



The following paper was originally published in the  
Proceedings of the 2nd USENIX Windows NT Symposium  
Seattle, Washington, August 3–4, 1998

## COM on a Multicast Transport

P. Emerald Chung and Yennun Huang  
*Bell Laboratories, Lucent Technologies*  
Yi-Min Wang  
*Microsoft Research*

For more information about USENIX Association contact:

1. Phone: 510 528-8649
2. FAX: 510 548-5738
3. Email: [office@usenix.org](mailto:office@usenix.org)
4. WWW URL: <http://www.usenix.org/>

# COM on a Multicast Transport

P. Emerald Chung and Yennun Huang

{emerald, yen}@research.bell-labs.com  
Bell Laboratories, Lucent Technologies  
Murray Hill, New Jersey

Yi-Min Wang

ymwang@microsoft.com  
Microsoft Research  
Redmond, Washington

## Introduction

Due to the wide availability of the Internet and intranets, the use of multicast applications, such as document sharing, collaborative groupware and multimedia delivery, has grown quickly. Another trend in software development is rapid application development by using off-the-shelf components. Microsoft Component Object Model (COM) [COM 95] has become a popular paradigm for fast component integration. In this paper, we extend the COM object model to integrate with a multicast transport protocol, called RMTP [Paul 96]. The goal is to develop a framework that enables building multicast applications from off-the-shelf components. We demonstrate our initial results with a stock quotes dissemination system.

## COM over RMTP

RMTP is built upon the foundation of IP multicast, and provides sequenced, lossless delivery of a data stream from one sender to a group of receivers. There is little correlation among the group of RMTP receivers. The Internet Group Management Protocol (IGMP) establishes a way of joining and leaving multicast groups. RMTP has been used commercially for data dissemination in wide area network where strong state consistency among the receivers is not required.

COM specifies an architecture, a binary standard, and a supporting infrastructure for building, using, and evolving component-based applications. It extends the benefits of object-oriented programming such as encapsulation, polymorphism, and software reuse to a dynamic and cross-process setting.

The overall architecture of COM+RMTP is shown in the figure. We leverage many mechanisms from the *Distributed COM (DCOM)* [Brown 96] technologies. DCOM is an object-oriented RPC protocol. It provides the infrastructure that allows a client process to create server objects on remote machines and to invoke their methods in a location-transparent way. While the standard DCOM implementation is designed for one-to-one, synchronous invocation, RMTP is used for one-to-many, asynchronous invocation. By encapsulating such semantic difference in the channel and stub manager

components (see figure), we are able to implement our system by exploiting the DCOM custom marshaling mechanism to replace the RPC transport with the RMTP protocol, while reusing standard proxy/stub components provided by DCOM for data marshaling.

## An Application

We use the COM/RMTP integration to implement a stock quote dissemination system. A client gets the stock quote from the Internet and makes COM method calls to multicast the data to a display component running on many (server) machines. The client and the servers communicate through a well-defined COM interface. It is interesting to note that, as we added more types of data to be delivered, new client and a new interface in the server were built to accommodate the changes without breaking old clients.

## Research Issues

There are several issues to be addressed in order to make this framework practical: (1) a protocol to advertise a group and its COM interfaces, (2) a mechanism to handle multiple callbacks if servers need to notify the client, and (3) a tool to automate the development of multicast-aware components. We are currently working on these issues and exploring potential applications.

## Reference:

- [Paul 96] J. C. Lin and S. Paul, "RMTP: A Reliable Multicast Transport Protocol", INFOCOM '96, pp. 1414-1424.
- [COM 95] The Component Object Model Specification, <http://www.microsoft.com/oledev/olecom/title.htm>
- [Brown 96] N. Brown, C. Kindel, Distributed Component Object Model Protocol -- DCOM/1.0, [http://premium.microsoft.com/msdn/library/techart/msdn\\_dcomprot.htm](http://premium.microsoft.com/msdn/library/techart/msdn_dcomprot.htm)

