Network Working Group
Request for Comments: 3051
Category: Informational

J. Heath
J. Border
Hughes Network Systems
January 2001

IP Payload Compression Using ITU-T V.44 Packet Method

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

Abstract

This document describes a compression method based on the data compression algorithm described in International Telecommunication Union (ITU-T) Recommendation V.44. Recommendation V.44 is a modem standard but Annex B, Clause B.1, of the recommendation describes the implementation of V.44 in packet networks (e.g., V.44 Packet Method). This document defines the application of V.44 Packet Method to the Internet Protocol (IP) Payload Compression Protocol (RFC 2393). RFC 2393 defines a method for applying lossless compression to the payload portion of IP datagrams.

V.44 Packet Method is based upon the LZJH data compression algorithm. Throughout the remainder of this document the terms V.44 Packet Method and LZJH are synonymous.

Table of Contents

1. Introduction

1.1 General

This document specifies the application of LZJH data compression, a lossless data compression algorithm, to IP datagram payloads. LZJH data compression is to be used in conjunction with the IP Payload Compression Protocol (IPComp) [RFC2393]. This document is written with the assumption that the reader has an understanding of the IPComp protocol.

1.2 Background of LZJH Data Compression

LZJH is similar to the algorithm described in [LZ78] although it also has aspects which are similar to the algorithm described in [LZ77]. As such, it provides the execution speed and low memory requirements of [LZ78] with compression ratios that are better than [LZ77]. Originally developed for the satellite industry to compress IP datagrams independently, it is ideal for the IPComp application. The LZJH algorithm was modified to compress a continuous stream of data for a modem environment and this modified version is the basis for

Recommendation V.44. LZJH is an adaptive, general purpose, lossless data compression algorithm. It was selected by the ITU-T as the basis for Recommendation V.44 based on its performance across a wide variety of data types, particularly web HTML's, and based on its compression ratio characteristics, per MIP and memory utilized (as compared to other candidate algorithms). Its encoder is extremely efficient and can encode a two character string with 3 bits the second time that string is encountered in the data.

A typical [LZ78] compression algorithm, such as V.42bis, is not suitable for an IPComp application since it takes too long to build up its dictionary, resulting in poor compression ratios on IP datagrams that are compressed independently. It also requires too many cycles to reset an [LZ78] dictionary between datagrams which adversely affects execution times.

Similarly, a typical [LZ77] compression algorithm suffers in the IPComp application due to poor execution times. Hash tables, that help improve execution times when compressing continuous data, may cause deterioration of execution times in an IPComp application since they must be reset to an initial state between each datagram.

LZJH not only has superior execution times when encoding or decoding packet data, but the reset of the dictionary between IP datagrams is trivial. The encoder requires only the initialization of a 256 word array and a handful of variables while the decoder requires only the initialization of a handful of variables.

The LZJH algorithm uses a dictionary of 1525 entries, a total of only 16K of dictionary memory, for the IPComp application. During the encode process unmatched characters are encoded as ordinals and matched redundant strings of characters are encoded as codewords or string-extension lengths that represent the redundant strings. During the decode process the ordinals, codewords, and string-extension lengths are interpreted to re-create exactly the original datagram payload.

The details of LZJH data compression can be found in [V44].

1.3 Intellectual Property Rights

The IETF has been notified of intellectual property rights claimed in regard to some or all of the specifications contained in this document. For more information, consult the online list of claimed rights.

1.4 Specification of Requirements

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119].

2. Compression Process

The compression of datagrams is performed by a function called the Encoder.

2.1 Encoder Dictionary

The transmitting entity MUST reset the encoder dictionary prior to processing each datagram's payload, as specified in clause 7.5.1 of [V44]. This ensures that each datagram's payload can be correctly decompressed independently of any other, as is required in an environment where datagrams may be lost or received out of order.

The transmitting entity MUST flush unprocessed encoder data after the last byte of the datagram has been passed into the encoder such that the compressed datagram can be transmitted as a unit. The flush ensures that all data is processed and included in the output, i.e., the compressed datagram is complete and no data from the current datagram will be processed with the next datagram.

2.2 Encoder Output

The input to the payload compression algorithm is an IP datagram payload. The output of the algorithm is a new (and hopefully smaller) payload. The output payload contains the input payload's data in either compressed or uncompressed format. The input and output payloads are each an integral number of bytes in length.

If the uncompressed form is used, the output payload is identical to the input payload and the IPComp header is omitted. If the compressed form is used, the output payload is prepended with the IPComp header and encoded as defined in clause 6.3 of [V44].

2.3 Padding

A datagram payload compressed using LZJH always ends with a FLUSH codeword in the last one or two compressed data bytes. The FLUSH codeword may start in the 2nd to the last compressed data byte and end in the last compressed data byte or be totally within the last data byte. The FLUSH codeword is used to signal the end of the

compressed data and differentiate compressed data from padding. Any bits or bytes beyond the FLUSH codeword within the compressed payload are to be considered padding.

The size of a compressed payload MUST be in whole octet units.

3. Decompression Process

The decompression of datagrams is performed by a function called the Decoder.

3.1 Compressed Datagram

If the received datagram is compressed, the receiver MUST reset the decoder dictionary prior to processing the datagram. This ensures that each datagram can be decoded independently of any other datagram in the event datagrams are lost or received out of order. Beginning with the decoder dictionary in the initial state, as specified in clause 7.5.2 of [V44], the receiver decodes the payload data field of the datagram according to the procedure specified in clause 6.4 of [V44].

3.2 Original Uncompressed Datagram

If the received datagram is not compressed, the receiver does not perform compression decoding and passes the payload data field of the datagram unaltered to the next protocol layer.

4. IPComp Association (IPCA) Parameters

IKE [RFC2409] MAY be used to negotiate the use of the LZJH compression algorithm to establish an IPCA, as defined in [RFC2393].

4.1 Transform ID

The value of the LZJH Transform ID is IPCOMP_LZJH. This value is used to negotiate the use of the LZJH data compression algorithm using IKE.

4.2 Security Association Attributes

There are no other parameters required for the negotiation of the LZJH compression algorithm using IKE.

4.3 Manual configuration

The CPI value IPCOMP_LZJH is used for manually configured IPComp Compression Associations.

4.4 Minimum packet size threshold

As stated in [RFC2393], small packets may not compress well. Informal tests using the LZJH algorithm on internet web pages and e-mail files show that the average payload size that typically produces expanded data is approximately 50 bytes. Thus, implementations may prefer not to attempt to compress payloads of approximately 50 bytes or smaller.

4.5 Compressibility test

The LZJH algorithm, as described in [V44], is easily modified to incorporate an adaptive compressibility test, as referenced in [RFC2393]. Annex B of [V44] specifies the mechanism for including such a test in LZJH.

5. Security Considerations

This document does not add any further security considerations to those discussed in [RFC2393].

6. IANA Considerations

This document does not introduce any new name spaces. The value of IPCOMP_LZJH is assigned from the IPsec IPCOMP transform identifier space defined in [RFC2407]. IANA has assigned a value of 4 for this purpose.

7. Acknowledgements

This document is modeled upon [RFC2395].

8. References

[LZ77] Lempel, A., and Ziv, J., "A Universal Algorithm for Sequential Data Compression", IEEE Transactions On Information Theory, Vol. IT-23, No. 3, May 1977.

[LZ78] Lempel, A., and Ziv, J., "Compression of Individual Sequences via Variable Rate Coding", IEEE Transactions On Information Theory, Vol. IT-24, No. 5, Sep 1978.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC2393] Shacham, A., "IP Payload Compression Protocol (IPComp)", RFC 2393, December 1998.

[RFC2395] Friend, R. and R. Monsour, "IP Payload Compression Using LZS", RFC 2395, December 1998.

[RFC2407] Piper, D., "The Internet IP Security Domain of Interpretation for ISAKMP", RFC 2407, November, 1998.

[RFC2409] Harkins, D. and D. Carrel, "The Internet Key Exchange", RFC 2409, November 1998.

[V44] ITU Telecommunication Standardization Sector (ITU-T) Recommendation V.44 "Data Compression Procedures", November 2000.

9. Authors' Addresses

Jeff Heath
Hughes Network Systems
10450 Pacific Center Ct.
San Diego, CA 92121

Phone: 858-452-4826
Fax: 858-597-8979
EMail: jheath@hns.com

John Border
Hughes Network Systems
11717 Exploration Lane
Germantown, MD 20876

Phone: 301-601-4099
Fax: 301-601-4275
EMail: border@hns.com

10. Full Copyright Statement

Acknowledgement

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