

# A Coordination Architecture for Internet Groupwork

H.-Peter Dommel  
Santa Clara University  
Computer Engineering  
500 El Camino Real  
Santa Clara, CA 95053-0566  
hpdommel@scu.edu

J. J. Garcia-Luna-Aceves  
School of Engineering  
University of California  
Santa Cruz, CA 95064, USA  
jj@cse.ucsc.edu

## Abstract

*This paper discusses a group coordination architecture to support Internet-wide distributed collaboration in the context of legacy Internet protocols. Group coordination in distributed systems and multimedia systems has many faces manifested in a variety of user interfaces and network protocols. To date, no standardized methodology for engineering group coordination protocols exists. We perceive coordination as the third complementary component in the trinity of group-communication services, next to membership and dissemination. With the current surge in e-commerce and Web-leveraged information exchange among users, the need for systems offering better telepresence and interaction capabilities becomes tangible. Services to support distributed group interaction at (near) real-time, with user-specified Quality-of-Service, and at Internet scope are of particular interest in this mosaic of telepresence and remote collaboration. In this paper, we propose a general group coordination architecture for heterogeneous networks, as a framework to leverage the rapid development of group-oriented distributed collaborative applications in the Internet, for example for distance education, distributed scientific simulation or visualization, and similar applications.*

## 1. Introduction

With the advent of networked desktop computers, new opportunities arise to collaboratively utilize distributed information systems. Information processing is not limited anymore to a singular, finite time-sharing domain. Client-server architectures have been a catalyst for new enabling technologies such as cluster computing, where groups of cooperating processors communicate seamlessly with each other. Computer-Supported Cooperative Work (CSCW),

specifically multimedia telecollaboration, promises to deliver more powerful ways of cooperation and coordination, compared to first-generation rudimentary examples of groupware such as email, data exchange, or online coordination of calendars.

From a distributed systems perspective, transaction processing [11] between users and a database server is based on the Commitment, Concurrency, and Recovery (CCR) infrastructure defined by ISO/IEC 10026, entailing atomicity, consistency, isolation, and durability (ACID) properties. Atomicity means that, to an outside observer, either all of the operations are completed, or none of them is executed. Consistency warrants that the operations are performed correctly with respect to the application semantics. The isolation property means that any partial results of the operations composing the atomic action are not accessible before the completion of the atomic action. Finally, durability means that a transaction must endure a communication or an application failure. We observe that currently no such standard exists for building and deploying interaction management protocols for computer-supported cooperative work, independent of such middleware architectures as CORBA [29]. A group coordination architecture may provide a more lightweight repository for crafting applications to feature collaborative services transparently.

From a networking perspective, much research has been invested in recent years in designing more effective group communication mechanisms, however, without much consideration of end-user interactivity. While multicast [5] is a necessity for effective group communication (designed to have a source send a packet only once to the network interface, and multicast routers replicating the packet on its transmission path to multiple receivers), only few collaborative tools build on such efficient group communication to date. In addition, with IP-Multicast, no guarantees are given for reliable or order-preserving delivery of packets, and a message is delivered on a best-effort basis to all members of a multicast group. These shortcomings have spurred much

research on multicast routing, reliable multicast, Quality-of-Service, streaming protocols, and resource reservation. While group membership is tackled by such protocols as IGMP (Internet Group Management) [9] or session directory services such as SDP [13], and multipoint dissemination is handled by multicasting routing and reliable multicast protocols [23] to achieve efficient, reliable packet dissemination, group coordination support for interactive hosts has received only scattered attention in the research literature to date. One such exception is the MCP protocol [45] which uses a conversational abstraction for coordination in form of floor control on top of the transport-layer multicast protocol XTP [42]. The goal is to achieve temporal and causal synchronization among concurrent flows between multiple hosts using a token-passing mechanism. We surmise that a formal and methodological framework for group coordination, providing an integrated view on dissemination, joint access and presentation issues from both a user and network perspective, is still missing.

We define coordination as an interactive scheduling process between two or more users forming a group to achieve joint work goals. Coordination correlates with cooperation, which we understand as the joint acting of individuals for a mutual benefit - in our context the mutual sharing of information for data mining or other forms of data exchange. Coordination and cooperation among users in networked multimedia systems support the process of *multimedia collaboration* [37], which is the actual act of users working together online. A system facilitating remote group collaboration with multimedia support is referred to as a *Collaborative Multimedia Environment* (CME). In contrast to earlier text-centered groupware systems, CMEs process multimodal resources, that is discrete and continuous data such as text, voice, video, music, sensor data from remote instruments, images and graphics. A group is understood as an association of users with a common interest and for a common benefit. The timeframe within which a group meets in a distributed fashion is called a session, defining the modalities of information exchange between users.

Our objective is to lay the foundations for a framework and architecture for group coordination, in particular at Internet scope and for larger groups. The rationale of this architecture is that many networked multimedia applications exhibit similar media semantics and coordination needs, justifying the deployment of such services as middleware at a sub-application layer. Our goal is to achieve a better understanding of the group coordination problem, as a component of Internet multimedia collaboration, bridging the gap between CSCW and networking research. We envision a new generation of CME based on improved group coordination protocols, facilitating scalable and efficient multiparty and multimedia communication. In such systems, groups and individuals can selectively, securely, and effi-

ciently cocreate and disseminate information with improved telepresence and mutual awareness. In contrast to the majority of commercial and experimental CME existing to date, where session coordination and control are performed via distinct servers and reflectors, we look at collaboration as an inherently distributed process enacted collectively by participating hosts.

The rest of this paper is organized as follows: Section 2 outlines related research efforts and discusses coordination from various angles, including dimensions and communication primitives in generic coordination exchanges, as they may typically occur in groupware systems. Section 3 discusses an Internet group coordination architecture, looking at the various system implementation options, offering a design rationale, and the idea to integrate coordination services as a higher-layer function with multicasting at the network and transport layer. Section 4 summarizes the contributions of this report.

## 2. Background

In their coordination theory framework, Malone and Crowston [27] define coordination as the “act of managing interdependencies between activities performed to achieve a goal”, looking at components of actors (people) and agents (computerized procedures), identifying workgoals, mapping goals to activities, and managing interdependencies among actors and activities. They distinguish between generic interdependencies, for instance sequenced or simultaneous actions on shared resources, and domain-specific interdependencies, e.g., specific data elements that must be passed between team members to achieve successful groupwork. Ellis and Rein [8] define groupware as “computer based systems that support two or more engaged in common tasks, providing an interface to a shared environment”. Schmidt and Simone [39] present an empirical characterization of computational coordination mechanisms useful as a general blueprint for the design of coordination protocols, proposing for instance the construction of such a mechanism such that “actors are able to control its execution and make local and temporary modifications of its behavior to cope with unforeseen contingencies”.

A related issue, cooperation, has been a major research focus in distributed artificial intelligence. Lux *et al.* [25] define a language of cooperation between humans and agents to assess how humans and computational agents estimate payoff in resource usage and relinquishment to optimize their strategies. Rosenschein and Zlotkin [38] developed a framework on negotiation protocols among agents, where utility metrics define strategies and outcomes in cooperative resource allocation. Decker [4] understands coordination as the “act of managing interdependencies between activities and dissects coordination processes into the specifica-

tion of shared goals, solution planning, and task scheduling, starting with signals for attention, acceptance or refusal of a connection, information transfer, acknowledgment, and teardown. Blackboard architectures [16] have been extensively studied in artificial intelligence as a mechanism for sharing information among expert system modules for purposes such as multiple-task planning.

Holt [17] discusses a coordination language based on Petri nets, as an element in a larger theoretical framework referred to as coordination mechanics, with the idea to establish relationships between tasks and products. Winograd [44] presents an analysis of group action based on speech act theory and discuss the Coordinator tool to support users in keeping track of requests and commitments to each other. Rabin [35] defines *choice coordination* as the problem to design a wait-free protocol for  $n$  concurrent processes causing all correct processes to agree on choosing one out of  $k$  possible alternatives. Communication is performed via registers associated with choices and the question is, how choice coordination can be solved best deterministically. If a small probability of failure is acceptable, a protocol on a small alphabet can solve the problem, however, a “perfect” protocol is very expensive.

Joung and Smolka [22] proposed a taxonomy of languages describing multiparty interaction, discerning between interactions with fixed vs. variable participants, conjunctive vs. disjunctive parallelism in the execution of interactions, synchronous vs. asynchronous task execution, and biparty vs. multiparty scenarios. For synchronous systems, multiparty interaction with disjunctive parallelism is accordingly NP-complete, and biparty interaction is solvable in polynomial time, when using conjunctive and disjunctive parallelism in isolation.

### 2.1. Network-centric Coordination

Coordination can be manifested explicitly, e.g., through signals like acknowledgments, or implicit through interpretation of the behavior and responses of partners. The process of coordinating activities can hence be either transparent or tangible to applications. Coordination revolves around resource allocation in terms of scheduling and availability. Allocating a resource at time  $t$  may create contingencies for resources and users at time  $t + 1$ . Coordinative stages are therefore often non-separable.

Messaging may be based on ad hoc input or use a predefined vocabulary. Partners may be located at the same or different places, and coordinate at the same or different times, i.e., coordination processes may operate synchronously or asynchronously in the relative timing of partners. Participants are either active initiators, or passive receivers. Coordination flow may be unicast and either half-duplex or full-duplex, or multi-way, using broadcast and filtering, or

multicast.

### 2.2. Dimensions

Coordination is intertwined with conflict [18], and its prediction, avoidance, or resolution. Conflict can arise before the allocation of shared resources, implying contention or race conditions, or during and after resource allocation, creating collisions, congestion, inconsistencies or information loss. Conflicts are resolved by blocking or queuing contending processes or by aborting them. A specific conflict prevention or resolution scheme may be unfair in that it benefits specific users and neglects others.

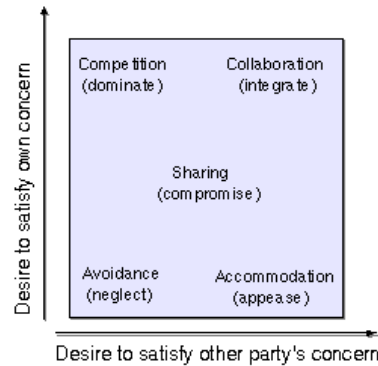


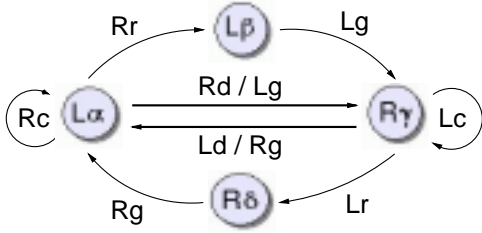
Figure 1. Conflict resolution in terms of own vs. group concerns.

Figure 1 [7] depicts the strategies of conflict resolution in the light of personal vs. group interests. We can partition the set of  $N$  users into two subsets, cooperating and conflicting users  $\{u_i\} = \{u_j\} \cup \{u_k\}$ , for  $i, j, k = 1, \dots, N$  and  $j \neq k$ . The objective of group coordination is to minimize the conflict set, e.g., by establishing effective turn-taking protocols.

### 2.3. Dissection of Coordination Processes

An abstraction for the flow of turn-taking messages between a local user  $L$  and a remote user  $R$  is depicted in Figure 2. The model can be interpreted twofold, depicting the flow of coordination primitives for control, or activity states and transitions among the two parties:

Interpreting the abstraction as a state transition diagram, the four nodes represent control or activity states of users  $L$  and  $R$ , and the transition labels represent the events or control signals triggering the state transitions. This model is a basic two-party rendition for a generic coordination mechanism without moderation and can be generalized to the  $n$ -party case with maximally  $\frac{N(N-1)}{2}$  pairwise flows. Turn-taking results in a  $L - R - L - R - L - R \dots$  hand-off



**Figure 2. Abstraction of coordination and control flow among two parties.**

pattern for a particular coordination primitive, with varying holding times per turn. Each node may assume the role of source or receiver at the same time for multiple, concurrent interactions.

The set of states and events models the *control flow* in two-party interaction, with  $\alpha$  and  $\gamma$  denoting “continuing the activity”, and  $\beta$  and  $\delta$  denoting “concluding and shifting the activity”.  $Rr$  denotes the event “ $R$  requests”,  $Rg$  represents “ $R$  gives up” (alternatively, “ $R$  grants”),  $Rc$  indicates “ $R$  continues”, and  $Rd$  “ $R$  demands” (symmetric for  $L$ ). The events  $r, g, c, d$  indicates state transitions to trigger the respective activities. Certain activities may be forbidden or invalid in a given combination, depending on media types.



**Figure 3. Conceptual model for coordination turn structure.**

Figure 3 depicts the generic coordination timeline of a turn on some resource  $R$  during session time  $t$ , consisting of the contention period  $X$ , the activity or busy period  $B$ , and an idle time  $I$ . The contention period may overlap with the busy and idle periods of the previous person in turn, and models the time, in which the previous holder, a moderator, or a mechanism determine the next turn taker. The activity time models the lifecycle of a coordinative turn. The idle time models the handover period, in which the next collaborator decides to issue a requesting coordination signal and the current user gives up his or her turn. Turn-taking relevance points (TRPs) mark the handover points from one coordination interval to the next one. While this model is fairly generic and simplistic, it allows to identify the basic coordination functions in group interaction, modeled with a point-to-point case.

### 3. Coordination Architecture

We discuss now relevant concepts for the implementation of a group coordination architecture. Navarro *et al.* [30] identify the following requirements for CSCW systems: support for information sharing, communication, tailorability, and transparency of organization, time, views, and activities. Among the many systems built to date, we can distinguish between architectures where computers and video enrich real-world media spaces [32], such as CAVECAT [3]; systems fusing virtual and face-to-face groupwork, such as the DIGITALDESK [43], CLEARBOARD [20], or the TEAMWORKSTATION [21]; and conferencing and collaboration based on networked computers [41]. The latter being our focus, it is not clear to date where group coordination services should be deployed in the network protocol stack to fulfill these requirements, under the constraints of individual, group, system, and network behavior. Previous coordination protocols have been implemented at the transport layer, in the session layer as middleware component, or integrated into an application itself.

We distinguish between *collaboration-unaware*, *collaboration-aware*, and *collaboration-transparent* architectures. Unaware systems offer no intrinsic collaborative capabilities, and require external mechanisms to tap into, filter, and multiplex operations on shared resources of the session. Aware systems offer inherent collaborative functionality visible to the user and are of limited use when operated stand-alone. Bentley and Rodden [2] discern between view-level sharing on identical information with different views, and object-level sharing using replication for independent manipulation. Collaborative systems can be *tightly* or *loosely coupled* [33]. Tight coupling presumes explicit membership control of participants and conference control enforcing a formal session style with floor control, which is for instance desirable with IP telephony. Loose coupling is based on a “light-weight” session model [10] and does not require explicit membership or conference control to synchronize media and regulate interactions. This model, implemented with the MBONE conferencing tools [26], allows for anonymous, large-scale conferences, and benefits from cross-media coordination mechanisms, since users want to see their activities manifested in coherent responses by all tools. Mechanisms to bridge the gap, such as a moderated blackboard for posting questions and stimulate interaction have been built for the MBONE [28].

#### 3.1. Components

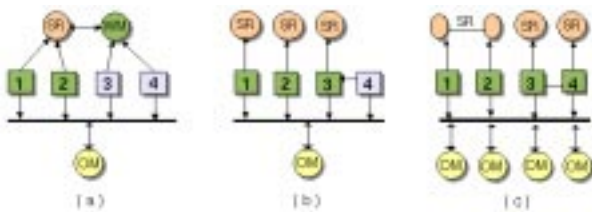
At the highest level of abstraction, a prototypical collaboration environment contains five building blocks:

- A conference manager for orchestrating session setup,

membership tracking (invitation, joining, leaving) and teardown (OM).

- A coordination manager for exercising group coordination services such as floor control on communication channels and network resources in the shared workspace (CM).
- A workspace or window manager, which integrates the local, private workspace, and the public, shared workspace, where the latter can be rendered in local, independent views, or in synchronization with remote hosts (WM). The underlying information base for WM may be local or obtain data from remote servers.
- One or more shared resources, such as an application or device (SR).
- A network interface linking the above components.

The CM may exercise control by intercepting messages sent between WM and SR, or by processing control requests from the various nodes. For implementing group coordination management, we have a choice of centralized or distributed coordination management, or hybrids of both. OM and CM may be centralized, a hybrid of centralization and distribution, or purely distributed, as depicted as in Figure 4.



**Figure 4. Coordination configurations: (a) Centralized; (b) Hybrid; (c) Distributed.**

The dark-shaded squares (hosts), in contrast to light-shaded squares, have an integrated WM and have the option of independent views. In *central sequencing*, all changes to shared resources are funneled through a single host (“coordination server”), with the benefit of predictable delays, simple implementation logic, and consistency.

Pendergast [34] distinguishes between three types of central processing, terminal linking, centralized data, and distributed data. Terminal linking yields strict WYSIWIS, centralized data lets hosts obtain data through a central server, but process them locally, and in distributed data all shared information is resident at the hosts’ local storage and a central node sequences all operations. The first and second solution offer inherent consistency, but incur high network traffic. The third solution implies partial or full replication of data in a federation of servers or hosts, and is the

predominant approach for collaborative systems, with the advantages of low network traffic and superior response, but difficulty in maintaining consistency.

In Figure 4 (a), all hosts sequence operations through a central *OM*, with hosts 1 and 2 accessing a resource directly with WYSIWIS, whereas hosts 3 and 4 use relaxed WYSIWIS by employing a WM for independent views. In the hybrid approach (b), host 4 communicates solely through host 3 to access *SR*, whereby hosts 1 – 3 maintain their own replica of *SR* and sequence exclusively through a central *OM*. In the distributed case (c), hosts 1 and 2 possess partial information on a resource *SR* and are linked to indicate exchange of information to mutually complete their picture of the resource. Hosts 3 and 4 access *SR* as independent objects, which exemplifies the special case suggested also by Pendergast [34] that no special coordination is needed to manipulate state information. In all cases, we assume for simplification that *OM* contains *CM*.

Most existing conferencing and collaboration systems can be subsumed under one of these models. In a centralized scheme, all hosts tap into the same application and use a single CM, with the shortcoming that hosts cannot be decoupled and depend on CM and the application server as a performance bottleneck under high load and point of failure. Centralized coordination incurs no integration cost. In a hybrid scheme, each host has its own replicated information base coordinated through a central coordination protocol. Hosts from the same local network partaking in a wide-area session communicate with a local conference server through a local bus architecture, which attaches to another conference server on the conference backbone. In a distributed architecture, CM, WM, and SR are all host-individual, and coordination is peer-based using a conference bus, with the benefits of flexibility in session organization, load-balancing, scalability, and fault-tolerance.

### 3.2. Design Rationale

We postulate henceforth a coordination architecture based on the following requirements: simplicity of implementation and maintenance; scalability in the number of users and hosts [40]; security with regard to the exchange of coordination information and data; extensibility for new resources and session models; efficiency in coordination, concerning low latency and protocol state overhead; reliability with regard to failures of hosts, resources, and network elements; persistence of coordination information at hosts despite the ephemeral nature of coordination signals; and interoperability between heterogeneous platforms.

The maintenance and distribution of coordination state information should be *partially or fully distributed* to avoid, for instance, the problem that a single coordination server becomes a bottleneck and single point of failure for a ses-

sion. Distributed session orchestration also allows for continuation within partitions of a split session, in case that one or more links between hosts fail. Session information should be *logged* at several hosts for the case that disconnected sites resurface and attempt to rejoin a session, retrieving updates on the collective session state from one of the active session partitions. Shared workspaces should be *asymmetric*, that is, each host is allowed to maintain its own individual view on the shared resources, but may synchronize itself a WYSIWIS state, if required. Session updates, rather than flooding the network, should be sent in increments between hosts, encoding the difference in previous and current work states.

A *hierarchical* host organization and representation of the workspace allows for more efficient naming, addressing, and state keeping of large sessions, and inheritance of globally valid session attributes. A hierarchy models face-to-face meetings better, because users may want to temporarily step out of a larger session to conduct a side-chat without having to define a new multicast group. Sessions should allow for various orchestration models, from tight control to loose control, to cater to the need of registered, formal meetings and informal meetings. Ideally, coordination services would be based a common structuring principle, such as a common dissemination and control geometry. Most available literature treats network coordination services as a black box without providing details on operation or implementation.

Figure 5 shows the layering of host-based services in a networked multimedia system, with coordination services located between the application layer and the media services and network support layer. Synchronization refers to playback synchronization of media streams [24] and is attributed to the media layer, in contrast to activity synchronization between users [1].

<b>Application</b>	Toolkits, GUI, Device Control, Scripts, Object Access
<b>Coordination</b>	Floor Control, Synchronization, Ordered Multicast, Security
<b>Media</b>	Mixing, Scaling, Tuning, QoS, Payout
<b>System</b>	File System, Replication, X-Server, Access Control
<b>Hardware</b>	Codecs, Network Interface, I/O

**Figure 5. Group coordination architecture.**

While floor control handles coordination, session control keeps track of membership, start-up and tear-down of the communication paths, and the use of compatibility issues

with particular media. Session control complements coordination mechanisms such as floor control and performs membership management [36], directory services [13], announcement [12], invitation [14], and teardown. Integrated session and floor control services have been referred to as conference control [15]. Typical membership functions fall into four categories: registration calls such as invite, initiate, join, etc.; modification of status and authorization; polling of information about the remote or local status; and termination, including functions such as leave, withdrawal, or expel. The group coordination architecture put forward in this paper assume communication based on message passing in a packet-switched network.

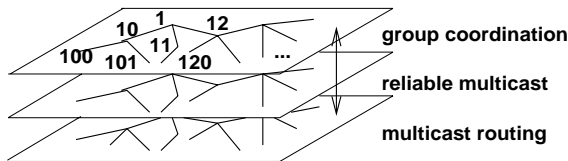
### 3.3. Multicast-supported Coordination

Group coordination relates to multicast routing and reliable multicasting [31], because control messages must be routed among hosts in the control tree built for managing session interactions. Failed control directives must be retransmitted, similar to packet loss recovery in reliable multicast. The IP-multicast model lacks support for more sophisticated addressing among hosts in the context of group interactivity and coordination. Shortlived subsets of members of multicast groups cannot interact with each other without affecting the session as a whole or defining an entirely new group. We observe that the standard model for IP-multicasting is not sufficiently powerful to support the full spectrum of group dynamics in distributed multimedia applications. As a stepping stone towards more elaborate group coordination support from the network layer up, we discuss the idea of integrating multicast routing, reliable and ordered multicasting and multicast-enabled group coordination.

A multicast tree is either a shortest path tree, which is a directed tree by one source reaching all members of the multicast group, or a shared tree, which is constructed for a group and shared by all sources. The multicast delivery tree is constantly pruned or extended by a multicast routing protocol such as DVMRP, CBT or PIM-SM (cf. [19] for a general review), reflecting the current subscription state to multicast groups and the presence of adjunct network resources. Multicast group membership is provided by a membership protocol such as IGMP [9].

Figure 6 depicts the relationship between a group coordination mechanism and underlying multicast services, divided into end-to-end reliable multicast as a transport service and multicast routing at the network layer. The coordination layer is hence a cohesive middleware substrate, on which various distributed applications can be built.

Consider a scenario where a multicast group MC consists of members collaborating on a distributed workspace, where specific subsets of members are inter-



**Figure 6. Multicast and group coordination protocol layering.**

ested in receiving certain information updates concerning the workspace. Multicast may reduce bandwidth consumption for updates within MC, but still impact members that do not want to be involved in certain transactions. A straightforward solution would be to define new multicast groups for such coteries, however, such groups may overlap, members may get duplicate updates, and for a large MC result in a very large amount of small subgroups with high management overhead. It is hence desirable to allow transmissions to subgroups of multicast groups and for data to be subcast on a per-packet basis.

For such highly interactive group work, the per-source tree model would require hosts to join a new tree per turn, and subsequently tear down the temporary multicast tree, which is impractical. Source trees are suited for a scenario where one source incites a long-lived transmission to other session members, and no further individual source trees in the session must be built. In an alternative model, a single shared tree is constructed in the beginning of a session and hosts join the session by being added into the tree. The tree is built and managed in close resemblance to the routing and reliable multicast topology. For multimedia collaboration with frequently switching sources, a shared tree is a more effective solution, with the shortcoming that delivery paths in the shared tree may be suboptimal and affect the latency of continuous media.

When a host become active, it transmits its data to its children, if the target hosts are located in its subtree, or to its parent host, if the target is located elsewhere in the tree. Each transmission involves therefore only as many hosts as the branching factor of the tree indicates. In case of stale links or failed hosts, many heuristics have been proposed on how to reconstruct and optimize shared trees. This motivates a refined intra-group addressability service to allow selective multicast of control information and data to subgroups of large groups on a per-turn or per-packet basis. This approach permits more practical implementation and inherent adaptation to membership and link changes in the shared multicast infrastructure. Furthermore, such an approach enables important functionality for collaboration of small user groups within multicast groups, such as selective subgroup addressing and subcasting. Integration of floor control with tree-based reliable multicasting [6] is one pos-

sible exemplification of this paradigm.

## 4. Conclusion

Our objective is to motivate the foundations of a group coordination architecture for “Internet groupware”. We have discussed various approaches for such architectures and reasoned for a distributed service model. We dissected dissecting cooperative turns among distributed users, with the goal to characterize the foundation of a “coordination API” which can be used by collaborative applications to negotiate and exchange coordination information in the process of distributed groupwork. We proposed the idea of integrating group coordination services with the IP-Multicast infrastructure and framework, which is currently gradually deployed on the Internet.

The central idea is to deploy coordination services on top of reliable multicast and achieve cohesive operation across the same logical network topology. This approach allows to eliminate the need to build a separate control structure for tracking, routing, withholding, or forwarding coordination directives. We view coordination as the third integral component in group-oriented communication services in the Internet, complementing group dissemination and membership protocols and enriching the current IP-Multicast service model. With steadily improving networking technology, workgroup computing and collaboration technology such as video conferencing and application sharing, particularly at Internet scope and with streaming media support, may represent the next paradigm for “killer applications”.

## References

- [1] I. F. Akyildiz and W. Yen. Multimedia group synchronization protocols for integrated services networks. *IEEE J. on Sel. Areas in Comm.*, 14(1):162–173, Jan. 1996.
- [2] R. Bentley, T. Rodden, P. Sawyer, and I. Sommerville. Architectural support for cooperative multiuser interfaces. *Computer*, pages 37–46, May 1994.
- [3] S. A. Bly, S. R. Harrison, and S. Irwin. Media Spaces: Bringing people together in a video, audio and computing environment. *Comm. of the ACM - Special Issue on Multimedia in the Workplace*, 36(1):28–47, Jan. 1993.
- [4] K. S. Decker and V. R. Lesser. Designing a family of coordination algorithms. Technical Report TR 94-14, Univ. of Massachusetts, Amherst, MA, Aug. 1995.
- [5] S. Deering. Host extensions for IP multicasting. RFC-1112, August 1989.
- [6] H.-P. Dommel and J. J. Garcia-Luna-Aceves. Efficacy of floor control protocols in distributed multimedia collaboration. *Cluster Computing J.*, 2(1):17–33, 1999.
- [7] S. Easterbrook. *CSCW: Cooperation or Conflict?* Computer Supported Cooperative Work. Springer-Verlag, London, 1993.

- [8] C. A. Ellis, S. J. Gibbs, and G. L. Rein. Groupware - some issues and experiences. *Comm. of the ACM*, 34(1):38–58, Jan. 1991.
- [9] W. Fenner. Internet Group Management Protocol, Version 2. RFC 2236, Nov. 1997.
- [10] S. Floyd, V. Jacobson, S. McCanne, C.-G. Liu, and L. Zhang. A reliable multicast framework for light-weight sessions and application level-framing. In *Proc. ACM SIGCOMM*, pages 342–356, Cambridge, MA, Aug./Sep. 1995.
- [11] J. Gray and A. Reuter. *Transaction Processing: Concepts and Techniques*. Morgan-Kaufmann, San Mateo, CA, 1993.
- [12] M. Handley and V. Jacobson. Session Announcement Protocol. Internet Draft draft-ietf-mmusic-sap-v2-03.txt, November 1997.
- [13] M. Handley and V. Jacobson. SDP: Session Description Protocol. RFC 2327, April 1998.
- [14] M. Handley, H. Schulzrinne, E. Schooler, and J. Rosenberg. SIP: Session Initiation Protocol. RFC 2543, March 1999.
- [15] M. Handley, I. Wakeman, and J. Crowcroft. The Conference Control Channel Protocol (CCCP): A scalable base for building conference control applications. In *Proc. ACM SIGCOMM*, pages 275–287, Cambridge, MA, Aug./Sep. 1995.
- [16] B. Hayes-Roth. A blackboard architecture for control. *Artificial Intelligence*, 26(3):251–321, July 1985.
- [17] A. W. Holt. Diplans: A new language for the study and implementation of coordination. *ACM Trans. on Office Information Systems*, 6(2):109–125, Apr. 1988.
- [18] P. H. Hughes. Going off the rails: Understanding conflict in practice. In *S. Easterbrook (Ed.) - CSCW: Cooperation or Conflict?*, pages 161–169. Springer, 1993.
- [19] C. Huitema. *Routing in the Internet*. Prentice Hall, Englewood Cliffs, NJ, 1995.
- [20] H. Ishii, M. Kobayashi, and J. Grudin. Integration of interpersonal space and shared workspace: ClearBoard design and experiments. In *Proc. CSCW'92*, pages 33–42, Nov. 1992.
- [21] H. Ishii and N. Miyake. Toward an open shared workspace: Computer and video fusion approach of TeamWorkStation. *Comm. of the ACM - Special Issue on Collaborative Computing*, 34(12):36–50, Dec. 1991.
- [22] Y.-J. Joung and S. A. Smolka. A comprehensive study of the complexity of multiparty interaction. *J. of the ACM*, 43(1):75–115, Jan. 1996.
- [23] B. N. Levine and J. J. Garcia-Luna-Aceves. A comparison of known classes of reliable multicast protocols. In *Proc. IEEE Int. Conf. on Network Protocols*, pages 112–21, Columbus, OH, Oct. 1996.
- [24] T. Little and A. Ghafoor. Synchronization and storage models for multimedia objects. *IEEE J. on Sel. Areas in Comm.*, 8(3):413–427, Apr. 1990.
- [25] A. Lux, P. de Greef, F. Bomarius, and D. Steiner. A generic framework for human computer cooperation. In *Proc. IEEE Int. Conf. on Intelligence and Cooperative information Systems*, pages 89–97, Rotterdam, Netherlands, May 1993.
- [26] M. R. Macedonia and D. P. Brutzman. Mbone provides audio and video across the Internet. *Computer*, 27(4):30–36, Apr. 1994.
- [27] T. W. Malone and K. Crowston. Toward an interdisciplinary theory of coordination. Technical Report CCS TR 120, MIT Sloan School of Management, Cambridge, MA, 1991.
- [28] R. Malpani and L. Rowe. Floor control for large Mbone seminars. In *Proc. ACM Multimedia*, pages 155–163, Seattle, WA, Nov. 1997.
- [29] K. Minsoon, H. Sungjune, J. Sangjin, H. Sunyoung, et al. Scalable and reliable synchronous collaboration environment on CORBA using WWW. In *Proc. High-Assurance Engineering Workshop*, Washington, DC, Aug. 1997. IEEE.
- [30] L. Navarro, W. Prinz, and T. Rodden. Towards open CSCW systems. In *Proc. IEEE 3rd Workshop on Future Trends of Distributed Computing Systems*, pages 4–10, Taipei, Taiwan, Apr. 1992.
- [31] K. Obraczka. Multicast transport protocols: a survey and taxonomy. *IEEE Communications Magazine*, 36(1):94–102, Jan. 1998.
- [32] M. H. Olson and S. A. Bly. The Portland experience: a report on a distributed research group. *Int. J. of Man-Machine Studies*, 34(2):211–228, Feb. 1991.
- [33] J. Ott, C. Perkins, and D. Kutscher. Requirements for local conference control. Internet Draft draft-ott-mmusic-mbusreq-00.txt, June 1999.
- [34] M. O. Pendergast. Multicast channels for collaborative applications: Design and performance evaluation. *Computer Communication Review*, 23(2):25–38, April 1993.
- [35] M. O. Rabin. The choice coordination problem. *Acta Informatica*, 17:121–34, 1982.
- [36] B. Rajagopalan. Consensus and control in wide-area group communication. Technical report, AT&T Bell Laboratories, Holmdel, NJ 07733-3030, Oct. 1993.
- [37] P. V. Rangan and H. M. Vin. Multimedia collaboration as a universal paradigm for collaboration. In *Multimedia - Principles, Systems and Applications*, pages 3–15. Springer-Verlag, Apr. 1991.
- [38] J. S. Rosenschein and G. Zlotkin. *Rules of Encounter - Designing Conventions for Automated Negotiation among Computers*. The MIT Press, Cambridge, MA, 1994.
- [39] K. Schmidt and C. Simone. Coordination mechanisms: Towards a conceptual foundation of CSCW systems design. *Computer-Supported Cooperative Work*, 5(2-3):155–200, 1996.
- [40] E. M. Schooler. The impact of scaling on a multimedia connection architecture. *Multimedia Systems*, 1:2–9, 1993.
- [41] E. M. Schooler. Conferencing and collaborative computing. *Multimedia Systems J.*, 4(5):210–225, Oct. 1996.
- [42] W. T. Strayer, B. Dempsey, and A. Weaver. *XTP: The Xpress Transfer Protocol*. Addison Wesley, 1992.
- [43] P. Wellner. Interactive with paper on the DigitalDesk. *Comm. of the ACM - Special Issue on Computer Augmented Environments*, 36, No. 7:86–97, July 1993.
- [44] T. Winograd. A language/action perspective on the design of cooperative work. *Human-Computer Interaction*, 3(1):3–30, 1987/88.
- [45] R. Yavatkar and K. Lakshman. Communication support for distributed collaborative applications. *Multimedia Systems J.*, 2(2):74–88, Aug. 1994.