Office de la Propriété Intellectuelle du Canada
Un organisme d'Industrie Canada

Canadian Intellectual Property Office
An agency of Industry Canada

CA 2199647 C 2004/01/27

(11)(21) 2 199 647

(12) BREVET CANADIEN
CANADIAN PATENT

(13) C

(22) Date de dépôt/Filing Date: 1997/03/11

(41) Mise à la disp. pub./Open to Public Insp.: 1998/09/11

(45) Date de délivrance/Issue Date: 2004/01/27

(51) Cl.Int.$^{6}$/Int.Cl.$^{6}$ H04L 12/56, H04L 29/02

(72) Inventeur/Inventor:
WOOD, ROBERT, CA

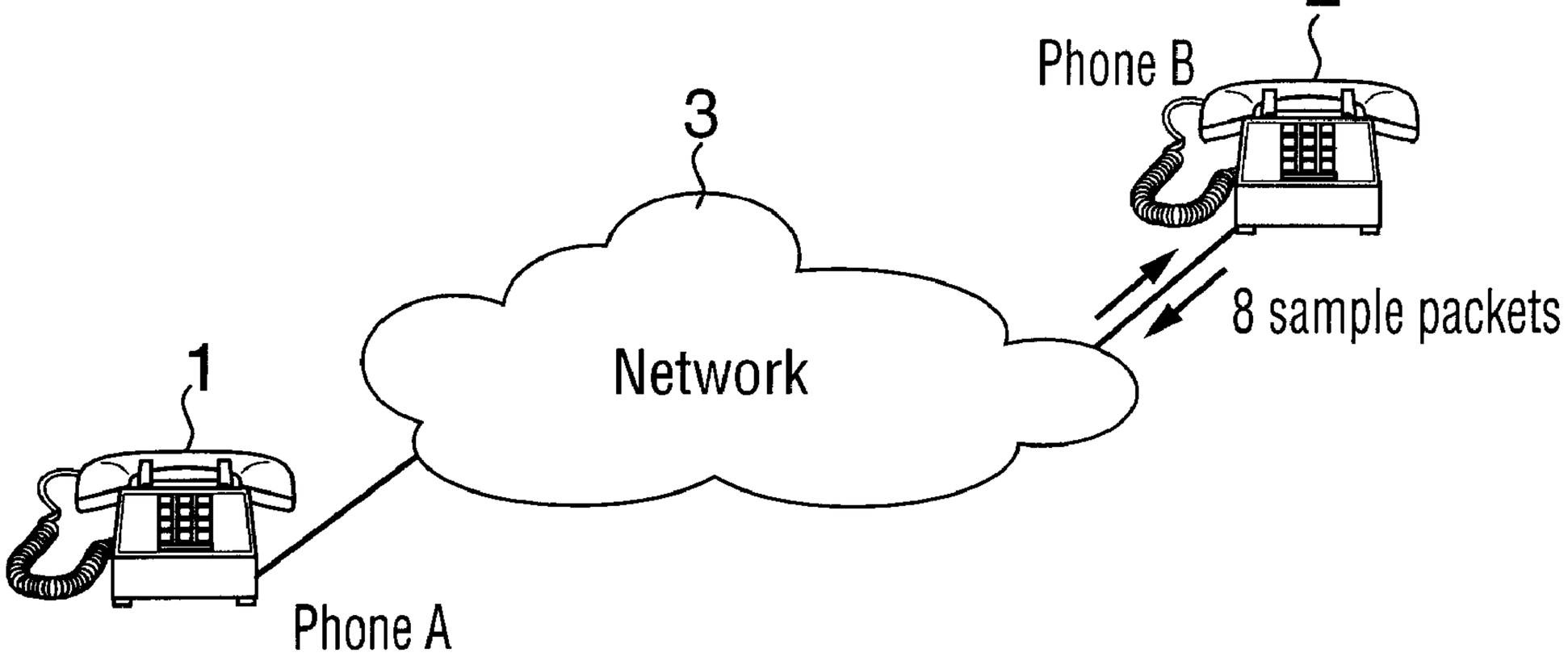
(73) Propriétaire/Owner:
MITEL NETWORKS CORPORATION, CA

(74) Agent: MARKS & CLERK

(54) Titre : SYNCHRONISATION DE DISPOSITIFS DE COMMUNICATIONS RACCORDES PAR UN LIEN ASYNCHRONE

(54) Title: SYNCHRONIZATION OF COMMUNICATION DEVICES CONNECTED OVER AN ASYNCHRONOUS LINK






(57) Abrégé/Abstract:
In a method of establishing communication between synchronous devices over an asynchronous communications channel, each device samples a signal to be transmitted at a rate determined by a local clock. The rate of transmission of samples over the channel is determined by the local clock in each device. At least one of the devices is periodically adjusted to match the rates of transmission and arrival of samples over the channel, for example, by adjusting the frequency of the local clock or by dropping or duplicating samples. With this arrangement there is no need to exchange explicit timing information.

## ABSTRACT OF THE DISCLOSURE

In a method of establishing communication between synchronous devices over an asynchronous communications channel, each device samples a signal to be transmitted at a rate determined by a local clock. The rate of transmission of samples over the channel is determined by the local clock in each device. At least one of the devices is periodically adjusted to match the rates of transmission and arrival of samples over the channel, for example, by adjusting the frequency of the local clock or by dropping or duplicating samples. With this arrangement there is no need to exchange explicit timing information.

# SYNCHRONIZATION OF COMMUNICATION DEVICES CONNECTED OVER AN ASYNCHRONOUS LINK

The invention relates to a method and apparatus for establishing communication between synchronous communication devices, such as telephones, over an asynchronous link.

Digital telephones sample data at a predetermined rate, which is generally 8KHz, as determined by a local timer. This gives the familiar frame period of 125 μsecs. In the case of TDM transmission over synchronous networks, synchronization is maintained between the sending and receiving telephones so that reconstruction of the analog signal from the received samples is relatively straightforward. However, when the samples are sent in packets across an asynchronous network, substantial jitter is introduced by the network in the packet arrival times. In order to reconstruct the original samples, some means must be available to synchronize the receiving and transmitting devices.

In the prior art, one technique for synchronizing the telephones is to send special packets through the network containing time information, the so-called "residual time stamp", RTS. This technique requires sophisticated algorithms to implement.

In another technique, the telephones are allowed to free-run using their own internal oscillators. Occasionally samples are dropped or added to maintain synchronization at a gross level. This technique results in poor transmission quality.

In yet another technique, a common clock is sent to both telephones to maintain them in synchronization. This technique requires sophisticated algorithms to implement.

An object of the invention is to alleviate the problems of the prior art.

According to the present invention there is provided a method of establishing communication between synchronous devices over an asynchronous communications channel, wherein each device samples a signal to be transmitted at a rate determined by a local clock, transmits and receives samples over said channel at a rate determined by the local clock, and makes periodic adjustments to match the rates of transmission and arrival of samples over the channel.

Normally at least one of the local clocks is periodically adjusted to match the rates of transmission and arrival of samples over the channel, although it is possible to drop or add samples to match the transmission and arrival rates.

This mechanism enables two independent devices, for example, telephones, which communicate through packets of samples, to maintain synchronization with each other to an accuracy greater than their local oscillators, while transferring no timing information directly between each them.

One local clock may run at a fixed rate, in which case only one of the clocks will be adjusted. Alternatively, both clocks may be adjusted.

The invention also provides a synchronous communication device capable of establishing communication with another synchronous communication device over an asynchronous communications channel, comprising means for sampling a signal to be transmitted at a rate determined by a local clock, means for transmitting and receiving samples over said channel at a rate determined by a local clock, and means for making periodic adjustments to match the rates of transmission and arrival of samples over the channel.

This invention allows telephones to be built with an inexpensive oscillator and processor which will still maintain synchronization across a network, and without the need for sophisticated time reference algorithms.

The invention will now be described in more detail, by way of example, only with reference to the accompanying drawings, in which:-

Figure 1 shows two telephones connected across an asynchronous network;

Figure 2 is a block diagram of a telephone in accordance with the invention connected to a network;

Figure 3 is a more detailed block diagram of a network in accordance with the invention.

Figure 4 is a detailed block diagram of an actual implementation of the invention;

Figure 5 is a flow chart illustrating the operation of the invention; and

Figure 6 is a routing for generating the counter value.

Referring to Figure 1, two digital telephones 1, 2 transmit and receive data between each other in packets over a network 3, such as an ATM or Ethernet network, which introduces substantial jitter on the packet arrival times. The telephones include a processor that samples data with a nominal 125uS period, determined by a local timer.

The timer can be either hardware or software based and is driven by a local oscillator of low quality, ~+/- 200ppm. The "transmit" sample rate is determined by this oscillator in the transmitting telephone. The "receive" sample rate is determined by the oscillator in the other telephone.

Each telephone stores consecutive samples in a "transmit" FIFO (First In First OUT) memory. 8 samples from this FIFO are transmitted in a packet every 1 mS as determined by the local clock of the transmitting telephone. More than 8 samples are stored in the transmit FIFO before a packet is transmitted to allow for differences between transmit and receive sample rates.

Each telephone receives a packet of 8 samples shall be every 1 mS as determined by the local clock of the remote telephone and stores them in a "receive" FIFO. When more than 8 samples are received, as in the transmit case, the receiving telephone starts reconstructing the received audio with the 125μS samples.

A pointer is used to inform the processor in each telephone of the number of samples in both transmit and receive FIFOs.

If the two telephone clocks are different in frequency, then over a period of time, there will be a difference in the number of samples in the receive and transmit FIFOs. By periodically checking the receive and transmit FIFO pointer values, the processor in the telephones can make adjustments to their local sample frequency.

To maintain an adequately accurate average of the receive and transmit rates, the FIFO pointers must be compared over a substantial period of time to smooth out the effects of packet jitter. This period, however, must be less than the time it would take to build up a slip of a single packet. The minimum jitter expected is 8 samples due to packet arrival.

The defining parameters for the behaviour of the telephone are the depth of the FIFO, which introduces delay into the signal; the sample time adjustment period and FIFO pointer average period which together determine the dynamic behaviour of end to end frequency lock.

To allow for packet jitter, the depth of the FIFOs should be a minimum of 16 samples. For the transmit FIFO, data will not be output to the network until 16 complete samples are stored in the FIFO. For the receive FIFO, samples will not be extracted from the FIFO until 16 samples have arrived. This creates a maximum delay in the telephone of 2mS.

The receive and transmit FIFO pointers should be sampled every 1 mS and their difference summed over a period of 900mS. If this sum is greater than 3, then 8 is subtracted from the sum to account for the packet jitter; if the sum is less than 3, then 8 is added. The level of four is arrived at as 3 is the maximum possible legitimate FIFO pointer difference due to sampling frequency errors over the period of 900mS, given the oscillator tolerance assumed above.

If the resulting sense number is positive, then the sample frequency is low, if negative, the USB sample frequency is high.

There are two possible ways to adjust sample rate in the telephone. The preferred way is to program the timer in the processor of each telephone to generate a processor interrupt with a period of 125uS and time resolution of +/- 166 nS., given a 12MHz oscillator. Upon interruption, the processor writes an audio sample into the transmit FIFO and reads a sample from the receive FIFO, feeding it to a digital-to-analog conversion circuit.

After performing this. operation, the frequency sensing algorithm is executed. If the frequency sensing algorithm determines the need to adjust the sampling frequency, this is done by incrementing or decrementing the time period programmed into the timer. If the sense number is positive, then the period must be decremented; if negative, then incremented. This period can be adjusted with a resolution of approximately +/- 100ppm.

There are several variables in this system which are available for adjustment, depending upon system response required. These are: packet size, oscillator frequency, FIFO depth and frequency sense algorithm period.

Another approach is not to adjust the frequency of the local oscillator, but instead to duplicate or discard samples. Thus, if the sense number is positive, that number of samples is skipped; if negative, a sample is repeated that number of times. As there is potentially 375 μS of sample to be affected, this algorithm is preferably run over a 300 mS period.

Referring now to Figure 2, the USB telephone 1 is connected to a USB interface 11 of a workstation 12, which has a network interface card 13 connected to a LAN 12, for example, an Ethenet or ATM LAN. To reduce cost, the USB telephone 1 should have a software phase-locked loop to synchronize with the network 12.

Data is received from the network 12 via a link 14 in packets of multiple 125 μS samples. The source end of the link's clock is synchronized to the PSTN. The received packets are transferred to a PC workstation memory buffer 15 via DMA and NIC driver software. The USB driver 16 outputs samples from this buffer to the USB telephone 1 in packets of 8 samples every 1 μS.

The microcontroller 20 in the telephone 1 is responsible for sending a sample every 1 25μS to the digital-to-analog converter (not shown) in the telephone. The microcontroller 20 also forms a packet of 8 sample every 1 mS to return to the source end of the LAN via link 21. The incoming and outgoing samples are buffered in FIFOs 23, 24.

The 125μS time period in the USB 1 is set by a hardware counter 22 generating an interrupt to the microcontroller. The microcontroller software also keeps a count of received number of samples (not packets) RXCNT and number of samples transmitted to the D/A converter DACNT. These counts are averaged over a period of time to give rates of incoming and outgoing samples and the frequency of outgoing samples adjusted periodically by adjusting the hardware counter value.

Assuming the above accuracy, a 125μS count requires a count of 500. This count will have an accuracy of 0.01% due to the oscillator and 0.2% resolution for a total

accuracy of 0.21%. The minimum time that RXCNT and DACNT must be averaged over to detect an error due to this counter accuracy is 125 μS / 0.21% or 59.5 mS. Therefore, the frequency should be adjusted every 60mS.

The method of adjusting the frequency shown in the diagram will result, worst case, in a square wave frequency modulation of 0.21% of the voice at a frequency of 8Hz.

Another method of adjusting the synchronization is to skip a sample if the frequency need to be adjusted upwards and duplicate a sample if the frequency needs to be adjusted downwards. This arrangement is simpler to implement and requires no change to the hardware counter interrupt value.

To lock the USB data samples to the source end of the LAN a similar approach is taken. This time, however, the USB, on detecting a need to adjust frequency, will either transmit an additional sample in a packet or one less sample in the packet. In other words, if the PC clock is running slow, then periodically, the USB will transmit 9 samples in a packet; if fast, then only 7 samples will be transmitted. It is for this reason that RXCNT should be a sample count rather than a packet count.

The invention permits two devices with independent timing sources that are used to generate periodic data samples, otherwise referred to as TDM data, to communicate. These data samples are gathered into a packet of samples and transmitted to the other source over anisochronous communication links. The transmission time delay of the packets over the communications links is unknown, variable and unrelated to each other. Upon reception, the data contained in these packets is transformed back to TDM by the receiving unit. The invention permits the sample rates of the two sources sample rates to remain in step with each other, without the transmission of specific timing reference information.

The device need not be implemented in both sources. In the situation where one of the sources is a fixed rate, i.e. does not implement the mechanism described herein, the other unit will still perform correctly and synchronize its sample rate to that of the fixed source.

Figure 3 shows the components of a telephone in more detail. The telephone is driven by a low quality, high frequency, constant frequency generator 30. Typically, this may be a microprocessor's system clock of 8 MHz. Alternatively, it can be of any frequency and function. Strict accuracy is not required.

The clock 30 drives a programmable divider 31 which divides down this clock to generate the TDM sample clock 32. This sample clock controls transfer of data samples from the incoming packet buffer 33 to the output hold register for output as a TDM signal on link 35 to a D/A converter (not shown). This sample clock 30 also controls sampling of the input TDM 36 from the A/D converter 37 in the input sample register 38 and subsequent transfer of a TDM sample to the outgoing packet buffer 37.

The sample clock 30 defines the emptying rate of the incoming packet buffer 33 and the filling rate of the outgoing packet buffer 37. There are two incoming buffer pointers 40. One keeps track of the number of data samples received from incoming packets, and the other the number of output samples. The difference between these two values, represented by signal 41, represents the difference in rate between incoming packet data and outgoing TDM data.

Similarly, there are two outgoing buffer pointers 50 and a difference signal 51 for the sampled TDM data and outgoing packet rates.

The two signals, 41, 51 are compared and smoothed by comparator and smoother 60. This unit generates two signals which will increment (+) or decrement (-) the controlling value of the programmable counter 31. The form of the comparator and smoothing algorithm implemented by unit 60 control the dynamical behaviour of the system, source-to-source.

Alternatively, for a simpler implementation, the incoming and outgoing buffer pointers can be single pointers representing the number of incoming packets received and the number of outgoing packets awaiting transmission. The comparator function will then make a difference of these two values. The smoothing function can remain the same or be different.

Referring now to Figure 4, incoming packets are stored in a FIFO 80 consisting of a RAM 84; a receive output address pointer 77 comprising a counter; a receive incoming pointer 74 comprising a counter, and an address multiplexer 73.

Circuit 85 extracts timing information from the incoming packets. This generates a signal to increment the address for the incoming data samples to be stored in the FIFO 80 and a signal that informs the processor of the arrival of another packet 76. This can either interrupt the processor directly or be polled periodically by the processor. In this example, there are 32 samples per packet. This signal has a frequency of 250Hz and therefore will be polled.

The samples are read from FIFO 80 to the sample output register 70 at the frequency of the locally generated clock 79. The output address is incremented by this clock.

The local sample clock is generated from the terminal count output of a 10-bit programmable divide-by-N counter is driven by an 8MHz oscillator 78. The value in this counter is set by the processor, in accordance with the algorithm shown in Figure 5, and thus determines the frequency of the local sample clock. This counter value, CNTVAL, 77 is set to an initial default value of 1000, providing for a nominal 8KHz clock.

A similar FIFO 81 is used to generate the outgoing packets. Both the writing of samples into the FIFO memory and the transmitting of sample packets out of the FIFO memory are controlled by the local sample clock. The transmit clock and timing circuit 41 controls packet transmission. If there are S samples in a packet, then this circuit contains a divide-by-S counter and generates a new packet at the frequency of 8?S kHz. Thus, the outgoing packet rate is determined by the local sample clock and the number of samples per packet.

In this example, there are 32 samples per packet, and the nominal packet rate is thus 250Hz.

The algorithm executed by the processor 100 for initializing the programmable counter and receive FIFO 84 is shown in Figure 5. First at step 200, the counter 77 is initialized to 1000. The signal 76 is sampled at step p201. If two or more packets are

detected at steps 202, 203, step 204 enables the sample output. After a packet detect at step 205, step 206 reads the receive FIFO pointers and step 207 sets the receive output pointer to the incoming pointer. The counter maintains a constant 64 samples in the FIFO84, the value of RxN in Figure 4. The algorithm depends upon the processor polling signal 76 of Figure 4.

In an alternative embodiment, this signal can be used to generate a processor interrupt. When this signal indicates a new packet has been loaded, the processor receives both the Receive_FIFO_output pointer 75 and the Receive_FIFO_incoming pointer 74. It then calculates the instantaneous value of the difference between the two values, RxN. This routine is repeated continuously.

Figure 6 is the routine that generates the counter value. Its entry point 300 can be either by a periodic timer interrupt or a direct call from step 201 in Figure 5. Block 301 is a low pass filter which generates a smoothed average value for RxN, RxN'. In this system, the packets arrive at a rate of 250Hz. There will also be an unspecified amount of jitter in the arrival time of the packets. Thus, the filter will have to average the values of RxN over a period of one second.

The value of RxN' is compared with the ideal value for that system of 64 at steps 302 and 303. If the value of RxN' is greater, the local clock is too slow and the counter value, CNTVAL, must be decremented. If RxN' is less, the sample clock is too fast and CNTVAL must be incremented.

The system can be implemented in hardware or software as will be understood by one skilled in the art.

The described system can be used in any application which requires end-to-end synchronization of signals across a jitter prone network while employing minimum cost components. In accordance with the invention there is no need to exchange explicit timing information. A typical application would be a wireless transceiver.

A major advantage of this invention is there is no requirement for explicit timing information to be signaled between two end units and the end units do not need to know anything about the other end units.

THE EMBODIMENTS OF THE INVENTION IN WHICH AN EXCLUSIVE PROPERTY OR PRIVILEGE IS CLAIMED ARE DEFINED AS FOLLOWS:

1. A method of establishing communication between synchronous devices over an asynchronous communications channel, wherein each device creates, at a rate determined by a local clock associated therewith, samples of a signal to be transmitted to the other device; transmits said samples over said channel at a rate determined by its local clock; and receives samples from the other device at a rate determined by the local clock associated with the other device, and wherein at least one of said local clocks is periodically adjusted to match the rates of transmission and arrival of said samples over the channel at the device associated therewith.

2. A method of establishing communication between synchronous devices over an asynchronous communications channel, comprising at each device carrying out the steps of:

a) sampling a signal to be transmitted at a nominal rate determined by a local clock associated with the device;

b) storing successive samples of the signal to be transmitted in a transmit buffer until a predetermined number of samples have been stored and then transmitting the stored samples over the communications channel at a rate determined by the local clock of the device;

c) storing received samples in a receive buffer at a rate determined by the local clock of the other device until number of a predetermined number of received samples have been stored;

d) reconstructing a received signal from the received samples at said nominal rate determined by the local clock of the device; and

e) periodically adjusting the local clock of the device to match rate of transmission and arrival of samples in the respective buffers.

3. A method as claimed in claim 2, wherein each buffer is associated with a counter that keeps track of the number of incoming and outgoing samples, a difference signal is derived for each buffer that represents the difference between these two values, and the local clock is adjusted to match the difference signal associated with each buffer.

4. A method as claimed in claim 3, wherein said difference signals are smoothed prior to adjusting said local clock.

5. A method as claimed in claim 3, wherein each buffer is associated with a counter that keeps track of the number of samples in the buffer, and the local clock is adjusted to match the samples in each buffer.

6. A method as claimed in claim 5, wherein the count in the counter associated with each buffer is passed through a smoothing circuit prior to adjusting the local clock.

7. A synchronous communication device capable of establishing communication with another synchronous communication device over an asynchronous communications channel, comprising means for sampling a signal to be transmitted at a rate determined by a local clock associated with said device, means for transmitting samples of said signal over said channel at a rate determined by said local clock, means for receiving samples of said signal at a rate determined by the local clock of the other device, and means for adjusting at least one of the local clocks periodically to match the rates of transmission and arrival of said samples over the channel.

8. A synchronous communication device capable of establishing communication with another synchronous communication device over an asynchronous communications channel, comprising:

a) means for sampling a signal to be transmitted at a nominal rate determined by a local clock associated with the device;

b) means for storing successive samples of the signal to be transmitted in a transmit buffer until a predetermined number of samples have been stored and then transmitting the stored samples over the communications channel at a rate determined by said local clock;

c) means for storing received samples in a receive buffer until number of a predetermined number of received samples have been stored at a rate determined by a local clock associated with said another synchronous communication device;

d) means for reconstructing a received signal from the received samples at said nominal rate determined by said local clock associated with said device; and

e) means for periodically adjusting said local clock of said device to match rate of transmission and arrival of samples in the respective transmit and receive buffers.

9. A device as claimed in claim 9, further comprising a counter associated with each buffer that keeps track of the number of incoming and outgoing samples, and means for generating a difference signal for each buffer that represents the difference between these two values, said clock adjusting means adjusting the local clock to match the difference signal associated with each buffer.

10. A device as claimed in claim 9, wherein said difference signals are smoothed prior to adjusting said local clock.

11. A device as claimed in claim 10, wherein each buffer is associated with a counter that keeps track of the number of samples in the buffer, and said clock adjusting means adjusts the local clock to match the difference signal associated with each buffer.

12. A device as claimed in claim 11, wherein the count in the counter associated with each buffer is passed through a smoothing circuit prior to adjusting the local clock.

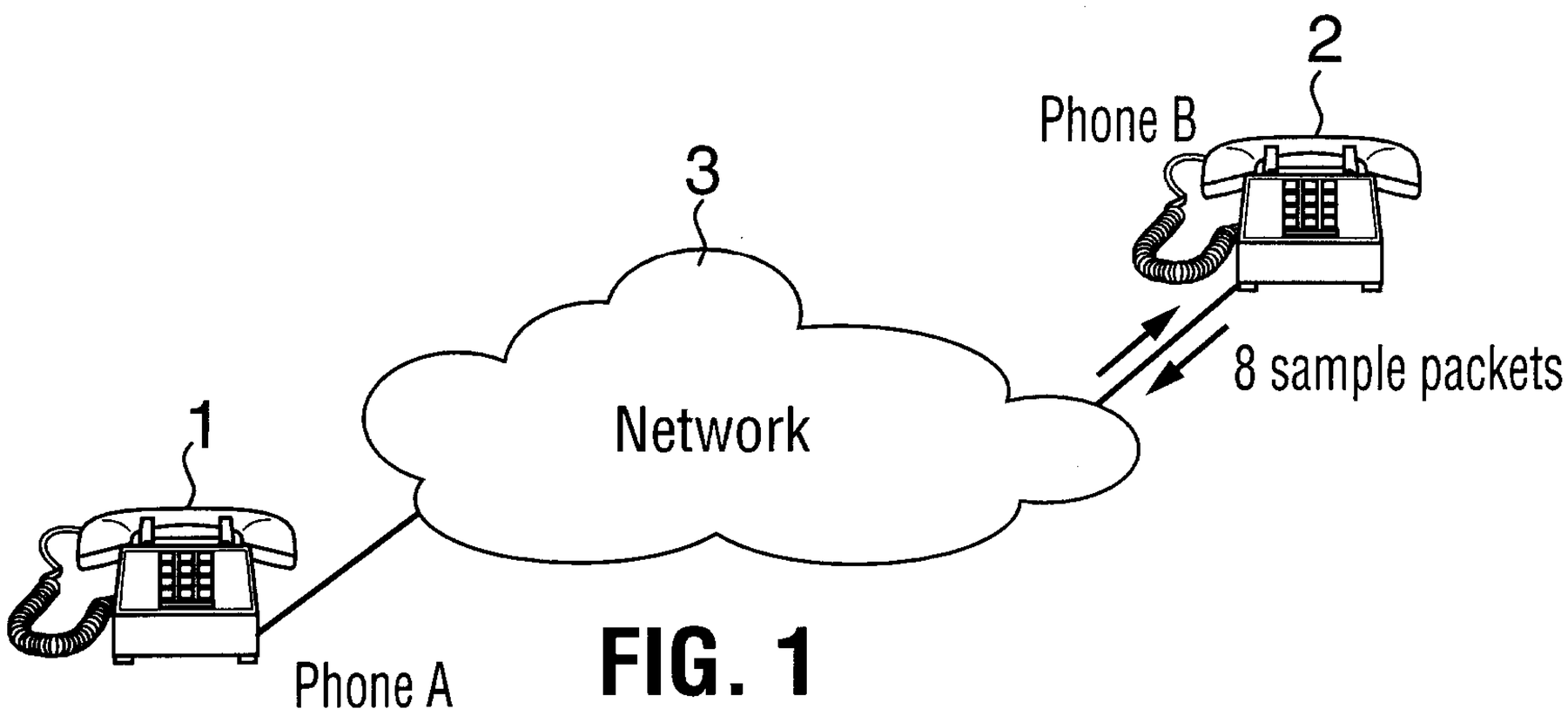

2
Phone B
3
Network
1
8 sample packets
Phone A

FIG. 1

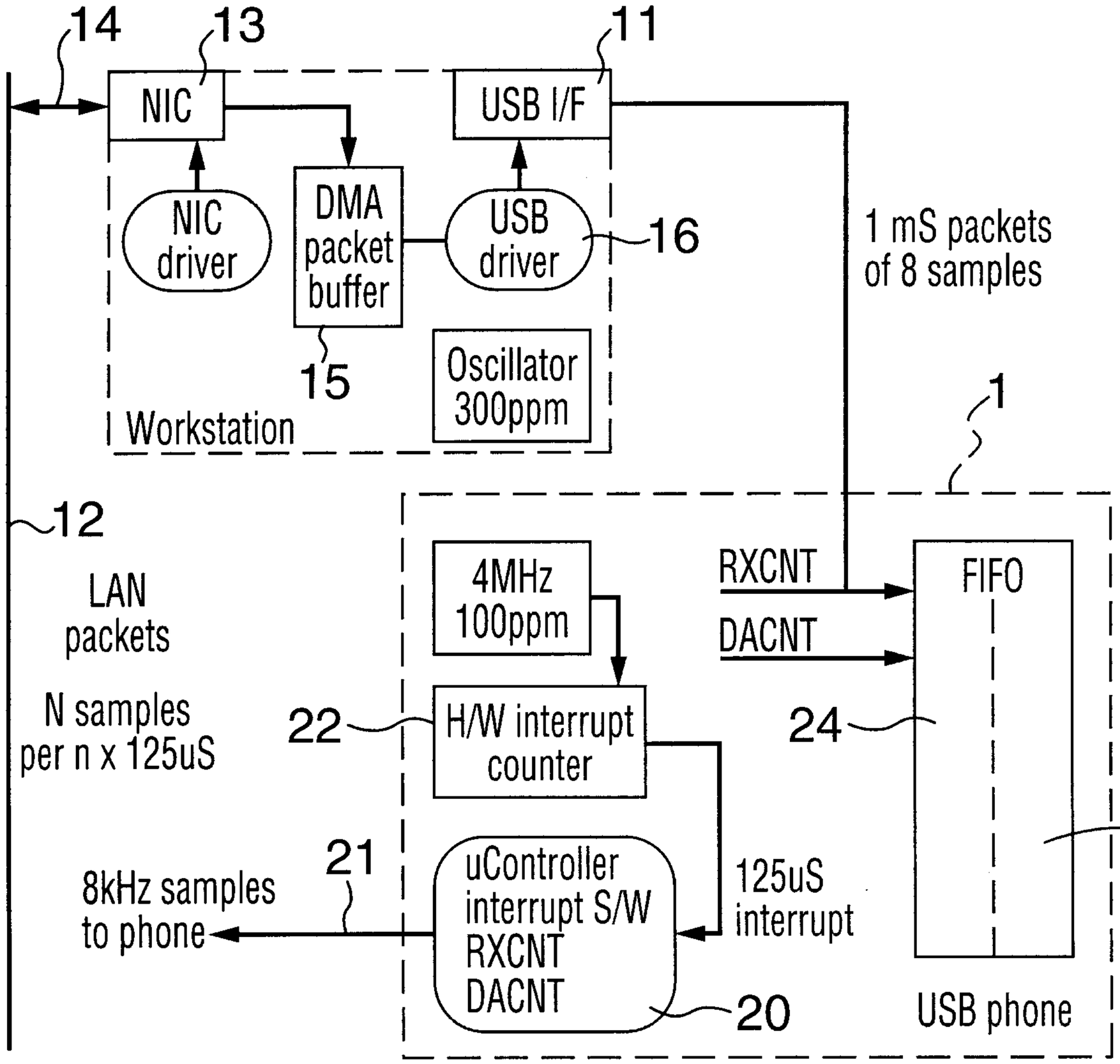

14
13
11
NIC
USB I/F
NIC driver
DMA packet buffer
USB driver
16
1 mS packets of 8 samples
15
Oscillator 300ppm
Workstation
1
12
LAN packets
N samples per n x 125uS
4MHz 100ppm
RXCNT
DACNT
FIFO
22
H/W interrupt counter
24
23
21
8kHz samples to phone
uController interrupt S/W
RXCNT
DACNT
125uS interrupt
20
USB phone

FIG. 2

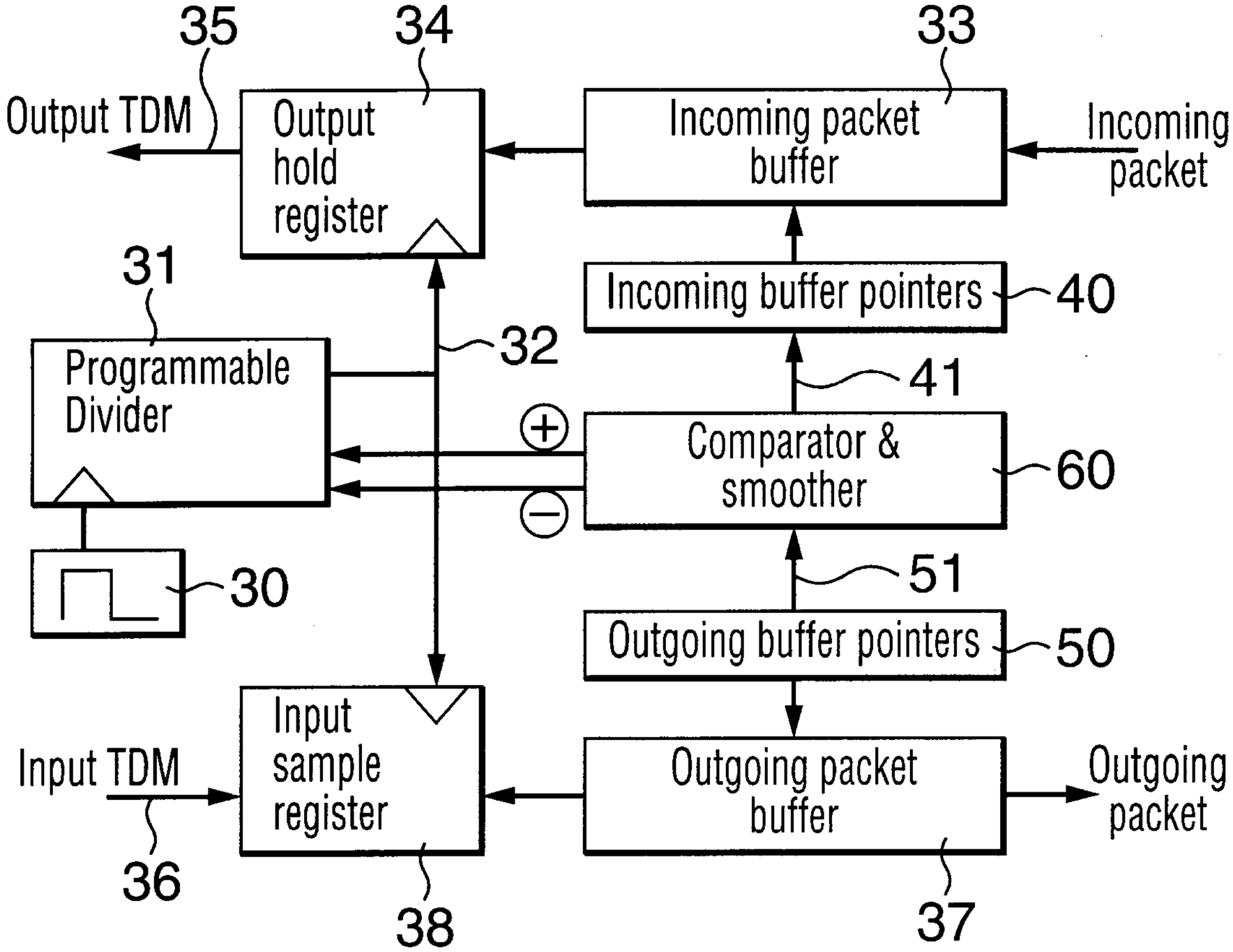

35
34
33
Output TDM
Output hold register
Incoming packet buffer
Incoming packet
31
Incoming buffer pointers
40
32
Programmable Divider
41
+
Comparator & smoother
60
−
30
51
Outgoing buffer pointers
50
Input sample register
Outgoing packet buffer
Input TDM
Outgoing packet
36
38
37

FIG. 3

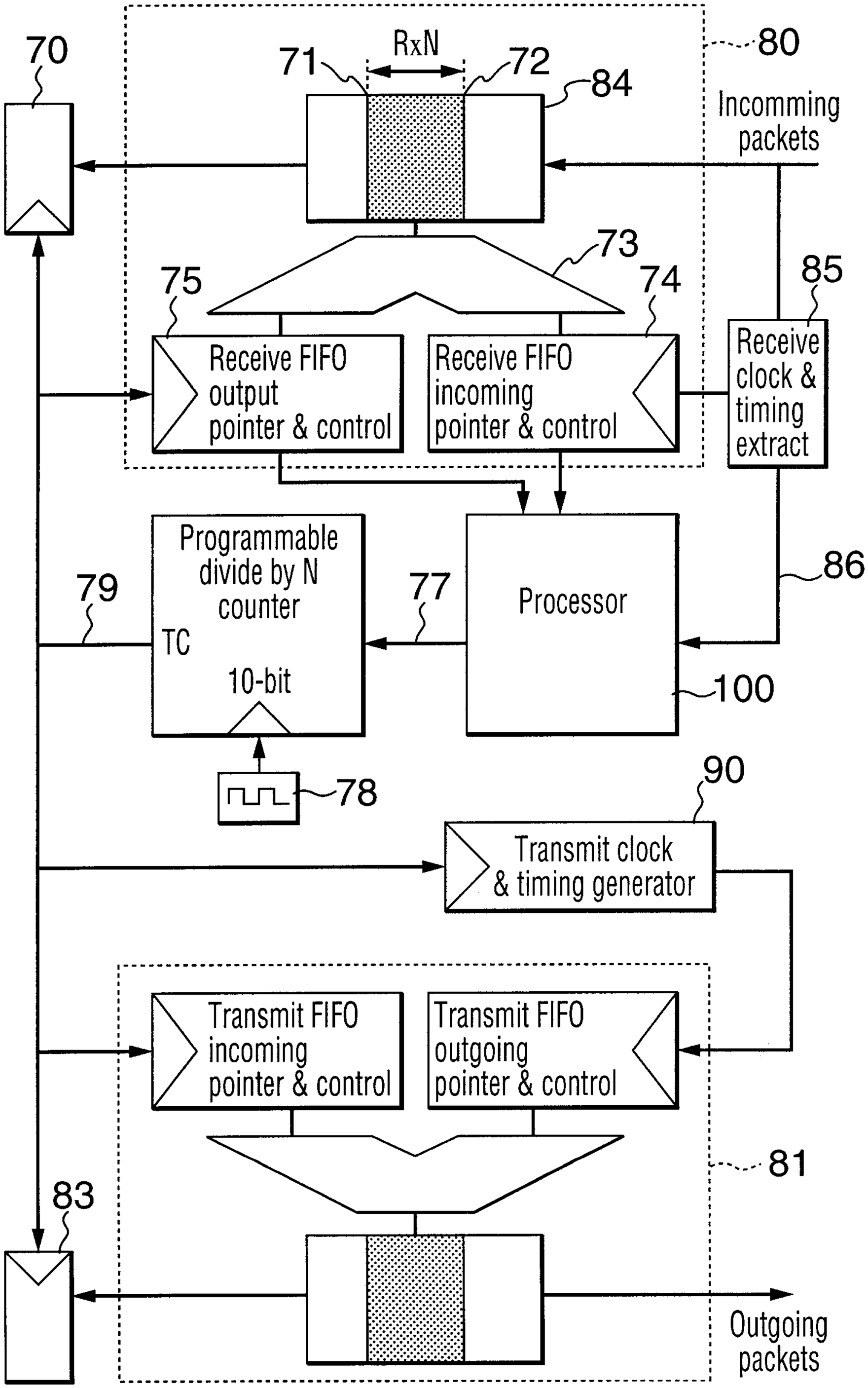

70
71
RxN
72
80
84
Incomming packets
73
75
74
85
Receive FIFO output pointer & control
Receive FIFO incoming pointer & control
Receive clock & timing extract
Programmable divide by N counter
TC
10-bit
79
77
Processor
86
100
78
90
Transmit clock & timing generator
Transmit FIFO incoming pointer & control
Transmit FIFO outgoing pointer & control
81
83
Outgoing packets

FIG. 4

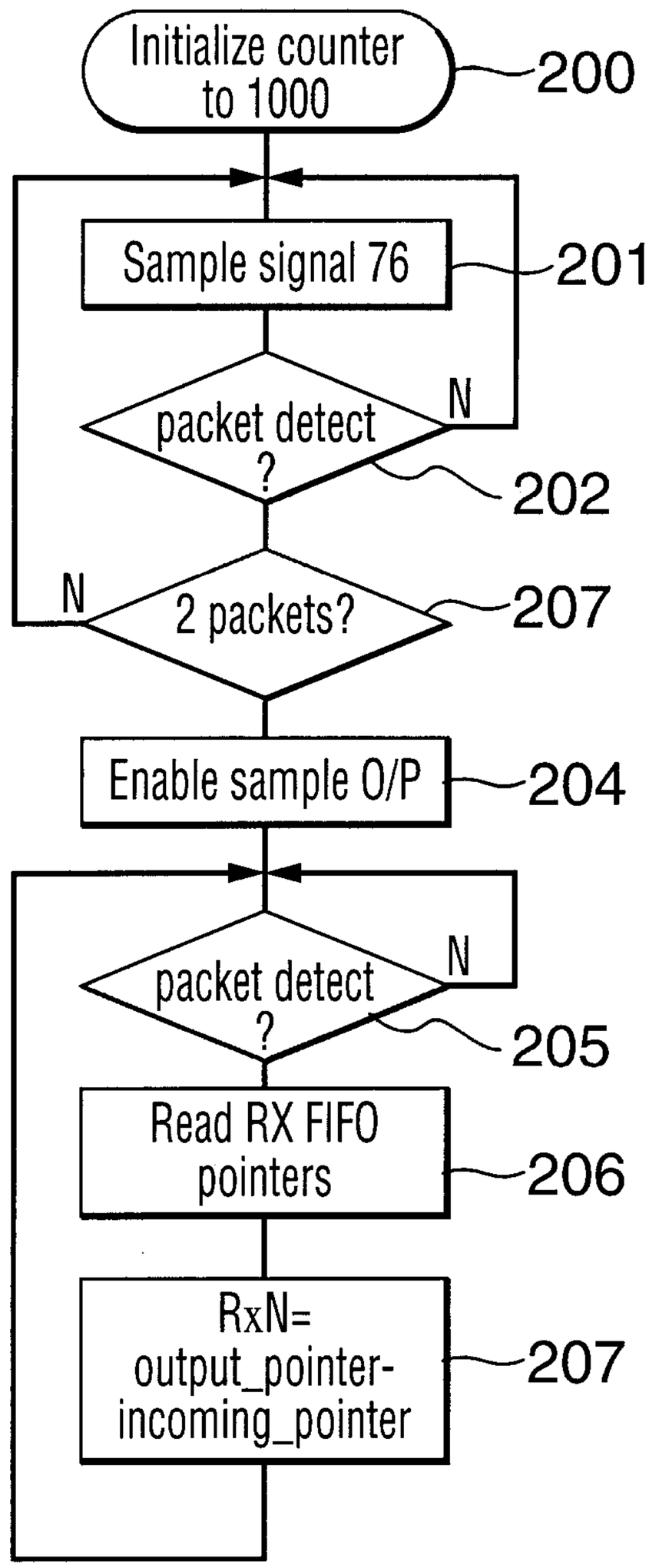

Initialize counter to 1000
200
Sample signal 76
201
packet detect ?
N
202
2 packets?
N
207
Enable sample O/P
204
packet detect ?
N
205
Read RX FIFO pointers
206
RxN= output_pointer- incoming_pointer
207

FIG. 5

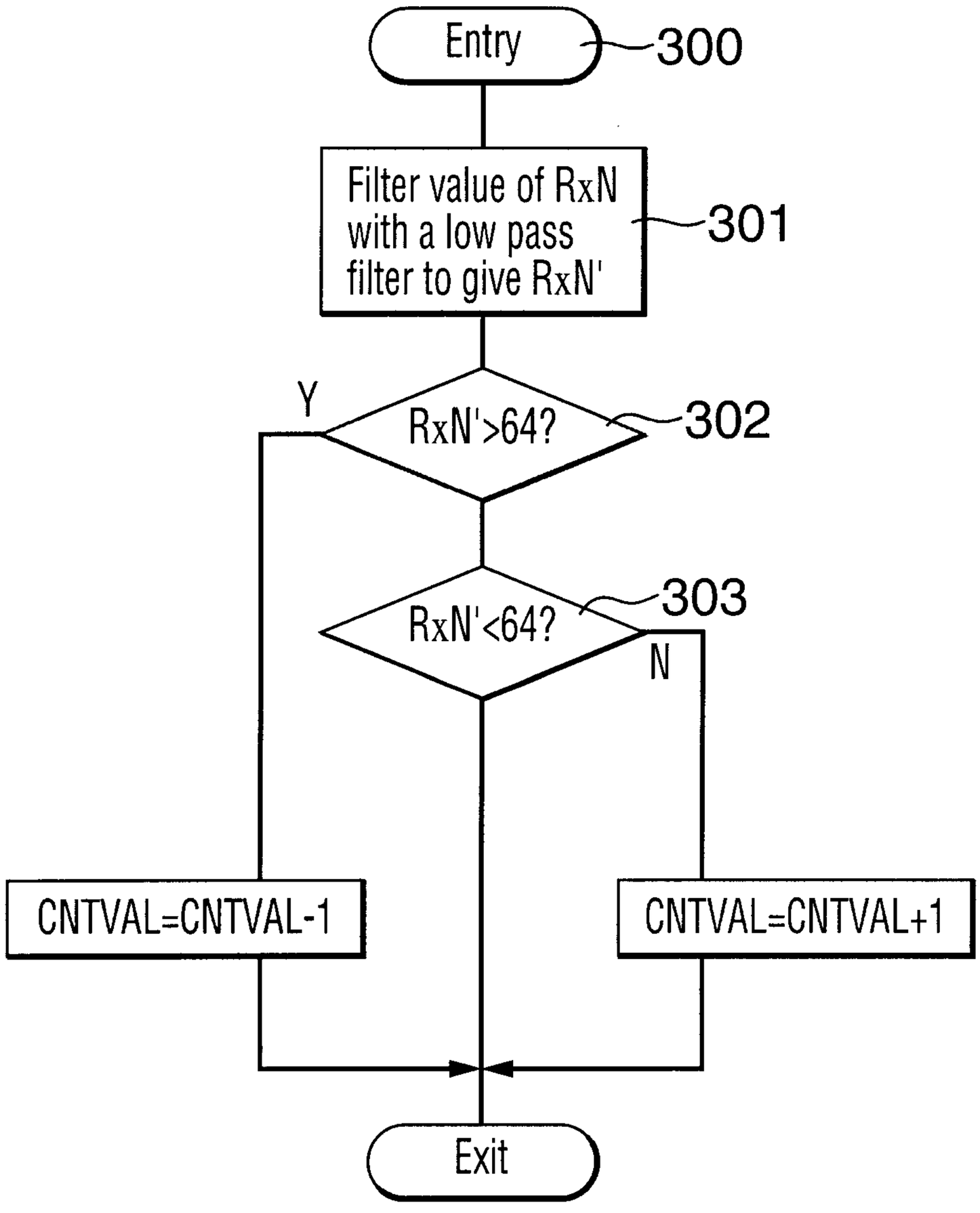
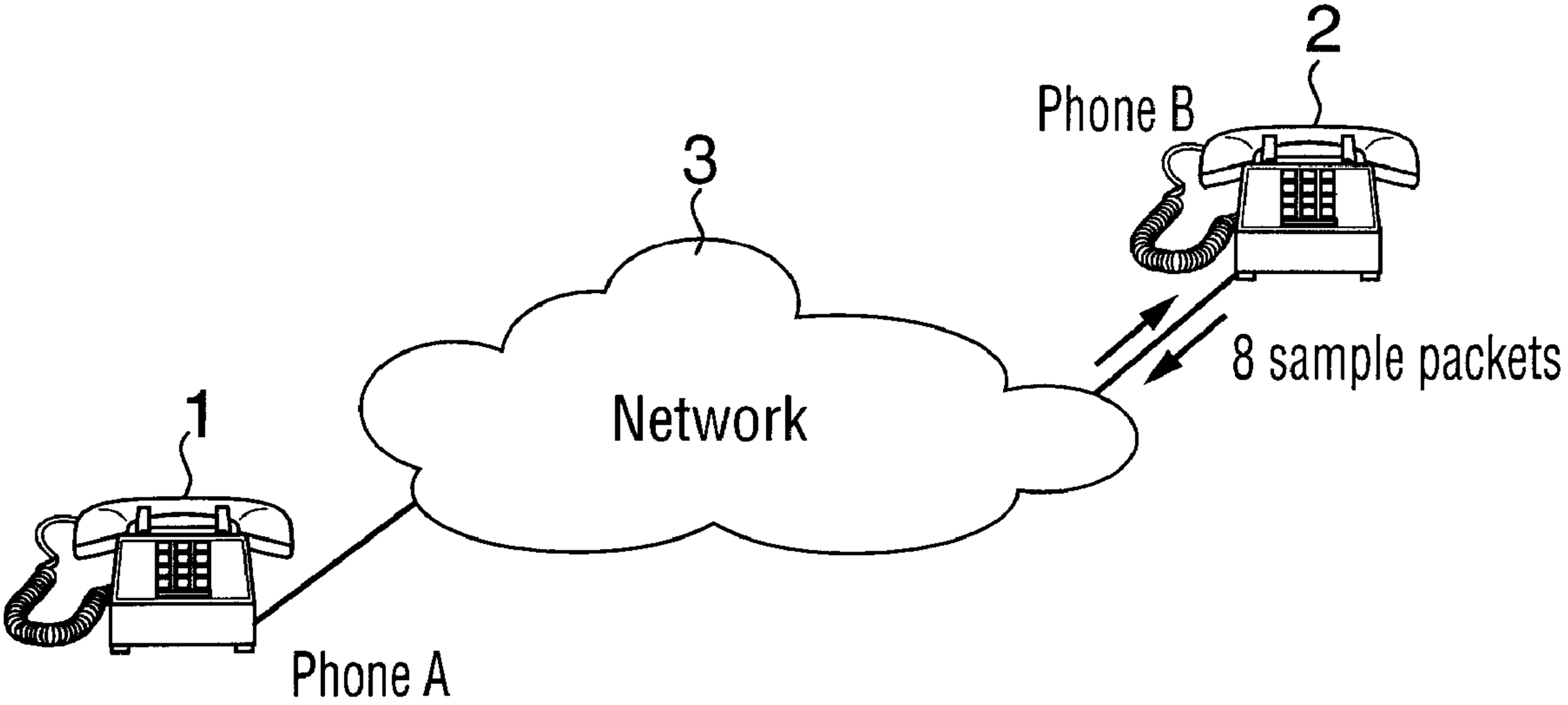



FIG. 6

Marks & Clerk


2
Phone B
3
Network
8 sample packets
1
Phone A