Interoperable End-to-End Space Communications Architectures Using CCSDS Building Blocks

Peter M. Shames [1]
Jet Propulsion Laboratory, California Institute of Technology, Pasadena, CA, 91109

End-to-end space communication architectures must connect system elements that may be in space, on the ground in mission operations centers, or are shared assets such as ground communications stations. End-to-end connectivity involves space communications over RF links, but also cross support services, terrestrial network circuits, and a variety of application layer protocols for commanding, telemetry, and mission operations. CCSDS has developed a large suite of interoperable, and cross-supportable, protocols for these purposes. Each of these defines a specific "layer" of functionality, such as: RF modulation, space link error coding, cross support frame delivery, or network layer routing. CCSDS has recently published a Space Communication Cross Support Architecture Requirements Document (SCCS-ARD) that describes how many of these standards fit together and how they are intended to be used. This paper provides an overview of this document, presented so as to explain the concepts so that others may use them. These concepts will be described from several key viewpoints.

I. Introduction

THE designers of all kinds of communication systems are faced with choices as to the interfaces and protocols to use: networked or serial links; copper or fiber; USB 3.0 or Thunderbolt; Wi-Fi or Bluetooth. Designers of space communication systems face similar choices and more, since these systems inherently include elements that may be at a great distance in space (and time) and where round trip light times may be measured in 10's of hours and received signal strength in fractions of a watt. Furthermore, these End-to-End space data systems are inherently systems-of-systems. They are typically composed of spacecraft and mission operations systems (MOS) belonging to one (or more) organizations and multi-mission communication assets belonging to other organizations.

Because of these characteristics it is essential that systems adopt interface standards that support space communications, and that have the features of interoperability and cross support. Such standards allow missions to take advantage of expensive, shared, multi-mission communication assets (DSN, ESTRACK, commercial, …). They may also provide interoperability among spacecraft owned by multiple agencies (e.g. Mars and Lunar exploration). As there is a move toward Solar System Internetworking (SSI) in the (near) future such standards will enable assets in space, on the surface of remote bodies, and on Earth to exchange data at a network level using delay and disruption tolerant networking protocols.

A key question that arises in designing such systems is: "Are there any standard system building blocks or protocol stacks available to build such systems?"

A building-block approach can provide the means for specifying, configuring, and designing interoperable, cross-supportable, end-to-end space communications systems. This approach can enhance system element and component reuse, support operational consistency, take advantage of existing well tested standards and services, and leverage interoperable CCSDS service, data link, and space internetworking protocols and interfaces to reduce cost and risk.

Although technical details will vary from one space mission to another, such as where the spacecraft are destined, what science is to be done, and whether they are robotic or human, there are major elements that tend to be similar: a spacecraft, one or more ground stations, and a mission operations center. Depending upon the mission configuration there may also be science and spacecraft operations centers, involvement of other spacecraft using some data relay configuration, or a need for higher level services like message or file transfer. There may also be considerations like voice and video services for humans, and, in the near future, high rate communications services such as gigabit Ka-band or optical comm. CCSDS has protocols and service interfaces, either in existence or in

[1] Mgr, JPL Data System Standards Program, Interplanetary Network Directorate, MS 301-490.

work, that support all of these features. This paper provides some high level guidance as to how to assemble these eighty or more standards into useful configurations for space missions.

II. Background

The CCSDS has recently published the Space Communication Cross Support Architecture Requirements Document (SCCS-ARD, CCSDS 901.1-M-1) [1] and the companion Space Communication Cross Support Architecture Description Document (SCCS-ADD, CCSDS 901.0-G-1) [2]. This paper is intended to introduce these building block concepts they document in a way that makes them accessible. Most of the figures in this document are reproduced from those standards and are reprinted with permission of the Consultative Committee for Space Data Systems (CCSDS). The SCCS terminology that is adopted here includes names for the major system nodes, the Space User Node (SUN, spacecraft), the Earth Space Link Terminal (ESLT, ground station), and the Earth User Node (EUN, MOC). Other terminology for relay and routing elements is also introduced as needed.

Most of the focus in this paper is on "traditional" single space link missions that include a spacecraft, its MOC, and a ground station. These are called ABA configurations under the assumption that Agency (or organization) "A" will own the spacecraft and its MOC, but Agency "B" will own or provide the ground station. These elements may, of course, all belong to the same space agency, but the configuration remain essentially the same. Relay missions, such as those at Mars or the Moon, and Solar System Internetworking (SSI) configurations are also covered in the SCCS documents [1 & 2], but these are just introduced in this paper in the interest of brevity.

The major service interfaces, including those used for planning, scheduling, configuration, management, data delivery, and reporting on space link communications are described. These service interfaces are offered by the ESLT service providers to the EUN service users on the ground. The are connected using terrestrial Internet interfaces on the ground and using space link protocols to the SUN. These inherently asymmetric, end-to-end, communication paths are produced by the service elements running in the ESLT, which has a "middle-man", service provider, role.

Views are provided to address the kinds of end-to-end configurations that are typical today at for link layer services, and briefly introduces the Solar System Internetwork, including relay and routing architectures using transport and network layer protocols. Missions are deployed today that offer relay services, but these tend to be "hand made" affairs not using any sort of standard architecture. One of the strong motivations for the SCCS standards are to provide guidance for deploying end-to-end relay architectures that have excellent interoperability, re-use, and accountability properties.

The SCCS approach is to define a set of system element building blocks and their associated functions. There is a basic set for ABA style missions, and an extended set for SSI style missions. There are also building blocks for transitional "hybrid" science/relay missions that have primary science goals, but that also have requirements levied to provide relay support for other missions. Multi-mission and multi-agency examples of these sorts of configurations are operational today around Mars, but they do not yet use standard interfaces.

All of these system element building blocks connect at interfaces, sometimes using RF links, sometimes using terrestrial links. At each interface there is a protocol stack that must be described accurately if interoperability is to be possible. For some purposes this stack may be described using a short-hand form, e.g. TC/TM space link (TeleCommand forward / TeleMetry return) [3, 4]. But full characterization, end-to-end, requires knowledge of the whole protocol stack, from application or file down to physical layer, and also knowledge of the ESLT interface stacks, which typically include Space Link Extension (SLE) forward and return frame services [5, 6, 7], Cross Support Transfer Service (CSTS) monitor and tracking data services [8, 9], and the means to request and configure space communications services using Service Management (SM) [10, 11] interfaces. These interfaces are all specified as protocol stack "building blocks", and two of the many possible end-to-end compositions are provided as examples.

The various security standards that may be applied to construct secure interoperable systems are also briefly described. Security, as with many of the other topics that are just touched upon here, including link design, coding and modulation choices, ranging and tracking approaches, retransmission and reliability, is itself a deep subject. This survey paper is not going to treat security in any significant detail and the reader is directed to other sources such as the CCSDS Security Architecture [12] and the CCSDS Guide for Secure System Interconnection [13].

Interested readers are encouraged to delve more deeply into any of these topics using the SCCS documents as a guide to the CCSDS standards that address each of these disciplines in depth. There are many other CCSDS descriptive documents (Green Books) that describe the motivation, rationale and applicability of these standards. And there are a number of other CCSDS recommended standards (Blue Books) and recommended practices (Magenta Books) that provide the technical details for building and operating compliant systems.

III. Service and Interface Interoperability Considerations

CCSDS has developed a large suite of interoperable, and cross-supportable, protocols and other standards. Each of these standards typically define a single specific "layer" of functionality, such as: RF modulation, space link protocols, error coding, cross support frame delivery, or network layer routing. What had not been provided until the SCCS was an architectural view, any sort of building block approach, nor end-to-end views of how all of these many parts fit together to provide solutions for different missions.

The concepts of using generic system nodes in End-to-End space missions deployments and the main kinds of interfaces used among these elements is introduced next. Both "traditional" single space links (called ABA) or a relay or network approach (SSI) may be used.

A. Generic "ABA" End-to-End Configuration

The generic "ABA" configuration, as shown in Figure 1, has the following main characteristics:

- Agency A develops the spacecraft (Space User Node) and the MOC (Earth User Node).
- Agency B owns/operates the ground station (Earth Space Link Terminal).

Main Interfaces:

- Each of these system building blocks has defined interfaces and component behaviors
- The ABA ESLT has a Service Management interface that is used by the EUN to plan, schedule, configure and report on provided space link services
- Terrestrial Service Provider interfaces between EUN and ESLT (SLE and CSTS).
- Space Link Service Provider interfaces between ESLT and SUN (CCSDS space link)
- Each of these interfaces has an associated set of protocols (the stack) and protocol behaviors

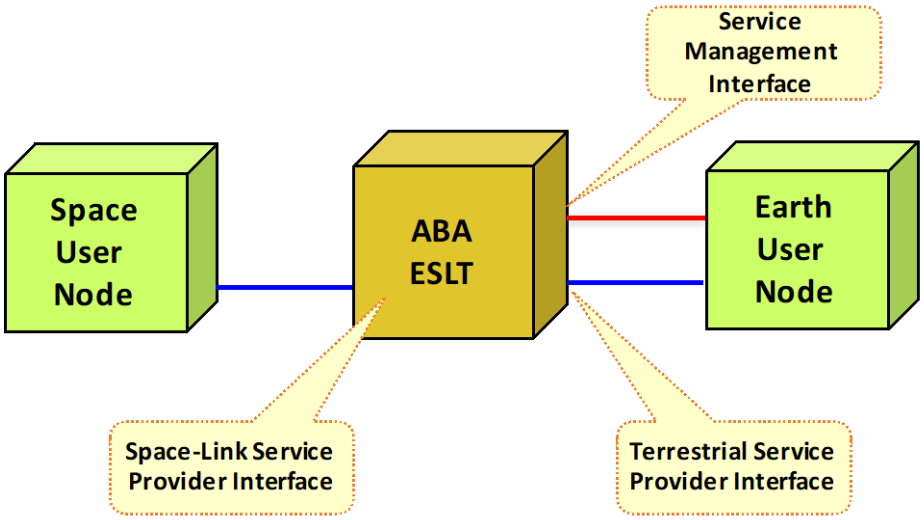

Figure 1. Generic "ABA" Configuration

A mission may utilize ABA ESLT services that are provided by more than one agency, which is why adoption of cross support service interfaces is important. All of these standard services have been recommended by the Interagency Operations Advisory Group (IOAG) and are documented in their Service Catalog, version 1 [14]. Most of the world's civilian space agencies have already adopted these standards, as have many commercial service providers. The EUN is considered to be the service user that has responsibility for coordinating, scheduling, management and control of the ESLT interfaces to the space link but real mission deployments may also include science operations centers, separate mission and spacecraft operations, and/or archives.

B. Generic "SSI" End-to-End Configuration

The generic "SSI" configuration, as shown in Figure 2, has the following main characteristics:

- Agency A develops the SSI spacecraft (Space User Node) and the MOC (Earth User Node).
- Agency B owns/operates the SSI Earth and Space Routing nodes (ERN, SRN).
- Agency C owns/operates the SSI capable ground station (SSI Earth Space Link Terminal).

Interfaces

- Each of these system building blocks has defined interfaces and component behaviors.
- The SSI ESLT has a Service Management interface that is used by the ERN to plan, schedule, configure and report on SSI ESLT provided space link services
- Terrestrial Wide Area Network (WAN) interfaces occur between ERN and ESLT (SLE and CSTS).
- Space Link interfaces occur between ESLT and SRN (CCSDS space link)
- Space Link interfaces occur between the SRN and the SUN (CCSDS space link or prox link)
- Terrestrial WAN component(s) and interfaces to provide Internet services
- Each of these interfaces has an associated set of protocols (the stack) and protocol behaviors

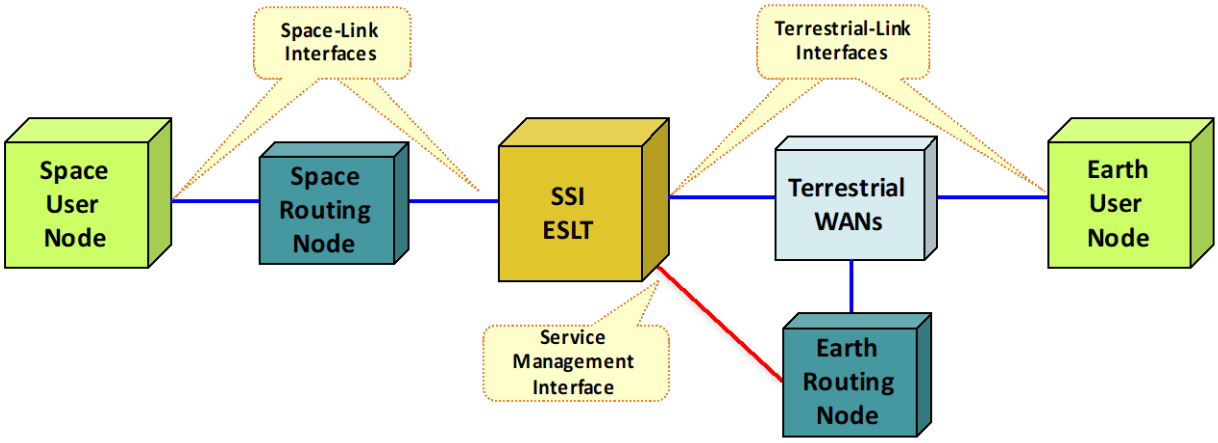

Figure 2. Generic "SSI" Configuration

The mission providing routing services (ERN) may utilize SSI ESLT services provided by more than one agency, and the ERN is considered to be the service user that has responsibility for coordinating, scheduling, management and control of the ESLT interfaces to the space link. As in any internetworked deployment, once these space links are up and operational SSI traffic may flow from the EUN to SUN, as well as among all other nodes. All of these standard internetworking services have been recommended by the Interagency Operations Advisory Group (IOAG) and are documented in their Service Catalog, version 2 [15]. Several of the world's civilian space agencies are planning to adopt these standards and experimental deployments are currently in use. The mission deployment may include science operations centers, separate mission and spacecraft operations, and/or archives.

IV. CCSDS Building Block Approach

The SCCS-ARD document describes service, system, and protocol building blocks based on more than 80 CCSDS standards. In order to provide an understanding of how to use this large set of protocols this section introduces a set of building blocks. These are arranged into related topics, each presented in their own sub-section. Each of these views on the deployments is different, but they all address the same kinds of elements: services, nodes, their functions, and the protocol stacks that are used to communicate among them.

The main topics in this section will be presented as follows:

1. Service interfaces as provided by ESLT (ABA and SSI)
    a. Describe how missions may plan, schedule, configure, use, and monitor cross supported service interfaces
2. Component building blocks, ABA and SSI nodes
    a. Show the major physical elements that are used to compose end-to-end systems
    b. Describe their typical functions and interfaces
    c. Support development of a coarse grained composition of the major system elements
    d. Show the nature of the connections and the high level data flows
3. Interface protocol building blocks for ABA and SSI nodes
    a. Show the primary protocol stack assemblies for ABA and SSI configurations
    b. Show the protocol configurations as they are intended to be used

c. Describe appropriate assemblies for different kinds of services: link layer, radiometric, network layer, application layer, and security

4. Example end-to-End deployments
   a. Use the component and protocol building blocks to demonstrate how to construct typical end-to-end ABA and SSI system deployments

It is worth noting that the focus in this paper (and in the SCCS) is on data transfer and underlying communications services; all mission user applications, using whatever application layer protocols they adopt, are constructed upon these services.

A. Service Interfaces

Service interfaces provide the means for missions to plan, schedule, configure, request, and report on cross supported services. There are basically two kinds of interfaces, those for management of services and those for delivery of data. And there are two basic kinds of deployments, ABA and SSI.

1. ABA Service Interfaces

Before any sort of space communication can take place arrangements must be made for space communications services. These might be private to an agency, or even to a mission, but most often they are shared use assets. The CCSDS Service Management Interfaces (SM) are used by the EUN to plan, schedule, configure, and report on cross supported services. Figure 3. shows these as an interface between the EUN and the ESLT. The exchange of requests and responses between the EUN and the ESLT are defined as a related set of interfaces that operate over terrestrial Internet protocols. They cover the exchange of spacecraft configuration information, required service coverage (period, frequency, mission trajectory or orbit), performance, service agreements, specific requests for and schedules for service.

The other kinds of service interfaces between the EUN and the ESLT are Service Delivery Interfaces. There are two related sets of data delivery interfaces, Space Link Extension (SLE) and Cross Support Transfer Services (CSTS). The space link extension (SLE) services were designed to provide an extension of the space link from the ESLT, which has all of the RF hardware (antennas, transmitters, receivers), to the user. These are the primary interoperability interfaces for data transfer over space links from the EUN to the ESLT.

1. SLE services include: forward CLTU, return all frames, and return channel frames.
2. CSTS services include: an extensible framework, and services for tracking data, monitor data, service control, and a new forward frame service, all built on that framework.

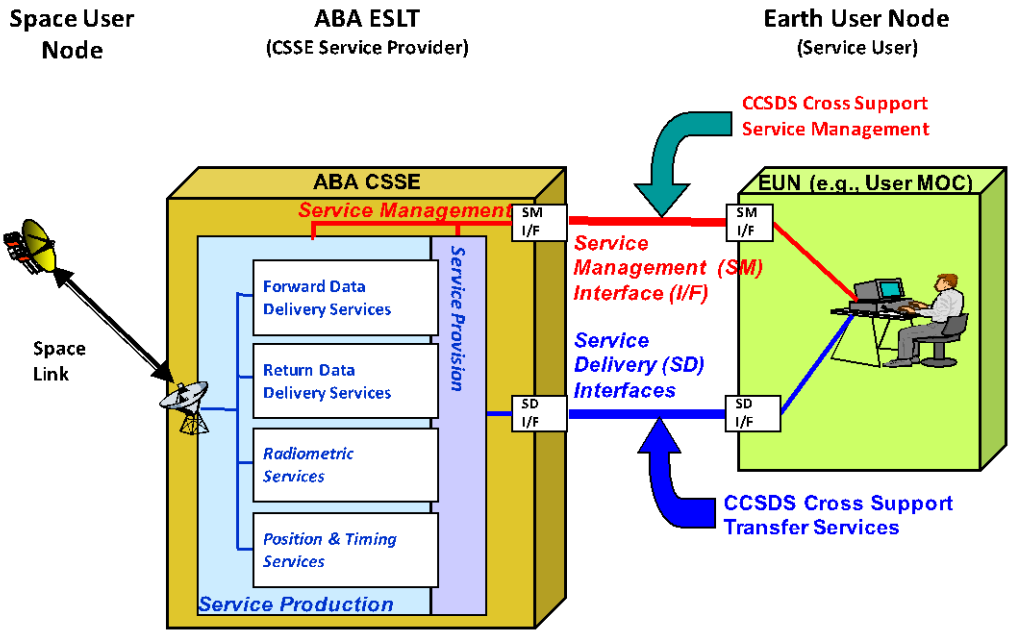

Figure 3. ABA Service Interfaces

Missions may use any or all of these services, and agreements for their configuration and use will be negotiated using the SM interfaces. These services all are designed for real-time delivery of data, but they also have off-line, after the fact, data delivery options. It is a peculiarity of space link performance and terrestrial bandwidth costs that near Earth missions, in particular, may be able to get data from the spacecraft to the ground at much higher rates (>1 Gbps) than it is possible to deliver it from the ESLT to the users on the ground. Off-line delivery is very useful in these circumstances.

In addition to the essential services for modulating and transmitting data in the forward direction, and for demodulating, decoding, and transferring data in the return direction, the ESLT also provides return of radiometric tracking data, monitor data describing the state of service production, and data that is time tagged and annotated as to the quality of the link and the signal processing. An ABA ESLT may also provide other services like Delta-Differential One-way Ranging (D-DOR) [16] or CFDP file delivery [17].

2. SSI Service Interfaces

For SSI deployments, as shown in Figure 4., these same Service Management Interfaces (SM) and Service Delivery interfaces are expected to be used between the ERN and the SSI ESLT. The ERN uses SM to plan, schedule, configure, and report on the cross supported service interfaces it requires to communicate to the SRN. In SSI deployments the ERN "owns" the SM interfaces to the ESLT, as well as the space link interfaces from ESLT to SRN.

The same is true of Service Delivery Interfaces (SLE, CSTS) for the SSI deployment as well as SSI store and forward routing services. The ERN uses the SLE and CSTS data delivery interfaces when it requires link layer access to the SRN. At the link layer the ERN to SRN connections look like an ABA mission. Depending on the design of the relay services this link layer access may just be used for control and monitoring of the SRN, or it may also be used for data transfers. At the same time, once the SRN is up and is running SSI layer services, this internetworking layer may be used for all communications between the ERN and the SRN. At the network layer the EUN and ERN use SSI services end-to-end to route data through the SSI ESLT and SRN, and all may operate as SSI nodes.

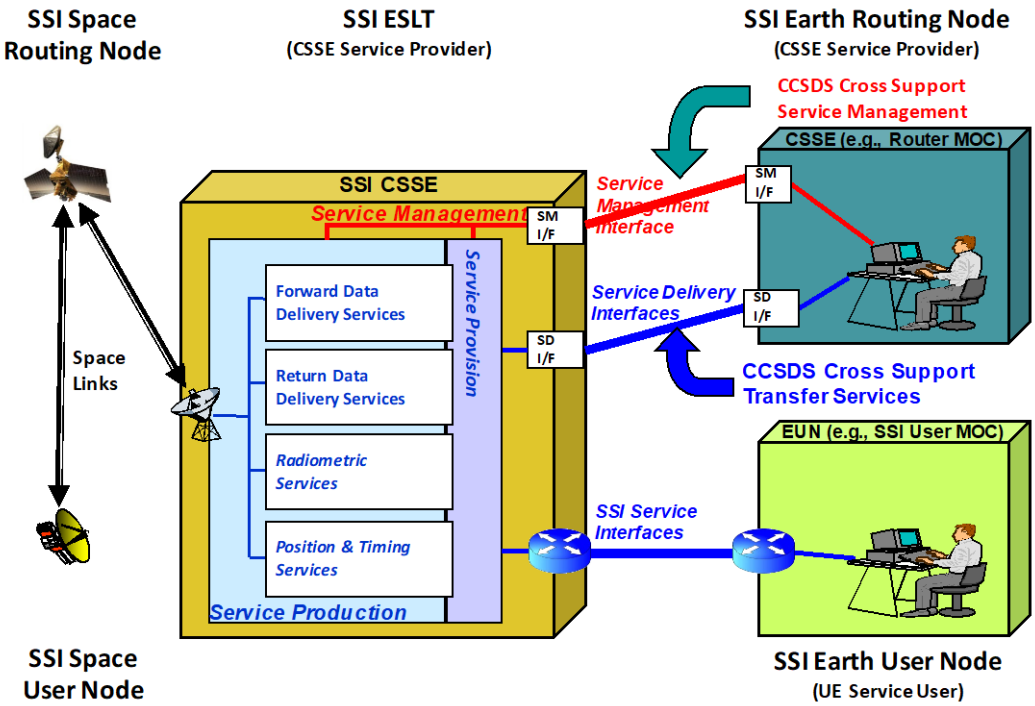

Figure 4. SSI Service Interfaces

The services that are to be offered by a fully compliant SSI ESLT include routing in the node, and this implies the ability to accept SSI (DTN bundle or IP [18, 19]) traffic and to merge it into the space link frame stream. And this requires the functions to create space link frames from whatever DTN bundles or IP data are allowed to be

processed, to encode these frames, and to multiplex them into the frame stream being sent to the SRN. The SRN itself must also include these routing, multiplexing, and forwarding functions.

During the period while the SSI is being incrementally deployed, it will initially supported by hybrid science / routing nodes. Agreements for cross-support will likely be done on a case by case basis between users and providers. This is expected to evolve over time and a vision of what this internetworked future will look like is provided in the SSI Architecture document [20]. One important aspect of this approach is to consider how to transition from ABA to SSI deployments and how to integrate ABA end nodes once the SSI is deployed. The service functions of SRN nodes are described in the next sub-section.

B. Component Building Blocks

The component building blocks, otherwise called nodes, are modeled as physical system elements that support a set of functions. These nodes are of two major types: ABA and SSI. This sub-section introduces the major nodes that are used to compose end-to-end systems. Only the typical functions and interfaces that are relevant to space communications will be described, but each of these nodes may include other functions, such as science planning and processing, or spacecraft operations and execution functions, but they are left out in these diagrams for simplicity's sake. Where thse funcitons do appear they are just shown as "Application". The intent of this somewhat coarse grained building block approach is to capture the composition of the major system elements and to show the nature of the connections and the high level data flows.

The ABA and SSI nodes share certain functions and interfaces, but there are also several that are different, and hybrid science / routing nodes may have features of both. All of the identified node types are listed below, but only a sub-set of them are included here, so as to introduce the overall concepts.

1. ABA Component Building Blocks

The identified set of ABA Component Building Blocks are these:

- ABA Earth Space Link Terminal
- ABA Earth User Node
- ABA Space User Node

Figure 5. shows a generic version of an ABA node. It includes functions for link layer processing, data forwarding (assumed to be a simple "path" service), element management (which may not be either standardized or even explicit), a data store of some sort, and zero or more applications. As noted in the previous section on services, some nodes, like an ABA ESLT, may also include functions for planning, scheduling, and configuring services (element management) and for data delivery (data forwarding). All nodes contain some sort of element management functions, but in many cases there may not be an explicit function identified, and the selection of management functions options may be built in at compile time, or configured via a table loaded at run time.

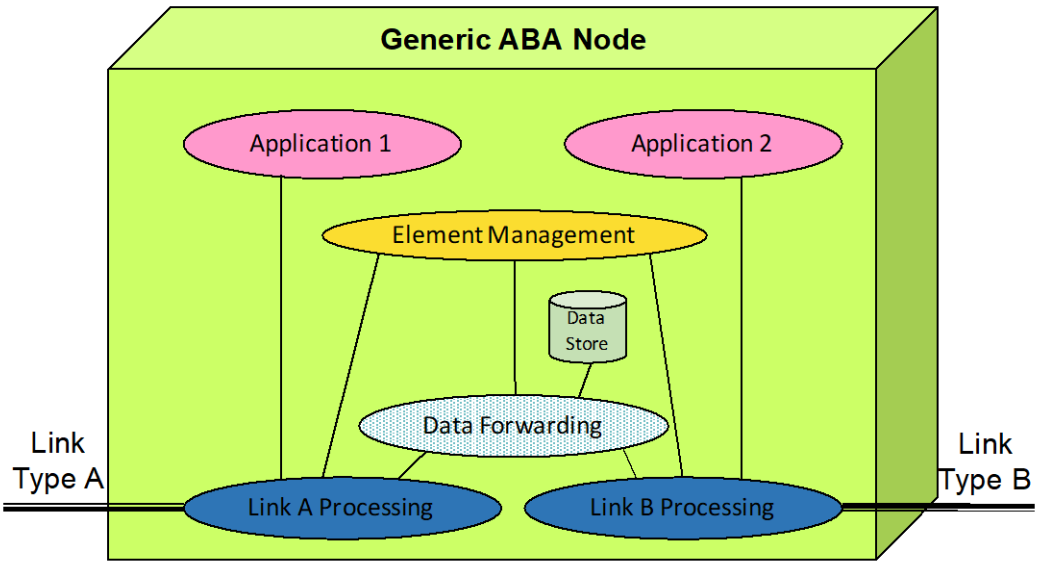

Figure 5. ABA Generic Component

Each ABA node will usually have some specialized characteristics, as described below:

- Not all ABA building blocks will include all functions, user building blocks are often "one-sided"
- The functions for link layer processing will be specified by the requirements from the selected link layer protocols and related standards
- The data forwarding and data storage functions may be defined, but may not necessarily be in a fully specified form
- Some "data path" functions, like that specified in CCSDS Space Packet protocol (SPP) are really just defined by a tag field, the definition of the "function" is entirely dependent upon implementation choices
- Some functions, like element management, may be present and well specified in some nodes, but not specified or use interfaces that are not standardized in others
- Application functions may or may not be present, but all are expected to use the ABA link layer functions

2. SSI Component Building Blocks

The identified set of SSI Component Building Blocks are these:

- SSI Earth Space Link Terminal
- SSI Earth Routing Node
- SSI Earth User Node
- SSI Space Routing Node
- SSI Space User Node
- Wide Area Network (WAN) Routing Node
- Planet Space Link Terminal
- Hybrid Science / Routing Node

Figure 6. shows a generic version of an SSI node. In addition to the functions of an ABA node, that are essential to provide the underlying link layer functions, it also includes functions for network layer processing, routing (defined to be real routing), network management (including routing table updates, which may or may not be standardized), and zero or more applications.

Some nodes, like an SSI ESLT, may include functions for frame creation and encoding, frame merging, as well as others like file and/or message layer functions. Some nodes, like WAN routing nodes, may only include the network layer functions and below, with no upper layer services aside from network management. And some SSI nodes, at the "edges" of the SSI, will need to include specialized last-hop and first-hop functions in order to service ABA nodes that wish to use the SSI "backbone" to transfer data. These last/first hop services are also designed to be used to "reboot" SSI nodes when they require it.

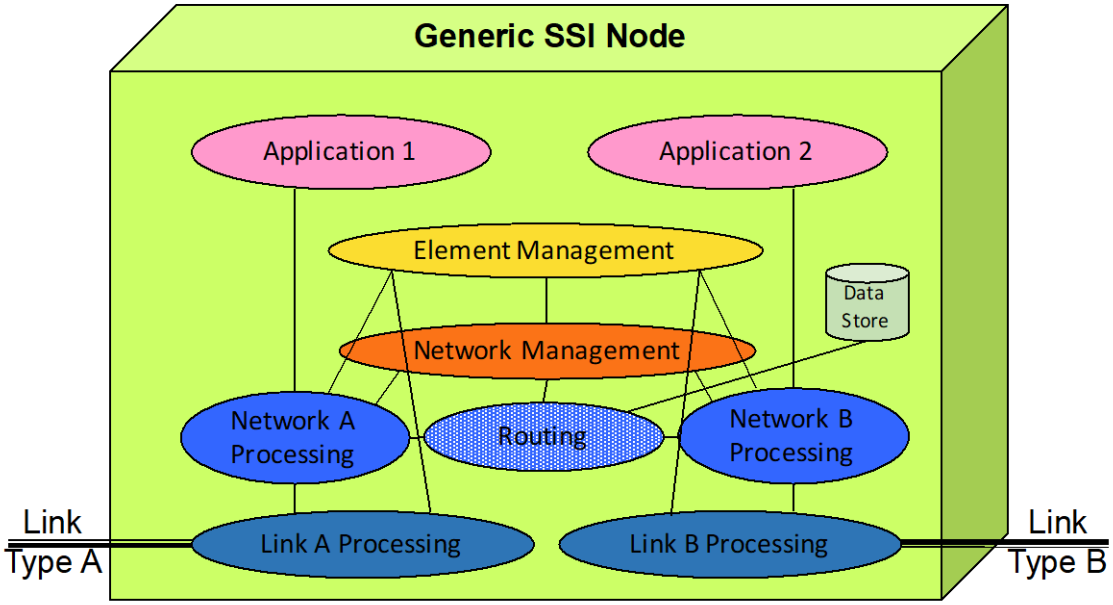

Figure 6. SSI Generic Component

Each SSI node may have specialized characteristics, as described below:

- Not all SSI building blocks will include all functions, user building blocks are often "one-sided"
- The link layer functions are essentially the same as for ABA configurations, with the exception of the ESLT frame creation, merging, and encoding functions
- The SSI network management and routing functions are specified in the relevant SSI standards, some of which are still in development (marked as [Future] in the SCCS-ARD)
- Some functions, like element management and cross-domain coordination, will be present, but are presently only described in the SSI conceptual architecture
- Application functions may or may not be present, but all are expected to use the SSI internetworking functions; some may still use ABA style link layer functions

C. Protocol Building Blocks

With a similar motivation as component building blocks, protocol building blocks are assembled sets of appropriate specifications modeled as a stack of protocol elements that implement an interface. In some cases these are only partial sets that can themselves be combined in useful ways. As with the component building blocks they are of two major types: ABA and SSI. This sub-section introduces the protocol stacks that are used to specify the interfaces on the different node types. Only the typical protocols and interfaces that are relevant to space communications will be described. There may also be "Application" layer protocols, but these are not shown. The intent of this protocol building block approach is to support specification of the major interoperable interfaces for the system nodes and to show the technical protocol stack configurations that support the interconnections among nodes.

The ABA and SSI stacks share certain protocols and interfaces, but the SSI nodes add a number of protocols for internetworking as well as others that provide higher layer protocol services. Many of the possible protocol stack types are included here as examples, but there are many other combinations of protocols, and selections of options, that are possible. An exhaustive treatment is impractical, but a useful and complete enough subset is provided here to permit an understanding of the approach and the range of options.

1. ABA Protocol Building Blocks

These CCSDS ABA protocol building blocks show various stack configurations as they are intended to be used. Many of these are specific for only one kind of node, EUN, SUN, or ESLT. This is largely because these stacks are specialized for the types of nodes and many are also inherently asymmetric. The traditional CCSDS protocols differentiate forward from return flows for a variety of reasons, some of which just have to do with the physics of space communications. Spacecraft, for obvious reasons, cannot carry 34M antennas and powerful transmitters with 10's of Kw of power, but these are both realistic deployments on the ground. As a result, forward links and return links tend to have quite different performance and functional characteristics, and the earliest protocol specs were been designed to work within these constraints. Some of this protocol asymmetry is going away with the use of protocols like AOS and Proximity-1 being used in symmetric, forward and return, deployments.

This sub-section describes appropriate assemblies intended for different kinds of protocol layers: physical (including modulation), link layer (including error correcting coding and synchronization), radiometric, application layer, and security. These stacks are designed to be used as the interface bindings for the different services and functions within the nodes that were described in the previous sub-sections. The descriptions included here are brief. For a fuller understanding and explicit references for all the stack elements refer to the SCCS-ARD [1].

The following diagrams include:

- ABA Earth user protocol stacks, forward and return (typically asymmetric)
- ABA ESLT protocol stacks
- ABA Space user protocol stacks
- Specialized "round trip" protocol stacks such as radiometric tracking
- An example of a security protocol deployment
- Many other examples, including security, are in the SCCS-ARD

Figure 7. shows the ABA service provider forward and return building blocks that are intended for deployment within an ESLT. The right hand sides describe the terrestrial interface bindings offered to an EUN. The left hand sides describe the space link bindings offered to an SUN. Note that the details of the right hand side stacks below TCP/IP are left out for this purpose. While a knowledge of what is actually used in a specific deployment is essential, Cat-6 Ethernet, 802.11AC Wi-Fi, fiber optics, or whatever, for our purposes knowing that SLE runs over TCP/IP and not UDP/IP is sufficient.

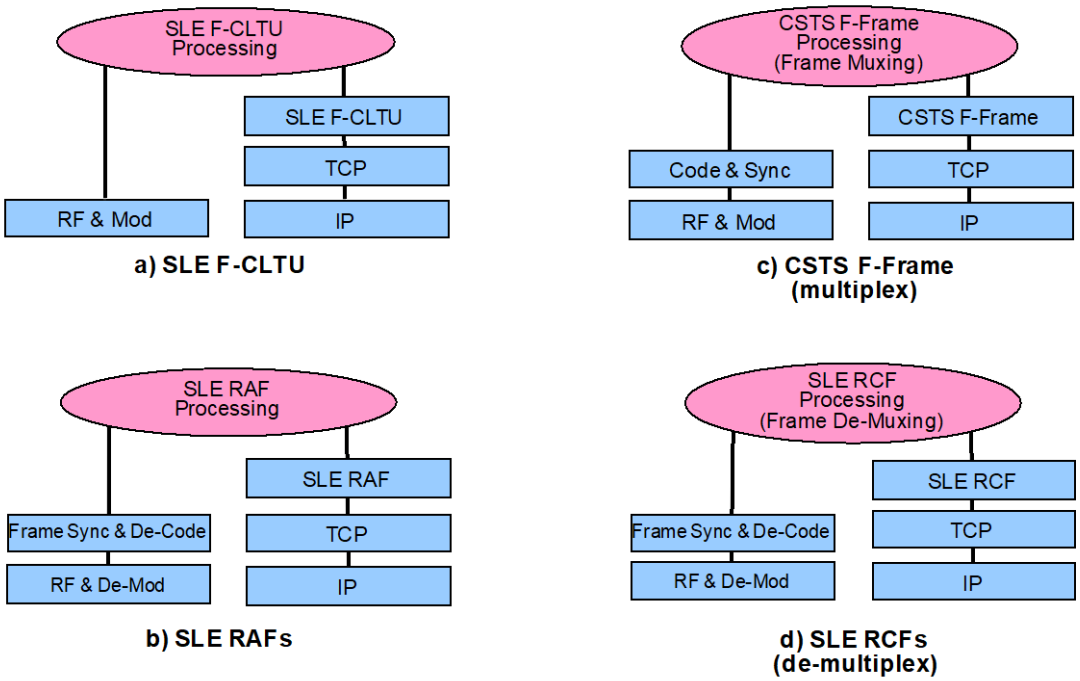

Figure 7. ABA Service Provider Protocol Building Blocks

Figure 7a shows the forward CLTU service, which is the one typically in use today. This requires the EUN to construct its data stream (say packets in TC frames), to encode them, and to send these Command Link Transmission Units (CLTU) in a form ready to be radiated. In Figure 7c the CSTS Forward Frame service [21] is shown. It is of interest in part because it offers a completely symmetric frame service to that shown in Figure 7b and it also has all the features required to support future SSI services. F-Frame accepts use data frames as does the encoding and multiplexing of frames from multiple sources. As such including this service protocol configuration would be ideal for a new ELST that is to provide SSI services in the future.

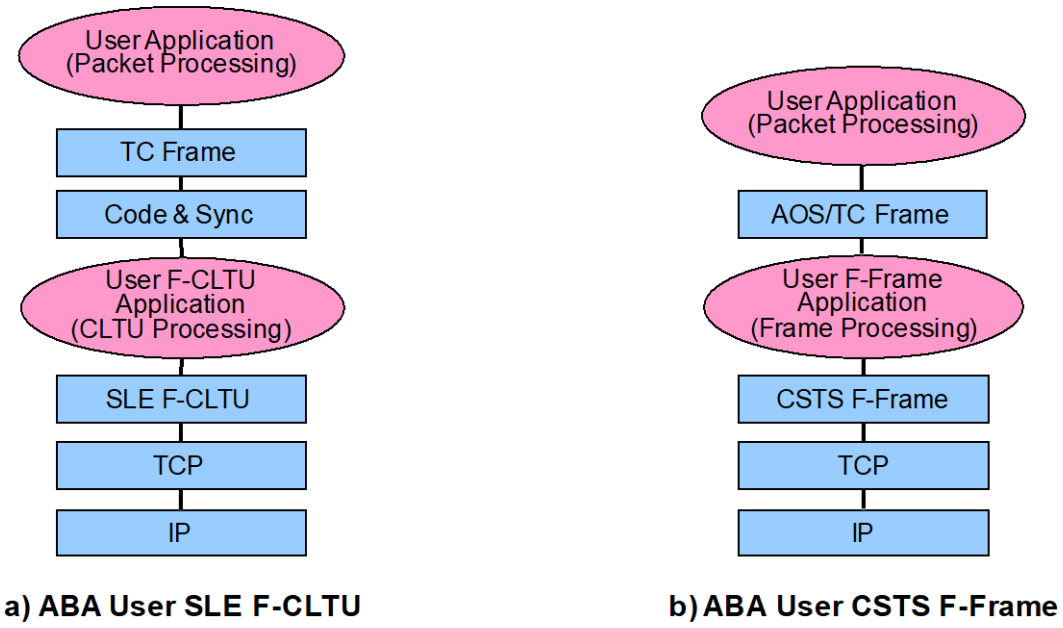

Figure 8. ABA Service User Protocol Building Blocks

Figure 8. shows the ABA service user forward building blocks that correspond to the Figure 7 ESLT. Both F-CLTU and F-Frame versions are shown. The return versions corresponding to figure 7b and 7d are not shown, but these are roughly parallel construction to figure 8b, returning de-coded frames.

Figure 9. shows the ABA ESLT service provider and service user interfaces for Service Management. These interactions are defined to be exchanges of standard XML formatted navigation data [22] using web services, HTTP REST-style [23], protocol interfaces. This is an example of using terrestrial protocols where they are appropriate. Figure 9b also shows the use of one of two navigation data standards (NDM or ODM) for encoding the orbit or trajectory data necessary to point the ESLT at the spacecraft needing services.

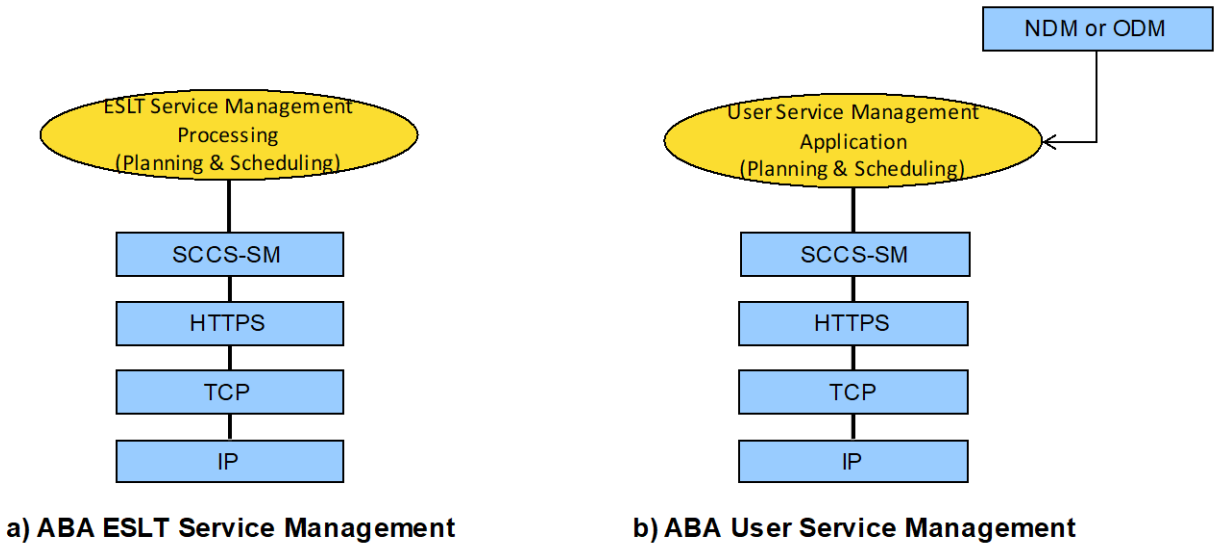

Figure 9. ABA Service Management User Protocol Building Blocks

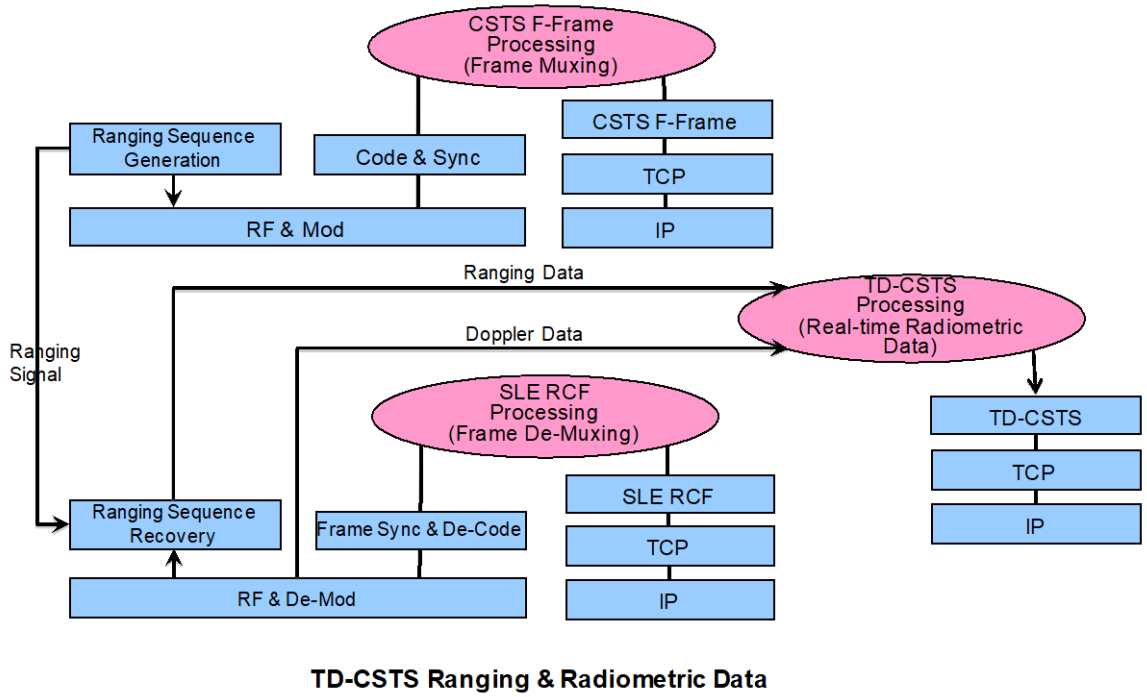

Figure 10. ABA Tracking Data End-to-End Protocol Building Blocks

Figure 10 shows both the forward and return protocol stacks that are involved in providing range and Doppler data. Radiometric tracking requires sending a ranging signal out to the spacecraft that is then returned and processed

on the ground. The resulting range and Doppler data measurements are returned to the user via the TD-CSTS service protocol.

As was mentioned earlier, various security options are available for use where these are required. Figure 11. shows a possible use of the CCSDS authentication, or authenticated encryption in the space data link as the means for securing the command path using frames or files.

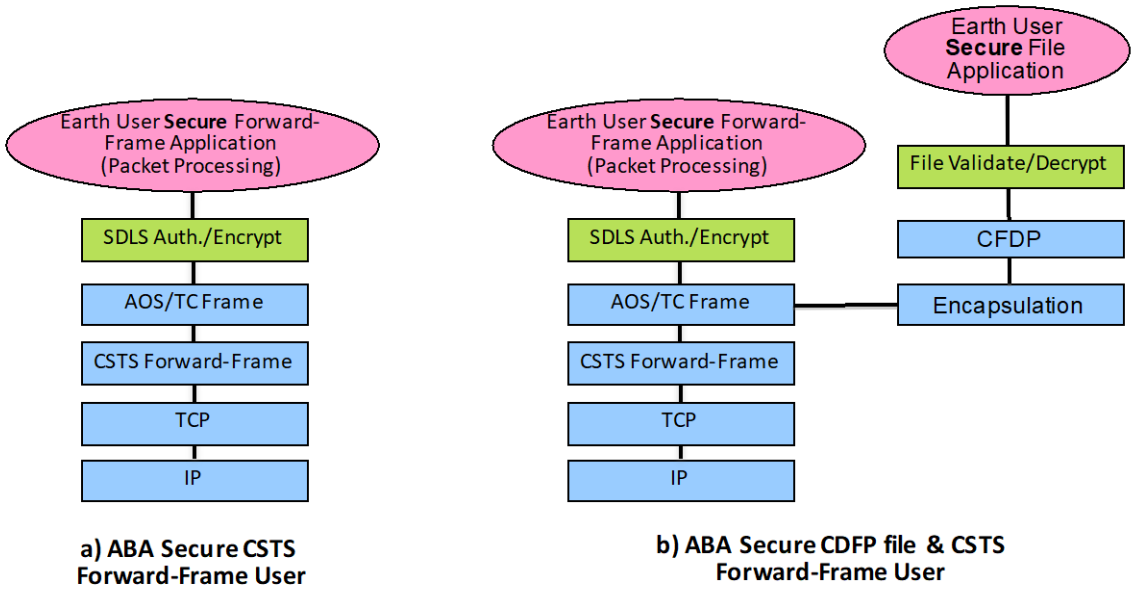

Figure 11. ABA Link & File Layer Security Protocol Building Blocks

Figure 11b. also shows the use of encryption to secure a whole file of data. This may be used by itself or in combination with link layer security protocols.

2. SSI Protocol Building Blocks

The following protocol building blocks show a few of the possible CCSDS SSI protocol stack configurations as they are intended to be used. This sub-section only describes a limited number of assemblies for SSI services: examples include the SSI EUN use of bundle processing and CFDP above the network layer, the protocol stacks required in the SSI ESLT, and examples of the protocol stacks in the SRN and SUN. These SSI stacks are designed to be used as the interface bindings for the different services and functions within the SSI nodes that were described in the previous sub-section. These descriptions are brief and cannot be all inclusive because there are so many possible permutations in any networked deployment. For a fuller understanding and explicit references for all the stack elements refer to the SCCS-ARD.

Appropriate assemblies for different kinds of SSI services as provided; specifically the network layer that uses underlying link layer protocols and the application application layers. The DTN BP network protocol is used end-to-end, and this is an essential feature of SSI deployments. IP may be used in some deployments where the round-trip light time is less than 1-2 seconds and the connectivity is continuous, but DTN protocols do not have those constraints and may be used everywhere. The underlying link layer protocols will change as necessary to accommodate long haul space links, proximity links between orbiters and landers, or terrestrial segments that run as an overlay network, effectively using the Internet for link layer connectivity between nodes acting as BP bundle agents.

The following diagrams include:

- SSI Earth user and relay protocol stacks, forward path (return is roughly symmetric)
- SSI ESLT protocol stacks showing frame creation, merging, and encoding in the ESLT
- SSI Space relay and user protocol stacks running over CCSDS long haul and proximity links
- Many other examples, including security, are in the SCCS-ARD

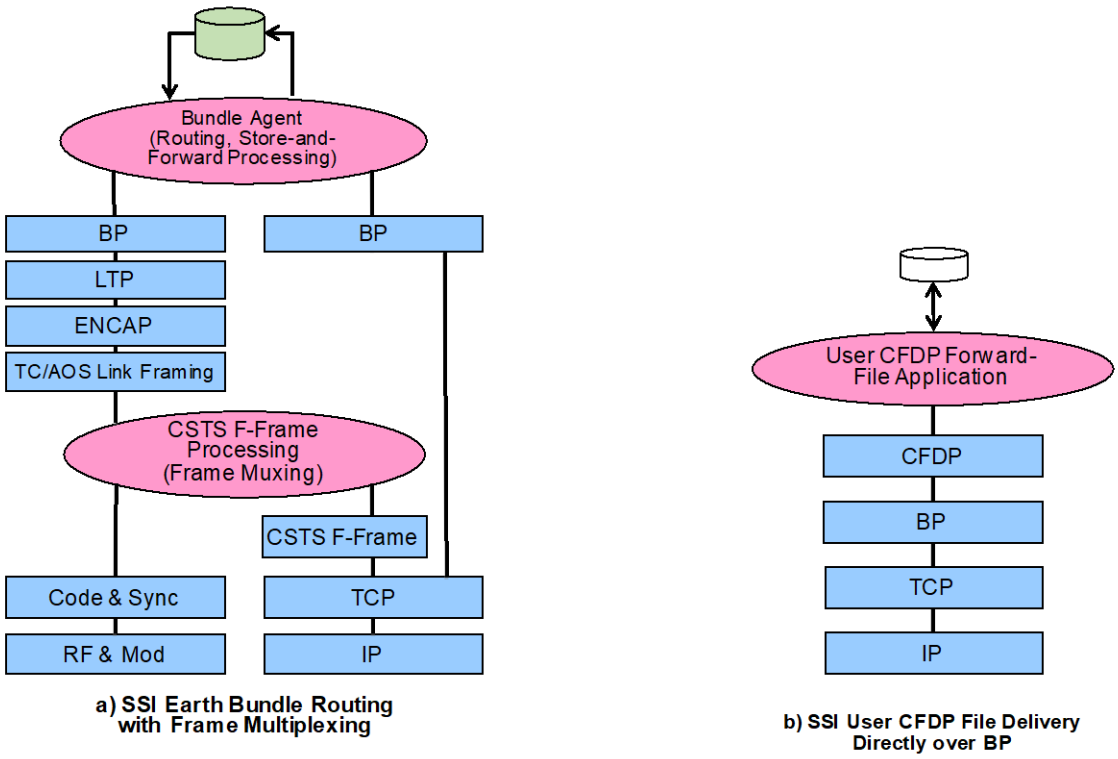

Figure 12. SSI User & Service Provider Protocol Building Blocks

Figure 12b. shows the SSI user protocol stack, with the Internet layer, BP, running over TCP/IP. This is sometimes referred to as an "overlay network" since it is really one network layer (BP) running over another network later (TCP). Figure 12a shows the protocol stack configuration within the SSI ESLT. This uses CSTS F-Frame, as was shown in Figure 7c, but adds the BP layer and the associated Bundle Agent with store and forward processing functions. It also shows the use of LTP [24], which provides a retransmission layer for link layer reliability, and ENCAP [25] as the means to insert BP into the forward space link.

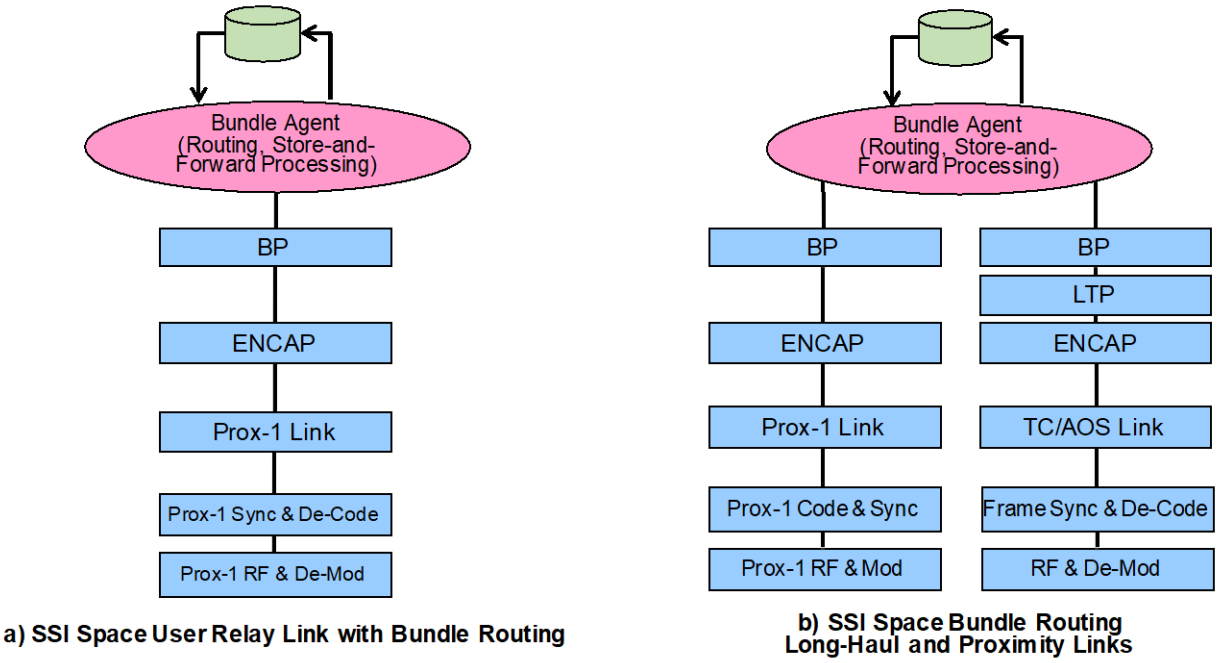

Figure 13. SSI Space User & Bundle Routing Protocol Building Blocks

Figure 13. completes the protocol stacks required for a simple SSI end-to-end path that uses an intermediate SSI Space Routing Node (SRN) to communicate with the SSI Space User Node. The SRN stack shown in Figure 13b includes the BP network layer and the associated Bundle Agent with store and forward processing functions. It also includes two different link layers, LTP and TC or AOS [26] as long haul space links to reliably connect to the SSI ESLT, and Proximity-1 [27], which itself offers reliability, to connect to other landed or orbiting assets. The SSI SRN may, of course, also include science applications and other functions peculiar to its mission. And those functions on the SRN may use the SSI layers to communicate or they may tap into the protocol stack at the link layer, but this is not shown here.

Figure 13a shows a representative protocol stack for an SSI SUN that connects over Prox-1 to the SSI SRN protocols and services shown in figure 13b. The SSI SUN stack just shows BP, but it could include the CFDP or AMS [28] protocols running over BP, or other application layer protocols as required by the mission. Not shown in these figures are the last-hop or first-hop services that would be run in the SSI SRN in order to re-boot the SSI SUN or to acquire tracking or other open loop data from the SUN. These capabilities are described in the SCCS documents.

D. End-to-End Deployments

Once all of the service, node, and protocol building blocks have been defined it is possible to construct a large number of possible end-to-end deployments that may be composed of these elements. Much like Lego™ blocks these building blocks can be used to construct a wide variety of mission deployments. Only two such configurations are presented here, one ABA and one SSI, but many more are possible. A number of other ABA and SSI examples are provided in the SCCS ADD and ARD documents.

1. ABA End-to-End Deployments

The following example shows three ABA Component building blocks and also a typical protocol stack building block configuration for an end-to-end deployment using a very "traditional" forward TC protocol stack. Figure 14. shows the EUN node sending TC frames to the SUN by using the ABA ESLT F-CLTU service interface. Of course, these TC frames might contain user data in the form of CCSDS Space Packets, or CFDP files, or forward data in a number of other standard or private forms.

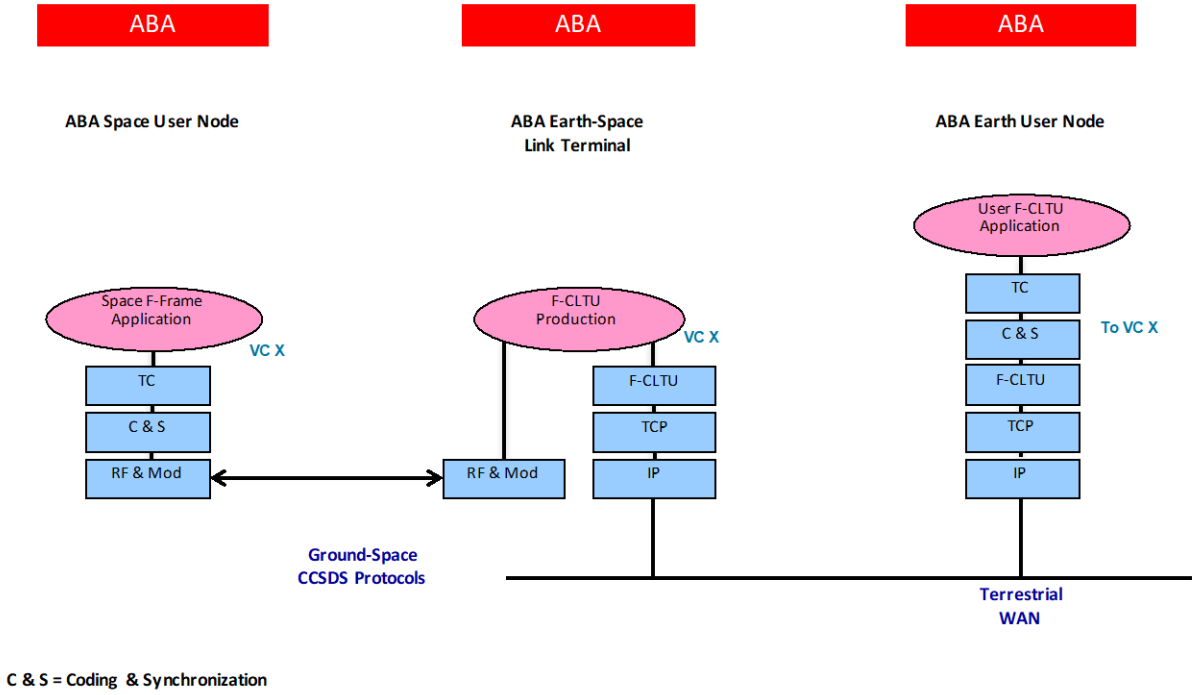

Figure 14. ABA End-to-End Deployment Example

Similar end-to-end deployments for return flows, radiometric processing, file delivery, and secured services are also possible, but all of these would use these same end-to-end configurations. The one exception is the AOS forward protocol, which requires the use of the CSTS F-Frame service in the ABA ESLT, as shown in Figures 7c and 8b.

These are not shown here, but SLE return flows, Service Management (SM), would also be required, and use of radiometric or D-DOR services, or of monitor data or service control [29], would require CSTS services to be used between the EUN and the ESLT, as described earlier.

2. SSI End-to-End Deployments

The use of building block nodes and typical protocol stack building block configurations for an SSI end-to-end deployment are shown next. This example uses an AOS forward protocol stack and also CFDP end-to-end file delivery, from EUN to SUN, running over BP. The flow also uses an SSI ESLT and Earth and Space Routing Nodes. Figure 15. shows the SSI EUN node running CFDP over BP to send a file to the SSI SUN via a space relay node. The file might contain any sort of user data, and other protocols could also be used over BP in addition to CFDP. Which application layer service protocols to use is strictly a mission choice and no changes are required in the BP end-to-end path to accommodate them.

This SSI example also shows the possibility of both a forward command path using space packets and a forward file path from the ERN to the SRN. Once the space link from the SSI ESLT to the SRN is operational it is possible to exchange many different data formats with many different nodes, using the BP network layer.

The ERN plays a role in providing services to the EUN, but it may not necessarily participate in the actual transfer of data from the EUN to the SUN. The BP data may flow direct from the EUN to the SSI ESLT, and it may be stored there until the ESLT to SRN space link is available. These flows and the routing of data are all managed by the protocol layers. While most of the core SSI protocols and services are available today this specific example requires the use of protocols like CSTS F-Frame and BP routing and network management protocols that are still in development within CCSDS. In the SCCS documents these are labelled as [Future], but they are all in active development and should emerge as full standards in the next 2-3 years.

Similar examples for return flows and secured services are also possible, but these are not shown here. As mentioned in Section III.C.1 security protocols may also be deployed where needed. These could include the same sort of user data encryption or authentication that was shown in Figure 11. or network layer security could be employed.

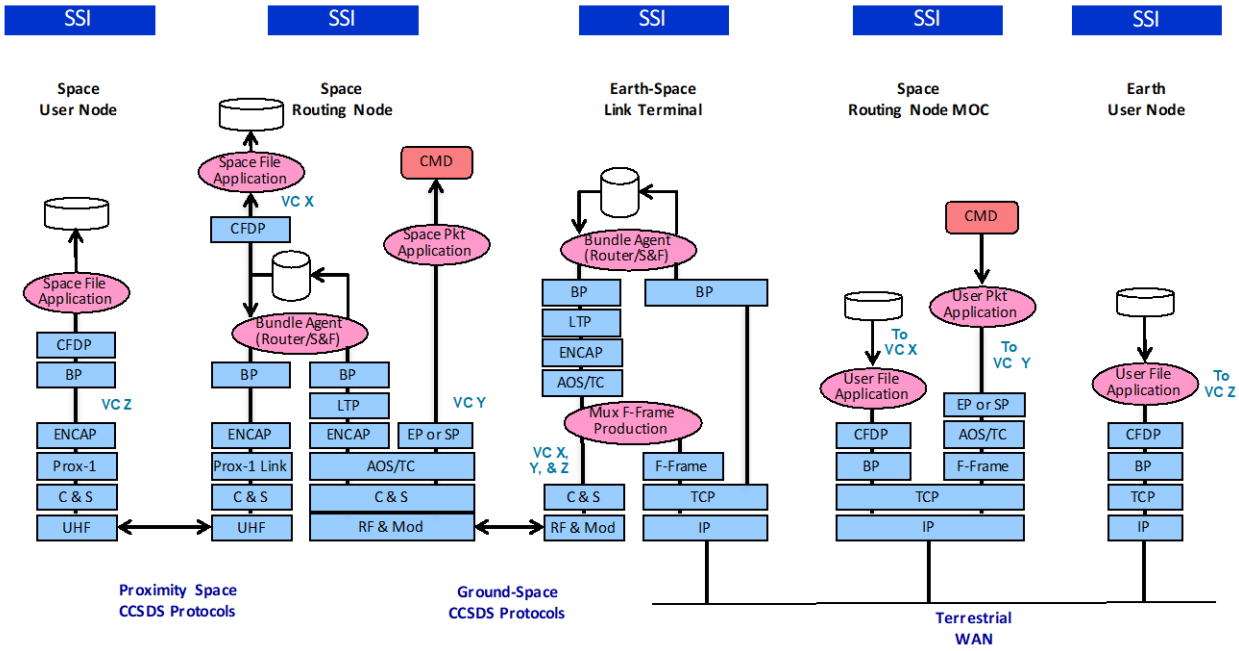

Figure 15. SSI End-to-End Deployment Example

V. Alternative Representations

All of the figures in this paper use the kinds of viewpoints, methods, and drawn representations that have been documented in the Reference Architecture for Space Data Systems [30]. Stylistically these figures may look like typical "Powerpoint cartoons" that have limited architectural coherence relative to what can be accomplished with more modern Model Based System Engineering (MBSE) methods such as SysML [31] and the tools that support this modeling language. However, all of these figures adhere to to the RASDS rules for constructing views that are compliant with the defined viewpoint specifications, and these same viewpoints and the fundamentals of the methods may also be represented very effectively using SysML or other formal methods. In fact, failure to use some sort of modeling formalisms, such as the ones defined in RASDS, can lead to "SysML" models that are no more useful than PPT cartoons.

There is a related SpaceOps 2016 paper by Shames, Sarrel, and Friedenthal [32], that provides a description of a SysML modeling approach for systems and interfaces. This MBSE method is completely aligned with RASDS, but takes advantage of the added power of SysML by developing a related set of views, methods, a profile with a set of stereotypes, and a principled approach to this sort of communications system interface modeling.

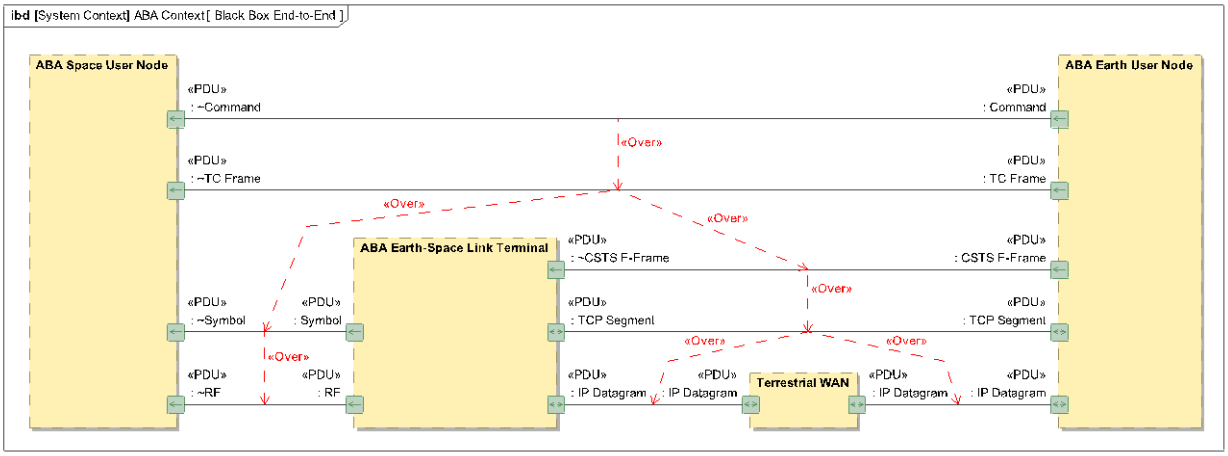

Figure 16. Example of SysML Alternative Representation

Figure 16. shows a SysML End-to-End deployment example that is similar to the one shown in Figure 14. This example shows the exposed protocol interfaces on each of the nodes, EUN, ESLT, and SUN. This figure does not show the internal functional or protocol details of each node, but the SpaceOps paper does include methods for accurately describing all of those details using SysML.

What Figure 16. does show are the major communications systems nodes from an End-to-End context view and also the high level protocol flows. The interfaces (ports) show the typical CCSDS Standards in relationship to these major system nodes. These ports are labeled with the names of actual CCSDS protocol specifications, so in a quite compact form a lot of information is conveyed, and the method shows how to naturally extend this to fuller, and more accurate, specification of components, their construction, and definitions of their interfaces.

Using the methods presented in that paper [32] the top level ports may be decomposed to show the complete interface bindings, the associated protocol stacks (and their behaviors or performance if these details are required), the details of the data types exchanged, and the internals of the end-to-end data flows. One of the great benefits of SysML over Powerpoint or any other tools that just produce drawings is that SysML "drawings" are actually views of an underlying formal model of the objects, functions, information, and relationships, and the tools help maintain the internal consistency of that model.

VI. Conclusion

The building-block approach presented here provides the means for describing, specifying, analyzing, and designing interoperable, cross-supportable, end-to-end space communications systems. This approach enhances component specification, reuse, consistency, conformance to standards, and provides support for adopting CCSDS service, data link, and space internetworking interfaces. Much like Lego™ blocks, these service, component, and protocol building blocks may be used to develop a wide variety of end-to-end mission communication architectures, both in the traditional ABA style and for future Solar System Internetworking (SSI) deployments.

Adopting standard component interfaces and services that conform to CCSDS standards, will support high levels of interoperability and cross support. Adopting the best set of standards should be of particular interest to the developers of service providing elements like ESLTs, as a way of "future proofing" these expensive assets. Furthermore, any new missions that plan to provide, or to use, space relay services or future internetworking should be considering how to design these for longevity. Adopting these building blocks will support system connectivity and composition, and will enable re-use of expensive ground and space communications assets into the future.

As more agencies adopt collaborative approaches to developing and operating space missions interoperability becomes essential. Failure to design these systems-of-systems deployments without adequate attention to the interfaces and protocols will inevitably lead to N**2 problems that will inexorably get worse over time. But this system building block approach, used appropriately, will support multi-mission and multi-agency space deployments and can also lead to the increment deployment of a Solar System Internet (SSI).

Acknowledgments

The figures in this paper are adopted from the CCSDS SCCS references [1, 2] and they all utilize the CCSDS Reference Architecture for Space Data Systems [30] methodology. RASDS was developed in the CCSDS System Architecture Working Group which was most ably led by Takahiro Yamada. The RASDS itself was strongly influenced by the Reference Model for Open Distributed Processing, RM-ODP [33,] and the adaptation of it was aided by discussions with Antonio Vallecillo and Bryan Wood. These methods are both compliant with the ISO Recommended practice for architectural description of software-intensive systems [34]. The RASDS protocol viewpoint, used here for the protocols stacks, is derived directly from the ISO Open System Interconnection - Basic Reference Model [35].

The SCCS standards themselves were developed in the CCSDS Cross Support Architecture Working Group which was led by Takahiro Yamada and then Peter Shames. Velma Anderson, Erik Barkley, James Border, Gian Paolo Calzolari, JP Chamoun, Wolfgang Hell, Greg Kazz, Dennis Lee, Nestor Peccia, John Pietras, Keith Scott, Jean-Marc Soula, and others contributed substantially to the formulation, refinement, and review of the SCCS-ADD and ARD. Wallace Tai strongly encouraged the production of these standards.

All of the work was carried out at the Jet Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration.

References

[1] *Space Communications Cross Support—Architecture Requirements Document*, Issue 1, Recommendation for Space Data System Practices (Magenta Book), CCSDS 901.1-M-1, Washington, D.C., CCSDS, May 2015.

[2] *Space Communications Cross Support—Architecture Description Document*, Issue 1, Report Concerning Space Data System Standards (Green Book), CCSDS 901.0-G-1, Washington, D.C., CCSDS, November 2013.

[3] *TC Space Data Link Protocol*, Issue 3. Recommendation for Space Data System Standards (Blue Book), CCSDS 232.0-B.3, Washington, D.C., CCSDS, September 2015.

[4] *TM Space Data Link Protocol*, Issue 2, Recommendation for Space Data System Standards (Blue Book), CCSDS 132.0-B-2, Washington, D.C., CCSDS, September 2015.

[5] *Space Link Extension—Forward CLTU Service Specification*, Issue 3, Recommendation for Space Data System Standards (Blue Book), CCSDS 912.1-B-3, Washington, D.C., CCSDS, July 2010.

[6] *Space Link Extension—Return All Frames Service Specification*, Issue 3, Recommendation for Space Data System Standards (Blue Book), CCSDS 911.1-B-3, Washington, D.C., CCSDS, January 2010.

[7] *Space Link Extension—Return Channel Frames Service Specification*, Issue 2, Recommendation for Space Data System Standards (Blue Book), CCSDS 911.2-B-2, Washington, D.C., CCSDS, January 2010.

[8] *Monitored Data—Cross Support Transfer Services*, Proposed Draft Recommendation for Space Data System Standards (White Book), CCSDS 922.1-B, forthcoming.

[9] *Tracking Data—Cross Support Transfer Service*, Proposed Draft Recommendation for Space Data System Standards (White Book), CCSDS 922.2-B, forthcoming.

[10] *Space Communications Cross Support—Service Management—Service Specification*, Issue 1, Recommendation for Space Data System Standards (Blue Book), CCSDS 910.11-B-1. Washington, D.C., August 2009.

[11] *Extensible Space Communication Cross Support Service Management—Concept (ESCCS-SM)*, Issue 1, Report Concerning Space Data System Standards (Green Book), CCSDS 902.0-G-1, Washington, D.C., September 2014.

[12] *Security Architecture for Space Data Systems*, Issue 1, Recommendation for Space Data System Practices (Magenta Book), CCSDS 351.0-M-1, Washington, D.C., CCSDS, November 2012.

[13] *CCSDS Guide for Secure System Interconnection*, Issue 1, Report Concerning Space Data System Standards (Green Book), CCSDS 350.4-G-1, Washington, D.C., CCSDS, November 2007.

[14] *IOAG Service Catalog #1*, Washington, D.C., IOAG, March 2010.

[15] *IOAG Service Catalog #2*, IOAG.T.SC2.2011.V1.0, Washington, D.C., IOAG, February 2011.
[16] *Delta-Differential One Way Ranging (Delta-DOR) Operations*, Issue 1, Recommendation for Space Data System Practices (Magenta Book), CCSDS 506.0-M-1, Washington, D.C., CCSDS, April 2011.
[17] *CCSDS File Delivery Protocol (CFDP)*, Issue 4, Recommendation for Space Data System Standards (Blue Book), CCSDS 727.0-B-4, Washington, D.C., CCSDS, January 2007.
18 *CCSDS Bundle Protocol Specification*, Issue 1, Recommendation for Space Data System Standards (Blue Book), CCSDS 734.0-B-1, Washington, D.C., CCSDS, September 2015.
[19] Postel, J, *Internet Protocol*, STD 5, Reston, Virginia, ISOC, September 1981.
[20] *Solar System Internetwork* (*SSI) Architecture*, Report Concerning Space Data System Standards (Green Book), CCSDS 730.1-G-1, Washington, D.C., CCSDS, July 2014.
[21] *Forward Frame—Cross Support Transfer Service*, Proposed CCSDS Draft, forthcoming.
[22] *XML Specification for Navigation Data Messages*, Issue 1, Recommendation for Space Data System Standards (Blue Book), CCSDS 505.0-B-1, Washington, D.C., CCSDS, December 2010.
[23] R. Fielding, et al. *Hypertext Transfer Protocol—HTTP/1.1*, RFC 2616, Reston, Virginia, ISOC, June 1999.
[24] M. Ramadas, et al. *Licklider Transmission Protocol—Specification*, RFC 5326, Reston, Virginia, ISOC, September 2008.
[25] *Encapsulation Service*. Issue 2. Recommendation for Space Data System Standards (Blue Book), CCSDS 133.1-B-2. Washington, D.C., CCSDS, October 2009.
[26] *AOS Space Data Link Protocol*. Issue 3, Recommendation for Space Data System Standards (Blue Book), CCSDS 732.0-B-3, Washington, D.C., CCSDS, September 2015.
[27] *Proximity-1 Space Link Protocol—Data Link Layer*, Issue 5, Recommendation for Space Data System Standards (Blue Book), CCSDS 211.0-B-5, Washington, D.C., CCSDS, December 2013.
[28] *Asynchronous Message Service (AMS)*, Issue 1, Recommendation for Space Data System Standards (Blue Book), CCSDS 735.1-B-1, Washington, D.C., CCSDS, September 2011.
[29] *Service Control—Cross Support Transfer Service*, Proposed CCSDS Draft, forthcoming.
[30] *Reference Architecture for Space Data Systems*, Issue 1, Recommendation for Space Data System Practices (Magenta Book), CCSDS 311.0-M-1, Washington, D.C., CCSDS, September 2008.
[31] *OMG Systems Modeling Language (OMG SysML™)*, Version 1.4, OMG Document formal/2015-06-03, Object Management Group, September 2015. [2]
[32] Shames, Peter M, Sarrel, Marc A, Freidenthal, Sanford A, *Modeling systems-of-systems interfaces with SysML***,** AIAA 2016, SpaceOps 2016, AIAA, May 2016 (submitted for publication).
[33] *Information Technology—Open Distributed Processing—Reference Model: Architecture*, International Standard, ISO/IEC 10746-3:1996, Geneva: ISO, 1996.
[34] *Systems and software engineering — Recommended practice for architectural description of software-intensive systems*, ISO/IEC 42010, International Standards Organization, July 2007, revised 2011.
[35] Information *Technology—Open Systems Interconnection—Basic Reference Model: The Basic Model*, Issue-1, International Standard, ISO/IEC 7498-1:1994, Geneva: ISO, 1994.

Biography

Peter Shames has been engaged in the process of turning computers into useful tools for scientists for the bulk of his professional career. His specific expertise is architecting large-scale space data systems, including space communications protocols and standards. Peter manages JPL's Data Systems Standards Program in the Interplanetary Network Directorate (IND). He is Director of the System Engineering Area for Consultative Committee for Space Data System (CCSDS). Within CCSDS he was lead editor of the CCSDS Reference Architecture for Space Data Systems (RASDS, CCSDS 311.0-M-1) and Space Communications Cross Support Architecture Documents (SCCS-ADD, CCSDS 901.0-G-1 and SCCS-ARD, CCSDS 901.1-M-1).