A Distributed Content Adaptation framework for Content Distribution Networks

Khalil El-Khatib, Gregor v. Bochmann, and Abdulmotaleb El Saddik
School of Information Technology & Engineering, University of Ottawa
800 King Edwards, Ottawa, Ontario, Canada K1N 6N5
elkhatib@site.uottawa.ca

Abstract. *The purpose of this research is to present topology aware overlays for efficient distributed multimedia application support. The discussed infrastructure for trans-coding multimedia streams uses self-organizing, resilient data distribution algorithms. The infrastructure 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 infrastructure 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 PC users, and cannot be rendered directly on all types of client devices. Yahoo and e-Bay have taken recently 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-coders have been implemented in hardware and can be deployed on intermediate 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-coders can be chained effectively together to perform a complex trans-coding task. So, instead of having all trans-coding done by one single trans-coder, a number of trans-coders 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 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 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 trans-coders 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. The function of the algorithm is to find the most appropriate chain of trans-coders between the sender and the receiver, and also to select the configuration for each trans-coder. The proposed algorithm uses the user's satisfaction with the quality of the trans-coded content as the optimization metric for the path selection function. Our approach is inspired by the work of Mao *et. al* [5], in the way we construct a chain of trans-coders to match the capabilities of the sender and receiver. Our approach is different though in the way we select the sequence of trans-coders. While Mao *et. el.* used network based characteristics such as data throughput, jitter, or delay to select the trans-coders, our approach is more user centric and uses the user's satisfaction as the only selection criterion. This approach is based on the observation [6,7,8,9,10] that different transport level QoS may generate similar user satisfaction, and that it is best to select a trans-coding path based on the end result, which is the user's satisfaction, and not based on single, independent, low-level factors such as delay, bandwidth, or throughput.

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 **Error! Reference source not found.** to construct a graph of trans-coders; the algorithm for selecting the chain of trans-coders 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.

Deleted: 3

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 [11], making application-level adaptation a necessity to cover the wide population of clients. The problem is even more challenging when multicasting the content to a large number of receivers with heterogeneous device capability and preferences.

There are two main approaches for handling this diversity in content formats: static content adaptation and dynamic content adaptation, with a number of hybrids combining both approaches [12,13]. These two approaches differ in the time when the different content variants are created [14] 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.* [15] 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 [12]: (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 [16], 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 [17,18], 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 [1,19,20,21], 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 control themselves 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, and 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 each of these aspects; interested readers might refer to [22] 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. Some other information in the user profile might include also the user's authorization, authentication and accounting information.

One of the most notable work on user profiles is the MPEG-21 standard [23], which describes attributes of the end user of multimedia content, including besides name and contact information, also content preferences, presentation preferences, accessibility and mobility preferences. These preferences are used for instance to provide effective and efficient access (search, filtering and browsing) to multimedia content.

Content Profile: Multimedia content might enclose different media types, such as audio, video, text, and each type can have different formats [14]. Each type and format has a number of characteristics and parameters that can be used to describe the media. Such information, referred to as meta-data information, is usually included in the content profile. Some of this meta-data about the content may include:

- Information about the storage features of the content, such as the type of media (video, audio, etc), the transport protocol (RTP/UDP/IP, H.320, etc), and the format (H.261 video, MPEG video, etc).
- Information about available variants of the content, such as colored-and-black and white variants,
- Information about the author and production of the content, such as the title, and date of creation.
- Information related to the usage of the content, such as copyright, application adaptations, and usage history.

The MPEG-7 standard [24], formally named "Multimedia Content Description Interface", offers a comprehensive set of standardized description tools to describe multimedia content. These tools allow for a complete description of what is depicted in the content, the form (coding format and size), the condition for accessing the material, the classification, the context, and the links to other relevant material. MPEG-7 provides also tools for describing variations of the content such as summaries and abstracts; scaled, compressed and low-resolution versions; and versions with different languages and modalities – audio, video, image, text, and so forth. Using the content profile, a content adaptation system can decide what type of adaptations can be applied to the content.

Context Profile: The notion of context and its implications has been a research topic for a number of research groups [25,26,27] and is still attracting more interest. According to [28] and [29], the context can be generally defined as: "any information that can be used to characterize the situation of an entity. An entity is a person, place or object that is considered relevant to the interaction between a user and an application, including the user and the application themselves." Based on this definition, 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). Some context information, such as the role or task of the user, can be manually keyed in by the user, while other information, such as location, time of the day, weather condition, can be easily gathered using sensing devices. Some other information, such as the current status of the user, can be gathered from other sources such as the calendar of the user or from a meeting attendees list. 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 is properly rendered on the user's device, it is essential to include the capabilities and characteristics of the device into the content personalization 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 [30] and the MPEG 21 standard [31], 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 [32]. Failing to meet these requirements may lead to a bad experience of the user [33,34]. With a large variety of wired and wireless network connectivity, it is necessary to include the network characteristics into content personalization and to dynamically adapt the multimedia content to the fluctuating network resources [35]. 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 [19,21]. Using intermediaries for applying adaptations alleviates the problem of clients with limited-resources [36] and overloaded servers [16].

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 [37], SLP [38], or WSDL [39]. 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. Note that the available bandwidth through an intermediary can also be included in the intermediary profile, but for clarity reasons, we have decided to include it in the network profile.

4. QoS selection algorithm

In this section, we will describe the overall QoS selection algorithm that finds the most appropriate path of trans-coders between the sender and the receiver, and also selects the configuration for each trans-coder. 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-coders. 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 them 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 [40] 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.

Formatted: Justified, Indent: First line: 0.5"

Figure 1 shows a possible satisfaction function for the frame rate variable.

Formatted: Font: 10 pt

Deleted: ¶
Figure 1

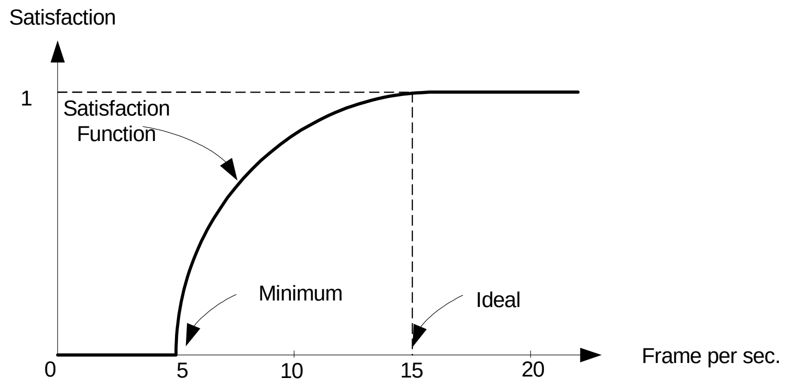

Figure 1. Possible satisfaction function for the frame rate.

In the case when there are more then one application parameter (frame rate, resolution, color depth, audio quality,…), Richards *et. al.* proposed using a combination function f_{comb} that determines the total satisfaction S_{tot} from the satisfactions s_i for the individual parameters as follows:

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

The function f_{comb} has two important properties:

- One individual low satisfaction is enough to bring the total satisfaction to a low value.
- The total satisfaction of equal individual satisfactions s_i is equal to the satisfactions s_i.

We note also that f_{comb} is a many to one mapping function, and hence different combinations of individual satisfaction values are possible for one value of S_{tot}. To find out what is the best possible combination of individual satisfaction functions, another selection criterion is needed. The most reasonable selection criterion is the charging cost. Providing a tariff structure which determines the cost for the different values x_i of the individual application parameters, one can devise an optimization strategy for finding application parameter values that minimize the cost for a given global satisfaction S_{tot}, or maximize the satisfaction S_{tot} for a given cost value.

4.2 Extending user's satisfaction to support weighted combination and multi-user conference sessions

We think that the approach described above is a major step towards a simple user-friendly interface for user level QoS specification; however, further considerations could be taken into account as described below. A first improvement results from the observation that users in telecommunication session might find some media types more important than others. For instance, a user of a news-on-demand service might prefer to receive high quality audio with low quality video as compared to average quality audio and average quality video. In the case of a user watching a sport event the situation may be the opposite (if the user does not care about the audio of the commenter).

This preference to individual media can play a factor when it comes to the calculation of the total satisfaction S_{tot}. By assigning different weights w_i to the different parameters x_i, S_{tot} will reflect the user preference for different media types. The combination function for the total user satisfaction can be redefined as follows:

$$S_{tot}^{user} = f_{comb}(s_1, s_2, s_3 \ldots, s_n, w_1, w_2, w_3 \ldots, w_n) = \frac{n\overline{w}}{\sum_{i=1}^{n} \frac{w_i}{s_i}} \qquad \text{(Equa. 2)}$$

where w_i is the weight for the individual satisfaction s_i and $\overline{w} = \frac{\sum_{i=1}^{n} w_i}{n}$.

These constants weights factors can be selected by the user, and stored in the user profile (AudioWeightFactor, VideoWeightFactor,..). The selection of these weights depends on the type of service the user is willing to receive when using a specific service or communicating with a given callee.

Additionally, we have so far considered only the QoS preferences of a single user. But all conversational multimedia applications involve several users. It is therefore important to determine how the possibly conflicting preferences of the different users are reconciled in order to come up with QoS parameters that are suitable for all participating users.

In certain circumstances, some given parameters may be determined simply based on the preferences of a single user. This may be the case in a two-way teleconference between two users A and B, where the parameters of the video visible by User A would be determined based on the preferences of User A alone, and the video in the opposite direction based on the preferences of User B. However, the situation may be more complex if the cost of the communication is paid by User A and the selection of the video received by User B has an impact on the communication cost.

In other circumstances, as for instance in the case of the joint viewing of a video clip by several participants in a teleconference, the selected quality parameters should be determined based on the preferences of all participating users. In such circumstances, we propose to use the same combination function for user satisfaction considered above and (optionally) introduce a weight for each of the participating users, called the *QoS selection weight*, which determines how much the preferences of the user influences overall QoS parameter selection. The total satisfaction (computed for all users) is then given by

$$S_{tot} = f_{comb}(s_{tot}^{usr_1}, s_{tot}^{usr_2} \ldots, s_{tot}^{usr_n}, a_1, a_2 \ldots, a_n) \qquad \text{(Equa. 3)}$$

where $s_{tot}^{usr_i}$ is the total satisfaction for user *i*, and a_i is the QoS selection weight for user *i*. In the case that the weight of a given user is zero, the preferences of this user are not taken into account for the selection of the QoS parameters.

4.3 Constructing a directed graph of trans-coders

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-coders. 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 intermediate trans-coders. Each vertex of the graph has a number of properties, including the computation and memory requirements of the corresponding trans-coder. Each vertex has a number of input and output links. The input links to the vertex represent the possible input formats to the trans-coder. The output links are the output formats of the trans-coder. Figure 2 shows a trans-coder 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 intermediaries profile. Each intermediary profile includes the list of available trans-coders, 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.

Formatted: Font: 10 pt, Not Bold

Deleted: Figure 2

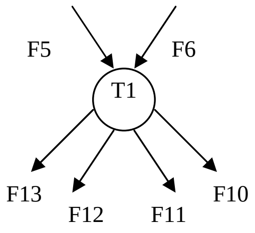

Figure 2. Trans-coder with multiple input and output links

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 from the sender 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-coder. As we can see from the graph, the sender node is connected to the trans-coder T1 along the edge labeled F5. This means that the sender S can deliver the

Deleted: Figure 3

Formatted: Font: 10 pt, Not Bold

content in format F5, and trans-coder T1 can convert this format into format F10, F11, F12, or F13.

In order to find the adaptations services that can be used for constructing the adaptation graph, we have used the concept of service directories, where each administrative domain in the Internet would use one or more directory services to advertise its publicly available adaptation services. This concept of service directories is not new to the area of distributed computing, and different standards are already well established [41,42]. Before constructing the graph, the sender sends a request to number of directory services to find what adaptation services are available and can be used in the graph. Another alternative to find all adaptation services would be to have each administrative domain add the available adaptation services to the content request message, as it travels from the receiver to the sender.

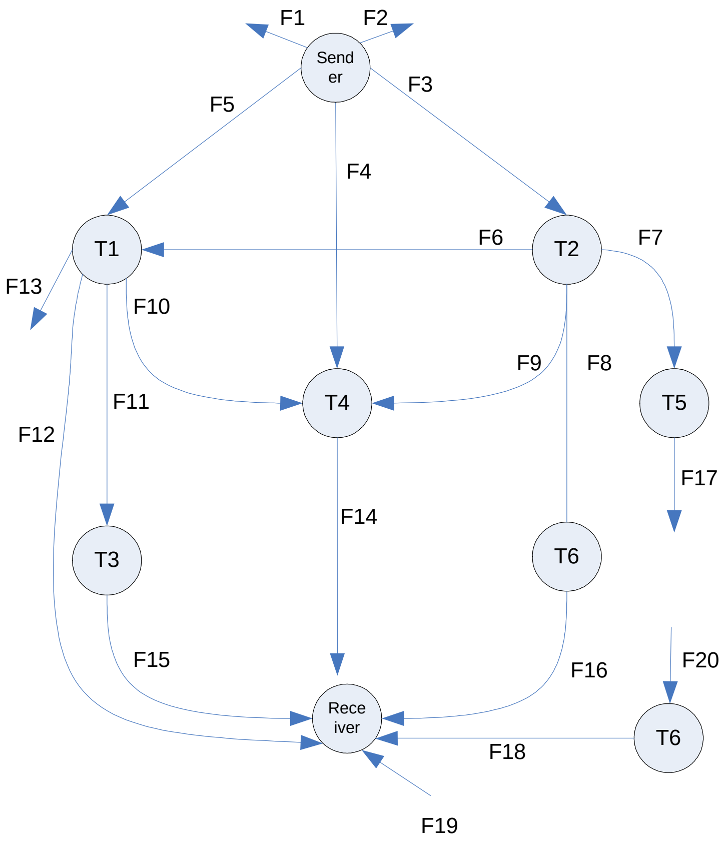

Figure 3. Directed trans-coding graph

4.4 Graph Optimization

By looking at the graph in Figure 3, we can see that there are some edges like F1, F2 or F17 that are connected only to one trans-coder. These edges cannot be a part of any path from the sender to the receiver. The same principle applies also to trans-coders other than the sender and receiver that are not on any path from the sender to the receiver. T6 is an example of a trans-coder that cannot be used to send data through on its way from the sender to the receiver. Removing these edges and vertices help reduce the computational time of the algorithm, since it helps pruning dead-ends from the graph. Applying

Formatted: Font: 10 pt, Not Bold

Deleted: Figure 3

optimization for the graph in Figure 3 would result in the graph shown in Figure 4.

Deleted: Figure 3

Formatted: Font: 10 pt, Not Bold, Do not check spelling or grammar

Deleted: Figure 4

Formatted: Font: 10 pt, Not Bold

Formatted: Font: 10 pt, Not Bold

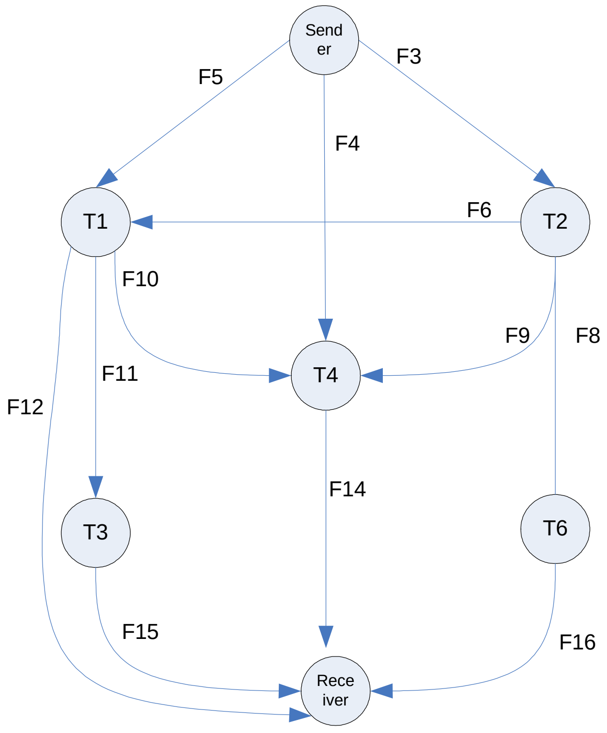

Figure 4. Optimized directed trans-coding graph

4.5 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 4.1. The maximum satisfaction achieved by using a trans-coder T_i depends actually on a number of factors.

The first factor is the bandwidth available for the data generated by the trans-coder T_i. The more bandwidth is available to the trans-coder, the more likely the trans-coder will be able to generate trans-coded content that is more appreciated by the receiver. The available bandwidth between two trans-coders is restricted by the amount of bandwidth available between the intermediate servers where the trans-coder T_i is running and the intermediate server where the next trans-coder or receiver is running. We can assume that connected trans-coders 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-coder.

4.6 Topology aware overlays for efficient application support

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-coders, 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, except that the optimization criterion is the user's satisfaction, and not the available bandwidth or the number of hops.

The algorithm uses two variables representing two sets of trans-coders, the set of already considered trans-coders, called VT, and the set of candidate trans-coders, called CS, which can be added next on the partially selected path. The candidate trans-coders set contains the trans-coders that have input edges coming from any trans-coder 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-coders in the graph that are connected to *Sender*, and also the *Receiver*. At each step of the protocol, the satisfaction of the user is evaluated for adding each of the trans-coders in the CS set, and the trans-coder T_i that generates the highest satisfaction is selected and added to VT. The CS set is then updated with all the neighbor trans-coders 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 below:

Step 1:	Let VT = {*Sender*} be the set of all considered trans-coders. Let CS be the set of all downstream neighboring transcoders of the *Sender*. Add the *Receiver* also to the set, in case there was a direct link from the *Sender* to the *Receiver.*
Step 2:	If CS is empty, then **TERMINATE**(FAILURE)
Step 3:	Compute the perceived user's satisfaction for using all the trans-coders in CS.
Step 4:	Select the trans-coder T_i that has the highest satisfaction value. Delete T_i from CS.
Step 5:	Let T_{prev} be the trans-coder in CS connected to T_i;
Step 6:	Let T_i.*previous* = T_{prev};
Step 7:	If the selected trans-coder T_i is the *Receiver* node, then **GOTO** Step 8.
Step 8:	Add to CS all the trans-coders to which T_i is directly connected.
Step 9:	**GOTO** Step 2
Step 10:	Print the reverse path from the *Receiver* to the *Sender*, by following the link "*previous"* of all transcoders, starting from the *Receiver.*

As indicated in Step 4, 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. 3, subject only to the constraint of bandwidth availability that connects T_i to T_{prev} in VT. i.e.

$$bandwith\_requirement(x_1 .. x_n) \leq Bandwidth\_AvailableBetween(T_i , \mathrm{T}_{prev}).$$

Since each trans-coder can only reduce the quality of the content, when the algorithm terminates, the algorithm would have computed the best path of trans-coders 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 $\{T_{11},\dots T_{1n}\}$ in Figure 5. If the path $\{T_{21},\dots T_{2m}\}$ is a better path, then T_{12m} 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 $\{T_{21},\dots T_{2m}\}$, 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.

Formatted: Font: 10 pt, Not Bold

Deleted: Figure 5

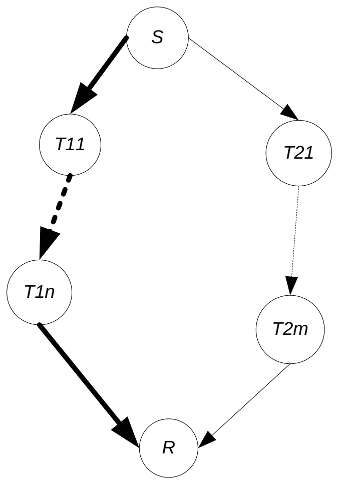

Figure 5. Graph selection

4.7 Characteristics of the algorithm

The proposed algorithm has a number of characteristics, including:

- **Self-organizing:** In general, the design of protocols for networks with changing resources is non-trivial. In a highly dynamic environment such as the Internet, one can not make any assumption regarding the number and location of adaptation services. By using directory services, the system can easily adapts to the changes in the set of adaptation services available in the network. New adaptation services can be easily introduced and removed from the network to meet the dynamic requirements of the users.
- **Resilient against service failure:** If any of the adaptation services fails, the sender can execute the algorithm again to find another chain of adaptation services. The sender node might even take a chance and try to select a path from the already constructed graph to shorten the service disruption period

4.8 Example

In this section, we will present an example to show how the algorithm works. We will assume that the graph construction part of the algorithm has generated the graph shown in Figure 6. The graph shows also the selected path with and without trans-coder T_7 as part of the graph. The selected trans-coders, 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.

Deleted: Figure 6

Formatted: Font: 10 pt, Not Bold

Deleted: Table 1

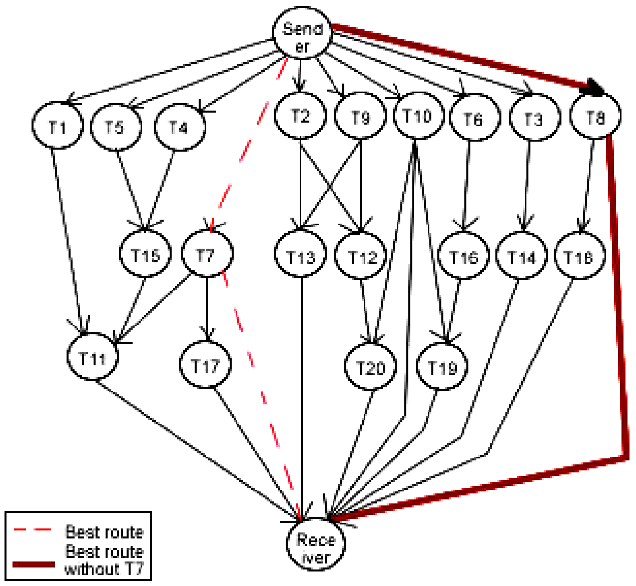

Figure 6. Example of trans-coding graph

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

Round	Considered Set (VT)	Candidate set (CS)	Selected trans-coder	Selected Path	User satisfaction
1	{Sender}	{T1, T2, T3, T4, T5, T6, T7, T8, T9, T10}	T10	Sender,T10	1.00
2	{Sender, T10}	{T1, T2, T3, T4, T5, T6, T7, T8, T9, T19, T20, Receiver}	T20	Sender,T10,T20	1.00
3	{Sender, T10, T20}	{T1, T2, T3, T4, T5, T6, T7, T8, T9, T19, Receiver}	T5	Sender,T5	0.90
4	{Sender, T10, T20, T5}	{T1, T2, T3, T4, T6, T7, T8, T9, T19, T15, Receiver}	T4	Sender,T4	0.90
5	{Sender, T10, T20, T5, T4}	{T1, T2, T3, T6, T7, T8, T9, T19, T15, Receiver}	T3	Sender,T3	0.76
6	{Sender, T10, T20, T5, T4, T3}	{T1, T2, T6, T7, T8, T9, T19, T15, T14, Receiver}	T2	Sender,T2	0.76
7	{Sender, T10, T20, T5, T4, T3, T2}	{T1, T6, T7, T8, T9, T19, T15, T14, T12, T13, Receiver}	T1	Sender,T1	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	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	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	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	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	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	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	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	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.

Reference

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 . Z. Morley Mao, H. Wilson So, B. Kang, and R. H. Katz, “Network Support for Mobile Multimedia using a Self-adaptive Distributed Proxy,” 11th International Workshop on Network and Operating Systems Support for Digital Audio and Video (NOSSDAV-2001).
6. R. Procter, M. Hartswood, A. McKinlay, S. Gallacher, “An investigation of the influence of network quality of service on the effectiveness of multimedia communication,” Group 99, ACM Press (1999), 160-168.
7. J. Gowan and J. Downs, “Video conferencing human-machine interface: A field study,” Information and Management, 27 (1994), 341-356.
8. I. Busse, B. Deffner, and H. Schulzrinne, “Dynamic QoS control of multimedia applications based on RTP,” Computer Communications, 19, 1996, pp. 49-58.
9. A. Bouch, A. Kuchinsky, and N. Bhatti, “Quality is in the eye of the beholder: Meeting users’ requirements for internet quality of service,” Proc. CHI 2000, CHI Letters 2(1), 297-304.
10. R. Apteker, J. Fisher, V. Kisimov, and H. Neishlos, “Video acceptability and frame rate,” IEEE Multimedia, 2, 3 (1995), 32-40.
11. 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.
12. 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.
13. 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.
14. 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
15. 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
16. 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.
17. 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
18. 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.
19. 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.

20. R. Floyd and B. Housel, “Mobile Web Access Using eNetwork Web Express,” IEEE Personal Communications, 5(5):47–52, 1998.
21. 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.
22. K. El-Khatib and G.v Bochmann, G.v., “Profiles in Content Adaptation,” Technical report. University of Ottawa, Canada. Dec 2003.
23. http://www.chiariglione.org/mpeg/standards/mpeg-21/mpeg-21.htm
24. http://www.chiariglione.org/mpeg/standards/mpeg-7/mpeg-7.htm
25. Ubiquitous Computing (Xerox PARC) http://www.ubiq.com/hypertext/wiser/UbiHome.html
26. Contextual Computing Group (Georgia Tech). http://www.cc.gatech.edu/ccg/
27. Ambiante (Collaborative Buildings) (GMD) www.darmstadt.gmd.de/ambiente
28. A.K. Dey, “Providing architectural Support for Building Context-Aware Applications,” PhD thesis, College of Computing, Georgia Institute of Technology, December 2000.
29. A. Schmidt, M. Beigl, and H. Gellersen, “There is More to Context than Location”, Computers & Graphics, vol. 23, no. 6, Dec. 1999, pp. 893-902; http://www.elsevier.com.
30. Wireless Application Forum, http://www.wapforum.org/
31. http://www.chiariglione.org/mpeg/standards/mpeg-21/mpeg-21.htm
32. Ng, C.W., Tan, P.Y., and Cheng, H., "Quality of Service Extension to IRML," IETF INTERNET-DRAFT, 'draft-ng-opes-irmlqos-00.txt', July 2001.
33. M. Katchabaw, H. Lutfiyya, and M. Bauer, “Driving resource management with application-level quality of service specifications,” ICE 98, ACM Press (1998), 83-91.
34. C. Poellabauer, H. Abbasi, and K. Schwan, “Cooperative run-time management of adaptive applications and distributed resources,” Proc. Multimedia ’02, ACM Press (2002).
35. 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.
36. S. Hollfelder, A. Kraiss, and T.C. Rakow, “A Client-Controlled Adaptation Framework for Multimedia Database Systems,” in R. Steinmetz and L.C. Wolf, editors, Proc. IDMS'97, pp 397-409, Darmstadt, Germany, September 10-12, Springer 1997.
37. JINI (TM): Http: //java.sun.com/product/JINI/. 1998.
38. E. Guttman, C. Perkins, J. Veizades, and M. Day, “Service Location Protocol. Version 2.,” http://ietf.org/rfc/rfc2608.txt.
39. W3C. WSDL: Web Service Description Language, http://www.w3.org/TR/wsdl, 2002.
40. 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.
41. Jini Technology Core Platform Specification (Version 2.0 Beta), Sun Microsystems, Februray 2003
42. UDDI: http:\\www.uddi.org