The Effects of Network Delays on Group Work in Real-Time Groupware

Carl Gutwin

Department of Computer Science, University of Saskatchewan

Saskatoon, Saskatchewan, Canada, S7N 5A9

carl.gutwin@usask.ca

Abstract. Network delays are a fact of life when using real-time groupware over a wide area network such as the Internet. This paper looks at how network delays affect closely-coupled group work in real-time distributed groupware. We first determine the types and amounts of delay that can happen on the Internet, and then identify types of collaborative interactions that are affected by delay. We then examine two interaction types more closely: predicting others' movements, and coordinating shared access to artifacts. We carried out experiments to measure the effects of two kinds of delay (latency and jitter). When these interactions are isolated and repeated, we found that even small delays can lead to significant increases in completion time and errors. Although people in real-world tasks are often able to adapt their actions to accommodate network delays, we conclude that designing groupware to minimise the effects of delay can improve usability for closely-coupled collaboration.

Introduction

The goal of real-time distributed groupware is to allow people in different places to work together, as naturally and simply as they do in face-to-face settings (e.g. Stefik et al 1987, Tatar et al 1991). Consequently, much attention has been paid to the design, implementation, and evaluation of multi-user software, and considerable gains have been made in improving groupware usability (e.g. Beaudoin-Lafon and Karsenty 1992, Gaver 1991, Gutwin and Greenberg 1998). However, there is one aspect of groupware that that has not received much consideration from CSCW, but one that can cause usability problems: the underlying network.

Distributed operation introduces communication delays between the computers running a groupware application, and in wide area networks, these delays can be substantial and unpredictable. Most people who have used groupware over the Internet have experienced delays of one kind or another, particularly in the real-time display of other people's movements and actions. For example, many shared workspace systems provide each person with a telepointer; under delay conditions, a person's telepointer may seem to get stuck as it moves across the screen, or may be out of synchrony with accompanying speech. Delays are especially noticeable when people work in a closely-coupled fashion and need continuous and up-to-date feedback about another person's activities (Salvador et al 1996).

The goal of this research is to investigate the effects of network delays on the usability of groupware systems. We are primarily concerned with groupware that involves visual artifacts in a shared workspace, and with tasks that involve close coupling amongst the group.

In this paper, we look at three questions: what kinds of delay happen in groupware systems, what magnitude of delay occurs on a real-world wide area network such as the Internet, and how different types and magnitudes of delay affect groupware usability. The main part of the paper reports on a study of how delay affects specific kinds of closely-coupled interactions. The results of these investigations will assist developers in designing and building groupware that is appropriate for the network environment in which it must operate.

Types of delay: latency and jitter

Groupware applications communicate information by sending messages to one another. For discrete information, such as commands and model updates, the order of the messages is important, but timing of a message is independent of other messages. For continuous real-time information, however, messages must be considered as part of a temporal stream. Telepointer positions and other information about people's movements and activities are examples of this type of data stream. Continuous streams have temporal dependencies, in that the timing and pacing of the stream has an effect on how the stream is interpreted. Streams are therefore sensitive to two kinds of delay that can be caused by network communication: *latency* and *jitter* (e.g. Fluckiger 1995, Tannenbaum 1996). These are described below and illustrated in Figure 1.

Latency is the lag between the sending and the receiving of a message (see Figure 1b). Since messages cannot be delivered instantly, latency will always exist to some degree. Even in a face-to-face conversation, there is a (usually unnoticeable) communication latency due to the speed of sound. In network communication, substantially larger latencies exist, caused by the transmission time of the network medium, slowdowns due to traffic, the overhead of routing messages, and by the processing time required to unpack and process messages.

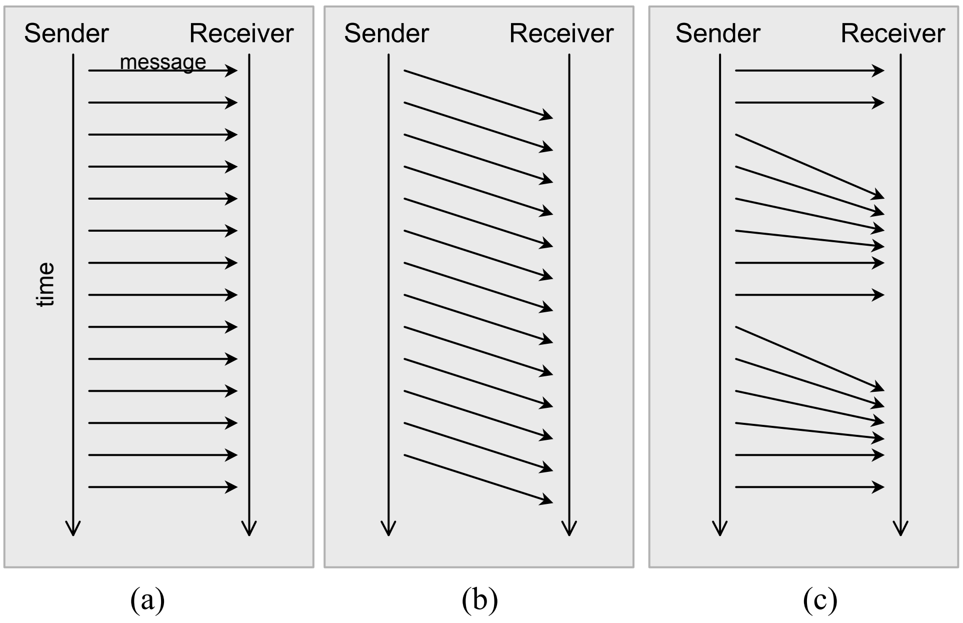

Figure 1. Time-series of messages from a sender to a receiver, with (a) no latency or jitter, (b) latency but no jitter, (c) jitter but no latency

From the groupware user's perspective, latency means that a data stream (such as another person's telepointer motion) is late compared to when it was produced. The motion of the telepointer will look normal in other respects, however; if the user has no indicator of when the motion started, then the latency will be difficult to detect. Problems begin to occur with latency in two situations. The first happens when two streams (such as voice and telepointer motion) are supposed to be synchronised, but are transmitted with different latencies. The second happens when interaction involves taking turns. Previous research suggests that turn-taking audio interaction such as telephone conversations become difficult to coordinate when latency is greater than about 300ms (e.g. Scholl 2000).

Jitter, in contrast, affects the pacing of the stream rather than its lateness. Jitter is variance in transmission time, and measures whether the amount of time between two messages at the receiving end is the same as the time between them when they were sent (see Figure 1c). For example, if messages are sent at 10ms intervals, but the receiving interval varies from 10ms, then there is jitter in the transmission. Jitter does not exist in face-to-face communication, because all the data in an utterance or a movement travel at exactly the same speed. In networked groupware, however, each message in a stream is encoded as an independent packet; two consecutive messages may be sent to the destination on different routes, or may encounter different overheads and traffic conditions along the way. Furthermore, a message may be lost altogether, and if the transmission protocol enforces in-order delivery, all messages behind the lost message must wait unprocessed while it is resent from the source. These factors imply two means of characterising jitter: size of delay, and percentage of messages that are delayed.

From the user's perspective, jitter appears as halting or jerky movement: a moving telepointer, for example, will appear to stick when a message is delayed, and will then catch up when messages begin flowing again. Research into the delivery of streaming audio and video suggests that people are able to notice even small amounts of jitter (tens of milliseconds), and quickly become annoyed by larger amounts (e.g. Cisco 2000, Scholl 2000). Audio and video applications strive to reduce jitter to zero, usually by buffering the stream before playback begins. Buffering (also called smoothing) reduces jitter by increasing overall latency; the stream starts later but plays smoothly.

The effects of latency and jitter on interaction in real-time groupware are not well known. Limited previous research has considered delay effects on collaborative task performance in 3D virtual environments (e.g. Park and Kenyon 1999, Vaghi et al 1999). One study suggests that performance is negatively affected by latencies of only 200ms (Park and Kenyon 1999), but the "two-person thread the needle" task used differs substantially from tasks in most 2D shared-workspace groupware systems. A qualitative study (Vaghi et al 1999) saw a variety of strategy changes in the presence of latencies from 200ms - 1000ms, and suggests that the task became difficult at about 500ms latency.

The next step for the current investigation is in determining a rough idea of the magnitude of latency and jitter that can be expected for a groupware system working across a real-world network such as the Internet.

Delay magnitude on the Internet

Real-world groupware applications encounter different amounts of latency and jitter, depending upon where and when they use the system. Delays on the internet depend on the power of the client machines, the bandwidth of the network segments, the distance that messages must travel, the number of routers that the message goes through, and the current traffic level (e.g. Acharya and Saltz 1996). This means that it will be impossible to state the exact latency or jitter that developers should design for; nevertheless, we wished to get a general idea of the possible range that could be encountered.

There are ongoing research projects that measure internet performance characteristics such as round-trip latency (Caida 2000, Matrix 2000, NLANR 2000). These projects report mean round-trip times between 100ms and 200ms (ping times) and approximately 5% packet loss between various internet sites (e.g. Matrix 2000). These low results do not entirely reflect reality for groupware applications, however, since they use hosts that are on or near a major backbone, since there is almost no processing of messages required for the tests (which adds to overall delay), and since they do not measure jitter directly.

To get a broader sample of real-world groupware latency and jitter, we recorded and analysed message communication to several sites using an instrumented GroupKit (Roseman and Greenberg 1996) application. Latency was measured by timestamping and sending messages with immediate reply by the other groupware system. Jitter was measured by timestamping arrival times and calculating the difference between inter-send times and inter-arrival times for each consecutive pair of messages. Several different network bandwidths and distances were tested, from local area networks to dialup connections and worldwide Internet links. All trials were carried out during the day (based on the remote host's location). A few examples from these tests are shown in Table 1.

Approx. distance	Slowest network link	Max. latency (ms)	Median latency (ms)	Max. delay from jitter (ms)	Median jitter delay (ms)	% delayed by jitter
100 m	Ethernet	99	99	51	31	5
3 km	Dial-up	1040	476	320	303	33
3 km	Cable	4420	492	847	200	34
700 km	Cable	950	500	1245	280	26
3000 km	Ethernet	910	303	343	97	52
12000 km	Ethernet	1000	590	2509	244	41

Table 1. Example jitter and latency between skorpio.usask.ca and various hosts. Latency is calculated by dividing round-trip time in half.

In our tests, latency and jitter varied widely, and maximum delays could be several seconds. Average latencies ranged from 100 to 700 milliseconds, and jitter delays ranged from 40 to 1000 milliseconds. These results were used to determine the test conditions for the experiments that are described in the next section. More detail on these delay benchmarks can be found in (Gutwin 2001).

These results give a rough idea of what groupware users can expect in heterogeneous real-world settings. The next issues to be addressed are whether these delay types and magnitudes have a negative effect on the usability of a groupware system, and if so, how and why those effects happen.

Effects of delay on group work

We are interested in several questions about the ways that network delays affect closely-coupled collaborative work, where people have to keep close track of others in order to carry out their own work correctly. Our questions are:

♦ What types of collaborative interactions are affected by delay?

- Does delay reduce task performance and system usability?
- What magnitude of delay is required before usability suffers?
- How do task strategies change in the presence of delays?
- Which kind of delay is worse, latency or jitter?

We considered these questions in two stages. First, we observed groups using a real groupware system (a real-time game) both with and without delays. From these observations, we identified several specific kinds of interactions that seemed to be most affected by delay. Second, two interaction types (predicting another person's movement, and coordinating access to a shared artifact) were examined in greater detail through two lab experiments.

Observations of delay in a real groupware activity

Five pairs of people were observed for approximately one hour each as they played the real-time groupware game GK-Pipedreams (Figure 2). The game is a multi-player version of the arcade game Pipedreams, in which players must place sections of pipe to stay ahead of the water that flows through the pipeline. GK-Pipedreams represents players in the workspace with telepointers, shows all player movements and manipulations as they happen, and gives all players full access to the artifacts in the workspace. People usually play GK-Pipedreams in a mixed-focus fashion: they engage in both independent and shared work, moving back and forth (often rapidly) between loosely- and tightly-coupled interaction.

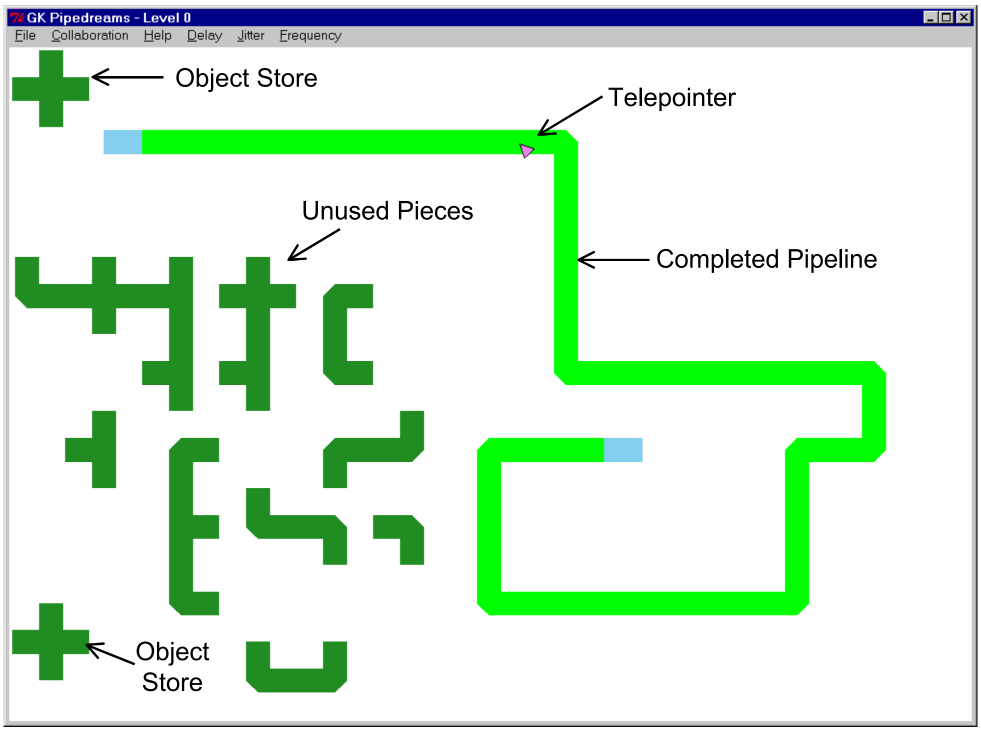

Figure 2. GK-Pipedreams groupware system. Players take pipe sections from the stacks at top left and bottom left, and build a pipeline from an inlet to an outlet.

For this study, GK-Pipedreams was altered to be able to introduce artificial communications delays between the two machines. In particular, the visual representation of the other person's movements could be subjected to controlled latency and jitter. Latency could be set to a fixed time between 0 and 1000 milliseconds. A simple model of jitter was adopted in which a certain percentage of messages would be delayed by a fixed time between 0 and 1000 milliseconds.

Pairs played the game first without any delays, and then with a range of artificial jitter and latency conditions. The players were in the same room (separated by a divider), so they were able to talk normally about their tasks. Two observers watched the on-screen interaction, looking for situations where delays appeared to cause some kind of problem for the group.

In general, people were able to deal with the latency and jitter conditions fairly well. Game performance did not appear to suffer even when large delays were imposed (c.f. Monk et al 1996), and people did not report major difficulties in working together on the task. However, the observers did notice changes in people's behaviour when delays were large. First, several people appeared to ignore the other person's activities entirely when the delay (particularly the jitter) was greater than about 500ms; that is, they reverted to working independently. Second, and perhaps as a result, groups appeared to run into more problems of coordination when delays were large: for example, players would both try to grab the same pipe piece, or would both try to move pieces into the same position. These problems were for the most part repaired quickly and without incident by the players; in discussions after the game, players only sometimes recalled that these problems had occurred at all.

Even though latency and jitter did not cause major problems for the groups, our observations suggested that certain kinds of collaborative interactions were affected by delay, and deserved closer study. These involve the activities of predicting movement and coordinating access.

- *Predicting another person's movement*. When a person moves around in a shared workspace, others use that movement to predict where the person is going (Gutwin and Greenberg 1998). These predictions are used to anticipate actions, to join someone at their destination, and to plan motion so as to avoid bumping into one another.
- *Coordinating access to shared artifacts*. Up to date knowledge of another person's activities in a closely-coupled situation is essential for managing access to a shared object or tool. This information is the basis of "social protocols" of concurrency control (Greenberg and Marwood 1994).

We carried out two experiments to test the effects of delay on prediction and coordination interactions in simple repeated tasks. We built two groupware systems with the GroupKit toolkit that implemented the interaction type. The groupware systems were set up such that two participants sat at different workstations and carried out the task by looking only at their own monitor.

Artificial delays were introduced using the same mechanism described earlier.

Experiment 1: Prediction

One type of prediction in a shared workspace is being able to determine where someone else is moving by watching their telepointer. We built a simple groupware system to isolate and test this type of interaction. The following sections describe the task, participants, experimental methods, and results.

The Prediction Task

This task involves two people, one attempting to predict as quickly and accurately as possible where the other person is pointing. One person (the follower) must click on a screen target that has been pointed to by the other person (the leader). Both participants see a set of boxes on their screen (see Figure 3); however, only the leader's screen showed which of these boxes was the target.

Leaders were instructed to move as quickly and accurately as possible to the yellow target as soon as it appeared on their screens. Followers were instructed to move to the box that was indicated by the leader, and click on the box when they arrived. Followers did not see the target highlighted in any way, but they could see the leader's telepointer. Groups were asked not to talk during this task.

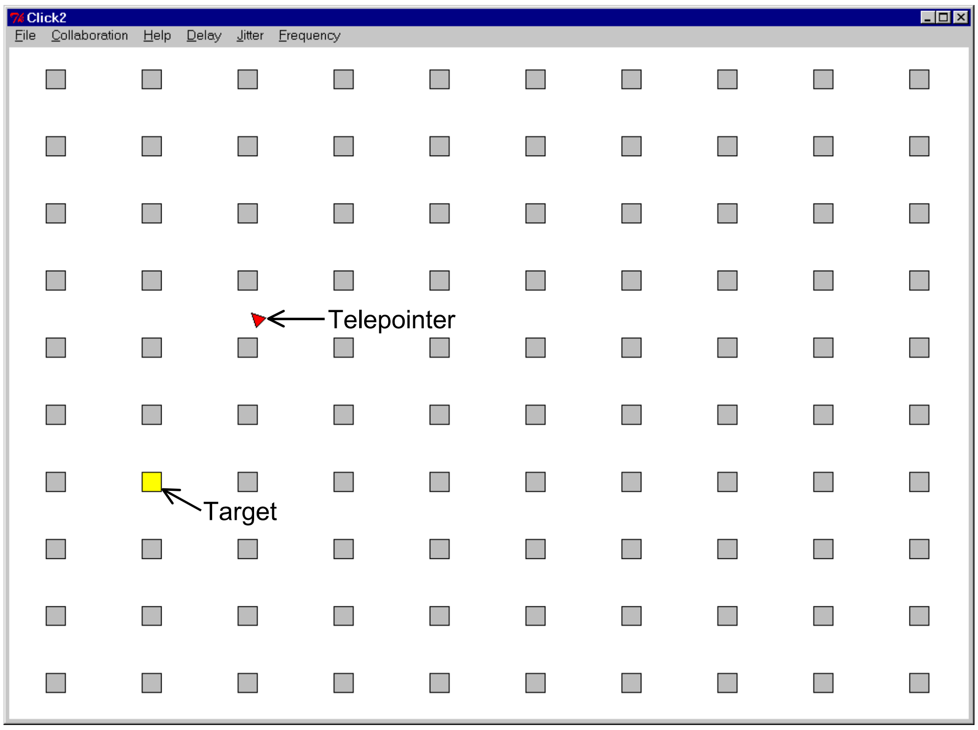

Figure 3. Prediction task system (leader's view). Target is coloured yellow.

Participants

Ten pairs of undergraduate computer science students participated in the study, and were given extra course credit for their involvement. Eight participants were women, and 12 were men; the mean age was 23 years. All participants were regular users of basic networked software (email and web browsers), but they had less experience with real-time groupware. Three participants had used a shared whiteboard system, and six had used internet voice conferencing. Five people were experienced with multi-player networked games. Finally, eight of the ten pairs were familiar with one another (more than three interactions per week).

Study hypotheses and conditions

The primary hypothesis in the prediction experiment was that increasing jitter would result in longer completion times and higher error rates. A secondary goal was to determine the level of jitter at which performance degrades significantly. We considered only jitter in this task because it was clear from pilot testing (see Gutwin and Penner 2000) that simple latency did not have any effect on performance.

The effects of delay were explored by applying several levels of artificial jitter to the display of the leader's telepointer on the follower's screen. The experiment was run on a network that had very low real jitter (mean 10ms, frequency 2%); in the summaries below we consider this as zero jitter. For the non-zero jitter conditions, the groupware system delayed the display of the leader's telepointer by the jitter amount and with a frequency of 10%; however, the system also ensured that at least one jitter event was randomly introduced into the display for each trial. Data was gathered by recording task completion times and errors for several blocks of pointing tasks.

Block	Type	Jitter (ms)	Trials
1	Practice	0	20
2	Practice	1000	5
3-7	Test	0, 200, 400, 600, 1000	12

Table 1. Conditions for prediction task; order of blocks 3-7 was randomised.

Pairs completed practice blocks with and without jitter, and then completed test blocks using different delay conditions, as shown in Table 1. Test block order was randomised for each group. The first two trials of each block were discarded as additional practice trials, leaving 10 test trials in each condition. Participants were allowed to rest between each block of trials. When the session was complete, the participants changed workstations and repeated the experiment in the other role, resulting in 20 total participants.

Results

Completion time and errors were recorded and logged by the groupware system. Results for the five test conditions are shown in Table 2 and Figures 4 and 5.

Jitter (ms)	N	Completion time per block		Errors per block	
		Mean (ms)	SD	Mean	SD
0	20	20910	5395	0.15	0.3663
200	20	21166	3242	0.15	0.3663
400	20	23349	5813	0.05	0.2236
600	20	25369	6123	0.15	0.3663
1000	20	25060	4773	0.75	1.1642

Table 2. Completion times and error rates for prediction task

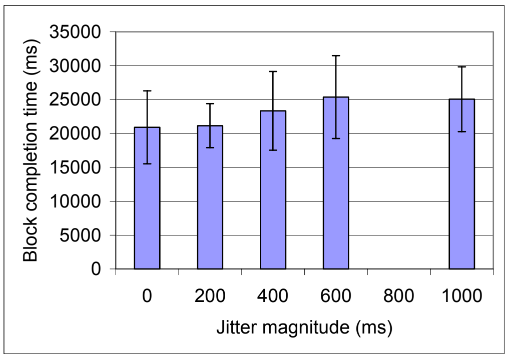

Figure 4. Mean completion times for prediction. One block = 10 trials; error bars represent standard deviation.

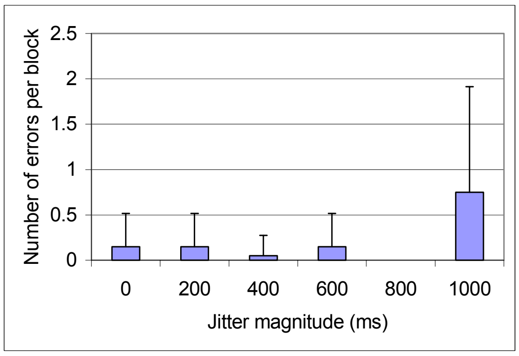

Figure 5. Mean errors for prediction. One block = 10 trials; error bars represent standard deviation.

We examined the primary hypothesis by carrying out repeated-measures ANOVA for completion time data and error data. Both analyses showed main effects of jitter magnitude: for completion time, F(4,76)=4.983, p<0.05; for error rate, F(4,76)=6.909, p<0.05.

To determine the magnitude at which jitter makes a substantial difference, post hoc one-tailed t-tests were performed comparing jitter conditions with the no delay condition. A Bonferroni correction was used to maintain alpha at 0.05; therefore, only p-values of 0.0125 or less were considered significant. For completion time data, t-tests showed significant differences for jitter of 600 ms and 1000ms (p<.0125), but not for jitter of 200ms (p=0.429) or 400ms (p=0.089). For error data, there were no significant differences between conditions.

In addition to the quantitative data, the experimenter also observed the way that groups carried out the tasks. It was clear that in low-delay conditions, people did use the feedback of the telepointer to carry out the task more quickly. People almost always started moving their mouse before the other person had stopped moving. People would track the other cursor closely, and then click the target as soon as the telepointer stopped moving. When there was larger jitter, however, people adopted different strategies. They would often still track the telepointer, but would pause when it stopped to wait and see if the pointer would begin moving again. In a few cases, people had difficulty finding the telepointer after it had made a large jitter-induced jump. Some participants resorted to a strategy of not starting their move until they were sure that the other person had finished.

Experiment 2: Coordination

The second type of interactive activity we studied was coordinating access to shared resources. A second experiment and a second groupware system were designed to test the effects of delay on coordination.

The Coordination Task

In the coordination task, participants were asked to drag objects from a shared central stack and drop them onto a target region. There were two drop regions, one for each participant, but only one central stack of objects (see Figure 6). In this system, both participants see the same objects on their screens, and carry out the same task actions. Participants were instructed to drag and drop a set number of objects from the stack, but were explicitly cautioned to minimise errors—that is, the number of times they grabbed an object that the other person had already taken. Groups were allowed to talk to each other during this task.

Participants

The same pairs from the prediction study were used in the coordination study, and were run through the experiment later the same day. This meant that all of the study participants had experienced jitter conditions in both the leader and follower role of the prediction task.

Study Hypotheses and Conditions

The primary hypothesis in the coordination experiment was that increasing jitter and latency would result in longer completion times and higher error rates. Again, the secondary goal was to determine the level at which performance degrades. In this task, several levels of either jitter or latency were applied to the display of both participants' telepointers and object moves as shown on the other person's screen. The system was run on a fast network with a consistent latency of 40 ms; as this is the lowest latency we could achieve, this amount was used in the 'no delay' condition.

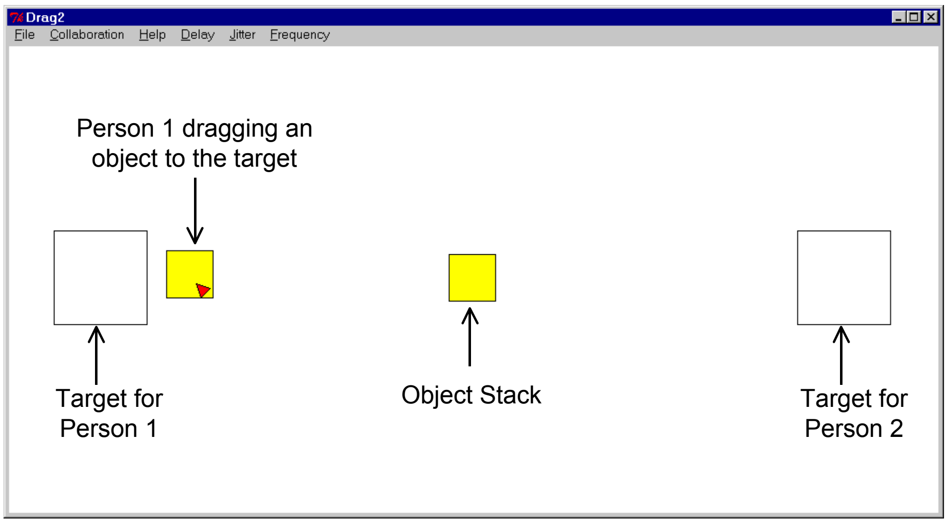
Drag2
File Collaboration Help Delay Jitter Frequency
Person 1 dragging an object to the target
Target for Person 1
Object Stack
Target for Person 2

Figure 6. Coordination task system. Rectangles at left and right are the drop regions; filled rectangles are objects to be moved; object stack is at centre.

Groups first completed three practice trials, and then carried out a set of test trials with different delay conditions (see Table 3). In each block of test trials, groups completed 50 object moves, of which the first five were discarded as additional practice trials, leaving 45 test trials per block. Block order was randomised for each group. Data was gathered by recording task completion times and errors for several blocks of trials. Errors were defined as attempts by one person to grab an object with their mouse that another person was already holding. Participants were specifically instructed to try and minimise these "double grabs."

Block	Type	Latency (ms)	Jitter (ms)	Trials
1	Practice	40	0	50
2	Practice	1040	0	10
3	Practice	40	1000	10
4	Practice	440	400	10
5-9	Test	40, 240, 440, 640, 1040	0	50
10-13	Test	40	40, 240, 440, 640, 1040	50

Table 3. Study conditions for prediction task.
The order of blocks 5-13 was randomised.

Results - Coordination

Completion time and errors were recorded and logged by the groupware system. Results for the test conditions are shown below in Table 4 and Figures 7 and 8.

Jitter (ms)	Latency (ms)	N	Completion time (45 trials)		Errors (45 trials)	
			Mean (ms)	SD	Mean	SD
0	40	10	31497	6063	0.9	1.728
200	40	10	32523	11893	1.1	0.994
400	40	10	31133	5463	1.8	1.475
600	40	10	32088	5135	1.4	1.712
1000	40	10	33041	5006	3.1	1.969
0	40	10	31497	6063	0.9	1.72
0	240	10	32940	4980	4.6	2.95
0	440	10	33876	5167	10.8	3.29
0	640	10	33942	5377	14.6	3.4
0	1040	10	31983	5896	16.8	2.89

Table 4. Completion time and error rates for all study conditions.

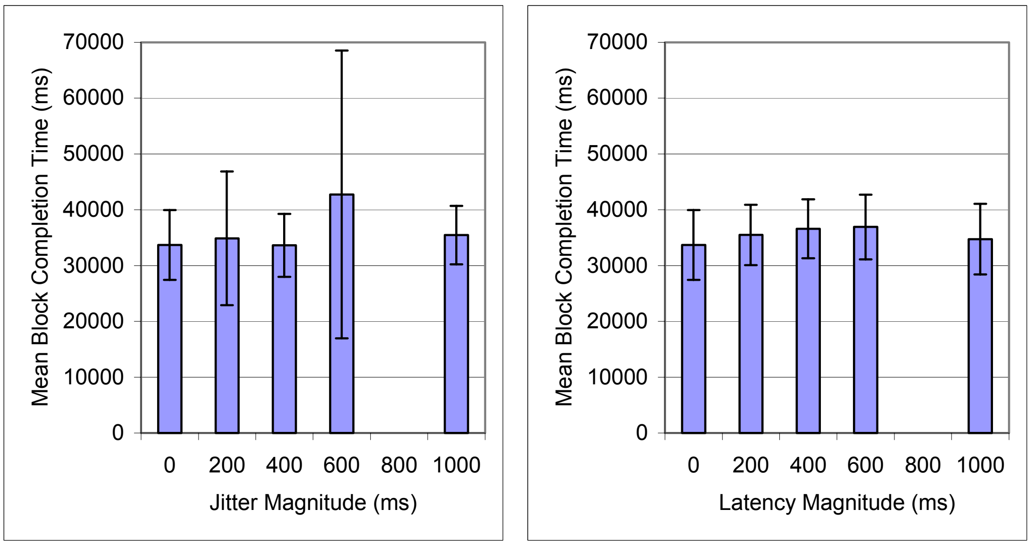

Figure 7. Mean block completion times for jitter (left) and latency (right) conditions. Blocks consist of 45 trials. Error bars show standard deviation.

The a priori hypothesis for the coordination task was that increased jitter and latency would increase completion time and errors. We considered latency conditions and jitter conditions separately, using the no-delay condition in both analyses. Repeated-measures analyses of variance were carried out for both error rates and completion time. For completion time, there were no main effects found for either jitter (F(4,36)=0.841, p=0.509) or latency (F(4,36)=1.6, p=0.195). For error rates, significant main effects were found for both jitter (F(4,36)=6.445, p<0.05) and latency (F(4,36)=73.11, p<0.05) conditions.

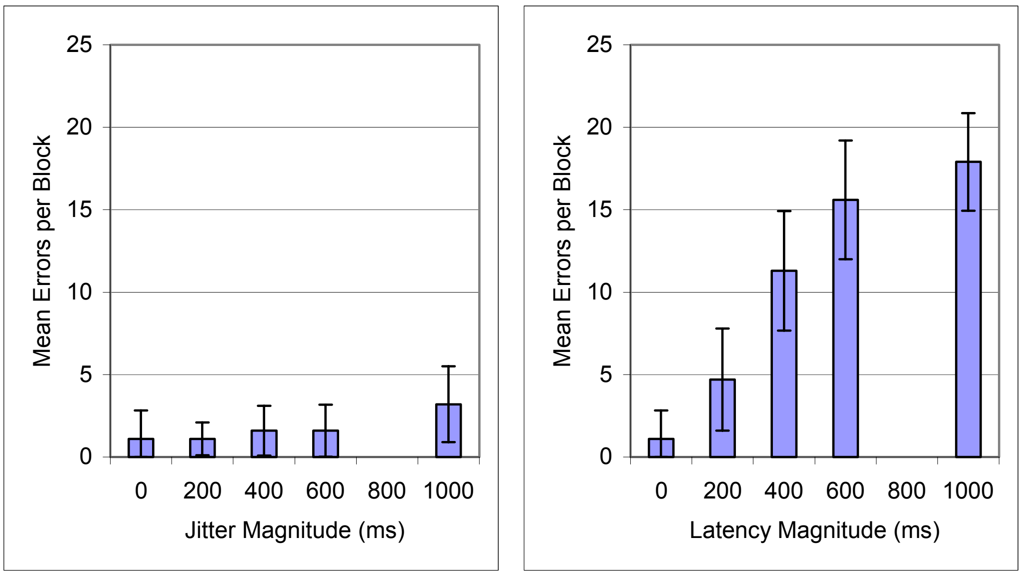

Figure 8. Mean errors per block for jitter (left) and latency (right) conditions. Blocks consist of 45 trials. Error bars show standard deviation.

To explore the error results more closely, post hoc one-tailed t-tests were performed to compare the different delay conditions with the no delay condition. A Bonferroni correction was again used to maintain alpha at 0.05. For the latency results, t-tests showed significant differences for all conditions compared to the no delay condition (all p<0.0125). For the jitter results, no significant differences between conditions were found.

Observations were also made of the participants in the coordination task. Strategies in this task all used turn-taking; all pairs used some kind of you-then-me-then-you approach to taking objects from the stack. However, when one person appeared to be taking longer with a move, the other person would occasionally fill in the time by taking an extra object. In low-delay conditions, errors were rare, and usually happened when people were re-starting the turn-taking. With larger delays errors happened regularly, and often occurred when the screen appeared to show that it was safe to take an object from the stack. When people had difficulty coordinating in the high-delay conditions, they would generally slow down and try to establish for certain where the other person was (e.g. by both participants going to their drop regions). They would then restart the turn-taking strategy.

Discussion

Our goals in this research were to find out what kinds of real-world interaction are affected by network delays, to determine the level at which performance starts to suffer, and to determine how task strategies change in the presence of delays. The next sections summarise our findings, interpret the findings in light of our general goals, consider the generalisation of the results, and discuss design approaches that can improve groupware usability in the face of network delay.

Summary of results

An observation of a real groupware system (GKPipedreams) suggested that overall task performance is resilient to the presence of even large delays. However, this resiliency occurred because people were able to alter their work style to be more independent and less collaborative. In addition, certain types of interaction appeared to cause problems even when people ignored the other person's movements. Two of these in particular were predicting others' movements, and coordinating access to shared objects.

In experiments to explore these two interactions in more detail, several of the priori hypotheses were supported by the results. Table 5 shows a summary of our quantitative findings. We found significant relationships between jitter magnitude and prediction performance, and between both types of delay and coordination errors. We also found that prediction time is significantly affected with jitter of 600ms or more, and that coordination errors are significantly increased with latency of 240ms or more.

Task	Hypothesis	Sig. Effect	1st sig. level
Prediction	Jitter magnitude increases CT	Yes	600 ms
Prediction	Jitter magnitude increases errors	Yes	None
Coordination	Jitter magnitude increases CT	No	—
Coordination	Jitter magnitude increases errors	Yes	None
Coordination	Latency magnitude increases CT	No	—
Coordination	Latency magnitude increases errors	Yes	240 ms

Table 5. Summary of quantitative results.
CT = completion time; Sig = significant.

The two most obvious effects were that of jitter on prediction time, and that of latency on coordination errors. These results can be converted back to real-world terms. With no jitter, predicting another person's pointing action took about two seconds; with jitter of 600ms, prediction took about half a second longer. With no latency, pairs made one coordination error in 50 repeated manipulations of a shared object. When latency was 240ms, they made one error in every 10 manipulations, and when latency was 1000ms, they made one error in every three.

We also found that people's strategies changed in the presence of delays. High jitter in the prediction task resulted in a change from a "start following immediately" strategy to a "wait and see" strategy. Latency in the coordination task seemed to force a "rhythmic turn-taking" strategy, but even this strategy often broke down when latencies were larger. From the experimenter's observations, it seemed clear that delays made the tasks more difficult, and that they reduced satisfaction with overall systems. It was clear that when there were delays in the system, people had to slow down and pay more attention to both their own and others' actions.

Explaining the results

In closely coupled visual tasks, people use the representation of the other person as a kind of visual evidence of the state of the action (e.g. Brennan 1990, Clark 1996). In the real world, people quickly learn to use this information to improve their efficiency. They can become expert at a shared task by learning the task boundaries that are presented by the other person, and gradually pushing towards those limits. This expertise, however, depends upon the information being accurate. In the prediction and coordination tasks, a main reason that delays impaired performance is that delay introduces uncertainty into a situation where certainty is required for expertise. When visual evidence is uncertain, fast perceptual tasks (e.g. simply watching the other person) are changed into time-consuming cognitive tasks (e.g. mentally calculating the other person's location or current activity).

In the prediction task, the most common strategy in the no-delay condition was to follow the other person's cursor as closely as possible, and to choose a target as soon as the telepointer stopped moving. When jitter was introduced, however, it made the telepointer appear as if the leader had stopped moving, forcing the follower to think about whether they had really stopped. In addition, attempts to "think ahead" and predict the leader's location in jitter conditions required the follower to calculate the likely path of the leader's cursor based on its previous speed and trajectory (also noticed in (Vaghi et al 1999)).

In the coordination task, latency made people unsure about what their partner was doing, and made it difficult to determine whether it was safe to grab the next object from the stack. When coordination is based on visual feedback, and the feedback cannot be trusted, coordination becomes more difficult. Perhaps as a result, almost all of the groups fell into rhythmic turn-taking during the delay conditions—an example of a non-visual way to coordinate the task and gain certainty about the other person. As long as people maintained the rhythm, they did not have to depend on untrustworthy visual evidence; unfortunately, the rhythm broke down often when people paused or got out of step, and it was often in restarting where errors occurred.

Jitter was less of a problem in the coordinated-moving task. This may have been because a jittery telepointer was still up-to-date most of the time, and along with the rhythm of turn-taking, may have provided enough information to keep the activity properly coordinated. One situation where jitter did appear to cause a problem, however, was when one person's cursor stuck while they were inside their drop area. This occasionally fooled their partner into thinking that they had paused, and into attempting to grab a second object from the stack. This problem is similar to that of the prediction task, where a telepointer freeze forces the observer to guess whether motion has really stopped.

Generalising the results

The two experiments showed that delays can affect both performance and strategy during closely-coupled activities. However, the experiments were set up to isolate and repeat specific types of interaction, so the question remains of what these results imply for real world group work.

First, the effects of delay on real world task performance will be proportional to the number of delay-sensitive interactions that the task requires. In tasks where predicting movement and coordinating access are common, it is likely that jitter and latency will lead to measurable performance loss. In other tasks these types of interactions will be less frequent (as in GKPipedreams) and so performance losses will not be apparent. Even if overall performance is not affected, however, delays will cause problems that affect system usability: when people have to keep track of others in a groupware system, latency and jitter are simply annoying. Despite the fact that people can concentrate harder and still get the task done, usability suffers.

Second, the strategy change that we observed in the GKPipedreams observation suggests that people tend to work more as individuals when visual information about others‘ activities cannot be trusted. This strategy protects performance, but means that the benefits of working as a group are largely lost. When groups are fully aware of others in the workspace, they are able to provide assistance, monitor one another, discuss artifacts, and share joint tasks. When people work as individuals, they do not maintain the awareness that provides for these sorts of collaborative benefits. These more subtle effects of delay on real-world distributed work will be the focus of our continued research in this area.

Designing for delay

What can a groupware developer do to design for delay, given that the network is not under their control? The most important things from the designer's perspective are to assess both the coupling requirements of the groupware system and the likely delays in the situation where the application will be installed. If the group task requires close coupling and the network delays are high, the developers may have to rethink their design, since unreliable information about others is often worse than no information at all. Once the requirements are known, there are several approaches that can be used to reduce delays or work around them.

Network techniques. First, different network configuration can be used that have lower delays than the Internet (e.g. dedicated ISDN modems). Second, different message protocols provide different quality-of-service (QoS) parameters than the TCP/IP protocol currently used in most groupware toolkits. For example, the UDP protocol has considerably lower latency and jitter than TCP. It does not provide guaranteed delivery, but for messages such as telepointer position, guarantees are not usually required. Other protocols such as LRMP (lightweight reliable multicast protocol) add some control ability on top of UDP, and may also be useful in groupware.

Smoothing. Buffering incoming messages is one way of reducing jitter at the cost of increasing latency. In situations where tasks are affected more by jitter than by latency, smoothing can be used.

Adaptation. If groupware systems are able to measure the current delay, they could be able to adapt the interface to present the users with interaction techniques that are most appropriate for the current delay conditions. For example, as jitter increases beyond a certain level, the system could offer to switch from the use of telepointers for user awareness to a participant list, which is not affected by jitter.

Explicit indications of delay. People may be better able to adapt their behaviour to delays if they are aware of the presence and magnitude of the delay. Delay can be represented in the interface by devices like delay gauges, or by "ghosting" effects on moving objects (Vaghi et al 1999).

Conclusions

Delay is a fact of Internet life, and is one aspect of a groupware system that is difficult for the developer to control. Since many groupware applications are being built to run across the Internet, we undertook an investigation into the effects of delay, particularly latency and jitter, on group work in real time groupware. Based on observations of a real groupware system, we identified two kinds of interaction in closely-coupled tasks that seemed to be affected by delays, and designed experiments to study them more closely. Significant relationships between delay and task performance were found, and it was clear that task strategies changed to accommodate the delay. Our main direction for future work is in studying the effects of delay on more realistic collaboration. From our observations of GK-Pipedreams, it is evident that there are changes to strategy, interaction, and conversation when information is delayed in a real-world application, but identifying these subtle changes will likely require more sensitive qualitative and process measures.

Acknowledgements

Thanks to Reagan Penner for running an earlier version of the experiments, and to Sean Cocks for conducting the version reported on here.

References

1. Acharya, A., and Saltz, J. (1996) *A Study of Internet Round-Trip Delay*. Report CSTR-37-36, Dept. of Computer Science, University of Maryland, 1996.
2. Beaudouin-Lafon, M., and Karsenty, A. (1992), 'Transparency and Awareness in a Real-Time Groupware System', *Proceedings of the Conference on User Interface and Software Technology*, Monterey, CA, 1992, 171-180.
3. Brennan, S. (1990), *Seeking and Providing Evidence for Mutual Understanding*, Ph.D. thesis, Stanford University, Stanford, CA, 1990.
4. Cisco corporation (2000), *Designing Internetworks for Multimedia*. Currently at: www.cisco.com/ univercd/cc/td/doc/cisintwk/idg4/nd2013.htm.
5. Clark, H. (1996), *Using Language*, Cambridge University Press, Cambridge, 1996.
6. Cooperative Association for Internet Data Analysis (2000). http://www.caida.org
7. Fluckiger, F (1995). *Understanding Networked Multimedia*, Prentice Hall, 1995.
8. Gaver, W. (1991), 'Sound Support for Collaboration', *Proceedings of the 2nd European Conference on Computer Supported Cooperative Work*, 1991, 293-308.
9. Greenberg, S., and Marwood, D. (1994), 'Real Time Groupware as a Distributed System: Concurrency Control and its Effect on the Interface', *Proceedings of the Conference on Computer-Supported Cooperative Work*, 1994, 207-217.
10. Gutwin, C. (2001) *Groupware Latency and Jitter on the Internet*. Technical Report, Department of Computer Science, University of Saskatchewan, 2001.
11. Gutwin, C., and Penner, R. (2000) *Slow and Sticky: Network Delay Effects on Groupware*. Report 00-05, Department of Computer Science, University of Saskatchewan, 2000.
12. Gutwin, C., and Greenberg, S. (1998) 'Effects of Awareness Support on Groupware Usability'. *Proceedings of ACM CHI'98*, Los Angeles, ACM Press, 1998.

13. Hall, R., Mathur, A., Jahanian, F., Prakash, A., Rassmussen, C. (1998) 'Corona: A Communication Service for Scalable, Reliable Group Collaboration Systems'. *Proceedings of CSCW'98*, pp. 140-149, 1998.
14. Matrix Information and Directory Services, Inc. (2000). *The Internet Weather Report*. http://www.mids.org/weather/
15. Monk, A., McCarthy, J., Watts, L., and Daly-Jones, O. (1996), 'Measures of Process', in P. Thomas (ed): *CSCW Requirements and Evaluation*, Springer-Verlag, London, 1996, 125-139.
16. National Laboratory for Applied Network Research (2000). www.nlanr.net
17. Park, K. and Kenyon, R. (1999), 'Effects of Network Characteristics on Human Performance in the Collaborative Virtual Environment'. *Proceedings of IEEE Virtual Reality '99*, March 14-17, 1999.
18. Roseman, M., and Greenberg, S. (1996), Building Real-Time Groupware with GroupKit, a Groupware Toolkit, *ACM Transactions on CHI*, 3(1), 66-106, 1996.
19. Salvador, T., Scholtz, J., and Larson, J. (1996), The Denver Model for Groupware Design, *SIGCHI Bulletin*, 28(1), 52-58, 1996.
20. Scholl, F. (2000), *Planning for Multimedia*. White paper, Monarch Information Networks. Currently available at: http://www.monarch-info.com/pmm.html.
21. Segal, L. (1995), 'Designing Team Workstations: The Choreography of Teamwork', in P. Hancock, J. Flach, J. Caird and K. Vicente (eds): Local Applications of the Ecological Approach to Human-Machine Systems, 392-415, Lawrence Erlbaum, Hillsdale, NJ, 1995.
22. Stefik, M., Foster, G., Bobrow, D., Kahn, K., Lanning, S., and Suchman, L. (1987), 'Beyond the Chalkboard: Computer Support for Collaboration and Problem Solving in Meetings', *Communications of the ACM*, 30(1), 32-47, 1987.
23. Tanenbaum, A. (1996), *Computer networks*, Prentice-Hall, Englewood N.J., 1996.
24. Tang, J. (1991), 'Findings from Observational Studies of Collaborative Work', *International Journal of Man-Machine Studies*, 34(2), 143-160, 1991.
25. Tatar, D., Foster, G., and Bobrow, D. (1991), 'Design for Conversation: Lessons from Cognoter', *International Journal of Man-Machine Studies*, 34(2), 185-210, 1991.
26. Kum, H., and Dewan, P. (2000), 'Supporting Real-Time Collaboration Over Wide Area Networks', *Video Proceedings of ACM CSCW 2000.*
27. Vaghi, I., Greenhalgh, C., and Benford, S. (1999), 'Coping with Inconsistency due to Network Delays in Collaborative Virtual Environments'. *Proceedings of the ACM Workshop on Virtual Reality and Software Technology*, 1999, 42-49.