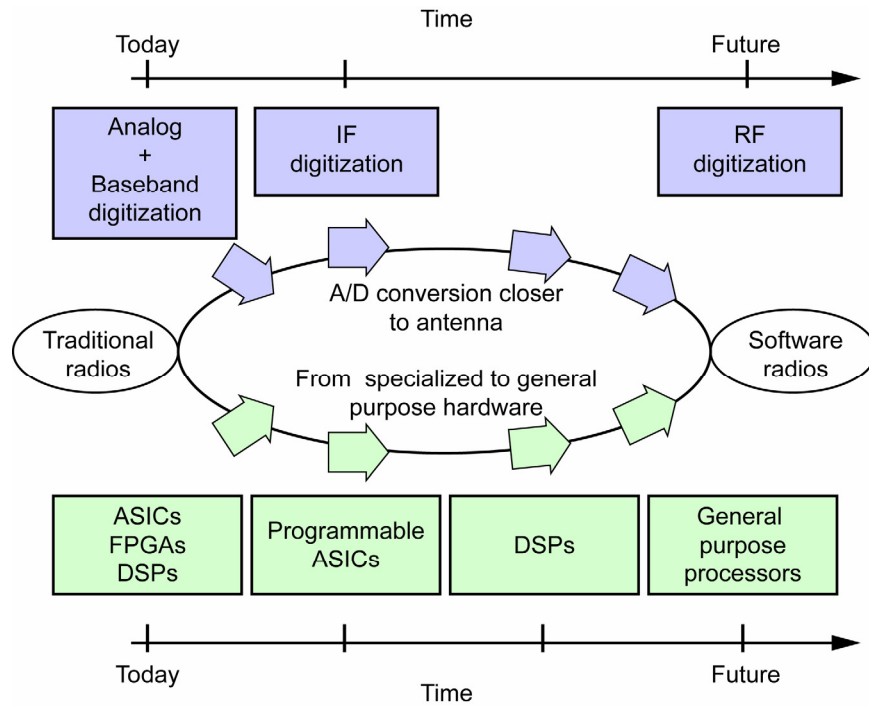


Figure 2.—Radio evolution.

The design expects architecture evolution for each generation of radios. For example, legacy radios have been largely hardware-based, residing primarily in application specific integrated circuits (ASICs). Figure 2 shows how radios evolve due to advances in hardware technology, where the functionality of radios have moved and will move into firmware and software-based processors such as field programmable gate arrays (FPGAs), digital signal processors DSPs and general purpose processors (GPPs).



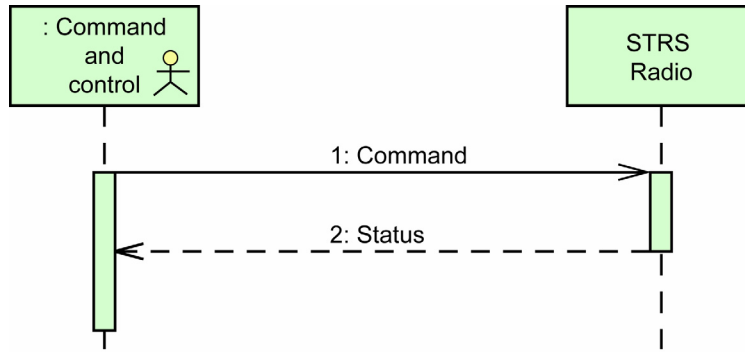


Figure 3.—Generic sequence diagram.

The state machine diagrams, also known as state-transition diagrams or even simply state diagrams, depict the various states in which a system or object may be and the transitions between those states. The sequence diagrams (fig. 3) show the main flow of activity as described in the use cases. The sequence diagrams model the dynamic behavior and are helpful to document and validate the logic.

The processing capability of the SDR provides the opportunity for the radio to implement functionality previously limited to the flight computer. The flight computer is assumed to be the only entity that can physically turn the power on and off for the radio. The flight computer contains a watchdog timer to monitor the health of the radio and cycles power if necessary. The flight computer can send commands to the radio. The radio will have a command and control subsystem to validate commands and translate them to appropriate method calls.

The design supports software evolution where new functionality is incorporated into the radio. New or modified waveforms may be uploaded as required. Also, new parameters may be set for a waveform. When a waveform function becomes a required part of the radio, the waveform is transitioned to a service. Waveforms become services by being configured as part of the infrastructure. To make a waveform into a service, merely include it in the configuration file used at power on. The configuration file may be changed by uploading it as an STRS OE component. Examples of likely functions, eventually to become part of the STRS infrastructure, are global positioning system (GPS) or ranging, navigation, and dynamic discovery service.

## 2.0 Documents

The requirements of the STRS Reference Implementation design were obtained from the Concepts and Functions Review for the STRS, Document No.: 99-P54849D Revision A, also known as CFD (Concepts and Functions Document). The CFD was developed by General Dynamics under contract with NASA Glenn Research Center.

Appendix B of this document contains the use cases and high-level sequence diagrams for the STRS architecture and includes traceability to the parent CFD requirements. POSIX standards are defined in the following two documents:

- (1) ISO/IEC 9945:2003 (IEEE Std 1003.1,2003 Edition) POSIX
- (2) IEEE Std 1003.13–2003 Standardized Application Environment Profile (AEP)—POSIX Realtime and Embedded Application Support  
<http://ieeexplore.ieee.org/iel5/9307/29578/01342418.pdf?tp=&isnumber=29578&arnumber=1342418>

The key terms defined in appendix B were obtained from the STRS Taxonomy Document, TBD.

### 3.0 Use Case Process

A use case is a set of scenarios that capture the different ways that external users interact with the system to achieve a specific goal. Use cases are textual and are in the following format:

Use Case Identifier: Example Use Case				
<b>Description:</b> A description of the use case goes here.				
<b>External actors:</b> External actor(s) involved with this specific use case.				
<b>Related use cases:</b> Use cases that are related to this use case by interaction or similarity.				
<b>Precondition:</b> Any precondition that must exist before this use case can occur. Usually the initial RADIO state and WAVEFORM state are listed. <b>Triggering event:</b> Event that triggers use case. Not all use cases have triggering events				
<b>Main Flow:</b> <ol style="list-style-type: none"> <li>1. This will be an ordered list of steps necessary to perform interaction. This is the nominal flow. Alternate flows will be listed below.</li> <li>2. Step 2</li> <li>3. Step 3</li> <li>4. Step 4</li> <li>5. etc.</li> </ol>				
<b>Resulting event:</b> If completion of use case results in an event, it is listed here. Usually the resulting RADIO state and WAVEFORM state are listed. <b>Post condition:</b> Description of the results of the use case interaction. (This is the post condition from the nominal flow.)				
<b>Alternatives:</b> (1a) This is where alternate flows are identified. The alternative will be identified by the number of the main flow where the branch occurs followed by a letter a to z. (3a) This is an alternative flow for step 3. (3b) This is an additional alternative flow for step 3. (7a–8a) This is an alternative flow for a range of steps from the main flow.				
<b>CFD Requirements:</b> A list of CFD requirements related to this use case.				
<table border="1" style="width: 100%;"> <thead> <tr> <th style="width: 50%;">Requirement Name</th> <th style="width: 50%;"></th> </tr> </thead> <tbody> <tr> <td>Requirement Description</td> <td></td> </tr> </tbody> </table>	Requirement Name		Requirement Description	
Requirement Name				
Requirement Description				
<b>Comments:</b> Comments on use case.				

Use case diagrams provide a graphical representation of the use cases and how they are related to external actors and other use cases.

Figure 4 shows the high-level use case diagram for the STRS radio. Each oval in the diagram represents a general kind of interaction between the system and its users. Each of the high-level use cases shown in the figure are composed of more detailed use cases that describe the interactions at a finer level of detail. This high-level use case only lists the key interactions to give a general overview. Refer to the actual use cases starting in 0 for details.

The users are shown as stick figures both inside and outside the spacecraft system boundary. The figure shows three general types of user, “Over the Air,” “STRS Command and Control,” and “Payload”. The “Ground Station” and “Other Space Vehicle” actors are a type of Over the Air user. Likewise “External Port” and “Flight Computer” are types of STRS Command and Control user. This generalization is shown on the diagram with the hollow closed triangular arrow pointing from the specific actor to the general actor.

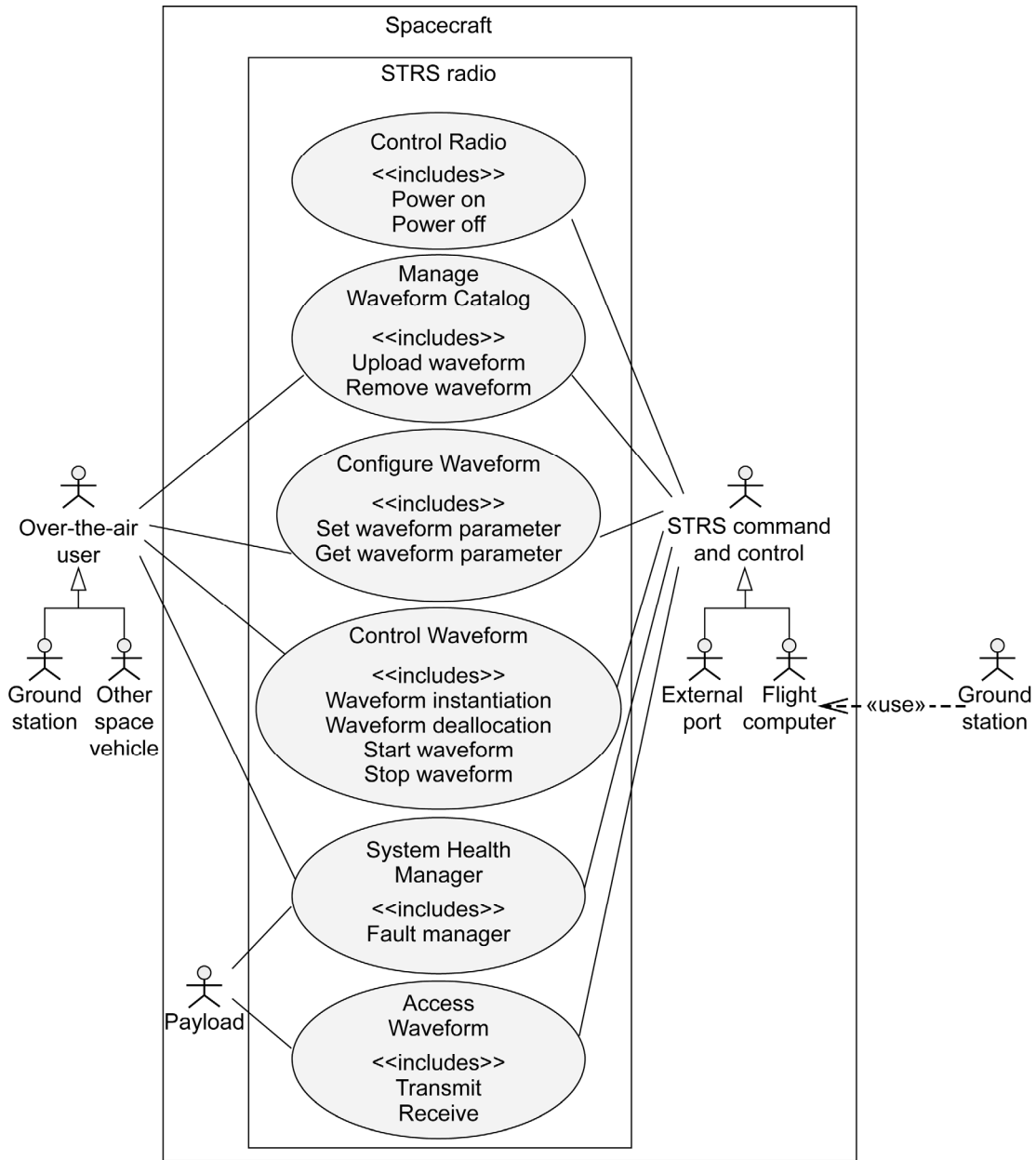


Figure 4.—Top-level use case diagram.

The Over the Air user is communicating with or controlling the radio by means of a communication channel realized by the STRS radio itself. The STRS Command and Control users are communicating with the radio by an onboard interface. Note that a Ground Station may configure a waveform over the air directly or could configure a waveform indirectly by sending the command to the onboard flight computer by means of another radio. This is why ground station is shown twice. From the perspective of the radio the indirect command would be coming from the flight computer. The interaction of the ground station and the flight computer shown on the right would be read as the ground station uses the flight computer to command the STRS radio.

Some use cases are followed by a high-level sequence diagram. A sequence diagram is a UML diagram that shows the dynamic behavior of the system. For example, see figure 3. The boxes at the top represent classes, subsystems, or actors that interact. The vertical lines represent life lines of the interacting entities with time progressing downward on the diagram. The horizontal lines represent

interactions between the participants. The sequence diagrams in this document show interactions at a very high level. Design-level sequence diagrams will be created later to illustrate the lower-level interactions of the system.

Glenn Research Center  
National Aeronautics and Space Administration  
Cleveland, Ohio, June, 2008

## Appendix A—Key Terms in Use Cases

**Application programming interface (API):** a formalized set of software calls and routines that can be referenced by an application program in order to access supporting system or network services.

**Built-in test (BIT):** software used to test whether the STRS radio and all of its subsystems are working properly.

**Data publisher:** part of the system that transmits data to one or more data subscribers.

**Data subscriber:** part of the system that receives data from a data publisher.

**External interface:** consists of spacecraft interfaces as well as other interfaces used to process input and output data of the radio.

**Flight computer:** a separate computer connected to the STRS radio that is used to monitor and control the STRS radio. It may contain the Watchdog timer for the STRS radio.

**General purpose module (GPM):** a hardware module used for general purpose processing that contains the STRS OE.

**Hardware abstraction layer (HAL):** a HAL is defined to specify the communication and integration specification between physical hardware modules. A HAL is the abstraction of control of the specialized hardware. It is the part of the STRS OE that implements the HAL API.

**Over-the-air programming (OTAP):** a method of distributing new software updates with the necessary settings with which to access services.

**Persistent storage:** location in the hardware from which data can be obtained and into which data can be stored. It includes nonvolatile memory, hard drives, flash, etc.

**Portable Operating System Interface (POSIX):** a family of IEEE standards 1003.n, which describes the fundamental operating system services and functions necessary to provide a UNIX-like kernel interface to applications. POSIX is not an operating system but is instead the guaranteed programming interfaces available to the application programmer.

**Quality of Service (QoS):** a networking term that specifies a guaranteed throughput level. (Throughput = the amount of data transferred from one place to another or processed in a specified amount of time measured in Kbps, Mbps, and Gbps, etc.)

**Real-time operating system (RTOS):** an operating system that guarantees a certain capability within a specified time constraint.

**Software-defined radio (SDR):** a collection of hardware and software technologies that enable reconfigurable system architectures for wireless networks and user terminals. SDR provides an efficient and comparatively inexpensive solution to the problem of building multimode, multiband, and multifunctional wireless devices that can be enhanced using software upgrades.

**Specialized hardware:** hardware that may be initialized or controlled using software. STRS specialized hardware is controlled using the HAL, and permits certain algorithms or functions to be optimized in hardware. Specialized hardware can include FPGA, DSP, ASIC, etc.

**Space Telecommunications Radio System (STRS):** the project under which various SDR architectures were investigated.

**STRS command source:** abstracts the command functionality usually found in the interface to the flight computer.

**STRS infrastructure:** the portion of the STRS radio that implements the required functionality. Such functionality includes configuration and control of STRS waveforms as well as the specialized hardware via HAL.

**STRS operating environment (OE):** the portion of the STRS radio that contains the STRS infrastructure, the POSIX conformant RTOS, HAL, and optionally middleware software. Any STRS waveform may only use the POSIX interface and the STRS infrastructure interface to access the required functionality.

**STRS radio:** an SDR designed for the STRS project whose software requirements include portability and open source. It contains waveforms and the STRS OE.

**STRS waveform:** a waveform implemented as an executable software application that uses a standardized interface to the STRS OE to access the required functionality of the hardware and software. Using a standard interface aids portability and reuse.

**Waveform (WF):** the set of transformations applied to information that is transmitted over the air and the corresponding set of transformations to convert received signals back to their information content. A waveform comprises the end-to-end functionality (e.g., modulation, coding, frequency conversion, and filtering).

**Watchdog Timer (WDT):** software and/or hardware that monitors the health of a system and if a problem is detected, takes the appropriate action to restore the system back to health.

## Appendix B—Acronyms

AEP	application environment profile
API	application programming interface
ASIC	application specific integrated circuit
BIT	built-in test
CFD	Concepts and Functions Document
DSP	digital signal processor
FPGA	field programmable gate array
GPM	general purpose module
GPP	general purpose processor
GPS	global positioning system
HAL	hardware abstraction layer
OE	operating environment
OTAP	over-the-air programming
POSIX	Portable Operating System Interface
QoS	Quality of Service
RTOS	real-time operating system
SDR	software-defined radio
STRS	Space Telecommunications Radio System
UML	unified modeling language
WDT	Watchdog timer
WF	waveform





# Appendix C—State Diagrams

## C.1 Radio State Diagram

State	Description
POWERDOWN FAULT	No power to the radio. Must be turned on by the Flight Computer before anything can be done. An error was detected. STRS radio attempts to fix the problem. If the fault cannot be corrected, the radio can try a degraded mode or cycling power.
INITIALIZING READY	Power has been turned on, but the radio is not ready to perform communication tasks. The radio is ready to perform communication tasks.
BUSY	The radio has received a command and is processing that command.
SHUTDOWN	The radio is in the process of shutting down and powering off.

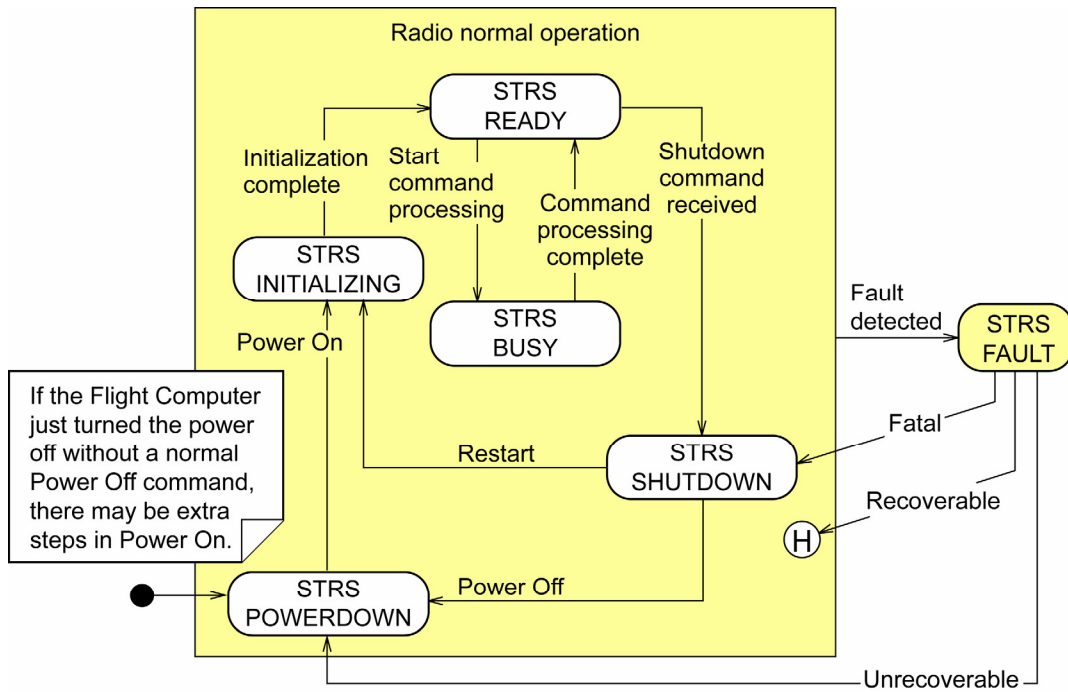


Figure 5.—STRS radio state diagram.

## C.2 Waveform State Diagram

State	Description
EMPTY	Waveform name is not recognized or waveform name is recognized but no components or fragments are present.
INSTANTIATED	Waveform components are loaded on the hardware and execution has begun.
LOADED	Waveform name is recognized and all components are present.
PARTIAL	Waveform name is recognized and some components or fragments are present.
RUNNING	Waveform start command has been received by the executing waveform.
FAULT	An error was detected by this waveform that only affects this waveform.

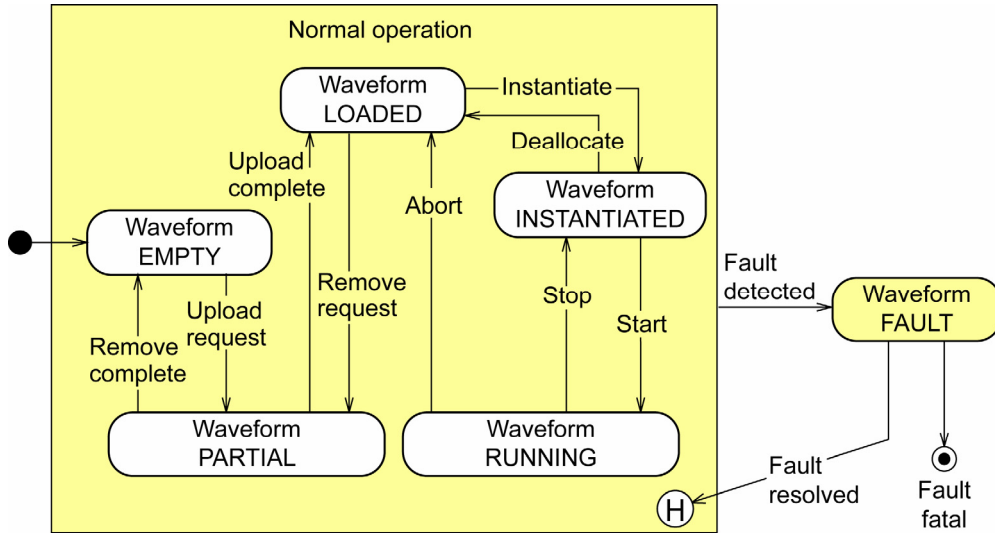


Figure 6.—STRS waveform state diagram.

The upload or removal of a waveform normally happens piecemeal so that the Waveform PARTIAL state indicates that only a portion of the waveform is loaded. Additional pieces may be loaded until upload is complete, and waveform is in the Waveform LOADED state. Additional pieces may be removed until removal is complete and waveform is in the Waveform EMPTY state.

The instantiate command allocates resources and prepares the waveform for execution. The start command begins execution after the waveform is instantiated or stopped. A running waveform can be stopped or aborted. The Specialized Hardware can be initialized during either instantiation or start, as appropriate. Typically, Specialized Hardware is initialized during instantiation then the start command will use the HAL to begin processing.

## Appendix D—STRS Use Cases

This appendix contains the use cases and high-level sequence diagrams for the STRS architecture.

### D.1 Power On

<b>Use Case Identifier: Power On</b>																						
<b>Description:</b> STRS radio is powered on and goes through initialization.																						
<b>External Actors:</b> Flight Computer, (Specialized Hardware via the HAL)																						
<b>Related Use Cases:</b> Waveform Instantiation and Waveform Start.																						
<b>Precondition:</b> STRS radio is powered off (cold Start with no prior conditions) <b>Triggering event:</b> STRS radio is powered on																						
<b>Main Flow:</b> <ol style="list-style-type: none"> <li>1. Boot Code executes low-level BIT on GPM hardware.</li> <li>2. Boot Code initializes and executes the RTOS image.</li> <li>3. RTOS executes BIT.</li> <li>4. RTOS initializes and executes the STRS Infrastructure.</li> <li>5. STRS infrastructure configures the Flight Computer interface.</li> <li>6. STRS infrastructure sends to the Flight Computer STRS INITIALIZING state.</li> <li>7. STRS infrastructure loads and initializes the HAL. During HAL initialization STRS Infrastructure reads Specialized Hardware configuration and installs identified HAL drivers/BSP.</li> <li>8. STRS infrastructure invokes BIT on each Specialized Hardware module via the HAL and writes telemetry to the Flight Computer interface</li> <li>9. STRS infrastructure logs results of all BIT.</li> <li>10. STRS infrastructure reads the STRS Application (waveform or service) configuration file and instantiates and starts STRS Applications as specified. See Waveform Instantiation and Waveform Start Use Case.</li> <li>11. STRS infrastructure sends to the Flight Computer STRS READY state.</li> </ol>																						
<b>Resulting event:</b> Flight Computer notified that the STRS radio is in READY state.																						
<b>Post condition:</b> STRS radio ready to accept commands from Flight Computer. STRS radio is in READY or STRS FAULT state.																						
<b>Alternatives:</b> <ol style="list-style-type: none"> <li>(1a) Low-level BIT identifies recoverable failure. Mark condition in RAM and proceed to 2.</li> <li>(1b) Low-level BIT identifies unrecoverable failure and does not proceed to 2. HW WatchDog Timer resets CPU.</li> <li>(3a) BIT identifies recoverable failure. Mark condition in RAM and proceed to 4.</li> <li>(3b) BIT identifies unrecoverable failure and does not proceed to 4. HW WatchDog Timer resets CPU.</li> <li>(4a) STRS infrastructure does not initialize properly or is corrupted. HW WatchDog Timer resets CPU.</li> <li>(7a–8a) Applicable if Specialized Hardware installed.</li> <li>(11a) If BIT test failed, STRS infrastructure sends to the Flight Computer STRS FAULT state.</li> </ol> <p>If the Flight Computer interface does not initialize or if the hardware has a fault, the flight computer will recognize that the STRS Radio is not communicating and will perform a Power Cycle. If state is permanent, the Flight Computer will Power Off module and attempt to use Redundant System.</p>																						
<b>Requirements:</b> <table border="1" style="width: 100%; border-collapse: collapse; margin-top: 10px;"> <thead> <tr> <th style="text-align: left; padding: 2px;">Requirement ID</th> <th style="text-align: left; padding: 2px;">Name</th> </tr> </thead> <tbody> <tr><td style="padding: 2px;">CUST13.1</td><td style="padding: 2px;">Power On Reset Testing</td></tr> <tr><td style="padding: 2px;">CUST14.1</td><td style="padding: 2px;">Diagnostic Reporting and Action</td></tr> <tr><td style="padding: 2px;">CUST22</td><td style="padding: 2px;">Standard Spacecraft Interfaces</td></tr> <tr><td style="padding: 2px;">SYS5</td><td style="padding: 2px;">Common Services</td></tr> <tr><td style="padding: 2px;">SYS10.1</td><td style="padding: 2px;">Persistent Storage Capacity</td></tr> <tr><td style="padding: 2px;">SYS12</td><td style="padding: 2px;">Device Diagnostic API</td></tr> <tr><td style="padding: 2px;">SYS15</td><td style="padding: 2px;">Power-on Modes</td></tr> <tr><td style="padding: 2px;">SYS15.1</td><td style="padding: 2px;">Default Initialization</td></tr> <tr><td style="padding: 2px;">SYS20</td><td style="padding: 2px;">Error and Fault Logs</td></tr> <tr><td style="padding: 2px;">SYS29</td><td style="padding: 2px;">RF Module Interface</td></tr> </tbody> </table>	Requirement ID	Name	CUST13.1	Power On Reset Testing	CUST14.1	Diagnostic Reporting and Action	CUST22	Standard Spacecraft Interfaces	SYS5	Common Services	SYS10.1	Persistent Storage Capacity	SYS12	Device Diagnostic API	SYS15	Power-on Modes	SYS15.1	Default Initialization	SYS20	Error and Fault Logs	SYS29	RF Module Interface
Requirement ID	Name																					
CUST13.1	Power On Reset Testing																					
CUST14.1	Diagnostic Reporting and Action																					
CUST22	Standard Spacecraft Interfaces																					
SYS5	Common Services																					
SYS10.1	Persistent Storage Capacity																					
SYS12	Device Diagnostic API																					
SYS15	Power-on Modes																					
SYS15.1	Default Initialization																					
SYS20	Error and Fault Logs																					
SYS29	RF Module Interface																					
<b>Comments:</b> All BIT and periodic background monitoring is provided to Flight Computer interface for telemetry.																						

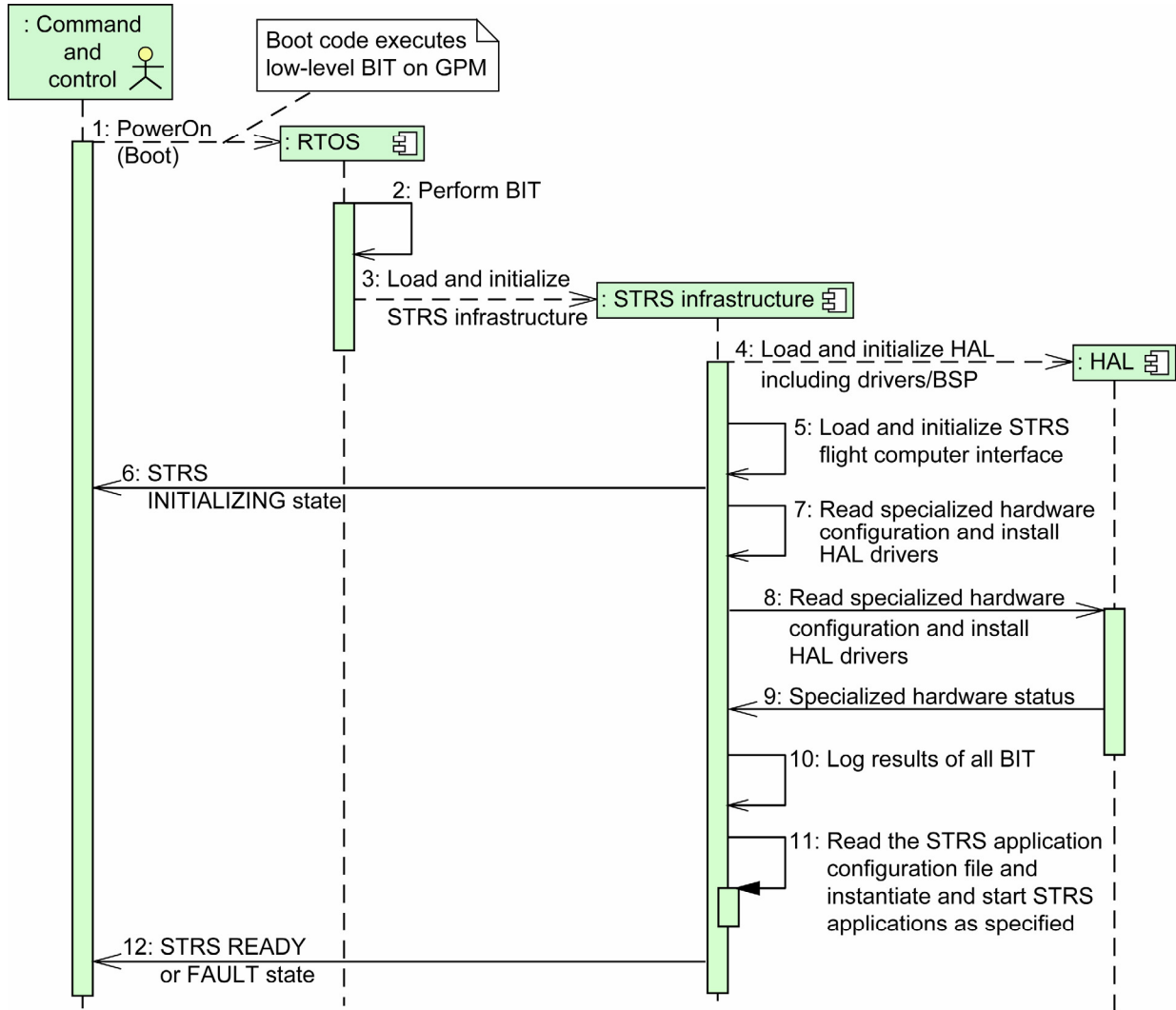


Figure 7.—Power on sequence.

## D.2 Power Off

<b>Use Case Identifier: Power Off</b>							
<b>Description:</b> STRS radio performs orderly shutdown and powers off.							
<b>External Actors:</b> Flight Computer, (Specialized Hardware via the HAL), Waveform							
<b>Related Use Cases:</b> Waveform Stop, Waveform Deallocate, Waveform Abort							
<b>Precondition:</b> STRS radio is in the READY or FAULT state <b>Triggering event:</b> Shutdown command received from STRS Command Source							
<b>Main Flow:</b> <ol style="list-style-type: none"> <li>1. Stop all RUNNING waveforms and services.</li> <li>2. Deallocate all INSTANTIATED waveforms and services.</li> <li>3. STRS infrastructure releases resources associated with specialized hardware.</li> <li>4. STRS infrastructure sends to the Flight Computer STRS SHUTDOWN state.</li> <li>5. STRS infrastructure commands HAL and RTOS to shutdown.</li> </ol>							
<b>Resulting event:</b> Flight Computer notified that the STRS Radio is in SHUTDOWN state. <b>Post condition:</b> STRS radio is powered off.							
<b>Alternatives:</b> (1a) Abort any waveforms and services that got an error in step 1.							
<b>Requirements:</b>							
<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: left; padding: 2px;">Requirement ID</th> <th style="text-align: left; padding: 2px;">Name</th> </tr> </thead> <tbody> <tr> <td style="padding: 2px;">CUST14.2</td> <td style="padding: 2px;">Network Status Monitoring</td> </tr> <tr> <td style="padding: 2px;">SYS21</td> <td style="padding: 2px;">Run-Time State Information</td> </tr> </tbody> </table>	Requirement ID	Name	CUST14.2	Network Status Monitoring	SYS21	Run-Time State Information	
Requirement ID	Name						
CUST14.2	Network Status Monitoring						
SYS21	Run-Time State Information						
<b>Comments:</b>							

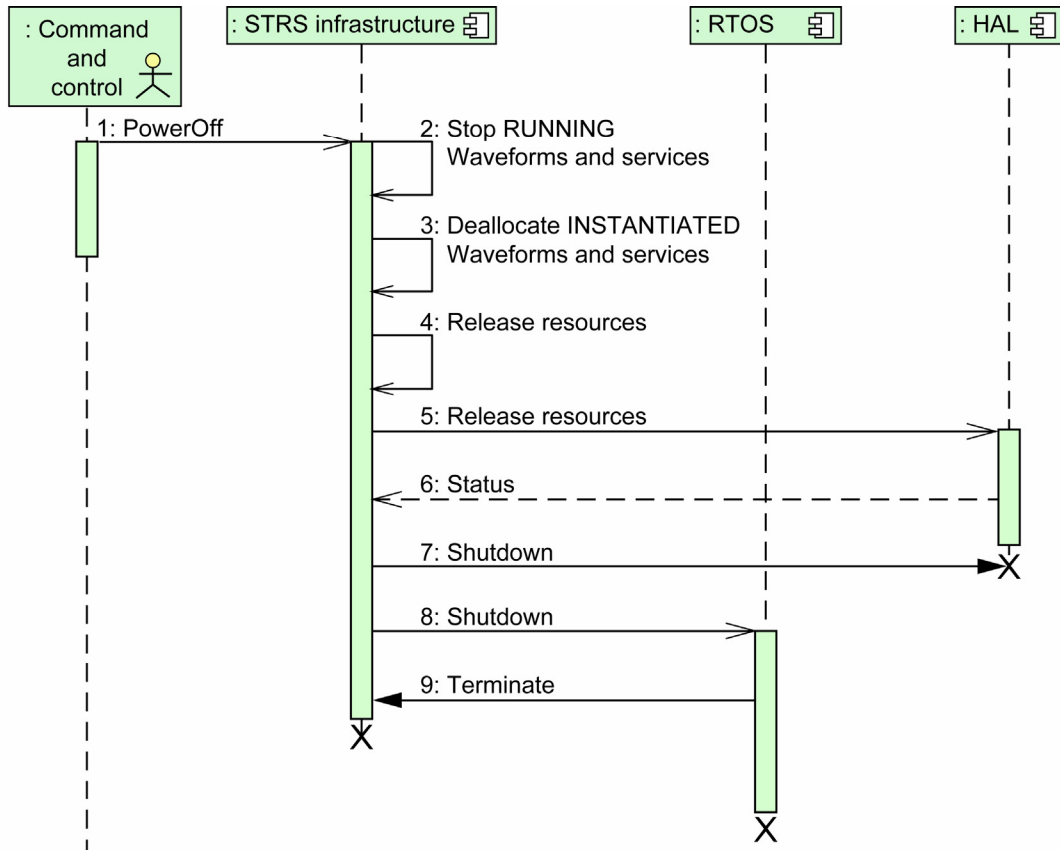


Figure 8.—Power off sequence.

## D.3 Waveform Upload

<b>Use Case Identifier: Waveform Upload</b>																							
<b>Description:</b> The STRS radio receives a waveform upload request from the STRS Command Source. It is assumed that the STRS radio decrypts and authenticates the files. The STRS Radio stores the received file(s) for installation. The STRS Radio returns any nonrepudiation of receipt data to the STRS Command Source.																							
<b>External Actors:</b> STRS Command Source																							
<b>Related Use Cases:</b> Waveform Instantiate and Waveform Remove																							
<b>Precondition:</b> STRS radio is ready to accept commands from STRS Command Source. STRS radio is in READY state. Waveform is in the Waveform EMPTY state. Triggering event: STRS Command Source requests waveform upload.																							
<b>Main Flow:</b> <ol style="list-style-type: none"> <li>1. STRS infrastructure receives a validated command from STRS Command Source.</li> <li>2. STRS infrastructure instantiates and starts a Dedicated Service. The Dedicated Service is in Waveform RUNNING state.</li> <li>3. The Dedicated Service obtains waveform fragments, decrypts, and validates each waveform fragment. Note a fragment is a partial receipt of the waveform.</li> <li>4. The waveform fragments are placed in Persistent storage. STRS Infrastructure marks the waveform as PARTIAL.</li> <li>5. Repeat from step 2 until entire waveform is received.</li> <li>6. The Dedicated Service provides nonrepudiation of receipt information to the STRS Command Source.</li> <li>7. The STRS infrastructure marks the waveform as LOADED.</li> </ol>																							
<b>Resulting event:</b> <b>Post condition:</b> The STRS radio has stored waveform files and the STRS radio continues with its normal operation. STRS radio is in READY state. Waveform is in the Waveform LOADED state. Dedicated Service state is dependent upon specific design.																							
<b>Alternatives:</b> <p>(2a) The Dedicated Service is responsible for handling any timeout error for obtaining the file, partial file resend, or invalid file attributes.</p> <p>(3a) The Dedicated Service encounters an error while storing the waveform. Errors are logged.</p> <p>STRS Command Source or STRS infrastructure may cancel software upload at any time. Need to add the ability to restart and continue from PARTIAL state.</p>																							
<b>Requirements:</b>																							
<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: left; padding: 2px;">Requirement ID</th> <th style="text-align: left; padding: 2px;">Name</th> </tr> </thead> <tbody> <tr><td style="padding: 2px;">CUST1</td><td style="padding: 2px;">Multiple Waveform Support</td></tr> <tr><td style="padding: 2px;">CUST8</td><td style="padding: 2px;">Remote Reconfiguration</td></tr> <tr><td style="padding: 2px;">CUST10.1</td><td style="padding: 2px;">Data Integrity Checks on Uploads</td></tr> <tr><td style="padding: 2px;">CUST22</td><td style="padding: 2px;">Standard Spacecraft Interfaces</td></tr> <tr><td style="padding: 2px;">SYS8</td><td style="padding: 2px;">Secure Transmissions</td></tr> <tr><td style="padding: 2px;">SYS9</td><td style="padding: 2px;">Upload Interfaces</td></tr> <tr><td style="padding: 2px;">SYS10</td><td style="padding: 2px;">File Management</td></tr> <tr><td style="padding: 2px;">SYS10.1</td><td style="padding: 2px;">Persistent Storage Capacity</td></tr> <tr><td style="padding: 2px;">SYS24</td><td style="padding: 2px;">Control and Management</td></tr> <tr><td style="padding: 2px;">SYS30</td><td style="padding: 2px;">Data and Command Integrity</td></tr> </tbody> </table>	Requirement ID	Name	CUST1	Multiple Waveform Support	CUST8	Remote Reconfiguration	CUST10.1	Data Integrity Checks on Uploads	CUST22	Standard Spacecraft Interfaces	SYS8	Secure Transmissions	SYS9	Upload Interfaces	SYS10	File Management	SYS10.1	Persistent Storage Capacity	SYS24	Control and Management	SYS30	Data and Command Integrity	
Requirement ID	Name																						
CUST1	Multiple Waveform Support																						
CUST8	Remote Reconfiguration																						
CUST10.1	Data Integrity Checks on Uploads																						
CUST22	Standard Spacecraft Interfaces																						
SYS8	Secure Transmissions																						
SYS9	Upload Interfaces																						
SYS10	File Management																						
SYS10.1	Persistent Storage Capacity																						
SYS24	Control and Management																						
SYS30	Data and Command Integrity																						
<b>Comments:</b> Dedicated Service may need to be started by the STRS infrastructure. Using a service rather than including this functionality in the STRS infrastructure gives greater flexibility.																							

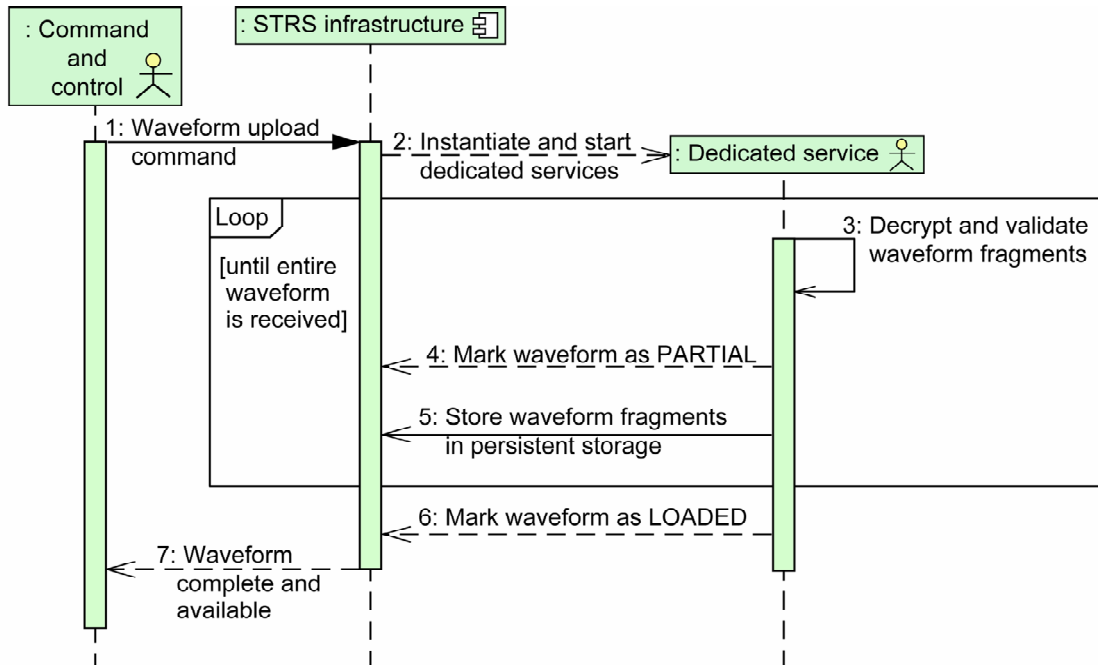


Figure 9.—Waveform upload sequence.



## D.4 STRS OE Upload

<b>Use Case Identifier: STRS OE Upload</b>																						
<p><b>Description:</b> The STRS Radio receives OE Upload request from the STRS Command Source. It is assumed that the STRS radio decrypts and authenticates the files. The STRS radio stores the received file(s) for installation. The STRS Radio returns any nonrepudiation of receipt data to the STRS Command Source.</p>																						
<p><b>External Actors:</b> STRS Command Source</p>																						
<p><b>Related Use Cases:</b> Power On</p>																						
<p><b>Precondition:</b> STRS radio ready to accept commands from STRS Command Source. STRS radio is in READY state.  <b>Triggering event:</b> STRS Command Source requests STRS OE upload.</p>																						
<p><b>Main Flow:</b></p> <ol style="list-style-type: none"> <li>1. STRS infrastructure receives a validated command from STRS Command Source.</li> <li>2. A Dedicated Service is instantiated and started. The Dedicated Service is in Waveform RUNNING state.</li> <li>3. A Dedicated Service obtains new STRS OE component fragments, decrypts, and validates each new STRS OE component fragment. Note: a STRS OE component fragment is a partial receipt of the STRS OE component.</li> <li>4. The STRS OE component fragments are placed in persistent storage. The new STRS OE component fragments do not overwrite the currently running STRS OE. The STRS Infrastructure marks the STRS OE component as PARTIAL.</li> <li>5. Repeat from step 2 until entire STRS OE component is received.</li> <li>6. The Dedicated Service provides nonrepudiation of receipt information to the STRS Command Source.</li> <li>7. The STRS infrastructure marks the new STRS OE component as LOADED.</li> <li>8. Repeat for all desired new STRS OE components.</li> <li>9. The STRS infrastructure marks the new STRS OE components as available.</li> </ol>																						
<p><b>Resulting event:</b>  <b>Post condition:</b> The STRS radio has stored new STRS OE component files and the STRS radio continues with its normal operation. STRS radio is in READY state. The Dedicated Service state is dependent upon specific design.</p>																						
<p>Alternatives:</p> <p>(2a) The Dedicated Service is responsible for handling any timeout error for obtaining the file, partial file resend, or invalid file attributes.</p> <p>(3a) The Dedicated Service encounters an error while storing the waveform. Errors are logged.</p> <p>STRS Command Source or STRS Infrastructure may cancel software upload at any time.            Need to add the ability to restart and continue from PARTIAL state.</p>																						
<p><b>Requirements:</b></p> <table border="1" style="width: 100%; border-collapse: collapse; margin-top: 10px;"> <thead> <tr> <th style="text-align: left; padding: 2px;">Requirement ID</th> <th style="text-align: left; padding: 2px;">Name</th> </tr> </thead> <tbody> <tr><td style="padding: 2px;">CUST1</td><td style="padding: 2px;">Multiple Waveform Support</td></tr> <tr><td style="padding: 2px;">CUST8</td><td style="padding: 2px;">Remote Reconfiguration</td></tr> <tr><td style="padding: 2px;">CUST10.1</td><td style="padding: 2px;">Data Integrity Checks on Uploads</td></tr> <tr><td style="padding: 2px;">CUST22</td><td style="padding: 2px;">Standard Spacecraft Interfaces</td></tr> <tr><td style="padding: 2px;">SYS8</td><td style="padding: 2px;">Secure Transmissions</td></tr> <tr><td style="padding: 2px;">SYS9</td><td style="padding: 2px;">Upload Interfaces</td></tr> <tr><td style="padding: 2px;">SYS10</td><td style="padding: 2px;">File Management</td></tr> <tr><td style="padding: 2px;">SYS10.1</td><td style="padding: 2px;">Persistent Storage Capacity</td></tr> <tr><td style="padding: 2px;">SYS24</td><td style="padding: 2px;">Control and Management</td></tr> <tr><td style="padding: 2px;">SYS30</td><td style="padding: 2px;">Data and Command Integrity</td></tr> </tbody> </table>	Requirement ID	Name	CUST1	Multiple Waveform Support	CUST8	Remote Reconfiguration	CUST10.1	Data Integrity Checks on Uploads	CUST22	Standard Spacecraft Interfaces	SYS8	Secure Transmissions	SYS9	Upload Interfaces	SYS10	File Management	SYS10.1	Persistent Storage Capacity	SYS24	Control and Management	SYS30	Data and Command Integrity
Requirement ID	Name																					
CUST1	Multiple Waveform Support																					
CUST8	Remote Reconfiguration																					
CUST10.1	Data Integrity Checks on Uploads																					
CUST22	Standard Spacecraft Interfaces																					
SYS8	Secure Transmissions																					
SYS9	Upload Interfaces																					
SYS10	File Management																					
SYS10.1	Persistent Storage Capacity																					
SYS24	Control and Management																					
SYS30	Data and Command Integrity																					
<p><b>Comments:</b> This use case is optional because components of the STRS OE may be in ROM. Normally this use case is followed by a shutdown, power off, power on sequence.</p> <p>Dedicated Service may need to be started by the STRS Infrastructure. Using a service rather than including this functionality in the STRS Infrastructure gives greater flexibility.</p>																						

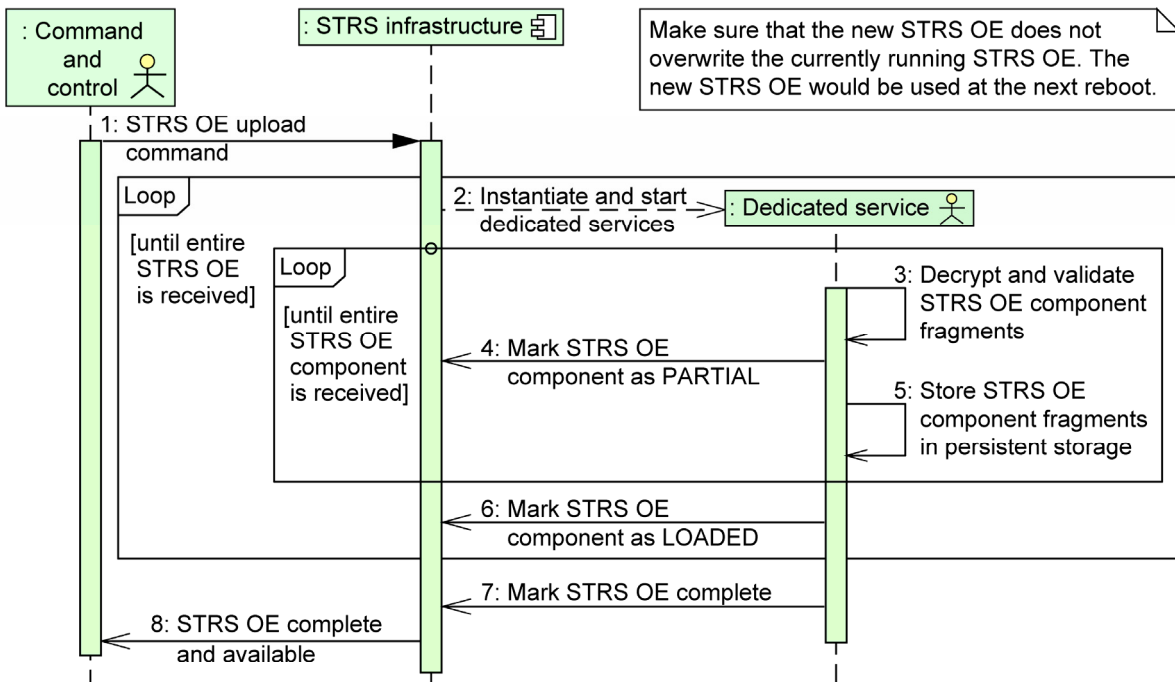


Figure 10.—STRS OE upload sequence.

## D.5 Waveform Instantiation

<b>Use Case Identifier: Waveform Instantiation</b>	
<b>Description:</b> The STRS radio receives a Waveform Instantiate request from the STRS Command Source.	
<b>External Actors:</b> STRS Command Source, Waveform	
<b>Related Use Cases:</b> Waveform Upload and Waveform Deallocate	
<b>Precondition:</b> STRS radio is in READY state. Waveform is in Waveform LOADED state. STRS Command Source requests that a new waveform can be instantiated.	
<b>Triggering event:</b> Instantiation request from STRS Command Source	
<b>Main Flow:</b>	
<ol style="list-style-type: none"> <li>1. STRS infrastructure receives a validated command from STRS Command Source.</li> <li>2. STRS infrastructure begins to process request by reading configuration file for waveform.</li> <li>3. STRS infrastructure loads the configuration images to the target hardware (either GPP or Specialized Hardware).</li> <li>4. STRS infrastructure instantiates the Waveform application.</li> <li>5. Waveform uses the STRS infrastructure to establish connections to the Specialized Hardware via the HAL.</li> <li>6. Waveform uses the STRS infrastructure to configure the hardware and software components.</li> <li>7. STRS infrastructure sends to the STRS Command Source the Waveform Instantiated Status</li> </ol>	
<b>Resulting event:</b>	
<b>Post condition:</b> Waveform ready to accept Start command from STRS Command Source. STRS radio is in READY state. Waveform is in Waveform INSTANTIATED state.	
<b>Alternatives:</b>	
(1–5a) If an error occurs, STRS infrastructure will take the appropriate error handling actions and inform the STRS Command Source.	
(6–7a) If an error occurs, waveform will take the appropriate error handling actions and inform the STRS Command Source	
<b>Requirements:</b>	
<b>Requirement ID</b>	<b>Name</b>
CUST8	Remote Reconfiguration
CUST9	RF Tunability
CUST17	Pre-Launch Hardware Configuration
SYS1	Layered Open Architecture
SYS1.2	Open Interfaces
SYS1.2.1	Waveform API
SYS1.2.2	Device API
SYS5	Common Services
SYS6.1	Data Coding Support
SYS6.2	Modulation Formats
SYS6.3	Coherent/Noncoherent Mode Support
SYS9.1	Reconfiguration APIs
SYS12	Device Diagnostic API
SYS20	Error and Fault Logs
SYS21	Run-Time State Information
SYS24	Control and Management
SYS25	Distributed Functionality
SYS29	RF Module Interface
SYS30	Data and Command Integrity
<b>Comments:</b> Resource allocation and teardown responsibility still needs to be determined in the elaboration phase.	

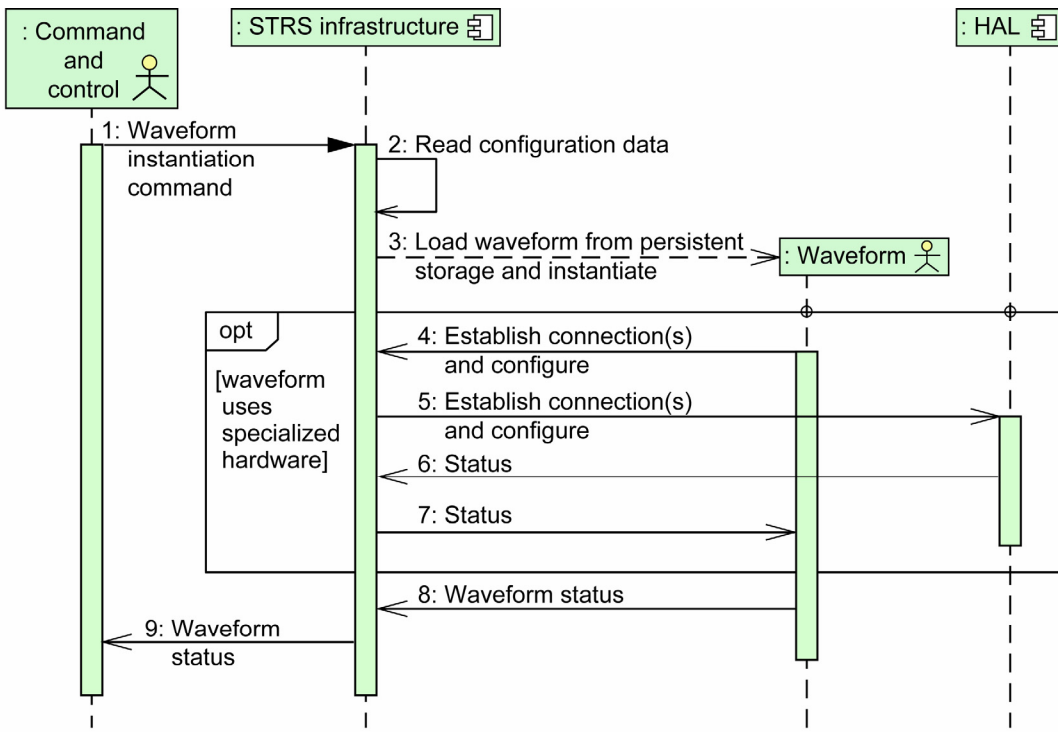


Figure 11.—Waveform instantiation sequence.

## D.6 Waveform Start

<b>Use Case Identifier: Waveform Start</b>																																									
<b>Description:</b> STRS radio receives a request from the STRS Command Source to start a waveform																																									
<b>External Actors:</b> STRS Command Source, Waveform																																									
<b>Related Use Cases:</b> Waveform Instantiate and Waveform Stop																																									
<b>Precondition:</b> STRS radio is in READY state. Waveform is in Waveform INSTANTIATED state. <b>Triggering event:</b> Start Waveform request from STRS Command Source																																									
<b>Main Flow:</b> <ol style="list-style-type: none"> <li>1. STRS infrastructure receives a validated command from STRS Command Source.</li> <li>2. When waveform uses Specialized Hardware and when HAL has already been configured during Waveform Instantiation, STRS infrastructure sends waveform START command to Specialized Hardware via the HAL.</li> <li>3. Waveform starts processing.</li> <li>4. Waveform sends the waveform RUNNING state to STRS infrastructure.</li> <li>5. STRS infrastructure sends to the STRS Command Source Waveform RUNNING state.</li> </ol>																																									
<b>Resulting event:</b> <b>Post condition:</b> STRS radio is in READY state. Waveform is in Waveform RUNNING state.																																									
<b>Alternatives:</b> <ol style="list-style-type: none"> <li>(1a) If an error occurs, STRS infrastructure will take the appropriate error handling actions and inform the STRS Command Source.</li> <li>(2a) When waveform uses Specialized Hardware and when HAL has not been configured during Waveform Instantiation, STRS infrastructure configures the HAL and sends waveform START command to Specialized Hardware via the HAL. Optionally, configuration may include loading the image to the Specialized Hardware, if it wasn't already performed during instantiation.</li> <li>(2b) If an error occurs, STRS infrastructure will take the appropriate error handling actions and inform the STRS Command Source.</li> <li>(3a) If an error occurs, waveform will take the appropriate error handling actions and inform the STRS Command Source via STRS infrastructure.</li> </ol>																																									
<b>Requirements:</b>																																									
	<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: left; padding: 2px;">Requirement ID</th> <th style="text-align: left; padding: 2px;">Name</th> </tr> </thead> <tbody> <tr><td style="padding: 2px;">CUST8</td><td style="padding: 2px;">Remote Reconfiguration</td></tr> <tr><td style="padding: 2px;">CUST9</td><td style="padding: 2px;">RF Tunability</td></tr> <tr><td style="padding: 2px;">CUST17</td><td style="padding: 2px;">Pre-Launch Hardware Configuration</td></tr> <tr><td style="padding: 2px;">SYS1</td><td style="padding: 2px;">Layered Open Architecture</td></tr> <tr><td style="padding: 2px;">SYS1.2</td><td style="padding: 2px;">Open Interfaces</td></tr> <tr><td style="padding: 2px;">SYS1.2.1</td><td style="padding: 2px;">Waveform API</td></tr> <tr><td style="padding: 2px;">SYS1.2.2</td><td style="padding: 2px;">Device API</td></tr> <tr><td style="padding: 2px;">SYS5</td><td style="padding: 2px;">Common Services</td></tr> <tr><td style="padding: 2px;">SYS6.1</td><td style="padding: 2px;">Data Coding Support</td></tr> <tr><td style="padding: 2px;">SYS6.2</td><td style="padding: 2px;">Modulation Formats</td></tr> <tr><td style="padding: 2px;">SYS6.3</td><td style="padding: 2px;">Coherent/Noncoherent Mode Support</td></tr> <tr><td style="padding: 2px;">SYS9.1</td><td style="padding: 2px;">Reconfiguration APIs</td></tr> <tr><td style="padding: 2px;">SYS12</td><td style="padding: 2px;">Device Diagnostic API</td></tr> <tr><td style="padding: 2px;">SYS20</td><td style="padding: 2px;">Error and Fault Logs</td></tr> <tr><td style="padding: 2px;">SYS21</td><td style="padding: 2px;">Run-Time State Information</td></tr> <tr><td style="padding: 2px;">SYS24</td><td style="padding: 2px;">Control and Management</td></tr> <tr><td style="padding: 2px;">SYS25</td><td style="padding: 2px;">Distributed Functionality</td></tr> <tr><td style="padding: 2px;">SYS29</td><td style="padding: 2px;">RF Module Interface</td></tr> <tr><td style="padding: 2px;">SYS30</td><td style="padding: 2px;">Data and Command Integrity</td></tr> </tbody> </table>	Requirement ID	Name	CUST8	Remote Reconfiguration	CUST9	RF Tunability	CUST17	Pre-Launch Hardware Configuration	SYS1	Layered Open Architecture	SYS1.2	Open Interfaces	SYS1.2.1	Waveform API	SYS1.2.2	Device API	SYS5	Common Services	SYS6.1	Data Coding Support	SYS6.2	Modulation Formats	SYS6.3	Coherent/Noncoherent Mode Support	SYS9.1	Reconfiguration APIs	SYS12	Device Diagnostic API	SYS20	Error and Fault Logs	SYS21	Run-Time State Information	SYS24	Control and Management	SYS25	Distributed Functionality	SYS29	RF Module Interface	SYS30	Data and Command Integrity
Requirement ID	Name																																								
CUST8	Remote Reconfiguration																																								
CUST9	RF Tunability																																								
CUST17	Pre-Launch Hardware Configuration																																								
SYS1	Layered Open Architecture																																								
SYS1.2	Open Interfaces																																								
SYS1.2.1	Waveform API																																								
SYS1.2.2	Device API																																								
SYS5	Common Services																																								
SYS6.1	Data Coding Support																																								
SYS6.2	Modulation Formats																																								
SYS6.3	Coherent/Noncoherent Mode Support																																								
SYS9.1	Reconfiguration APIs																																								
SYS12	Device Diagnostic API																																								
SYS20	Error and Fault Logs																																								
SYS21	Run-Time State Information																																								
SYS24	Control and Management																																								
SYS25	Distributed Functionality																																								
SYS29	RF Module Interface																																								
SYS30	Data and Command Integrity																																								
<b>Comments:</b> Resource allocation and teardown responsibility still needs to be determined in the elaboration phase. Normally the Specialized Hardware is loaded prior to the start command.																																									

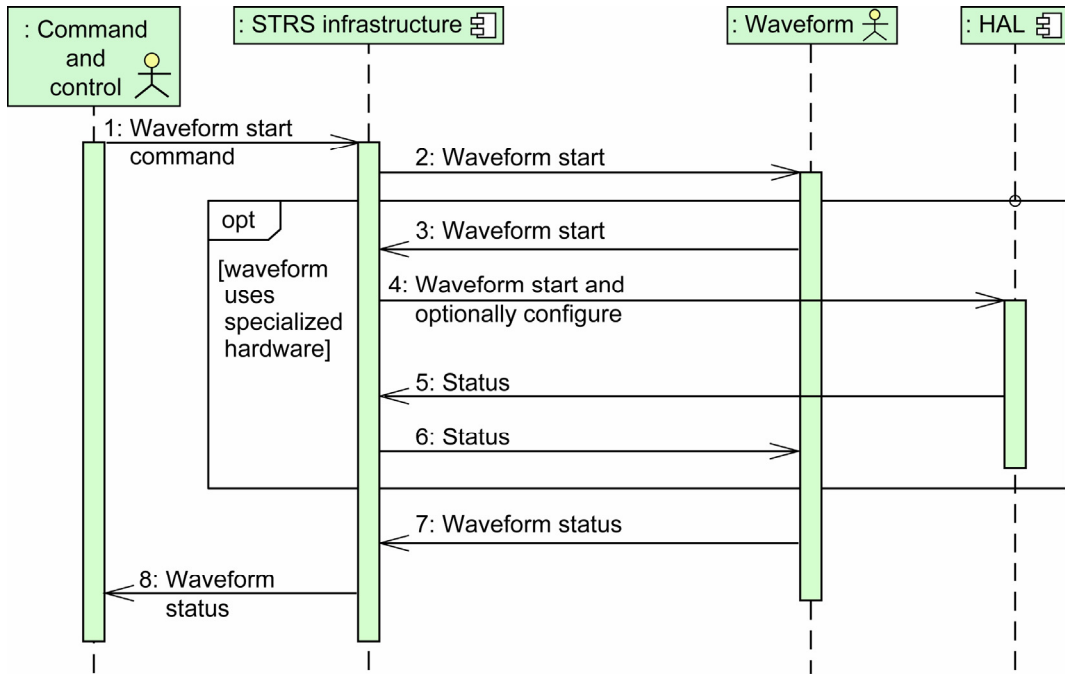


Figure 12.—Waveform start sequence.

## D.7 Processor Resource Sharing

<b>Use Case Identifier: Processor Resource Sharing with Flight Computer</b>	
<b>Description:</b> STRS radio receives a processing request from the Flight Computer to perform a processing workload sharing, e.g., perform data compression on a dataset.	
<b>External Actors:</b> Flight Computer, Sharing Application (Dedicated Service)	
<b>Related Use Cases:</b> Waveform Instantiation and Waveform Start	
<b>Precondition:</b> STRS radio is in READY state. The Sharing Application is in Waveform INSTANTIATED state. The Flight Computer determines that the mission schedule allows the radio to perform the process sharing services.	
<b>Triggering event:</b> Processing Sharing request received from Flight Computer.	
<b>Main Flow:</b>	
<ol style="list-style-type: none"> <li>1. STRS infrastructure receives a validated Process Sharing request from the Flight Computer</li> <li>2. The STRS infrastructure determines whether the radio has the capacity to perform the process sharing services.</li> <li>3. Sharing Application initiates processing (start processor resource sharing).</li> <li>4. Sharing Application sends to the Flight Computer Waveform Running Status</li> <li>5. Sharing Application receives a validated data from the Flight Computer or other data source via STRS Infrastructure.</li> <li>6. Sharing Application returns results to the Flight Computer or other data sink via STRS Infrastructure.</li> <li>7. Repeat from step 3 until processing complete or stop command received.</li> <li>8. STRS Infrastructure sends the Sharing Application status to the Flight Computer.</li> </ol>	
<b>Resulting event:</b> Processing results are returned to the Flight Computer or designated memory location.	
<b>Post condition:</b> STRS radio is in READY state. Sharing Application state is dependent upon specific design.	
<b>Alternatives:</b>	
(5a) Data received from external interface or memory location.	
(6a) Results sent to external interface or memory location.	
(1–2, 8a) If an error occurs, STRS infrastructure will take the appropriate error handling actions and inform the Flight Computer.	
(3–7a) If an error occurs, Sharing Application will take the appropriate error handling actions and inform the Flight Computer.	
<b>Requirements:</b>	
<b>Requirement ID</b>	<b>Name</b>
CUST 8	Remote Reconfiguration
CUST 11	Processing Power Sharing
CUST 13.2	Last Known Configuration
SYS 9.1	Reconfiguration APIs
SYS 10	File Management
SYS 11	Distributed Services
SYS 12	Device Diagnostic API
SYS 13	Availability
SYS 13.1	Process Protection
SYS 17	Event Notification
SYS 17.1	Configurable Event Responses
SYS 18	Background Monitoring
SYS 18.1	Run-Time Exception Reporting
SYS 19	Default Recovery
SYS 20	Error and Fault Logs
SYS 21	Run-Time State Information
SYS 25	Distributed Functionality
SYS 30	Data and Command Integrity
<b>Comments:</b>	
Process Sharing request is assumed to have fields to indicate the type of processing being requested, the location of the data set to be processed, and the destination of the processed data product. Other fields may be required to perform validation or processing capture duration, etc.	

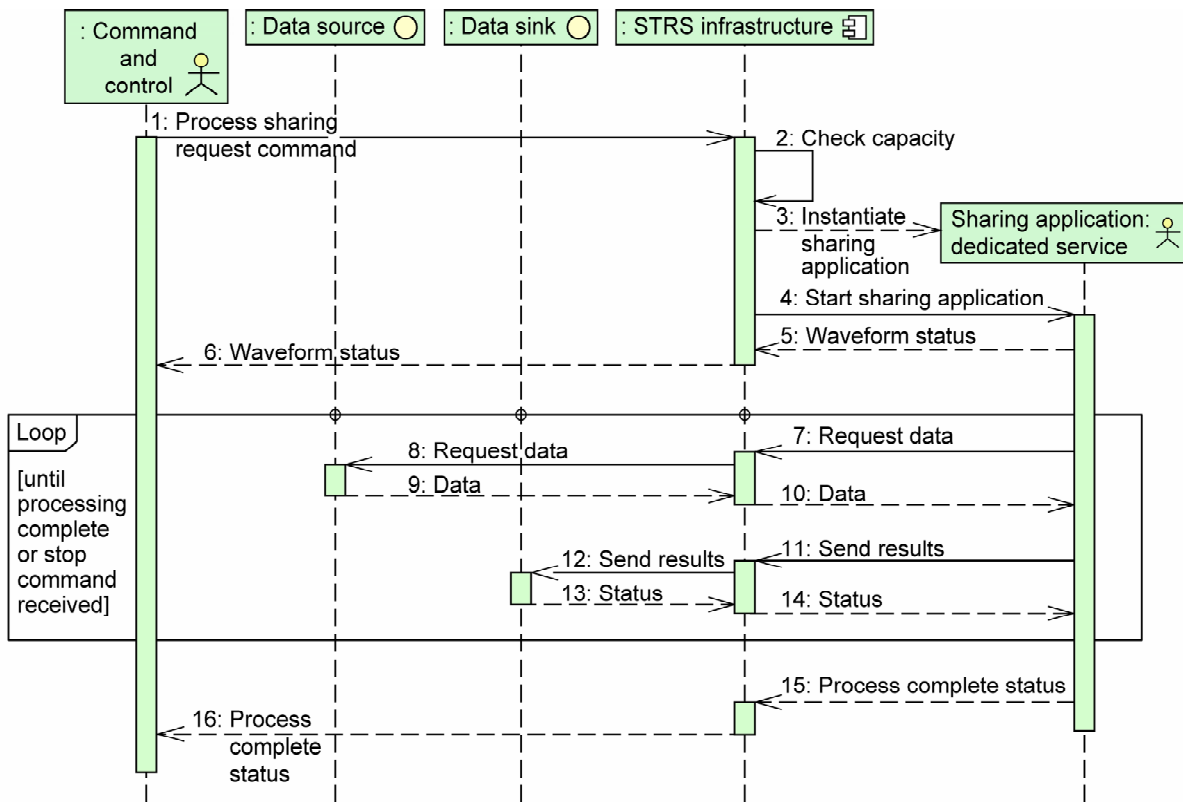


Figure 13.—Processor resource sharing sequence.



## D.8 Set Waveform Parameter

<b>Use Case Identifier: Set Waveform Parameter</b>																							
<b>Description:</b> STRS radio receives a request from the STRS Command Source to set waveform parameter(s).																							
<b>External Actors:</b> STRS Command Source, Waveform																							
<b>Related Use Cases:</b> Get waveform parameter																							
<b>Precondition:</b> STRS radio is in READY state. The waveform is in Waveform RUNNING state or waveform is in Waveform INSTANTIATED state.																							
<b>Triggering event:</b> Set Parameter request from STRS Command Source.																							
<b>Main Flow:</b> <ol style="list-style-type: none"> <li>1. STRS infrastructure receives a validated set parameter request.</li> <li>2. STRS infrastructure sends a set parameter request to waveform.</li> <li>3. The waveform verifies that the parameter(s) are acceptable and configures the resources accordingly optionally via the HAL.</li> <li>4. The waveform sends status to the STRS infrastructure.</li> <li>5. The STRS infrastructure indicates to the STRS Command Source that the waveform has been configured.</li> </ol>																							
<b>Resulting event:</b> <b>Post condition:</b> STRS radio is in READY state. The requested waveform parameter(s) have been set.																							
<b>Alternatives:</b> (3a) The waveform rejects the parameter(s) and notifies the STRS Command Source that the parameter(s) have been rejected.																							
<b>Requirements:</b> <table border="1" style="width: 100%; border-collapse: collapse; margin-top: 10px;"> <thead> <tr> <th style="text-align: left; padding: 2px;">Requirement ID</th> <th style="text-align: left; padding: 2px;">Name</th> </tr> </thead> <tbody> <tr><td style="padding: 2px;">CUST5</td><td style="padding: 2px;">Network Performance Monitoring</td></tr> <tr><td style="padding: 2px;">CUST8</td><td style="padding: 2px;">Remote Reconfiguration</td></tr> <tr><td style="padding: 2px;">CUST9</td><td style="padding: 2px;">RF Tunability</td></tr> <tr><td style="padding: 2px;">CUST15</td><td style="padding: 2px;">Commanded Built-in-Test and Status Reporting</td></tr> <tr><td style="padding: 2px;">CUST18</td><td style="padding: 2px;">Multiple Frequency Bands</td></tr> <tr><td style="padding: 2px;">CUST22</td><td style="padding: 2px;">Standard Spacecraft Interfaces</td></tr> <tr><td style="padding: 2px;">SYS10</td><td style="padding: 2px;">File Management</td></tr> <tr><td style="padding: 2px;">SYS10.1</td><td style="padding: 2px;">Persistent Storage Capacity</td></tr> <tr><td style="padding: 2px;">SYS24</td><td style="padding: 2px;">Control and Management</td></tr> <tr><td style="padding: 2px;">SYS30</td><td style="padding: 2px;">Data and Command Integrity</td></tr> </tbody> </table>		Requirement ID	Name	CUST5	Network Performance Monitoring	CUST8	Remote Reconfiguration	CUST9	RF Tunability	CUST15	Commanded Built-in-Test and Status Reporting	CUST18	Multiple Frequency Bands	CUST22	Standard Spacecraft Interfaces	SYS10	File Management	SYS10.1	Persistent Storage Capacity	SYS24	Control and Management	SYS30	Data and Command Integrity
Requirement ID	Name																						
CUST5	Network Performance Monitoring																						
CUST8	Remote Reconfiguration																						
CUST9	RF Tunability																						
CUST15	Commanded Built-in-Test and Status Reporting																						
CUST18	Multiple Frequency Bands																						
CUST22	Standard Spacecraft Interfaces																						
SYS10	File Management																						
SYS10.1	Persistent Storage Capacity																						
SYS24	Control and Management																						
SYS30	Data and Command Integrity																						
<b>Comments:</b> Parameters include frequency, power level, bit rate, etc. Multiple parameters may be set, but probably one at a time may be sent to the Specialized Hardware via the HAL.																							

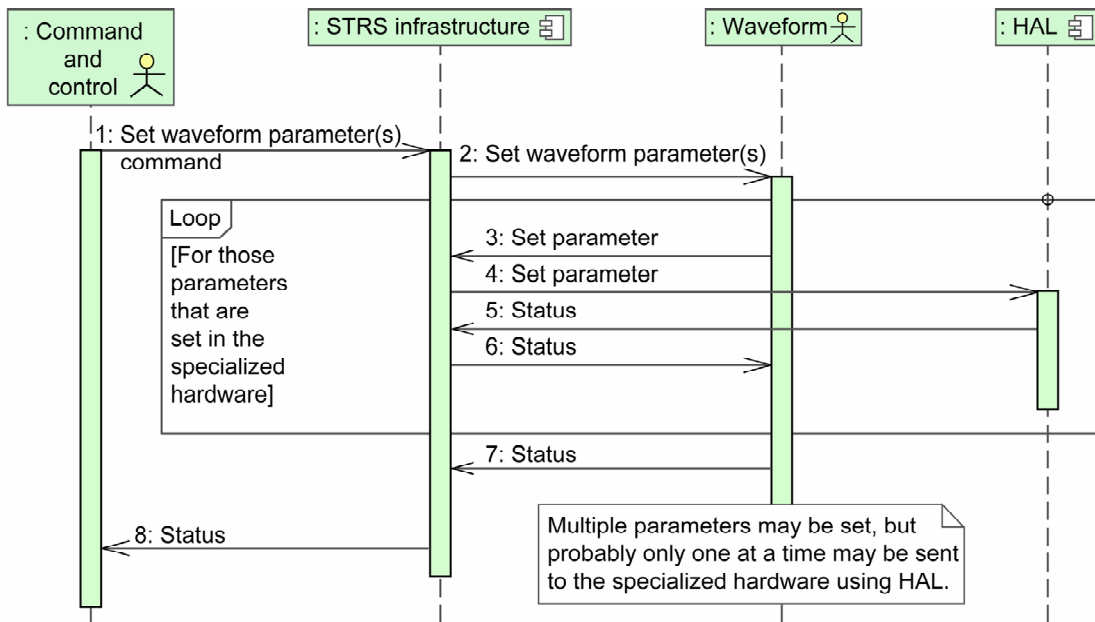


Figure 14.—Set waveform parameter sequence.

## D.9 Get Waveform Parameter

<b>Use Case Identifier: Get Waveform Parameter</b>	
<b>Description:</b> STRS radio receives a request from STRS Command Source to get Waveform parameter(s).	
<b>External Actors:</b> STRS Command Source, Waveform	
<b>Related Use Cases:</b> Set Waveform Parameter	
<b>Precondition:</b> STRS radio is in READY state. The waveform is in Waveform RUNNING state or waveform is in Waveform INSTANTIATED state.	
<b>Triggering event:</b> Get Parameter request from STRS Command Source.	
<b>Main Flow:</b>	
<ol style="list-style-type: none"> <li>1. STRS infrastructure receives a validated get parameter request.</li> <li>2. STRS infrastructure sends a get parameter request to waveform.</li> <li>3. The waveform verifies and retrieves the parameter(s) optionally via the HAL.</li> <li>4. The waveform sends requested parameter(s) to the STRS infrastructure.</li> <li>5. The STRS infrastructure returns requested parameters to the controlling STRS Application and/or STRS Command Source.</li> </ol>	
<b>Resulting event:</b>	
<b>Post condition:</b> STRS radio is in READY state. The requested waveform parameter(s) have been retrieved.	
<b>Alternatives:</b>	
(3a) The waveform rejects the request and notifies the STRS Command Source that the requested parameter does not exist.	
<b>Requirements:</b>	
<b>Requirement ID</b>	<b>Name</b>
CUST5	Network Performance Monitoring
CUST8	Remote Reconfiguration
CUST9	RF Tunability
CUST15	Commanded Built-in-Test and Status Reporting
CUST18	Multiple Frequency Bands
CUST22	Standard Spacecraft Interfaces
SYS10	File Management
SYS10.1	Persistent Storage Capacity
SYS24	Control and Management
SYS30	Data and Command Integrity
<b>Comments:</b> Parameters include frequency, power level, bit rate, etc. Multiple parameters may be retrieved, but probably one at a time may be obtain from the Specialized Hardware via the HAL.	

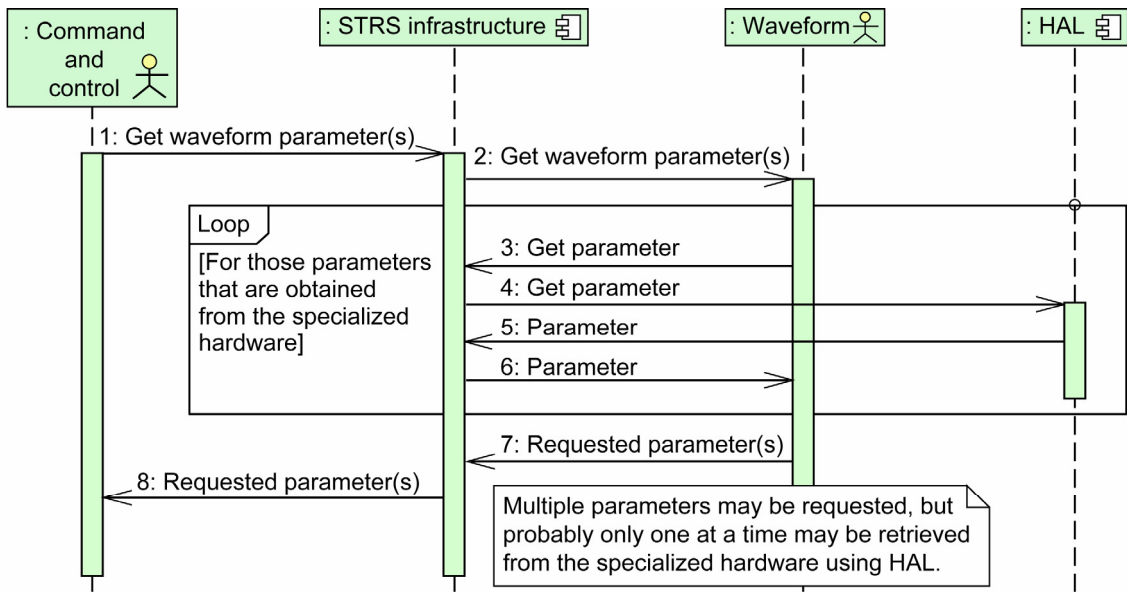


Figure 15.—Get waveform parameter sequence.

## D.10 Transmit a Packet

<b>Use Case Identifier: Transmit a Packet</b>																														
<b>Description:</b> A data packet from the Data Publisher or Source is transmitted by a waveform across the communication link.																														
<b>External Actors:</b> Data Publisher or Source, External Radio (Receiver), Waveform																														
<b>Related Use Cases:</b> Receive a Packet																														
<b>Precondition:</b> Radio is in READY state. Waveform is in Waveform RUNNING state. <b>Triggering event:</b> Data Source sends a packet.																														
<b>Main Flow:</b> <ol style="list-style-type: none"> <li>1. When the Data Source is separate from the waveform, user data packets that are sent from the Data Source are received by the infrastructure.</li> <li>2. When the Data Source is separate from the waveform, the infrastructure sends the user data packets to the waveform.</li> <li>3. The waveform transforms the user data packets into data packets by any necessary encryption, encoding, modulations, etc.</li> <li>4. When waveform uses Specialized Hardware, data packets are sent from the waveform to the infrastructure.</li> <li>5. When waveform uses Specialized Hardware, data packets are sent from infrastructure to the Specialized Hardware via the HAL.</li> <li>6. Status may be returned.</li> </ol>																														
<b>Resulting event:</b> STRS radio transmits a packet (or packets). <b>Post condition:</b> The STRS radio remains in READY state																														
<b>Alternatives:</b> (1–6a) The waveform detects the occurrence of a fault or alarm (Crypto Alarm, out of lock, over temperature, low output power, etc.) and disables transmission and notifies the Data Source and/or STRS Command Source of the fault. (1–6b) When the Data Source is not separate from the waveform, the waveform creates the user data packets.																														
<b>Nonfunctional requirements:</b> Mission dependent																														
<b>Requirements:</b>																														
<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: left; padding: 2px;">Requirement ID</th> <th style="text-align: left; padding: 2px;">Name</th> </tr> </thead> <tbody> <tr><td style="padding: 2px;">CUST1</td><td style="padding: 2px;">Multiple Waveform Support</td></tr> <tr><td style="padding: 2px;">CUST3</td><td style="padding: 2px;">Secure Transmission</td></tr> <tr><td style="padding: 2px;">CUST4</td><td style="padding: 2px;">Transparent Network Routing</td></tr> <tr><td style="padding: 2px;">CUST9</td><td style="padding: 2px;">RF Tunability</td></tr> <tr><td style="padding: 2px;">CUST13</td><td style="padding: 2px;">Automate System Recovery/Initialization</td></tr> <tr><td style="padding: 2px;">CUST19</td><td style="padding: 2px;">Multichannel Support</td></tr> <tr><td style="padding: 2px;">SYS5.1</td><td style="padding: 2px;">Protocol Support</td></tr> <tr><td style="padding: 2px;">SYS6.1</td><td style="padding: 2px;">Data Coding Support</td></tr> <tr><td style="padding: 2px;">SYS6.2</td><td style="padding: 2px;">Modulation Formats</td></tr> <tr><td style="padding: 2px;">SYS6.3</td><td style="padding: 2px;">Coherent/Noncoherent Mode Support</td></tr> <tr><td style="padding: 2px;">SYS6.4</td><td style="padding: 2px;">DSSS Support</td></tr> <tr><td style="padding: 2px;">SYS7</td><td style="padding: 2px;">Link Metric Reporting</td></tr> <tr><td style="padding: 2px;">SYS26</td><td style="padding: 2px;">Frequency Variability</td></tr> <tr><td style="padding: 2px;">SYS27.1</td><td style="padding: 2px;">Independent Data Rate</td></tr> </tbody> </table>	Requirement ID	Name	CUST1	Multiple Waveform Support	CUST3	Secure Transmission	CUST4	Transparent Network Routing	CUST9	RF Tunability	CUST13	Automate System Recovery/Initialization	CUST19	Multichannel Support	SYS5.1	Protocol Support	SYS6.1	Data Coding Support	SYS6.2	Modulation Formats	SYS6.3	Coherent/Noncoherent Mode Support	SYS6.4	DSSS Support	SYS7	Link Metric Reporting	SYS26	Frequency Variability	SYS27.1	Independent Data Rate
Requirement ID	Name																													
CUST1	Multiple Waveform Support																													
CUST3	Secure Transmission																													
CUST4	Transparent Network Routing																													
CUST9	RF Tunability																													
CUST13	Automate System Recovery/Initialization																													
CUST19	Multichannel Support																													
SYS5.1	Protocol Support																													
SYS6.1	Data Coding Support																													
SYS6.2	Modulation Formats																													
SYS6.3	Coherent/Noncoherent Mode Support																													
SYS6.4	DSSS Support																													
SYS7	Link Metric Reporting																													
SYS26	Frequency Variability																													
SYS27.1	Independent Data Rate																													
<b>Comments:</b> Note that this use case does not address the protocols associated with setting up the link, guaranteeing the packet is received correctly, or protocols associated with retransmission.  The STRS radio may have mission requirements for QoS, routing (point-to-point, broadcast or multicast address).  STRS Applications may contain combining or parsing of packets into frames, it may contain modification of the packet protocols (such as TCP-IP Performance Enhancing Proxies), and may contain encryption.																														



## D.11 Receive a Packet

<b>Use Case Identifier: Receive a Packet</b>	
<b>Description:</b> A data packet provided by an External Radio is received across the communication link by a waveform and provided to the Data Subscriber or Sink.	
<b>External Actors:</b> Data Subscriber or Sink, External Radio (Transmitter), Waveform	
<b>Related Use Cases:</b> Transmit a Packet	
<b>Precondition:</b> Radio is in READY state. Waveform is in Waveform RUNNING state. <b>Triggering event:</b> Waveform receives a packet from an External Radio	
<b>Main Flow:</b> <ol style="list-style-type: none"> <li>1. When waveform uses Specialized Hardware, data packets are sent to the infrastructure from the Specialized Hardware via the HAL.</li> <li>2. When waveform uses Specialized Hardware, data packets are sent to the waveform from the infrastructure.</li> <li>3. The waveform transforms the data packets into user data packets by any necessary demodulation, decoding, decryption, etc.</li> <li>4. When the Data Sink is separate from the waveform, the waveform sends the user data packets to the infrastructure.</li> <li>5. When the Data Sink is separate from the waveform, the infrastructure sends the user data packets to the Data Sink.</li> <li>6. Status may be returned.</li> </ol>	
<b>Resulting event:</b> A packet (or packets) is received by the Data Sink. <b>Post condition:</b> The STRS radio remains in READY state	
<b>Alternatives:</b> (1–6a) The waveform detects the occurrence of a fault or alarm (Crypto Alarm, out of lock, over temperature, low output power) and disables transmission and notifies the Data Sink of the fault. (1–2b) When the Data Sink is not separate from the waveform, the waveform processes the user data packets. (1–2c) When the Data Sink is separate from the waveform and the infrastructure cannot process the packet, the infrastructure notifies the Data Sink of a dropped packet. (3b) The waveform cannot process the packet and notifies the Data Sink of a dropped packet.	
<b>Nonfunctional requirements:</b> Mission dependent	
<b>Requirements:</b>	
<b>Requirement ID</b>	<b>Name</b>
CUST1	Multiple Waveform Support
CUST3	Secure Transmission
CUST4	Transparent Network Routing
CUST9	RF Tunability
CUST13	Automate System Recovery/Initialization
CUST19	Multichannel Support
SYS5.1	Protocol Support
SYS6.1	Data Coding Support
SYS6.2	Modulation Formats
SYS6.3	Coherent/Noncoherent Mode Support
SYS6.4	DSSS Support
SYS7	Link Metric Reporting
SYS26	Frequency Variability
SYS27.1	Independent Data Rate
<b>Comments:</b> Note that this use case does not address the protocols associated with setting up the link, guaranteeing the packet is received correctly, or protocols associated with retransmission.  The STRS radio may have mission requirements for QoS, routing (point-to-point, broadcast or multicast address).  STRS Applications may contain combining or parsing of packets into frames, it may contain modification of the packet protocols (such as TCP-IP Performance Enhancing Proxies), and may contain encryption.	

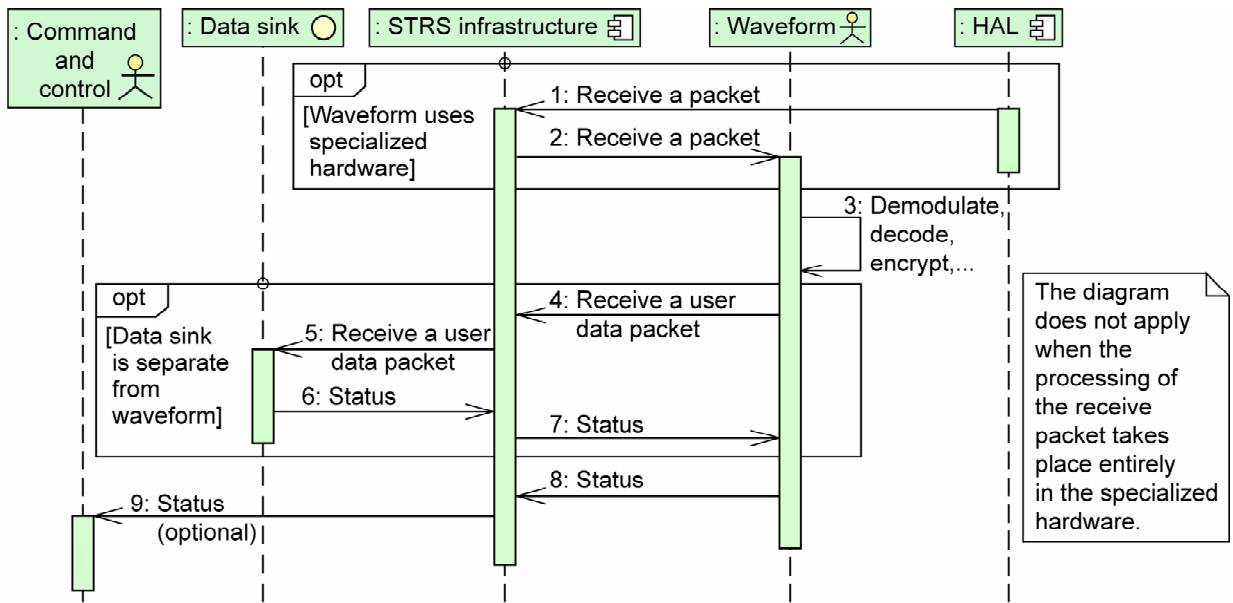


Figure 17.—Waveform receive a packet sequence.



## D.12 Waveform Stop

<b>Use Case Identifier: Waveform Stop</b>																																									
<b>Description:</b> STRS radio receives a request from the STRS Command Source to stop a waveform.																																									
<b>External Actors:</b> STRS Command Source, Waveform																																									
<b>Related Use Cases:</b> Waveform Start																																									
<b>Precondition:</b> STRS radio is in READY state. The waveform is in Waveform RUNNING state. <b>Triggering event:</b> Stop Waveform request from STRS Command Source.																																									
<b>Main Flow:</b> <ol style="list-style-type: none"> <li>1. STRS infrastructure receives a validated command from STRS Command Source.</li> <li>2. STRS infrastructure sends waveform STOP command.</li> <li>3. When waveform uses Specialized Hardware, the waveform uses the STRS infrastructure to send the STOP command to the Specialized Hardware via the HAL.</li> <li>4. Waveform stops processing.</li> <li>5. Waveform acknowledges with waveform stopped message to STRS infrastructure.</li> <li>6. STRS infrastructure sends to the STRS Command Source the Waveform INSTANTIATED state.</li> </ol>																																									
<b>Resulting event:</b> <b>Post condition:</b> STRS radio is in READY state. Waveform is stopped and is in Waveform INSTANTIATED state and ready to receive a Waveform start command or Waveform Deallocate command.																																									
<b>Alternatives:</b> (1–3a) If an error occurs, STRS infrastructure will take the appropriate error handling actions and inform the STRS Command Source. (4a) If an error occurs, waveform will take the appropriate error handling actions and inform the STRS Command Source.																																									
<b>Requirements:</b>																																									
<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: left; padding: 2px;">Requirement ID</th> <th style="text-align: left; padding: 2px;">Name</th> </tr> </thead> <tbody> <tr><td style="padding: 2px;">CUST8</td><td style="padding: 2px;">Remote Reconfiguration</td></tr> <tr><td style="padding: 2px;">CUST9</td><td style="padding: 2px;">RF Tunability</td></tr> <tr><td style="padding: 2px;">CUST17</td><td style="padding: 2px;">Pre-Launch Hardware Configuration</td></tr> <tr><td style="padding: 2px;">SYS1</td><td style="padding: 2px;">Layered Open Architecture</td></tr> <tr><td style="padding: 2px;">SYS1.2</td><td style="padding: 2px;">Open Interfaces</td></tr> <tr><td style="padding: 2px;">SYS1.2.1</td><td style="padding: 2px;">Waveform API</td></tr> <tr><td style="padding: 2px;">SYS1.2.2</td><td style="padding: 2px;">Device API</td></tr> <tr><td style="padding: 2px;">SYS5</td><td style="padding: 2px;">Common Services</td></tr> <tr><td style="padding: 2px;">SYS6.1</td><td style="padding: 2px;">Data Coding Support</td></tr> <tr><td style="padding: 2px;">SYS6.2</td><td style="padding: 2px;">Modulation Formats</td></tr> <tr><td style="padding: 2px;">SYS6.3</td><td style="padding: 2px;">Coherent/Noncoherent Mode Support</td></tr> <tr><td style="padding: 2px;">SYS9.1</td><td style="padding: 2px;">Reconfiguration APIs</td></tr> <tr><td style="padding: 2px;">SYS12</td><td style="padding: 2px;">Device Diagnostic API</td></tr> <tr><td style="padding: 2px;">SYS20</td><td style="padding: 2px;">Error and Fault Logs</td></tr> <tr><td style="padding: 2px;">SYS21</td><td style="padding: 2px;">Run-Time State Information</td></tr> <tr><td style="padding: 2px;">SYS24</td><td style="padding: 2px;">Control and Management</td></tr> <tr><td style="padding: 2px;">SYS25</td><td style="padding: 2px;">Distributed Functionality</td></tr> <tr><td style="padding: 2px;">SYS29</td><td style="padding: 2px;">RF Module Interface</td></tr> <tr><td style="padding: 2px;">SYS30</td><td style="padding: 2px;">Data and Command Integrity</td></tr> </tbody> </table>	Requirement ID	Name	CUST8	Remote Reconfiguration	CUST9	RF Tunability	CUST17	Pre-Launch Hardware Configuration	SYS1	Layered Open Architecture	SYS1.2	Open Interfaces	SYS1.2.1	Waveform API	SYS1.2.2	Device API	SYS5	Common Services	SYS6.1	Data Coding Support	SYS6.2	Modulation Formats	SYS6.3	Coherent/Noncoherent Mode Support	SYS9.1	Reconfiguration APIs	SYS12	Device Diagnostic API	SYS20	Error and Fault Logs	SYS21	Run-Time State Information	SYS24	Control and Management	SYS25	Distributed Functionality	SYS29	RF Module Interface	SYS30	Data and Command Integrity	
Requirement ID	Name																																								
CUST8	Remote Reconfiguration																																								
CUST9	RF Tunability																																								
CUST17	Pre-Launch Hardware Configuration																																								
SYS1	Layered Open Architecture																																								
SYS1.2	Open Interfaces																																								
SYS1.2.1	Waveform API																																								
SYS1.2.2	Device API																																								
SYS5	Common Services																																								
SYS6.1	Data Coding Support																																								
SYS6.2	Modulation Formats																																								
SYS6.3	Coherent/Noncoherent Mode Support																																								
SYS9.1	Reconfiguration APIs																																								
SYS12	Device Diagnostic API																																								
SYS20	Error and Fault Logs																																								
SYS21	Run-Time State Information																																								
SYS24	Control and Management																																								
SYS25	Distributed Functionality																																								
SYS29	RF Module Interface																																								
SYS30	Data and Command Integrity																																								
<b>Comments:</b> Resource allocation and teardown responsibility still needs to be determined in the elaboration phase.																																									

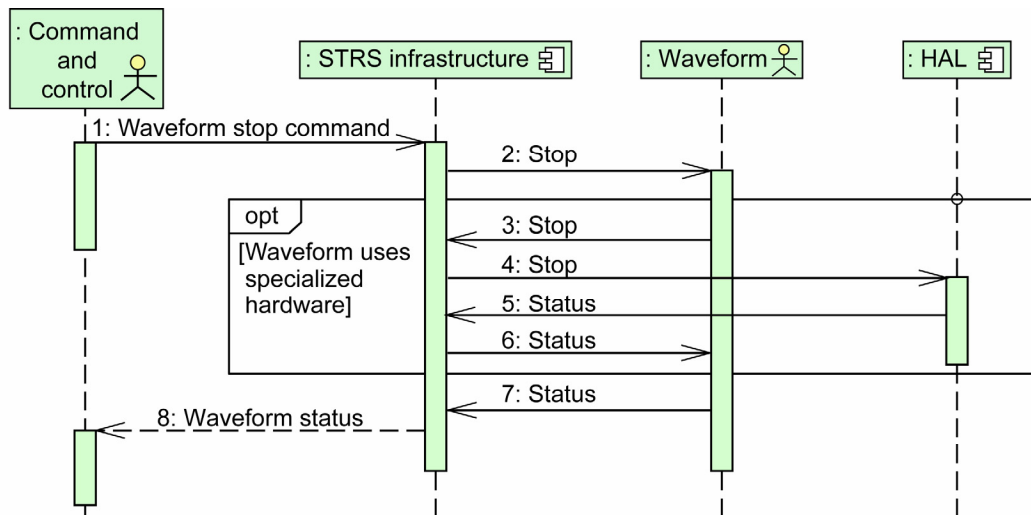


Figure 18.—Waveform stop sequence.

## D.13 Waveform Deallocate

<b>Use Case Identifier: Waveform Deallocation</b>																																									
<b>Description:</b> STRS radio receives a request from the STRS Command Source to deallocate a waveform.																																									
<b>External Actors:</b> STRS Command Source, Waveform, (Specialized Hardware via HAL)																																									
<b>Related Use Cases:</b> Waveform Instantiation																																									
<b>Precondition:</b> STRS radio is in READY state. The waveform is in Waveform INSTANTIATED state. <b>Triggering event:</b> Deallocation request from STRS Command Source																																									
<b>Main Flow:</b>																																									
<ol style="list-style-type: none"> <li>1. STRS infrastructure receives a validated command from STRS Command Source.</li> <li>2. STRS infrastructure sends the Waveform Deallocate request to Waveform.</li> <li>3. When waveform uses Specialized Hardware, the waveform uses the STRS infrastructure to release and close connections to the Specialized Hardware via the HAL.</li> <li>4. Waveform releases resources.</li> <li>5. Waveform terminates.</li> <li>6. STRS infrastructure releases resources associated with waveform</li> <li>7. STRS infrastructure sends to the STRS Command Source Waveform LOADED state.</li> </ol>																																									
<b>Resulting event:</b>																																									
<b>Post condition:</b> STRS radio is in READY state. The waveform and its resources are deallocated. The waveform is in Waveform LOADED state.																																									
<b>Alternatives:</b>																																									
(1–7a) If an error occurs, STRS infrastructure will take the appropriate error handling actions and inform the STRS Command Source.																																									
<b>Requirements:</b>																																									
<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: left; padding: 2px;">Requirement ID</th> <th style="text-align: left; padding: 2px;">Name</th> </tr> </thead> <tbody> <tr><td style="padding: 2px;">CUST8</td><td style="padding: 2px;">Remote Reconfiguration</td></tr> <tr><td style="padding: 2px;">CUST9</td><td style="padding: 2px;">RF Tunability</td></tr> <tr><td style="padding: 2px;">CUST17</td><td style="padding: 2px;">Pre-Launch Hardware Configuration</td></tr> <tr><td style="padding: 2px;">SYS1</td><td style="padding: 2px;">Layered Open Architecture</td></tr> <tr><td style="padding: 2px;">SYS1.2</td><td style="padding: 2px;">Open Interfaces</td></tr> <tr><td style="padding: 2px;">SYS1.2.1</td><td style="padding: 2px;">Waveform API</td></tr> <tr><td style="padding: 2px;">SYS1.2.2</td><td style="padding: 2px;">Device API</td></tr> <tr><td style="padding: 2px;">SYS5;</td><td style="padding: 2px;">Common Services</td></tr> <tr><td style="padding: 2px;">SYS6.1</td><td style="padding: 2px;">Data Coding Support</td></tr> <tr><td style="padding: 2px;">SYS6.2</td><td style="padding: 2px;">Modulation Formats</td></tr> <tr><td style="padding: 2px;">SYS6.3</td><td style="padding: 2px;">Coherent/Noncoherent Mode Support</td></tr> <tr><td style="padding: 2px;">SYS9.1</td><td style="padding: 2px;">Reconfiguration APIs</td></tr> <tr><td style="padding: 2px;">SYS12</td><td style="padding: 2px;">Device Diagnostic API</td></tr> <tr><td style="padding: 2px;">SYS20</td><td style="padding: 2px;">Error and Fault Logs</td></tr> <tr><td style="padding: 2px;">SYS21</td><td style="padding: 2px;">Run-Time State Information</td></tr> <tr><td style="padding: 2px;">SYS24</td><td style="padding: 2px;">Control and Management</td></tr> <tr><td style="padding: 2px;">SYS25</td><td style="padding: 2px;">Distributed Functionality</td></tr> <tr><td style="padding: 2px;">SYS29</td><td style="padding: 2px;">RF Module Interface</td></tr> <tr><td style="padding: 2px;">SYS30</td><td style="padding: 2px;">Data and Command Integrity</td></tr> </tbody> </table>	Requirement ID	Name	CUST8	Remote Reconfiguration	CUST9	RF Tunability	CUST17	Pre-Launch Hardware Configuration	SYS1	Layered Open Architecture	SYS1.2	Open Interfaces	SYS1.2.1	Waveform API	SYS1.2.2	Device API	SYS5;	Common Services	SYS6.1	Data Coding Support	SYS6.2	Modulation Formats	SYS6.3	Coherent/Noncoherent Mode Support	SYS9.1	Reconfiguration APIs	SYS12	Device Diagnostic API	SYS20	Error and Fault Logs	SYS21	Run-Time State Information	SYS24	Control and Management	SYS25	Distributed Functionality	SYS29	RF Module Interface	SYS30	Data and Command Integrity	
Requirement ID	Name																																								
CUST8	Remote Reconfiguration																																								
CUST9	RF Tunability																																								
CUST17	Pre-Launch Hardware Configuration																																								
SYS1	Layered Open Architecture																																								
SYS1.2	Open Interfaces																																								
SYS1.2.1	Waveform API																																								
SYS1.2.2	Device API																																								
SYS5;	Common Services																																								
SYS6.1	Data Coding Support																																								
SYS6.2	Modulation Formats																																								
SYS6.3	Coherent/Noncoherent Mode Support																																								
SYS9.1	Reconfiguration APIs																																								
SYS12	Device Diagnostic API																																								
SYS20	Error and Fault Logs																																								
SYS21	Run-Time State Information																																								
SYS24	Control and Management																																								
SYS25	Distributed Functionality																																								
SYS29	RF Module Interface																																								
SYS30	Data and Command Integrity																																								
<b>Comments:</b> Resource allocation and teardown responsibility still needs to be determined in the elaboration phase.																																									

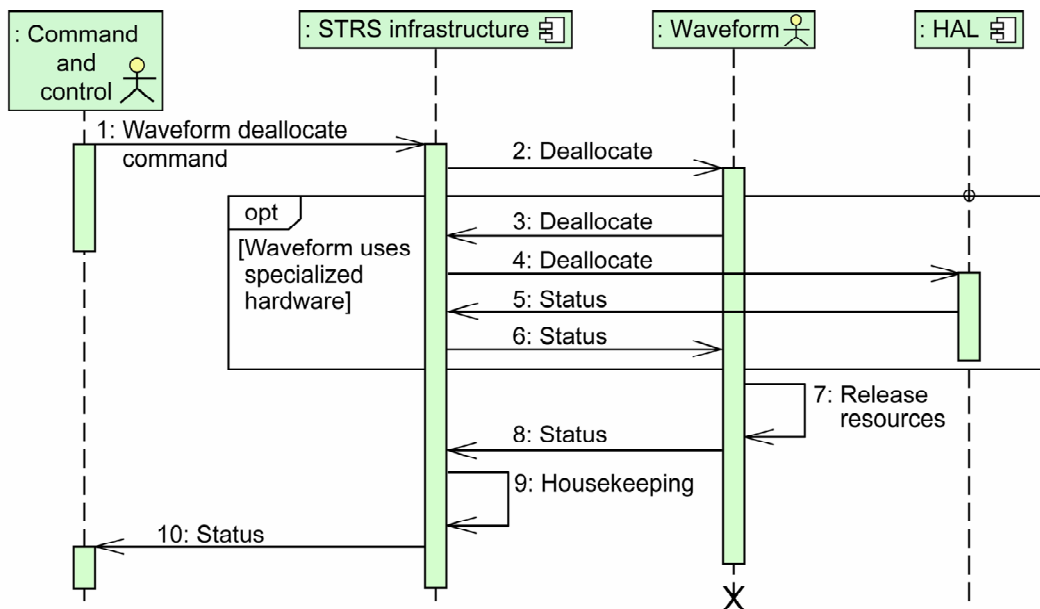


Figure 19.—Waveform deallocate sequence.

## D.14 Waveform Abort

<b>Use Case Identifier: Waveform Abort</b>																																								
<b>Description:</b> STRS radio receives a request from the STRS Command Source to abort a Waveform.																																								
<b>External Actors:</b> STRS Command Source, Waveform, [Specialized Hardware via the HAL]																																								
<b>Related Use Cases:</b> Waveform Stop and Waveform Deallocate																																								
<b>Precondition:</b> STRS radio is in READY state. The waveform is in Waveform RUNNING state. <b>Triggering event:</b> Abort request from STRS Command Source																																								
<b>Main Flow:</b> <ol style="list-style-type: none"> <li>1. STRS infrastructure receives a validated command from STRS Command Source.</li> <li>2. When waveform uses Specialized Hardware, the STRS Infrastructure releases and closes connections to the Specialized Hardware via the HAL.</li> <li>3. STRS infrastructure terminates threads and or processes associated with the waveform</li> <li>4. STRS infrastructure releases resources associated with the waveform.</li> <li>5. STRS infrastructure sends to the STRS Command Source Waveform LOADED state.</li> </ol>																																								
<b>Resulting event:</b> <b>Post condition:</b> STRS radio is in READY state. Waveform and associated resources released. The waveform is in Waveform LOADED state.																																								
<b>Alternatives:</b> (1–5a) If an error occurs, STRS infrastructure will take the appropriate error handling actions and inform the STRS Command Source.																																								
<b>Requirements:</b>																																								
<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: left; padding: 2px;">Requirement ID</th> <th style="text-align: left; padding: 2px;">Name</th> </tr> </thead> <tbody> <tr><td style="padding: 2px;">CUST8</td><td style="padding: 2px;">Remote Reconfiguration</td></tr> <tr><td style="padding: 2px;">CUST9</td><td style="padding: 2px;">RF Tunability</td></tr> <tr><td style="padding: 2px;">CUST17</td><td style="padding: 2px;">Pre-Launch Hardware Configuration</td></tr> <tr><td style="padding: 2px;">SYS1</td><td style="padding: 2px;">Layered Open Architecture</td></tr> <tr><td style="padding: 2px;">SYS1.2</td><td style="padding: 2px;">Open Interfaces</td></tr> <tr><td style="padding: 2px;">SYS1.2.1</td><td style="padding: 2px;">Waveform API</td></tr> <tr><td style="padding: 2px;">SYS1.2.2</td><td style="padding: 2px;">Device API</td></tr> <tr><td style="padding: 2px;">SYS5</td><td style="padding: 2px;">Common Services</td></tr> <tr><td style="padding: 2px;">SYS6.1</td><td style="padding: 2px;">Data Coding Support</td></tr> <tr><td style="padding: 2px;">SYS6.2</td><td style="padding: 2px;">Modulation Formats</td></tr> <tr><td style="padding: 2px;">SYS6.3</td><td style="padding: 2px;">Coherent/Noncoherent Mode Support</td></tr> <tr><td style="padding: 2px;">SYS9.1</td><td style="padding: 2px;">Reconfiguration APIs</td></tr> <tr><td style="padding: 2px;">SYS12</td><td style="padding: 2px;">Device Diagnostic API</td></tr> <tr><td style="padding: 2px;">SYS20</td><td style="padding: 2px;">Error and Fault Logs</td></tr> <tr><td style="padding: 2px;">SYS21</td><td style="padding: 2px;">Run-Time State Information</td></tr> <tr><td style="padding: 2px;">SYS24</td><td style="padding: 2px;">Control and Management</td></tr> <tr><td style="padding: 2px;">SYS25</td><td style="padding: 2px;">Distributed Functionality</td></tr> <tr><td style="padding: 2px;">SYS29</td><td style="padding: 2px;">RF Module Interface</td></tr> <tr><td style="padding: 2px;">SYS30</td><td style="padding: 2px;">Data and Command Integrity</td></tr> </tbody> </table>	Requirement ID	Name	CUST8	Remote Reconfiguration	CUST9	RF Tunability	CUST17	Pre-Launch Hardware Configuration	SYS1	Layered Open Architecture	SYS1.2	Open Interfaces	SYS1.2.1	Waveform API	SYS1.2.2	Device API	SYS5	Common Services	SYS6.1	Data Coding Support	SYS6.2	Modulation Formats	SYS6.3	Coherent/Noncoherent Mode Support	SYS9.1	Reconfiguration APIs	SYS12	Device Diagnostic API	SYS20	Error and Fault Logs	SYS21	Run-Time State Information	SYS24	Control and Management	SYS25	Distributed Functionality	SYS29	RF Module Interface	SYS30	Data and Command Integrity
Requirement ID	Name																																							
CUST8	Remote Reconfiguration																																							
CUST9	RF Tunability																																							
CUST17	Pre-Launch Hardware Configuration																																							
SYS1	Layered Open Architecture																																							
SYS1.2	Open Interfaces																																							
SYS1.2.1	Waveform API																																							
SYS1.2.2	Device API																																							
SYS5	Common Services																																							
SYS6.1	Data Coding Support																																							
SYS6.2	Modulation Formats																																							
SYS6.3	Coherent/Noncoherent Mode Support																																							
SYS9.1	Reconfiguration APIs																																							
SYS12	Device Diagnostic API																																							
SYS20	Error and Fault Logs																																							
SYS21	Run-Time State Information																																							
SYS24	Control and Management																																							
SYS25	Distributed Functionality																																							
SYS29	RF Module Interface																																							
SYS30	Data and Command Integrity																																							
<b>Comments:</b>																																								

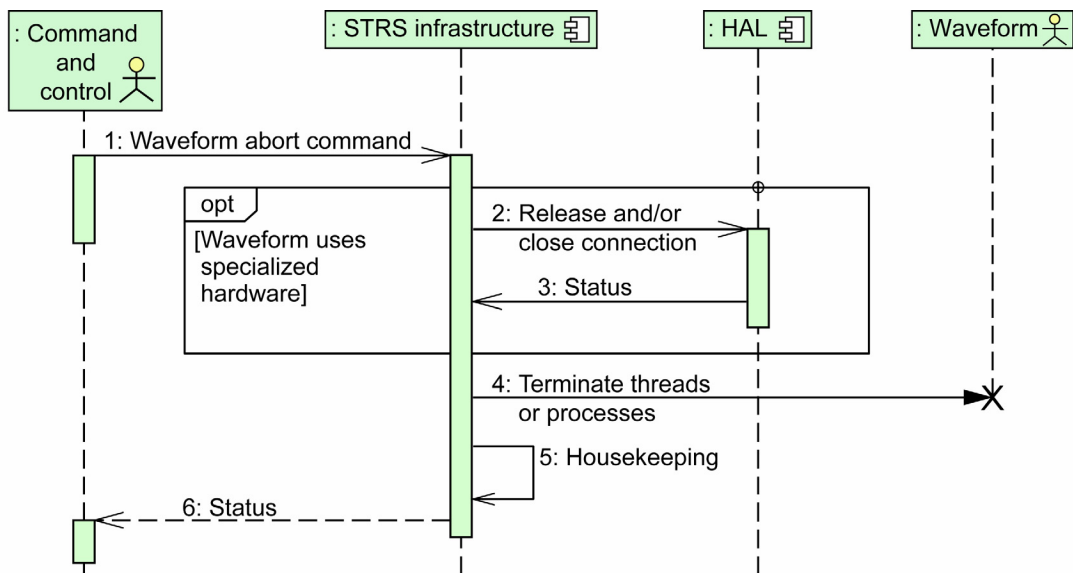


Figure 20.—Waveform abort sequence.

## D.15 Waveform Remove

<b>Use Case Identifier: Waveform Remove</b>	
<b>Description:</b> The STRS radio receives a request from the STRS Command Source to remove a waveform. The specified waveform is deleted from persistent storage.	
<b>External Actors:</b> STRS Command Source	
<b>Related Use Cases:</b> Waveform Upload and Waveform Deallocate	
<b>Precondition:</b> STRS radio ready to accept commands from STRS Command Source. STRS radio is in READY state. Waveform is in the Waveform LOADED or PARTIAL state.	
<b>Triggering event:</b> STRS Command Source requests waveform removal.	
<b>Main Flow:</b>	
<ol style="list-style-type: none"> <li>1. STRS infrastructure receives a validated command from STRS Command Source.</li> <li>2. STRS infrastructure removes a waveform fragment from persistent storage.</li> <li>3. STRS infrastructure marks the waveform as PARTIAL state.</li> <li>4. Repeat from step 2 until entire waveform is removed.</li> <li>5. STRS infrastructure marks the waveform as EMPTY state.</li> </ol>	
<b>Resulting event:</b>	
<b>Post condition:</b> The STRS radio has removed waveform file(s), and the STRS radio continues with its normal operation. STRS radio is in READY state. The waveform is in the Waveform EMPTY state.	
<b>Alternatives:</b>	
(2a) An error is encountered while removing the waveform. Errors are logged.	
<b>Requirements:</b>	
<b>Requirement ID</b>	<b>Name</b>
CUST1	Multiple Waveform Support
CUST8	Remote Reconfiguration
CUST10.1	Data Integrity Checks on Uploads
CUST22	Standard Spacecraft Interfaces
SYS8	Secure Transmissions
SYS9	Upload Interfaces
SYS10	File Management
SYS10.1	Persistent Storage Capacity
SYS24	Control and Management
SYS30	Data and Command Integrity
<b>Comments:</b>	

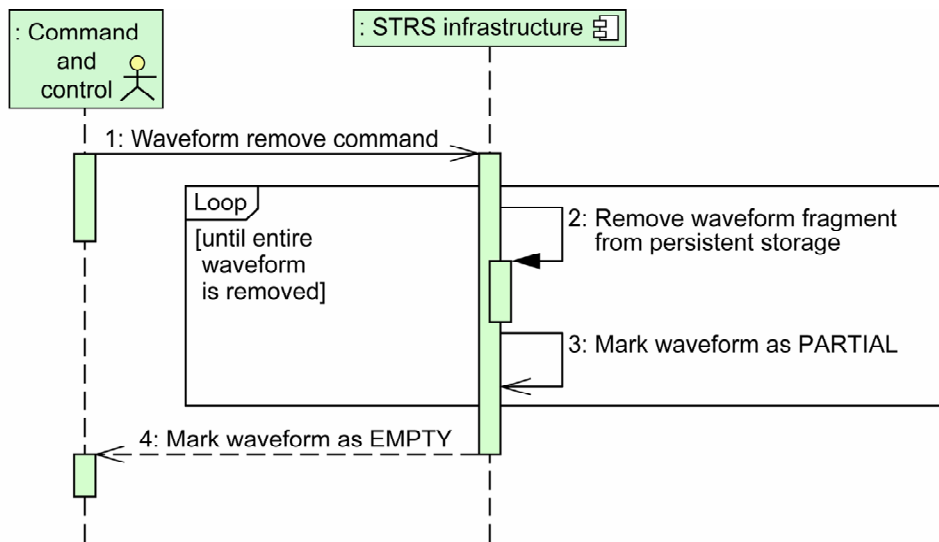


Figure 21.—Waveform remove sequence.



## D.16 Fault Management

<b>Use Case Identifier: Fault Management</b>																																																			
<b>Description:</b> STRS radio reacts to failure or subnominal performance as indicated by diagnostic and performance monitoring																																																			
<b>External Actors:</b> Flight Computer, Fault Source																																																			
<b>Related Use Cases:</b> TBD																																																			
<b>Precondition:</b> All states: power on through full operations <b>Triggering event:</b> detected error																																																			
<b>Main Flow:</b> <ol style="list-style-type: none"> <li>1. Determine error severity and error source.</li> <li>2. Log original error.</li> <li>3. If anomaly is recoverable, corrective action is taken.</li> <li>4. Log any corrective actions.</li> <li>5. Return status to fault source.</li> <li>6. Any change in health is reported to Flight Computer and/or waveform or service.</li> </ol>																																																			
<b>Resulting event:</b> Update STRS radio state of current configuration <b>Post condition:</b> STRS Radio has recovered fully, is degraded, or in fail-safe mode.																																																			
<b>Alternatives:</b> (3–5a) If a recoverable error is repeated treat as FATAL. The number of repetitions required is mission-dependent. (3–5b) If the error is FATAL, the STRS radio is configured as required by the mission. (3–5c) If the WatchDog timer repeats, use fail-safe mode.																																																			
<b>Requirements:</b>																																																			
<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: left; padding: 2px;">Requirement ID</th> <th style="text-align: left; padding: 2px;">Name</th> </tr> </thead> <tbody> <tr><td style="padding: 2px;">CUST5</td><td style="padding: 2px;">Network Performance Monitoring</td></tr> <tr><td style="padding: 2px;">CUST10.1</td><td style="padding: 2px;">Data Integrity Checks on Uploads</td></tr> <tr><td style="padding: 2px;">CUST13</td><td style="padding: 2px;">Automated System</td></tr> <tr><td style="padding: 2px;">CUST13.1</td><td style="padding: 2px;">Power ON Reset Testing</td></tr> <tr><td style="padding: 2px;">CUST13.2</td><td style="padding: 2px;">Last Known Configuration</td></tr> <tr><td style="padding: 2px;">CUST14.1</td><td style="padding: 2px;">Diagnostic Reporting and Action</td></tr> <tr><td style="padding: 2px;">SYS7</td><td style="padding: 2px;">Link Metric Reporting</td></tr> <tr><td style="padding: 2px;">SYS9.1</td><td style="padding: 2px;">Reconfiguration APIs</td></tr> <tr><td style="padding: 2px;">SYS12</td><td style="padding: 2px;">Device Diagnostic API</td></tr> <tr><td style="padding: 2px;">SYS13</td><td style="padding: 2px;">Availability</td></tr> <tr><td style="padding: 2px;">SYS13.1</td><td style="padding: 2px;">Process Protection</td></tr> <tr><td style="padding: 2px;">SYS14</td><td style="padding: 2px;">Default Waveform</td></tr> <tr><td style="padding: 2px;">SYS15</td><td style="padding: 2px;">Power-on Modes</td></tr> <tr><td style="padding: 2px;">SYS15.1</td><td style="padding: 2px;">Default Initialization</td></tr> <tr><td style="padding: 2px;">SYS16</td><td style="padding: 2px;">Redundancy</td></tr> <tr><td style="padding: 2px;">SYS17</td><td style="padding: 2px;">Event Notification</td></tr> <tr><td style="padding: 2px;">SYS18</td><td style="padding: 2px;">Background Monitoring</td></tr> <tr><td style="padding: 2px;">SYS18.2</td><td style="padding: 2px;">Exception Recovery</td></tr> <tr><td style="padding: 2px;">SYS18.3</td><td style="padding: 2px;">SEU Mitigation</td></tr> <tr><td style="padding: 2px;">SYS19</td><td style="padding: 2px;">Default Recovery</td></tr> <tr><td style="padding: 2px;">SYS20</td><td style="padding: 2px;">Error and Fault Logs</td></tr> <tr><td style="padding: 2px;">SYS21</td><td style="padding: 2px;">Run-Time State Information</td></tr> <tr><td style="padding: 2px;">SYS25</td><td style="padding: 2px;">Distributed Functionality</td></tr> <tr><td style="padding: 2px;">SYS30</td><td style="padding: 2px;">Data and Command Integrity</td></tr> </tbody> </table>	Requirement ID	Name	CUST5	Network Performance Monitoring	CUST10.1	Data Integrity Checks on Uploads	CUST13	Automated System	CUST13.1	Power ON Reset Testing	CUST13.2	Last Known Configuration	CUST14.1	Diagnostic Reporting and Action	SYS7	Link Metric Reporting	SYS9.1	Reconfiguration APIs	SYS12	Device Diagnostic API	SYS13	Availability	SYS13.1	Process Protection	SYS14	Default Waveform	SYS15	Power-on Modes	SYS15.1	Default Initialization	SYS16	Redundancy	SYS17	Event Notification	SYS18	Background Monitoring	SYS18.2	Exception Recovery	SYS18.3	SEU Mitigation	SYS19	Default Recovery	SYS20	Error and Fault Logs	SYS21	Run-Time State Information	SYS25	Distributed Functionality	SYS30	Data and Command Integrity	
Requirement ID	Name																																																		
CUST5	Network Performance Monitoring																																																		
CUST10.1	Data Integrity Checks on Uploads																																																		
CUST13	Automated System																																																		
CUST13.1	Power ON Reset Testing																																																		
CUST13.2	Last Known Configuration																																																		
CUST14.1	Diagnostic Reporting and Action																																																		
SYS7	Link Metric Reporting																																																		
SYS9.1	Reconfiguration APIs																																																		
SYS12	Device Diagnostic API																																																		
SYS13	Availability																																																		
SYS13.1	Process Protection																																																		
SYS14	Default Waveform																																																		
SYS15	Power-on Modes																																																		
SYS15.1	Default Initialization																																																		
SYS16	Redundancy																																																		
SYS17	Event Notification																																																		
SYS18	Background Monitoring																																																		
SYS18.2	Exception Recovery																																																		
SYS18.3	SEU Mitigation																																																		
SYS19	Default Recovery																																																		
SYS20	Error and Fault Logs																																																		
SYS21	Run-Time State Information																																																		
SYS25	Distributed Functionality																																																		
SYS30	Data and Command Integrity																																																		
<b>Comments:</b>																																																			

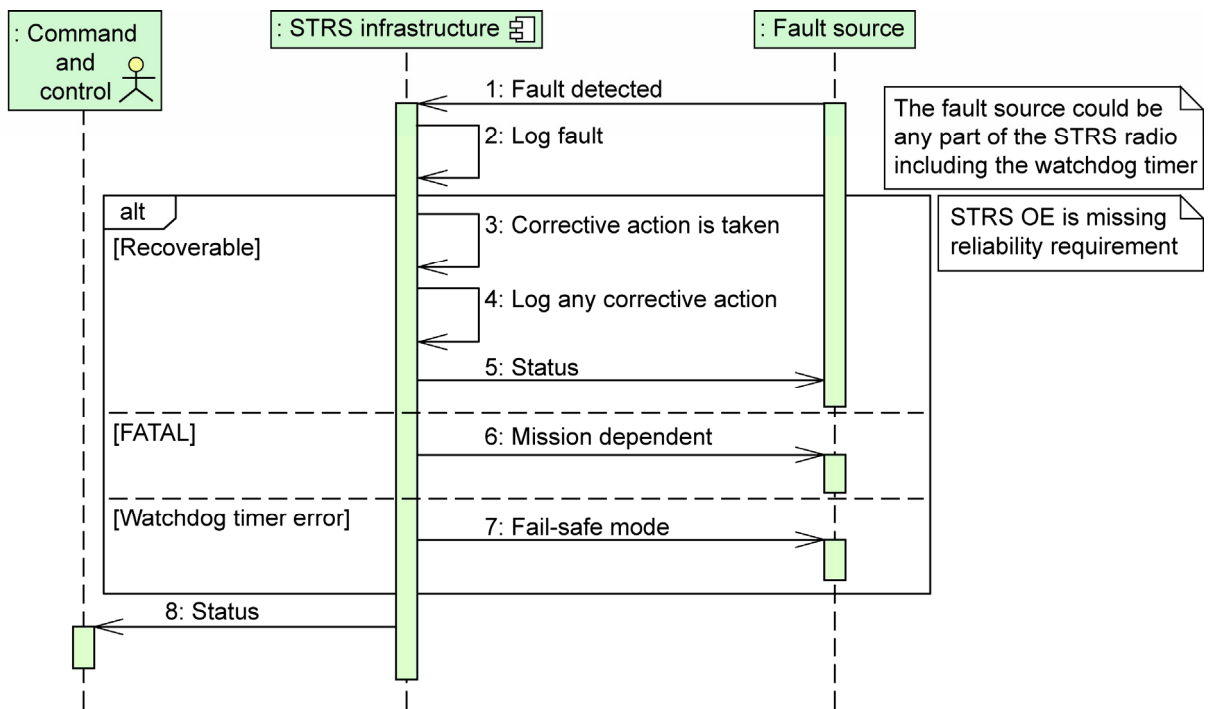


Figure 22.—Fault management sequence.

## D.17 Commanded Built-In Test

<b>Use Case Identifier: Built-In-Test</b>																							
<b>Description:</b> STRS radio receives a request from the STRS Command Source to perform a built-in test (BIT).																							
<b>External Actors:</b> STRS Command Source																							
<b>Related Use Cases:</b> Power On and Waveform Instantiation.																							
<b>Precondition:</b> STRS radio ready to accept commands from STRS Command Source. STRS radio is in READY state. <b>Triggering event:</b> STRS Command Source request Built-In-Test.																							
<b>Main Flow:</b>																							
<ol style="list-style-type: none"> <li>1. STRS infrastructure receives a validated command from STRS Command Source.</li> <li>2. STRS infrastructure requests a BIT from the RTOS.</li> <li>3. RTOS performs BIT.</li> <li>4. STRS infrastructure invokes BIT on its subsystems.</li> <li>5. STRS infrastructure invokes BIT on each Instantiated Waveform.</li> <li>6. STRS infrastructure invokes BIT on each Specialized Hardware module via the HAL and writes telemetry to the Flight Computer interface.</li> <li>7. STRS infrastructure logs results of all BIT.</li> <li>8. STRS infrastructure sends results to the Flight Computer.</li> </ol>																							
<b>Resulting event:</b> Flight Computer notified that the STRS radio is in READY state.																							
<b>Post condition:</b> STRS radio ready to accept commands from Flight Computer. STRS Radio is in READY or STRS FAULT state.																							
<b>Alternatives:</b>																							
<p>(3a) BIT identifies recoverable failure. Mark condition in RAM and proceed to 4.</p> <p>(3b) BIT identifies unrecoverable failure and does not proceed to 4. HW WatchDog Timer resets CPU.</p> <p>(6a) Applicable if Specialized Hardware installed.</p> <p>(8a) If BIT test failed, STRS infrastructure sends to the Flight Computer STRS FAULT state.</p> <p>If the hardware has a fault, the flight computer will recognize that the STRS radio is not communicating and perform a Power Cycle. If state is permanent, the Flight Computer will Power Off module and attempt to use Redundant System.</p>																							
<b>Requirements:</b>																							
<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: left; padding: 2px;">Requirement ID</th> <th style="text-align: left; padding: 2px;">Name</th> </tr> </thead> <tbody> <tr><td style="padding: 2px;">CUST15</td><td style="padding: 2px;">Commanded BIT and status reporting</td></tr> <tr><td style="padding: 2px;">CUST14.1</td><td style="padding: 2px;">Diagnostic Reporting and Action</td></tr> <tr><td style="padding: 2px;">CUST22</td><td style="padding: 2px;">Standard Spacecraft Interfaces</td></tr> <tr><td style="padding: 2px;">SYS5</td><td style="padding: 2px;">Common Services</td></tr> <tr><td style="padding: 2px;">SYS10.1</td><td style="padding: 2px;">Persistent Storage Capacity</td></tr> <tr><td style="padding: 2px;">SYS12</td><td style="padding: 2px;">Device Diagnostic API</td></tr> <tr><td style="padding: 2px;">SYS15</td><td style="padding: 2px;">Power-on Modes</td></tr> <tr><td style="padding: 2px;">SYS15.1</td><td style="padding: 2px;">Default Initialization</td></tr> <tr><td style="padding: 2px;">SYS20</td><td style="padding: 2px;">Error and Fault Logs</td></tr> <tr><td style="padding: 2px;">SYS29</td><td style="padding: 2px;">RF Module Interface</td></tr> </tbody> </table>	Requirement ID	Name	CUST15	Commanded BIT and status reporting	CUST14.1	Diagnostic Reporting and Action	CUST22	Standard Spacecraft Interfaces	SYS5	Common Services	SYS10.1	Persistent Storage Capacity	SYS12	Device Diagnostic API	SYS15	Power-on Modes	SYS15.1	Default Initialization	SYS20	Error and Fault Logs	SYS29	RF Module Interface	
Requirement ID	Name																						
CUST15	Commanded BIT and status reporting																						
CUST14.1	Diagnostic Reporting and Action																						
CUST22	Standard Spacecraft Interfaces																						
SYS5	Common Services																						
SYS10.1	Persistent Storage Capacity																						
SYS12	Device Diagnostic API																						
SYS15	Power-on Modes																						
SYS15.1	Default Initialization																						
SYS20	Error and Fault Logs																						
SYS29	RF Module Interface																						
<b>Comments:</b> All BIT and periodic background monitoring is provided to Flight Computer interface for telemetry.																							

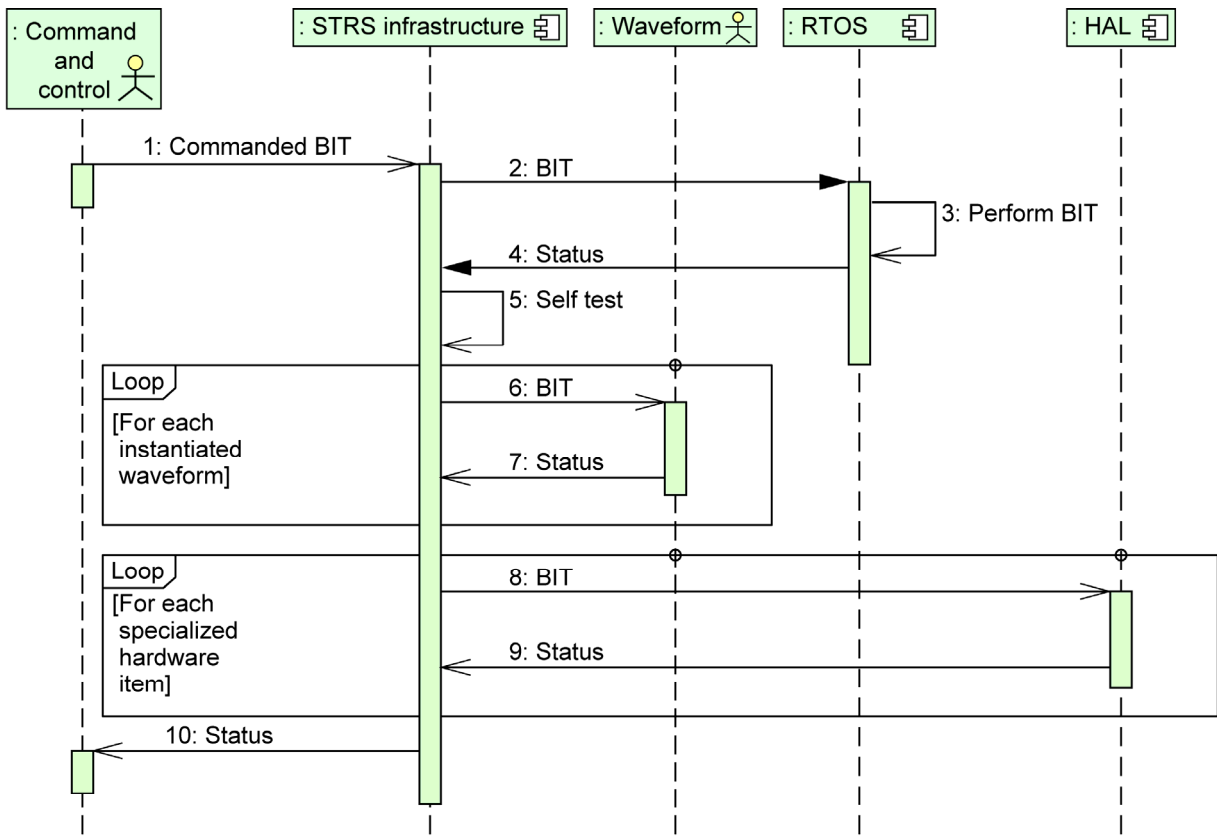


Figure 23.—Commanded BIT.

**REPORT DOCUMENTATION PAGE**

*Form Approved*  
OMB No. 0704-0188

The public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Department of Defense, Washington Headquarters Services, Directorate for Information Operations and Reports (0704-0188), 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to any penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number.

PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ADDRESS.

<b>1. REPORT DATE (DD-MM-YYYY)</b> 01-06-2008		<b>2. REPORT TYPE</b> Technical Paper		<b>3. DATES COVERED (From - To)</b>	
<b>4. TITLE AND SUBTITLE</b> Space Telecommunications Radio System Software Architecture Concepts and Analysis				<b>5a. CONTRACT NUMBER</b>	
				<b>5b. GRANT NUMBER</b>	
				<b>5c. PROGRAM ELEMENT NUMBER</b>	
<b>6. AUTHOR(S)</b> Handler, Louis, M.; Hall, Charles, S.; Briones, Janette, C.; Blaser, Tammy, M.				<b>5d. PROJECT NUMBER</b>	
				<b>5e. TASK NUMBER</b>	
				<b>5f. WORK UNIT NUMBER</b> WBS 439432.04.07.01	
<b>7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)</b> National Aeronautics and Space Administration John H. Glenn Research Center at Lewis Field Cleveland, Ohio 44135-3191				<b>8. PERFORMING ORGANIZATION REPORT NUMBER</b> E-15968	
<b>9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)</b> National Aeronautics and Space Administration Washington, DC 20546-0001				<b>10. SPONSORING/MONITORS ACRONYM(S)</b> NASA	
				<b>11. SPONSORING/MONITORING REPORT NUMBER</b> NASA/TP-2008-214813	
<b>12. DISTRIBUTION/AVAILABILITY STATEMENT</b> Unclassified-Unlimited Subject Categories: 17 and 62 Available electronically at <a href="http://gltrs.grc.nasa.gov">http://gltrs.grc.nasa.gov</a> This publication is available from the NASA Center for AeroSpace Information, 301-621-0390					
<b>13. SUPPLEMENTARY NOTES</b>					
<b>14. ABSTRACT</b> The Space Telecommunications Radio System (STRS) project investigated various Software Defined Radio (SDR) architectures for Space. An STRS architecture has been selected that separates the STRS operating environment from its various waveforms and also abstracts any specialized hardware to limit its effect on the operating environment. The design supports software evolution where new functionality is incorporated into the radio. Radio hardware functionality has been moving from hardware based ASICs into firmware and software based processors such as FPGAs, DSPs and General Purpose Processors (GPPs). Use cases capture the requirements of a system by describing how the system should interact with the users or other systems (the actors) to achieve a specific goal. The Unified Modeling Language (UML) is used to illustrate the Use Cases in a variety of ways. The Top Level Use Case diagram shows groupings of the use cases and how the actors are involved. The state diagrams depict the various states that a system or object may be in and the transitions between those states. The sequence diagrams show the main flow of activity as described in the use cases.					
<b>15. SUBJECT TERMS</b> Space communication; Spacecraft communication; Radio communication systems; Computer architecture; Software; Software engineering; Telemetry					
<b>16. SECURITY CLASSIFICATION OF:</b>			<b>17. LIMITATION OF ABSTRACT</b>	<b>18. NUMBER OF PAGES</b>	<b>19a. NAME OF RESPONSIBLE PERSON</b>
<b>a. REPORT</b>	<b>b. ABSTRACT</b>	<b>c. THIS PAGE</b>			<b>STI Help Desk (email:help@sti.nasa.gov)</b>
U	U	U	UU	52	<b>19b. TELEPHONE NUMBER (include area code)</b> 301-621-0390