Improving Media Services on P2P Networks

The Media Accelerating Peer Services system extends P2P infrastructures to improve multimedia services across heterogeneous computing platforms.

Rainer Lienhart, Matthew Holliman, Yen-Kuang Chen, Igor Kozintsev, and Minerva Yeung
Intel Labs

The Internet's advent as the dominant application, content, and service delivery pipe set off a dramatic evolution among computing paradigms. The evolution continued as applications and services gradually migrated from client-server to edge services, and has now arrived at peer-to-peer (P2P) computing frameworks that can accommodate vast numbers of users. To comprehend this evolution's impact, consider the great speed with which Napster amassed tens of millions of users, often with hundreds of thousands or even millions online concurrently at peak periods, while still maintaining acceptable service.[1] No existing client-server-based content delivery infrastructures can match that scalability. Even edge services with multiple caching servers would be hard pressed to accommodate a fraction of the delivery capacity popular P2P services make possible.

While distributed computing platform research usually focuses on platforms with roughly equivalent computing components, P2P computing emphasizes the heterogeneity in uptime, computing capabilities, and connectivity bandwidths between peers. This diversity will only increase as the infrastructure accommodates wireless connections and devices such as personal digital assistants, tablets, compact media players, and cell phones (see Table 1, next page).

In this article, we present an architecture and resource management and adaptation framework that transcends existing infrastructures to accommodate and accelerate multimedia peer applications and services. We also propose key technology components that support seamless adaptation of resources to enhance quality of service and the building of better tools and applications that utilize the peer-computing network's underlying power. As we'll show, we have developed a prototype system that integrates the various components and sample applications that can be built on the proposed infrastructure.

 1089-7801/02/$17.00©2002 IEEE http://computer.org/internet/ JANUARY • FEBRUARY 2002

Table 1. Inequality in heterogeneous computing platforms.

Devices	CPU	Memory	Storage	Connectivity	Screen
Server	Multiple 1GHz+	2 Gbyte	>100 Gbyte	1+ Gbps	N/A
PC	Single 1GHz+	256 Mbyte	40 Gbyte	100 Mbps	1,600 × 1,200
Laptop	600 MHz	128 Mbyte	10 Gbyte	100 Mbps	1,024 × 768
Media PDA	200 MHz	16 Mbyte	8 Mbyte	19.2 Kbps	320 × 240

Challenges and Opportunities with Multimedia

Among the many challenges associated with multimedia are volume of data, real-time requirements, and complexity of manipulation. It is particularly important to consider these issues in P2P networks given the diversity of content and users. Particular challenges include these:

- While more static content such as Web pages and images are available by download-style data transfers (such as HTTP), multimedia might require streaming and real-time viewing using an unreliable protocol such as RTP/UDP.
- Video and audio are inherently content-rich media. Queries must take advantage of this content rather than relying simply on such attributes as filenames or file sizes. Therefore, the network should extract as much meta-information as possible from media files to enable efficient search, including information explicitly stored inside the files and attributes automatically derived from the content.
- A user might choose to access content on the network through a handheld device connected by a low-bit-rate wireless link or from a high-end desktop machine connected by broadband. As the computational power and channel characteristics available to users vary widely, content that plays on one device might not necessarily be viewable on another without proper data-transformation operations.

Multimedia files can exist in a variety of different formats, resolutions, and qualities. Hence, depending on end users' requirements, a multimedia-aware delivery system might send the same content to users differently.

System Architecture

To achieve improved media service over heterogeneous peer networks, our proposed system architecture extends existing P2P infrastructures with several modules. These modules are collated into the Media Accelerating Peer Services (MAPS) shown in Figure 1. The system's software layers are, from bottom to top,

- the traditional underlying operating system and related services;
- the P2P service layer;
- the MAPS media support modules; and
- the application layer.

The P2P service layer provides basic functionality such as peer discovery, user and group management, name resolution, and delivery primitives. Examples of systems providing some or all of these services are Gnutella,[2] FreeNet,[3] and JXTA.[4] MAPS extends this layer with a plug-in architecture for customizing basic search and delivery operations. MAPS provides two extensions: a module enabling real-time media streaming (via RTP over UDP) and a module for enhanced search capabilities (via distributed SQL).

The MAPS media support modules use the P2P service layer for network and resource access and provide transformation and computation services on data obtained from or sent to the P2P service layer. Examples include a support module for MPEG video that provides transcoding support and a module that automatically analyzes images, videos, and music files on each peer to provide better search capabilities (see http://www.videoanalysis.org/).[5]

P2P Service Layer

Our P2P service layer provides the basic functionality common to many systems, implementing the underlying membership, transport, and lookup services. In addition, the service layer extends these capabilities by providing further functionality required to support higher-level media services. Here, we discuss only the differences or additions to existing architectures required to better support media data.

Considering the variety of network conditions and environments in which a P2P system can operate, a flexible architecture must support different transport mechanisms. Our framework supports the incorporation of different protocols (HTTP and RTP/UDP, among others), letting the application specify its required transport. For example, a video player on a wireless handheld device might request that the peer services layer send video only via RTP/UDP in combination with advanced forward error correction (FEC) and automatic repeat request methods.[6] In contrast, a delay-insensitive application that prefetches and downloads videos to a home server might specify the use of a reliable protocol such as TCP. The peer services layer instantiates the appropriate transport object on any source or intermediate hosts to supply the object as requested. In MAPS, we assume that the peer nodes are fully connected and can communicate to each other, such as through the IP network.

Using filenames to identify and describe data is a legacy artifact that is not necessarily appropriate for adequately describing rich content, particularly when dealing with audio and video. As a result, support for alternative naming systems is required. In our method, we query any media properties through a distributed heterogeneous database. For example, we have implemented a distributed SQL query engine to support alternative naming systems in the P2P service layer.

P2P networks should also be aware of wireless-specific characteristics — battery status, mobility, and so forth — in addition to parameters common to wired peers such as available bandwidth, screen resolution, and available media players. Wireless devices announce their capabilities and status to other peers for refining search and delivery. For example, if a wireless device's physical location changes, higher-layer services such as MAPS can personalize content delivered to the device to better reflect its new location.

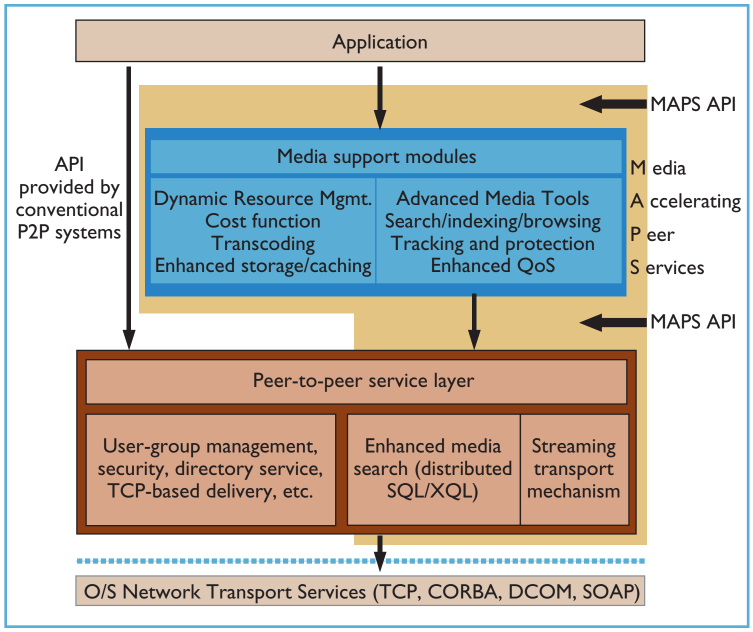

Figure 1. MAPS system architecture. The underlying P2P service layer handles basic functionality common to most P2P systems. MAPS extends this functionality by adding tools to enable the development of rich media applications over the same infrastructure.

Media Support Modules

In a network where data is shared and proliferated, multiple copies of a given file typically will exist throughout the system. To enhance P2P system efficiency, MAPS dynamic resource management handles transmission and retrieval of such content intelligently and seamlessly, such as by giving preference to copies on local hosts or to copies on hosts connected by high-speed connections. When the data of interest is multimedia, a larger array of possibilities arises. For example, different versions of the same content are likely to exist throughout the network at multiple bit rates and resolutions. Furthermore, MAPS can transcode or transform (morph) content, even on the fly, to generate versions at new bit rates and resolutions or to incorporate additional redundancy for error resiliency over noisy channels — in other words, to make the content adaptive to client device capabilities, storage capacities, and bandwidth constraints.

The best way to locate a media object version on the network is determined by ranking and prioritizing available paths to the object. In this context, a path consists of both a conventional network route and an optional sequence of transformations throughout the network necessary to deliver the object to the destination node in satisfying the requester's constraints. The cost function measures the aggregate computational complexity and transmission overhead required to deliver a given object. To find the most efficient sequence of operations, MAPS attempts to minimize the cost function over the set of possible paths.

Figure 2 (next page) shows how MAPS determines path selection when a node requests a media file from the network. The requesting node constructs a directed graph representing the relationships between participating nodes and their advertised costs for carrying out any needed transformations. The figure expresses transmission and transcoding costs as edge weights, while peer nodes and content transformations map to vertices. Finding the optimal sequence of operations at some instant corresponds to solving the single-pair shortest-path problem for this graph, so that a well-known algorithm such as Dijkstra's suffices to route the request.

In the example in Figure 2, node *A*, which possesses only an MPEG-4 decoder, requests an MPEG-

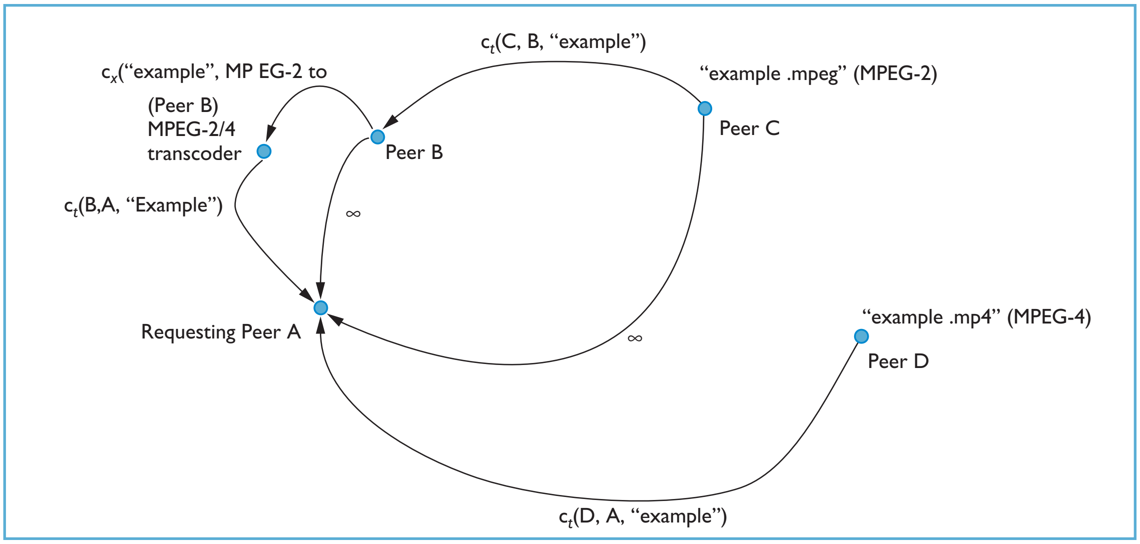

Figure 2. Transcoding and path selection graph construction. In the example, node A requests delivery of a video stream "example," which exists at nodes C and D, and compares the cost of obtaining the clip along different paths.

4 version of a stream ("example"). The requested content is available in MPEG-2 format from *C*, or directly as an MPEG-4 sequence from *D*. *A* compares the cost of obtaining the clip along several paths: directly from *D*, which involves a transmission cost $c_t(D, A)$; directly from *C*, which cannot satisfy *A*'s constraints and thus is assigned infinite cost; and finally from *C* to *B* to *A*, which involves the transmission cost $c_t(C, B) + c_t(B, A)$, plus the cost of transcoding at *B*, c_x("example", MPEG-2 to MPEG-4).

Beyond dynamic resource management, we also provide advanced media tools to enable new uses and users of a multimedia service platform. The MAPS advanced media tools provide a toolbox to extract syntactic and semantic attributes automatically from media exploiting two sources of metainformation: metainformation encoded in the media files such as date and time of recording, and information derived by automatically analyzing the media content such as locations of shot boundaries and faces, key frame-based video summarizations,[5,7] and transcriptions of text in images and videos into ASCII (see http://www.videoanalysis.org).

MAPS Prototype System

We built our prototype system using Java for the client programs and C/C++ for the MAPS core. All components run under Windows and Linux. Media players were developed in-house.[8] Figure 3 shows how our prototype can stream a media clip stored on a remote peer to a wireless peer, through an intermediate transcoding node. When a user with a handheld device requests a media file from the peer explorer, MAPS returns a list of possible downloads with their associated costs, including standard MPEG-1, high-quality MPEG-2, and low-bit-rate MPEG-4 sequences. When the user selects a version already on the network, the file is streamed to its destination.

If no media file matches the capability of the wireless handheld device or the user's request, MAPS transcodes the content. For instance, suppose the user requests an MPEG-4 version of a video clip that exists only in MPEG-2 format, possibly at a different rate and resolution. After performing the search, our P2P system discovers the MPEG-2 version of the requested clip and returns the cost of streaming it. In this scenario, transferring the MPEG-2 sequence over the wireless link is costly — the handheld device lacks the computational power to decode and render the sequence in real time, and the device has limited battery life so it should not transcode itself. MAPS thus assigns a peer to perform the transcoding based on the cost evaluation. In the example shown in Figure 3, MAPS chooses a different transcoding peer from the peer containing the media content, because the former is a more powerful machine in our testbed.

Conclusions

Media data possesses a number of unique characteristics that differentiate it from other data types, including error tolerance, real-time requirements, rich semantics, and transformability. To make such rich content pervasive and easy to use in a peer-to-peer system, we argue that it is crucial to consider these characteristics when designing the system.

In this work, we have shown how peer-to-peer systems can be enhanced to better support media. To this end, we have described MAPS, a flexible architecture on which peer-to-peer multimedia applications can be built. As the proliferation of media content on the Internet and peer-to-peer applications meet head on, we look forward to the development of systems that share the same goals of making rich media widely accessible to users.

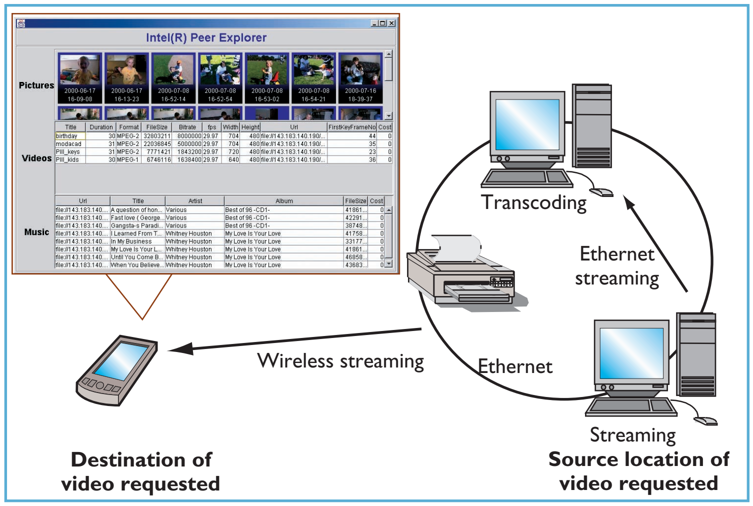

Figure 3. Media streaming with transcoding to a wireless peer device. The wireless handheld device requests a low-bit-rate, MPEG-4-encoded version of a high-bit-rate, high-quality video sequence from the source location. The video sequence is transcoded on-the-fly before the wireless gateway to fulfill the request.

References

1. M. Macedonia, "Distributed File Sharing: Barbarians at the Gates?," *Computer*, vol. 33, no. 8, Aug. 2000, pp. 99–101.
2. S. Tilley and M. DeSouza, "Spreading Knowledge about Gnutella: A Case Study in Understanding Net-Centric Applications," *Proc. 9th Int'l Workshop on Program Comprehension*, IEEE Computer Soc. Press, Los Alamitos, Calif., 2001, pp. 189–198.
3. I. Clarke et al., "Freenet: A Distributed Anonymous Information Storage and Retrieval System," *Designing Privacy Enhancing Technologies: Int'l Workshop on Design Issues in Anonymity and Unobservability*, H. Federrath, ed., Springer-Verlag, Berlin, 2001, pp.46-66.
4. L. Gong, "JXTA: A Network Programming environment," *IEEE Internet Computing*, vol. 5, no. 3, May/June 2001, pp. 88–95.
5. R. Lienhart, "Reliable Transition Detection in Videos: A Survey and Practitioner's Guide," *Int'l J. Image and Graphics*, vol. 1, no. 3, 2001, pp. 469-486.
6. D.G. Sachs et al., "Hybrid ARQ for Robust Video Streaming Over Wireless LANs," *Proc. Information Technology: Coding and Computing*, IEEE Computer Soc., Los Alamitos, Calif., 2001, pp. 317–321.
7. B. Davies, R. Lienhart, and B.-L. Yeo, "The Video Document," *Multimedia Storage and Archiving Systems IV*, vol. 3846, Sept. 1999, pp. 22–34.
8. M.M. Yeung, "MPL: MPEG Processing Library—Tools and Advanced Technology for Video-Centric Applications," *Intel Developer Forum*, Palm Springs, Calif., Sept. 1999.

Rainer Lienhart is a staff researcher with Intel Labs. His research interests includes image/video/audio content analysis, machine learning, scalable signal processing, scalable learning, ubiquitous and distributed media computing in heterogeneous networks, media streaming, and peer-to-peer networking and mass media sharing. He received a PhD in computer science from the University of Mannheim, Germany.

Matthew Holliman is with the Media and Internet Technology Group in Intel Labs. His research interests include media and Internet technology, focusing on content delivery and protection. He received a B.S. from the University of Illinois at Urbana-Champaign and an M.S. from Northern Illinois University, both in computer science.

Yen-Kuang Chen is a researcher with Intel Labs. His research interests include video compression and processing, architecture and algorithm design in multimedia computing, video and graphics hardware design, and performance evaluation. He received a PhD in electrical engineering from Princeton University. He is on the editorial board of the *Journal of VLSI Signal Processing Systems*, cochair of the MPEG-4 Intellectual Property Management and Protection Ad Hoc Group, and chair of the MPEG Study on Standard Digital Watermarking Technology Ad Hoc Group.

Igor Kozintsev is a senior software engineer with Intel Labs. His research interests include multimedia processing, wireless communications, computer vision and pattern recognition, information theory, coding theory, and statistical learning theory. He received a Diploma in electrical engineering from Moscow State Technical University and MS and PhD degrees in electrical engineering from the University of Illinois at Urbana-Champaign.

Minerva Yeung is a principal engineer and research manager at Intel's Media and Internet Technology Group, where she leads a team of researchers building next-generation digital media technology, Internet multimedia infrastructure, and related applications and services. She received a BSEE from Purdue University and MA and PhD degrees from Princeton University. She is on the editorial board of *IEEE Transactions on Multimedia* and *IEEE Signal Processing Letters*.

Readers can contact the authors at {rainer.lienhart, matthew.holliman, yen-kuang.chen, igor.v.kozintsev, minerva.yeung}@intel.com.