adlet template is generated for the requesting device. The logical flow then ends.

FIG. **12** illustrates a process for retrieving additional attributes from the requesting device, according to one embodiment of the invention. After a start block, the logical flow moves to block **1210** where the language used by the requesting device is determined. The determined language is used to provide the requesting device with content matching their language preference. For example, if the determined language is Spanish, then the content will be delivered to the requesting device in Spanish. Similarly, if the determined language is English, then the content is delivered in English. Flowing to block **1220**, the network bandwidth is determined. As will be appreciated, the network bandwidth may be determined many different ways. According to one embodiment of the invention, the bandwidth is determined automatically by performing a download test on the requesting device. The requesting device may download a very small file having a predetermined size for a given period of time. The bandwidth is determined based on the number of files successfully downloaded for the given period of time. As the network conditions may change over time, the download test may be repeated. Moving to block **1230**, a determination is made as to whether a firewall is present on the requesting device. Transitioning to block **1240** permissions on the requesting device are determined. For example, permissions may be set that disable the ability to perform certain operations across a network. Moving to block **1250**, other attributes are determined. Another attribute may be a protocol restrictions attribute or the other attributes may include any attributes that help to optimize content delivery to the requesting device.

FIG. **13** illustrates the process of requesting an adlet playlist, according to one embodiment of the invention. After a start block, the logical flow moves to block **1310** where the device requests the adlet playlist from a CDN on the network. Moving to decision block **1320**, a determination is made as to whether the adlet playlist is cached on the specified CDN. When the adlet playlist is not cached, a default adlet playlist is delivered to the requesting device (block **1330**). According to one embodiment of the invention, the default adlet playlist is the closest playlist stored on the CDN that is optimized for the requesting device. The process transitions to block **1340**, where the adlet playlist is created that is optimized for the requesting device (See FIG. **15** and related discussion). The created adlet playlist is then cached for future requesting devices on the CDN (block **1340**). When the adlet playlist is cached, the logical flow moves to block **1350** at which point the CDN sends the requested content that is optimized to play on the requesting device. The logical flow then ends.

FIG. **14** shows a process for monitoring the performance of an adlet on a requesting device, according to one embodiment of the invention. After a start block, the logic flows to block **1410** where the adlet playlist is monitored. Moving to decision block **1420**, a decision is made as to whether each cliplet within the adlet playlist is playing properly. To determine if the cliplet is playing properly the media player within the requesting device may be polled at predetermined times. For example, the media player may be polled after ten seconds to determine if the cliplet is playing properly. When the cliplet is playing properly, the logical flow returns to block **1410** to continue monitoring of the adlet playlist. When the cliplet is not playing properly the cliplet may be retrieved from an alternative location. The alternative location may be another location on the network. The cliplet is then delivered to the requesting device (block **1430**) and the monitoring is continued. The logical flow then ends.

FIG. **15** illustrates a process for generating instructions for delivering the on-demand content, according to one embodiment of the invention. After a start block, the logical flow moves to block **1510** where the cliplets within the adlet are determined. The cliplets may be identified as described above with reference to FIGS. **7**, **8**A, and **8**B.

Moving to block **1520**, the source and protocol for delivery of the cliplets is determined. A determination is made as to whether the cliplet should be downloaded in full, downloaded in chunks of data, or streamed to the requesting device. The delivery method is determined from the attributes of the requesting device as well as the size of the content. For example, if the cliplet may be downloaded within a predetermined time, then the content may be downloaded in full. For example, if the cliplet may be downloaded within ten seconds then the cliplet may be downloaded in full. As will be appreciated, the time to download the cliplet depends on the bandwidth of the requesting device, latency within the network, and the file size of the cliplet. Chunking may be used when a device may receive larger amounts of data than streaming allows. According to one embodiment of the invention, the cliplets are streamed for low bandwidth requesting devices. Generally, streaming is a technique for transferring content so that it can be processed as a steady and continuous stream by the requesting device. Streaming technologies are becoming increasingly important with the growth of the commonly used because most users do not have fast enough access to download large files quickly. Using streaming, the requesting device can start displaying the data before the entire file has been transmitted. For streaming to produce a smooth presentation to the user, the requesting device must be able to collect the data and process it as a steady stream.

Flowing to block **1530**, a message may be sent to the JIT encoder to encode the cliplet according to the determined attributes. The JIT encoder may encode the cliplets in real-time and deliver them to the CDN for delivery.

Next, at block **1540**, the media entity body is generated to perform the cliplet on the requesting device. According to one embodiment of the invention, the media entity body is created using Synchronized Multimedia Integration Language (SMIL). According to another embodiment, the media entity body is created using ASX. Briefly described, SMIL is a markup language based on extensible Markup Language (XML) and is being developed by the World Wide Web Consortium (W3C) that enables developers to divide multimedia content into separate files and streams (audio, video, text, and images), send them to a user's computer individually, and then have them displayed together as if they were a single multimedia stream. The ability to separate out the static text and images helps to make the multimedia content much smaller so that it doesn't take as long to travel over the network. Rather than defining the actual formats used to represent multimedia data, it defines the commands that specify whether the various multimedia components should be played together or in sequence.

ASX files, on the other hand, are small text files that can always sit on an HTTP server. When the browser interprets the ASX file, it accesses the streaming media file that is specified inside the ASX file, from the proper HTTP, mms, or file server. For example, the following is an exemplary ASX file:

```
<ASX VERSION="3">
  <ENTRY>
    <REF HREF="mediaplayer://'location to access file"/XXXX1.asf/">
    <REF HREF="mediaplayer://"location to access file"/XXXX2.asf/">
  </ENTRY>
</ASX>
```