A QoS-Based Framework for Distributed Content Adaptation

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 tremendous growth of the Internet has introduced a number of interoperability problems for distributed multimedia applications. These problems are related to the heterogeneity of client devices, network connectivity, content formats, and user's preferences. The purpose of this paper is to present a framework for trans-coding multimedia streams. The proposed infrastructure takes into consideration the profile of communicating devices, network connectivity, exchanged content format, context description, and available customization services to find a chain of services that could be applied to adapt the content to the required needed format. Part of the framework is a QoS-based selection algorithm that finds the best sequence of adaptation services which can maximize users' satisfaction with the delivered content.

1. Introduction

Diversity and heterogeneity of Internet clients is a major problem for multimedia content 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. Content adaptation [1,2,3,4,5] 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 and 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 [1,6]. To solve this problem, some transcoders have been implemented in hardware and can be deployed on intermediate nodes or proxies [7]. 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 [8], and that multiple transcoders can be chained effectively together to perform a complex trans-coding task. 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 these transcoders 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 transcoders between the sender and the receiver, and also to select the configuration parameters for each transcoder. The proposed algorithm uses the user's satisfaction with the quality of the trans-coded content as the optimization metric for the selection function.

Proceedings of the First International Conference on Quality of Service in Heterogeneous Wired/Wireless Networks (QSHINE'04)

The rest of the paper is organized as follows: Section 2 lists all the required elements for providing customized content adaptation. In Section 3, we present our methodology for using the required element from Section 2 to construct a graph of transcoders; the algorithm for selecting the chain of transcoders as well as the selection criterion for the algorithm are introduced in Section 3. Our conclusion is presented in Section 4.

2. Required elements for content adaptation

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 [9] 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.

Content profile: Multimedia content might enclose different media types, such as audio, video, text, and each type can have different formats [3]. 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, included in the content profile. Some of this meta-data may include:

- 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).
- Available variants of the content, such as colored-and-black and white variants,
- The author and production of the content, such as the title, and date of creation.
- The usage of the content, such as copyright, application adaptations, and usage history.

Context profile: According to [10] and [11], 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.

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, and display size. 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.

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 [12]. Failing to meet these requirements may lead to a bad experience of the user [13,14]. 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 [15]. 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.

Profile of intermediaries: When the content is delivered to the user across the network, it usually travels through a number of intermediaries. These intermediaries have been traditionally used to apply some added-value services, including on-the-fly content adaptations services [16,17,18]. Using intermediaries for applying adaptations alleviates the problem of clients with limited-resources [19] and overloaded content servers [20].

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 [21], SLP [22], or WSDL [23]. 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.

3. QoS selection algorithm

In this section, we will describe the overall QoS selection algorithm that finds the most appropriate path of transcoders between the sender and the receiver, and also selects the configuration for each transcoder. 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 transcoders. After constructing the graph, we will present the actual QoS path and parameter selection algorithm.

3.1. 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. As we mentioned earlier, the user's preferences expressed in the user's profile can be classified as application layer QoS parameters.

To compute the user's satisfaction with all values of the application layer configuration parameters, we have used the approach presented by Richards et. al. in [24], 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.

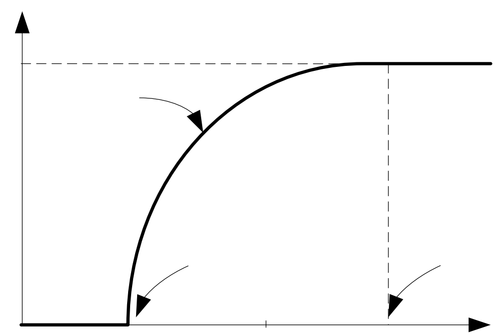

Figure 1. Possible satisfaction function for the frame rate

For applications that have more than one QoS 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:

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

3.2. Constructing directed graph

Now that we have decided on the selection criteria, the first step of the QoS selection algorithm is to construct a directed acyclic graph for adaptation, using the content profile, device profile, and the list of available transcoders. 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 adapted content. The elements of the directed graph are the following:

1. Vertices in the graph represent intermediate transcoders or adaptation service. Each vertex has a number of input and output links. The input links to the vertex represent the possible input formats to the transcoder. The output links are the output formats of the transcoder. Figure 2 shows a transcoder 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.

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 output links of the sender with all the input links 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.

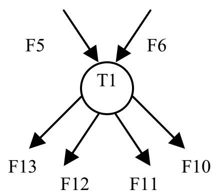

Figure 2. Transcoder with multiple input and output links

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

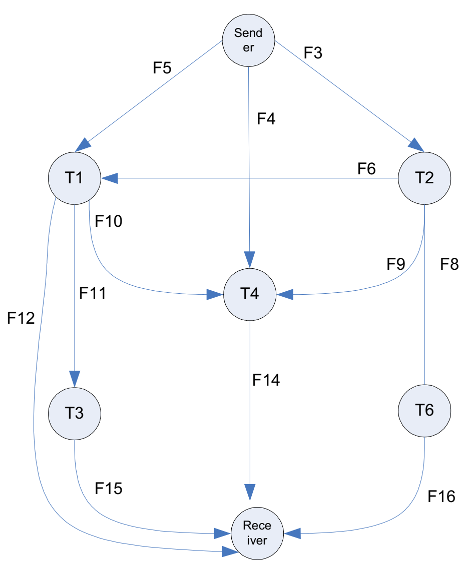

Figure 3. Directed trans-coding graph

3.3. 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 3.1. The maximum satisfaction achieved by using a transcoder T_i depends actually on a number of factors: The first factor is the bandwidth available for the data generated by the transcoder T_i. The more bandwidth is available to the transcoder, the more likely the transcoder will be able to generate trans-coded content that is more appreciated by the receiver. The available bandwidth between two transcoders is restricted by the amount of bandwidth available between the intermediate server where the transcoder T_i is running and the intermediate server where the next transcoder or receiver is running. We can assume that connected transcoders 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 transcoder.

3.4. Route 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 transcoders, starting from the sender node and ending with the receiver node, that generates the maximum satisfaction of the receiver.

The QoS selection algorithm presented here uses two sets of transcoders: the set of already considered transcoders, called VT, and the set of candidate transcoders, called CS, which can be added next on the partially selected path. The candidate transcoders set contains the transcoders that have input edges coming from any transcoder in the set VT.

At the beginning of the algorithm, the set VT contains only the sender node *S*, while CS contains all the other transcoders in the graph that are connected to the sender node *S*, as well as the receiver node *R*. At each step of the protocol, the satisfaction of the user is evaluated for adding each of the transcoders in the CS set, and the transcoder T_i that generates the highest satisfaction is selected and added to VT. The CS set is then updated with all the neighbor transcoders of T_i. The algorithm stops when the CS set is empty, or when the receiver node *R* 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 transcoders. Let CS be the set of all downstream neighbors of *Sender*.
Step 2: If CS is empty, then **TERMINATE**(FAILURE)
Step 3: Compute the perceived user's satisfaction for all the transcoders in CS.
Step 4: Select the transcoder T_i that has the highest satisfaction value.
Step 5: If the selected transcoder T_i is the *Receiver* node, then **GOTO** Step 8.
Step 6: Add to CS all the transcoders to which T_i is directly connected.
Step 7: **GOTO** Step 2
Step 8: Print path from the *Sender* to T_i

When the algorithm terminates, the algorithm would have computed the best path of transcoders from the sender node *S* to the receiver node *R*. The user's satisfaction value computed on the last edge to the receiver node is the maximum value the user can achieve.

4. Summary

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.

We have already coded the algorithm, and we are currently integrating it into a prototype in our Mobile Internet Telecommunication (MobInTel) [25] architecture. Performance results will be published in a future paper.

Acknowledgments

The authors acknowledge the financial assistance of the Natural Sciences and Engineering Research Council Canada (NSERC) and of the National Capital Institute of Telecommunications (NCIT).

Reference

1. 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.
2. 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.
3. 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
4. S. Björk, L.E. Holmquist, J. Redström, I. Bretan, R. Danielsson, J. Karlgren, and K. Franzén, "WEST: a Web browser for small terminals", *Proc. of the 12th annual ACM symposium on User interface software and technology*, p.187-196, November 07-10, 1999, Asheville, North Carolina, United States
5. B. Fisher, G. Agelidis, J. Dill, P. Tan, G. Collaud, and C. Jones, "CZWeb: Fish-Eye Views for Visualizing the World-Wide Web", *Proceedings. of the 7th Int. Conf. on Human-Computer Interaction (HCI International '97),* 719-722, 1997.
6. 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
7. Cisco CallManager. http://www.cisco.ca
8. J. R. Smith, R. Mohan, C.-S. Li, "Scalable Multimedia Delivery for Pervasive Computing", *ACM Multimedia '99*, Oct. 30 - Nov. 5, 1999.
9. K. El-Khatib, and G.v. Bochmann, "Profiles in Content Adaptation", Technical report. University of Ottawa, Canada. Dec 2003.
10. A.K. Dey, "Providing architectural Support for Building Context-Aware Applications", PhD thesis, College of Computing, Georgia Institute of Technology, December 2000.
11. 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.
12. 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.
13. M. Katchabaw, H. Lutfiyya, and M. Bauer, "Driving resource management with application-level quality of service specifications", *ICE 98, ACM Press* (1998), 83-91.
14. C. Poellabauer, H. Abbasi, and K. Schwan, "Cooperative run-time management of adaptive applications and distributed resources", *Proc. Multimedia '02*, ACM Press (2002).
15. 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.
16. 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.
17. R. Floyd and B. Housel, "Mobile Web Access Using eNetwork Web Express", *IEEE Personal Communications*, 5(5):47–52, 1998.
18. 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.
19. S. Hollfelder, A. Kraiss, and T.C. Rakow, "A Client-Controlled Adaptation Framework for Multimedia Database Systems", in *Proc. IDMS'97*, pp 397-409, Darmstadt, Germany, September 10-12, Springer 1997.
20. 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.
21. JINI (TM): Http: //java.sun.com/product/JINI/. 1998.
22. E. Guttman, C. Perkins, J. Veizades, and M. Day, "Service Location Protocol. Version 2.", http://ietf.org/rfc/rfc2608.txt.
23. W3C. WSDL: Web Service Description Language, http://www.w3.org/TR/wsdl, 2002.
24. 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.
25. K. El-Khatib, X. He, and G.v. Bochmann, "A communication services infrastructure including home directory agents", Technical report. U. of Ottawa, Canada. May 2000.