(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
22 March 2012 (22.03.2012)

PCT

(10) International Publication Number
WO 2012/034607 A1

(51) International Patent Classification:
H04L 29/06 (2006.01)

(21) International Application Number:
PCT/EP2011/001412

(22) International Filing Date:
22 March 2011 (22.03.2011)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
P201001199 17 September 2010 (17.09.2010) ES

(71) Applicant *(for all designated States except US)*: **TELEFONICA, S.A.** [ES/ES]; Gran Vía 28, E-28013 Madrid (ES).

(72) Inventors; and

(75) Inventors/Applicants *(for US only)*: **LAOUTARIS, Nikolaos** [GR/GR]; Gran Vía 28, E-28013 Madrid (ES). **RODRÍGEZ, Pablo** [ES/ES]; Gran Vía 28, E-28013 Madrid (ES). **YANG, Xiaoyuan** [ES/ES]; Gran Vía 28, E-28013 Madrid (ES). **SIRIVIANOS, Michael** [CY/CY]; Gran Vía 28, E-28013 Madrid (ES).

(74) Agent: **GONZALEZ-ALBERTO, Natalia**; Hermosilla 3, E-28001 Madrid (ES).

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

[Continued on next page]

(54) Title: A MULTI-HOP AND MULTI-PATH STORE AND FORWARD SYSTEM, METHOD AND PRODUCT FOR BULK TRANSFERS

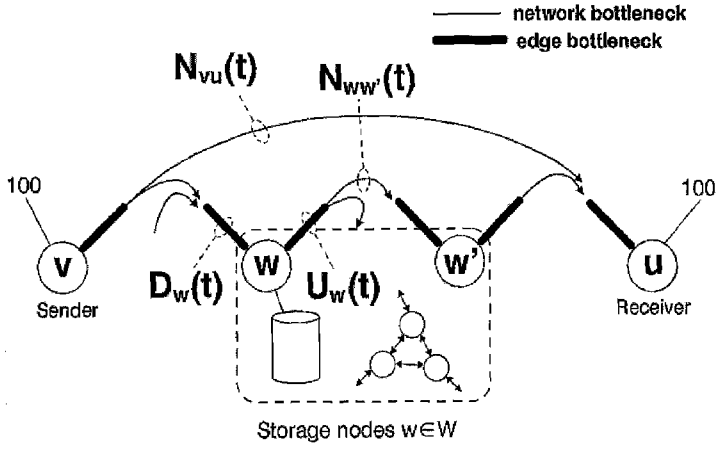

FIG. 2

(57) Abstract: A multi-hop and multi-path store and forward system for bulk transfers comprising a plurality of nodes (100) wherein each node (100) also comprises an overlay management module (1) that is arranged to add a new node (100) to the overlay, removing it, and maintaining the overlay connections during the node's participation in the system; a volume prediction module (2) which is arranged to maintain a time series with the predicted maximum volume of data that can be forward to each neighbour node during each one of the slots that make an entire day; bootstrapping, ISP - friendliness, and security means; a scheduling and routing module (3) for all data transfers between a sender v, storage nodes w and a receiver node u; and wherein the module (3) also comprises volume prediction means to calculate an initial transfer plan and keeps updating it periodically as it receives updated predictions from the nodes, and means for solving a maximum network flow optimization problem; and wherein a transmission management module (4) is arranged to receive, scheduling and routing commands from the scheduling and routing module (3) of senders v for which a local node is forwarding and executes them accordingly. Aim of the system is to split a large file into multiple chunks using a number of intermediate storage nodes to bypass MTABs (Multiple Time aligned Bottlenecks) and shorten the delivery times for bulk data.

WO 2012/034607 A1

Published:

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

A MULTI-HOP AND MULTI-PATH STORE AND FORWARD SYSTEM, METHOD AND PRODUCT FOR BULK TRANSFERS

Field of the invention

The object of the present invention uses the technique of splitting a large file into multiple chunks using a number of intermediate storage nodes to bypass MTABs (Multiple Time Aligned Bottlenecks) and shorten the delivery times for bulk data. It is employed a multi-path/multi-hop store-and-forward (SnF) routing and scheduling of chunks over extended periods of time.

The invention proposes a system and a method that schedules transfers of data over several hours in the future from a starting transmission time.

The present invention can be implemented as a generic bulk-transfer platform on top of a variety of systems.

BACKGROUND ART

The bottleneck link of a long-lived delay tolerant bulk (DTB) data flow can change during the course of a day as a result of pricing and diurnal demand intensity. Consider, for example, the case of a sender on the west coast of the US trying to send a 4.8GB DVD directly to a receiver on the east coast (3 hours time difference). Assume that both connect through ISPs that rate-limit the bulk flows of their residential customers during evening peak hours (e.g., from 8pm until 11pm) in order to make room for interactive traffic. It is easy to verify that even with fast 10Mbps symmetric access links the transfer can take up to 7 hours, whereas it should normally take only 1 hour given the access capacities. The problem is that if the sender initiates a direct transfer at 5pm pacific time (non-peak hour at the sending ISP), the transmission will be blocked for 3 hours by the receiving ISP who will be on its local peak hours. Passed those 3 hours, it will be the sender that enters its peak times and thus the transfer will be blocked for another 3 hours. Consequently, the transmission will finish at around midnight Pacific Time, with the sender transmitting at full speed during the last one hour. Similar problems can appear in corporate applications, e.g., between datacenters

situated at remote time zones that use each other for backup purposes. When one of them has free bandwidth due to reduced end-user traffic it cannot take full advantage of it and backup its data to a remote datacenter, because at that the very exact time the remote datacenter can be receiving its peak user traffic (due to its location at a remote time zone).

Similar situations can arise even within the same time zone. For instance, the sender and the receiver can set manually self-imposed bulk bandwidth caps at different times of the day to protect their interactive sessions, or their access networks can peak, due to their nature, at different local times (e.g., residential vs. enterprise vs. university networks).

The above situation is called Multiple Time Aligned Bottlenecks problem (or MTAB). The MTAB is a timing problem. Since all end-to-end (E2E) connections must cross the two edge links (first and last hop), spatial redirection in the form of single or multi-path overlay routing is not able to avoid the MTAB. What is needed instead, is a type of temporal redirection with the purpose of "time-shifting" the bandwidth of the sender while the receiver is blocked, thus rescuing it from being wasted. This can be achieved by performing Store-and-Forward (SnF) of data with the help of one or multiple intermediate storage nodes. Trading bandwidth for storage makes sense since storage costs keep dropping much faster than bandwidth costs [1]. Returning to the previous example, the sender can forward the entire file to such a storage node during any 1 hour window between 5pm to 8pm pacific, which in turn can deliver it to the receiver by 9pm pacific, yielding a 4- instead of 7-hour completion time.

Currently there exist four basic solutions for performing Delay Tolerant Bulk Transfers (DTB) on a daily basis:

- First Solution: With a dedicated network between source and destination. This solution has been applied in the case of the Large Hadron Collider (LHC) particle accelerator of CERN that generates 27 Terabytes of raw data daily that have to be transmitted to remote storing and analysis centers around the world. This is handled by the LHC Computing Grid (http://lcg.web.cern.ch/LCG/) which is a dedicated optical to networks with nodes in Europe, Asia and North America.

- Second solution: A simple solution for transferring DTB data is to send them in physical form using postal or courier services.

- Third solution: By sending the data through commercial ISPs using existing E2E connection oriented transfers. In this case the transmission can take place over ftp or http protocols using standard best effort services offered by ISPs. An alternative is to use interactive traffic friendly solutions like Scavenger [7] and TCP Nice [11] that take care to transmit the DTB data with the minimum possible negative impact on interactive traffic flowing through the same network. In the case of Scavenger this is achieved by tagging the DTB traffic as low priority and letting routers service it only if there is left over bandwidth from higher priority interactive traffic. In the case of TCP Nice, interactive traffic is protected by having TCP Nice back off and reduce its transmission rate faster than the standard TCP that carries the interactive traffic.

The third solution involves several drawbacks. In order to explain the related problems it should be considered an E2E-CBR policy of almost constant rate B/T that can deliver volume B within deadline T. In the case of LHC data this would require a stream of at least 2.5 Gbps (27 Terabytes per day). Assuming that the transfer has to reoccur every day, E2E-CBR would push up the 95-percentiles of the sending and receiving access ISPs by exactly B/T of 2.5 Gbps costing them anything between $75K and $225K in additional monthly transit costs ($30K-90K per Gbps according to recent prices). In other words, since E2E-CBR is bound to increase the charged volume by exactly its mean rate, it provides no advantage compared to buying dedicated lines of the exact same rate. Using Scavenger and TCP Nice one could send additional DTB data over the existing lines that carry interactive traffic, taking advantage of time intervals during which the interactive traffic is low. However, as it is eluded before, this can be problematic when there is underutilized capacity in some, but not all, involved links between two end points. In that case, E2E solutions, like Scavenger and TCP Nice, will adjust the transmission rate to the rate of the slowest link (the one with the least underutilized capacity to be used for DTB data). This prohibits using the underutilized capacity of all remaining links which in effect is being wasted.

In [2,3,4], of the same inventors of the present application, it is outlined a system for shipping a large amount of bulk (scientific data) at zero to low cost between access

ISPs that are subjected to 95/5-percentile pricing using single-path single-hop SnF through a unique storage node. This however has the problem that in some cases, depending on the amount of free bandwidth at each link, one needs multi-path and multi-hop SnF over multiple intermediate storage nodes. There are two reasons for this: (i) if the sender and the receiver are very fast, then a single storage node cannot utilize all their bandwidth; and (ii) sometimes there exist MTABs between the sender and the storage node or between the storage node and the receiver.

The solution to this is to use additional storage nodes and chain them together until no MTAB exists between any two nodes. However, to select among the different paths each with different number of hops, one needs to use information about the future availability of bandwidth between any two nodes.

- Fourth solution:

There has also been prior work that discloses employing store and forward to reduce the transfer time or the cost of transmission. We list them and compare them to current invention below:

In [2,3,4,14] the authors of this invention presented architectures for sending bulk data over a single hop of store and forward node. Under those proposed solutions, in addition to sending data directly to the final receiver over a TCP or TCP Nice channel, the sender can also upload data to an intermediate storage node which in turn, can forward the data to the final receiver. By doing so, it is possible to avoid wasting the bandwidth of the sender at times that due to the MTAB, the receiver cannot absorb data at the maximum speed that can be sustained by the sender.

The advantages of using multiple instead of a single storage node as proposes this invention are detailed at the end part of this description. In a nutshell they are: 1) One might need multiple intermediate storage nodes in order to saturate (use all the capacity of) fast sender-receiver nodes, and 2) Due to timing differences between sender and receiver, it might take multiple intermediate storage hops in order to get the data through. This is explained in Figure 3 of the drawings of this invention with an example. Because of requirements 1) and 2) the scheduling algorithm for controlling

the transmissions is much more elaborated than the simple one used in the single hop of storage nodes case.

Among others, the new algorithm proposed by this invention uses information about the future availability of bandwidth whereas the previous one was a much simpler online greedy algorithm that did not use such information.

In [15], the current inventors have proposed a single hop multiple path SnF solution that optimizes the cost instead of delivery time, and we do not consider edge bottlenecks, which require breaking nodes into virtual nodes. [15] considers perfect prediction and thus does not perform rescheduling. It provides a theoretical treatment of store and forward and does not take into account practical considerations.

To the best of the knowledge of the current inventors the invention described in [17] is the first to propose store and forward for the Internet. It stores, data in the form of messages of relatively small sizes. The purpose of storing the messages is to allow further processing of the message before it is forwarded to subsequent internet devices or to wait until an appropriate link with the subsequent device has been established. For example, packets can be compressed, transformed into packets of a different network type and so on.

The main differences between the current invention and the proposed solution in [17] are:

a) Store and forward (or message switching) is defined as a mode of transmission wherein data messages or their portions are accumulated, stored and retransmitted on a schedule or priority basis as desired and in accordance with channel and/or equipment availability to the next desired location, thereby maximizing the efficiency of transmission in accordance with a predetermined priority. Data are stored in intermediate nodes in order to be converted into appropriate formats such that they can be presented at the receiver terminal or in order to be compressed in order to reduce the transmission time. Such storing may be for a time duration of several hours or more when required.

In this invention instead store and forward is defined as a bulk data transfer method in which portions of the content are temporarily stored at intermediate nodes to avoid time aligned bottlenecks. Time-aligned bottlenecks are caused when the minimum of the capacity of the sender and the receiver is consistently low because of a difference in the times that the sender and the receiver are able to send and receive data at high rates. Our solution's gains come from the fact that we refrain from sending data to the receiver directly at a low rate and we instead send them at a faster rate to an intermediate node.

When the link from the intermediate node to the receiver becomes highly available, there are data at the intermediate node that can be sent at a high rate.

b) In this invention the IP protocol is used. It does not deal with packet level transmissions. Quoted prior art divide content in packets, whereas in this invention content is divided in data chunks.

This invention rely on TCP/IP to transfer data.

c) This invention incorporate an optimal maximum-flow-problem based scheduler, that uses time expansion to model storage.

Both systems consider large storing time scales (hours), however disclosed solution schedules only in terms of differentiating packets based on priority. It does not schedule in order to minimize the data transfer time.

d) In this invention inter-data center bulk traffic of much higher volumes is considered

The aim in the quoted prior art is minimizing the transfer times, while this invention aims at enabling messages of varying formats to be converted to the appropriate formats at the receiving terminals. It enables incompatible terminals to communicate with each other, while this invention enables data centers to minimize the time for their synchronization.

e) This invention does not aim at providing communication between terminals that use different communication protocols, but the aim of this invention is minimizing the transfer time of bulk data using TCP/IP. The cited prior art use SnF for transcoding and compression, which is an obvious application of storing at intermediate nodes. In this invention instead SnF is used to reduce the total transfer time. This is a profoundly innovative application of SnF.

f) Quoted prior art use SnF on a priority basis. This means that each packet has a specification by its source priority and is forwarded by intermediate storage nodes ahead of packets with lower priority. This invention does not specify a priority mechanism.

Also the different goals affects the design choices in the following ways:

A) In this invention resource prediction is required, while in quoted prior art they do not.

B) In this invention not concern exist about the short-time-scale dynamics of the transport protocol as having minimal impact on performance.

C) This invention has a transmission scheduler that optimizes the transfer time for a given bandwidth and intermediate storage prediction.

D) In this invention the intermediate storage nodes do not perform any conversion of the stored traffic (transcoding or compression).

Like this invention, the method and system described in [18] aims at content oriented routing in a storage-embedded network realizing reduced packet loss in order to reduce the transfer time of large data. It also, like this invention aims at increasing the efficiency of the network by increasing the utilization of existing bandwidth resources.

Unlike this invention, the method and system of [18] operates at the network layer instead of the application layer. This renders the deployment of the disclosed solution cumbersome, as it requires extensive upgrades of the network infrastructure. On the

other hand, the nodes of this invention can be easily installed at the edges of the network.

In addition, unlike this invention the method and system of [18] does not employ a max-flow-based optimal transmission scheduler and does not utilize resource prediction.

The solution described in [16] also employs store and forward.
In this invention, we aim at minimizing the transfer time of bulk data for a given cost, whereas in [16] the authors aim at reducing traffic bursts to reduce the customer's cost under 95th percentile pricing. To this end, they use a suboptimal greedy scheduler with limited knowledge of future resource availability, while in this invention a computationally tractable optimal scheduler is used with an elaborate resource prediction method.

SUMMARY OF THE INVENTION

In order to solve the afore-mentioned problems, the multi-hop and multi-path store and forward system for bulk transfers, object of the current invention, is based on a sender node v and a large file F that is sent by said sender node to a receiver node to a receiver node u; the sender can utilize any leftover uplink bandwidth that cannot be saturated by its direct connection to the receiver to forward additional pieces of the file to storage nodes w □ W . Nodes w can, in turn, store the pieces until they can forward them to the receiver or another storage node. The Minimum Transfer Time (MTT) problem is defined as follows:

Let MTT (F, v, u, W) denote the minimum transfer time to send a file of size F from v to u with the help of nodes w ϵ W under given uplink, downlink, storage, and network bottleneck constraints at all the nodes. The Minimum Transfer Time problem amounts to identifying a transmission schedule between nodes that yields the minimum transfer time MTT (F, v, u, W).

It is an object of the current invention to attain MTT (F, v, u, W) in controlled environments where the available bandwidth is a priori known, or approximate it given some error in bandwidth prediction and occasional node churn.

More concretely, in a first aspect of the invention, A multi-hop and multi-path store and forward system for bulk transfers comprises a plurality of nodes wherein each node also comprises:

(i) an overlay management module that is arranged to add a new node to the overlay, removing it, and maintaining the overlay connections during the node's participation in the system;

(ii) (ii) a volume prediction module which is arranged to maintain a time series with the predicted maximum volume of data that can be forward to each neighbour node during each one of the slots that make an entire day;

(iii) and bootstrapping, ISP-friendliness, and security means;

as already disclosed for Example in [8].

According to this invention the system further comprises:

(iii) a scheduling and routing module for all data transfers between the sender v, the storage nodes W and the receiver node u; and wherein the module also comprises volume prediction means to calculate an initial transfer plan and keeps updating it periodically as it receives updated predictions from the nodes, and means for solving a maximum network flow optimization problem; and

(iv) a transmission management module is arranged to receive scheduling and routing commands from the scheduling and routing module of senders for which the local node is forwarding and executes them accordingly.

In a second aspect of the invention, the multi-hop and multi-path store and forward method for bulk transfers, comprising the steps of: (i) managing the overlay adding a new node (100) to the overlay, removing it, and maintaining the overlay connections during the node's participation; (ii) maintaining a time series with the predicted maximum volume of data that can be forward to each neighbour node during each one of the slots that make an entire day; characterized in that it also comprises a (iii) scheduling and routing step for all data transfers between the sender v, the storage nodes W and the receiver node u; said step including a volume prediction step to calculate an initial transfer plan and keeping updating it periodically as it receives

updated predictions from the nodes; and solving a maximum network flow optimization problem step; and

(iv) a transmission management step wherein it is received the scheduling and routing commands of the senders for which the local node is forwarding and executing them accordingly.

In a third aspect of the present invention, it is disclosed a computer program product comprising computer-executable instructions embodied in a computer-readable medium for performing the steps of the aforementioned method.

Therefore, a first object of the current invention is to extend the SnF (Store and Forward policy to send data) scheduling beyond the simple case of single-path/single-hop, over a single unconstrained storage node. More specifically, (i) it is considered bottlenecks on the storage nodes and allow for multi-path transfers; (ii) it is allowed multi-hop transfers; (iii) it is added network bottlenecks, which were not considered in [2,3,4]; and (iv) it is addressed prediction error and churn.

These extensions render the proposed SnF solution of value not only to big applications (business or scientific data), but also to small residential users.

Moreover what sets the current invention further apart from existing bulk transfer systems is that it schedules transfers over several hours in the future. The simple shortsighted greedy algorithms proposed in [2,3,4] can fail in such cases and lead to data being stuck at intermediate storage nodes.

The current invention avoids this by scheduling transfers using network flow algorithms and a time expansion of the overlay graph based on predictions regarding the shape and the duration of future MTABs. Although seemingly cumbersome, coming up with such predictions is simplified by the fact that the system of the invention is only sensitive to aggregate transmitted volumes across hourly time scales (as opposed to highly volatile instantaneous bandwidth) and the fact that at such time scales, resource availability often follows predictable behavioral periodicities, e.g., 24-hour diurnal circles [5], or scheduled throttling periods [6].

Throughout the description and claims the word "comprise" and variations of the word, are not intended to exclude other technical features, components, or steps. Additional objects, advantages and features of the invention will become apparent to those skilled in the art upon examination of the description or may be learned by practice of the invention. The following examples and drawings are provided by way of illustration, and they are not intended to be limiting of the present invention. Furthermore, the present invention covers all possible combinations of particular and preferred embodiments described herein.

BRIEF DESCRIPTION OF THE DRAWINGS

FIG 1. Shows a block diagram of a node that is part of the system object of the present invention.

FIG 2. Shows a group of nodes included in the system object of the present invention, wherein Uw(t),Dw(t) are, respectively, uplink and downlink edge bottlenecks of w at time t; Nww' is the network bottleneck of overlay connection w → w' at time slot t; and Sw(t) represents the storage capacity of w at time slot t.

FIG 3. Shows the reduction of the MTT problem to a variant of the maximum flow problem using time expansion, wherein the upper portion of the figure shows the reduction when it is considered only network bottlenecks. The bottom (zoomed-in) portions of the figure shows the reduction when it is also considered edge bottlenecks.

DETAILED DESCRIPTION OF PARTICULAR EMBODIMENTS AND EXAMPLES

The multi-hop and multi-path store and forward system for bulk transfers, object of the current invention, comprises a plurality of nodes (100). FIG.1 summarizes the internal structure of a node that consists of the following modules:

- An overlay management module (1) that is arranged to add a new node (100) to the overlay, removing it, and maintaining the overlay connections during the node's participation in the system. After a call to a generic function join (v) (i.e. join the sender node), the node registers itself with a bootstrap server and starts its participation in the system. This means that it can be used to forward data for other nodes or initiate its own bulk transfer jobs through calls to the generic function send(v,u,F), i.e. send the

file F to the receiver node by the sender node. After the sending, the bootstrap server returns to it a set W of intermediate storage nodes that are used for performing store-and-forward.

- A volume prediction module (2) that is arranged to maintain a time series with the predicted maximum volume of data that can be forward to each neighbour node during each one of the slots that make an entire day. This is accomplished by monitoring the actual transfers, performing active probing, and using an appropriate estimator to update the predicted time series. The details of volume prediction are disclosed in [8].

- A scheduling and routing module (3) that is invoked at the sending node v following a call to send(v, u, F) and is arranged to schedule and route all data transfers between the sender v, the storage nodes W and the receiver node u . The module uses volume predictions to calculate an initial transfer plan and keeps updating it periodically as it receives updated predictions from the nodes. The transfer schedules are computed by solving a maximum network flow optimization problem. In a preferred embodiment the optimization problem is solved using the GLPK simplex-based solver of the PuLP Python package as it disclosed in [12].

- A transmission management module (4) that is arranged to receive scheduling and routing commands from the scheduling and routing module (3) of senders for which the local node is forwarding and executes them accordingly.

In addition, the above described system has to deal with bootstrapping, ISP-friendliness, and security.

The core feature of the invention is the scheduling and routing module (3). This module is invoked at the sending node v to send a file F to a receiver node u. This module is responsible for scheduling and routing all data transfers between the sender v, the storage nodes W and the receiver node u. The module uses volume predictions to calculate an initial transfer plan and keeps updating it periodically as it receives updated predictions from the nodes. The prediction module is out of the scope of this invention description and is described in detail in [8].

In order to better describe the behaviour of the scheduling and routing module (3), firstly it is described the scheduling under a perfect volume prediction scenario and afterwards, it is described the scheduling under imperfect volume prediction.

FIG.2 depicts a plurality of nodes (100) and summarizes the notation employed in the following description. Uw(t)and Dw(t) denote the volume of data from file F that can be sent on the physical uplink and downlink of node w during time slot t due to edge bottlenecks. In the simplest case, edge bottlenecks are given by the nominal capacity of the access links and thus are independent of time. Additional variability can be introduced by other applications or sessions running on the same host, thus claiming part of the capacity.

Nww'(t) denotes the volume of data that can be sent on the overlay link from w to w' due to network bottlenecks. The kind of network bottleneck we are primarily interested in is due to Deep Packet Inspection (DPI) devices that rate limit bulk flows at ISP peering points in order to reduce transit costs (under percentile pricing) and/or protect the QoS of interactive traffic. The reason is that such persistent rate limiting impacts the transferred volumes much more severely than transient congestion. Sw(t) denotes the maximum volume of data from file F that can be stored at w during time slot t. It is dependent on the node's total storage capacity and on the amount of storage required by other applications or sessions according with the present invention.

Next, it is described the scheduling and routing of data transfers between the sender v, the storage nodes W and the receiver node u.in three examples: a) perfect prediction of volume capacity of nodes under network bottlenecks only; b) perfect prediction under both network and edge bottlenecks; and c) imperfect prediction under network and edge bottlenecks.

Example of Perfect Prediction

In this case, Uw(t), Dw(t), Nww'(t) and Sw(t) are a priori known □ w □ W and □ t that make up an entire day.

(Case 1) Networks bottlenecks only

Assuming that there are no edge bottlenecks, the MTT problem of minimizing the delivery time of volume F using SnF relay nodes with time-varying storage and bandwidth on edges can be reduced into a maximum flow problem without storage and with fixed capacities on edges. The reduction is performed through time-expansion, as shown in Fig.3. Let Tmax be an upper bound on the minimum transfer time. It is possible to construct a max-flow problem over a flow network G(V,E) as follows:

- Node set V: For each SnF relay node w ϵ W in the definition of MTT as it is above described, we add to V Tmax virtual nodes w(t), $1 \leq t \leq Tmax$. Similarly for the sender v and the receiver u.
- Edge set E: For $1 \leq t \leq Tmax -1$, it is connected w(t) with w(t+1) with a directed edge of capacity Sw(t). It is repeated the same operation for the sender v and the receiver u. Also, for $1 \leq t \leq Tmax$, it is connected w(t) with w'(t), w,w' ϵ W with a directed edge of capacity Nww'(t) and similarly for the sender and the receiver.
- Single source and sink: The source is set to be the sender virtual node v(1) and the sink to be the receiver virtual node u(Tmax).

An optimal transmission schedule is obtained by performing a binary search to find the minimum Tmax for which the maximum flow from v(1) to u(Tmax) equals the volume to be transferred F. The mapping from the maximum flow solution to a transmission schedule is as follows: if the max-flow solution involves flow f crossing the edge w(t)→w'(t) then an optimal schedule should transmit volume f from w to w' during time slot t.

(Case 2) Edge and network bottlenecks

In this case it is taken into account the edge node uplink and downlink bottlenecks in the MTT problem. This requires splitting each virtual node w(t) into three parts, as shown in the bottom of Fig.3: the front part w(t)- is used for modelling the downlink bottleneck Dw(t), the middle part w(t)* models the storage capacity Sw(t), and the back part w(t)+ models the uplink bottleneck. The sender node has only a "+" part and the receiver has only a "–" part. The complete reduction is as follows:

- Node set V: For each storage node w ϵ W in the definition of the MTT problem, for $1 \leq t \leq Tmax$ add to V virtual nodes w(t), w(t)* and w(t)+. Similarly for the sender v and the receiver u.

- Edge set E: For $1 \leq t \leq Tmax - 1$ connect w(t)* and w(t+1)* with a directed edge of capacity Sw(t). Repeat the same for the sender v and the receiver u. For $1 \leq t \leq Tmax$, connect w(t)- with w(t)* and w(t)* with w(t)+ with a directed edge of capacity Dw(t) and Uw(t), respectively. Also, connect v(t)* with v(t)+ and u(t)- with u(t)* with a directed edge of capacity Uw(t) and Dw(t), respectively. In addition, for $1 \leq t \leq Tmax$ connect we connect w(t) with w'(t), $w,w' \in W$ with a directed edge of capacity Nww'(t) and similarly for the sender and receiver.
- Single source and sink: The source is set to be the sender virtual node v(1)* and the sink to be the receiver virtual node u(Tmax)*.

As before, we obtain the optimal MTT transmission schedule by finding the max-flow of the smallest Tmax that equals the volume to be transferred F.

Example of Imperfect Prediction

It has so far assumed perfect a priori knowledge of bandwidth bottlenecks. The system of the invention is designed to tap on the predictability of periodic patterns at large time scales, but is also equipped with the ability to adapt gracefully to estimation error end churn. Next it is described how the system attains its adaptability.

The core concept is to periodically re-compute the transmission schedule based on revised bottleneck predictions provided by the prediction module (2). The first computation of the transmission schedule is described above. However, for the subsequent computations, apart from the updated bottlenecks, it is needed to take into account that the sender may have already delivered some part of the file to intermediate storage nodes and the final receiver. It is captured this by augmenting our basic time-expansion: it is assigned a new demand at the source $Fv \leq F$ and assigning a new demand Fw at each SnF relay $w \in W$. Fw is equal to the volume of file data w currently stored. This reduces the MTT problem into a multiple source maximum flow problem. The details are as follows:

- Node and Edge sets: The node and edge sets between the sender, the receiver and the intermediate SnF relay nodes are obtained as it is described before.
- Multiple sources and single sink: It is represented the volume that has yet to be transferred to the receiver as a flow from multiple sources to the sink node u(Tmax)*.

Then, it is reduced the multiple source max-flow problem into a single source max-flow as follows [9]. It is created a virtual super-source node S. It is connected S to the sender virtual node v(1) with a directed edge of capacity equal to the demand Fv, which is equal to the volume of the file the sender has not yet transmitted to any storage node or the receiver. It is also connected S to each storage virtual node w(1) with a directed edge of capacity Fw on equal to the volume of file data it currently stores.

An optimal transmission schedule is obtained by finding the minimum Tmax for which the total flow from the super-source S to u(Tmax)* over the flow network G(V,E) equals the remaining undelivered volume:

Fv +

The mapping from the resulting flow into a transmission schedule is done as before. The reduction hinges on the important observation that all above demands carry distinct parts of the file, thus are equally important.

Example of the End Game Mode

The above design adapts to prediction error by periodically revising the transmission schedule. In extreme situations however, even periodic revision does not suffice. For example, a node can go off-line, or its uplink can drop unexpectedly to a very small value and thus hold back the entire transfer due to parts that get stuck at the node. Since it is considered this to be an extreme case rather than the norm, it is addressed it through a simple End-Game Mode (EGM) [10] approach as follows. Due to the low cost of storage it is easy to keep at some storage nodes "inactive" replicas of already forwarded pieces. Then, if the piece is deemed delayed by the time that the EGM kicks in, a replica can switch to "active" and be pulled directly by the receiver. Similar to "BitTorrent", it is further subdivided a piece in sub-pieces. EGM kicks in after c% (e.g., 95%) of the file has been delivered and pulls sub-pieces of the delayed piece simultaneously from the source and the storage nodes that store active replicas.

The system of the invention is arranged for carrying bulk data with the following characteristics: a) it uses store and forward (SnF) relays; b) it carries data over multiple

paths of intermediate SnF relays; c) it carries data over multiple hops of SnF relays; and d) it considers known or predicted capacities of SnF relays to forward certain data volumes at certain times in the future.

The advantage of store-and-forward besides on the fact that E2E connections over the native IP path or alternative paths provided by overlay routing cannot avoid crossing the first and the last hop of a transfer and thus are unable to solve MTABs that involve these two edge links. The requirement in these cases is to "time shift" the bandwidth of the sender during times that the receiver is blocked, saving it from being wasted. This is exactly what SnF is doing by feeding intermediate storage nodes during times that the receiver is blocked and cannot get data fast from its direct connection to the sender. From a bandwidth perspective and denoting Uv(t) the uplink speed of the sender and Du(t) the downlink speed of the receiver, we observe that by starting at t0 and continuing for T time units, SnF can transfer a volume up to

min(,)

On the other hand, any end-to-end policy is more tightly constrained by

The advantage of multi-path besides on the fact that just like the sender and the receiver, the storage nodes themselves have bandwidth and storage constraints. Thus, it is possible that a single storage node cannot fully exploit the access links of the sender and the receiver. This can easily be solved by using multiple storage nodes which translates into a need to support multipath transfers. In addition, multi-path transfers make it easier to work around network bottlenecks.

The advantage of multi-hop besides on the fact that a storage node can help alleviate MTABs as long as it has partial time overlap with both the sender and the receiver. That is, if it can download data fast from the sender but also upload fast at some later time to the receiver. However, there are cases that this is impossible.

In Fig.3, the sender v can upload data fast to storage node w1 which, however, cannot forward them to the receiver u at any point of time in the future. The problem can be

solved by involving a second storage node, w2 that receives the data from w1 and forwards it to u. This illustrates the advantage of multi-hop transfers.

The advantage of predicting and using information about the future besides on the fact that if storage (SnF relay) nodes are bottleneck free, the optimal store and forward policy is quite simple: the sender must use any leftover uplink bandwidth that cannot be exploited by a direct end-to-end path to the receiver, to upload as fast as possible additional data to the storage node, which in turn will drain as fast as possible to the receiver. However, as mentioned before, storage nodes have bottlenecks, thus one might need to use multiple of them. This immediately raises the question of how to route and schedule transmissions among end points and storage nodes. A fundamental design choice of the system of the invention is to schedule transmissions based on estimated of the capacity of each SnF relay node to move large volumes of data over extended periods of time.

The above design choice introduces complications that are not taken into account by existing applications dealing with bulk data, e.g., in the BitTorrent P2P system. In addition, P2P is of multicast nature, which implies that any successful piece delivery is guaranteed to be useful for at least the node receiving it, and potentially for other nodes that can fetch it from there. In the system of the invention however, a transfer is of value only if it eventually helps in delivering the chunk to the unique final receiver. Predicting the future capacity of SnF relay nodes is advantageous because it prevents waste of bandwidth and storage in transfers that yields no benefit.

Hence, it is essential for the system of the invention to know or be able to roughly estimate, the ability of a node to move during an extended period of time in the future some large volume of data. This might seem cumbersome. However, in many the future availability of bandwidth is a priori known, e.g., performing backups within or across datacenters over dedicated local or wide-area networks. Or the case of ISP's that sell residential broadband access with a priori advertised caps over certain periods of the day. Or in the case of terabyte transfers, where due to the large aggregation of independent flows on the background, the prediction of leftover diurnal bandwidth is almost perfect [10]. Even if a priori knowledge is not explicitly offered, as long as exist some strong periodic pattern, it is easy to detect and adapt to it. The key observation is

that due to its delay tolerant bulk nature, the system of the invention is only sensitive to aggregate volumes over extended periods of time (hours). It is not sensitive to bandwidth fluctuations at small (msec/sec) scales which are hard to measure and predict. The system of the invention, only needs to know about dominant events, such as throttling during ISP peak hours or the trigger of an end-user-defined timed bandwidth cap. Such events are easy to detect and communicate to nodes upon bootstrap. Each node can refine its predictions and adapt to errors during its participation in the system.

References

[1] J. Gray. Distributed computing economics. TechReportMSR-TR-2003-24.

[2] N. Laoutaris, G. Smaragdakis, P. Rodriguez, and R. Sundaram. Delay Tolerant Bulk Data Transfers on the Internet. In Proc. of ACM SIGMETRICS'09.

[3] N. Laoutaris and P. Rodriguez. Good things come to those who (can) waitor how to handle delay tolerant traffic and make peace on the Internet. In Proc. of ACM HotNets-VII, 2008.

[4] P-0800076. Moving Several Terabytes Everyday for Free.

[5] Anukool Lakhina, Konstantina Papagiannaki, Mark Crovella, Christophe Diot, Eric D. Kolaczyk, and Nina Taft. Structural analysis of network traffic flows. In Proc. of ACM SIGMETRICS / Performance 2004.

[6] G. Siganos, M. Iliofotou, X. Yang, J. Pujol, and P. Rodriguez. Apollo: Tapping into the Bittorrent Ecosystem. Telefonica Technical Report, 2009.

[7] S. Shalunov and B. Teitelbaum. Qbone Scavenger Service (QBSS) Definition. Internet2 technical
report, March 2001.

[8] N. Laoutaris, X. Yang, M. Sirivianos, M. Sanchez, and P. Rodriguez. Hermes: A Software Substrate for Bulk Transfers. Telefonica technical report, 2009.

http://research.tid.es/nikos/images/hermes_tech.pdf.

[9] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein. Introduction to Algorithms, 2nd edition. MIT Press, Cambridge, Massachusetts, 2001.

[10] B. Cohen. Incentives Build Robustness in BitTorrent. In P2P Econ, 2003

[11] A. Venkataramani, R. Kokku, M. Dahlin, TCP Nice: A Mechanism for Background Transfers. In OSDI, 2002.

[12] S. Mitchell and J. Roy. Pulp Python LP Modeler. http://code.google.com/p/pulp-or/

[13]Twisted Python library. http://twistedmatrix.com/trac/

[14] N. Laoutaris and P. Rodriguez. Delay Tolerant Bulk InternetTransfers. In Ercim News. April 2009.

[15] Algorithms for Constrained Bulk-Transfer of Delay Tolerant Data. In ICC 2009

[16] The Local and Global Effects of Traffic Shaping. Massimiliano Marcon, Marcel Dischinger, Krishna Gummadi, Amin Vahdat. MPI Technical Report. MPI-SWS-2009-003 (June 2009).

[17] Packet-switched Data Communications System. Crager et al. US 4058672

[18] Method and System for Content Oriented Routing. Mizutani Masahiko, Chiyoda-Ku. EP-A1-1411678.

CLAIMS

1.- A multi-hop and multi-path store and forward system for bulk transfers comprising a plurality of nodes (100) wherein each node (100) also comprises:

an overlay management module (1) that is arranged to add a new node (100) to the overlay, removing it, and maintaining the overlay connections during the node's participation in the system;

a volume prediction module (2) which is arranged to maintain a time series with the predicted maximum volume of data that can be forward to each neighbour node during each one of the slots that make an entire day; and bootstrapping, ISP-friendliness, and security means; characterized in that it comprises a scheduling and routing module (3) for all data transfers between a sender v, storage nodes w $\in$ W and a receiver node u; and wherein the module (3) also comprises volume prediction means to calculate an initial transfer plan and keeps updating it periodically as it receives updated predictions from the nodes, and means for solving a maximum network flow optimization problem; and wherein a transmission management module (4) is arranged to receive, scheduling and routing commands from the scheduling and routing module (3) of senders v for which a local node is forwarding and executes them accordingly.

2.- A multi-hop and multi-path store and forward method for bulk transfers, comprising the steps of:

managing the overlay adding a new node (100) to the overlay, removing it, and maintaining the overlay connections during the node's participation;

maintaining a time series with the predicted maximum volume of data that can be forward to each neighbour node during each one of the slots that make an entire day; characterized in that it comprises:

- a scheduling and routing step for all data transfers between the sender v, the storage nodes w $\in$ W and the receiver node u;

said scheduling and routing step including a volume prediction step to calculate an initial transfer plan and keeping updating it periodically as it receives updated predictions from the storage nodes w $\in$ W and providing a maximum network flow optimization; and

- a transmission management step wherein it is received the scheduling and routing commands of the senders for which the local node is forwarding and executing them accordingly.

3- A multi-hop and multi-path store and forward method according to claim 2, wherein said volume prediction step is set over at least several hours in the future from a given starting sending time.

4.- A computer program product comprising computer-executable instructions embodied in a computer-readable medium for performing said volume prediction steps of the method of claim 2.

; said step including a volume prediction step to calculate an initial transfer plan and keeping updating it periodically as it receives updated predictions from the nodes; problem step; and

(iv) a transmission management step wherein it is received the scheduling and routing commands of the senders for which the local node is forwarding and executing them accordingly.

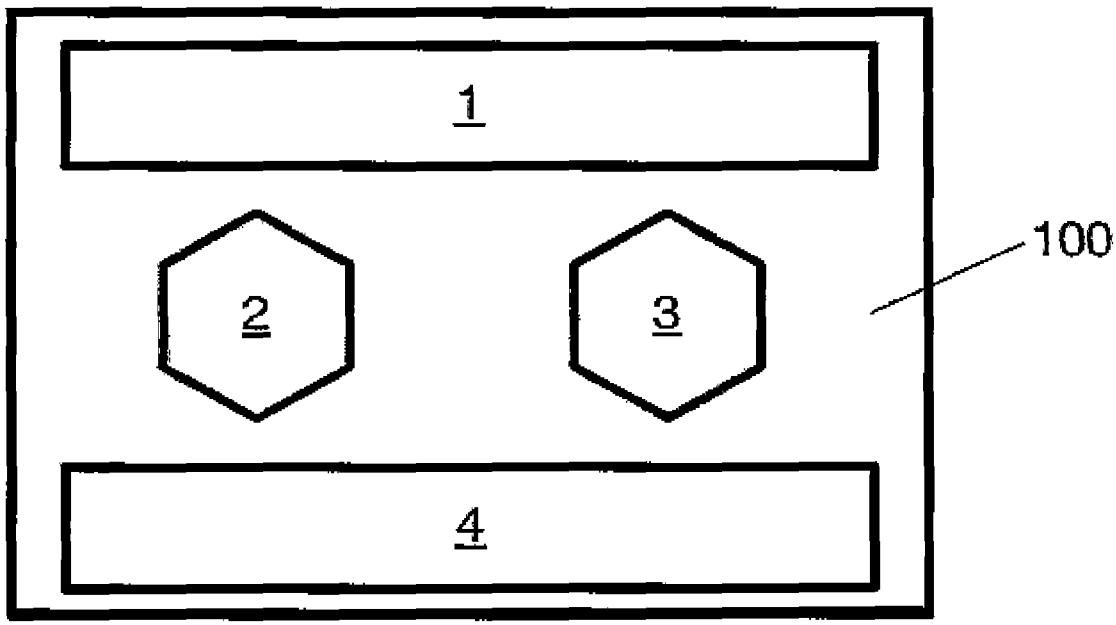

FIG. 1

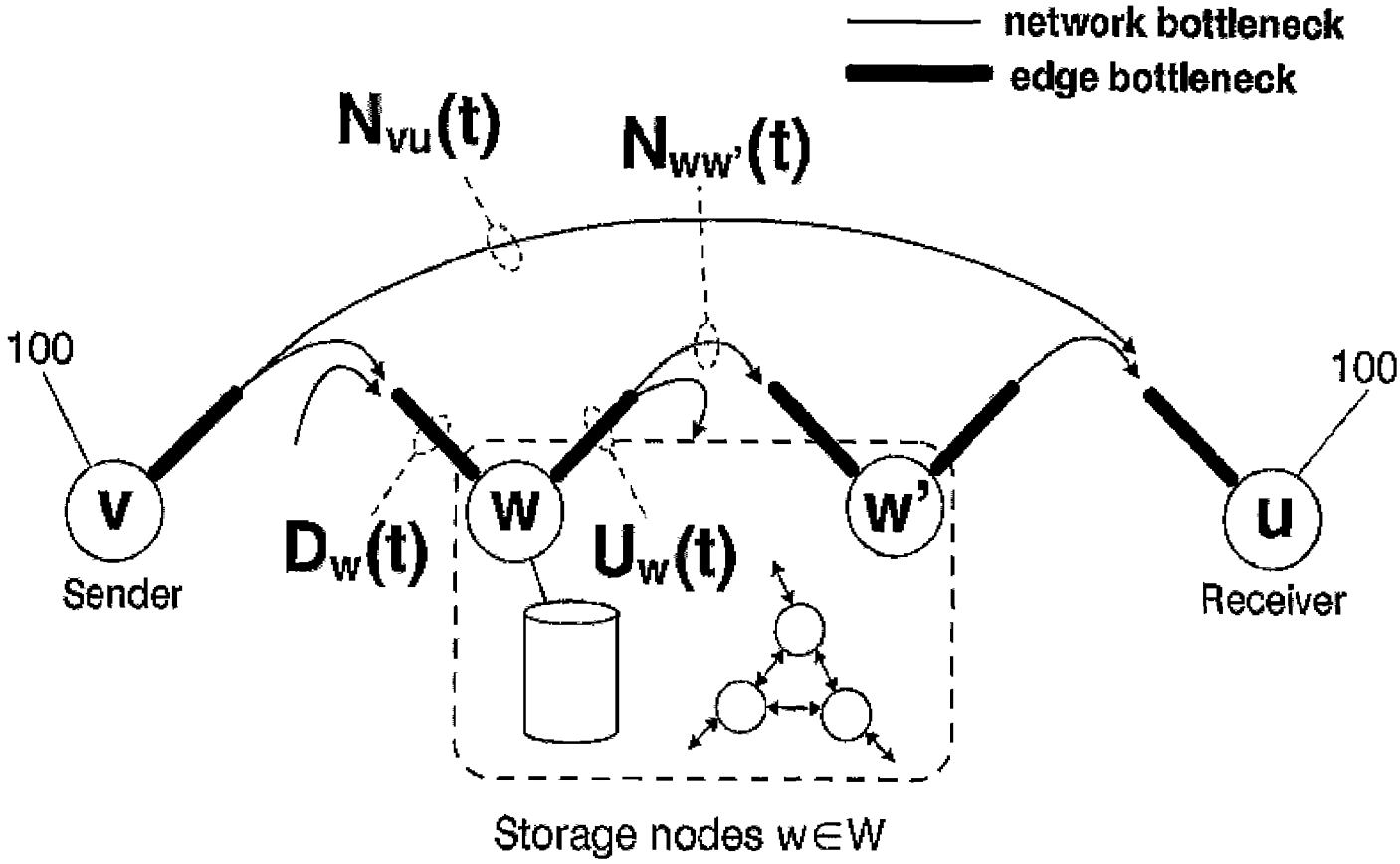
network bottleneck
edge bottleneck
$N_{vu}(t)$
$N_{ww'}(t)$
100
100
v
Sender
$D_w(t)$
w
$U_w(t)$
w'
u
Receiver
Storage nodes w∈W

FIG. 2

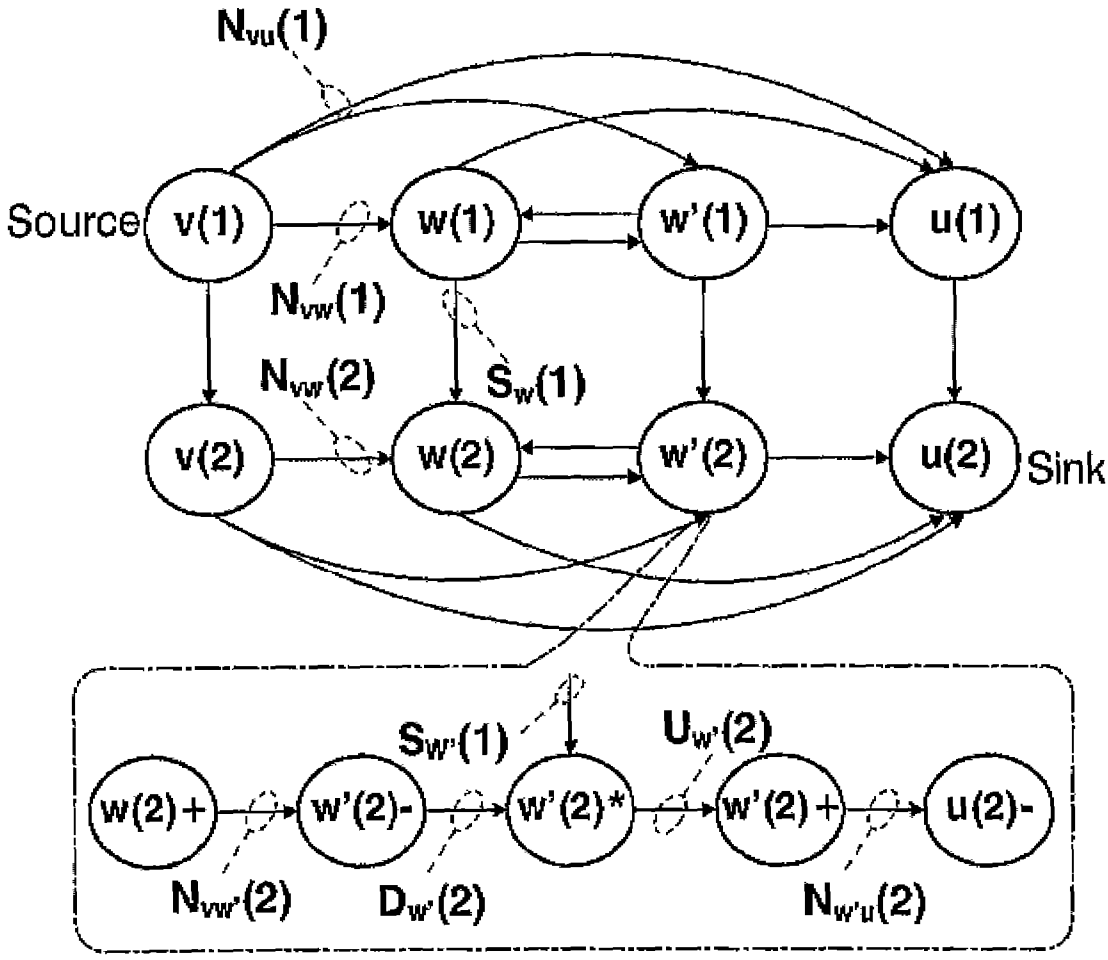
$N_{vu}(1)$
Source
v(1)
w(1)
w'(1)
u(1)
$N_{vw}(1)$
$N_{vw}(2)$
$S_w(1)$
v(2)
w(2)
w'(2)
u(2)
Sink
$S_{w'}(1)$
$U_{w'}(2)$
w(2)+
w'(2)-
w'(2)*
w'(2)+
u(2)-
$N_{vw'}(2)$
$D_{w'}(2)$
$N_{w'u}(2)$

FIG. 3

INTERNATIONAL SEARCH REPORT

International application No
PCT/EP2011/001412

A. CLASSIFICATION OF SUBJECT MATTER
INV. H04L29/06
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

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 practical, 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.
X	US 2005/015511 A1 (IZMAILOV RAUF [US] ET AL) 20 January 2005 (2005-01-20) paragraphs [0018], [0034] - [0043], [0069], [0080]	1-4
X	NIKOLAOS LAOUTARIS: "Bulk data transfers on the Internet or how to book some terabytes on red-eye bandwidth", INFORMATION THEORY WORKSHOP (ITW), 2010 IEEE, IEEE, PISCATAWAY, NJ, USA, 6 January 2010 (2010-01-06), page 1, XP031703946, ISBN: 978-1-4244-6372-5 the whole document	1-4

-/--

[X] Further documents are listed in the continuation of Box C.

[X] 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

"E" earlier document but published on or after the international filing date

"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)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"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

"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

"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.

"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
4 July 2011	11/07/2011

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

INTERNATIONAL SEARCH REPORT

International application No
PCT/EP2011/001412

C(Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	SMARAGDAKIS G ET AL: "Swarming on Optimized Graphs for n-Way Broadcast", INFOCOM 2008. THE 27TH CONFERENCE ON COMPUTER COMMUNICATIONS. IEEE, IEEE, PISCATAWAY, NJ, USA, 13 April 2008 (2008-04-13), pages 141-145, XP031263787, ISBN: 978-1-4244-2025-4 the whole document	1-4
A	ACM, 2 PENN PLAZA, SUITE 701 - NEW YORK USA, 6 November 2008 (2008-11-06), XP040466665, the whole document	1-4
A	US 2009/122697 A1 (MADHYASHA HARSHA V [US] ET AL MADHYASTHA HARSHA V [US] ET AL) 14 May 2009 (2009-05-14) paragraphs [0002], [0005], [0006], [0019] - [0023]	1-4
A	CHHABRA P ET AL: "Algorithms for Constrained Bulk-Transfer of Delay-Tolerant Data", COMMUNICATIONS (ICC), 2010 IEEE INTERNATIONAL CONFERENCE ON, IEEE, PISCATAWAY, NJ, USA, 23 May 2010 (2010-05-23), pages 1-5, XP031703113, ISBN: 978-1-4244-6402-9 cited in the application the whole document	1-4
A	US 4 058 672 A (CRAGER WILLIAM C ET AL) 15 November 1977 (1977-11-15) cited in the application abstract column 4, line 66 - column 5, line 45	1-4

INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No
PCT/EP2011/001412

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 2005015511 A1	20-01-2005	NONE	
US 2009122697 A1	14-05-2009	NONE	
US 4058672 A	15-11-1977	ES 463987 A1	01-07-1978
		SE 7712572 A	10-05-1978