TOWARDS ADAPTIVE DISTRIBUTION OF MULTIMEDIA CONTENT WITHIN COLLABORATIVE CONFERENCING SESSIONS

Daniel Schuster
Technische Universität Dresden
Chair for Computer Networks
01062 Dresden
Germany
schuster@rn.inf.tu-dresden.de

Sascha Kümmel
VidSoft GmbH
Lortzingstr. 35
01307 Dresden
Germany
kuemmel@vidsoft.de

ABSTRACT
This paper proposes a generic architecture and protocol for adaptive distribution of multimedia content within collaborative conferencing sessions, supplying applications like presentation sharing, co-browsing, or application sharing with extensive content.

Applications building upon it are able to use efficient decoupled content delivery with adaptation in a tightly-coupled environment while integrated interaction features like annotations, telepointer, and chat are still available. Building blocks from content networking research are combined with concepts of IP conferencing. A system for presentation sharing and photo sharing was implemented as a proof of concept.

KEY WORDS
Multimedia Conferencing, Content Distribution, CSCW, Shared Whiteboards

1 Introduction

Architectures, protocols, and compression techniques for IP-based conferencing have been studied extensively in the last years [1]. The developed solutions have to bridge the gap between high user demands and strong technical limitations, especially the decentralized structure of the Internet, which was originally not designed for real-time communication. Nevertheless, there have been remarkable efforts that incorporated in a number of standards that are usually gathered under the signaling protocols H.323 [2] of the ITU-T and the Session Initiation Protocol (SIP) [3] of the IETF.

As IP conferencing systems based on these standards families are primarily designed to establish real-time conversational data streams (audio, video, and text), data conferencing applications like shared whiteboard, presentation sharing, document sharing, and application sharing are commonly realized in a separate conferencing architecture with reliable (multipoint) transport.

Besides its original existence as a supplement for conversational audio/video conferencing, data conferencing functionality is also used as the main functionality in groupware applications [4] [5] [6] and commercial WebConferencing systems (WebEx, LiveMeeting) to enable teams to work together in real-time. Efficient exchange of content (documents, presentation slides, binary files) between team members is the essential functionality needed for these systems which is poorly supported by data conferencing frameworks available today.

Additionally, the growth in heterogeneity in both communication endpoints and characteristics of network connections is a second source of motivation for our research. As an example, we can think of attendees with PDAs and a GPRS connection that join a team meeting while on a business trip. A distributed presentation would be slowed down in an unacceptable way if all attendees had to wait until the next full slide is distributed. Hence, it is necessary to have a means of caching, prefetching, and data adaptation that can be used by any data conferencing application.

This leads to the following research question: Is it possible to create a generic service for multimedia content distribution within collaborative conferencing sessions that supports

- propagation of content with meta information by clients,
- n:m distribution of arbitrary media objects,
- adaptation to heterogeneous bandwidths and devices,
- scalability for up to 100 and more participants,
- integration with shared collaborative workspaces,
- integration with existing standards-based conferencing systems?

The requirements distribution and scalability are already met by existing content distribution networks. Adaptation capabilities are also available in a number of systems, especially for Web adaptation. But we are not aware of any system that integrates these features with collaborative conferencing and allows content propagation by the clients.

We survey related work in the problem area and propose a generic content distribution architecture that can be used in standards-based conferencing environments. The solution consists of services for content delivery, decoupling the content delivery process from state replication of collaborative applications. This leads to a hybrid distribution pattern, supporting both push and pull semantics for data exchange. Push semantics should be used for application state synchronization only, while pull semantics is appropriate for multimedia content.

The rest of the paper is organized as follows: Section 2 gives an overview of related work. Architecture and protocols are defined on this basis in Section 3. Implementation aspects are discussed in Section 4. We conclude the paper in Section 5 and discuss future research directions.

2 Related Work

2.1 IP Conferencing Models

The term IP conferencing is widely used for any type of real-time communication over IP networks. This covers IP-based telephony as well as multipoint communication, while only the latter is of interest here.

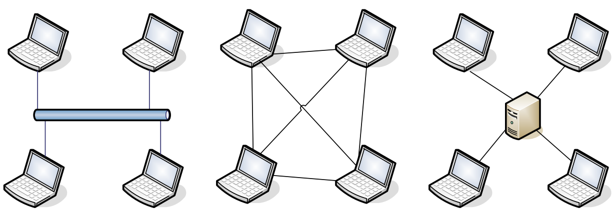

Figure 1. Conferencing Models

Figure 1 shows the different conferencing models in use today. Loosely-coupled conferences rely upon a multicast service for data distribution and feedback. Broadcast of mass data is most efficient with this model, as the routing occurs at the network level. But multicast services are not deployed yet by most internet providers due to a number of problems like lack of access control [7]. Besides this, a multicast solution does not allow individual adaptation, except of hierarchical coding. This is sufficient for continuous media streams but imposes problems for a large number of content objects and individual access that we want to consider here.

In the full mesh conferencing model, each participant initiates a signaling relationship to each other participant in the conferencing session [8]. Adaptive media distribution may be realized in this model, as media is delivered independently to each participant. But full mesh conferencing is restricted to a small number of participants only as it requires $0.5*n(n-1)$ signaling relationships and $n(n-1)$ media channels for n participants.

Tightly-coupled conferences use a central point of control for signaling relationships which is typically placed at a server but can also be realized by one of the clients. Only n signaling relationships are needed for n participants in this model. The central instance enables easy creation of cooperative services and allows for individual adaptation. We also assume centralized media handling for this model, although this is not obligatory as there can be different models for signaling and media within a conference.

Beyond these simple models, conferencing systems may use extended topologies or hybrid models to combine advantages of different basic approaches. Examples are tree topologies (T.120 [9]) or a P2P-network of servers (Jabber [10]). As the tightly-coupled approach offers best adaptation and cooperation opportunities, we will examine it more detailed in the following.

2.2 Existing Solutions for Tightly-Coupled Conferencing

The ITU-T defined the T.120 family of standards [9] as an architecture for tightly-coupled conferencing. T.120 uses a tree-based topology with a single top-node as central instance. This includes a number of topology variants like star or even a cascade of nodes. The architecture consists of a reliable Multipoint Communication Service (MCS, T.122/T.125) [11], a Generic Conference Control component (GCC, T.124), and a set of standardized application protocols including whiteboard (T.126) [12], file transfer (T.127), application sharing (T.128), and chat (T.134/T.140).

T.120 has been proved to have some scalability problems [13] which led to the definition of the Scalable Conferencing Control Service (SCCS) [14]. Despite these issues, T.120 is widely used mainly in commercial systems, but also in some research projects [5].

As the work of IETF conferencing standardization was mainly multicast-based in the 90s, the area of tightly-coupled conferencing is not yet well developed for architectures based on IETF’s Session Initiation Protocol (SIP) [3]. Starting in 2001, the centralized conferencing working group (XCON) is currently developing standards for tightly-coupled conferencing in coordination with MMUSIC and SIPPING working groups.

Although not the main focus, this architecture is likely to be used for data conferencing in the future. Singh [15] proposes a star topology, which is the direction of current protocol development within the standardization group. The upcoming architecture [16] will contain a focus as the central point of session initiation and synchronization, and services for conference control, floor control, conference policy control, and event notification that will have comparable features than the equivalent services within T.120.

2.3 Content Networking

Our research was motivated by various efforts in the well-known area of content networking. We rely on basic techniques from content delivery networks and peer-to-peer networks.

Content delivery networks (CDN) [17] consist of an explicitly controlled network of replica servers which are called surrogates. They are located close to a number of clients and thus are able to reduce request latency and avoid high load on origin servers as content is replicated in advance to client requests. We identify the principles of edge delivery and prefetching with an interconnected network of delivery servers as useful for our requirements.

This inherently includes higher-level caching as in Web proxy caching. Cooperation of multiple caches enables bundling of content requests and responses at certain points in the delivery network. We consider this principle of bundling for the design of our architecture.

The mentioned approaches deal with individual access to a large number of content objects. Besides this n:m content distribution scheme, collaborative applications often need efficient 1:n delivery for shared viewing. Scalability of such broadcast-like mass data delivery can be achieved if receiving network nodes forward received parts of the content to other receiving nodes.

This peer-to-peer technique, which is widely used in file sharing tools such as Gnutella, Freenet, and BitTorrent, is applicable in a lot more areas. Cohen [18] describes the use of the peer-to-peer approach for serving large files. The files are split into pieces of 256 KB that are registered independently by a tracker server and thus can be downloaded and uploaded to other clients.

We propose the application of this forwarding technique for 1:n content distribution to a number of homogeneous nodes. A typical 1:n distribution in a heterogeneous environment can thus be split up in a broadcast part and individual adaptive transfers.

2.4 Adaptation

Research in the field of adaptation has been done in a wide variety of projects mainly in the areas of distributed multimedia systems and mobile/ubiquitous computing. Springer [19] classifies various approaches of context-related adaptation and identifies basic adaptation mechanisms like data adaptation, caching, prefetching, and priorization.

We think of data adaptation primarily in the scenario of presentation sharing, where slides are provided in high resolution by a participant and may be scaled down or compressed for different viewers. But adaptation is not only limited to images. As an example, document adaptation [20] offers great opportunities for document sharing applications.

Data adaptation operations like scaling and transcoding reduce both the amount of data to be transferred over the network as well as the quality perceived by the viewer [21]. As our protocol has to take care about this, we need a language to describe media representations and viewer preferences, and a means of content negotiation to combine both. We choose the Media Feature Sets [22] for this purpose, as they fit our requirements.

Dynamic data adaptation can be done using adaptation paths as introduced by the Ninja project [23]. Distributed adaptation elements are chained to perform multi-staged adaptation as a means of combining basic services to a complex service.

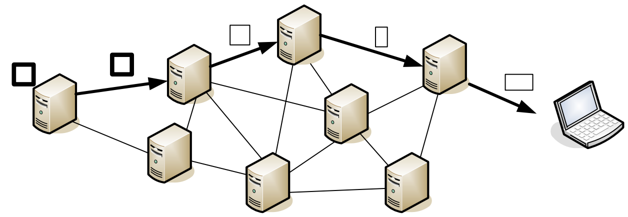

Figure 2. Adaptation Paths

This technique is especially useful in our scenario, as we think of clients providing content to a content network and other clients requesting from it. So there are good chances of doing multiple adaptation operations using adaptation paths.

2.5 Groupware Applications

As mentioned above, data conferencing functionality is widely used in collaborative systems that integrate synchronous and asynchronous collaboration. The older TeamRooms [4] system uses a single server where virtual team rooms are provided and made persistent. The virtual meeting tool of the later TeamSpace [5] system uses a centralized T.120 topology with a single top-node. As opposed to that, CINEMA [6] builds upon the SIP family of standards. Basic server elements like a whiteboard cache, screen sharing server, and instant messaging server are introduced. This inherently shows the need for a content delivery service used by multiple collaborative applications.

The more general problem of sharing resources in synchronous groupware systems has already been examined in [24]. Exisiting single-user applications may be hooked transparently, so that file access requests are routed via proxies to the original source. Special requirements of multimedia content like propagation, adaptation, and efficient distribution are not covered by this approach.

3 Architecture and Protocols for Adaptive Content Distribution

3.1 Architecture

As could be seen in Section 2.2, there exists a number of possible tightly-coupled target architectures. The older T.120 obviously offers more features at the moment and is widely deployed but is likely to become less important as soon as SIP-based solutions offer at least the same functionality in the future. Besides that, a Jabber-like topology of a P2P network of servers is also very popular in this area. Thus we define an abstract architecture model that can be used in the three mentioned environments.

The architecture (see Figure 3) consists of multiple client and server nodes that are all connected to an abstract conferencing session. These nodes can run a Content Delivery Client (CDC), a Content Delivery Service (CDS), or an instance of each of the two entities. A CDC is always connected to a CDS using a client-server protocol.

It is assumed, that the conferencing session has a notion of presence so that the nodes can identify each other and initiate direct connections. The second assumption is, that this presence is augmented with the type of content distribution entity, so that nodes can be identified either as running a CDC, a CDS, none, or both. Extendable presence services are provided by all tightly-coupled conferencing systems reviewed in this paper, so that these conditions can be easily fulfilled. The third necessary assumption is, that clients are able to identify the nearest or best-suited CDS for their purposes. This can be done by DNS address resolution within a SIP or Jabber system or by using the user database replication (roster exchange) of T.120.

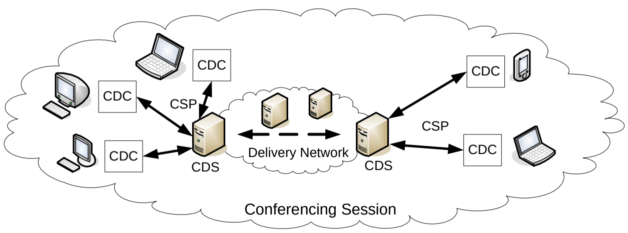

Figure 3. Architecture

CDS services can connect to each other to form a delivery network, but no implication is made on how this should be done. The necessary information arises out of the presence information of the conference session. The simplest case is the use of only one CDS instance in a conference session.

The CDS combines part of the functionality of surrogates and adaptation services that are useful in our application scenario. As a surrogate, the CDS is responsible for delivery of content to the edge of the network by providing a means of prefetching. It is directly addressed by associated clients on each request and caches delivered material for future requests from other clients. As an adaptation service, the CDS provides scaling, transcoding, or any other adaptation operation that can be processed on content entities prior to edge delivery.

In a content distribution network, an overlay session concludes all surrogates to enable efficient distribution and replication of content objects. In our architecture, this overlay session is enhanced to include the clients, too. The existing conferencing session is used for this purpose. This enables each client to act as an origin server and be an active part of content distribution. Content distribution is seen as a continuous activity throughout the lifetime of a conferencing session, using idle resources for caching and prefetching. CDS servers can keep track of their client's capabilities and load and thus enable seamless distribution of content.

3.2 The Content Sharing Protocol

We defined a client-server protocol for communication between CDC and CDS – the Content Sharing Protocol (CSP). A collaborative conferencing application uses a CDC to access resources of other conference participants that they offer in the conference session. CSP supports capability exchange, generic content offering, adaptive image distribution, and binary transmission of data. The CSP messages used for these purposes are described in the following, using an example of adaptive distribution of an image slide used within a presentation sharing application.

Capability Exchange

Exchange of capabilities occurs either if a CDC connects for the first time to a CDS or if its capabilities changed. In both cases, it sends an UPDATE to its associated server. Capabilities do not have to be distributed to the other members of the session as far as the CDS is responsible for adapting data flows that are headed to its clients. In case of the first connection attempt, the CDS registers the new CDC as a client and stores its capabilities.

Thus, the major element of UPDATE is a capability description in the Media Feature Sets format.

A simple example of UPDATE is shown as follows:

```
UPDATE <client URI> CSP/1.0
...
Accept: image/png, image/jpeg, image/gif
Capabilities: display-x<=1024, display-
y<=768, ua-media=screen, pix-x<=4096,
pix-y<=4096
```

The capabilities of a desktop computer are described that supports a maximum resolution of 1024 by 768 pixels on a scrollable screen. As media types, the three image formats PNG, JPEG, and GIF are accepted.

Content Offering

Content offering can be done by any participant at any time during a session using an OFFER message that is sent to the server. The server distributes this message to all other known CDS in the session and to its clients. Incoming offers from other CDS elements are forwarded to the connected clients, so that the offers are made available in the whole conferencing session.

An offer assigns one or more content URIs to a location in the virtual collaboration space defined by the collaborative application using the CDC. A line-based description format is used for this purpose. The location can be either a slide of a shared presentation or a rectangular view defined by some other protocol. It is also possible to change and remove offers with an OFFER message using different headers.

By using this generic concept of dynamic URI-assignment, any type of content can be exchanged using CSP as a basis. The content distribution is delegated to specific media protocols, which are defined for the requirements of an appropriate media viewer.

The CDS is also responsible for supporting latecomers with the necessary OFFERs. While sending offers to the late joined CDC, more OFFERs may arrive or existing offers may be altered or removed. We do not cover late join support more detailed in this paper due to space constraints.

Examining content characteristics

The characteristics of the original content may be transmitted along with an OFFER or by using a HEAD-request. In both cases, a source representation is described in the form:

```
Features: type=image/png, pix-x=720,
pix-y=540, color=full, color-bits=24,
size=124336
```

The example describes a slide in PNG format, which was exported from a PowerPoint presentation. The example slide contains photographic images and is thus not well-suited for lossless as shown by the size-tag. A typical text slide in PNG format is usually less than 10 KB in size (when using ~400 kpix spatial resolution).

Adaptive Image Distribution

As an example for generic media protocols, a protocol for adaptive image distribution within the Content Sharing Protocol was defined.

An image has to be requested by the host specified in the URI that was associated with a workspace location by an OFFER message. To enable bundling of requests and responses, the GET request has to be sent first to the associated CDS. It then decides whether to answer the request from the local cache or to issue a request to the offering client.

To enable adaptation decisions, a GET request contains user preferences as shown in the following example:

```
GET <uri> CSP/1.0
...
Preferences: pix>=100000, size<=50000,
time<=4000;q=0.4, compression-
quality>=80/100
```

A minimum of pixels is requested, which is one of the different dimensions of quality perceived by the viewer. The q-tag enables the specification of weighting. The compression quality is used as a quality constraint in addition to the number of pixels and refers to the quality parameter passed to an encoder with lossy compression.

Responses are given in a HTTP-like manner with different status codes. The content itself is not embedded in the response but transmitted separately using a protocol for efficient parallel transmission of multiple binary data buffers.

The features of the delivered object representation, which may be the result of multiple adaptation operations, are transmitted to the client:

```
200 OK
...
Features: type=image/jpeg, pix-x=504,
pix-y=378, color=full, size=35733,
adaptation-steps=2, compression-
quality=80/100
```

The original PNG slide was scaled down and then transcoded to the JPEG format. This instance of the content finally reaches the client and is displayed on the screen of the end-system, scaling it to the dimensions of the original slide as propagated with OFFER.

Other media distribution protocols might use very similar design principles while providing some more sophisticated adaptation methods. A protocol for document distribution might consider adaptation of embedded objects as an option while for video objects both frame rate and spatial resolution are candidates for adaptation.

Binary Transmission

The final step in the distribution process is the transmission of the adapted object representations from one node to the other. The mentioned forwarding technique from P2P-networks should be used in the case of multiple requests with nearly identical preferences to speed up content distribution for shared viewing.

A handle is declared between the two peer nodes within the media protocol that refers to the content transfer throughout its lifetime. It is now up to certain low-level delivery mechanisms to enforce different priorities, pause, continue, or cancel transfers.

4 Implementation

To evaluate the proposed approach, we implemented a system that realizes the architecture and the Content Sharing Protocol with the components CDS, CDC, a presentation sharing client application, and a presentation sharing server application. It supports presentation sharing and photo sharing with integrated annotations, telepointer, and chat. The components are implemented in C++ using state machines as a key technology to establish a distributed control flow.

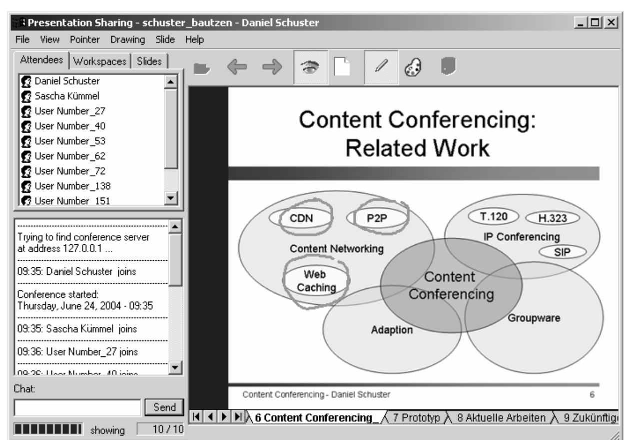

Figure 4. Presentation sharing client application

We chose the T.120 architecture for realization as it is currently the only collaboration environment with standardized applications like whiteboard and application sharing. We built upon T.126 to create a presentation sharing application with integrated drawings, telepointer, and chat. The push transfer of images in this protocol was replaced by integration of the CDC in the client application.

Besides caching and data adaptation, the system also supports client-controlled prefetching. The presentation sharing clients simply request the next slide of the presentation during idle time. Activated prefetching usually completely reduces the transfer time to some marginal cache-hit time in the common case of a sequential presentation.

Basic tests with a set of PowerPoint presentations gathered from the Web show the applicability of our approach. Dynamically transcoding the biggest 12% of the slides to JPEG reduces the overall transfer time of all slides to 75% compared to the original slides. Combining this with scaling both dimensions to 70% of the original size, reduces this to 64%. Nevertheless, these tests still have a limited meaning, as the different adaptation operations caching, prefetching, scaling, and transcoding influence each other. No common notion of quality is still available to control such combination of adaptation operations. So the definition of architecture and protocols described in this paper is only the first step towards adaptive distribution of multimedia content in a real-time setting.

5 Conclusions

We discussed the problem area of adaptive distribution of multimedia content within collaborative conferencing sessions. Based on broad review of related work, an architecture and protocols for adaptive multimedia distribution to be used by collaborative applications like presentation sharing or document sharing was defined.

The implementation and basic tests show the feasibility of our approach. We plan to examine adaptation operations as supported by the protocol more detailed to allow the definition of algorithms for efficient combination of adaptation operations in the mentioned scenario. Once this research work is done, a detailed evaluation with quantitative comparison to other approaches can be carried out.

6 References

[1] Handley, M., Crowcroft, J., Bormann, C., Ott, J., Very large conferences on the Internet: the Internet Multimedia Conferencing Architecture, *International Journal of Computer and Telecommunications Networking*, *31*(3), 1999, 191-204.

[2] Packet-based multimedia communications systems, ITU-T Recommendation H.323, International Telecommunication Union, 2000.

[3] Rosenberg, J., Schulzrinne, H., Camarillo, G., Johnston, A., Peterson, J., Sparks, R., Handley, M., Schooler, E., SIP: Session Initiation Protocol, Request for Comments 3261, IETF, 2002.

[4] Roseman, M., Greenberg, S., TeamRooms: Network Places for Collaboration, *ACM Conference on Computer Supported Cooperative Work (CSCW)*, Boston, MA, USA, 1996.

[5] Geyer, W., Shahrokh, D., et al., Virtual Meeting Support in TeamSpace, *ACM Conference on Computer-Supported Cooperative Work (CSCW'2000)*, Philadelphia, PA, USA, 2000.

[6] Singh, K., Wu, X., Lennox, J., Schulzrinne, H., Comprehensive Multi-platform Collaboration, *SPIE Conference on Multimedia Computing and Networking*, Santa Clara, CA, USA, 2004.

[7] Deering, S., Estrin, D., Farinacci, D., Jacobson, V., Liu, C., Wei, L., An Architecture for Wide-area Multicast Routing, *IEEE/ACM Transactions on Networking*, *4*(2), 1996.

[8] Lennox, J., Schulzrinne, H., A Protocol for Reliable Decentralized Conferencing, *NOSSDAV'03*, Monterey, CA, USA, 2003.

[9] ITU-T Recommendation T.120, Data Protocols for Multimedia Conferencing, International Telecommunication Union, 1996.

[10] Jabber Software Foundation, www.jabber.org, 2005.

[11] ITU-T Recommendation T.122, Multipoint Communication Service - Service Definition, International Telecommunication Union, 1998.

[12] ITU-T Recommendation T.126, Multipoint Still Image And Annotation Protocol, International Telecommunication Union, 1997.

[13] Trossen, D., Helbig, T., The ITU T.120 Series of Standards as Basis for Conference Applications, *SPIE Voice, Video and Data Communications*, Dallas, TX, USA, 1997.

[14] Trossen, D., Scalable Group Communication in Tightly Coupled Environments, PhD Thesis, University of Technology North-Rine Westfalia Aachen, Faculty of Mathematics, Computer Science, and Natural Science, 2000.

[15] Singh, K., Nair, G., Schulzrinne, H., Centralized Conferencing using SIP, *IP telephony workshop*, Columbia University, New York, NY, USA, 2001.

[16] Rosenberg, J., A Framework for Conferencing with the Session Initiation Protocol, draft-ietf-sipping-conferencing-framework-05, IETF SIPPING Working Group, May 2005.

[17] Peng, G., CDN: Content Distribution Network, Technical Report, State University of New York at Stony Brook, Stony Brook, NY, USA, 2003.

[18] Cohen, B., Incentives Build Robustness in BitTorrent, *Workshop on Economics of Peer-to-Peer-Systems*, Berkeley, CA, USA, 2003.

[19] Springer, T., A Component-based Meta-Model of Context-aware Adaptation Graphs for Mobile and Ubiquitous Applications [in German], PhD thesis, TU Dresden, Department of Computer Science, 2004.

[20] de Lara, E., Wallach, D. S., Zwaenepoel, W., Opportunities for Bandwidth Adaptation in Microsoft Office Documents, *4th Usenix Windows Systems Symposium*, Seattle, Washington, USA, 2000.

[21] Chandra, Chandra, S., Gehani, A., Ellis, C. S., Vahdat, A., Transcoding Characteristics of Web Images, *Multimedia Computing and Networking (MMCN'01)*, San Jose, CA, USA, 2001.

[22] Klyne, G., A Syntax for Describing Media Feature Sets, RFC 2533, IETF Network Working Group, 1999.

[23] Chandrasekaran, S., Madden, S., Ionescu, M., Ninja Paths: An Architecture for Composing Services Over Wide Area Networks, University of California, Berkeley, 2000.

[24] Begole, J., Smith, R. B., Struble, C. A., Clifford, A. S., Resource Sharing for Replicated Synchronous Groupware, *IEEE/ACM Transactions on Networking*, *9*(6), 2001.