Consultative
Committee for
Space Data Systems

**REPORT CONCERNING SPACE
DATA SYSTEM STANDARDS**

Next Generation Space Internet (NGSI)

CCSDS 730.0-G-1

GREEN BOOK

April 2003

AUTHORITY

Issue:	Current Issue
Date:	April 2003
Location:	Matera, Italy

This document has been approved for publication by the Management Council of the Consultative Committee for Space Data Systems (CCSDS) and reflects the consensus of technical panel experts from CCSDS Member Agencies. The procedure for review and authorization of CCSDS Reports is detailed in the *Procedures Manual for the Consultative Committee for Space Data Systems*.

This document is published and maintained by:

CCSDS Secretariat
Office of Space Communication (Code M-3)
National Aeronautics and Space Administration
Washington, DC 20546, USA

FOREWORD

This Report describes the proposed Next Generation Space Internet (NGSI) architecture.

Through the process of normal evolution, it is expected that expansion, deletion, or modification to this Report may occur. This Report is therefore subject to CCSDS document management and change control procedures which are defined in the *Procedures Manual for the Consultative Committee for Space Data Systems*. Current versions of CCSDS documents are maintained at the CCSDS Web site:

http://www.ccsds.org/

Questions relating to the contents or status of this report should be addressed to the CCSDS Secretariat at the address on page i.

At time of publication, the active Member and Observer Agencies of the CCSDS were:

Member Agencies

– Agenzia Spaziale Italiana (ASI)/Italy.
– British National Space Centre (BNSC)/United Kingdom.
– Canadian Space Agency (CSA)/Canada.
– Centre National d'Etudes Spatiales (CNES)/France.
– Deutsches Zentrum für Luft- und Raumfahrt e.V. (DLR)/Germany.
– European Space Agency (ESA)/Europe.
– Instituto Nacional de Pesquisas Espaciais (INPE)/Brazil.
– National Aeronautics and Space Administration (NASA)/USA.
– National Space Development Agency of Japan (NASDA)/Japan.
– Russian Space Agency (RSA)/Russian Federation.

Observer Agencies

– Austrian Space Agency (ASA)/Austria.
– Central Research Institute of Machine Building (TsNIIMash)/Russian Federation.
– Centro Tecnico Aeroespacial (CTA)/Brazil.
– Chinese Academy of Space Technology (CAST)/China.
– Commonwealth Scientific and Industrial Research Organization (CSIRO)/Australia.
– Communications Research Centre (CRC)/Canada.
– Communications Research Laboratory (CRL)/Japan.
– Danish Space Research Institute (DSRI)/Denmark.
– European Organization for the Exploitation of Meteorological Satellites (EUMETSAT)/Europe.
– European Telecommunications Satellite Organization (EUTELSAT)/Europe.
– Federal Service of Scientific, Technical & Cultural Affairs (FSST&CA)/Belgium.
– Hellenic National Space Committee (HNSC)/Greece.
– Indian Space Research Organization (ISRO)/India.
– Institute of Space and Astronautical Science (ISAS)/Japan.
– Institute of Space Research (IKI)/Russian Federation.
– KFKI Research Institute for Particle & Nuclear Physics (KFKI)/Hungary.
– MIKOMTEK: CSIR (CSIR)/Republic of South Africa.
– Korea Aerospace Research Institute (KARI)/Korea.
– Ministry of Communications (MOC)/Israel.
– National Oceanic & Atmospheric Administration (NOAA)/USA.
– National Space Program Office (NSPO)/Taipei.
– Space & Upper Atmosphere Research Commission/Pakistan.
– Swedish Space Corporation (SSC)/Sweden.
– United States Geological Survey (USGS)/USA.

DOCUMENT CONTROL

Document	Title	Date	Status
CCSDS 730.0-G-1	Next Generation Space Internet	April 2003	Current Issue

CONTENTS

CONTENTS (continued)

1 INTRODUCTION

1.1 PURPOSE

Space missions are evolving away from stovepipe software and single ground station/single spacecraft topologies to Internet-like operations and satellite constellations. This Report outlines a number of approaches that allow Principal Investigators (PIs) with computers on the Internet varying degrees of interaction with onboard systems on Earth-orbiting platforms. This Report then examines in detail a Next Generation Space Internet (NGSI) architecture that provides real-time end-to-end Internet access between PIs and orbiting assets, and shows that a combination of standard Internet technologies yields a promising solution. Additionally, this Report presents minor modifications designed to improve the 'space-friendliness' of the resulting architecture. These modifications are minimal in scope and confined in location to a few routers that can easily be assumed to be under the control of mission operators and, possibly, commercial ground station operators.

1.2 SCOPE

The NGSI architecture efforts are concentrated on providing the mechanisms necessary to allow science users to request and reserve communication resources over the entire path—from their payloads to their laboratories—in a secure manner. In particular the NGSI architecture has been designed to make the most of available spacecraft communication assets, rather than attempting to provide continuous coverage of spacecraft that would not already have such connectivity. As such there are four main components to the proposed Consultative Committee for Space Data Systems (CCSDS) standards-track activities:

a) researching and recommending mechanisms to support dynamic utilization of space link communications services(see annex B);

b) integrating end-to-end resource reservation mechanisms, such as the Resource Reservation Protocol (RSVP), with the dynamic link utilization mechanisms (reference [1]);

c) researching user-transparency via the Mobile Internet Protocol (IP) for real-time user-to-payload interaction (reference [2]);

d) providing efficient end-to-end security and key management mechanisms which take advantage of existing approaches in the terrestrial environment, such as IP Security (IPSEC), and providing 'space link-friendly' approaches for the space segment (reference [3]).

1.3 REFERENCES

The following documents are referenced in the text of this Report. At the time of publication, the editions indicated were valid. All documents are subject to revision, and users of this Report are encouraged to investigate the possibility of applying the most recent editions of the documents indicated below. The CCSDS Secretariat maintains a register of currently valid CCSDS Recommendations.

[1] *Next Generation Space Internet (NGSI)—End-to-End Resource Provisioning for Orbiting Missions*. Experimental Specification for Space Data System Standards, CCSDS 732.5-O-1. Experimental Specification. Issue 1. Washington, D.C.: CCSDS, April 2003.

[2] *Next Generation Space Internet (NGSI)—Supporting Spacecraft IP Mobility*. Experimental Specification for Space Data System Standards, CCSDS 733.0-O-1. Experimental Specification. Issue 1. Washington, D.C.: CCSDS, April 2003.

[3] *Next Generation Space Internet (NGSI)—End-to-End Security for Space Mission Communications*. Experimental Specification for Space Data System Standards, CCSDS 733.5-O-1. Experimental Specification. Issue 1. Washington, D.C.: CCSDS, April 2003.

[4] *Space Communications Protocol Specification (SCPS)—Security Protocol (SCPS-SP).* Recommendation for Space Data System Standards, CCSDS 713.5-B-1. Blue Book. Issue 1. Washington, D.C.: CCSDS, May 1999.

[5] Atkinson, R., *Security Architecture for the Internet Protocol*, RFC 1825, August 1995.

[6] Atkinson, R., *IP Authentication Header*, RFC 1826, August 1995.

[7] Atkinson, R., *Encapsulating Security Payload (ESP)*, RFC 1827, August 1995.

[8] Harkins, D. and Carrel, D., *The Internet Key Exchange (IKE)*, RFC 2409, November 1998.

[9] Braden, R. Ed., et. al., *Resource Reservation Protocol (RSVP)—Version 1 Functional Specification*, RFC 2205, September 1997.

[10] Wroclawski, J., *The Use of RSVP with IETF Integrated Services*, RFC 2210, September 1997.

[11] Terzis, A., Krawczyk, J., Wroclawski, J., and L. Zhang, *RSVP Operation Over IP Tunnels*, RFC 2746, January 2000.

[12] Fuller, V., Li, T., Yu, J., and K. Varadhan., *Classless Inter-Domain Routing (CIDR): an Address Assignment and Aggregation Strategy*. RFC 1519, September 1993.

[13] C. Perkins, *IP Mobility Support*, RFC 2002, October 1996.

[14] Swallow, G., et al. *RSVP-TE: Extensions to RSVP for LSP Tunnels*. Internet Draft, August 2000.

[15] Rosen, E., et. al. *Multiprotocol Label Switching Architecture*. Internet Draft, July 2000.

[16] Case, J., et. al. *A Simple Network Management Protocol (SNMP)*. RFC 1098, May 1990.

[17] Baker, F., et. al. *RSVP Management Information Base Using SMIv2*. RFC 2206, September 1997.

[18] Scott, K. *End-to-End Resource Provisioning for Orbiting Missions*. Internet Draft, April 2001.

[19] Jamoussi, B., et. al. *Constraint-based LSP Setup Using LDP*. Internet Draft, July 2000.

[20] Srinivasan, C., et al., MPLS Label Switch Router Management Information Base Using SMIv2,' Internet Draft, January 2001.

[21] *Proximity-1 Space Link Protocol*. Recommendation for Space Data Systems Standards, CCSDS 211.0-B-1. Blue Book. Issue 1. Washington, D.C.: CCSDS, October 2002.

[22] Rosen, E., et al. *MPLS Label Stack Encoding*. RFC 3032, January 2001.

[23] *Telemetry Channel Coding*. Recommendation for Space Data System Standards, CCSDS 101.0-B-6. Blue Book. Issue 6. Washington, D.C.: CCSDS, October 2002.

[24] Telecommand Part 1—Channel Service. Recommendation for Space Data System Standards, CCSDS 201.0-B-3. Blue Book. Issue 3. Washington, D.C.: CCSDS, June 2000.

[25] *Telecommand Part 2—Data Routing Service*. Recommendation for Space Data System Standards, CCSDS 202.0-B-3. Blue Book. Issue 3. Washington, D.C.: CCSDS, June 2001.

[26] *Advanced Orbiting Systems, Networks and Data Links: Architectural Specification*. Recommendation for Space Data System Standards, CCSDS 701.0-B-3. Blue Book. Issue 3. Washington, D.C.: CCSDS, June 2001.

2 FUTURE REQUIREMENTS

2.1 FUTURE MISSIONS

Internetworking technology has radically changed the way people work and communicate on Earth. Through the Internet, users have nearly instant access to a broad variety of resources including information, computational power, and data storage. Further, standard network interfaces for manipulating these resources make access to a local resource the same as access to one half-way around the world. Finally, standard interfaces between the application and communications stack make it easy for users to write new programs that communicate with each other via networks.

NASA's Advanced Information Systems Technology (AIST) program, part of the Earth Science Enterprise, is bringing these same advances to Earth-observing science instruments. AIST's long-term goal is to enable a highly interconnected 'sensor web' of satellites that allows long-range and detailed prediction/analysis of the Earth and its biosphere. The target is an integrated system of orbiting sensors that:

a) can be easily controlled/reconfigured from standard workstations on the Internet, allowing a large number of widely distributed users to place requests for information via the Internet, which the sensor web will arbitrate and respond to;

b) operates semi-autonomously, detecting events of interest, re-targeting multi-spectral sensors to observe them, and delivering data products to the ground;

c) operates in a coordinated manner, with the various orbiting sensor assets communicating among themselves to assign sensors to targets and schedule contemporary observations;

d) automatically transmits sensor data to terrestrial consumers via the Internet, although the destination of the data may be different depending on the task or event observed; for example, crop data may be transmitted to the Department of Agriculture (or directly to local farmers), while ocean temperature values might go to the National Oceanic and Atmospheric Administration (NOAA) and the Navy;

e) can process data on orbit, possibly collaboratively among satellites, to reduce the volume of data downlinked to Earth, where data from different sources may be further processed into finished information products.

Current operations procedures simply cannot accommodate this dynamic communications model. Under the current scheme, all aspects of communication are carefully and manually scheduled and managed. Direct access to satellites via the Internet is avoided for security reasons and, while IP is used as a bearer protocol to stage information at various points in the network, its primary application end-to-end has been for testing and demonstration purposes.

This Report describes elements of the communications infrastructure required to interconnect orbiting sensors and the Internet. By extending and modifying a number of Internet technologies, communications resources can be used efficiently while maintaining compatibility with the terrestrial Internet. This Report describes mechanisms for both ground-based controllers and onboard instruments to:

a) maintain connectivity as spacecraft change points of attachment to the ground;

b) request and reserve system resources throughout the entire communications path, to ensure that critical data are not lost because of network congestion, and

c) secure both control of the sensor web(s) and their data.

2.2 MODES OF OPERATION (ARCHITECTURAL OPTIONS)

2.2.1 CURRENT SPACECRAFT OPERATIONS

Current spacecraft operations contain a mix of real-time and non-real time spacecraft access. The purpose of this research is not to make a recommendation for either of these paradigms but, rather, to put in place mechanisms that allow science users of spacecraft to select that mix of real-time and non-real-time use that provides the best science. The approaches that provide different levels of interactivity are discussed in subsections 2.2.2 through 2.2.5.

2.2.2 REAL-TIME END-TO-END ACCESS

There has been much interest recently in providing access to orbiting assets from hosts on the Internet, essentially extending the Internet to orbit. This would provide PIs real-time access to their instruments for both commanding and data gathering. In addition, providing access via the Internet would allow instrument designers and builders to test their instruments in the laboratory using the same communications techniques that will be used on orbit. A drawback to this approach is that it requires the PI to be cognizant of many aspects of spacecraft operation that he might not otherwise need to know. For example, for real-time communications, PIs would have to know the communications schedule between the ground and the orbiting assets. Unless this connectivity is constant, the advantages to the PI of having real-time interaction with his payload may be largely offset by the reality of having to interact with payload during odd hours.

2.2.3 DELAY TOLERANT NETWORKING

Delay Tolerant Networking (DTN) is an emerging area of research that deals with providing communications service in severely stressed environments. One of DTN’s main objectives is to transparently handle cases where the network becomes partitioned so that real-time end-to-end communication is not possible. Note that when the network is disconnected, Internet Protocol, and all protocols based upon it, will fail. This happens because if the network remains partitioned for any significant period of time, all IP datagrams trying to cross the partition will be dropped as they reach the partition boundary. DTN handles this by defining a message-based store-and-forward overlay network. When the next hop for a message is unavailable, the message simply queues in a DTN router until connectivity is restored and it can move towards its destination. Using DTN, a user is free to launch a message at an orbiting host at any time, regardless of whether or not there is connectivity between the orbiting network and the ground. This frees the user from having to know the details of the communications schedule, with the caveat that with DTN there is no expectation of real-time delivery. In addition, because the DTN architecture uses a new set of protocols in the overlay network, it requires either new applications for command and data return, or a proxy-like service between Internet and DTN protocols. The new application development required for DTN is not too onerous, as special-purpose communications and processing applications are not uncommon today. A set of 'standard' DTN applications for services such as file transfers also provides a strong base of reusable communications tools.

2.2.4 MANAGED 'STAGED' ACCESS

Another possible architecture that is attractive from a user point of view is some sort of 'staged' access, whereby users place commands to be uploaded to spacecraft in a particular location. This could be done, for example, by using File Transfer Protocol (FTP) to send the commands to a particular directory, or loading them into a secure Web-based form. Data returned from the instruments could either be sent directly to interested parties or placed into a repository, with the location of the data mailed or posted to a Web page. This is attractive from a user's point of view because it shields the user from having to know the details of the communications, such as when contact is available and how much bandwidth can be used. This does not solve the problem, however, since it does nothing to mitigate the management of communications assets, both within an orbiting constellation and between that constellation and the ground.

3 AN ARCHITECTURE FOR END-TO-END COMMUNICATIONS BETWEEN INTERNET AND ORBITING HOSTS

3.1 GENERAL

This section presents an end-to-end architecture for communication between hosts on the terrestrial Internet and orbiting hosts, such as instruments and spacecraft. The basic architecture can be realized entirely with standard Internet protocols. The Internet solutions, however, have been designed for an environment where round trip times are low and bandwidth is relatively cheap. As such they do not perform particularly well in a space environment, where round trip times to spacecraft are longer, and bandwidth is considered a precious resource. This section contains a number of adaptations of the Internet protocols that take advantage of the nature of space communications and existing CCSDS work to improve efficiency and reduce overhead.

This section explores the real-time end-to-end access option described in subsection 2.2.2 as a means of providing end-to-end communications between hosts on the terrestrial Internet and orbiting assets. Note that due to the nature of IP routers, end-to-end communications also implies real-time communications, since if the orbiting assets were partitioned from the terrestrial network, packets would be dropped at the interface.

3.2 ARCHITECTURE

Figure 3-1 shows the proposed architecture for end-to-end communications.

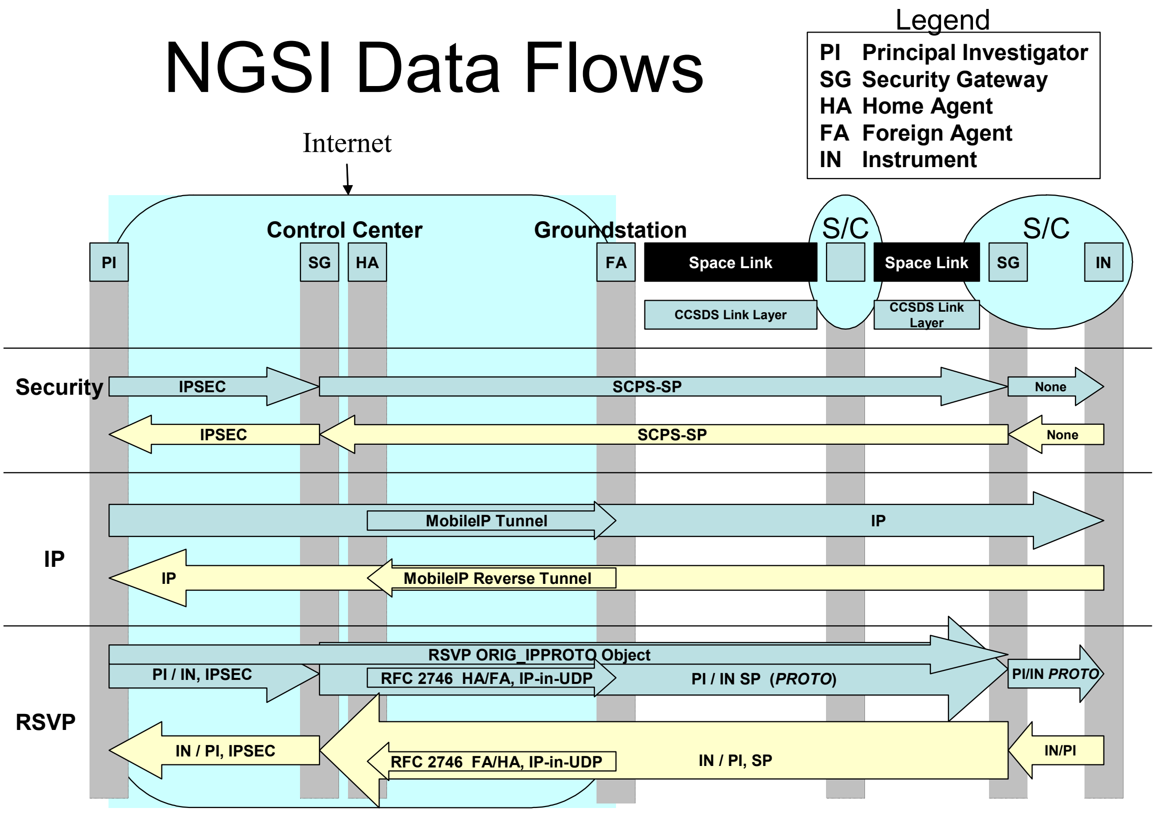


Figure 3-1: NGSI Architecture

The physical elements are shown at the top of figure 3-1. These include:

a) a principal PI;

b) a control center housing a security gateway and a MobileIP home agent;

c) a ground station hosting a MobileIP foreign agent (possibly a space-based network);

d) a destination spacecraft with a security gateway and one or more addressable end systems.

The PI may be connected to the control center via the Internet, and the control center may access one or more ground stations via the Internet.

Different views of the communications path related to the technologies proposed for use are shown in the lower part of figure 3-1. For example, the slice labeled ‘Security’ shows the various security protocols in use at different points in the communications path. The PI can use standard IPSEC to protect data in transit to the control center, which houses a trusted security gateway (discussed in subsection 3.3). The gateway translates the IPSEC to the more bit-efficient SCPS-SP security that traverses the space link. The middle slice shows the MobileIP tunnels used to forward datagrams between the home and foreign agents, and the bottom slice shows the various source/destination/IP protocol tuples used by RSVP to identify data flows.

For the RSVP slice, the notations in the arrows (e.g., PI/IN, IPSEC) refer to the source and destination IP addresses and IP protocol (IPPROTO) number of packets for which resources are reserved at that point. Thus as far as RSVP is concerned, a reservation for data flowing from PI to space in the portion of the network path between the control center and the ground station would carry an RSVP SESSION object reserving resources for packet from the home agent to the foreign agent of type User Datagram Protocol (UDP) (IP-in-UDP encapsulation).

All of the technologies described in this subsection operate at the IP layer and above. In particular they do not require any support from the underlying data link technology. Figure 3-1 also shows CCSDS data links for both the space-to-space and space-to-ground communications. No special modifications to existing CCSDS protocols are necessary, since these protocols already have the ability to carry IP datagrams.

3.3 SECURITY

3.3.1 GENERAL

Spacecraft connected to the Internet will present an irresistible lure to hackers, opening the door to all manner of security threats, including unauthorized disclosure of data, unauthorized modification of data, and Denial of Service (DoS) attacks. Rigorous security measures, designed to ensure that only authorized users are allowed access to the space links and the spacecraft themselves, must be employed if spacecraft are commanded via the Internet.

3.3.2 OVERHEAD

The Internet Engineering Task Force (IETF) Internet Protocol Security (IPSEC) working group has developed a set of security protocol standards that are just now being widely deployed on the terrestrial Internet. One drawback to using IPSEC for space missions is the additional overhead involved (i.e., a minimum of 10 bytes per IP packet). While this may not seem like much, the acknowledgement stream for a Transmission Control Protocol (TCP) connection typically contains 40, 48, or 52-byte packets, so that 10 bytes represents approximately 20% additional overhead.

3.3.3 CCSDS SECURITY LAYER

The CCSDS has developed a suite of protocols that parallel the Internet stack, but which have been extended and/or optimized for the space environment. The CCSDS security layer, known as Space Communications Protocol Standards-Security Protocol (SCPS-SP), (reference [4]) is a functional cousin to IPSEC (references [5] through [7]) and contains most of IPSEC's capabilities, but with only 2 bytes of overhead per IP packet. This makes it a prime candidate for use in the space segment. The reduced overhead has its price, however, and SCPS-SP is not interoperable with IPSEC. The NGSI architecture solution is to use a trusted security gateway that can convert between IPSEC and SCPS-SP.

3.3.4 TRUSTED GATEWAY

Figure 3-2 shows a 'trusted gateway' configuration that can manage IPSEC security on one side and SCPS-SP-style security on the other. The gateways are termed 'trusted' because in order to convert between the two security protocols, the data must be momentarily 'in the clear' inside the gateway box itself, and the gateway must hold both the IPSEC and SCPS-SP encryption keys.

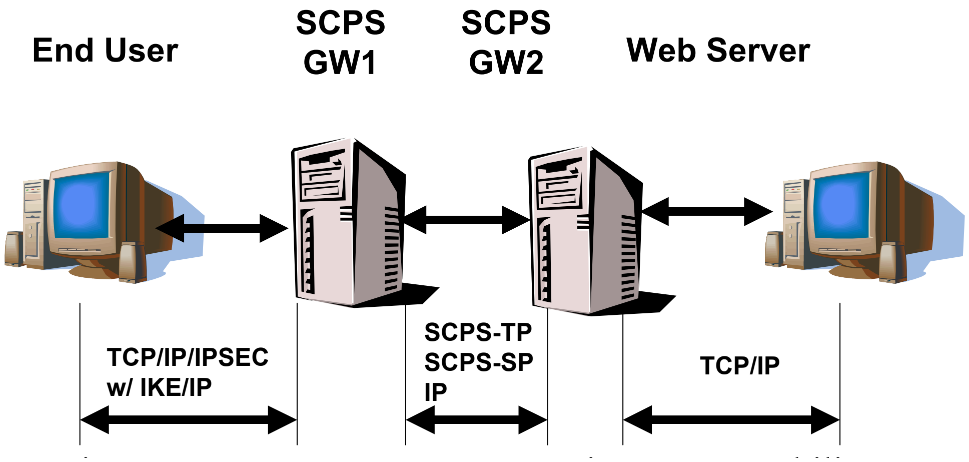


Figure 3-2: Trusted Gateways Translate Between IPSEC and SCPS-SP

3.3.5 KEY MANAGEMENT PROTOCOLS

3.3.5.1 General

A further issue complicating the use of security to protect orbiting assets is the lack of a bit-efficient key management protocol. Unlike security protocols, key management protocols generally do not add per-packet overhead. Instead they are run 'out of band' periodically to distribute cryptographic information. We have investigated a number of key exchange protocols being considered for use in the Internet to determine if any could either be adopted 'as-is' or adapted (like SCPS-SP) for the space environment.

3.3.5.2 Security Associations

Both the IPSEC and SCPS-SP protocols require the creation of *security associations*. A security association is the result of a negotiation between two parties who wish to communicate securely. Therefore, it would appear to make the most sense for the space community to either adopt the Internet Key Exchange (IKE) (reference [8]) as it presently exists, or develop a minimal profile for its use in a space communications environment while maintaining interoperability with the rest of the Internet.

3.3.5.3 Aggressive Exchange

The most promising candidate is an operational mode of IKE termed the 'aggressive exchange'. 'Aggressive Exchange' allows IKE security associations, key exchanges, and authentication payloads to be transmitted together in a single IKE message. This mode reduces the number of round-trips required to establish a security association and key exchange, and greatly mitigates the additional latency of space communications. A drawback of the aggressive exchange mode is that with it users must forgo identity protection. In the usual IKE/Internet Security Association and Key Management Protocol (ISAKMP) mode of operation, identities are exchanged only after a common shared secret key has been used to establish a secure communications channel. In this way the identity exchange is protected. However, when using an 'aggressive exchange,' there is no secure communications channel in place to protect the identity exchanges. The definition of the 'aggressive exchange' also allows only a single proposal and a single transform to be 'negotiated' (i.e., no choices are allowed).

Despite the loss of authenticated identity and the inability to send multiple proposals, security associations and key exchanges using IKE's aggressive exchange would still be interoperable with the ground-based Internet. This means that there appears to be a way to implement an existing Internet standard in a space communications environment in a bandwidth-preserving manner, while still maintaining compatibility with the ground.

3.4 RESOURCE RESERVATION

3.4.1 RESOURCE RESERVATION TO PREVENT CONGESTION LOSS

Resource reservation can vastly improve the amount of science a mission can perform by eliminating loss due to congestion. By reserving the resources needed for a particular flow along the entire communications path, the network can ensure that other cross-traffic cannot interfere with the protected flow.

TCP, the Internet protocol that supports the vast majority of applications today, fares particularly poorly in the face of loss. TCP is an end-to-end protocol running between the data source and destination (instrument and PI in figure 3-1). Thus any loss, even if it occurs in the terrestrial network, will cause data retransmission that originates at the instrument and crosses the space link. In addition to the lost bandwidth of the retransmission, TCP will respond to the loss by lowering its transmission rate (in order to unload the congested router). This may cause under-utilization of the space link and a lower total data return. Because the losses that trigger retransmission and a reduction in data transmission rate may occur outside the purview of the mission operations team (in the middle of the Internet, for example), simply managing or over-engineering the space segment will not alleviate the problem.

In addition to improving communications efficiency by reducing or eliminating congestion-based losses, resource reservation can provide guaranteed Quality of Service (QoS). That is, resource reservation can be used to bound communications parameters such as delay and jitter. This greatly simplifies the design of control loops such as those needed by constellations of spacecraft acting in tandem. Using standard IP, message delays in such a system would be unknown, and the messages themselves more subject to loss.

3.4.2 RESOURCE RESERVATION PROTOCOL (RSVP)

The Resource Reservation Protocol (RSVP) (references [9] through [11]) designed by the Internet community is an end-to-end signaling protocol that allows users to express their requirements to the network, and lets the network inform users as to whether or not those requirements can be met. When coupled with a bandwidth allocation mechanism such as that described below, RSVP can provide users with a mechanism for allocating communications resources along the entire path, from source to destination.

Figure 3-3 shows OPNET simulation results of a data transfer between an orbiting instrument and a terrestrial host. The horizontal axis reflects time, and the vertical axis reflects total data delivered. The heavy straight line on the left shows the data transfer with RSVP. In this case terrestrial cross-traffic in the model could not interfere with the space-to-ground data transfer. The curved line extending to the right shows the data transfer without RSVP. Here cross traffic on the ground necessitated retransmissions from the instrument. Because they occur between the end hosts, these retransmissions consumed bandwidth on the space link that could have been used for other data.

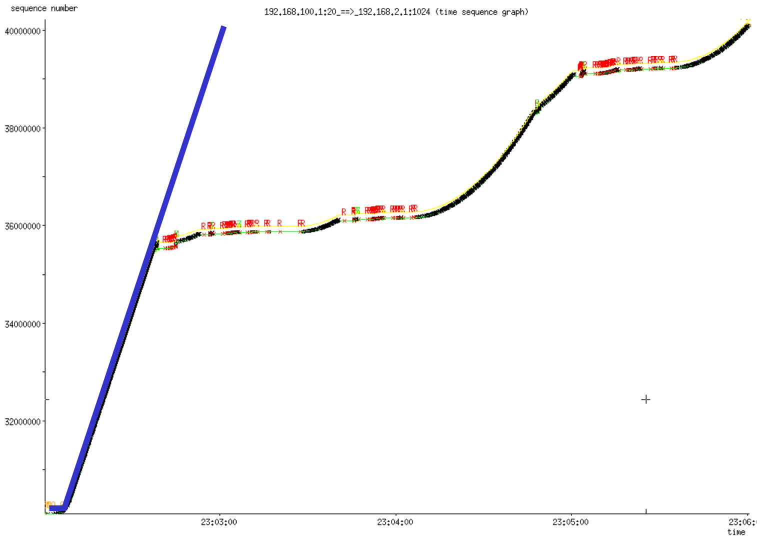


Figure 3-3: RSVP Protects Against Congestion Loss

3.4.3 RSVP AS A RESOURCE ARBITER

In addition to providing a mechanism to eliminate congestion-based data loss, RSVP also provides a mechanism to automatically arbitrate resource usage. Multiple applications vying for the same communications resources all make their requests through RSVP. Those applications that succeed can send data, and those that fail are forced to wait (or to send their data using an ‘unreserved’ service class). In either case, applications whose resource requests are granted will receive a very high quality of service from the network.

3.4.4 A NEW RSVP OBJECT TYPE SUPPORTING PROTOCOL TRANSLATING GATEWAYS

The only modification required beyond the current RSVP requests for comments is a new RSVP object type to carry the end system IP protocol through the security gateways. In particular, end systems will set the IP protocol (IPPROTO) field in RSVP PATH messages to reflect the data flow exiting the source—which may be 50 (IPSEC ESP), 51 (IPSEC AH), or 99 (private encryption, used for SCPS-SP)—if encryption is used at the source. The security gateways have enough information to modify this IPPROTO field for data flowing between the gateways. The downstream gateway, however, cannot 'undo' this modification without further information. Thus the NGSI architecture proposes a new 'ORIG_IPPROTO' RSVP Message type. Sources whose data traverse protocol translating gateways include this object type in PATH messages they source. The ORIG_IPPROTO contains enough information to allow a 'downstream' gateway to correctly set the IPPROTO field in outbound RSVP PATH messages, regardless of whether or not encryption is used.

3.4.5 RSVP AND ADVANCED TRANSPORT PROTOCOLS

The purpose of the NGSI project is to allow scientists and engineers to use common programming tools and methods when designing and building spacecraft instruments. In particular, for data communications we are interested in providing access to spacecraft and instruments via the Internet, where the most commonly used data transport protocols are the Transmission Control Protocol (TCP) and the User Datagram Protocol (UDP). By using RSVP as described in this Report, we can ensure that data is not lost in the network due to congestion from other flows.

There remains however the possibility of 'auto-congestion', whereby an RSVP-protected flow strays outside its reservation, resulting in data loss. For some applications this is not an issue. If a videoconferencing application generates data at a constant X bps and has a reservation for X bps, then no congestion will occur. However if there is no bound on the rate at which the application can generate data (file transfers provide a good example), the data source needs to take care to not inject data in such a manner as to violate the reservation. Data that is non-conformant with the reservation (e.g., data that is sent too quickly) may be dropped.

The main difficulty with controlling the rate at which data is sent is that with the standard sockets interface, an application has at most a coarse-grained notion of the instantaneous outbound rate. Because there is no rate control mechanism in the sockets interface, applications would have to manually attempt to control transmission rate by alternately sending data and pausing. This is complicated by the fact that application writes do not necessarily translate into packets being emitted and, in the case of TCP, retransmissions are invisible to the application. Thus while applications could shoulder the burden of rate-controlling their outbound traffic, it would be cumbersome and inexact at best.

A better solution would be for the transport machinery (usually provided by the operating system kernel) to provide a means to throttle the outbound data rate so as not to exceed any established RSVP reservations. TCP Tranquility (SCPS-TP) provides just such a mechanism via its ability to rate-control traffic, including retransmissions. Extensions to both RSVP and Tranquility would be required to allow Tranquility to match TCP control blocks to reservations and to set the transmission rates based on reservations found.

3.5 IP MOBILITY FOR SPACECRAFT

3.5.1 GENERAL

The IP is the glue that holds the Internet together. IP's common addressing scheme is what ensures that different computers on the Internet can find and communicate with each other, and the forwarding of IP packets by routers is the most basic service provided to end users. Thus when speaking of spacecraft communicating with users on the Internet, we assume the use of IP addressing.

To manage the vast number of addressable systems, IP addresses are grouped together in a topological hierarchy, and almost all forwarding is done via Classless Inter-domain Routing (CIDR) (reference [12]). What is important in the context of this disucssion is that *Internet routing assumes a fixed relationship between IP address and topological location within the Internet*. Low Earth Orbiting (LEO) satellites that communicate with first one ground station then another do not fit this mold. If these ground stations have different locations in the Internet topology (as they almost certainly would if run by competing providers, for example), then from the perspective of a user connected to the Internet, the satellites themselves appear to move within the network. If the IP address of the spacecraft remains the same, then it is nearly impossible to route packets to the correct ground station in order to reach the spacecraft. Packets sent from the spacecraft to locations on the ground can generally be routed without difficulty.

Mobile nodes in the terrestrial Internet have these same problems, and researchers have developed a number of methods for managing mobility from a routing perspective. One of the most popular of these is IP Mobility Support, commonly referred to as MobileIP reference [13]). Mobile IP specifies protocol enhancements for the transparent routing of IP datagrams to mobile nodes in the Internet. Using MobileIP, a mobile node is assigned a fixed home address, and other nodes wishing to reach the mobile node use that address. If the mobile node is away from its home, a care-of address is associated with it that provides information as to its current point of attachment to the Internet. The mobile node registers the care-of address with its home agent, which then tunnels datagrams destined for the mobile agent to this care-of address. The preferred method of acquiring a care-of address is through foreign agents, in which the foreign agent acts as the endpoint of the tunnel, un-encapsulates received datagrams, and delivers them to the mobile node.

Figure 3-4 illustrates the MobileIP data flows when a mobile user is connected to a foreign agent. Here the mobile node has already associated itself with the foreign agent, and the foreign agent has set up an IP tunnel with the mobile's home agent. Data coming from the mobile user is routed as usual, with the mobile's home IP address (A.B.C.G in this case) as the source address. Nodes that want to send data to the mobile send it to A.B.C.G, and the data is routed toward the mobile's home agent, which intercepts the IP datagrams and tunnels them to the foreign agent. The foreign agent un-encapsulates the tunneled datagrams and forwards them to the mobile user.

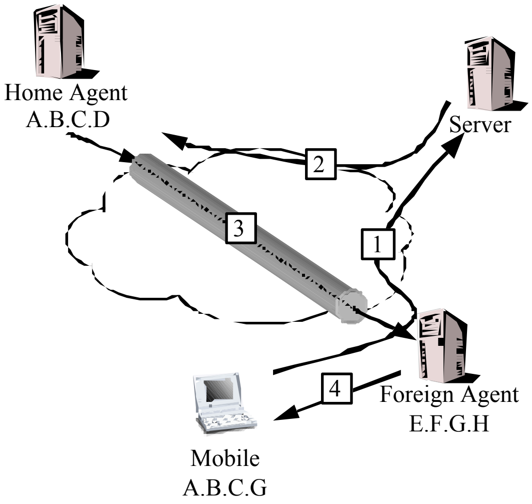


Figure 3-4: MobileIP Data Flows

One extra element is present in figure 3-1 that is not shown in 3-4, namely a *reverse tunnel* from the foreign agent to the home agent. This is necessary to force packets sent by the spacecraft toward the PI to go thorough the control center, and hence the ground-based security gateway. Without this reverse tunnel, IP packets injected into the terrestrial network would follow whatever paths were established by the routing algorithm when making their way towards the PI. These paths would not necessarily go through the control center. Thus if the packets were encrypted by the space-based security gateway using SCPS-SP, they would arrive at the PI as SCPS-SP packets and be uninterpretable.

3.5.2 MOBILE IP EXTENSIONS FOR SPACE OPERATIONS

Mobile IP was designed to permit mobile agents to move randomly while still receiving datagrams at a fixed address. Since Mobile IP cannot predict the movement of mobile nodes, the protocol specifies several mechanisms to associate a mobile node with a mobility agent (i.e., a home or foreign agent). Spacecraft, however, do not move randomly. Contacts between spacecraft and ground stations are scheduled, with *a priori* agreement of established state. If we think of the spacecraft as a mobile node, the ground station as a foreign agent, and the control center as a home agent, then Mobile IP is directly applicable to this environment. Moreover, since the contacts are planned, the mechanisms to associate a mobile node with a locally attached mobility agent are no longer necessary. Eliminating these exchanges will free space link resources during the contact period.

NGSI proposes a set of extensions to MobileIP that allow the ground station to register with the home agent(s) on behalf of the orbiting devices. Using OPNET, we have prototyped these extensions and have examined a scenario with a single satellite making contact with a ground station. As performance measures, we considered the bandwidth saved by having the ground station 'proxy-register' the appropriate IP addresses for the satellite, as well as the time savings involved (since the extensions remove handshaking across the long-delay space link). Figure 3-5 shows the utilization of the space-to-ground link for this scenario. The 15-20 seconds saved reflects the time needed for the handshaking and set-up of traditional MobileIP.

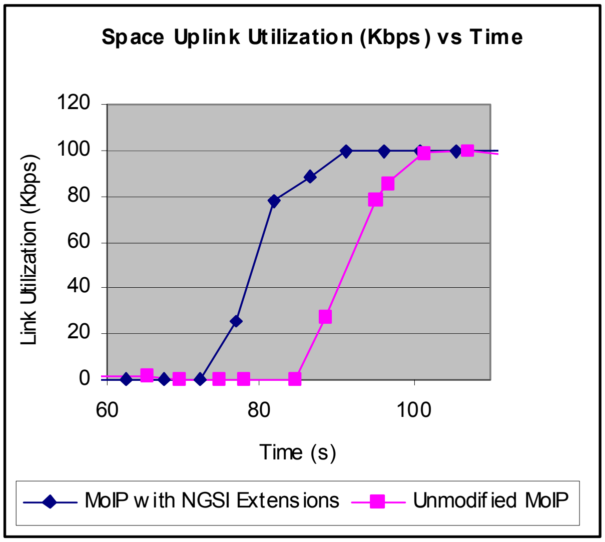


Figure 3-5: Proxy-Registration Savings

3.6 DYNAMIC SPACE LINK COMMUNICATION SERVICES

NOTE – See annex B for a detailed discussion of dynamic space link communications services. See annex C for a related discussion of MPLS stack encoding for CCSDS links. See annex D for a related discussion of CCSDS transfer frames.

The RSVP protocol does not actually implement or enforce resource reservations; it merely provides the mechanism for signaling between applications and network elements. A crucial piece of the NGSI work is to allow a protocol such as RSVP to reserve resources across communications links, either between spacecraft and/or between spacecraft and ground stations. This, coupled with the ability to reserve resources in the terrestrial portion of the path, will allow for end-to-end reservation and, hence, QoS. Because of the large number of missions implementing the CCSDS standards, and because of their international support, we are interested in methods for reserving bandwidth over CCSDS data link layers.

The key technical capabilities needed to allow allocation of communications resources on space links include ‘Traffic Classifying and Filtering’, ‘Dynamic Route Modification,’ and ‘Output Scheduling’. ‘Traffic Classifying and Filtering’ addresses the ability to identify the critical data flows and to filter them into classes for special processing. ‘Dynamic Route Modification’ concerns the ability to change portions of the route of the data flows without reestablishing routes end-to-end. ‘Output Scheduling’ includes the ability to effect special processing for different classes of data vying for resources.

4 DEPLOYMENT ISSUES

4.1 LIMITED 'CUSTOM CODE'

The architecture shown in figure 3-1 involves a minimal amount of custom code, and that code resides only in hosts that could reasonably be assumed to be under the control of mission designers or, possibly, commercial ground station providers. In particular, no modifications are necessary to any machines in the 'Internet' piece of the communications path.

4.2 DEPLOYMENT ISSUES: RSVP AND PROVIDER BOUNDARIES

4.2.1 GENERAL

A number of factors may prevent us from simply setting up RSVP reservations from instruments to PIs across the Internet. For example, most Internet Service Providers (ISPs) block RSVP at their boundaries to prevent requests from outside to reserve resources within their networks. Figure 4-1 shows the architecture we envision, where sensor web users can access the spacecraft directly via the Internet, along with the border routers that separate different ISPs from one another.

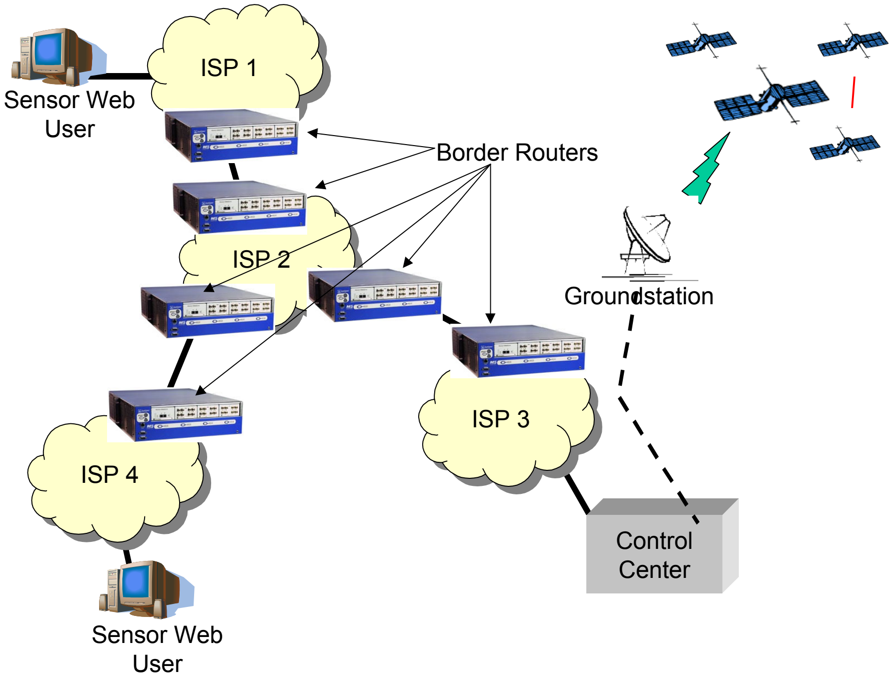


Figure 4-1: Architecture of an Internet-Accessible Spacecraft

NOTE – This subsection describes a number of example deployment scenarios where mission operations has varying degrees of control over the networks connecting the PI to the s\pace link terminus. For simplicity we will assume that the space link terminates at the ground station.

4.2.2 A SINGLE ISP FROM SPACE LINK TERMINUS TO PI

The simplest case is where there is only one closed network connecting the control center and all of the users, and that network is controlled by the space agency. In this case the agency can simply enable RSVP on the routers.

4.2.3 NEGOTIATING WITH ISPs TO ALLOW RSVP TO SELECT USERS

There may be ways to arrange for RSVP to flow between providers, allowing us to set up end-to-end resource reservations. The technical difficulty in allowing RSVP to cross provider boundaries (and providers) depends on exactly how RSVP requests are being blocked, which in turn depends on the make, model, and operating system running in the border router.

Some routers can be configured to allow or disallow RSVP on a per interface basis. In these cases, providers usually disable RSVP on the 'outside' interfaces, effectively preventing RSVP requests from outside the provider from allocating provider resources. To allow resource requests from a space-based sensor web to pass would require RSVP to be enabled on all interfaces in all routes to the destinations. Providers can still restrict the end systems allowed to source RSVP requests by installing IP filters that drop requests from undesirable addresses. RSVP INTEGRITY objects (reference [11]) could be used in conjunction with the filters to further ensure that only authorized users are allowed to reserve resources.

4.2.4 RSVP AND DIFFERENTIATED SERVICES

Section 2 contains a brief overview of differentiated services, or diffserv (reference [9]). Recall that diffserv uses the IP type of service byte to differentiate traffic into different classes of service, and provides different levels of service to each class. By limiting admission to the different network classes, a diffserv network can maintain acceptable QOS for those flows that are admitted.

RFC 2996 (reference [13]) describes a method for integrating RSVP and a diffserv network. Using RFC 2996, standard RSVP signaling is used between the endpoints, and some node (probably at or near the ingress point to the diffserv network) maps RSVP flows to diffserv classes of service (called code points). Note that the RSVP/diffserv mapper may deny RSVP requests if the admission control for the requested diffserv class of service fails. If however the RSVP request, or more specifically if the flow that it defines as mapped to a diffserv code point, can be admitted, then the flow will enjoy the QOS afforded to its code point within the diffserv network. Thus if the networks separating the groundstation from the PI are diffserv networks supporting RFC 2996 (reference [13]), we can still use RSVP to signal QOS.

4.2.5 NON-RSVP, NON-DIFFSERV NETWORKS

Some networks may not block RSVP signaling, but may not react to it either. In these cases the RSVP signaling simply passes through the network unmodified, and no reservations are made in the routers. This might be the case, for example, if traffic were tunneled, as with a Virtual Private Network (VPN) or applied to a Multi-Protocol Label Switched (MPLS) path. In these cases RSVP can still reserve resources in the 'RSVP-capable' pieces of the communications path, but not in the non-RSVP-capable cloud.

An example of this type of service would be if a PI used a VPN to access the control center. In this case there would be no reservation between the control center and the PI, but one might not be necessary. If the VPN provides a guaranteed amount of bandwidth, and if there is no other congesting traffic on the VPN, then so long as the VPN bandwidth exceeds that required by the instrument, there will be no congestion loss.

4.2.6 PROVIDERS THAT COMPLETELY BLOCK RSVP

More problematic would be if the RSVP implementation running on a provider's routers (both border and internal) simply blocked all RSVP requests. For example, some router vendors use RSVP primarily to set up and manage Label Switched Paths (LSPs). Since end hosts generally do not set up label switched paths themselves, an RSVP implementation that ignored requests that did not contain an LSP setup object would effectively block the resource request. If a provider completely blocks RSVP traffic, then RSVP reservations will fail and the applications will have to resort to IP's best effort service.

4.2.7 RESOURCE RESERVATION FROM INSTRUMENT TO GROUND STATION

Even if providers block all RSVP signaling in the terrestrial network, requests originating at instruments might still be able to flow across the space network and through the downlink to the terminus of the space link. At the space link terminus, a router running a modified RSVP daemon could terminate the RSVP dialog, essentially stopping the request from flowing further, but allowing the reservation to succeed. Using this method, resources would be reserved in the space segment, preventing congestion loss from instrument to groundstation. Once the data reaches the ground it can be archived and/or gatewayed onto the terrestrial portion of the network.

Note that even though flows are not protected through the terrestrial Internet, allowing RSVP from instrument to ground station allows RSVP to be used as the arbiter of communications resources in the space segment. Thus congestion can be prevented in the space network, and RSVP can still provide the automated scheduling feature that will be absolutely necessary for managing the large numbers of dynamic flows we expect from orbiting sensor webs.

5 CONCLUSIONS

Current mission operations rely on careful scheduling and management of communications resources. This level of management cannot support future orbiting sensor webs consisting of tens, hundreds, or even thousands of spacecraft, each making independent and autonomous decisions regarding communications. By using internet protocols to connect these sensor webs to the terrestrial Internet we can:

a) support the dynamic communications flows future the sensor webs will generate;

b) use standard interfaces when designing instruments;

c) test communications on the ground with the same protocols that will be used in operation.

The Internet Protocol provides a best-effort service that is not well-suited to spacecraft operations. Recent advances by the Internet community to support mobility, security, and quality of service can be used to re-establish a communications environment suitable for both spacecraft command and data return. This paper describes an architecture that uses MobileIP, RSVP, and secure gateways to provide efficient, end-to-end, secure communications between terrestrial-based and orbiting hosts. Slight space-specific modifications proposed here exploit the unique space environment to provide improved performance. Finally we note that the architecture functions entirely at the IP layer and above. In particular it requires no modification of existing CCSDS protocols, making use only of CCSDS' existing ability to carry IP datagrams.

ANNEX A

ABBREVIATIONS AND ACRONYMS

AIST	Advanced Information Systems Technology
ARQ	Automatic Request Queuing
ASM	Attached Synchronization Marker
BCH	Bose-Chaudhuri-Hocquenghem
CBQ	Class Based Queuing
CCSDS	Consultative Committee for Space Data Systems
CIDR	Classless Inter-domain Routing
CLTU	Command Link Transmission Unit
CRC	Cyclic Redundancy Code
DoS	Denial of Service
DTN	Delay Tolerant Networking
FIFO	First-In-First-Out
FTP	File Transfer Protocol
IETF	Internet Engineering Task Force
IKE	Internet Key Exchange
IP	Internet Protocol
IPSEC	Internet Protocol Security
IPv4	Internetworking Protocol Version 4
IPv6	Internetworking Protocol Version 6
ISAKMP	Internet Security Association and Key Management Protocol
LDP	Label Distribution Protocol
LEO	Low Earth Orbit

LSP	Label Switched Path
LSR	Label Switching Router
MIB	Management Information Base
MPLS	Multiprotocol Label Switching
NGSI	Next Generation Space Internet
NOAA	National Oceanic & Atmospheric Administration
OCF	Operational Control Field
PDU	Protocol Data Unit
PI	Principal Investigator
QoS	Quality of Service
RFC	Request For Comment
R-S	Reed-Solomon
RSVP	Resource Reservation Protocol
SCPS	Space Communications Protocol Specification
SCPS-SP	Space Communications Protocol Specification-Security Protocol
SNMP	Simple Network Management Proocol
TC	Telecommand
TCP	Transmission Control Protocol
TTL	Time To Live
UDP	User Datagram Protocol
VC	Virtual Channel
VCID	Virtual Channel Identifier
VPN	Virtual Private Network

ANNEX B

DYNAMIC SPACE LINK COMMUNICATIONS SERVICES

B1 OVERVIEW

This annex examines ways to provide dynamic space link communications services to space networks so that they can be easily managed and adapt to a variety of communications needs. Emerging techniques being developed for the Internet are discussed as candidates for providing these services over space links using Consultative Committee for Space Data Systems (CCSDS) link protocols. These techniques include:

a) Reserving Resources, using the Resource Reservation Protocol (RSVP);

b) Traffic Classifying and Filtering, using Multiprotocol Label Switching (MPLS);

c) Dynamic Route Modification;

d) Output Scheduling.

B2 BACKGROUND OF DYNAMIC SPACE LINK COMMUNICATIONS SERVICES

In the future there will be networks of spacecraft that will dynamically establish and re-allocate Internet-like connectivity and communications capacity among multiple orbiting satellites, and between those satellites and ground facilities. These networks will function like terrestrial networks in that they will be able to route packet data efficiently from and to almost any end point in the network. Additionally, it is likely that the participants in these networks will be re-configurable spacecraft with programmable sensors and real-time adaptive sensor swarms. There will be a need to adapt rapidly to changing requirements and, at the same time, ensure that critical data flows are not disturbed by the changing environment. The key technical capabilities that are needed include ‘Reserving Resources’, ‘Traffic Classifying and Filtering’, ‘Dynamic Route Modification’ and ‘Output Scheduling’.

Reserving Resources addresses the ability to communicate the need to provide specified data flows with a specific set of resources. Along with this is included the ability to dynamically change the reservations as needed. The Resource Reservation Protocol (RSVP) [26] has been established by the Internet Engineering Task Force (IETF) as a method for distributing resource reservation requests to Internet nodes. Extensions [14] have been proposed for RSVP that also distributed routing information that can be used to dynamically reroute sections along the path of an end-to-end data flow in order to avoid congestion or outages.

Traffic Classifying and Filtering addresses the ability to identify the critical data flows and to filter them into classes for special processing. Multiprotocol Label Switching (MPLS) [15] provides a mechanism for labeling traffic at its source so that it can be easily recognized and segregated for special processing.

Dynamic Route Modification concerns the ability to change portions of the route of the data flows without reestablishing routes end-to-end.

Output Scheduling includes the ability to effect special processing for different classes of data vying for resources. New traffic control mechanisms provide the ability to enforce differentiated services for specified data flows. Through these mechanisms, the concept of Class Based Queuing (CBQ) provides an elegant method for specifying special processing required for sharing resources.

B3 CONCEPTUAL FLOW COMPONENTS

Figure B-1 illustrates a conceptual flow of packetized data through a node in a dynamically managed space network that supports the key technical capabilities described earlier.

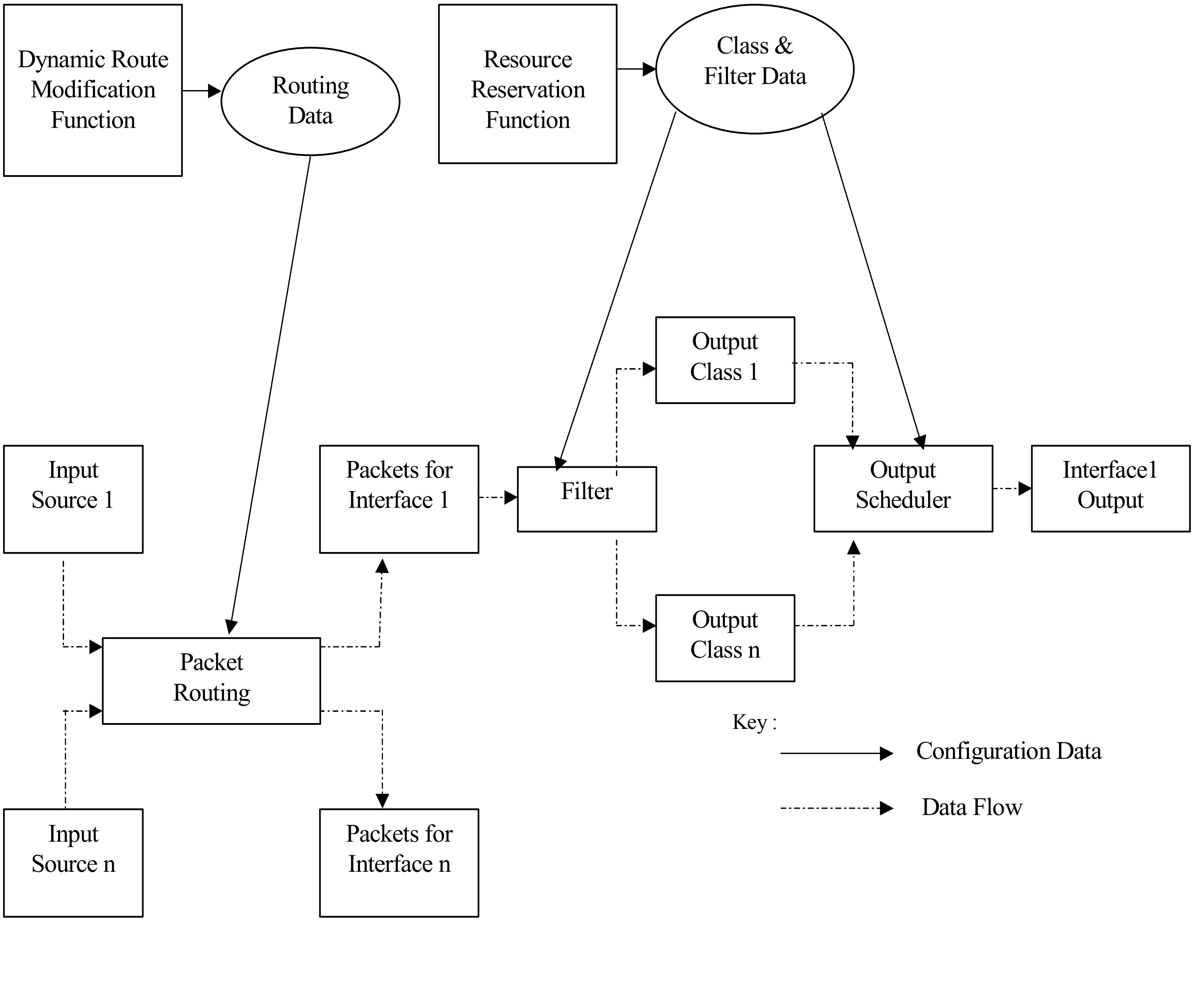


Figure B-1: Conceptual Data Flow Through a Node

A typical node would have multiple sources of input packets. Sources could be local instruments on board the spacecraft which host applications that packetize data to be downlinked to earth or forwarded to another spacecraft. The sources could also be instruments on other spacecraft that are forwarding their packets via the routing capabilities of this node. In any case, the headers of the input packets must contain enough information to determine to which output interface the packet should be routed. In order to determine this, the routing mechanism must have access to appropriate routing data. The dynamic maintenance of this routing data is the primary concern of the Dynamic Route Modification function.

The output interfaces in this discussion can literally represent physical interfaces, CCSDS Virtual Channels, or local applications for which the packets are destined. For most of this discussion we assume that they represent CCSDS Virtual Channels.

Once packets have been segregated for specific output interfaces, they can be passed to a filtering process that separates them into classes and places them into queues based on classification and filter data. The maintenance of this data is the primary concern of the Resource Reservation function. The classification data define the various classes and how members of a class are to be treated, while the filter data define the tests that must be passed in order to assign a packet to a designated class. (The routing capabilities of a node are actually an instance of the filtering function but, because routing is by definition implemented in all nodes and is not a new capability introduced at this time, it is addressed separately in this discussion.)

Packets that have been classified and sorted into output queues can then be forwarded to the output interface. Packets are selected from the awaiting queues and inserted into link layer frames. An output scheduler applies a queue discipline to the assembled queues so that each class gets an appropriate Quality Of Service (QoS). QoS may be expressed as an amount of bandwidth available to a specified class of traffic. This usually means that specified classes will receive an allocated portion of the output bandwidth of a shared link over some time interval, given sufficient demand. Any excess bandwidth not allocated to these classes is distributed according to some scheme appropriate to the mission. The output scheduler determines the amount of bandwidth to reserve for each class from associated reservation data. The maintenance of this reservation data is the primary concern of the Reserving Resources function.

B4 KEY TECHNICAL CAPABILITIES

B4.1 RESERVING RESOURCES

Reserving network resources requires some mechanism for a source application or network management capability to communicate resource requirements to network nodes along a communications path, and for those nodes to effect the reservation of those resources. The communication of the resource requirements can be accomplished with a signaling protocol such as the Resource Reservation Protocol (RSVP) or by explicit management via a management protocol such as the Simple Network Management Protocol (SNMP) [16].

Using RSVP, a sending application can initiate the establishment of a session with a receiving application. The receiving application can establish a reservation for resources along a path from the sender such that a specified QOS can be guaranteed. Using SNMP, a management application can establish a reservation of resources along a path by communicating the need for reservations directly to individual nodes along the path. Reference [17] documents a Management Information Base (MIB) designed to allow RSVP and SNMP to complement each other in the same environment. In any case, the nodes must effect the reservations and maintain reservation state information that can be monitored by the creator of the reservation. Because RSVP does not present a single point of failure, and because there are proposed extensions to RSVP that directly support other functional capabilities, this paper assumes the use of RSVP for the Reserving Resources function. Figure B-2 illustrates how an RSVP entity interacts with other network components.

An RSVP daemon program runs in the sender, the receiver, and intermediate routing platforms. Session establishment, reservation requests, state information, and error messages are communicated from RSVP daemon to RSVP daemon in order to effect an appropriate resource reservation. The details of the RSVP messaging can be found in [26]. The details of the specific RSVP implementation for the CCSDS environment are described in [18].

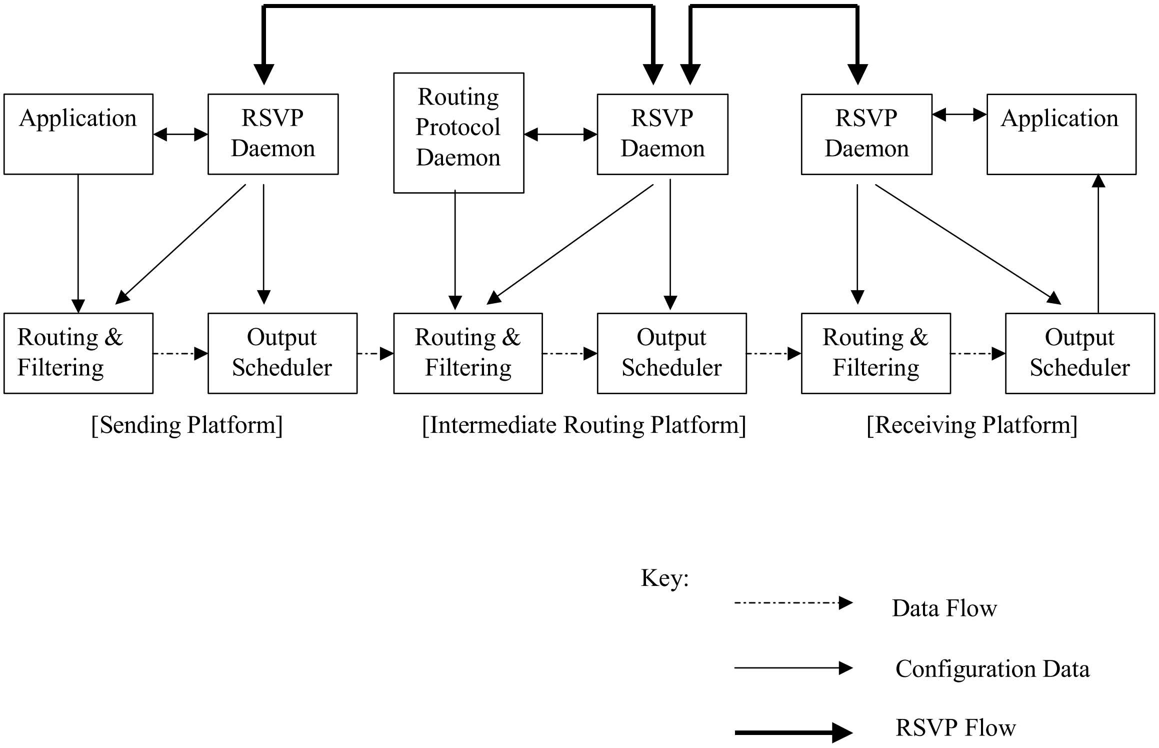


Figure B-2: RSVP Interactions

B4.2 CLASSIFYING TRAFFIC

Mission requirements dictate needs for special treatment of some data flows. As an example, real-time sensor signals require some amount of certainty that their data will arrive at a destination at the same rate at which it was generated. For a network to provide guaranteed bandwidth for this data flow, there must be some means to identify this data and segregate it for special handling. Defining the uniqueness of data flows and specifying the special processing, or QOS, to be utilized for each is termed classification. In a network where Reserving Resources is supported, resources are reserved for specified 'classes' of traffic to provide them with specified QOS. Agreement on the definition of a class is essential to the ability of the network to reserve resources.

There are many possibilities for establishing classification. Examples include classifying traffic by source, destination, network protocol type, transport protocol type, sensitivity to delay or jitter, and many others. Obviously what is important to individual missions should determine the classification scheme.

B4.3 FILTERING

Closely aligned with classification is 'filtering'. This is the actual act of segregating traffic into the classes. A filter is a mechanism for examining packets and determining their class. The most obvious way to assign a packet to a class is by examining its header information for certain characteristics. The most common example of this is basic routing, where a packet is classified for delivery to the correct 'next hop', and subsequent delivery to its final destination, based upon examination of its destination address.

Simple filtering can be accomplished without tying up a great deal of resources at each node. Under many circumstances, however, filtering can become very complex. Complex filtering can require each node in a path to consume excessive resources tearing data structures apart to examine their contents to determine their class designation. The complexity can be appreciated when it is considered that space links can handle CCSDS Source Packets, CCSDS Encapsulation Packets, SCPS Network Protocol Datagrams, and Internet Protocol Datagrams (Ipv4). There may also be a need to handle unencapsulated Internet Protocol Datagrams (Ipv6), unformatted octet blocks, and others.

In some circumstances class may be determined by external factors and, therefore, may not be determined by examining the contents of the packet. To address this situation, the concept of label switching has been devised. With label switching, data units are classified once as they enter a path and are assigned a label that identifies their class. The label is prepended to the data unit as a 'shim' header. As the data units pass along the path, each node need only know how to handle one commonly formatted header, the shim, in order to be able to classify the data unit. The flexibility of this approach allows a packet to be classified based on the shim alone. The contents need not be specifically formatted, and the classification factors need not be based on the packet contents.

In order to effect the label switching technique, each node must support a label switching standard, and there must be a mechanism for communicating label information to each node.

‘Multi-Protocol Label Switching’ (MPLS) is designed for handling label switching in such networks and is the obvious standard for this application. There are competing methods for setting up the label switching in each node. There is a ‘Label Distribution Protocol’ (LDP) [19] specifically designed for setting up nodes with MPLS. Extensions to RSVP have also been proposed that allow MPLS set-up information to be distributed as part of the RSVP messages during the resource reservation process. There is also a proposal for an SNMP MIB [20] that would allow MPLS setup via SNMP. All of these methods will work but, in an environment that uses RSVP, it seems most efficient to consider the use of that method. (Label encoding for MPLS shims is not specified in the standard and can vary by link type. Annex C describes an encoding scheme for CCSDS links.)

B4.4 DYNAMIC ROUTE MODIFICATION

The CCSDS protocols are not well-equipped with routing capabilities. There is nothing in the CCSDS packet data units that is specifically intended to support routing capability. Only the Application Process ID can vaguely be considered routing information, in that it is used to identify the source application of data. To be unique this ID must be combined with a Spacecraft ID from the link layer Transfer Frame. This scheme was designed for individual spacecraft to communicate telemetry directly to a ground station, and for ground stations to communicate telecommands directly to a single spacecraft. No inter-spacecraft communication or packet switching in space was anticipated. The new Proximity-1 protocol [21] expands the inter-spacecraft communication concept somewhat, but still does not approach the needs of a constellation of routers.

Work on the use of conventional packet switching techniques based on the Internet protocols is under study, but attempting to route conventional CCSDS packets and Internet-like packets on the same network would present many problems. This is due mostly to the dependence of many techniques on the conventions of Transmission Control Protocol (TCP)/Internet Protocol (IP) formats and addressing. Label switching protocols like MPLS would provide a simple solution to these problems. With MPLS, routing would be based upon the label shim, and various network layer data units could be mixed.

A single routing scheme based on Label Switched Paths (LSP) would allow RSVP (or SNMP) to dynamically change a portion of the route of an LSP without reestablishing the complete end-to-end route. This would allow the path to bypass any nodes that are having outage or congestion problems. This technique would also allow a route to be reconfigured dynamically as a spacecraft moves from ground system to ground system.

B5 OUTPUT SCHEDULING

B5.1 GENERAL

Once packets are received, assigned to output interfaces, and filtered into class queues, they are candidates for transmission to the next hop. Output scheduling is the mechanism that enforces some discipline over the output so that specified qualities of service are maintained for the traffic flows through the interface. The quality of service of most interest in this application is ‘Bandwidth Availability’. Output scheduling for this instance, therefore, is the management of bandwidth over space link channels. There are various techniques for performing this function that are collectively referred to as Queue Disciplines.

B5.2 MANAGED BANDWIDTH

For the purposes of this document, 'managing bandwidth' refers to the data throughput in terms of bits per second (bps) available, and not the actual management of the physical radio frequency (RF) bandwidth in terms of transmission spectrum.

The amount of network layer data throughput in terms of bps available is determined by the data rate of a physical channel, the allocation of bandwidth to specific Virtual Channels, the overhead presented by Transfer Frame headers and encoding, and the efficiency with which data are inserted into the data fields of the Transfer Frames. In this paper, we address the management of data flows presented to CCSDS space links by network layer protocols. The overhead presented by the data units of the network and higher layer protocols is, therefore, not considered when calculating the throughput. The overhead that may be presented by the use of shim layers such as those used for some label switching protocols is also not considered. All of these sources of overhead would be considered in order to compute the throughput of application data.

B5.3 SPACE LINK CHANNEL STRUCTURES

CCSDS Physical Links are logically partitioned into multiple Virtual Channels. Each Virtual Channel is allocated a portion of the total throughput of a Physical Channel. CCSDS Transfer Frames are assigned to and transmitted via one of the Virtual Channels. Part of the identification of a CCSDS Transfer Frame is its Virtual Channel Identification (VCID). Packets and other application data are packaged in the Transfer Frames to be delivered across a CCSDS space link. It is this capacity of the CCSDS Transfer Frames that constitutes the bandwidth resource that can be managed by the techniques examined in this paper. Figure B-3 illustrates the CCSDS link architecture.

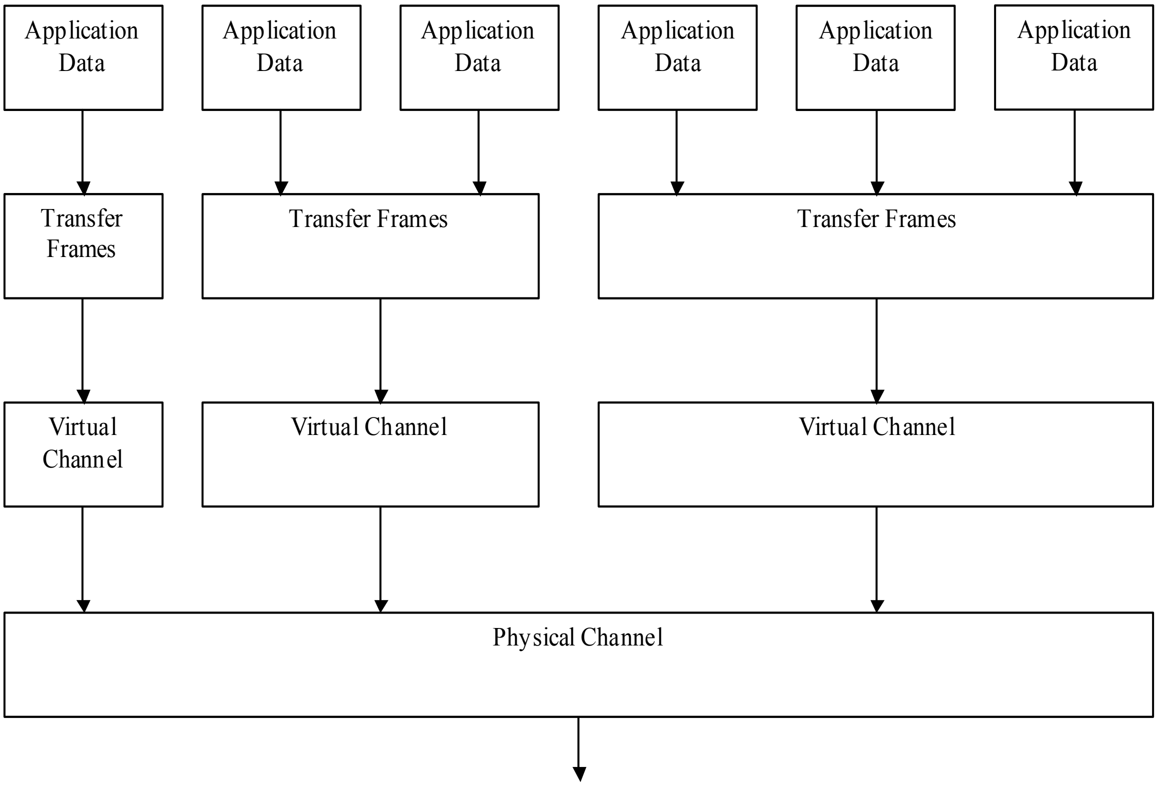


Figure B-3: CCSDS Link Architecture

B5.4 QUEUE DISCIPLINES FOR BANDWIDTH MANAGEMENT

The management of bandwidth as it is addressed in this document is in essence the management of queues of Network Layer Packets waiting to be transmitted across a space link. The queue discipline used determines how these packets are selected from the queues and placed into the data fields of CCSDS Transfer Frames. Annex C describes the various transfer frame formats used with CCSDS Space Links.

When there is only one source of packets and only one queue to be sent, there is no congestion as long as the storage capacity and transmission rate is great enough to handle the flow. A simple First-In-First-Out (FIFO) queue discipline is sufficient for handling the flow. Even when there are multiple sources of packets, a simple FIFO discipline is usually sufficient if there is sufficient storage and transmission capacity. The FIFO discipline is used in most networks.

When there is either insufficient storage, insufficient transmission resources, or there is a need to consider other quality of service aspects for some packets, a more involved discipline is needed. If simple equity is the only consideration, a Stochastic Fairness Queuing (round-robin) discipline is effective. If the problem is insufficient storage space, however, all users usually suffer under this method in that all will have packets discarded when there is no more room to add packets at the end of the queues, since the packets are arriving faster than they can be transmitted.

Sometimes certain packets need special handling and must be isolated into special queues for this purpose. Even packets from the same sources may need to be handled differently. In terms of bandwidth distribution, some packets may need to be transmitted ahead of other traffic. An example of this would be a packet that carries vital network management information that might relieve network congestion if it makes it to its destination. In this case, a simple priority discipline could have the appropriate effect. A single queue of packets, however, could be given top priority, take all of the available bandwidth and choke off all the other flows.

When data needs to flow at a specified rate in order to be useful but should not choke off all other traffic, there is a need to do more than just prioritize the packet flow. There must be a way to ensure that a nominal bandwidth allocation is provided even under highly congested conditions. One discipline that can do this involves isolating bandwidth so that it can only be used by certain queues. This in effect is the discipline used by CCSDS implementations when Virtual Channels are used to isolate traffic flows.

This discipline has two disadvantages. One, it does not utilize the total bandwidth of the link very efficiently. If there is intermittent ('bursty') or small amounts of traffic on an isolated channel, its excess capacity is not shared with other flows that may be constricted. Two, it limits flows to the bandwidth of the isolated channel. Even for short periods, a flow is constricted by the size of the isolated channel. This is a good approach, however, when users want to independently manage their designated portion of the total bandwidth, or there is competition such that one user does not want to give support to another user by lending them bandwidth from their share.

A more efficient use of the available bandwidth would have specific characteristics. Such a scheme would ensure that specified data flows:

a) are given at least a minimal specified throughput;

b) have an opportunity to utilize even more bandwidth if it is available;

c) do not interfere with other flows that also have a guaranteed throughput;

d) do not totally choke off other flows if there is more than enough bandwidth to meet all guaranteed capabilities.

These characteristics can be integrated into bandwidth management by utilizing two disciplines. One discipline is to allocate designated bandwidth to flows (classes) with guaranteed throughput. The other is to distribute any remaining bandwidth in some equitable fashion.

The first discipline must estimate the bandwidth that is being consumed by the guaranteed classes and designate them as being 'over limit' or 'under limit' (with the limit being the minimal bandwidth to which the classes are guaranteed). If they are under limit and have packets (data) to transmit, a packet is extracted from their queue, perhaps using a FIFO mechanism, and submitted to the link. (There is an equity issue to address in that there needs to be a mechanism for allowing all the guaranteed classes appropriate access to the bandwidth. This could be accomplished by checking the guaranteed classes using a round-robin, weighted round-robin, or prioritization mechanism based on the equality of the various guaranteed classes.) If they are over limit or have no data to send, they are ignored until the second discipline is addressed.

Once the bandwidth guarantees have been satisfied (all guaranteed classes are not under limit, or do not have packets to send), the second discipline should be used to distribute any remaining bandwidth. This discipline might be something like round-robin, weighted round-robin, or prioritization based upon implementation needs.

Applying a scheme such as that described above constitutes a low-level implementation of Class Based Queuing (CBQ). More elaborate implementations are structured such that there is a hierarchy of class definitions that can be used to specify different queuing disciplines over various portions of the channel resources. In this paper, analysis is limited to the management of bandwidth over a single channel. This abstract channel might be thought of a single CCSDS Virtual Channel, or it might be applied to a complete Physical Channel. The only hierarchy of classes that might be considered is the hierarchy constructed by virtual channelization. The Virtual Channels might be thought of as one level of isolated classes that may be subdivided into managed subclasses, between which bandwidth can be shared.

ANNEX C

MPLS STACK ENCODING FOR CCSDS LINKS

C1 INTRODUCTION

The utilization of ‘Multi-Protocol Label Switching (MPLS)’[15] over CCSDS data links requires that network layer data units be augmented with a stack of label headers thereby turning them into ‘labeled packets’. These labeled packets are the basic units of data routed over ‘MPLS Label Switched Paths (LSP)’. Communication nodes (including CCSDS nodes) that route LSPs are known as ‘Label Switching Routers (LSR)’. (A CCSDS Node exchanges data with other nodes using CCSDS link protocols.) In order to transmit labeled packets on a particular CCSDS link, LSRs must support an encoding technique which, given a label stack and a network layer data unit, produces a labeled packet that can be multiplexed with unlabeled packets via CCSDS Transfer Frames.

This annex examines the trade-offs involved in the use of various encoding schemes by LSRs in transmitting labeled packets over CCSDS links.

This document also specifies rules and procedures for processing the various fields of the label stack encoding. Since MPLS is independent of any particular network layer protocol, some of these rules and procedures are protocol-independent and are quoted or referenced from other documents. A few, however, differ for CCSDS protocols. In this document, we reference the protocol-independent procedures, and propose the protocol-dependent procedures for CCSDS encoding.

C2 ALTERNATIVES CONSIDERED

Various trade-offs are involved in selecting an encoding scheme for CCSDS labeled packets. There is always a conflict between the need to reduce the amount of processing and storage required at each LSR and the need to minimize the overhead of packet headers required for the labeling. There are advantages to strictly following standards and recommendations, while there are also advantages to optimizing for unique implementations. Section C.2 of this appendix describes an encoding scheme that closely adheres to RFC 3032 [22], ‘MPLS Label Stack Encoding’. The only variation from RFC 3032 is the use of a single Time To Live (TTL) Field (refer to C.2.1). Although this scheme has advantages, it does not minimize the size of the packet header to reduce transmission needs, nor does it reduce the buffer space required at the LSRs.

A more transmission-efficient scheme could be proposed that would reduce the size of label entries further by reducing the size of the label value by four bits and removing the Experimental Use and Bottom of Stack bits. The Experimental Use bits are currently not used, and the function of the Bottom of Stack bit can be handled by use of the Label Depth field. The length of the label entries might even be further reduced by inclusion of a label length field and by making the label field variable length. In any case, there are many possible alternatives with advantages and disadvantages that could be used to optimize the labeled packet header. The one proposed here, however, probably is the best scheme for fitting in with future implementations (with the inclusion of the Experimental Use bits) and for reducing the processing needed by each LSR.

C3 MPLS LABELED PACKET

C3.1 LABEL STACK

The label stack consists of a series of label entries. The top of the label stack appears earliest in the packet, and the bottom appears latest. The stack represents a hierarchy of LSPs. Each entry in the stack indicates an LSP being traversed by the labeled packet.

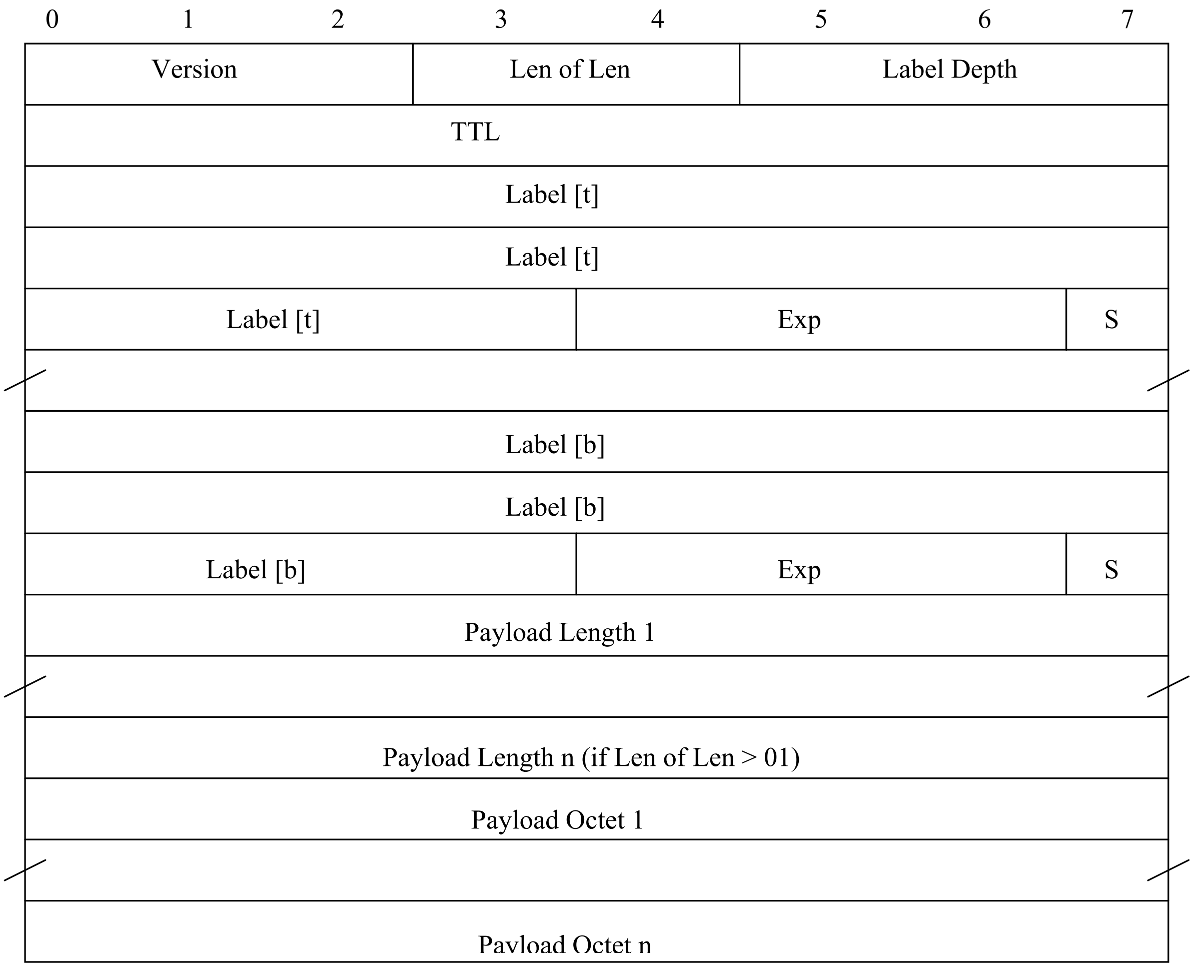
C3.2 LABEL STACK ENTRIES

A label stack is composed of a series of individual label entries. RFC 3032, 'MPLS Label Stack Encoding' (reference [22]), specifies a 32-bit label stack entry format that includes a 20-bit Label value, 3 Experimental bits, 1 Bottom of Stack flag, and an 8-bit TTL field. This format is used as guidance for the label stack encoding proposed in this document for CCSDS MPLS label switching, but is not followed exactly in order to minimize transmission overhead and reduce redundancy. Refer to RFC 3032 for a more extensive introduction to MPLS label stack encoding.

C3.3 CCSDS LABELED PACKET

In this encoding scheme labeled packets consist of a CCSDS-specific packet header and a payload. The header carries the packet's version, time-to-live, label stack, and length.

The payload contains an encapsulated network layer PDU. The proposed CCSDS labeled packet format is illustrated by figure C-1.



Version: CCSDS Packet Version Identifier (TBD)

Len of Len: Length of Length of Payload Field
01 = One octet
10 = Two octets
11 = Four octets

Label Depth: Number of label entries in stack

TTL: Time to Live, 8 bits

Label: Label Value, 20 bits

t: Top of label stack

b: Bottom of label stack

Exp: Experimental Use, 3 bits

S: Bottom of Stack flag, 1 bit

Payload Length: One to four octet value indicating the length of the data unit encapsulated in the labeled packet

Payload: The actual data unit encapsulated in the Labeled Packet

Figure C-1: CCSDS MPLS Labeled Packet

C3.4 RESERVED LABELS

Label values 0 - 15 are reserved as specified in RFC 3032.

C3.5 DETERMINING THE NETWORK LAYER PROTOCOL

The labeled packet header does not contain any field which explicitly identifies the contents of the Payload. The Payload could contain a conventional CCSDS Source Packet, a SCPS Network Protocol Datagram, an Internet Protocol Datagram (Ipv4), an Encapsulation Packet, or any other data unit desired. When the last label is popped from a packet's label stack, the process that acts on the contents of the Payload must have some mechanism to know what the Payload contains. If this process deals with labeled packets that contain only CCSDS Source Packets, SCPS Network Protocol Datagrams, Internet Protocol Datagrams (Ipv4), or CCSDS Encapsulation Packets, the process can determine the type of packet/datagram from the first three bits, Version field, of the Payload. If the process also deals with other PDUs in the Payload such as ‘raw’ Internet Protocol Datagrams (Ipv6), the process must depend on the label value to determine the contents of the Payload. When the paths are established, they must be associated with a specific type of contents. The CCSDS Source Packets, SCPS Network Protocol Datagrams, Internet Protocol Datagrams (Ipv4), and CCSDS Encapsulation Packets may share such association because the process can determine their format from their Version field. Figure C-2 illustrates this determination process.

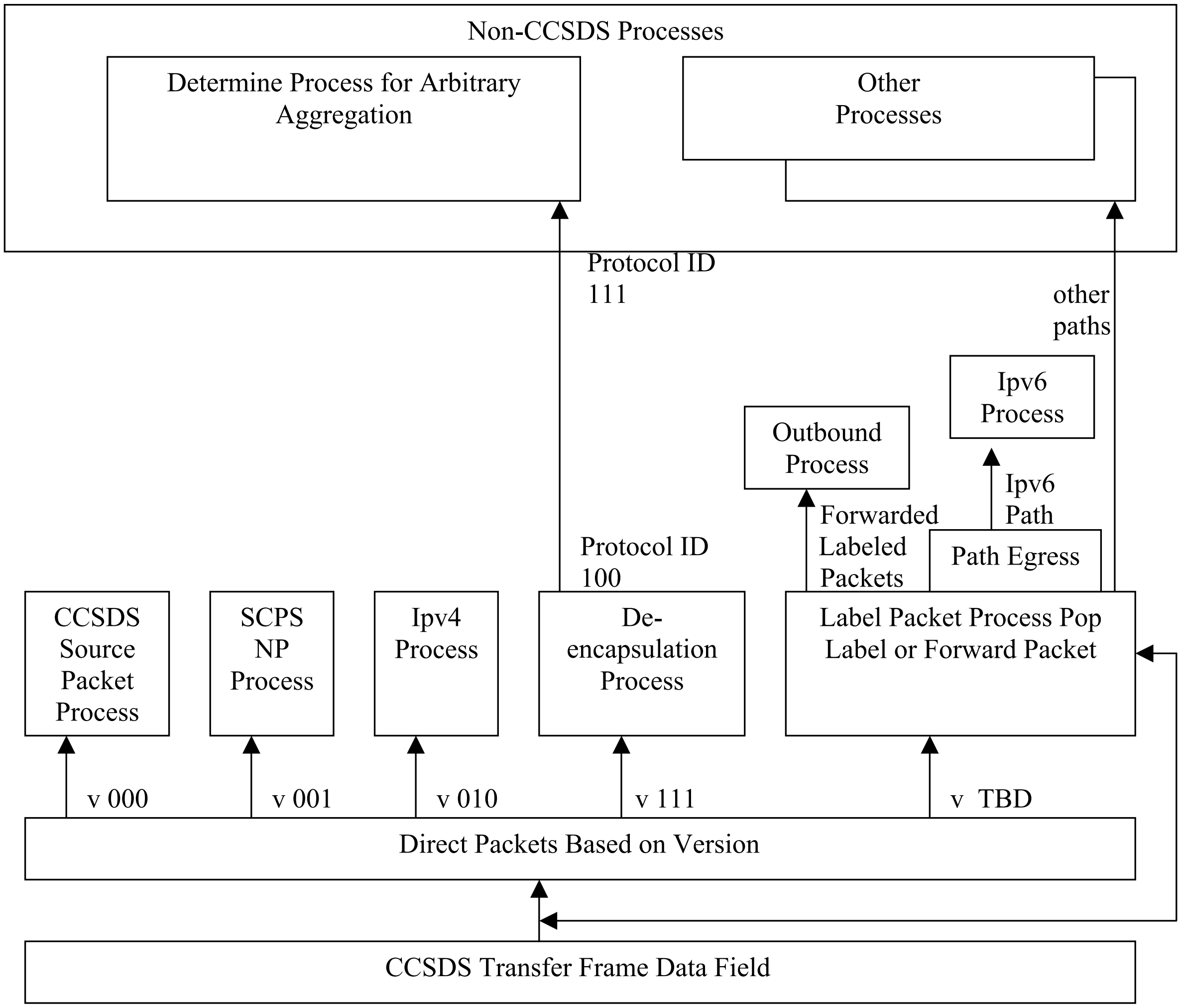

Non-CCSDS Processes
Determine Process for Arbitrary Aggregation
Other Processes
Protocol ID 111
other paths
Ipv6 Process
Outbound Process
Ipv6 Path
Forwarded Labeled Packets
Path Egress
Protocol ID 100
CCSDS Source Packet Process
SCPS NP Process
Ipv4 Process
De-encapsulation Process
Label Packet Process Pop Label or Forward Packet
v 000
v 001
v 010
v 111
v TBD
Direct Packets Based on Version
CCSDS Transfer Frame Data Field

Figure C-2: Determining the Network Layer Protocol

C3.6 PROCESSING THE TIME TO LIVE (TTL) FIELD

C3.6.1 Definitions

The 'incoming TTL' of a CCSDS labeled packet is defined to be the value of the TTL field when the labeled packet is received. This is slightly different from RFC 3032 in that there is only one TTL in the proposed encoding scheme. This has the same effect as the definition in RFC 3032, however, in that RFC 3032 states:

> 'Note that the outgoing TTL value is a function solely of the incoming TTL value, and is independent of whether any labels are pushed or popped before forwarding. There is no significance to the value of the TTL field in any label stack entry which is not at the top of the stack'.

The 'outgoing TTL' of a CCSDS labeled packet is defined to be one less than the 'incoming TTL'.

C3.6.2 Protocol-independent TTL Processing

The Protocol-independent Time to Live processing for CCSDS labeled packets is the same as is stated in RFC 3032 for other labeled packets with the exception that there is only one TTL Field in the proposed encoding scheme and, therefore, all references to the ‘TTL field of the label stack entry at the top of the label stack’ should be considered a reference to the singular TTL of the CCSDS labeled packet.

C3.6.3 CCSDS-dependent Rules

There is no concept of ‘Time to Live’ in CCSDS Source or Encapsulation Packets. The TTL field of the labeled packet, therefore, must be set to the maximum number of hops (LSRs) that a labeled packet is permitted to experience before being discarded.

C3.6.4 SCPS-dependent Rule

When a SCPS NP packet is first labeled, the TTL field of the CCSDS labeled packet MUST BE set to the value of the SCPS NP Datagram Hop Count field if it exists. If the SCPS NP datagram Hop Count field needs to be decremented, as part of the SCPS NP processing, it is assumed that this has already been done. If the SCPS NP datagram Hop Count field does not exist, the TTL field of the CCSDS labeled packet field MUST BE set to the maximum number of hops (LSRs) that a CCSDS labeled packet is permitted to experience before being discarded.

When a label is popped, and the resulting label stack is empty, then the value of the SCPS NP datagram Hop Count field should be replaced with the outgoing TTL value, as defined above.

C3.6.5 IP-dependent Rules

The IP-dependent rules are the same for CCSDS labeled packets as is stated in RFC 3032 for other labeled packets with the exception that there is only one TTL Field in the proposed encoding scheme and, therefore, all references to the ‘TTL field of the label stack entry’ should be considered a reference to the singular TTL of the CCSDS labeled packet.

C3.7 FRAGMENTATION

The fragmentation considerations expressed in RFC 3032 do not apply to CCSDS links because the frame encoding provides for transporting data units of unlimited size.

ANNEX D

CCSDS TRANSFER FRAMES

D1 TRANSFER FRAME VERSIONS

Currently there are two adopted and one proposed version of the CCSDS Transfer Frame. Version 1 was defined originally and was recommended for ‘conventional missions, as characterized by bi-directional (but asymmetric) transfer of data between moderately complex space and ground systems, serving a moderate number of users, at low-to-medium data rates’. Version 2 was later defined ‘to accommodate extra services needed by Advanced Orbiting Systems’ such as ‘manned and man-tended space stations’. Version 3 has most recently been proposed for the ‘efficient transfer of space data of various types and characteristics over proximity space links’.

For each version there are several optional features and variations that affect the throughput of the Virtual Channel to which it is assigned. There are also options in the error detection and forward error correction coding that affect the frame overhead. The following sections lightly discuss each version of the CCSDS Transfer Frame and the overhead presented by various options. For a complete discussion of each version of the Transfer Frame, refer to their respective specifications in the CCSDS Recommendation for ‘Packet Telemetry’ [23], ‘Telecommand’ [24] and [25], ‘AOS Networks and Data Links’ [26], and ‘Proximity-1 Space Link Protocol’ [21].

D2 CONVENTIONAL TRANSFER FRAMES

NOTE – The asymmetric nature of Version 1 conventional transfer is apparent in its use of different frame formats, one for Packet Telemetry and one for Telecommand.

D2.1 CONVENTIONAL PACKET TELEMETRY TRANSFER FRAMES

Figure D-1 illustrates the format of the Packet Telemetry Transfer Frame along with a prepended Attached Synchronization Marker (ASM) and a set of Reed-Solomon (R-S) check symbols. This structure constitutes the complete bit requirement for transmitting a Version 1 Packet Telemetry Transfer Frame.

The ASM for a Version 1 Packet Telemetry Transfer Frame is a fixed 32 bits in length. The Transfer Frame itself is fixed in length for each implementation, but may vary in size up to a maximum length of 8,920 bits not including ASM or R-S Coding. All Version 1 Packet Telemetry Transfer Frames have a 48-bit primary header field that reduces the maximum data that can be carried by each data unit to 8,872 bits. If needed for a specific mission, a secondary header may be required that can be as much as 512 bits in length. Space for a 32-bit Operational Control Field and a 16-bit Frame Error Control Field may also be included.

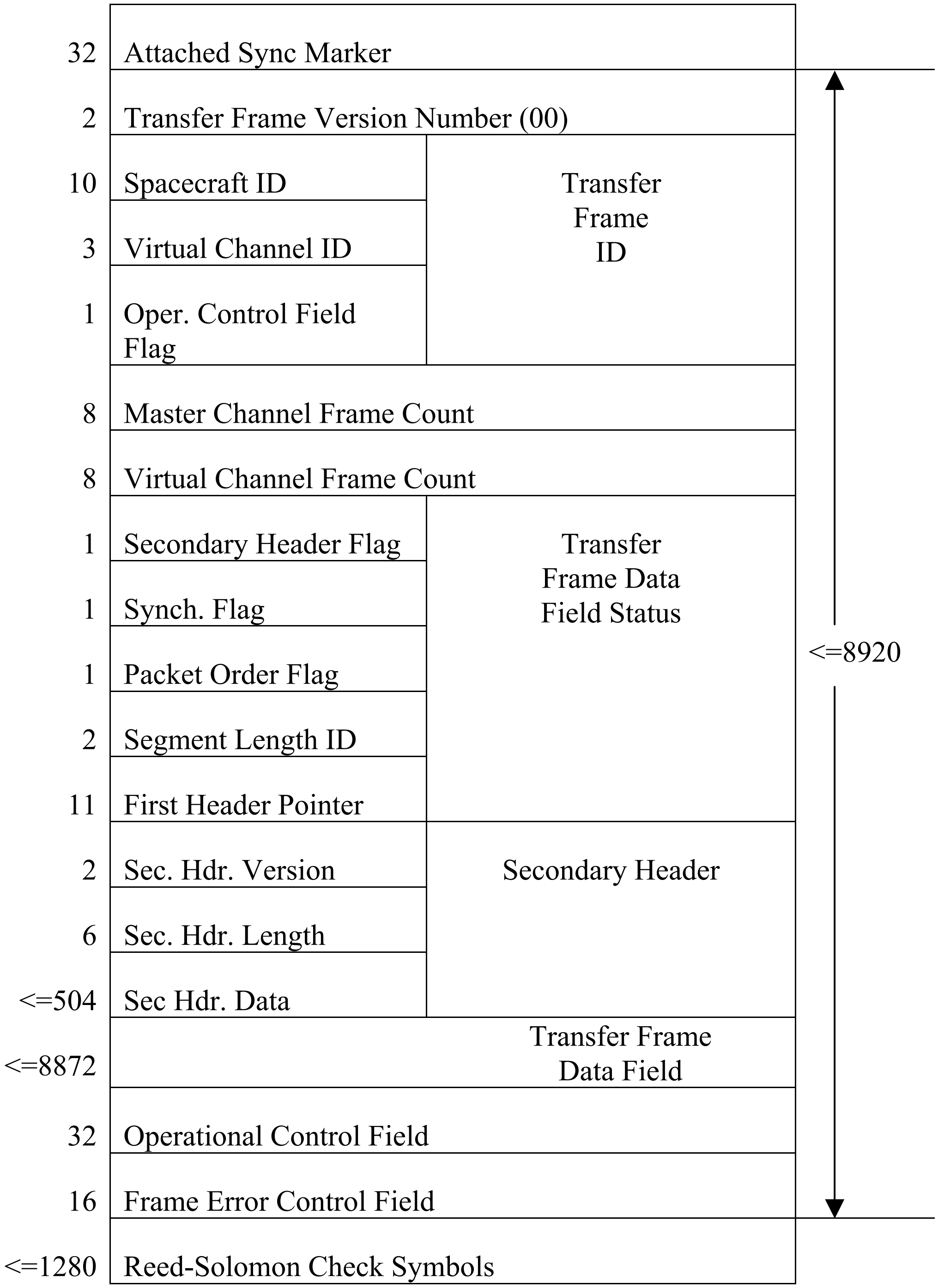
Although figure D-1 includes 1280 bits of R-S code symbols, the Frame Error Control Field coding option may apply. In some implementations the R-S coding may not be performed, and the Frame Error Control Field will be used to detect errors. Either the Frame Error Control Field, the R-S code symbols, or both must be used.

The managed bandwidth for a Version 1 Packet Telemetry Virtual Channel is computed as follows:

$$b = (v / (a + t + r)) * (t - (h + s + o + f))$$

Where:

b = Managed Bandwidth
v = Bandwidth Allocation of the Virtual Channel
a = ASM Length (32)
t = Transfer Frame Length
r = Reed-Solomon Code Length = $(\text{int}((t + 222)/223) * 32)$
h = Primary Header Length (48)
s = Secondary Header Length (if present)
o = OCF Length (if present)
f = Frame Error Control Field Length (if present)



32	Attached Sync Marker		
2	Transfer Frame Version Number (00)		<=8920
10	Spacecraft ID	Transfer Frame ID	
3	Virtual Channel ID		
1	Oper. Control Field Flag		
8	Master Channel Frame Count		
8	Virtual Channel Frame Count		
1	Secondary Header Flag	Transfer Frame Data Field Status	
1	Synch. Flag		
1	Packet Order Flag		
2	Segment Length ID		
11	First Header Pointer		
2	Sec. Hdr. Version	Secondary Header	
6	Sec. Hdr. Length		
<=504	Sec Hdr. Data		
<=8872		Transfer Frame Data Field	
32	Operational Control Field		
16	Frame Error Control Field		
<=1280	Reed-Solomon Check Symbols		

Figure D-1: Version 1 CCSDS Conventional Packet Telemetry Transfer Frame with ASM and R-S Coding

D2.2 CONVENTIONAL TELECOMMAND TRANSFER FRAMES

Figure D-2 illustrates the format of the Version 1 Telecommand Transfer Frame. This structure does not constitute the complete bit requirement for transmitting a Version 1 Telecommand Transfer Frame. The CCSDS Conventional Telecommand Frame is coded using a modified Bose-Chaudhuri-Hocquenghem (BCH) code that produces a 64-bit code block for every 56 bits, or portion of 56 bits, in the transfer frame. In addition, a 16-bit Start Sequence precedes each coded frame, and a 64-bit Tail Sequence follows each coded frame. This structure constitutes the complete bit requirement for transmitting a Version 1 Conventional Telecommand Transfer Frame.

The Telecommand Transfer Frame itself may vary in size up to a maximum length of 8,192 bits. This maximum is not affected by the presence or absence of the 16 bit Frame Error Control Word optionally used with these frames. All Version 1 Conventional Telecommand Transfer Frames have a 40-bit header field.

The Telecommand Transfer Frame has a Frame Data Field sized to hold one TC Frame Data Unit, the data unit passed from the immediately higher Segmentation Layer. This Frame Data Field is limited to a maximum of 8,152 bits if no Frame Error Control Word is used, and 8,136 bits if a Frame Error Control Word is used. The maximum managed bandwidth for a Telecommand Virtual Channel is not a constant as in a fixed length frame implementation. The amount of frame overhead is a function of the size of the TC Frame Data Units. The amount of bandwidth consumed by each frame can be expressed as:

$b = s + h + c + t$

Where:

b = Bandwidth Consumed by a Frame
s = Start Sequence Length ($>=16$)
h = Header Length (40)
d = TC Frame Data Unit Length
c = Code Block Area Length ($\mathrm{int}((d + 55)/56) * 64$)
t = Tail Sequence Length (64)

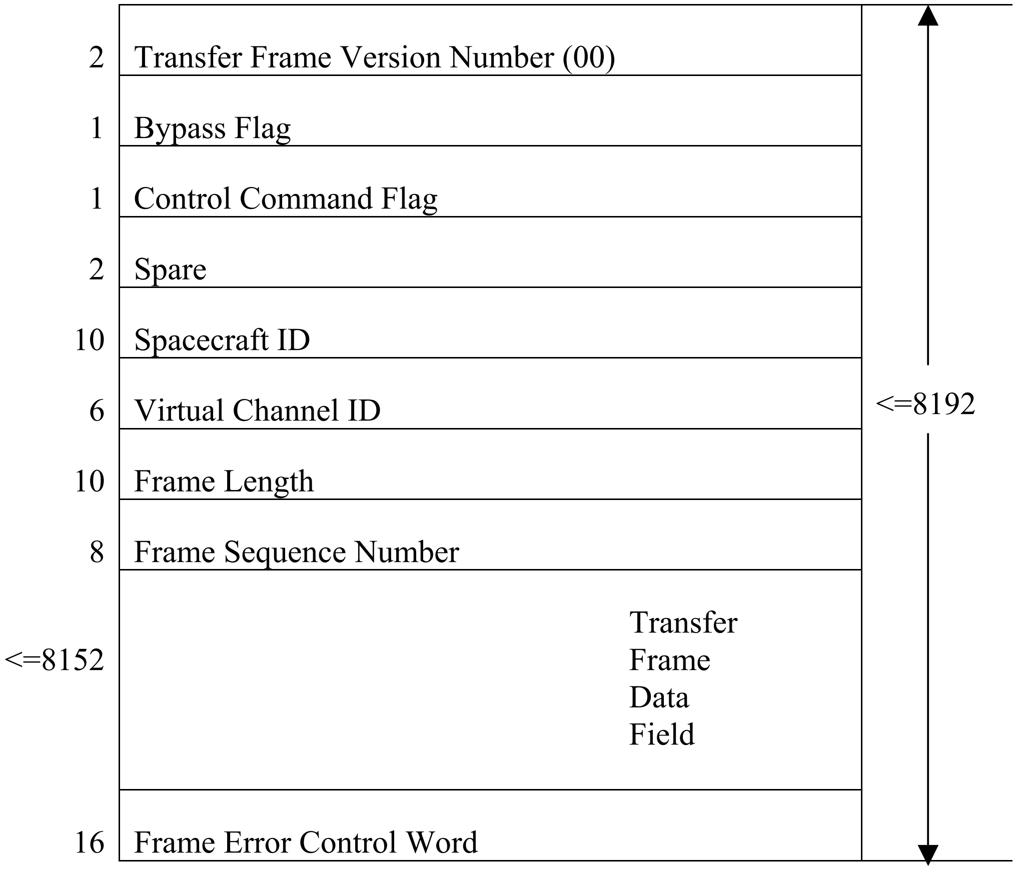


Figure D-2: Version 1 CCSDS Conventional Telecommand Transfer Frame

D3 VERSION 2: ADVANCED ORBITING SYSTEMS (AOS) TRANSFER FRAME

Figure D-3 illustrates the format of the AOS Transfer Frame along with a prepended Attached Synchronization Marker (ASM) and a set of R-S check symbols. This structure constitutes the complete bit requirement for transmitting a Version 2 Transfer Frame.

The ASM for a Version 2 Transfer Frame is a fixed 32 bits in length. The AOS Transfer Frame itself is fixed in length for each implementation, but may vary in size up to a maximum length of 10,200 bits if there are no R-S check symbols. If there are attached R-S check symbols, the maximum length of the AOS Transfer Frame is 8,920 bits. Similar to Version 1 Telemetry Frames, all Version 2 Transfer Frames have a 48-bit primary header field. Optionally, the primary header field may be extended by a 16-bit error checking and correction code. In order to insert isochronous data into every transmitted frame, a fixed length (determined on an implementation by implementation basis) insert zone may be reserved in each transmitted PDU. Space for a 48-bit AOS Transfer Frame trailer consisting of a 32-bit Operational Control Field and a 16-bit Frame Error Control Field may also be included.

Although figure D-3 includes 1,280 bits of R-S code symbols, no such symbols are present for Grade-3 service.

The managed bandwidth for a Version 2 Virtual Channel is, therefore, computed as follows:

$$b = (v / (a + t + r)) * (t - (h + i + o + f))$$

Where:

b = Managed Bandwidth
v = Bandwidth Allocation of the Virtual Channel
a = ASM Length (32)
t = Transfer Frame Length
r = Reed-Solomon Code Length = (int ((t + 222)/223) * 32)
h = Primary Header Length (48)
i = Insert Zone Length (if present)
o = OCF Length (if present)
f = Frame Error Control Field Length (if present)

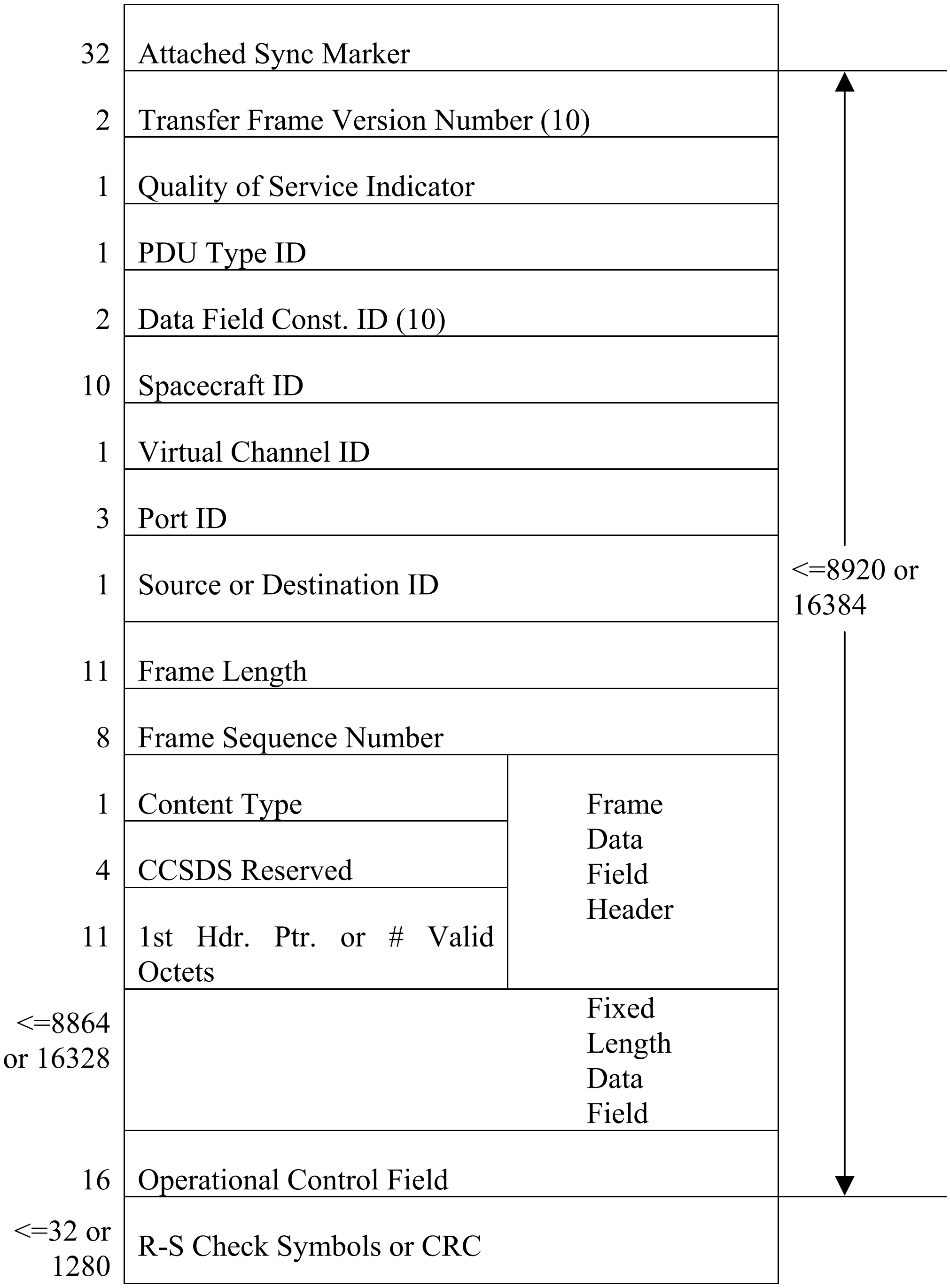
32	Attached Sync Marker		
2	Transfer Frame Version Number (01)		<=8920
8	Spacecraft ID	VCDU ID	
6	Virtual Channel ID		
24	Virtual Channel Data Unit Counter		
1	Replay Flag	Signaling Field	
17	Spare		
16	VCDU Header Error Control		
Fixed	VCDU Insert Zone		
<=8872		Fixed Length VCDU Data Unit Zone	
32	Operational Control Field		
16	Frame Error Control Field		
<=1280	Reed-Solomon Check Symbols		

Figure D-3: Version 2 CCSDS Transfer Frame with ASM and R-S Coding

D4 VERSION 3: PROXIMITY-1 TRANSFER FRAMES

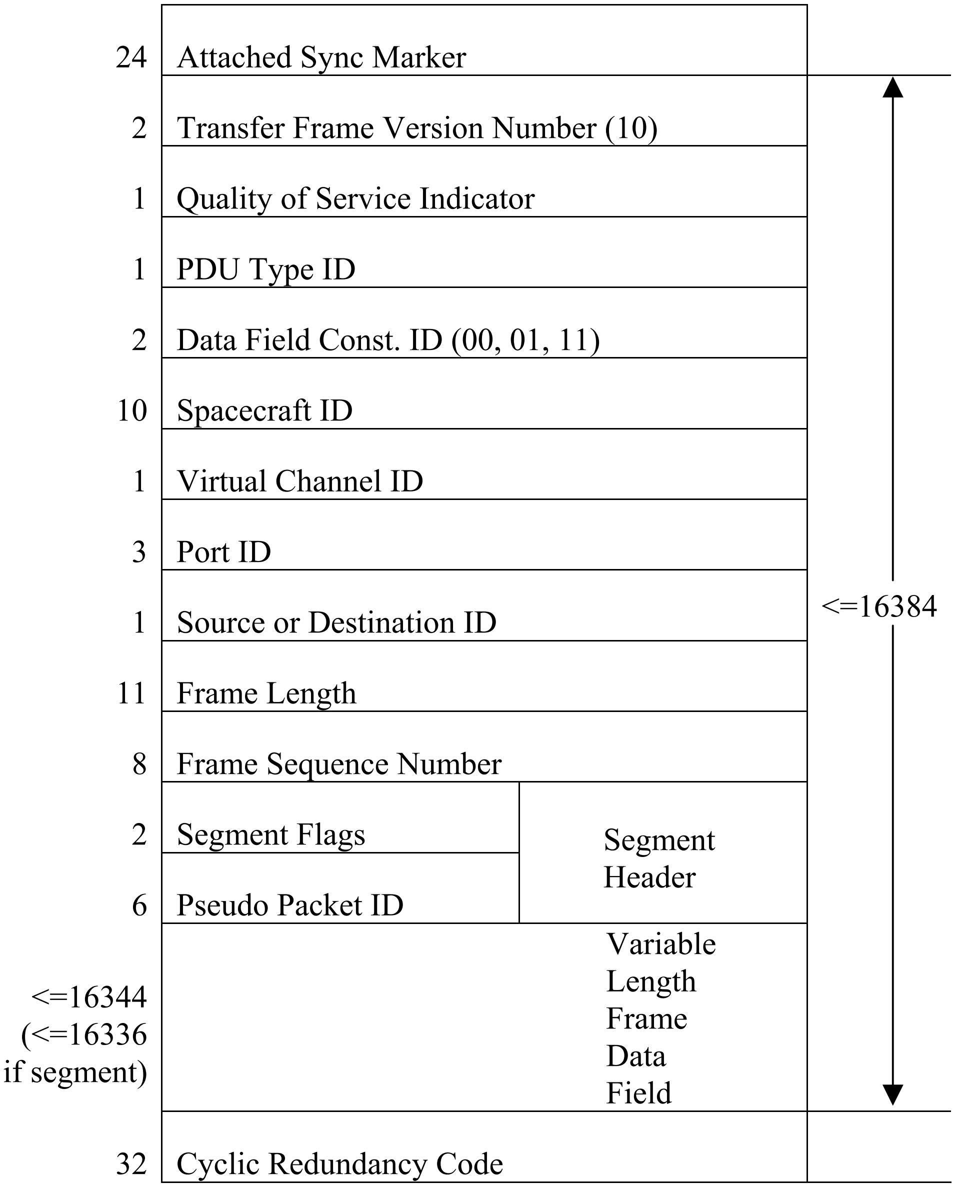
D4.1 GENERAL

There are two basic types of Version 3 Transfer Frames. One is fixed length and used in synchronous implementations, and one is variable length and used for asynchronous implementations. Figure D-4 illustrates the format of the Proximity-1 fixed length Transfer Frame and figure D-5 illustrates the Proximity-1 variable length Transfer Frame. Both figures include a prepended ASM and either a set of R-S check symbols for fixed length data units, or a Cyclic Redundancy Code (CRC) for either fixed or variable length Transfer Frames. These structures constitute the complete bit requirements for transmitting Version 3 Transfer Frames.



32	Attached Sync Marker		
2	Transfer Frame Version Number (10)		<=8920 or 16384
1	Quality of Service Indicator		
1	PDU Type ID		
2	Data Field Const. ID (10)		
10	Spacecraft ID		
1	Virtual Channel ID		
3	Port ID		
1	Source or Destination ID		
11	Frame Length		
8	Frame Sequence Number		
1	Content Type	Frame Data Field Header	
4	CCSDS Reserved		
11	1st Hdr. Ptr. or # Valid Octets		
<=8864 or 16328		Fixed Length Data Field	
16	Operational Control Field		
<=32 or 1280	R-S Check Symbols or CRC		

Figure D-4: Version 3 Fixed Length CCSDS Proximity Link Transmission Unit



24	Attached Sync Marker		
2	Transfer Frame Version Number (10)		<=16384
1	Quality of Service Indicator		
1	PDU Type ID		
2	Data Field Const. ID (00, 01, 11)		
10	Spacecraft ID		
1	Virtual Channel ID		
3	Port ID		
1	Source or Destination ID		
11	Frame Length		
8	Frame Sequence Number		
2	Segment Flags	Segment Header	
6	Pseudo Packet ID		
<=16344 (<=16336 if segment)		Variable Length Frame Data Field	
32	Cyclic Redundancy Code		

Figure D-5: Version 3 Variable Length CCSDS Proximity Link Transmission Unit

D4.2 SYNCHRONOUS IMPLEMENTATIONS

The ASM for a fixed length Version 3 Transfer Frame is a fixed 32 bits in length. The Transfer Frame itself is fixed in length for each implementation but may vary in size up to a maximum length of 8,920 bits if R-S forward error correction coding is used and 16,384 bits if CRC is used. This maximum is not reduced by the presence of the R-S check symbols or CRCs. The fixed length Version 3 Transfer Frames contain a 40-bit Header Field and a 16-bit Frame Data Field Header. The data capacity of the Transfer Frame may also be reduced by the presence of an Operational Control Field.

For the purposes of this analysis, we are assuming that Network Layer Packets may be carried in these fixed length data units with a Frame Data Field Header that will contain a

First Header Pointer. This allows for packets to be split over multiple frames. We also assume that all packets destined for an associated Virtual Channel will be destined to the same Port so that they can be multiplexed together in Transfer Frames. For Internet-like traffic it is appropriate to leave Automatic Request Queuing (ARQ)-type actions to the Transport Layer, so we assume that all Network Layer data units are submitted for Expedited Service. The managed bandwidth for a synchronous Version 3 Virtual Channel is, therefore, computed as follows:

$$b = (v / (a + t + c)) * (t - (h + p + o))$$

Where:

b = Managed Bandwidth
v = Bandwidth Allocation of the Virtual Channel
a = ASM Length (32)
t = Transfer Frame Length
c = R or C
R = Reed-Solomon Code Length = (int((t + 222)/223) * 32)
C = Cyclic Redundancy Code Length = 32
h = Header Length (40)
p = Frame Data Field Header Length (16)
o = OCF Length (if present)

D4.3 ASYNCHRONOUS IMPLEMENTATIONS

The ASM for the variable length Version 3 Transfer Frames used in asynchronous implementations is a fixed 24 bits in length. The Transfer Frame itself may vary in size up to a maximum length of 16,384 bits. This maximum is not affected by the presence of the 32-bit CRC used with these frames. All Version 3 Transfer Frames have a 40-bit header field. The variable length Version 3 Transfer Frames may also contain an 8-bit segment header.

The maximum managed bandwidth for an asynchronous Version 3 Virtual Channel is not a constant as it is in a synchronous implementation. The amount of frame overhead is a function of the size of the Network Layer data units. The amount of bandwidth consumed by each frame can be expressed as:

$$b = a + h + s + t + c$$

Where:

b = Bandwidth Consumed by a Frame
a = ASM Length (24)
h = Header Length (40)
s = Segment Header Length (8)
t = Segment/N-PDU Length
c = Cyclic Redundancy Code Length (32)