ADAPTIVE CONTENT DELIVERY: A NEW APPLICATION AREA FOR MEDIA COMPUTING RESEARCH

Hongjiang Zhang

Microsoft Research, China
No.5 Zhichun Road, Haidian District
Beijing 100084, China

ABSTRACT

The explosive growth of the Internet has resulted in increasing diversity and heterogeneity in terms of client device capability, network bandwidth, and user preferences. To date, most Web content has been designed with desktop computers in mind, and often contains rich media such as images, audio, and video. In many cases, this content is not suitable for devices like WebTVs, personal digital assistants, and smart phones with relatively limited display capability, storage, processing power, and network access. To support universal access and ubiquitous browsing, there is a need to develop alternative approaches for information delivery. In this paper, we review a framework for adaptive content delivery in heterogeneous environments, aiming at improving content accessibility under changing network and viewer conditions. The framework includes content adaptation schemes, client capability and network bandwidth discovery methods, and a Decision Engine for adaptation selection. We also address research issues in the deployment of such systems technologies.

1. INTRODUCTION

The explosive growth of the Internet and the increasing amount of online electronic information have made the Internet an important information source for people today. However, the expansion of the Internet also comes with increasing diversity and heterogeneity in terms of the types of client devices and network connections that people use to access the Web, as well as the special needs and preferences that end users might have. In addition to traditional desktop computers, many new devices are being used to access the Internet, such as handheld computers, personal digital assistants (PDAs), set-top boxes, and smart phones. Network connections are also highly diverse, ranging from low bandwidth (9.6-28.8 kbs/sec), such as in cellular and wire-line modems, to middle bandwidth (128 Kbs/sec-1.5 Mbs/sec), such as with ISDN, DSL, and cable modems, to high bandwidth (10-100 Mbs/sec), such as with local Ethernet. On the server side, more and more Web sites provide rich multimedia content, an integration of text, images, audio, video, graphics, and applications, without differentiated services to address the limitations of individual client devices and network access. As a result, users of devices with limited capabilities or slow connections experience frustration due to slow content delivery or inability to view certain media types. The lack of Internet infrastructure to accommodate this growing heterogeneity raises challenging research issues for enabling effective and ubiquitous information access over the Internet.

Adaptive content delivery is a system technology that transforms Web content and delivery schemes according to viewers' heterogeneous and changing conditions to enable universal access. The goal of adaptive content delivery is to take into account these heterogeneous and changing conditions and provide the best information accessibility and perceived quality of service over the Internet. Ultimately, adaptive content delivery aims at *Universal access* to multimedia information in a heterogeneous network environment, by accommodating the special needs of users and the constraints of client devices and network characteristics. In other word, the adaptive content delivery effort is to provide the necessary Internet infrastructure to allow users to *access any information over any network from anywhere through any type of client device* [28].

Adaptive content delivery has beneficial business implications beyond just reaching a wider audience for Web content. One of the main benefits is to decrease the Web access time for users. In a user survey conducted by Georgia Technical Institute's Graphics, Visualization, and Usability Group, 53% of respondents reported that they had left a web site while searching for product information simply because the site was too slow [27]. The improved perceived quality of service by adaptive content delivery means that shoppers are more likely to stay and return, thus resulting in a greater profit for e-commerce sites. Adapting content to have more aesthetic appearance on the user device or allowing the user to have wider access may encourage the user to appreciate the site more. This can also result in higher hit rates and return rates, implying higher sales for e-commerce sites and higher advertising revenues.

In order to provide adaptive content delivery over heterogeneous network environments, many technologies from different aspects of the delivery environment need to be developed and integrated. These technologies include

- Media processing and analysis algorithms to support content adaptation.
- A set of mechanisms for reliably detecting the software and hardware capabilities of a client device.
- A way to effectively measure the characteristics of the current network connection between a client and a server.
- A standard approach for defining user preferences and a mechanism for tracking them from session to session.
- Decision rules on when and how to perform a particular content adaptation process based on various conditions.

In this paper, we present an overview of research issues in developing an adaptive content delivery systems and technologies, mainly based on the work in [9]. We focus on the general framework of adaptive content delivery systems, some key technologies to realize such framework, and related research issues, each will be discussed in more detail in the following sections.

2. CURRENT STATUS

Numerous companies, academic communities, and standards organizations have recognized the issues for delivering content under heterogeneous clients and network conditions. Examples of commerical products and research prototypes in this area include Spyglass [22], ProxiNet [24], Intel QuickWeb [23], OnLineAnywhere [21], IBM Transcoding proxy [12], TranSend [6], Digestor [4], Mobiware [2], Smart Client [14], and Odyssey [11]. They usually design their systems only for narrow needs. The types of content adaptation they looked into are mostly image-centric transformation. In contrast, our framework is developed to provide a broad range of Web content adaptation for all different types of devices under heterogeneous and changing network conditions.

The W3C and the IETF have existing standards and on-going discussions on facilitating server/proxy decision making on the mechanisms of content adaptation and content delivery. Most of these protocols are new Web techniques that have yet to gain the recognition of their potential in facilitating Web content delivery. One notable success is the Synchronized Multimedia Integration Language (SMIL) [5][19]. SMIL is a markup language that enables the synchronized delivery of multiple video streams, audio streams, and images. It provides conditional constructs to switch tasks (e.g. request different content) based on bandwidth conditions. The Extensible Markup Language (XML) [18] describes the logical representation of data and can be utilized to facilitate serving content to different types of clients under heterogeneous network conditions. The logical representation of data can be converted into an appropriate representation for display using the Extensible Style Sheet Language (XSL) [17].

The HTTP/1.1 content negotiation capability [25] and the CC/PP [15] are mechanisms for the client to convey along with its request its preferred version of content and its user agent information. In HTTP/1.1 content negotiation, a user agent can specify in the HTTP header that, for example, English documents are preferred over French, or that JPEG images are preferred over GIF images. CC/PP specifies client capabilities and user preferences as a collection of URIs and RDF (Resource Description Framework) text [26], which is sent by the client along with a HTTP request. The URIs point to an RDF document which have the details of the clients capabilities. The RDF text can be used to provide additional details that the referenced RDF documents do not provide. RDF provides a way to express "metadata" for a Web document. The CC/PP scheme allows proxies and servers to collect information about the client, from the client directly, and to make decisions based on this information for content adaptation and delivery. If CC/PP becomes widely deployed it holds great promise for adaptive content systems.

3. ADAPTATION FRAMEWORK AND ARCHITECTURE

A typical framework for our adaptive content delivery system is shown in Figure 1. It consists of three main modules --

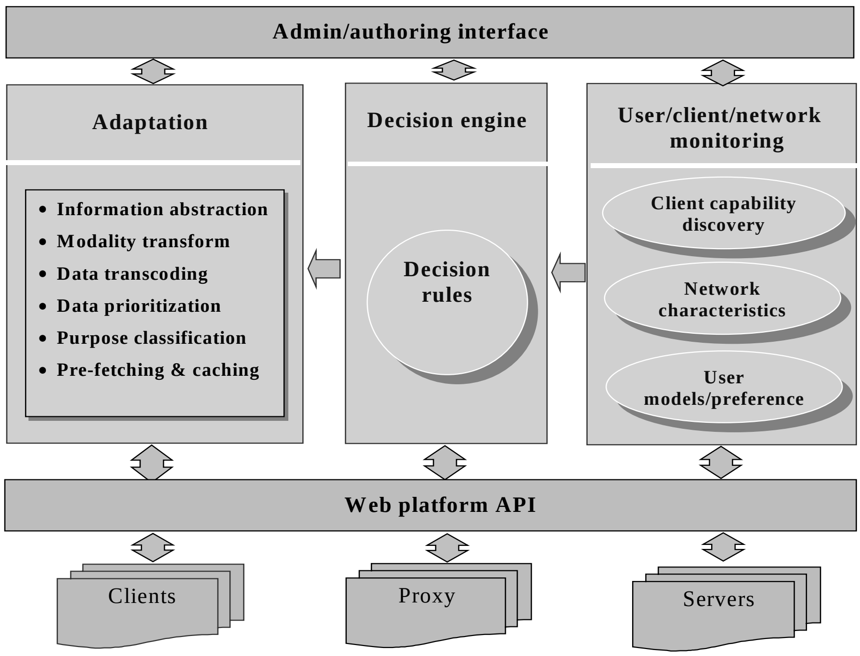

Figure 1. The basic framework of our adaptive content delivery system. It consists of three main modules including user/client/network-monitoring, decision engine, and content adaptation modules along with interfaces to administration and authoring tools and Internet

user/client/network-discovery, decision engine, and content adaptation algorithm modules -- along with interfaces to administration and authoring tools and system devices such as client, proxy, and server. The user/client/network-discovery module detects and collects all necessary information that the Decision Engine needs to know in order to dispatch a particular content adaptation scheme. In this framework, the Decision Engine is the center of the system. Depending upon where the Decision Engine and the content adaptation algorithms are located, we can classify the system into one of the following architectures: server-based adaptation, proxy-based adaptation, or combinations thereof. Each design choice has its advantages and drawbacks. The main issues to consider include how deployable is the system, how efficient is the utilization of the bandwidth, CPU, and memory storage, how effective is the content adaptation, as well as the related copyright issues for content transformation.

3.1 Server-based Adaptation

In a server-based architecture, the server is responsible for discovering what the client capabilities are and how much effective bandwidth is available. It then determines the best adaptation strategy. Using the server-based architecture has the advantage that it allows both static (off-line) and dynamic (on-the-fly) content adaptation. The former refers to automatically creating multiple versions on the authored content, at anytime after the content has been created; the latter refers to performing an on-the-fly adaptation as each request comes in. The server architecture provides more author control since the adaptation can be tied to the content authoring process, allowing the author to provide hints on the adaptations for different circumstances. For a secured environment, such as in e-commerce applications, pages are usually encrypted, in which case only the server can perform adaptation. However, the adaptation results in additional computational load and resource consumption on the server; thus the server design needs to take into account load balancing as well. If a static adaptation approach is used to generates multiple versions of the content, it will make content management more cumbersome and requiring more storage.

3.2 Proxy-based Adaptation

In a proxy-based architecture, the client connects through a proxy that makes the request to the server on behalf of the client. The proxy intercepts the reply from the server, decides and performs the adaptations, and then sends the adapted content to the client. It is usually true that the bandwidth between the proxy and the server is much higher than that between the client and the proxy, thus the time to download the original content from the server to the proxy is negligible. This is the cases when the proxy resides at the point-of-presence (POP) of an Internet Service Providers, and when a client is connecting via a slow modem.

Adapting on the proxy means that there is no need to change existing clients and servers. The proxy can transform existing Web content so that existing content does not have to be re-authored. Because a proxy takes the content from many servers, there are many documents with widely varying appearances, created with many different authoring tools. Hence, in the proxy-based adaptation architecture, there is less author control on the outcome of the adaptation, and it is difficult to determine what alteration "looks good" for any general content. Thus the author of the Web page may find the resulting transformation objectionable, and it may be difficult to reliably ensure that a transformed page will look aesthetically pleasing. For secured or proprietarily encoded content, the organization deploying the proxy will need to coordinate with the service or content provider in order to access the content for performing adaptation.

Several commercial proxies have been deployed for the purpose of adapting Web content for universal access [22][24]. Because of these difficult issues, the commercial adaptation proxies have either targeted specific clients and usage environments, or have partnered with portal and content provider companies. Partnering with content providers also circumvents the copyright issue.

4. CONTENT ADAPTATION TECHNIQUES

In order to increase content accessibility and improve the user's experience within a heterogeneous network environment, many media processing technologies can be used to enable more intelligent information delivery. Several existing content adaptation systems apply image-processing techniques to adapt the inline images of a Web page according to characteristics of the client display, such as screen size or color depth [6][12][23][24]. This is only a small subset of technologies that will find application in this increasingly important area. In Table 1, all the content adaptation technologies useful for adaptive content delivery are listed and classified into five categories, based on their applications: Information abstraction; Modality transformation; Data transcoding; Data prioritization; Purpose classification; and Form factor adaptation. This categorization is useful in developing a general decision-making framework for optimizing adaptive content delivery over the Internet. As the number of input parameters and possible output actions becomes large, the problem of developing an efficient and effective decision framework can become quite complex.

4.1 Information Abstraction

The goal of information abstraction is to reduce the bandwidth requirement for delivering the content by compressing the data, while preserving the information that has the highest value to the user. Examples of information extraction include text summarization, image thumbnail generation, and video highlighting and key-frame extraction. Such algorithms can also be used to improve the user's browsing experience by providing a preview of the content. In this way, users are able to quickly browse though more information even though their network bandwidth is constrained. If users feel that some information is interesting or critical to them, they can decide to download the full content at that time. Moreover, information abstraction can be very useful when the client device has limited display capability, such as on palmtops and smart phones. For example, summarizing each paragraph by a few words and shrinking the size/resolution of each image in a Web page will help to fit this page on the small screens of those devices.

4.2 Modality Transform

Modality transform is the process of transforming content from one mode to another so that the content can become useful for a particular client device. For instance, most handheld computers are not capable of handling video data because of both hardware and software constraints. In order to make the information contained in the video accessible on these devices, we can transform video into sets of images, extracted audio or closed caption text. In this way, users will be able to receive useful information in whatever form that their devices can handle. Other examples of modality transform include speech-to-text and text-to-speech transform and table-to-plain-text or table-to-list transform for HTML. The primary goal of modality transform is to adapt the content representation to client device capabilities. In some cases it may even reduce data volume and, thus save bandwidth in delivery.

	Video	Image	Audio	Text
Purpose Classification	• Removal • Substitute	• Removal • Substitute	• Removal • Substitute	• Removal • Substitute
Information Abstraction	• Video highlight • Video frame rate reduction • Video resolution reduction • Keyframe extraction	• Image Dimension reduction • Data size reduction (by increasing compression rate)	• Audio highlight • Audio sub-sampling • Stereo-to-mono conversion	• Text summarization • Outlining • Font size reduction • Text white space removal
Modality Transform	• Video-to-image • Video-to-audio • Video-to-text • Removal	• Image-to-text • Removal	• Audio-to-text • Removal	• Text-to-audio • Removal • Table-to-list • Table-to-plain text • Language translation
Data Transcoding	• Format conversion • Color-depth reduction	• Format conversion • Color-depth reduction • Color-to-grayscale	• Format conversion	• Format conversion
Data Priroitization	• Layered coding • Frame prioritization • Audio prior to video	• Multi-resolution image compression	• Audio prior to video	• Text prior to image/audio/video

Table 1: A list of content adaptation technologies that are classified into five categories based on their applications. This classification helps to develop a general decision-making framework for adaptive content delivery over the Internet.

4.3 Data Transcoding

Data transcoding is the process of converting data format according to client device capability. For example, some client devices may not be able to display color GIF images due to the lack of viewing or rendering software or the constraint of hardware display capability, such as a black-and-white screen. In such cases, there is a need to transcode the original images into another appropriate format, such as GIF-to-JPEG or color-to-grayscale transformation, so that they can be viewed on the client device. Other examples of data transcoding include video format conversion (such as MPEG-to-QuickTime), audio format conversion (such as WAV-to-MP3), and document format conversion (such as Postscript-to-PDF).

4.4 Data Prioritization

The goal of data prioritization is to distinguish the more important part of the data from the less important part so that different quality of service levels can be provided when delivering the data through the network. For example, we can allow less important data to be dropped under network bandwidth constraints. Or, we can provide progressive delivery to send out the more important data first (such as low-resolution images) and then deliver the less important data to enhance the information later (such as the reconstruction of high-resolution images). In this way, we can improve the user's browsing experience by efficiently utilizing available network bandwidth. Data prioritization can be achieved within a single media type by using special encoding schemes such as layered coding [9][13] and multi-resolution compression for images [20]. It can also be done across multiple media types by, for example, giving audio higher priority than video and text higher priority than other types of media.

Video content abstraction can be considered a type of data abstraction, modality transform, or transcoding processes. Key-frame extraction can invoked when the client does not have a video playing capability and the user prefers to receive an image version of the video (i.e., video-to-image modality transform). A parameter indicating the degree of aggressiveness in selecting key-frames can be used to control the number of images generated in the process. This parameter can be determined by the decision engine based upon the client's bandwidth and video support features, as well as users' preferences.

4.5 Purpose Classification

A typical Web page contains a lot of information and media objects that are redundant or may not be of interest to a user. For example, an e-commerce web site may have multiple images for linking to the same product site on the top, bottom and the side of the page. A portal site usually contains many images of banners, logos, and advertisements. These data often consume a good deal of network bandwidth and, therefore, decrease the efficiency of information delivery. If we can classify the purpose of each media object in a Web page, we can improve the efficiency of information delivery by either removing redundant objects (assuming the related copyright issues have been properly addressed) or prioritizing them according to their importance.

Purpose classification of a media object can be done using content analysis techniques. It can also be achieved to some extent by matching URL strings with a pre-established database or via heuristics for associating meanings with certain text contained in the URLs. For example, advertisement images can be detected and blocked by matching URL strings with a list of keywords like "ad", "banner", "advertisement", "promotion", or a list of known advertising web hosts. Objects with names or "alt" tags containing "bullet" and "logo" are deemed less important, or even redundant. We can easily use this hint to classify them.

4.6 Form factor adaptation

Form factor adaptation aims at re-arranging Web page layout adaptively based on browser devices form factors such that the Web content will be most perceivable for the user using a particular browser. We can define a number of form factor adaptation rules for HTML document in an adaptive content system. These include:

- Minimizing white space on small displays by collapsing sequences of paragraphs into single text blocks.
- Optimizing text layout on small displays by utilizing text summarization algorithms. Such algorithms can include keyword-driven text synopsis as well as coarser measures, based upon "speed reading" heuristics.

Removing from a Web page those image links that are determined to be redundant. For example, if an image is associated with a navigation link and that navigation link appears again as a textual anchor within the same page, then that image is considered to be redundant in that it does not add any additional information to the Web page. Eliminating such images not only reduces the size of the target Web page, but also saves on the computational overhead of component-level adaptation such as scaling and color down-sampling.

More sophisticated form factor adaptation will require more studies in user preference modeling and interface design.

5. DECISION ENGINE

Figure 2 shows a typical decision engine. The inputs to the decision engine are a set of media objects contained in a document to be downloaded and the information about their content types, content lengths, and purposes of usage in a Web page.

Document Object is useful data structure to represent the logical structure and contents of a Web page after it is parsed in content adaptation systems. This defines the logical structure and contents of the Web page that is to be adapted. The XML Document Object Model (DOM) [16] technology can be used to represent the represent the hierarchical structure of a Web page, with node elements representing entities such as text, images, audio, and video.

By using a Document Object Model to represent Web page structure, we can to perform a number of document-level adaptations to the Web page. This complements the component-level adaptations that we implement, such as image scaling and video modality transform. By analyzing the contents of the Web page as a whole, we can optimize the adaptation process by identifying redundant information that can be removed from the Web page without overly compromising the informational integrity of the Web page. This page-level adaptation can be considered to be a "re-authoring" of the content [4].

Once the structure of and the objects in the document to be down loaded are extracted, the decision engine should first check the user preferences to see if it needs to remove or substitute any redundant objects in order to save bandwidth. Other information for making decisions, such as how much a user is willing to trade off image quality for download time, is acquired by the decision engine at this time. The decision engine further checks the client capability and the network characteristics of the client. Based upon the collected information, the decision engine then determines if it needs to launch a particular content adaptation algorithm for a particular object.

One can separate the decision-making process into three stages. The first stage involves a binary decision that controls the dispatch of modality transform and data transcoding. Since these two types of schemes are usually performed to enable the viewing of media objects whose original representation cannot be handled by the client, the resulting decision is either "yes" (i.e., perform the transformation) or "no" (i.e., do not perform it).

The second stage of decision-making involves a careful examination of various parameters to find out the best trade-off between information abstraction and download time. These parameters should include the user specified preference in terms of quality and response time, the current network bandwidth between the client and the server, the estimated processing time for conducting information abstraction, and the predicted output data size. The goal is to determine when it is beneficial to perform information abstraction and how much abstraction (compression) should be done. In [8], a scheme for making such a decision in an image transcoding proxy has been proposed. In [9], a similar scheme was adapted and extended to other types of media.

The final stage of the decision-making involves data prioritization. At this stage the decisions for transforming all media objects into appropriate representations have been made and the server now needs to decide how to arrange them in the delivery pipeline according to their importance. Simple rules, such as text before image, image before audio, and audio before video, can be used to prioritize different media objects. Within the same media object, if an encoding scheme such as layered coding or multi-resolution image compression is supported, we can also prioritize the data accordingly. For a more advanced application, we can incorporate the user's real-time interaction or feedback to prioritize the data on the fly. The method proposed in [7] for progressive interactive web delivery can be a natural mechanism to be performed at this stage.

6. DISCOVERING CLIENT, USER PREFERENCES, AND NETWORK CHARACTERISTICS

Many methods can be used to measure and collect the information about client capabilities, user preferences, and network characteristics. The following is a list of methods that can be used:

- **Analyzing the request**: The HTTP request header [3] contains useful information about the client device. For example, some browsers (e.g. IE 3.0) directly send the screen size information in a header field called *UA-pixels*. We can also infer useful information indirectly from some header strings. One can also guess that the bottleneck bandwidth is 20kps since this is the approximate maximum throughput of a palm size computer serial cable. This method works for Palms, WebTV (544x200 screen, 56kps modem), and other specific platforms.

  There exist established and on-going standards that facilitate adaptive content delivery. The HTTP/1.1 content negotiation capability [25] provides a mechanism to specify in the HTTP headers what content versions the client prefers. The World Wide Web Consortium (W3C) is developing a standard for the process of discovering client capability and user preferences. This standard is called Composite Capability/Preference Profiles (CC/PP) [15]. It is a mechanism that allows the client to describe the capabilities and preferences associated with its user and user agent. This information includes the hardware platform, system software, applications and user preferences.

- **Tracking sessions**: We can use a hash of the *UserAgent* string, IP Address, and an optional cookie from the client request to establish an ID for the user. All states about a session are maintained with this ID.

- **Automatic measurement**: At the server, measurements can be actively performed to discover information about system load [1] and network bandwidth.

- **User advice**: A user interface can be provided for the user to submit information about his/her preferences, device capability, and the type of network connection that he/she uses. For example, the client may not have a video player (as with many PDA's) or may not accept certain types of image formats. A user may indicate a preference for summarized text instead of a long document; thus only the requested portion needs to be sent. A user can also indicate that the sound on his/her device has been turned off, so either the audio data needs not be sent, or a speech-to-text conversion could be used. These preferences can be submitted to the server through a server-side interface such as a Web page with forms.

- **Analyzing user browsing behavior**: Log analysis over time can give an indication of user preferences. For example, if connections keep ending prematurely, this may be an indication that bandwidth is low, and the adaptive system should adapt more aggressively.

7. CONCLUSIONS

In this paper, we have reviewed a typical framework for adaptive content delivery in heterogeneous network environments has been presented. To fully support adaptive content delivery, we need to develop a set of system technologies including the discovery modules for detecting client capabilities, user preferences and network characteristics, various content adaptation techniques to improve web accessibility and information delivery, and a general decision-making framework for optimizing adaptive content delivery over the Internet. Developing of all of these technologies poses many research challenges, and is a promising new area networking and multimedia research.

8. ACKNOWLEDGEMENTS

The overview presented in this paper is mainly based on the work presented in [9].

9. REFERENCES

[1] Tarek F. Abdelzaher and N. Bhatti, "Web content adaptation to improve server overload behavior," Proceedings of the 8th International World Wide Web Conference, pp. 485-499, May 1999.

[2] O. Angin, A.T. Campbell, M. E. Kounavis, and R. R.-F. Liao, "The Mobiware Toolkit: Programmable support for adaptive mobile networking," IEEE Personal Communications, Vol. 5, No. 4, August 1998, pp. 32-43.

[3] T. Berners-Lee, R. Fielding, H. Frystyk, “Hypertext Transfer Protocol -- HTTP/1.0” RFC 1945

[4] T. Bickmore and B. Schilit, "Digestor: Device Independent Access to the World Wide Web", proceedings of the Sixth International World Wide Web Conference, Santa Clara, California, 1999.

[5] L. Bouthillier, "Synchronized multimedia on the Web," Web Techniques Magazine, 3(9), Sep. 1998.

[6] A. Fox, S. D. Gribble, Y. Chawathe, and E. A. Brewer, "Adapting to network and client variation using active proxies: lessons and perspectives," IEEE Personal Communication, Vol. 5, No. 4, pp. 10-19, August 1998.

[7] J. M. Gilbert and R. W. Brodersen, "Globally progressive interactive web delivery," Proceedings of IEEE Infocom 99, New York, March 1999.

[8] R. Han, P. Bhagwat, R. Lamaire, T. Mummert, V. Perret, and J. Rubas, "Dynamic adaptation in an image transcoding proxy for mobile web browsing," IEEE Personal Communications, pp. 8-17, December 1998.

[9] W. Y. Ma, et al., “A framework for adaptive content delivery in heterogeneous network environments,” Proc. Of SPIE/ACM Conference on Multimedia Computing and Networking, San Jose, Feb. 2000.

[10] S. McCanne, M. Vetterli, and V. Jacobson, "Low-complexity video coding for receiver-driven layered multicast," IEEE JSAC, vol. 16, no. 6. Pp. 983-1001, Aug. 1997.

[11] B. D. Noble, M. Satyanarayanan, D. Narayanan, J. E. Tilton, J. Flinn, and K. R. Walker, "Agile application-aware adaptation for mobility," Proceedings of the 16th ACM Symposium on Operating System Principles.

[12] J. Smith, R. Mohan, and C. Li, "Scalable multimedia delivery for pervasive computing," ACM Multimedia, 1999.

[13] D. Taubman and A. Zakhor, "Multi-rate 3D subband coding of video," IEEE Trans. on Image Processing, vol. 3, no. 5, pp.572-540, Sep. 1994.

[14] C. Yoshikawa et al, "Using Smart Clients to Build Scalable Services", Proc. Winter 1997 USENIX Tech. Conf., January 1997.

[15] W3C Composite Capability/Preference Profiles (CC/PP). http://www.w3.org/TR/NOTE-CCPP/.

[16] W3C Document Object Model Level-1 Recommendation. http://www.w3.org/TR/REC-DOM-Level-1/.

[17] W3C Extensible Stylesheet Language (XSL) http://www.w3.org/TR/WD-xsl/.

[18] W3C Extensible Markup Language (XML) http://www.w3.org/TR/REC-xml.

[19] W3C Synchronized Multimedia Integration Language http://www.w3.org/TR/REC-smil/.

[20] HP OpenPix. http://image.hp.com.

[21] OnLineAnywhere. http://www.onlineanywhere.com.

[22] Spyglass-Prism. http://www.spyglass.com.

[23] Intel QuickWeb. http://www.intel.com/quickweb.

[24] ProxiNet. http://www.proxinet.com.

[25] IETF Working Group on Content Negotiation. http://www.ietf.org/html.charters/conneg-charter.html

[26] W3C Resource Description Framework (RDF) Schema Specification. http://www.w3.org/TR/PR-rdf-schema/.

[27] Geogia Tech Graphic, Visualization, & Usability Center's (GVU) 9th WWW User Survey, April, 1998, http://www.cc.gatech.edu/gvu/user_surveys/survey-1998-04/

[28] World Wide Web Consortium, Web Content Accessibility Guidelines 1.0, http://www.w3.org/TR/WAI-WEBCONTENT/.