SSC20-X11-05

A Novel Approach to Transport-Layer Security for Spacecraft Constellations

Edward J. Birrane, Sarah Heiner
Johns Hopkins University, Applied Physics Laboratory, Space Exploration Sector
11101 Johns Hopkins Rd, Laurel, MD 20723; 443-778-7423
Edward.Birrane@jhuapl.edu

ABSTRACT

Spacecraft constellations seek to provide transformational services from increased environmental awareness to reduced-latency international finance. This connected future requires trusted communications. Transport-layer security models presume link characteristics and encapsulation techniques that may not be sustainable in a networked constellation. Emerging transport layer protocols for space communications enable new transport security protocols that may provide a pragmatic alternative to deploying Internet security mechanisms in space. The Bundle Protocol (BP) and Bundle Protocol Security (BPSec) protocol have been designed to provide such an alternative.

BP is a store-and-forward alternative to IP that carries session information as secondary headers. BPSec uses BP's featureful secondary header mechanism to hold security information and security results. In doing so, BPSec provides an in-packet augmentation alternative to security by encapsulation. BPSec enables features such as security-at-rest, separate encryption/signing of individual protocol headers, and the ability to add secondary headers and secure them at waypoints in the network. These features provided by BPSec change the system trades associated with networked constellations. They enable security at rest, secure content caching, and deeper inspection at gateways otherwise obscured by tunneling.

I. INTRODUCTION

The promises of global communications include increased quality of life, better understanding and utilization of our planet, and more efficient industry. The pragmatic buildout of such a system should include space-based networking nodes, but must also consider existing infrastructure, spacecraft constraints, and a model for federating individual networks into a functioning internetwork[1].

Multiple actors within industry and government are constructing near-Earth communications constellations. Most constellations are envisioned as forming a dedicated backhaul interfacing with the terrestrial Internet (and third-party intranets) at ground stations. Regardless of design, a constellation's success comes from its ability to provide trusted communications.

Trusted communications are achieved by the integration of network architecture, topology, and multiple layers of security. However, security protocols in networked constellations will have to overcome multiple obstacles unique to the space environment in addition the usual set of security considerations. For example, security endpoints may be established between spacecraft in the constellation. In cases where two spacecraft are not in regular contact, this topological change can complicate network and transport layer endpoint-centric models of security services. The traditional space link approach of restricting security to only the link layer leads to other security issues. Packets may be rerouted in transit or may share links with other packets representing users from different administrative domains or with different trust models and credentials[2].

Like secured communications on the terrestrial Internet, an end-to-end security model at the transport layer reduces reliance on establishing a "chain of trust" across a data path. Terrestrial approaches to end-to-end transport security function well where spacecraft communications replicate the reliable, end-to-end links characteristic of the Internet. In cases where spacecraft cannot (or will not) support this connectivity model, these approaches may fail.

Sometimes this failure stems from the links required to make a cryptographic algorithm function. For example, a stream cipher will not function across lossy paths. In other cases, the security protocol - the mechanism used to communicate cryptographic material - fails to carry information effectively. When security failures are caused by protocol issues, new security protocols can be

designed to better carry cryptographic material for more challenged or diverse environments.

NASA, in coordination with other international space agencies, industry, and academia, has standardized the Bundle Protocol[3] (BP) as an alternative to the Internet Protocol (IP) in cases where networks are challenged by significant signal propagation delays or frequent link disruptions. While these specific impairments may or may not be present in any given near-Earth constellation, the features and structure of BP "bundles" enable new techniques for data handling and data security. A security protocol built around the unique features of BP is the Bundle Protocol Security[4] (BPSec) Protocol.

BPSec provides end-to-end, store-and-forward-friendly security services with unique features such as data-at-rest, different cipher suites for different portions of the BP bundle, and the ability to add security to a bundle at a waypoint without losing or hiding important semantic information.

Specifically, BPSec enables an augmentation approach to network security. In such an approach, a bundle can carry network, transport, and application data with BPSec applying different cipher suites to each of these differently scoped information elements. By offering an alternative to security-through-encapsulation, BP and BPSec enable fewer constraints, and thus more flexibility, in the design of networked constellations.

The remainder of this paper is organized as follows. Section II discusses both traditional and emerging approaches to constellation networking, transport, and security. Section III discusses the motivation for a new security protocol. Sections IV and V present the systems overview of BPSec and its implementation mechanics, respectively. Section VI describes the roles and responsibilities of BPSec network agents and Section VII provides examples of BPSec features operating in a network. Section VIII concludes the work with a summary of BPSec contributions.

II. APPROACHES TO CONSTELLATION NETWORKING AND SECURITY

Fully networked constellations are receiving renewed interest by government and industry. The most common approaches to building these constellations are to either reuse as much as possible from the terrestrial internet or to treat SATCOM links as a special tunnel connecting networks. Therefore, to review current approaches to securing networked constellations is to review the security of the terrestrial Internet and the security of dedicated SATCOM tunnels.

This section discusses the networking assumptions upon which terrestrial Internet security is predicated, the unique constraints often present in space-based communications that may violate those assumptions, and emerging protocols that may be useful in addressing these gaps.

Encapsulation and the TCP/IP Model

As the most successful, scaled networking model in human history, the TCP/IP model used by the terrestrial Internet serves as a reasonable starting point for a networked constellation. The protocols and their behaviors are understood, and industry has already developed commercial tools for like systems.

This networking model is characterized by a layered architecture with each layer responsible for different networking features. These logical "layers" are typically instantiated with an encapsulation approach; network information from one layer is "encapsulated" by network information from its lower layer just as it encapsulated information from its higher layer. This concept is illustrated in Figure 1.

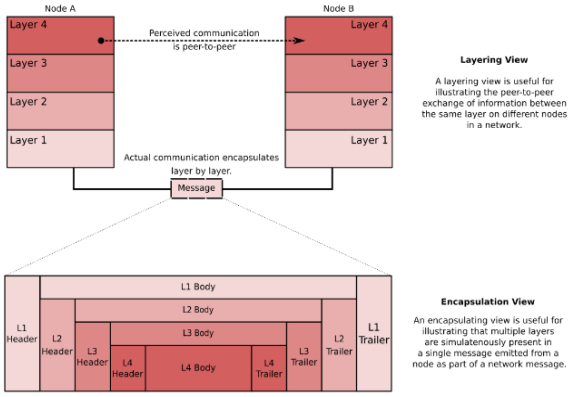

Figure 1 - Layering is typically implemented using encapsulation[5].

For example, encapsulation is fundamental to the design of the protocols used to secure the internetwork and transport layers of the terrestrial Internet. Familiarity with these approaches, and their behavior, and limitations, is important to understanding the motivation for, and desired characteristics of, an alternative approach.

Securing the Internetworking Layer

The internetworking layer of the TCP/IP model[6] routes packets across paths through either a single network or multiple, distinct network segments. The Internet Protocol (IP) that operates at this layer is secured with the Internet Protocol Security[7] (IPSec) protocol.

IPSec uses separate mechanisms based on whether authentication or confidentiality is being applied to an IP packet. The Authentication Header[8] (AH) provides authenticated data integrity over immutable parts of the packet whereas the Encapsulating Security Payload[9] (ESP) provides signed confidentiality to the IP payload by replacing its plain text contents with cipher text. The ESP and AH can be applied in either tunnel or transport mode, depending on how much packet data may be exposed to an external observer.

As illustrated in Figure 2, tunnel mode treats the entire secured IP packet as the data of a new, encapsulating packet thus securing the entirety of the encapsulated packet. Transport mode preserves header portions of the IP packet. The specific way in which IPSec implements transport mode differs based on whether the packet is IPv4 or IPv6.

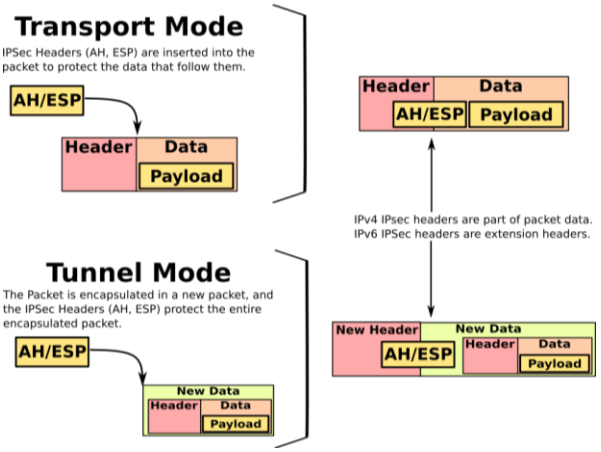

Figure 2 - IPSec either augments or encapsulates packets based on mode and IP protocol version.

The features and configuration of IPSec policy must be tuned to the network in which it is deployed. One such example of a secure architecture based on IPSec is the High Assurance Internet Protocol Interoperability Specification (HAIPIS) which defines how to combine multiple terrestrial Internet protocols to communicate highly sensitive data. High Assurance Internet Protocol Encryptor (HAIPE) devices conform to this specification and are required in circumstances requiring high levels of security[10]. HAIPE devices implement the IPSec protocol for network security but restrict certain capabilities and provide custom enhancements - such as pre-placed, symmetric keys.

IPSec and HAIPE present both capabilities and challenges for networked constellations. By focusing on the networking layer, security can be agnostic of higher-layer considerations at the transport layer. However, tunnel mode cannot address the situation where only a subset of an IP packet needs security and transport mode cannot address the situation where different parts of the packet need different types of security. As we discuss later, both situations can occur in a fully networked constellation.

The use of IPSec headers, particularly in transport mode, demonstrates that headers can serve as an alternative to encapsulation for carrying security information. However, header semantics associated with IPv4 and IPv6 may not be featureful enough to support multiple types of security in a single packet.

Securing the Transport Layer

The transport layer of the TCP/IP model is responsible for reliable communication between messaging endpoints, to include in-order receipt, deduplication, and re-transmission. While IPSec is implemented between internetworking endpoints, transport security is used to secure information across multiple IPSec tunnels and to secure information after it has been taken out of an IP packet as part of delivery or re-encoding at the transport layer. Approaches to securing the transport layer differ based on whether the transport layer is connection-oriented or connectionless. Connection-oriented transport uses TCP or QUIC for communication whereas connectionless transport uses UDP.

Transport-Layer Security 1.3[11] (TLS) is the current implementation of security services for the TCP/IP transport layer. TLS is an umbrella term for a group of protocols requiring timely, ordered delivery for their proper operation. This includes protocols for session establishment, key negotiation, application data fragmentation, and record creation that are segmented into a ‘handshake layer” and an “application layer”.

TLS requires a reliable transport layer for four reasons. (1) TLS records use implicit sequence numbers. If a TLS record is lost the wrong sequence number may be used to remove protection on that record. (2) TLS handshakes are dependent on precise ordering to succeed. (3) TLS handshakes omit some acknowledgements based on the assumption of receipt. (4) Handshake messages may be large enough to require fragmentation, the re-assembly of which requires reliable transport[12].

Connectionless transport mechanisms, such as UDP, do not retain information related to a sent message; the message is not explicitly acknowledged or re-transmitted. The Datagram Transport Layer Security[12] (DTLS) protocol operates similarly to TLS with the exception that it has been modified for operation over a non-reliable transport. This is accomplished by adding explicit sequence numbers, retransmit timers, and devising special rules for handling segment

fragmentation. DTLS functions in cases where transport layers are unreliable. Its overall security design remains like TLS in that it seeks to establish synchronized information between two security endpoints.

The QUIC[13] protocol, originally developed by Google and currently being adopted by the Internet Engineering Task Force (IETF), is a lighter-weight alternative to TCP for connection-oriented transport. QUIC reduces the number of handshake messages required for security negotiation and achieves reliability by federating multiple UDP streams that are managed by the QUIC layer itself. QUIC incorporates some aspects of the TLS handshake and can also integrate with TLS directly for security[14]. While QUIC provides a faster web loading experience, its connection approach suffers over satellite links[15].

Securing Traditional Spacecraft Communications

There exists no single architecture for a space communications system because these systems exist for different purposes and are built by diversified sets of government agencies, commercial industry, and academia. Like the terrestrial Internet, space communications can be discussed in general terms by the standards used in their construction. The Consultative Committee for Space Data Systems (CCSDS) is one such standards organization whose members comprise most of the world's space agencies. Their standards enable interoperable commercial products and document common solutions to common problems.

CCSDS protocols define both a Space Packet Protocol[16] (SPP) for path-based routing of a packet through a space network and an encapsulation service[17] for tunneling IP (or similar) traffic through the space network. In either case, security for space systems is applied at some layer above the encapsulation service, built into a space-based data link layer, or both.

The CCSDS protocol used to secure the data link layer is the Space Data Link Security (SDLS) Protocol. This approach treats space communications as a special tunnel between other parts of a secured network[18]. While this is a useful operating mode in cases where a network uses a spacecraft as a link, it does not scale in cases where the constellation, itself, is the network.

The entry and exit points of a satellite communications system may be subject to link disruptions, low bandwidth, or both. To flow terrestrial Internet traffic, these systems deploy Performance Enhancing Proxies (PEPs) that "improve the performance of the Internet protocols on network paths where native performance suffers due to characteristics of a link or subnetwork on the path" [19].

For example, when used at the transport layer, a TCP PEP may either generate additional local acknowledgements (or window acknowledgements) in an effort to provide better flow control. Alternatively, a TCP PEP can also provide TCP spoofing in which the PEP itself is the endpoint of the "local" TCP connection. Spoofing requires changing information in protocol headers and sending messages as if they came from other nodes - the exact types of behaviors that are usually protected against by security services.

PEPs are often required infrastructure components, but can be expensive to deploy, require tuning and configuration, and present architectural problems related to how security can be maintained to and through PEP endpoints.

A Networking Paradigm for Space and Ground Networks

The Bundle Protocol (BP) version 6 (BPv6) was published in 2007 as an experimental[20] approach to packetized internetworking for Delay-Tolerant Networks[21]. BPv6 was profiled by the CCSDS[22] in 2015 with version 7 (BPv7)[3] being proposed for standardization in 2020 by the IETF.

The BP data unit is called the "bundle" and represents a bundle of "blocks". Each bundle contains a "primary block", a "payload block", and zero or more "extension blocks" that are like secondary headers. BP provides two motivating features in this context: a flexible protocol extension mechanism and standardized behavior for store-and-forward operations.

Like IPv6 extension headers, BP extension blocks can be defined to carry additional information in a bundle. Unlike IPv6 extension headers, BP extension blocks can be re-ordered, can be of variable length, and can be added/processed/removed by any BP agent in the network, as illustrated in Figure 3.

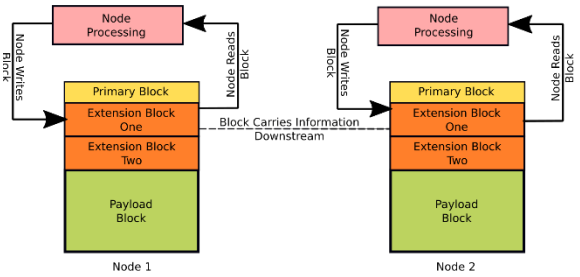

Figure 3 - BP provides a more featureful extension header mechanism than IPv6[5].

BP extension blocks are expected to carry information with application, network, or node scope. For example,

the following types of information may be included in various extension blocks in a bundle.

- Annotative information about the payload to expedite processing at waypoint nodes.
- Control-channel information sent amongst nodes independent of any specific payload.
- Session information when sessions cannot be maintained at endpoints.

The motivation for providing a more featureful extension mechanism stems from BPs intended use as a store-and-forward protocol for deep space networking. Deep-space links are characterized by long signal propagation delays and frequent disruptions. Networks operating in these conditions would need to store data for long periods of time and messages would need to carry information that would otherwise be assumed resident and synchronized at communication endpoints.

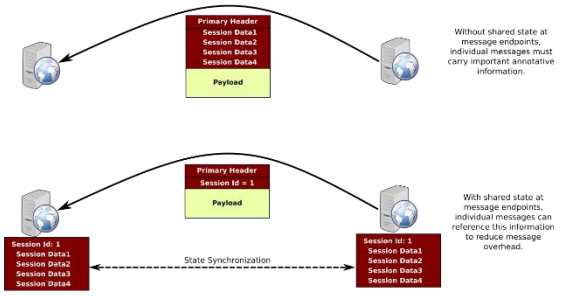

Figure 4 - BP carries session information that cannot be synchronized end-to-end[5].

Because BP requires that its protocol agents store bundles, and because bundles can carry session information in extra headers, BP offers a viable alternative to PEPs for SATCOM[23,24]. For example, a spoofing PEP at the transport layer would not be necessary because the node running the spoofing proxy would already have the ability to store BP bundles in its role as a BP agent. BP improves communications performance when either physical or service interruption is experienced[25], making it a compelling replacement for PEPs that would otherwise be used in a network to address these same service issues.

III. MOTIVATION TO DEVELOP A NEW SECURITY PROTOCOL

This section discusses the motivation for a new security protocol in the context of known issues when using (or combining) existing security protocols. This is not to say that the use of IPSec, HAIPE, TLS, DTLS, and PEPs is insecure, but that the use of these protocols necessitates either accepting significant inefficiencies or placing significant constraints on the network architecture.

This section also discusses the conditions present in a networked constellation that would be considered stressing for existing security protocol designs, and what new features should be present in any security protocol proposed for use in these at-scale architectures.

Issues with Existing Security Protocols

There are four concerns related to the deployment of security protocols that may be impactful to networked constellation architectures: coarse-grained packet security, super-encapsulation, delay intolerance, and reliance on synchronized data.

Coarse-Grained Packet Security. Existing security approaches are often all-or-nothing with respect to confidentiality. For example, tunnel modes encrypt entire packets whereas transport modes encrypt entire payloads, but not header information (IPv4) or some header information (IPv6). This is consistent with a strict layered approach to security but is unable to support architectures where the responsibilities between layers are not so clear. For example, if an encrypting "network layer" security tunnel is defined between two endpoints, then a node within that tunnel would be unable to see any transport layer information, even if a certain subset of that information would be useful.

Such non-standard functionality would be useful and secure in a networked constellation if messages mixed network, transport, and application layer information. For example, an unmodified TCP/IP stack is unable to use IPSec through TCP PEPs and blending these layers of information together usefully requires additional architectural components, non-standard protocol modifications, and new algorithms[29].

Super-Encapsulation. One approach to avoiding the information-hiding problem caused by encapsulation is to use more encapsulation. For example, if TCP headers are encrypted by IPSec or HAIPE, then the entire packet can be placed as the payload of another TCP segment for reliability to and through the security tunnel. In this way, important plain text information can be preserved in the encapsulating TCP segment rather than hidden in the encapsulated, encrypted TCP segment[26].

This approach incurs the obvious inefficiencies of repeated headers and header information. Further, it presupposes that the encapsulating packets can be populated with appropriate header values from the encrypted, encapsulated packets. Finally, these approaches must take special care to avoid problems with nested protocol control loops, such as the TCP meltdown problem[27,28]. In this situation, conflicts between timers and acknowledgements of the

encapsulating TCP segment cause problems with the retransmission of the encapsulated TCP segment.

Delay-Intolerance. Most of the terrestrial Internet exists within ready access to capable infrastructure providing timely and reliable communications and reducing the impact of transport-layer retransmissions. In cases where end-to-end communications are delayed as a function of capacity and congestion, pre-placed information can load balance information for geographic distribution.

Networked constellations may not have ready access to such infrastructure, and rapidly moving, sparsely populated constellations may not be able to effectively pre-place information. In such situations, protocols must be able to handle delays and disruptions as part of the normal operation of the network, rather than as some transient error condition that must be "waited out".

Reliance on Synchronized Data. Security protocols such as TLS and IPSec rely on the establishment of security sessions prior to the secure exchange of information. This session establishment implies that the endpoints of the secure tunnel allocate and maintain session state. If session information times-out or is changed (such as when rotating keys), or if messages are lost or received out of order, then the secure tunnel may no longer be usable. Protocols such as DTLS require less session information but preserve some concepts as part of TLS compatibility.

Networked Constellation Stressing Conditions

Security tunnels in resourced networks can be established in milliseconds and if a tunnel collapses, it can be re-established just as quickly. Networks challenged by delays and disruptions may not be able to establish tunnels as quickly and must architect the network to avoid needless renegotiation of security tunnels.

For example, a common "passthrough" spacecraft architecture involves using end-to-end transport, IPSec or HAIPE encryption, and super-encapsulation through PEPs to handle individual delays. Architectures such as the one illustrated in Figure 5 are built from necessity, not efficiency. Depending on the way in which security is passed through the PEPs, and how much super-encapsulation is used, the system may also be brittle to change and significantly reduce data throughput. This architecture also does not represent a networked constellation which would need to establish transport and network security services that may both originate and terminate within the constellation itself.

There is no single definition of a networked constellation architecture. Architectural concepts span the spectrum from linear pass-through systems to hub-and-spoke models to opportunistic mesh networks. When reasoning about the potential stressing conditions encountered by spacecraft in such a constellation, the most complex formulation must be considered; if a security solution can operate in the most stressing environment then it would presumably operate in less stressing environments.

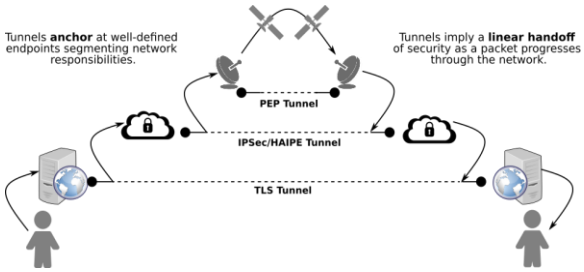

Figure 5 - Existing security mechanisms lock networks to a specific model.

There are five stressing conditions present in an opportunistically meshed constellation. Such an architecture is characterized by ad-hoc contacts and ill-structured data paths. Specifically, these conditions are intermittent connectivity, congested paths, partitioned topologies, limited link state information, and multiple administrative controls.

Intermittent connectivity. Inter-spacecraft links, like any wireless link, may be attenuated, unidirectional, occulated, or otherwise disrupted. These disruptions may be foreseeable, planned on short notice, or occur randomly as a function of the external environment. In opportunistic networks nodes may join and leave the network without prior planning which also affects connectivity.

Unless specifically designed to support constant, high-rate communications and unless operating under nominal conditions, a networked constellation will likely encounter disruptions leading to data delivery delays. This is the case with even space-based pass-through links, and why PEPs are often deployed as part of a SATCOM solution.

Congested Paths. While there are significant advancements in the bandwidth achievable through inter-satellite links, networked constellations of any size and data capacity will likely still see congestion across data paths. For example, optical inter-satellite links can support speeds of at least 10Gbps[30] representing at least an order-of-magnitude increase in inter-satellite bandwidth. However, the sufficiency of the path can only be evaluated relative to the data in the network. For example, ultra-high definition cameras can output 12Gbps[31] streams and 8K cameras can output 16Gbps

streams. Notwithstanding protocol overhead, a single camera can saturate an optical link and, thus, any path needing to use that link. Data volumes and data rates depend on how the constellation is used, but it is unlikely that such links would avoid congestion.

Partitioned Topologies. The source and destination of messages within the networked constellation may never have a path connecting them at any one instant in time. These nodes may exist in different network partitions or, in an ad-hoc network, may leave the network entirely during some part of the message exchange. This partitioning may persist for extended periods of time such that it is the rule not the exception. Approaches that assume timely connectivity between messaging endpoints may not function in an opportunistic mesh network.

Limited Link State Information. Intermittent connectivity, congested links, partitioned topologies, and asymmetric data rates complicate the measurement and exchange of link state. Because the magnitude of link state data grows with the number of links, measuring aggregate link state across an evolving networked constellation requires a high volume of information exchange which, itself, can be limited by congested links.

Multiple Administrative Controls. The spacecraft comprising a networked constellation may be owned, administered, and operated by a single organization. Alternatively, the network may be built from a federation of spacecraft representing multiple organizations with diverse ownership, operational concepts, and administrative policies and configurations.

Administrative domains may be enforced as part of the hierarchy of a federated network. For example, a message generated in a "source segment" of a network may have one set of policy applied to it, and then other policies as it traverses other "transport" segments of the network, and possibly other sets of policy once received at the destination segment of the network. Such a circumstance is illustrated in Figure 6.

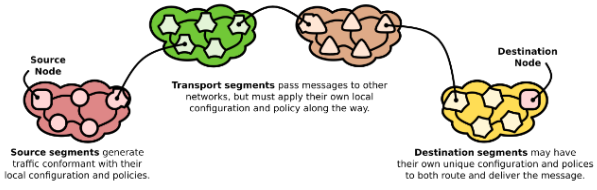

Figure 6 - Federated networks cross administrative domains[5].

Desirable Properties of a Security Protocol

An ideal security protocol is one which provides needed security services while placing the fewest constraints on the architecture of the network it is securing. In particular, network constellations may not behave the same as terrestrial networks.

There are five properties that make a security protocol more adaptable to the stressing conditions of a networked constellation without otherwise constraining the architecture of that constellation. These properties are fine-grained security, augmentation before encapsulation, delay tolerance, topology independence, and self-sufficiency.

Fine-Grained Packet Security (FGPS). Not all information in a data packet needs to be secured in the same way. For example, in both IPv6 and BP, secondary headers can be used to carry a variety of annotative information describing characteristics of the link, the payload, the path, the prior hop, quality of service, and other information. While some of this information may, itself, need security, it may not need the same security as the payload. A flexible security protocol is needed which allows for different kinds of information in a packet to be secured differently.

Augmentation Before Encapsulation (ABE). Fine-grained security cannot be achieved with encapsulation, which, by definition, hides encapsulated data. An augmentation approach allows for the creation of headers in a packet for the purpose of carrying security information. The potential efficiency of augmentation over super encapsulation is illustrated in Figure 7.

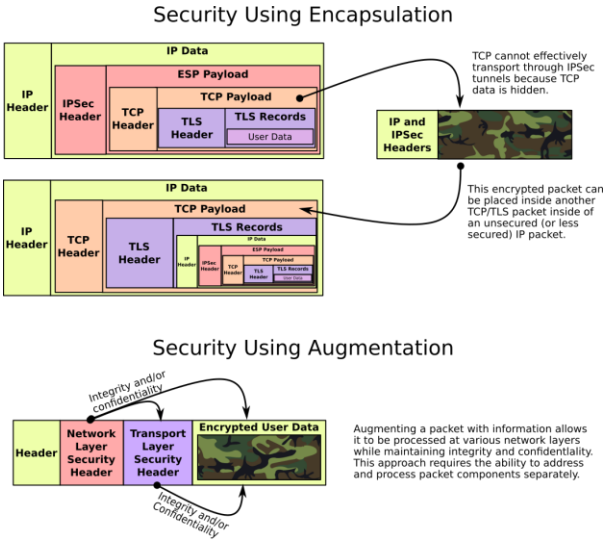

Figure 7 - Protocol augmentation provides an alternative to encapsulation for security services.

Delay-Tolerance (DT). A desirable security protocol would be one which can tolerate delivery delays and link disruptions – particularly as it pertains to the establishment of initial cryptographic material and the maintenance of session state.

Topological Independence (TI). Not every networked constellation will maintain a fixed topology. While changes to waypoint nodes in a network might be opaque to the processing of IPSec and TLS, changes to the end points of a secure tunnel can break the tunnel when new endpoints have not synchronized on the same session state.

Existing security protocols achieve their topological independence by re-purposing protocol data fields in non-standard ways. For example, security tunnel endpoints could be specified as local loopback addresses and a separate mechanism could be deployed to "re-route" tunnels to new endpoints.

A desirable security protocol would allow for dynamic tunnel endpoints without requiring additional layers of redirection. Such a common problem should be handled in a common way.

Self-Sufficiency (SS). As a function of communication limitations, topological change, node resources, or data volume there may be times when endpoints in a network cannot maintain required session state. A desirable security protocol would support carrying necessary information within the secured packet itself.

Table 1: Desirable security properties address stressing conditions in networked constellations.

	FGPS	ABE	DT	TI	SS
Intermittent Connectivity			X		X
Congested Paths	X	X			
Partitioned Topologies		X		X	
Limited Link State			X	X	
Multiple Administrative Controls	X				X

As illustrated in Table 1, these desirable properties address the stressing conditions of networked constellations. Fine-grained packet security and augmentation reduce message size and better support multiple administrative domains and changing network topologies. Delay-tolerance means that loss of link connectivity and lack of knowledge of link state will not prevent eventual secure message delivery. Topological independence does not hard-code security endpoints allowing them to change as link state and topologies change. Finally, self-sufficiency does not require state to be held at endpoints, which may exist in different administrative domains or be inaccessible due to loss of connectivity.

IV. BPSEC: A NOVEL APPROACH TO TRANSPORT-LAYER SECURITY

The predilection for layer-based security and super encapsulation is understandably driven by the existence of standards, standards-based products, and their successful deployments in other networking use cases. To the extent that traditional security approaches cannot address the needs of emerging networked constellations, a new approach should be investigated.

The BP represents a novel transport protocol with an expressive header syntax. Unlike IPv6 extension headers, multiple extension blocks of the same type can exist (enumerated by an instance ID), block ordering is not mandated, and waypoint nodes can insert blocks as necessary. This allows extension blocks to act as "secondary payloads" that can be secured individually.

Such a novel approach to security at the transport layer is a fusion of concepts present in traditional security approaches.

- Like IPv6, extension headers can hold security information, and be the targets of security services such as authentication and confidentiality.
- Like IPSec, authentication and confidentiality can be applied separately based on need.
- Like QUIC and DTLS, information can be bundled to reduce the round-trip handshakes necessary for negotiation.
- Like TLS, data can be segmented - using BP extension blocks instead of TLS records.

The concepts incorporated by BPSec, as inspired by useful properties of other security protocols, are listed in Table 2.

Table 2: BPSec incorporates useful concepts from other protocols.

BP Security Approach	IPv6	TCP	QUIC	IPSec	TLS	DTLS
Security in headers				X	X	X
Nodes add headers	X					
Data bundled to reduce round-trips			X			X
Segment secured data		X		X	X	X
Apply different cipher suites				X	X	X

Key Negotiation and Management

Key negotiation and management enable security mechanisms based on shared secrets. Unlike IPSec and TLS, BPSec does not mandate a specific mechanism for the generation of shared secrets and session keys or for managing longer-lived security keys. If security nodes can generate keys, those keys can be used with BPSec and the protocol is not otherwise limited.

In environments where terrestrial Internet Key Exchange (IKE) can occur, BPSec allows this formulation. BPSec also supports environments requiring symmetric, preconfigured keys or other out-of-band mechanisms.

Early Attempts to Secure the Bundle Protocol

The concept of extension blocks used to implement BP security features was first proposed in the experimental Bundle Security Protocol[32] (BSP). The BSP defined "security blocks" to carry cryptographic material associated with authentication, integrity, or confidentiality of other bundle blocks. These blocks were like the authentication header (AH) and encapsulating security payload (ESP) of IPSec in IPv6. This experimental work was refined as the Streamlined Bundle Security Protocol[33] (SBSP) defined by the CCSDS for securing BPv6. BPSec is the security protocol for BPv7 and beyond.

BPSec Design Principles

The design of BPSec was guided by the principles of block-level granularity, multiple security sources, mixed security policy, user-selectable security contexts, and deterministic processing. Adherence to these principles ensures that BPSec meets the desirable properties of a networked constellation security protocol.

Block-Level Granularity (BLG). By definition, a bundle is a collection of blocks. This means that any security service present in the bundle must be captured in a block and the target of that security service must, itself, be captured in one or more blocks. This "block-level" granularity allows BPSec operations to be targeted to different types of information in the bundle. For instance, as illustrated in Figure 8, confidentiality may be applied to one block while a signed integrity mechanism may separately be applied to some other block.

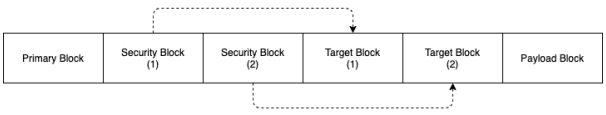

Figure 8 - Blocks in a bundle carry security information for other blocks in the bundle.

Multiple Security Sources (MSS). BPSec allows multiple nodes to augment the bundle with new security blocks. This allows different network segments to add security on blocks within the bundle while keeping other blocks within the bundle unmodified. Any node which adds a security block to a bundle is the "security source" for that block. Importantly, adding a new security service into a bundle does not require altering other information in the bundle not related to the security service.

Supporting multiple security sources also means supporting multiple security endpoints. BPSec assigns security-related processing functions to nodes as a matter of the policy configuration of the nodes themselves. Security services provided by the BPSec do not require "anchoring" at specific nodes in a network.

Mixed Security Policy (MSP). BPSec policy allows nodes to assert their role in the creation, verification, and acceptance of security services in the bundle. This is like IPSec and TLS services where nodes are configured to serve as tunnel endpoints. However, the BPSec approach allows these roles to change even after security has been applied to the bundle without requiring out-of-band address translation schemes such as terminating security tunnels at loopback addresses.

User-Selectable Security Contexts (USSC). Unlike IPSec which requires a security association between two endpoints, and unlike TLS which requires a synchronized session, BPSec can operate in environments where there is no negotiated information between a security source and a security endpoint. This behavior is required given the environments in which BP may be deployed.

To address this scenario, BPSec introduces the concept of a "security context" as the union of security policy, cipher suite configuration, and session parameters necessary to implement a security service. Security contexts can be populated based on the needs of the networking environment.

Security contexts may contain all information needed to process the security operations in the bundle (fully asynchronous mode), they may only include necessary information to identify security associations (fully synchronous mode), or they may carry default or other synchronizing information (hybrid mode).

Deterministic Processing (DP). The benefit of super encapsulation is that the processing order of security services in a packet is always clear. For example, if a packet is encapsulated in another packet, the encapsulating packet must be deciphered prior to addressing security for the encapsulated packet.

When choosing to augment packets, processing order is no longer encoded in the encapsulation hierarchy. This may cause problems in understanding the order in which security blocks should be processed. The BPSec defines processing rules and restrictions to prevent block processing ambiguities; the processing order for BPSec security blocks is deterministic.

Mapping BPSec Principles to Desirable Properties

The mapping of BPSec design principles to network constellation desirable properties is given in Table 3.

Securing individual blocks using multiple security sources provides fine-grained security without mandating an encapsulation approach and without incurring the challenges of that approach in a dynamic topology. Flexible security policies and security contexts provide standardized mechanisms for adapting to delivery delays and changes to the network topology. Security-by-augmentation could not occur without a deterministic processing order.

Table 3: BPSec design properties satisfy the desirable properties of a security protocol.

	BLG	MSS	MSP	USSC	DP
Fine-Grained Security	X	X	X		
Augmentation before encapsulation	X	X			X
Delay-tolerance			X	X	
Topological independence	X	X	X	X	
Self-sufficiency				X	X

V. BUNDLE PROTOCOL SECURITY MECHANICS

BPSec provides two security services for blocks within a bundle: plain text integrity and signed confidentiality. A BPSec security operation is defined as the application of a security service to a specific target block within the bundle. The physical representation of a security operation in a bundle is a BP extension block called a security block. The security block used to apply plain text integrity is called the Block Integrity Block (BIB) and the security block used to apply signed confidentiality is called the Block Confidentiality Block (BCB).

The Block Integrity Block (BIB)

Similar to an IPSec AH, a BIB carries the output of an integrity mechanism as one or more security results. The BIB is calculated by feeding the contents of its target block to a selected integrity mechanism and storing important parameters, optional session information, and the results of the integrity mechanism.

The BIB may carry information for one or more security targets in cases where the integrity mechanism uses shared parameters as applied by the same node in the network. In this case, multiple sets of security results are captured in the block, one for each security target.

While conceptually like the AH, BIBs may pick and choose which blocks in the bundle they apply to, and integrity can be preserved if the ordering of blocks in the bundle changes.

The Block Confidentiality Block (BCB)

Similar to the IPSec ESP, a BCB carries cryptographic material relating to the cipher suite used to encrypt the contents of some other block or blocks in a bundle. The BPSec requires the use of Authenticated Encryption with Additional Data (AEAD) cipher suites to generate signed cipher text and other authenticated but unencrypted data.

When a BCB is added to a bundle its target blocks are encrypted in place. When the BCB is removed from the bundle, the target blocks are decrypted.

The BCB carries plain text parameters, encrypted parameters, and optional session information associated with the cipher suite. The BCB may also carry additional signatures and overflow cipher text (in cases when produced cipher text is larger than the original plain text of the target block) or this information may be merged with the cipher text itself and stored in the target block in cases where the target block is allowed to increase in size.

VI. BPSEC ROLES AND RESPONSIBILITIES

Security policy at the node determines the role the node will play for any given security block in the bundle. For example, a single node may add a new security block, verify the integrity of another security block, and accept/remove a third security block within the same bundle.

Security sources add security blocks when necessary to implement security operations as required by node policy. As illustrated in Figure 9, a node acting as a security source accepts a bundle from the local Bundle Processing Agent (BPA) which contains at least one target block requiring the application of a security service. The security source adds a security block for the appropriate target block prior to forwarding the bundle.

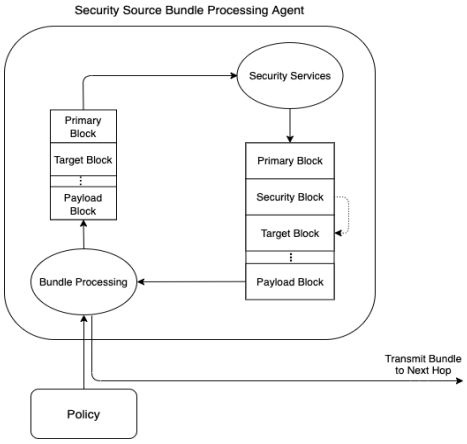

Figure 9 - Security Sources add security blocks to bundles as required by node policy.

Security verifiers check that the contents of an existing security block are consistent with the conditions that were present when the security block was created. In the case of integrity, security results can be re-calculated and compared to those in the security block. In the case of confidentiality, any associated plain text authenticated data may be verified in a similar process. Security verifiers do not decrypt data.

As illustrated in Figure 10, a node acting as a security verifier accepts a bundle with one or more security operations represented in it and, as a matter of node policy, verifies some subset of security blocks. If a verifier cannot verify the integrity of a security block, the block may be removed from the bundle, the target block may be removed from the bundle, or the bundle may be removed from the network.

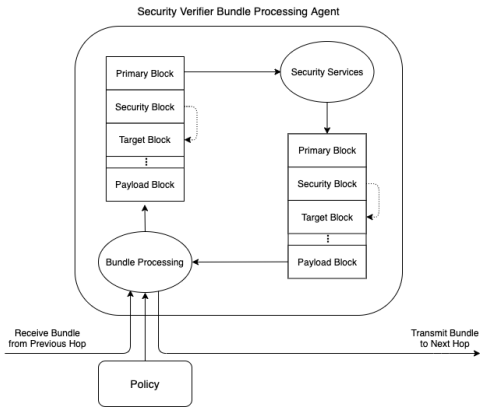

Figure 10 - Security verifiers ensure that data has not been changed in transit.

Security acceptors both verify the security operation in the bundle and process/remove the security block from the bundle, as illustrated in Figure 11. In the case of integrity, the security acceptor operates in the same way as a security verifier, with the addition that the security block is then removed from the bundle. In the case of confidentiality, the appropriate cipher suite is given the cipher text body of the target block as well as other parameters from the security context. The calculated plain text returned from the cipher suite replaces the cipher text contents of the target block and the security block is removed from the bundle.

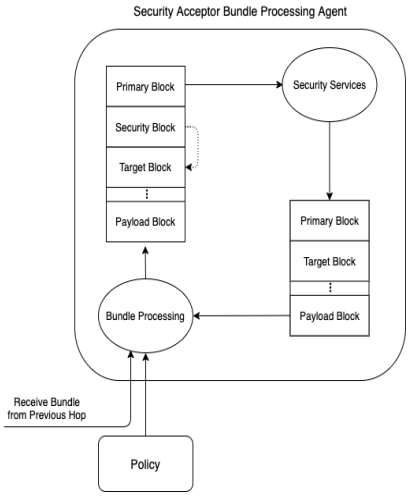

Figure 11 - Security acceptors are the endpoints of a security operation in a bundle.

Like other security protocols, bundles with security blocks may pass through nodes that are not otherwise security aware. The relationship of bundles with security blocks through BPSec aware and BPSec unaware nodes is illustrated in Figure 12.

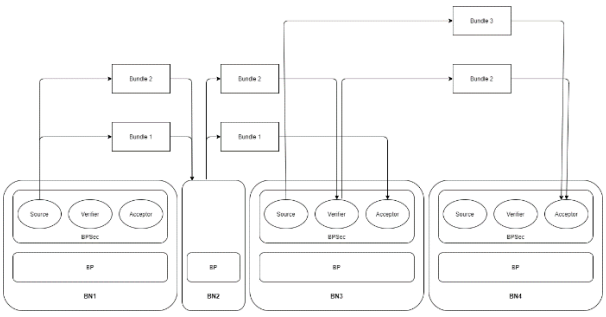

Figure 12 - Bundle Transmission by BPAs with Various Roles

In this figure BP nodes BN1, BN3, and BN4 are BPSec aware while BN2 cannot process security blocks. Consider the role of these four nodes with respect to three different bundles. BN1 is a security source for bundles 1 and 2. BN3 is the bundle destination and, thus, security acceptor for bundle 1, a security verifier for bundle 2, and a security source for bundle 3. BN4 is the acceptor for bundles 2 and 3.

VII. BPSEC EXAMPLES

This section discusses the use of the BPSec augmentation approach to achieve common security protocol functions that would normally be spread across application, transport, and network layers using various encapsulation schemes. The four examples presented are single-source confidentiality, multiple-source confidentiality, data at rest, and secure content aggregation.

Single-Source Confidentiality

Single-source confidentiality illustrates the familiar concept of an encrypted security tunnel between two endpoints, such as could be implemented with either IPSec or TLS. Similar to IPSec, this tunnel in BP could be instantiated as part of the networking layer covering all bundles traversing a portion of the network. Like TLS, the tunnel could be applied only to bundles with certain destinations or only to certain blocks in the bundle, providing finer-grained discernment of how traffic is encrypted as a function of the applications that generate it.

Multiple-Source Confidentiality

A bundle may travel through different network segments enforcing different security policies. BPSec treats these security policy gateways as security sources and not new encapsulation points for a new security tunnel.

Consider the scenario where a bundle is created with an integrity signature on its primary block. Later, at a gateway node, the payload of the bundle is encrypted. At some point after the payload encryption, but before eventual payload decryption, a new extension block is added to the bundle and also encrypted. This case is illustrated in Figure 13 and represents a stressing case for encapsulation because the "security tunnels" have overlapping endpoints. However, this situation can exist in a networked constellation attempting to adhere to the desirable security properties of topological independence, self-sufficiency, and delay-tolerance.

In this illustration, the scope of the various security tunnels are illustrated over a set of 6 nodes (1-6) comprising the path traversed by the bundle. A stacked representation of transmitted bundle blocks is shown between each node in the diagram. Dashed arrows are drawn from a BPSec security block to its target block.

At the bundle source (1) a BIB is added to the bundle holding an integrity security result over the primary block. Policy at the next node (2) requires that payloads be encrypted and so a BCB is added which carries appropriate parameters and replaces the contents of the payload with cipher text. At node 3, separate policy adds a new extension block to carry additional, encrypted data associated with the data exchange (e.g., a metadata annotation such as a GPS tag[34]). Node 4 ends the payload security tunnel, and the BCB associated with the payload is removed and the payload is decrypted in place. Node 5 processes and removes the added extension block, leaving the bundle to arrive at node 6 in the same condition that it left node 1.

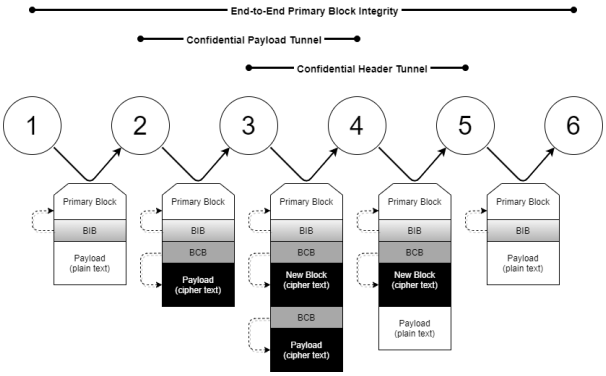

Figure 13 - BPSec provides security via augmentation as well as via encapsulation.

In this example, BPSec provides security operations that target network layer concerns (routing information), transport layer concerns, and application layer concerns while also tolerating changes to node policy.

Security for Data at Rest

As a store-and-forward protocol, bundles may be persisted at nodes for extended periods of time. Because BPSec carries security information as blocks within the bundle, when the bundle is stored the security associated with that bundle is also stored. Encrypted data is never decrypted during transmission until the bundle has arrived at its security acceptor.

This is different from current approaches to network and transport security where the security is associated with the transmission of data. Security protocols are not self-sufficient if they require session information to be present at endpoints. The BPSec concept of security context provides additional information in the security blocks themselves to avoid this occurrence, and those security blocks are also persisted with the bundle.

Secure Content Aggregation

Content caches allow data to be pre-placed where it can be readily accessed. Content aggregators are used frequently on the terrestrial Internet to distribute traffic load across geographic areas. Similarly, sparse networks utilize content caches to reduce data return times, especially in cases where re-transmitting data incurs significant node resources and delays[35].

BP agents must persistently store bundles as part of supporting store-and-forward networking. Because BPSec provides security at rest, the bundle store can be used as a local, secure content cache where the payloads of bundles can be encrypted with BCBs. Extension blocks holding annotative information on these payloads can be added to assist with the organization and retrieval of information in this caching scheme.

VIII. CONCLUSION

Existing security solutions at both the networking and transport layer make assumptions on the nature of the networks they secure. These assumptions include stable topologies and rapid renegotiation or convergence after times of change. These features are rarely present in even the simplest SATCOM architectures. Attempts to apply security to these SATCOM architectures limit the practical system engineering trades associated with these constellations.

A common approach to dealing with different roles and responsibilities in securing a network is encapsulation. Nested encapsulation, or super encapsulation, constrains the performance of a network by adding processing and data overhead. It also constrains the design of the network to have fixed endpoints for the encapsulation and subsequent decapsulation. Forcing networked constellation architectures to reuse terrestrial security protocols may limit the functionality of the constellations themselves.

An alternative to super encapsulation is packet augmentation, wherein security services are flattened in a packet such that security boundaries can be enforced where needed without hiding otherwise unaffected data. However, this approach requires a more featureful secondary header mechanism than provided by either TCP or IP.

The BP, originally designed for use in deep space missions, provides a more expressive extension mechanism. BPSec uses this mechanism to provide a flexible augmentation approach to security. While BPSec can work in an encapsulation scheme, it also can provide security features without incurring the inefficiencies or architectural constraints imposed by those approaches.

References

1. E. J. Birrane, D. J. Copeland and M. G. Ryschkewitsch, "The path to space-terrestrial internetworking," 2017 IEEE International Conference on Wireless for Space and Extreme Environments (WiSEE), Montreal, QC, 2017
2. Birrane, Edward, Vignesh Ramachandran, and Samantha Jacobs. "Security Standards for Space–Terrestrial Internetworks—A Multidimensional Approach to Securing Shared Circuits." Space Operations: Innovations, Inventions, and Discoveries (2014): 375.
3. S. Burleigh, K. Fall, and E. Birrane. Bundle Protocol Version 7. Internet Engineering Task Force (IETF), 9 Mar. 2020, tools.ietf.org/html/draft-ietf-dtn-bpbis-24.
4. Birrane, E., and K. McKeever. Bundle Protocol Security Specification. Internet Engineering Task Force (IETF), 8 Mar. 2020, tools.ietf.org/html/draft-ietf-dtn-bpsec-22.
5. Birrane, Ed, and Soloff, Jason. Designing Delay-Tolerant Applications for Store-and-Forward Networks. Norwood, MA., Artech House, 2020.
6. Alani, Mohammed M. "Tcp/ip model." Guide to OSI and TCP/IP models. Springer, Cham, 2014. 19-50.
7. Doraswamy, Naganand, and Dan Harkins. IPSec: the new security standard for the Internet, intranets, and virtual private networks. Prentice Hall Professional, 2003.
8. RFC4302, S. Kent. "IP Authentication Header S." Kent December (2005).
9. Kent, S. "RFC 4303-IP Encapsulating Security Payload (ESP), 2005." URL https://tools. ietf. org/html/rfc4303 (2006).
10. No, CNSS Policy. "National Policy Governing the Use of High Assurance Internet Protocol Encryptor (HAIPE) Products." (2007).
11. Rescorla, E. "RFC 8446: The Transport Layer Security (TLS) Protocol Version 1.3." Internet Engineering Task Force (IETF) (2018).
12. Rescorla, E., et al., “The Datagram Transport Layer Security (DTLS) Protocol Version 1.3”, Work in Progress, draft-ietf-tls-dtls13-38, May 2020.
13. Iyengar, J., et al., “QUIC: A UDP-Based Multiplexed and Secure Transport”, Work in Progress, draft-ietf-quic-transport-27, February 2020.
14. Thomson, M., and Turner, S., “Using TLS to Secure QUIC”, Work in Progress, draft-ietf-quic-tls-29, June 2020.
15. Thomas, Ludovic, et al. "Google QUIC performance over a public SATCOM access." International Journal of Satellite Communications and Networking 37.6 (2019): 601-611.

16. Space Packet Protocol. CCSDS 133.0-B-1. Blue Book. Issue 1. Washington, D.C.:CCSDS, September 2003.
17. Encapsulation Packet Protocol. CCSDS 133.1-P-2.1. Pink Book. Issue 1. Washington, D.C.:CCSDS, August 2019.
18. Fischer, Daniel, et al. "Finalizing the CCSDS space-data link layer security protocol: setup and execution of the interoperability testing." AIAA SPACE 2015 Conference and Exposition. 2015.
19. Border, John, et al. "Rfc3135: Performance enhancing proxies intended to mitigate link-related degradations." (2001).
20. Scott, K., and S. Burleigh. "RFC 5050: bundle protocol specification." IRTF DTN Research Group (2007).
21. Cerf, Vinton, et al. "RFC 4838, delay-tolerant networking architecture." irtf Dtn research group 2.4 (2007): 6.
22. CCSDS Bundle Protocol Specification. CCSDS 734.2-B-1. Blue Book. Issue 1 Washington, D.C.:CCSDS, September 2015.
23. Caini, C., et al. "TCP, PEP and DTN performance on disruptive satellite channels." 2009 International Workshop on Satellite and Space Communications. IEEE, 2009.
24. Caini, C. "Delay-tolerant networks (DTNs) for satellite communications." Advances in Delay-Tolerant Networks (DTNs). Woodhead Publishing, 2015. 25-47.
25. Celandroni, Nedo, et al. "On the applicability of reliable transport protocols in satellite delay tolerant and disruptive networks." International Journal of Satellite Communications and Networking 32.2 (2014): 141-161.
26. Y. Kim, J. Jo, R. Harkanson and K. Pham, "TCP-GEN Framework to Achieve High Performance for HAIPE-Encrypted TCP Traffic in a Satellite Communication Environment," 2018 IEEE International Conference on Communications (ICC), Kansas City, MO, 2018, pp. 1-7.
27. Coonjah, Irfaan, Pierre Clarel Catherine, and K. M. S. Soyjaudah. "An Investigation of the TCP Meltdown Problem and Proposing Raptor Codes as a Novel to Decrease TCP Retransmissions in VPN Systems." Information Systems Design and Intelligent Applications. Springer, Singapore, 2019. 337-347.
28. Honda, Osamu, et al. "Understanding TCP over TCP: effects of TCP tunneling on end-to-end throughput and latency." Performance, Quality of Service, and Control of Next-Generation Communication and Sensor Networks III. Vol. 6011. International Society for Optics and Photonics, 2005.
29. Djeddai, Lekhemissi, and Rong Ke Liu. "IPSecOPEP: IPSec over PEPs architecture, for secure and optimized communications over satellite links." 2016 7th IEEE International Conference on Software Engineering and Service Science (ICSESS). IEEE, 2016.
30. Thappa, Veena Kumari, Bindu Sharma, and Abhishek Sharma. "High speed 2× 10 Gbps WDM enabled inter-satellite optical wireless communication link." Journal of Optical Communications 1.ahead-of-print (2019).
31. Grubbs, Rodney P. "Enabling Technologies for Deep Space Motion Imagery." 2018 SpaceOps Conference. 2018.
32. Symington, S., et al. "RFC 6257, Bundle Security Protocol Specification." IRTF Delay-Tolerant Networking Research Group (2011).
33. CCSDS Streamlined Bundle Security Protocol Specification. CCSDS 734.5-R-1. Issue 1. Washington, D.C., March 2018.
34. Symington, Susan. Delay-Tolerant Networking Metadata Extension Block. RFC 6258, May, 2011.
35. Li, Chao, et al. "Geo-DMP: A DTN-Based Mobile Prototype for Geospatial Data Retrieval." ISPRS International Journal of Geo-Information 9.1 (2020): 8.