



Exploring Overheads and a Peer-to-Peer Collaborative



Communication Locking Policies in Synchronous Editing System



Motivation

- Synchronous collaboration during periods of “rapid development”
- Examine collaborative, distributed software engineering
- Examine general-purpose document editing
- Explore communication overhead in a peer-to-peer environment



Related Work

- Computer-supported cooperative work (CSCW)
- Notification algorithms
- Configuration management (CM) from Software Engineering
- Adjust the granularity of the lock
- Avoid edit collisions and improve concurrent edit possibilities



Other Work in this Area

- Many HCI related issues (awareness, presence, etc.)
- Integrating software configuration management (SCM) into the IDE
- Hierarchical locking and “soft locks”
- Explicitly supporting informal software development methods

The Collaborative Environment

The screenshot shows the Synch Edit 1.0 IDE interface. The title bar reads "Synch Edit 1.0 ID = -815206130". The main window is titled "File" and shows a code editor with the following C++ code:

```
#include <stdio.h>

void main (int argc, char** argv)
{
    int i;

    for(i=0; i < 100; i++)
    {
        cout << "The variable 'i' is = " << i << endl;
        i += 2;
    }
}
```

Below the code editor, the status bar shows "lower", "shift off", "Current Line: 11", and "OWNED".

On the right side, there is a chat window with the following messages:

<Jon Preston> Ok - we're both in now
<David Gilmour> Great - thanks for starting the code up
<Jon Preston> Now - what do we need to edit next?
<David Gilmour> Let's run it to see output first

Below the chat window is a "Send" button. Below that is another chat window with the following messages:

<David Gilmour> 0155 0
<Jon Preston> on line -1
<Jon Preston> on line 10
<David Gilmour> on line 11
<David Gilmour> on line 10
<David Gilmour> on line 9
<David Gilmour> on line 10

User is Now Locked Out of Line

The screenshot shows the Synch Edit 1.0 IDE interface. The title bar reads "Synch Edit 1.0 ID = -814157553 (HOST)". The main window is titled "File" and shows a C++ program being edited. The code is as follows:

```
#include <stdio.h>

void main (int argc, char** argv)
{
    int i;

    for(i=01 i < 100; i++)
    {
        cout << "The variable 'i' is = " << i << endl;
        i += 2;
    }
}
```

The status bar at the bottom indicates "lower", "shift off", "Current Line: 11", and "not owned".

On the right side, there is a chat window with the following messages:

<Jon Preston> Ok - we're both in now
<David Gilmour> Great - thanks for starting the code up
<Jon Preston> Now - what do we need to edit next?
<David Gilmour> Let's run it to see output first

Below the chat messages is a "Send" button. At the bottom of the chat window, there is a scrollable list of messages:

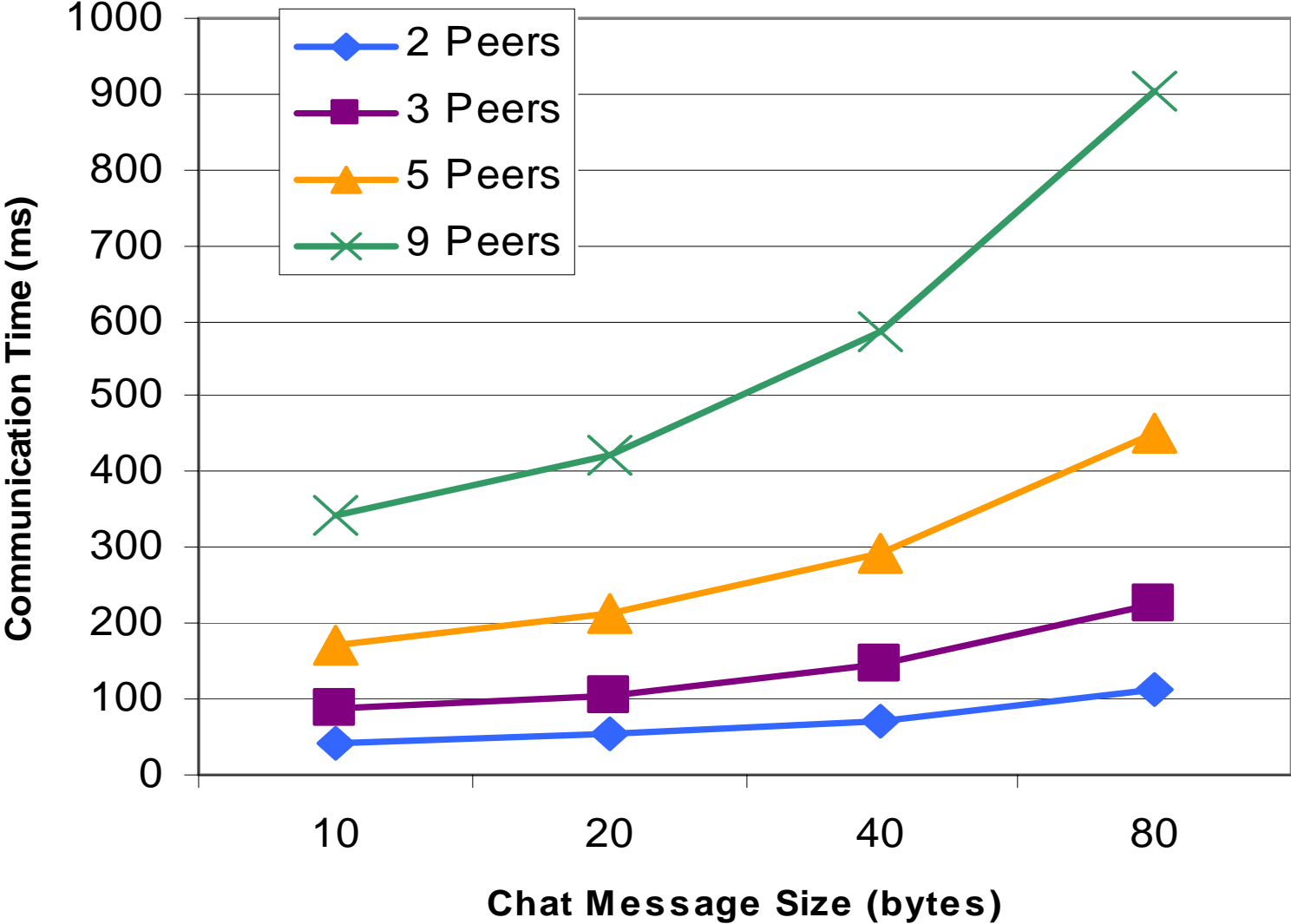
<David Gilmour> | 155 0
<Jon Preston> on line -1
<Jon Preston> on line 10
<David Gilmour> on line 11
<David Gilmour> on line 10
<David Gilmour> on line 9
<David Gilmour> on line 10
<Jon Preston> on line 11
<Jon Preston> on line 10
<Jon Preston> on line 9
<Jon Preston> on line 8
<Jon Preston> on line 7
<Jon Preston> on line 6



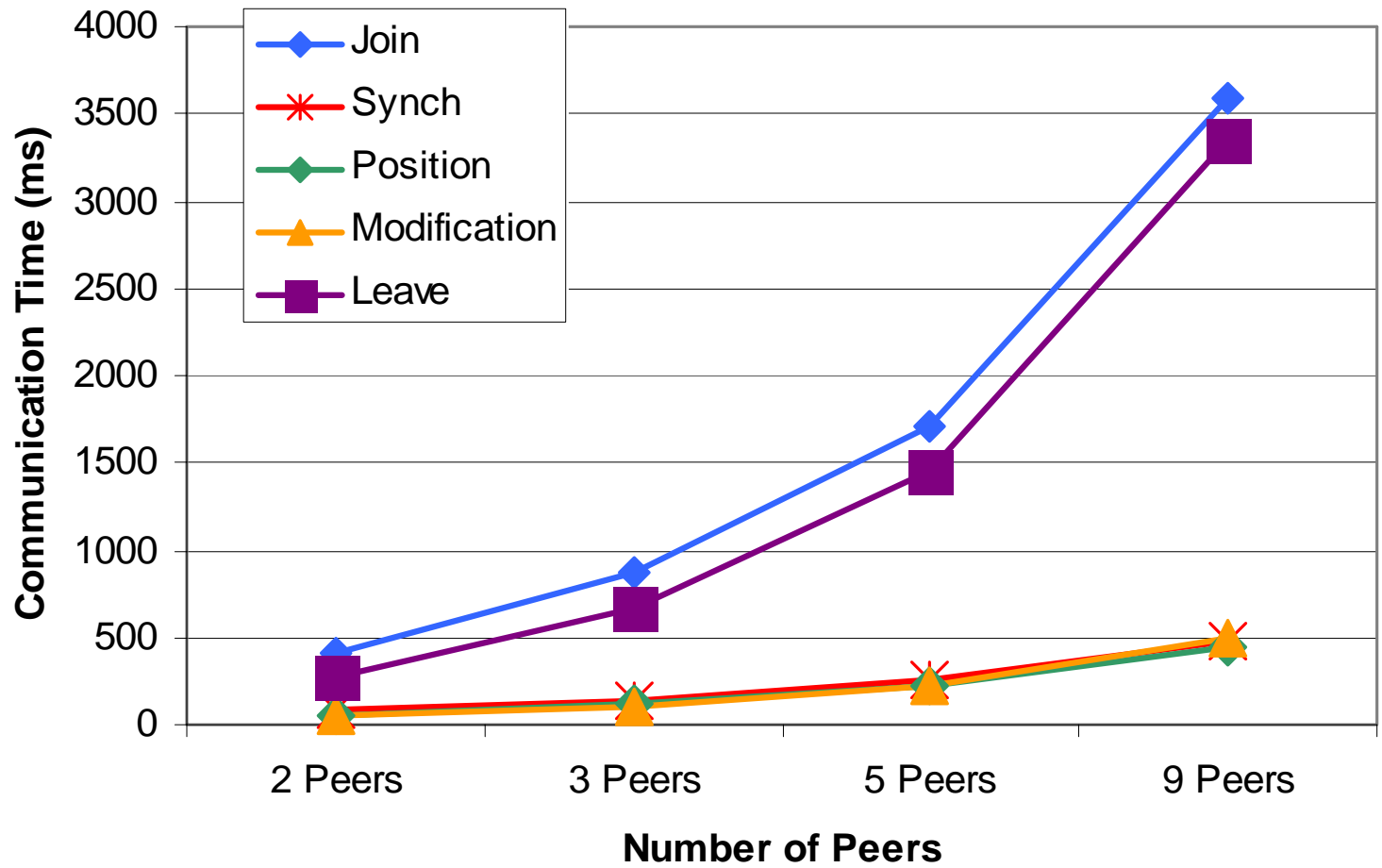
Communication Analysis

- The system is I/O bound, so communication time dominates
- Five types of messages:
 - Peer joins and broadcasts to others
 - Peer requests synchronize of content
 - Chat content is sent (to all peers)
 - Movement (position update)
 - Modification to the shared space

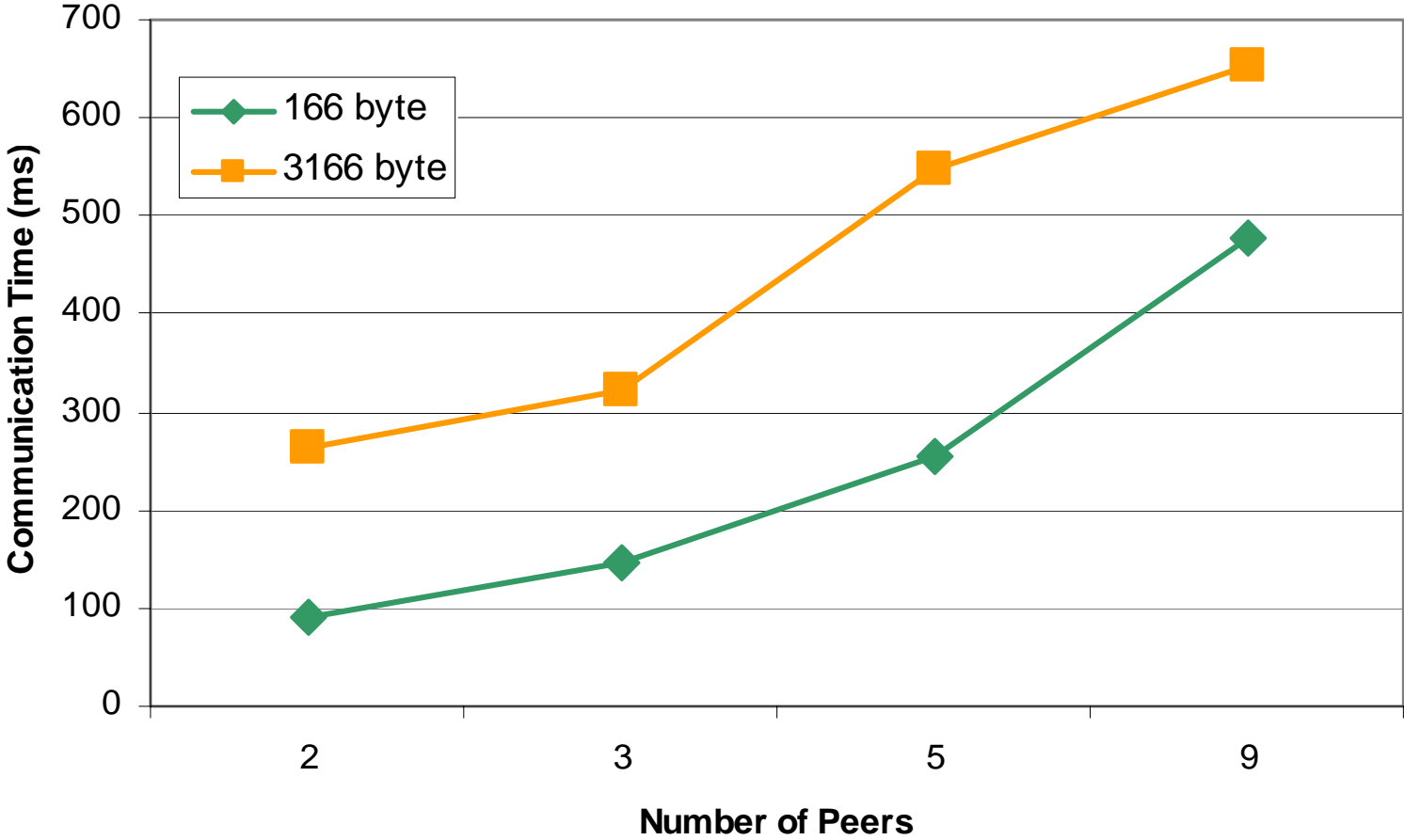
Communication Cost: Chat Message



Communication Costs



Communication Cost: 166 byte vs. 3166 byte Shared Content Synchronize





Contribution & Uniqueness

- Synchronous collaboration with optimistic locking
- Line-level locking is novel (varying the lock granularity)
- This research implements the collaborative environment via peer-to-peer (DirectX 9)
- Demonstrates that the communication overhead is reasonable



Future Research

- Examine various locking granularities (class, method, block)
- Cache changes and adjust frequency of updates to minimize packet overhead
- Accurately simulate edit patterns via CMS log-file analysis
- Develop Web services-based API for general-purpose collaborative editing
- Improve the usability/notification
- Implications for Extreme Programming (XP)



Contact

Jon A. Preston

jon.preston@acm.org

Department of Computer Science
Georgia State University

PhD Advisor: Dr. Sushil K. Prasad

References

- J. Estublier, "Software Configuration Management: a Roadmap", Proceedings of the Conference on the Future of Software Engineering, Limerick, Ireland, pp. 279-289.
- C. Gutwin and S. Greenberg, "The Importance of Awareness for Team Cognition in Distributed Collaboration", Team Cognition: Understanding the Factors that Drive Process and Performance, APA Press, Washington, pp. 177-201.
- C. Gutwin, R. Penner, and K. Schneider, "Group Awareness in Distributed Software Development", Proceedings of CSCW'04, ACM, Chicago, IL, pp. 72-81.
- W.H. Harrison H. Ossher, and P.F. Sweeney, "Coordinating Concurrent Development", In Proceedings of CSCW'90, 157-168, October 1990.
- B. Magnusson, U. Asklund, and S. Minör, "Fine-Grained Revision Control for Collaborative Software Development", Proceedings of ACM SIGSOFT'93 – Symposium on the Foundations of Software Engineering, Los Angeles, CA, December 7-10, 1993.
- J. M. Marques and L. Navarro, "WWG: a Wide-Area Infrastructure to Support Groups", In Proceedings of GROUP'01, Boulder CO, pp. 179-187, September 2001.
- A. Mehra, J. Grundy, and J. Hosking, "Supporting Collaborative Software Design with a Plug-in, Web Services-based Architecture", Workshop on Directions in Software Engineering Environments (WoDiSEE) from Proceedings of the 26th International Conference on Software Engineering (ICSE'04), IEEE, Edinburgh, Scotland, 2004.
- C. O'Reilly, "A Weakly Constrained Approach to Software Change Coordination", Proceedings of the 26th International Conference on Software Engineering (ICSE'04), IEEE, Edinburgh, Scotland, 2004.
- H. Shen and C. Sun, "Flexible Notification for Collaborative Systems", CSCW'02, New Orleans, LA, ACM Press, pp. 77-86.
- J. Tam and S. Greenberg, "A Framework for Asynchronous Change Awareness in Collaboratively-Constructed Documents", CRIWG 2004, Springer-Verlag, Berlin, pp. 67-83.
- J. Tam, L. McCaffrey, F. Maurer, and S. Greenberg, Change Awareness in Software Engineering Using Two Dimensional Graphical Design and Development Tools. Report 2000-670-22, Department of Computer Science, University of Calgary, Alberta, Canada, October 2000.
- A. van der Hoek et al, "Continuous Coordination: A New Paradigm for Collaborative Software Engineering Tools", Workshop on Directions in Software Engineering Environments (WoDiSEE) from Proceedings of the 26th International Conference on Software Engineering (ICSE'04), IEEE, Edinburgh, Scotland, 2004.
- R. van der Lingen and A. van der Hoek, "An Experimental, Pluggable Infrastructure for Modular Configuration Management Policy Composition", Proceedings of the 26th International Conference on Software Engineering (ICSE'04), IEEE, Edinburgh, Scotland, 2004.
- M. Younas and R. Iqbal, "Developing Collaborative Editing Applications using Web Services", Proceedings of the 5th International Workshop on Collaborative Editing, Helsinki, Finland, September 115, 2003.