Network Working Group
Internet Draft: Network Byte Order
Document: draft-newman-network-byte-order-01.txt

C. Newman
Innosoft

February 1999

Network Byte Order

Status of this memo

This document is an Internet-Draft and is in full conformance with all provisions of Section 10 of RFC 2026.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

The list of current Internet-Drafts can be accessed at http://www.ietf.org/ietf/1id-abstracts.txt.

The list of Internet-Draft Shadow Directories can be accessed at http://www.ietf.org/shadow.html.

Abstract

This memo defines the term "network byte order" and explains why it should be used in IETF protocols and media types.

1. Terminology Used in this Document

The key words "SHOULD" and "SHOULD NOT" in this document are to be interpreted as described in "Key words for use in RFCs to Indicate Requirement Levels" [KEYWORDS].

A byte is the smallest natural grouping of bits that can be directly addressed by hardware. Modern hardware and network protocols use 8-bit bytes, also known as octets.

1.1. Background and Motivation

The document "ON HOLY WARS AND A PLEA FOR PEACE" [IEN-137] written in 1980 argues that the industry should settle on a single byte order. Since then, the IETF has largely settled on a single byte order known as "Network Byte Order" and this memo is intended to record that rough concensus. Unfortunately, the "holy war" continues among CPU manufacturers.

2. Definition of Network Byte Order

When a number is too large to fit in a single byte, multiple bytes are used to encode that number. When such numbers are sent over a byte-oriented protocol (e.g., TCP is 8-bit-byte oriented) an order for the bytes must be selected so both ends interpret the numbers in the same way independent of CPU architecture. When the bytes which make up such multi-byte numbers are ordered from most significant byte to least significant byte, that is called "network byte order" or "big endian."

For example, take the unsigned hexidecimal number 0xFEEDFACE (decimal 4,277,009,102). If this is sent as a sequence of 8-bit bytes using network byte order (big endian), the sequence would be: 0xFE, 0xED, 0xFA, 0xCE. In little endian (least significant byte to most significant byte), this would be: 0xCE, 0xFA, 0xED, 0xFE.

3. Byte Order Recommendations

The following byte ordering options have been used in protocols or media types in the past:

(1) Network byte order only.

(2) Little endian only.

(3) Run-time selection between (1) or (2)

In practice, run-time selection has often been a failure. It adds a negotiation step and creates two variants of the same protocol or media type which have to be tested independently (and often aren't, so things break).

A well-known case of (3) is the TIFF media type [TIFF]. Versions of TIFF software were released which only handled one of the two byte orders. Because of this, most modern TIFF software gives the user a choice between the two TIFF variants and calls them "Mac format" (big endian) and "PC format" (little endian) in an attempt to make the problem partially comprehensible. In this case, (3)

resulted in interoperability problems, extra user interface development work and user confusion. Even the risk of such serious consequences outweighs the minor benefits used to promote (3), therefore run-time byte-order selection SHOULD NOT be used.

The choice between (1) and (2) is arbitrary given that the cost of byte reversal on modern processors is normally negligible. However, IETF protocols and media types SHOULD use network byte order both for consistancy with existing IETF work, and also to take advantage of the network-byte-order macros (ntohl, htonl, ntohs, htons) present in the majority of TCP libraries.

3.1. Example Exception to Recommendations

In certain exceptional circumstances (usually involving high speed networks and/or low speed processors), the cost of byte reversal when sharing data between two little-endian machines may outweigh the cost of developing, testing and maintaining the two codepaths necessary for run-time selection of byte order.

3.2. Network Bit Order

For certain low-level protocols or compression-oriented media types, bit-order may be an issue. When possible, big-endian is encouraged for consistancy with Network Byte Order.

4. Security Considerations

Security sensitive code which allowed selectable byte order would have two code paths to verify for correctness rather than one. Otherwise byte order has no impact on security.

5. Multinational Considerations

Some coded character sets [CHAR-POLICY] have more codepoints than would fit in a byte. When a multi-byte character encoding scheme is used, the byte order issues in this memo apply to that encoding. An octet-based character encoding scheme such as UTF-8 [UTF-8] can avoid the issue.

6. References

[CHAR-POLICY] Alvestrand, H., "IETF Policy on Character Sets and Languages", RFC 2277, UNINETT, January 1998.

[IEN-137] Cohen, D., "ON HOLY WARS AND A PLEA FOR PEACE", IEN 137, USC/ISI, April 1980.

[TIFF] Parsons, G., Rafferty, J., Zilles, S., "Tag Image File Format (TIFF) - image/tiff MIME Sub-type Registration", RFC 2302, Northern Telecom, March 1998.

[KEYWORDS] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", RFC 2119, Harvard University, March 1997.

[UTF-8] Yergeau, F. "UTF-8, a transformation format of ISO 10646", RFC 2279, Alis Technologies, January 1998.

7. Author's Address

Chris Newman
Innosoft International, Inc.
1050 Lakes Drive
West Covina, CA 91790 USA

Email: chris.newman@innosoft.com