

Network Working Group  
Request for Comments: 4824  
Category: Informational

J. Hofmueller, Ed.  
A. Bachmann, Ed.  
I.O. zmoelnig, Ed.  
1 April 2007

## The Transmission of IP Datagrams over the Semaphore Flag Signaling System (SFSS)

### Status of This Memo

This memo provides information for the Internet community. It does not specify an Internet standard of any kind. Distribution of this memo is unlimited.

### Copyright Notice

Copyright (C) The IETF Trust (2007).

### Abstract

This document specifies a method for encapsulating and transmitting IPv4/IPv6 packets over the Semaphore Flag Signal System (SFSS).

### Table of Contents

1. Introduction .....	2
2. Definitions .....	2
3. Protocol Discussion .....	3
3.1. IP-SFS Frame Description .....	3
3.2. SFS Coding .....	4
3.3. IP-SFS Data Signals .....	5
3.4. IP-SFS Control Signals .....	6
3.5. Protocol Limitations .....	7
3.6. Implementation Limitations .....	7
4. Interface Discussion .....	7
4.1. Data Link Control .....	8
4.2. Establishing a Connection .....	8
4.3. State Idle .....	8
4.4. Session Initiation .....	8
4.5. State Transmitting .....	9
4.6. State Receiving .....	10
4.7. Terminating a Connection .....	11
4.8. Further Remarks .....	11
5. Security Considerations .....	11
6. Acknowledgements .....	11
7. References .....	12

## 1. Introduction

This document specifies IP-SFS, a method for the encapsulation and transmission of IPv4/IPv6 packets over the Semaphore Flag Signaling System (SFSS). The SFSS is an internationally recognized alphabetic communication system based upon the waving of a pair of hand-held flags [JCroft, Wikipedia]. Under the SFSS, each alphabetic character or control signal is indicated by a particular flag pattern, called a Semaphore Flag Signal (SFS).

IP-SFS provides reliable transmission of IP datagrams over a half-duplex channel between two interfaces. At the physical layer, SFSS uses optical transmission, normally through the atmosphere using solar illumination and line-of-sight photonics. A control protocol (Section 4) allows each interface to contend for transmission on the common channel.

This specification defines only unicast transmission. Broadcast is theoretically possible, but there are some physical restrictions on channel direction dispersion. This is a topic for future study.

The diagram in Figure 1 illustrates the place of the SFSS in the Internet protocol hierarchy.

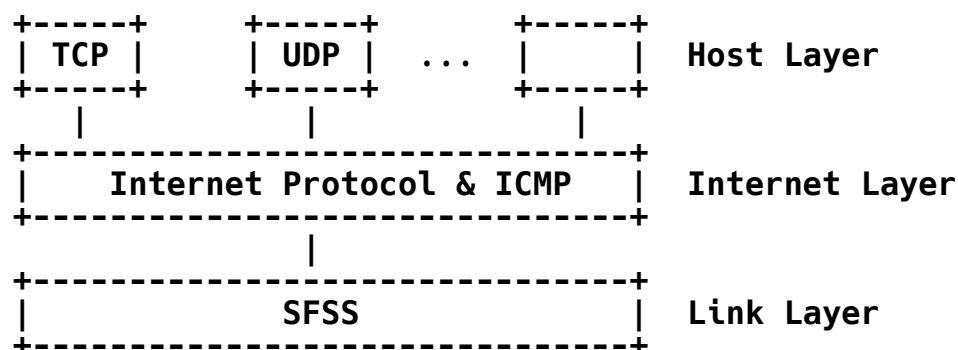


Figure 1: Protocol Relationships

## 2. Definitions

**Link:** A link consists of two (2) interfaces that share a common subnet.

**Link Partner:**  
The opposite interface.

**Session:** The transmission of an entire IP datagram.

**SFS:** One Semaphore Flag Signal, i.e., one flag pattern (Section 3.2).

**SFSS:** The Semaphore Flag Signaling System.

**IP-SFS:** IP over Semaphore Flag Signaling System.

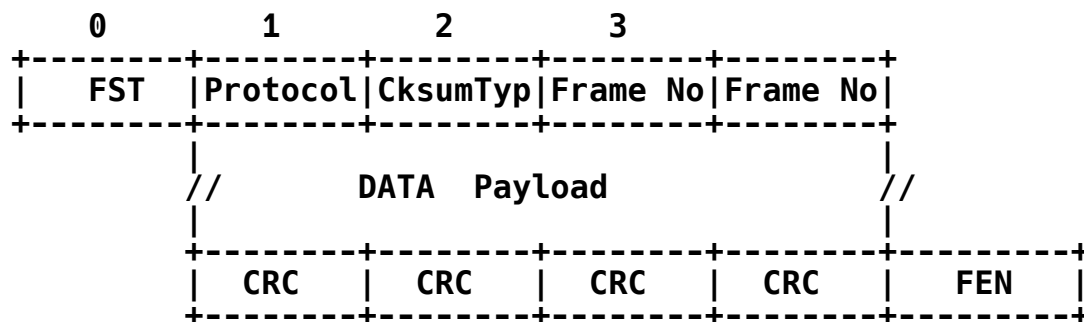
### 3. Protocol Discussion

IP-SFS adapts the standard SFSS to encode an alphabet of 16 signals (flag patterns) to represent data values 0-15 (Section 3.2.1) and 9 signals to represent control functions (Section 3.2.2). With 16 data signals, IP-SFS transmission is based upon 4-bit nibbles, two per octet. Each of the signal patterns defined in Section 3.2 is called an SFS.

#### 3.1. IP-SFS Frame Description

IP datagrams are formatted into IP-SFS frames by adding IP-SFS headers and trailers. Figure 2 shows the format of one IP-SFS frame. The frame is delimited by a control SFS called FST (Frame Start) and a control SFS called FEN (Frame End). It is composed of a series of 4-bit nibbles, one per SFS.

An IP datagram will be fragmented into multiple successive IP-SFS frames if necessary. When an IP datagram is fragmented into N frames, the first frame will be sent with frame number N-1, the second with frame number N-2, ..., and the last with frame number 0.



Note that each field represents one SFS or 4 bits.

Figure 2: IP-SFS Frame Format

**FST:** Frame Start control SFS

**Protocol:** 4 bits -- Internetwork-layer protocol code

- 0 None.
- 1 For IPv4.
- 2 For IPv6.
- 3 For IPv4 frame gzip-compressed.
- 4 For IPv6 frame gzip-compressed.
- 5...15 Reserved for future use.

**CksumTyp:** 4 bits (one data SFS) -- Checksum Type

- 0 none.
- 1 CCITT CRC 16 (polynomial:  $x^{16} + x^{12} + x^5 + 1$ ).
- 2...15 Reserved for future use.

**Frame No:** 8 bits (2 data SFSs):  
Frame number for fragmented IP datagram.

**DATA:** 0 to 510 data SFSs (Section 3.2.1) representing 0 to 255 octets of payload.

**CRC:** 16 bits as four data SFSs.  
CRC checksum. Preset to 0xFFFF. One's complement of checksum is transmitted.

**FEN:** Frame ENd control SFS.

The number of transmitted SFSs per minute (Spm) depends on the experience of participating interfaces. Resulting link speed in bits per second for IP-SFS is  $(\text{Spm}/60)*4$ , not counting framing overhead.

### 3.2. SFS Coding

Data signals and control signals are based upon standard SFS encoding, as described by [JCroft], [Wikipedia], and other sources on the Internet. The 16 data signals are interpreted as 4-bit nibbles, while the 9 control signals are used for data link control.

IP-SFS defines the 16 data signals by the original SFSS encodings for letters A to P and the 9 control signals represented by SFSS encodings Q to X.

### 3.3. IP-SFS Data Signals

Figure 3 illustrates the 16 SFSs used to transmit data frames over the link. The illustrations show each SFS as seen from the receiving side.

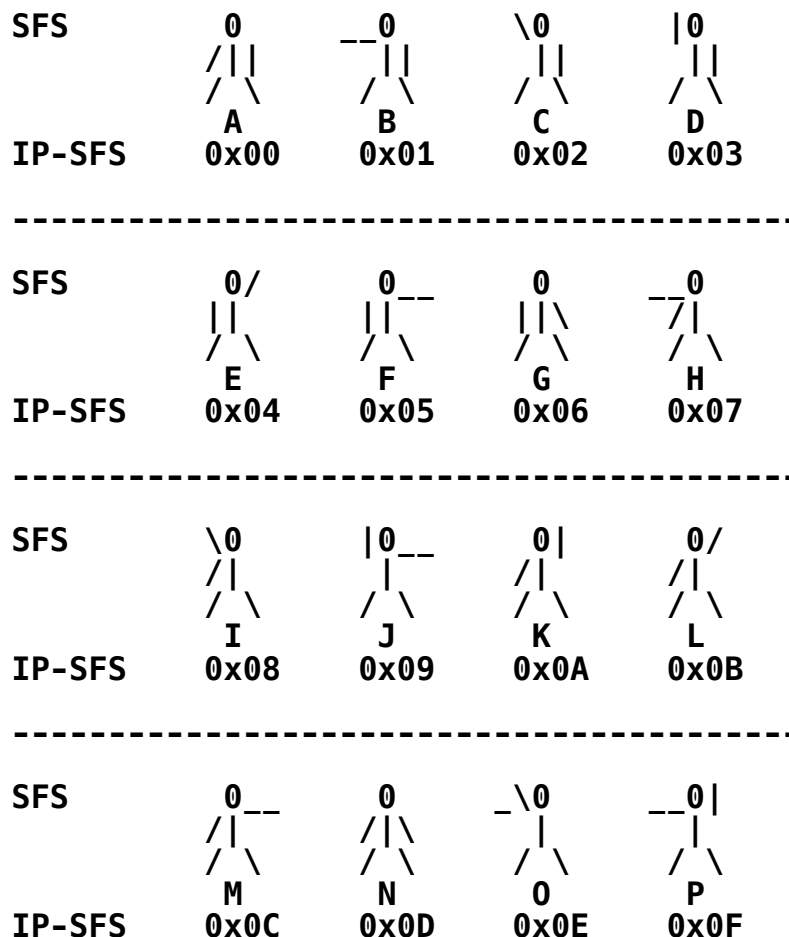


Figure 3: IP-SFS Data Signals.

### 3.4. IP-SFS Control Signals

Nine control signals are used to signal special IP-SFS conditions. Their meanings are listed in Figure 4. The illustrations show each SFS as seen from the receiving side.

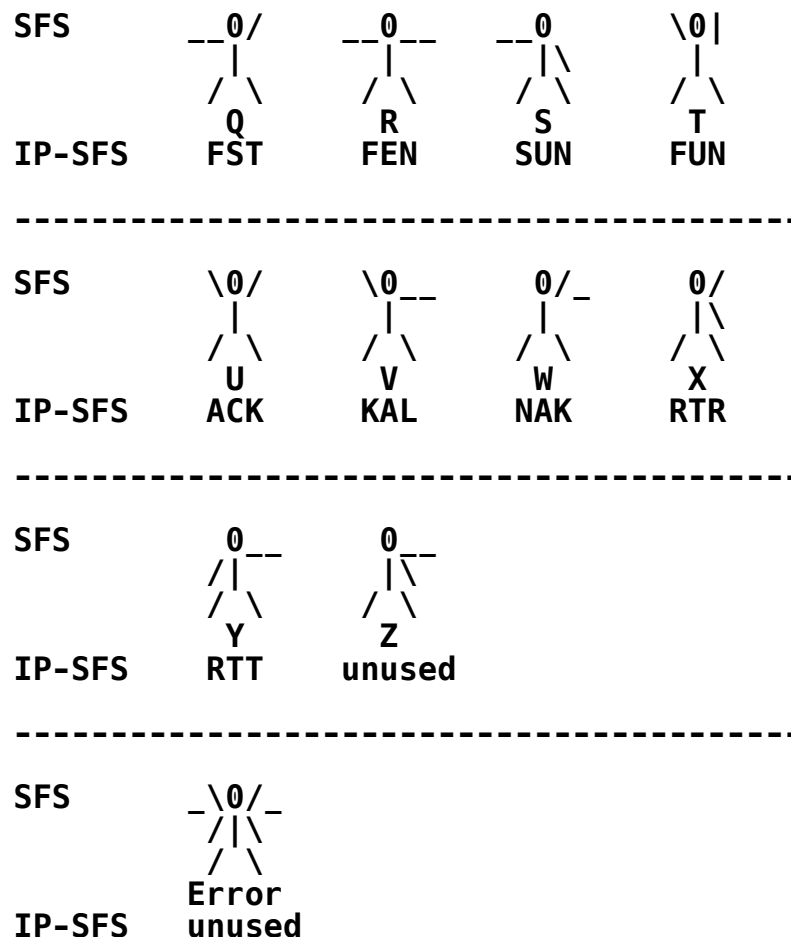


Figure 4: IP-SFS Control Signals.

**FST:** Frame S**T**art. Signals the start of a new frame.

**FEN:** Frame E**N**d. Signals the end of one frame.

**SUN:** Signal U**N**do. Cancels the transmission of one or more individual SFSSs within the current frame. This signal will be unacknowledged by the receiver.

**FUN:** Frame UNDo. As long as Frame ENd is not sent, the transmitter or the receiver may send a FUN to restart the transmission of the current frame. This signal will be unacknowledged and may be ignored by the receiver.

**ACK:** Frame ACK. Acknowledges reception of one frame.

**KAL:** KeepALive. Keep a connection alive. Is to be transmitted in State Idle at a frequency of at least KAL\_FREQ (see Section 4.2). This signal will be unacknowledged.

**NAK:** Frame No AcK. The frame received is incorrect.

**RTR:** Ready To Receive. Receiver acknowledges it is ready to receive.

**RTT:** Ready To Transmit. Sender requests permission to initiate transmission.

### 3.5. Protocol Limitations

Due to the physical characteristics of the transfer channel, bit error rates are expected to be in the range of  $1e-3$  (boy scout) to  $1e-4$  (professional sailor), and also depend a number of physical factors. Poor visibility due to weather conditions or lack of illumination (e.g., night time) can drastically increase the error rate.

IP-SFS provides no means to handle frame reordering or dual (multiple) frame reception. Thus, the protocol is not suitable in environments where interfaces are moving fast and/or when the path of light is long.

### 3.6. Implementation Limitations

Maximum payload per frame: 510 SFS (0...510) nibbles (0 to 255 octets)

Maximum SFS per frame: 518

Maximum frames per session: 255 (0...254)

## 4. Interface Discussion

An interface is constructed through the participation of one or more humans. A knowledge of the SFSS is recommended, but its absence can be compensated by a well-designed GUI.

#### 4.1. Data Link Control

This section describes the control protocol used to allocate the half-duplex data link to either interface.

Interfaces know three States: Idle, Transmitting (TX), and Receiving (RX).

#### 4.2. Establishing a Connection

IP-SFS is strictly point-to-point. Unless interface members are acquainted with each other, a brief introduction of both sides is suggested prior to setting up a link to reduce the likelihood of interface-spoofing attacks.

Interfaces must agree on two different IP addresses on the same subnet.

Interfaces are free to negotiate any period of time as TIMEOUT. Possible values for TIMEOUT are the time it takes to smoke a cigarette or the time it takes to drink a glass of water. If negotiation fails, TIMEOUT defaults to 60 seconds.

The same procedure may be applied for the KeepALive FRequency (KAL\_FRQ). The period of KAL\_FRQ ( $1/KAL\_FRQ$ ) should be at least  $5 \times TIMEOUT$ .

#### 4.3. State Idle

Interfaces in State Idle must be ready to send an IP datagram provided by a local higher-level protocol or to receive a datagram from the link-partner. Interfaces in State Idle must send keep-alive signals KAL at a frequency of at least KAL\_FRQ.

There are no further definitions for State Idle, but we recommend staying away from alcoholic beverages or other types of drugs that could lead to an increased number of FUN and/or SUN signals, a decrease in bandwidth, or an increase of line latency.

If the number of IP datagrams in the transmission queue is  $>0$ , the interface may try to initiate a session by sending an RTT to the link partner. If the link partner ready to receive, it returns an RTR signal.

#### 4.4. Session Initiation

An interface receiving a datagram from an Internet layer protocol level may start signaling RTT.



If the link partner does not respond with RTR within TIMEOUT, or the link partner responds with RTT, the interface switches to State Idle for a random period of time between 2 seconds and TIMEOUT, before resending the RTT.

If the link partner transmits RTR, the interface transmits the number of IP-SFS frames to be transmitted in this session as two SFSSs followed by another RTT. If the link partner does not transmit the same number of IP-SFS frames followed by RTR within 3\*TIMEOUT, the interface switches to State Idle.

If the link partner transmits the same number of IP-SFS frames followed by RTR, the interface switches to State Transmitting.

Unless obstructed through environmental noise or great distance, hollering can be used to request line discipline from the link partner in State Idle. The use of cellphones is also an option, whereas throwing objects or using guns is not recommended, since it could result in interface damage or destruction. This would be counterproductive.

#### 4.5. State Transmitting

Transmission of one IP-SFS frame starts with FST. After FST and before FEN have been transmitted, the interface may acknowledge a received FUN and restart the transmission of the active frame or discard the signal and continue transmission of the active IP-SFS frame.

If an error occurs by transmitting a wrong data SFS, the interface may invalidate the last data SFS by transmitting SUN followed by the correct signal. A series of incorrectly transmitted data SFSSs may be invalidated by sending a SUN for each invalid SFS, effectively back-spacing the sequence.

Control SFSSs cannot be invalidated.

If an error occurs, the interface may also transmit FUN and restart transmission of the active IP-SFS frame.

Whether the interfaces choose SUN or FUN for error correction is up to the interface, but it is suggested to use SUN for a single invalid SFS, and FUN whenever the interface failed to transmit several SFSSs in a row.

After FEN has been transmitted, the transmitting interface waits for the link partner to transmit ACK or NAK.

If ACK has been received, the transmitting interface removes the active frame and starts transmitting the next IP-SFS frame. If no frames are left, the interface switches to State Idle.

If NAK has been received, the transmission failed, and the interface must start transmitting the active frame again.

If the link partner does not transmit ACK or NAK within TIMEOUT, the transmission failed, and the interface must start retransmitting the active IP-SFS frame.

If transmission of the same IP-SFS frame fails 5 times, the interface leaves the IP datagram in the transmission queue and switches to State Idle.

#### 4.6. State Receiving

In State Receiving, the interface stores each SFS received from the link partner in the receiving queue in the order of appearance.

After FST and before FEN have been received, the interface may transmit FUN at any time to request a retransmission of the entire IP-SFS frame.

If the time between two received SFSS exceeds TIMEOUT, the receiving interface must discard all data from the active IP-SFS frame and may additionally transmit FUN. If the link partner does not continue transmitting within a second TIMEOUT period, the interface must clear the receiving queue and switch to State Idle.

If the interface receives SUN from the link partner, it must discard the last received data SFS (if any). If N SUNs are received in a row, then the last N data SFS must be discarded, unless there are no more data SFS in the frame. If there are no more data SFS in the frame to be discarded, the SUN signal must be ignored by the interface.

If the receiving interface receives FUN from the link partner, it is free to discard the frame received thus far. We suggest honoring FUN since discarding the signal will decrease bandwidth.

After FEN has been received, the receiving interface evaluates the checksum. If the checksum is good, the interface transmits ACK. If the Frame Number of the active frame is 0, the interface passes the entire data from the receiving queue to the higher level protocol, clears the receiving queue, and switches to State Idle.

If the checksum is incorrect, the interface transmits NAK.

#### 4.7. Terminating a Connection

If the interface is in State Idle and the link partner did not transmit any kind of SFS for at least five times  $1/KAL\_FRQ$ , the connection is terminated and the interfaces are free to disband.

#### 4.8. Further Remarks

Interfaces are connected to computer hardware by means of a suitable input device (RX) and a suitable output device (TX). Possible devices include keyboard, mouse, and monitor. Other means of connection are subject to availability and budget.

Although it is theoretically possible to combine the TX and RX parts of an interface in one human being, we suggest dividing the operations among at least two people per interface. For longer transmissions (multimedia streaming, video conferencing, etc.), consider rotating and providing substitutes.

Bandwidth tends to vary. Typically TX starts at about 2-4 bits/s and will decrease over time due to exhaustion and lack of concentration. When an interface in TX State signals at a rate higher than the RX interface is able to receive, signal loss occurs.

#### 5. Security Considerations

By its nature of line-of-sight signaling, IP-SFS is considered insecure. The transmission of sensitive data over IP-SFS is strongly discouraged unless security is provided by higher level protocols.

Interfaces tend to keep data in their memory. There is no way to determine the lifetime of data in memory. As a side effect, data might show up in unwanted locations at undesired times.

We are currently not aware of a technique to reliably delete data from interfaces' memory without permanent interface destruction.

#### 6. Acknowledgements

We thank Eva Ursprung and Doris Jauk-Hinz from Womyn's Art Support (WAS), Anita Hofer, Reni Hofmueller, Ulla Klopff, Nicole Pruckermayr, Manfred Rittler, Martin Schitter, and Bob Braden for all their contributions and support of this project.

## 7. References

- [JCroft] Croft, J., "Semaphore Flag Signalling System", <<http://www.anbg.gov.au/flags/semaphore.html>>.
- [Wikipedia] Wikipedia, "Modern semaphore", <[http://en.wikipedia.org/wiki/Semaphore#Modern\\_semaphore](http://en.wikipedia.org/wiki/Semaphore#Modern_semaphore)>.

## Authors' Addresses

Jogi Hofmueller (editor)  
Brockmanngasse 65  
Graz 8010  
AT

E-Mail: [ip-sfs@mur.at](mailto:ip-sfs@mur.at)

Aaron Bachmann (editor)  
Ulmgasse 14 C  
Graz 8053  
AT

E-Mail: [ip-sfs@mur.at](mailto:ip-sfs@mur.at)

IOhannes zmoelnig (editor)  
Goethestrasse 9  
Graz 8010  
AT

E-Mail: [ip-sfs@mur.at](mailto:ip-sfs@mur.at)

## Full Copyright Statement

Copyright (C) The IETF Trust (2007).

This document is subject to the rights, licenses and restrictions contained in BCP 78, and except as set forth therein, the authors retain all their rights.

This document and the information contained herein are provided on an "AS IS" basis and THE CONTRIBUTOR, THE ORGANIZATION HE/SHE REPRESENTS OR IS SPONSORED BY (IF ANY), THE INTERNET SOCIETY, THE IETF TRUST AND THE INTERNET ENGINEERING TASK FORCE DISCLAIM ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

## Intellectual Property

The IETF takes no position regarding the validity or scope of any Intellectual Property Rights or other rights that might be claimed to pertain to the implementation or use of the technology described in this document or the extent to which any license under such rights might or might not be available; nor does it represent that it has made any independent effort to identify any such rights. Information on the procedures with respect to rights in RFC documents can be found in BCP 78 and BCP 79.

Copies of IPR disclosures made to the IETF Secretariat and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementers or users of this specification can be obtained from the IETF on-line IPR repository at <http://www.ietf.org/ipr>.

The IETF invites any interested party to bring to its attention any copyrights, patents or patent applications, or other proprietary rights that may cover technology that may be required to implement this standard. Please address the information to the IETF at [ietf-ipr@ietf.org](mailto:ietf-ipr@ietf.org).

## Acknowledgement

Funding for the RFC Editor function is currently provided by the Internet Society.