US 20080086484A1

(19) United States

(12) Patent Application Publication

(10) Pub. No.: US 2008/0086484 A1

(43) Pub. Date: Apr. 10, 2008

Darnell et al.

(54) SERVER SYSTEM FOR SERVING EXTENDED CONTENT FEEDS TO CLIENTS

(76) Inventors: **Benjamin G. Darnell**, Palo Alto, CA (US); **Carl Laurence Gonsalves**, Atherton, CA (US)

Correspondence Address:
MORGAN, LEWIS & BOCKIUS, LLP.
2 PALO ALTO SQUARE, 3000 EL CAMINO REAL
PALO ALTO, CA 94306

(21) Appl. No.: **11/869,493**

(22) Filed: **Oct. 9, 2007**

Related U.S. Application Data

(60) Provisional application No. 60/828,610, filed on Oct. 6, 2006.

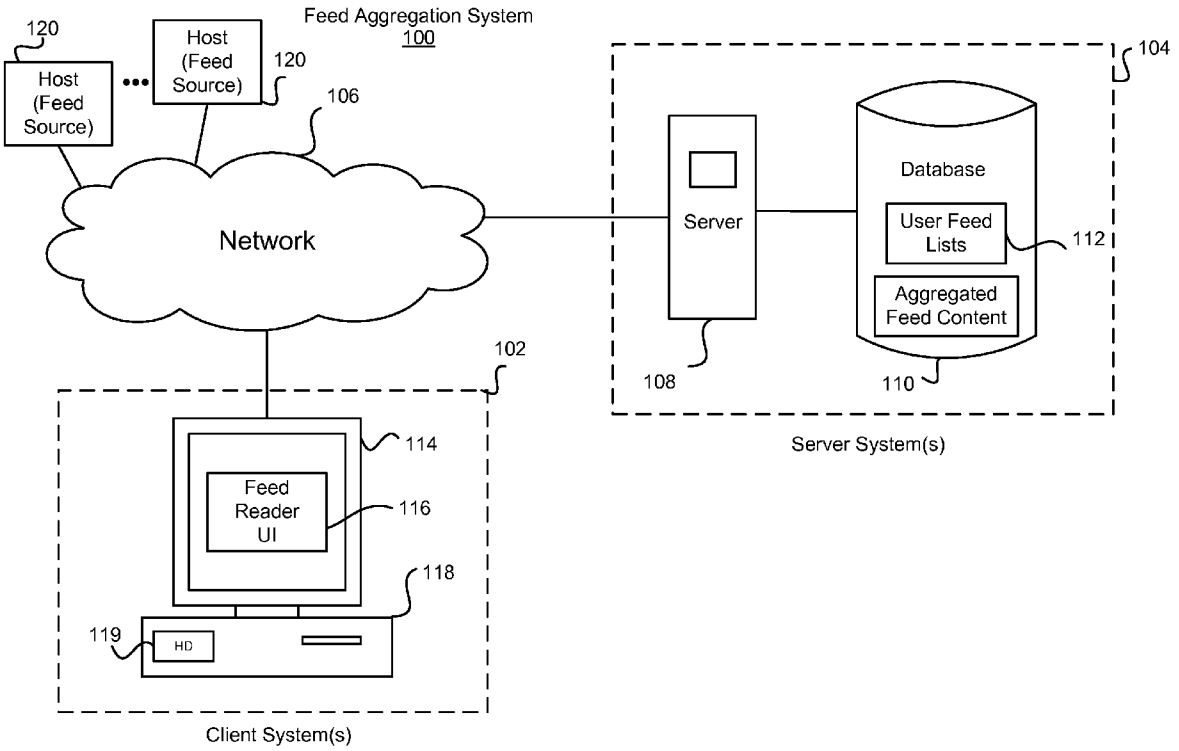
Publication Classification

(51) **Int. Cl.**
G06F 17/30 (2006.01)

(52) **U.S. Cl.** **707/10**; 707/E17.005

(57) **ABSTRACT**

At a system remotely located from a client retrieves, for each of a plurality of content feeds, a currently published set of content items for a respective content feed, and merges the retrieved content items into a previously stored set of previously published content items for the respective content feed. A database storing content items for the respective content feed includes both the currently published set of content items and historical content items no longer published by the respective content feed. In response to a request from the client, the system transmits to the client an ordered list of content items for a specified stream. The specified stream corresponds to one or more content feeds, and the transmitted content items include at least one historical content item for at least one of the content feeds corresponding to the specified stream.



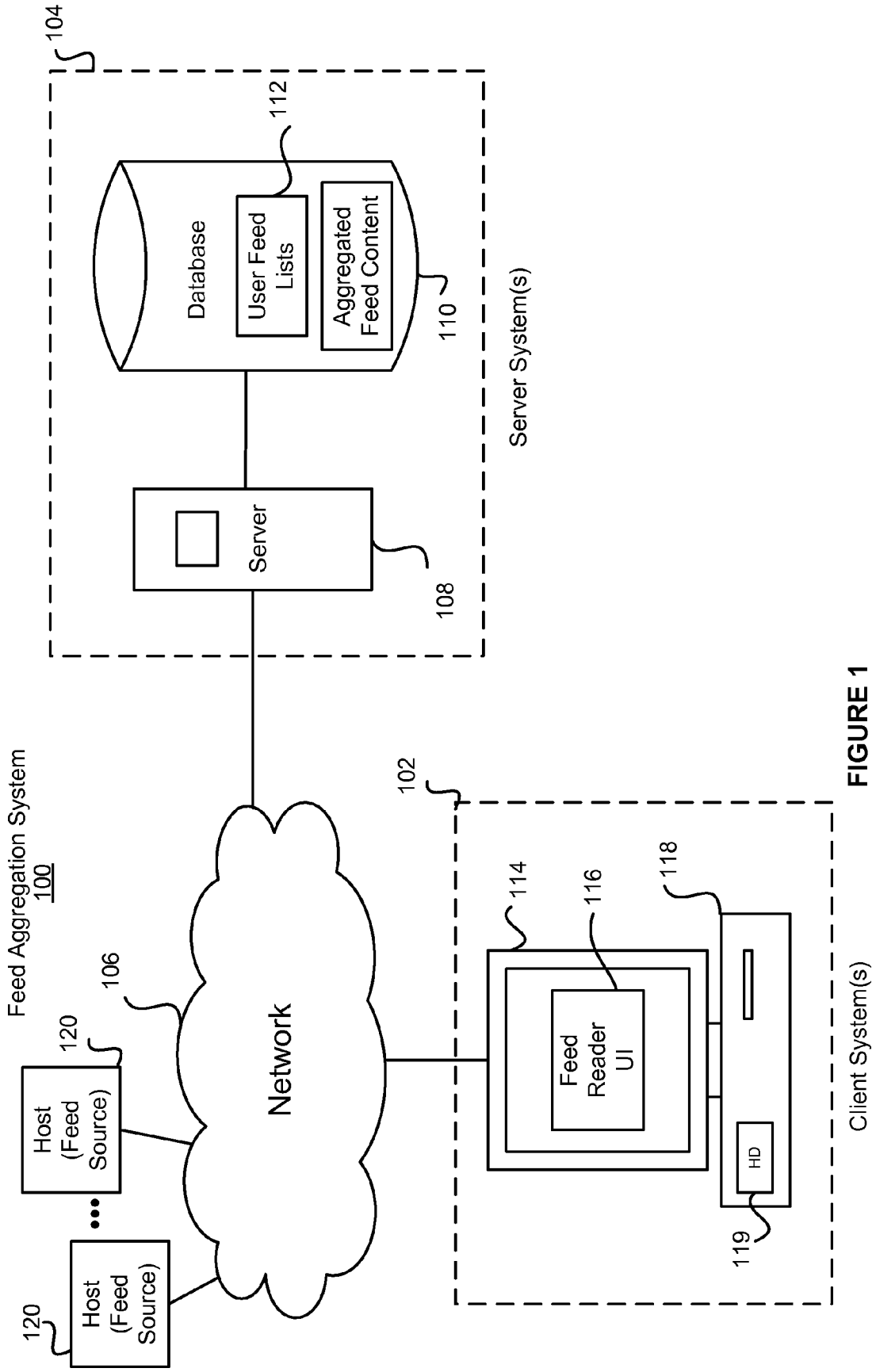

Feed Aggregation System
100
104
Database
User Feed Lists
112
Aggregated Feed Content
110
Server
108
Server System(s)
Network
106
Host (Feed Source)
120
Host (Feed Source)
120
102
114
Feed Reader UI
116
118
HD
119
Client System(s)

FIGURE 1

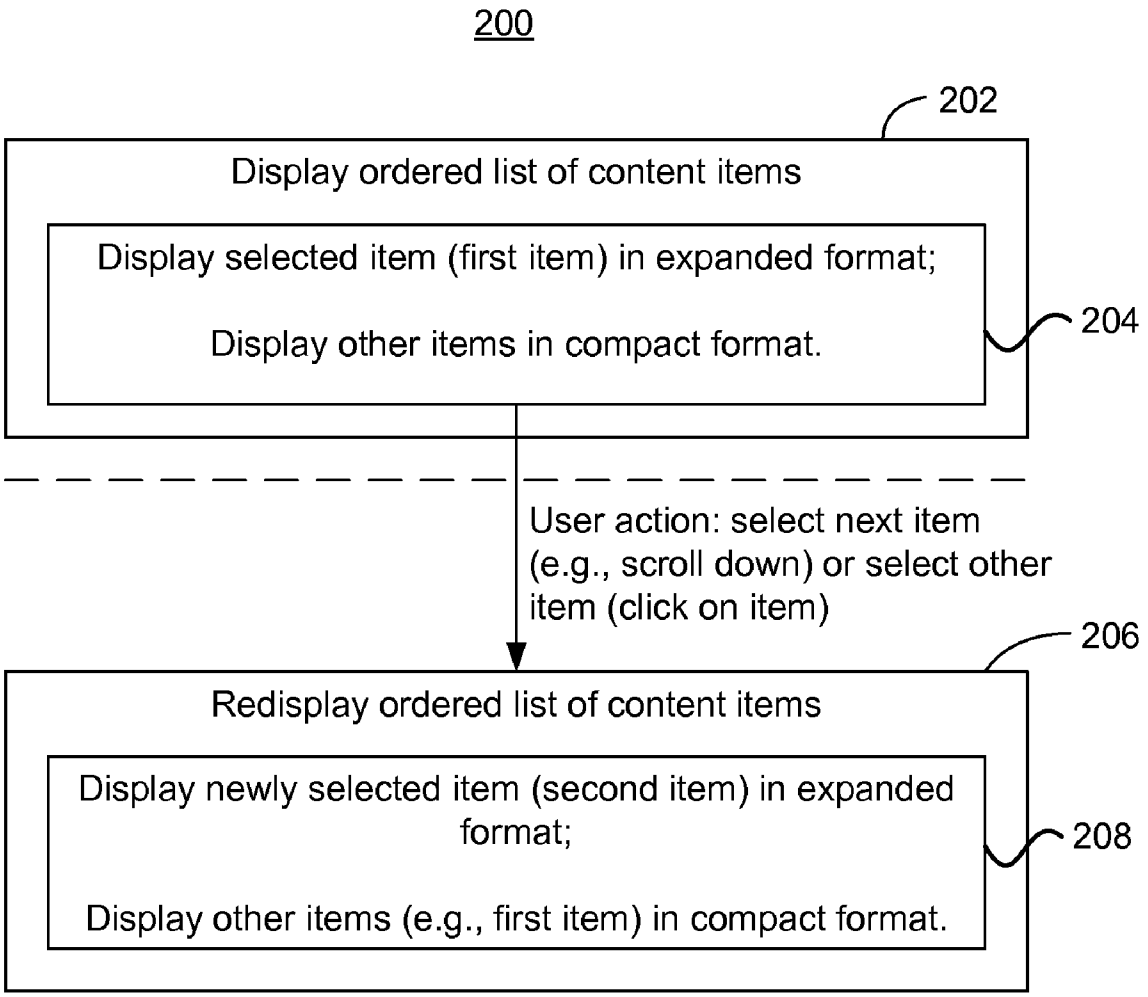

200
202
Display ordered list of content items
Display selected item (first item) in expanded format;
Display other items in compact format.
204
User action: select next item (e.g., scroll down) or select other item (click on item)
206
Redisplay ordered list of content items
Display newly selected item (second item) in expanded format;
Display other items (e.g., first item) in compact format.
208

Figure 2A

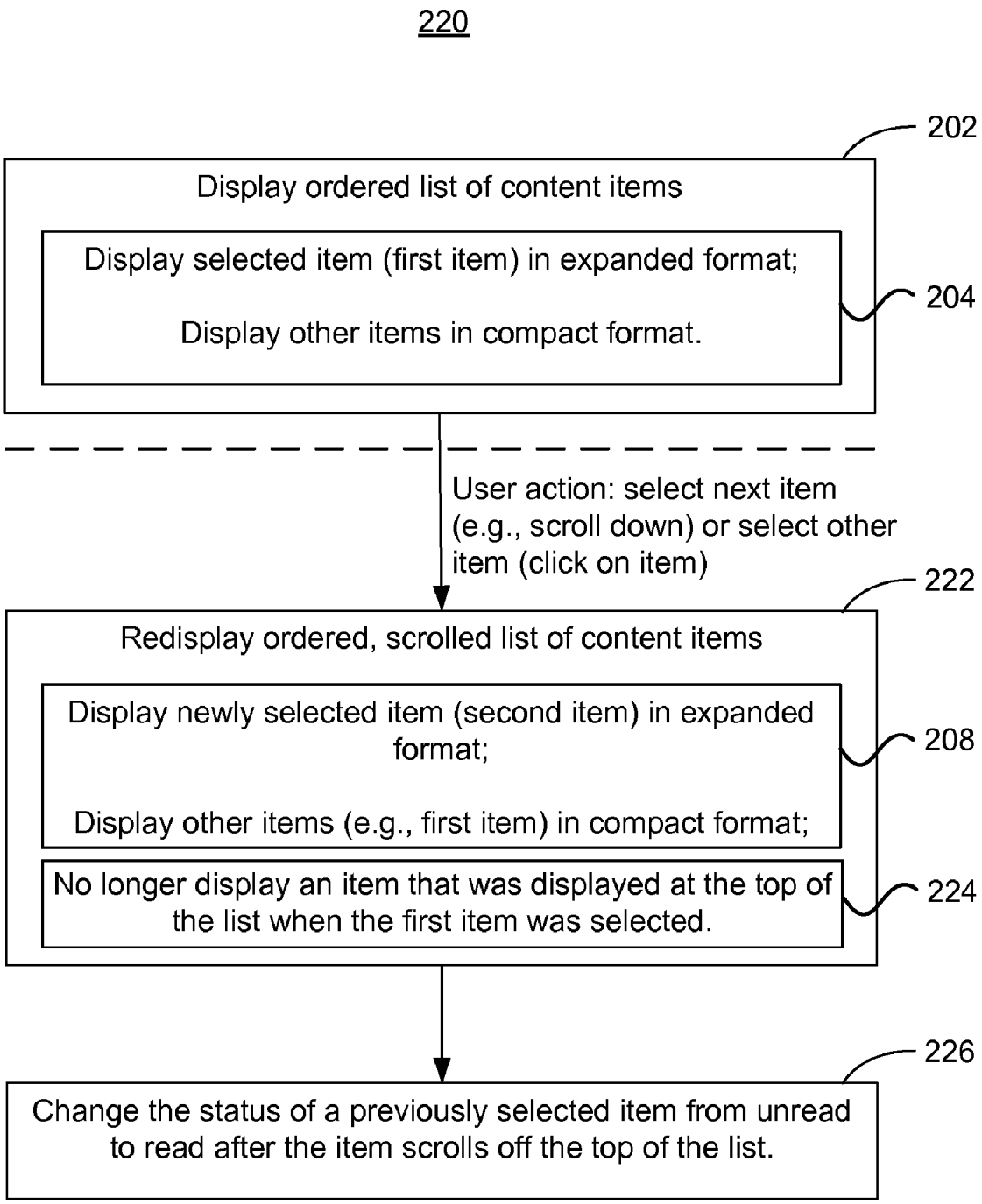

220
202
Display ordered list of content items
Display selected item (first item) in expanded format;
Display other items in compact format.
204
User action: select next item (e.g., scroll down) or select other item (click on item)
222
Redisplay ordered, scrolled list of content items
Display newly selected item (second item) in expanded format;
Display other items (e.g., first item) in compact format;
208
No longer display an item that was displayed at the top of the list when the first item was selected.
224
226
Change the status of a previously selected item from unread to read after the item scrolls off the top of the list.

Figure 2B

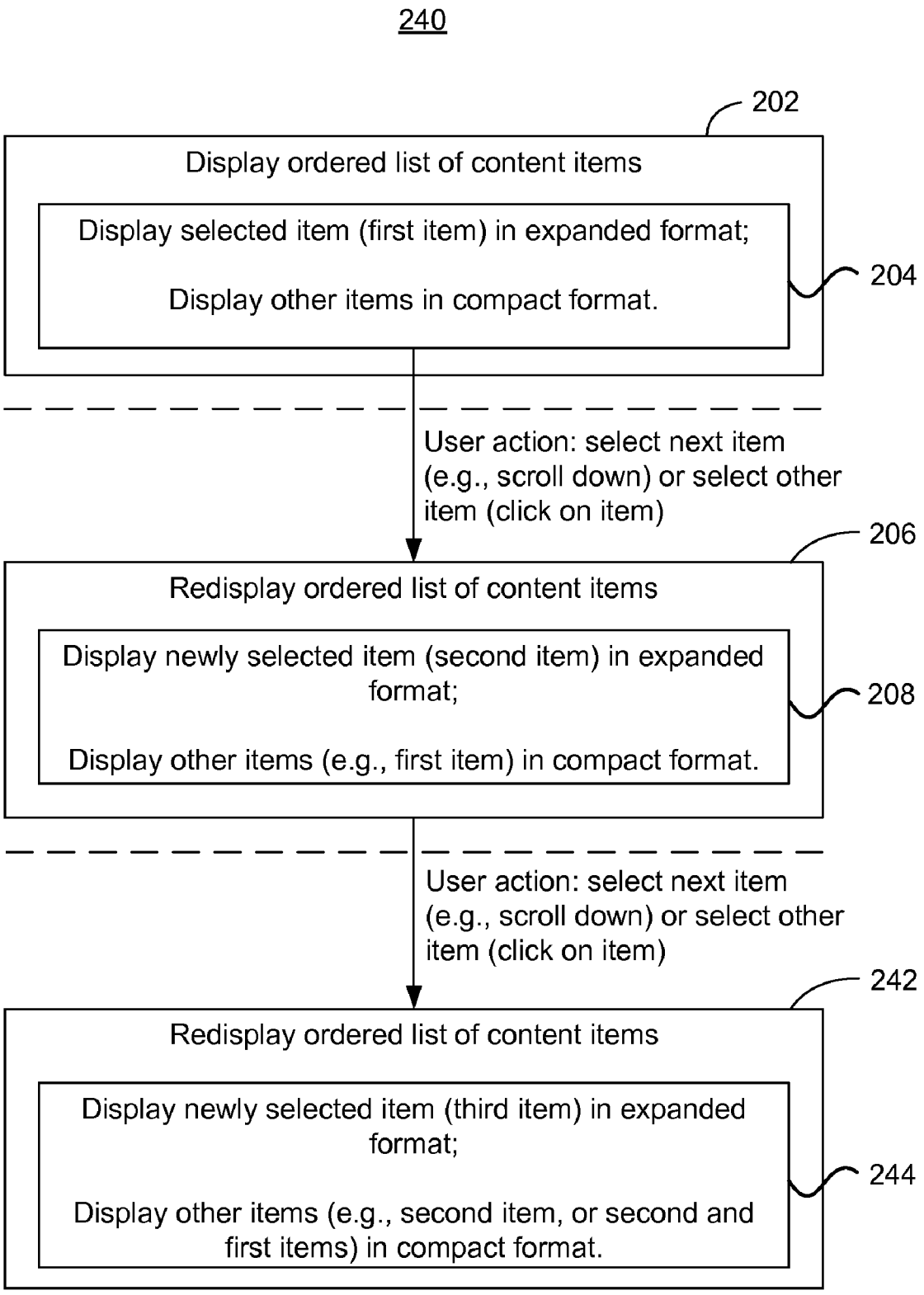

240
202
Display ordered list of content items
Display selected item (first item) in expanded format;
Display other items in compact format.
204
User action: select next item (e.g., scroll down) or select other item (click on item)
206
Redisplay ordered list of content items
Display newly selected item (second item) in expanded format;
Display other items (e.g., first item) in compact format.
208
User action: select next item (e.g., scroll down) or select other item (click on item)
242
Redisplay ordered list of content items
Display newly selected item (third item) in expanded format;
Display other items (e.g., second item, or second and first items) in compact format.
244

Figure 2C

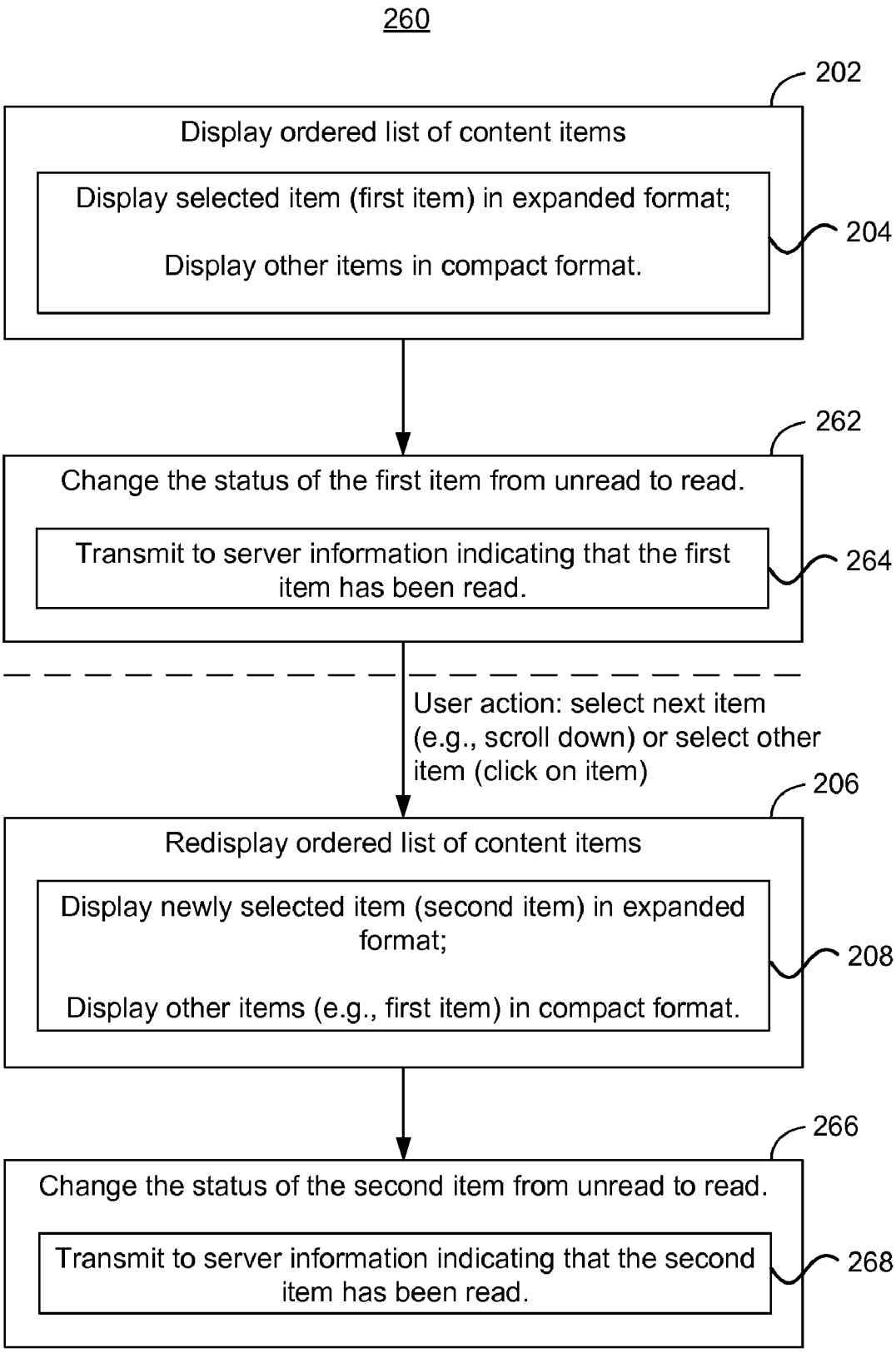

260
202
Display ordered list of content items
Display selected item (first item) in expanded format;
Display other items in compact format.
204
262
Change the status of the first item from unread to read.
Transmit to server information indicating that the first item has been read.
264
User action: select next item (e.g., scroll down) or select other item (click on item)
206
Redisplay ordered list of content items
Display newly selected item (second item) in expanded format;
Display other items (e.g., first item) in compact format.
208
266
Change the status of the second item from unread to read.
Transmit to server information indicating that the second item has been read.
268

Figure 2D

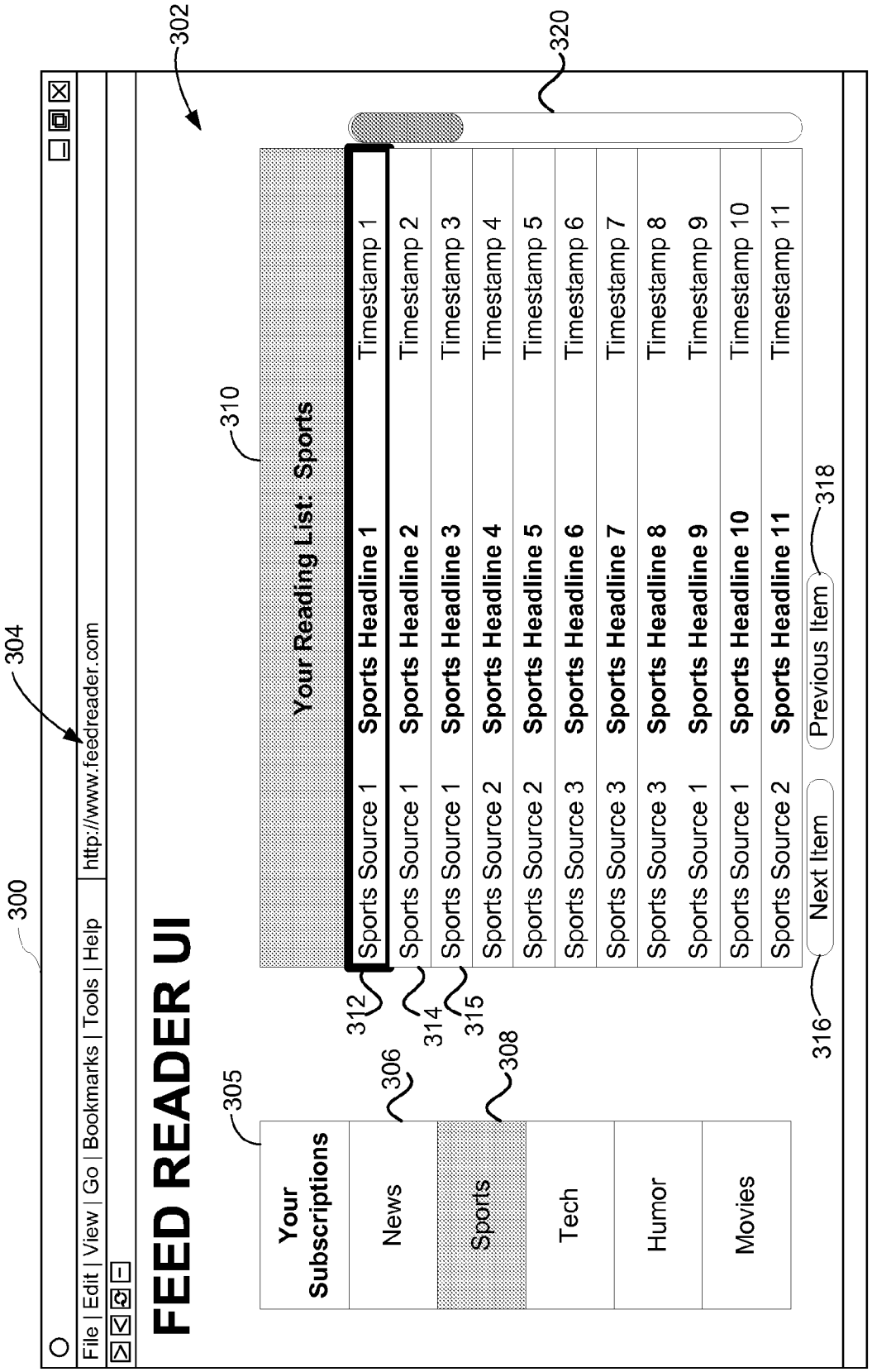

300
304
302
File | Edit | View | Go | Bookmarks | Tools | Help
http://www.feedreader.com
FEED READER UI
305
Your Subscriptions
News
Sports
Tech
Humor
Movies
306
308
310
Your Reading List: Sports
Sports Source 1 Sports Headline 1 Timestamp 1
Sports Source 1 Sports Headline 2 Timestamp 2
Sports Source 1 Sports Headline 3 Timestamp 3
Sports Source 2 Sports Headline 4 Timestamp 4
Sports Source 2 Sports Headline 5 Timestamp 5
Sports Source 3 Sports Headline 6 Timestamp 6
Sports Source 3 Sports Headline 7 Timestamp 7
Sports Source 3 Sports Headline 8 Timestamp 8
Sports Source 1 Sports Headline 9 Timestamp 9
Sports Source 1 Sports Headline 10 Timestamp 10
Sports Source 2 Sports Headline 11 Timestamp 11
312
314
315
316
318
320
Next Item
Previous Item

Figure 3A

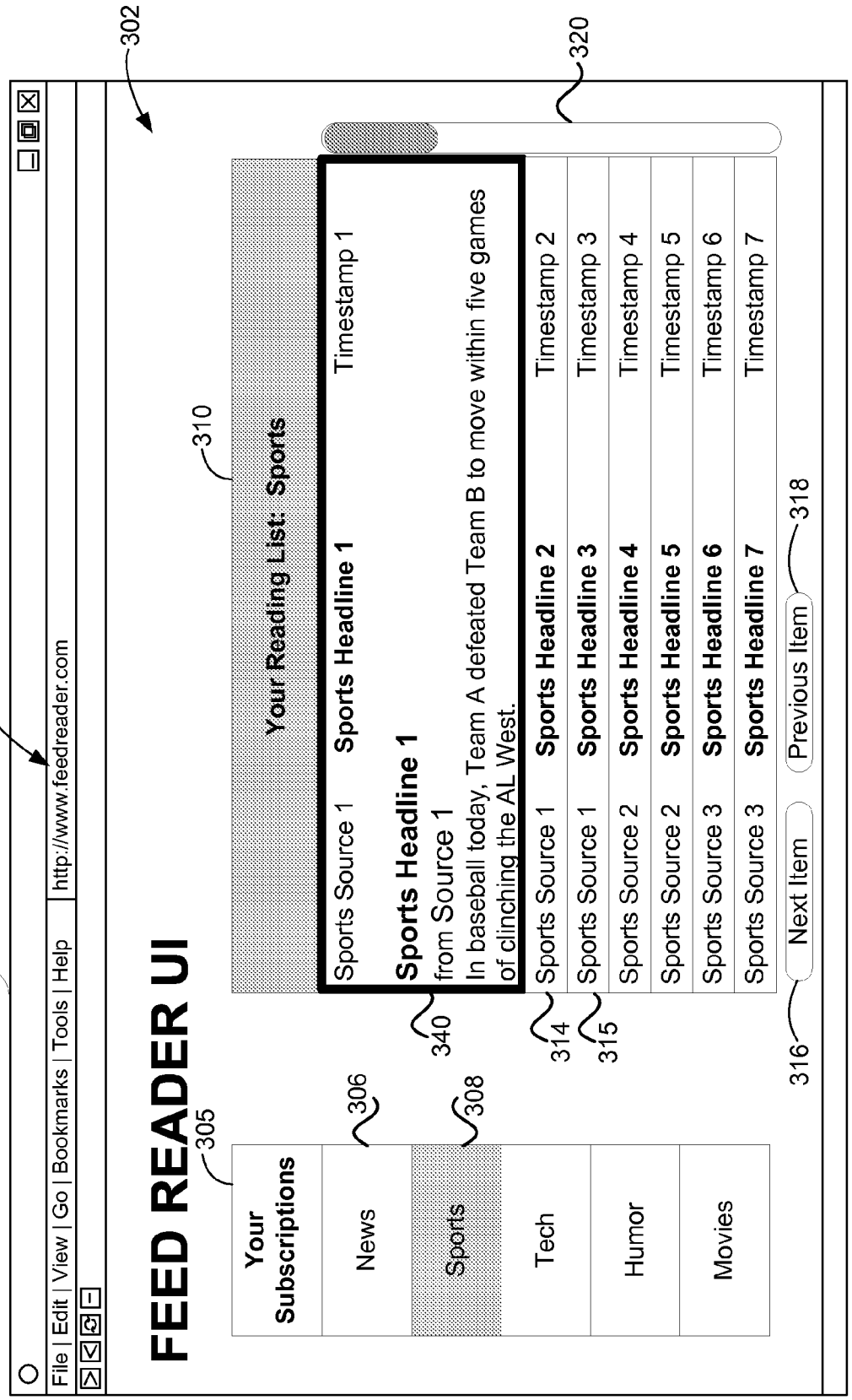

300
302
304
File | Edit | View | Go | Bookmarks | Tools | Help
http://www.feedreader.com
FEED READER UI
305
Your Subscriptions
306
News
308
Sports
Tech
Humor
Movies
310
Your Reading List: Sports
Sports Source 1
Sports Headline 1
Timestamp 1
Sports Headline 1
from Source 1
In baseball today, Team A defeated Team B to move within five games of clinching the AL West.
340
Sports Source 1
Sports Headline 2
Timestamp 2
314
Sports Source 1
Sports Headline 3
Timestamp 3
315
Sports Source 2
Sports Headline 4
Timestamp 4
Sports Source 2
Sports Headline 5
Timestamp 5
Sports Source 3
Sports Headline 6
Timestamp 6
Sports Source 3
Sports Headline 7
Timestamp 7
316
Next Item
Previous Item
318
320

Figure 3B

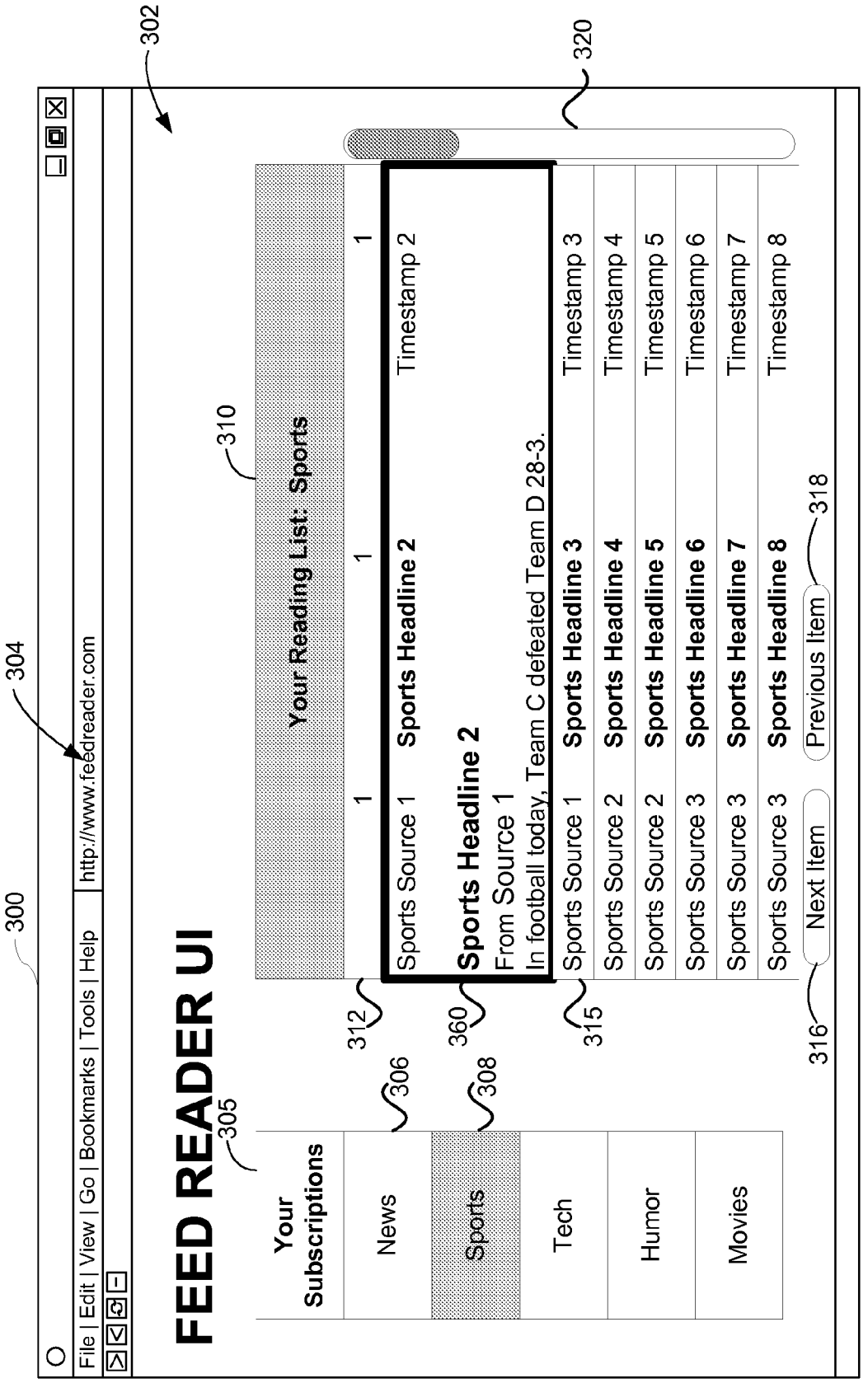

300
302
304
File | Edit | View | Go | Bookmarks | Tools | Help
http://www.feedreader.com
FEED READER UI
305
Your Subscriptions
News
306
Sports
308
Tech
Humor
Movies
310
Your Reading List: Sports
312
Sports Source 1
1
Sports Headline 2
1
Timestamp 2
1
360
Sports Headline 2
From Source 1
In football today, Team C defeated Team D 28-3.
315
Sports Source 1
Sports Headline 3
Timestamp 3
Sports Source 2
Sports Headline 4
Timestamp 4
Sports Source 2
Sports Headline 5
Timestamp 5
Sports Source 3
Sports Headline 6
Timestamp 6
Sports Source 3
Sports Headline 7
Timestamp 7
Sports Source 3
Sports Headline 8
Timestamp 8
316
Next Item
318
Previous Item
320

Figure 3C

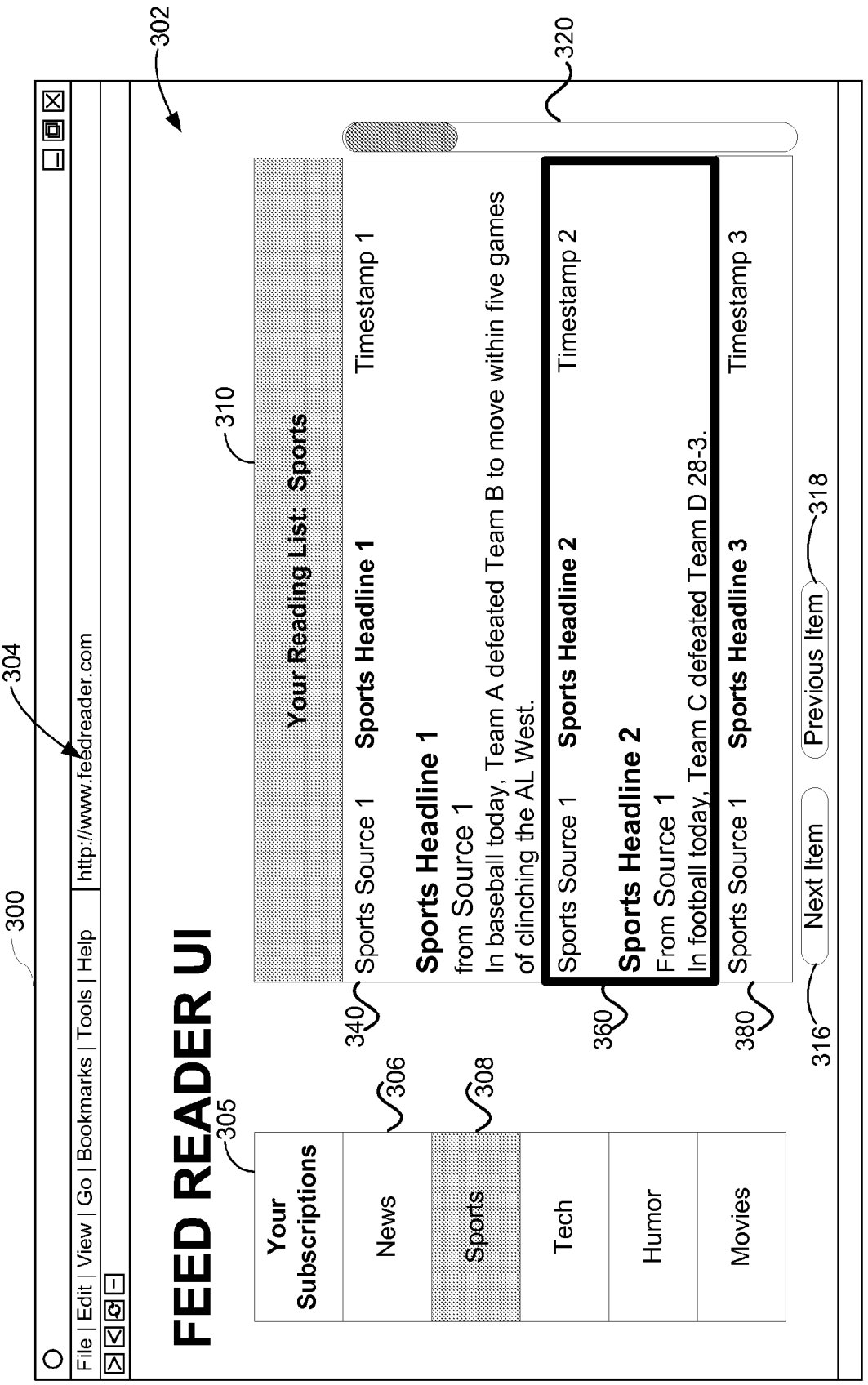

300
302
304
File | Edit | View | Go | Bookmarks | Tools | Help
http://www.feedreader.com
FEED READER UI
305
Your Subscriptions
News
306
Sports
308
Tech
Humor
Movies
310
Your Reading List: Sports
Sports Source 1
Sports Headline 1
Timestamp 1
340
Sports Headline 1
from Source 1
In baseball today, Team A defeated Team B to move within five games of clinching the AL West.
Sports Source 1
Sports Headline 2
Timestamp 2
360
Sports Headline 2
From Source 1
In football today, Team C defeated Team D 28-3.
Sports Source 1
Sports Headline 3
Timestamp 3
380
320
Next Item
316
Previous Item
318

Figure 3D

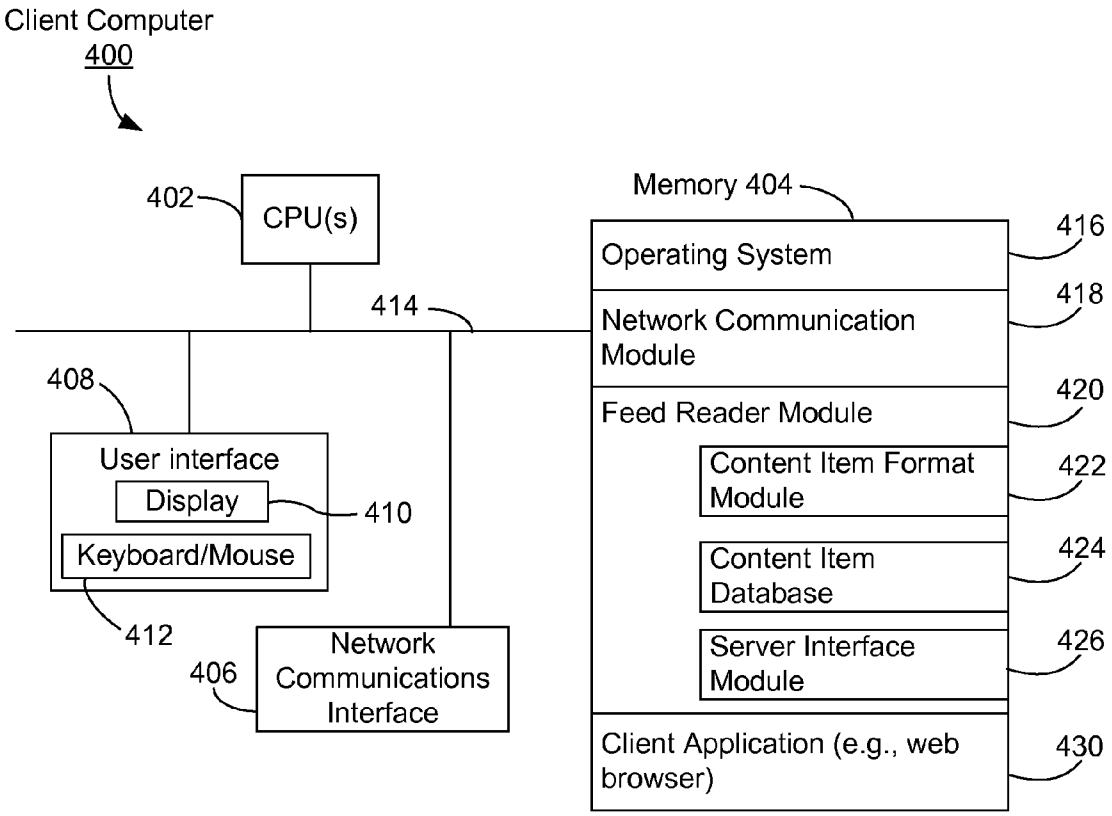


Figure 4

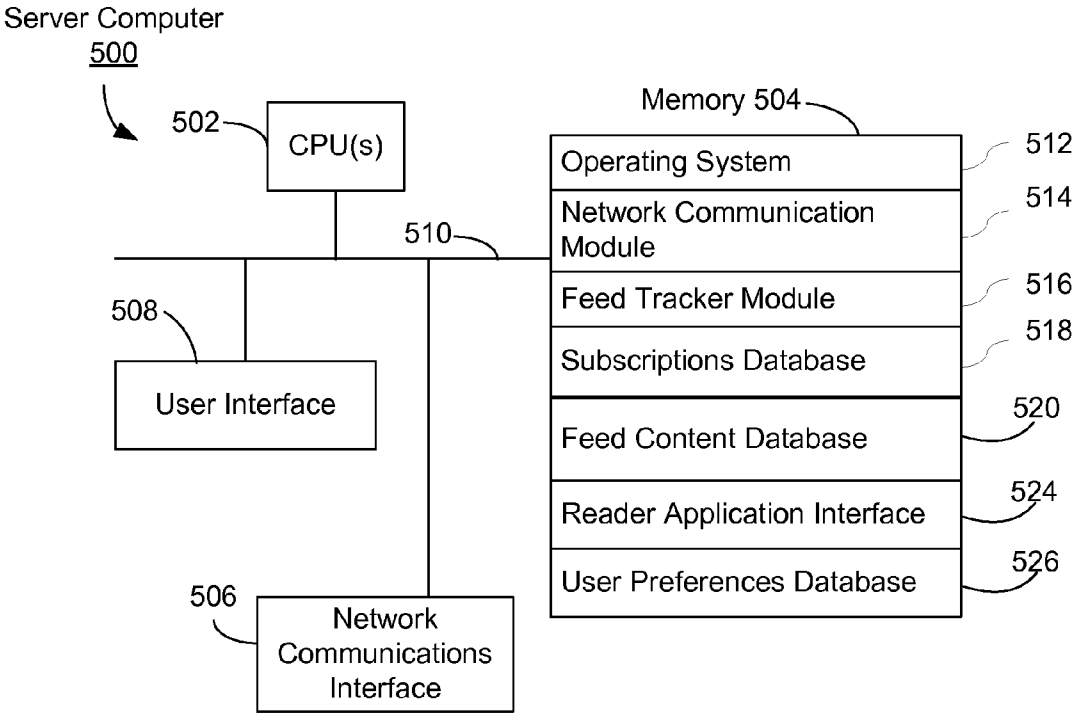


Figure 5

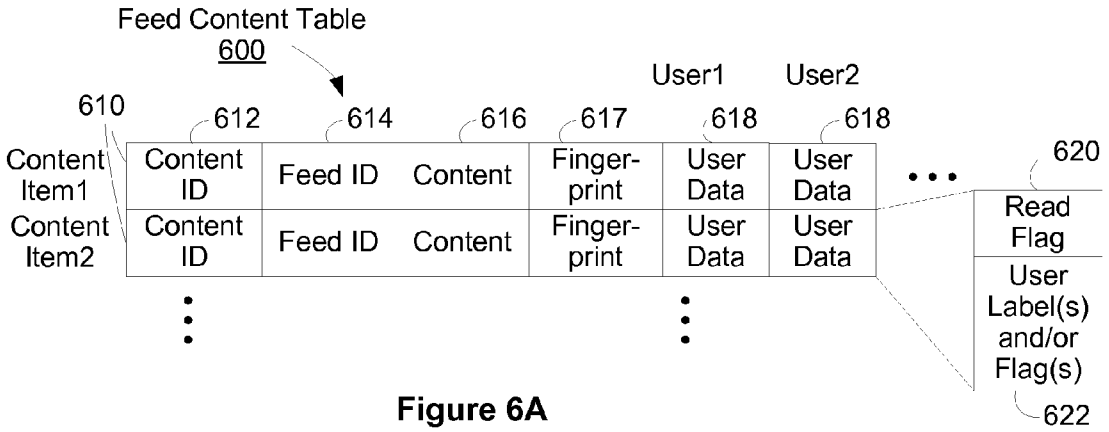


Figure 6A

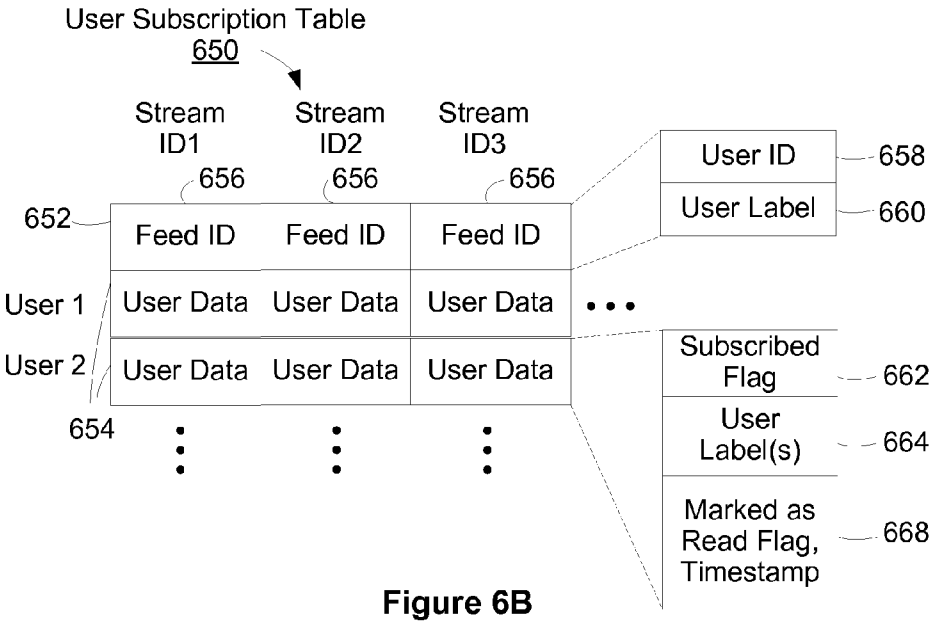


Figure 6B

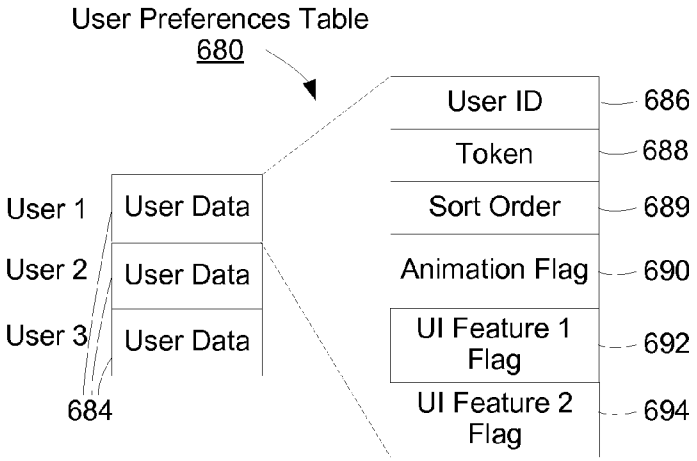


Figure 6C

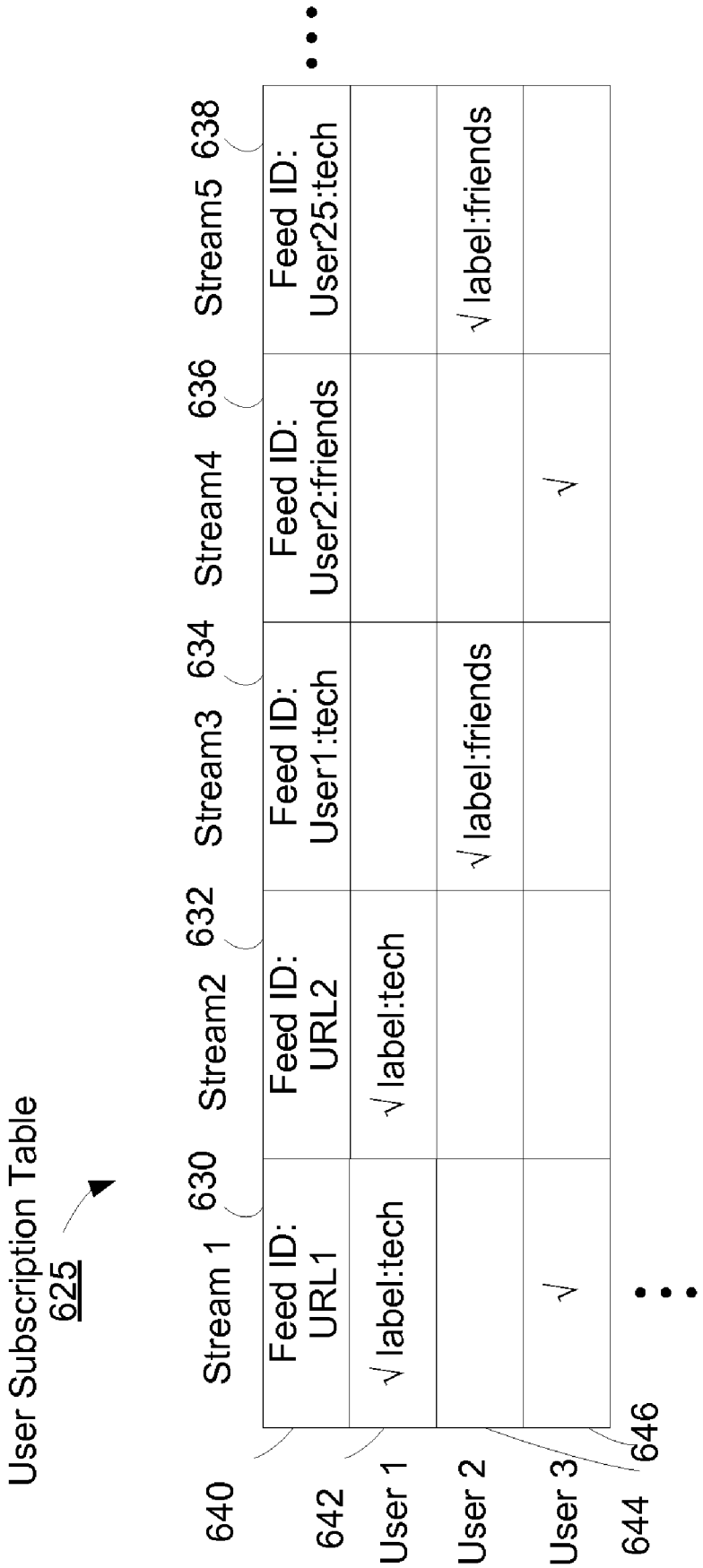

User Subscription Table
625
Stream 1
630
Stream2
632
Stream3
634
Stream4
636
Stream5
638
Feed ID: URL1
Feed ID: URL2
Feed ID: User1:tech
Feed ID: User2:friends
Feed ID: User25:tech
√ label:tech
√ label:tech
√ label:friends
√ label:friends
√
640
642
User 1
User 2
User 3
644
646

Figure 6D

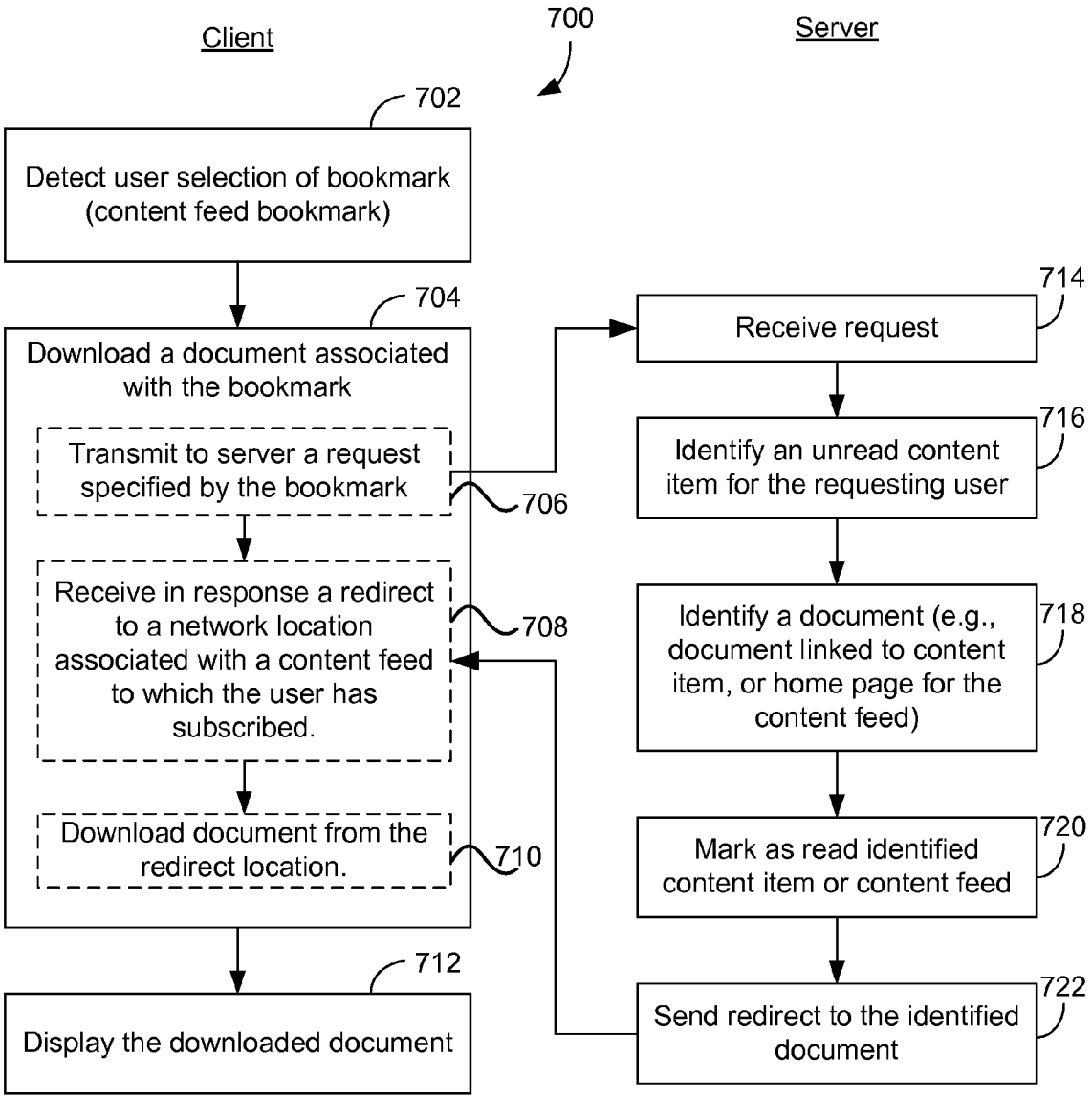

Client
700
Server
702
Detect user selection of bookmark (content feed bookmark)
704
Download a document associated with the bookmark
Transmit to server a request specified by the bookmark
706
Receive in response a redirect to a network location associated with a content feed to which the user has subscribed.
708
Download document from the redirect location.
710
712
Display the downloaded document
714
Receive request
716
Identify an unread content item for the requesting user
718
Identify a document (e.g., document linked to content item, or home page for the content feed)
720
Mark as read identified content item or content feed
722
Send redirect to the identified document

Figure 7A

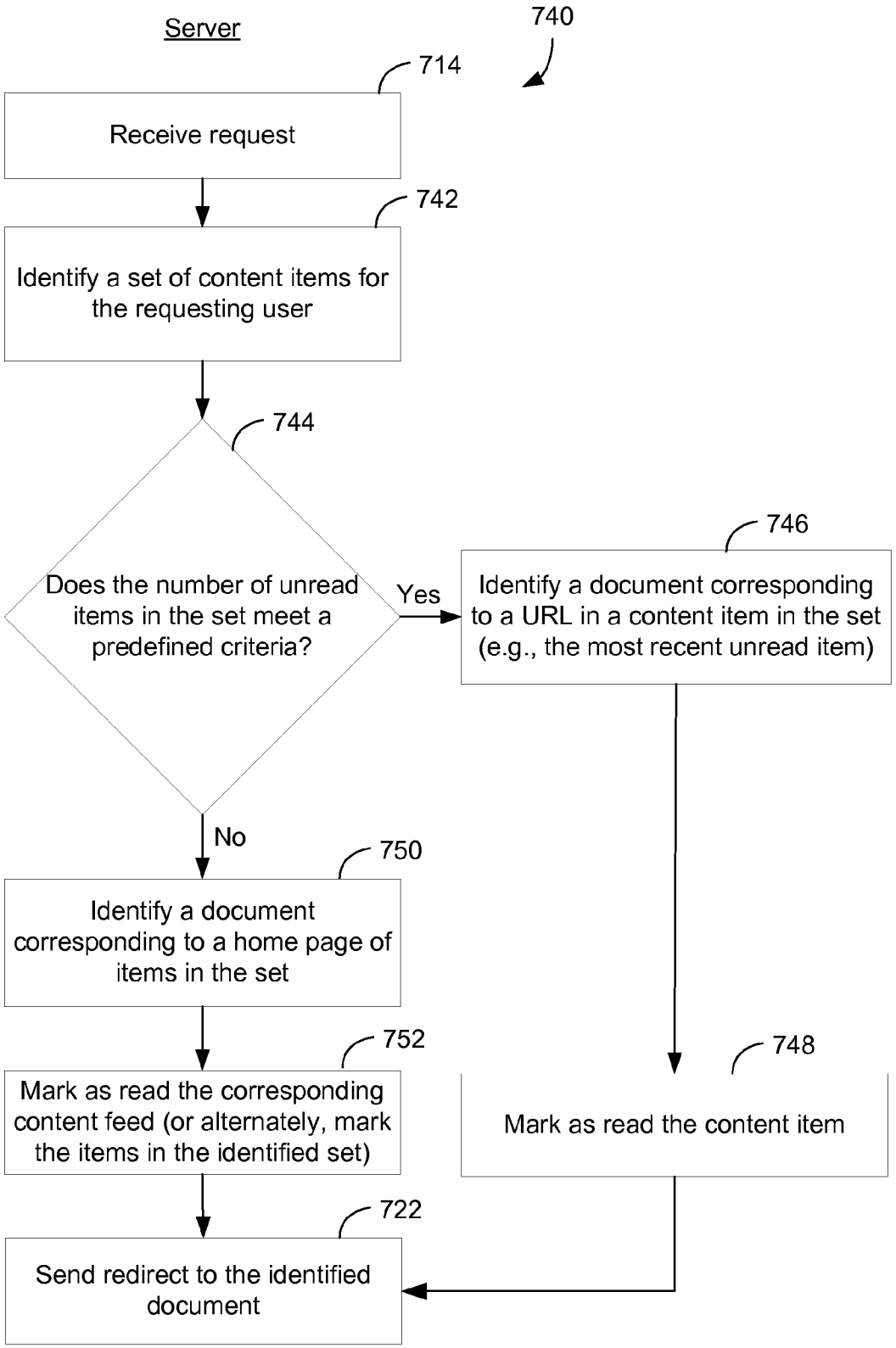

Server
740
714
Receive request
742
Identify a set of content items for the requesting user
744
Does the number of unread items in the set meet a predefined criteria?
Yes
746
Identify a document corresponding to a URL in a content item in the set (e.g., the most recent unread item)
No
750
Identify a document corresponding to a home page of items in the set
752
Mark as read the corresponding content feed (or alternately, mark the items in the identified set)
748
Mark as read the content item
722
Send redirect to the identified document

Figure 7B

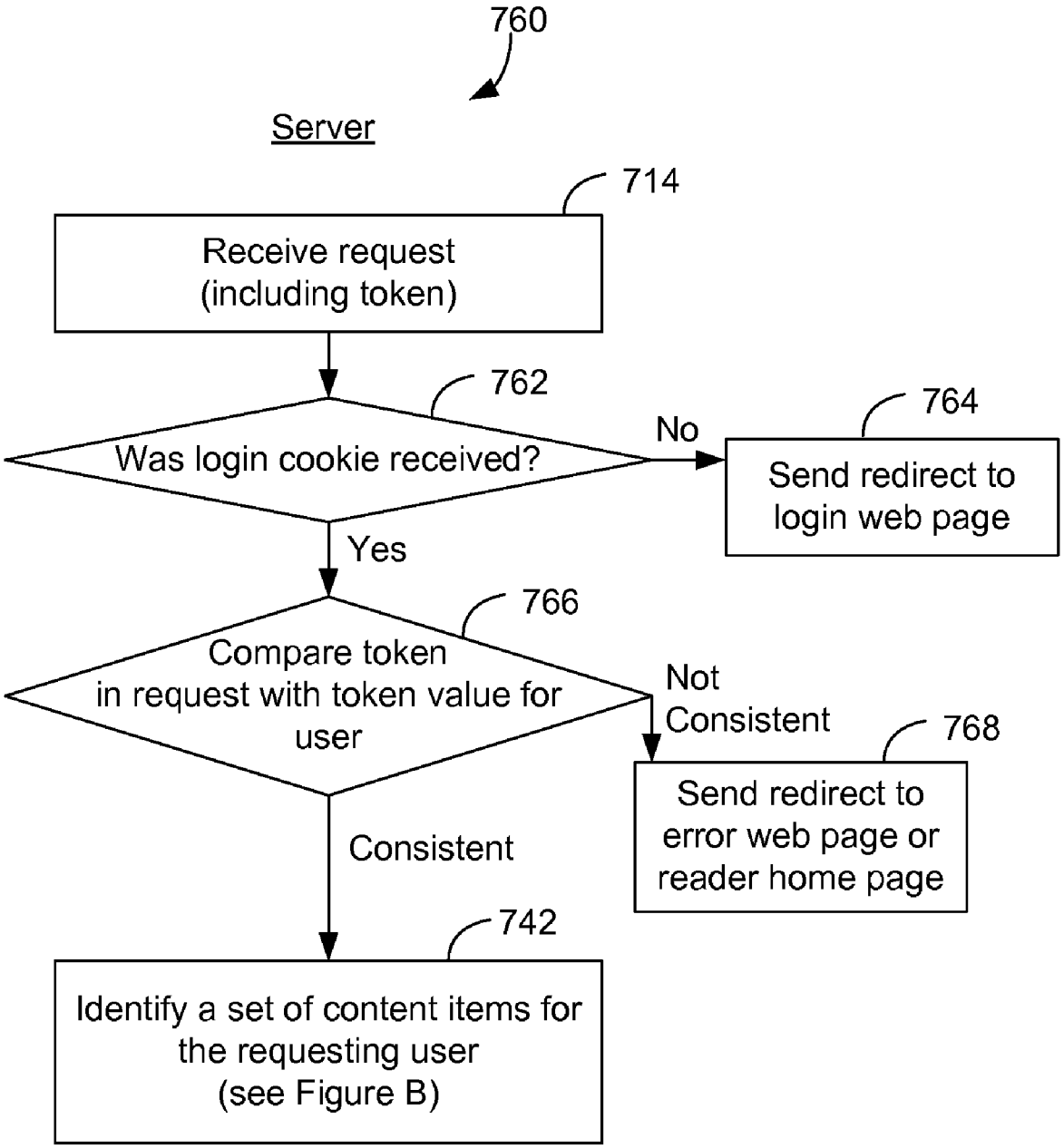

760
Server
714
Receive request (including token)
762
Was login cookie received?
No
764
Send redirect to login web page
Yes
766
Compare token in request with token value for user
Not Consistent
768
Send redirect to error web page or reader home page
Consistent
742
Identify a set of content items for the requesting user (see Figure B)

Figure 7C

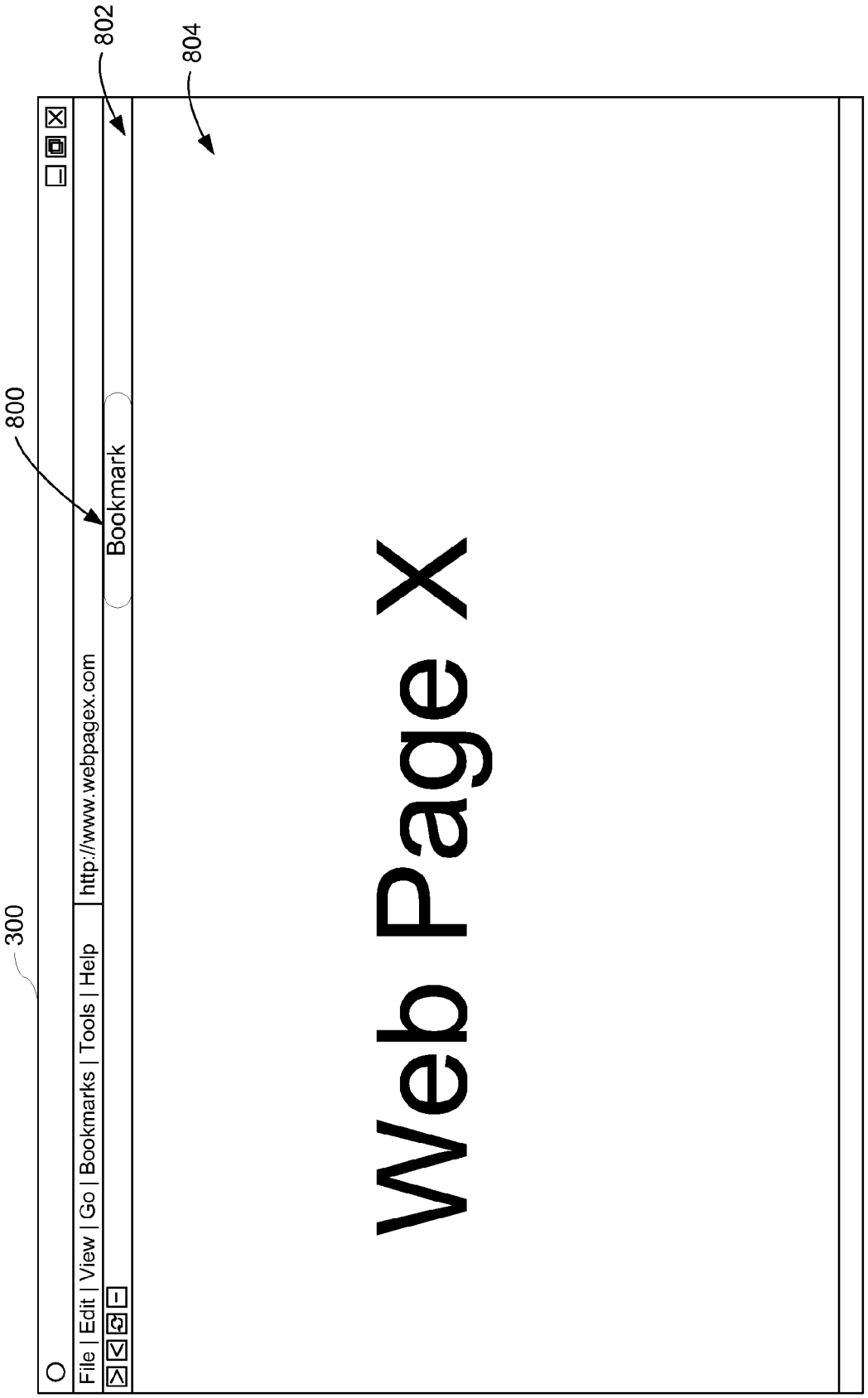

300
800
802
804
File | Edit | View | Go | Bookmarks | Tools | Help
http://www.webpagex.com
Bookmark
Web Page X

Figure 8A

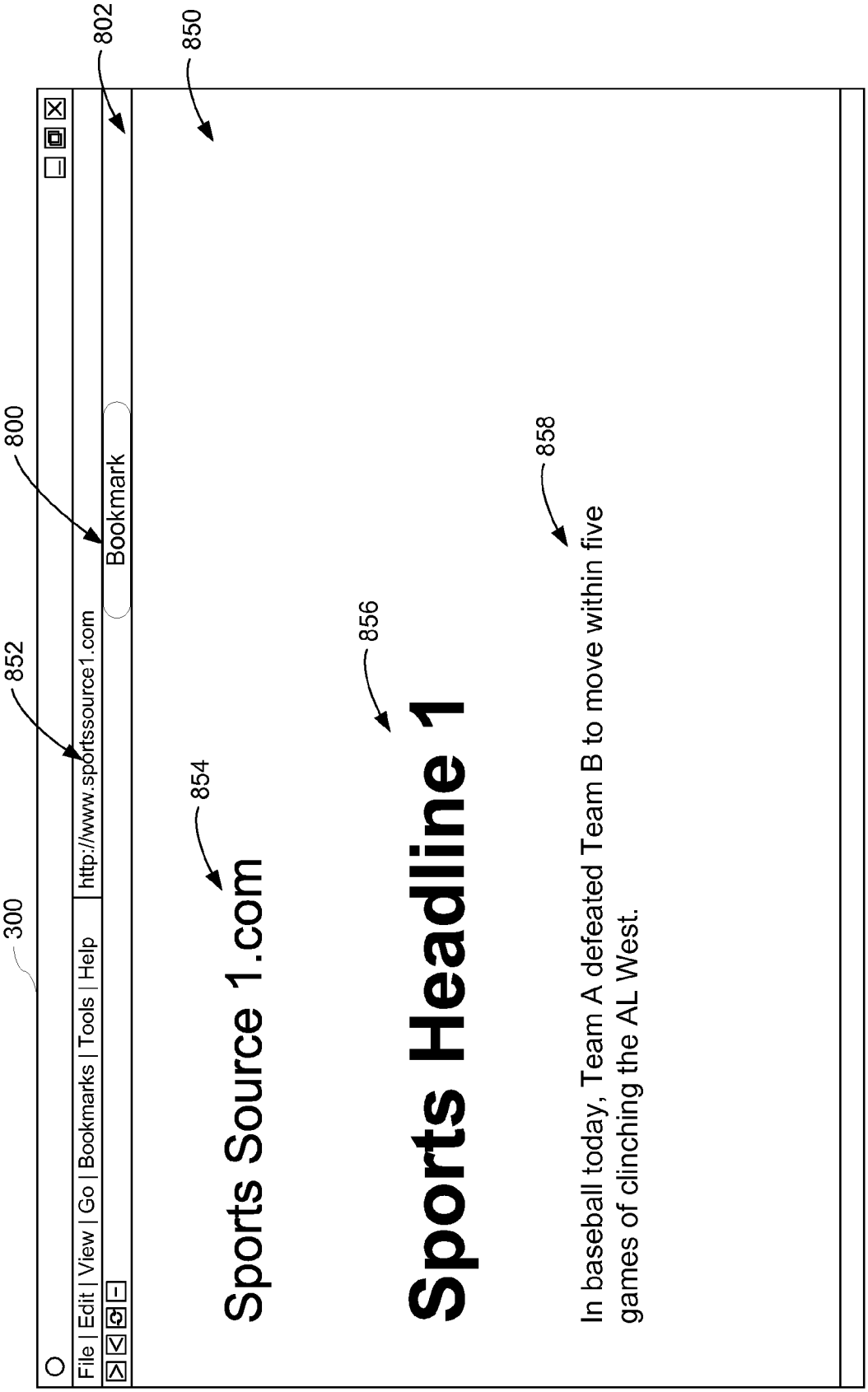

300
852
800
802
850
File | Edit | View | Go | Bookmarks | Tools | Help
http://www.sportssource1.com
Bookmark
854
Sports Source 1.com
856
Sports Headline 1
858
In baseball today, Team A defeated Team B to move within five games of clinching the AL West.

Figure 8B

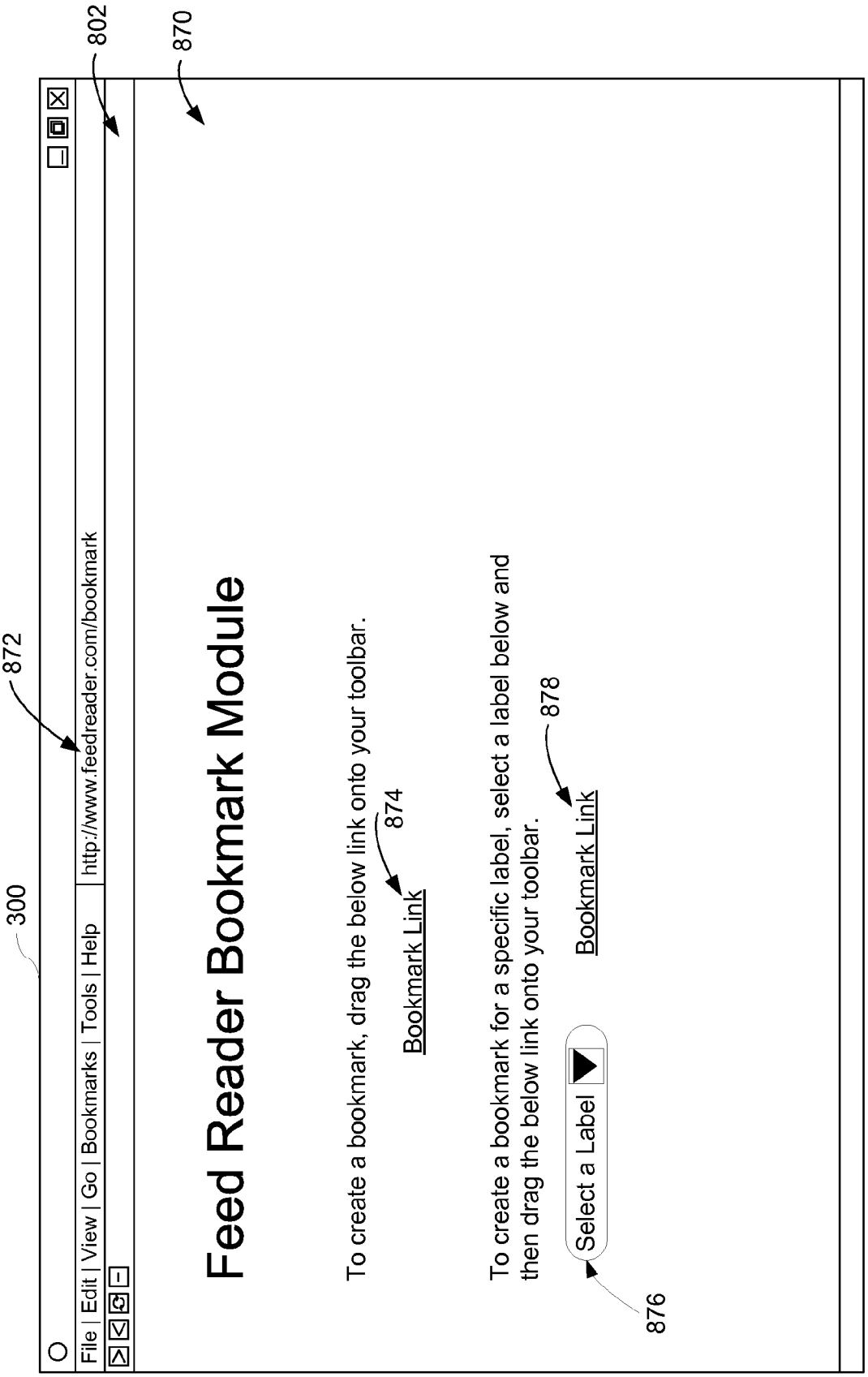

300
872
802
870
File | Edit | View | Go | Bookmarks | Tools | Help
http://www.feedreader.com/bookmark
Feed Reader Bookmark Module
To create a bookmark, drag the below link onto your toolbar.
874
Bookmark Link
To create a bookmark for a specific label, select a label below and then drag the below link onto your toolbar.
878
Bookmark Link
Select a Label
876

Figure 8C

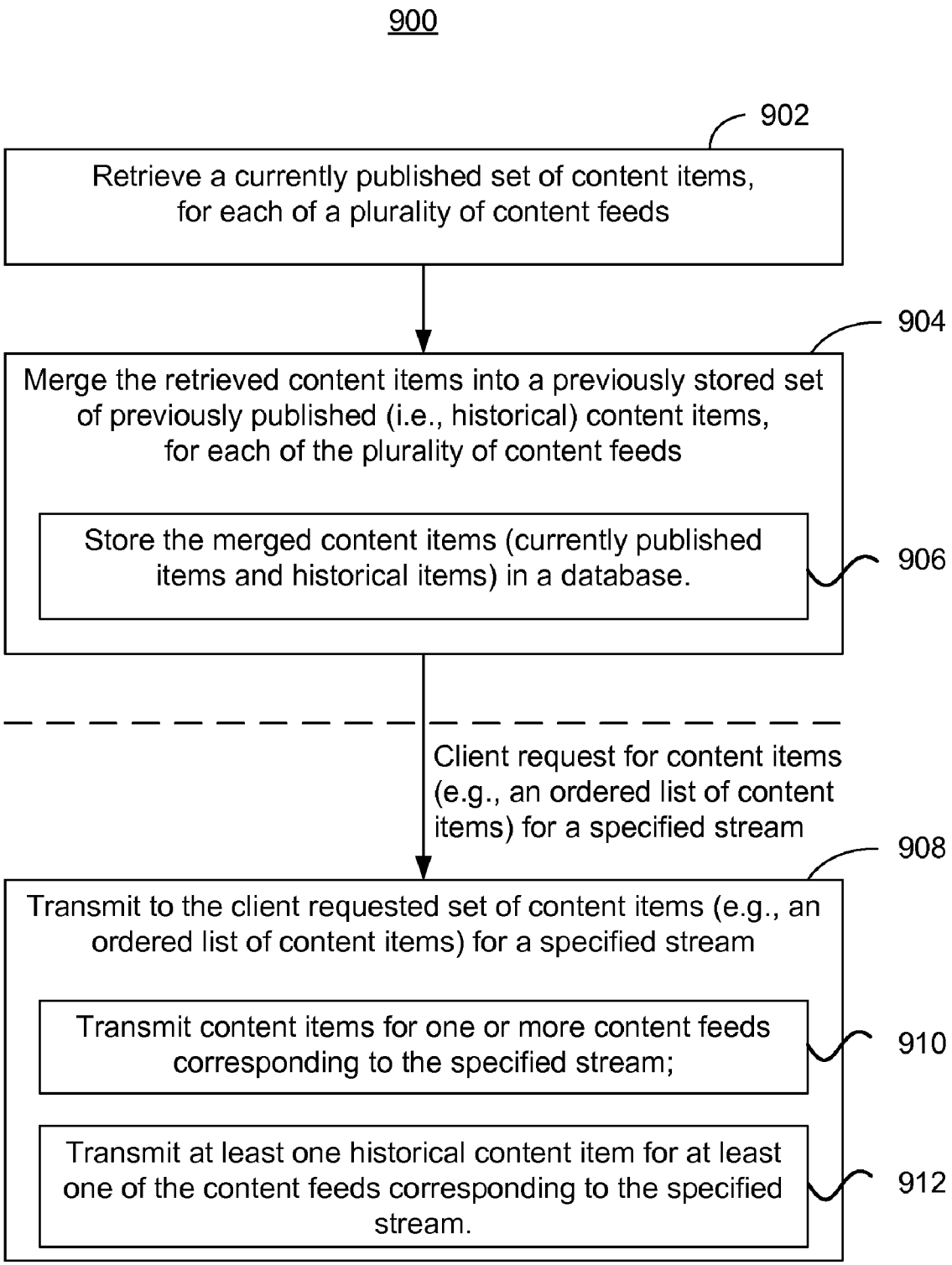

900
902
Retrieve a currently published set of content items, for each of a plurality of content feeds
904
Merge the retrieved content items into a previously stored set of previously published (i.e., historical) content items, for each of the plurality of content feeds
Store the merged content items (currently published items and historical items) in a database.
906
Client request for content items (e.g., an ordered list of content items) for a specified stream
908
Transmit to the client requested set of content items (e.g., an ordered list of content items) for a specified stream
Transmit content items for one or more content feeds corresponding to the specified stream;
910
Transmit at least one historical content item for at least one of the content feeds corresponding to the specified stream.
912

Figure 9A

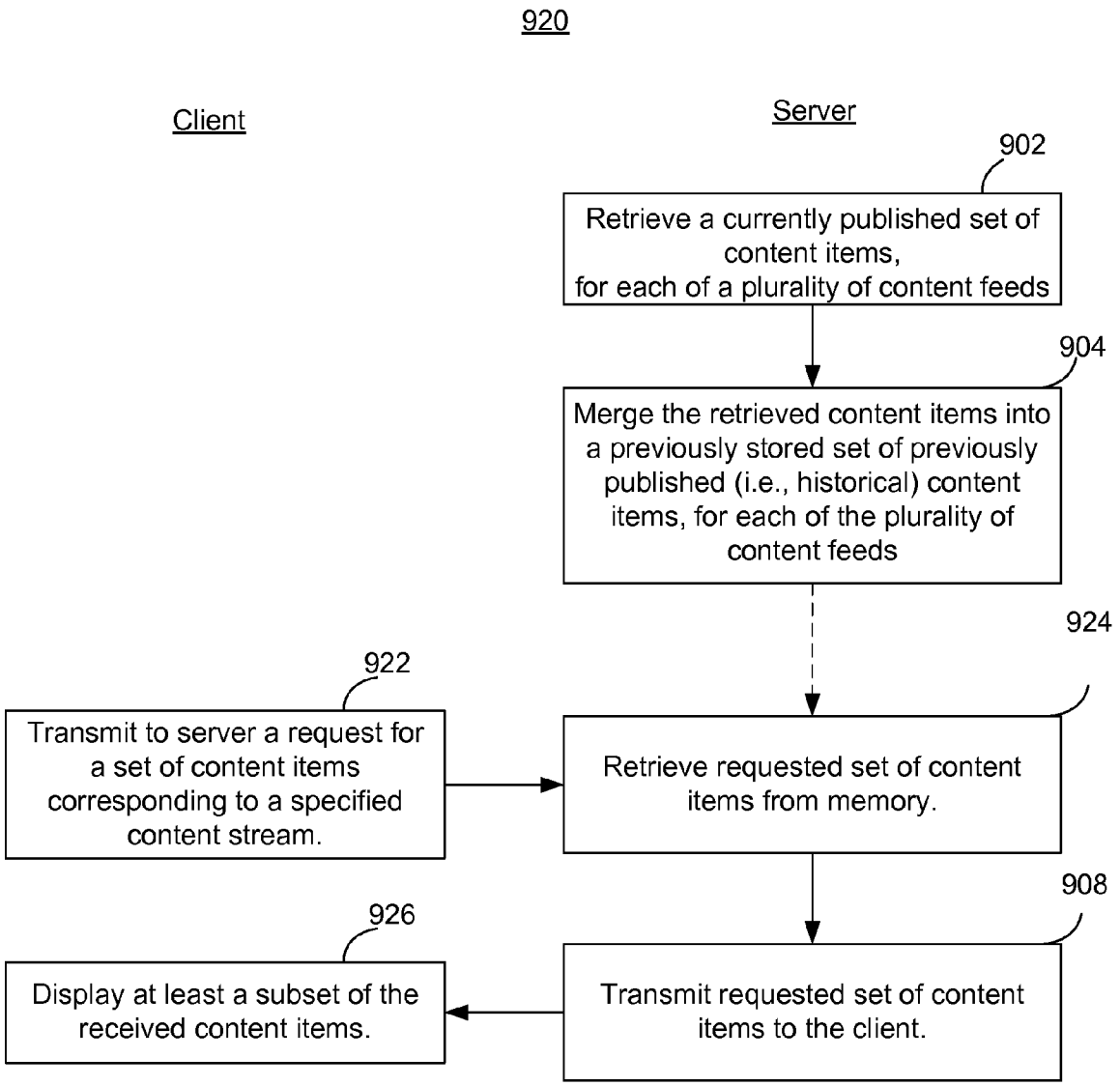

920
Client
Server
902
Retrieve a currently published set of content items, for each of a plurality of content feeds
904
Merge the retrieved content items into a previously stored set of previously published (i.e., historical) content items, for each of the plurality of content feeds
924
922
Transmit to server a request for a set of content items corresponding to a specified content stream.
Retrieve requested set of content items from memory.
908
926
Display at least a subset of the received content items.
Transmit requested set of content items to the client.

Figure 9B

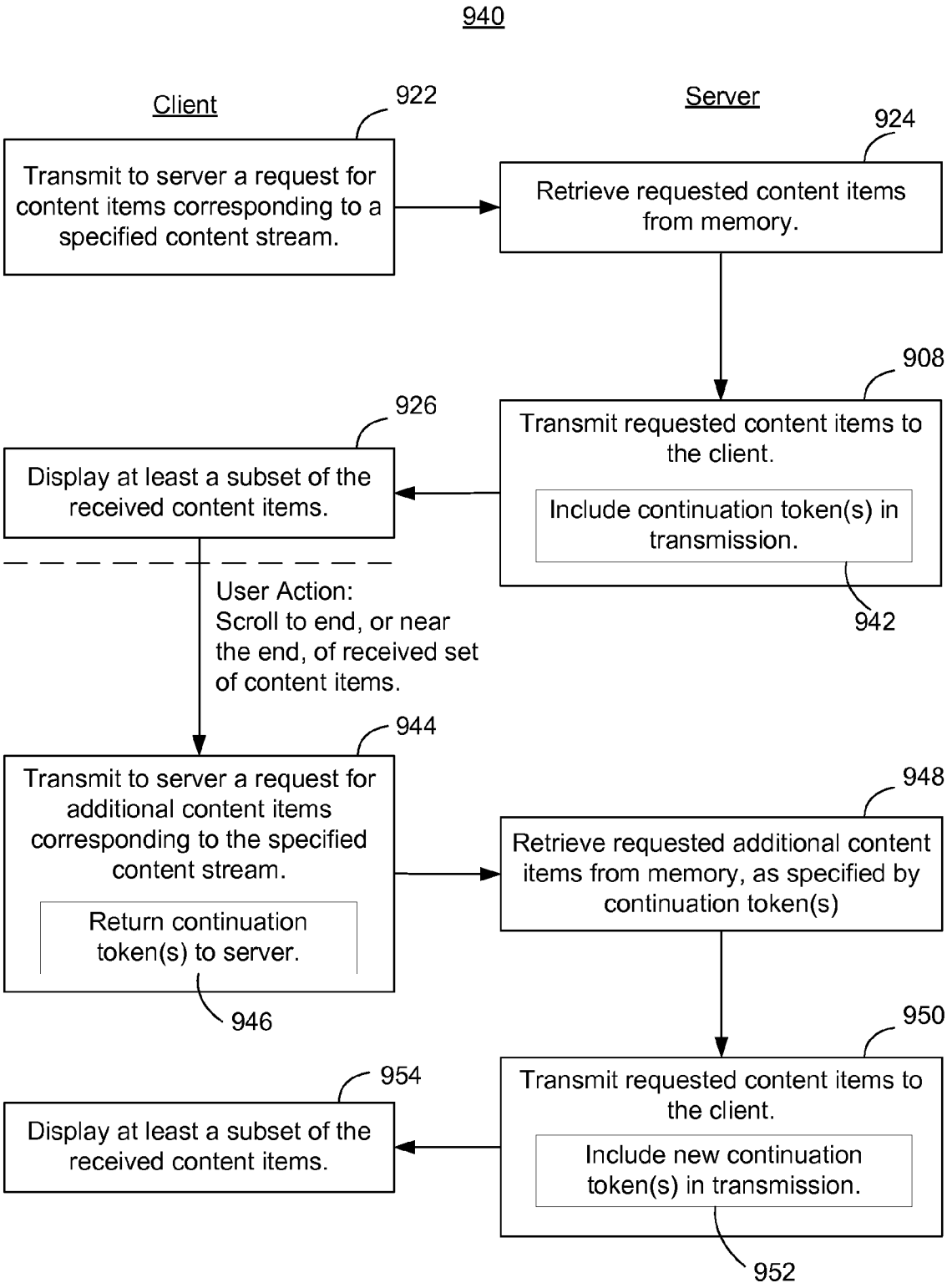

940
Client
Server
922
Transmit to server a request for content items corresponding to a specified content stream.
924
Retrieve requested content items from memory.
908
Transmit requested content items to the client.
Include continuation token(s) in transmission.
942
926
Display at least a subset of the received content items.
User Action:
Scroll to end, or near the end, of received set of content items.
944
Transmit to server a request for additional content items corresponding to the specified content stream.
Return continuation token(s) to server.
946
948
Retrieve requested additional content items from memory, as specified by continuation token(s)
950
Transmit requested content items to the client.
Include new continuation token(s) in transmission.
952
954
Display at least a subset of the received content items.

Figure 9C

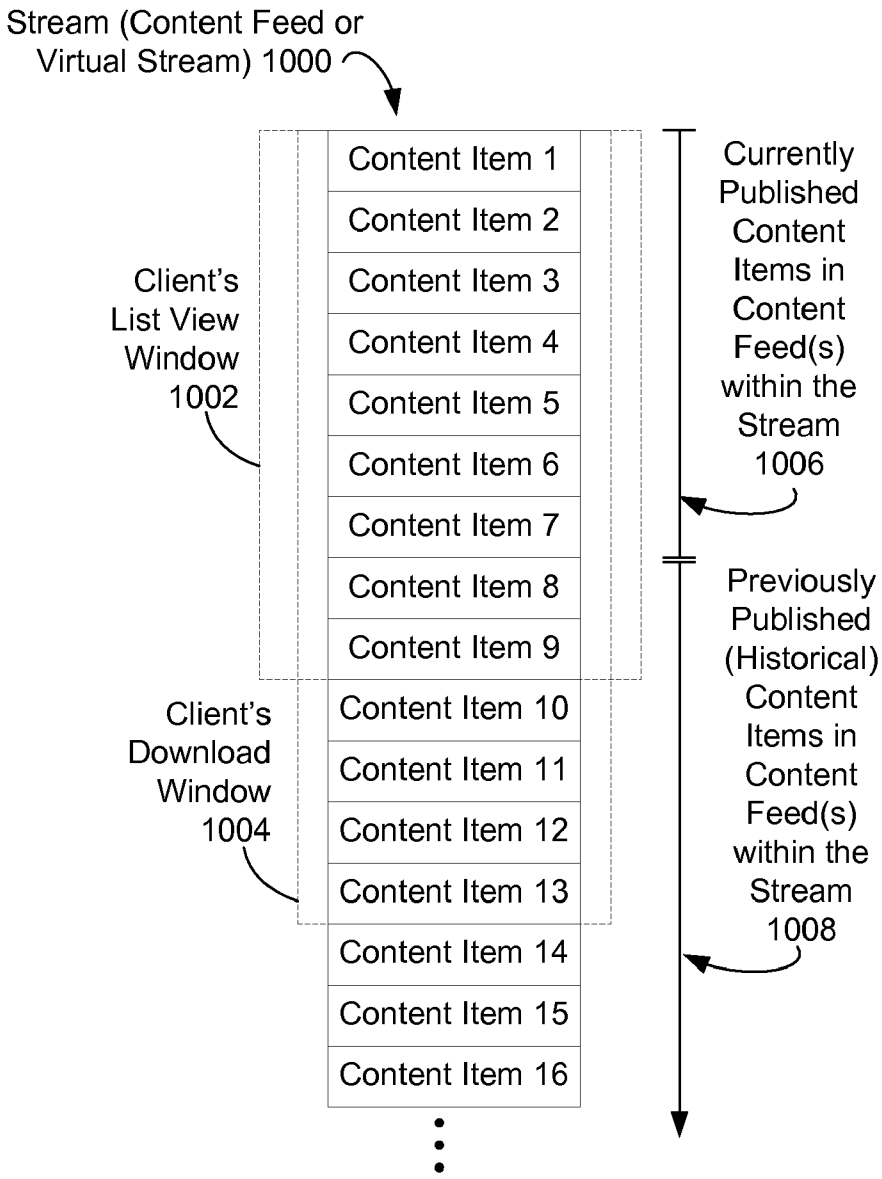


Figure 10A

Content Item(s) Marked as Read

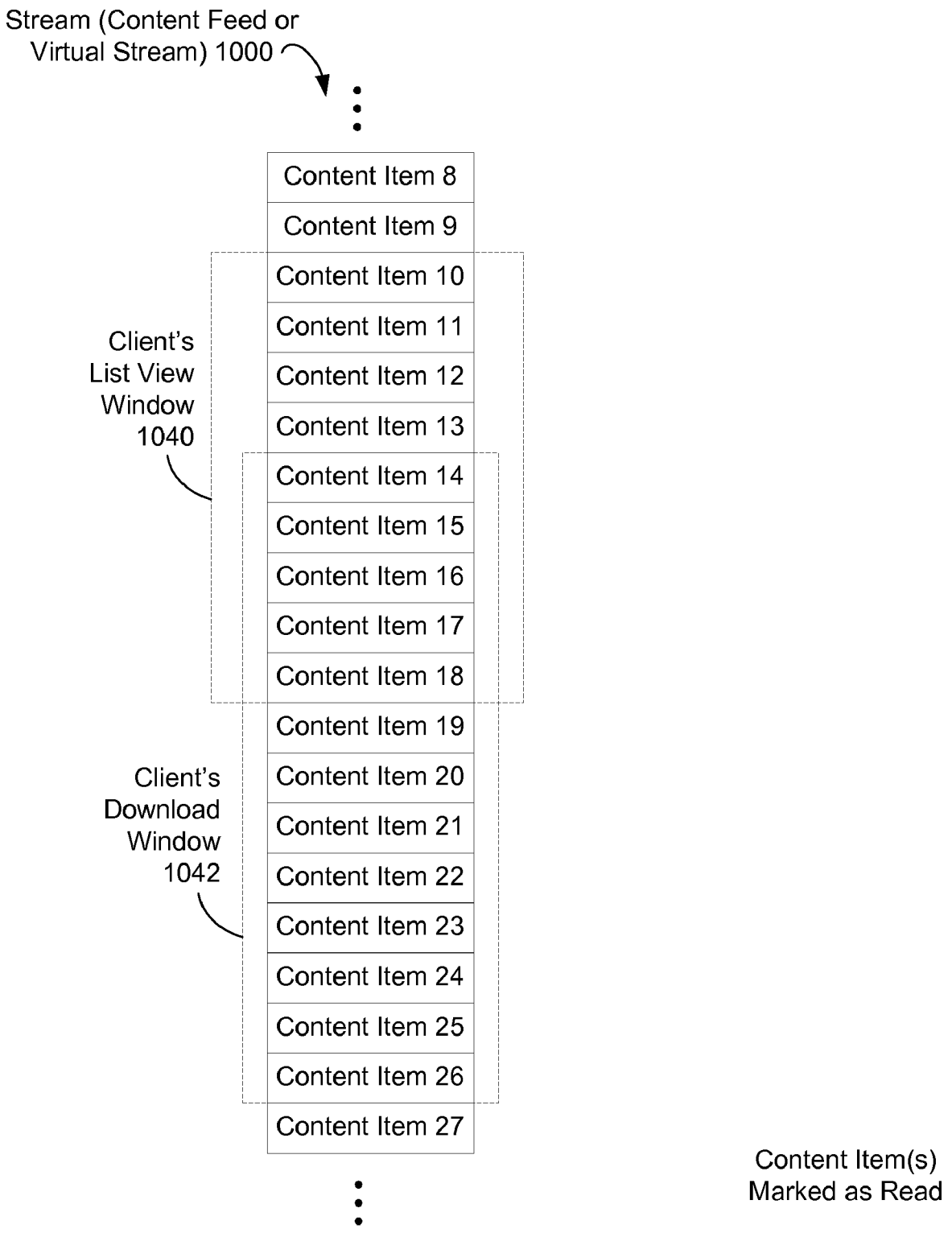

Stream (Content Feed or Virtual Stream) 1000
Content Item 8
Content Item 9
Content Item 10
Content Item 11
Content Item 12
Content Item 13
Content Item 14
Content Item 15
Content Item 16
Content Item 17
Content Item 18
Content Item 19
Content Item 20
Content Item 21
Content Item 22
Content Item 23
Content Item 24
Content Item 25
Content Item 26
Content Item 27
Client's List View Window 1040
Client's Download Window 1042
Content Item(s) Marked as Read

Figure 10B

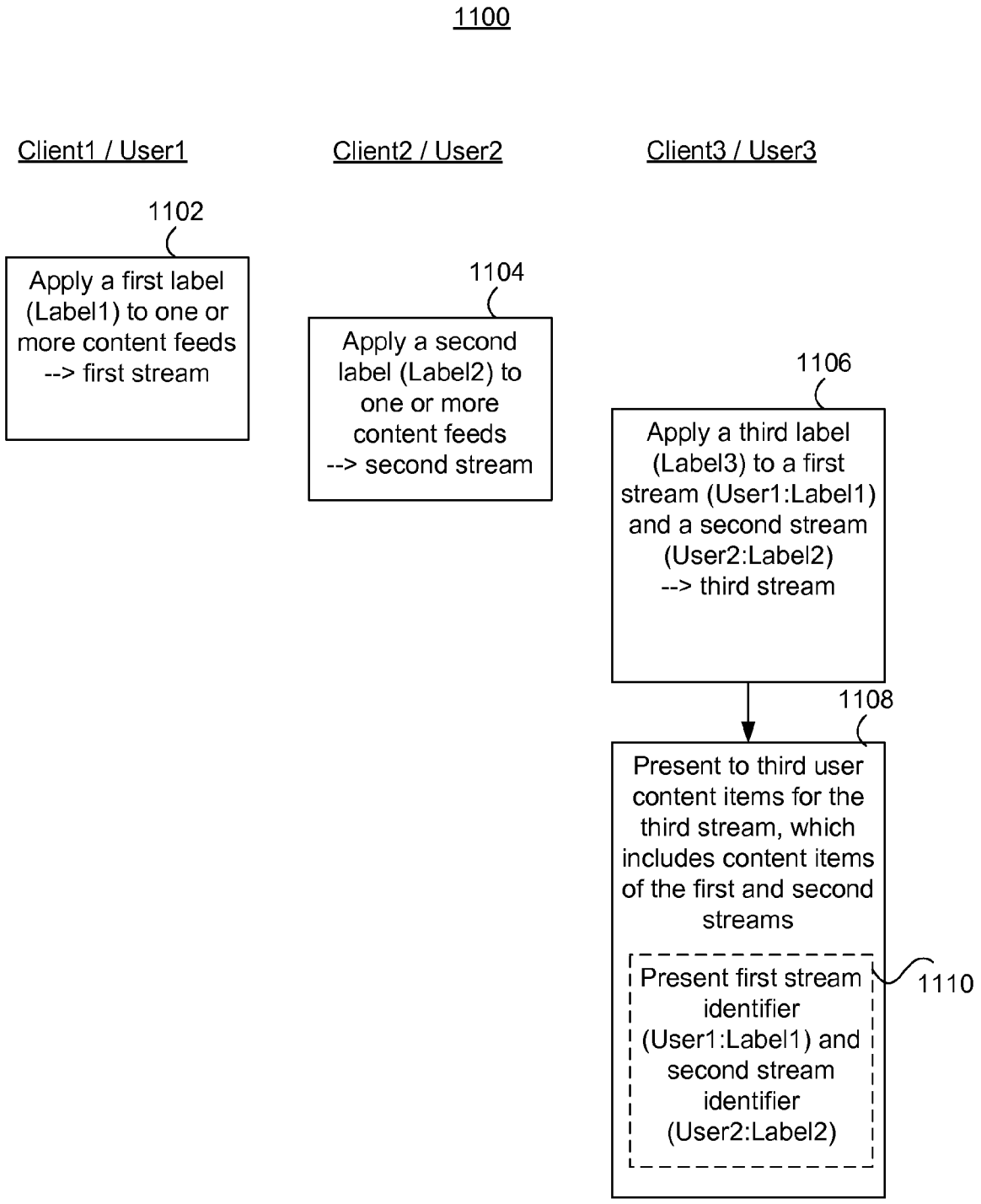

1100
Client1 / User1
Client2 / User2
Client3 / User3
1102
Apply a first label (Label1) to one or more content feeds --> first stream
1104
Apply a second label (Label2) to one or more content feeds --> second stream
1106
Apply a third label (Label3) to a first stream (User1:Label1) and a second stream (User2:Label2) --> third stream
1108
Present to third user content items for the third stream, which includes content items of the first and second streams
Present first stream identifier (User1:Label1) and second stream identifier (User2:Label2)
1110

Figure 11

SERVER SYSTEM FOR SERVING EXTENDED CONTENT FEEDS TO CLIENTS

RELATED APPLICATIONS

[0001] This application claims priority to U.S. Provisional Patent Application No. 60/828,610, titled "Server System for Serving Extended Content Feeds to Clients," filed Oct. 6, 2006, which is incorporated by referenced herein in its entirety.

[0002] This application is related to the following U.S. patent applications, which are incorporated by reference herein in their entirety:

[0003] U.S. patent application Ser. No. 11/246,597, titled "Indirect Subscriptions to A User's Selected Content Feed Items," filed Oct. 7, 2005;

[0004] U.S. patent application Ser. No. 11/246,596, titled "Personalized Content Feed Suggestions Page," filed Oct. 7, 2005;

[0005] U.S. patent application Ser. No. 11/246,657, titled "Content Feed User Interface with Gallery Display of Same-Type Items," filed Oct. 7, 2005; and

[0006] U.S. patent application Ser. No. 11/246,656, titled "Indirect Subscriptions to Top N Lists of Content Feeds," filed Oct. 7, 2005.

TECHNICAL FIELD

[0007] The disclosed embodiments relate generally to content syndication, and more particularly, to a content feed user interface with multiple modes for viewing content feed items.

BACKGROUND

[0008] Content syndication is growing in popularity as a way to distribute frequently updated information, such as news and blog postings, on the Internet. Using content syndication formats such as Really Simple Syndication (RSS), content providers can include content and/or links to such content in a content feed. Users may subscribe to these content feeds using an application known as a feed reader, content feed reader, feed aggregator or content feed aggregator. When the content feed is updated with new content items, the new content items are reflected in the user's reader.

[0009] A user may subscribe to many content feeds, with each feed listing multiple content items. Looking through a dense list of content items to locate items of interest can be a cumbersome task for the user.

[0010] Accordingly, there is a need for a more efficient manner of presenting content items to the user.

SUMMARY

[0011] A method of serving content is performed at a system remotely located from a client. For each of a plurality of content feeds, a currently published set of content items for a respective content feed is retrieved. The retrieved content items are merged into a previously stored set of previously published content items for the respective content feed. A database storing content items for the respective content feed includes both the currently published set of content items and historical content items no longer published by the respective content feed. In response to a request from the client, an ordered list of content items for a specified stream is transmitted to the client. The specified stream corresponds to one or more content feeds in the plurality of content feeds. The transmitted content items include at least one historical content item for at least one of the content feeds corresponding to the specified stream.

[0012] A system for serving content, located remotely from a client, includes memory, one or more processors, and one or more programs stored in the memory and configured for execution by the one or more processors. The one or more programs include instructions for retrieving, for each of a plurality of content feeds, a currently published set of content items for a respective content feed, and for merging the retrieved content items into a previously stored set of previously published content items for the respective content feed. A database storing content items for the respective content feed includes both the currently published set of content items and historical content items no longer published by the respective content feed. The one or more programs also include instructions for transmitting to the client, in response to a request from the client, an ordered list of content items for a specified stream. The specified stream corresponds to one or more content feeds in the plurality of content feeds. The transmitted content items include at least one historical content item for at least one of the content feeds corresponding to the specified stream.

[0013] A computer readable storage medium storing one or more programs is configured to be executed by a server computer system remotely located from a client. The one or more programs include instructions for retrieving, for each of a plurality of content feeds, a currently published set of content items for a respective content feed, and for merging the retrieved content items into a previously stored set of previously published content items for the respective content feed. A database storing content items for the respective content feed includes both the currently published set of content items and historical content items no longer published by the respective content feed. The one or more programs also include instructions for transmitting to the client, in response to a request from the client, an ordered list of content items for a specified stream. The specified stream corresponds to one or more content feeds in the plurality of content feeds. The transmitted content items include at least one historical content item for at least one of the content feeds corresponding to the specified stream.

[0014] A system for serving content includes means for retrieving, for each of a plurality of content feeds, a currently published set of content items for a respective content feed, and for merging the retrieved content items into a previously stored set of previously published content items for the respective content feed. A database storing content items for the respective content feed includes both the currently published set of content items and historical content items no longer published by the respective content feed. The system also includes means for transmitting to the client, in response to a request from the client, an ordered list of content items for a specified stream. The specified stream corresponds to one or more content feeds in the plurality of content feeds. The transmitted content items include at least one historical content item for at least one of the content feeds corresponding to the specified stream.

BRIEF DESCRIPTION OF THE DRAWINGS

[0015] FIG. **1** is a block diagram illustrating a content feed aggregation system in accordance with some embodiments.

[0016] FIGS. **2**A-**2**D are flow diagrams illustrating processes for presenting content items to a user, in accordance with some embodiments.

[0017] FIGS. **3**A-**3**D are schematic screenshots of the user interface of a content feed reader in accordance with some embodiments.

[0018] FIG. **4** is a block diagram illustrating a client computer in accordance with some embodiments.

[0019] FIG. **5** is a block diagram illustrating a server computer in accordance with some embodiments.

[0020] FIGS. **6**A, **6**B, **6**C and **6**D are diagrams illustrating data structures for feed content items and user subscriptions, in accordance with some embodiments.

[0021] FIGS. **7**A, **7**B and **7**C are flow diagrams illustrating a process for presenting content items to a user, in accordance with some embodiments.

[0022] FIG. **8**A is block diagram illustrating a feed reader user interface in accordance with some embodiments.

[0023] FIG. **8**B is a block diagram illustrating a content item provided via the user interface of FIG. **8**A, in accordance with some embodiments.

[0024] FIG. **8**C is a block diagram illustrating a user interface for installing a bookmark on a browser toolbar, where the bookmark is for accessing one or more items of one or more content feeds to which the user has subscribed, in accordance with some embodiments.

[0025] FIG. **9**A is a flow diagram illustrating a process for retrieving and storing content items, in accordance with some embodiments.

[0026] FIGS. **9**B and **9**C are flow diagrams illustrating processes for presenting content items to a user, in accordance with some embodiments.

[0027] FIGS. **10**A and **10**B are block diagrams illustrating a list of content items corresponding to a stream, in accordance with some embodiments.

[0028] FIG. **11** is a flow diagram illustrating a process for recursively defining streams of content items, in accordance with some embodiments.

[0029] Like reference numerals refer to corresponding parts throughout the drawings.

DESCRIPTION OF EMBODIMENTS

[0030] In a content feed aggregation system, a user may subscribe to one or more content feeds and be presented with the content items included in those content feeds. The content items may be presented to the user in an ordered list, such that the format of each item in the list is uniform. However, the user may desire to view additional information about selected content items beyond that provided by a uniform list. The disclosed embodiments provide the user a way to view additional information about selected items. For example, in one mode of operation, when a user selects a first content item, the selected item is displayed in an expanded format, and other items are displayed in a compact format. When the user subsequently selects a second content item, the second item is displayed in the expanded format and the first item is displayed in the compact format.

[0031] Content feeds are also herein called feed streams. The feed items from a content feed source can be considered to be a stream of items. Furthermore, some embodiments of the content feed aggregation and distribution system described herein allow users to define virtual streams that incorporate content items from content feeds and/or other virtual streams. Thus, in these embodiments, the term "feed streams" also encompasses virtual streams defined by the users of the content feed aggregation and distribution system.

[0032] FIG. **1** is a block diagram illustrating a content feed aggregation system in accordance with some embodiments. The content feed aggregation system **100** includes a server system **104**. The server system **104** is coupled to one or more client systems **102** and to one or more hosts **120** (or "feed sources") by a network **106**. The network **106** may be any suitable wired and/or wireless network and may include a local area network (LAN), wide area network (WAN), the Internet, metropolitan area network (MAN), or any combination of such networks.

[0033] The server system **104** accesses content feeds from the feed sources **120**. The server system **104** includes a server **108** and a database **110**. Server **108** serves as a front-end for the server system **104**. Server **108**, sometimes called a front end server, retrieves the content feeds from the feed sources **120**, and also provides an interface between the server system **104** and the client systems **102**. In some embodiments, the functions of server **108** may be divided or allocated among two or more servers.

[0034] The server system **104** stores content items in the content feeds in the database **110**. In some embodiments, the database **110** stores both metadata (e.g., title, description, URL, date/time, and possibly other metadata) and the content of each content item. However, in some other embodiments, the database **110** stores the metadata but not the content for each content item. The database **110** also stores user feed information **112** for a plurality of users. The user feed information for a particular user identifies content feed subscriptions, as well as sources, filters, and read states for that particular user. That is, user feed information associated with a user identifies the content feeds to which the user has subscribed, any filters the user has defined for the feeds, any labels the user has associated with individual feed items, and an indication of whether each feed item has been marked as "read" by the user.

[0035] It should be appreciated that the server system **104**, including the server **108** and the database **110**, may be implemented as a distributed system of multiple computers. However, for convenience of explanation, the server system **104** is described below as being implemented on a single computer, which can be considered a single logical system.

[0036] A content feed (or channel) is a resource or service that provides a list of content items that are present, recently added, or recently updated at a feed source **120**. A content item in a content feed may include the content associated with the item itself (the actual content that the content item specifies), a title (sometimes called a headline), and/or a description of the content, a network location or locator (e.g., URL) of the content, or any combination thereof. For example, if the content item identifies a text article, the content item may include the article itself inline, along with the title (or headline), and locator. Alternatively, a content item may include the title, description and locator, but not the article content. Thus, some content items may include the content associated with those items, while others contain links to the associated content but not the full content of the items. A content item may also include additional metadata that provides additional information about the content. The full version of the content may be any machine-readable

data, including but not limited to web pages, images, digital audio, digital video, Portable Document Format (PDF) documents, and so forth.

[0037] In some embodiments, a content feed is specified using a content syndication format, such as RSS. RSS is an acronym that stands for "rich site summary," "RDF site summary," or "Really Simple Syndication." "RSS" may refer to any of a family of formats based on the Extensible Markup Language (XML) for specifying a content feed and content items included in the feed. In some other embodiments, other content syndication formats, such as the Atom syndication format or the VCALENDAR calendar format, may be used to specify content feeds.

[0038] Further information regarding the RSS syndication format known as "Really Simple Syndication" is disclosed in the *RSS* 2.0 *Specification* by D. Winer, which is hereby incorporated by reference herein in its entirety. M. Nottingham et al. disclose further information regarding the Atom syndication format in *The Atom Syndication Format*, which is hereby incorporated by reference herein in its entirety.

[0039] A user interfaces with the server system **104** and views content items at a client system or device **102** (hereinafter called the client system for ease of reference). The client system **102** includes a computer **118** or computer controlled device, such as a personal digital assistant (PDA), cellular telephone or the like. The computer **118** typically includes one or more processors (not shown); memory, which may include volatile memory (not shown) and non-volatile memory such as a hard disk drive **119**; and a display **114**. The computer **118** may also have input devices such as a keyboard and a mouse (not shown).

[0040] In some embodiments, a user may interact with the server system **104** via a feed reader user interface **116** that may be presented on the display **114**. The user may create a list of feed subscriptions via the feed reader user interface **116**. In some embodiments, using the feed reader user interface **116**, the user may also create filters or labels to be applied to content feeds and/or content items, and modify the read states of content feeds and/or content items. The feed reader user interface transmits a list of content feed subscriptions, or modifications to a list of content feed subscriptions, to the server system **104** for storage at the database **110**. The feed reader user interface **116** presents content items stored at the database **110** to the user based on the user's list of content feed subscriptions. That is, feed reader user interface **116** presents to the user content items specified in the content feeds to which the user has subscribed. A user may view the full version of the content of a content item in the feed reader user interface **116** by selecting it (e.g., by clicking on it with a mouse pointer). In some embodiments, a copy of the user's list of content feed subscriptions and copies of the presented content items may be cached locally at the client system **102**.

[0041] In some embodiments, the feed reader user interface **116** may be a web-based user interface. That is, the feed reader user interface **116** includes one or more web pages. It is noted that a single web page can contain multiple frames, each of which may appear (when displayed by a browser application) to be a distinct web page. The web page(s) may be written in the Hypertext Markup Language (HTML), Extensible Markup Language (XML), or any other suitable language for preparing web pages, and may include one or more scripts for interfacing with the server system **104**. For example, the web page(s) may include a Javascript application that interfaces with the server system **104** via an application programming interface (API). The Javascript application receives content items from the server system **104**, manages the rendering of those content items at the client, and also performs the client-side aspects of other tasks, such as marking items as read, adding and removing content feed subscriptions, labeling content feed subscriptions and/or individually selected content items, selecting or revising user preferences, and the like. The user may subscribe to content feeds, view content items, and otherwise interact with the server system **104** by interacting with the web page(s) of the feed reader user interface **116**. In other words, the server system **104**, including the feed reader user interface **116**, provides a web-based content aggregation service. The server system **104** aggregates and stores content items in accordance with the user's content feed subscriptions. In some embodiments, the server system **104** can also apply filters or labels, or change the read states of content items in accordance with user actions or instructions. When the user accesses the feed reader user interface **116**, the content items are presented to the user via the feed reader user interface **116**.

[0042] In some other embodiments, the feed reader user interface **116** may be a part of a standalone application that is run on the client system **102**. The standalone application may interface with the server system **104** via an application programming interface (API).

[0043] FIG. **2**A is a flow diagram illustrating a process **200** for presenting content items to a user, in accordance with some embodiments. Process **200** occurs at a client device, on a client application that permits the user to interact with server system **104** via feed reader user interface **116**. User interface **116** displays an ordered list of content items (**202**). The content items are specified in content feeds to which the user has subscribed. A selected first item is displayed in expanded format, and other items are displayed in compact format (**204**). FIG. **3**B depicts an exemplary schematic screen shot of a user interface showing a first item, in an ordered list of content items, in expanded format and the other items in a compact format. In one embodiment, the compact format includes the publication source, headline, and timestamp associated with each item, with each item presented in a single line. In another embodiment, the compact format further includes a snippet (a small portion) of the content, also displayed on the single line. In an embodiment, the expanded format displays the same information as the compact format on a first line of the expanded format. In addition, the expanded format displays the title (sometimes called the headline) and source of the content on subsequent lines, in a larger font than the font used on the first line, and displays some or all of the content below the title and source. In other embodiments, the expanded format may include additional information associated with the content item. For example, a content item displayed in the expanded format may include a link to the full content associated with the content item. The link includes the URL of a web page or other document that contains the full content of the content item.

[0044] A user may select a next item (second item), for example by scrolling down the list, or may select another item (second item), for example by clicking on it. In response, the ordered list is redisplayed (**206**). The newly selected second item is displayed in expanded format, while other items, such as the first item, are displayed in compact

format (**208**). FIG. **3**C depicts an exemplary schematic screen shot of a user interface showing a second item in expanded format while the first item and other items are displayed in compact format.

[0045] FIG. **2**B is a flow diagram illustrating a process **220** for presenting content items to a user, in accordance with some embodiments. An ordered list of contents items is displayed with a selected first item in expanded format and other items in compact format, as described above (**202**, **204**). When a user selects a second item, the ordered list is redisplayed (**222**). If the user has selected a second item by scrolling down the list, the redisplayed list is scrolled appropriately. The newly selected item is displayed in expanded format, while other items, such as the first item, are displayed in compact format (**208**). If the user has scrolled down the list, one or more items previously displayed at the top of the list when the first item was selected are no longer displayed (**224**). In some embodiments, the status of a previously selected item is changed from unread to read when the item scrolls off the top of the displayed list (**226**). More specifically, when an item that has been viewed in expanded format scrolls off the top of the display list (**226**), the client sends a message to the server system identifying the content item to be marked as read, and the server system then updates the status of the content item accordingly.

[0046] As discussed below with reference to FIG. **2**D, in some other embodiments, content items are marked as read when they are displayed in expanded format. Alternatively, content items are marked as read when they are selected, regardless of whether they are displayed in expanded or compact format. Furthermore, in some embodiments, content items that have been marked as read, which have not yet been scrolled off the displayed list, are displayed in a manner that is distinct from the display of unread items. For example, one or more portions of the displayed text (e.g., the displayed title or headline) of items marked as read may be displayed using a light shade of gray while the corresponding text or text portions of unread items may be displayed using black text.

[0047] FIG. **2**C is a flow diagram illustrating a process **240** for presenting content items to a user, in accordance with some embodiments. Process **240** begins with operations **202**-**208**, described above with regards to FIG. **2**A. After operation **208**, the displayed ordered list includes a newly selected second item displayed in expanded format and other items, such as the previously selected first item, displayed in compact format. A user may select a next item (third item), for example by scrolling down the list, or may select another item (third item), for example by clicking on another item. In response, the ordered list is redisplayed (**242**). The newly selected third item is displayed in expanded format, while other items, such as the second and first items, are displayed in compact format (**244**). However, if the user scrolls through the list to select the third item, the first or second items may not be displayed (e.g., because they have been scrolled off the top or bottom of the displayed list of content items), depending on how far through the list the user scrolls.

[0048] FIG. **2**D is a flow diagram illustrating a process **260** for presenting content items to a user, in accordance with some embodiments. Operations **202** and **204** are performed as described above with regards to FIG. **2**A. Selection of the first item results in the status of the first item being changed from unread to read (**262**). In one embodiment, the status is changed by transmitting information to server **108** indicating that the first item has been read (**264**). Server **108** then updates database **110** to reflect that the first item has been read by the user.

[0049] A user may select a next item (second item), for example by scrolling down the list, or may select another item (second item), for example by clicking on another item. In response, the ordered list is redisplayed (**206**). The newly selected second item is displayed in expanded format, while other items, such as the first item, are displayed in compact format (**208**). Selection of the second item results in the status of the second item being changed from unread to read (**266**). In one embodiment, the status is changed by transmitting information to server **108** indicating that the second item has been read (**268**). Server **108** then updates database **110** to reflect that the second item has been read by the user.

[0050] FIG. **3**A is a schematic screenshot of a web browser's user interface **300** (sometimes hereinafter called the web browser for convenience), presenting a web page that displays a feed reader user interface **302**, in accordance with some embodiments. The web browser **300** can be any type of browser application that permits a user to display and interact with web pages or other documents or information. The web browser's user interface includes a URL field **304** for entering the URL of any web page that a user wants to display, as well as for displaying the URL of the web page currently being displayed. The user may access a web page displaying a feed reader's user interface **302** by entering in the URL field **304** the URL associated with feed reader user interface **302**.

[0051] Feed reader user interface **302** includes a list **305** of labels, also referred to as folders, associated with content feeds to which a user has subscribed. Exemplary labels shown in FIG. **3**A include "News" **306** and "Sports" **308**. The labels or folder names may be user defined names. User interface **302** also includes an ordered list **310** of content items associated with a selected label. For example, in FIG. **3**A, the "Sports" label **308** has been selected by the user, as indicated by a shaded background. In this example, ordered list **310** includes items **312**, **314**, and **315**. In some embodiments, items in ordered list **310** are ordered according to timestamps associated with the items. In some other embodiments, items in ordered list **310** are ordered in accordance with a score that takes into account both the content of the items and the timestamps of the items.

[0052] A user may select a different label, for example by clicking on a different label. If a user selected the "News" label **306**, ordered list **310** would display content items associated with "News" label **306**. Also, user interface **302** would indicate that the "News" label **306** was selected by displaying it with a shaded background. In this example, the background for "Sports" label **308** would no longer be shaded after the user selected "News" label **306**.

[0053] In FIG. **3**A, all items in ordered list **310**, including items **312**, **314**, and **315**, are displayed in compact format. In this embodiment, compact format includes the source, headline, and timestamp for each item, displayed in a single line. For example, the compact format display for item **312** includes "Sports Source 1," "Sports Headline 1," and "Timestamp 1," all displayed on the same line of ordered list **310**. In other embodiments, compact format might include additional information. For example, a brief portion of the content could be displayed next to the headline.

[0054] User interface **302** shows that an item in ordered list **310** is selected by displaying that item with a bold border. In FIG. **3**A, item **312** is selected, as indicated by its bold border. Items **314** and **315** are not selected, as indicated by their narrow borders. In other embodiments, a selected item may have a border with a different color from that of other items (e.g., a thick blue border for the selected item and a gray or thin black border for the unselected items). A user may select another item (second item) by clicking on "Next Item" button **316** or "Previous Item" button **318**, by clicking directly on another item, or by scrolling through ordered list **310** using scrollbar **320** and then clicking on another item. For example, in FIG. **3**A, a user could select item **314** by clicking "Next Item" button **316** or by clicking directly on item **314**. Item **314** then would be displayed with a bold border, and other items, such as item **312**, would appear with narrow borders.

[0055] FIG. **3**B is a schematic screenshot of a web browser **300** presenting a web page that displays a feed reader user interface **302**, in accordance with some embodiments. As in FIG. **3**A, user interface **302** includes a list **305** of labels associated with content feeds to which a user has subscribed and an ordered list **310** of content items associated with a selected label. However, a selected content item **340** is displayed in expanded format, while other content items, such as a second item **314**, are displayed in compact format. Item **340** corresponds to item **312** displayed in compact format in FIG. **3**A. In an embodiment, the expanded format displays the same information as the compact format on a first line, such as the source, headline, and timestamp for each item. The headline then is displayed again in larger type on a second line, the source is displayed again on a third line, and the content is displayed on a fourth and subsequent lines as necessary. This embodiment of expanded format is illustrated by content item **340**. In a different embodiment, the expanded format does not display the same information as the compact format on a first line; instead, it displays the headline and source in larger type than in the compact format, followed by the content on subsequent lines.

[0056] As discussed above in regards to FIG. **3**A, a user may select another item (second item). For example, in FIG. **3**B, the user might select item **314**, by clicking on item **314** or by clicking "Next Item" button **316**. FIG. **3**C, a schematic screenshot of a web browser **300** presenting a web page that displays a feed reader user interface **302**, shows the result of this user action according to an embodiment. The selected item **360** (corresponding to item **314** in compact format) appears in expanded format. Other items, including first item **312** (corresponding to item **340** in expanded format) are displayed in compact format.

[0057] In the example of FIGS. **3**B and **3**C, item **360** is not as large as item **340**: it has only one line of content as opposed to two. Therefore, the ordered list **310** in FIG. **3**B displays fewer items than the ordered list **310** in FIG. **3**C: ordered list **310** in FIG. **3**B displays seven items, while ordered list **310** in FIG. **3**C displays eight items. In general, the number of items displayed in ordered list **310** will vary depending on the size in expanded format of the selected item.

[0058] FIG. **3**D is a schematic screenshot of a web browser **300** presenting a web page that displays a feed reader user interface **302**, in accordance with some embodiments. As in FIGS. **3**A-**3**C, user interface **302** includes a list **305** of labels associated with content feeds to which a user has subscribed and an ordered list **310** of content items associated with a selected label. However, all items in ordered list **310** are displayed in expanded format. Items **340** and **360** (corresponding respectively to items **312** and **314** in compact format) are displayed in their entirety. Only the top portion of item **380** (corresponding to item **315** in compact format) is displayed, because there is not enough space to display the entire item in expanded format. Because the size of each item in expanded format varies, depending for example on the amount of content, the number of displayed items will vary in general.

[0059] Some, but not necessarily all, content items displayed in expanded format include a link to the full content associated with those content items. The link may be activated by a user clicking on the headline in the content item, or by clicking on a link icon (not shown in FIGS. **3**A-**3**D) displayed in the expanded format of a respective content item. Activation of the link typically causes the browser to open a new browser window or tab and to download and display the document (if any) located at the location (e.g., URL) specified by the link.

[0060] Although not shown in FIGS. **3**A-**3**D, the feed reader user interface will typically include icons, menu items or the like for reviewing and managing the user's content feed subscriptions, including removing and adding content feed subscriptions, and optionally labeling a respective content feed subscription with one or more user defined labels (or, alternately, assigning the respective content feed subscription to one or more user defined folders).

[0061] FIG. **4** is a block diagram illustrating a client computer in accordance with some embodiments. The client computer **400** typically includes one or more processing units (CPU's) **402**, one or more network or other communications interfaces **406**, memory **404**, and one or more communication buses **414** for interconnecting these components. The communication buses **414** may include circuitry (sometimes called a chipset) that interconnects and controls communications between system components. The client computer **400** may also include a user interface **408** comprising a display device **410** and a keyboard and/or mouse (or other pointing device) **412**. Memory **404** includes high-speed random access memory, such as DRAM, SRAM, DDR RAM or other random access solid state memory devices; and may include non-volatile memory, such as one or more magnetic disk storage devices, optical disk storage devices, flash memory devices, or other non-volatile solid state storage devices. Memory **404** may optionally include one or more storage devices remotely located from the CPU(s) **402**. In some embodiments, memory **404** stores the following programs, modules and data structures, or a subset thereof:

[0062] an operating system **416** that includes procedures for handling various basic system services and for performing hardware dependent tasks;

[0063] a network communication module **418** that is used for connecting the client system **400** to other computers via the one or more communication network interfaces **406** and one or more communication networks, such as the Internet, other wide area networks, local area networks, metropolitan area networks, and so on;

[0064] a feed reader module **420** for handling content items from content feeds; and

[0065] a client application **430**, such as a web browser.

[0066] The feed reader module **420** includes a content item format module **422** for determining the format in which to display content items, a local database **424** for storing content items sent by the server, and a server interface module **426** for interfacing with server computer **500**.

[0067] In some embodiments, the feed reader module **420** may be a script-based module, embedded in a web page served from the server system **104** (FIG. **1**). The web page may be rendered by a client application **430**, such as a web browser, at the client computer **400**. When the web page is rendered, the feed reader module **420** is executed, thereby providing a web-based interface to the server system **104**. The script-based feed reader module may be written in JavaScript, ECMAScript or any other suitable scripting language.

[0068] In some other embodiments, the feed reader module **420** may be a standalone application stored in memory **404** of the client computer **400**. The standalone application may include, but is not limited to a feed aggregator application. In further other embodiments, the feed reader module **420** may be an add-on or a plug-in to another application. For, example, the feed reader module **420** may be a plug-in or extension to a web browser application or an email application.

[0069] In some embodiments, received content items may be cached locally in memory **404**. Similarly, a user's list of content feed subscriptions may also be cached locally in memory **404**.

[0070] Each of the above identified elements in FIG. **4** may be stored in one or more of the previously mentioned memory devices. Each of the above identified modules corresponds to a set of instructions for performing a function described above. The above identified modules or programs (i.e., sets of instructions) need not be implemented as separate software programs, procedures or modules, and thus various subsets of these modules may be combined or otherwise re-arranged in various embodiments. In some embodiments, memory **404** may store a subset of the modules and data structures identified above. Furthermore, memory **404** may store additional modules and data structures not described above.

[0071] FIG. **5** is a block diagram illustrating a server computer **500** in accordance with some embodiments. The server computer **500** typically includes one or more processing units (CPU's) **502**, one or more network or other communications interfaces **506**, memory **504**, and one or more communication buses **510** for interconnecting these components. The communication buses **510** may include circuitry (sometimes called a chipset) that interconnects and controls communications between system components. The server system **500** optionally may include a user interface **508**, which may include a display device (not shown), and a keyboard and/or a mouse (not shown). Memory **504** includes high-speed random access memory, such as DRAM, SRAM, DDR RAM or other random access solid state memory devices; and may include non-volatile memory, such as one or more magnetic disk storage devices, optical disk storage devices, flash memory devices, or other non-volatile solid state storage devices. Memory **504** may optionally include one or more storage devices remotely located from the CPU(s) **502**. In some embodiments, memory **504** stores the following programs, modules and data structures, or a subset thereof:

[0072] an operating system **512** that includes procedures for handling various basic system services and for performing hardware dependent tasks;

[0073] a network communication module **514** that is used for connecting the server system **500** to other computers via the one or more communication network interfaces **506** and one or more communication networks, such as the Internet, other wide area networks, local area networks, metropolitan area networks, and so on;

[0074] a feed tracker module **516** for retrieving content items for storage periodically and/or as content feeds are updated;

[0075] a subscriptions database **518** for storing information about the content feed subscriptions of users of the system;

[0076] a feed content database **520**, for storing content items from content feeds and user information about content items that respective users have read, labeled, and so on;

[0077] a reader application interface **524** for exchanging information with the feed reader modules in one or more client computers; and

[0078] a user preferences database **526** for storing user-specific information, including user preferences with regard to the user interface.

[0079] Each of the above identified elements in FIG. **5** may be stored in one or more of the previously mentioned memory devices. Each of the above identified modules corresponds to a set of instructions for performing a function described above. The above identified modules or programs (i.e., sets of instructions) need not be implemented as separate software programs, procedures or modules, and thus various subsets of these modules may be combined or otherwise re-arranged in various embodiments. In some embodiments, memory **504** may store a subset of the modules and data structures identified above. Furthermore, memory **504** may store additional modules and data structures not described above.

[0080] Although FIG. **5** shows a "server computer," FIG. **5** is intended more as a functional description of the various features which may be present in a set of servers than as a structural schematic of the embodiments described herein. In practice, and as recognized by those of ordinary skill in the art, items shown separately could be combined and some items could be separated. For example, some items shown separately in FIG. **5** could be implemented on single servers and single items could be implemented by one or more servers.

[0081] FIG. **6**A is a diagram illustrating a data structure for feed content items, in accordance with some embodiments. As described above, a server computer **500** (FIG. **5**) may store a plurality of feed content items in feed content database **520**. The feed content items may be stored in a data structure, such as a table data structure **600**. The feed content table **600** includes a row **610** for each content item from each feed source. Each row includes one or more fields (**612**, **614**) that identify the content item, such as a content ID **612** and a feed ID **614** (which identifies the feed source of the content item). In some embodiments, the content ID may include information that uniquely identifies the feed source, in which case the feed ID **614** may be omitted. Each row **610** may further include content **616** and a content fingerprint **617** of the content item. The content **616** may include the

metadata of the content item (e.g., title, description, URL, date/time, and possibly other metadata), and may further include the actual content of the content item. In some embodiments, the content fingerprint **617** of a content item may be generated by applying a hash function to the content **616** of the item, and the web page or document, if any, referenced by the content item (e.g., referenced by a URL or publisher-assigned ID in the content item). In addition, each row **610** includes a column **618** or field for each registered user of the system. The user data in this column may include a read flag **620**, which indicates whether the user corresponding to this column has read the content item associated with the row **610** containing the user data. The user data in column **618** also may optionally include additional user information, such as one or more user specified labels or flags **622**. A user may tag or label content items, as well as content feeds, and a record of the tag or label that the user associated with each labeled content item is stored in the user data **618**. In some embodiments, the system may allow users to tag or label individual content items with predefined tags or labels (e.g., star, red, blue, etc.), and some embodiments further allow users to tag or label individual content items with user-defined labels (e.g., "news," "music," etc.). Furthermore, a content item, represented by a row **610**, may by tagged by any number of users.

[0082] FIG. **6**B is a diagram illustrating a data structure for user subscriptions, in accordance with some embodiments. As described above, a server computer **500** (FIG. **5**) may store user subscription information in subscriptions database **518**. The user subscription information may be stored in a data structure, such as a table data structure **650**. A user subscription information table **650** stores feed identifications for feed streams known to the system and user data for those feed streams. The user subscription table **650** includes fields or columns **656** corresponding to the stream ID's for each content stream known to the system, including both real content streams (typically corresponding to content streams from publication sources at well defined network locations) and virtual content streams. A virtual content stream is a content stream derived from one or more other content streams within the context of the server system **104**. For example, a virtual content stream may include a set of real content streams assigned a label by a user. In another example, a virtual stream may include individual content items assigned a particular label by a user; the labeled content items may be from one or more real content streams. In one embodiment, a virtual stream can be identified by the user who created the stream and the user label assigned to the virtual stream by the user. Thus, the stream ID of a virtual stream may include the user ID **658** of the user who created the virtual stream and the user label **660** assigned to the virtual stream by that user. Alternately, the stream ID of a virtual stream may be generated or assigned by a mapping function that maps virtual streams to virtual stream ID's.

[0083] Table **650** further includes a row **654** for each user, with fields or columns **656** containing user data for each content stream. In some embodiments, the user data for a respective content stream includes a subscribed flag **662** indicating whether the user has subscribed to the content stream associated with the column **656** containing the user data. The user data may also include a marked-as-read flag and timestamp **668**, which indicates whether and when the user corresponding to the row has viewed the items in the content stream associated with the column **656**. The user data may further include additional user information, such as one or more user specified labels **664**. A user may label content streams that the user may want to view again in the future, and a record of the label that the user associated with a content stream may be stored in the user data **656**.

[0084] FIG. **6**C is a diagram illustrating a data structure for user preferences, in accordance with some embodiments. As described above, a server computer **500** (FIG. **5**) may store user preference information in user preferences database **526**. The user preference information may be stored in a data structure, such as a table data structure **680**. User preferences table **680** contains user data **684** (e.g., in a row or record of the table **680**) for each user. The user data **684** optionally includes a user ID **686** (which may be implied by the location of the user data **684** in the user preferences table **680**), and a token **688** used by server **500** to authenticate requests from the user. In some embodiments, the token **688** is generated using a random or pseudo-random number generator or function, and is of sufficient length that it would be impractical to guess or independently replicate the token. In some embodiments, users who have not implemented the bookmark feature discussed below are not assigned a token **688**. Instead, a respective token **688** is generated and stored in the user data **684** of a user when the user first implements the bookmark feature. As described in more detail below, the token is used in the bookmark feature to prevent interlopers from forging a user's feed reader bookmark(s) (sometimes herein called a content feed bookmark), which would enable unauthorized access to a user's feed reader account, and which in turn would result in content items being marked as read even though they had never been seen by the user.

[0085] The user data optionally may include one or more additional fields **689**, **690**, **692**, **694**. For example, the user data optionally may include a sort order flag **689**, for specifying the order in which to display content items on a user interface, such as user interface **302** (FIG. **3**A). For example, content items may be displayed in chronological order or according to a ranking based on predefined criteria (e.g., the quantity and/or quality of the content in the content items, as well as the regency of the content items). The user data may further include an animation flag **690** to indicate whether or not to display animations on the user interface. The user data may include user interface feature flags or parameter values (**692**, **694**) to specify other user interface characteristics associated with the user. For example, user interface feature flags may be used to specify the formatting and color of a user interface. In an embodiment, users may set the sort order flag, the animation flag and user interface feature flags via menus accessible from the user interface (not shown).

[0086] Feed content table **600**, user subscription table **650**, and user preferences table **680**, or portions of these tables, may be stored using sparse table storage mechanisms, since large portions of the user data in these tables may be empty or set to default values.

[0087] FIG. **6**D is a diagram illustrating an example of a user subscription table, such as user subscription table **650** (FIG. **6**B), in accordance with some embodiments. As explained above, user subscription table **650** includes rows for each user known to the system and columns for each content stream known to the system, including both real and virtual content streams. In the example of user subscription table **625**, columns **630** and **632** are associated with real content streams, corresponding to streams from publication

sources at well-defined network locations. Specifically, stream **1** (**630**) is available at a network location indicated by URL**1** and stream **2** (**632**) is available at a network location indicated by URL**2**. In an embodiment, the URL's associated with real streams are used as feed ID's and are listed in row **640** of user subscription table **625**. The check mark under stream **1** (**630**) for user **3** (**646**) indicates that user **3** has subscribed to stream **1**.

[0088] Columns **634**, **636** and **638** are associated with virtual content streams, defined as streams derived from one or more other content streams. For example, user **1** (**642**) has subscribed to streams **1** (**630**) and **2** (**632**) and assigned the label "tech" to both streams, as indicated by the check mark and text "label:tech" at the intersection of row **642** with columns **630** and **632**. User **1** thereby created stream **3** (**634**), a virtual stream. In some embodiments, a respective virtual stream is given a feed ID that includes a user name or identifier and a label assigned by a respective user (e.g., the feed ID may have the form "user#:label"). Examples of such feed ID's are listed in row **640** of FIG. **6**D. In this example, stream **3** has the feed ID "User**1**:tech." Because the feed ID of a respective virtual stream includes the user name or identifier of the user who defined the virtual stream, multiple users may use the same label without conflict. For example, user **25** (row not shown) has assigned the label "tech" to one or more streams, resulting in stream **5** (**638**) with feed ID "User**25**:tech".

[0089] Some virtual streams may correspond to (or may include) individual content items on one or more content feeds that have been labeled by a respective user. For example, a user (e.g., "user**99**" may mark or label specific content feed items as "favorite" items, or the user may assign a user-specified label (e.g., "best" or "reviews") to specific feed items. If the set of user labeled items is made available for subscription by others, the user-labeled items become a virtual stream that is assigned a stream identifier, such as "user**99**:best". This virtual stream is assigned a column in the user subscription table **625** and subscription and user labeling information for the virtual stream is stored in this column.

[0090] Check marks in column **634** associated with stream **3** indicate which other users have subscribed to stream **3**. In this example, user **2** (**644**) has subscribed to stream **3** and assigned the label "friends" to it. User **2** also has assigned the label "friends" to stream **5** (**638**), created by user **25**. User **2** thereby created stream **4** (**636**), which has feed ID "User**2**:friends". Stream **4** (**636**) thus is a virtual stream comprising other virtual streams (stream **3** and stream **5**), demonstrating that multiple levels of recursion are allowed for virtual streams. User **3** (**646**) has subscribed to stream **4** (**636**). If user **3** desired, he could label stream **4**, thereby creating another virtual stream with an additional level of recursion, and another user could subscribe to that virtual stream.

[0091] FIG. **7**A is a flow diagram illustrating a process **700** for presenting content items to a user, in accordance with some embodiments. Operations **702-712** are performed by client computer **400**, while operations **714-722** are performed by server computer **500**. Client **400** detects user selection of a bookmark (sometimes herein called a feed reader bookmark or content feed bookmark) (**702**). In response, client **400** downloads a document associated with the bookmark (**704**) and displays the downloaded document to the user (**712**).

[0092] In one embodiment, the process of downloading the document begins with transmitting to server **500** a request specified by the bookmark (**706**). As discussed in more detail below, in some embodiments, if the request is a valid request, it includes a token that uniquely corresponds to the user identifier of the requesting user, and a login cookie (also variously called an object, cookie file or cookie object) containing the requesting user's user identifier (sometimes called the user ID). Server **500** receives the request (**714**), identifies the user or user account of the user, and identifies an unread content item for the requesting user (**716**). Typically, in order for the server to successfully process the user's request, the requesting user must be a registered user of the content feed aggregation system, and must have subscribed to at least one content feed. The item identified by the server may, for example, be the most recent unread item in a set of content feeds associated with the selected bookmark. In another example, an unread item is identified according to other predefined selection criteria.

[0093] A document associated with the identified item is identified (**718**). Exemplary identified documents include a document corresponding to a URL in the identified content item, or the home page of a content feed containing the identified content item. For example, if the identified item is a blog posting, the identified document could be a web page containing the posting, or it may be the home page of the blog. If the identified item is a news article, the identified document could be a web page containing the article, or it may be the home page of a news service that published the article.

[0094] Server **500** marks the identified content item or a corresponding feed stream as read (**720**). In one embodiment, when the document identified in operation **718** is linked to a specific content item, the specific content item is marked as read. On the other hand, when the document identified in operation **718** is a home page for a content feed, the corresponding feed stream is marked as read, with a timestamp indicating the last time that the feed stream was marked as read. Marking an entire feed stream as read is equivalent (from the user's perspective) to marking all content items in the feed stream as having been read by the user, but is much more efficient because the only update made to the server's database is to update the "marked-as-read" flag and timestamp **668** in the user data for the feed stream (in the user subscription table **650**). In contrast, marking as read all the individual items in the feed stream would potentially require updating the user data for many items in the feed content table **600**. Once a feed stream has been marked as read for a particular user, only content items that are newer than the marked-as-read timestamp can be considered to be unread by the user (excluding, of course, any individual content items that have been marked as read).

[0095] Server **500** then sends client **400** a redirect to the identified document, or more specifically, to a network location associated with the document (**722**). Client **400** receives the redirect (**708**) and downloads the identified document from the redirect location (**710**). The downloaded document is displayed to the user (**712**).

[0096] FIG. **7**B is a flow diagram illustrating a process **740**, performed by server computer **500**, for identifying content to provide to a user, in response to user selection of a bookmark. For example, user selection of the bookmark may cause an HTTP request to be sent to the server and the URL may include one or more parameters. Server **500**

receives a request specified by the bookmark (**714**). A set of content items associated with the bookmark request is identified (**742**). For example, the identified set of items could include items associated with a content feed to which the user has subscribed.

[0097] During operation **744**, the number of unread items in the identified set of items is tested against predefined criteria. If the number of unread items meets the predefined criteria, a document corresponding to a URL in a content item in the set is identified (**746**). The item corresponding to the identified document is marked as read (**748**), and a redirect to the identified document is sent to client computer **400** (**722**). However, if the number of unread items does not meet the predefined criteria, a document corresponding to a home page of unread items in the set is identified (**750**). In addition, the content feed corresponding to the unread items is marked as read (**752**), and a redirect to the identified document is sent to the client computer (**722**). Alternately, all unread items in the set associated with the home page are marked as read (**752**), and a redirect to the identified document is sent to the client computer (**722**).

[0098] For example, consider a situation in which the most recent unread item from all the content feeds to which a user subscribes is a news article from a particular news provider. In response to user selection of a bookmark, server **500** receives a request specified by the bookmark (**714**) and identifies a set of items comprising the most recent unread item (the news article) and all other items provided by the same particular news provider (**742**). Now consider that the predefined criteria is whether the number or unread items in the set is less than ten. If the user's feed has less than ten unread items from the particular news provider, a web page containing the most recent unread item (the news article) will be identified (**746**), the item will be marked as read (**748**), and a redirect to the web page will be sent to client **400** (**722**). However, if the user's feed has ten or more unread items, a home page for the particular news provider will be identified (**750**), the feed stream associated with the particular news provider will be marked as read (**752**) (or, alternately, all the unread items in the feed stream will be marked as read), and a redirect to the home page will be sent to client **400** (**722**).

[0099] FIG. **7**C is a flow diagram illustrating a process **760**, performed by server computer **500**, for receiving and verifying a request resulting from user selection of a bookmark. For security reasons, it is desirable to verify the request is, in fact, an authorized request from a respective user of the feed reader system before processing the request. In some embodiments, a valid request (**702**) initiated by user selection of the feed reader bookmark includes a token that uniquely corresponds to the user identifier of the requesting user, and a login cookie (also variously called an object, cookie file or cookie object) containing the requesting user's user ID. In one embodiment, the token is embedded as a parameter in the URL specified by the bookmark. The login cookie, if included in the request, is previously generated when the user logged in to the feed reader system or logged into a related system, such as a parent server system that provides a plurality of online services. It is noted that in the embodiments described here, the token and the user identifier are stored in separate objects (e.g., a bookmark and a cookie) so as to protect the integrity of the user's feed reader account. As will be described next, unless the user is in possession of both a login cookie and a bookmark having a token that corresponds to the user identifier in the cookie, the request generated by activating the bookmark will be rejected by the feed reader system.

[0100] Upon receiving a request (**714**), server **500** checks to see whether a login cookie, creating by previously logging in to the feed reader system, containing a user ID was received with the request (**762**). If no login cookie was received, server **500** sends client computer **400** a redirect to a login web page (**764**), from which the user may log in to the feed reader. Once the user has logged in to the feed reader, the user may select a bookmark and thereby initiate process **700** (FIG. **7**A). If a login cookie was received, server **500** compares the token in the request to a stored token associated with the user (as identified by the user identifier in the login cookie), for example, token **688** stored in user preferences table **680**. If the received token is consistent with the stored token, server **500** proceeds to identify a set of content items for the requesting user (**742**) and then perform the subsequent steps of process **740** (FIG. **7**B), described above. However, if the received token is not consistent with the stored token, server **500** sends client **400** a redirect to, for example, a web page displaying an error message or a feed reader home page (**768**).

[0101] In some embodiments, a content feed bookmark functions as a link to a URL that corresponds to an online content feed aggregator. An example of a URL specified by a bookmark is:

[0102] http://www.domain1.com/reader/bookmark?mode=nextitem&token=5782174&label=news

In this example, the value "5782174" in the URL is the token that uniquely corresponds to the user identifier of a user. The value shown for this token is merely illustrative; in practice, the number of digits of the token will be chosen based on cryptographic concerns. The parameters "mode=nextitem" and "label=news" specify that the document to be identified in step **718** of process **700** is the next unread item in News label **306**. Different mode settings may be specified, for example, to specify that the item to be identified in step **718** is the home page of a content feed, or to invoke process **740** (FIG. **7**B). If no label is specified, the document to be identified in response to selection of the bookmark may be provided by any content stream to which the user has subscribed.

[0103] FIG. **8**A is a schematic screenshot of a web browser **300** displaying a web page **804**. Web browser **300** has a tool bar **802** containing a bookmark **800**. Selecting bookmark **800**, for example by clicking on it, initiates process **700**, described with regards to FIG. **7**A. Thus, instead of using a feed reader user interface such as the interface shown in FIG. **3**A, a user may access feed reader content items simply by selecting bookmark **800**.

[0104] FIG. **8**B is a schematic screenshot of a web browser **300** displaying a web page **850**, in response to user selection of bookmark **800**. In an embodiment, web page **850** corresponds to the most recent unread content item in the content feeds to which a user subscribes. Web page **850** is identified by a URL **852** corresponding to the source of the content item. In the example of FIG. **8**B, web page **850** corresponds to content item **312** in FIG. **3**A and content item **340** in FIG. **3**B. In this example, web page **850** includes a source **854**, a headline **856**, and text **858**, corresponding to the source, headline, and text of items **312** and **340**. In another embodiment, web page **850** corresponds to an unread item identified according to other predefined selection criteria.

[0105] FIG. **8**C is a block diagram illustrating a user interface for installing a bookmark on a browser toolbar, where the bookmark is for accessing one or more items of one or more content feeds to which the user has subscribed, in accordance with some embodiments. Web browser **300** displays a web page **870**, identified by a URL **872**, that allows a user to create a bookmark on toolbar **802** of browser **300**. Dragging bookmark link **874** onto toolbar **802** will create a bookmark **800** (see FIG. **8**A). In some embodiments, selecting the bookmark created using bookmark link **874** will cause the browser to display the most recent unread content item from all of the content feeds to which a user subscribes. However, the user may prefer to create a bookmark for a specific label or folder in the user's feed reader account, by selecting a label from pull-down menu **876** and then dragging bookmark link **878** onto toolbar **802**. In an embodiment, the resulting bookmark **800**, when selected, will display the most recent unread content item in a set of content feeds associated with the selected label or folder. In other embodiments, other content items will be displayed in accordance with predefined selection criteria.

[0106] FIG. **9**A is a flow diagram illustrating a process **900** for retrieving and storing content items for subsequent transmission to client computer **400**, in accordance with some embodiments. Server computer **500** retrieves a currently published set of content items for a content feed, for each of a plurality of respective content feeds (**902**). Step **902** is performed, for example, by feed tracker module **516** (FIG. **5**). The retrieved content items are merged into a previously stored set of previously published content items, otherwise known as historical content items, for each of the plurality of content feeds (**904**). Currently published content items are available from respective publication sources at well-defined network locations, while historical content items were previously available from the respective publication sources, but are no longer available. The merged content items are stored in a database (**906**), such as feed content database **520** (FIG. **5**).

[0107] In an embodiment, an XML file containing a currently published set of content items for a content feed is parsed by item. Each content item includes a title, date, and text context. Some items also include a link, such as a URL associated with the item, or an ID assigned by the feed publisher (herein called the "publisher-assigned ID"). For each item in the XML file, server computer **500** computes an identifier (sometimes called the content ID) and a content fingerprint. For example, the identifier may be a 64-bit number based on the URL of the corresponding content feed and either the publisher-assigned ID, the item's link, or the item's title, in that order of preference, depending on which information is available. In an embodiment, a hash function is applied to this information to produce the identifier. In one embodiment, the identifier is a 64-bit value, but may be a longer value in other implementations to further reduce the chance of a collision between two distinct content items. In some embodiments, the content fingerprint or a respective content item is generated by applying a hash function to the entire content of the content item, and the web page or document, if any, referenced by the content item (e.g., referenced by a URL or publisher-assigned ID in the content item).

[0108] Server computer **500** compares each identifier against the content ID's **612** of feed content table **600** (FIG. **6**A) stored in feed content database **520** (FIG. **5**). If an identifier does not match any of the stored content ID's, the corresponding content item is inserted into database **520**, along with its fingerprint. If the identifier does match a stored content ID, then server **500** compares the content fingerprint of the corresponding item to the content fingerprint of the stored item (see, for example, content fingerprint **617** in FIG. **6**A). If the content has not changed, the corresponding item is not inserted into database **520**. However, if the content has changed, the corresponding item is inserted into database **520** along with its content ID and fingerprint. In some embodiments, this is accomplished by replacing the previously stored content item that matches the content ID of the new content item. In another embodiment, the new content items is added to the database **520** without replacing or deleting the previously stored content item that matches the fingerprint of the new content item.

[0109] In response to a request from client computer **400**, server computer **500** transmits to client computer **400** an ordered list of content items for a specified stream (**908**). The specified stream may correspond to a particular content feed (i.e., may correspond to a particular publication source at a well-defined network location) or may be a virtual stream derived from one or more other content streams within the context of server system **104**. Therefore, the transmitted ordered list of content items contains items from one or more content feeds corresponding to the specified stream (**910**). The transmitted ordered list of content items also may include at least one historical item for at least one of the content feeds corresponding to the specified stream (**912**). In an embodiment, the list is ordered by time, based on timestamps associated with the content items. Alternately, the list may be ordered based on other predefined selection criteria (e.g., a predefined scoring function of the content and the timestamps of the content items).

[0110] FIG. **9**B is a flow diagram illustrating a process **920** for presenting an ordered list of content items to a user, in accordance with some embodiments. Server computer **500** retrieves and stores content items as described above with regard to FIG. **9**A (**902**, **904**). In response to a user action, client computer **400** transmits to server computer **500** a request for a set of content items corresponding to a specified stream (**922**). Examples of user actions prompting this request may include selecting a label, for example by clicking on a label listed under "Your Subscriptions" **305** in FIG. **3**A, or scrolling through a list of content items such as list **310** in FIG. **3**A. Server computer **500** receives the request and retrieves the requested set of content items from server memory **504** (**924**) and responds by transmitting a set (e.g., an ordered list) of content items for a specified stream (**908**). Client computer **400** receives the transmitted set of content items and displays at least a subset of the content items to the user (**926**). Examples of displayed sets of content items include lists **310** in FIGS. **3**A-**3**D.

[0111] FIG. **9**C is a flow diagram illustrating a process **940** for presenting an ordered list of content items to a user, in accordance with some embodiments. A request from client computer **400** (**922**) prompts server computer **500** to retrieve and transmit a set (e.g., an ordered list) of content items corresponding to a specified stream (**924**, **908**), as described above with regards to FIG. **9**B. In some embodiments, transmission **908** includes one or more continuation tokens for enabling subsequent requests for additional content items corresponding to the specified stream (**942**). Continuation tokens contain sufficient information to allow server com-

puter **500** to resume its ordered retrieval in response to a subsequent request from client computer **400**. For example, where the requested set of content items is ordered by time and is associated with a particular content feed, transmission **908** includes a continuation token containing the timestamp of the next item in the particular content feed to be retrieved (or the timestamp of the last item sent to the client) and transmitted in response to a subsequent request. In other words, if a request **922** is for the twenty most recent unread items in a particular feed, the continuation token returned along with the results of the request contains the timestamp of the twenty-first most recent unread item in the feed. In another example, where the requested set of content items is ordered by time and is associated with a plurality of respective content feeds, transmission **908** may contain multiple continuation tokens, each corresponding to a respective content feed. In yet another example, where the requested set of content items is arranged according to some other predefined criteria, the continuation token(s) may contain a parameter specifying the predefined criteria (e.g., one of a small number of predefined options) and another parameter specifying the depth of a next item to be retrieved (or the depth of the last item previously retrieved) after applying the criteria to content items in the specified stream.

[0112] A user viewing the displayed list of content items (**926**) may attempt to scroll beyond the end of the displayed list. This action by the user will prompt client computer **400** to fetch from server computer **500** an additional set of content items, corresponding to the specified stream, to display to the user. In some embodiments, when the user scrolls within a predefined number (e.g., a number between five and fifteen, inclusive) of content items of the end of the displayed list, client computer **400** fetches additional items from the server. Specifically, client computer **400** transmits to server computer **500** a request for addition content items (**944**). The request includes one or more continuation token (s) previously transmitted to client computer **400** in step **942** (**946**). Server computer **500** uses the returned continuation token(s) to retrieve the requested additional content items from server memory **504** (**948**). For example, where the second request is for the next N (e.g., twenty) unread items for the specified stream, the request sent by the client computer **400** includes a continuation token containing the timestamp of the first content item in the specified stream that follows the last content item (of the same specified stream) previously sent to the client computer. Server computer **500** will retrieve twenty items from the specified stream, starting with the one identified by the continuation token in the request. In another example, where the request is for the next twenty unread items for a particular feed, ranked according to predefined criteria, client computer **400** returns a continuation token containing a parameter specifying the predefined criteria and one or more parameters that specify either the number of items previously retrieved and transmitted in response to previous requests, or the depth of the next item to be retrieved and transmitted to the client computer. Server computer **500** ranks the content items corresponding to the specified feed according to the criteria stated in the continuation token and retrieves from memory **504** the requested number of content items starting at the specified depth in the ranking.

[0113] Content items retrieved in accordance with the returned continuation token(s) are transmitted to client computer **400** (**950**), and at least a subset of those content items are displayed to the user (**954**). For example, the client computer may simultaneously display some of the previously received content items and some of the newly received content items. Transmission **950** includes new continuation token(s) that may be returned in subsequent requests for content items (**952**), according to the process described above.

[0114] FIG. **10**A is a block diagram illustrating a list of content items corresponding to a stream **1000**, in accordance with some embodiments. Stream **1000** may correspond to a particular content feed or may be a virtual stream. Content items are stored in memory **504** of server computer **500**, for example in content item database **524** (FIG. **5**). The content items are ordered by time, such that content item **1** is the most recent item in the stream. The content items include currently published content items **1006** and previously published content items, or historical content items, **1008**. Currently published content items **1006** are available from respective publication sources at well-defined network locations, while historical content items **1008** are no longer available from the respective publication sources. In the example of FIG. **10**A, content items **1-7** are currently published, while content item **8** and all subsequent content items are historical.

[0115] In response to a request **922** (FIG. **9**B) from client computer **400**, an ordered list (or, more generally, a set) of items corresponding to stream **1000** is transmitted to client computer **400**. The transmitted ordered list contains a specified number of items, referred to as client's download window **1004**. In the example of FIG. **10**A, client's download window **1004** includes **13** items, meaning that the ordered list transmitted to client computer **400** includes **13** items. In this example, the **13** transmitted items include **7** currently published items and 6 historical items. At least a subset of the transmitted ordered list is displayed to the user (**926**; see FIG. **9**B), as determined by the size of client's list view window **1002**. In the example of FIG. **10**A, client's list view window includes 9 items, meaning that 9 items are displayed to the user via the feed reader user interface. Examples of formats for displaying items are shown in FIGS. **3**A-**3**D. In general, the size of client's list view window **1002** will vary depending on the space required to display transmitted content items.

[0116] FIG. **10**B is a block diagram illustrating content items in stream **1000** to be transmitted and displayed in response to a second request from client computer **400** following the initial request described with regards to FIG. **10**A. Client computer **400** generates the second request in response to a user action, such as scrolling through a list of content items, such as list **310** in FIG. **3**A. In response to the second request, server computer **500** identifies a next set of untransmitted content items in stream **1000**, where the number of identified content items is determined by the client's download window **1042**. In the example of FIG. **10**B, client's download window **1042** includes **13** items. Since content items **1-13** previously were transmitted in response to the first request, server computer **500** identifies content items **14-26** and transmits an ordered list of those items to client computer **400**.

[0117] The items actually displayed to a user in response to the second request depend on the action taken by the user. In the example of FIG. **10**B, the user has scrolled past the entire list of content items originally displayed in client's list view window **1002** (FIG. **10**A). Therefore, in this example,

client's list view window **1040** now displays content items **10-18**. In general, which content items are displayed will depend on the manner in which the user scrolls through the displayed list and the size of the displayed items. The displayed list may include content items retrieved via multiple requests. For example, in FIG. **10**B, displayed content items **10-13** were retrieved in response to the first request, as part of download window **1004** (FIG. **10**A), while displayed content items **14-18** were retrieved in response to the second request, as part of download window **1042**. However, all nine content items are displayed to the user in a seamless list.

[0118] FIG. **11** is a flow diagram illustrating a distributed process **1100** for recursively defining streams of content items, in accordance with some embodiments. FIG. **11** shows operations at three client devices (Client**1**, Client**2**, Client**3**), performed in response to instructions or commands from three respective users (User**1**, User**2**, User**3**). The corresponding operations at the server system are discussed above with reference to FIG. **6**D. At a first client device, Client**1**, in response to instructions or commands from a first user, User**1**, the first client device applies a first label (e.g., Label**1**) to one or more user-selected or user-specified content feeds (**1102**). If the labeled set of user-specified content feeds is made available for subscription by other users, the labeled set of user-specified content feeds becomes a virtual stream that is assigned a stream identifier, such as "User**1**: Label**1**". This virtual stream is assigned a column in the user subscription table **625** (FIG. **6**D) and subscription and user labeling information for the virtual stream is stored in this column.

[0119] Similarly, at a second client device, Client**2**, in response to instructions or commands from a second user, User**2**, the second client device applies a second label (e.g., Label**2**) to one or more user-selected or user-specified content feeds (**1104**). If the labeled set of user-specified content feeds is made available for subscription by other users, the labeled set of user-specified content feeds becomes a virtual stream that is assigned a stream identifier, such as "User**2**: Label**2**". This virtual stream is assigned a column in the user subscription table **625** (FIG. **6**D) and subscription and user labeling information for the virtual stream is stored in this column.

[0120] At a third client device, Client**3**, in response to instructions or commands from a third user, User**3**, the third client device applies a third label (e.g., Label**3**) to the first and second streams (**106**). The resulting set of labeled streams becomes a third stream if the set of labeled streams is made available for subscription by others. When the user selects the third label, which may be represented as a folder name, content items for the third stream are presented to the third user (**1108**). In addition, in some embodiments the names of the labeled streams (e.g., User**1**:Label**1** and User**2**: Label**2**) are presented to the third user (**1110**). For example, the names of the labeled streams may be shown in a subscription list at one side of the GUI or window in which content items of the third stream are displayed.

[0121] Additional users (e.g., a fourth user at a fourth client device) may subscribe to the third stream, and thus the process of recursive stream definitions may continue at the client devices of those additional users.

[0122] The foregoing description, for purpose of explanation, has been described with reference to specific embodiments. However, the illustrative discussions above are not intended to be exhaustive or to limit the invention to the precise forms disclosed. Many modifications and variations are possible in view of the above teachings. The embodiments were chosen and described in order to best explain the principles of the invention and its practical applications, to thereby enable others skilled in the art to best utilize the invention and various embodiments with various modifications as are suited to the particular use contemplated.

What is claimed is:

1. A method of serving content, comprising:

at a system remotely located from a client:

for each of a plurality of content feeds, retrieving a currently published set of content items for a respective content feed, and merging the retrieved content items into a previously stored set of previously published content items for the respective content feed; wherein a database storing content items for the respective content feed includes both the currently published set of content items and historical content items no longer published by the respective content feed; and

in response to a request from the client, transmitting to the client an ordered list of content items for a specified stream, the specified stream corresponding to one or more content feeds in the plurality of content feeds, the transmitted content items including at least one historical content item for at least one of the content feeds corresponding to the specified stream.

2. The method of claim **1**, wherein the ordered list of content items transmitted to the client includes at least one content item in the currently published set of content items for the specified stream and the at least one historical content item for the specified stream.

3. The method of claim **1**, wherein the specified stream comprises two or more content feeds in the plurality of content feeds, and the transmitted content items include at least one historical content item for at least one of the content feeds corresponding to the specified stream.

4. The method of claim **1**, wherein the specified stream includes a plurality of contents items tagged by a user, wherein the plurality of tagged content items are from one or more of the content feeds.

5. The method of claim **1**, wherein the specified stream comprises two or more content feeds in the plurality of content feeds and includes a plurality of contents items tagged by a user, wherein the plurality of tagged content items are from one or more of the content feeds.

6. The method of claim **1**, wherein the specified stream comprises one or more other streams labeled by a user, wherein a respective stream of the one or more other streams comprises two or more content feeds in the plurality of content feeds.

7. The method of claim **1**, wherein the specified stream comprises one or more other streams labeled by a user, wherein a respective stream of the one or more other streams comprises two or more content feeds in the plurality of content feeds; and

wherein the specified stream includes a plurality of content items tagged by a user, wherein the plurality of tagged content items are from one or more of the content feeds.

8. The method of claim **1**, further comprising:

at the client:

receiving from the system, in response to the request, the ordered list of content items; and

displaying at least a subset of the received ordered list of content items.

9. A system for serving content, located remotely from a client, comprising:

memory;

one or more processors;

one or more programs stored in the memory and configured for execution by the one or more processors, the one or more programs including:

instructions for retrieving, for each of a plurality of content feeds, a currently published set of content items for a respective content feed, and for merging the retrieved content items into a previously stored set of previously published content items for the respective content feed; wherein a database storing content items for the respective content feed includes both the currently published set of content items and historical content items no longer published by the respective content feed; and

instructions for transmitting to the client, in response to a request from the client, an ordered list of content items for a specified stream, the specified stream corresponding to one or more content feeds in the plurality of content feeds, the transmitted content items including at least one historical content item for at least one of the content feeds corresponding to the specified stream.

10. The system of claim **9**, wherein the ordered list of content items transmitted to the client includes at least one content item in the currently published set of content items for the specified stream and the at least one historical content item for the specified stream.

11. The system of claim **9**, wherein the specified stream comprises two or more content feeds in the plurality of content feeds, and the transmitted content items include at least one historical content item for at least one of the content feeds corresponding to the specified stream.

12. The system of claim **9**, wherein the specified stream includes a plurality of contents items tagged by a user, wherein the plurality of tagged content items are from one or more of the content feeds.

13. The system of claim **9**, wherein the specified stream comprises two or more content feeds in the plurality of content feeds and includes a plurality of contents items tagged by a user, wherein the plurality of tagged content items are from one or more of the content feeds.

14. The system of claim **9**, wherein the specified stream comprises one or more other streams labeled by a user, wherein a respective stream of the one or more other streams comprises two or more content feeds in the plurality of content feeds.

15. The system of claim **9**, wherein the specified stream comprises one or more other streams labeled by a user, wherein a respective stream of the one or more other streams comprises two or more content feeds in the plurality of content feeds; and

wherein the specified stream includes a plurality of content items tagged by a user, wherein the plurality of tagged content items are from one or more of the content feeds.

16. A computer readable storage medium storing one or more programs configured to be executed by a server computer system remotely located from a client, the one or more programs comprising:

instructions for retrieving, for each of a plurality of content feeds, a currently published set of content items for a respective content feed, and for merging the retrieved content items into a previously stored set of previously published content items for the respective content feed; wherein a database storing content items for the respective content feed includes both the currently published set of content items and historical content items no longer published by the respective content feed; and

instructions for transmitting to the client, in response to a request from the client, an ordered list of content items for a specified stream, the specified stream corresponding to one or more content feeds in the plurality of content feeds, the transmitted content items including at least one historical content item for at least one of the content feeds corresponding to the specified stream.

17. The computer program product of claim **16**, wherein the ordered list of content items transmitted to the client includes at least one content item in the currently published set of content items for the specified stream and the at least one historical content item for the specified stream.

18. The computer program product of claim **16**, wherein the specified stream comprises two or more content feeds in the plurality of content feeds, and the transmitted content items include at least one historical content item for at least one of the content feeds corresponding to the specified stream.

19. The computer program product of claim **16**, wherein the specified stream includes a plurality of contents items tagged by a user, wherein the plurality of tagged content items are from one or more of the content feeds.

20. The computer program product of claim **16**, wherein the specified stream comprises two or more content feeds in the plurality of content feeds and includes a plurality of contents items tagged by a user, wherein the plurality of tagged content items are from one or more of the content feeds.

21. The computer program product of claim **16**, wherein the specified stream comprises one or more other streams labeled by a user, wherein a respective stream of the one or more other streams comprises two or more content feeds in the plurality of content feeds.

22. The computer program product of claim **16**, wherein the specified stream comprises one or more other streams labeled by a user, wherein a respective stream of the one or more other streams comprises two or more content feeds in the plurality of content feeds; and

wherein the specified stream includes a plurality of content items tagged by a user, wherein the plurality of tagged content items are from one or more of the content feeds.

23. A system for serving content, comprising:

means for retrieving, for each of a plurality of content feeds, a currently published set of content items for a respective content feed, and for merging the retrieved content items into a previously stored set of previously published content items for the respective content feed; wherein a database storing content items for the respec-

tive content feed includes both the currently published set of content items and historical content items no longer published by the respective content feed; and

means for transmitting to the client, in response to a request from the client, an ordered list of content items for a specified stream, the specified stream corresponding to one or more content feeds in the plurality of content feeds, the transmitted content items including at least one historical content item for at least one of the content feeds corresponding to the specified stream.

* * * * *