

User's Manual for the SCPS Network Protocol

April 1997

Robert C. Durst
Mary Jo Zukoski

Sponsor: SMC/AXE
Dept. No.: W159

Contract No.: F-19628-94-C-0001
Project No.: 03970638

Washington C³ Center
McLean, Virginia

Abstract

This document provides the information necessary to write programs that use the Space Communications Protocol Standards (SCPS) Network Protocol (SCPS-NP). The primary content of this document are example routing tables and requirements structures that are used in the operation of SCPS-NP.

This document does NOT provide information necessary to build a distribution of the SCPS-NP software. Such information is provided as documentation with the reference implementation of the SCPS protocols or with whatever distribution of the software the user might have.

KEYWORDS: SCPS, SCPS-NP, SCPS-TP, Routing, Precedence, Multicast

Table of Contents

Section	Page
Introduction	1
Overview of SCPS Network Protocol Capabilities	3
2.1 Background	3
2.2 Terminology	3
2.3 Requirements	4
2.4 Shortcomings of using IP	6
2.5 Capabilities of the SCPS Network Protocol	8
2.6 Invoking the Capabilities of the SCPS Network Protocol	11
2.6.1 Configuring Routing Tables	11
2.6.2 Configuring Address Translation Tables	12
2.6.3 Completing a SCPS-NP Requirements Structure	12
2.6.4 Configuring Network Control Requirements	12
2.6.5 Setting MIB Configuration Parameters and Thresholds	13
2.6.6 Sending a Datagram Directly From the Requirements Structure	13
2.6.7 Acquiring a SCPS-NP Template	13
2.6.8 Sending a Datagram Using the SCPS-NP Template	13
2.7 Configuration Alternatives	14
2.7.1 Encapsulation	14
2.7.2 Translation	15
Configuring SCPS-NP to Enable Specific Capabilities	17
3.1 Checksum	17
3.2 Timestamps	17
3.3 Precedence	18
Format Requirements for Input Files	19
4.1 Translation File:	19
4.1.1 npIP_NP_File	19
4.1.2 npPathFile	20
4.2 Routing File:	21
4.2.1 npNextHopFile	21
4.2.2 npMultiNextHopFile	22

Section	Page
Suggested Reading	25
Glossary	27

List of Figures

Figure		Page
1	Encapsulating Network Configuration	14
2	Translating Network Configuration	15
3	Format for setsockopt call: checksum	17
4	Format for setsockopt call: timestamps	17
5	Format for setsockopt call: precedence	18
6	Example npIP_NP_File	20
7	Example npPathFile	20
8	Example npNextHopFile for End System A	21
9	Example npNextHopFile for End System B	22
10	Example npNextHopFile for End System C	22
11	Example npMultiNextHopFile for End System A	23
12	Example npMultiNextHopFile for End System B	23
13	Example npMultiNextHopFile for End Systems C & D	23
14	Example npMultiNextHopFile for End System C	24
15	Example npMultiNextHopFile for End System F	24

Section 1

Introduction

In the fall of 1992, NASA and the DOD jointly established a technical team (the SCPS Technical Working Group, or "SCPS-TWG") to explore possibilities for developing common space data communications standards, with a principal focus on the activities associated with in-flight monitoring and control of civil and military spacecraft. In practical terms, these activities involve a ground control center conducting a dialog with a remote spacecraft to transmit telecommands, to up-load and verify onboard software loads, and to confirm correct spacecraft performance via a flow of telemetry.

The team adopted a two-pronged approach in its study phase: part of the team conducted a top-down survey of representative civil and military space data communications requirements, while the remainder of the team conducted a bottom-up analysis of available standard data communications protocols. Together they compared the results to see how capabilities matched requirements, and formulated recommendations for future work. In evaluating existing capabilities, first priority was given to commercially-supported "off the shelf" standards. However, recognizing unique requirements of the space mission environment (long propagation delays, noise-induced errors, and limited spacecraft data processing resources and communications capacity), the team also considered other options. By the end of 1993 the team concluded that wide segments of the US civil and military space communities have common needs for:

- An efficient file handling protocol, capable of supporting file transfers initiated either from ground-based systems or space-based systems
- A data transport protocol that provides the user with selectable levels of reliability, based on operational need, between computers that are communicating over a network containing one or more space data transmission paths
- Optional data protection mechanisms to assure the end-to-end security and integrity of such message exchange
- An efficient protocol to support connectionless routing of messages through networks containing space data links.

Following the study phase, the SCPS-TWG began development of four specifications, one for each of the protocols, that address the above requirements: the SCPS File Protocol (SCPS-FP), the SCPS Transport Protocol (SCPS-TP), the SCPS Security Protocol (SCPS-SP), and the SCPS Network Protocol (SCPS-NP). These draft specifications have been submitted for adoption as military standards and as international standards. At the completion of these standards activities, resulting standards may then be adopted by any military, civil, or commercial organization for use in any space system. It is the intent of NASA and DOD that commercial vendors produce the SCPS protocols as widely-distributed commercial products, thus helping to reduce the cost of space systems while increasing their interoperability.

Part of the protocol development includes development of a reference implementation of each of the protocols. This reference implementation is being made available for the purposes of evaluation and experimentation with the protocols by potential users of the SCPS capabilities. As the SCPS-NP is not used directly by the user (it is called by the SCPS-TP and underlying link layer protocols), the predominant interface the user will have with the SCPS-NP are SCPS-TP requirements structures, routing tables and translation tables.

This user's guide continues in Section 2 with an overview of the major SCPS-NP capabilities. Section 3 illustrates the format of the requirements structure, translation tables and routing tables. Finally, Section 4 presents suggested reading for those readers not familiar with IP or routing in general.

Section 2

Overview of SCPS Network Protocol Capabilities

2.1 Background

The SCPS Network Protocol operates at Layer 3 of the OSI Basic Reference Model. Its primary goal is to route data from the source of the data to its ultimate destination with the user's requested quality of service.

The SCPS-NP is a new protocol. It is not a subset of the Internet Protocol (IP), although it draws on concepts and technology from IP, and shares some IP numbering in its service interface. The SCPS-NP also draws on concepts and technology from the CCSDS Path service, but is neither a subset nor a superset of CCSDS Path.

The drivers for generating a new protocol were two: to provide very high bit-efficiency (primarily compared to IP), and to provide for user-required qualities of service not provided by either IP or the existing CCSDS capabilities.

As a result of the study phase mentioned in Section 1, several technical requirements were allocated to the network layer of the OSI Basic Reference Model. Section 2.2 presents some terminology useful to the subsequent text, then Section 2.3 presents and discusses the requirements that were allocated to the network layer. Section 2.4 examines how the Internet Protocol (IP) might be used to meet those requirements and identifies some of the shortcomings of such an approach. Section 2.5 describes the capabilities within the SCPS-NP that address the requirements. Section 2.6 discusses configuring a network to support the required SCPS-NP capabilities and the means by which applications request those capabilities.

2.2 Terminology

In this subsection, we present terms that are appropriate to the subsequent discussion and are either not widely known or are prone to misinterpretation.

A connection, in communication terms, is a term that describes state information that is named, persistent, and shared across the systems supporting the communication. Data sent via that connection make use of the shared state, thus gaining bit-efficiency and possibly processing advantages. One implication of the use of a connection is that all data flowing on the connection is treated in the same manner, as specified by the state information that defines

the connection. Rather than carrying the state information itself, the data is accompanied by an identifier that is used to reference the state information. This state information is typically the source and destination, but may include information such as precedence. A “managed connection” is one in which the shared state is distributed via network management mechanisms (outside the scope of the SCPS-NP).

Some refer to a “datagram” as a unit of data transmission for connectionless networks and a “packet” as a unit of transmission for connection-oriented networks. We make no such distinction here - the terms are used interchangeably to represent a variable-length, octet-aligned protocol data unit. Neither the term “packet” nor the term “datagram” imply any particular layer of the OSI reference model when used in this document.

The term flood routing describes a routing technique in which a packet is replicated and transmitted to all adjacent nodes. The adjacent nodes then replicate the packet and repeat the process. Packet identification techniques are used to prevent a node replicating a packet more than once. This technique is typically applied in networks with rich connectivity (meaning that each node is connected to several other nodes), and is used to improve the probability of receipt of important packets. Flood routing generates a substantial amount of traffic, so it is typically used sparingly. However, its use improves the probability that all nodes in the network will receive at least one copy of the packet (the packet identification techniques ensure that *only* one copy of a particular packet will be delivered to upper layer protocols). Appropriate uses are for very high priority traffic and for routing updates that should be delivered to all systems in the network.

2.3 Requirements

This section summarizes the technical requirements that have been allocated to the network layer, and provides some discussion of how those requirements relate to SCPS communications environments. Prospective users and network designers should consider which of these requirements apply to their operational environments, in order to make appropriate configuration decisions based on information presented later in this section.

The key technical requirements that were allocated to the network layer are summarized below. The SCPS Network protocol must:

- Route data from source to destination
- Provide efficient operation in constrained-bandwidth environments
- Provide precedence- (priority-) based data handling
- Provide packet lifetime control
- Provide selectable routing treatments

- Provide signaling of network conditions to upper layer protocols

The ability to route data from source to destination is characteristic of essentially all protocols that operate at the network layer of the OSI Basic Reference Model. The SCPS Network Protocol was based on requirements derived from several types of networks ranging from simple to complex. The simplest networks involve a single link and dedicated end systems at either end of that link, such as is found with typical current military SATCOM configurations. Other networks involve a single destination end system (satellite onboard computer) communicating through one or more ground stations over a ground network to an operations center consisting of several end systems. Some spacecraft may have onboard networks. The more complex networks that served as sources of requirements involve networks with changing topologies, such as those found in satellite constellations or in military tactical radio networks. In these topologies, end systems may communicate with other mobile end systems or may communicate through the mobile network to the ground-based (wired) network.

Common to all of the prospective environments is that bandwidth may be constrained, either unidirectionally or bidirectionally. This constraint results in a requirement to operate with high bit-efficiency. Bit-efficiency quantifies the fraction of transmitted bits that are user data. Improving bit-efficiency may be accomplished in two ways: by increasing the amount of user data per unit of protocol control information (i.e., header information), or by decreasing the amount of protocol control information per unit of user data. The first approach, making packets longer, is simple, but does not work well in environments that are prone to bit-errors. It also does not work well when the user's data does not lend itself to aggregation. The second approach, reducing protocol header overhead, is the approach used throughout the SCPS Network Protocol design. Several requirements derive from the need to operate in bandwidth-constrained environments: multicasting, support for managed-connections, and precedence-based data handling all address bandwidth-related constraints and are described in subsequent paragraphs.

Multicasting is a technique for improving bit-efficiency. The technique of multicasting allows addressing of data to a group of destination systems. Rather than sending a unique copy of the data to each remote system, data are sent to the group address, and intermediate systems replicate the multicast packet only as necessary in order to reach all of the destination systems in the multicast group.

Managed-connections can enhance bit-efficiency in networks that can be characterized as having a few source-destination pairs that account for most of the network's traffic. For these flows, the source and destination addresses can be replaced by an identifier for the managed connection. In the SCPS-NP, this identifier is called a "Path address."

Precedence improves operation in bandwidth-constrained environments in two ways. First, it controls the order of service, which reduces queuing delay and variation in queuing delay for high precedence traffic. Second, precedence controls the order of packet discarding when congestion occurs, to ensure that if packet discarding is required, low precedence packets are discarded before higher precedence packets.

Packet lifetime control provides protection against transient routing loops. A transient routing loop is formed when routing tables in the network are not synchronized. This condition can occur as a result of using certain routing protocols, such as Shortest Path First [Bertsekas, pp. 410-414]. While a routing loop is in existence, the links forming the loop may become progressively more congested. Packet lifetime control ensures that data packets do not remain in the network indefinitely, as they are discarded once they have exceeded their “lifetime.” This, combined with either automated or manual means to update the routing tables provides control over routing loops.

The requirement for selectable routing treatments provides the ability to switch between “normal” routing and other routing treatments, such as flood routing. This is of use in the more complex network topologies that involve relatively rich connectivity, such as satellite constellations with cross links or some mobile tactical radio networks. The ability to flood route packets in these networks can improve the probability of receipt and reduce the propagation time of the flood routed packet through the network.

Signaling of network conditions to upper-layer protocols is required to allow those protocols to become aware of and to adapt to changing conditions within the network. Signals that may be passed to the upper-layer protocols include indications of network congestion, network corruption, and link outages. This requires the network to identify these conditions at points in the network that may be remote from the end systems that host the upper-layer protocols, and to propagate network-internal signals to the affected end systems for delivery to the upper-layer protocols.

2.4 Shortcomings of using IP

The Internet Protocol (IP) is a highly capable, broadly distributed protocol. It is an appropriate protocol for many environments, and may be appropriate for some SCPS environments. Due to its broad commercial support, if it will meet the requirements of a mission, it should be seriously considered.

Table 1, below, lists the requirements presented in Section 2.3 and identifies whether IP meets those requirements. Clearly, IP can route data from its source to its destination, although as

with the SCPS Network Protocol, the choice of specific routing protocol is dependent on the local networking environment.

Table 1. Support of SCPS Network Requirements by IP

Requirement	Support in IP?
Route from source to destination	Yes
Support for constrained bandwidth	No
Multicasting	Yes
Managed-connection operation	No
Precedence- (priority-)based handling	Yes*
Selectable routing treatment	No
Packet lifetime control	Yes
Signaling to support upper-layer processing and network control	Partial

In general, IP does not provide any explicit support for operating in constrained-bandwidth environments. IP headers are a minimum of 20 octets in length, and may be made longer with the addition of options. IP provides support for multicasting, but has no mechanism for shortening its headers by using managed connections. (Note that there are techniques for compressing TCP and IP headers at the link layer, however these techniques were designed for use on low-speed serial links without significant propagation delays. The technique is defined in RFC 1144, and a discussion of its use in SCPS environments appears in Section 2 of the SCPS-TP User's Guide.)

The IP header contains a field to carry eight levels of precedence. However, commercial equipment typically does not make any use of the field, hence the asterisk (*) in the table. In particular, high precedence packets would not benefit from any reduced probability of discard in congested routers, nor would they receive any reduced queuing delays in routers.

There is no concept of flood routing in IP. While an IP option could be defined to signal flood routing, there is no routing support for it in commercially-available implementations.

The capabilities in IP for packet lifetime control are adequate for most environments.

With respect to signaling of network conditions, some IP implementations provide partial support. The Internet Control Message Protocol (ICMP), the companion protocol to IP that handles such signaling, has the ability to generate congestion signals. However, the use of this signal has been deprecated due to the inability of routers to control the rate at which the

congestion signals are generated (this problem has been solved with the advent of Random Early Detection (RED), but RED is not widely deployed nor is its Explicit Congestion Notification (ECN) option). There is no signaling provided to indicate loss due to corruption nor to link outage.

2.5 Capabilities of the SCPS Network Protocol

The SCPS Network Protocol provides support for all of the requirements identified in Section 2.3. The protocol is designed in such a way that unnecessary header elements are not incorporated into the header. This design decision increases the processing to format and parse SCPS-NP headers in favor of reducing the number of bits that are transmitted.

Table 2 reprises the requirements presented in Section 2.3 and identifies the support for those requirements within the SCPS-NP. The following paragraphs describe the capabilities of the SCPS-NP and how they meet the requirements.

Table 2. Support of SCPS Network Requirements by SCPS-NP

Requirement	Support in SCPS-NP
Route from source to destination	Yes
Support for constrained bandwidth	Short, variable length headers
Multicasting	Yes
Managed-connection operation	Path addressing
Precedence- (priority-)based handling	Yes
Selectable routing treatment	“Normal” and Flood Routing
Packet lifetime control	Hop Count and Timestamp
Signaling to support upper-layer processing and network control	Separate signals for congestion, corruption, and link outage

Network protocols typically route data from source to destination by selecting the “next hop” router based on the destination address. There are many methods by which the next hop router is selected. The SCPS-NP selects its next hop router by means of routing tables, which may be statically or dynamically configured. Routing tables that are statically configured are typically maintained either with network management or by distributing the tables in files. Some network configurations benefit from the use of routing protocols to maintain the routing tables. The routing protocols are not part of the SCPS-NP, but interact with the SCPS-NP routing tables. This release of the SCPS Network Protocol supports statically-configured routing tables. However, there is nothing that prevents the addition of dynamic routing protocols or prevents them from interacting with the SCPS-NP routing tables.

The SCPS-NP is designed for constrained-bandwidth operation. The protocol has only a few elements that are present in every packet header: a version number, the packet length, the transport protocol identifier, and a control field. The transport protocol identifier indicates the network user (e.g., the SCPS-TP, TCP, UDP, or Compressed TCP, or the SCPS-SP) to which the packet’s data should be delivered. The control field is a variable-length bit-field that signals what protocol header elements appear in the remainder of the header. These optionally-appearing header elements include both source and destination addresses, fields for precedence and routing requirements, fields to support packet lifetime control, and a header checksum for header error detection.

The SCPS-NP may perform address translation to improve bit-efficiency. Typically, the protocols that use SCPS-NP operate using IP addresses. This is especially important to consider if the SCPS-NP protocol operates only in the space (or wireless) segment of the

network and protocol translation rather than encapsulation is performed. Section 2.7 discusses this topic further. However, IP addresses are four octets in length, and there are two of them. These can be carried without translation in the SCPS-NP header, or may be translated into more bit-efficient representations. The address formats are described in the SCPS-NP specification. An IP source-destination pair may be translated into three alternate versions: an Extended Path Address, which represents the two addresses with a single four-octet address; a pair of Basic End System Addresses, which (in the Red-1 version of the specification) represents each of the two four-octet addresses with a two-octet address; or a Basic Path Address, which represents the pair of four-octet addresses with a single two-octet address. The use of a single address to represent an address pair is what is meant by a ‘managed connection.’ The SCPS-NP decides whether to translate addresses based on address translation tables that are configured statically. These translation tables are identical throughout the network. Address translation table formats are discussed in Section 2.6.

The SCPS-NP supports multicasting, and identifies multicast (group) addresses based on the address format. Either end-system addresses or path addresses may signify multicast address groups. (Refer to the SCPS-NP specification for the details of multicast address formats.) Currently in the SCPS-NP, the systems that belong to multicast groups are defined statically. Multicast group configuration is discussed in Section 2.6.

The SCPS-NP carries precedence in its Basic Quality of Service header field. When packets are queued for transmission within the SCPS-NP, the precedence field controls the order of transmission (higher-precedence packets are transmitted first), and the order of packet discard in the event of congestion (lower-precedence packets are discarded first).

The SCPS-NP supports selection of routing treatments as another element within the Basic Quality of Service header field. Signaling of four separate routing treatments is supported by the protocol, with two of those being defined in the current version. The two routing treatments that are defined are ‘normal’ and flood routing, and are described in Section 2.3, above.

The SCPS-NP supports two forms of packet lifetime control. The first uses a hop-count, which is initialized at the packet’s source and decremented every time the packet is routed. If it reaches zero before the packet reaches its destination, the packet is discarded. (This prevents packets that are caught in routing loops from remaining in the network indefinitely.) All networks that have the possibility of routing loops can make use of the hop count capability. The second form of packet lifetime control is based on a source timestamp, which is carried in the SCPS-NP header and indicates the time at which the packet was sent. A system that receives the SCPS-NP packet checks the timestamp to determine whether the packet is too old to be forwarded. This decision is made based on a Maximum Packet

Lifetime configuration parameter. This form of packet lifetime control depends on having synchronized system clocks on all of the systems that host the SCPS-NP. It offers a bit-efficiency advantage in some cases, in that the source timestamp may be the same one used by the transport layer for round-trip timing. In this version of the SCPS Network Protocol Reference Implementation, clock sources are not assumed to be synchronized throughout the network. When packet lifetime control is required, the hop count parameter is used by default. Source timestamps submitted by the transport protocol will be carried in addition to the hop count field.

The SCPS-NP provides the SCPS Control Message Protocol (SCMP) to accomplish necessary signaling between SCPS-NP entities. It supports essentially similar error and query services as are found in the Internet Control Message Protocol (ICMP), but with additions for corruption-experienced and link-outage signaling. (ICMP already has signaling to report congestion - it is called the “Source Quench” message.)

2.6 Invoking the Capabilities of the SCPS Network Protocol

As noted in Section 1, the SCPS-NP is not called directly by the user; the SCPS-TP is the primary “user” of the NP. Invoking many of the capabilities of the NP is done within the TP code and is conditional upon the state of the connection and/or network. However, the user may *indirectly* affect certain capabilities of the NP by setting socket options or by manipulating input files used at run time. Both manners of invoking the NP are described in this section.

2.6.1 Configuring Routing Tables

Routing tables are initialized by configuring the input files *npNextHopFile* and *npMultiNextHopFile*. They define the unicast and multicast topology, respectively, of the network. The files will be read at run time and the routing tables built from the information therein. Note that this version of the Reference Implementation of SCPS is limited to static routing.

The first file, *npNextHopFile*, specifies the unicast next hop information from the view of a particular end system. This is accomplished by listing the NP destination address for each destination as well as its corresponding next hop link layer address. For a detailed explanation and examples of this file, see Section 4.2.1 of this document.

The second routing file, *npMultiNextHopFile*, defines the multicast next hop information from the view of a particular end system. It is similar in format to *npNextHopFile*, with the main

difference being the possible existence of *multiple* next hop link layer addresses following a NP destination address. For a detailed explanation and examples of this file, see Section 4.2.2.

2.6.2 Configuring Address Translation Tables

Address translation tables provide mappings between different address types used in the SCPS Network. There are two translation tables currently defined: IP-End System and NP-Path. These tables may be initialized by the input files *npIP_NP_File* and *npPathFile*, respectively.

npIP_NP_File defines the one-to-one mappings of IP addresses to their unique end system NP addresses. It is important to recall that IP plays a dual role in the operation of SCPS. It can co-exist as a network layer along with SCPS-NP (and therefore provide IP destination addresses) as well as serve as a pseudo-link layer underneath SCPS-NP. Both of these roles necessitate the use of the *npIP_NP_File*. For a detailed explanation and example of this file, see Section 4.1.1.

npPathFile specifies, for each path address, the virtual circuit between each source and its destination(s). Therefore, each line in the file must include a path address, its corresponding NP end system address, followed by one or more destination end system addresses. For a detailed explanation and example of this file, see Section 4.1.2.

2.6.3 Completing a SCPS-NP Requirements Structure

The requirements structure allows the NP to precompute as much header information as possible. Fields that remain constant for the duration of the connection - such as source address, destination address, and transport protocol identifier - are filled in immediately. Fields that can change during a connection - e.g., checksums, timestamps, etc. - have space reserved for their values.

The requirements structure is filled out by the network service user (typically, the TP) and is forwarded to the NP upon completion. The TP performs this activity; it is **not** an application call. However, there are three fields within the requirements structure that the user may request via socket options: the use of checksums, the use of timestamps, and the level of priority (precedence). The setting of these socket options is described in detail in Section 3.

2.6.4 Configuring Network Control Requirements

The use of header checksums and timestamps are configurable network control requirements that may be set by the user. As described in Section 2.6.3, these may be requested by setting socket options. If checksums are requested, they shall be provided. Timestamps will only be provided upon request if there is sufficient clock synchronization in the network. If there is not, an error will return from the request.

Hop counts are also a network control requirement, but they cannot be directly requested by the user. Their use is dictated by the configuration setting `LOOP_CONTROL` within the file `scps_defines.h`. The default value is `TRUE` (that is, loop control is turned “on”). This setting is recommended where routing loops or misrouted datagrams are possible. If loop control is turned on and timestamps are not set via a socket option, then hop counts will be used instead. To turn off loop control, the setting must be changed to `FALSE` and the SCPS source code recompiled and linked.

2.6.5 Setting MIB Configuration Parameters and Thresholds

The MIB configuration parameters and thresholds (as defined `mib.h`) may be set in the routine `scps_np_init()`. This routine is located in the source file `scps_np.c`. The format for setting these fields is:

```
mib.field = VALUE;
```

For example, to set the parameter `npDefaultHopCount` to 15:

```
npmib.npDefaultHopCount = 15;
```

The SCPS source code must then be recompiled and linked.

2.6.6 Sending a Datagram Directly From the Requirements Structure

Datagrams may be sent by the network service user by calling the routine `scps_np_dg_request()`. This call is within the TP code and is **not** an application call. Moreover, this routine cannot be used with static routing due to the connection-oriented nature of such routing.

2.6.7 Acquiring a SCPS-NP Template

Once the requirements structure is completed by the network service user (as discussed in Section 2.6.3), a call is made to `scps_np_get_template()`. This routine returns a SCPS-NP template, which includes a partially-completed NP header. This header contains all the static information entered into the requirements structure (destination address, transport protocol identifier, etc.) as well as space reserved for non-static fields.

2.6.8 Sending a Datagram Using the SCPS-NP Template

This is the default method for sending a datagram using the current release of the Reference Implementation of SCPS.

The network service user obtains a SCPS-NP template via the procedure outlined in Section 2.6.7. When the network service user has data it wishes to send, it makes a call to

scps_np_trequest() with the data and the SCPS-NP template. The call is performed by the TP and is **not** an application call.

2.7 Configuration Alternatives

There are two main alternatives for configuring the SCPS Network Protocol to operate in IP-based ground networks. Each alternative has its own advantages, which are discussed in this subsection. The two alternatives may be referred to as “encapsulation” and “translation”.

2.7.1 Encapsulation

Figure 1 illustrates the basic concept of the encapsulation approach: SCPS-NP packets are formed at the data source, and routed through the ground network by *encapsulating* them in IP packets or UDP/IP packets. On the right half of the figure, which represents the typical ground network, IP packets are used to carry SCPS-NP packets that, in turn, carry TP packets and user data. (In this figure, link headers are not shown, and the SCPS-SP is assumed to be not in use.) The center box in the figure represents a router at the point where the wired network meets the wireless (space) network. In this router, the IP header is removed (for space-bound packets) or added (for packets coming from the space-based portion of the network). On the left half of the figure, which represents the wireless portion of the network, packets do not carry the IP headers, reducing header overhead.

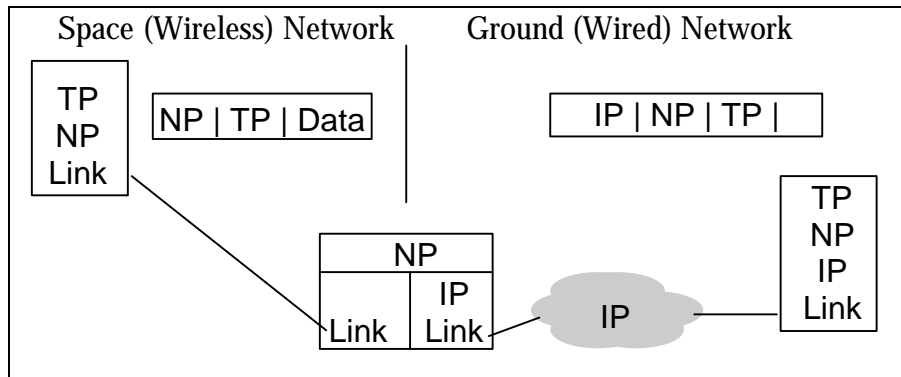


Figure 1. Encapsulating Network Configuration

The primary advantage of the encapsulating approach is that the signaling capabilities of the SCPS Control Message Protocol are preserved end-to-end throughout the network. At routers and end systems that support the SCPS-NP, signals indicating congestion, corruption,

and link-outage can be generated (assuming that those conditions can be detected in the local system). Note, however, that congestion or corruption loss or link outages that occur *within* the IP network are not signaled by the NP. Until the techniques of RED with Explicit Congestion Notification are widely deployed throughout the Internet, congestion signaling will not be available. Corruption is not a significant problem within the Internet, so no signaling for corruption is currently necessary. Likewise, link outage is not currently a significant problem within the wired portions of the Internet.

Note that this version of the Reference Implementation supports encapsulation in either IP or UDP/IP, controlled by a compile-time definition.

2.7.2 Translation

Figure 2 illustrates the translation approach to routing through the ground network. In this approach, the ground-based system on the right supports IP, but not the SCPS-NP. At the router in the center of the figure, for ground-bound packets the information from the NP headers are used to form IP headers, which replace the NP headers. Similarly, space-bound packets have the IP headers removed and replaced with NP headers containing similar information.

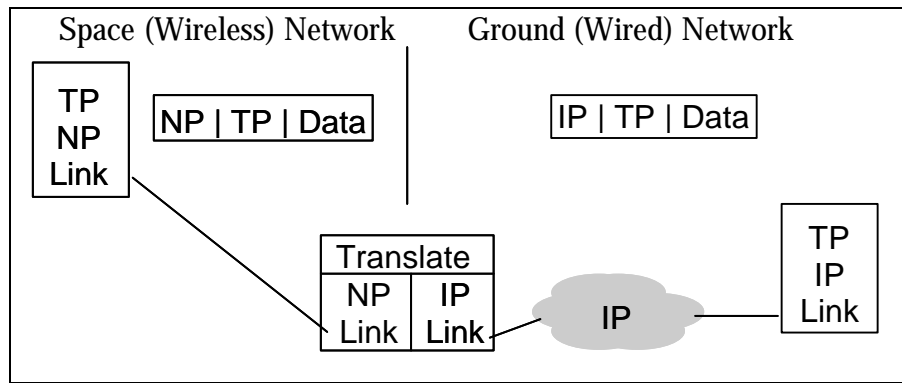


Figure 2. Translating Network Configuration

The main advantage of this approach is that the ground-based system on the right may use either a commercial implementation of TCP (at a loss of the SCPS enhancements) or a SCPS-enhanced TCP implementation over regular IP. The SCPS Control Message Protocol's congestion signal can be propagated to the ground-based system via the Internet Control

Message Protocol (although many commercial TCP implementations do not respond to it or respond incorrectly).

Software for IP-NP translation is not provided with this release of the Reference Implementation.

Section 3

Configuring SCPS-NP to Enable Specific Capabilities

As noted earlier, applications do not interface directly with SCPS-NP. However, SCPS-NP *can* receive instructions through the use of SCPS socket options. (For a more detailed discussion of SCPS-TP and socket options, see the SCPS-TP user's manual referenced in Section 5 of this document.) The SCPS Reference Implementation provides a large degree of flexibility in the enabling and disabling of capabilities, as well as queries and modification of operational parameters. The following commands describe the mechanisms that allow an application to perform such operations for SCPS-NP.

3.1 Checksum

Applications have the ability to dictate the use of checksums on a per socket basis. This can be achieved by making the `scps_setsockopt()` call shown in Figure 3.

```
one = 1;

scps_setsockopt(socket, NP_PROTO_NP, SO_CHECKSUM, &one, sizeof(one));
```

Figure 3. Format for setsockopt call: checksum

Checksums should be used whenever the possibility of bit errors exists.

3.2 Timestamps

Applications have the ability to enable the use of timestamps on a per socket basis. This can be achieved by making the following `scps_setsockopt()` call, where “fmt” specifies of the defined SCPS timestamps of type “ts_fmt”:

```
fmt = SCPS32;

scps_setsockopt(socket, NP_PROTO_NP, SO_TIMESTAMP, &fmt, sizeof(fmt));
```

Figure 4. Format for setsockopt call: timestamps

An application making this `setsockopt` call is **not** guaranteed the use of timestamps. Timestamps may be used whenever sufficient clock synchronization exists. However, if the SCPS-NP determines that such synchronization does not exist, the SCPS-NP will return the `setsockopt` call with an error indication.

3.3 Precedence

Applications may dictate the level of precedence (or priority) on a per socket basis. This can be achieved by making the `scps_setsockopt()` call shown in Figure 5, where “lev” is any valid precedence level (0-15):

```
lev = DEF_PREC;  
  
scps_setsockopt(socket, NP_PROTO_NP, SO_PRECEDENCE, &lev, sizeof(lev));
```

Figure 5. Format for setsockopt call: precedence

Section 4

Format Requirements for Input Files

Unlike the SCPS-TP, the SCPS-NP is not called directly by the application. Instead, the *network service user* is typically the SCPS-TP or the underlying link-layer protocol(s). Aside from the socket options discussed in Section 3, the only other interface to the SCPS-NP is through routing and translation files. The files are input at run-time and will be used to create internal routing tables.

It should be noted that the current release of the SCPS software does not reside in kernel. As such, this version of SCPS-NP relies on an underlying IP network protocol to act as its link layer. This necessitates that all ‘link layer’ addresses passed by the SCPS-NP be IP addresses. It also impacts the current format of the SCPS-NP input files.

4.1 Translation File:

The two files described below provide mappings between different address types and are used by the SCPS-NP for proper routing.

4.1.1 npIP_NP_File

This file provides the IP-to-NP (and vice versa) translations for each address passed to the SCPS-NP layer and is the same for all end systems in a particular SCPS Network. The format of the file should be as follows, where all numbers are in hexadecimal notation:

```
IP_Addr_1 NP_Addr_1
IP_Addr_2 NP_Addr_2
IP_Addr_3 NP_Addr_3
```

Example 1: Figure 6 shows a network of six end systems, each with an IP address and its SCPS-NP Basic End System (2 octets) translation. Recall that a SCPS-NP End System address can be any two-octet number with the proper format. In particular, the multicast indicator must be in the least significant (right-most) bit. Therefore, odd numbers are not allowed in the unicast address space. This is why c0307291 is not mapped to 7291 -- an odd number -- in Figure 6. Similarly, c0307295 is not mapped to 7295. Instead, these are mapped to other, unique, even numbers.

Note: for the following several examples, consider the address mappings of Figure 6 to represent the end systems A, B, C, D, E, F (in that order).

c0307290	7290
c0307291	7292
c0307294	7294
c0307295	7296
c0307298	7298
c03072a6	72a6

Figure 6. Example npIP_NP_File

Warnings: (1) Every end system in the network must have a copy of this file. The simplest method is to create the file once and then copy it to all SCPS systems. (2) Both the name of the file and its format must be preserved for the SCPS-NP to read the data correctly. i.e., the file must be named *npIP_NP_File* and its entries must be in hexadecimal notation.

4.1.2 npPathFile

Recall that a path address is basically a permanent virtual circuit between a source and one or more destinations. This file contains the source-destination(s) associations for each path address in the network. The format of this file is the path address in the first column, the source NP address in the second column, and the destination(s) NP address(es) in the remaining column(s).

```

path_addr_A  src_A_NP_addr_1  dst_A_NP_addr_1  [dst_A_NP_addr_2  etc...]
path_addr_B  src_B_NP_addr_1  dst_B_NP_addr_1  [dst_B_NP_addr_2  etc...]
path_addr_C  src_C_NP_addr_1  dst_C_NP_addr_1  [dst_C_NP_addr_2  etc...]

```

Example 2: As an illustration of this file, consider the sample *npPathFile* shown in Figure 7. In the first line, path address 0x55b2 defines a source-destination pair from 0x7290 and 0x7292. The second line, 0x55b3, is a multicast path address with source 0x7290 and destinations 0x7292, 0x7294, and 0x7296.

55b2	7290	7292
55b3	7290	7292 7294 7296

Figure 7. Example npPathFile

Only if path addresses are defined in a SCPS Network need this file exist. However, if path addresses are to be used, then this file *must* be created and it must distributed throughout the SCPS Network in which the addresses are defined. Moreover, both the name of the file

and its format must be preserved for the SCPS-NP to read the data correctly. i.e., the file must be named *npPathFile* and its entries must be in hexadecimal notation.

4.2 Routing File:

The following routing files specify the next hop information for each end system in a SCPS Network.

4.2.1 npNextHopFile

This file contains the unicast next hop information *from the view of a particular end system*. In addition, the first line of the file should specify that end system's NP address. The remainder of the file should list (in the first column) the NP destination address and (in the second column) the link layer next-hop address. (For routing NP over IP, the link layer next-hop address will be the next-hop IP address.) It is this information that allows the SCPS-NP to carry out its own routing while restricting the IP to link layer duties. The current release of the SCPS Reference Implementation is limited to static routing.

The format of the *npNextHopFile* is as follows:

```
my_local_IP_addr
dest1_NP_addr      nexthop1_linklayer_addr
dest2_NP_addr      nexthop2_linklayer_addr
dest3_NP_addr      nexthop3_linklayer_addr
```

Example 3: As an illustration of this file, consider a network of three end systems: A, B, C, where A is linked to B and B is linked to C. All links are bi-directional. Observe that if A wants to talk to B, it can do so directly. However, if it wants to talk to C, it must first go through B. Likewise, C can talk to B directly but must use B as an intermediary if it wants to talk to A. B, on the other hand, can talk to both A and C directly. The *npNextHopFiles* of A, B and C are given in Figure 8, Figure 9, and Figure 10, respectively.

c0307290
7292 c0307291
7294 c0307291

Figure 8. Example npNextHopFile for End System A

c0307291
7290 c0307290
7294 c0307294

Figure 9. Example npNextHopFile for End System B

c0307294
7290 c0307291
7292 c0307291

Figure 10. Example npNextHopFile for End System C

Warnings: Warnings: (1) Every end system in the network must have a copy of this file. It should be clear, however, that each end system's version may be different due to the connectivity of the network. (2) Both the name of the file and its format must be preserved for the SCPS-NP to read the data correctly. i.e., the file must be named *npNextHopFile* and its entries must be in hexadecimal notation with the local IP address as the first line.

4.2.2 npMultiNextHopFile

This file contains the next hop information for each predefined multicast group *from the view of a particular end system*. In addition, the first line should contain the multicast groups to which this particular end system belongs. The remainder of the file should list (in the first column) the SCPS-NP multicast destination End System address, while the remaining column(s) on a line should list the link layer addresses of the next hop(s). There may be multiple next hops.

The format of *npMultiNextHopFile* is as follows, where the brackets indicate entries that may or may not always be listed :

```
local_mc_addr1(possibly 0) [local_mc_addr2      etc....]
dest_mc_addr_A  nxthp_linklayr_addr_A1  [nxthp_linklayr_addr_A2  etc..]
dest_mc_addr_B  nxthp_linklayr_addr_B1  [nxthp_linklayr_addr_B2  etc..]
dest_mc_addr_C  nxthp_linklayr_addr_C1  [nxthp_linklayr_addr_C2  etc..]
```

(Recall that a SCPS-NP address denotes multicast by setting the multicast flag, M-ID, of the address. The following examples use a 2-octet Basic End System address format with M-ID set.)

Example 4: As an illustration of this file, consider the example network described above: A connected to B, B connected to C. Now add a third end system D (IP address 0xc0307296), connected to B. Define a multicast group 0x8001 which consists of the source A and destinations C and D. If A sends a message to its multicast group 0x8001, it would travel from A to B, from B to C and D. The *npMultiHopNextFile* for these end systems in the network are illustrated in Figure 11, Figure 12, and Figure 13, respectively. Observe that C and D have copies of the same file, shown in Figure 13.

8001
8001 c0307292

Figure 11. Example npMultiNextHopFile for End System A

In Figure 11, we see that A belongs to multicast group 0x8001 and only forwards datagrams sourced from this group to B's IP address.

0
8001 c0307294 c0307296

Figure 12. Example npMultiNextHopFile for End System B

Recall that the first line specifies the multicast group(s) belonged to by the local system. End system B is not in any multicast groups and so will have a "0" entry in its first line in Figure 12. However, it is along the multicast path and, moreover, is at a fork in the multicast path. Hence, B has *two* next hop entries for multicast destination 0x8001, as shown in the second line of Figure 12. This means that it must make a copy of any datagrams received from A and send one to C and the other to D.

8001

Figure 13. Example npMultiNextHopFile for End Systems C & D

End systems C and D, on the other hand, have a single-line file specifying the multicast groups to which they belong, as shown in Figure 13. In this case, this consists of just one group: 8001. Since there are no more end systems to which to forwards datagrams from either C or D, they have no entries for next hop information.

Example 5: Suppose we add two more end systems to the example given above. End system E is connected to D but is not in any multicast group, while end system F is connected to C and is also a member of multicast group 0x8001. If A wants to send a multicast message

to its group, the message would go from A to B, B to C and D, C to F. The *npMultiNextHopFiles* for A, B and D would remain unchanged. (D's would remain unchanged because E is not a member of the group.) End system E would not have a *npMultiNextHopFile* because it is neither a member of any multicast group nor is it along the path for any multicast group. Note that this is the difference between B (which has a *npMultiNextHopFile*) and F (which does not): although both are not members of any multicast groups, B is along a multicast path. The *npMultiNextHopFiles* for C and F are given in Figure 14 and Figure 15, respectively.

8001	
8001	c03072a6

Figure 14. Example *npMultiNextHopFile* for End System C

Previously, C's *npMultiNextHopFile* indicated just what multicast groups to which it belonged. Now, however, C must forward multicast messages from group 0x8001 to F. Figure 14 illustrates this new responsibility. Together, they dictate to C that it must not only listen for datagrams sourced from 0x8001, but also forward them.

Figure 15 specifies that F is a member of multicast group 0x8001 but will not forward any datagrams received from that (or any) group.

8001

Figure 15. Example *npMultiNextHopFile* for End System F

Section 5

Suggested Reading

This section presents some suggested reading for readers with varying levels of familiarity with IP and routing in general.

Readers with little previous familiarity with IP should consider reviewing an introductory text on the subject. One excellent example is TCP/IP Illustrated, Volume 1, by W. Richard Stevens (Copyright 1994, Addison-Wesley Professional Computing Series).

A primer on networks and routing is Data Networks, Second Edition, by Dimitri Bertsekas and Robert Gallager. (Copyright 1992, Prentice Hall). Sections 1 and 5 are particularly relevant.

Another introductory text for understanding IP and general routing concepts is Internetworking with TCP/IP, Volume II, Design, Implementation and Internals by Douglas E. Comer and David L. Stevens. (Copyright 1994, Prentice Hall).

The internals of the IP code are explained in detail in TCP/IP Illustrated, Volume 2, The Implementation, by Gary R. Wright and W. Richard Stevens (Copyright 1995, Addison-Wesley Professional Computing Series).

The following Requests for Comments provide the specifications for the Internet Protocols on which SCPS-NP, SCMP and the MIB are adapted. Universal Resource Locators (URLs) are provided for world-wide web access.

R. Braden. *Requirements for Internet Hosts*. AIB STD 3. RFC 1122, October 1989. <URL: <http://ds.internic.net/rfc/rfc1122.txt>>

K. McCloghrie and M. Rose. *Management Information Base for Network Management of TCP/IP-Based Internets: MIB-II*. AIB STD 17. RFC 1213, March 26, 1991. <URL: <http://ds.internic.net/rfc/rfc1213.txt>>

J. Postel. *Internet Protocol - DARPA Internet Program Protocol Specification*. RFC 791, September 1981. <URL: <http://ds.internic.net/rfc/rfc791.txt>>

J. Postel. *Internet Control Message Protocol*. AIB STD 5. RFC 792, September 1981. <URL: <http://ds.internic.net/rfc/rfc792.txt>>

SCPS-NP addressing is derived, in part, from the concepts outlined in the Path Service from the Consultative Committee for Space Data Systems (CCSDS) Recommendation for Advanced Orbiting Systems (CCSDS-701.0-B-2).

The following is the companion document to the SCPS-NP and explains socket options in greater detail:

R. Durst, P. Feighery, E. Travis, *User's Manual for the SCPS Transport Protocol*, MITRE WN97W0000018, March 1997.

Glossary

CCSDS	Consultative Committee for Space Data Systems
DOD	Department of Defense
ECN	Explicit Congestion Notification
IAB	Internet Activities Board
ICMP	Internet Control Message Protocol
IP	Internet Protocol
MIB	Management Information Base
RED	Random Early Detection
RFC	Request for Comments
SCMP	SCPS Control Message Protocol
SCPS	Space Communications Protocol Standards
SCPS-FP	SCPS File Protocol
SCPS-NP	SCPS Network Protocol
SCPS-SP	SCPS Security Protocol
SCPS-TP	SCPS Transport Protocol
SCPS-TWG	SCPS Technical Working Group
STD	Standard
TCP	Transmission Control Protocol
UDP	User Datagram Protocol
URL	Universal Resource Locator