TransMIDI: A System for MIDI Sessions Over the Network Using Transis

Dan Gang * Gregory V. Chockler Tal Anker Alex Kremer
Tomas Winkler
Institute of Computer Science
The Hebrew University of Jerusalem, Israel
{dang, grishac, anker, kresa, tomasw}@cs.huji.ac.il

Abstract

We built a system that allows musical performers (and listeners) who wish to play together to organize into multiple session groups. The users interact in real time over the network, and may dynamically join or leave a session group. The players contribute to the session by playing on their MIDI controllers, using General MIDI protocol. We assume a totally asynchronous environment in which failures of both end systems and communication links are possible. TransMIDI was implemented using the Transis group communication system for fault tolerance and coordination. Transis is a transport layer that supports efficient and reliable multicast and membership services. The advanced Transis group services allow for different forms of cooperation among performers, so that many different kinds of musical ensembles can be easily established. Moreover, TransMIDI provides its users with means for exploring novel ways of musical interaction.

1 Introduction

Increasing improvement in communication technology will invite widespread utilization of multimedia interactive services that involve multiple parties communicating via the Internet. Current research effort in this area concentrates on audio and video transmission over the network. Examples of such systems are [Cor, ACK+97, CD]. In this paper we present a system called *TransMIDI* for conducting interactive musical session among a group of musicians. Our system utilizes the Transis group communication system. The Transis system supports reliable multicast services with various ordering options, as well as the multicast group abstraction. These services have been proven to be very useful for networking multimedia applications [CHKD96, ACDK97].

The large amount of data associated with audio and video files, raises considerable problems for developing real-time multimedia applications over the network. One solution is the improvement of the hardware performance and software efficiency. Another solution is to devise new compact media representations and compression schemes. In TransMIDI we used the MIDI standard for musical transmission because it is compact and sufficiently powerful for a wide variety of musical applications.

Presently, we are witnesses to a surge of software solutions for various musical activities, which exploit modern communication and information technologies. For example, the project group MusicWeb [BJH96] was formed from a few participating universities and research centers in Europe to design a system to solve the requirements of teaching systems with information retrieval abilities. NetMuse [Arn96] is a project to link all university Music departments in Scotland by means of ATM-based multimedia network.

2 Background

In this paper we describe the *TransMIDI* system, which enables its users to interact in real time over the network using different forms of cooperation. We chose MIDI for data transmission of digital music information. As opposed to audio files, which require a huge amount of (sampled) data, MIDI uses only a very restricted amount of data (since no information of the sound's characteristics that needs to be transferred). Industrial developments with MIDI have led

*Dan Gang is supported by the Eshkol Fellowship, Israeli Ministry of Science

to the emergence of high-quality low-price sound generators and sound cards for personal computers. The massive production of MIDI keyboards, MIDI guitars and wind controllers provide various means to perform music. Such controllers produce streams of events which are easily stored and transmitted. MIDI-based applications and libraries provide tools for controlling and editing MIDI data. The digital nature of MIDI enables flexible mutual transformations between graphics, movements and MIDI information. Our project relies upon the General MIDI protocol, which allows end users to obtain similar sounds, regardless of the type of musical hardware used.

Despite all this, MIDI does have its limitations, particularly concerning the definition of new tuning systems and the dynamic control of continuous data. Additionally, it cannot dynamically control audio parameters. While alternatives that overcome these limitations do exist (such as the ZIPI protocol [ZIP94]), we nevertheless chose MIDI because of its widespread availability.

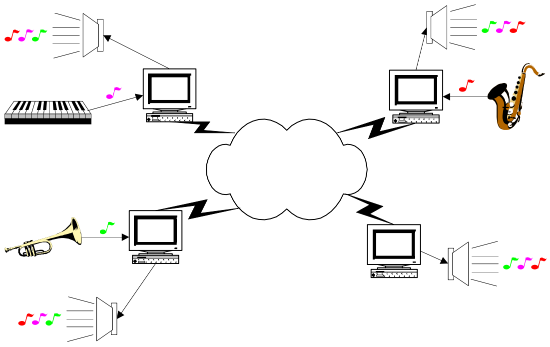

Figure 1: A Jam Session Over the Network

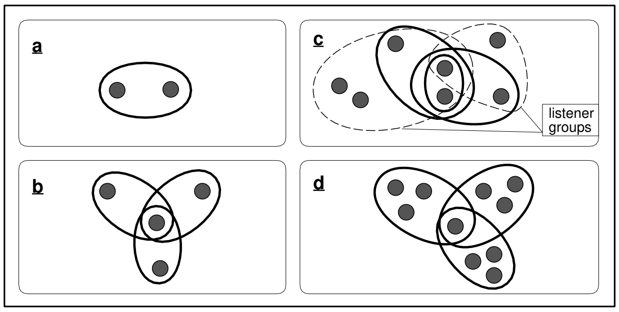

Figure 2: Various group topologies for different musical jam sessions

3 The Transis Group Communication System

Transis is a group communication system that has acquired a wide recognition in the academic community. Transis [DM96] supports efficient and reliable multicast and membership services. The Transis multicast services offer various types of message ordering. For example, the *Causal* multicast service guarantees that the reply to a message is never delivered before the message itself at any target. The *Total* multicast service extends the *Causal* service in such a way that all messages are delivered in the same order at all their targets.

Transis allows the processes to be easily arranged into multicast groups. A multicast group is identified by the *logical name* assigned to it when the group is created. Each message targeted to the group's logical name is guaranteed to be delivered to all currently connected and operational members of the group. This allows handling of a set of processes as a single logical connection. Furthermore, processes may dynamically join or leave these groups. Transis provides the group members with indication of the current group membership, i.e. the group of currently connected processes. These membership reports are delivered within the flow of regular messages.

Using Transis' services, it is easy to create groups with complex layouts. This feature is particularly useful for cooperative networking applications. In the next section we show how to use Transis to arrange performers into various topologies, with multiple ensemble possibilities available for each topology.

One of our current efforts, is the gearing of the Transis system towards new emerging network technologies such as ATM. We intend to run our system over classical UDP/IP with LAN emulation over ATM (LANE) [ATM95]. We have also developed the CONGRESS group resolution protocol [ABDL96], which we intend to exploit as a basis for a membership service over ATM networks.

4 TransMIDI Architecture

The TransMIDI system (depicted in Figure 1) enables musical performers and listeners to organize into multiple session groups. The users interact in real time over the network, and are able to join and leave the session group as they wish. The players contribute to the session by playing on their MIDI controllers, using General MIDI protocol to achieve similar sounds for each computer station.

Transis group services allow for different forms of cooperation among performers, enabling easy establishment of various kinds of musical ensembles. Within the session group, members may perform in user-predefined modes. For example, some users could be allowed to join a group actively in a "performer" mode, while others could join passively, in a "listener" mode. Additionally, some members might enjoy privileged operations within the group (for instance, a member might be a "conductor", with more liberty to control the overall performance). In this manner, the system provides different possibilities for cooperation and organization of groups.

Figure 2a shows the simplest logical group topology, in which all members perform, and every member can listen to every other member at the same time.

Figure 2b shows a leader who can hear and be heard by all other members, but these members cannot hear each other: as far as each of these members are concerned, she is playing a duet with the leader, who is playing only with her. This could occur if the leader plays a harmonic line and the others improvise their own melodies independently. The leader (alone) may collect and use data from the real-time performances.

In Figure 2c, we have a set session nucleus. New members may join this session, but not hear each other (the members of the nucleus may hear whomever they choose). Further, listeners may choose any subset of the performers. An example use of this topology might be a running "minus one" session, in which soloists can come and go at will, and listeners can hear any combination that they may choose.

Figure 2d presents a topology similar to the topology presented in Figure 2b, but with several members in disjointed groups. This topology might be interpreted as a "teacher" and a few disjointed "class groups".

There exist many other possible topologies and many other interpretations for those presented above. One of the important issues of this project is to explore new useful topologies of groups which enable new ways of musical interaction.

5 Implementation

The application is based upon two basic software packages: the Transis groupware package described in section 3 and the SGI DMEDIA MIDI support library.

SGI's DMEDIA library provides API for control over MIDI devices connected via standard Mac MIDI interface to the machine's standard RS232 serial port. The library enables easy fetching and sending of MIDI events from/to the devices connected.

The basic building block supported by TransMIDI is a group of performers and listeners. To create a new MIDI session, one is required to form a singleton group by simply joining it (specifying its name). Others can afterwards *join* the group. Members depart from the group by performing the *leave* action. When the last member leaves the group, it is automatically destroyed.

Transis provides an easy and reliable way to implement such groups. Once the Transis daemon is started on a machine, the Transis client-side API gives all the necessary group communication functionality. User actions, *leaving* and *joining*, are directly mapped to respective Transis library calls.

Once the desired session configuration is created, the system works as depicted in Figure 3. Events produced by MIDI device, are fetched by the TransMIDI application. The application passes the message containing the event to Transis, specifying all the groups the user belongs to. Transis guarantees that the message will be delivered to all the applications that constitute the specified groups. Upon the delivery of a message from Transis the application sends it to the attached MIDI device.

Note that the application process may join many groups, and receive the events produced by any one of their members. However, it is possible to filter out the events coming from a specific group. This enables, for instance, the teacher/students model depicted in Figure 2d. In this configuration the teacher is a member of three different groups. The teacher can turn off the input coming from any group and concentrate on the rest.

We use Transis' Causal multicast service to ensure that no member hears a feedback to a musical sequence, before receiving the sequence itself. On fast networks, due to the low communication overhead events sent to a group are played almost immediately on other members' machines.

The prototype system was implemented and tested on SGI machines running IRIX 5.3 and 6.2 connected to ATM based LAN. We chose the SGI platform because of its convenient MIDI support. The primary GUI will be extended as soon as we gain more experience of the various ways to exploit the system.

6 Conclusions and Future Directions

The TransMIDI system allows musical performers (and listeners) to organize into multiple session groups and interact in real time over the network.

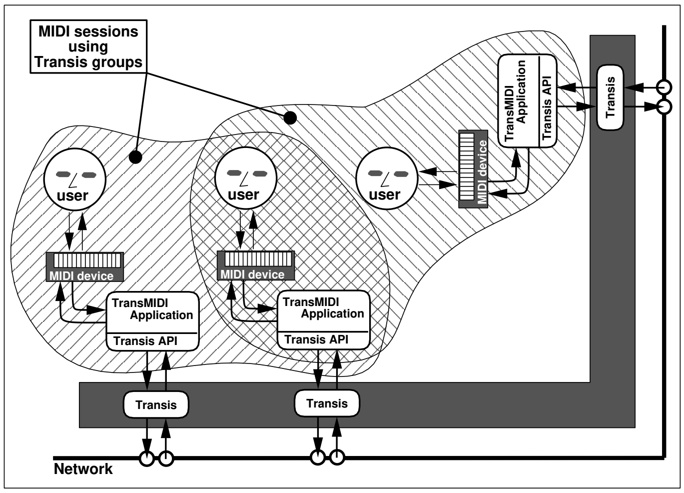

Figure 3: TransMIDI Implementation

Transis' group management has proven invaluable for effective solution of MIDI sessions. The system provides a solid platform for experimenting with different forms of cooperation among performers, exploring new group topologies in order to achieve new ways of musical interaction.

7 Acknowledgments

The support of Profs. Daniel Lehmann and Danny Dolev in this project is gratefully acknowledged. Special thanks go to Aviva Dayan who carefully read our drafts and helped to significantly improve the presentation.

References

[ABDL96] T. Anker, D. Breitgand, D. Dolev, and Z. Levy. Congress: CONnection-oriented Group-address RESolution Service. Tech. Report CS96-23, Institute of Computer Science, The Hebrew University of Jerusalem, Jerusalem, Israel, December 1996. Available from: `http://www.cs.huji.ac.il/labs/transis/`.

[ACDK97] T. Anker, G. V. Chockler, D. Dolev, and I. Keidar. The Caelum Toolkit for CSCW: The Sky is the Limit. In *Proceedings of the 3rd International Workshop on Next Generation Information Technologies and Systems (NGITS 97)*, June 1997.

[ACK+97] T. Anker, G. Chockler, I. Keidar, M. Rozman, and J. Wexler. Exploiting Group Communication for Highly Available Video-On-Demand Services. In *Proceedings of the 13th International Conference on Advanced Science and Technology (ICAST97) and the 2nd International Conference on Multimedia Information Systems (ICMIS 97)*, April 1997.

[Arn96] S. Arnold. NetMuse: a very-high-performance wide-area network for music. In *Proceedings of the International Computer Music Association*, 1996.

[ATM95] The ATM Forum. *LAN Emulation Over ATM Specification - Version 1.0*, February 1995.

[BJH96] C. Bohm, K. Jong, and C. Hempel. MusicWeb, new communication and information technologies in the music classroom. In *Proceedings of the International Computer Music Association*, 1996.

[CD] S. Casner and S. Deering. First IETF Internet audiocast. In *ConneXions 6,6 (1992)*, pages 10–17.

[CHKD96] G. Chockler, N. Huleihel, I. Keidar, and D. Dolev. Multimedia Multicast Transport Service for Groupware. In *TINA Conference on the Convergence of Telecommunications and Distributed Computing Technologies*, September 1996. Full version available as Technical Report CS96-3, The Hebrew University, Jerusalem, Isreal.

[Cor] Cornell University. *The CU-SeeMe Home Page, URL: http://cu-seeme.cornell.edu/.*

[DM96] D. Dolev and D. Malki. The Transis Approach to High Availability Cluster Communication. *Communications of the ACM*, 39(4), April 1996.

[ZIP94] ZIPI. The ZIPI Music Interface Language. *Computer Music Journal*, 18(4), 1994.