

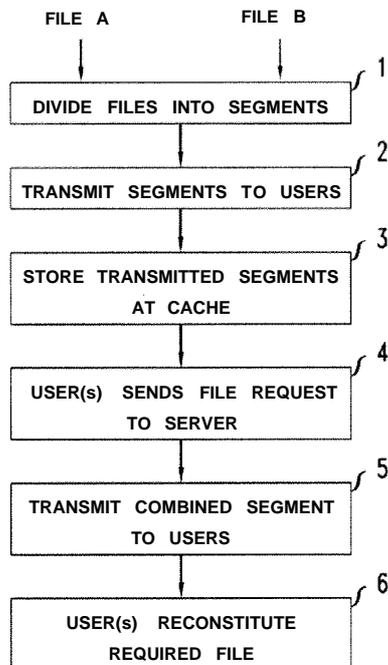


- (51) **International Patent Classification:**
H04L 29/08 (2006.01) H04N 21/433 (201 1.01)
- (21) **International Application Number:**
PCT/US20 12/030090
- (22) **International Filing Date:**
22 March 2012 (22.03.2012)
- (25) **Filing Language:** English
- (26) **Publication Language:** English
- (30) **Priority Data:**
13/077,420 31 March 2011 (31.03.2011) US
- (71) **Applicant (for all designated States except US):** ALCA-TEL LUCENT [FR/FR]; 3, avenue Octave Greard, F-75007 Paris (FR).
- (72) **Inventor; and**
- (75) **Inventor/Applicant (for US only):** MADDAH-ALI, Mohammadali [CA/US]; 6 Marvin Lane, Piscataway, NJ 08854 (US).
- (74) **Agent:** MURGIA, Gregory, J.; Alcatel-lucent Usa Inc., Attention: Docket Administrator-room 3d-201, 600-700 Mountain Avenue, Murray Hill, NJ 07974-0636 (US).
- (81) **Designated States (unless otherwise indicated, for every kind of national protection available):** AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IS, JP, KE, KG, KM, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LT, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.
- (84) **Designated States (unless otherwise indicated, for every kind of regional protection available):** ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK,

[Continued on next page]

(54) **Title:** MANAGING DATA FILE TRANSMISSION

FIG. 1



(57) **Abstract:** In a method for managing transmission of data files to users, a first file is divided into a plurality of first segments and a second file is divided into a plurality of second segments. A first segment is sent to a first user and a different first segment is sent to a second user. A second segment is sent to the first user and a different second segment is sent to the second user. At least part of a segment sent to the first user is combined with at least part of a segment sent to the second user to produce a combined segment which is of smaller size than the total size of the at least parts of segments before combining. The combined segment is transmitted to the first user and to the second user for each user to recover a segment using the combined segment and at least part of a segment.

WO 2012/134941 A1



SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

— before the expiration of the time limit for amending the claims and to be republished in the event of receipt of amendments (Rule 48.2(h))

Published:

— with international search report (Art. 21(3))

MANAGING DATA FILE TRANSMISSION

FIELD OF THE INVENTION

The present invention relates to a method and apparatus for managing transmission of data files to users, and more particularly, but not exclusively, to
5 transmission of data files to be cached locally to a user.

BACKGROUND

A server may hold data files to be sent to a user over a network, which may be, for example, the Internet or some other network. The data files may be media content
10 files, or of some other type, and sent to the user when the user requests them, or when the content creator or provider wants to distribute them, for example. If a user does not require immediate access to a data file, or wishes to keep it available for later access, the data file may be stored in a cache memory local to the user. For example, the cache memory may be a hard drive included in the user equipment or the cache
15 memory may be separately provided and arranged to be readily accessible by the user, for example, by being located nearer the edge of a network than the server. When the user needs a cached file, it may be obtained from the cache memory with reduced or no network resources.

Data transmission may be efficiently managed by sending data files to be
20 cached at times when network resources are at high capacity and/or low cost. For example, a data file may be sent if a WiFi network is available, or when the user is in a low-traffic area or it is a low-traffic time period.

BRIEF SUMMARY

25 According to a first aspect of the invention, in a method for managing transmission of data files to users, a first file is divided into a plurality of first segments and a second file is divided into a plurality of second segments. A first segment is sent to a first user and a different first segment is sent to a second user. A second segment is sent to the first user and a different second segment is sent to the
30 second user. At least part of a segment sent to the first user is combined with at least part of a segment sent to the second user to produce a combined segment which is of

- 2 -

smaller size than the total size of the at least parts of segments before combining. The combined segment is transmitted to the first user and to the second user for each user to recover a segment using the combined segment and at least part of a segment.

The smaller size of the combined segment compared to the total size of the at least parts of segments before combining means that the combined segment requires less space if it is to be cached in a cache, or caches, local to the users. Additionally, as the combined segment is of smaller size, it requires fewer network resources for transmission than if the segments were to be sent individually in uncombined form. This is advantageous even for arrangements in which segments are not cached at a memory cache or caches local to a user.

The user or users may be any node in a network. For example, in a wireless network, a user could be a router or an end user or some other network node.

A method in accordance with the invention may be applied to wireless, fixed line or other types of network and is not limited to one particular technology type.

A method in accordance with the invention may be applied where there are f files and K users, and includes partitioning each file into 2 to the power of K segments, where for each subset of users, there is a segment which is saved on all of the users in that subset.

In one embodiment, the first and second segments sent to the first user may be cached a first cache memory local to the first user. Similarly, the first and second segments sent to the second user may be cached at a second cache memory local to the second user. Thus, a cache memory may be arranged to store only part of the first file and part of the second file, which may be advantageous when the cache memory is of insufficient size to store the complete files. When at a later time, one or both of the users want to access the first file or the second file, the entire file need not be sent to them as they already have access to some segments locally. Thus, even though remaining segments may in some cases need to be sent at a relatively expensive time, or during high network loads, as the entire file need not be sent then, overall costs for sending the file may be reduced. Additionally, the combined segment is smaller than the total size of the contributing segments, which may provide efficient file transmission management.

In an embodiment, the at least part of a segment sent to the first user is

- 3 -

combined with at least part of a segment sent to the second user by using addition in a finite field. In one embodiment, the finite field is the binary field.

In one method, the first and second segments sent to the first and second users are sent prior to the combined segment being sent to the first and second users.

5 However, in another embodiment, the data is sent in a different order.

In one embodiment, a segment is divided into first and second parts, the first part is incorporated into the combined segment and the second part transmitted to the first and second users. The second part may be transmitted at the same time as the combined segment or as a separate transmission.

10 According to a second aspect of the invention, a data transmission manager for managing transmission of data files to users comprises: a divider for dividing a first file into a plurality of first segments and for dividing a second file into a plurality of second segments; transmitter apparatus for sending a first segment to a first user and a different first segment to a second user, and for sending a second segment to the
15 first user and a different second segment to the second user; a combiner for combining at least part of a segment sent to the first user with at least part of a segment sent to the second user to produce a combined segment which is of smaller size than the total size of the at least parts of segments before combining; and said transmitter apparatus being operative to send the combined segment to the first user
20 and to the second user for each of the first and second users to recover a segment using the combined segment and at least part of a segment.

BRIEF DESCRIPTION OF THE DRAWINGS

Some embodiments of the present invention will now be described by way of
25 example only, and with reference to the accompanying drawings, in which:

Figure 1 schematically illustrates a flowchart of a method in accordance with the invention;

Figure 2 is a schematic graph illustrating transmission; and

Figure 3 schematically illustrates an apparatus for implementing the method of
30 Figure 1.

DETAILED DESCRIPTION

In a first embodiment, assume that there is a first file A and a second file B available from a server, each file being 1MB in size. There are first user U1 and second user U2, and each user has an associated local cache memory M1 and M2 respectively, M1 and M2 being 1MB each. It is assumed that the history of these two users U1 and U2 shows that each may require file A and file B with equal probability.

In this caching scheme, it is necessary to broadcast on average 0.5 MB in expensive networks. Moreover, the peak rate is 0.5 MB.

With reference to Figure 1, at 1, the first and second files are divided into several segments. File A is divided into two equal size segments A1 and A2, where each segment is 0.5 MB, such that $A=(A1, A2)$.

Similarly, file B is divided into two equal size segments B1 and B2, where each segment is 0.5 MB, and $B=(B1, B2)$.

At 2, segments A1 and B1 are transmitted to the first user U1 and, at 3, saved on the associated cache memory M1. Also, A2 and B2 are transmitted to the second user U2 and saved on the associated cache memory M2. The caching strategy is summarized in Table 1 and the probability in Table 2.

Caching Table	
User One	A1, B1
User Two	A2, B2

Table 1

Probability Table	File A	File B
User One	$p1A=0.5$	$p1B=0.5$
User Two	$p2A=0.5$	$p2B=0.5$

Table 2

Thus, each user has part of each file.

At a later time, for example, if the first user U1 wants to access file A and the second user U2 requires file B, neither has sufficient segments cached to provide a

- 5 -

complete file. The users transmit their requirements to the server at 4.

At 5, the server arranges for the transmitter to broadcast a combined segment A_2+B_1 , where + indicates summation in a finite field to generate the combined segment. In this example, combination is carried out in the binary field and therefore + is simply bit-wise XOR operation. The combined segment A_2+B_1 has 0.5 MB size. This compares with a size of 0.5MB for segment A_2 and 0.5 MB for segment B_1 , that is, 1MB in total.

In the next stage, shown at 6, the first user U_1 receives A_2+B_1 and already has B_1 in its cache memory M_1 . Thus, the first user U_1 can recover A_2 by the operation $(A_2+B_1)-B_1$. The first user U_1 also has A_1 in its cache memory M_1 . Therefore, the first user U_1 has both A_1 and A_2 and is able to reconstitute the required file A.

The second user U_2 also receives A_2+B_1 broadcast by the transmitter. The second user U_2 already has A_2 in its cache memory, and therefore can recover B_1 by the operation $(A_2+B_1)-A_2$. The second user U_2 already has B_2 in its cache memory M_2 . Therefore, both B_1 and B_2 are available to the second user U_2 to reconstitute the file B it requires.

In another scenario, assuming that the starting point is that shown in Table 1, assume that the first and second users both require file A. In this case, the transmitter broadcasts A_2+A_1 , where + again indicates combination is carried out in the binary field and is a bit-wise XOR operation. The combined segment A_2+A_1 has a size of 0.5 MB compared to a size of 0.5MB for segment A_2 and 0.5 MB for segment A_1 , that is, 1MB in total

The first user U_1 thus receives A_2+A_1 and already has A_1 in its cache memory. Therefore it can recover A_2 by $(A_2+A_1)-A_1$. Therefore, the first user U_1 has both segments A_1 and A_2 and can thus reconstitute file A.

The second user U_2 also receives the broadcast A_2+A_1 and already has A_2 in its cache memory M_2 . Therefore it can recover A_1 by $(A_2+A_1)-A_2$. Therefore, it will have both A_1 and A_2 and can thus reconstitute file A.

For other cases, the broadcasting strategy and also the recovery method are shown in Table 3 below. It can be seen that no matter which file each user requests, the transmitter needs to broadcast only 0.5 MB. Therefore, the average rate of data that the transmitter broadcasts is 0.5 MB.

Demand	Modified Requirements Due to Caching	Transmitter Sends	Decoding Strategy
User One Wants A User Two Wants B	User One Wants A2 User Two Wants B1	A2+B1	User One: (A2+B1)-B1 User Two: (A2+B1)-B2
User One Wants B User Two Wants A	User One Wants B2 User Two Wants A1	A1+B2	User One: (A1+B2)-A1 User Two: (A1+B2)-B1
User One Wants A User Two Wants A	User One Wants A2 User Two Wants A1	A1+A2	User One: (A1+A2)-A1 User Two: (A1+A2)-A2
User One Wants B User Two Wants B	User One Wants B2 User Two Wants A1	B1+B2	User One: (B1+B2)-B1 User Two: (B1+B2)-B2

Table 3

5 In this embodiment, the caching scheme is optimized and the average load or maximum load of the network is minimized. In the simple example given above, the caching scheme can reduce the average load up to 50% and peak load of 100%.

Figure 2 illustrates the normalized average transmission rate against the normalized cache-memory size for two files, of size F, and two users, each with cache size M. In addition, the probability that each user may need one of the files is equally likely.

In another embodiment, it is assumed that there are first and second users User 1 and User 2 with available caching memory size of M_1 and M_2 respectively. In addition, assume that the server has two files A and B with sizes F_A and F_B respectively, i.e.

$$|A|=F_A$$

$$|B|=F_B.$$

The users may need one of the files with some probabilities listed in the following Table 4:

User 1 wants	A	A	B	B
User 2 wants	A	B	A	B
Probability	PAA	PAB	PBA	PBB

Table 4

Each file is divided into 5 parts, which need not be equal, to give:

$$A = \{A_0, A_1, A_2, A_{12}\}$$

5 $B = \{B_0, B_1, B_2, B_{12}\}$

with sizes

$$|A_0| = x_0, |A_i| = x_i, |A_2| = x_2, |A_{12}| = x_{12},$$

$$|B_0| = y_0, |B_i| = y_i, |B_2| = y_2, |B_{12}| = y_{12},$$

Therefore,

10 $x_0 + x_i + x_2 + x_{12} = F_A$

$$y_0 + y_i + y_2 + y_{12} = F_B$$

Parts of the files denoted by A_i and B_i are cached on User 1 memory.

Parts of the files denoted by A_2 and B_2 are cached on User 2 memory.

15 Parts of the files denoted by A_{12} and B_{12} are cached on both users' memories.

Parts of the files denoted by A_0 and B_0 are cached on none of the memories,

Therefore,

$$x_i + y_i + x_{i_2} + y_{i_2} \leq M_i$$

20 $x_2 + y_2 + x_{12} + y_{12} \leq M_2$

In a first scenario, User 1 asks for file A and User 2 asks for file B. Thus,

User 1 requires A_0 which is only available at the server

25 User 2 requires B_0 which is only available at the server

User 1 already has A_i and A_{12} .

User 2 already has B_i and B_{12} .

User 1 wants A_2 while User 2 has it on its caching memory

User 2 wants B_i while User 1 has it on its caching memory

Therefore the server must send both A_0 and B_0 to Users 1 and 2.

Assuming that $|A_2| > |Bi|$, then the server partitions segment A_2 into two parts, denoted by A_{2U} and A_{2L} , i.e. $A_2 = \{ A_{2U}, A_{2L} \}$, where $|A_{2U}| = |Bi|$. Then the server
 5 sends combined segment $A_{2U} + Bi$, where + is addition in binary field or any other finite field, and also sends A_{2L} . Note that:

$$|A_{2U} + Bi| \text{ plus } |A_{2L}| = |A_2| = \max \{ |A_2|, |Bi| \}$$

10 where $\max \{ |A_2|, |Bi| \}$ is the maximum of the size of A_2 and the size of Bi .
 Then user 1 can use $A_{2U} + Bi$, A_{2L} , and Bi to recover $A_2 = \{ A_{2U}, A_{2L} \}$. Also user 2 can use $A_{2U} + Bi$ and A_2 to recover Bi .

On other hand, if $|A_2| < |Bi|$, then the server divides segment Bi into two parts, denoted by B_{iU} and B_{iL} , i.e. $Bi = \{ B_{iU}, B_{iL} \}$, where $|B_{iU}| = |A_2|$. Then the server
 15 sends combined segment $B_{iU} + A_2$ and also sends B_{iL} and again + is addition in binary field or any other finite fields. Note that

$$|B_{iU} + A_2| \text{ plus } |B_{iL}| = |Bi| = \max \{ |A_2|, |Bi| \}.$$

Then user 2 can use $B_{iU} + A_2$, B_{iL} , A_2 , to recover $Bi = \{ B_{iU}, B_{iL} \}$. Also User
 20 1 can use $B_{iU} + A_2$, and Bi to recover A_2 .

Then each user has the necessary segments to reconstitute the file requested by it.

In a second scenario, User 1 and User 2 both request file A. Note:

25 Users 1 and 2 want A_0 which is only available at the server memory

Both Users 1 and 2 already have A_{12} .

User 1 wants A_2 while User 2 has it on its caching memory

User 2 wants A_i while User 1 has it on its caching memory

Therefore the server has to sends A_0 to users 1 and 2.

30 Assume that $|A_2| > |Ai|$, then the server partitions segment A_2 into two parts, denoted by A_{2U} and A_{2L} , i.e. $A_2 = \{ A_{2U}, A_{2L} \}$, where $|A_{2U}| = |Ai|$. Then the server sends combined segment $A_{2U} + Ai$ and A_{2L} where + is addition in binary field or any other finite fields. Note that

$$|A_{2U} + Ai| + |A_{2L}| = |A_2| = \max \{ |A_2|, |Ai| \}.$$

Then User 1 can use $A_{2U} + Ai$, A_{2L} , and Ai to recover $A_2 = \{ A_{2U}, A_{2L} \}$. Also User 2 can use $A_{2U} + Ai$ and A_2 to recover Ai .

On other hand if $|A_2| < |Ai|$, then the server partitions Ai into two parts, denoted by A_{1U} and A_{1L} , i.e. $Ai = \{ A_{1U}, A_{1L} \}$, where $|A_{1U}| = |A_2|$. Then the server sends $A_{1U} + A_2$ and A_{1L} where $+$ is addition in binary field or any other finite fields. Note that

$$|A_{1U} + A_2| + |A_{1L}| = |Bi| = \max \{ |A_2|, |Bi| \}.$$

Then User 2 uses $A_{1U} + Ai$, A_{1L} , and A_2 to recover $Ai = \{ A_{1U}, A_{1L} \}$. Also User 1 can use $A_{1U} + A_2$, and Ai to recover A_2 .

Following this strategy, the rate required for different cases is illustrated in Table 5 below:

User 1 wants	A	A	B	B
User 2 wants	A	B	A	B
Probability	p_{AA}	p_{AB}	p_{BA}	p_{BB}
Required	$x_0 +$	$x_0 + y_0 +$	$x_0 + y_0 +$	$y_0 +$
Rate	$\max\{x_1, x_2\}$	$\max\{x_2, y_1\}$	$\max\{x_1, y_2\}$	$\max\{y_1, y_2\}$

Table 5

Therefore, the average rate is equal to:

$$p_{AA} (x_0 + \max\{x_1, x_2\}) + p_{AB} (x_0 + y_0 + \max\{x_2, y_1\}) + p_{BA} (x_0 + y_0 + \max\{x_1, y_2\}) + p_{BB} (y_0 + \max\{y_1, y_2\})$$

Therefore, $x_0, x_1, x_2, x_{12}, y_0, y_1, y_2$, and y_{12} may be chosen to minimize the average rate:

$$\text{Min } p_{AA} (x_0 + \max\{x_1, x_2\}) + p_{AB} (x_0 + y_0 + \max\{x_2, y_1\}) + p_{BA} (x_0 + y_0 + \max\{x_1, y_2\}) + p_{BB} (y_0 + \max\{y_1, y_2\})$$

Subject to:

$$x_0 + x_1 + x_2 + x_{12} = F_A$$

- 10 -

$$\begin{aligned}
y_0 + y_1 + y_2 + y_{12} &= F_B \\
x_1 + y_1 + x_{12} + y_{12} &\leq M_1 \\
x_2 + y_2 + x_{12} + y_{12} &\leq M_2 \\
x_0, x_1, x_2, x_{12}, y_0, y_1, y_2, \text{ and } y_{12} &\geq 0
\end{aligned}$$

5

If the maximum rate is the main concern, the following optimization can be used:

$$\begin{aligned}
&\text{Min Max } [p_{A,A}(x_0 + \max\{x_1, x_2\}), p_{A,B}(x_0 + y_0 + \max\{x_2, y_1\}), p_{B,A} \\
10 \quad &(x_0 + y_0 + \max\{x_1, y_2\}), p_{B,B}(y_0 + \max\{y_1, y_2\})]
\end{aligned}$$

Subject to

$$\begin{aligned}
&x_0 + x_1 + x_2 + x_{12} = F_A \\
15 \quad &y_0 + y_1 + y_2 + y_{12} = F_B \\
&x_1 + y_1 + x_{12} + y_{12} \leq M_1 \\
&x_2 + y_2 + x_{12} + y_{12} \leq M_2 \\
&x_0, x_1, x_2, x_{12}, y_0, y_1, y_2, \text{ and } y_{12} \geq 0
\end{aligned}$$

20 Other aspects may be used to refine the method. For example, if caching has some costs, the corresponding cost may be added to the objective function of the optimization of the data rate. For example, if sending data to both users has different costs than sending data to one user, the objective function may be correspondingly modified. Also, if one user already has some parts of the files, then this may be

25 exploited in the optimization. Also, users may have some priorities on segments of a file which should be cached on their memory. Such priorities may be taken into account in the optimization. For transmission of data to the users, users priorities may be considered. For example, files may be transmitted in a manner that allows a user to recover its file in a specific order.

30 The method can be extended to any number of files and any number of users. For example, assume that there are three users, user 1, user 2, and user 3, and three files A, B, and C.

Each file is partitioned into 8 segments as follows:

$$A = \{A_0, A_1, A_2, A_3, A_{12}, A_{13}, A_{23}, A_{i_{23}}\}$$

$$B = \{B_0, B_1, B_2, B_3, B_{12}, B_{13}, B_{23}, B_{i_{23}}\}$$

$$C = \{C_0, C_1, C_2, C_3, C_{12}, C_{13}, C_{23}, C_{i_{23}}\}$$

5

Then the segments are saved as follows:

Segments X_0 at none of the users, for $X=A,B,C$

Segments X_i at user i , for $i=1,2,3$, and $X=A,B,C$

Segments X_{ij} at both users i and j for $i,j=1,2,3$, and $X=A,B,C$

10 Segments $X_{i_{23}}$ at all users i and j for $X=A,B,C$

For simplicity, it is assumed in this example that Segments X_i for $i=1,2,3$, and $X=A,B,C$ have the same size, and also X_{ij} for $i,j=1,2,3$, and $X=A,B,C$ have the same size. This assumption is just for this example, and is not general

15 requirement.

Then, if for example, user 1 wants A, user 2 wants B, and user 3 wants C, then the transmitter sends

$$A_0$$

$$B_0$$

$$C_0$$

$$A_2+B_1$$

$$A_3+C_1$$

$$B_3+C_2$$

$$A_{23}+B_{13}+C_{12}$$

20

25 Then each user has received enough segments and combined segments to retrieve the desired file.

If for example all users want A, then the transmitter sends

$$A_0$$

$$e_1A_1+ e_2A_2+ e_3A_3$$

$$g_1A_1+ g_2A_2+ g_3A_3$$

$$q_1A_{i_{23}}+ q_2A_{i_{13}}+ q_3A_{i_{23}}$$

30

- 12 -

where the operations are any large enough finite-field, and g_j, q_i are from the same field. Then each user has enough equations to solve for entire A.

The size of the segments can be optimized as explained above. The size of the files does not need to be the same.

5 If there are f files and K users, then each file is partitioned into 2 to the power of K segments, where for each subset of users, there is a segment which is saved on all of the users in that subset. The size of the some of these segments may be zero.

With reference to Figure 3, a data transmission manager 7 for implementing the embodiment described with reference to Figure 1 includes a content store 8 which
10 holds data files A and B. A divider 9 accesses the content store 8 to obtain the data files A and B and divides the files into segments. Some of the segments are to be transmitted to users 10 and 11 at a time when network capacity is large and/or resources required are not expensive. Those segments to be initially transmitted to the users 10 and 11 are selected by a control processor 12 which also maintains a record
15 of which segments are transmitted. The control processor 12 instructs a server 13 as to which segment is to be transmitted to which user. The server 13 acquires the relevant segments and sends them via transmitter 14 to the users 10 and 11 over a network. The users 10 and 11 each have an associated memory cache 15 and 16 respectively in which to store the data segments sent to them from a server 11. Each
20 user receives segments from file A and also from file B.

When the users 10 and 11 wish to have a complete file A or B, they send a message to the server 13 over the network. The server 13 and control processor 12 determine what combined segment is required to fulfill the users requests. The combined segment is produced from file segments by combiner 17 and delivered via
25 server 13 to the users 10 and 11. The users 10 and 11 are then able to reconstitute the complete files using the previously transmitted segments and the combined segment.

The apparatus illustrated in Figure 3 may be adapted to perform more complex data file delivery to implement other methods as set out above.

The functions of the various elements shown in the figure, including any
30 functional blocks labeled as "processors", may be provided through the use of dedicated hardware as well as hardware capable of executing software in association with appropriate software. When provided by a processor, the functions may be

- 13 -

provided by a single dedicated processor, by a single shared processor, or by a plurality of individual processors, some of which may be shared. Moreover, explicit use of the term "processor" should not be construed to refer exclusively to hardware capable of executing software, and may implicitly include, without limitation, digital
5 signal processor (DSP) hardware, network processor, application specific integrated circuit (ASIC), field programmable gate array (FPGA), read only memory (ROM) for storing software, random access memory (RAM), and non volatile storage. Other hardware, conventional and/or custom, may also be included.

The present invention may be embodied in other specific forms without
10 departing from its spirit or essential characteristics. The described embodiments are to be considered in all respects only as illustrative and not restrictive. The scope of the invention is, therefore, indicated by the appended claims rather than by the foregoing description. All changes that come within the meaning and range of equivalency of the claims are to be embraced within their scope.

15

- 14 -

I CLAIM:

1. A method for managing transmission of data files to users, comprising:
 - dividing a first file into a plurality of first segments;
 - dividing a second file into a plurality of second segments;
 - 5 sending a first segment to a first user and a different first segment to a second user;
 - sending a second segment to the first user and a different second segment to the second user;
 - combining at least part of a segment sent to the first user with at least part of a
 - 10 segment sent to the second user to produce a combined segment which is of smaller size than the total size of the at least parts of segments before combining; and
 - transmitting the combined segment to the first user and to the second user for each user to recover a segment using the combined segment and at least part of a
 - segment.
- 15 2. The method as claimed in claim 1 wherein the first and second segments sent to the first and second users are sent prior to the combined segment being sent to the first and second users.
- 20 3. The method as claimed in claim 1 and including dividing a segment into first and second parts; incorporating the first part into the combined segment; and transmitting the second part to the first and second users.
- 25 4. The method as claimed in claim 3 and including transmitting the second part with the combined segment.
5. The method as claimed in claim 1 and including choosing the sizes of the first segments and of the second segments to minimize the average rate.
- 30 6. The method as claimed in claim 1 and including using the probability that the first file and the second file will be required by the first user and/or the second user in

- 15 -

optimizing the sizes of the first segments and of the second segments.

7. Data transmission manager for managing transmission of data files to users, comprising: a divider for dividing a first file into a plurality of first segments and for
5 dividing a second file into a plurality of second segments; transmitter apparatus for sending a first segment to a first user and a different first segment to a second user, and for sending a second segment to the first user and a different second segment to the second user; a combiner for combining at least part of a segment sent to the first
10 user with at least part of a segment sent to the second user to produce a combined segment which is of smaller size than the total size of the at least parts of segments before combining; and said transmitter apparatus being operative to send the combined segment to the first user and to the second user for each of the first and second users to recover a segment using the combined segment and at least part of a
15 segment.
8. The data transmission manager as claimed in claim in claim 7 wherein the divider is operative to divide a segment into first and second parts; the combiner is operative to incorporate the first part into the combined segment; and said transmitter apparatus is operative to send the second part to the first and second users.
20
9. The data transmission manager as claimed in claim 7 and including a processor for choosing the sizes of the first segments and of the second segments to minimize the average rate.
- 25 10. The data transmission manager as claimed in claim 7 and including a processor for using the probability that the first file and the second file will be required by the first user and/or the second user to optimize the sizes of the first segments and of the second segments.

FIG. 1

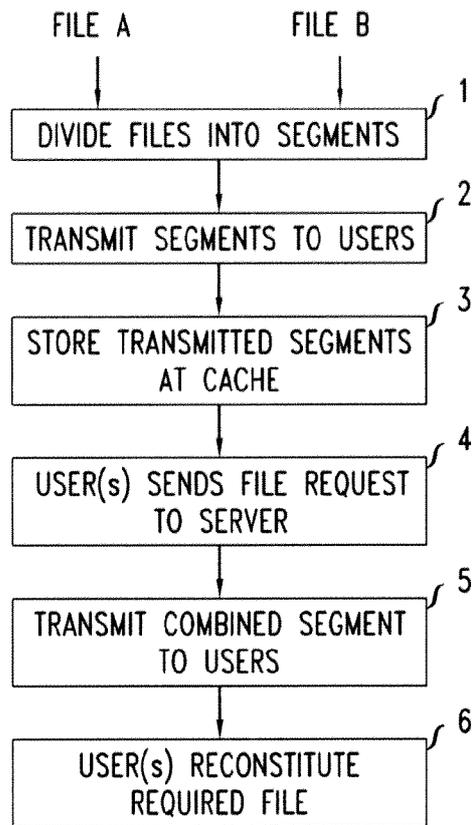


FIG. 2

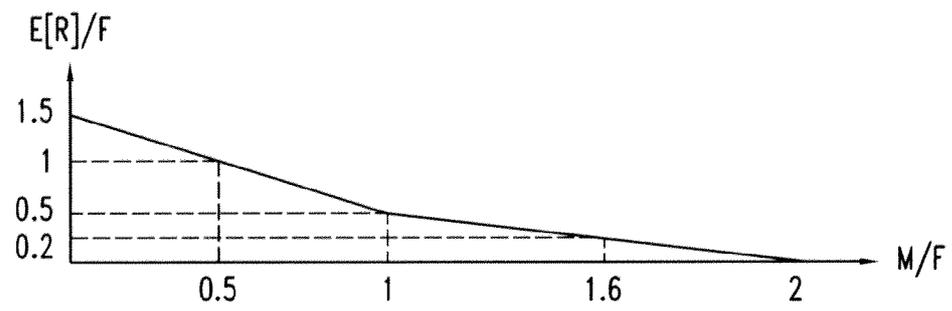
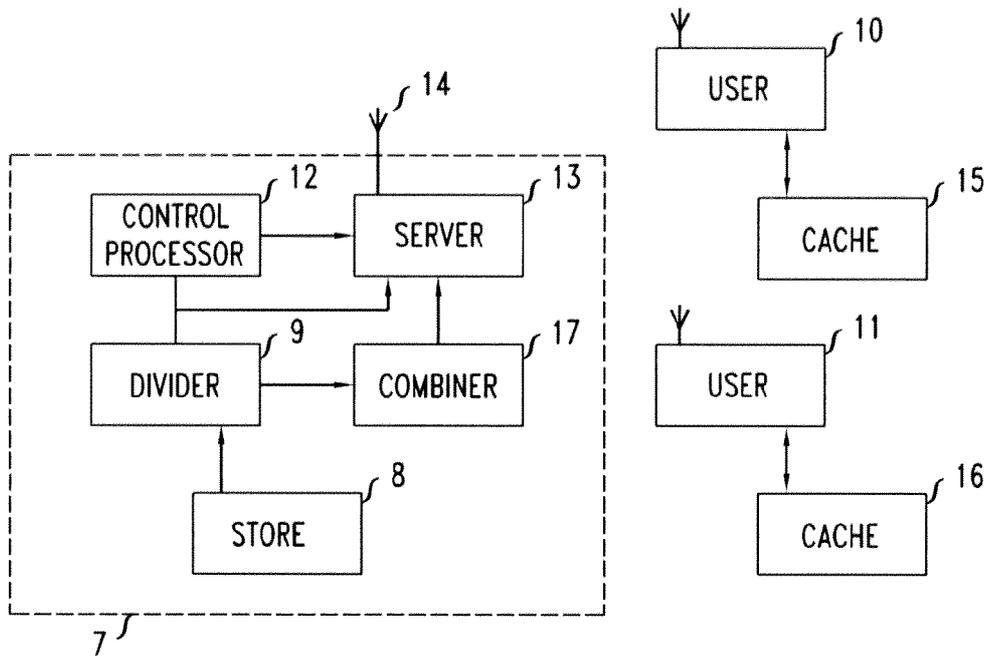


FIG. 3



INTERNATIONAL SEARCH REPORT

International application No
PCT/US2012/030090

A. CLASSIFICATION OF SUBJECT MATTER
INV. H04L29/08 H04N21/433
 ADD.
 According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED
 Minimum documentation searched (classification system followed by classification symbols)
H04L G06F H04N

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)
EPO-Internal , WPI Data

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	US 2009/222509 AI (KING CHAO [US] ET AL) 3 September 2009 (2009-09-03) paragraphs [0017] - [0039] paragraphs [0079] - [0104] figures 6,7 -----	1-10
A	US 2007/255844 AI (SHEN GUO BIN [CN] ET AL) 1 November 2007 (2007-11-01) paragraphs [0048] - [0090] -----	1-10
A	US 6 816 872 B1 (SQUIBB MARK [US]) 9 November 2004 (2004-11-09) column 1, line 41 - column 4, line 27 -----	1-10

Further documents are listed in the continuation of Box C. See patent family annex.

* Special categories of cited documents :

"A" document defining the general state of the art which is not considered to be of particular relevance	"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
"E" earlier application or patent but published on or after the international filing date	"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)	"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art
"O" document referring to an oral disclosure, use, exhibition or other means	"&" document member of the same patent family
"P" document published prior to the international filing date but later than the priority date claimed	

Date of the actual completion of the international search 18 July 2012	Date of mailing of the international search report 27/07/2012
--	---

Name and mailing address of the ISA/ European Patent Office, P.B. 5818 Patentlaan 2 NL - 2280 HV Rijswijk Tel. (+31-70) 340-2040, Fax: (+31-70) 340-3016	Authorized officer Vel oso Gonzal ez , J
--	--

INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No

PCT/US2012/030090

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 2009222509	AI	03-09-2009	NONE
US 2007255844	AI	01-11-2007	NONE
US 6816872	BI	09-11-2004	NONE