A QoS-based Service Composition for Content Adaptation[1]

Khalil El-Khatib
University of Ontario Institute of Technology
2000 Simcoe Street North, Oshawa, Ontario, Canada L1H 7K4
Khalil.el-khatib@uoit.ca

Gregor v. Bochmann and Abdulmotaleb El-Saddik
School of Information Technology & Engineering, University of Ottawa
800 King Edwards, Ottawa, On., Canada K1N 6N5
{bochmann, elsaddik}@site.uottawa.ca

Abstract

Today's Internet suffers from the problem of heterogeneity in client devices, network connectivity, content format, and users' preferences. The framework presented in this paper tackles this problem using the approach of service composition to support distributed multimedia applications. The discussed framework for trans-coding multimedia streams uses self-organizing, resilient data distribution algorithms. The framework takes into consideration the profile of communicating devices, network connectivity, exchanged content formats, context description, and available adaptation services to find a chain of adaptation services that could be applied to the content. Part of the framework is a selection algorithm that finds the best sequence of adaptation services that can maximize the user's satisfaction with the delivered content.

1. Introduction

Diversity and heterogeneity of Internet clients is a major problem for multimedia delivery over the Internet. Clients range from a small single-task audio player to a complex, multi-task, multi-function desktop computer. The diversity of clients varies along different axes including display capabilities, storage space, processing power, as well as the forms of network connectivity that these clients use to access the Internet. Clients differ also in the data formats they can consume and produce, installed applications and services, and personal preferences of their users. Today, vast amount of multimedia content already exists on the Internet. Most of this content is created and formatted for the personal computers (PC's), and cannot be rendered directly on all types of client devices. A number of content providers, such as Yahoo and e-Bay, have taken the costly approach of creating different versions of content for different access devices.

Content adaptation is considered an effective and attractive solution to the problem of mismatch in content format, device capability and user's preferences. The process of adaptation, also referred to as trans-coding, is usually applied to the sender's content in order to satisfy the device constrains of the receiver client and the preferences of its user. Possible adaptations include, but are not limited to: text summarization, format change, reduction of image quality, removal of redundant information, audio to text conversion, video to key frame or video to text conversion, content extraction to list a few.

Most currently available content adaptation schemes are best suitable for Web content. Examples of such adaptations schemes include conversion of HTML pages to WML (Wireless Markup Language) pages, conversion of *jpeg* images to black and white *gif* images, conversion of HTML tables to plain text, or stripping of Java applets / JavaScript. These adaptation schemes do not have the same requirements and challenges of real-time multimedia content adaptations. Real-time multimedia applications involve large volumes of data making trans-coding a computationally very expensive task [1,2]. To solve this problem, some trans-coding services have been implemented in hardware and can be deployed on intermediate network nodes or proxies [3]. This approach cannot keep the pace with the constant and quick introduction of new types of clients, and requires investments in specialized hardware devices. Another more suitable approach to address the computational challenge of multimedia trans-coding is based on the observation that the general trans-coding process can be defined as combinatorial [4], and that multiple trans-coding services can be chained effectively together to perform a complex trans-coding task. So, instead of having all trans-coding done by one single trans-coding service, a number of trans-coding services can collaborate to achieve a composite adaptation task. For instance, trans-coding a 256-color depth *jpeg* image to a 2-color depth *gif* image can be carried out in two stages: the first

[1] This work was partially supported by the Natural Sciences and Engineering Research Council of Canada.

1-4244-0832-6/07/$20.00 ©2007 IEEE.

stage covers converting 256-color to 2-color depth, and the second step converts jpeg format to gif format. Transcoders can then be built in software and deployed and advertised easily and quickly to meet the needs of the users. Trans-coding would also be fast and reliable since its components can be simpler and they can also be replicated across the network.

Given a composite adaptation task that can be carried out in a number of stages, and given that there could be a number of possible configurations to adapt the sender's content to make it presentable at the receiver's device, the challenge is to find the appropriate chain of available trans-coding services that best fits the capabilities of the device, and at the same time, maximizes the user's satisfaction with the final delivered content. In this paper, we will discuss a Quality of Service (QoS) selection algorithm for providing personalized content through service composition. The function of the algorithm is to find the most appropriate chain of available trans-coding services between the sender and the receiver, and also to select the configuration for each trans-coding service. The proposed algorithm uses the user's satisfaction with the quality of the trans-coded content as the optimization metric for the path selection algorithm.

The rest of the paper is organized as follows: In Section 2, we will advocate content adaptation as a solution for interoperability, and the different approaches used in content adaptation. Section 3 lists all the required elements for providing customized content adaptation. In Section 4 we present our methodology for using the required element from Section 3 to construct a graph of trans-coding services; the algorithm for selecting the chain of trans-coding services is then presented. The selection criterion for the algorithm as well as its characteristics is also presented in Section 4, and finally, we end Section 4 with an example that shows step-by-step the results of the algorithm. Our conclusion is presented in Section 5.

2. Content Adaptation

In today's Internet, there is a wide range of client devices in terms of both hardware and software capabilities. Device capabilities vary in different dimensions, including processing power, storage space, display resolution and color depth, media type handling, and much more. This variety on device capabilities makes it extremely difficult for the content providers to produce a content that is acceptable and appreciated by all the client devices [5], making application-level adaptation a necessity to cover the wide variety of clients.

There are three main approaches for handling this diversity in content formats: a static content adaptation, a dynamic content adaptation, and a hybrid of the static and dynamic approaches [6,7]. The first two approaches differ in the time when the different content variants are created [8] to match the requested format. In static adaptation, the content creator generates and stores different variants of the same content on a content server, with each variant formatted for a certain device or class of devices. Hafid *et. al.* [9] presented an architecture for news-on-demand using this scheme. Static adaptation has three main advantages: (1) it is highly customized to specific classes of client devices, and (2) it does not require any runtime processing, so no delay is incurred, and (3) the content creator has the full control on how the content is formatted and delivered to the client. On the other hand, static adaptation has a number of disadvantages, mainly related to the management and maintenance of different variants of the same content [6]: (1) different content formats need to be created for each sort of device or class of devices, and needs to be re-done when new devices are introduced, and (2) it requires large storage space to keep all variants of the same content.

With dynamic content adaptation, the content is trans-coded from one format to the other only when it is requested. Depending on the location where the trans-coding takes place, dynamic content adaptation technologies can be classified into three categories: server-based, client-based, and proxy-based. In the server-based approach [10], the content server is responsible for performing the trans-coding; the content provider has all the control on how the content is trans-coded and presented to the user. Additionally, it allows the content to be trans-coded before it is encrypted, making it secure against malicious attacks. On the other hand, server-based adaptation does not scale properly for a large number of users and requires high-end content and delivery server to handle all requests.

As for the client-based approach [11,12], the client does the trans-coding when it receives the content. The advantage of this approach is that the content can be adapted to match exactly to the characteristics of the client. But at the same time, client-based adaptation can be highly expensive in terms of bandwidth and computation power, especially for small devices with small computational power and slow network connectivity, with large volume of data might be wastefully delivered to the device to be dropped during trans-coding.

The third adaptation approach is the proxy-based approach [1,13,14,15], where an intermediary computational entity can carry out content adaptation on the fly, on behalf of the server or client. Proxy adaptation has a number of benefits including leveraging the installed infrastructure and scaling properly with the number of clients. It also provides a clear separation between content creation and content adaptation. On the other hand, some content provider may argue that they prefer to have full control on how their content is presented to the user. Also, using proxies for adaptation does not allow the use of end-to-end security solutions.

3. Characterization and requirements for content adaptation

Advances in computing technology have led to a wide variety of computing devices, which made interoperability very difficult. Added to this problem is the diversity of user preferences when it comes to multimedia communications. This diversity in devices and user preferences has made content personalization an important requirement in order to achieve results that satisfy the user. The flexibility of any system to provide content personalization depends mainly on the amount of information available on a number of aspects involved in the delivery of the content to the user. The more information about these aspects is made available to the system, the more the content can be delivered in a format that is highly satisfactory to the user. These relevant aspects are: user preferences, media content profile, network profile, context profile, device profile, and the profile of intermediaries (or proxies) along the path of data delivery. We will briefly describe here each of these aspects; interested readers might refer to [16] for more details.

User Profile: The user's profile captures the personal properties and preferences of the user, such as the preferred audio and video receiving/sending qualities (frame rate, resolution, audio quality…). Other preferences can also be related to the quality of each media types for communication with a particular person or group of persons. For instance, a customer service representative should be able to specify in his profile his/her preference to use high-resolution video and CD audio quality when talking to a client, and to use telephony quality audio and low-resolution video when communicating with a colleague at work. The user's profile may also hold the user's policies for application adaptations, such as the preference of the user to drop the audio quality of a sport-clip before degrading the video quality when resources are limited. The MPEG-21 standard [17] is the most notable standards on user profiles.

Content Profile: Multimedia content might enclose different media types, such as audio, video, text, and each type can have different formats [8]. Each type has its format characteristics and parameters that can be used to describe the media. Such information about the content may include storage features, variants, author and production, usage, and many other metadata. The MPEG-7 standard [18], formally named "Multimedia Content Description Interface", offers a comprehensive set of standardized description tools to describe multimedia content.

Context Profile: A context profile would include any dynamic information that is part of the context or current status of the user. Context information may include physical (e.g. location, weather, temperature), social (e.g. sitting for dinner), or organizational information (e.g. acting senior manager). The MPEG-21 standard includes tools for describing the natural environment characteristics of the user, including location and time, as well as the audio and illumination characteristics of the user's environment. Resource adaptation engines can use these elements to deliver the best experience to the user.

Device Profile: To ensure that a requested content can be properly rendered on the user's device, it is essential to include the capabilities and characteristics of the device into the content adaptation process. Information about the rendering device may include the hardware characteristics of the device, such as the device type, processor speed, processor load, screen resolution, color depth, available memory, number of speakers, the display size, and the input and output capabilities. The software characteristics such as the operating system (vendor and version), audio and video codecs supported by the device should also be included in the device profile. The User Agent Profile (UAProf) created by the WAP Forum [19] and the MPEG 21 standard [20], both include description tools for describing device capabilities.

Network Profile: Streaming multimedia content over a network poses a number of technical challenges due to the strict QoS requirements of multimedia contents, such as low delay, low jitter, and high throughput [21]. Failing to meet these requirements may lead to a bad experience of the user [22,23]. With a large variety of transport networks, it is necessary to include the network characteristics into content personalization and to dynamically adapt the multimedia content to the fluctuating network resources [24]. Achieving this requires collecting information about the available resources in the network, such as the maximum delay, error rate, and available throughput on every link over the content delivery path. A description tool for network capabilities, including utilization, delay and error characteristics are included in the MPEG 21 standard.

Profile of Intermediaries: When the content is delivered to the user across the network, it usually travels over a number of intermediaries. These intermediaries have been traditionally used to apply some added-value services, including on-the-fly content adaptations services [13,15].

For the purpose of content adaptation, the profile of an intermediary would usually include a description of all the adaptation services that an intermediary can provide. These services can be described using any service description language such as JINI [25], SLP [26], or WSDL [27]. A description of an adaptation service would include, for instance, the possible input and output format to the service, the required processing and computation power of the service, and maybe the cost for using the service. The intermediary profile would also include information about the available resources at the intermediary (such as CPU cycles, memory) to carry out the services.

4. QoS selection algorithm

In this section, we will describe the overall QoS selection algorithm that finds the most appropriate chain of trans-coding services between the sender and the receiver, and also selects the configuration for each trans-coding service. We will first start by defining the user's satisfaction as the selection criterion for the algorithm, and then show how to construct the directed graph for adaptation, using the sender's content profile, receiver's device profile, and the list of available trans-coding services. After constructing the graph, we will show how to apply some optimization techniques on the graph to remove the extra edges in the graph, and finally present the actual QoS path and parameter selection algorithm.

4.1. User's satisfaction as selection criteria

Most Internet users are indifferent about the underlying technologies such as protocols, codecs, or resource reservation mechanisms that enable their communication session. They are also indifferent about network level QoS characteristics, such as bandwidth, delay, or throughput. All what is important for these users in the end is making the communication session work in a satisfactory way: for instance, hearing without jitter and seeing without irregularity.

As we mentioned earlier, the user's preferences expressed in the user's profile can be classified as application layer QoS parameters. In order to compute the user's satisfaction with all values of the application layer configuration parameters, we have used the approach presented in [28] by Richards *et. al.*, where each application level QoS parameter is represented by a variable x_i over the set of all possible values for that QoS parameter. The satisfaction or appreciation of a user with each quality value is expressed as a satisfaction function $S_i(x_i)$. All satisfaction functions have a range of [0..1], which corresponds to the minimum acceptable (M) and ideal (I) value of x_i. The satisfaction function $S_i(x_i)$ can take any shape, with the condition that it must increase monotonically over the domain. Figure 1 shows a possible satisfaction function for the frame rate variable.

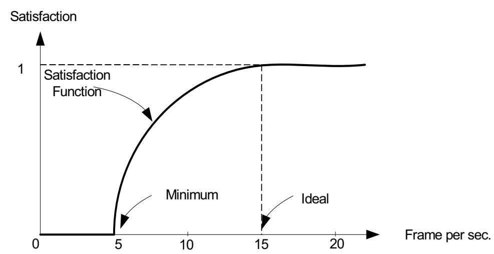

Figure 1. Possible satisfaction function for the frame rate.

In the case when there are more than one application parameter (frame rate, resolution, color depth, audio quality,…), Richards *et. al.* proposed using a combination function f_{comb} that computes the total satisfaction S_{tot} from the satisfactions s_i for the individual parameters (Equa. 1). An extension to this model is presented in [29].

$$S_{tot} = f_{comb}(s_1, s_2, s_3 \ldots, s_n) = \frac{n}{\sum_{i=1}^{n} \frac{1}{s_i}} \quad \text{(Equa. 1)}$$

4.2. Constructing a directed graph of trans-coding services

Now that we have decided on the selection criteria, the first step of the QoS selection algorithm would be to construct a directed acyclic graph for adaptation, using the content profile, device profile, and the list of available trans-coding services. Using this graph, the route selection algorithm would then determine the best path through the graph, from the sender to the receiver, which maximizes the user's satisfaction with the final received adapted content. The elements of the directed graph are the following:

1. Vertices in the graph represent trans-coding services. Each vertex of the graph has a number of properties, including the computation and memory requirements of the corresponding trans-coding service. Each vertex has a number of input and output links. The input links to the vertex represent the possible input formats to the trans-coding service. The output links are the output formats of the trans-coding service. Figure 2 shows a trans-coding service T1, with two input formats, F5 and F6, and four possible output formats, F10, F11, F12 and F13. The sender node is a special case vertex,

with only output links, while the receiver node is another special vertex with only input links.

To find the input and output links of each vertex, we rely on the information in different profiles. The output links of the sender are defined in the content profile, which includes as we mentioned earlier, meta-data information (including type and format) of all the possible variants of the content. Each output link of the sender vertex corresponds to one variant with a certain format. The input links of the receiver are exactly the possible decoders available at the receiver's device. This information is available through the description of the receiver's device in the device profile. The input and output links of intermediate vertices are described in the service description part of the intermediaries profile. Each intermediary profile includes the list of available trans-coding services, each with the list of possible input and output formats. Each possible input format is represented as an input link into the vertex, and the output format is represented as an output link.

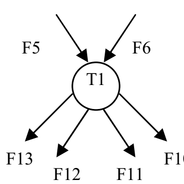

Figure 2. Trans-coding service with multiple input and output links

2. Edges in the graph represent the network connecting two vertices, where the input link of one vertex matches the output link of another vertex.

To construct the adaptation graph, we start with the sender node, and then connect the outgoing edges of the sender with all the input edges of all other vertices that have the same format. The same process is repeated for all vertices. To make sure that the graph is acyclic, the algorithm continuously verifies that all the formats along any path are distinct.

Figure 3 shows an example of an adaptation graph, constructed with one sender, one receiver, and seven intermediate vertices, each representing a trans-coding service. As we can see from the graph, the sender node is connected to the trans-coding service T1 along the edge labeled F5. This means that the sender S can deliver the content in format F5, and trans-coding service T1 can convert this format into format F10, F11, F12, or F13.

4.3. Adding constraints to the graph

As we have discussed earlier, the optimization criterion we have selected for the QoS selection algorithm is the user's satisfaction computed using the function f_{comb} presented in Section 4.1. The maximum satisfaction achieved by using a trans-coding service T_i depends actually on a number of factors.

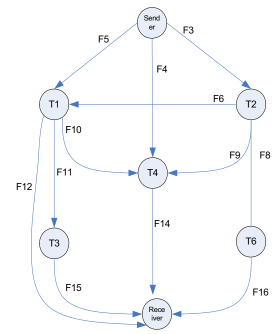

Figure 3. Directed trans-coding graph

The first factor is the bandwidth available for the data generated by the trans-coding service T_i. The more bandwidth is available to the trans-coding service, the more likely the trans-coding service will be able to generate trans-coded content that is more appreciated by the receiver. The available bandwidth between two trans-coding services is restricted by the amount of bandwidth available between the intermediate servers where the trans-coding service T_i is running and the intermediate server where the next trans-coding service or receiver is running. We can assume that connected trans-coding services that run on the same intermediate server have an unlimited amount of bandwidth between them.

Other factors that can affect the user's satisfaction are the required amount of memory and computing power to carry out the trans-coding operation. Each of these two factors is a function of the amount of input data to the trans-coding service.

4.4. QoS selection algorithm

Once the directed acyclic adaptation graph has been constructed, the next step is to perform the QoS selection algorithm to find a chain of trans-coding services, starting from the sender node and ending with the receiver node, which generates the maximum satisfaction of the receiver. Finding such as path can be similar to the problem of finding the shortest path in a directed weighted graph with similar complexity, except that the optimization criterion is the user's satisfaction, and not the available bandwidth or the number of hops.

Our proposed algorithm uses two variables representing two sets of trans-coding services, the set of already considered trans-coding services, called VT, and the set of candidate trans-coding services, called CS, which can be added next on the partially selected path. The candidate trans-coding services set contains the trans-coding services that have input edges coming from any trans-coding service in the set VT. At the beginning of the algorithm, the set VT contains only the *sender* node, and CS contains all the other trans-coding services in the graph that are connected to *sender*, and also the *receiver*. In each iteration, the algorithm selects the trans-coding service T_i that , when using it, generates the highest user satisfaction. The user satisfaction is computed as an optimization function of the audio and video parameters for the output format for T_i, subject to the constraint of available bandwidth between T_i and its ancestor trans-coding service, and also subject to the remaining user's budget. T_i is then added to VT. The CS set is then updated with all the neighbor trans-coding services of T_i. The algorithm stops when the CS set is empty, or when the *Receiver* node is selected to be added to VT. The complete description of the algorithm is given in Figure 4.

```
Step 1:   // Let VT be the set of all considered trans-coding services.
          VT = {sender};
          // Let CS be the set of all direct neighbor transcoders of all
          // transcoders in VT
          CS = neighbor(sender);
          // Let user_budget be the amount of money the user is willing to
        pay
Step 2:   // Each trans-coding service keeps a track of its parent trans-coding
          // service. Let T_prev be the trans-coding services in CS connected to
          // T_i; Compute the perceived user's satisfaction for using all the
        trans-
          // coding services in CS, subject to two constraints: the remaining
          // user budget and the available bandwidth between T_i and T_prev.
          For ∀ T_i ∈ CS
            Optimize( user_profile, input_format, output_format, Sat_T[i],
                                   user_budget,cost,available_bandwith)
Step 3:   // If there are no more transcoders to consider and the receivers can
          // not be reached from the sender through any transcoding path.
          if is_empty(CS) then
              TERMINATE(FAILURE)
Step 4:   Select the trans-coding service T_i that has the highest satisfaction
              value Sat_T[i], for the user.
          CS = CS – { T_i };
Step 5:   VT = VT + { T_i };
Step 6:   Let T_i. previous_selected_transcoder = T_prev;
          T_i.accumulated_cost = T_i.previous.accumulated_cost +
              transcoding_and_transmission_cost [i];
Step 7:   if T_i = receiver , then GOTO Step 10
Step 8:   // compute the satisfaction for using all the neighboring transcoders
          // of T_i and add them to CS
          For ∀ T_j ∈ neighbors( T_i );
            Optimize( user_profile, input_format, output_format, Sat_T[i],
                                   user_budget,cost[i],available_bandiwth)
              CS = CS ∪ {T_j };
Step 9:   GOTO Step 3
Step 10:  Print the reverse path from the Receiver to the Sender, by following
              the link "previous" of all transcoders, starting from the Receiver.
```

Figure 4. Pseudo-code for the route selection algorithm

As indicated in Step 2 and Step 8, the algorithm selects from CS the transcoder T_i that can generate the highest satisfaction value for the receiver. To compute the satisfaction value for each transcoder T_i in CS, the algorithm selects the QoS parameter values x_i that optimize the satisfaction function in Equa. 2, subject only to the constraint remaining user's budget and the bandwidth availability that connects T_i to T_{prev} in VT. i.e.

$$bandwith\_requirement(x_1..x_n) \leq Bandwidth\_AvailableBetween(T_i, T_{prev}). \quad \text{(Equa. 2)}$$

Since each trans-coding service can only reduce the quality of the content, when the algorithm terminates, the algorithm would have computed the best path of trans-coding services from the *sender* to the *receiver*, and the user's satisfaction value computed on the last edge to the receiver node is the maximum value the user can achieve. To show this, assume that the selected path is the path $\{T_{11},\ldots T_{1n}\}$ in Figure 5. If the path $\{T_{21},\ldots T_{2m}\}$ is a better path, then T_{2m} should have converted the content into variant that is more appreciated by the user than the variant generated by T_{1n}. Since transcoders can only reduce the quality of content, all transcoders along the path $\{T_{21},\ldots T_{2m}\}$, should have also produced a content with higher satisfaction function than the variant produce by T_{1n}, and hence all these transcoders should have been selected before T_{1n}, which contradicts with the assumption.

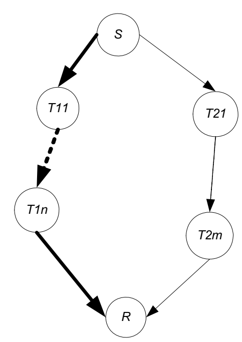

Figure 5. Graph selection

4.5. Example

In this section, we will present an example to show how the QoS path selection algorithm works. We will assume that the graph construction algorithm has generated the graph shown in Figure 6. The graph also shows the selected path with and without trans-coding service T_7 as part of the graph. The selected trans-coding services, user satisfaction, as well as the best current path produced by the algorithm are also shown in Table 1. Each row in the table shows the results for one iteration of the algorithm.

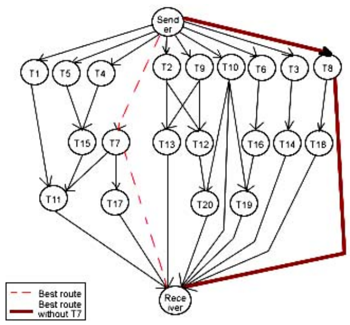

Figure 6. Example of trans-coding graph

Table 1. Results for each step of the path selection algorithm

Round	Considered Set (VT)	Candidate set (CS)	Selected trans-coding service	Selected Path	Deliverd Frame Rate	User satisfaction
1	{ *sender* }	{T1, T2, T3, T4, T5, T6, T7, T8, T9, T10}	T10	*sender*,T10	30	1.00
2	{ *sender*, T10}	{T1, T2, T3, T4, T5, T6, T7, T8, T9, T19, T20, *receiver*}	T20	*sender*,T10,T20	30	1.00
3	{ *sender*, T10, T20}	{T1, T2, T3, T4, T5, T6, T7, T8, T9, T19, *receiver*}	T5	*sender*,T5	27	0.90
4	{ sender, T10, T20, T5}	{T1, T2, T3, T4, T6, T7, T8, T9, T19, T15, receiver}	T4	sender,T4	27	0.90
5	{ sender, T10, T20, T5, T4}	{T1, T2, T3, T6, T7, T8, T9, T19, T15, receiver}	T3	sender,T3	23	0.76
6	{ sender, T10, T20, T5, T4, T3}	{T1, T2, T6, T7, T8, T9, T19, T15, T14, receiver}	T2	sender,T2	23	0.76
7	{ sender, T10, T20, T5, T4, T3, T2}	{T1, T6, T7, T8, T9, T19, T15, T14, T12, T13, receiver}	T1	sender,T1	23	0.76
8	{ sender, T10, T20, T5, T4, T3, T2, T1}	{T6, T7, T8, T9, T19, T15, T14, T12, T13, T11, receiver}	T11	sender,T1, T11	23	0.76
9	{ sender, T10, T20, T5, T4, T3, T2, T1, T11}	{T6, T7, T8, T9, T19, T15, T14, T12, T13, receiver}	T13	sender,T2, T13	23	0.76
10	{ sender, T10, T20, T5, T4, T3, T2, T1, T11, T13}	{T6, T7, T8, T9, T19, T15, T14, T12, receiver}	T12	sender,T2, T12	23	0.76
11	{ sender, T10, T20, T5, T4, T3, T2, T1, T11, T13, T12}	{T6, T7, T8, T9, T19, T15, T14, receiver}	T14	sender,T3, T14	23	0.76
12	{ sender, T10, T20, T5, T4, T3, T2, T1, T11, T13, T12, T14}	{T6, T7, T8, T9, T19, T15, receiver}	T8	sender, T8	20	0.66
13	{ sender, T10, T20, T5, T4, T3, T2, T1, T11, T13, T12, T14, T8}	{T6, T7, T9, T19, T15, receiver}	T7	sender, T7	20	0.66
14	{ sender, T10, T20, T5, T4, T3, T2, T1, T11, T13, T12, T14, T8, T7}	{T6, T9, T19, T15, receiver}	T6	sender, T6	20	0.66
15	{ sender, T10, T20, T5, T4, T3, T2, T1, T11, T13, T12, T14, T8, T7, T6}	{T9, T19, T15, receiver}	receiver	sender, T7, receiver	20	0.66

5. Conclusion

Content adaptation is a natural solution to the problem of heterogeneity in client devices, network connectivity, content format, and users' preferences. This paper presented a framework for adding several adaptation services to multimedia to make the content more satisfactory to the user. An important part of the framework is the QoS path selection algorithm that decides on the chain of adaptation services to add and the configuration parameters for each service.

6. References

[1] S. Chandra and C.S. Ellis, "JPEG Compression Metric as a Quality Aware Image Transcoding," in Second Usenix Symposium on Internet Technologies and Systems (USITS '99), Oct 12, 1999.

[2] R. Han, P. Bhagwat, R. LaMaire, T. Mummert, V. Perret, and J. Rubas, "Dynamic adaptation in an image trans-coding proxy for mobile WWW browsing," IEEE Personal Communication, vol. 5 (6), Dec. 1998.

[3] Cisco CallManager. http://www.cisco.ca

[4] J. R. Smith, R. Mohan, C.-S. Li, "Scalable Multimedia Delivery for Pervasive Computing," ACM Multimedia '99, Oct. 30 - Nov. 5, 1999.

[5] A. Fox, S.D. Gribble, and Y. Chawathe, "Adapting to Network and Client Variation Using Infrastructural Proxies: Lessons and Perspectives," IEEE Personal Communications, August 1998.

[6] W.Y. Lum and F.C.M. Lau, "On Balancing Between Trans-coding Overhead and Spatial Consumption in Content Adaptation," Mobicom 2002, Atlanta, USA, September 2002, 239-250.

[7] C. Y. Chang and M. S. Chen "Exploring Aggregate Effect with Weighted Transcoding Graphs for Efficient Cache Replacement in Transcoding Proxies," in Proceedings of the 18th IEEE International Conference on Data Engineering (ICDE-O), 2002.

[8] Z. Lei and N.D. Georganas, "Context-based Media Adaptation in Pervasive Computing," Proc. Can.Conf. on Electr. and Comp. Engg (CCECE'2001), Toronto, May 2001.

[9] A. Hafid and G.v. Bochmann, "Quality of Service Negotiation in News-on-Demand Systems: an Implementation," In Proceedings of the Third International Workshop on Protocols for Multimedia Systems, Madrid, Spain, 1996.

[10] R. Mohan, J.R. Smith and C.S. Li, "Adapting Multimedia Internet Content for Universal Access," IEEE Trans. on Multimedia, 1(1):104–114, 1999.

[11] S. Björk, L.E. Holmquist, J. Redström, I. Bretan, R. Danielsson, J. Karlgren, and K. Franzén, "WEST: a Web browser for small terminals," Proceedings of the 12th annual ACM symposium on User interface software and technology, p.187-196, November 07-10, 1999, Asheville, North Carolina, United States.

[12] B. Fisher, G. Agelidis, J. Dill, P. Tan, G. Collaud, and C. Jones, "CZWeb: Fish-Eye Views for Visualizing the World-

Wide Web," Proc. of the 7th Int. Conf. on Human-Computer Interaction (HCI International '97), 719-722, 1997.

[13] S. Chandra, C. Ellis, and A. Vahdat, “Application-Level Differentiated Multimedia Web Services Using Quality Aware Transcoding,” IEEE Journal on Selected Areas in Communications, 2000.

[14] R. Floyd and B. Housel, “Mobile Web Access Using eNetwork Web Express,” IEEE Personal Communications, 5(5):47–52, 1998.

[15] A. Fox, S.D. Gribble, Y. Chawathe, E.A. Brewer, and P. Gauthier, “Cluster-Based Scalable Network Services,” in Proc. 16th ACM Symp. On Operating Systems Principles, pages 78–91, Saint-Malo, France, 1997.

[16] K. El-Khatib and G.v Bochmann, G.v., “Profiles in Content Adaptation,” Technical report. University of Ottawa, Canada. Dec 2003.

[17] MPEG-21. http://www.chiariglione.org/mpeg/standards /mpeg-21/ mpeg-21.htm. Accessed on Jan 10, 2007.

[18] MPEG-7. http://www.chiariglione.org/mpeg/standards/ mpeg-7/mpeg-7.htm. Accessed on Jan 10, 2007.

[19] Wireless Application Forum, http://www.wapforum.org/

[20] MPEG-21 Standard. http://www.chiariglione.org/mpeg/ standards/mpeg-21. Accessed on Jan 10, 2007.

[21] C.W. Ng, P.Y. Tan, and H. Cheng, "Quality of Service Extension to IRML," IETF INTERNET-DRAFT, 'draft-ng-opes-irmlqos-00.txt', July 2001.

[22] M. Katchabaw, H. Lutfiyya, and M. Bauer, “Driving resource management with application-level quality of service specifications,” ICE 98, ACM Press (1998), 83-91.

[23] C. Poellabauer, H. Abbasi, and K. Schwan, “Cooperative run-time management of adaptive applications and distributed resources,” Proc. Multimedia ’02, ACM Press (2002).

[24] D. Wu., Y.T. Hou, and Y. Zhang, "Scalable Video Coding and Transport over Broad-band Wireless Networks," Proc. IEEE, vol. 89, no. 1, pg 6-20, Jan 2001.

[25] JINI (TM): Http: //java.sun.com/product/JINI/. 1998.

[25] E. Guttman, C. Perkins, J. Veizades, and M. Day, “Service Location Protocol. Version 2.,” http://ietf.org/rfc/rfc2608.txt.

[26] W3C. WSDL: Web Service Description Language, http://www.w3.org/TR/wsdl, 2002.

[27] A. Richards, G. Rogers, V. Witana, and M. Antoniades, “Mapping user level QoS from a single parameter,” In Second IFIP/IEEE International Conference on Management of Multimedia Networks and Services, Versailles, November 1998.

[28] K. El-Khatib, Zhen E. Zhang, N. Hadibi and G. v. Bochmann, “Personal and Service Mobility in Ubiquitous Computing Environments", Journal of Wireless communications and Mobile Computing, v. 4, pp. 595-607, 2004.