METHOD FOR CONTENT DELIVERY INVOLVING A POLICY DATABASE

INVENTORS: Per Karlsson, Tallinage (SE); Victor Souza, Stockholm (SE)

ASSIGNEE: Telefonaktiebolaget L M Ericsson (Publi), Stockholm (SE)

Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 425 days.

Appl. No.: 13/410,932

PCT Filed: Dec. 18, 2008

PCT No.: PCT/SE2008/051507
§ 371(e)(1), (2), (4) Date: Jun. 20, 2011

PCT Pub. No.: WO2010/071516
PCT Pub. Date: Jun. 24, 2010

Prior Publication Data

Int. Cl.
H04L 29/08 (2006.01)
H04L 29/06 (2006.01)

U.S. Cl.
CPC  H04L 67/104 (2013.01); H04L 65/602 (2013.01); H04L 67/1076 (2013.01); H04L 67/2823 (2013.01); H04L 67/2828 (2013.01); H04L 69/24 (2013.01)

Field of Classification Search
CPC : H04L 65/602; H04L 67/104; H04L 67/1076; H04L 67/2823; H04L 67/2828; H04L 69/20
USPC : 709/224, 217

ABSTRACT
The present invention comprises a content delivery system for delivering content to clients. The content delivery system comprises at least one mobile object adapted to be executed on the content delivery system and on other content delivery systems or multimedia devices adapted for mobile objects. Each mobile object comprises a media file and is further adapted to determine the capabilities of the other content delivery systems or multimedia devices and to monitor data related to the clients’ access of the media file. The content delivery system also comprises a transcoding unit for transcoding the media file to another media file prior sending it to the other content delivery system or multimedia device. The invention solves problems that can arise when sending large media files to content delivery systems or multimedia devices having limited capabilities.

11 Claims, 11 Drawing Sheets
References Cited

U.S. PATENT DOCUMENTS

6,970,902 B1  11/2005  Moon
7,024,466 B2  4/2006  Osten et al.
7,155,531 B1  12/2006  Lango et al.
7,415,537 B1*  8/2008  Maes .............................. 709/246
8,150,897 B2  4/2012  Shoji et al.
8,260,881 B1  9/2012  Paleja et al.
8,266,357 B2*  9/2012  Knowles et al. ................. 710/240
8,423,496 B1  4/2013  Hauser
8,510,596 B1*  8/2013  Gupta et al. ..................... 714/15
2004/0010990 A1  1/2004  Manzino

OTHER PUBLICATIONS

Department of Computer Science/ Thayer School of Engineering,
Dartmouth College, 1999.

* cited by examiner
EXECUTE MOBILE OBJECT

MONITOR DATA RELATED TO THE ACCESS OF THE MEDIA FILE STORED IN THE MOBILE OBJECT AND/OR IN THE EXECUTION ENVIRONMENT

REMOVE MEDIA FILE?

MOVE MEDIA FILE TO OTHER SERVER?

HALT AND DELETE MOBILE OBJECT

HALT ACCESS TO MOBILE OBJECT AND PAUSE ACTIVE CONNECTIONS

SEND MOBILE OBJECT TO SECOND SERVER

Figure 3A
COPY MOBILE OBJECT?

CREATE COPY OF THE MOBILE OBJECT

KEEP EXECUTION STATES IN COPY?

RESET EXECUTION STATES IN COPY

SEND COPY OF MOBILE OBJECT

Figure 3B
Figure 7

1. Execute Parent Mobile Object
2. Monitor data related to the access of the media file
3. Copy Mobile Object?
   - Yes (522)
     - Send Child Mobile Object excluding media file
     - Receive capabilities
   - No (703)
4. Transcoding needed?
   - Yes (706)
     - Transcode media file and create reference to child object
   - No (706)
5. Send media file (or reference)
Figure 8

EXECUTE PARENT MOBILE OBJECT 801

REQUEST FOR MEDIA FILE 802

COPY MOBILE OBJECT? 803

STREAM MEDIA FILE 809

SEND CHILD MOBILE OBJECT EXCLUDING MEDIA FILE 804

RECEIVE CAPABILITIES 805

TRANSCODING NEEDED? 806

TRANSCODE MEDIA FILE AND CREATE REFERENCE TO CHILD OBJECT 807

SEND MEDIA FILE (OR REFERENCE) 808
USER GENERATED EVENT

SEND REQUEST FOR MEDIA FILE

RECEIVE MOBILE OBJECT EXCLUDING MEDIA FILE

DETERMINING CAPABILITIES IN THE PORTABLE DEVICE

SEND CAPABILITIES

RECEIVE TRANSCODED MEDIA FILE

STORE TRANSCODED MEDIA FILE IN MOBILE OBJECT

Figure 10
METHOD FOR CONTENT DELIVERY INVOLVING A POLICY DATABASE

TECHNICAL FIELD OF THE INVENTION

The present invention relates to a method and a system for content delivery.

BACKGROUND

A content delivery network or content distribution network (CDN) is a system of computers (such as content servers) networked together across the Internet and that delivers content (especially media content) to clients. Content delivery networks are difficult to control and manage as requests for content come from different locations sometimes following certain viewing patterns. In order to optimize performance, locations of the content servers and media content close to the client may be chosen. Many parameters have to be taken into account to optimize the media delivery in content delivery networks, for example:

- Number of content servers
- Locations of the clients and the content servers
- Location of an asset (media content)
- Popularity of an asset
- Available bandwidth in the network
- Cost for moving a content
- Processing capabilities

The number of parameters and unpredictability when it comes to user behavior makes this problem very complex if one wants to control a global CDN using a central management system.

Existing solutions try to solve this problem in a centralized approach. Taking all the parameters into account the different solutions try to best place content in the network. However, the dynamicity of such an environment makes the algorithms highly complex and global decisions take a long time to be made and executed.

Another technical area relevant to the current invention is mobile objects (also called mobile agents). Mobile objects are programs (software) and associated data that can migrate from host to host in a network at times and to places of their own choosing. In the new host they can then continue to run, possibly interacting with the local execution environment. Mobile objects have for example been discussed in the paper ‘Mobile Agents and the Future of the Internet’ by Kotz et al published in ‘ACM Operating Systems Review, August 1999 pp 7-13. An example of using mobile agents is disclosed in U.S. Pat. No. 7,254,608. This patent discloses a system and a method for using mobile agents for managing distribution of content in peer-to-peer networks. In the patent the mobile agent may visit the peer nodes on an itinerary to search for and collect information on distributor content stored on the visited peer.

SUMMARY

The present invention relates to the problem of how to avoid the disadvantages mentioned above of managing a content delivery network.

The problem is in the current invention solved by placing the content itself (such as media files) in the mobile objects (agents). The invention comprises a content delivery system for delivering content to clients where the content itself (such as media files) is placed within the mobile objects (agents). This content delivery system comprises at least one mobile object adapted to be executed on the content delivery system but also adapted to be executed on other content delivery systems or multimedia devices adapted to execute mobile objects. What characterizes these mobile objects is that they comprise:

- a media file accessible by the client,
- a policy data base and execution logic.

The execution logic is adapted to determine the capabilities of the other content delivery systems or multimedia devices.

The execution logic is also adapted to alternatively monitor data related to the client’s access of the media file, and to interrogate the policy data base and to copy the mobile object excluding the media file towards the other content delivery system or multimedia device when certain conditions stored in the policy data base are met, or to receive a request for the media file from the client and to copy the mobile object excluding the media file towards the other content delivery system or multimedia device.

The content delivery system also includes a transcoding unit adapted to transcode the media file prior sending it to other content delivery systems or multimedia devices having limited capabilities.

The invention also includes a method for placing content in the content delivery system or multimedia device.

In the method, the mobile object (comprising the policy data base and the media file accessible by the client as described above) executes in one content delivery system. The mobile object either:

- monitors data related to the client’s access of the media file and when certain conditions stored in the policy data base are met, the mobile object sends a copy of itself excluding the media file to the other content delivery system or multimedia device
- or when receiving a request from the client, a copy of the mobile object excluding the media file is sent to the other content delivery system or multimedia device.

After sending the copy excluding the media file, the mobile object also determines the capabilities of the other content delivery system or multimedia device. If the capabilities in the other content delivery system or multimedia device are restricted, the media file is transcoded before it is sent to the copy of the mobile object in the content delivery system or multimedia device.

The invention further includes a method and a multimedia device for receiving and executing mobile objects comprising media files.

The current invention makes the content (e.g., a media file) autonomous. By autonomous is here meant that the mobile object is intelligent enough to monitor data related to the client’s access of the media file (e.g., number of times the media file has been downloaded, local link utilization measurements, link costs, etc) and to determine capabilities of other content delivery systems or multimedia devices and to take appropriate decisions. Instead of utilizing a central server (or a couple of servers) to gather global information and take global decisions the invention comprises a mobile object that takes local decisions based on local information.

This will highly simplify the way content is placed and controlled through content delivery networks.

The objective with the current invention is therefore to overcome at least one of the disadvantages mentioned above.

The invention has several advantages:

- Content such as media files can be placed in content delivery systems and multimedia devices having limited capabilities.
Efficient utilization of radio link (e.g., GPRS) and network resources (e.g., more powerful network-based execution environments).

Avoid utilizing scarce resources on the device with limited resources.

Allow mobile objects residing in portable devices to be utilized by the content delivery system.

Enhanced end-user experience (faster transcoding time, faster download time).

Robustness; the system is more robust since it features strong fault isolation characteristics. The mobile objects are independent pieces of software relying on the local environment and not on centralized systems. It is harder to launch a denial of service attack towards such a distributed system.

No central point of failure; equal distribution of control and management functionalities brings a nice load distribution and no central point of failure.

Simplicity; the complexity of traditional global optimization algorithms is eliminated by allowing local decisions based on local knowledge to be taken.

Lower maintenance/management costs; simplicity leads to lower cost of maintenance of the entire system.

The invention will now be described in more detail and with preferred embodiments and referring to accompanying drawings.

BRIEF DESCRIPTION OF THE DRAWINGS

FIG. 1 is a block diagram illustrating a content delivery network.

FIG. 2 is a block diagram illustrating an embodiment of a content delivery system and a mobile object.

FIGS. 3a and 3b are flow charts showing a method of placing content in a content delivery network.

FIG. 4a is a block diagram illustrating an embodiment of content delivery systems in a content delivery network.

FIG. 4b is a block diagram illustrating an embodiment of an implementation of the content delivery system.

FIG. 5 is a block diagram illustrating an embodiment of a content delivery system.

FIG. 6 is a block diagram illustrating an embodiment of a content delivery system.

FIG. 7 is a flow chart showing a method of placing content in a content delivery system.

FIG. 8 is a flow chart showing the method of placing content in a multimedia device.

FIG. 9 is a block diagram illustrating different types of mobile objects.

FIG. 10 is a flow chart showing a method of receiving content in a multimedia device.

DETAILED DESCRIPTION

FIG. 1 illustrates an example of a content delivery network CDN 100. The CDN 100 comprises in this example, four content servers 111, 112, 113 and 114. These content servers 111-114 are interconnected by a network comprising a number of routers or switches 121, 122 and 123. In content server 111, a media file 1111 is stored. The media file 1111 can in this example be accessed by seven different clients 131-135, 520, 620. As a majority of the clients are located closer to server 112 than server 111, it is worth to consider moving the content, i.e., the media file 1111 from server 111 to server 112. Closer can here be a network based distance, e.g., number of node hops in between the server 111, 112 and the clients 131 etc. By moving the content closer to the clients 133-135 and 620 transmission resources are saved in the network 100, there will be lower delays and the response time will be faster.

In known prior art, the content servers 111, 112, 113 and 114 are managed by a central operation and management center, OMC 190. As discussed above, this solution has a number of drawbacks.

FIG. 2 describes an embodiment of a content delivery system, here a Content Server 1 210 and a mobile object 212 according to the current invention. Content Server 1 210 comprises an execution environment 211 and at least one mobile object 212. The mobile object 212 comprises a media file 213. This media file 213 could be a data file such as a film, a streaming video, a game, an audio file, a document etc that can be accessed in action 231 by a client 230, 240 for on-line viewing or for download.

The mobile object 212 further comprises execution logic 214. The execution logic comprises program code 215 and a data area 216 that is used to execute the mobile object 212. The execution logic 214 is also handling requests in action 231 from the clients 230, 240 that want to access the media file 213. The programming language for the program code 215 could preferably be Java which has been the most used programming language to implement mobile objects as it is platform independent. The data area 216 comprises also data related to the client’s 230, 240 access of the media file 213 as for example:

- Number of active connections (between the clients 230, 240 and the mobile object 212).
- Number of total accesses to the mobile object 212.
- Number of times the media file 213 has been accessed per time unit (e.g., number of accesses per minute).
- Location of the clients 230, 240 accessing the media file 213 (using for example the source IP address of the clients 230, 240).
- Location of other servers.

In addition to monitor data in the data area 216 in the mobile object 212, the execution logic 214 is also adapted to monitor in action 241 data related the execution environment 211 as for example:

- Available bandwidth and/or processor load.
- Available communication links between the content server 210 and the clients 230, 240.
- Communication link usage and/or link costs.

The mobile object 212 further comprises a policy data base 217. This data base 217 is adapted to comprise any kind of application specific policies that can trigger actions by the mobile object 212 as for example:

- The maximum and minimum acceptable number of accesses to the media file 213 per time unit.
- The maximum and minimum acceptable bandwidth to be consumed by the mobile object 212.
- The maximum network cost of transmission of a media file (213).

Furthermore, utilizing the inventive concept a skilled person can create different policies and corresponding behaviors that are more suitable and tailored to the management of a particular content delivery network.

Using the data 216 related to the access of the media file 213 and the execution environment 211 and interrogating the policy data base 217, the execution logic 214 can for example determine that a majority of the clients 240 accessing the media file 213 are located close to another server, Content Server 2 220.

Fulfilling certain conditions stored in the policy data base 217 the execution logic 214 may trigger a process to move, in
In the process of moving, the mobile object 212 moves from the original Content Server 1 210 to Content Server 2 220 without leaving any traces in Content Server 1 210.

Fulfilling other conditions stored in the policy data base 217 the execution logic 214 may trigger a process to copy the mobile object 212 from Content Server 1 210 to a new mobile object 222 in Content Server 2 220. In this case, the mobile object 212 in Content Server 1 210 continues to execute in parallel with the copy 222.

The copy 222 of mobile object 212 could either be adapted to keep its execution states when starting to execute in Content server 2 220 or it could be adapted to reset the execution states prior to the execution.

In the event of clients not previously connected to find the new location of the mobile object 222, a name resolution process is started using for example DNS (Domain Name Service) redirection.

A situation that also can occur is that very few (or no) clients at all have accessed the media file 213 for a certain period of time. This can be an indication that the media file 213 has become less popular or that the clients 240 are located close to another Content Server 2 220 already hosting a copy of the media file 213. In this situation, the execution logic 214 can take a decision to simply let the mobile object 212 in Content Server 1 210 ‗die‘ and delete itself.

The flow charts in FIGS. 3a and 3b illustrate an embodiment of the method of placing content in a content delivery network 100 according to the current invention. The mobile object 212 is executing in the original server 210 in step 301. During execution it monitors in step 302 data related to the client’s access of the media file 213. This data can be collected by the mobile object itself 212 and stored in the data area 216. The data can also be located in the execution environment 211 and accessed in action 241 by the mobile object 212.

In step 303, a check is made if the media file 213 can be deleted. If a certain period of time has elapsed without any client 230, 240 accessing the media file 213 or that the media file 213 has been accessed very seldom, the mobile object 212 can make the decision to halt the execution and delete itself in step 304. The conditions for this are stored in the policy data base 217. If, on the contrary, the media file 213 is very popular but mainly accessed by clients 240 located closer to the other content server 220, the mobile object 212 makes the decision in step 305 to move the mobile object 212 to the other server 220. But before sending the mobile object to the other content server 220 in step 307, the mobile object 212 halts in step 306 the access to the media file 213 and pauses the active connections between clients 230, 240 and the mobile object 212.

If the decision in step 305 is to move the media file 213, the flow chart continues (digit ‘2’ encircled) in FIG. 3b. If the media file 213 is popular in both regions where clients 230 (close to Content Server 1 210) and clients 240 (close to Content Server 2 220) are located, the mobile object 212 can make the decision in step 308 (now turning to FIG. 3b) to copy itself in step 309 instead of moving. Before the copy 222 of the mobile object 212 is sent to Content Server 2 220, a check is made in step 310 if the current execution states stored in data area 216 in the copy 222 is to be kept or reset.

If the states are to be kept, the copy 222 (a ‘clone‘) of the mobile object 212 is sent in step 312 to content server 2 220. If the states are to be reset, the states are reset in step 311 and the copy 222 (a new ‘instance‘) is sent in step 312. After sending the copy 222 in step 312 the process starts all over again (digit ‘1‘ encircled) as illustrated by FIG. 3a. This is also the case if the mobile object 212 decides to not copy itself in step 308.

Yet another aspect of the invention is that the content delivery system 220 can be seen as a content delivery end-point (source of content) that is created dynamically when the mobile object 212 is moved or copied into the execution environment 221. As an example, a server that is originally not a content server but having an execution environment 221 adapted for mobile objects in general can become a Content Server 2 220 when the mobile object 212 according to the invention is copied or moved into the execution environment 221 in the server 220.

The current invention is in the embodiments described above applied to a mobile object 212 that is moved and/or copied from one content server 210 to another content server 220. The inventive concept does also allow for the mobile object 212 to migrate between other content delivery systems or to other systems or devices to execute mobile objects 212.

One example of this is a peer-to-peer P2P network. FIG. 4a illustrates a P2P network 400 comprising a plurality of terminals 401, 402, 403, 404. In this example all terminals 401-404 are connected to each other fully meshed. The content delivery system 210 (corresponding to any of the terminals 401-404. Assuming that a mobile object 405 is located in terminal 401, the mobile object 405 may decide to move itself in action 407 to another terminal 404.

Another example are portable devices such as PDAs, smart phones, game consoles and so on. The portable devices can themselves be content delivery systems and part of the content delivery network 100, but can also be a portable device only adapted to receive and execute mobile objects. In principle, the portable devices can in addition comprise a client that itself can request content from the content delivery network 100.

FIG. 5 illustrates a first content delivery system 510 and a second content delivery system 520 having limited capabilities, such as a portable device. The first content delivery system 510 is here a server but could in principle also be a portable device.

The first content delivery system 510 comprises an execution environment 511, at least one mobile object 512 adapted to run in the execution environment 511. The mobile object 512 in turn comprises execution logic 514, a policy data base 517 and a media file 513. The execution logic 514 is also adapted to monitor in action 5143 data related to the execution environment 511. The first content delivery system 510 further comprises a transcoder 519 adapted to transcoding the media files 513.

The term transcoding does here include various types of transformation of the media file 513. If for example the media file 513 is a video file, transcoding can mean digitally compressing, changing sample rate, converting between different coding formats etc.

The second content delivery system 520 comprises an execution environment 521 adapted for mobile objects. It
further comprises an execution interface 540 to the execution logic 521 that is adapted so that mobile objects 522 can access information about the capabilities of the second content delivery system 520. In addition, the second content delivery system 520 optionally comprises a client C 528. FIG. 6 illustrates a first content delivery system 510 as in FIG. 5, but the second content delivery system 520 is replaced by a multimedia device, here another portable device 620. One difference between the second content delivery system 520 and the portable device 620 is that the second content delivery system 520 is part of the content delivery network 100, whereas the portable device 620 is not. The portable device 620 comprises however an execution environment 621 adapted for executing mobile objects and an execution interface 640. In addition it comprises a client M 628. The client M 628 in the portable device 620 is further adapted to send requests 630 to the first content delivery system 510.

The client X 660 is a traditional client that can access the media file 513 but is neither part of the content delivery network 100 nor having capabilities of executing mobile objects.

When sending the mobile object 512 from the first content delivery system 510 to the second content delivery system 520 or the multimedia device 620, a number of considerations have to be made with regard to the capabilities of this system 520 and this device 620. The capabilities can be limited by the system/device 520.620 itself such as screen resolution, processing power, memory, supporting decoding types, battery life-time etc. Other capabilities can be network related such as link capacity and link costs (especially for wireless devices). If for example the media file 513 is a high definition video file, this video file may be unsuitable for being presented on a small wireless smart phone screen. Alternatively, the link capacity between the first content delivery system 510 and the system/device 520.620 can be limited (such as low bandwidth) which makes the distribution too slow. To overcome this, the invention includes the feature to transcode in action 707 the media file 513 to a format more suitable for the system/device 520.620 prior sending it in action 708 to the system/device 520.620.

The method to place content in a content delivery system 520 having limited capabilities is illustrated more in detail in by the flow chart in FIG. 7. The original mobile object 512 (here called a parent mobile object) executes in step 701 in the content delivery system 510. During execution, the mobile object 512 monitors in step 702 data related to the client’s access to the media file 513. In step 703 a check is made if the mobile object 512 needs to be copied. If not, the mobile object 512 continues to execute in step 701. If, yes, the mobile object 512 creates and sends in step 704 a copy 522 of itself (here called a child mobile object), excluding the media file 513 to the content delivery system 520.

When the child mobile object 522 is received by the content delivery system 520, it starts to collect information about the capabilities in the system 520. This can be done by simple system calls within the system 520 and can also include network probing. The probing can be performed as measurements between the child mobile object 522 and the parent mobile object 512 in the content delivery system 510 to detect current link conditions such as bandwidth etc.

When the information of the capabilities has been collected by the child mobile object 522 in the content delivery system 520, it returns this information in step 705 in a message to the parent mobile object 512 in the content delivery system 510. The parent mobile object 512 determines in step 706 if transcoding is necessary. If not, the media file 513 is sent unchanged in step 707 to the child mobile object 522. This can for example be the case when the content delivery system 520 itself can perform the transcoding. If transcoding is needed in step 706, the media file 513 is transcoded in step 707 by the transcoder 519 and sent in step 708 to the child mobile object 522. The execution of the parent mobile object 512 is continued as before in step 701.

In addition to transcoding the media file 513 in step 707, optionally a reference between the parent mobile object 512 and the child mobile object 522 is created and stored in the parent mobile object 512. The use of this reference will be discussed more in detail further below.

The method to place content in the multimedia device 620 is illustrated more in detail by the flow chart in FIG. 8. The parent mobile object 512 executes in step 801 in the content delivery system 510. At some point in time, the mobile object 512 receives in step 802 a request 630 from the client M 628 in the portable device 620 for the media file 513. The parent mobile object 512 can interrogate its policy data base 517 and in step 803 decide to stream the media file 513 in step 809 or to send in step 804 a copy 622 of itself (a child mobile object), to the multimedia device 620, but where the media file 513 is excluded.

The knowledge that the client M 628 is residing in a portable device 620 can be determined, for example, by analyzing the HTTP request and identifying the specific portable device’s browser. The ability to execute mobile objects can for example be determined by receiving an explicit indication in the request 630.

When the information of the capabilities of the portable device 620 has been collected by the child mobile object 622, it returns this information in step 805 in a message to the parent mobile object 512.

The parent mobile object 512 determines in step 806 if transcoding is necessary. If not, the media file 513 is sent unchanged in step 808 to the child mobile object 622. This can for example be the case when the portable device 620 itself can perform the transcoding. If transcoding is needed in step 806, the media file 513 is transcoded in step 807 by the transcoder 519 to a transcoded media file 523 and sent in step 808 to the child mobile object 622 in the portable device 620. The execution of the parent mobile object 512 is continued as before in step 801.

As a further option, a user of the portable device 620 can specify some parameters related to the capabilities of the device 620 to be used for playback of the content. These parameters can be accessed by the child mobile object 622. The parameters can for example comprise an override parameter that says that irrespectively of the capabilities of the portable device 620, the media file 513 should be delivered unchanged. This option could be used when the user desires to receive the full media file 513 but intends to transfer it locally to another device having a more suitable display such as a desktop computer or a HDTV set. By returning the overridden parameter in step 805, the parent mobile object 512 considers this in step 806 and sends in step 808 the media file 513 unchanged to the child mobile object 622.

The method to receive content in the portable device 620 seen from the portable device’s 620 point of view is illustrated by the flow chart in FIG. 10. The event that triggers the method in step 1000 is normally a user of the portable device 620 who wants to access some sort of multimedia, such as a video file. The portable device 620 sends a request to the content delivery system 510 in step 1001. Instead of receiving a streamed video file, the portable device 620 receives the child mobile object 622, excluding the media file 513.

When receiving the child mobile object 622, the portable device 620 starts in step 1003 to collect information about the
The content delivery system 210, 510, 520 also includes a transcoder unit 440. This transcoder unit 440 is used for transcoding the media file 513 when necessary as described above.

The invention does also include a multimedia device 620 for playing media files 523 received from the content delivery system 510. This device is illustrated in FIG. 6.

The multimedia device 620 comprises a client 628. Triggered by a user event in step 1000 (see flow chart in FIG. 10) this client 628 sends in step 1001 a request 630 for the media file 513 located in the mobile object 512 in the content delivery system 510. In order to receive and execute mobile objects 622 in step 1002, the multimedia device 620 is provided with an execution environment 621 for mobile objects 622.

Certain requested media files 513 may not be suitable to be played on the multimedia device 620 due to limited screen resolution, processing power, memory, supporting decoding types, battery life-time etc. In order to overcome this, the execution environment 621 in the multimedia device 620 is provided with an execution interface (640) accessible by the mobile object (622). Through this execution interface (640) the mobile object (622) can access information about the capabilities of the multimedia device 620. This information is sent to the mobile object 512 in the content delivery system 510. This mobile object 512 will then, if necessary, initiate a transcoding of the media file 513 to a transcoded media file 623. The transcoded media file 623 (or the original media file 513) is then sent to and received by the multimedia device 620.

An alternative for the parent mobile object 910 to transcode the media file 914 before sending it to the third mobile object 940 is to refer to an already existing transcoded media file 924, 934 in an already existing child mobile object 920, 930. This is done by storing references 921, 931 to the child objects 920, 930 in the parent mobile object 910. In the case with the third child mobile object 940, the parent mobile object 910 can for example decide to send in action 941 the reference 931 to the child mobile object 930 instead of sending a transcoded media file. The procedure to do this is similar to the procedure illustrated by the flow chart in FIG. 8, but with the difference that the parent mobile object 910 in step 808 sends the reference 931 to the child mobile object 940 instead of transcoding the original media file 914 in step 807 and sending the transcoded media file in step 808.

It is then up to the child mobile object 940 to contact the other child mobile object 930 in action 950 and request a transfer of a copy of the transcoded media file 934. This latter scenario is suitable when the two children mobile objects 940, 930 possess enough resources such as bandwidth to perform this transfer.

The parent mobile object 910 is adapted to keep the references 921, 931 alive only if certain conditions are met. If the parent mobile object 910 judges that the available bandwidth or the processing power for the systems or devices carrying the child objects 920, 930 is too restricted, it may erase the references 921, 931. The parent mobile object 910 can also erase the references 921, 931 when no request relating to the existing transcoded media files 924, 934 has been received for a certain period of time.

FIG. 6 illustrates an example of an implementation of the content delivery system 210, 510, 520. The system 210, 510, 520 comprises a processor 410 designed to execute software in a memory area 420. This memory area 420 comprises an execution environment 430 and at least one mobile object MO 421, 422, 423. Each of the mobile object 421, 422, 423 is adapted to communicate with the execution environment 430.
3. A content delivery system for delivering content to at least one client, said content delivery system comprising:
   a processor,
   a memory area, and
   at least one mobile object adapted to be stored in the memory area and to be executed in the content delivery system and in at least one multimedia device adapted to execute mobile objects and comprising the client and where the mobile object comprises the following parts:
   a first media file accessible by the client;
   a policy data base;
   execution logic comprising program code and a data area and where the data area comprises client access data related to the client's access of the first media file and where the execution logic is adapted to receive a request for the first media file from the client, to determine capabilities of the multimedia device, to copy the mobile object excluding the first media file to the multimedia device when certain conditions stored in the policy data base are met by the client access data and create a reference to the copied mobile object in the mobile object; and
   where the content delivery system further comprises a transcoding unit adapted to transcode the first media file into a second media file prior to the sending of the second media file to the copied mobile object in the multimedia device.

4. The content delivery system as in claim 3 further comprising an execution environment adapted for mobile objects and where the execution logic in the mobile object is further adapted to monitor data related to the execution environment.

5. A method for placing content in a content delivery network comprising the steps of:
   executing in a first content delivery system a first mobile object comprising execution logic, a policy data base and a first media file accessible by at least one client and where the execution logic comprises client access data related to the clients' access of the first media file;
   monitoring the client access data;
   when certain conditions stored in the policy data base are met by the client access data, sending a copy of and created by the first mobile object excluding the first media file to a second content delivery system;
   determining capabilities of the second content delivery system for receiving the first media file;
   creating a reference to the copied first mobile object in the first mobile object;
   transcoding the first media file into a second media file if the second content delivery system is not capable of receiving the first media file; and
   sending the second media file to the copied first mobile object in the second content delivery system.

6. The method as in claim 5 where the step of determining the capabilities of the second content delivery system comprises the step of receiving from the copied first mobile object a message comprising data about the capabilities of said second content delivery system.

7. The method as in claim 6 further comprising the step of erasing the reference after a certain period of time.

8. The method as in claim 5, where the client is located in the second content delivery system.

9. A method for placing content in a multimedia device comprising the steps of:
   executing in a content delivery system a first mobile object comprising execution logic, a policy data base and a first media file accessible by a client in the multimedia device and where the execution logic comprises client access data related to the client's access of the first media file;
   receiving a request from the client to access the first media file;
   when certain conditions stored in the policy data base are met by the client access data, sending a copy of and created by the first mobile object excluding the first media file to the multimedia device;
   determining capabilities in the multimedia device for receiving the first media file;
   creating a reference to the copied first mobile object in the first mobile object;
   transcoding the first media file into a second media file if the multimedia device is not capable of receiving the first media file; and
   sending the second media file to the copied first mobile object in the multimedia device.

10. The method as in claim 9 where the step of determining the capabilities of the multimedia device comprises the step of receiving from the copied first mobile object a message comprising data about the capabilities of said multimedia device.

11. The method as in claim 9 further comprising the step of erasing the reference after a certain period of time.

* * * * *