

```
1 -----
2 3/27/95
3
4 Modify the protocol classes and location class for
5 signed location coordinates.
6
7 Made the port numbers for text and binary connections command line variables.
8 -----
9 3/31/95
10
11 Do not send state messages automatically.
12
13 Replace assert with something that doesn't abort in some of the network layer.
14 -----
15 4/6/95
16
17 Rewrite to conform to protocol14. Added Teleport, Disappear, and Appear
18 messages. Modified RoomChange message.
19
20 Added checking for duplicate User name.
21 -----
22 4/9/95
23
24 Added capability for ProtocolObject to accept and forward messages to
25 UserObjects other than the one for its Client.
26
27 Add PropertyCommand and PropertyRequestCommand.
28
29 Add WhisperCommand.
30
31 Additional error checking to make sure both user and world are set before
32 accepting commands besides SessionInit and AppInit.
33
34 -----
35 4/11/95
36
37 Moved main loop to main.cc
38
39 Added brute force crowd control. In Room.cc, compute a list of
40 users proximate to a given location by estimating the distance to
41 every user in the room.
42
43 Added a fix suggested by Bo to net_mbuf.cc. Modifies error checking
44 in the read() function immediately following the recv() call.
45
46 Tracked down several problems with the server crashing when the client
47 exits improperly.
48
49 Added some signal trapping in the network layer.
50
51 Added a command line argument to set the number of user updates.
52
```

53 [NOTE: This version lost on grotto?]
54
55 -----
56 4/19/95
57
58 Integrated Mitra's shortObjId class.
59
60 Changed select timeout in networking layer to 100 microseconds.
61
62 Changed timeout on an individual connection to 3 minutes.
63
64 Improved message logging in BinaryProtocolObject.
65
66 Modified SessionInit and AppInit to parse unknown properties.
67
68 Added a message logging switch to the command line.
69
70 Sets the number of file descriptors to MAXCONNECTIONS (net_comm.cc)
71
72 Fixed memory leak - Room not deleting location update messages.
73
74 -----
75 4/24/95
76
77 Added the visibleUser table and processing to User.cc. This processing
78 will absolutely limit the number of updates that get sent to a
79 Client, and gives us a cleaner algorithm for providing the Client with
80 Appear and Disappear message.
81
82 Added an "I see you" list to each User. These are used to notify
83 Users that are sending me updates when I leave the World (not doing
84 this causes the server to core dump when a client crashes). This
85 list is also used to allow some transitivity (if you can see me, I can see you)
86 by sending location updates to Users on both the "I see you"
87 list and the "you see me" list.
88
89 Changed the timeout on the select in net_comm.cc to 50000 microseconds.
90
91 Ordinary Users set to 30, priority Users set to 10, connection timeout
92 set to 5 minutes.
93
94 Temporarily removed text port functionality to prevent crashes due
95 to Satan executions over the network.
96
97 -----
98 4/28/95
99
100 Implemented a modified crowd control algorithm that is room-based.
101 Briefly, all users send their location updates to the room which stores
102 them in a list. At periodic intervals (run off the same timer as for
103 client updates) the room compiles a proximateUser list for each user
104 and sends them out to the users. Each user updates their visibleUser

105 table utilizing the new list. This approach limits the sending of
106 appear/disappear to the client in the same way that location updates
107 are limited.
108
109 Added login limits as a command line argument.
110
111 -----
112 8/28/95
113
114 Modified constants in net_mbuf.h to set MBUF_BLKSIZE to (1024 - 32)
115 and MBUF_MAX_TFR to 1024. These values were suggested by Bo.
116
117 Trying to track down crash bug - during abort sequence a segmentation
118 violation occurs during the propagation of the aborting users teleport
119 message to its proximate users. Removed a few error messages being printed
120 so that the server can be run with its output redirected to a file.
121 Initialized the exiting teleport message's new room to an empty string.
122 On receipt of a teleport message from another user, checked for both
123 entryType == 0 and newRoom != this room, as the condition under which
124 to remove the sending user from our proximate list
125
126 Fixed crash bug on receipt of SIGINT by moving the signal handler for
127 this into main.cc where it now waits until the end of a cycle to terminate.
128
129 Clean up and optimize.
130 - Simplify ProtocolObject by removing the text protocol. This functionality
131 should be provided by a translation process outside the server.
132 - Removed ControlObject and TextControl. When groups are implemented, it
133 should be done using an approach similar to that used for video and
134 audio conferencing.
135 - Changed the name "Attribute" to "Property" to conform to the protocol
136 documentation.
137 - Replaced pervasive use of Strings as object identifiers with a numerical
138 Identifier class which consists of a server number and numerical object
139 identifier. In this implementation, the server number is always 0 and
140 the object identifier is the pointer to the object. This construct
141 supports future extensions to multiple servers, while reducing the overhead
142 involved with String processing. This modification required extensive
143 changes to Message, ObjectList, Room, User, ProtocolObject, and ObjId
144 classes.
145 - Remove loop through all 256 entries of the ObjId table in each call
146 to sendMessage(ObjIdArray), instead only loop through active entries.
147 - Change update frequency to once per second.
148 - Give rooms a two byte numeric identifier by which they are referred to
149 in all commands. Rooms are stored in an array rather than a linked list.
150 Also changed the exit and entry type of the teleport command to a single
151 byte each. Required changes to World, Message, and ProtocolObject classes.
152 - Combine command for appear, disappear and teleport commands.
153 - Improved efficiency of getProximateUsers and doUpdate functions in Room.cc.
154
155 Added statistic logging throughout. Every hour log the number of Users
156 in each Room. Create a log entry for every User that logs in giving

157 login time, avatar, number of text messages sent, logout time.
158 Every week this log file is closed and a new file is opened.
159
160 Check SessionInit properties for protocol = 16 and client software version
161 greater than 95071120 - send NAK for old client or old protocol
162 (ProtocolObject.cc).
163
164 Moved call to setsockopt() in net_comm.cc to occur *before* the bind().
165 This solves the problem whereby the port can't be reused immediately
166 after the server is killed.
167
168 -----
169 10/5/95
170
171 Fixed error in ObjId.cc in the function shortLocOK() which was checking
172 the X coordinate twice and not checking the Y coordinate.
173
174 Fixed error in getProximateUsers where the variable "largest" was not being
175 initialized when searching for the new largest distance member in the table.
176
177 Fixed error in ObjId.cc where an exiting teleport for a given shortObjId
178 could be overwritten by an entering teleport for a given shortObjId.
179
180 -----
181 10/18/95
182
183 Replaced brute-force search for nearest neighbors with a cell-based
184 algorithm (Room.cc).
185
186 Added a length variable to Objectlist and keep a running total of the
187 length of the list (rather than traversing the whole list to find the
188 length each time it is needed).
189
190 Modified the doUpdate function in Room.cc to create lists by making
191 calls directly to the user objects rather than creating them internally
192 and then sending the entire list to the user object. This should reduce
193 the amount of list processing that is done (the biggest cycle hog now).
194
195 Implement a configuration file. The server now creates only one world
196 which is specified in the config file.
197
198 Rooms are created on the fly as they are referenced by clients and deleted
199 when they are empty.
200
201 Implemented two kinds of updates: a full visibility update where the
202 visibleUser table is updated, and a location only update for users already
203 in the visibleUser table.
204
205 Made Location updates received from the client be passed directly via a
206 call to the User object, rather than being passed as a message. The
207 text, teleport, prop, and roomchange objects could be done the same way
208 (anything for which only the client will be changing its own state, and

209 won't be changing the state of other objects).
210
211 Parallelized room updates (both visibility and location). This is #ifdef'd
212 out in this release until the networking layer can be made thread safe to
213 improve performance.
214
215 Added a lot of code to gather timing information. This is also #ifdef'd out.
216 -----
217 10/23/95
218
219 Modified ProtocolObject.cc to detect size errors in LONGLOCCMD, TELEPORTCMD,
220 and ROOMCHNGCMD and log the message in which the error was detected.
221 Modified ProtocolObject.cc to log messages in which an error is detected.
222 Server seems to be getting a lot of confusing messages.
223
224 Modified Room.cc so that it doesn't delete itself when empty (patch to
225 try to solve crash bug - deleting empty rooms should work).
226
227 Modified User so that when a teleport is received, the validity of the new
228 Room is checked before exiting the current Room. This also shouldn't
229 really be necessary - all new rooms should be valid since they are
230 created if they don't exist.
231
232 Modified User so that proximateList is cleared after notifying other Users
233 that a teleport out of the Room has been done. I think this is the change
234 that fixed the crash bug!
235 -----
236 12/13/95
237
238 Removed old networking layer, modified World.cc to handle the select, and
239 ProtocolObject.cc to do the send and recv.
240
241 Added some more multithreading, and some more timing statistics (these
242 are ifdef'd).
243
244 Modified main.cc to catch SIGINT and abort cleanly.
245
246 Modify makefiles and combine subdirectories into one source directory.
247
248 Remove use of String class and switch to SGI compiler.
249
250 Replace use of Baselist in World class (for connection list, user list, and
251 room list) with hash tables. This should speed up whispers (where the user
252 being whispered to must be found in the World) and room changes/teleports
253 (where the room the user is changing to must be found in the world).
254
255 Change name "ProtocolObject" to "Protocol".
256 Add checkName() function to Protocol to check for legality of user name.
257
258 -----
259 Begin Gamma development
260 -----

Explore Litigation Insights

Docket Alarm provides insights to develop a more informed litigation strategy and the peace of mind of knowing you're on top of things.

Real-Time Litigation Alerts



Keep your litigation team up-to-date with **real-time alerts** and advanced team management tools built for the enterprise, all while greatly reducing PACER spend.

Our comprehensive service means we can handle Federal, State, and Administrative courts across the country.

Advanced Docket Research



With over 230 million records, Docket Alarm's cloud-native docket research platform finds what other services can't. Coverage includes Federal, State, plus PTAB, TTAB, ITC and NLRB decisions, all in one place.

Identify arguments that have been successful in the past with full text, pinpoint searching. Link to case law cited within any court document via Fastcase.

Analytics At Your Fingertips



Learn what happened the last time a particular judge, opposing counsel or company faced cases similar to yours.

Advanced out-of-the-box PTAB and TTAB analytics are always at your fingertips.

API

Docket Alarm offers a powerful API (application programming interface) to developers that want to integrate case filings into their apps.

LAW FIRMS

Build custom dashboards for your attorneys and clients with live data direct from the court.

Automate many repetitive legal tasks like conflict checks, document management, and marketing.

FINANCIAL INSTITUTIONS

Litigation and bankruptcy checks for companies and debtors.

E-DISCOVERY AND LEGAL VENDORS

Sync your system to PACER to automate legal marketing.