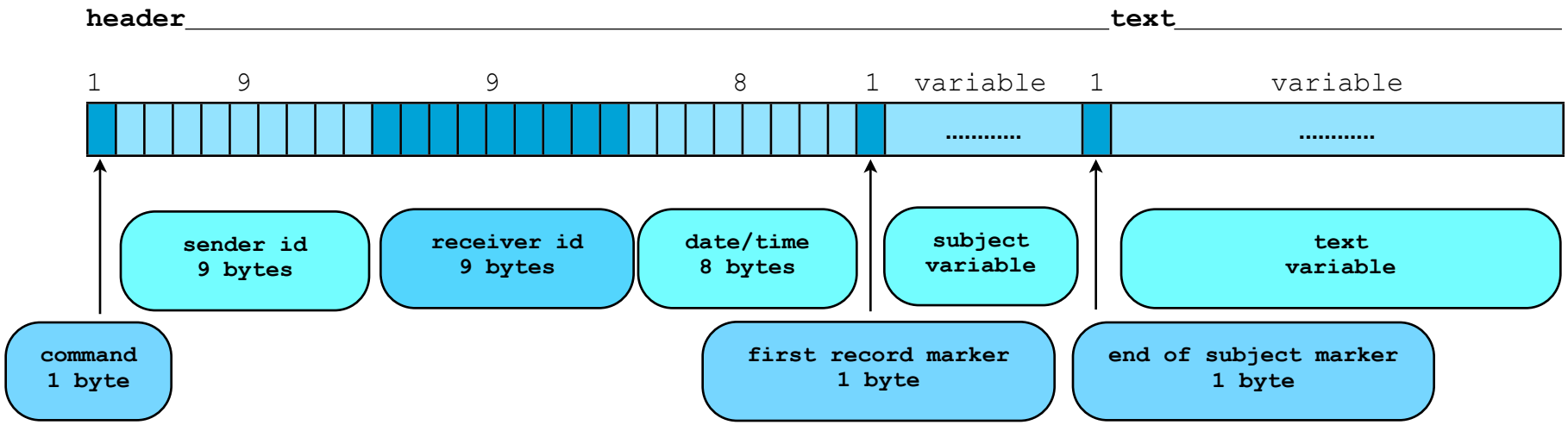
Structure of a message part I: Globals class

A message is composed of a **header** followed by **text**

The **header** contains information about the message being sent, such as the identification of the sender and the receiver, time, subject, etc. With this information we will be able to quickly locate a message in our messages file extremely quickly, even when the messages file contains millions of messages. Some of this information is stored in the message as fixed length and some of it as variable length.

The **text** contains the actual text of the message. It can be of any size.

The structure of a **message** is



field	description
command	The first byte is a command that tells the server what to do with the message. For example, when command has the value 'S' it means that the client is sending the message. When command is 'D' the client wants to delete the message. See the constants below for all possible values of this command
sender id	The sender identification is a unique nine byte identifier.
receiver id	The receiver identification is a unique nine byte identifier.
date/time	The server stamps the date and time when the message is received by the server. The date will be read from the server as an 8-byte long and then transformed when necessary
first record marker	This field identifies if a particular record is the first record of a message. We use this field for recovery in case of a major breakdown. Structures like the availableList and index trees (to come soon) will need to be rebuilt.
subject	Contains the subject of the message
end of subject marker	This is neeeded because the subject is of variable size
text	Body of text. This is of unlimited size. Since the messages are stored in fixed size records we do not need an end of text delimiter here

Globals

The following are integer and character constants. We use symbolic constants as much as possible and minimize the use of numbers in our definitions in Globals

constant	value	notes
COMMAND_POS	0	0
COMMAND_LEN	1	1
CLIENT_ID_LEN	9	9
SENDER_POS	COMMAND_POS + COMMAND_LEN	0 + 1 = 1
SENDER_LEN	CLIENT_ID_LEN	9
RECEIVER_POS	SENDER_POS + SENDER_LEN	1 + 9 = 10
RECEIVER_LEN	CLIENT_ID_LEN	9
DATE_TIME_POS	RECEIVER_POS + RECEIVER_LEN	10 + 9 = 19
DATE_TIME_LEN	8	8
IDENTIFICATION_LEN	SENDER_LEN + RECEIVER_LEN + DATE_TIME_LEN	9 + 9 + 8 = 26
FIRST_RECORD_MARKER_POS	DATE_TIME_POS + DATE_TIME_LEN	19 + 8 = 27
FIRST_RECORD_MARKER_LEN	1	1
END_OF_SUBJECT_MARKER_LEN	1	1
TEXT_LEN	END_OF_SUBJECT_MARKER_LEN + 51	1 + 51 = 52
RECORD_DATA_LEN	COMMAND_LEN + SENDER_LEN + RECEIVER_LEN + DATE_TIME_LEN + FIRST_RECORD_MARKER_LEN + TEXT_LEN	1 + 9 + 9 + 8 + 1 + 52 = **80**

constant	value	notes
INT_LEN	4	4
NEXT_RECORD_LEN	INT_LEN	4
RECORD_LEN	RECORD_DATA_LEN + NEXT_RECORD_LEN	80 + 4 = **84**
FIRST_RECORD_MARKER	'+' to be changed later to 200	first record of message
END_OF_SUBJECT_MARKER	'@' to be changed later to 201	subject is variable size
DELETED	'*' to be changed later to 202	record marked as deleted
SEND_MESSAGE	'S'	command from client to server
DELETE_MESSAGE	'D'	command from client to server
IN_BOX	'I'	command from client to server
OUT_BOX	'O'	command from client to server
SERVER_SHUTDOWN	'Q'	command from client to server