

(19) World Intellectual Property Organization
International Bureau



(43) International Publication Date
28 December 2000 (28.12.2000)

PCT

(10) International Publication Number
WO 00/79764 A1

- (51) International Patent Classification⁷: **H04L 29/06**
- (21) International Application Number: PCT/SE00/01270
- (22) International Filing Date: 16 June 2000 (16.06.2000)
- (25) Filing Language: English
- (26) Publication Language: English
- (30) Priority Data:
09/335,557 18 June 1999 (18.06.1999) US
- (71) Applicant: **TELEFONAKTIEBOLAGET L M ERICSSON (publ)** [SE/SE]; S-126 25 Stockholm (SE).
- (72) Inventors: **SVANBRO, Krister**; Mjölkkudsvägen 133, 1 tr, S-973 43 Luleå (SE). **JONSSON, Lars-Erik**; Docentvägen 28, S-977 52 Luleå (SE). **DEGERMARK, Mikael**; Skeppsbrogatan 40, S-972 31 Luleå (SE). **HANNU, Hans**; Professorsvägen 274, S-977 51 Luleå (SE).
- (74) Agents: **MILDH, Christer** et al.; Ericsson Radio Systems AB, Ericsson Research/Patent Support Unit, S-164 80 Stockholm (SE).
- (81) Designated States (*national*): AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, CA, CH, CN, CR, CU, CZ, DE, DK, DM, DZ, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, TZ, UA, UG, UZ, VN, YU, ZA, ZW.
- (84) Designated States (*regional*): ARIPO patent (GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GW, ML, MR, NE, SN, TD, TG).

Published:

— With international search report.

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.



WO 00/79764 A1

(54) Title: **ROBUST DELTA ENCODING WITH HISTORY INFORMATION**

(57) Abstract: In the compression of header field values (12) to produce compressed header portions (17) of associated data packets (18) to be transmitted across a communication channel, there is included in each transmitted packet history information about delta values (ID) of a certain number of previous packets (18) in the transmission sequence. This makes the compression scheme more robust and tolerant of packet loss, because lost delta values (ID) can be reconstructed using the history information.

ROBUST DELTA ENCODING WITH HISTORY INFORMATION

FIELD OF THE INVENTION

The invention relates generally to packet communications and, more particularly, to header compression in packet communications.

BACKGROUND OF THE INVENTION

Due to the tremendous success of the Internet, it has become a challenging task to make use of the Internet Protocol IP (see, e.g., Jon Postel, *Internet Protocol*, DARPA RFC 791, September 1981, incorporated herein by reference) over all kind of links. However, because the IP protocols were designed for wired links with high bandwidth capabilities, and because packet headers of the IP protocols are rather large, it is not always a simple task to use IP protocols with narrow band links, for example cellular links. If we consider the scenario when the IP protocols are used for real-time data, for example ordinary speech, the User Datagram Protocol UDP (see, e.g., Jon Postel, *User Datagram Protocol*, DARPA RFC 768, August 1980, incorporated herein by reference) and the Real-Time Transport Protocol RTP (see, e.g., Henning Schulzrinne, Stephen L. Casner, Ron Frederick and Van Jacobson, *RTP: A Transport Protocol for Real-Time Applications*, IETF RFC 1889, IETF Audio/Video Transport Working Group, January 1996, incorporated herein by reference) are applied on top of IP. Together they require a total amount of 40 header octets (IP 20, UDP 8 and RTP 12 octets). If we combine these header requirements with ordinary speech usage, which may have frame sizes as low as 15-20 octets, the header part will disadvantageously represent more than 70% of the packet. With the upcoming new IP version 6 (see, e.g., Steven Deering and Robert Hinden, *Internet Protocol Version 6 (Ipv6) Specification*, RFC 2460, IETF Network Working Group, December 1998, incorporated herein by reference), which has a header of 40 bytes, this problem will increase. Reducing the header sizes would improve the spectrum efficiency and save a lot of money when transmitting over wireless links.

-2-

The term header compression (HC) comprises the art of minimizing the necessary bandwidth for information carried in headers on a per-hop basis over point-to-point links. Header compression takes advantage of the fact that some fields in the headers are not changing within a flow, and that most header changes are small and/or predictable. Conventional header compression schemes make use of these facts and send static information only initially, while changing fields are sent either as uncompressed values (e.g., for completely random information) or as differences (or deltas) from packet to packet, the latter typically referred to as difference (or delta) encoding. When difference encoding is used, the compression scheme can be fragile, with its performance very dependent on link quality. For example, if packet loss is common on the link, quality suffers because many consecutive packets are typically lost each time a loss occurs.

Conventional header compression/decompression schemes are often realized using state machines, and the challenging task is to keep the compressor and decompressor states, (or contexts), consistent with each other.

In general, there are two different conventional techniques to keep the decompressor context updated. The first technique uses periodic refreshes wherein absolute header data is sent. An advantage of this solution is that its performance is not affected by the round-trip-time (RTT) of the link, due to the fact that no messages are sent from the de-compressor to the compressor. This means that it also works over simplex links. On the other hand, there are a number of disadvantages with periodic refreshing. For example, the average header overhead will be high due to the high number of large refresh headers, most of which are unnecessary. Also the number of lost packets will be high if errors on the link are common.

The other common way of keeping the context updated is to let the compressor send refreshing information (i.e., absolute header data) only when requested by the decompressor. This requires a duplex link but reduces the average header overhead because no unnecessary updates are performed. Provided that the RTT is small, this solution also reduces the number of lost packets due to inconsistent context states after a link error. The obvious disadvantages are dependence on the back channel of the duplex link, sensitivity to lost packets on the link, and the high number of consecutive

-3-

lost packets that will occur in case of an invalid context (and associated refresh request) when the RTT is high.

For all header compression schemes, two measures describe their performance. Compression efficiency describes how much the headers are compressed. This can be expressed by the average or maximal header size, combinations of both, or in other ways. Robustness describes how well the scheme handles loss on the link. Will loss of a packet make the header contexts inconsistent resulting in a large number of subsequent lost packets?

Normally, most conventional header compression schemes perform well, but they require links with low error rates and small RTT's.

Currently, there exist a number of different conventional header compression schemes. In fact, they are not really different schemes but different development states of the same one. The earliest proposals (see, e.g., Van Jacobson, *Compressing TCP/IP Headers for Low-Speed Serial Links*, IETF RDC 1144, IETF Network Working Group, February 1990, incorporated herein by reference) handle only compression of TCP (see, e.g., Jon Postel, *Transmission Control Protocol*, DARPA RFC 761, January 1980, incorporated herein by reference) flows, while ideas have later evolved to make compression of UDP and also RTP headers possible (see, e.g., Mikael Degermark, Björn Nordgren and Stephen Pink, *IP Header Compression*, IETF RFC 2507, IETF Network Working Group, February 1999, incorporated herein by reference; and Steven Casner and Van Jacobson, *Compressing IP/UDP/RTP Headers for Low-Speed Serial Links*, IETF RFC 2508, IETF Network Working Group, February 1999, incorporated herein by reference). Today it could therefore be stated that for real-time data flows there is just one scheme for header compression (see Casner and Jacobson above), which is currently being standardized within the IETF by the Audio/Video-Transport working group, and which is referred to herein as CRTP.

CRTP compresses the 40 octets of RTP/UDP/IP headers down to 2 octets for most packets and, as long as the links are reliable, this minimal size will almost be equal to the average. CRTP uses difference encoding for three fields: the RTP sequence number field; the RTP time stamp field; and the ID field of the IP header.

-4-

CRTP uses update requests, as described above, to bring invalidated de-compressor contexts up to date.

A more general scheme for compression of UDP/IP headers (see Degermark, et al. above), which uses the periodic refreshing principle, may also be used, but the RTP headers are then sent uncompressed, resulting in 12 extra header octets in each packet.

CRTP performs well as long as the used link has a low bit-error rate and/or the RTT is small. However, this is often not the case for wireless links. The RTT is generally also of a magnitude that results in a large number of consecutive lost packets before the de-compressor receives a context update. This is in general undesirable for applications such as real-time audio and video. The overall packet-loss-rate will therefore also be too high and it is not considered possible to improve the wireless link characteristics to make the result better. Both reduction of the bit-error rate (BER) and the RTT would be too expensive. Thus, the robustness of CRTP is identified to be its weakness.

The present invention provides a principle of including, in each packet sent, history information about difference (delta) values of a certain number of previous packets. By doing that, the compression scheme becomes more robust and tolerant for packet loss because the lost difference information can be reconstructed using this history information.

BRIEF DESCRIPTION OF THE DRAWINGS

Figure 1 illustrates an exemplary packet data transmitting station according to the invention.

Figure 2 illustrates an exemplary embodiment of the header compressor of Figure 1.

Figure 3 illustrates exemplary operations which can be performed by the header compressor of Figures 1 and 2.

Figure 4 illustrates an exemplary packet data receiving station according to the invention.

-5-

Figure 5 illustrates an exemplary embodiment of the header decompressor of Figure 4.

Figure 5A illustrates an alternative to the embodiment of Figure 5.

Figure 6 illustrates exemplary operations which can be performed by the header decompressor embodiments of Figures 4-5A.

DETAILED DESCRIPTION

As indicated above, loss of a packet on the link can result in an inconsistency between the respective context states of the header compressor and the header decompressor. This inconsistency would degrade the quality of the communication service, because packets that arrive during periods of context inconsistency cannot be passed to the user application. If the header compression/decompression scheme could tolerate the loss of some packets without experiencing context inconsistencies, then unacceptable degradations in quality might be avoided. According to the invention, assuming a sequence of timewise consecutive packets, Packet P-2, Packet P-1, Packet P, etc., if the header of Packet P can also carry information about the headers of preceding Packets P-1, P-2, etc., then some occurrences of lost packets on the link can be advantageously tolerated without context inconsistencies. In order to accommodate in the header of a given packet information about the headers of previous packets, the total information can be advantageously coded in order to avoid an unacceptable increase in the size of the header. Thus, as described generally above, and as described in more detail below, it is advantageous to include in the header of a given packet at least some historical information about the headers of previous packets in order to permit recovery from packet loss without context inconsistencies. The scope of the historical information included in a given header can differ from one case to another, as described below.

The following definitions are used in this description.

Header decompression (HD) refers to the process of reconstructing desired (uncompressed) header information from the compressed header information produced by a header compression (HC) process.

-6-

Loss between HC and HD (LCD) is a packet loss parameter for the link between header compression and header de-compressor. This parameter describes the maximal number of consecutive lost packets on the link that a header compression scheme can handle if the proposed encoding principles for changing fields are used.

5 Of course, this requires that no other mechanism of the scheme is more sensitive to loss.

Loss before header compression (LBC) is loss that has happened to the packet stream before it reaches the header compressor. This might be loss on the other end of the connection, for an example on an identical narrow band link using the same header compression scheme, but it could also be loss on a link in between (core network). Because of the low expectation of losing packets on such a reliable link, this loss is treated as insignificant compared to the loss in a possible narrow band link on the sending side. The reason for doing this simplification is that it then seems reasonable to set the requirement on LBC to the same value as for LCD.

10

Individual Delta (ID) represents the change of the field since previous packet. If a packet has sequence number 42 and the previous one number 40 the individual delta of packet 42 is ID=2.

15

Accumulated Delta (AD) is a sum over the ID values of the previous K packets, where K depends on how well the scheme is supposed to remember changes. As will be evident from the following description, larger values of K provide greater robustness, and therefore can accommodate larger LBCs.

20

Encoded Delta Values (ED) is an encoding of the two parameters ID and AD so that they together can be represented with one parameter.

The aforementioned ID for a given header field of the Pth packet of a sequence of timewise consecutive packets, Packet P-2, Packet P-1, Packet P, etc., is given by Equation 1 below.

25

$$ID_p = S_p - S_{p-1} \quad (1)$$

Thus, the Individual Delta, ID_p for the Pth Packet of the sequence is given by the difference between the field (the sequence number field S_p in the example of Equation 1) of Packet P and the corresponding field of immediately preceding Packet

30

-7-

P-1. The sequence number fields of Packets P and P-1 are designated in Equation 1 as S_p and S_{p-1} , respectively.

The Accumulated Delta value described above can be defined for Packet P as shown in Equation 2 below.

$$AD_p = \sum_{X=1}^K ID_{p-X}, (K \geq 2) \quad (2)$$

5 Thus, the Accumulated Delta value AD_p represents the sum of the respective individual deltas of a selected number (K) of packets which were transmitted prior to Packet P in the transmission sequence.

As shown in Equation 3 below, the Individual Delta and the Accumulated Delta values, ID_p and AD_p of Packet P can be encoded using an encoding function f to produce an Encoded Delta value, ED_p :

$$ED_p = f(ID_p, AD_p) \quad (3)$$

This Encoded Delta value ED_p is then transmitted as part of the compressed header. At the receiving end, the inverse of f , namely f^{-1} is applied to the Encoded Delta value to recover the Individual Delta value and the Accumulated Delta value as shown in

15 Equation 4 below.

$$(ID_p, AD_p) = f^{-1}(ED_p) \quad (4)$$

The present invention utilizes Equations 1 and 2 above to successfully maintain a valid decompressor context in situations when a number of consecutive packets are lost on the communication channel. For example, if K is set equal to two in

20 Equation 2 above, then a loss of two consecutive packets can be handled at the receiving end. If K is set equal to two in Equation 2 above, then the Accumulated Delta value for Packet P is given by Equation 5 below.

$$AD_p = ID_{p-1} + ID_{p-2} \quad (5)$$

Equations 6, 7 and 8 below respectively represent first, second and third guesses that can be performed by a header decompression scheme according to the present

25 invention.

$$S_p = S_{LAST} + ID_p \quad (6)$$

-8-

$$S_P = S_{LAST} + ID_P + AD_P - ID_{LAST} \quad (7)$$

$$S_P = S_{LAST} + ID_P + AD_P \quad (8)$$

In each of Equations 6-8 above, the first term S_{LAST} represents the field value (in this example the sequence number field value) of the packet received at the receiver immediately before (i.e., the last packet received before) Packet P, and ID_{LAST} in Equation 7 represents the corresponding ID value. Thus, as seen from Equation 6, if S_{LAST} is S_{P-1} (no packet loss), then Equation 6 can be expected to yield the correct value of S_P , as shown by Equation 1.

However, if S_{LAST} is not S_{P-1} , then Equation 6 will not yield the correct value of S_P , so a second attempt to guess S_P can be made using Equation 7. Equation 7 can be expected to yield the correct value of S_P if S_{LAST} is S_{P-2} (Packet P-1 lost). Otherwise Equation 7 will yield an incorrect value, whereupon Equation 8 can be used to guess S_P . If S_{LAST} is S_{P-3} (Packets P-1 and P-2 lost), then Equation 8 can be expected to yield a correct value of S_P . Otherwise, Equation 8 will also fail to yield the correct value. It can be seen from the foregoing discussion that, as long as no more than two consecutive packets have been lost before arrival of Packet P, one of the three successive guesses corresponding to Equations 6-8 can be expected to identify the correct value S_P .

Returning to Equation 7, and as mentioned above, this equation can be expected to provide the correct value if only one packet has been lost. In such case, the last packet arrival would be Packet P-2, so S_{LAST} and ID_{LAST} are S_{P-2} and ID_{P-2} , respectively. Substituting S_{P-2} , ID_{P-2} , and AD_P (from Equation 5) into Equation 7, yields Equation 7A below.

$$S_P = S_{P-2} + ID_P + ID_{P-1} \quad (7A)$$

Recognizing that S_{P-2} can be expressed as shown in Equation 7B below

$$S_{P-2} = S_{P-1} - ID_{P-1} \quad (7B)$$

and substituting this expression for S_{P-2} in Equation 7A, we have

$$S_P = S_{P-1} + ID_P \quad (7C)$$

Comparison of Equation 1 with Equation 7C shows that Equation 7 can be expected to yield the correct result if the last received packet was Packet P-2.

Returning to Equation 8, and as mentioned above, this equation can be expected to provide the correct value of S_p if two consecutive packets have been lost. In such case, the last received packet would be Packet P-3, so S_{LAST} would be S_{P-3} . Substituting S_{P-3} , and AD_p (from Equation 5), into Equation 8 yields Equation 8A below.

$$S_p = S_{P-3} + ID_p + ID_{P-1} + ID_{P-2} \quad (8A)$$

Recognizing that S_{P-3} can be expressed as shown in Equation 8B below.

$$S_{P-3} = S_{P-2} - ID_{P-2} \quad (8B)$$

And recalling that S_{P-2} can be expressed as shown in Equation 7B above, then S_{P-3} can be expressed as Equation 8C below.

$$S_{P-3} = S_{P-1} - ID_{P-1} - ID_{P-2} \quad (8C)$$

Substituting the Equation 8C expression for S_{P-3} into Equation 8A yields Equation 8D below.

$$S_p = S_{P-1} + ID_p \quad (8D)$$

Comparison of Equation 1 above with Equation 8D shows that Equation 8 can be expected to yield the correct result if the last received packet was Packet P-3.

Equation 9 below corresponds to Equation 2 above with $K=3$.

$$AD_p = ID_{P-1} + ID_{P-2} + ID_{P-3} \quad (9)$$

Using this formulation of the Accumulated Delta, Equations 10-13 below can be used to successfully maintain a valid decompressor context when losses of up to three consecutive packets occur.

$$S_p = S_{LAST} + ID_p \quad (10)$$

$$S_p = S_{LAST} + ID_p + AD_p - ID_{LAST} - ID_{NEXTLAST} \quad (11)$$

$$S_p = S_{LAST} + ID_p + AD_p - ID_{LAST} \quad (12)$$

$$S_p = S_{LAST} + ID_p + AD_p \quad (13)$$

More specifically, Equation 10 assumes that Packet P-1 was received immediately before Packet P (no packet loss), Equation 11 assumes that Packet P-2 was received immediately before Packet P (one lost packet), Equation 12 assumes that Packet P-3 was received immediately before Packet P (two lost packets), and Equation 13 assumes that Packet P-4 was received immediately before Packet P (three lost packets). Thus, Equation 10 can be expected to provide the correct value S_p if S_{LAST}

-10-

is S_{P-1} , and Equations 11, 12 and 13 can be expected to yield the correct value of S_p if S_{LAST} is S_{P-2} , S_{P-3} , and S_{P-4} , respectively.

Note that $ID_{NEXTLAST}$ in Equation 11 represents the Individual Delta value of the packet received immediately before the packet received immediately before Packet P, that is, the next-to-last received packet.

In generally the same manner described above with respect to Equations 6-8, Equations 10, 11, 12 and 13 respectively represent first, second, third and fourth guess attempts which can be made by an exemplary header decompression scheme according to the invention, using the appropriate values associated with the packets received last and next-to-last before Packet P.

There is one rare scenario in which the guess of Equation 11 will fail to produce the correct value of S_p in the case of a loss of one packet, even though Equation 11 is normally expected to provide the correct value in such case. The scenario is as follows: Packet P-4 is received; Packet P-3 is lost; Packet P-2 is received; Packet P-1 is lost; and Packet P is received. Equation 11 fails in this situation because, $ID_{NEXTLAST}$ will in fact be ID_{P-4} , not ID_{P-3} , because Packet P-3 was not in fact received, contrary to the assumption of Equation 11 that only Packet P-2 was lost, and that Packet P-3 was received.

This situation can be handled by making a fifth guess after the four guesses corresponding to Equations 10-13 have failed. This fifth guess basically treats the above-described scenario as if Packet P-2 was not received, thus handling the situation as if there were three lost packets, namely Packets P-1, P-2 and P-3. In this fifth guess, Equation 13 can be used again but, for this guess, S_{P-4} (which would correspond to the last-received packet if Packets P-1, P-2 and P-3 were indeed all lost) is inserted for S_{LAST} .

Figure 1 illustrates an exemplary embodiment of a packet data transmitting station according to the invention which can perform the exemplary header compression operations described above. A conventional communications application 11 provides header information 12 and payload information 13. The payload information can be handled in conventional fashion by a payload processor 15, which outputs a payload 16. The header information is applied to a header compressor 14

-11-

which compresses the header information to produce a compressed header 17. The payload 16 and the compressed header 17 comprise a packet 18. A conventional transmitter 19 can receive the packet 18, and using well known techniques, transmit the packet across a radio communication link such as a cellular communication link.

5 The packet data transmitting station of Figure 1 can be, for example, either a fixed-site or mobile radio transmitting station operating in a cellular communication network.

Figure 2 illustrates an exemplary portion of the header compressor of Figure 1. Header information corresponding to a desired field, in this example the sequence value S_p described above, is input to a delta encoder 21 and a checksum generator 22.

10 The delta encoder performs conventional delta encoding generally according to Equation 1 above to produce the Individual Delta value ID_p corresponding to S_p of Packet P. This Individual Delta value is input to a buffer 24 which maintains a record of the Individual Delta values of the previous K packets. A summing apparatus 25 is coupled to the buffer to receive the Individual Deltas, and is also provided with the value of K, in order to sum desired ones of the previous Individual Deltas according to Equation 2 to produce the Accumulated Delta AD_p for Packet P. An encoder 26 receives the Individual Delta ID_p and Accumulated Delta AD_p of Packet P, and encodes these into an Encoded Delta ED_p for Packet P.

15

The checksum generator 22 uses the sequence number value S_p to generate a checksum (for example, a CRC checksum), namely checksum P as shown in Figure 2. Checksum P at 29 is combined with the Encoded Delta value ED_p at 20 to form a compressed header field 28 representing the sequence number S_p . This compressed header field 28 can be included in a compressed header such as shown at 17 of Figure 1.

20

25 Although Figure 2 illustrates compression of a single header field, it will be understood that other desired header fields can also be compressed using the techniques of Figure 2.

The encoder 26 of Figure 2 maps the Individual Delta ID_p and the Accumulated Delta AD_p into the combined Encoded Delta ED_p . In one example, a unique code value can be assigned for each possible combination of values of the Individual Delta and the Accumulated Delta, which values can be determined, for example, by

30

-12-

empirical observations. The encoder 26 can thus be implemented as, for example, a look-up table having plural codes indexed against ID_p/AD_p combinations. In some embodiments, the most common values of the Encoded Delta ED_p could be identified, and these most common values could be encoded using a relatively small amount of bits while the other less common values of ED_p could be encoded using relatively more bits.

Figure 3 illustrates exemplary operations which can be performed by the exemplary header compressor embodiment of Figure 2. The header field information is received at 31, and the checksum is generated therefrom at 32. The Individual Delta is calculated at 33, and the Accumulated Delta is calculated at 34. At 35, the Encoded Delta is produced in response to the Individual and Accumulated Deltas. At 36, the Encoded Delta is combined with the checksum to form the compressed header field.

Figure 4 illustrates an exemplary packet data receiving station according to the invention. A conventional receiver 46 can use well known techniques to receive from a radio communication link, such as a cellular communication link, a received version 18' of a transmitted packet such as Packet 18 of Figure 1. The received version 18' includes a payload version 16' and a received compressed header version 17'. The received payload version 16' is input to a payload processor 45 which can use conventional techniques to produce at 43 received payload information for input to a conventional packet data communications application 41. The received compressed header version 17' is applied to a header decompressor 44, which decompresses the received header version and provides at 42 received header information for the communications application 41.

Figure 5 illustrates an exemplary portion of the header decompressor of Figure 4. In Figure 5, a received version 20' of an Encoded Delta ED_p such as shown at 20 in Figure 2 is applied to a decoder 51 which can use Equation 4 to produce the Individual Delta and Accumulated Delta corresponding to Packet P. The decoder, which can be implemented, for example, with a look-up table having ID_p/AD_p combinations indexed against ED_p values, outputs the Accumulated Delta AD_p and the Individual Delta ID_p to a reconstructor 53 which can, in example embodiments, make the header field guess attempts associated with Equations 6-8 above or

-13-

Equations 10-13 above, depending on the value of K provided thereto. The current guess of S_p , namely S_p' is output from the reconstructor at 55, and is applied to a checksum generator 56.

5 The checksum generator generates a checksum from the current guess S_p' , and this checksum is input at 57 to a comparator 58 which compares the generated checksum 57 to received version 29' of an original checksum such as shown at 29 in Figure 2. If the compared checksums match, then the guess is considered to be correct, and the comparator activates an output 500 which operates a connection unit 59 in order to provide the guess S_p' from the reconstructor as received header information
10 to the communications application 41. This correct guess is also fed back to the reconstructor 53 for storage as S_{LAST} for use by the reconstructor in the next sequence of guess attempts using Equations 6-8 or 10-13. Upon receipt of the new S_{LAST} , the reconstructor can shift ID_p into a two-stage shift register 52, thereby retaining a running record of ID_{LAST} and $ID_{NEXTLAST}$ for use in Equations 6-8 or 10-13.

15 If the generated checksum does not match the received checksum in the comparator, then the comparator output 500 notifies the reconstructor to make another attempt, for example try Equation 7 after having unsuccessfully tried Equation 6. If none of Equations 6-8 (or Equations 10-13 in another embodiment) result in a checksum match at the comparator 58, then the reconstructor 53 can output a fail
20 signal to the communications application 41, thereby indicating that the header field cannot be accurately reconstructed.

Although Figure 5 illustrates decompression of a single header field, it will be understood that other desired header fields can also be decompressed using the techniques of Figure 5.

25 In order to implement the above-described special scenario guess wherein Equation 13 is re-used with S_{p-4} (which corresponds to the next-to-last received packet) inserted as S_{LAST} instead of S_{p-2} (which actually corresponds to the last received packet), the reconstructor of Figure 5 will need to maintain a running record of the S values of the last two received packets, S_{LAST} and $S_{NEXTLAST}$. This can be done, for
30 example, by maintaining a two-stage shift register similar to the register 52 of Figure 5. An example 52A of the $S_{LAST}/S_{NEXTLAST}$ register is shown in Figure 5A. The

-14-

above-described re-use of Equation 13 for a fifth guess can thus be described by Equation 14 below:

$$S_p = S_{\text{NEXTLAST}} + ID_p + AD_p \quad (14)$$

5 Figure 6 illustrates exemplary operations which can be performed by the exemplary header decompressor portions of Figures 5 and 5A. At 61 the received version of the compressed header field arrives. At 62, the Encoded Delta is decoded to produce the Individual Delta and the Accumulated Delta. At 63, the reconstructor uses, for example, one of Equations 6-8 or 10-14 to make a reconstruction attempt or guess. At 64, a checksum is generated from the reconstruction guess. If the generated
10 checksum equals the received checksum at 65, then at 66 the reconstruction guess is accepted as the header field value. If the generated checksum does not match the received checksum at 65, then the reconstructor makes another reconstruction guess at 63 (using another of Equations 6-8 or 10-14), unless it is determined at 67 that all attempts (i.e., all equations) have been exhausted. If so, then a failure indication is
15 provided at 68.

It will be evident to workers in the art that the exemplary embodiments described above can be readily implemented by suitable modifications in software, hardware or both in header compressors and decompressors of conventional packet data transmitting and receiving stations.

20 It will also be evident that the above-described invention is applicable to packet communications over any lossy, narrow bandwidth link, including real-time communications applications such as, for example, real-time audio and video applications.

25 Although exemplary embodiments of the present invention have been described above in detail, this does not limit the scope of the invention, which can be practiced in a variety of embodiments.

WHAT IS CLAIMED IS:

1. A method of compressing a current header field value to produce a compressed header portion of an associated current data packet to be transmitted across a communication channel, comprising:

5 determining a first difference between the current header field value and a corresponding header field value associated with a previous packet that precedes the current packet in a sequence of packets provided for transmission across the communication channel; and

10 providing in the compressed header portion information indicative of the first difference and further indicative of a second difference between header field values which correspond to the current header field value and are respectively associated with first and second previous packets which precede the current packet in the sequence.

15 2. The method of Claim 1, wherein said providing step includes encoding the first and second differences to produce an encoded representation of the first and second differences.

3. The method of Claim 2, wherein said encoding step includes using a look-up table to determine the encoded representation in response to the first and second differences.

20 4. The method of Claim 1, wherein said providing step includes determining a third difference between header field values which correspond to the current header field value and are respectively associated with the first previous packet and a third previous packet that precedes the current packet in the sequence, and determining a fourth difference between header field values which correspond to the current header field value and are respectively associated with the second previous
25 packet and the third previous packet, and summing the third and fourth differences to produce the second difference.

-16-

5. The method of Claim 1, including generating a checksum from the current header field value, and providing the checksum in the compressed header portion along with the information indicative of first and second differences.

5 6. The method of Claim 1, wherein the communication channel includes a wireless communication link.

7. The method of Claim 1, wherein the header field value includes a sequence number.

8. A method of decompressing a compressed header portion of a current data packet received from a communication channel to produce a desired header field value, comprising:

10 obtaining from the compressed header portion first information indicative of a first difference between the desired header field value and a corresponding header field value of a previous packet that preceded the current packet in a sequence of packets transmitted across the communication channel, and said obtaining step including obtaining from the compressed header portion second information indicative of a second difference between header field values which correspond to the desired header field value and are respectively associated with first and second previous packets which preceded the current packet in the transmission sequence; and

15 20 guessing the desired header field value based on the first and second information.

9. The method of Claim 8, wherein said guessing step includes guessing the desired header field value based on the first and second information and also based on a header field value corresponding to the desired header field value and associated with a previous packet that preceded the current packet in the transmission sequence and was received last before the current packet.

25

-17-

10. The method of Claim 8, including extracting from the compressed header portion a received version of a coded representation which represents the first and second differences and was produced at a transmitting end of the communication channel, said obtaining step including decoding the received version of the encoded representation to obtain the first and second information.

5

11. The method of Claim 10, wherein said decoding step includes using a look-up table to determine the first and second information in response to the received version of the encoded representation.

10

12. The method of Claim 8, including generating a checksum from the guess of the desired header field value, extracting from the compressed header portion a received version of a checksum generated from the desired header field value at a transmitting end of the communication channel, and comparing the generated checksum to the received checksum version to determine whether the guess is correct.

15

13. The method of Claim 8, wherein the communication channel includes a wireless communication link.

20

14. The method of Claim 8, wherein the second difference is a sum of third and fourth differences, wherein the third difference is a difference between header field values which correspond to the current header field value and are respectively associated with the first previous packet and a third previous packet that preceded the current packet in the transmission sequence, and wherein the fourth difference is a difference between header field values which correspond to the current header field value and are respectively associated with the second previous packet and the third previous packet.

25

15. The method of Claim 8, wherein the header field value includes a sequence number.

-18-

16. An apparatus for compressing a current header field value to produce a compressed header portion of an associated current data packet to be transmitted across a communication channel, comprising:

an input for receiving header field values;

5 a difference determining apparatus coupled to said input for determining a first difference between the current header field value and a corresponding header field value associated with a previous packet that precedes the current packet in a sequence of packets provided for transmission across the communication channel, and for determining a second difference between header field values which correspond to the
10 current header field value and are respectively associated with first and second previous packets which precede the current packet in the sequence; and

an output coupled to said difference determining apparatus for outputting the compressed header portion including information indicative of the first and second differences.

15 17. The apparatus of Claim 16, including an encoder coupled between said output and said difference determining apparatus for encoding the first and second differences to produce at said output and encoded representation of the first and second differences.

20 18. The apparatus of Claim 16, wherein said difference determining apparatus is operable to determine a third difference between header field values which correspond to the current header field value and are respectively associated with the first previous packet and a third previous packet that precedes the current packet in the sequence, and to determine a fourth difference between header field values which correspond to the current header field value and are respectively associated with
25 the second previous packet and the third previous packet, and said difference determining apparatus operable for summing the third and fourth differences to produce the second difference.

-19-

19. The apparatus of Claim 16, including a checksum generator coupled to said input for receiving the current header field value and generating therefrom a checksum, said checksum generator coupled to said output for providing the checksum in the compressed header portion along with said information indicative of first and second differences.

5

20. The apparatus of Claim 16, wherein the communication channel includes a wireless communication link.

21. The apparatus of Claim 16, wherein the header field value includes a sequence number.

10

22. An apparatus for decompressing a compressed header portion of a current data packet received from a communication channel to produce a desired header field value, comprising:

an input for receiving the compressed header portion;

15

a reconstructor coupled to said input for receiving first information obtained from the compressed header portion and indicative of a first difference between the desired header field value and a corresponding header field value of a previous packet that preceded the current packet in a sequence of packets transmitted across the communication channel, said reconstructor further for receiving second information obtained from the compressed header portion and indicative of a second difference between header field values which correspond to the desired header field value and are respectively associated with first and second previous packets which preceded the current packet in the transmission sequence; and

20

said reconstructor responsive to said first and second information for guessing the desired header field value.

25

23. The apparatus of Claim 22, wherein said reconstructor has an input for receiving a header field value corresponding to the desired header field value and associated with a previous packet that preceded the current packet in the transmission

-20-

sequence and was received last before the current packet, said reconstructor operable to guess the desired header field value responsive to said first and second information and said header field value received at said reconstructor input.

5 24. The apparatus of Claim 22, including a decoder coupled between said input and said reconstructor for receiving from said compressed header portion a received version of a coded representation which represents the first and second differences and was produced at a transmitting end of the communication channel, said decoder operable for decoding the received version of the encoded representation to obtain the first and second information.

10 25. The apparatus of Claim 22, wherein said reconstructor has an output for outputting a guess of the desired header field value, and further including a checksum generator coupled to said reconstructor output for receiving said guess and generating therefrom a checksum, said checksum generator including an output for outputting said checksum, and further including a comparator coupled to said input for
15 receiving from said compressed header portion a received version of a checksum that was generated from the desired header field value at a transmitting end of the communication channel, said comparator coupled to said checksum generator output for receiving therefrom said checksum, said comparator operable to compare said received checksum version to said checksum generated by said checksum generator
20 to determine if said guess is correct.

26. The apparatus of Claim 22, wherein the communication channel includes a wireless communication link.

27. The apparatus of Claim 22, wherein the header field value includes a sequence number.

FIG. 1

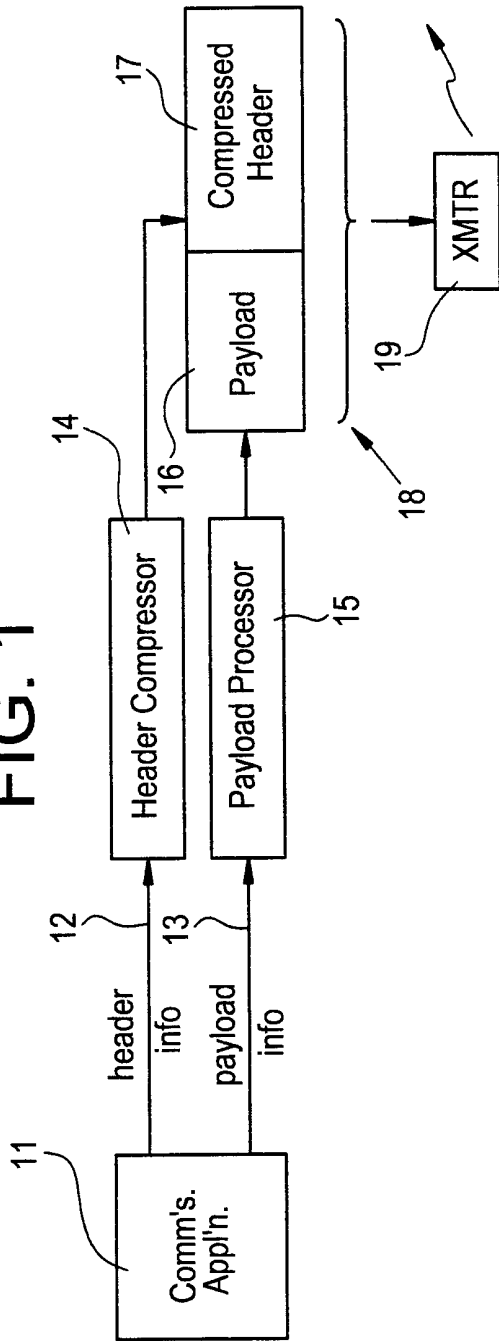


FIG. 4

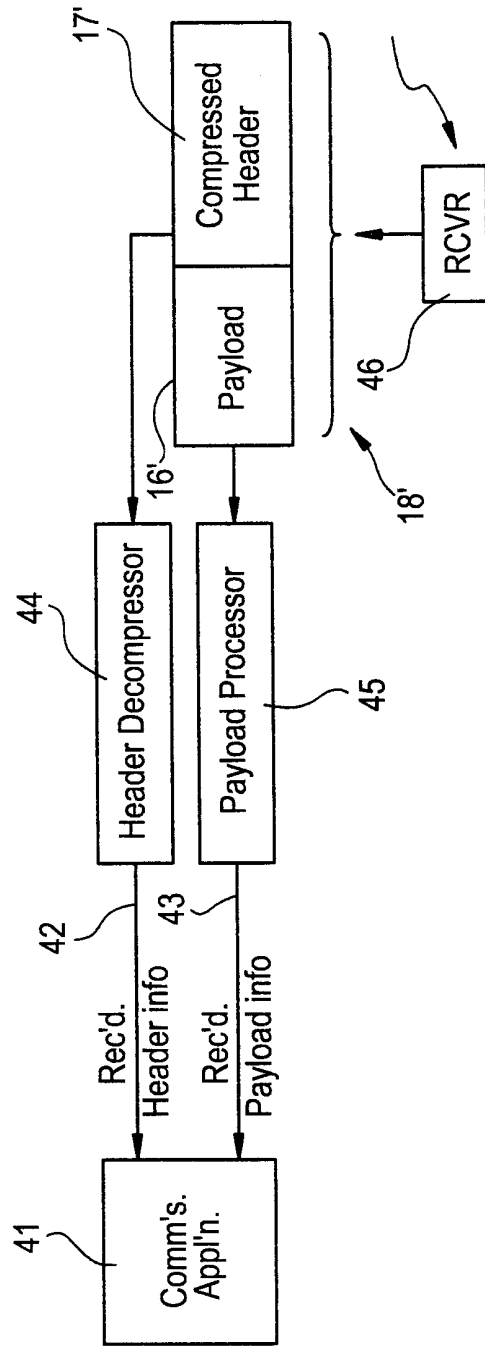


FIG. 2

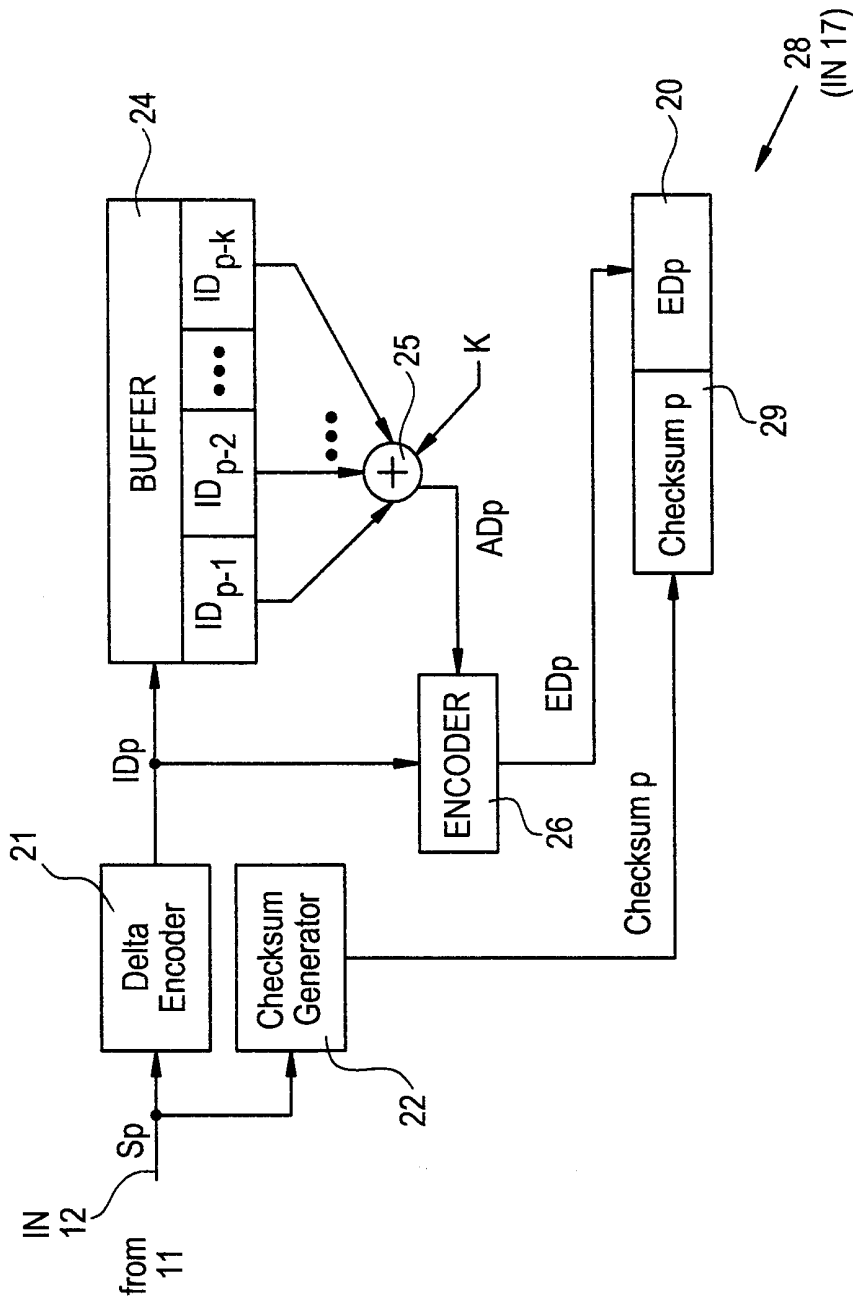


FIG. 3

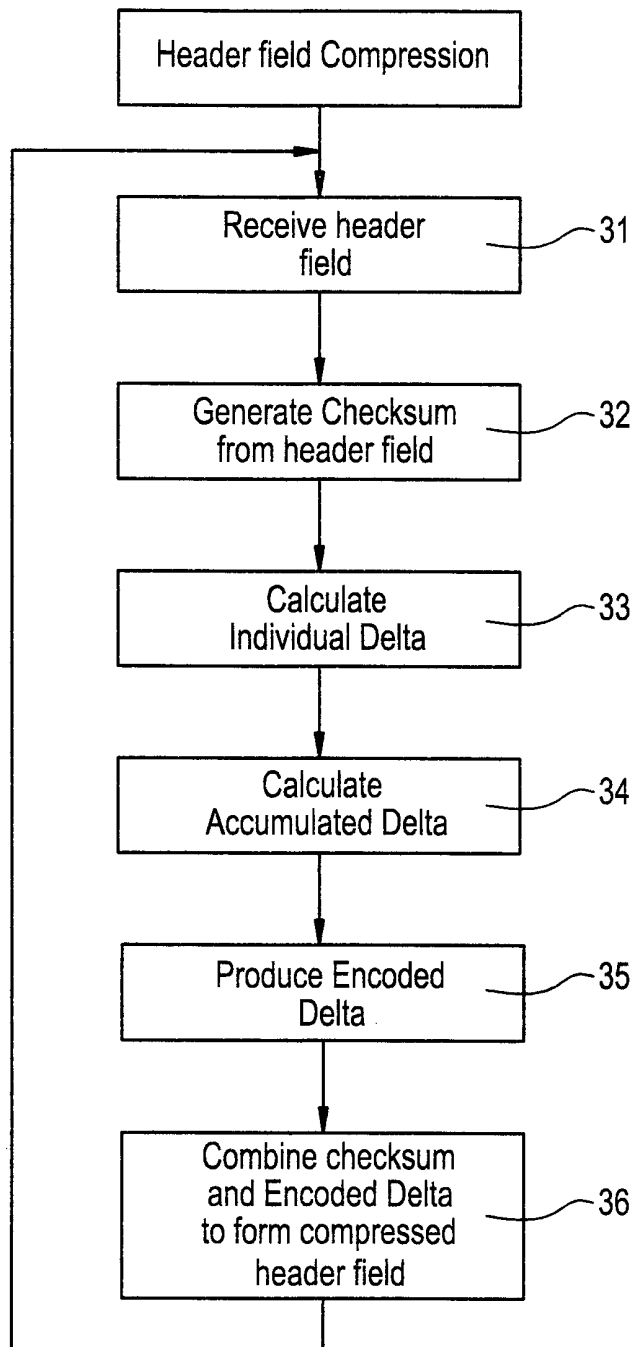


FIG. 5

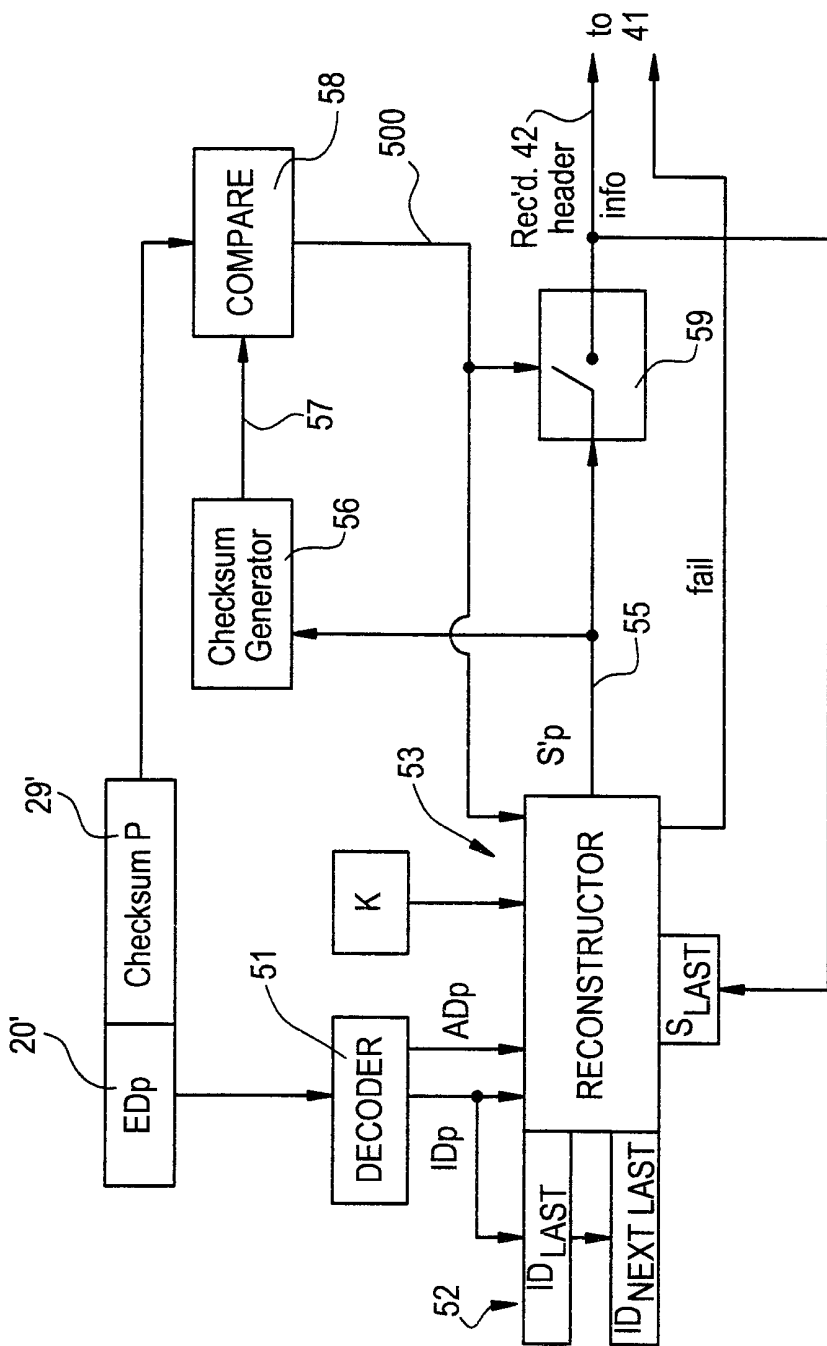


FIG. 5A

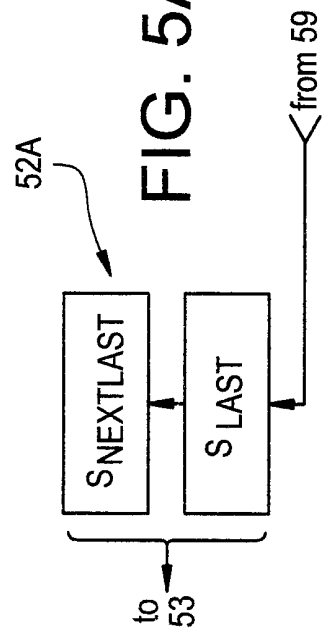
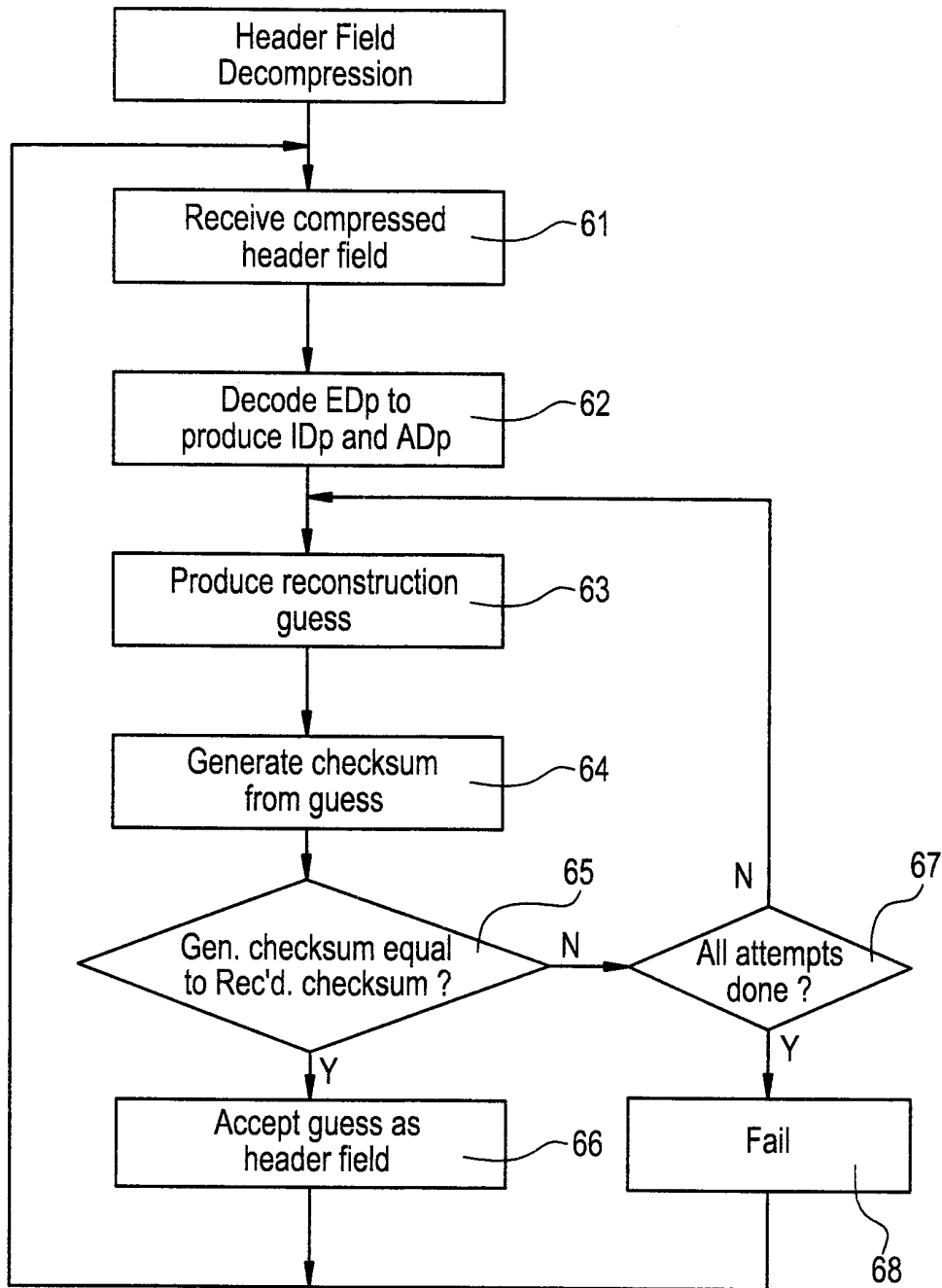


FIG. 6



INTERNATIONAL SEARCH REPORT

International application No.

PCT/SE 00/01270

A. CLASSIFICATION OF SUBJECT MATTER

IPC7: H04L 29/06

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

IPC7: H04L, H03M

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

SE,DK,FI,NO classes as above

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	RFC2507, February 1999, M. Degermark et al: "IP Header Compression", pages 1-31 see section 6; 10.1; abstract --	1-27
A	1997 IEEE International Conference on Communications, 1997. ICC'97 Montreal, Towards the Knowledge Millennium. 1997, Stephen J. Perkins et al:"Dependency removal for transport protocol header compression over noisy channels", vol. 2, pages 1025-1029, see section 4.2 --	1-27



Further documents are listed in the continuation of Box C.



See patent family annex.

* Special categories of cited documents:

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier document but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance: the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance: the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art

"&" document member of the same patent family

Date of the actual completion of the international search

4 October 2000

Date of mailing of the international search report

12 -10- 2000

Name and mailing address of the ISA/

Swedish Patent Office

Box 5055, S-102 42 STOCKHOLM

Facsimile No. +46 8 666 02 86

Authorized officer

Peder Gjervaldsaeter/AE

Telephone No. +46 8 782 25 00

INTERNATIONAL SEARCH REPORT

International application No.

PCT/SE 00/01270

C (Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT		
Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	Wireless Networks, US, ACM, vol. 3, no. 5, page 375-387 XP000728935 ISSN: 1022-0038, Degermark M. et al: "Low-Loss TCP/IP Header Compression for Wireless Networks", see section 4.1, 4.2; abstract --	1-27
A	RFC2508, February 1999, S. Casner et al: "Compressing IP/UDP/RTP Headers for Low-Speed Serial Links", pages 1-17, see section 3.1; 3.2 --	1-27
A	Third IEEE Symposium on Computers and Communications, 1998. ISCC'98. Proceedings. pages 543-548, G. Mamais et al: "Evaluation of the Casner-Jacobson Algorithm for Compressing the RTP/UDP/IP Headers", see section II -- -----	1-27