# Europäisches Patentamt European Patent Office Office européen des brevets



(19)

\*EP001191760B1\*

(11) **EP 1 191 760 B1**

(12) **EUROPEAN PATENT SPECIFICATION**

(45) Date of publication and mention of the grant of the patent: **10.08.2005 Bulletin 2005/32**

1. Application number: **01120765.1**
2. Date of filing: **07.09.2001**

(51) Int Cl.7: **H04L 29/06**, H04L 29/14

**EP 1 191 760 B1**

(54) **Apparatus and method for header decompression** Vorrichtung und Verfahren für Kopfdekomprimierung Appareil et procédé permettant la décompression d’en-têtes

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| (84) Designated Contracting States:  **DE FI FR GB SE** | | | * **Ido, Daiji**   **Yokohama-shi, Kanagawa-ken (JP)** | |
| (30) | Priority: **11.09.2000** | **JP 2000274615** | (74) | Representative: **Lang, Johannes, Dipl.-Ing.** |
|  | **01.12.2000** | **JP 2000367514** |  | **Bardehle Pagenberg Dost** |
|  |  |  |  | **Altenburg Geissler Isenbruck** |
| (43) | Date of publication of application: | |  | **Postfach 86 06 20** |
|  | **27.03.2002 Bulletin 2002/13** | |  | **81633 München (DE)** |
| (60) | Divisional application: | | (56) | References cited: |
|  | **04009502.8 / 1 447 956** | | **•** | **CARSTEN BORMANN ET. AL.: "Robust Header** |
| (73) | Proprietor: **MATSUSHITA ELECTRIC INDUSTRIAL** | |  | **Compression (ROHC)" INTERNET DRAFT,**  **[Online] 14 July 2000 (2000-07-14), pages 1-34,** |
|  | **CO., LTD.** | |  | **XP002247470 Retrieved from the Internet:** |
|  | **Kadoma-shi, Osaka-fu (JP)** | |  | [**&lt;URL:http://www.watersprings.org/pub/id/dr**](http://www.watersprings.org/pub/id/dr) |
| (72) | Inventors: | |  | **aft-ietf-rohc-rtp-01.txt&gt; [retrieved on**  **2003-07-14]** |
| **•** | **Hata, Koichi** | | **•** | **ROHC MAILING LIST ARCHIVE, [Online] 10 July** |
|  | **Katano-shi, Osaka-fu (JP)** | |  | **2000 (2000-07-10), pages 1-9, XP002247471** |
| **•** | **Miyazaki, Akihiro** | |  | **ROHC mailing list archive Retrieved from the** |
|  | **Sakai-shi, Osaka-fu (JP)** | |  | **Internet:** |
| **•** | **Imura, Koji** | |  | [**&lt;URL:http://www.cdt.luth.se/rohc/msg00559.**](http://www.cdt.luth.se/rohc/msg00559) |
|  | **Machida-shi, Tokyo-to (JP)** | |  | **html&gt; [retrieved on 2003-07-14]** |

Note: Within nine months from the publication of the mention of the grant of the European patent, any person may give notice to the European Patent Ofﬁce of opposition to the European patent granted. Notice of opposition shall be ﬁled in a written reasoned statement. It shall not be deemed to have been ﬁled until the opposition fee has been paid. (Art. 99(1) European Patent Convention).

Printed by Jouve, 75001 PARIS (FR)

# Description

BACKGROUND OF THE INVENTION

Field of the Invention *5*

**[0001]** The present invention relates to methods for header compression/decompression in packet trans- mission and, more specifically, to a method for header compression/decompression, where reference informa- *10* tion is requested to be updated when an error occurs in packet transmission.

Description of the Background Art

*15*

**[0002]** Typical protocols recently known for data transmission over the Internet include TCP/IP (Trans- mission Control Protocol/Internet Protocol) and UDP/IP (User Datagram Protocol/Internet Protocol). In data transmission under such transmission protocol over a *20*

low and midrange bit rate transmission path, the header specified by TCP, UDP, IP, or other protocols is larger in size, disadvantageously causing overhead associated with communication. For example, to transmit 10-byte data under UDP/IP, the transmitting side has to add a *25*

28-byte header to the original data, resultantly forming a 38-byte packet, which is approximately four times larg- er in size than the original data. If such increase in size happens quite often, the transmission path is substan-

tially decreased in effective speed. *30*

**[0003]** In order to reduce communication overhead caused by the header, a header compression scheme developed by V. Jacobson and defined in RFC 1144 and RFC2508 has been known. In this scheme, among the fields of the header included in the packet, transmitted *35*

are only any field changed in value from the one includ- ed in the previous packet. Such field changed in value are not so many in the header, and therefore, in this scheme, header compression is successfully carried

out. This header compression scheme, however, is a *40*

standard for wired communication with a low transmis- sion error rate, as shown in FIG. 5, and is not efficient for a transmission path with a high transmission error rate.

**[0004]** FIG. 6 shows a communication network for *45*

wireless terminals over a cellular phone network such as W-CDMA. In recent years, the number of users of such communication network is rapidly growing. The communication network of FIG. 6 includes a wireless

transmission section where errors frequently occur. To *50*

reduce overhead caused by the header in a wireless section, one header compression scheme is known as ROHC (RObust Header Compression) studied by IETF (Internet Engineering Task Force). The detail of ROHC

is described in "draft-ietf-rohc-rtp-00.txt (29 June *55*

2000)", and in the successive "draft-ietf-rohc-rtp-01.txt (14 July 2000)". Further details to said IETF drafts can be found in a posting to the IETF ROHC WG mailing list,

(10 July 2000), "Re: AW: [rohc] SO packets with key- words", available under URL: <http://www.cdt.luth.se/ro-> hc/msg0059.html, which discloses that a refresh proce- dure is triggered if two consecutive CRCs indicate an error.

**[0005]** In ROHC, for data compression at the trans- mitting side (compressing side) and data decompres- sion at the receiving side (decompressing side), refer- ence information is shared by both sides for reference. That is, the reference information referred to for data compression at the transmitting side is also referred to for data decompression at the receiving side. By sharing the reference information, data decompression can be correctly achieved. FIG. 7 shows one example of data transmission adopting ROHC.

**[0006]** In FIG. 7, at the start of data transmission, the transmitting side and the receiving side each have held correct reference information . Consider first a case where the transmitting side transmits a header *H1* and data *D1* to the receiving side. Before transmission, the transmitting side carries out data compression on the header *H1* by referring to the reference information a. Here, the header H1 and a compressed header H'1 for transmission to the receiving side have such a relation as represented by the following equation (1).

*H'1* = *H1*\* (1).

**[0007]** In the above equation (1), an operation repre- sented by \* varies for each field of the header to be com- pressed. For example, the operation is so carried out as follows: the field does not vary if representing a UDP port number; the field is generally increased in value by 1 if representing an RTP sequence number; and the field is increased in value by 50 if representing an RTP times- tamp.

**[0008]** As such, the reference information  includes

all information required for compression of each field as described above. Therefore, if the receiving side holds the correct reference information  having the same contents as that held in the transmitting side, the receiv- ing side can correctly decompress the received com- pressed header *H'1* into the original header *H1*, thereby obtaining the correct header *H1* and data *D1.* Similarly, headers and data *H2* and *D2*, *H3* and *D3*, and *H4* and *D4* are transmitted after each header is compressed by referring to the reference information .

**[0009]** Next, consider a case where the reference in- formation is changed. FIG. 8 shows an example of data transmission where the reference information is changed during the transmission. In FIG. 8, after the header *H2* and the data *D2* are transmitted, the refer- ence information is changed from  to , and the header H3 is compressed by referring to the changed reference information .

**[0010]** For example, assume that the RTP timestamp of the header to be transmitted is increased by 50, but,

at the time of transmission of the data D3, such increase is changed to by 100. Under this assumption, the trans- mitting side changes the reference information  held so far containing that "The RTP timestamp is increased

by 50" into the reference information  containing that *5*

"The RTP timestamp is increased by 100". To update the reference information, as shown in FIG. 8, the re- ceiving side refers to update information further provid- ed to the compressed header to be transmitted (here, a

header H'3). *10*

**[0011]** In some cases, the reference information may be updated even if the update information is not explic- itly transmitted. One example header compression scheme taken in such cases is briefly described below.

In the compressed header, the sequence number is as- *15*

signed 4 bits capable of representing integers from 0 to 15, but not 16 or more. Therefore, any integer *N* equal to 16 or more is represented by *Nmod16*. Thus, the re- ceiving side finds the sequence number by using an

equation *L*\*16 + (received sequence number), where *L 20*

is incremented by 1 whenever the received sequence number is changed from the maximum value (here, 15) to the minimum value (here, 0). Here, the update infor- mation is not explicitly transmitted. Instead, when the

sequence number becomes larger than the maximum *25*

value, the reference information is updated on both sides.

**[0012]** FIG. 9 is a block diagram showing the structure of a header decompression apparatus that achieves the

header decompression as described above. *30*

**[0013]** In FIG. 9, a header decompression apparatus 1007 includes a packet output unit 1001, an error detec- tor 1002, a header decompressor 1003, a packet receiv- er 1004, a reference information manager 1005, and an

update request unit 1006. *35*

**[0014]** The packet receiver 1004 receives a header- compressed packet from a transmitting side, and out- puts the packet to the header decompressor 1003. The header decompressor 1003 refers to reference informa-

tion managed by the reference information manager *40*

1005 to decompress the compressed header, and out- puts the header-decompressed packet to the error de- tector 1002. If the compressed header is provided with update information, the header decompressor 1003 up- dates the reference information managed by the refer- *45* ence information manager 1005 with the update infor- mation provided to the compressed header. The error detector 1002 detects any error in the header-decom- pressed packet. If not detecting an error, the error de- tector 1002 outputs the correctly-decompressed packet *50*

to the packet output unit 1001. If detecting an error, the error detector 1002 discards the packet as not having been correctly decompressed. The update request unit 1006 receives a notification that an error is detected by

the error detector 1002, and transmits an update re- *55*

quest to the transmitting side. Specifically, according to the above document, "draft-ietf-rohc-rtp-00.txt (29 June 2000)", the update request unit 1006 transmits a NACK

packet. The reference information manager 1005 man- ages the reference information for header decompres- sion. The packet output unit 1001 outputs the header- decompressed packet.

**[0015]** As such, the header decompression apparatus 1007 detects any error in the compressed header. Here, typically, the compressed header is provided with a CRC (Cyclic Redundancy Code) for determining whether the header-decompressed packet has any error or not. Therefore, any error that occurred in the compressed header or a payload due to noise during wireless trans- mission can be detected, and the erroneous packet can be discarded.

**[0016]** FIG. 10 shows one example of data transmis- sion where an error occurs due to noise during wireless transmission. In FIG. 10, a header *H2* is compressed to be a header *H'2*, and the header *H'2* and data *D2* are wirelessly transmitted. During the wireless transmis- sion, noise or other factors affect the compressed head- er *H'2*, causing an error, which is denoted by a dotted cross in FIG. 10. As a result, as denoted by a solid cross in FIG. 10, the header cannot be correctly decom- pressed at the receiving side, and therefore the entire packet is discarded.

**[0017]** Such error as described above may occur also in the compressed header with the update information provided thereto. FIG. 11 exemplarily shows a case where an error occurs in the header with the update in- formation provided thereto, and the reference informa- tion is erroneously updated. In FIG. 11, a header *H3* is compressed to be a header *H'3*, and the header *H'3* and data *D3* is wirelessly transmitted. During the wireless transmission, noise or other factors affect the update in- formation provided to the compressed header *H'3*, caus- ing a change in the update information, which is denoted by a dotted cross in FIG. 11. Therefore, at the receiving side, the reference information is erroneously updated to reference information ', based on the changed up- date information, and the header *H3* decompressed by referring to the erroneous reference information is not the same as the original header *H3* before compression at the transmitting side. This also applies to the following headers *H4* and thereafter. As a result, as denoted by a solid cross in FIG. 11, the header cannot correctly de- compressed at the receiving side, and is generally re- garded as having an error. Therefore, the entire packet is discarded.

**[0018]** In some cases, however, the header is not re-

garded as having an error even if it has not been cor- rectly decompressed, and therefore the packet is not discarded. FIG. 12 exemplarily shows a case where a packet is not regarded as having an error even if the reference information is erroneously updated. In FIG. 12, the reference information is erroneously updated at the receiving side to become receiving-side reference information ', which is different from the reference in- formation  at the transmitting side. As a result, headers *H1* to *H4* are erroneously decompressed at the receiv-

ing side. Therefore, in general, CRC errors occur and the entire packet are discarded. However, according to principles of CRC, not all errors cannot be detected, and any erroneously-decompressed header may be acci-

dentally determined as being correct. In FIG. 12, a pack- *5* et containing the header *H3* and the data *D3* is acciden- tally determined as being correct, and is not discarded. **[0019]** As stated above, even if no error is detected, there may be a decompression error caused by noise

or erroneous reference information. Also, even if one *10*

decompression error is detected, it is impossible to tell the cause of the decompression error, that is, either noise or erroneous reference information. Therefore, according to the above background art, an update re-

quest typified by NACK is transmitted whenever an error *15* occurs. Such update request, however, is unnecessary when the reference information is correct and the error

is caused only by noise. Thus, according to the back- ground art, the more errors caused by noise, the more unnecessary update requests are transmitted, and *20* therefore the lower header compression efficiency be- comes.

SUMMARY OF THE INVENTION

*25*

**[0020]** Therefore, an object of the present invention is to provide a header decompression method in which on- ly a necessary request for updating reference informa- tion is made based on the state of an error in a header-

decompressed packet. *30*

**[0021]** This is achieved by a header decompression apparatus as defined in independent claim 1, a header decompression method as defined in independent claim 3, a computer-readable recording medium with a pro-

gram recorded therein, the program being executed in *35*

a computer system for carrying out header decompres- sion as defined in independent claim 5, and a computer program product for carrying out header decompression as defined in independent claim 6. Further preferred em-

bodiments are the subject of dependent claims 2 and 4. *40* **[0022]** The present invention has the following fea- tures to attain the object above.

**[0023]** A first aspect of the present invention is direct- ed to a header decompression apparatus for decom-

pressing a compressed header of a packet for transmis- *45* sion by referring to reference information being the same as reference information referred to for header compression by a transmitting side. In the apparatus, a packet receiver receives the packet from the transmit-

ting side. A reference information manager stores and *50*

manages the reference information. A header decom- pressor is provided with the received packet, and carries out header decompression by referring to the reference information stored in the reference information manag-

er. *55*

**[0024]** An error detector detects an error in the packet including the decompressed header. A counter/storage counts and stores the number of errors detected by the

error detector. An update request unit transmits, to the transmitting side, update information for updating the reference information, when determining, that the refer- ence information stored in the reference information manager should be updated. When the update informa- tion is transmitted from the transmitting side, the refer- ence information manager updates the reference infor- mation stored in the reference information manager based on the transmitted update information.

**[0025]** The counter/storage includes a successive er- ror counter and a successive decompression success counter. The successive error counter counts and stores the number of times *X* the packet having the error de- tected by the error detector successively appeared. The successive decompression success counter counts and stores the number of times *Y* the packet having no error successively appeared after the error detector detects the error.

**[0026]** Moreover, the update request unit determines, based on values of *X* and *Y* counted by the counter/stor- age, whether the reference information stored in the ref- erence information manager should be updated. **[0027]** As described above, in the first aspect, based on the number of decompression errors and decom- pression successes that successively appeared in the receiving side, an update request is transmitted to the transmitting side. Thus, the number of packets to be dis- carded due to error in the reference information can be reduced, and efficient packet transmission can be achieved.

**[0028]** According to a second aspect, in the first as- pect,

when *X* ÷ *Y,* the update request unit determines

that the reference information stored in the reference in- formation manager should be updated.

**[0029]** As described above, in the second aspect, when *X* ÷ *Y*, a large number of successive decompres- sion errors have occurred, and therefore there is a high possibility that the reference information has an error. Thus, the number of packets to be discarded due to error in the reference information can be reduced, and effi- cient packet transmission can be achieved. These and other objects, features, aspects and advantages of the present invention will become more apparent from the following detailed description of the present invention when taken in conjunction with the accompanying draw- ings.

BRIEF DESCRIPTION OF THE DRAWINGS

# [0030]

FIG. 1 is a block diagram showing the structure of a data compression apparatus 607 according to a first embodiment of the present invention;

FIG. 2 is a block diagram showing the structure of a data decompression apparatus 709 according to the first embodiment of the present invention;

FIG. 3 is a diagram exemplarily showing the state of decompression and values of *X* and *Y* when a packet *P1* to a packet *P11* are received;

FIG. 4 is a block diagram showing the structure of

a data decompression apparatus 908 according to *5*

a second embodiment of the present invention; FIG. 5 is a diagram showing a header compression section in wired communication;

FIG. 6 is a diagram showing a header compression section in wireless communication; *10*

FIG. 7 is a diagram exemplarily showing data trans- mission under a conventional header compression scheme of ROHC;

FIG. 8 is a diagram showing an example of data transmission where reference information is *15*

changed during the transmission in conventional header compression;

FIG. 9 is a block diagram showing the structure of a header decompression apparatus that achieves

the header decompression in conventional art; *20*

FIG. 10 is a diagram showing one example of data transmission where an error occurs due to noise during wireless transmission in the conventional header compression;

FIG. 11 is a diagram exemplarily showing a case *25*

where, in the conventional header compression, an error occurs in the header with reference update in- formation provided thereto, and the reference infor- mation is erroneously updated; and

FIG. 12 is a diagram exemplarily showing a case *30*

where, in the conventional header compression, a packet is not regarded as having an error even if the reference information is erroneously updated.

DESCRIPTION OF THE PREFERRED *35*

EMBODIMENTS

(First Embodiment)

**[0031]** A method for header compression/decom- *40* pression according to a first embodiment of the present invention is realized by a header compression appara-

tus and a header decompression apparatus as shown in FIGS. 1 and 2, respectively. With reference to the

drawings, the structure of each apparatus is described *45*

below.

**[0032]** FIG. 1 is a block diagram showing the structure of a header compression apparatus 607. In FIG. 1, the header compression apparatus 607 includes a packet

input unit 601, a CRC provider 602, a header compres- *50*

sor 603, a packet transmitter 604, a reference informa- tion manager 605, and an update request receiver 606. **[0033]** The packet input unit 601 outputs a received packet to the CRC provider 602. The CRC provider 602

provides the received packet with a CRC for output to *55*

the header compressor 603. The header compressor 603 refers to reference information managed by the ref- erence information manager 605 to compress the head-

er of the received packet, and outputs the resultant packet to the packet transmitter 604. How to compress the header will be described later. The packet transmit- ter 604 transmits the received header-compressed packet to the receiving side. The reference information manager 605 manages the reference information re- ferred to by the header compressor 603. The update re- quest receiver 606 receives an update request from the transmitting side, and notifies the reference information manager 605 of the update request. The notified refer- ence information manager 605 outputs the managed reference information to the header compressor 603, and instructs it to provide update information corre- sponding to the reference information to the com- pressed header.

**[0034]** FIG. 2 is a block diagram showing the structure of a header decompression apparatus 709. In FIG. 2, the header decompression apparatus 709 includes a packet output unit 701, an error detector 702, a header decompressor 703, a packet receiver 704, a successive decompression error counter (hereinafter, successive error counter) 705, a successive decompression suc- cess counter 706, a reference information manager 707, and an update request unit 708.

**[0035]** The packet receiver 704 receives, from the transmitting side, the header-compressed packet, and outputs it to the header decompressor 703. The header decompressor 703 refers to the reference information managed by the reference information manager 707 to decompress the compressed header, and outputs the resultant packet to the error detector 702. The error de- tector 702 detects a CRC error in the header-decom- pressed packet, and outputs, to the packet output unit 701, only any packet whose header has been correctly decompressed. The successive error counter 705 counts the number of successive decompression errors detected by the error detector 702. The successive de- compression success counter 706 counts the number of successive decompression successes detected by the error detector 702. The update request unit 708 re- fers to the numbers counted by the successive error counter 705 and the successive decompression suc- cess counter 706 to determine whether an update re- quest is required or not in a manner described later, and transmits the update request as required to the trans- mitting side. The reference information manager 707 manages the reference information for header decom- pression.

**[0036]** Note that, in the present invention, "succes- sive" decompression errors mean that successive two or more error states are observed in a header, or that only one error state is observed therein, although the latter is not generally applicable to the meaning of the word "successive". Similarly, "successive" decompres- sion successes means that successive two or more er- ror-free states are observed in a header, or that only one error-free state is observed therein. How to determine "successive" states in the present invention will be de-

scribed later.

**[0037]** Described specifically below are the operation of each of the above structured header compression ap- paratus and header decompression apparatus accord-

ing to the present embodiment. Note that a packet sup- *5* plied to the apparatuses contains data, such as video and audio, with an RTP/UDP/IP header.

**[0038]** First, the operation of the header compression apparatus shown in FIG. 1 is specifically described. The packet input part 601 outputs an externally-inputted *10* RTP/UDP/IP packet to the CRC provider 602. The CRC provider 602 computes a CRC for the entire packet, and provides the CRC to the packet.

**[0039]** The header compressor 603 refers to the ref- erence information managed by the reference informa- *15* tion manager 605 for header compression. In this head-

er compression, if the header can be decompressed with its sequence number, the reference information is not updated, and only the sequence number is included

in the header. If the header cannot be decompressed *20* with its sequence number, update information of the ref- erence information and the sequence number are in- cluded in the header. The packet transmitter 604 trans- mits the header-compressed packet to the receiving side. *25*

**[0040]** The reference information manager 605 stores and manages the reference information referred to by the header compressor 603. Notified by the header compressor 603 of update, the reference information

manager 605 updates the stored reference information. *30* Notified by the update request receiver 606 that an up- date request has been received, the reference informa-

tion manager 605 instructs the header compressor 603 to provide the update information to the header.

**[0041]** The update request receiver 606 receives an *35*

update request from the receiving side. On receiving the update request, the update request receiver 606 notifies the reference information manager 605 that the update request has been received.

**[0042]** Next, the operation of the header decompres- *40* sion apparatus 709 shown in FIG. 2 is specifically de- scribed. The packet receiver 704 receives the header- compressed packet transmitted from the packet trans- mitter 604 shown in FIG. 1, and outputs the packet to

the header decompressor 703. *45*

**[0043]** The header decompressor 703 refers to the reference information stored in the reference informa- tion manager 707 to decompress the compressed- header of the packet. The header-decompressed pack-

et is outputted to the error detector 702. If the header is *50*

provided with update information, the header decom- pressor 703 notifies the reference information manager 707 of the update information.

**[0044]** The error detector 702 checks whether any er-

ror occurs or not in the header-decompressed packet *55*

by using the CRC. Also, the error detector 702 notifies the successive error counter 705 and the successive de- compression success counter 706 of the presence or

absence of an error. If detecting any error, the error de- tector 702 discards the packet. If not detecting, the error detector 702 outputs, to the packet output unit 701, the packet with the CRC removed therefrom. The packet output unit 701 externally outputs the error-undetected RTP/UDP/IP packet.

**[0045]** The reference information manager 707 stores and manages the reference information required for header decompression. If the compressed header in- cludes the update information, the reference information manager 707 updates the stored reference information with the update information coming from the header de- compressor 703.

**[0046]** The successive error counter 705 counts the number of successive errors *X* based on the error de- tection results in the error detector 702. For example, if an error is detected in a packet, *X* is 1. Then, if an error is detected also in the next packet, *X* becomes 2. Then, if an error is detected still in the following packet, *X* is further incremented by 1 to become 3. If no error is de- tected, successiveness is interrupted, and the error counter 705 stops counting.

**[0047]** The successive decompression success counter 706 counts the number of decompression suc- cesses *Y* based on the error detection results in the error detector 702. For example, if no error is detected in a packet, *Y* is 1. Then, if no error is detected also in the next packet, *Y* becomes 2. Then, if no error is detected still in the following packet, *Y* is further incremented by 1 to become 3 . If any error is detected, successiveness is interrupted, and the successive decompression suc- cess counter 706 stops counting.

**[0048]** The update request unit 708 determines whether to request update based on a set of *X* outputted from the successive error counter 705 and *Y* outputted from the successive decompression success counter 706. This determination operation is further described in detail with reference to FIG. 3.

**[0049]** FIG. 3 is a diagram exemplarily showing de- compression operation and values of *X* and *Y* when packets *P1* to *P11* are received. In FIG. 3, assume that an error occurred in a packet *P12* not shown.

**[0050]** In FIG. 3, an error is detected in the packet *P1*, and therefore *X* = 1. In the packet *P2*, no error is detect- ed, and therefore *X* is held as 1 and *Y* = 1. Then, no error is detected from the packets *P3* and *P4*, and there- fore *X* = 1 and *Y* = 3. Then, when an error is detected in the packet *P5*, the values of *X* and *Y* for the packets *P1* to *P4* are regarded as a set representing the pres- ence or absence of an error, and determined as *X* = 1, *Y* = 3. The values of *X* and *Y* for the packet *P5* and there- after are regarded as a new set. An error is detected in the packets *P5* to *P8*, and therefore *X* = 4. No error is detected in the packets *P9* to *P11*, and therefore *Y* = 3. Thereafter, when an error is detected in the packet *P12*, the values of *X* and *Y* for the packets *P5* to *P11* are re- garded as a set, and determined as *X* = 4 and *Y* = 3. **[0051]** As such, the update request unit 708 regards

the values of *X* and *Y* as a set representing successive decompression errors and subsequent decompression successes, and compares these values with predeter- mined values. By way of example, when *X* is larger than

10 and smaller than 2, it is determined that an error oc- *5* curs in the reference information, and requests the transmitting side of updating the reference information.

These predetermined values are example only, and not restrictive. If *X* ÷ *Y*, a large number of successive de- compression errors have occurred, and therefore there *10* is a high possibility that the reference information has

an error.

**[0052]** As such, the header decompression apparatus 709 according to the present embodiment requests the

transmitting side of updating the reference information *15* based on the number of successive decompression er- rors and successes at the receiving side. Thus, the number of packets discarded due to error in the refer- ence information can be reduced, and efficient packet transmission can be achieved. Such capability of the *20* header decompression apparatus 709 can be achieved

in a general-purpose computer system. In this case, this capability is realized by a program executed in the com- puter system. The program is typically stored in a recod-

ing medium such as CD-ROM, or transmitted through a *25*

communication medium. (Second Embodiment)

**[0053]** A method for header compression/decom- *30* pression according to a second embodiment of the present invention is realized by a header compression apparatus and a header decompression apparatus as shown in FIGS. 1 and 4, respectively. Therefore, the header compression apparatus according to the second *35* embodiment is the same in structure as the header com- pression apparatus 607 shown in FIG. 1, and not de- scribed herein. Described below is the structure of the header decompression apparatus according to the present embodiment with reference to FIG. 4. *40*

**[0054]** FIG. 4 is a block diagram showing the structure of a header decompression apparatus 908. In FIG. 4, the header decompression apparatus 908 includes a packet output unit 901, an error detector 902, a header

decompressor 903, a packet receiver 904, a decom- *45* pression error detection result storage 905, a reference information manager 906, and an update request unit 907.

**[0055]** The packet receiver 904 outputs a header- compressed packet supplied from the transmitting side *50*

to the header decompressor 903. The header decom- pressor 903 refers to the reference information in the reference information manager 906 to decompress the compressed header, and outputs the resultant packet to

the error detector 902. The error detector 902 detects *55*

any error in the header-decompressed packet, and out- puts, to the packet output unit 901, any packet whose header has been correctly decompressed. The decom-

pression error detection result storage 905 counts the number of errors or successes detected in the packet by the error detector 902 for storage. The update re- quest unit 907 is supplied with the number counted by the decompression error detection result storage 905, determining whether an update request is required or not and transmitting the update request as required to the transmitting side. The reference information manag- er 906 manages the reference information for header decompression.

**[0056]** Described specifically below is the operation of the above structured header decompression appara- tus according to the present embodiment. Note that the operation of the header compression apparatus 607, that is, the transmitting side, is similar to that according to the first embodiment. Also assume that a packet sup- plied to the apparatuses contains data such as video and audio with an RTP/UDP/IP header.

**[0057]** In a header decompression apparatus 908 of FIG. 4, a packet receiver 904 receives a header-com- pressed packet transmitted from the packet transmitter 604 of FIG. 1, and outputs the packet to the header de- compressor 903.

**[0058]** The header decompressor 903 refers to the reference information stored in the reference informa- tion manager 906 to decompress the compressed- header of the packet. The header-decompressed pack- et is outputted to the error detector 902. If the header is provided with update information of the reference infor- mation, the header decompressor 903 outputs the up- date information to the reference information manager 906.

**[0059]** The error detector 902 checks the presence or absence of an error in the header-decompressed packet by using the CRC. Also, the error detector 902 notifies the detection result to the decompression error detec- tion result storage 905. If detecting any error, the error detector 902 discards the packet. If not detecting, the error detector 902 outputs, to the packet output unit 901, the packet with the CRC removed therefrom. The packet output unit 901 externally outputs the error-undetected RTP/UDP/IP packet.

**[0060]** The reference information manager 906 stores and manages the reference information required for header decompression. If the compressed header in- cludes the update information, the reference information manager 906 updates the stored reference information with the update information coming from the header de- compressor 903.

**[0061]** The decompression error detection result stor- age 905 stores the detection results of the error detector 902, that is, the number of errors *R* for preceding *W* packets. Instead of *W* or *R*, the number of decompres- sion successes may be stored.

**[0062]** The update request unit 907 determines whether to request update by comparing the values of *W* and *R* outputted from the decompression error detec- tion result storage 905 with predetermined values. For

example, if, among received 50 packets, 45 packets have errors, that is, *W* = 50 and *R* ÷ 45, the update re- quest unit 907 determines that the reference information has errors equal to or more than a predetermined value, and requests the transmitting side to update the refer- *5* ence information. Note that the above values are exam-

ple only, and not restrictive.

**[0063]** As described above, the header decompres- sion apparatus 908 according to the present embodi- ment requests updating of the reference information *10* when the number of decompression errors becomes rel- atively or substantially large in the packets previously received by the receiving side. Thus, the number of packets discarded due to error in the reference informa-

tion can be reduced, and efficient packet transmission *15* can be achieved. Such capability of the header decom- pression apparatus 908can be achieved in a general- purpose computer system. In this case, this capability

is realized by a program executed in the computer sys- tem. The program is typically stored in a recoding me- *20* dium such as CD-ROM, or transmitted through a com- munication medium.

**[0064]** While the invention has been described in de- tail, the foregoing description is in all aspects illustrative

and not restrictive. It is understood that numerous other *25* modifications and variations can be devised without de- parting from the scope of the invention.

**Claims** *30*

1. A header decompression apparatus (709, 908) for decompressing a compressed header of a packet for transmission by referring to reference informa-

tion being the same as reference information re- *35* ferred to for header compression by a transmitting side, said apparatus (709, 908) comprising:

a packet receiver (704, 904) for receiving the packet from said transmitting side; *40*

a reference information manager (707, 906) for storing and managing said reference informa- tion;

a header decompressor (703, 903), provided with the received packet, for carrying out head- *45* er decompression by referring to the reference information stored in said reference information manager (707, 906);

an error detector (702, 902) for detecting an er-

ror in the packet including the decompressed *50*

header; and

an update request unit (708, 907) for transmit- ting, to said transmitting side, an update re- quest for requesting update information for up- dating said reference information, wherein *55*

when said update information is transmitted from said transmitting side, said reference in- formation manager (707, 906) updates the ref-

erence information stored in said reference in- formation manager (707, 906) based on the transmitted update information,

# characterized in that

it is further comprised a counter/storage (705, 706, 905) for counting the number of errors detected by said error detector (702, 902), said counter/storage (705, 706) including

a successive error counter (705) for counting and storing the number of times *X* a packet having an error detected by said error detector (702) suc- cessively appeared; and

a successive decompression success coun- ter (706) for counting and storing the number of times *Y* a packet not having an error detected by said error detector successively after counting by said successive error counter (705) is stopped; and **in that**

said update request unit (708, 907) transmits said update request when determining, based on values of *X* and *Y* counted by said counter/storage (705, 706), that the reference information stored in said reference information manager (707, 906) should be updated.

1. The header decompression apparatus according to claim 1, wherein when *X*  *Y,* said update request unit (708) determines that the reference information stored in said reference information manager (707) should be updated.
2. A header decompression method for decompress- ing a compressed header of a packet for transmis- sion by referring to reference information that is the same as reference information referred to for head- er compression by a transmitting side, said method comprising:

a packet receiving step of receiving the packet from said transmitting side;

a header decompressing step, provided with the received packet, of carrying out header de- compression by referring to the reference infor- mation stored in said reference information stored;

an error detecting step of detecting error in the packet including the decompressed header; and

an update requesting step of transmitting, to said transmitting side, an update request for re- questing update information for updating said reference information;

# characterized in

further comprising a counting/storing step of count- ing the number of errors detected in said error de- tecting step, said counting/storing step including

a successive error counting step of counting and storing the number of times *X* a packet having an error detected by said error detecting step suc- cessively appeared; and

a successive decompression success count- *5*

ing step of counting and storing the number of times *Y* a packet not having an error detected by said error detecting step successively appeared after count- ing by said successive error counting step is

stopped; and **in that** *10*

in said update requesting step said update request is transmitted when determining, based on values of *X* and *Y* counted in said counter/storage step, that the stored reference information should be up-

dated. *15*

1. The header decompression method according to claim 3, wherein in said update requesting step, when *X*  *Y*, it is determined that the stored refer- ence information should be updated. *20*
2. A computer-readable recording medium with a pro- gram recorded therein, the program being executed in a computer system for carrying out header de-

compression of decompressing a compressed *25*

header of a packet for transmission by referring to reference information that is the same as reference information referred to for header compression by a transmitting side, said program comprising:

*30*

a packet receiving step of receiving the packet from said transmitting side;

a header decompressing step, provided with the received packet, of carrying out header de-

compression by referring to the reference infor- *35* mation stored in said reference information stored;

an error detecting step of detecting an error in the packet including the decompressed head-

er; and *40*

an update requesting step of transmitting, to said transmitting side, an update request for re- questing update information for updating said reference information;

*45*

# characterized in

further comprising a counting/storing step of count- ing the number of errors detected in said error de- tecting step, said counting/storing step including

a successive error counting step of counting *50*

and storing the number of times *X* a packet having an error detected by said error detecting step suc- cessively appeared; and

a successive decompression success count-

ing step of counting and storing the number of times *55*

*Y* a packet not having an error detected by said error detecting step successively appeared after count- ing by said successive error counting step is

stopped; and **in that** in said

update requesting step said update request is transmitted, when determining, based on values of *X* and *Y* counted in said counter/storage step, that the stored reference information should be updat- ed.

1. A computer program product for carrying out head- er decompression of decompressing a compressed header of a packet for transmission by referring to reference information the same as reference infor- mation referred to for header compression by a transmitting side, said program comprising code means adapted to perform, when said program is run on a data-processing system, the method steps of:

a packet receiving step of receiving the packet from said transmitting side;

a header decompressing step, provided with the received packet, of carrying out header de- compression by referring to the reference infor- mation stored in said reference information stored;

an error detecting step of detecting an error in the packet including the decompressed head- er; and

an update requesting step of transmitting, to said transmitting side, an update request for re- questing update information for updating said reference information;

# characterized in

further comprising a counting/storing step of count- ing the number of errors detected in said error de- tecting step, said counting/storing step including

a successive error counting step of counting and storing the number of times *X* a packet having an error detected by said error detecting step suc- cessively appeared; and

a successive decompression success count- ing step of counting and storing the number of times *Y* a packet not having an error detected by said error detecting step successively appeared after count- ing by said successive error counting step is stopped; and **in that**

in said update requesting step said update request is transmitted, when determining, based on values of *X* and *Y* counted in said counter/storage step, that the stored reference information should be up- dated.

# Patentansprüche

1. Header-Dekomprimierungsvorrichtung (709, 908) zum Dekomprimieren eines komprimierten Hea- ders eines Pakets zur Übertragung durch Bezug-

nehmen auf Referenzinformation, welche dieselbe ist wie Referenzinformation, auf welche sich bezo- gen wird zur Header-Komprimierung von einer übertragenden Seite, wobei die Vorrichtung (709,

908) umfasst: *5*

einen Paketempfänger (704, 904) zum Emp- fangen des Pakets von der übertragenden Sei- te;

einen Referenzinformationsverwalter (707, *10*

906) zum Speichern und Verwalten der Refe- renzinformation;

einen Header-Dekomprimierer (703, 903), ver- sehen mit dem empfangenen Paket, zum Aus- führen von Header-Dekomprimierung durch *15*

Bezugnehmen auf die Referenzinformation, die in dem Referenzinformationsverwalter (707, 906) gespeichert ist;

einen Fehlererkenner (702, 902) zum Erken-

nen eines Fehlers in dem Paket, welches den *20*

dekomprimierten Header enthält; und

eine Aktualisierungsanforderungseinheit (708, 907) zum Übertragen, an die Übertragungssei- te, einer Aktualisierungsanforderung zum An-

fordern von Aktualisierungsinformation zum *25*

Aktualisieren der Referenzinformation, worin wenn die Aktualisierungsinformation von der Übertragungsseite übertragen wird, der Refe- renzinformationsverwalter (707, 906) die in

dem Referenzinformationsverwalter (707, 906) *30*

gespeicherte Referenzinformation basierend auf der übertragenen Aktualisierungsinformati- on aktualisiert,

[dadurch gekennzeichnet, dass *35*](#_TOC_250000)

weiterhin umfasst ist ein Zähler/Speicher (705, 706, 905) zum Zählen der Anzahl an Fehlern, die durch den Fehlererkenner (702, 902) erkannt wurden, wo- bei der Zähler/Speicher (705, 706) enthält

einen Zähler (705) für aufeinander folgende *40*

Fehler zum Zählen und Speichern der Anzahl an Malen *X*, wie oft ein Paket, bei dem durch den Feh- lererkenner (702) ein Fehler erkannt wird, aufein- ander folgend erschienen ist; und

einen Zähler für aufeinander folgenden De- *45* komprimierungserfolg (706) zum Zählen und Spei- chern der Anzahl an Malen *Y*, wie oft aufeinander folgend ein Paket, bei dem durch den Fehlererken-

ner kein Fehler erkannt wird, nachdem das Zählen durch den Zähler für aufeinander folgende Fehler *50*

(705) angehalten ist; und dadurch dass

die Aktualisierungsanforderungseinheit (708, 907) die Aktualisierungsanforderung überträgt, wenn sie bestimmt, basierend auf Werten von *X* und *Y*, die

durch den Zähler/Speicher (705, 706) gezählt wur- *55*

den, dass die in dem Referenzinformationsverwal- ter (707, 906) gespeicherte Referenzinformation aktualisiert werden soll.

1. Header-Dekomprimierungsvorrichtung nach An- spruch 1, worin wenn *X*  *Y*, die Aktualisierungsan- forderungseinheit (708) bestimmt, dass die Refe- renzinformation, die in dem Referenzinformations- verwalter (707) gespeichert ist, aktualisiert werden soll.
2. Header-Dekomprimierungsverfahren zum Dekom- primieren eines komprimierten Headers eines Pa- kets zur Übertragung durch Bezugnehmen auf Re- ferenzinformation, welche dieselbe ist wie Refe- renzinformation, auf die sich zur Header-Kompri- mierung durch eine Übertragungsseite bezogen wird, wobei das Verfahren umfasst:

einen Paketempfangsschritt des Empfangens des Pakets von der Übertragungsseite;

einen Header-Dekomprimierungsschritt, ver- sehen mit dem empfangenen Paket, des Aus- führens von Header-Dekomprimierung durch Bezugnehmen auf die Referenzinformation in der gespeicherten Referenzinformation;

einen Fehlererkennungsschritt des Erkennens eines Fehlers in dem Paket, welches den de- komprimierten Header enthält; und

einen Aktualisierungsanforderungsschritt des Übertragens, an die Übertragungsseite, einer Aktualisierungsanforderung zum Anfordern von Aktualisierungsinformation zum Aktualisie- ren der Referenzinformation;

# dadurch gekennzeichnet, dass

weiter umfasst ist ein Zähl-/Speicherschritt des Zählens der Anzahl an Fehlern, die in dem Fehler- erkennungsschritt erkannt wurden, wobei der Zähl-/ Speicherschritt enthält

einen Schritt des Zählens aufeinander folgen- der Fehler, des Zählens und Speicherns der Anzahl an Malen *X*, die ein Paket, das durch den Fehlerer- kennungsschritt einen Fehler erkannt hat, aufein- ander folgend erschienen ist; und

einen Schritt des Zählens aufeinander folgen- der Dekomprimierungserfolge, des Zählens und Speicherns der Anzahl an Malen *Y*, die ein Paket, das keinen Fehler durch den Fehlererkennungs- schritt erkannt hat, aufeinander folgend erschienen ist, nachdem das Zählen durch den Schritt des Zäh- lens aufeinander folgender Fehler angehalten wur- de; und dadurch, dass

in dem Aktualisierungsanforderungsschritt die Ak- tualisierungsanforderung übertragen wird, wenn bestimmt wird, basierend auf Werten von *X* und *Y*, die in dem Zähl-/Speicherschritt gezählt wurden, dass die gespeicherte Referenzinformation aktua- lisiert werden soll.

1. Header-Dekomprimierungsverfahren nach An- spruch 3, worin in dem Aktualisierungsanforde-

rungsschritt, wenn *X*  *Y,* bestimmt wird, dass die gespeicherte Referenzinformation aktualisiert wer- den soll.

1. Computer-lesbares Aufzeichnungsmedium, mit ei- *5* nem darin aufgezeichneten Programm, wobei das Programm in einem Computersystem ausgeführt wird zum Durchführen von Header-Dekomprimie- rung des Dekomprimierens eines komprimierten Headers eines Pakets zur Übertragung durch Be- *10* zugnehmen auf Referenzinformation, welche die- selbe ist wie Referenzinformation, auf welche sich

zur Header-Komprimierung durch eine Übertra- gungsseite bezogen wird, wobei das Programm umfasst: *15*

einen Paketempfangsschritt des Empfangens des Pakets von der Übertragungsseite;

einen Header-Dekomprimierungsschritt, ver- sehen mit dem empfangenen Paket, des Aus- *20* führens von Header-Dekomprimierung durch Bezugnehmen auf die Referenzinformation in

der gespeicherten Referenzinformation;

einen Fehlererkennungsschritt des Erkennens eines Fehlers in dem Paket, welches den de- *25* komprimierten Header enthält; und

einen Aktualisierungsanforderungsschritt des Übertragens, an die Übertragungsseite, einer Aktualisierungsanforderung zum Anfordern

von Aktualisierungsinformation zum Aktualisie- *30*

ren der Referenzinformation;

# dadurch gekennzeichnet, dass

weiter umfasst ist ein Zähl-/Speicherschritt des Zählens der Anzahl an Fehlern, die in dem Fehler- *35* erkennungsschritt erkannt wurden, wobei der Zähl-/ Speicherschritt enthält

einen Schritt des Zählens aufeinander folgen- der Fehler des Zählens und Speicherns der Anzahl

an Malen *X*, die ein Paket, das durch den Fehlerer- *40* kennungsschritt einen Fehler erkannt hat, aufein- ander folgend erschienen ist; und

einen Schritt des Zählens aufeinander folgen- der Dekomprimierungserfolge des Zählens und

Speicherns der Anzahl an Malen *Y*, die ein Paket, *45*

das keinen Fehler durch den Fehlererkennungs- schritt erkannt hat, aufeinander folgend erschienen ist, nachdem das Zählen durch den Schritt des Zäh- lens aufeinander folgender Fehler angehalten wur-

de; und dadurch, dass *50*

in dem Aktualisierungsanforderungsschritt die Ak- tualisierungsanforderung übertragen wird, wenn bestimmt wird, basierend auf Werten von *X* und *Y*, die in dem Zähl-/Speicherschritt gezählt wurden,

dass die gespeicherte Referenzinformation aktua- *55*

lisiert werden soll.

1. Computerprogrammprodukt zum Durchführen von

Header-Dekomprimierung des Dekomprimierens eines komprimierten Headers eines Pakets zur Übertragung durch Bezugnehmen auf Referenzin- formation, die gleich ist wie Referenzinformation, auf die sich zur Header-Komprimierung durch eine Übertragungsseite bezogen wird, wobei das Pro- gramm Codemittel umfasst, die angepasst sind, wenn das Programm auf einem Datenverarbei- tungssystem ausgeführt wird, die Verfahrensschrit- te durchzuführen:

einen Paketempfangsschritt des Empfangens des Pakets von der Übertragungsseite;

einen Header-Dekomprimierungsschritt, ver- sehen mit dem empfangenen Paket, des Aus- führens von Header-Dekomprimierung durch Bezugnehmen auf die Referenzinformation in der gespeicherten Referenzinformation;

einen Fehlererkennungsschritt des Erkennens eines Fehlers in dem Paket, welches den de- komprimierten Header enthält; und

einen Aktualisierungsanforderungsschritt des Übertragens, an die Übertragungsseite, einer Aktualisierungsanforderung zum Anfordern von Aktualisierungsinformation zum Aktualisie- ren der Referenzinformation;

# dadurch gekennzeichnet, dass

weiter umfasst ist ein Zähl-/Speicherschritt des Zählens der Anzahl an Fehlern, die in dem Fehler- erkennungsschritt erkannt wurden, wobei der Zähl-/ Speicherschritt enthält

einen Schritt des Zählens aufeinander folgen- der Fehler des Zählens und Speicherns der Anzahl an Malen *X*, die ein Paket, das durch den Fehlerer- kennungsschritt einen Fehler erkannt hat, aufein- ander folgend erschienen ist; und

einen Schritt des Zählens aufeinander folgen- der Dekomprimierungserfolge des Zählens und Speicherns der Anzahl an Malen *Y*, die ein Paket, das keinen Fehler durch den Fehlererkennungs- schritt erkannt hat, aufeinander folgend erschienen ist, nachdem das Zählen durch den Schritt des Zäh- lens aufeinander folgender Fehler angehalten wur- de; und dadurch, dass

in dem Aktualisierungsanforderungsschritt die Ak- tualisierungsanforderung übertragen wird, wenn bestimmt wird, basierend auf Werten von *X* und *Y*, die in dem Zähl-/Speicherschritt gezählt wurden, dass die gespeicherte Referenzinformation aktua- lisiert werden soll.

# Revendications

1. Appareil de décompression d'en-tête (709, 908) pour décompresser un en-tête compressé d'un pa- quet pour transmission en se référant à des infor-

mations de référence étant les mêmes que les in- formations de référence auxquelles il s'est référé pour la compression d'en-tête par un côté émission, ledit appareil (709, 908) comprenant :

*5*

un récepteur de paquet (704, 904) pour rece- voir le paquet dudit côté émission ;

un gestionnaire d'informations de référence (707, 906) pour mémoriser et gérer lesdites in- *10* formations de référence ;

un décompresseur d'en-tête (703, 903), pourvu du paquet reçu, pour effectuer la décompres-

sion d'en-tête en se référant aux informations *15* de référence mémorisées dans ledit gestion- naire d'informations de référence (707, 906) ;

un détecteur d'erreur (702, 902) pour détecter

une erreur dans le paquet incluant l'en-tête *20*

décompressée ; et

une unité de demande de mise à jour (708, 907) pour transmettre, vers ledit côté émission, une demande de mise à jour pour demander des *25* informations de mise à jour pour mettre à jour lesdites informations de référence, dans lequel

lorsque lesdites informations de mise à jour sont transmises depuis ledit côté émission, ledit *30* gestionnaire d'informations de référence (707, 906)

met à jour les informations de référence mémori- sées dans ledit gestionnaire d'informations de réfé- rence (707, 906) sur la base des informations de mise à jour transmises, *35*

# caractérisé en ce que

il comprend, en outre, un compteur/mémoire (705, 706, 905) pour compter le nombre d'erreurs détectées par ledit détecteur d'erreur (702, 902), le-

dit compteur/mémoire (705, 706) incluant *40*

un compteur d'erreurs successives (705) pour compter et mémoriser le nombre de fois *X* où un paquet ayant une erreur détectée par ledit détec- teur (702) apparaît successivement ; et

un compteur de décompression successive *45*

satisfaisante (706) pour compter et mémoriser le nombre de fois *Y* où un paquet n'ayant pas d'erreur détectée par ledit détecteur d'erreur apparaît suc- cessivement après que le comptage par ledit comp- teur d'erreurs successives (705) soit arrêté ; et **en** *50*

# ce que

ladite unité de demande de mise à jour (708, 907) transmet ladite demande de mise à jour lors de la détermination, sur la base des valeurs de *X* et

*Y* comptées par ledit compteur/mémoire (705, 706), *55*

que les informations de référence mémorisées dans ledit gestionnaire d'informations de référence (707, 906) devraient être mises jour.

1. Appareil de décompression d'en-tête selon la re- vendication 1, dans lequel lorsque *X*  *Y*, ladite unité de demande de mise à jour (708) détermine que les informations de référence mémorisées dans ledit gestionnaire d'informations de référence (707) de- vraient être mises à jour.
2. Procédé de décompression d'en-tête pour décom- presser un en-tête compressé d'un paquet pour transmission en se référant aux informations de ré- férence qui sont les mêmes que les informations de référence auxquelles il s'est référé pour la compres- sion d'en-tête par un côté émission, ledit procédé comprenant :

une étape de réception de paquet consistant à recevoir le paquet dudit côté émission ;

une étape de décompression d'en-tête, pourvu du paquet reçu, consistant à effectuer la dé- compression d'en-tête en se référant aux infor- mations de référence mémorisées dans ledit gestionnaire d'informations de référence ;

une étape de détection d'erreur consistant à détecter l'erreur dans le paquet incluant l'en-tê- te décompressée ; et

une étape de demande de mise à jour consis- tant à transmettre, audit côté émission, une de- mande de mise à jour pour demander des in- formations de mise à jour pour mettre à jour les- dites informations de référence ;

# caractérisé en ce que

il comprend, en outre, une étape de compta- ge/mémorisation consistant à compter le nombre d'erreurs détectées dans ladite étape de détection d'erreur, ladite étape de comptage/mémorisation in- cluant

une étape de comptage d'erreurs successi- ves consistant à compter et mémoriser le nombre de fois *X* où un paquet ayant une erreur détectée par ladite étape de détection d'erreur apparaît successivement ; et

une étape de comptage de décompression successive satisfaisante consistant à compter et à mémoriser le nombre de fois *Y* où un paquet n'ayant pas d'erreur détectée par l'étape de détection d'er- reur apparaît successivement après que le comp- tage par ladite étape de comptage d'erreur succes- sive soit arrêté ; et **en ce que**

dans ladite étape de demande de mise à jour, ladite demande de mise à jour est transmise lors de la détermination, sur la base des valeurs de *X* et *Y* comptées dans ladite étape de comptage/mémori- sation, que les informations de référence mémori- sées devraient être mises jour.

1. Procédé de décompression d'en-tête selon la re- vendication 3, dans lequel dans ladite étape de de- mande de mise à jour, lorsque *X*  *Y,* il est déterminé que les informations de référence mémorisées de- vraient être mises à jour. *5*
2. Support d'enregistrement lisible par ordinateur avec un programme enregistré dans celui-ci, le pro- gramme étant exécuté dans un système d'ordina- teur pour effectuer la décompression d'en-tête de *10*

la décompression d'un en-tête compressé d'un pa- quet pour transmission en se référant aux informa- tions de référence qui sont les mêmes que les in- formations de référence auxquelles il s'est référé pour la compression d'en-tête par un côté émission, *15*

ledit programme comprenant :

une étape de réception de paquet consistant à recevoir le paquet dudit côté émission ;

*20*

une étape de décompression d'en-tête, pourvu du paquet reçu, consistant à effectuer la dé- compression d'en-tête en se référant aux infor- mations de référence mémorisées dans ledit

gestionnaire d'informations de référence ; *25*

une étape de détection d'erreur consistant à détecter une erreur dans le paquet incluant l'en-tête décompressée ; et

*30*

une étape de demande de mise à jour consis- tant à transmettre, vers ledit côté émission, une demande de mise à jour pour demander des informations de mise à jour pour mettre à jour

lesdites informations de référence ; *35*

# caractérisé en ce que

il comprend, en outre, une étape de compta- ge/mémorisation consistant à compter le nombre d'erreurs détectées dans ladite étape de détection *40*

d'erreur, ladite étape de comptage/mémorisation in- cluant

une étape de comptage d'erreurs successi- ves consistant à compter et mémoriser le nombre

de fois *X* où un paquet ayant une erreur détectée *45*

par ladite étape de détection d'erreur apparaît successivement ; et

une étape de comptage de décompression successive satisfaisante consistant à compter et à mémoriser le nombre de fois *Y* où un paquet n'ayant *50*

pas d'erreur détectée par ladite étape de détection d'erreur apparaît successivement après que le comptage par ladite étape de comptage d'erreur successive soit arrêté ; et **en ce que**

dans ladite étape de demande de mise à jour, *55*

ladite demande de mise à jour est transmise lors de la détermination, sur la base des valeurs de *X* et *Y* comptées dans ladite étape de comptage/mémori-

sation, que les informations de référence mémori- sées devraient être mises jour.

1. Produit de programme d'ordinateur pour effectuer la décompression d'en-tête consistant à décom- presser un en-tête compressé d'un paquet pour transmission en se référant à des informations de référence qui sont les mêmes que les informations de référence auxquelles il s'est référé pour la com- pression d'en-tête par un côté émission, ledit pro- gramme comprenant un moyen de code conçu pour effectuer, lorsque ledit programme est exécuté sur un système de traitement de données, les étapes du procédé de :

une étape de réception de paquet consistant à recevoir le paquet dudit côté émission ;

une étape de décompression d'en-tête, pourvu du paquet reçu, consistant à effectuer la dé- compression d'en-tête en se référant aux infor- mations de référence mémorisées dans ledit gestionnaire d'informations de référence ;

une étape de détection d'erreur consistant à détecter une erreur dans le paquet incluant l'en-tête décompressée ; et

une étape de demande de mise à jour consis- tant à transmettre, vers ledit côté émission, une demande de mise à jour pour demander les in- formations de mise à jour pour mettre à jour les- dites informations de référence ;

# caractérisé en ce que

il comprend, en outre, une étape de compta- ge/mémorisation consistant à compter le nombre d'erreurs détectées dans ladite étape de détection d'erreur, ladite étape de comptage/mémorisation in- cluant

une étape de comptage d'erreurs successi- ves consistant à compter et mémoriser le nombre de fois *X* où un paquet ayant une erreur détectée par ladite étape de détection d'erreur apparaît successivement ; et

une étape de comptage de décompression successive satisfaisante consistant à compter et à mémoriser le nombre de fois *Y* où un paquet n'ayant pas une erreur détectée par ladite étape de détec- tion d'erreur apparaît successivement après que le comptage par ladite étape de comptage d'erreur successive soit arrêté ; et **en ce que**

dans ladite étape de demande de mise à jour, ladite demande de mise à jour est transmise lors de la détermination, sur la base des valeurs de *X* et *Y* comptées dans ladite étape de comptage/mémori- sation, que les informations de référence mémori- sées devraient être mises jour.



















