US007248777B2

(12) **United States Patent**
**Feininger et al.**

(10) **Patent No.: US 7,248,777 B2**
(45) **Date of Patent: Jul. 24, 2007**

(54) **METHODS AND APPARATUS TO DETECT CONTENT SKIPPING BY A CONSUMER OF A RECORDED PROGRAM**

(75) Inventors: **William A. Feininger**, Palm Harbor, FL (US); **Scott Brown**, Clearwater, FL (US)

(73) Assignee: **Nielsen Media Research, Inc.**, New York, NY (US)

( * ) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 681 days.

(21) Appl. No.: **10/483,825**

(22) PCT Filed: **Apr. 17, 2003**

(86) PCT No.: **PCT/US03/12001**

§ 371 (c)(1),
(2), (4) Date: **Jan. 13, 2004**

(87) PCT Pub. No.: **WO2004/100559**

PCT Pub. Date: **Nov. 18, 2004**

(65) **Prior Publication Data**

US 2004/0250281 A1 Dec. 9, 2004

(51) **Int. Cl.**
***H04N 5/76*** (2006.01)
***H04N 7/00*** (2006.01)
***H04N 5/00*** (2006.01)

(52) **U.S. Cl.** ..... **386/46**; 386/125

(58) **Field of Classification Search** ..... 386/46, 386/125

See application file for complete search history.

(56) **References Cited**

U.S. PATENT DOCUMENTS

| | | | | |
|---|---|---|---|---|
| 4,750,213 | A | 6/1988 | Novak | 455/67 |
| 4,752,834 | A | 6/1988 | Koombes | 358/335 |
| 4,782,401 | A | 11/1988 | Faeber et al. | 358/335 |
| 4,792,864 | A | 12/1988 | Watanabe et al. | 358/335 |
| 4,888,638 | A | 12/1989 | Bohn | 358/84 |
| 4,931,865 | A | 6/1990 | Scarampi | 358/84 |
| 4,943,963 | A | 7/1990 | Waechter et al. | 370/94.1 |
| 5,270,829 | A * | 12/1993 | Yang | 386/83 |

(Continued)

FOREIGN PATENT DOCUMENTS

| | | |
|---|---|---|
| CA | 2323539 | 10/1999 |

(Continued)

OTHER PUBLICATIONS

International Preliminary Examination Report for PCT/US03/12001 dated Dec. 15, 2003.

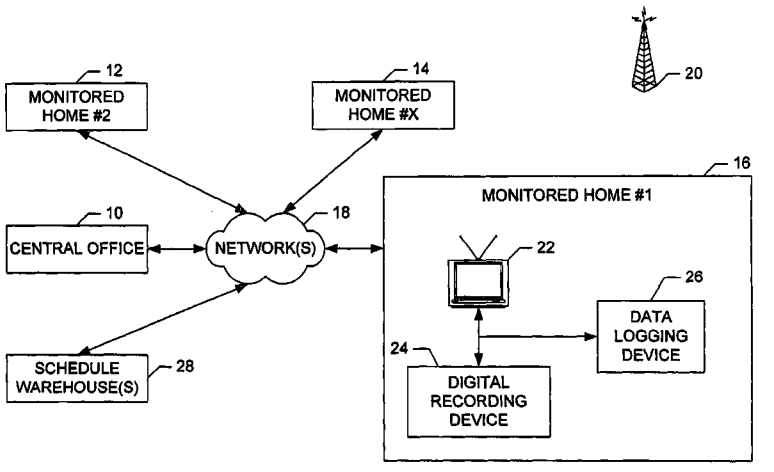
(Continued)

*Primary Examiner*—Thai Q. Tran
*Assistant Examiner*—Nigar Chowdhury
(74) *Attorney, Agent, or Firm*—Hanley, Flight & Zimmerman, LLC

(57) **ABSTRACT**

Methods and apparatus to detect content skipping by a consumer of a recorded program are disclosed. In a disclosed method, a plurality of recording times of sections of a recorded program are stored in association with playback times of the recorded program. Playback intervals between sequential pairs of the playback times are compared with corresponding recording intervals between sequential pairs of the recording times to determine if any portion of the recorded program was not played back in real time.

**79 Claims, 18 Drawing Sheets**




Roku EX1005
IPR2024-01056
U.S. Patent No. 10,572,896

U.S. PATENT DOCUMENTS

| | | | | |
|---|---|---|---|---|
| 5,333,091 | A | 7/1994 | Iggulden et al. | 360/14.1 |
| 5,481,296 | A | 1/1996 | Cragun et al. | |
| 5,521,631 | A | 5/1996 | Budow et al. | |
| 5,535,209 | A | 7/1996 | Glaser et al. | 370/84 |
| 5,661,526 | A | 8/1997 | Hamamoto et al. | |
| 5,699,370 | A | 12/1997 | Kaniwa et al. | 371/57.1 |
| 5,703,655 | A | 12/1997 | Corey et al. | 348/468 |
| 5,819,156 | A | 10/1998 | Belmont | |
| 5,999,689 | A | 12/1999 | Iggulden | |
| 6,002,443 | A | 12/1999 | Iggulden | |
| 6,184,918 | B1 | 2/2001 | Goldschmidt Iki et al. | 348/1 |
| 6,233,389 | B1 | 5/2001 | Barton et al. | |
| 6,343,179 | B1 | 1/2002 | Suito et al. | |
| 6,404,977 | B1 | 6/2002 | Iggulden | 386/46 |
| 6,513,161 | B2 | 1/2003 | Wheeler et al. | 725/14 |
| 6,850,691 | B1 | 2/2005 | Stam et al. | |
| 7,057,986 | B2 * | 6/2006 | Fukuchi | 369/47.31 |
| 2002/0034374 | A1 | 3/2002 | Barton | |
| 2002/0056087 | A1 | 5/2002 | Berezowski et al. | |
| 2002/0090198 | A1 | 7/2002 | Rosenberg et al. | |
| 2002/0120925 | A1 | 8/2002 | Logan | |
| 2003/0154128 | A1 | 8/2003 | Liga et al. | 705/14 |
| 2004/0250281 | A1 | 12/2004 | Feininger | |
| 2005/0138658 | A1 | 6/2005 | Bryan | |

FOREIGN PATENT DOCUMENTS

| | | |
|---|---|---|
| EP | 1183689 | 3/2002 |
| EP | 1254561 | 11/2002 |
| WO | WO 01/47279 | 6/2001 |
| WO | WO 0147238 | 6/2001 |
| WO | WO2004051997 | 6/2004 |
| WO | WO2006014495 | 2/2006 |

OTHER PUBLICATIONS

International Search Report for PCT/US03/12001 dated Jun. 27, 2003.

* cited by examiner

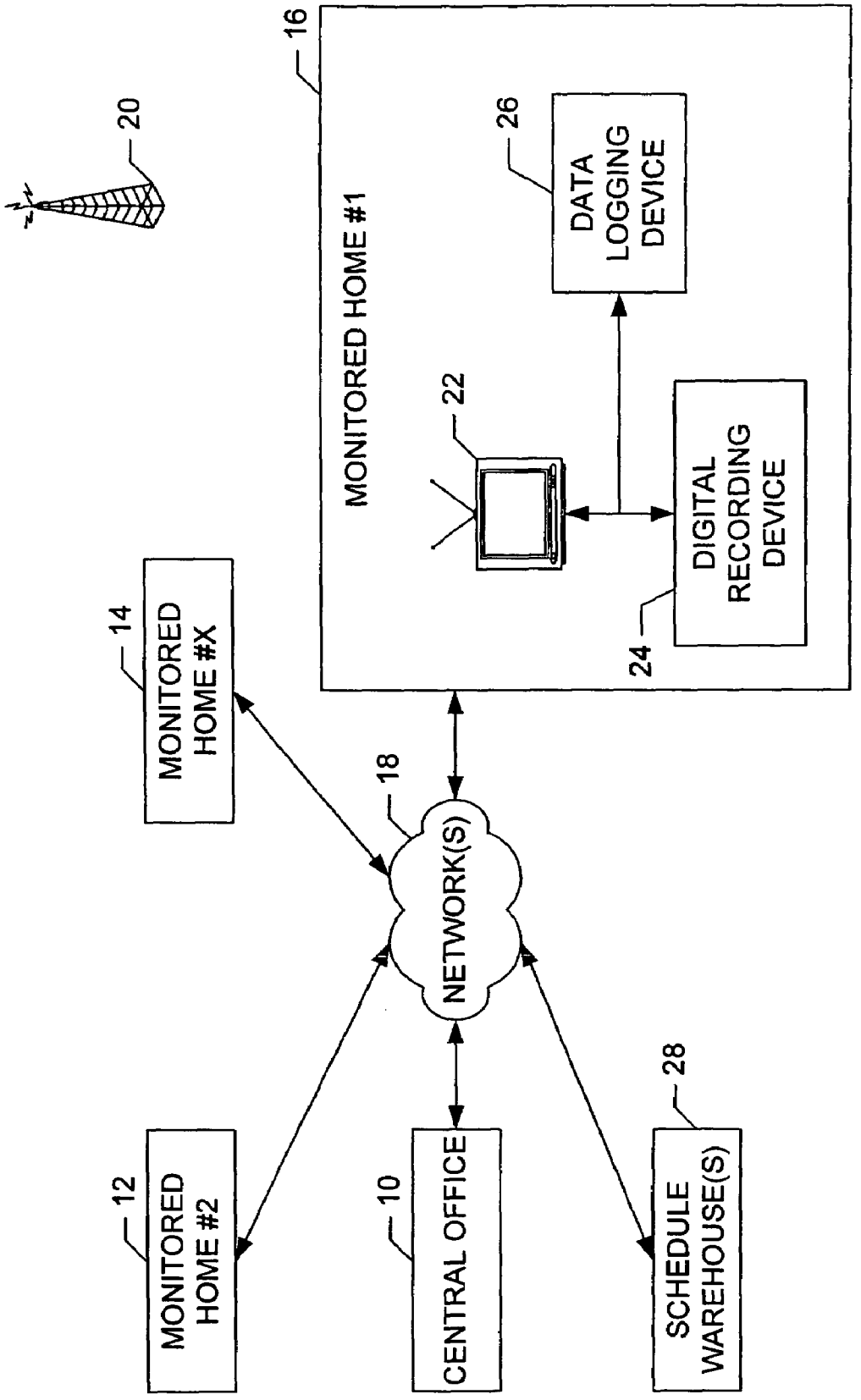

16
MONITORED HOME #1
26
DATA LOGGING DEVICE
22
DIGITAL RECORDING DEVICE
24
20
14
MONITORED HOME #X
18
NETWORK(S)
12
MONITORED HOME #2
10
CENTRAL OFFICE
28
SCHEDULE WAREHOUSE(S)

FIG. 1

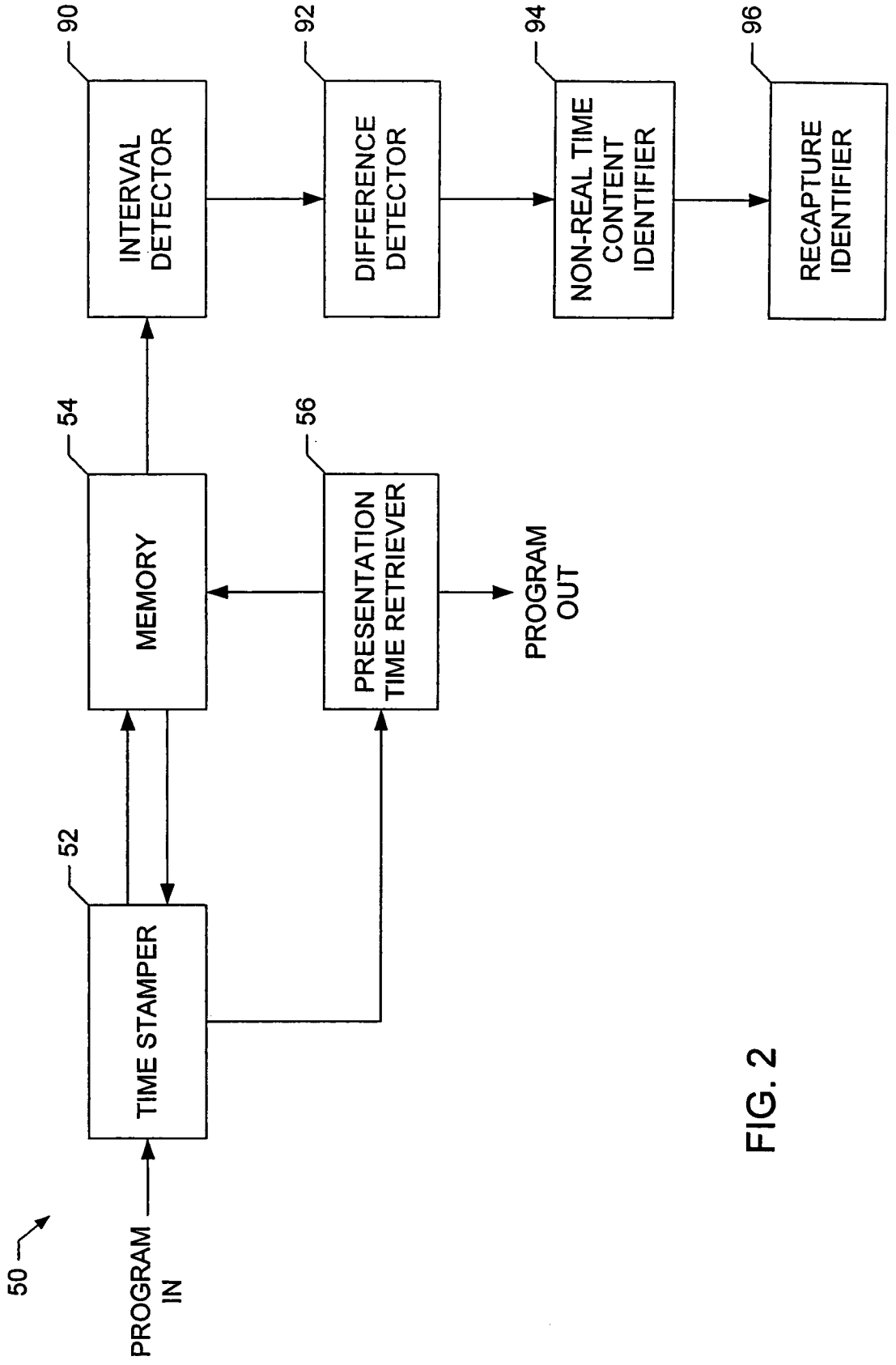

50
PROGRAM IN
52
TIME STAMPER
54
MEMORY
56
PRESENTATION TIME RETRIEVER
PROGRAM OUT
90
INTERVAL DETECTOR
92
DIFFERENCE DETECTOR
94
NON-REAL TIME CONTENT IDENTIFIER
96
RECAPTURE IDENTIFIER

FIG. 2

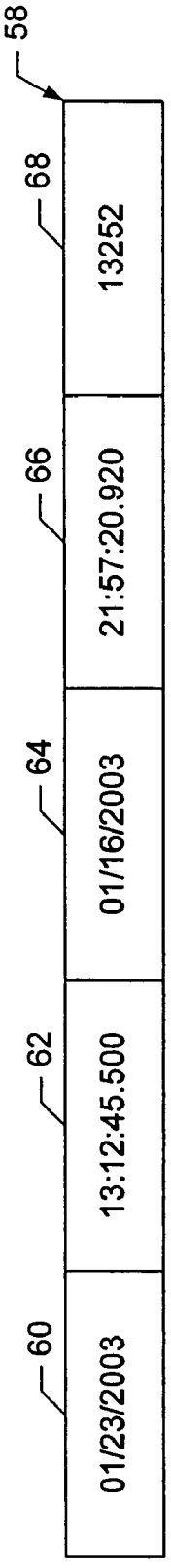



FIG. 3A

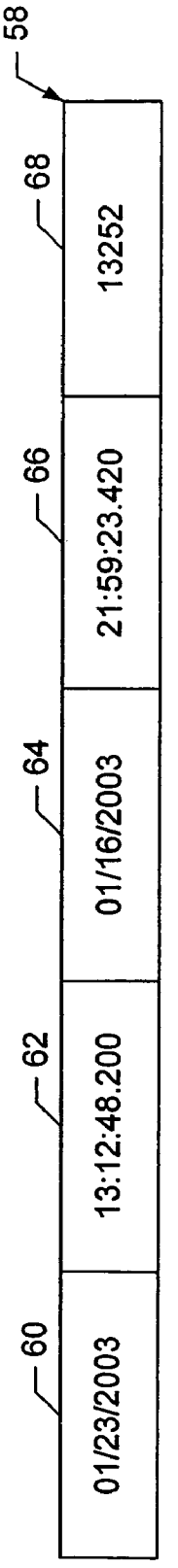



FIG. 3B

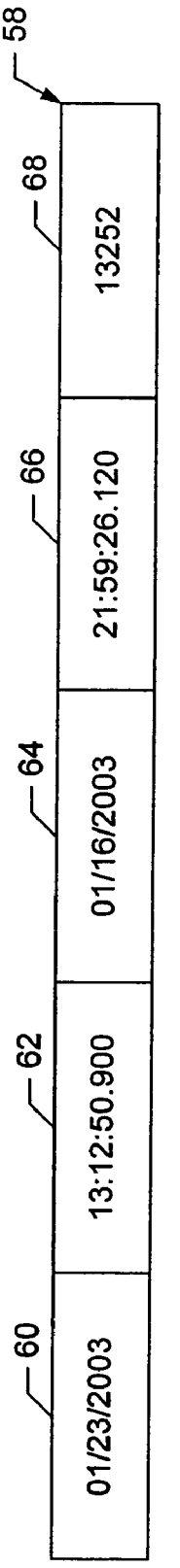



FIG. 3C

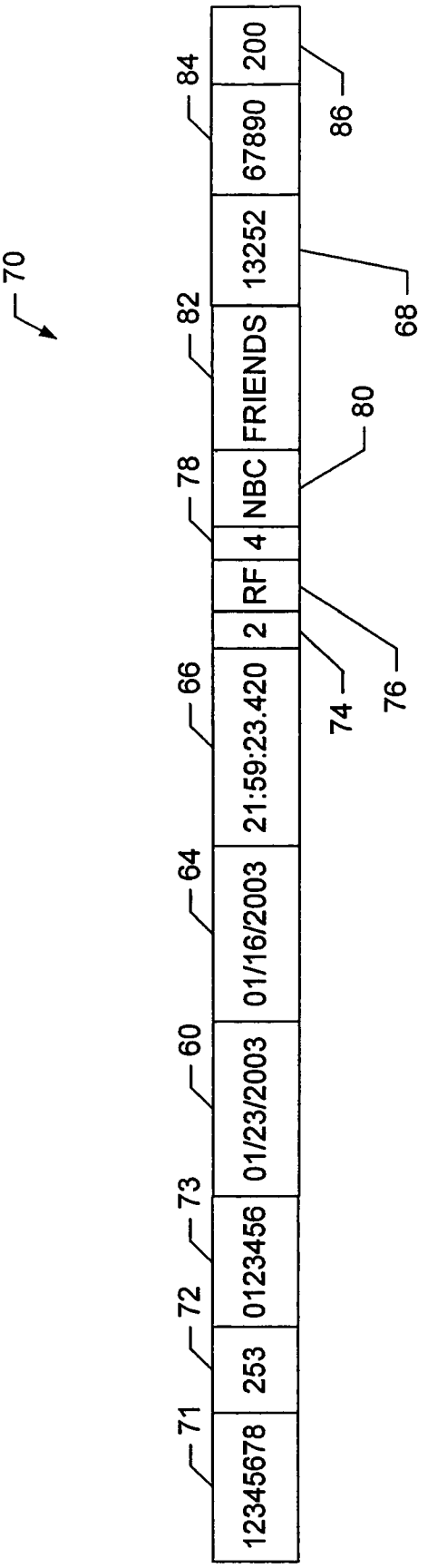

70
71
12345678
72
253
73
0123456
60
01/23/2003
64
01/16/2003
66
21:59:23.420
74
2
76
RF
4
78
NBC
80
82
FRIENDS
68
13252
84
67890
86
200

FIG. 4

FIG. 5

| NAME | START TIME | END TIME |
|---|---|---|
| INTRODUCTION | 00:00:00.000 | 00:03:30.003 |
| COMMERCIAL 1 | 00:03:30.003 | 00:06:01.015 |
| CONTENT 1 | 00:06:01.015 | 00:13:35.920 |
| COMMERCIAL 2 | 00:13:35.920 | 00:15:55.723 |
| CONTENT 2 | 00:15:55.723 | 00:20:43.513 |
| COMMERCIAL 3 | 00:20:43.513 | 00:23:53.819 |
| CONTENT 3 | 00:23:53.819 | 00:28:09.875 |
| ... | ... | ... |
| COMMERCIAL N | 00:57:20.920 | 00:59:18.562 |
| ... | ... | ... |

FIG. 6

| ADVANCED SEGMENT NAME(S) | REWOUND SEGMENT NAME(S) |
|---|---|
| 1 | 4 |
| 5 | 5 |
| 8 | 6 |
| 8 | 7 |
| 9 | 8 |
| | 9 |

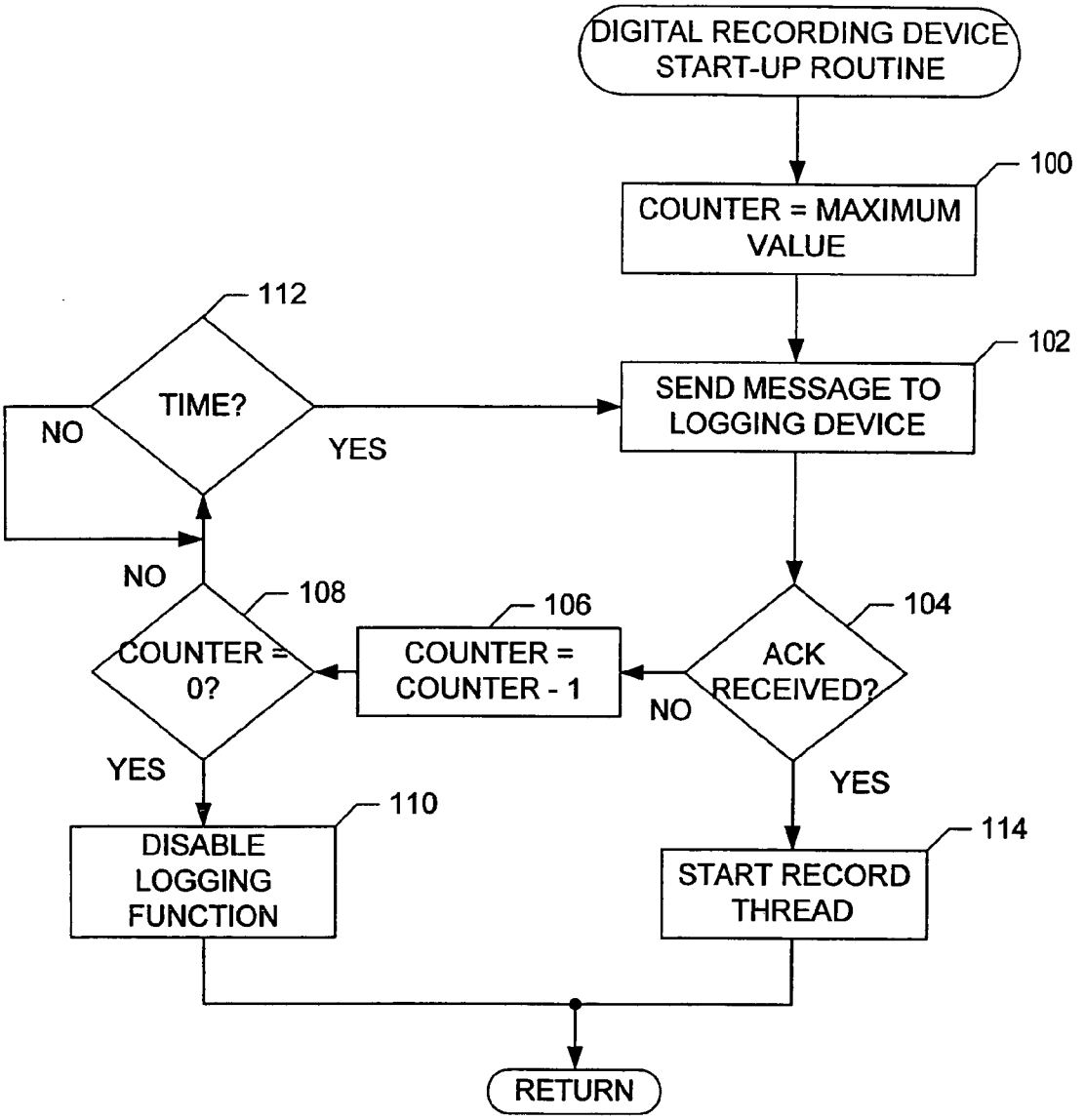

DIGITAL RECORDING DEVICE START-UP ROUTINE
100
COUNTER = MAXIMUM VALUE
102
SEND MESSAGE TO LOGGING DEVICE
104
ACK RECEIVED?
NO
YES
106
COUNTER = COUNTER - 1
108
COUNTER = 0?
NO
YES
112
TIME?
NO
YES
110
DISABLE LOGGING FUNCTION
114
START RECORD THREAD
RETURN

FIG. 7

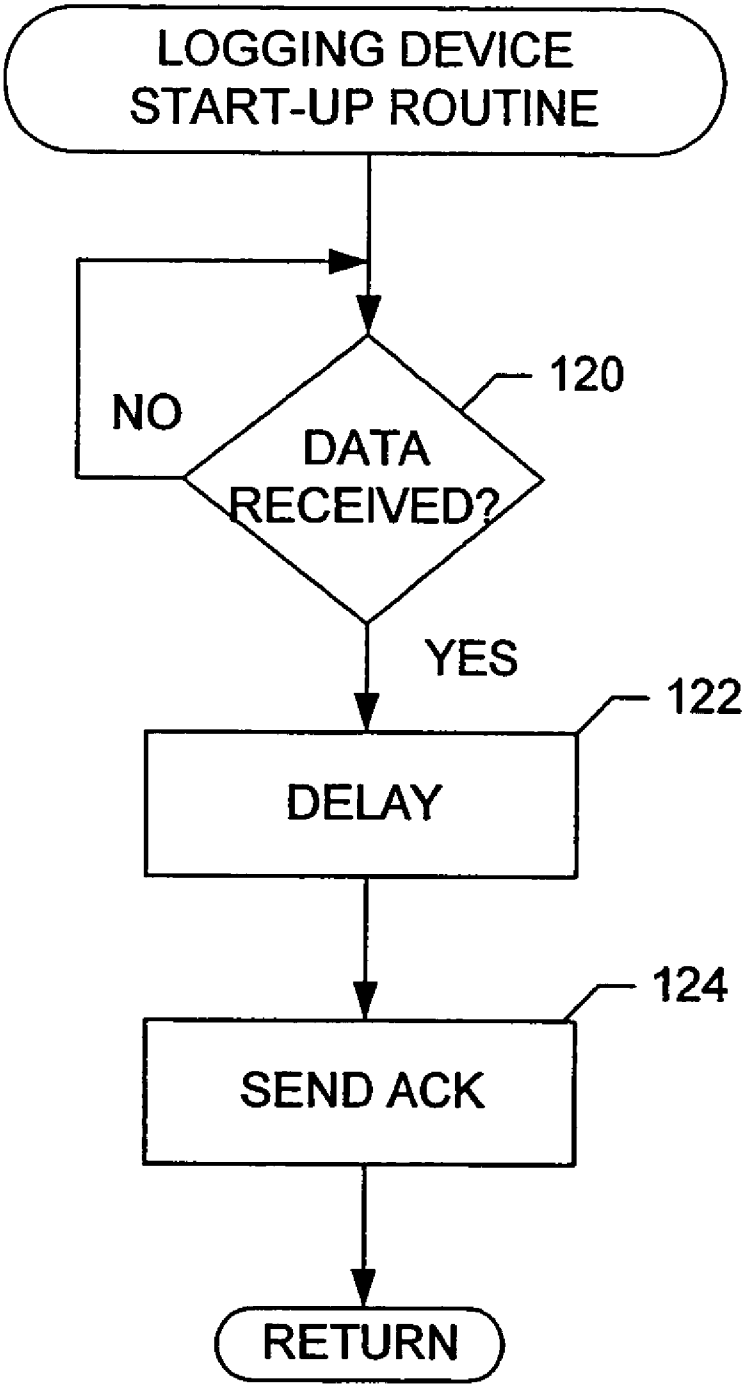

LOGGING DEVICE START-UP ROUTINE
120
NO
DATA RECEIVED?
YES
122
DELAY
124
SEND ACK
RETURN

FIG. 8

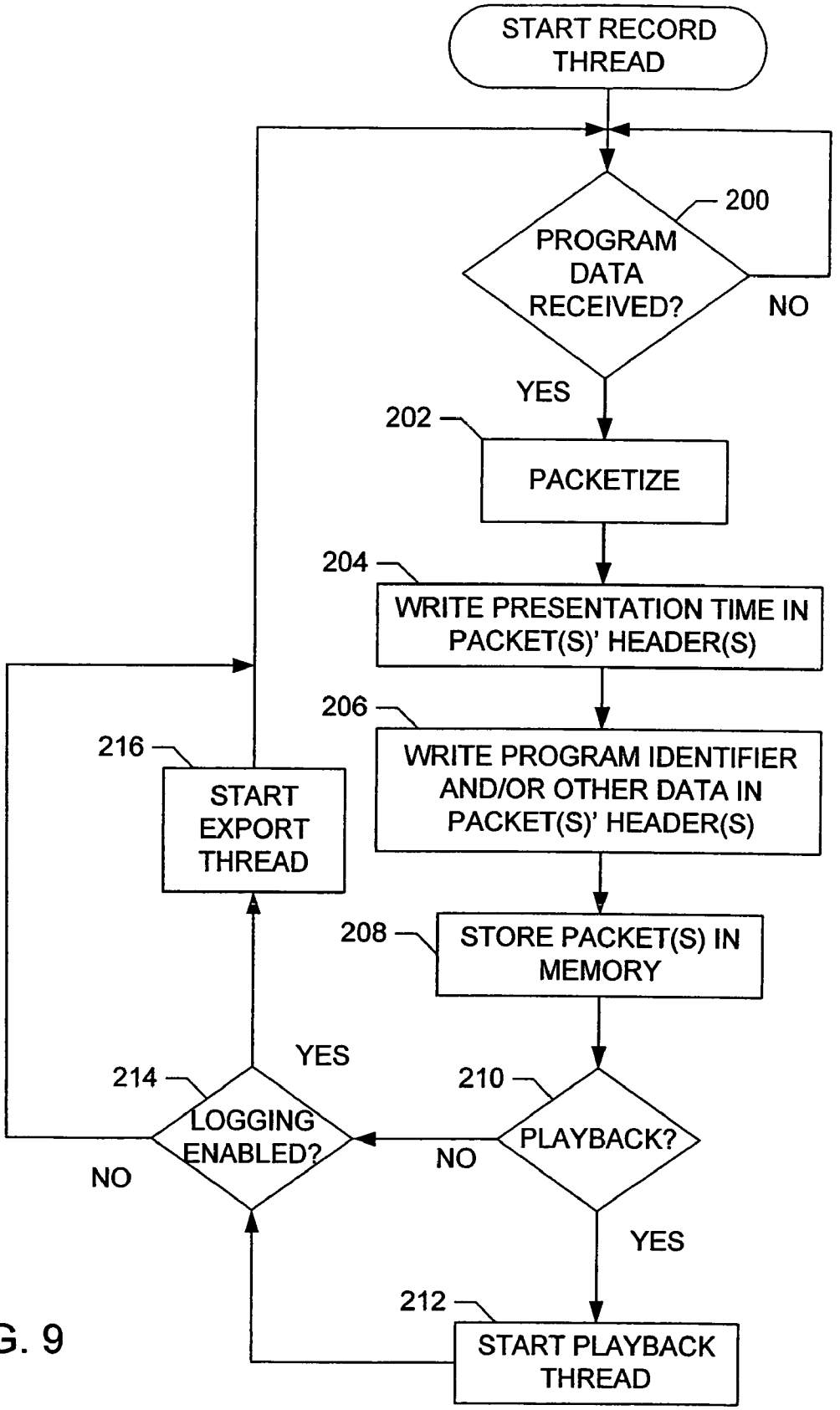

START RECORD THREAD
200
PROGRAM DATA RECEIVED?
NO
YES
202
PACKETIZE
204
WRITE PRESENTATION TIME IN PACKET(S)' HEADER(S)
206
WRITE PROGRAM IDENTIFIER AND/OR OTHER DATA IN PACKET(S)' HEADER(S)
208
STORE PACKET(S) IN MEMORY
210
PLAYBACK?
NO
YES
212
START PLAYBACK THREAD
214
LOGGING ENABLED?
YES
NO
216
START EXPORT THREAD

FIG. 9

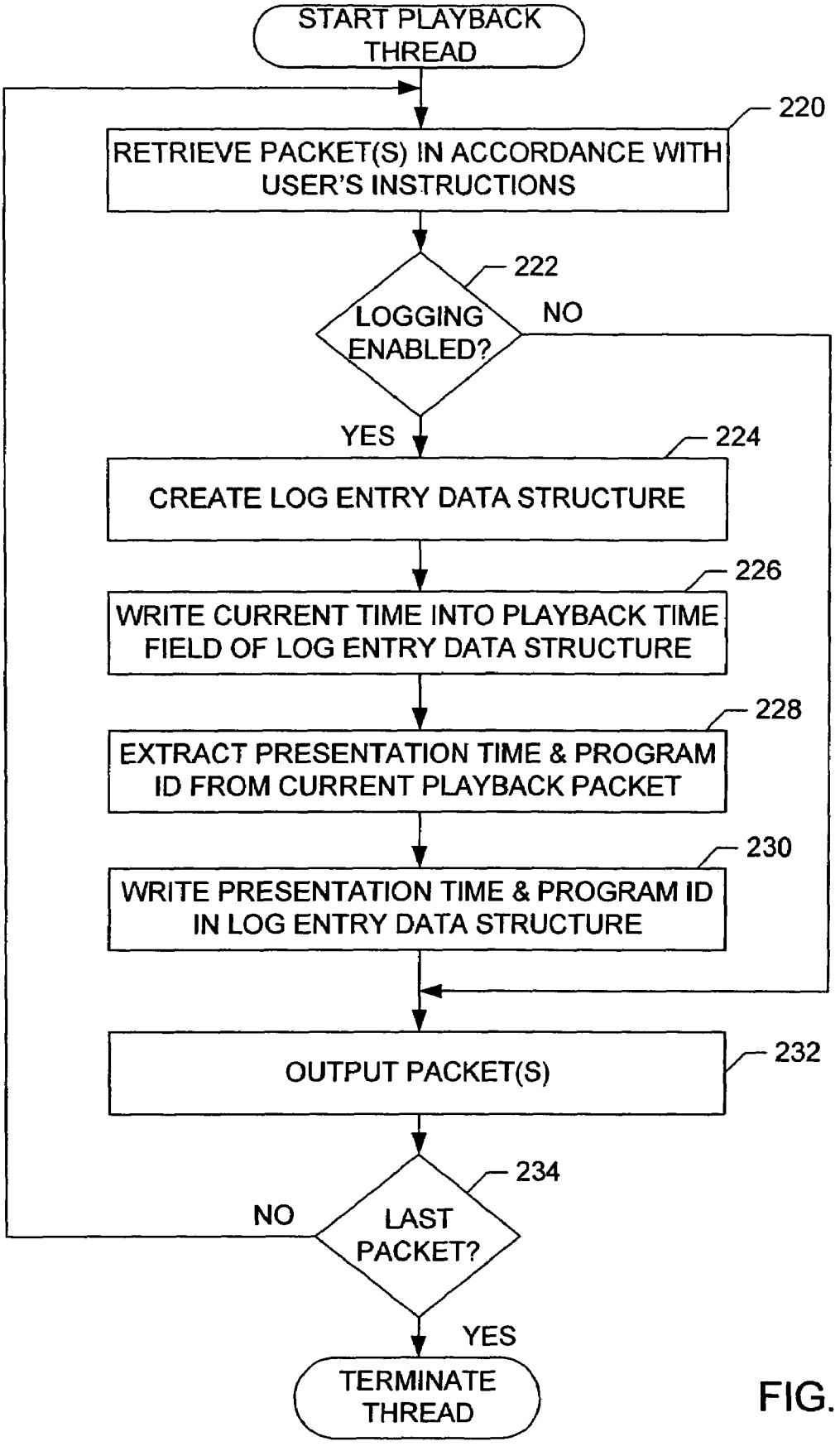

START PLAYBACK THREAD
220
RETRIEVE PACKET(S) IN ACCORDANCE WITH USER'S INSTRUCTIONS
222
LOGGING ENABLED?
NO
YES
224
CREATE LOG ENTRY DATA STRUCTURE
226
WRITE CURRENT TIME INTO PLAYBACK TIME FIELD OF LOG ENTRY DATA STRUCTURE
228
EXTRACT PRESENTATION TIME & PROGRAM ID FROM CURRENT PLAYBACK PACKET
230
WRITE PRESENTATION TIME & PROGRAM ID IN LOG ENTRY DATA STRUCTURE
OUTPUT PACKET(S)
232
234
NO
LAST PACKET?
YES
TERMINATE THREAD

FIG. 10

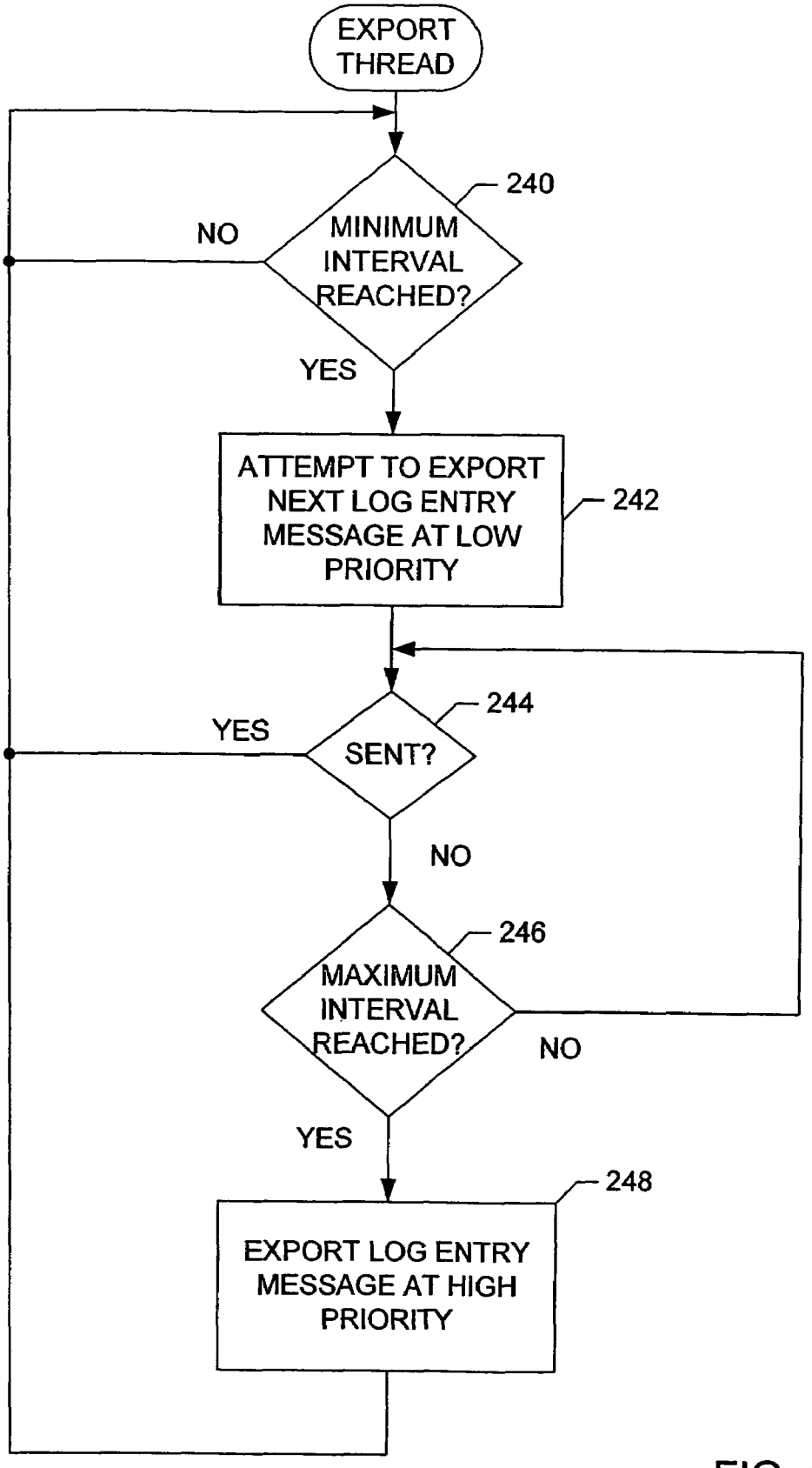

EXPORT THREAD
240
MINIMUM INTERVAL REACHED?
NO
YES
ATTEMPT TO EXPORT NEXT LOG ENTRY MESSAGE AT LOW PRIORITY
242
244
SENT?
YES
NO
246
MAXIMUM INTERVAL REACHED?
NO
YES
248
EXPORT LOG ENTRY MESSAGE AT HIGH PRIORITY

FIG. 11

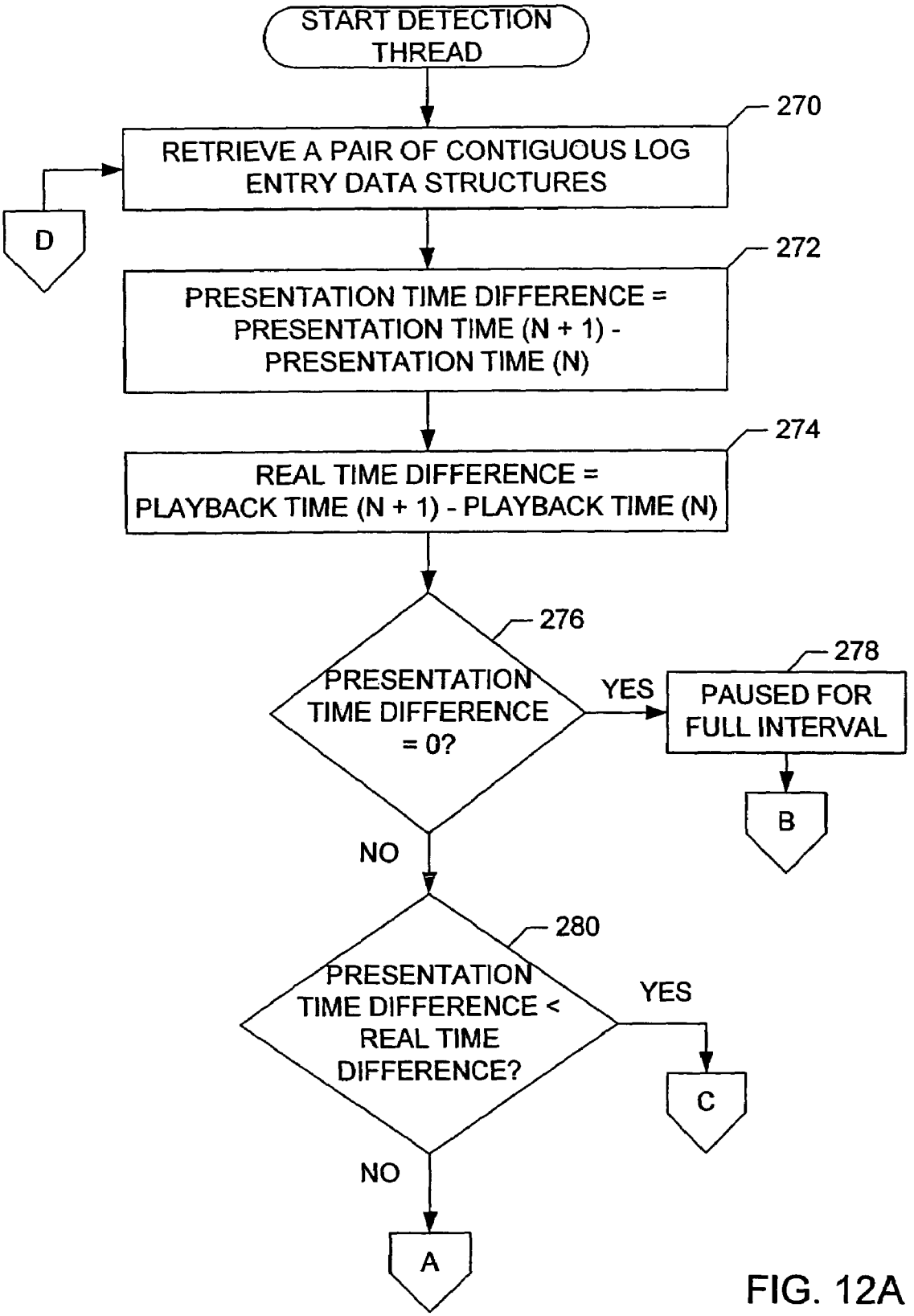

START DETECTION THREAD
270
RETRIEVE A PAIR OF CONTIGUOUS LOG ENTRY DATA STRUCTURES
D
272
PRESENTATION TIME DIFFERENCE = PRESENTATION TIME (N + 1) - PRESENTATION TIME (N)
274
REAL TIME DIFFERENCE = PLAYBACK TIME (N + 1) - PLAYBACK TIME (N)
276
PRESENTATION TIME DIFFERENCE = 0?
YES
278
PAUSED FOR FULL INTERVAL
B
NO
280
PRESENTATION TIME DIFFERENCE < REAL TIME DIFFERENCE?
YES
C
NO
A

FIG. 12A

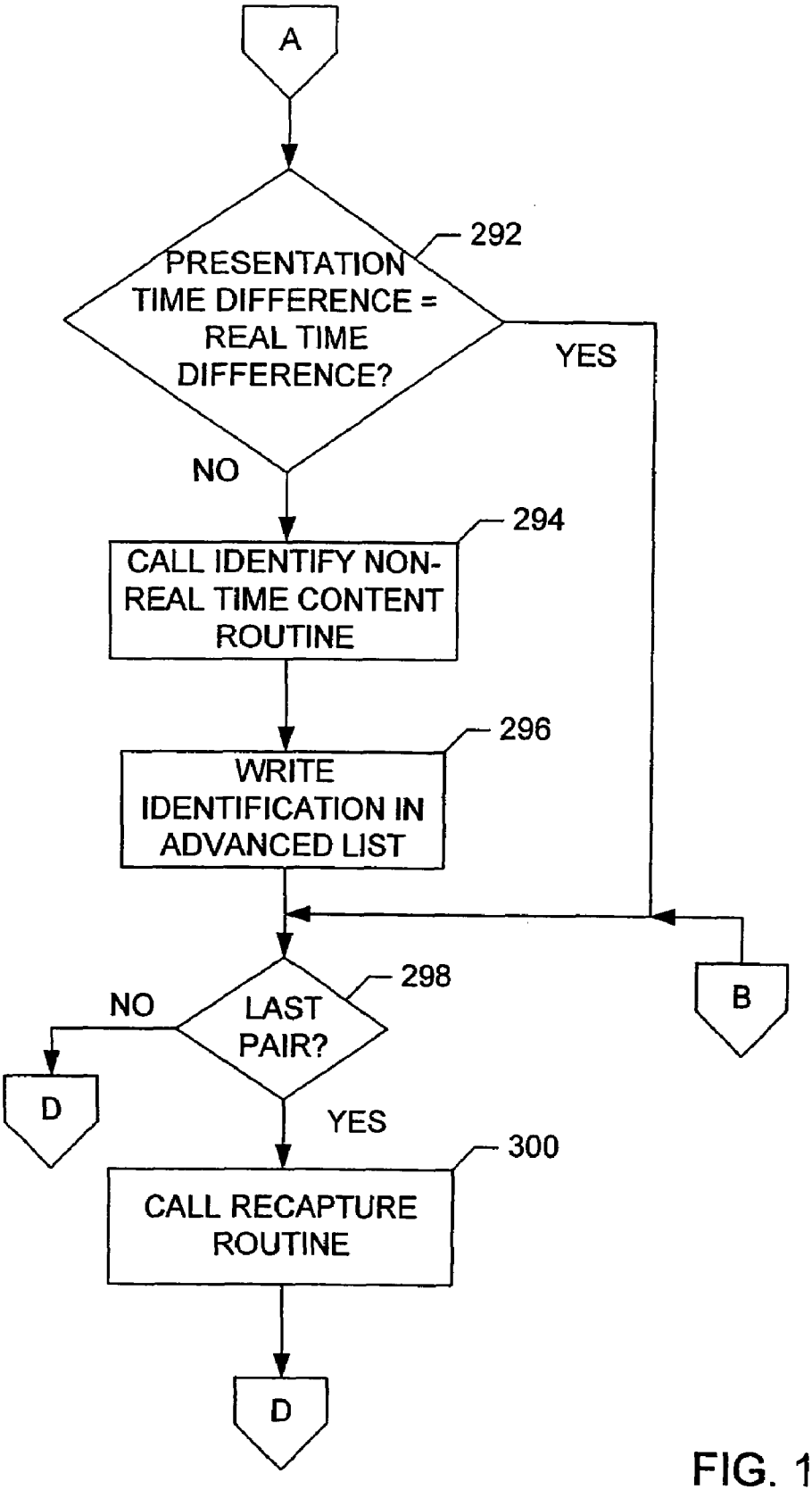

A
PRESENTATION TIME DIFFERENCE = REAL TIME DIFFERENCE?
292
YES
NO
CALL IDENTIFY NON-REAL TIME CONTENT ROUTINE
294
WRITE IDENTIFICATION IN ADVANCED LIST
296
B
LAST PAIR?
298
NO
D
YES
CALL RECAPTURE ROUTINE
300
D

FIG. 12B

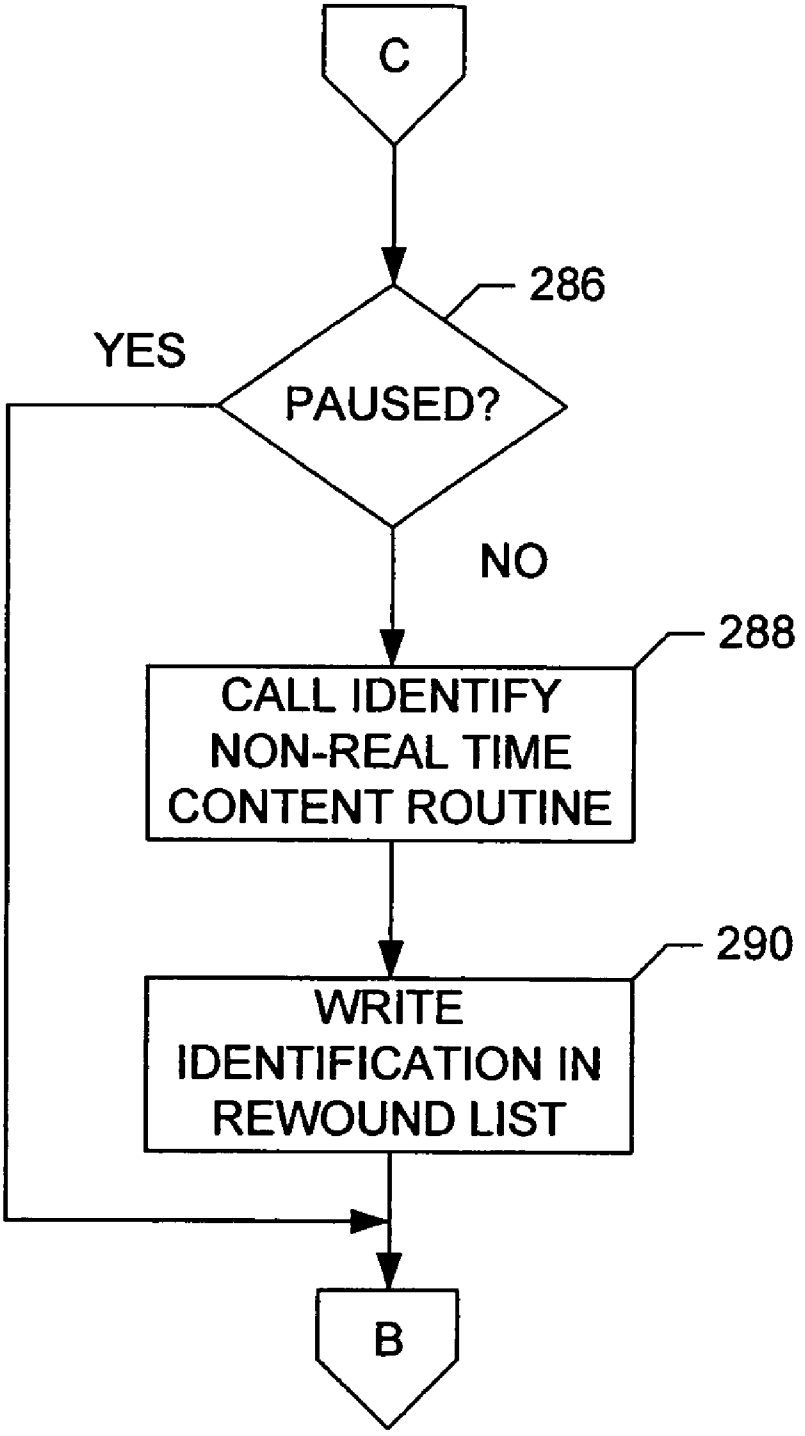

C
286
YES
PAUSED?
NO
288
CALL IDENTIFY NON-REAL TIME CONTENT ROUTINE
290
WRITE IDENTIFICATION IN REWOUND LIST
B

FIG. 12C

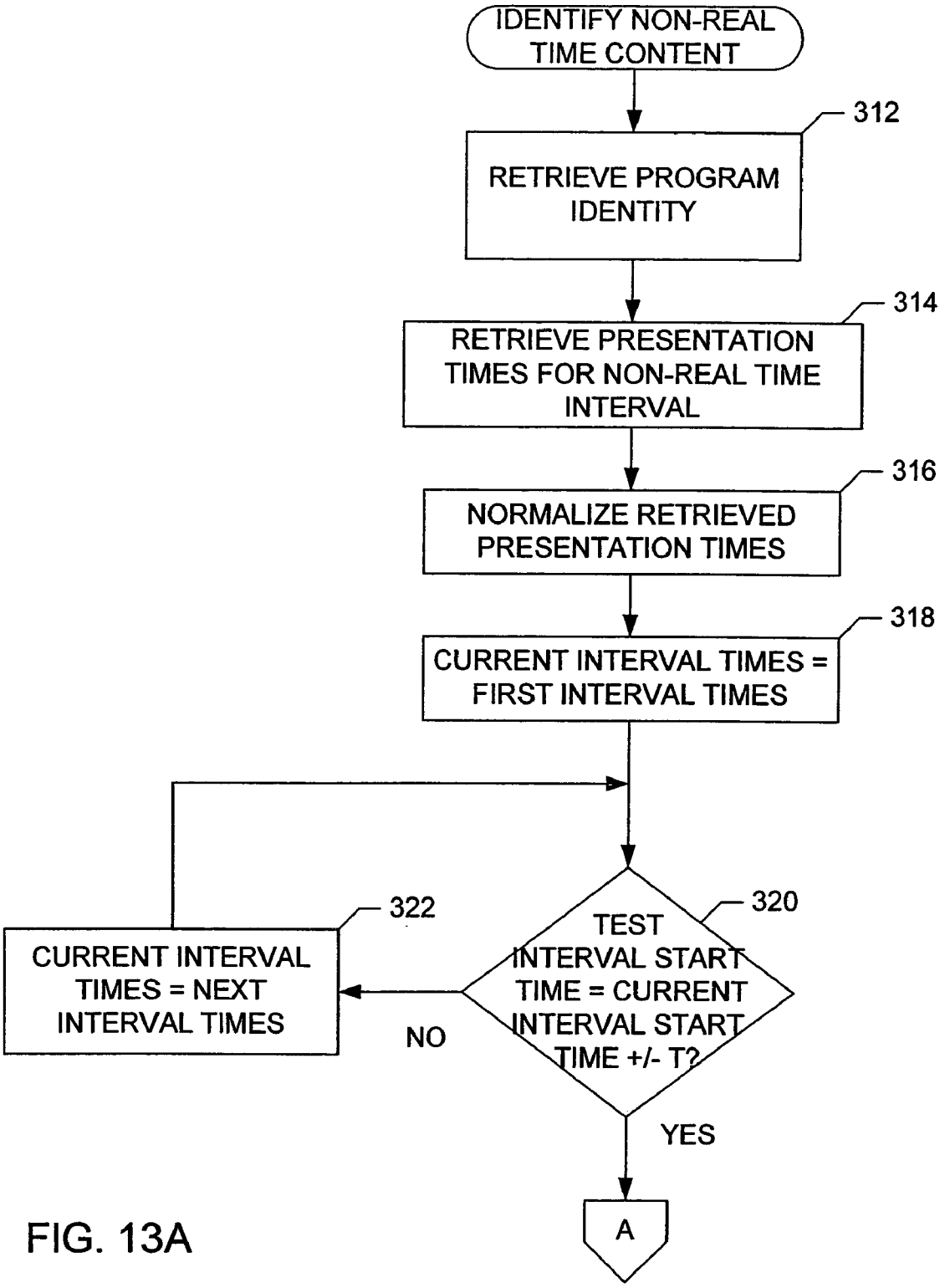

IDENTIFY NON-REAL TIME CONTENT
312
RETRIEVE PROGRAM IDENTITY
314
RETRIEVE PRESENTATION TIMES FOR NON-REAL TIME INTERVAL
316
NORMALIZE RETRIEVED PRESENTATION TIMES
318
CURRENT INTERVAL TIMES = FIRST INTERVAL TIMES
320
TEST INTERVAL START TIME = CURRENT INTERVAL START TIME +/- T?
322
CURRENT INTERVAL TIMES = NEXT INTERVAL TIMES
NO
YES
A

FIG. 13A

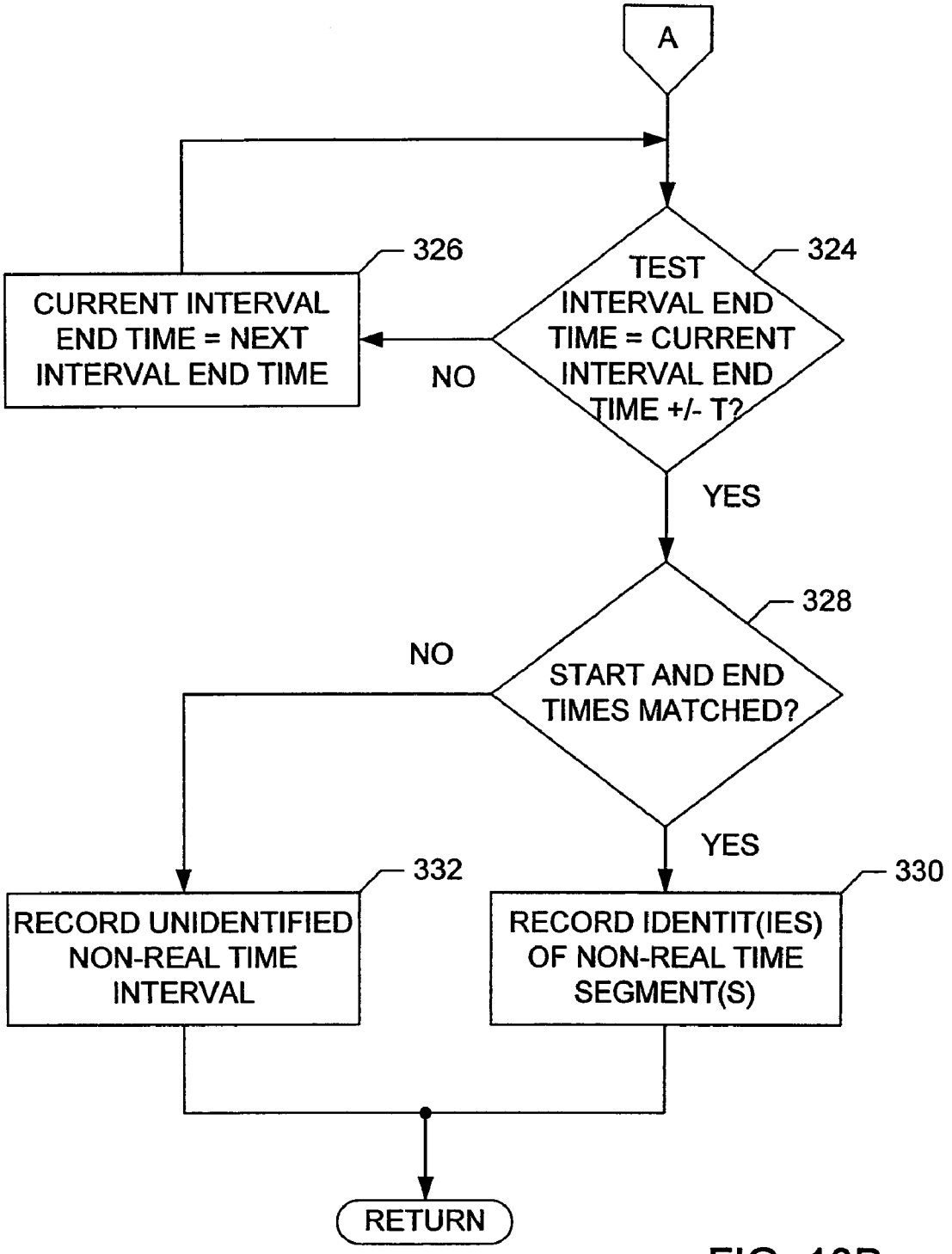

A
TEST INTERVAL END TIME = CURRENT INTERVAL END TIME +/- T?
324
NO
CURRENT INTERVAL END TIME = NEXT INTERVAL END TIME
326
YES
START AND END TIMES MATCHED?
328
NO
YES
RECORD IDENTIT(IES) OF NON-REAL TIME SEGMENT(S)
330
RECORD UNIDENTIFIED NON-REAL TIME INTERVAL
332
RETURN

FIG. 13B

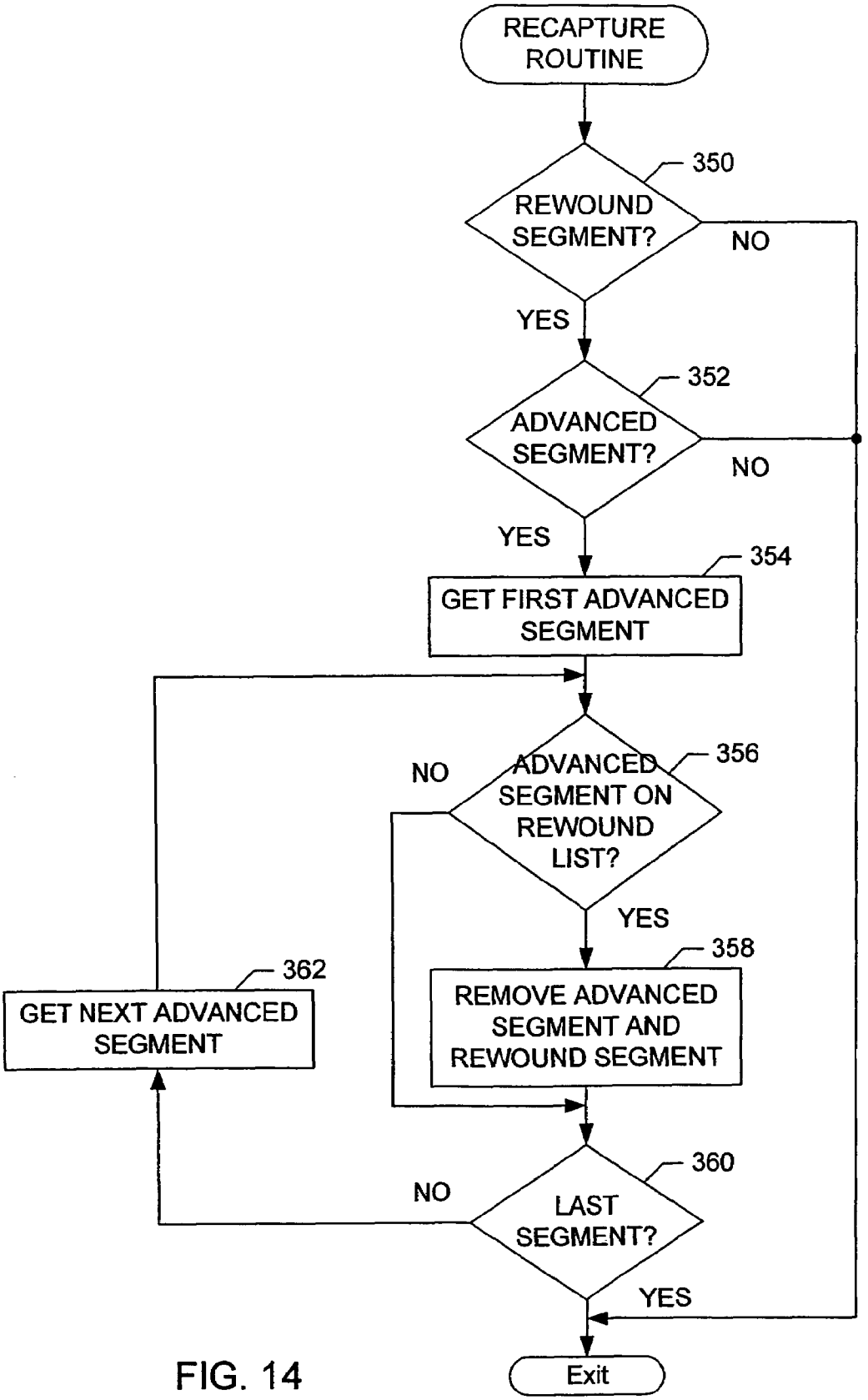

RECAPTURE ROUTINE
350
REWOUND SEGMENT?
NO
YES
352
ADVANCED SEGMENT?
NO
YES
354
GET FIRST ADVANCED SEGMENT
356
NO
ADVANCED SEGMENT ON REWOUND LIST?
YES
358
REMOVE ADVANCED SEGMENT AND REWOUND SEGMENT
362
GET NEXT ADVANCED SEGMENT
360
NO
LAST SEGMENT?
YES
Exit

FIG. 14

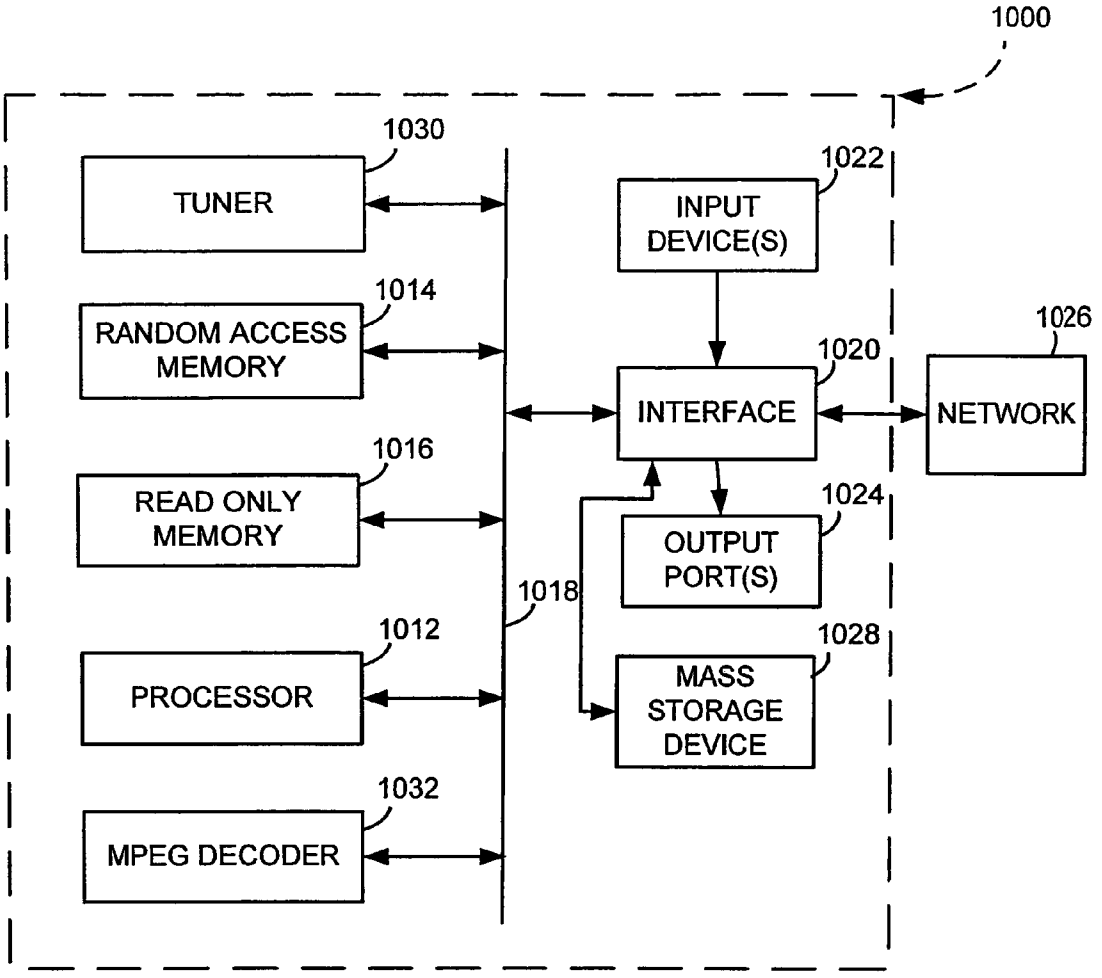

1000
1030
TUNER
1014
RANDOM ACCESS MEMORY
1016
READ ONLY MEMORY
1012
PROCESSOR
1032
MPEG DECODER
1018
1022
INPUT DEVICE(S)
1020
INTERFACE
1024
OUTPUT PORT(S)
1028
MASS STORAGE DEVICE
1026
NETWORK

FIG. 15

# METHODS AND APPARATUS TO DETECT CONTENT SKIPPING BY A CONSUMER OF A RECORDED PROGRAM

## RELATED APPLICATION

This patent arises from a national phase application filed under 35 U.S.C. § 371 which claims the benefit of international patent application serial number PCT/US03/12001, which was filed on Apr. 17, 2003.

## FIELD OF THE DISCLOSURE

This disclosure relates generally to audience measurement, and, more particularly, to methods and apparatus to detect content skipping by a consumer of a recorded program.

## BACKGROUND

Companies that rely on broadcast video and/or audio programs for revenue, such as advertisers, television networks and content providers, wish to know the size and demographic composition of the audience(s) that consume their program(s). Audience measurement companies address this need by measuring the demographic composition of a set of statistically selected households and the program consumption habits of the member(s) of those households. For example, audience measurement companies may collect viewing data on a selected household by monitoring the content displayed on that household's television(s) and by identifying which household member(s) are watching that content.

Traditionally, broadcast programs have been consumed at the time of broadcast. Therefore, it was safe to assume that audience members using an information presenting device such as a television or radio consumed the entire broadcast stream during the period in which the information presenting device was in use. Recently, however, recording devices such as audio cassette players, video cassette recorders (VCR's), set top boxes, digital video recorders, and personal video recorders, such as SonicBlue's ReplayTV®, TiVo® and other devices that permit content to be recorded and replayed in accordance with the desires of the audience members have become commonplace. These devices have increased the audience members' ability to time shift the consumption of broadcast programs (i.e., to record a program at the time of broadcast and consume that same program at a later time that suits the consumer). This ability to time shift has also provided the consumer with enhanced power to consume only selected portions of broadcast programs by, for example, skipping or fast-forwarding through portions of recorded content. Some consumers have used this enhanced ability to avoid viewing advertising commercials or other portions of the broadcast program.

## BRIEF DESCRIPTION OF THE DRAWINGS

FIG. **1** is a schematic illustration of an example system to develop audience measurement statistics.

FIG. **2** is a schematic illustration of an example apparatus to detect content skipping by a consumer of a recorded program.

FIGS. **3**A–**3**C illustrate example log entry data structures that may be created by the apparatus of FIG. **2**.

FIG. **4** illustrates another example log entry data structure that may be created by the apparatus of FIG. **2**.



FIG. **5** is an example chart identifying example segments, segment start times, and segment end times for an example program.

FIG. **6** is an example chart that may be developed by the apparatus of FIG. **2** to record fast forwarded and/or skipped segments of a recorded program and rewound/replayed segments of the recorded program.

FIG. **7** is a flowchart representative of example machine readable instructions that may be executed by the digital recording device of FIG. **1** to determine if a data logging device is coupled thereto.

FIG. **8** is a flowchart representative of example machine readable instructions that may be executed by the data logging device of FIG. **1** to advise the digital recording device of its presence.

FIG. **9** is a flowchart representative of example machine readable instructions that may be executed to implement the apparatus of FIG. **2**.

FIG. **10** is a flowchart representative of additional example machine readable instructions that may be executed to implement the apparatus of FIG. **2**.

FIG. **11** is a flowchart representative of more example machine readable instructions that may be executed to implement the apparatus of FIG. **2**.

FIGS. **12**A–**12**C are a flowchart representative of additional example machine readable instructions that may be executed to implement the apparatus of FIG. **2**.

FIGS. **13**A–**13**B are a flowchart representative of still more example machine readable instructions that may be executed to implement the apparatus of FIG. **2**.

FIG. **14** is a flowchart representative of still more example machine readable instructions that may be executed to implement the apparatus of FIG. **2**.

FIG. **15** is a schematic illustration of an example device that may be used to execute the instructions represented by the flowcharts of FIGS. **7**–**14** to implement the apparatus of FIG. **2**.

## DETAILED DESCRIPTION

FIG. **1** is a schematic illustration of an example system to develop audience measurement statistics. In the illustrated example, a central data collection office **10** is connected to a plurality of statistically selected households **12**, **14**, **16** via one or more public and/or private networks **18**. The monitored home sites **12**, **14**, **16** are provided with sensors and/or other passive and/or active data collection engines to gather data identifying the broadcast programs consumed by the members of the monitored households **12**, **14**, **16**, and the household members that consumed those programs. The data collected in the monitored home sites **12**, **14**, **16** is exported periodically or continuously to the central office **10** via the network **18**. The data collection office **10** includes one or more computers that analyze the data received from the monitored home sites **12**, **14**, **16** to develop meaningful audience measurement statistics such as, for example, television ratings, audience share measurements, etc. The network **18** can be implemented by any desired network such as, for example, the Internet, the public switched telephone network, a wireless connection, dedicated connections, etc.

In the illustrated example, the home sites **12**, **14**, **16** receive video and/or audio programs broadcast from one or more broadcasting systems **20**. Although FIG. **1** schematically illustrates the broadcasting systems **20** by a radio tower, persons of ordinary skill in the art will readily appreciate that the broadcasting systems may be implemented by one or more of a terrestrial broadcasting system,

a cable broadcasting system, a satellite broadcasting system, the Internet, or any other broadcasting system.

The broadcast programs may include primary content (e.g., entertainment, informational and/or educational content such as movies, television network programs, sporting events, news, etc.) and may also include secondary content (e.g., commercials) interspersed within the primary content. It is frequently the case that the primary content is more desirable to consumers then the secondary content, but this is not always the case (e.g., commercials (i.e., secondary content) during the NFL Superbowl are sometimes of greater interest than the football game (i.e., the primary content)). For simplicity of discussion, in the following it will be assumed that the broadcast program contains a television program (i.e., primary content) interspersed with commercials (i.e., secondary content) advertising products of a sponsor of the television program.

The home sites **12**, **14**, **16** may include any number of information presenting devices **22** such as a television, a computer, a radio, a stereo, an Internet appliance, etc. The home sites **12**, **14**, **16** may also include one or more recording devices such as an analog recording device (e.g., a magnetic tape recording device for recording audio programs and/or a video cassette recorder (VCR)), and/or a digital recording device **24** such as a set top box, a digital video recorder, a personal video recorder, a computer, etc. The home site **12**, **14**, **16** may also be provided with a data logging device **26** which is coupled to the digital recording device to collect, record and/or analyze data reflecting usage of the digital recording device **24**. The data collected and/or analyzed by the data logging device **26** may be periodically exported to the central office **10** via the network **18** for further analysis and/or distribution. The data exportation may be performed at regularly scheduled time intervals and/or may be triggered by predetermined events.

If a digital or personal video recorder **24** is present in the home site **12**, **14**, **16**, the home site **12**, **14**, **16** may be connected to one or more schedule warehouses **28** via one or more public and/or private networks **18**. In the illustrated example, the schedule warehouse **28** stores media schedules of broadcast programming for use by digital and/or personal video recorders **24**. The media schedules stored in the schedule warehouse **28** may be compiled from information provided by a number of sources. For example, they may be developed from information provided by broadcasters and/or content providers. The media schedules may include, among other things, the titles of programs to be broadcast, the scheduled broadcast times for the programs, the broadcast channels that are to carry the scheduled programs at the scheduled times, and the identities of the networks that intend to broadcast the programs. Some of the information present in the media schedules may not be provided to the digital/personal video recorders **24**, but may instead be supplied to audience measurement companies (e.g., the central office **10**) to assist in computing ratings and/or audience demographics information. Thus, as shown in FIG. **1**, the central office **10** may also be coupled to the schedule warehouse **28** via the network **18**.

On a periodic basis, the digital/personal video recorders **24** retrieve media schedule(s), or portion(s) thereof, from the schedule warehouse **28** via the network **18**. The retrieved schedules are stored in a local memory of the digital and/or personal video recorder **24** for use in selecting and recording programs of interest. In the personal video recorder context, the media schedules may include data that enables the personal video recorder **24** to select and store programs that may be of interest to the members of the household **12**, **14**, **16** based on prior viewing habits and/or otherwise identified interests of those members.

FIG. **2** is a schematic illustration of an example apparatus **50** to detect portions of a recorded program that were not played back in real time (e.g., that were skipped, fast forwarded, paused, and/or rewound). As explained in further detail below, the example apparatus **50** of FIG. **2** may be implemented by one or more of the digital recording device **24**, the data logging device **26**, and/or the central office **10**. The apparatus **50** is equipped to record and analyze data reflecting usage of, for example, a digital recording device **24** in a home site **12**, **14**, **16**.

For the purpose of logging recording or presentation times of a broadcast program recorded at a home site **12**, **14**, **16**, the apparatus **50** is provided with a time stamper **52**. The time stamper **52** records a plurality of sequential times in association with the recorded program. In particular, the time stamper **52** records the times at which various sections of a received broadcast program are received and/or stored in a memory **54** of, for example, a digital recording device **24**. For example, if the digital recording device **24** is implemented by a digital/personal video recorder such as the TiVo® video recorder, the recorded program may be processed in accordance with a modified version of the MPEG-2 standard. In such an example, the broadcast program received by the digital video recorder may be compressed and stored as two datastreams, namely, an audio datastream and a video datastream. Each of the datastreams contains a series of data packets. Each of the packets includes a header and a payload. The header contains a number of fields to store information about the associated content of the packet and/or the relationship of the packet to other packets in the datastream (e.g., a packet sequence number). The payload section of the packet stores the program content (e.g., a section of the audio and/or video content of the broadcast program). The time stamper **52** writes the time at which the corresponding payload was recorded by the digital/personal video recorder **24** in a field of the header of the packet. In the example of the TiVo® digital/personal video recorder, the time stamps provided by the time stamper **52** are presentation times (e.g., the times at which the corresponding packets were received and/or stored by the TiVo® product) and are intended for use in synchronizing the packets of the audio datastream and the packets of the video datastream during playback of the recorded program.

For the purpose of recording the playback rate of the recorded program, the apparatus **50** is further provided with a presentation time retriever **56**. When the recorded program is played back from the memory **54** (which may be within seconds or minutes of receipt of the broadcast program by the digital recording device **24** if the consumer is "live" viewing the same, or may be hours, days, weeks, months or longer after receipt of the broadcast program by the digital recording device **24**), the presentation time retriever **56** retrieves the recording/presentation times of some or all of the sections of the program being played back and stores those recording times in association with the playback time of the corresponding sections of the program. In other words, as the program is played back, the presentation time retriever **56** periodically creates a running list of log entry data structures **58** such as those shown in FIGS. **3**A–**3**C. The log entry data structures **58** are preferably created and populated at evenly spaced intervals of real time. In particular, each log entry data structure **58** is preferably created

at substantially the time that a corresponding section of the recorded program is being played.

Each log entry **58** may include a playback date field **60** to store data representative of the playback date (in this example, formatted as MM/DD/YYYY for month/day/year) of a corresponding section of the recorded program, a playback time field **62** to store data representative of the playback time (in this example, formatted as HH:MM:SS.mmm for hour/minute/second/millisecond) of the corresponding section of the program being played, a recording date field **64** to store data representative of the recording date (in this example, formatted as MM/DD/YYYY) of the section of the recorded program being played, and a recording time field **66** to store data representative of the recording time (in this example, formatted as HH:MM:SS.mmm) of that same section of the program. In the example of the TiVo® video recorder, the presentation time retriever retrieves the presentation date/time from the header(s) of the packet(s) currently being played. The presentation time retriever **56** then respectively stores the retrieved presentation date/time in the recording date field **64** and recording time field **66** of a log entry data structure **58** to associate that data with the playback time of the corresponding packet(s). Additionally, the presentation time retriever **56** may store a current date and time in the playback date field **60** and the playback time field **62** of that same log entry data structure **58**.

The presentation time retriever **56** may also retrieve a program identifier from the packet header of the section of the recorded program currently being played back and store that identifier in one or more program identification fields **68**. The program identifier may be any form of information that identifies the program being played back (e.g., a program name, a program identification number, etc.), and/or the station that broadcast the program (e.g., a broadcast channel number, a station name, or a standardized station identifier). In the example of FIGS. **3**A–**3**C, the program identification field **68** contains a universal program identifier. As is know to persons or ordinary skill in the art, a universal program identifier is a program code that identifies a program based on a pre-defined, standardized code arrangement.

In the example of FIG. **2**, the playback dates and times stored by the presentation time retriever **56** are provided by the time stamper **52**. However, persons of ordinary skill in the art will appreciate that the playback dates and times may alternatively be developed by the presentation time retriever **56** and/or by a clock separate from the time stamper **52**. Irrespective of their source, the playback times stored by the presentation time retriever **56** preferably correspond to real time and are evenly spaced. The playback spacing or intervals between the playback times/log entry data structures **58** are a matter of design choice. However, persons of ordinary skill in the art will appreciate that shorter intervals between the recorded playback times/log entry data structures **58** (i.e., the more playback times recorded, the fewer packets that are played without logging their recording times), translate into more accurate identification of portions of the recorded program that were not played back in real time, and that shorter playback intervals also translate into higher data collection volume and, thus, more severe memory resource demands. In the illustrated example, the playback intervals are 2.7 seconds long (i.e., a log entry data structure **58** is created every 2.7 seconds). Although the evenly spaced, equal playback intervals are currently preferred, persons of ordinary skill in the art will appreciate that, instead of being triggered by time, the creation of a log entry data structure **58** may be triggered by one or more predetermined events such that the playback intervals are not equal and/or not evenly spaced in time.

Persons of ordinary skill in the art will appreciate that, in examples in which the same playback interval is consistently used between log entries, it is possible to not record the playback time in the running log because the log entry data structures **58** themselves inherently provide both an indication of the time lapse between entries (i.e., it is always a consistent value such as, for example, 2.7 seconds) and the aggregate amount of time that has passed during playback. In other words, because we know that a log entry data structure **58** recording the playback time of the section of the program currently being played back is made every, for example, 2.7 seconds, then we know that the twenty-first log entry (e.g, the log entry data structure **58** of FIG. **3**B) occurred 2.7 seconds after the twentieth log entry (e.g, the log entry data structure **58** of FIG. **3**A) without having to compute a playback interval between those data structures. However, even in cases where the same playback interval is consistently used, it may be desirable to record the playback times as a mechanism for sorting and/or distinguishing between log entries.

An alternative example data structure **70** is shown in FIG. **4**. Like the example data structure **58** of FIGS. **3**A–**3**C, the example data structure **70** of FIG. **4** includes a playback date field **60**, a recording date field **64**, a recording time field **66**, and a program identification field **68**. However, unlike the data structure **58** of FIGS. **3**A–**3**C, the data structure **70** of FIG. **4** does not contain the playback time field **62** for the reasons explained above. The playback date field **60** is retained because it might be interesting to learn how much time passed between broadcast and playback of the recorded program. Similarly, although not used in the example of FIG. **4**, the playback time field **62** is preferably retained in the data structure **70** because it might be interesting to learn the time of day at which the playback occurred.

As shown in FIG. **4**, the example log entry data structure **70** includes many fields not present in the example data structure **58** of FIGS. **3**A–**3**B. For instance, the data structure **70** of FIG. **4** includes the following fields: (a) a digital recording device serial number field **71**, (b) a sequence number field **72**, (c) a tuner identification number field **73**, (d) a playback mode indicator field **74**, (e) an input source indicator field **76**, (f) a channel number field **78**, (g) a network name field **80**, (h) a program name field **82**, (i) a universal network identifier field **84** and (j) a checksum **86**.

The digital recording device serial number field **71** may contain the serial number of the digital recording device **24** being monitored. The serial number may be a variable-length string (e.g., 12345678) that uniquely identifies a particular digital recording device **24** such as a set top box, a digital video recorder, and/or a personal video recorder. In monitored households **12**, **14**, **16** having multiple digital recording devices **24**, the serial number of the digital recording devices **24** can be used to track the viewing habits within a household **12**, **14**, **16** on a digital recording device **24** by digital recording device **24** basis.

The sequence number field **72** may be used to number the data structures **58**, **70** to ensure that they are transmitted, analyzed, and/or processed in the correct order. The sequence number field may, for example, store an unsigned integer. For example, the integer may be a value in the range of 0–255. When the sequence number reaches 255, the sequence number value is returned to zero to again begin numbering the next consecutive sequence of data structures **58**, **70** from 0–255.

The tuner number field **73** may contain a variable length string identifying the tuner that received the recorded program. For example, the tuner number field **73** may be used to indicate which tuner in a digital recording device **24** having multiple tuners was used to receive the recorded program.

The playback mode indicator field **74** may contain playback mode indicator data designating the operational mode of the digital recording device **24** at the time the log file data structure **70** was generated. In the illustrated example, the playback mode indicator data may be a coded such that the code "**1**" indicates the device **24** is playing a live broadcast, the code "**2**" indicates the device **24** is playing a recorded program, the code "**3**" indicates the device **24** is playing an alternative stream (e.g., a movie trailer which originates from the hard disk of the device **24** after being, for example, downloaded via the Internet or a television broadcast channel), the code "**4**" indicates the device **24** is recording a user selected program, the code "**5**" indicates the device is recording a program based on a user profile, the code "**6**" indicates the user is accessing a full screen user interface or menu (without audio), and the code "**7**" indicates that the user is accessing a transparent user interface screen that overlays a playing program (with audio). If the playback mode data is code "**6**", the broadcast time field **66**, the input source field **76**, the channel number field **78**, the station name field **80**, and the program name field **82** may be blank in this circumstance and the log data may not identify the user's activity.

The input source indicator field **76** may contain a code that represents the input source through which the stored broadcast **120** was received by the digital recording device **24**. The input source indicator field **76** may designate codes for different input sources to the digital recording device **24**. For example, the code "RF" may represent a radio frequency input, the code "SAT**1**" may represent a first satellite input, the code "SAT**2**" may represent a second satellite input, the code "COMP" may indicate a composite input, the code "SVID" may represent an S-video input and the code "other" may indicate that the recorded program did not come from a broadcast source, such as when the video was pre-loaded onto a hard disk or the video was delivered through a private channel or broadband connection. In such circumstances, the video being viewed will not be identified, and the playback time and channel number fields **62**, **78** will be left blank.

The channel number field **78** may contain data indicative of the channel number of the internal tuner of the digital recording device **24** (e.g., a numerical value of 0–99999), data indicative of a channel number of an external device such as a television (e.g., a numerical value of 0–99999) coupled to the digital recording device **24**, data indicative of a minor channel, the code "manual," and/or nothing. In the illustrated example, a numeric value stored in the channel number field **78** indicates a tuned channel number. However, the stored channel number in the channel number field **78** may not be the same as the channel number that is displayed in the user interface of the digital recording device **24** because the user interface may display logical channel numbers that have been remapped into a single logical channel space.

In the illustrated example, the channel number field **78** contains data indicative of the tuned channel of the external device, not the tuned channel of the digital recording device **24**, if an external device is connected to the RF or satellite input (e.g., on channel **2**, **3** or **4**). In such circumstances, the input source field **76** indicates that the input source is "RF." Therefore, interpretation of the data in the channel number field **78** is dependent on knowledge of the identity of the device connected to the RF input of the digital recording device **24**.

In the illustrated example, the channel number field **78** contains the code "manual" when the user has requested a manual record from a particular input source and has specified that the digital recording device **24** should not change channels on the device before starting the recording process. The "manual" channel code may be used with devices that cannot be tuned (e.g., camcorders and other devices). The "manual" channel number may also be used with pay-per-view (PPV) events or other types of programs where the user is required to perform some special action to tune to the desired channel.

In the illustrated example, the network name field **80** contains a variable-length string of data indicative of the call letters or other associated name (e.g. "NBC") of the broadcasting station that broadcast the recorded program. Thus, the data in the network name field **80** corresponds to the data contained in the channel number field **78**. The station name may be determined, for example, by comparing the tuned channel number from the tuned channel field **78** with a program schedule that has been downloaded from the schedule warehouse **28**. If the station name cannot be determined, then the network name field **80** may be left blank.

In the illustrated example, the program name field **82** contains a variable-length string of data representative of the name of the recorded program (e.g. "Friends"). The program name may be determined, for example, by comparing the tuned channel number contained in the channel number field **78** with a program schedule that has been downloaded from the schedule warehouse **28**. If the program name cannot be determined, then the program name field **82** may be left blank.

In the illustrated example, the network identifier field **84** contains data representative of the network that broadcast the recorded program. For example, the data contained in the network identifier field **84** may be a universal network identifier that is based on a standardized arrangement such as the universal broadcast/cable station identifier arrangement. The universal broadcast/cable station identifier arrangement identifies each broadcast/cable station by a unique code. Alternatively, a multi-system operation (MSO) specific code may be used.

In the illustrated example, the checksum field **84** contains data representative of a computed checksum. The computed checksum may be, for example, a numerical value based on the number of bits set in the corresponding data structure **58**, **70**. Any known software technique for generating checksums may be used to calculate and/or check the checksum value.

In order to determine if the audience has played a portion of the recorded program back in something other than real time, the apparatus **50** is further provided with an interval detector **90** and a difference detector **92**. The interval detector **90** computes recording intervals between the recording times appearing in successive ones of the log entry data structures **58**, **70** developed by the presentation time retriever **56**. The interval detector **90** computes the recording interval by finding the difference between the recording times appearing in successive log entries **58**, **70**. Persons of ordinary skill in the art will appreciate that, if the recording times appearing in successive log entries **58**, **70** are identical, then the recorded program has been paused for the duration of the corresponding playback interval (i.e., for the entire duration of the time between the log entries **58**, **70**

corresponding to the identical recording times). If, on the other hand, the difference between the recording times appearing in successive log entries **58**, **70** has a first sign (e.g., positive), then either the recorded program was played back in real time or the recorded program was played back at a rate different than real time (e.g., a portion of the program was skipped or fast forwarded, or a portion of the program was played back in slow motion or was paused for less than the entire duration of the playback interval). If instead the difference between the recording times appearing in successive log entries **58**, **70** has a second sign (e.g., negative), then the recorded program was rewound.

The interval detector **90** may also compute playback intervals. In particular, the interval detector **90** may optionally calculate the difference between the playback times stored in successive log entries **58**, **70** to determine a playback interval corresponding to each pair of successive log entry data structures **58**, **70**. However, as explained above, if log entries **58**, **70** are generated at regular intervals, calculating playback intervals for each pair of successive log data structures **58**, **70** may be dispensed with since the playback intervals will always be the same. In such circumstances, the same, known constant value (e.g., 2.7 seconds) may be used for the playback intervals.

To determine if any portion(s) of the recorded program have been played at a rate different then real time, the apparatus **50** is provided with a difference detector **92**. The difference detector **92** determines if a difference exists between any of the recording intervals calculated by the interval detector **90** and the corresponding playback interval. In particular, the difference detector **92** preferably subtracts a value representative of the recording interval calculated by the interval detector **90** from a value representative of the corresponding playback interval. (As mentioned above, the corresponding playback interval may be a value calculated for each pair of data structures **58**, **70**, or it may be a constant value based on the understanding that the log entries **58**, **70** are developed at a constant periodic rate.) If the difference calculated by the difference detector **92** is zero, then the portion of the recorded program corresponding to the recording times stored in the data structures **58**, **70** was played back in real time. If, on the other hand, the difference calculated by the difference detector **92** is non-zero, then the portion of the recorded program corresponding to the recording times stored in the data structures **58**, **70** was not played back in real time. In particular, if the playback interval is smaller than the recording interval such that the difference calculated by the difference detector **92** is negative, then the portion of the recorded program corresponding to the recording times stored in the data structures **58**, **70** was advanced faster than real time (e.g., the portion was at least partially skipped and/or fast forwarded). Otherwise, if the playback interval is larger than the recording interval (or the recording interval has a negative sign due to a rewind event) such that the difference calculated by the difference detector **92** is positive, then the portion of the recorded program corresponding to the recording times stored in the data structures **58**, **70** was at least partially played in slow motion, rewound, and/or paused for less than the full time interval between the corresponding data structures **58**, **70**.

To identify the segment(s) in the portion of the recorded program that were not played back in real time, the apparatus **50** is further provided with a non-real time content identifier **94**. The non-real time content identifier **94** responds to detection of non-zero differences by the difference detector **92** by searching a database to determine the identities of the program segments that were skipped, fast forwarded, and/or rewound. For example, the database may include a plurality of program tables identifying the start times, end times, and names of the segments of broadcast programs. Each of the program tables may be associated with a respective broadcast program, and the start and end times appearing in those program tables may be normalized as explained below (e.g., such that the start of broadcast time is time zero).

The program tables may be created by the schedule warehouse **28**, the central office **10**, content providers, and/or broadcast stations. An example program table that may be used by the non-real time content identifier **94** to identify the segment(s) of an example recorded program that were not played in real time is shown in FIG. **5**.

In the example of FIG. **5**, the program table includes a name column, a start time column, and an end time column. The name column is populated with unique names of the segments of the corresponding program. For instance, in the example of FIG. **5**, the program includes the following segments: (a) introduction, (b) commercial **1**, (c) content **1**, (d) commercial **2**, (e) content **2**, (f) commercial **3**, (g) content **3**, etc. The start time and end time columns are populated with the start and end times of the corresponding program segments. Preferably the start times and end times stored in the program table are normalized to the start of a broadcast interval rather than reflective of actual broadcast times. For example, the program of FIG. **5** is to run the introduction segment from time zero (i.e., the start of the broadcast interval) until time 00:03:30.003 (i.e., for a length of three minutes, thirty seconds, and 3 milliseconds). By way of another example, the segment entitled "commercial **1**" is to start three minutes, thirty seconds, and 3 milliseconds after the broadcast interval begins.

Normalized times are preferably used in the program tables instead of actual broadcast times because, due to reflections during transmission, geographic distances, etc., different home sites **12**, **14**, **16** may receive the same broadcast program at slightly different times. However, all of the relative lengths of the broadcast segments as received at the home sites **12**, **14**, **16** will be the same from site to site regardless of transmission delays. Therefore, by normalizing the start and end times of the program tables to the start times of the programs they represent, and by normalizing the start times and the end times recorded in the log entry data structures **58**, **70** to the overall start (i.e., received) time of the recorded program, the effect of transmission differences between households can be reduced to thereby allow the non-real time content identifier **94** to use the same table to identify the program segments that were subjected to non-real time playback activity regardless of the location of the monitored home site **12**, **14**, **16** that exhibited the non-real time playback event.

The non-real time content identifier **94** may begin the search process by locating the program table (if any) corresponding to the recorded program. The located program table may then be searched for the segment(s) corresponding to the portion of the recorded program which experienced something other than real-time playback. For example, the non-real time content identifier **94** may compare the normalized start time of the recording interval of the recorded program at issue to the start times appearing in the program table associated with the corresponding program (see FIG. **5**). Persons of ordinary skill in the art will readily appreciate that it is unlikely that the normalized start time of the recording interval that experienced non-real time activity will correspond exactly to one of the start times appearing in the program chart, because the audience member causing the

non-real time effect is unlikely to send a command to, for example, fast forward, pause, or rewind, at precisely the normalized start time of a program segment. Instead, it is likely that the normalized start time of the recording interval at issue will fall within a range (e.g., +/−30 seconds) around a start time of one of the intervals. Therefore, the non-real time content identifier **94** is structured to find a match between the start time of the recording interval at issue and a start time of a segment appearing in the program table, if the corresponding start times are within some threshold amount (T) of one another.

Because the portion of the recorded program that experienced non-real time activity may encompass more than one segment of the recorded program, in addition to searching the program table (see FIG. **5**) for a corresponding start time, the non-real time content identifier **94** also searches the program table associated with the recorded program for an end time that matches the normalized end time of the recording interval that was subjected to non-real time playback. As with the start time search, it is likely that the normalized end time of the recording interval at issue will fall within a range (e.g., +/−30 seconds) around an end time of one of the intervals. Therefore, the non-real time content identifier **94** is structured to find a match between the end time of the recording interval at issue and an end time of a segment appearing in the program table, if the corresponding end times are within some threshold amount (T) of one another.

Once a matching start time and a matching end time are identified in the program table, the non-real time content identifier **94** identifies the segment(s) corresponding to the substantially matching start and end times as the segment(s) of the recorded program that were subjected to non-real time playback. For instance, in the example of FIG. **5**, if the start time of the recording interval that was subjected to non-real time activity substantially matches the start time of the segment entitled "commercial **2**," and the end time of the recording interval substantially matches the end time of the segment entitled "commercial **2**," then the non-real time content identifier **94** identifies the segment "commercial **2**" as the portion of the recorded program that was subjected to non-real time playback (e.g., skipped or fast forwarded). As another example, if the start time of the recording interval that was subjected to non-real time activity substantially matches the start time of the segment entitled "commercial **2**," and the end time of the recording interval substantially matches the end time of the segment entitled "content **2**," then the non-real time content identifier **94** identifies both the segment entitled "commercial **2**" and the segment entitled "content **2**" as the portion of the recorded program that was subjected to non-real time playback (e.g., skipped or fast forwarded).

If the non-real time activity being analyzed by the non-real time content identifier **94** was an advancing event (e.g., skipping, fast forwarding, pausing or slow playing), then the recording interval will have a positive sign and the non-real time content identifier **94** will use the normalized playback time of the first created data structure in the pair of data structures **58**, **70** associated with the playback interval as the start time and the normalized playback time of the second created data structure in the pair of data structures **58**, **70** associated with the playback interval as the end time. If, on the other hand, the non-real time activity was a rewinding event, then the recording interval will typically have a negative sign (unless the rewind was very short (e.g., less than one half the length of the playback interval)) and the non-real time content identifier **94** will use the normalized playback time of the second created data structure in the pair of data structures **58**, **70** associated with the playback interval as the start time and the normalized playback time of the first created data structure in the pair of data structures **58**, **70** associated with the playback interval as the end time. This latter approach enables the non-real time content identifier **94** to record the segments of the recorded program encompassed within the rewound interval. Persons of ordinary skill in the art will appreciate, however, that, just because the segments were rewound, does not mean that all of those segments were watched. Therefore, if any of the rewound segments are subjected to further non-real time playback (e.g., rewinding, fast forwarding, pausing, and/or skipping), the apparatus **50** will log that activity for later reconciliation as explained below.

Persons of ordinary skill in the art will appreciate that alternative methods of identifying the segments of the recorded program that were subjected to non-real time playback activity may alternatively be employed. For example, rather than requiring a start or end time of the recording interval to respectively fall within a range around a start time and end time appearing in the program table, the non-real time content identifier **94** may alternatively search the program table to determine if the recording interval overlaps with at least a predetermined portion (e.g., 10%) of one or more of the known intervals reflected in that table. The non-real time content identifier **94** may then identify the segment(s) associated with the known interval(s) that are sufficiently overlapped by the recording interval that experienced non-real time playback as the segment(s) that were not played back in real time.

Irrespective of the methodology employed for determining the identities of the segments of the recorded program that were subjected to non-real time playback activity (e.g., fast forwarding, pausing, skipping, and/or rewinding), the non-real time content identifier **94** preferably records the determined identities of the segments in a recapture table such as the example recapture table shown in FIG. **6**. In the example of FIG. **6**, the recapture table includes two columns. A first column is used to store the identities of all segments of the recorded program that were advanced in time during the recording interval at a rate faster than real time. Thus, the first column stores the identities (in this example, represented by segment numbers), of any segments that the non-real time content identifier **94** identifies as having been skipped and/or fast forwarded. The second column of the recapture table of FIG. **6** is used to store the identities of all segments of the recorded program that were rewound.

As mentioned above, a segment of a recorded program that is initially skipped and/or fast forwarded may later be viewed in real time if a portion of the recorded program encompassing that segment is rewound. To determine if a portion of the recorded program that was passed at a rate greater than real time was viewed in real time as a result of rewinding the recorded program, the apparatus **50** is further provided with a recapture identifier **96**. In the illustrated example, the recapture identifier **96** compares the segment names appearing in the first column of the recapture table of FIG. **6** with the segment names appearing in the second column of the recapture table of FIG. **6**. Whenever the recapture identifier **96** encounters the same segment name in both columns, the recapture identifier **96** removes the pair of segment names from the table. The segment names remaining in the advanced column of the recapture table after the recapture identifier **96** has checked every entry of the advanced column for a corresponding entry in the rewound column, represent the segments that were skipped and/or

fast forwarded without being recaptured by a rewind. The segment names remaining in the rewound column of the recapture table after the recapture identifier **96** has checked every entry of the advanced column for a corresponding entry in the rewound column, represent the segments that were rewound and reviewed.

It must be emphasized that the recapture identifier **96** cancels one segment name in each column of the recapture table when a match is identified. Thus, if the same segment name appears twice in the advanced column and once in the rewound column, the recapture identifier **96** will remove one instance of the segment name from the advanced column and the one instance of that segment name from the rewound column, but the second instance of the segment name will remain in the advanced column, because the second entry of the segment name in the advanced column indicates that the segment was again skipped and/or fast forwarded after the rewound event. For instance, in the example of FIG. **6**, segment "**5**" was fast forwarded and/or skipped one time and rewound one time such that the fast forwarding/skipping event and rewinding event cancel each other out and, absent other data, segment "**5**" was played at real time one time. Therefore, when the recapture identifier **96** processes the recapture table, it will remove both references to segment "**5**" from the recapture table. As a result, when the processing of the recapture table is complete, there will be no reference to segment "**5**" in the recapture table, thereby indicating that segment "**5**" was viewed in real time.

On the other hand, segment "**8**" appears twice in the advanced column and once in the rewound column because, although it was rewound once, it was fast forwarded twice (i.e., it was never played in real time). Therefore, when the recapture identifier **96** processes the recapture table of FIG. **6**, it will remove one instance of the segment "**8**" from the advanced column and the only instance of segment "**8**" from the rewound column. As a result, when the processing of the recapture table of FIG. **6** is complete, there will be one reference to segment "**8**" in the advanced column, but no reference to that segment in the rewound column, thereby indicating that segment "**8**" was fast forwarded and/or skipped and not recaptured by a rewind event.

In another example, segment "**7**" only appears in the rewound column of the example recapture table of FIG. **6**. As a result, when the processing of the recapture table of FIG. **6** is complete, there will be one reference to segment "**7**" in the rewound column, but no reference to that segment in the advanced column, thereby indicating that segment "**7**" was twice viewed in real time.

Persons of ordinary skill in the art will appreciate that the apparatus **50** may be located in a single device (e.g., the digital recording device **24**) or distributed across multiple devices. For instance, in the example of FIG. **1**, the apparatus **50** may be implemented by the digital recording device **24**, the data logging device **26**, and/or the central office **10**. In a more specific example, the time stamper **52**, a portion of the memory **54**, and the presentation time retriever **56** are implemented by the digital recording device **24**, and the interval detector **90**, the difference detector **92**, the non-real time content identified **94**, a portion of the memory **54**, and the recapture identifier **96** are implemented by the data logging device **26** and/or the central office **10**. If the interval detector **90**, the difference detector **92**, the non-real time content identified **94** and the recapture identifier **96** are implemented by the data logging device **26**, the processed data entry structures **50**, **78** and recapture table(s) (see FIG. **6**) may be periodically exported to the central office via the network **18** for compilation with data from other home sites **12**, **14**, **16** and/or further analysis. If the interval detector **90**, the difference detector **92**, the non-real time content identified **94** and the recapture identifier **96** are implemented by the central office **10**, the data logging device **26** may be excluded or may function as a store and forward repository for the log entry data structures **58**, **70**.

Flowcharts representative of example machine readable instructions for implementing the apparatus **50** of FIG. **2** are shown in FIGS. **7**–**14**. In this example, the machine readable instructions comprise one or more programs for execution by one or more processors such as the processor **1012** shown in the example device **1000** discussed below in connection with FIG. **15**. The program(s) may be embodied in software stored on a tangible medium such as a CD-ROM, a floppy disk, a hard drive, a digital versatile disk (DVD), or a memory associated with a processor, but persons of ordinary skill in the art will readily appreciate that the entire program (s) and/or parts thereof could alternatively be executed by a device other than a processor and/or embodied in firmware or dedicated hardware in a well known manner. For example, any or all of the time stamper **52**, the presentation time retriever **56**, the interval detector **90**, the difference detector **92**, the non-real time content identifier **94**, and/or the recapture identifier **96** could be implemented by software, hardware, and/or firmware. Further, although the example program(s) are described with reference to the flowcharts illustrated in FIGS. **7**–**14**, persons of ordinary skill in the art will readily appreciate that many other methods of implementing the example apparatus **50** may alternatively be used. For example, the order of execution of the blocks may be changed, and/or some of the blocks described may be changed, eliminated, or combined.

The programs of FIGS. **7** and **8** may be used in examples wherein the data logging device **26** is used to store, analyze and/or forward the log entry data structures **58**, **70** output by the digital recording device **24**. In such an example, the digital recording device **24** may execute the program of FIG. **7** and the data logging device **26** may execute the program of FIG. **8** to enable the digital recording device **24** to determine if the data logging device **26** is connected and, thus, if logging of user activity is desired. The digital recording device **24** and the data logging device **26** may be connected in any desired manner. For example, they may be connected via a serial connection (e.g., a DB-9 connection or a USB connection).

Irrespective of how the digital recording device **24** and data logging device **26** are connected, the program of FIG. **7** begins at block **100** where the processor of the digital recording device **24** sets an attempt counter to a maximum value (e.g., 30). The processor of the digital recording device **24** then send a message to the digital logging device **26** (block **102**) to determine if a digital logging device **24** is connected to its output port (e.g., a DB-9 serial port). If the digital recording device **24** receives an acknowledgement signal back from the data logging device **26** (block **104**), control advances to block **114**. Otherwise, control advances to block **106**.

Assuming for purposes of discussion that an acknowledgement has not been received (block **104**), the processor of the digital recording device **24** decrements the attempts counter by one (block **106**). If the attempts counter has reached zero (block **108**), it is determined that no data logging device **26** is coupled to the digital recording device **24**. As a result, the logging function of the digital recording device **26** is disabled such that no usage data is collected and stored in the data structures **58**, **70** (block **110**). The program of FIG. **7** then terminates.

If, on the other hand, the attempts counter has not reached zero (block **108**), then control advances to block **112**. At block **112**, the processor of the digital recording device **24** determines if sufficient time has passed to again try to contact the logging device **26**. When sufficient time has passed (block **112**), control returns to block **103** where the processor of the digital recording device **24** sends another message to the data logging device **24**. Control continues to loop through blocks **102**–**112** until the attempt counter reaches zero (block **108**) and the logging function is disabled (block **110**), or until an acknowledgement message is received back from an attached data logging device **26** (block **104**).

If an acknowledgement signal is received back from a data logging device (block **104**), the processor of the digital recording device **24** starts a recording thread as explained below in connection with FIG. **9**.

FIG. **8** illustrates a program that may be executed by the data logging device **26** at start-up. The program begins at block **120** where the processor of the data logging device **26** enters a loop wherein the data logging device **26** waits to receive data from the digital recording device **24**. If a data signal is received (block **120**), the processor of the data logging device **26** waits a predetermined length of time (block **122**) and then sends an acknowledgement signal back to the digital recording device **24** (block **124**). (The returned acknowledgement signal is the signal the digital recording device is seeking at block **104** of FIG. **1**.) The data logging device **26** then stores, analyzes, and/or forwards the data structures **58**, **70** received from the digital recording device **24** as explained in further detail below.

From the foregoing, persons of ordinary skill in the art will appreciate that the example of FIGS. **7**–**8** employs a handshaking method wherein the presence or absence of an acknowledgement signal is used to detect if a data logging device **26** is coupled to a digital recording device **24**. In the example of FIGS. **7**–**8**, if no data logging device **26** is attached, the digital recording device **24** will not log data reflecting its usage pattern. In the illustrated example, once the data logging functionality is disabled, there will be no attempt to re-enable the same unless a re-set or power-up event of the digital recording device **24** occurs.

Persons of ordinary skill in the art will further appreciate that the programs of FIGS. **7**–**8** may be eliminated in examples wherein the digital recording device **24** alone implements the apparatus **50** and/or in examples in which the digital recording device **24** implements the apparatus **50** in cooperation with the central office **10** without the use of a data logging device **26**. Although as explained above, there are numerous ways to implement the functionality of the apparatus **50**, for simplicity of explanation, the discussion of FIGS. **9**–**14** below assumes the presence of a data logging device **26** wherein the data logging device **26** implements the interval detector **90**, the difference detector **92**, the non-real time content identified **94** and the recapture identifier **96**, and the data logging device periodically exports the data it collects and analyzes to the central office **10** for compilation with data from other monitored sites **12**, **14**.

Assuming that the digital recording device **24** has initiated the record thread (block **114**, FIG. **7**), the record thread begins by entering a loop wherein it awaits the receipt of program data by the digital recording device **24** (block **200**). Program data may be received from many sources. For example, it may be tuned data developed by a tuner of the digital recording device **24**, it may be data received by the digital recording device **24** from an external device via, for example, an RF input, a Satellite input, or an S-video input, etc. When program data is received (block **200**), the digital recording device **24** packetizes the received program data into one or more datastreams (block **202**). For example, if the digital recording device **24** is a TiVo® digital/personal video recorder, the received program data may be packetized into a stream of video packets and a stream of audio packets. The time stamper **52** then writes the present time into one or more headers of the packets in the datastream(s) (block **204**). The recording of these presentation times in the datastream (s) may be intended, for example, for use in synchronizing a video datastream and an audio datastream, but is also used for identifying non-real time activity as explained further below.

Other data may also be written into the datastream(s) (block **206**). For example, any or all of a program identifier, a digital recording device serial number, the input port from which the data was received, the channel number, the station name, the program name, and the station identifier associated with the program being recorded may be recorded in the datastream(s). After the packet(s) are completely populated (block **206**), they are stored in the memory **54** (block **208**).

The digital recording device **24** then determines if a playback command (e.g., play, rewind, pause, fast forward, etc.) has been received (block **210**). If so, the digital recording device **24** initiates the playback thread (block **212**) as explained below in connection with FIG. **10**. Otherwise, control advances to block **214**.

At block **214**, the digital recording device **24** determines if the logging functionality has been initiated. If so, the digital recording device **24** initiates the export thread as explained below in connection with FIG. **11** (block **216**). Otherwise, if the logging functionality has not been initiated (block **214**), control returns to block **200** without passing through block **216**.

Persons of ordinary skill in the art will appreciate that typically only one playback thread and one export thread will execute at a given time.

Assuming a playback thread has been initiated, at block **220** of FIG. **10**, the digital recording device **24** retrieves the packet(s) of one or more datastream(s) corresponding to the program specified by the user from the memory **54** for playback in accordance with the user's instructions. The digital recording device **24** then determines if the logging functionality is enabled by, for example, checking a flag set at block **110** of FIG. **7** (block **222**). If the logging functionality is not enabled (block **206**), control advances to block **232**. Otherwise control proceeds to block **224**.

Assuming for purposes of discussion that the logging functionality is enabled (block **222**), the digital recording device **24** creates a log entry data structure (e.g., data structure **58** or data structure **70**) (block **224**). The time stamper **52** then writes one or more of the current time and date into the playback fields **60**, **62** of the data structure **58**, **70** (block **226**). Control then advances to block **228**.

At block **228**, the presentation time retriever **56** extracts the presentation/recording time from the packet(s) currently being played (block **228**). Other data may also be extracted from the packet(s) (block **228**). For example, any or all of a program identifier, a digital recording device serial number, the input port from which the data was received, the channel number, the station name, the program name, and the station identifier associated with the recorded program may be extracted from the packet(s) that are currently being played.

The data extracted by the presentation time retriever **56** is written into the corresponding field of the log entry data structure **58**, **70** such that the extracted data is associated

with the playback time of the corresponding packet(s) (block **230**). Control then advances to block **232**.

Irrespective of whether control reaches block **232** from block **222** or block **230**, at block **232** the digital recording device **24** outputs the packet(s) to an information presenting device such as a television. The digital recording device **24** then determines if all of the packets of the recorded program requested by the user of the digital recording device **24** have been output (block **234**). If so, the playback thread terminates. Otherwise, control returns to block **220**.

Control continues to loop through blocks **220**–**234** until all of the packet(s) associated with the recorded program requested by the consumer have been output (block **234**). The playback thread then terminates as explained above.

Assuming the export thread has been initiated, at block **240** of FIG. **11**, the digital recording device **24** determines if a minimum interval of time has passed since the last message containing a log entry data structure was exported to, for example, the data logging device **26**. If not, control remains at block **240** until that time interval has passed. The digital recording device **24** then attempts to export the next message containing a log entry data structure **58**, **70** (block **242**). The message sent at block **242** is queued in a low priority thread such that it does not impact the perceptible usage of the device **24** by the consumer. The queue may contain multiple messages. The messages contained in the queue may be sent in the order they were created or out of order. In the case where out or order exporting is possible, sequence numbers may be used to re-order the messages upon receipt.

At block **244**, the digital recording device **24** determines if the message is still pending in the queue. If the message has been sent, control returns to block **240** to await the time to send the next message containing the next log entry data structure **58**, **70**. Otherwise, the digital recording device **24** determines if the maximum time interval (e.g., 30 seconds) between exporting of messages containing log entry data structures has passed (block **246**). If that allotted time has not passed (block **246**), control returns to block **244**. Otherwise, if the maximum permitted delay in transmitting a queued message has passed (block **246**), the message containing the excessively delayed message containing the log entry data structure is converted to a high priority message and immediately exported (block **248**). Control then returns to block **240**.

Control continues to loop through blocks **240**–**248** until the export thread is terminated by, for example, turning off the digital recording device **24**. Thus, once the export thread is initiated (block **216**, FIG. **9**), it will continue to export messages containing log entry data structures **58**, **70** until operation of the digital recording device **24** is interrupted. Since, in the illustrated example, a data structure **58**, **70** is created every 2.7 seconds, the digital recording device **24** may export up to 32,000 data structures **58**, **70** per day describing what the user is viewing via the digital recording device **24** at any given time. Of course, asynchronous models which create data structures based on detected events in addition to or in place of the expiration of set time intervals would export different quantities of data structures.

An example program for implementing the interval detector **90** and the difference detector **92** is shown in FIGS. **12**A–**12**C. The example program of FIGS. **12**A–**12**C may be executed, for example, by the data logging device **26** and/or the central office **10**. Alternatively, the program of FIGS. **12**A–**12**C could be executed by the digital recording device **24**.

The program of FIGS. **7**A–**7**C begins at block **270** where the interval detector **90** retrieves a pair of contiguous log entry data structures **58**, **70**. The interval detector **90** then calculates a presentation time difference (i.e., a recording time interval) by subtracting the presentation time value recorded in the older data structure **58**, **70** from the presentation time value recorded in the newer data structure **58**, **70** (block **272**). The interval detector **90** may then calculate a real time difference (i.e., a playback interval) between the data structures **58**, **70** by subtracting the playback time value recorded in the older data structure **58**, **70** from the playback time value recorded in the newer data structure **58**, **70** (block **274**). As explained above, block **272** may be eliminated if the log entry data structures **58**, **70** are created at a constant, fixed interval so that the real time difference between log entry data structures **58**, **70** is always a known constant.

After the recording time interval and playback time intervals are obtained (blocks **272** & **274**), the difference detector **92** determines if the presentation time difference (i.e., the recording interval) between the pair of log entry data structures **58**, **70** is zero (block **276**). If so, the recorded program was paused for the entire playback interval between the data structures (block **278**), that fact is logged and control advances to block **298** (FIG. **12**B). At block **298**, the interval detector **90** determines is the last pair of log entry data structures **58**, **70** have been analyzed. If not, control returns to block **270** (FIG. **12**A). Otherwise, control advances to block **300** where the recapture identifier **96** is invoked as explained below in connection with FIG. **14**.

If the presentation time difference (i.e., the recording interval) between the pair of log entry data structures **58**, **70** is not zero (block **276**), a full interval pause did not occur and control advances to block **280**.

At block **280**, the difference detector **92** determines if the recording interval is less than the playback interval. If the recording interval (i.e, the presentation time difference) is smaller than the playback interval (i.e., the real time interval) between the data structures **58**, **70** (block **280**), the corresponding portion of the recorded program has been paused or rewound and control advances to block **286**. If the presentation time difference is a positive value, then a brief pause event or slow motion playback occurred, the same is noted, and control returns to block **298** of FIG. **12**B.

If the presentation time difference is negative, a rewind event has occurred and control advances to block **288**. At block **288**, the non-real time content identifier **94** is invoked. As explained below in connection with FIGS. **13**A–**13**B, the non-real time content identifier **94** returns an identification of the segment(s) of the recorded program that were rewound. The returned segment names are written in the rewound column of a recapture table such as that shown in FIG. **6** (block **290**). Control then advances to block **298** of FIG. **12**B as explained above.

If the recording interval is not less than the playback interval (block **280**, FIG. **12**A), control advances to block **292** of FIG. **12**B. At block **292**, the difference detector **92** determines if the recording interval is equal to the playback interval. If the recording interval (i.e, the presentation time difference) is not equal to the playback interval (i.e., the real time interval) between the data structures **58**, **70** (block **292**), the corresponding portion of the recorded program has been skipped and/or fast forwarded and control advances to block **294**. At block **294**, the non-real time content identifier **94** is invoked. As explained below in connection with FIGS. **13**A–**13**B, the non-real time content identifier **94** returns an identification of the segment(s) of the recorded program that were skipped and/or fast forwarded. The returned segment

names are written in the advanced column of a table such as that shown in FIG. **6** (block **296**). Control then advances to block **298**.

If the recording interval (i.e, the presentation time difference) is equal to the playback interval (i.e., the real time interval) between the data structures **58**, **70** (block **292**), the corresponding portion of the recorded program has not been skipped and/or fast forwarded and control advances directly to block **298**.

Control continues to loop through blocks **270**–**298** until every pair of log entry data structures **58**, **70** in a block of such data (e.g., a block of data corresponding to 30 minutes of monitoring) have been analyzed for non-real time activity (block **298**). When that occurs, the recapture identifier **96** is invoked to operate on that block of data (block **300**). After the recapture identifier **96** analyzes the data as explained below in connection with FIG. **14**, control returns to block **270** to start the analysis of the next block of data.

As mentioned above, the non-real time content identifier **94** is invoked at block **288** of FIG. **12**C and block **294** of FIG. **12**B. The operation of the non-real time content identifier **94** will now be explained in connection with FIGS. **13**A–**13**B. The identify non-real time content routine begins at block **312** when the non-real time content identifier **94** retrieves the identity of the recorded program from one of the corresponding log entry data structures **58**, **70** and a program table such as the example program table shown in FIG. **5** for the recorded program from a database of program tables. The non-real time content identifier **94** then retrieves the recording times for the portion of the recorded program that experienced non-real time activity from the log entry data structures **58**, **70** (block **314**). The retrieved times are then normalized by, for example, subtracting the start time of the recorded program as recorded in the log entry data structures **58**, **70** from the recording times retrieved from the log entry data structures **58**, **70** (block **316**).

Once the retrieved times are normalized (block **316**), a pointer is set to point to the first segment identified in the program table corresponding to the recorded program (see FIG. **5**) and the start and end times of the pointed to interval are retrieved from the program table (block **318**). The non-real time content identifier **94** then determines if the start time for the interval being identified is substantially equal to (e.g., falls within some range of) the start time of the interval pointed to in the program table (block **320**). If not, the pointer is moved to point to the next segment in the program table (block **322**), and control returns to block **320**. Control continues to loop through blocks **320**–**322** until a match between the start time for the interval being identified and the start time of the interval pointed to in the program table occurs (block **320**). When a match occurs, the segment corresponding to the matched start time is recorded and control then advances to block **324** (FIG. **13**B).

At block **324**, the non-real time content identifier **94** determines if the end time for the interval being identified is substantially equal to (e.g., falls within some range of) the end time of the interval pointed to in the program table. If not, the pointer is moved to point to the next segment in the program table (block **326**), and control returns to block **324**. Control continues to loop through blocks **324**–**326** until a substantial match between the end time for the interval being identified and the end time of the interval pointed to in the program table occurs (block **324**). When a match occurs, the segment corresponding to the matched end time is recorded and control then advances to block **328**.

If the non-real time content identifier **94** has recorded a segment corresponding to the start time of the portion of the recorded program that experienced non-real time activity and a segment corresponding to the end time of the portion of the recorded program that experienced non-real time activity (block **328**), the non-real time content identifier **94** records the identities of the segment corresponding to the start time, the segment corresponding to the end time, and any segment(s) falling between the start time and the end time of the portion of the recorded program that experienced non-real time activity. Persons of ordinary skill in the art will appreciate that if the start and end times identified at block **320** and **324** correspond to a single segment, then only one segment is recorded at block **330**. With the segment identity (ies) recorded (block **330**), control returns to the calling block (i.e., block **288** or block **294**).

If the non-real time content identifier **94** has not recorded a segment corresponding to the start time of the portion of the recorded program that experienced non-real time activity and a segment corresponding to the end time of the portion of the recorded program that experienced non-real time activity (block **328**), the non-real time content identifier **94** records the identity of the portion of the recorded program that experienced non-real time activity as "unidentified segment" (block **332**). Control then returns to the calling block (i.e., block **288** or block **294**).

It will be appreciated by persons of ordinary skill in the art that the start time of the portion of the recorded program employed at block **320** is the normalized start time retrieved from the earlier created log entry data structure **58**, **70** if the non-real time content identifier **94** is invoked from block **294** of FIG. **12**B (i.e., a skipped and or fast forwarded portion is being identified). Similarly, the end time of the portion of the recorded program employed at block **324** is the normalized end time retrieved from the later created log entry data structure **58**, **70** if the non-real time content identifier **94** is invoked from block **294** of FIG. **12**B (i.e., a skipped and or fast forwarded portion is being identified).

Conversely, it will be appreciated by persons of ordinary skill in the art that the start time of the portion of the recorded program employed at block **320** is the normalized start time retrieved from the later created log entry data structure **58**, **70** if the non-real time content identifier **94** is invoked from block **288** of FIG. **12**C (i.e., a rewound portion is being identified). Similarly, the end time of the portion of the recorded program employed at block **324** is the normalized end time retrieved from the earlier created log entry data structure **58**, **70** if the non-real time content identifier **94** is invoked from block **288** of FIG. **12**C (i.e., a rewound portion is being identified).

As mentioned above, the recapture identifier **96** is invoked at block **300** of FIG. **12**B. The operation of the recapture identifier **96** will now be explained in connection with FIG. **14**. The recapture routine begins at block **350** when the recapture identifier **96** determines if there are any rewound segments identified in the recapture table (e.g., the recapture table of FIG. **6**). If there are no rewound segments in the recapture table (block **350**), the recapture routine immediately terminates and control returns to block **270** of FIG. **12**A. If there are rewound segments in the recapture table (block **350**), the recapture identifier **96** determines if there are any fast forwarded and/or skipped segments identified in the recapture table (block **352**). If there are no fast forwarded and/or skipped segments in the recapture table (block **352**), the recapture routine immediately terminates and control returns to block **270** of FIG. **12**A. If there are fast forwarded and/or skipped segments in the recapture table (block **352**), control advances to block **354**.

At block **354**, the recapture identifier **96** retrieves the first fast forwarded and/or skipped segment from the recapture table. The recapture identifier **96** then compares the retrieved fast forwarded and/or skipped segment to the segment(s) listed in the recapture table as rewound segments (block **356**). If the retrieved fast forwarded and/or skipped segment appears on the rewound list of the recapture table (block **356**), the retrieved segment is deleted from both the advanced list and the rewound list of the recapture table (block **358**) and control advances to block **360**. Otherwise, control advances directly from block **356** to block **360**.

At block **360**, the recapture identifier **96** determines if every segment in the advanced segment list of the recapture table has been compared to the segment(s) appearing in the rewound list of the recapture table. If not, control advances to block **362** where the next segment appearing in the advanced list of the recapture table is retrieved. Control then returns to block **356**.

Control continues to loop through blocks **356**–**360** until every segment in the advanced list has been compared to the segment(s) in the rewound list (block **360**). The recapture routine then terminates and control returns to block **270** of FIG. **12**A.

FIG. **15** is a block diagram of an example device **1000** capable of implementing the apparatus and methods disclosed herein. In particular, the device **1000** can be, for example, a server, a personal computer, a personal digital assistant (PDA), an Internet appliance, a DVD player, a CD player, a digital video recorder, a personal video recorder, a set top box, device, an application specific device such as the data logging device **26**, or any other type of computing device.

The system **1000** of the instant example includes a processor **1012**. For example, the processor **1012** can be implemented by one or more Intel® microprocessors from the Pentium® family, the Itanium® family or the XScale® family. Of course, other processors from other families are also appropriate.

The processor **1012** is in communication with a main memory including a volatile memory **1014** and a non-volatile memory **1016** via a bus **1018**. The volatile memory **1014** may be implemented by Synchronous Dynamic Random Access Memory (SDRAM), Dynamic Random Access Memory (DRAM), RAMBUS Dynamic Random Access Memory (RDRAM) and/or any other type of random access memory device. The non-volatile memory **1016** may be implemented by flash memory and/or any other desired type of memory device. Access to the main memory **1014**, **1016** is typically controlled by a memory controller (not shown) in a conventional manner.

The device **1000** also includes a conventional interface circuit **1020**. The interface circuit **1020** may be implemented by any type of well known interface standard, such as an Ethernet interface, a universal serial bus (USB), a high performance serial bus such as an IEEE 1394 bus or firewire, and/or a third generation input/output (3GIO) interface.

One or more input devices **1022** may be connected to the interface circuit **1020**. The input device(s) **1022** permit a user to enter data and commands into the processor **1012**. The input device(s) can be implemented by, for example, an infrared hand held controller, a keyboard, a mouse, a touch-screen, a track-pad, a trackball, isopoint and/or a voice recognition system.

One or more output ports **1024** are also connected to the interface circuit **1020**. The output ports **1024** can be implemented, for example, by serial ports (e.g., DB-9, USB, etc) or parallel ports.

The interface circuit **1020** also includes a communication device (e.g., communication device **56**) such as a modem or network interface card to facilitate exchange of data with external computers via a network **1026** (e.g., an Ethernet connection, a digital subscriber line (DSL), a telephone line, coaxial cable, a cellular telephone system, etc.).

The device **1000** also includes one or more mass storage devices **1028** for storing software and data. Examples of such mass storage devices **1028** include flashcards, jump drives, floppy disk drives, hard drive disks, compact disk drives and digital versatile disk (DVD) drives.

The device **1000** may also include a tuner **1030** for receiving broadcast programs and/or an MPEG decoder **1032** for compressing, encoding, decompressing and/or decoding the received programs.

The device of FIG. **15** may be representative of the digital recording device **24** and/or the data logging device **26**. The data logging device **26** may be similarly constructed to the digital recording device **24**, but it would typically not include input device(s) **1022**, a tuner **1030** or an MPEG decoder **1032**.

Although the above examples focused on video broadcast programs intended for consumption on a visual information presenting device **22** (e.g., a television) coupled to a digital recording device **24** such as a digital video recorder, a personal video recorder and/or a set top box, persons of ordinary skill in the art will appreciate that the methods and apparatus described herein could also be used with broadcast programs intended for consumption on a purely audio information presenting device (e.g., a radio) and/or to analog recording devices (e.g., a cassette recorder and/or a VCR).

Further, although certain example methods and apparatus have been described herein, the scope of coverage of this patent is not limited thereto. On the contrary, this patent covers all methods, apparatus and articles of manufacture fairly falling within the scope of the appended claims either literally or under the doctrine of equivalents.

What is claimed is:

**1**. A method of detecting portions of a recorded program that were not played back in real time comprising:

storing a plurality of recording times of sections of the recorded program in association with playback times of the recorded program; and

comparing playback intervals between sequential pairs of the playback times with corresponding recording intervals between sequential pairs of the recording times to determine if any portion of the recorded program was not played back in real time.

**2**. A method as defined in claim **1** wherein the plurality of recording times substantially correspond to times at which respective sections of the recorded program are recorded by a local device.

**3**. A method as defined in claim **1** further comprising storing the recorded program as a plurality of data packets with the recording times stored in headers of the data packets.

**4**. A method as defined in claim **3** wherein each of the recording times is stored in association with at least one respective packet.

**5**. A method as defined in claim **3** wherein storing the plurality of recording times of sections of the recorded program in association with playback times of the recorded program further comprises:

retrieving a recording time from a header of a data packet being played back; and

storing the recording time in association with one of the playback times.

6. A method as defined in claim 1 wherein the playback times substantially correspond to times at which respective sections of the recorded program are played back by a local device.

7. A method as defined in claim 1 wherein storing the plurality of recording times of sections of the recorded program in association with playback times of the recorded program comprises:

creating a plurality of log entry data structures; and

storing respective ones of the playback times in respective ones of the plurality of log entry data structures.

8. A method as defined in claim 7 further comprising storing respective ones of the plurality of recording times in respective ones of the plurality of log entry data structures.

9. A method as defined in claim 1 wherein storing the plurality of recording times of sections of the recorded program in association with playback times of the recorded program comprises:

creating a plurality of log entry data structures; and

storing respective ones of the recording times in respective ones of the plurality of log entry data structures.

10. A method as defined in claim 1 wherein, if a playback interval is different than a corresponding one of the recording intervals, then a portion of the recorded program at least partially corresponding to the recording interval was not played back in real time.

11. A method as defined in claim 10 further comprising identifying the portion of the recorded program that was not played back in real time.

12. A method as defined in claim 11 wherein identifying the portion of the recorded program that was not played back in real time comprises:

comparing a start time of the recording interval associated with the portion of the recorded program that was not played back in real time to known start times associated with segments of the recorded program;

comparing an end time of the recording interval associated with the portion of the recorded program that was not played back in real time to known end times associated with segments of the recorded program; and

if the start time of the recording interval associated with the portion of the recorded program that was not played back in real time substantially matches the known start time of a first segment, and the end time of the recording interval associated with the portion of the recorded program that was not played back in real time substantially matches the known end time of the first segment, identifying the first segment as the portion of the recorded program that was not played back in real time.

13. A method as defined in claim 12 wherein if the start time of the recording interval associated with the portion of the recorded program that was not played back in real time substantially matches the known start time of the first segment, and the end time of the recording interval associated with the portion of the recorded program that was not played back in real time substantially matches the known end time of a second segment, identifying the first segment, the second segment and any segments between the first and second segments as the portion of the recorded program that was not played back in real time.

14. A method as defined in claim 12 wherein the start times and the end times of the recording interval associated with the portion of the recorded program that was not played back in real time are normalized to a broadcast duration of the recorded program.

15. A method as defined in claim 11 wherein identifying the portion of the recorded program that was not played back in real time comprises:

comparing the recording interval associated with the portion of the recorded program that was not played back in real time to known intervals associated with segments of the recorded program; and

if the recording interval associated with the portion of the recorded program that was not played back in real time overlaps with at least a predetermined amount of at least one of the known intervals, identifying at least one segment associated with the at least one of the known intervals as the portion of the recorded program that was not played back in real time.

16. A method as defined in claim 15 wherein the recording interval associated with the portion of the recorded program that was not played back in real time is normalized to a broadcast duration of the recorded program.

17. A method as defined in claim 11 wherein if a first portion of the recorded program is identified as having been skipped, and a second portion of the recorded program is identified as having been rewound, comparing the first and second portions to determine if at least a segment of the first portion was viewed with the second portion of the recorded program.

18. A method as defined in claim 1 wherein the recorded program is recorded on a local recording device from a broadcast signal.

19. A method as defined in claim 18 wherein the local recording device comprises at least one of: a digital video recorder, a set top box, a personal video recorder, and an analog video recorder.

20. A method as defined in claim 1 further comprising storing at least one of: (a) a channel identifier, (b) a broadcast station identifier, (c) an input source code, (d) a program identifier, (e) a name of the recorded program, (f) a universal network identifier, (g) a universal program identifier, (h) a digital recording device serial number, (i) a tuner identification number, and (j) a playback mode code in association with the playback time.

21. A method as defined in claim 1 wherein (a) if a recording interval is zero, then playback was paused for a duration of the corresponding playback interval, (b) if a recording interval has a first sign, then a corresponding portion of the recorded program was played at a rate different than real time, and (c) if a recording interval has a second sign opposite the first sign, then a rewind event has occurred.

22. A method as defined in claim 1 further comprising transmitting at least some of the recording times to a collection facility, and wherein comparing playback intervals between sequential pairs of the playback times with corresponding recording intervals between sequential pairs of the recording times to determine if a portion of the recorded program was not played back in real time is performed at the collection facility.

23. A method as defined in claim 22 further comprising transmitting at least some of the playback times to the collection facility.

24. An article of manufacture storing machine readable instructions which, when executed, cause a machine to:

store a plurality of recording times of sections of a recorded program in association with playback times of the recorded program; and

compare playback intervals between sequential pairs of the playback times with corresponding recording intervals between sequential pairs of the recording times to

determine if any portion of the recorded program was not played back in real time.

**25**. An article of manufacture as defined in claim **24** wherein the plurality of recording times substantially correspond to times at which respective sections of the recorded program are recorded by a local device.

**26**. An article of manufacture as defined in claim **24** wherein the machine readable instructions cause the machine to store the plurality of recording times as a plurality of data packets with the recording times stored in headers of the data packets.

**27**. An article of manufacture as defined in claim **26** wherein the machine readable instruction cause the machine to store each of the recording times in association with at least one respective packet.

**28**. An article of manufacture as defined in claim **26** wherein the machine readable instructions cause the machine to store the plurality of recording times of the sections of the recorded program in association with the playback times of the recorded program by:

retrieving a recording time from a header of a data packet being played back; and

storing the recording time in association with one of the plurality of playback times.

**29**. An article of manufacture as defined in claim **24** wherein the playback times substantially correspond to times at which respective sections of the recorded program are played.

**30**. An article of manufacture as defined in claim **24** wherein the machine readable instructions cause the machine to store the plurality of recording times of the sections of the recorded program in association with the playback times of the recorded program by:

creating a plurality of log entry data structures; and

storing respective ones of the playback times in respective ones of the plurality of log entry data structures.

**31**. An article of manufacture as defined in claim **30** wherein the machine readable instructions cause the machine to store respective ones of the plurality of recording times in respective ones of the plurality of log entry data structures.

**32**. An article of manufacture as defined in claim **24** wherein the machine readable instructions cause the machine to store the plurality of recording times of the sections of the recorded program in association with the playback times of the recorded program by:

creating a plurality of log entry data structures; and

storing respective ones of the recording times in respective ones of the plurality of log entry data structures.

**33**. An article of manufacture as defined in claim **24** wherein, if a playback interval is different than a corresponding one of the recording intervals, then a portion of the recorded program corresponding to the recording interval was not played back in real time.

**34**. An article of manufacture as defined in claim **33** wherein the machine readable instructions cause the machine to identify the portion of the recorded program that was not played back in real time.

**35**. An article of manufacture as defined in claim **34** wherein the machine readable instructions cause the machine to identify the portion of the recorded program that was not played back in real time by:

comparing a start time of the recording interval associated with the portion of the recorded program that was not played back in real time to known start times associated with segments of the recorded program;

comparing an end time of the recording interval associated with the portion of the recorded program that was not played back in real time to known end times associated with segments of the recorded program; and

if the start time of the recording interval associated with the portion of the recorded program that was not played back in real time substantially matches the known start time of a first segment, and the end time of the recording interval associated with the portion of the recorded program that was not played back in real time substantially matches the known end time of the first segment, identifying the first segment as the portion of the recorded program that was not played back in real time.

**36**. An article of manufacture as defined in claim **35** wherein if the start time of the recording interval associated with the portion of the recorded program that was not played back in real time substantially matches the known start time of the first segment, and the end time of the recording interval associated with the portion of the recorded program that was not played back in real time substantially matches the known end time of a second segment, identifying the first segment, the second segment and any segments between the first and second segments as the portion of the recorded program that was not played back in real time.

**37**. An article of manufacture as defined in claim **35** wherein the start times and the end times of the recording interval associated with the portion of the recorded program that was not played back in real time are normalized to a broadcast duration of the recorded program.

**38**. An article of manufacture as defined in claim **34** wherein the machine readable instructions cause the machine to identify the portion of the recorded program that was not played back in real time by:

comparing the recording interval associated with the portion of the recorded program that was not played back in real time to known intervals associated with segments of the recorded program; and

if the recording interval associated with the portion of the recorded program that was not played back in real time overlaps with at least a predetermined amount of one of the known intervals, identifying at least one segment associated with the at least one of the known intervals as the portion of the recorded program that was not played back in real time.

**39**. An article of manufacture as defined in claim **38** wherein the recording interval associated with the portion of the recorded program that was not played back in real time is normalized to a broadcast duration of the recorded program.

**40**. An article of manufacture as defined in claim **34** wherein if a first portion of the recorded program is identified as having been skipped, and a second portion of the recorded program is identified as having been rewound, the machine readable instructions cause the machine to compare the first and second portions to determine if at least a segment of the first portion was viewed with the second portion of the recorded program.

**41**. An article of manufacture as defined in claim **24** wherein the recorded program is recorded on a local recording device from a broadcast signal.

**42**. An article of manufacture as defined in claim **41** wherein the local recording device comprises at least one of: a digital video recorder, a set top box, a personal video recorder, and an analog video recorder.

**43**. An article of manufacture as defined in claim **24** wherein the machine readable instructions cause the

machine to store at least one of: (a) a channel identifier, (b) a broadcast station identifier, (c) an input source code, (d) a program identifier, (e) a name of the recorded program, (f) a universal network identifier, (g) a universal program identifier, (h) a digital recording device serial number, (i) a tuner identification number, and (j) a playback mode code in association with the playback time.

**44**. An article of manufacture as defined in claim **24** wherein (a) if a recording interval is zero, then playback was paused for a duration of the corresponding playback interval, (b) if a recording interval has a first sign, then a corresponding portion of the recorded program was played at a rate different than real time, and (c) if a recording interval has a second sign opposite the first sign, then a rewind event has occurred.

**45**. An article of manufacture as defined in claim **24** wherein the machine readable instructions further cause the machine to transmit at least some of the recording times to a collection facility, and wherein the machine readable instructions that cause the machine to compare playback intervals between sequential pairs of the playback times with corresponding recording intervals between sequential pairs of the recording times to determine if a portion of the recorded program was not played back in real time are executed at a collection facility.

**46**. An article of manufacture as defined in claim **45** wherein the machine readable instructions further cause the machine to transmit at least some of the playback times to the collection facility.

**47**. An apparatus to detect portions of a recorded program that were not played back in real time comprising:

a time stamper to record a plurality of recording times in association with a recorded program;

a memory to store the recorded program;

a presentation time retriever to retrieve the recording time of a section of the recorded program being played back and to store the retrieved recording time in association with a playback time; an interval detector to identify recording intervals between sequential ones of the plurality of recording times; and

a difference detector to determine if a difference exists between a recording interval and a playback interval, wherein the difference detector compares playback intervals between sequential pairs of playback times with corresponding recording intervals between sequential pairs of recording times to determine if any portion of the recorded program was not played back in real time.

**48**. An apparatus as defined in claim **47** wherein the playback time is provided by the time stamper.

**49**. An apparatus as defined in claim **47** wherein the recorded program is stored as a plurality of packets, and the time stamper records the recording times in headers of the packets.

**50**. An apparatus as defined in claim **47** further comprising a non-real time content identifier responsive to a detected difference between the recording interval and the playback interval to identify a portion of the recorded program that was not played back in real time.

**51**. An apparatus as defined in claim **50** wherein the non-real time content identifier identifies the portion of the recorded program that was not played back in real time by:

comparing a start time of the recording interval associated with the portion of the recorded program that was not played back in real time to known start times associated with segments of the recorded program;

comparing an end time of the recording interval associated with the portion of the recorded program that was not played back in real time to known end times associated with segments of the recorded program; and

if the start time of the recording interval associated with the portion of the recorded program that was not played back in real time substantially matches the known start time of a first segment, and the end time of the recording interval associated with the portion of the recorded program that was not played back in real time substantially matches the known end time of the first segment, the non-real time content identifier identifies the first segment as the portion of the recorded program that was not played back in real time.

**52**. An apparatus as defined in claim **51** wherein if the start time of the recording interval associated with the portion of the recorded program that was not played back in real time substantially matches the known start time of the first segment, and the end time of the recording interval associated with the portion of the recorded program that was not played back in real time substantially matches the known end time of a second segment, the non-real time content identifier identifies the first segment, the second segment and any segments between the first and second segments as the portion of the recorded program that was not played back in real time.

**53**. An apparatus as defined in claim **51** wherein the start times and the end times of the recording interval associated with the portion of the recorded program that was not played back in real time are normalized to a broadcast duration of the recorded program.

**54**. An apparatus as defined in claim **50** wherein the non-real time content identifier identifies the portion of the recorded program that was not played back in real time by:

comparing the recording interval associated with the portion of the recorded program that was not played back in real time to known intervals associated with segments of the recorded program; and

if the recording interval associated with the portion of the recorded program that was not played back in real time overlaps with at least a predetermined amount of at least one of the known intervals, identifying at least one segment associated with the at least one of the known intervals as the portion that was not played back in real time.

**55**. An apparatus as defined in claim **54** wherein the recording interval associated with the portion of the recorded program that was not played back in real time is normalized to a broadcast duration of the recorded program.

**56**. An apparatus as defined in claim **50** further comprising a recapture identifier to determine if a portion of the recorded program that was passed at a rate greater than real time was viewed in real time as a result of rewinding the recorded program.

**57**. An apparatus as defined in claim **56** wherein at least one of the time stamper, the memory and the presentation time retriever are located at a monitored home, and at least one of the interval detector, the difference detector, the non-real time content identifier, and the recapture identifier are located at least one of a collection facility and a data logging device.

**58**. An apparatus as defined in claim **47** wherein the plurality of recording times substantially correspond to times at which respective sections of the recorded program are recorded by a local device.

59. An apparatus as defined in claim 47 wherein the recorded program is stored in the memory as a plurality of data packets, and the recording times are stored in headers of the data packets.

60. An apparatus as defined in claim 59 wherein each of the recording times is stored in association with at least one respective packet.

61. An apparatus as defined in claim 47 wherein the playback time substantially corresponds to a time at which the section of the recorded program is played.

62. An apparatus as defined in claim 47 wherein the playback time is stored in a log entry data structure.

63. An apparatus as defined in claim 47 wherein the presentation time retriever stores the retrieved recording time in a log entry data structure.

64. An apparatus as defined in claim 47 wherein the presentation time retriever stores the retrieved recording time in association with a playback time by storing the retrieved recording time in a log entry data structure representative of a playback time.

65. An apparatus as defined in claim 47 wherein, if the difference detector detects a difference between a playback interval and one of the recording intervals, then a portion of the recorded program corresponding to the one of the recording intervals was not played back in real time.

66. An apparatus as defined in claim 47 wherein the recorded program is recorded on a local recording device from a broadcast signal.

67. An apparatus as defined in claim 66 wherein the local recording device comprises at least one of: a digital video recorder, a set top box, a personal video recorder, and an analog video recorder.

68. An apparatus as defined in claim 47 wherein the presentation time retriever stores at least one of: (a) a channel identifier, (b) a broadcast station identifier, (c) an input source code, (d) a program identifier, (e) a name of the recorded program, (f) a universal network identifier, (g) a universal program identifier, (h) a digital recording device serial number, (i) a tuner identification number, and (j) a playback mode code in association with the playback time.

69. An apparatus as defined in claim 47 wherein (a) if a recording interval is zero, then playback was paused for a duration of the corresponding playback interval, (b) if a recording interval has a first sign, then a corresponding portion of the recorded program was played at a rate different than real time, and (c) if a recording interval has a second sign opposite the first sign, then a rewind event has occurred.

70. A method of identifying a portion of a recorded program that was not played in real time comprising:

computing a first difference between a first recording time and a second recording time; and

comparing the first difference to a second difference between a first playback time and a second playback time to determine if a portion of the recorded program associated with the first difference was played in real time.

71. A method as defined in claim 70 further comprising computing the second difference between the first playback time and the second playback time.

72. A method as defined in claim 70 further comprising receiving a first data structure containing the first playback time and the first recording time.

73. A method as defined in claim 72 further comprising receiving a second data structure containing the second playback time and the second recording time.

74. A method as defined in claim 70 wherein, if the second difference is larger than the first difference, a portion of the recorded program was at least one of skipped and fast forwarded.

75. An apparatus to detect portions of a recorded program that were not played in real time comprising:

an interval detector to compute a recording interval between a first recording time and a second recording time; and

a difference detector to compare a playback interval and the recording interval to determine if the recording interval is indicative of real time playback of a corresponding portion of the recorded program.

76. An apparatus as defined in claim 75 wherein the playback interval is a difference between a first playback time and a second playback time.

77. An apparatus as defined in claim 76 wherein the playback interval is computed by the interval detector.

78. An apparatus as defined in claim 75 further comprising a non-real time content identifier to identify the corresponding portion of the recorded program that was not played back in real time.

79. An apparatus to identify a portion of a recorded program that was not played in real time comprising:

at least one processor;

a memory coupled to the at least one processor and storing machine readable instructions which, when executed by the at least one processor, cause the at least one processor to:

compute a first difference between a first recording time and a second recording time; and

compare the first difference to a second difference between a first playback time and a second playback time to determine if a portion of the recorded program associated with the first difference was played in real time.

* * * * *