UMA-based Wireless and Mobile Video Delivery Architecture

Latha Sampath[*], Abdelsalam Helal[†], John R. Smith[‡]

1. Introduction

As pervasive computing is becoming a reality that has already begun to shape our lives, the need for multimedia delivery systems in wireless and mobile environments is direr. The richness of multimedia data, the limitations of the wireless networks, and the restrictions imposed by the portable device modalities give rise to a complex problem that is in need of immediate solutions.

The Moving Picture's Experts Group (MPEG) has started work on a new standardization effort called the "Multimedia Content Description Interface," also known as MPEG-7. The goal of MPEG-7 is to enable fast and efficient searching and filtering of audio-visual material. The effort is being driven by specific requirements gleaned from a large number of applications related to image, video and audio databases, media filtering and interactive media services (radio, TV programs), scientific image libraries, and so forth.

Recently, multimedia content description information for enabling Universal Multimedia Access (UMA) has been proposed [1], as part of the MPEG-7 specifications. The basic idea of universal multimedia access (UMA) is to enable client devices with limited communication, processing, storage and display capabilities to access rich multimedia content [5]. The use of MPEG-7 with UMA descriptors would be ideal for our efforts towards fast and efficient multimedia data transmission.

In this paper we propose a conceptual model for describing the network conditions under which the multimedia data is being transmitted, as well as the modality of the device which receives the data. We present an architecture for adaptation of multimedia data to such wireless network conditions and device capabilities, under constraints imposed by user preferences and multimedia content, to ensure effective, meaningful, and acceptable delivery of video data to mobile users. The adaptability is achieved through careful application of a combination of off-line and on-line reductions to the video streams. In doing so, we make use of the concept of descriptor schemes for describing the content of video data, and its lendability to different kinds of reductions [1].

The architecture consists of an MPEG-7 server in the fixed network, and an MPEG-7 player on the mobile host. In addition, an encoder/decoder layer is used to perform physical frame bit reductions on transcoded video frames. The server also maintains index structures and a programmable two-dimensional matrix of reductions. An offline packager module is included with the server, which packages all the MPEG-7 data with the video when the video is first registered with the database.

2. Content Description for Universal Multimedia Access (UMA)

The main idea of UMA (Universal Multimedia Access) is that any kind of device with any minimal capabilities should be able to access any multimedia data, over any network. In order to be able to do this, the basic solution is to modify the multimedia content to suit the conditions of transfer.

Different variations of multimedia data have been classified in [1], and ways of describing the content of such data have been studied, to determine their adaptability to reductions. The variations can be derived from other multimedia data, such as through methods for extraction, summarization or translation, or can simply represent substitutions of multimedia data [1].

For example, given a video program with large resource requirements, the program may not be easily handled by small hand-held computer (HHC) devices. In this case, a content-publisher may create a substitute program with lower resource requirements for users of these devices. Alternatively, variations of the program may be generated by translation, summarization and extraction methods, as explained in [1]. For example, the program may be translated from video to the audio modality to be delivered to auto-PC devices. Alternatively, variations may be generated by extraction methods, such as through key-frame

[*] Computer and Information Science and Engineering Dept. University of Florida, lsampath@cise.ufl.edu
[†] Computer and Information Science and Engineering Dept. University of Florida, helal@cise.ufl.edu
[‡] IBM T.J. Watson Research Center, jsmith@us.ibm.com

extraction [10] in order to produce an animated view of the program, which consists only of important key-frames.

2.1Variations-DS – classification of reduction schemes

Several entities are defined by [1] for describing variations of multimedia data: substitution, translation, summarization [1], and extraction. The variation entity is defined by [1] as an abstract entity from which the other different types of entities (substitution, translation, summarization, extraction and visualization) are derived. The variation entity also contains attributes that give the selection conditions for which a variation should be selected as a replacement for a multimedia item [1]. The derived entities are as in Table 1.

In UMA applications, the variations of multimedia material can be selected as replacement, if necessary, to adapt to client terminal capabilities or network conditions. If the network conditions are bad, a variation with the least possible effort needed for transfer may be selected, so that the work is reduced. At the same time, if the device used by the client has excellent display characteristics and other good resources needed for display of multimedia, the variation selected must as far as possible have good visible characteristics – for example, key frame extraction can be selected instead of color reduction. On the other hand, if the device does not have some necessary quality, such as a color display, then that quality can be dropped from the multimedia that needs to be streamed to that client.

In general, the Variation-DS provides important information not only for UMA applications but also for managing multimedia data archives since in many cases the multimedia items are variations of others.

2.2 UMA Attributes used to describe multimedia content

The MPEG-7 specifications document proposed several descriptive attributes for multimedia data in general, which were used to reduce the capacity requirements of large MPEG multimedia videos. MPEG-7 is targeted at the problem of browsing a video file to detect frames of interest to the user. These attributes, called Descriptors, were used to describe various types of multimedia information, in the form of video, audio, graphics, images, 3D models and so on.

Reduction	Description	Examples
Substitution	One program substitutes for another when there need not be any actual derivation relationship between the two	Text passages used to substitute for images that he browser is not capable of handling
Translation	Conversion from one modality to another – the input program generates the output program by means of translation	text-to-speech (TTS), speech-to-text (speech recognition), and video-to-image (video mosaicing)
Summarization	Input program is summarized to generate the output program – may involve compaction and possible loss of data	Thumbnail generation, text summarization
Extraction	Information is extracted from the input program to generate the output program – involving analysis of the input program	key-frame extraction from video, embedded-text and caption extraction from images and video

Table 1. Classification of reductions performed on multimedia data.

When UMA was proposed as a new part of the MPEG-7 requirements, several new attributes were proposed as additions too enable UMA [1]. With these additions, it becomes simpler to select any of the different possible kinds of variations, suitable for the nature of multimedia content in any particular video. For example, we may select speech to text translation for a foreign language film. Thus we are able to take into consideration the lendability of the multimedia data to particular reductions.

These descriptive attributes, or Descriptors, are summarized in Table 2. Using these descriptors, we shall see, in later sections, how multimedia content descriptions are obtained, and used to select the ideal variation for the situation for the multimedia data, to be sent to the client.

Descriptor Scheme	Purpose
Published Multimedia Data Hint-DS	Describes the need for and the relative importance of a multimedia data item in a presentation, and may contain hints as to the type of reduction the presentation lends itself easily to
Media-D	Standardized descriptive information about the image, video and audio material, which help in UMA, such as resource requirements, functions of network conditions to preferred scaling operations etc.
Meta-D	Data about data – rights, ownership etc.
Spatio-temporal Domain-D	Information about the source, acquisition and use of visual data
Image Type-D	Describes display characteristics of images
Region Hint-D	Information about the importance of particular regions within an image, relative to each other
Audio Domain-D	Information about the source and usage of audio data
Segment Hint-D	Analogous to the region hint, but applicable to the temporal dimension

Table 2. UMA attributes.

3. Conceptual View

In order to match the terminal device capabilities and connection quality to the different variations, we propose a three dimensional structure – the three dimensions being device modality, connection quality and variation type.

Depending on the quality of the specific connection, and the capabilities of the specific device, the mapping into the three-dimensional space is decided. A number of specific descriptors describe the possible combinations of media resources available for different devices and at different connection qualities. The different variations to the multimedia data can be specified in the form of indices into the multimedia material. We also propose meta-data that influences the content adaptation process, by describing the mapping from connection quality and device modality into required variations. Additionally, we propose a number of descriptors that can be considered to be transcoding hints; that is, they can be used to guide the adaptation process. We discuss some issues that arise in the implementation of such a scheme.

The relationship between the different entities guiding the three-dimensional mapping is illustrated in Figure 1 below. The following subsections examine these factors in depth.

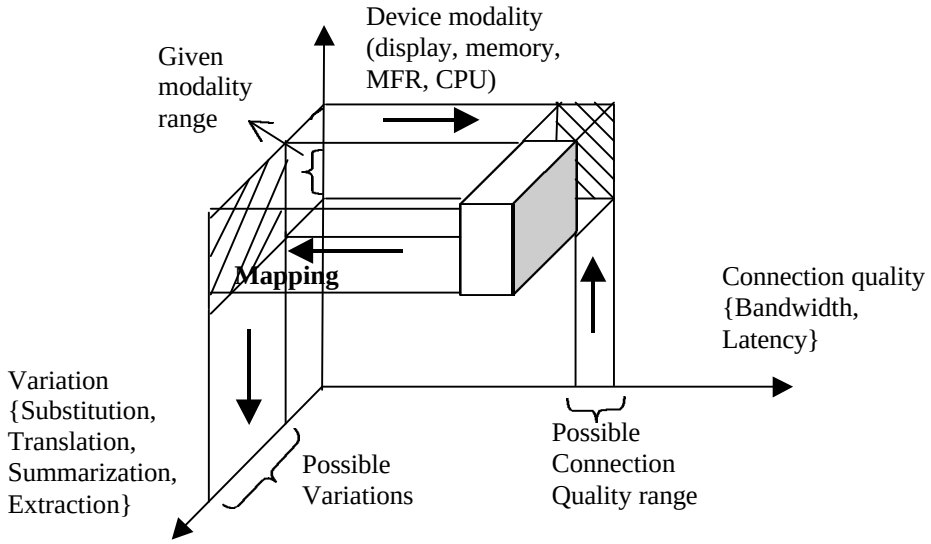

Figure 1. Three-dimensional view of MPEG-7 and the description schema for UMA.

3.1 Network conditions

Different variations in connection quality that may be possible in the network can be described in terms of connection attributes. The two variants in connection quality of a network are Bandwidth and Latency. The quality of connection ranges from the strong connection quality of a fixed network, to the weak connection quality of a wireless network, to loss of connection, which is quite possible in mobile networks. You have three possible Connections attributes:

- Bandwidth – This describes the existing bandwidth of the network. Depending on the value of this attribute, the connection quality of the network varies and the variation to be applied to the multimedia data changes statically as well as dynamically. This will be measured in the usual unit of bps. This can have a range *from 0.0 bps to 50 Mbps*.
- Latency – This factor describes the connectivity of the network in terms of initial connecting time and time of connection maintenance. This can be measured as *with* or *without latency*. This affects the variation to be applied to the multimedia data. The importance of latency can be seen when a large amount of video data needs to be streamed over a network, and due to a latency problem that cannot be ignored, the quality of the streaming is poor. Of course we have to consider that when chunking is allowed in a packet-switched network (as in iDEN), the problem can be ignored.
- BER (Bit Error Rate) **–** This factor describes the accuracy of the network. Accuracy is an important factor to be considered in wireless networks, especially in multimedia applications, since errors in data are so visible to the user.

NETWORK CLASS	BANDWIDTH AND LATENCY SPECS.	EXAMPLE NETWORK
CLASS A1	2.0 Kbps – 9.6 Kbps, with latency	Mobitex (RAM), early CDMA PCS systems
…	…	…
…	…	…
CLASS F2	> 10 Mbps, <= 50 Mbps, without latency	W-LAN, HyperLAN

Table 3. Classification of devices based on connection quality.

Depending on the values of these factors, the various network connections can be classified into certain classes of connections, as illustrated in Table 3. The class of network connection is one of the major factors in the function, that maps the existing conditions to the appropriate variation suggested, for the multimedia data that needs to be transferred over that network.

3.2 Device modality

The capabilities of the device that is currently attempting to view the multimedia data can be described in terms of the variants Display, Audio, Memory, Maximum Frame Rate (MFR) and CPU. The capabilities of a device range over all possible combinations of all possible values of these attributes.

- Display – The display type (resolution, size, refresh rate and color capabilities) of the device. This can range from a handheld device's display to that for a desktop. This will have a domain ranging *from B/W to 64K color or more*.
- Audio – The audio capabilities of the device. The benchmark for this could be measured in a sound present or absent flag**.** We could also have different quality audio, such as CD quality, stereo quality and so on. The audio would be useful in determining whether audio information needs to be downloaded to the device at all or not. Hence the domain of this descriptor would be *{absent, CD quality, stereo quality}*.
- Memory – The amount of storage space available to store buffered multimedia data and run the application. The memory used by a mobile device can generally be divided into slow access memory, under which we have RAM and flash memory, and fast access memory, under which we have

microdrives and hard drives. With more flash memory, we can buffer more amount of information for the streaming, and with the presence of slow access memory (which all mobile devices don't have), we can store even more information. This descriptor will have a domain of *less than 32 MB to more than 1 GB.*

- MFR – The maximum permissible rate of display of frames on the device. The device capabilities can limit the rate at which the screen can be regenerated each time a new frame needs to be displayed. This may be related to the refresh rate of the device screen, which is dependent upon the type of screen material.
- CPU – The CPU capabilities of the device. This can be measured in speed of the CPU, or computational capability – maybe we can use a fixed unit such as flops. This mainly affects the complexity of computations that is permitted on the device, and hence, the extent of decoding possible on the device.

Depending on the values of these attributes, that describe the capabilities of wireless devices, the various devices can be classified into certain classes, as described in Table 4.

CLASS OF DEVICE	DISPLAY	AUDIO	MEMORY	CPU	COLOR	E.G. OF DEVICE
CLASS I a	Low resolution (64X64 – 256X128)	Not present	Low (upto 32 MB)		Low (B/W, grayscale)	Web phones (e.g. Nokia 7110)
…	…	…	…	…	…	…
…	…	…	…	…	…	…
CLASS IV b	High (1024X640 and above)	Stereo	High (1 GB and above)	Good (above 500 MHz)	64 K	

Table 4. Classification of devices based on device modality.

What we have done so far covers the restrictions imposed by the physical conditions in which the multimedia data is being asked for – network conditions and device limitations. The "ideal" variation, however, depends not only on what we can get, but also on the preferences of the user, and the adaptability of the multimedia data to the reductions selected. These factors are discussed below.

3.3 Multimedia Content

The next matter to be considered is the importance that has to be given to the content of multimedia. This is what can be called the "lendability" of the multimedia data to certain types of reductions. Multimedia data is rich in content, and different types of data have parts with different importance, depending on the context in which it is used. For example, the weather channel may give prime importance to the audio, and the video may be less important. But the sports channel may give higher preference for video. A foreign language film may take off audio and leave captions in. So all that seems to be required is some means of describing the content of the multimedia, and associating this description with the video at the server.

This is where MPEG-7 and the additional descriptive data detailed in Section 2 comes in. Originally MPEG-7 was designed to be a content-descriptive language. This concept of content description is extended to include content description specifically useful for UMA [1], so that it gives us an idea about the variations suggested for any particular multimedia data item, the relative importance of different multimedia components within the data, its resource requirements and any other information needed to describe the content absolutely to the server.

When we have this content description with us, it becomes easier to use this to select the set of variations (reductions) ideal for that particular content. These suggestions are also included as part of the content description as transcoding hints [1]. These descriptors can be used to affect the nature of variations suggested for the multimedia data.

We can express this as a restriction on the usage of reductions by the multimedia server. So on the implementation side, when the server gets a request for a particular video from a client, the server uses the network conditions and device modality of the current situation to decide the full set of variations suitable for that situation. Then the Multimedia Content (MC) comes into the picture to restrict the full set of available variations to a smaller set of reductions that are permitted on that content.

3.4 User preferences

The final issue that has to be considered is the user's opinion of all these changes being made to what had originally been requested. The "ideal" variation may be totally in contrast with what the user considers essential, to make sense out of the data. The user may, for example, find audio data more important as compared to video data, and might not like the idea of sound being cut out, to fit the result within the network conditions. If the ideal variation derived from the existing network conditions is speech-to-text conversion, the user will find the result dissatisfactory.

To take the preferences of the user into consideration, we introduce the concept of loss profiles. The user gives a set of limits, within which the variation of the stream from the desired quality should stay, as much as possible. This would help in making a choice between different types of reduction possibilities available.

We can extend this concept a little more to include the data relevant to that user, in the form of a Loss Profile (LP). This would include data in the form of client preferences, the network conditions that the client normally operates within, and the modality of the device that the client normally works with.

We can move further on the lines we have followed above for multimedia content, and impose another restriction on the final result obtained, depending on the user Loss Profile, to get the final set of restrictions to apply on the multimedia data.

4. Implementation strategy

The Transcoding Function describes the mapping between the different static and dynamic variations available with MPEG-7, and their applicability, i.e., the (Connection, Modality) combination situation in which each is relevant. The mapping may be influenced by loss profiles that prescribe user preferences in variation options, and also by the content of multimedia data. These options are discussed in later sections. There can be two classes of variation application:

- **Static Scaling Profile** - This suggests the relationship between the variations suggested, and the conditions of the network connection quality and the device modality as calculated at the time of connection establishment. The structure of this attribute is suggested as a two dimensional grid, with rows and columns represented by different network connection qualities and device modalities specified, and the actual grid containing the variations to be applied in each case.
- **Dynamic Scaling Profile -** In the case of rapid network condition changes, some scaling operations can be done in the network without negotiation with the server (which may involve long delays). The scaling profile indicates sequence of preferred scaling operations on multimedia data and. the impact on corresponding utility functions. These scaling profiles can be made available to the server along with the other MPEG-7 descriptors packaged with the multimedia data, so that the server can take appropriate decisions based on the conditions.

Basically, the mapping of (Connections, Modality) to Variations can be summarized in Table 5. Now we add in the effect of the nature of multimedia content, and the user loss profiles, in the form of the following equations.

Let C be the range of connection quality, and D, the range of device modality into which the current conditions happen to fall. The resultant region of permitted variations, V, into which the system maps can be represented as a function of C and D.

$$\mathbf{V = C \ X \ D}$$

Let MC be a Multimedia Content and V_{MC} be the possible set of applicable variations for this particular multimedia data, given by

$$\mathbf{V_{MC}} = \cap_{\mathbf{MC}} \mathbf{V}$$

where **V** is the set of variations obtained so far and $\cap_{\mathbf{MC}}$ is variation restriction on **V.**

Let $\mathbf{V_F}$ represent the region of finally permitted variations when the user loss profile, LP, is taken into consideration. Hence $\mathbf{V_F}$ is a restricted version of $\mathbf{V_{MC}}$, as specified by LP. This can be expressed using the following equation, where $\cap_{\mathbf{LP}}$ is the loss profile restriction on $\mathbf{V_{MC}}$.

$$\mathbf{V_F} = \cap_{\mathbf{LP}} \mathbf{V_{MC}}$$

Effectively, when a client tries to get some multimedia data from the server, the server will gauge the connection quality of the network. The server will also try to judge the modality of the client device. This can be achieved by having the user select some class of device from among a menu.

Then the server will use this range of C and D to map into the C X D table and the result will be V, a set of variations, which may be applied to the multimedia data to fit it within the physical conditions.

Then the server will use MC to further restrict the set of possible variations suited for that particular multimedia data. Finally the server will use LP to choose, from among the suggested variations, those that are acceptable to the user. This will yield another set of variations, which may be applied to the multimedia content.

On the implementation side, the user LP can be obtained from the user, by having the user move a set of sliders that indicate the acceptable quality of the multimedia needed [11]. These sliders can represent the values for different relevant multimedia information, which can be selected from the descriptors specified for MPEG-7. For example, we can have sliders for color quality varying from B/W to 64 K color, audio quality varying from text captions to stereo quality audio and so on.

When the user selects the values of these attributes, they form an attribute for user loss profile (LP). This is used to restrict $\mathbf{V_{MC}}$ to $\mathbf{V_F}$.

The user's loss profile can be obtained dynamically at the time of connection, and also stored as part of the user's profile.

We can actually do most of this process statically at the time of registration of the video into the database. As the video is being registered into the database, the set of MPEG-7 descriptors could be created for that video and stored along with the video into the database (It should also be possible to insert a video with pre-defined descriptors into the database).

The various variations of the video are also statically created at the same time and stored along with the video. This raises the issue of the amount of storage space required for each video, along with its different variations. The solution to this would involve some sort of complex indexing scheme that does not store the variations separately, but instead indices into the existing video bitstream, to dynamically assemble the particular bitstream required for any particular variation of the video. These indices can also be created and stored with the video at the time of registration.

The mapping of all possible combinations of (C X D) to the respective set of V can also be created and stored statically at registration.

Finally, since the content of multimedia does not change with the user, the restrictions imposed by MC on V can also be stored statically at registration. So finally, after registration, what we have is a new video with its accompanying descriptors (MC), table of C X D, and the indices for all possible variations with that video.

MODALITY					
CONNE-CTION		**CLASS I a**	**…**	**…**	**CLASS IV b**
	CLASS A1	{substitution}≡ {text for images}	**…**	**…**	{translation} ≡ {video-to-image}
	CLASS A2	{substitution}≡ {text for images}	**…**	**…**	{translation} ≡ {video-to-image}
	CLASS B1	{substitution, translation} ≡ {text for images, video-to-image, voice-to-text, color reduction}	**…**	**…**	{translation, summarizat-ion} ≡ {video-to-image, thumbnail generation }
	…	**…**	**…**	**…**	**…**
	CLASS E2	{summarizati-on} ≡ {voice scaling}	{summar-ization} ≡ {frame-dropping}	{summar-ization} ≡ {voice scaling}	No reduction

Table 5. Mapping of (Connections-DS, Modality-DS) to Variations-DS.

Online, when the client establishes connection with the server, the client's Loss Profile (LP) is downloaded, along with information about the network conditions (C) and the device modality (D) of the client. When the client asks for a particular video, the set of $\mathbf{V_{MC}}$ is determined, and then the user's LP comes into the picture, to determine $\mathbf{V_F}$ for that video. The ideal order of performing the final set of variations is determined dynamically, and then the indices are used to select the portions of the bitstream which, when assembled together and streamed to the client, would construct the perfect video as per all the restrictions and user requirements.

This scheme involves the discussion of a lot of issues. Firstly, we need to remember that the ultimate goal of MPEG-7 is not to impose so many restrictions that the final result that the users get is much smaller than the original need. There has to be a way of ensuring that within the given limitations, the maximal quality video is delivered to the users.

Secondly, the imposing of all these stages in the encoding and delivery of the data makes the scheme increasingly complex. We shall discuss these issues and possible solutions in following sections.

5. Architecture

So now when we try to put all this together into a feasible and practical architecture, we have an MPEG-7 server that contains the database of multimedia data. The server will also contain Table 3 and Table 4. The server will be able to ascertain the class of network and client device, and will accordingly use the matrix in Table 5 to decide the appropriate reductions needed to be performed. Each element of the matrix will index into the video to locate the parts of the video needed for processing and streaming.

We can separate the MPEG-7 packaging functions from the actual downloading function of the server, by having a packager module on the server end, which packages multimedia data into an MPEG-7 scheme at the input end of a multimedia database. Each time a new video is added to the database, a registry process will have to be carried out by this module, which will involve filling in descriptor values for the descriptors described in Section 2. Then the packager will, according to the content of the multimedia data, decide which variations can be applied to the multimedia data, and these will form MC – Multimedia Content (see Section 3), that is packaged along with the multimedia data. Thus the packager does the following main tasks offline, at registration:

- Creation of MPEG-7 (and UMA) descriptor values for the video

- Creation of the ideal matrix for that video in particular, with ideal reductions for each level of device modality and network conditions
- Creation of indices for each element of the matrix. Each index is basically a linked list, with links to each piece of the video that needs to be streamed after optional processing stages
- Creation of the mapping table of **(C X D)** to **V**, including the restrictions imposed by **MC**, so that we get $\mathbf{V_{MC}}$

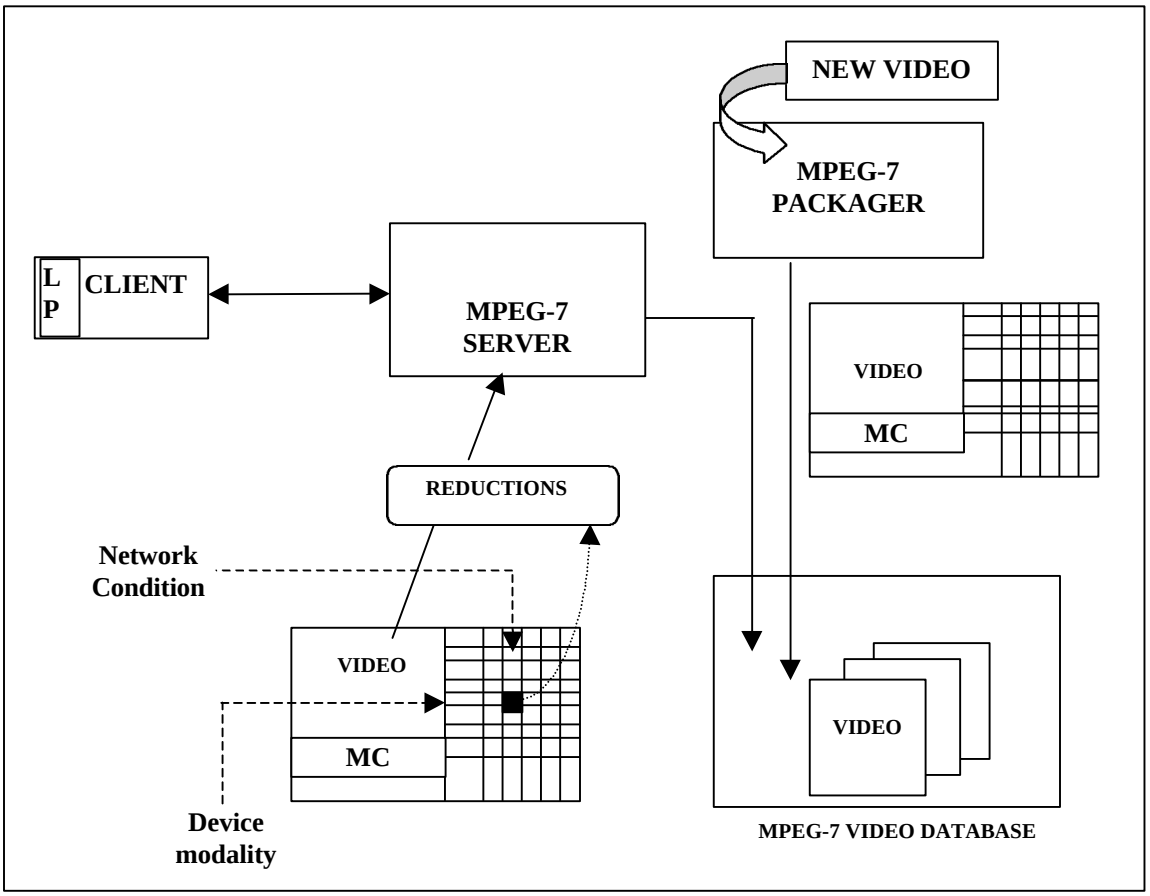

Figure 2. Overview of the process.

When the server gets a request from a client for a particular video, the server checks for the ranges of C (Connection) and D (Device modality), as discussed in Section 3. This gives the server V, the set of variations according to the physical conditions, from the table, and following from that, V_{MC}, depending on MC (Multimedia Content). Then the server uses LP (the user Loss Profile) to restrict the variations provided by V, forming V_F. Finally the server gets an acceptable set of variations. The server then uses algorithms to decide the order in which these variations must be applied to the data, which is used by the server to call the appropriate function that will stream only the required data, after applying the variations, to the client. The entire process is summarized in Figure 2 above. The indexing scheme is used along with this process to assemble portions of the information needed for any particular variation of the video. When the server chooses a particular variation to be applied, it will follow the index and stream the appropriate bits to the client, where these will be reassembled to form the intended reduced video according to the network quality, device modality, content of multimedia requested and the user's preferences.

6. Practical Issues

6.1 Efficiency

One of the main issues that have to be dealt with in the implementation of this architecture is the efficiency of storage. The prominent downside of using several variations, of the same multimedia data,

would be the redundancy in storage of the same data, in different ways, to accommodate all these variations.

One way of reducing, or possibly eliminating redundancy, that was suggested in the previous section, would be to not store different versions of the same data separately, but have a sophisticated indexing structure and scheme. This indexing scheme would have several indices into the base multimedia data file. Depending on which variation was selected as the best one for the situation, the corresponding index into the data would be selected. This would then access, in the appropriate order, only that data relevant to the selected variation.

This indexing can be done by the packager, mentioned in Section 6, when the video is being entered into the database. This means that the packager will have to have a copy of the C, D and C X D tables. The packager can then manufacture a table of its own, consisting of the same rows and columns as the C X D table but with the entries being pointers (indices) into the multimedia data.

Secondly, the imposing of all these stages in the encoding and delivery of the data makes the scheme increasingly complex. This complexity is handled by removing the necessity of maintaining a global table of mappings, from CXD to V, and from V to VU, and refer to this for each access to the database of multimedia data. What we instead do is construct a personal table with each video, suited to its contents, along with other MPEG-7 descriptors, in the registry phase. Then when each user asks to access particular multimedia data, this local table can be referred to, since it would already have taken the content of the multimedia data into account.

Hence there would have to be a C X D table for each video in the database, each with its own set of pointers for reductions. Of course, we can always have the global table for reference purposes.

6.2 Delivery jitter

Another important issue to be considered, when we play around with dynamic variation of data, would be the quality of the data delivered. It may not be acceptable to the users to have their perfect video, with CD quality sound, suddenly change into a text only stream, because of fluctuations in the network.

The solution to this problem would be simply to ask the users what is acceptable to them, and stay within those limits as much as possible. This may be represented in the form of a loss-profile. This would include data from the users as to what is acceptable to them in case of changes in network conditions.

6.3 Synchronization

When we mention jumping from one part of the data to another using indices, we would also have to consider the different streams of data going over the network simultaneously – audio, video, and possibly text captions/ translations. These have to be synchronized throughout the stream. The design may dictate the dropping of audio data if the network deteriorates. If the network comes back to its earlier high quality, we have to restart audio data streaming. However the audio data cannot pick up where it left off – it has to match with the part of video being streamed currently.

6.4 Perceived Video Quality

We may devise any sophisticated scheme to determine the perfect reduction strategy for a video under given conditions of network and device modality. However all this would be of no use if the user is not happy with the results. The acceptability of a video quality is highly subjective and personal. The measurement of acceptability of perceived video quality is a difficult task that involves user participation, and was not performed as part of the tests mentioned in Section 7.

7. Experimental data

7.1 Description

The experimental data was collected in the following manner. A particular video was selected as a case study – this video can be found at [12]. This video was repeatedly downloaded and played under different circumstances, to measure the performance of the scheme. The experiments were divided into two main sections, those with the index and those without. Within each separation, the performance of the server was measured under different schemes of reduction applied to the video.

The reduction schemes were categorized into frame dropping, size reduction and color reduction. In the experiments without indexing, the scheme for application of these reductions was a carefully selected algorithm that yielded the best results for all possibilities of network conditions and device modality. In the experiments with indexing, the same algorithm was used to construct the index mentioned in Section 5, and this index was used later to access and stream the video to the client. The overhead of time and space in the construction of the index was also measured.

To further illustrate the importance of a good indexing scheme, three separate levels of indexing were considered:

- A 1 X 1 index, which just performs the basic streaming, without taking into consideration the network conditions and device modality.
- A 2 X 2 index, which has different combinations of two possible network condition levels and two possible device modality levels. This index combines different types of reduction for better streaming at the lower levels of these conditions.
- A 3 X 3 index, which is similar to the 2 X 2 index, with one more level of device modality as well as network conditions.

7.2 Results

The video selected for the case study was a 4 minute long video clip of 20.361 Mbytes, that contained variations in factors such as extent of movement in the video and importance given to color and clarity of the video.

The results for the lowest level of sophistication in the reduction scheme are shown in Table 6(a) and Table 6(b), one without indexing and one with indexing. For both schemes we used a reduction policy most appropriate for most kinds of videos – 25% size reduction, followed by frame dropping. The results of Table 6(a) were obtained after a straightforward application of the available reductions for the current network and device modality conditions, all being performed at execution time (without the use of the indexing scheme to get pointers to the relevant frames). Table 6(b) also shows the time taken to create the index entries for the video, as well as the space overhead required for storing the index.

Total time taken to download video (sec)
3906

Table 6(a). Performance of streaming without indexing, and one algorithm for all download conditions.

Total download time (sec)	**Overhead in index creation Space/ Time in bytes/sec**
1620	45124 bytes/ 1360 sec

Table 6(b). Performance of streaming with index, and one algorithm for all download conditions.

The results for the next level of sophistication in the reduction scheme are shown in Table 7(a) and Table 7(b), the former without indexing and the latter with indexing. The results show that though the indexing adds overhead in terms of time and space, the improvement in the streaming quality cannot be ignored. The point to remember is that the index generation is done offline and the index is stored on the server, and hence the process is not at the client's cost.

		Device type 1	**Device type 2**
Network class 1	**Download time (sec)**	1023 • Frame dropping • Size reduction	3685
Network class 2	**Download time (sec)**	3876 • Size reduction	3906

Table 7(a). Performance of server without indexing, with 2 possible levels each of network conditions and device modality.

		Device type 1	Device type 2
Network class 1	**Download time (sec)**	934	2986
Network class 2	**Download time (sec)**	1567	3910
Space/ time overhead		49345 bytes/ 2106 sec	

Table 7(b). Performance of server with indexing, with 2 possible levels each of network conditions and device modality.

The results for the highest level of sophistication in the reduction scheme are shown in Table 8(a) and Table 8(b). This clearly demarcates the scheme with a good index, and the scheme without an index. The scheme without the index suffers from the complexity of the algorithm that has to perform reduction at runtime, and does not have the helping hand of reduction hints stored previously in the indexing scheme.

These experiments clearly illustrate the importance of not just performing reductions on the streamed video, but getting aid in performing these reductions with a sophisticated indexing scheme that will allow reduction of the server's access time to the video and processing time of the video.

		Device type 1	Device type 2	Device type 3
Network Class 1	**Download time (sec)**	876 • Frame • Size • Color	1253 • Frame • Size • Color	3218 • Frame • Size
Network class 2	**Download time (sec)**	1106 • Frame • Size • Color	1378 • Frame • Size	3689 • Frame
Network class 3	**Download time (sec)**	1067 • Frame • Size	3889 • Size	3938

Table 8(a). Performance of server without indexing, with 3 possible levels each of network conditions and device modality.

		Device type 1	Device type 2	Device type 3
Network class 1	**Download time (sec)**	556	896	2989
Network class 2	**Download time (sec)**	678	945	3105
Network class 3	**Download time (sec)**	697	801	3916
Space/ time overhead			57089 bytes/ 4576 sec	

Table 8(a). Performance of server with indexing, with 3 possible levels each of network conditions and device modality.

8. Comments and conclusion

Universal Multimedia Access is an important application that will be easily and conveniently enabled, using the principles of MPEG-7. We have shown how the different conditions of transfer of video data can be measured and classified, to form a table of network conditions and device modalities. We have also shown how this can be used to determine an ideal set of reductions for any video. We further described how the nature of each video can help in describing the ideal reductions that video would lend itself to, so that this information can aid in selecting a set of reductions most appropriate for the set of conditions at the

time. We then described a way to make the task of reduction simpler, using an indexing scheme that would index into the video and pick just the information necessary to be streamed to the client given the set of input conditions. We also showed the improved performance on the use of an indexing scheme.

One of the main problems we faced in preparing the tables for this document was the unavailability of much technical data for the device capabilities. The actual data available on the Web is minimal and the characteristics such as processing capabilities and refresh rate, which are important for multimedia data over a wireless network, for devices with limited capabilities, are not available. This data can be added on in future revisions on this document.

We have also not considered all possible reduction techniques that could be used to improve bitrate. An important reduction method we did not include in the tests is rate reduction. The indexing scheme can be easily extended to include this reduction method, expanding the range of possible combinations of download conditions and allowing more fine tuning of the index to different conditions.

9. References

[1] John Smith et al. "MPEG-7 Content Description for Universal Multimedia Access", ISO/IEC Y17CI/SC29/WGII MPEG99IM4949, MPEG-7 Proposal draft.

[2] Generic Audio Visual Description Scheme for MPEG-7 (V0.3), *ISO/IEC JTC1/SC29/WG11 m4677*, Vancouver, July, 1999.

[3] MPEG 7 Applications document, *ISO/IEC JTC1/SC29/WG11/*, MPEG99, Seoul (Korea).

[4] MPEG 7 Requirements document, *ISO/IEC JTC1/SC29/WG11/*, MPEG99, Seoul (Korea).

[5] C. Christopoulos, T. Ebrahimi, V. V. Vinod, J. R. Smith, R. Mohan, and C.-S. Li, "MPEG-7 application: Universal Access Through Content Repurposing and Media Conversion", ", *ISO/IEC JTC1/SC29/WG11 MPEG99/M4433*, March 1999, Seoul.

[6] Composite Capability/ Preference Profiles (CC/ PP): A user side framework for content negotiation. *W3C Note*, http://www.w3.org/TR/NOTE-CCPP/ *(11/1998)*.

[7] J. R. Smith, R. Mohan, and C. -S. Li, "Content-based Transcoding of Images in the Internet", *IEEE Intern. Conf. Image Processing*, Oct. 1998.

[8] J. R. Smith, R. Mohan and C.-S. Li. "Scalable Multimedia Delivery for Pervasive Computing," *ACM Multimedia*, Orlando, FL, November 1999.

[9] R. Mohan, J. R. Smith and C.-S. Li. "Adapting Multimedia Internet Content for Universal Access, *IEEE Transactions on Multimedia*," Vol. 1, No. 1, March 1999.

[10] Y. Abdeljaoued, T. Ebrahimi (EPFL, Switzerland), C. Christopoulos, I. Mas Ivars, "A New Algorithm for Video Summarization," *ISO/IEC JTC1/SC29/WG11 MPEG99/M4738*, July 1999, Vancouver.

[11] Malcolm McIhagga, Ann Light and Ian Wakeman. "Giving Users the Choice Between a Picture and a Thousand Words," *School of Cognitive and Computing Sciences*, University of Sussex, Brighton, May 18 1998.

[12] http://www.harris.cise.ufl.edu/research/wvideo