Overview of Joint editing systems: synchronous and asynchronous modes and applications in musical communication over the internet

Álvaro Mendes Barbosa
Pompeu Fabra University - Audiovisual Institute
Estació de França Paseig de Cincurvalacio, 8 Desp. 391
08003 Barcelona, España – http://www.iua.upf.es
tel.: +34 93 542 2588, email: alvaro.barbosa@tecn.upf.es

Joint editing as a group communication process

By joint editing is meant the process of developing a Multimedia Document by more than one author, communicating partly or wholly via computer networks.

In particular, the exchange of partial content which will add up to a finished product, is done via computer networks.

Communication models for distributed operation systems

Common Network Protocols

For the various users in a distributed multi-user application to share the same virtual space and interact, their host machines must communicate with each other via a network.

While there are many different protocols available, two of the most commonly used are the **Transmission Control Protocol (TCP)** and the **User Datagram Protocol (UDP)** .

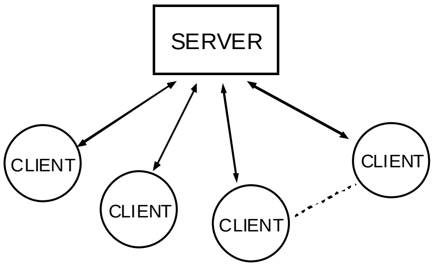
TCP guarantees reliable delivery of messages sent, while **UDP** makes no guarantees, however, TCP is much slower, and therefore less suitable for real-time communication, than UDP.

Real-time applications, like multi-user virtual environments, which require the high-speed of UDP must be careful not to rely on all message transmissions being successful, or use a hybrid TCP/UDP approach to send slow but reliable messages when necessary.

Centralized Network Model [1]

A client/server, or centralized network model, burdens a single host with the task of communicating with each of the clients to determine and report the current state of the system. The server simply maintains the database, while the clients handle computation and rendering.

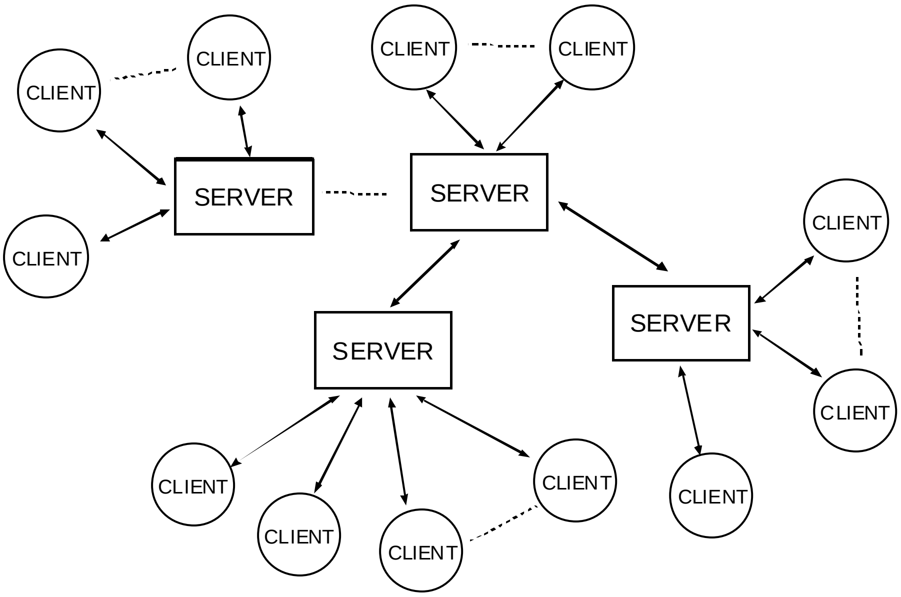
This is typically the easiest approach to implement, but is not scalable. As the number of users increases, the performance between the server and each of the clients decreases.



One way researchers have found to overcome this scaling problem, is to create multiple communicating servers.

Each client communicates directly with the closest (in terms of network distance) server, which takes care of communicating updates with the other servers, who in turn communicate with each of their clients.

This increases the complexity of maintaining a coherent database, but decreases the impact of adding new clients (as long as there are enough servers).

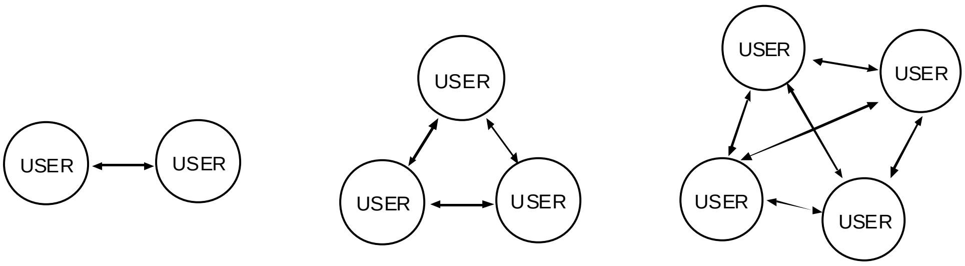


An example of a public domain centralized network model is the Colorado University **Harvest** Proxy System, now called **Squid** [2].

Distributed Network Model [1] [3]

A serverless, peer-to-peer, point-to-point, or distributed network model makes no distinction between clients and servers. Each user maintains a local copy of the database as well as handles computation and rendering.

When changes are made to the database, the user must communicate those change to all other users in the system.



In the most simple case where we only have two users, the process of information transmission is classified as **UNICAST**, since for each update message sent there is only one sender and one receiver.

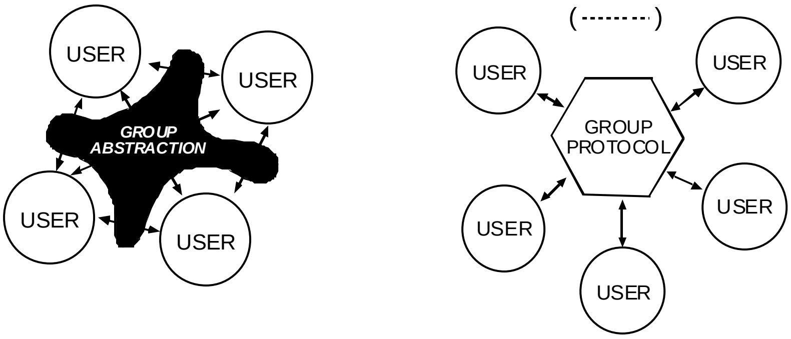
However when increasing the number of users the approach for information transmission must be **MULTICAST**, since each time an update message is sent, it must be sent to all the users in the network.

This approach has a scaling problem because the number of messages being sent by each user steadily increases with the number of users, since each user has the responsibility to physically send the messages to all the others.

Techniques can be employed to help reduce the number of messages sent by determining which peers will be interested in any given update message.

The group communication paradigm

Group communication is a powerful paradigm for joint editing of documents. It introduces the notion of group abstraction whish is based on the concept that one can consider the multiple connections between the users as a hole and in that sense each user only needs to have one communication channel opened with this group protocol.



In this case the process of information transmission is classified as **BROADCAST**, since rather than sending update messages to each of the other users in turn, each user only sends a single message that is received by all other users in the system.

While this results in fewer total messages being sent over the network, broadcast communication has the negative side effect of sending each message to everyone on the network, including those not participating in the virtual environment simulation.

This can cause an overwhelming burden to processes on the network.

An example of a public domain group protocol is the **Spread** Wide Area Group Communication System [4]

Operation Systems

Group systems can operate in different ways:

Transference – The users only interchange information documents, but the work development is done individually, without a common sense of the global information structure (Ex: document exchange by email).

Manipulation – The users can remotely manipulate some of the characteristics of documents in the project, but he cannot change the project structure. (Ex: remote manipulation off an audio track of a song)

Conference – All the users participate and collaborate at the same level in the collaborative event. (Ex: video conference)

Collaboration – All the users actuate at the same level in the collaborative development of a project. (Ex: white boards, or voting and rating systems)

Synchronous and asynchronous modes [5] [6]

Joint editing can be performed in a **synchronous** or **asynchronous** mode.

In the synchronous mode, all the participants are active simultaneously on the common document.

In the asynchronous mode, the participants do not need be active simultaneously, although of course the system must support the situation where several participants happen to be active at the same time.

Experience with computer conferencing systems, which were initially designed to operate in both a synchronous and an asynchronous mode, tells us that almost all usage is in the asynchronous mode.

Practical experience shows that when both modes are available to users, users almost always chose the asynchronous mode for serious interchanges.

In fact, the advantages of the asynchronous mode are often quoted as important reasons for using computer conferencing systems:

- ?? All participants need not find a time where they can all be active simultaneously. If one participant cannot participate at a certain time, that participant can enter the discussion a few days later and still is able to contribute to the outcome of the work.
- ?? You are not forced into rash decisions because of the time limits of a simultaneous meeting, if for example there is a need to think more about a problem, collect facts or make tests, this can be done and new input to the decision can be given the next day, without delaying the whole decision until the next time when all are together simultaneously.
- ?? Some people need more time than others to read and reflect on a problem, because of the asynchronous nature each participant can choose to spend more or less time on the topics of discussion as needed.

The assumption is thus that the same reasons that cause people to prefer an asynchronous mode of operation in computer conferencing are equally valid for joint editing, and that thus systems and standards for joint editing should be designed to work well particularly in asynchronous circumstances.

International Telecommunications Union [7]

Series T Recommendations: Terminal equipments and protocols for telematic services

T.190 – Cooperative Document Handling (Cdh) – Framework and Basic Services

This Recommendation specifies basic services, like storing-and-retrieval or manipulation of documents. Complex services, such as asynchronous document production or joint document presentation, are founded on basic services, and are specified in other Recommendations.

Some of these basic services can be used as stand-alone services, but all of them are candidates to build more complex services, such as joint synchronous editing and joint document presentation/viewing.

T.191 – Cooperative document handling (CDH) – Joint synchronous editing (point-to-point)

This Recommendation specifies one complex service, "joint synchronous editing (point-to-point)", where two users are communicating for the purpose to produce a new document, revise an existing document and view an existing document jointly.

T.192 – Cooperative Document Handling – Complex services: Joint synchronous editing and joint document presentation/viewing

This Recommendation specifies document communication services to be provided on top of existing base standards or profiles, giving constraints on them and rules on how to use and combine them.

Applications for cooperative music composition

Existing Music Oriented **Groupware Tool Kits** currently support Remote and distributed operation of digital sound.
Public domain examples:

The OpenSound Control (OSC) / OpenSound Control Kit [8]

The OSC is a protocol for communication among computers, sound synthesizers, and other multimedia devices that is optimized for modern networking technology.
The OpenSound Control Kit, consists of a library, API, documentation, and other goodies that implement most of the features of OSC and make it fairly easy for developers to support to their applications.

The Synthesis Toolkit (STK) [9]

STK is a set of audio signal processing C++ classes and instruments for music synthesis. STK also has support for multichannel streaming of digital audio

The Network Audio System (NAS) [10]

NAS is a network-transparent, client/server audio transport system.
More than providing the possibility of sending and receiving streams of audio over the network, it also provides easy use of audio over the network with the possibility to play, record and manipulate audio remotely.
The NAS architecture separates the hardware level from the audio application interaction level with a Network Audio Server. It can be described as the audio equivalent of an X server.

The Transis Communication System /
The Caelum toolkit for CSCW [11]

Transis is a multicast communication layer that facilitates the development of fault tolerant distributed applications in a network of machines. Transis supports reliable group communication for high availability applications. Transis contains a novel protocol for reliable message delivery that optimizes the performance for existing network hardware and tolerates network partitioning.

The Transis communication system supports *process group* communication. Groups are conveniently identified by a name (a string) that is selected by the user, such that messages are addressed to the entire group by specifying the group name.

Using the group abstraction, the communication subsystem relieves the user from identifying the targets of messages explicitly, and from finding the network routes to them. In addition, it guarantees *all-or-none* delivery semantics, and handles message losses and transient network failures transparently.

The Transis communication system complies with the concept of MMTS (Multimedia multicast Transport Service) that supports QoS (Quality of Service) group communication, providing synchronization of messages for systems with different QoS requirements.

About Reliability and Quality of Service

Reliability carries different meanings for different applications. For example, in a replicated database setting, reliability means that messages are never lost, and that messages arrive in the same order at all sites. In order to guarantee this reliability property, it is acceptable to sacrifice real-time message delivery: some messages may be greatly delayed, and at certain periods message transmission may even be blocked. While this is perfectly acceptable behavior for a reliable database application, this behavior is intolerable for a reliable video server. For a continuous MPEG video player, reliability means real-time message delivery, at a certain bandwidth; It is acceptable for some messages to be lost, as long as the available bandwidth complies with certain predetermined stochastic assumptions. Introducing database style reliability (i.e. message recovery and order constraints) may violate these assumptions, rendering the MPEG decoding algorithm incorrect.

A desktop and multi-media conferencing tool, is a Computer Supported Cooperative Work (CSCW) application incorporating various activities such as video transmission and management of replicated work space. These activities obviously require different qualities of service, and yet are part of the same application. Furthermore, CSCW applications often need to be fault-tolerant, and need to support smooth reconfiguration when parties join or leave. Groupware is a powerful tool for the construction of fault-tolerant applications, providing reliable multicast and membership services with strong semantics. In this paper, we incorporate multiple quality of service options within the framework of groupware systems. This way, a single application can exploit multiple quality service options, and can also benefit from the groupware semantics.

The use of MIDI in collective composition

Another common approach for the implementation of network joint editing systems is to use the **MIDI** protocol for the cooperative manipulation of audio documents.

TransMIDI [12]

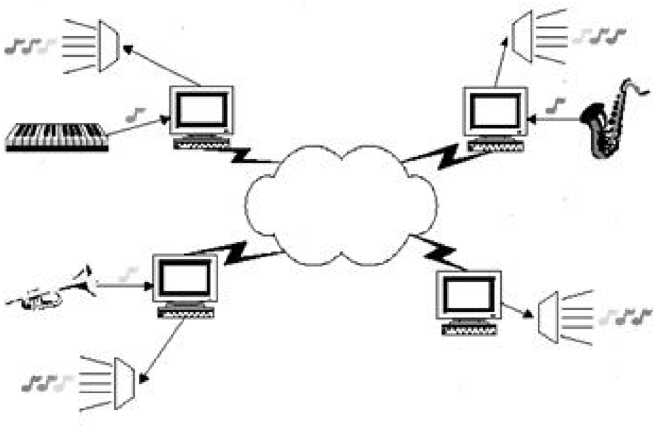
TransMIDI is a system that allows musical performers (and listeners) who wish to play together to organize into multiple session groups.

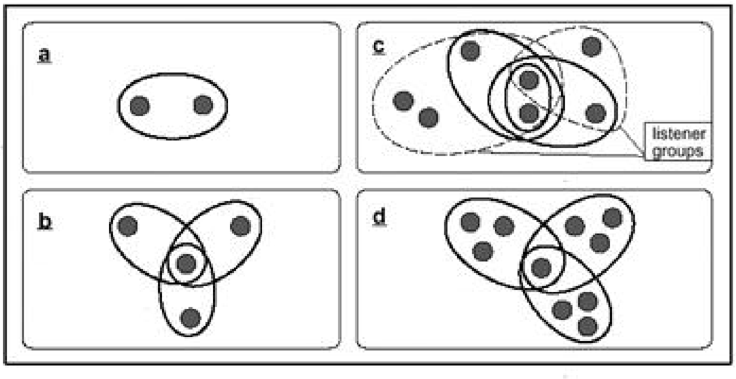
The users interact in real time over the network, and may dynamically join or leave a session group. The players contribute to the session by playing on their MIDI controllers, using General MIDI protocol.

The environment is asynchronous therefore failures of both end systems and communication links are possible.

TransMIDI was implemented using the Transis group communication system for fault tolerance and coordination.

Possible TransMIDI group topologies:





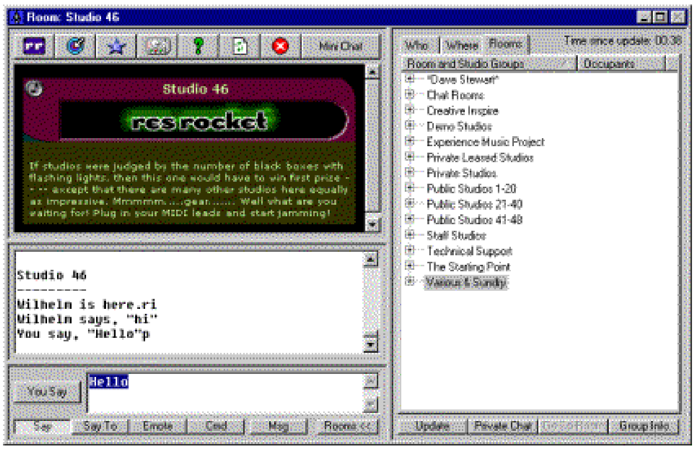
It is possible to have different topologies including the formation of hybrid groups of participants and cores with one or more leaders, and also permitting the access to listener groups.

The ResRocket / Rocket Networks Virtual Workplaces [13]

As a concept the ResRocket software is a distributed system similar to the TransMIDI, however it also has a centralized server that is part of the system.

It is a successfully freely distributed application with a reasonable amount of users forming a community of musicians that actually are creating music cooperatively over the Internet.
The system allows the performers (and listeners) to organize into multiple session groups called Virtual Studios, and to lay down tracks of MIDI in a overall composition.

Somehow it also resembles Chat application where the user can browse to the database of on-line users and join or leave sessions any time, as long as he haves the proper permissions defined by the creators of the session.

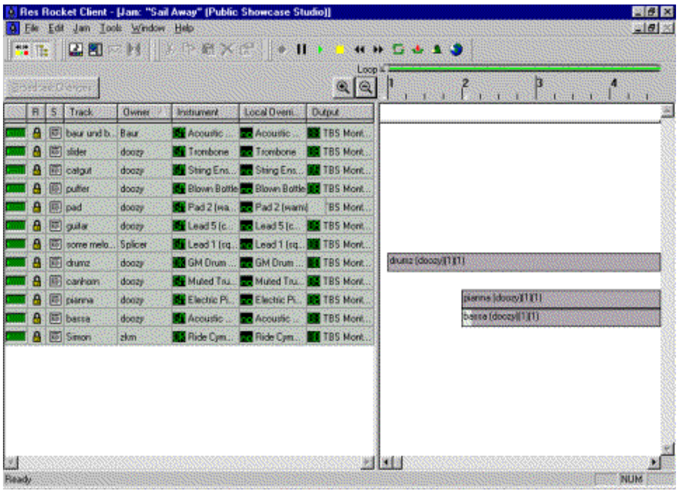


If studios were judged by the number of black boxes with flashing lights, then this one would have to win first prize - - -- except that there are many other studios here equally as impressive. Mmmmm....gear...... Well what are you waiting for! Plug in your MIDI leads and start jamming!

```
Studio 46
----------
Wilhelm is here.ri
Wilhelm says, "hi"
You say, "Hello"p
```

```
Hello
```

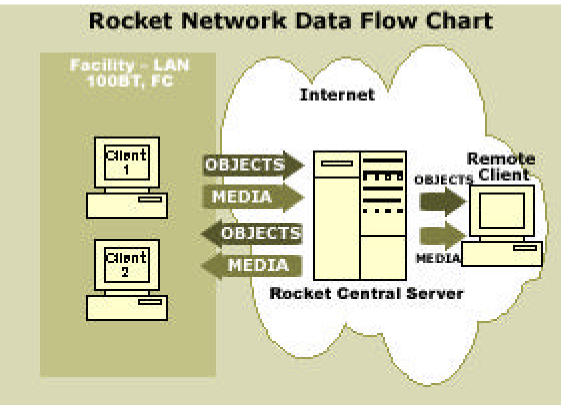
- "Dave Stewart"
- Chat Rooms
- Creative Inspire
- Demo Studios
- Experience Music Project
- Private Leased Studios
- Private Studios
- Public Studios 1-20
- Public Studios 21-40
- Public Studios 41-48
- Staff Studios
- Technical Support
- The Starting Point
- Various & Sundry



Track	Owner	Instrument	Local Overri...	Output
baur und b...	Baur	Acoustic ...	Acoustic ...	TBS Mont...
slider	doozy	Trombone	Trombone	TBS Mont...
catgut	doozy	String Ens...	String Ens...	TBS Mont...
puffer	doozy	Blown Bottle	Blown Bottle	TBS Mont...
pad	doozy	Pad 2 (wa...	Pad 2 (warm)	TBS Mont...
guitar	doozy	Lead 5 (c...	Lead 5 (c...	TBS Mont...
some melo...	Splicer	Lead 1 (sq...	Lead 1 (sq...	TBS Mont...
drumz	doozy	GM Drum ...	GM Drum ...	TBS Mont...
canhorn	doozy	Muted Tru...	Muted Tru...	TBS Mont...
pianna	doozy	Electric Pi...	Electric Pi...	TBS Mont...
bassa	doozy	Acoustic ...	Acoustic ...	TBS Mont...
Simon	zkm	Ride Cym...	Ride Cym...	TBS Mont...

The ResRocket Software can work either in a synchronous or asynchronous mode.

The company Rocket Networks that developed this software as simultaneously introduced the **Rocket Power Audio Software**, that is based on a Centralized Network Model aiming professional recording that supports digital audio and midi (named media and objects), that works in asynchronous way due to latency over the internet, but however is very effective in a joint editing situation where all the users are present at the same time.



The Rocket Power hallows the creation of Virtual Work Places and Synchronization is provided by central server.

The Rocket Power Software is becoming the industry standard for the support to long distance collaboration on digital non editing recording software packages, beeing currently supported by:

- Digidesign's Protools
- Emagic's Logic Audio
- Steinberg's Cubase VST

Collective composition with alternative audio formats

The F@ust Music On-Line (FMOL) [14]

Standalone software program written for W95 and DirectX. Once downloaded and installed it automatically manages all the Internet connections with the database server, with whom it interchanges small scorefiles.

Current FMOL synth engine supports eight stereo audio channels. Each channel is made of a generator (sine, square, sample player, etc.) and three serial processors (filters, reverbs, resonators, etc.), to be chosen by each composer between more than a hundred different synthesis methods, algorithms or variations.

References:

[1] Pulkka, A. (1995). Spatial Culling Of Interpersonal Communication Within Large-Scale Multi-User Virtual Environments. University of Washington, Human Interface Technology Laboratory.

[2] Squid Web Proxy Cache - *http://www.squid-cache.org/*

[3] Obraczka, K. (1997). Multicast Transport Protocol: A Survey and Taxonomy. University of southern California, Information Sciences Institute

[4] Spread Wide Area Group Communication System - *http://www.spread.org/*

[5] T.191 – Cooperative document handling (CDH) – Joint synchronous editing (point-to-point)

[6] Palme, Jacob (1996). Joint Editing as a Group Communication Process. Stockholm University, department of computer and systems sciences.

[7] International Telecommunications Union - *http://www.itu.int/*

[8] The OpenSound Control (OSC) / OpenSound Control Kit -
http://www.cnmat.berkeley.edu/OpenSoundControl/

[9] The Synthesis Toolkit (STK)
http://www-ccrma.stanford.edu/software/stk/

[10] The Network Audio System (NAS)
http://radscan.com/nas.html

[11] The Transis Communication System / The Caelum toolkit for CSCW
http://www.cs.huji.ac.il/labs/transis/transis.html

[12] TransMIDI
http://www.cs.huji.ac.il/~grishac/Abstracts/transmidi.html

[13] The ResRocket / Rocket Networks Virtual Workplaces
{ http://www.resrocket.com/, http://www.rocketnetwork.com/ }

[14] The F@ust Music On-Line (FMOL)
http://www.iua.upf.es/~sergi/FMOL/Project Description.htm