(12) UK Patent Application (19) GB (11) 2 403 877 (13) A

(43) Date of A Publication 12.01.2005

(21) Application No: 0316054.6

(22) Date of Filing: 09.07.2003

(71) Applicant(s):
Motorola Inc
(Incorporated in USA - Delaware)
Corporate Offices,
1303 East Algonquin Road, Schaumburg,
Illinois 60196, United States of America

(72) Inventor(s):
Konstantin Dolgov
Tom Risager

(74) Agent and/or Address for Service:
Optimus
Grove House, Lutyens Close,
Chineham Court, BASINGSTOKE,
Hampshire, RG24 8AG, United Kingdom

(51) INT CL[7]:
H04L 29/06

(52) UK CL (Edition X):
H4P PDCFP PPEC

(56) Documents Cited:
EP 1137237 A3 **WO 2003/030435 A1**
S. Casner et al, "Compressing IP/UDP/RTP Headers for Low-Speed Serial Links", Internet Engineering Task Force (IETF), Network Working Group, Request for Comments RFC 2508, 1999, Retrieved from the Internet: <URL: http://www.ietf.org/rfc/rfc2508.txt>
M. Degermark et al, "IP Header Compression", Internet Engineering Task Force (IETF), Network Working Group, Request for Comments RFC 2507, 1999, Retreived from the Internet: <URL: http://www.ietf.org/rfc/rfc2507.txt>

(58) Field of Search:
UK CL (Edition V) **H4P**
INT CL[7] **H04L**
Other: **Online: WPI, EPODOC, JAPIO, INSPEC, Selected Internet sites**

(54) Abstract Title: **Packet communication with header compression**

(57) The invention relates to a system for communicating data packets comprising compressed header information. An apparatus operates in a first mode which comprises the step of receiving data packets (301). Data packet sequence information is extracted (303) from the data packets and a disruption indication is determined (305) in response to the packet sequence information. If the disruption indicates that no disruption has occurred, the apparatus remains in the first mode and decompresses (309) the compressed headers and generates (311) decompressed data packets. If the disruption indicates that a disruption has occurred, the apparatus enters a second mode wherein a header compression resynchronisation process is initialised (313), while at the same time compressed headers are decompressed (315) and decompressed data packets are generated (317) without any validity check being performed. The second mode is exited (321) when the header compression resynchronisation process is terminated successfully. The invention allows for improved error recovery and is particularly suited for data packets using compressed IP/UDP/RTP headers.

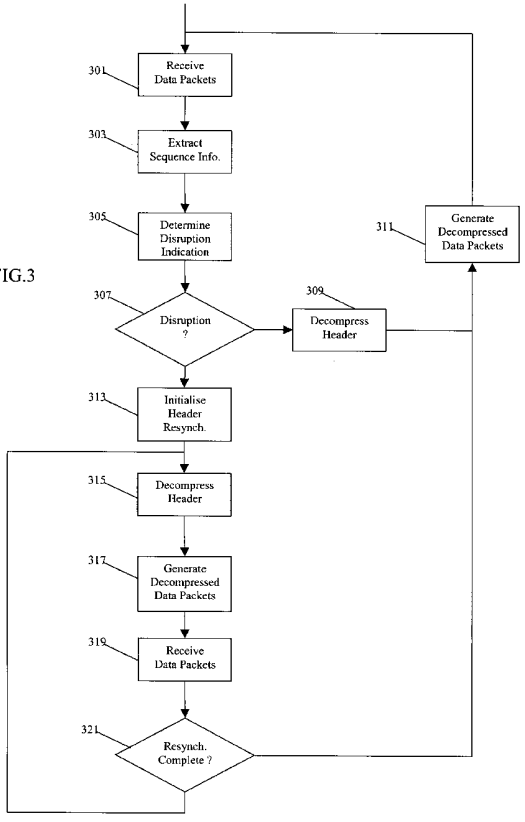


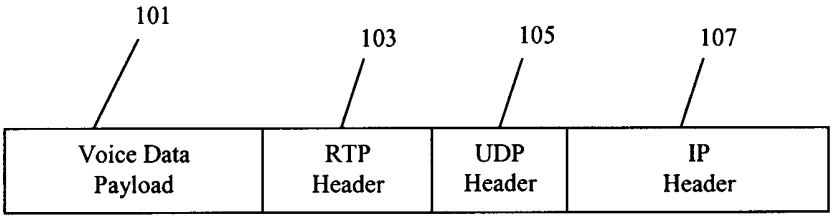
FIG.3

At least one drawing originally filed was informal and the print reproduced here is taken from a later filed formal copy.

This print takes account of replacement documents submitted after the date of filing to enable the application to comply with the formal requirements of the Patents Rules 1995

Original Printed on Recycled Paper

GB 2 403 877 A


101
103
105
107
Voice Data Payload
RTP Header
UDP Header
IP Header

100

FIG. 1

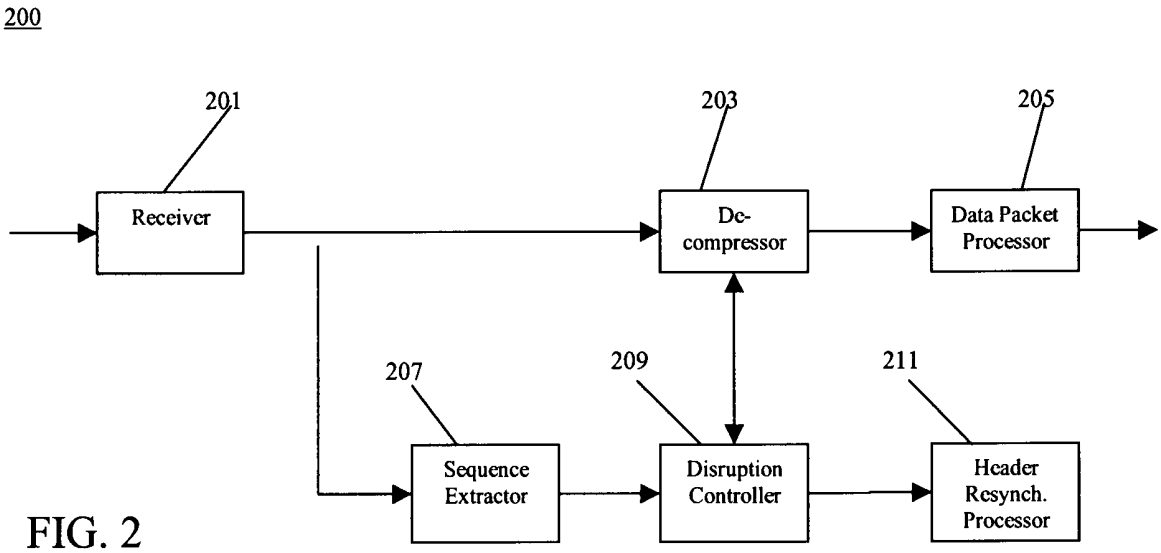
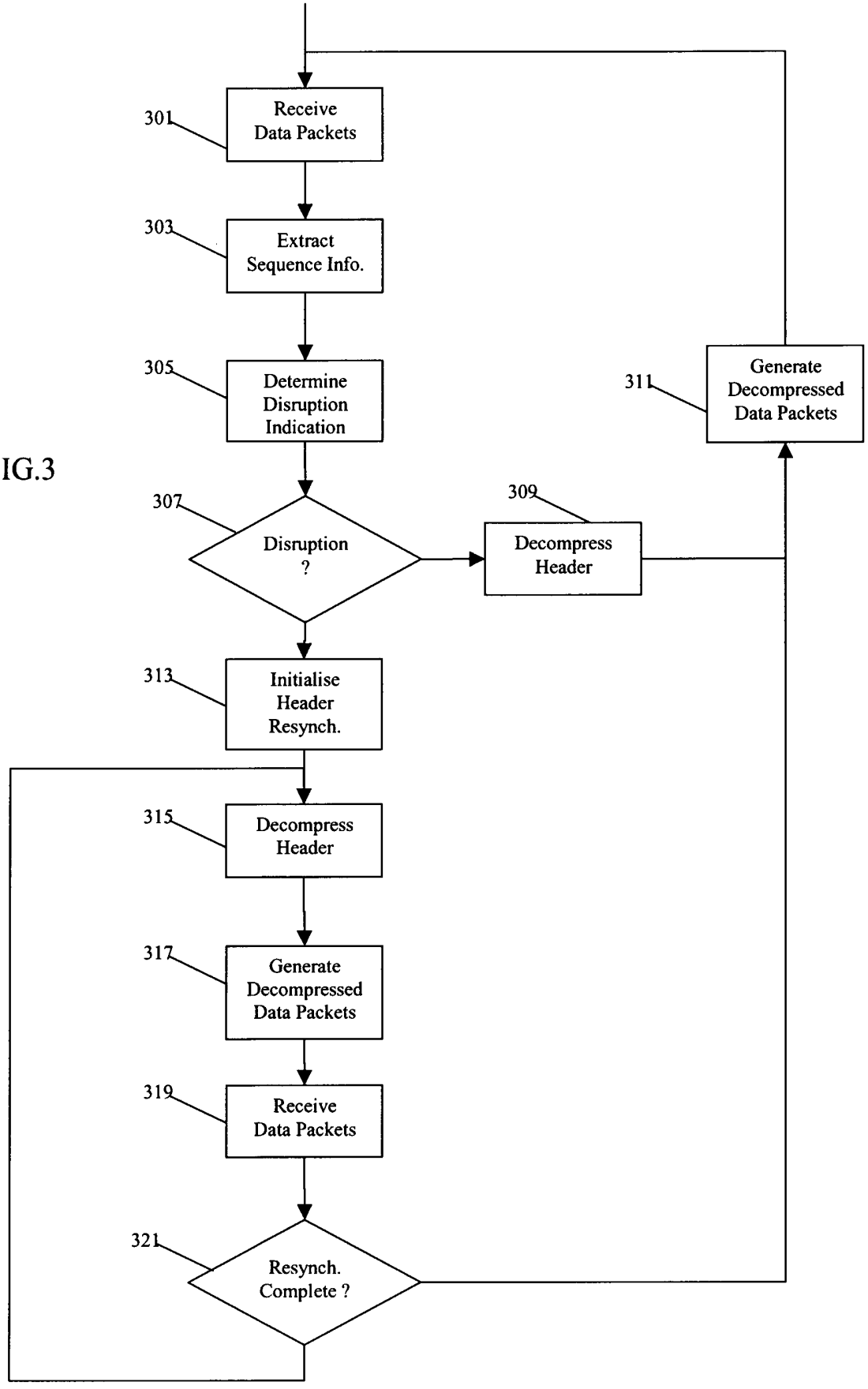

200
201
Receiver
203
De-compressor
205
Data Packet Processor
207
Sequence Extractor
209
Disruption Controller
211
Header Resynch. Processor

FIG. 2

FIG.3


Receive Data Packets
301
Extract Sequence Info.
303
Determine Disruption Indication
305
Disruption ?
307
Decompress Header
309
Generate Decompressed Data Packets
311
Initialise Header Resynch.
313
Decompress Header
315
Generate Decompressed Data Packets
317
Receive Data Packets
319
Resynch. Complete ?
321

METHOD AND APPARATUS FOR COMMUNICATING DATA PACKETS

Field of the invention

The invention relates to a method and apparatus for communicating data packets and in particular for communicating data packets of a data service which comprise compressed header information

Background of the Invention

In recent decades, the emergence of new radio communication systems and in particular cellular communication systems such as GSM or TETRA have led to a huge increase in wireless and mobile communication.

Initially, communication services provided by such communication systems were predominantly circuit switched communication services, wherein a continuous connection is set up between the parties of a call. A circuit switched communication service is well suited for many services, such as e.g. voice services, and provides advantages in terms of guaranteed delay and throughput characteristics. However, circuit switched connections tend to result in an inefficient use of the available radio resource. For example, in a two-way voice call, each party typically speaks for less than 40% of the time and the continuous connection is therefore not utilised for more than 60% of the total time. Furthermore, data calls are typically very bursty in nature. For example, Internet browsing results in a data communication wherein short periods of high data rates are interspersed by long periods of virtually no

communication. Accordingly, a circuit switched connection is idle for the majority of the duration of the call.

Therefore, conventional circuit switched services are increasingly being supplemented, enhanced or replaced by packet based services wherein data is communicated in data packets. Each data packet is typically individually routed from source to destination and no permanent connection is typically set up. Thus, the source may communicate data packets as and when it is necessary and thus the traffic flow and resource utilisation may be more flexibly adjusted to the current communication requirements.

Packet data services were initially aimed at burst data services such as Internet browsing but have in the latter years also been considered for services such as voice communication.

However, packet data services and communication systems tend to be relatively complex and involve communication and routing through many different entities and interfaces. The level of control and data exchanged between different entities is wide ranging from specific parameters associated with the physical communication medium to general application parameters. Therefore a number of protocols are used in the communication system. The exact protocol used for a specific communication link depends on which entities the communication is between and on the nature of the data communicated.

In order to achieve a structured approach, each protocol is divided into layers with each layer dealing with a specific set of parameters of the communication. Each layer provides a specific communication medium to the higher layers meeting a given set of requirements. For example, the lowest layer guarantees a medium meeting a certain error

characteristic to the next layer. However, meeting this error characteristic is achieved by functionality in the lowest layer without any interaction from the next layer. Hence, communication systems, such as TETRA, GPRS or UMTS, employ a layered network communication protocol.

In a packet based system, the packets comprise headers containing information related to the implementation of the protocols. Each layer will typically take a packet from the higher layer and add a header to this data packet thereby implementing the required communication for the layer. E.g. the lowest layer for the physical interface may add a header comprising information related to the received bit error rate. Thus, headers for the different layers are typically concatenated (specifically higher level headers may comprise lower layer headers as an embedded sub-header), resulting in the total header data which comprise the headers of all layers. This may result in a header of a very significant size.

Communicating this information is wasteful in terms of resource, and therefore it is preferable to minimise the header data communicated. Especially, the interface resources from base stations is limited and accordingly it is important to reduce the overhead incurred by the packet headers. For example, a packet data voice service may typically comprise as little as two bytes of voice data in a data packet whereas a full concatenated header may comprise several hundred bytes of data thus resulting in a payload efficiency of only one or two percent.

In order to resolve this disadvantage, many communication systems use header compression whereby only part of the header data is transmitted. The header compression depends on how much the header information changes and on how much information has already been communicated.

The header compression is implemented in a given layer of a protocol. The header is then communicated using the communication medium provided by the lower layer. This communication medium is chosen at call setup based on Quality of Service (QoS) parameters such as the required error rate, the maximum throughput delay etc.

As an example of a header compression technique, a packet source may initially start with no compression and gradually a compression index of known unchanged information may be built up. Thus the compression algorithm operates by omitting header information, which has not changed from previous packets, and as the compression index of known information is increased, more and more data can be omitted, and the size of the header can be reduced. Thus compression initially starts poor and improves until all fields that can be compressed have been optimally compressed. An example of a header compression scheme can be found in Internet Engineering Task Force (IETF) specification RFC 2508.

A problem associated with communication of data packets is that errors may occur during the communication. For example communicating data packets over a radio interface is that errors may occur which cause the data packets not to be received or to be received with errors causing the data packets to be discarded by the receiver.

Lost packets are a particular problem in association with header compression. As header compression relies on information from headers of previous data packets, the loss of packets may render it impossible to accurately decompress a subsequent header. For example, if a data packet that comprised an update to a particular header parameter has been lost, this parameter cannot be derived for the following packets. Accordingly, a lost data packet typically results in all subsequent data

packets being discarded until the full header information has been restored.

An example of an algorithm for recovering from lost data packets is described Internet Engineering Task Force (IETF) specification RFC 2507 (section 10.1) and in Internet Engineering Task Force (IETF) specification RFC 2508 (section 3.3.5).

This algorithm relies on header check sum data being icluded in the data packet. In summary, following a data packet loss, the algorithm provides for a decompression of a header of a subsequent data packet under the assumption that no updates were included in the lost data packets. Subsequently, a check sum is calculated for the decompressed header and compared to the check sum data of the data packet. If the checksums match, the data packet is accepted and the communication continues unchanged for the following data packets. However, if the check sums do not match, the data packet and all subsequent data packets are discarded. The communication is then re-started when a data packet is received comprising the full header information.

However, in many cases the loss of a plurality of data packets is significantly more detrimental than losing a single data packet. For example, in the case of highly compressed audio information, the loss of two consecutive audio samples results in a significant drop in audio quality in comparison to the loss of a single audio packet. The above described algorithm frequently results in many data packets being lost while the receiver awaits the receipt of the full header information. This results in a significant degradation in the quality of the data service and specifically in a much reduced voice quality for a voice service.

Furthermore, the described algorithm relies on the presence of checksum data in the data packet. Such checksum data increases the overhead and thus results in an increased resource usage. For example, check sum data may correspond to a data content of two bytes which is comparable to the data payload of some voice services. Thus, the algorithm is inapplicable to protocols which do not use check sum data and furthermore prevents an efficient resource utilisation in systems wherein such checksum data may be optional.

Hence, the inventors have recognised that an improved system for communicating data packets would be advantageous and in particular a system allowing for increased flexibility, resource utilisation, error insensitivity and/or improved performance and/or quality would be advantageous.

Summary of the Invention

Accordingly, the present invention seeks to mitigate, alleviate or eliminate one or more of the above mentioned disadvantages singly or in any combination.

According to a first aspect of the invention, there is provided a method of communicating data packets of a data service comprising compressed header information; the method comprising the steps of: in a first mode of operation: receiving data packets, extracting data packet sequence information from at least one of the data packets, determining a disruption indication in response to the packet sequence information, and if the disruption indicator indicates a disruption entering a second mode of operation for the data service; the second mode of operation comprising: initialising a header compression resynchronisation process, decompressing compressed header information of received data packets,

and generating decompressed data packets comprising the decompressed header information.

Specifically, the steps of decompressing compressed header information of received data packets, and generating decompressed data packets comprising the decompressed header information are preferably carried out during the header compression resynchronisation process.

The inventors of the present have realised that in many applications it is not desirable to perform a validity check following a lost data packet or to continue or discard data packets and await header information in response thereto. Rather, the inventors have realised that in many applications, improved performance can be achieved by initialising a header resynchronisation process while continuing to generate decompressed data packets based on a decompression of headers. Accordingly, data packets are not lost unnecessarily because the system is awaiting re-synchronisation. Also, a continuation based on erroneous header information is corrected when the header resynchronisation process is carried out.

Hence, the invention allows for improved performance and allows for a system wherein fewer data packets are discarded and/or wherein header synchronisation is always achieved after a data packet loss.

For example, in situations wherein data packets are lost that comprise information required for generating decompressed data packets the data packets generated in the second mode may be erroneous. However, in most applications, this is no worse than discarding the data packets. Furthermore, the header resynchronisation is performed which will cause the system to recover in the same way as a conventional validity check based approach. In situations where the lost data packet

comprises no information required to generate decompressed data packets, no data packets are discarded and the generated data packets are all correctly generated. Furthermore, in situations where some non-critical information is lost, header resynchronisation will result in the information being restored but in the mean time useful decompressed data packets are generated using the existing data packets. Thus, the useful information of data packets that would otherwise have been discarded may be utilised.

In particular, the invention does not require the presence of check sum data and may be used in a large variety of communication systems. A significant complexity reduction may be achieved as no validity check is required.

According to a feature of the invention, no header information validity check is performed in the second mode of operation.

Hence, the invention allows for efficient performance without requiring any validity check and thus may be used in a wide range of systems and protocols and/or may provide reduced complexity/resource requirement and/or may decrease the overhead of the data packets thereby increasing the performance of the communication system as a whole.

It will be appreciated that not performing a header information validity check is equivalent to executing a validity check process and discarding or ignoring the outcome of this check.

According to another feature of the invention, the data packets do not comprise check sum data. Hence, the invention allows for an efficient error management and recovery even for data packets not comprising any check sum data.

According to another feature of the invention, the data packets comprise check sum data and the step of generating decompressed data packets is independent of a validity of the check sum data. For example, a validity check of the check sum data may be ignored or may not be performed. The invention thus allows for improved performance and may specifically allow for useful decompressed data packets being generated even if a check sum validity check would indicate that the data packet comprised errors. The check sum data may specifically be header check sum data and the generation of decompressed data packets may independent of a validity of the header check sum data but take the validity of other check sum data into account.

According to another feature of the invention, the second mode of operation is exited when the header compression resynchronisation process terminates. The method may specifically return to the first mode of operation when normal header decompression functionality can be resumed. This allows for a suitable and high performance error management process.

According to another feature of the invention, the header compression resynchronisation comprises requesting uncompressed header information from a transmitter of the data packets. This allows for a convenient and robust means of restoring any lost header compression information.

According to another feature of the invention, the second mode of operation is exited when the uncompressed header information is received. This allows for a suitable indication of a successful recovery from an error situation and typically means that all information required for normal operation is available.

According to another feature of the invention, the header resynchronisation process is specified in section 3.3.5 of the document Request For Comment RFC2508 of the Internet Engineering Task Force (IETF). This is a particularly suitable method for re-synchronising headers following one or more lost data packets.

According to another feature of the invention, the first mode of operation further comprises decompressing compressed header information of received data packets; and generating decompressed data packets comprising the decompressed header information. Preferably, the decompressed data packets are generated both in the first and second mode of operation and thus a continuous stream of data packets may be provided. Specifically, this may allow for e.g. a voice data service where interruptions caused by lost data packets are minimised.

According to another feature of the invention, the data packets comprise sequence information and the step of determining a disruption comprises determining a disruption indication in response to the sequence information. This provides for a particularly simple and robust way of detecting interruptions in a flow of data packets.

According to another feature of the invention, the sequence information comprises a data packet sequence number and the step of determining a disruption indication comprises: determining an expected sequence number for a first data packet; extracting a sequence number of the first data packet; and setting the disruption indication corresponding to a disruption if the expected sequence number does not match a sequence number of the first data packet. This provides for a particularly simple and robust way of detecting interruptions in a flow of data packets.

According to another feature of the invention, the step of decompressing comprises determining a likely header parameter value of a first parameter of the decompressed header in response to an expected change from a header of a previous data packet. Hence, data packets may e.g. be generated following a lost data packet and before header resynchronisation by extrapolating parameter values to subsequent data packets.

According to another feature of the invention, an update relationship between parameter values of data packets is known from a previous data packet and the likely header parameter value is determined in response to the update relationship and a previous parameter value of the first parameter. This provides for a particularly suitable method of extrapolating data. For example, data packets may comprise a header parameter which increments a specific amount between each header and the parameter value of a data packet following a lost data packet may be incremented by a value corresponding to the number of data packets that have been lost.

According to another feature of the invention, the step of decompressing comprises setting a parameter value of a second parameter of the decompressed header to a default parameter value. This may for example allow for decompressed data packets to be generated even if no information is available for a specific parameter. Specifically, this may be used for parameters which are not critical or where a wrong parameter value will result in acceptable consequences.

According to another feature of the invention, the method further comprises the step of setting an uncertainty indication for the data service when entering the second mode of operation. This allows for equipment, applications or users to take into account that the possibility

of data of the data packets being invalid is increased. For example, a vocoder may take into consideration that voice data of the decompressed data packets are less reliable than for data packets received when in the first mode of operation.

According to another feature of the invention, the method further comprises the step of including uncertainty information in the data packets generated in the second mode. This allows for any equipment, applications or users to take into account that the data of the data packets may be less reliable than for data packets generated in the first mode of operation. The uncertainty indication may for example be a simple binary value included in the decompressed data packet when in the second mode of operation. Additionally or alternatively, a more detailed uncertainty identification indicating the level of uncertainty may for example be included.

According to another feature of the invention, the method further comprises the step of transmitting the generated data packets. Decompressed data packets of the first mode, the second mode or preferably both may be transmitted over a suitable medium to another physical, logical or structural entity.

Preferably, the compressed header information comprises a compressed Internet Protocol (IP) header, a compressed User Datagram Protocol (UDP) header and/or a compressed Real Time Protocol (RTP) header.

Preferably the data service is a voice data packet service. Preferably, the data packets are received over a radio interface. The invention may advantageously be implemented in a cellular communication system, such as a TETRA (Terrestrial Trunked Radio operating according to standards defined by the European Telecommunications Institute), GSM

(Global System for Mobile communication) or UMTS (or other 3G) cellular communication system.

According to a second aspect of the invention, there is provided an apparatus for communicating data packets of a data service comprising compressed header information; the apparatus comprising: means for receiving data packets in a first mode of operation, means for extracting data packet sequence information from at least one of the data packets in the first mode of operation, means for determining a disruption indication in response to the packet sequence information and if the disruption indicator indicates a disruption entering a second mode of operation for the data service; initialising a header compression resynchronisation process in the second mode of operation, decompressing compressed header information of received data packets in the second mode of operation, and generating decompressed data packets comprising the decompressed header information in the second mode of operation.

These and other aspects, features and advantages of the invention will be apparent from and elucidated with reference to the embodiment(s) described hereinafter.

An embodiment of the invention will now be described, by way of example only, with reference to the accompanying drawings, in which

Brief Description of the Drawings

FIG. 1 illustrates an example of a composition of a data packet;

FIG. 2 is a block schematic diagram of an apparatus for communicating data packets of a data service in accordance with a preferred embodiment of the invention; and

FIG. 3 is a flowchart illustrating a method of communicating data packets of a data service comprising compressed header information in accordance with an embodiment of the invention.

Detailed Description of a Preferred Embodiment of the Invention

The following description focuses on an embodiment of the invention applicable to a voice data packet service communicated over a network interface in a cellular communication system such as GPRS, UMTS or TETRA. However, it will be appreciated that the invention is not limited to this application but may be applied to e.g. many other data services, interfaces and/or communication systems.

In the preferred embodiment, a voice data packet service is set up in the fixed network for a mobile terminal of a TETRA radio communication system. In this embodiment, the base station receives voice samples over a dedicated circuit switched link on the air interface. The data samples are converted to packet data in the base station and transmitted as a data packet stream to another network element. Typically, back haul links from base stations are very costly and therefore it is desirable to limit the required capacity. Accordingly, the packet data stream uses header compression in the preferred embodiment. The data packet stream is in the preferred embodiment converted back to a circuit switched service at another base station for communication to another mobile terminal over an air interface. In other embodiments, the mobile terminal may directly communicate data packets over the air interface.

The communication of the data packets is in the preferred embodiment in accordance with a layered communication protocol comprising a plurality of sub-protocols. Specifically, the data service employs a Real Time Protocol (RTP) over a User Datagram Protocol (UDP) which resides on top of an Internet Protocol (IP). Thus, in the preferred embodiment the base station packetizes the received voice data and adds the required header information.

Thus, when a data packet is to be transmitted into the fixed network from the base station, the base station generates the payload data for the data packet and passes this to the layer implementing the RTP protocol. Accordingly, the RTP process ads an RTP header which comprises information required to implement the RTP protocol. Specifically, the RTP header may comprise a time stamp which allows the receiving application to order the received packets and to playback the data in the right order and at the right time (i.e. in real time).

The data packet comprising the RTP header is then passed to a process implementing the UDP protocol. The process adds a UDP header which comprises the required information for implementing the UDP protocol. Specifically, UDP comprises information such as a port number allowing for different applications to maintain individual channels for data. The UDP header may also comprise a checksum which allows the receiving end to check the validity of the received data. The use of check sum data is optional in UDP and may be used in some embodiments but not in others.

The UDP data packet (also known as a datagram) is passed to the process which implements the IP protocol. This process adds an IP header comprising information such as a source address, destination address etc used for implementing the IP protocol.

The IP/UDP/RTP data packet is then passed to the lower layers for transmission from the base station. FIG. 1 illustrates an example of a composition of a IP/UDP/RTP data packet. As illustrated, the data packet 100 comprises a data payload portion 101 comprising the speech data generated by the voice application. The data packet 100 furthermore comprises the added RTP header 103, the UDP header 105 and the IP header 107. As illustrated, a significant overhead is added to the original data payload. As a typical example, a data packet may be generated for every 10 msec of voice and 20 bytes of data may be generated by the voice coder every 10 msec (corresponding to an encoded data rate of 2kbps). The data payload 101 will thus comprise 20 bytes. The RTP header is typically 12 bytes, the UDP header 8 bytes and the IP header (IP version 4) is typically 20 bytes. Thus, a 60 byte data packet is generated to carry 20 bytes of data resulting in an efficiency of only 33%. This efficiency is further reduced for IP version 6 where the header is a minimum of 40 bytes.

In order to increase the efficiency and reduce the overhead, it is desirable to use header compression.

Header compression utilises the principle that many fields of the IP/UDP/RTP header may be constant between consecutive data packets or change in a predictable way. For example, data packets between the same receiver and transmitter will always have the same IP source and destination address fields. As another example, an IP identifier value may increase by one for each consecutive data packet and can thus be predicted by the receiver.

Accordingly, a header compression algorithm may compress the header by omitting data which does not change or change predictably between

data packets. Specifically, it is known for a header compression algorithm to establish which fields in a particular stream of packets are constant or predictable. Once that has been accomplished, there is no need to transmit the contents of these fields over the link and the IP/UDP/RTP header stack is replaced by a compressed header that identifies the stream that the packet belongs to (known as a context) and which only contains the values of fields which change unpredictably. The decompressor reconstructs the IP/UDP/RTP header by combining the information received over the link with pre-stored values for the constant fields. Similarly, the predictable fields are determined based on stored information.

If the value of a field is considered to be constant or predictably changing is modified in an unexpected way, the compressed header will comprise information informing the decompressor of this change. For example, it is possible in some situations that predictable fields may at some point change unexpectedly. For instance, an IP identifier value that has incremented by one in each packet (21, 22, 23,...) may be modified to increment by two (...27, 28, 30). The compressor responds by encoding the difference in the compressed packet and instructing the decompressor to expect subsequent packets belonging to this context to increment by this new value. Thus, the decompressor will now expect IP identifiers to increment 30, 32, 34... until instructed otherwise.

It is important for the performance of the header compression process, as well as for the data service and the communication system in general, that the system is robust and efficient in the presence of lost data packets. Especially, in radio communications, communication errors may exist that result in data packets not being correctly received. Radio communication, such as microwave links, is frequently used for back haul from remote or inaccessible base stations.

The compressed IP/UDP/RTP header of the preferred embodiment comprises a 4-bit sequence number that allows the decompressor to detect if a compressed packet was lost on the transmission link. If a data packet has been lost, it is generally uncertain whether a header of a subsequent data packet can be successfully decompressed, and thus whether a decompressed data packet with an uncompressed IP/UDP/RTP header can be generated. If the lost data packet comprised no updates or changes to the stored information or applied rules for generating the compressed data, the header may successfully be decompressed using the stored information and taking the number of lost data packets into account. However, if the lost data packet comprises information that modifies the stored information or the rules by which the predictable fields change, the decompressor will generate an erroneous header. For example, if the packet comprising information of the IP identifier changing from incrementing by one to incrementing by two is lost, the de-compressor will not have the required information for decompressing the subsequent data packets.

FIG. 2 illustrates a block schematic of an apparatus 200 for communicating data packets of a data service in accordance with a preferred embodiment of the invention.

The apparatus 200 comprises a receiver 201 which receives data packets of the data service. The data packets comprise a compressed IP/UDP/RTP header. In the preferred embodiment, the apparatus is a network element part receiving data packets from a base station. The receiver 201 is coupled to a decompressor 203 which is operable to decompress the compressed header information of received data packets.

The decompressor 203 is coupled to a data packet processor 205 which is operable to generate compressed data packets comprising the payload data and the uncompressed IP/UDP/RTP header. The data packet processor 205 is operable to transmit the data packets to other external or internal units. For example, the data packet processor 205 may communicate the generated data packets to an internal voice decoder. However, in the preferred embodiment the data packet processor 205 transmits the generated data packet to a destination element through a packet data network. As the IP/UDP/RTP header is decompressed, the data packet is suitable for communication in a conventional IP/UDP/RTP based communication network which does not utilise header compression. In a first mode of operation, where no lost data packets have been detected, the apparatus receives data packets comprising compressed IP/UDP/RTP headers, decompresses the headers and generates the corresponding uncompressed data packets.

The receiver 201 is furthermore coupled to a sequence extractor 207 which extracts data packet sequence information from the received data packets. Specifically, the sequence extractor 207 extracts a sequence number from the received data packets. The sequence number is in the preferred embodiment, the 4 bit UDP sequence number comprised in the UDP header section.

The sequence extractor 207 is coupled to a disruption controller 209 which is operable to determine a disruption indication in response to the packet sequence information. Specifically, the disruption controller 209 detects that an interruption has occurred if there is a gap between the UDP sequence numbers of consecutively received data packets. Thus, the disruption controller 209 sets the disruption indication to indicate a disruption if it is detected that one or more data packets have been lost.

The disruption controller 209 is operable to control the other functional modules of the apparatus and is specifically operable to control the apparatus to operate in a second mode of operation following a disruption. The second mode may in the preferred embodiment be seen as an error recovery mode wherein the operation of the apparatus is aimed at reducing the detrimental effect of one or more lost data packets.

The disruption indication need not be a specific, explicit or accessible indication but is in the preferred embodiment simply the result of the sequence number check which is internally used by the disruption controller 209.

The disruption controller 209 is coupled to a header resynchronisation processor 211. In the second mode of operation, the disruption controller 209 is operable to control the header resynchronisation processor 211 to initialise a header compression resynchronisation process. Accordingly, when the disruption controller 209 detects that a data packet has been lost, it controls the header resynchronisation processor 211 to resynchronise the header information between the transmitting end and the apparatus. Specifically, the header resynchronisation processor 211 may cause a request to be sent to the transmitter which results in the transmitter sending the information required by the decompressor 203. Specifically, the header resynchronisation processor 211 may request that a full uncompressed header is transmitted.

In accordance with the preferred embodiment, the initialisation of the header compression resynchronisation does not cause the decompressor 203 or the data packet processor 205 to stop decompressing headers or to generate decompressed data packets. Rather, the decompressor 203 is

controlled by the disruption controller 209 to continue to decompress the IP/UDP/RTP header based on the existing information.

Thus, while in the second mode, the apparatus continues to generate data packets based on the available information while at the same time proceeding to resynchronise the header regardless of whether a validity check has indicated that a subsequent data packet is valid or not.

Since the data packets generated in the second mode may have been decompressed by use of erroneous information there is an increased probability that these may comprise errors. However, although the data packets have reduced reliability compared to normal packets, it is also possible that correct data packets are generated. Furthermore, it is possible that even if the wrong information has been used for the decompression of the header, the resulting error may be acceptable. For example, the voice data of a data packet may be valid and can be used by a decoder even if a field indicating an IP option is erroneously set. Thus the approach ensures that no data packets are discarded unnecessarily and allows for information of data packets that would otherwise be discarded to possibly be used.

At the same time, it is ensured that header resynchronisation is started instantly and therefore that the duration of the error recovery state is minimised. Furthermore, the approach allows for increased error robustness as the header synchronisation is always carried out whenever one or more data packets have been lost. Furthermore, there is no requirement for a validity check to be performed on the received data packet in order to determine if header resynchronisation is necessary.

Thus in comparison to an algorithm performing a validity check on the first data packet received after a lost data packet and either continuing unaffected or pausing the communication until an uncompressed header is received, a number of advantages is achieved. Specifically, less information and fewer data packets may be lost and increased robustness may be achieved. For example, for a voice service, it would in most cased be preferable to forward the packet with incorrectly restored IP/UDP/RTP headers as the voice information may possibly still be used. The worst case is that the generated data packets cannot be used but this is no worse than if the data packets had been discarded anyway. Thus the resulting performance will be as good as or better than a validity check based algorithm.

Furthermore, the need for a validity check and thus the presence of checksums may be obviated thus permitting a reduced overhead and increased performance.

FIG. 3 illustrates a flowchart of a method of communicating data packets of a data service comprising compressed header information in accordance with a preferred embodiment of the invention. The method is applicable to the apparatus 200 of FIG. 2 and will be described with reference to this.

The method initiates in step 301 wherein the receiver 201 receives data packets comprising compressed IP/UDP/RTP header information. The data packets are in the preferred embodiment received by a network element over a network interface.

Step 301 is followed by step 303 wherein the sequence extractor 207 extracts data packet sequence information from the data packets.

Specifically, the sequence extractor 207 extracts the 4 bit UDP sequence number of the IP/UDP/RTP header.

Step 303 is followed by step 305 wherein the disruption controller 209 determines a disruption indication in response to the packet sequence information. Specifically, the disruption controller 209 compares the UDP sequence number of the current data packet with that of the previous packet. If the UDP sequence number is incremented by one in comparison to the previous UDP sequence number, the disruption indication is set to indicate that no disruption has occurred as this indicates that no data packets have been lost (Due to the 4 bit word length of the UDP sequence indication, the detection will not be able to detect if a multiple of 16 data packets have been lost. However, this is an extremely unlikely situation). If the extracted UDP sequence number does not correspond to that of the previous packet incremented by one, the disruption indication is set to indicate that a disruption has occurred as this indicates that one or more data packets have been lost.

Thus in the preferred embodiment, an expected sequence number is determined as that of the previous data packet incremented by one. This expected sequence number is compared to the extracted sequence number and if the expected sequence number does not match the extracted sequence number, the disruption indication is set to correspond to a disruption.

Step 305 is followed by step 307 wherein the method branches depending on whether the disruption indication indicates that a disruption has occurred or not. If a disruption has occurred, the apparatus enters a second mode of operation comprising steps 313 to 321. If the disruption indication indicates that no disruption has occurred, the method continues in step 309 wherein the decompressor

309 decompresses the IP/UDP/RTP header in accordance with the header decompression protocol. Step 311 is followed by step 311 wherein the data packet processor 205 generates decompressed data packets. In the preferred embodiment, this simply comprises combining the pay load section of the received data packet with the decompressed IP/UDP/RTP header generated by the decompressor 203.

If the disruption indication indicates that one or more data packets have been lost, the apparatus continues in a second mode of operation. In this case, step 307 is followed by step 313 wherein a header compression resynchronisation process is initialised by the header resynchronisation processor 211. In the preferred embodiment, the header compression resynchronisation process is initialised by the header resynchronisation processor 211 controlling a request message to be sent to the data packet transmitter. The request message comprises a request for a full uncompressed header to be transmitted in the next transmitted data packet. Upon receipt of the uncompressed header, the decompressor 203 will regain all information required for successfully decoding data packets. Specifically, the header resynchronisation process is equivalent to the resynchronisation process specified in section. 3.3.5 of RFC2508 of the Internet Engineering Task Force (IETF).

Thus, a header compression resynchronisation process is initialised whenever the apparatus enters the second mode of operation. The resynchronisation process is started independently of any validity check and is not dependent on a possible decompression of the IP/UDP/RTP header. This allows for increased reliability.

Furthermore, step 313 is followed by step 315 wherein the decompressor 203 decompresses the compressed header information of the received data packets and step 317 wherein decompressed data packets

comprising the decompressed header information are generated. These steps are performed in addition to the header resynchronisation process being initialised and are not dependent on any validity check of any data of the received or decompressed data. Thus in the preferred embodiment, an error recovery mode is entered wherein decompressed data packets are generated in parallel to a header resynchronisation process being initialised. Hence, no specific evaluation of whether the information available to the decompressor is sufficient is required. Rather, the inventors have realised that it is advantageous to perform in parallel operations that may correct the loss of required information as well as operations that can generate useful data packets in case no critical information has been lost. Thus, the desired operation can be achieved for both possible error situations without requiring any evaluation of which error situation has occurred. This provides improved performance and reliability of the communication.

In the preferred embodiment, the decompressor, when in the second mode of operation, utilises the stored information and rules to generate a decompressed header assuming that no updates to these have occurred during the lost data packets.

Thus, for fields which for the last received data packet were indicated as constant, the decompressor 203 sets the parameter value to a likely parameter value which is identical to the stored value.

For a parameter that changes predictably (e.g. is incremented by one between each packet), the update relationship between parameter values of consecutive data packets is known. The decompressor 203 accordingly determines the likely parameter value as the last stored parameter value updated in accordance with the update relationship. For example, a parameter may increment by one between each data

packet and the parameter value of the last received data packet may have been X. If the UDP sequence number indicates that three packets have been lost between the last received and the current data packet, the decompressor sets the parameter value to X+4. Of course, the lost packets may have comprised an update to the update relationship causing the determined value to be incorrect. However, this is likely to happen rarely in most applications.

Some parameter values may change unpredictably or the decompressor 203 may have no information related to this parameter value. In this case, the decompressor 203 may simply insert a default value. For example, an IP header may comprise flags indicating whether a specific option is used and the decompressor may simply set one or more of these parameters to indicate a default option configuration.

Hence, data packets are generated which may or may not have correctly decompressed headers. If the header is correctly decompressed, the data packets can be used at the destination. If the header is incorrectly decompressed, the data packet may be discarded in the system or may not be useable at the destination. However, this is no worse than if the data packet had already been discarded for failing a validity check. In fact, in many cases an erroneously decompressed header may still be used at the destination which may still be able to receive the data packet and to extract the payload. Thus the throughput of useful data packets to the destination may be improved but will not be worse than for an algorithm discarding data packets that fail a validity check.

Step 317 is followed by step 319 wherein the next data packet is received.

Step 319 is followed by step 321 wherein it is determined whether the second mode of operation should be terminated and the apparatus return to the normal first mode of operation. Specifically, the second mode of operation is exited when the header compression resynchronisation process terminates. The second mode of operation is thus exited when the header compression resynchronisation process has ensured that the decompressor 203 has all available information required for decompressing the IP/UDP/RTP headers of the received data packets. In the preferred embodiment, the second mode of operation is exited when the uncompressed header information is received. Hence, the apparatus stays in the second mode of operation until a data packet is received which has an uncompressed IP/UDP/RTP header. This corresponds to the situation when the data service is initialised and the header compression and communication in general may thus proceed in accordance with the protocol used.

If it is determined in step 319 that the header compression resynchronisation process is still ongoing, the method returns to step 315 wherein the IP/UDP/RTP header of the received data packet is decompressed. If it is determined that the header compression resynchronisation has terminated, the method exits the second mode of operation and returns to the first mode of operation by returning to step 311 wherein a decompressed data packet is generated based on the last received data packet. The method then continues normal operation in the first mode of operation.

Hence, the method of the preferred embodiment allows for an efficient and reliable recovery after one or more data packets have been lost. The method does not require or rely on a validity check of the received data packet or decompressed header and accordingly may be used even for data packets that do not comprise check sum data. Furthermore, if check

sum data is included in the IP/UDP/RTP headers this may be ignored by the recovery process of the preferred embodiment.

In the preferred embodiment, an uncertainty indication is set for the data service when entering the second mode of operation. Thus, when the error recovery mode is entered, it is indicated that the generated data packets have an increased probability of comprising errors. This may be used by an application which may change the operation in accordance. For example, a voice decoder terminal may ignore as many header fields as possible and simply attempt to extract the data payload of the data packets. As an example, the destination may ignore any changes in optional settings while the uncertainty indication for the data service indicates that the IP/UDP/RTP header is less reliable. In the preferred embodiment, the uncertainty indication is included as an indication in data packets which are generated and communicated to external network elements.

The invention can be implemented in any suitable form including hardware, software, firmware or any combination of these. However, preferably, the invention is implemented as computer software running on one or more data processors and/or digital signal processors. The elements and components of an embodiment of the invention may be physically, functionally and logically implemented in any suitable way. Indeed the functionality may be implemented in a single unit, in a plurality of units or as part of other functional units. As such, the invention may be implemented in a single unit or may be physically and functionally distributed between different units and processors.

Although the present invention has been described in connection with the preferred embodiment, it is not intended to be limited to the specific form set forth herein. Rather, the scope of the present invention is

limited only by the accompanying claims. In the claims, the term comprising does not exclude the presence of other elements or steps. Furthermore, although individually listed, a plurality of means, elements or method steps may be implemented by e.g. a single unit or processor. Additionally, although individual features may be included in different claims, these may possibly be advantageously combined, and the inclusion in different claims does not imply that a combination of features is not feasible and/or advantageous. In addition, singular references do not exclude a plurality. Thus references to "a", "an", "first", "second" etc do not preclude a plurality.

CLAIMS

1. A method of communicating data packets of a data service comprising compressed header information; the method comprising the steps of:

in a first mode of operation:

receiving data packets:

extracting data packet sequence information from at least one of the data packets;

determining a disruption indication in response to the packet sequence information;

and if the disruption indicator indicates a disruption entering a second mode of operation for the data service; the second mode of operation comprising:

initialising a header compression resynchronisation process;

decompressing compressed header information of received data packets; and

generating decompressed data packets comprising the decompressed header information.

2. A method as claimed in claim 1 wherein no header information validity check is performed in the second mode of operation.

3. A method as claimed in any previous claim wherein the data packets do not comprise check sum data.

4. A method as claimed in any previous claim wherein the data packets comprise check sum data and the step of generating decompressed data packets is independent of a validity of the check sum data.

5. A method as claimed in any previous claim wherein the second mode of operation is exited when the header compression resynchronisation process terminates.

6. A method as claimed in any previous claim wherein the header compression resynchronisation comprises requesting uncompressed header information from a transmitter of the data packets.

7. A method as claimed in claim 6 wherein the second mode of operation is exited when the uncompressed header information is received.

8. A method as claimed in any precious claim wherein the header resynchronisation process is the resynchronisation process specified in section. 3.3.5 in the document Request For Comment RFC2508 of the Internet Engineering Task Force (IETF).

9. A method as claimed in any previous claim wherein the first mode of operation further comprises decompressing compressed header information of received data packets; and generating decompressed data packets comprising the decompressed header information.

10. A method as claimed in any previous claim wherein data packets comprise sequence information and the step of determining a disruption comprises determining a disruption indication in response to the sequence information.

11. A method as claimed in claim 10 wherein the sequence information comprises a data packet sequence number and the step of determining a disruption indication comprises:

determining an expected sequence number for a first data packet;

extracting a sequence number of the first data packet; and

setting the disruption indication corresponding to a disruption if the expected sequence number does not match the sequence number of the first data packet.

12. A method as claimed in any previous claim wherein the step of decompressing comprises determining a likely header parameter value of a first parameter of the decompressed header in response to a expected change from a header of a previous data packet.

13. A method as claimed in claim 12 wherein an update relationship between parameter values of data packets is known from a previous data packet and the likely header parameter value is determined in response to the update relationship and a previous parameter value of the first parameter.

14. A method as claimed in any previous claim wherein the step of decompressing comprises setting a parameter value of a second parameter of the decompressed header to a default parameter value.

15. A method as claimed in any previous claim further comprising the step of setting an uncertainty indication for the data service when entering the second mode of operation.

16. A method as claimed in any previous claim further comprising the step of including uncertainty information in the data packets generated in the second mode.

17. A method as claimed in any previous claim further comprising the step of transmitting the generated data packets.

18. A method as claimed in any previous claim wherein the compressed header information comprises a compressed Internet Protocol (IP) header.

19. A method as claimed in any previous claim wherein the compressed header information comprises a compressed User Datagram Protocol (UDP) header.

20. A method as claimed in any previous claim wherein the compressed header information comprises a compressed Real Time Protocol (RTP) header.

21. A method as claimed in any previous claim wherein the data service is a voice data packet service.

21. A method as claimed in any previous claim wherein the data packets are received over a radio interface.

22. The method of any previous claim used in a cellular communication system.

23. A computer program enabling the carrying out of a method according to any previous claim.

24. A record carrier comprising a computer program as claimed in claim 23.

25. An apparatus for communicating data packets of a data service comprising compressed header information; the apparatus comprising:
means for receiving data packets in a first mode of operation;

means for extracting data packet sequence information from at least one of the data packets in the first mode of operation;

means for determining a disruption indication in response to the packet sequence information and if the disruption indicator indicates a disruption entering a second mode of operation for the data service;

initialising a header compression resynchronisation process in the second mode of operation;

decompressing compressed header information of received data packets in the second mode of operation; and

generating decompressed data packets comprising the decompressed header information in the second mode of operation.

26. A method according to any one of claims 1 to 22 and substantially as herein described with reference to the accompanying drawings.

27. An apparatus according to claim 25 and substantially as herein described with reference to the accompanying drawings.

INVESTOR IN PEOPLE

Application No: GB 0316054.6
Claims searched: 1-27

Examiner: Matthew Nelson
Date of search: 20 October 2003

Patents Act 1977 : Search Report under Section 17

Documents considered to be relevant:

Category	Relevant to claims	Identity of document and passage or figure of particular relevance
X	1-3, 5-14 & 17-25.	S. Casner et al, "Compressing IP/UDP/RTP Headers for Low-Speed Serial Links", Internet Engineering Task Force (IEFT), Network Working Group, Request for Comments RFC 2508, dated February 1999. Retrieved from the Internet: <URL: http://www.ietf.org/rfc/rfc2508.txt>. See section 3.3.5.
X	1-3, 5-14 & 17-25.	M. Degermark et al, "IP Header Compression", Internet Engineering Task Force (IEFT), Network Working Group, Request for Comments RFC 2507, dated February 1999. Retrieved from the Internet: <URL: http://www.ietf.org/rfc/rfc2507.txt>. See sections 10.1 and 10.2.
X	1,5-14, 17-25	EP 1137237 A3 (NTT) See the abstract and paragraphs [0066]-[0068].
A		WO 03/030435 A1 (MATSUSHITA) See the abstract.

Categories:

X	Document indicating lack of novelty or inventive step	A	Document indicating technological background and/or state of the art.
Y	Document indicating lack of inventive step if combined with one or more other documents of same category.	P	Document published on or after the declared priority date but before the filing date of this invention.
&	Member of the same patent family	E	Patent document published on or after, but with priority date earlier than, the filing date of this application.

Field of Search:

Search of GB, EP, WO & US patent documents classified in the following areas of the UKC[V]:

H4P

Worldwide search of patent documents classified in the following areas of the IPC[7]:

H04L

The following online and other databases have been used in the preparation of this search report :

WPI, EPODOC, JAPIO, INSPEC, Selected Internet sites