own copy of the document to note which of the collaborators have acknowledged receipt of the document.

[0190] Arrows **1745** and **1750** illustrate a logout scenario. Before broadcasting a LOGOUT message, the computer **252** constructs the LOGOUT message from the predetermined LOGOUT opcode, the username of the collaborator logging out (i.e., Z) of a current editing session, and the UUID of the document. Then the computer **252** being operated by Z transmits the LOGOUT message to the computer **200**, as represented by the arrow **1745**. The computer **252** also transmits the LOGOUT message to the computer **250** being operated by Y, as reference by the arrow **1745**.

[0191] **FIG. 18** is a sequence diagram showing the process of performing an online merge of two versions of a document with both simultaneous and sequential modifications being made to the document. Again, the collaborator X is operating the computer **200** and the collaborator Y is operating the computer **250**. The arrows referenced as **1810**, **1815**, **1820**, and **1825** show a scenario where the collaborators X and Y make simultaneous modifications while online. The computer **200** constructs a MODIFY message from the predetermined MODIFY opcode, the username (X) of the modifier, the UUID of the document, the old version string (0), the new version string (1), and the difference between the two versions of the document. The computer **250** also constructs a MODIFY message from the predetermined MODIFY opcode, from the predetermined MODIFY opcode, the username (Y) of the modifier, the UUID of the document, the old version string (0), the new version string (1), and the patch.

[0192] Each copy of the document preferably contains a history of modifications to the document and a record of who made the modifications and, insofar as the document is aware, which collaborators have been made aware of each of the modifications. Accordingly, as described above, associated with each document is version information comprising a version identifier or version string and a list of "known acknowledgers", one of whom is the author. This is represented in **FIGS. 17, 18**, and **19** by the notation doc(version identifier, known acknowledgers), e.g. doc(0, {X, Y}) to indicate that as far as a particular collaborator is aware, version 0 has been acknowledged by X and Y, but not by Z.

[0193] The computer **200** transmits the constructed MODIFY message to the computer **250** as represented by the arrow **1810**. The MODIFY message includes the difference between the two versions of the document. Then the computer **250** being operated by Y transmits the MODIFY message which has been constructed by the computer **250** to the computer **200**. This MODIFY message includes the patch representing the changes made to the document by Y during the editing session.

[0194] The computers **200** and **250** being operated by X and Y must then construct acknowledgment messages for the modifications that they have sent each other. The computer **200** constructs a modification acknowledgment message, ACK_MOD, from the predetermined ACK_MOD opcode. The ACK_MOD message includes the modifier username (i.e., Y), the username of the acknowledger (i.e., X), the UUID of the document, the version string of the version to which the patch applied, and, optionally, a hash of the patch included with the MODIFY message sent by Y (i.e., for verification purposes). The computer **250** constructs a modification acknowledgment message, ACK_MOD, from the predetermined ACK_MOD opcode. Again, the ACK_MOD message includes the username of the modifier (i.e., X), the username of the acknowledger (i.e., Y), the document UUID, the version string of the version to which the patch applied, and, optionally, a hash of the patch representing the modifications to the document made by X (i.e., again for verification purposes). As represented by the arrow **1820**, the computer **250** being operated by Y transmits the ACK_MOD message to computer **200**. Then, the computer **200** being operated by X transmits the ACK_MOD message to computer **250**, as represented by the arrow **1825**.

[0195] The arrows **1830** and **1835** show the sequence where user X modifies the document, the computer **200** constructs and transmits a MODIFY message, as described above, and the computer **250** being operated by the user Y acknowledges the modification by constructing and transmitting an ACK_MOD message to computer **200**. Similarly, the arrows **1840** and **1845** show the sequence where user Y modifies the document, the computer **250** constructs and transmits a MODIFY message, as described above, and the computer **200** being operated by the user X acknowledges the modification by constructing and transmitting an ACK_MOD message to computer **250** which will update the copy of the document to record X's acknowledgement.

[0196] Note that throughout **FIG. 18** no messages are transmitted to collaborator Z. This is because collaborators X and Y removed collaborator Z from their respective lists of active collaborators in response to the LOGOUT message broadcast by collaborator Z at steps **1745** and **1750**.

[0197] **FIG. 19** is a sequence diagram showing the sequence of events for an online merge of two versions of a document with both simultaneous and sequential modifications. Again, the collaborator X is operating the computer **200** and the collaborator Y is operating the computer **250**. Then, the collaborator Z logs into an editing session using the computer **252**. The arrows **1915**, **1920**, **1925**, **1930**, **1935**, **1940**, **1945**, **1950**, **1955**, **1960**, **1965**, **1970**, **1975**, and **1980** show sequence of event when user Z logs back into the editing session.

[0198] Before attempting to log in, the computer **252** must construct a login message, LOGIN, from the predetermined LOGIN opcode, the username (i.e., Z), and the document UUID. The computer **252** then transmits the LOGIN message to the computer **200**, as represented by the arrow **1915**. The computer **252** also transmits the LOGIN message to the computer **250**, as represented by the arrow **1920**.

[0199] The computers **200** and **250** then acknowledge the LOGIN message transmitted by the computer **252** so that Z can discover which collaborators are online and can provide an update. As illustrated in **FIG. 6**, the computer **200** constructs an ACK_LOGIN message from the predetermined ACK_LOGIN opcode, the username (X), and the document's UUID, and then transmits the message to the computer **252**, as represented by the arrow **1925**. Similarly, the computer **250** constructs an ACK_LOGIN message from the predetermined ACK_LOGIN opcode, the username (Y), and the document's UUID, and then transmits the message to the computer **252**, as represented by the arrow **1930**.

[0200] The computer **252** being operated by Z must now update the version of the document stored locally on the