

(19) World Intellectual Property Organization
International Bureau



(43) International Publication Date
8 November 2007 (08.11.2007)

PCT

(10) International Publication Number
WO 2007/127332 A2

(51) International Patent Classification:
G01R 31/08 (2006.01)

(21) International Application Number:
PCT/US2007/010176

(22) International Filing Date: 26 April 2007 (26.04.2007)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
60/745,736 26 April 2006 (26.04.2006) US
11/740,918 26 April 2007 (26.04.2007) US

(71) Applicant (for all designated States except US): **BITTOR-RENT, INC.** [US/US]; 201 Mission Street, Suite 900, San Francisco, CA 94105 (US).

(72) Inventors; and

(75) Inventors/Applicants (for US only): **COHEN, Bram** [US/US]; 201 Mission Street, Suite 900, San Francisco, CA 94105 (US). **HARRISON, David** [US/US]; 201 Mission Street, Suite 900, San Francisco, CA 94105 (US). **HAZEL, Gregory** [US/US]; 201 Mission Street, Suite 900, San Francisco, CA 94105 (US).

(74) Agents: **HAYES, David, L.** et al.; Fenwick & West LLP, Silicon Valley Center, 801 California Street, Mountain View, CA 94041 (US).

(81) Designated States (unless otherwise indicated, for every kind of national protection available): AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BH, BR, BW, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, 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, PG, PH, PL, PT, RO, RS, RU, SC, SD, SE, SG, SK, SL, SM, SV, SY, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.

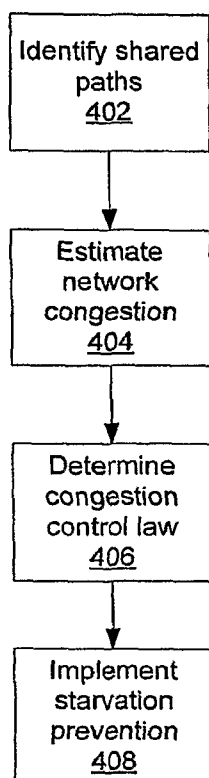
(84) Designated States (unless otherwise indicated, for every kind of regional protection available): ARIPO (BW, GH, GM, KE, LS, MW, MZ, NA, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European (AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HU, IE, IS, IT, LT, LU, LV, MC, MT, NL, PL, PT, RO, SE, SI, SK, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

Published:

— without international search report and to be republished upon receipt of that report

[Continued on next page]

(54) Title: END-SYSTEM DYNAMIC RATE LIMITING OF BACKGROUND TRAFFIC



(57) Abstract: Dynamic rate limiting of background traffic to alleviate congestion in the access network is enabled. ICMP echo round-trip times and ICMP losses to a nearby node outside the local area and just beyond the divergence in end-to-end paths are measured, allowing unambiguous discrimination of nearby from distant congestion points. Using round-trip time samples, either short-run delay or short-run variance in delay can be measured to estimate congestion. When combined with an appropriate control law, background traffic can be rapidly reduced to allow interactive traffic to traverse unhindered through the access network. The described system and methods can be implemented in the application-layer and without any additional support from the network.



For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

END-SYSTEM DYNAMIC RATE LIMITING OF BACKGROUND TRAFFIC

5

CROSS-REFERENCE TO RELATED APPLICATIONS

[0001] This application claims the benefit of provisional application 60/745,736, filed on
10 April 26, 2006, and which is incorporated by reference herein in its entirety.

BACKGROUND OF THE INVENTION**Field of the Invention**

[0002] The present invention relates generally to peer-to-peer networking
environments. In particular, the present invention is directed towards a system and
15 method for adaptively rate limiting traffic in a peer-to-peer network based on network
congestion.

Description of Background Art

[0003] A number of applications exist for exchanging files in a peer-to-peer
environment. These peer-to-peer file sharing applications suffer the deserved reputation of
20 bullying other traffic on the network. Due to difficult-to-remedy limitations in Internet
congestion-control algorithms, large file downloads tend to build up a backlog inside the
network. Backlogs increase delays that are most often noticed by users running interactive
applications such as web browsers. The conventional approach to solving this problem has
been to rate-limit peer-to-peer traffic at all times, even when no web traffic is present. This
25 leads to an inefficient result, since if no competing traffic such as that from a web browser is
present, artificially limiting the rate at which the peer-to-peer application can operate serves
no purpose.

[0004] In an alternative strategy, peer-to-peer traffic is treated as background traffic.
Background traffic defers to higher-priority traffic, but otherwise consumes excess capacity.

30

SUMMARY OF THE INVENTION

[0005] The present invention includes methods for end-systems in a peer-to-peer network to dynamically rate limit background traffic to alleviate congestion in the access network. This differs from the traditional end-to-end congestion control problem as addressed by TCP in at least three ways: 1) end-to-end congestion control measures congestion across all bottlenecks in the path even when a typical user is more motivated to protect nearby bottlenecks, e.g., his own access point; 2) end-to-end congestion control schemes typically treat all traffic equally pushing the duty of service differentiation to the underlying network; and 3) end-to-end congestion control typically controls only a single flow as opposed to the aggregate of flows sharing a bottleneck.

[0006] The present invention measures ICMP echo round-trip times and ICMP losses to a nearby node outside the local area and just beyond the divergence in end-to-end paths allowing unambiguous discrimination of nearby from distant congestion points. Using round-trip time samples, either short-run delay or short-run variance in delay can be measured to estimate congestion. When combined with an appropriate control law, background traffic can be rapidly reduced to allow interactive traffic to traverse unhindered through the access network. The present invention can be implemented in the application-layer and without any additional support from the network.

BRIEF DESCRIPTION OF THE DRAWINGS

[0007] Fig. 1 is an illustration of a peer-to-peer networking environment.

[0008] Fig. 2 is a block diagram of a system for providing adaptive rate-limiting in a peer-to-peer network in accordance with an embodiment of the present invention.

[0009] Fig. 3 illustrates the determination of a shared path in accordance with an embodiment of the present invention.

[0010] Fig. 4 is a flowchart illustrating a method for providing adaptive rate-limiting in a peer-to-peer network in accordance with an embodiment of the present invention.

[0011] The figures depict preferred embodiments of the present invention for purposes of illustration only. One skilled in the art will readily recognize from the following discussion that alternative embodiments of the structures and methods illustrated herein may be employed without departing from the principles of the invention described herein.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENTS

[0012] Fig. 1 illustrates a context for the present invention. A network 116 such as the Internet connects remote peers 110, 112, 114 with a group of peers 102, 104, 106 on a local area network (LAN) 118. For example, peers 102, 104, 106 may be three computers in a single household, or three computers on a college campus, or three computers at a commercial location. Further, while three computers in the LAN and three remote peers are shown in Fig. 1, this is simply for purposes of illustration – an arbitrary number of peers may be involved in the peer-to-peer networking environment to which the present invention has application. Peers 102, 104, 106 communicate with network 116 via access router 108—for example, network traffic between peer 102 and peer 110 would travel through access router 108 (in addition to making several additional hops as is known in the art).

[0013] A system of the present invention in one embodiment is executed by a peer involved in peer-to-peer file sharing, as part of a file-sharing application. Fig. 2 illustrates functional components of a system 200 for providing dynamic rate limiting in accordance with an embodiment of the present invention. System 200 includes a shared path identification module 202 for detecting the location of an access router 108 is likely to be the location of a packet throughput bottleneck; a congestion estimation module 204 for estimating the level of congestion in the network by observing traffic or probing; a congestion control module 206 for applying a congestion control law to adaptively control a rate limit to which the peer-to-peer traffic is subject, thereby governing how much traffic is allowed to enter the network based on the congestion estimate; and a starvation prevention module 208 for ensuring on a longer timescale that an appropriate balance exists between peer-to-peer and other types of traffic over the network by setting a lower bound on the rate limit imposed on peer-to-peer (background) traffic and thus avoiding starvation.

[0014] Each of the modules of system 200 is described further below.

Shared Path Identification

[0015] Some conventional approaches to shared path identification operate by identifying the location of access router 108 by looking for the last common node in the paths to peers. The routes are obtained by tracerouting to each new peer and updating the shared path. For example, referring to Fig. 3, a path from peer 102 to peer 302 travels via

routers 306, 308 and 310. A path from peer 102 to peer 304 travels via routers 306, 308, and 312. Accordingly, the shared path includes routers 306 and 308, and the access router would be predicted to be router 308.

[0016] Shared path identification module 202, by contrast, in one embodiment ignores all connections to nodes within the same network prefix when determining the last common node.

Congestion Estimation

[0017] Once the location in the network of access router 108 has been determined, system 200 next estimates the level of congestion in the network.

[0018] Since system 200 is detecting congestion in the nearby network, it exploits two properties of such networks to improve congestion control: with high likelihood there is only one bottleneck (usually access point 108) and thus this single bottleneck can be well-characterized according to buffer size and capacity.

[0019] In one embodiment, system 200 can use two congestion estimators. A first method, auto-threshold ping (ATP), measures congestion based on smoothed ping round-trip time, setting delay thresholds that require less sensitive input parameters than conventional methods. A second method, variance ping (VP), eschews using smoothed round trips in favor of reacting based on variance in round-trip times.

Auto-Threshold-Pinging

[0020] Congestion estimation module 204 begins by obtaining smoothed ping round trip times. In one embodiment, the minimum round-trip time seen so far, known as `base_rtt`, is subtracted from the smoothed ping times to obtain an unbiased estimate of queuing delay. Alternatively, because propagation delay is likely to be miniscule compared to queuing delays, subtracting the `base_rtt` may have negligible effect and therefore may be skipped.

[0021] To smooth round trip estimates, congestion estimation module 204 may use exponentially weighted moving averaging (EWMA); mean over a moving window; or median over a moving window. All three techniques require one parameter: the weight for EWMA or moving window size. For all three smoothing mechanisms, performance remains good across a wide range of scenarios without modifying parameter settings, EWMA of 0.1 (smaller is slower convention) or window size of 10 samples. Prior work in

congestion control has largely avoided using moving windows because of the additional state and computations involved. However, since an aggregate of all peer-to-peer connections is being controlled, these additional computations are likely to be miniscule compared to the overhead already present in the underlying TCP layer.

5 [0022] A single delay threshold is used in one embodiment to signal congestion. Congestion estimation module 204 stores the k -largest round trip times and uses the median of these measurements to estimate the delay that occurs when the bottleneck buffer is full or near full. This estimate, called a delay-on-full estimation, is denoted `delay_on_full`. In one embodiment, k can be 1, in which case the delay-on-full estimation is equivalent to
 10 using the maximum round trip time (RTT) seen so far. Median is used in one embodiment because it is less affected by outliers. When an ICMP echo loss occurs, the largest sample is dropped. Thus the delay-on-full estimate will eventually recover if it becomes far off due to spurious noise.

[0023] Once the delay-on-full estimate is made, the delay threshold (`max_thresh`)
 15 is set in one embodiment as follows:

`max_thresh = thresh_factor * delay_on_full`

[0024] By setting `max_thresh` dynamically, system 200's throughput sensitivity is reduced across scenarios with different bottleneck sizes. However, larger bottleneck buffers will result in larger delays. This dynamic setting also eliminates errors found in
 20 conventional methods when the threshold is set so large that congestion is never detected, and it reduces the rate of false positives whenever there is a reasonably provisioned bottleneck buffer.

Variance Pinging

[0025] Auto-threshold pinging by itself does not explicitly take into account delay
 25 variance. Because observed round-trip time variance is high, system 200 exploits the high variance as a measure of congestion.

[0026] Queuing delay exhibits high variance, but not in the case of low or very high utilization. When the access network has low utilization, a queue is not given a chance to build. When the access network has high utilization, the buffer is not given a chance to
 30 drain. System 200 adjusts the send rate to keep the system near the point of maximum variance.

[0027] Variance `var` is measured across a window of the last `max_samples` where `max_samples` is in one embodiment set to 10. Whenever a ping arrives, the following is done:

```

    var = measure over window
5    if var > max_var then max_var = var
    if var > var_factor * max_var:
        network is congested

```

[0028] `Max_var` will tend to rise over time with noise and as a result there is concern that it might drift so high that the access network never becomes congested. However, when this occurs, the buffer will begin to overflow resulting in ping loss. When a ping is lost, congestion estimation module 204 reduces `max_var` by `reduce_factor`. In one embodiment, `reduce_factor` is set to 0.8.

[0029] Note that variance reduces when the bottleneck becomes near full. The described algorithm increases the send rate whenever variance is below `var_factor * max_var` under the assumption that variance is in the regime where it increases with send rate. As a result, the rate limit increases until the buffer overflows and pings begin to be lost. Thus in one embodiment system 200 multiplicatively backs off the rate limit whenever ping loss occurs. This multiplicative back-off is steeper than the back-off described below in order to ensure that the buffer is given a chance to drain.

20 Congestion Control Law

[0030] Congestion control module 206 in one embodiment uses Additive Increase with Multiplicative De-crease (AIMD) as a control law as follows:

```

    if network is congested:
        rlim *= beta
25    else if upspeed within
        epsilon of rlim:
        rlim += delta

```

where "rlim" represents the upload rate limit, and upspeed represents.

[0031] The congested state is signaled as described above. In one embodiment, `beta` is set to 0.8 and `delta` to 1 KBps.

[0032] AIMD improves upon conventional controls for a peer-to-peer environment in that it is rate-based.

Starvation Prevention

[0033] A starvation prevention mechanism places bounds on how low congestion control module 206 rate limits a peer's background traffic. An appropriate value for the rate limit is determined by first characterizing the access network's capacity over longer time periods and then setting an appropriate bound. In one embodiment, this is done using capacity fraction starvation prevention; alternatively it is done using long-run throughput fraction starvation prevention.

Capacity Fraction Starvation Prevention

[0034] The benefit a user derives from interactive traffic and from background traffic both exhibit diminishing returns with increasing bandwidth use. More specifically, the utility functions for both interactive and background traffic are continuously differentiable, concave, and increasing. From convex optimization, under these conditions a unique solution will exist. If the utility functions are additionally logarithmic then the optimal point resides at a fraction of capacity.

[0035] Consider utility U , bitrate x allocated to foreground traffic, and bitrate y allocated to background traffic. a and b are constants denoting relative importance of foreground versus background traffic. Let c denote the access capacity:

$$\begin{array}{ll} \text{utility} & U = a \log x + b \log y, \\ \text{maximize} & U \end{array} \quad (1)$$

$$\text{such that} \quad x + y \leq c \text{ and } x, y \geq 0. \quad (2)$$

[0036] Given that utility is an increasing function of band-width, the optimum will reside along the line $x + y = c$. The maximal utility occurs where

$$\frac{dU}{dx} = 0 = \frac{a}{x} - \frac{b}{c-x}, \quad (4)$$

which solves to

$$x = \frac{a}{a+b}c, y = \frac{b}{a+b}c. \quad (5)$$

[0037] Thus, for a choice of utility functions, the optimum minimum background traffic rate limit occurs at a fraction of $\frac{b}{a+b}c$ regardless of the value of c . The capacity fraction starvation prevention building block thus takes as input a fraction cap_frac . The rate limit on background traffic is bounded such that

`rlim >= cap_frac * cap_est`

where `cap_est` is an estimate of access capacity. A number of existing capacity estimators can be used to set `cap_est`. See, for example, Van Jacobson, "Pathcar, a tool to infer characteristics of internet paths," <http://ftp.ee.lbl.gov/pathchar>; A. Downey, "Using pathcar to estimate internet link characteristics," in Proceedings of SIGCOMM '99, Boston, MA, August 1999; and Robert L. Carter and Mark E. Crovella, "Measuring bottleneck link speed in packet switched networks," in Performance Evaluation, 1996, 27-28, pp. 297-318, all of which are incorporated by reference herein.

[0038] If the user desires emulation of high priority queuing, then this is handled as a special case. The user sets `cap_frac` to zero. Multiplicative decrease can get arbitrarily close to zero, unlike conventional methods which are limited by the granularity of the decrease delta.

Long-run throughput fraction starvation prevention

[0039] Long-run throughput fraction starvation prevention is similar to capacity fraction starvation prevention building block, except that the rate limit prevented from falling below a fraction of the long-run aggregate upload rate, `long_avg_uprate`:

`rlim >= cap_frac * long_avg_uprate`

[0040] This does not require a capacity estimator as required by capacity fraction starvation prevention, but using long-run throughput only prevents starvation to the extent that the throughput is averaged over a much longer time span than the timescale used by the congestion estimator building block. A long period of congestion would cause the long-run average to diminish resulting in a slow progression toward starvation.

[0041] Accordingly, and referring to Fig. 4, a method for dynamically rate-limiting background traffic in accordance with an embodiment of the present invention includes identifying 402 shared paths; estimating network congestion 404; determining 406 an appropriate congestion control law; and implementing 408 starvation prevention, all as described above.

[0042] The present invention has been described in particular detail with respect to a limited number of embodiments. Those of skill in the art will appreciate that the invention may additionally be practiced in other embodiments. First, the particular naming of the components, capitalization of terms, the attributes, data structures, or any other

programming or structural aspect is not mandatory or significant, and the mechanisms that implement the invention or its features may have different names, formats, or protocols.

Further, the system may be implemented via a combination of hardware and software, as described, or entirely in hardware elements. Also, the particular division of functionality

5 between the various system components described herein is merely exemplary, and not mandatory; functions performed by a single system component may instead be performed by multiple components, and functions performed by multiple components may instead performed by a single component. For example, the particular functions of the congestion estimation module 204, congestion control module 206, and so forth may be provided in
10 many or one module.

[0043] Some portions of the above description present the feature of the present invention in terms of algorithms and symbolic representations of operations on information. These algorithmic descriptions and representations are the means used by those skilled in the art of peer-to-peer networking to most effectively convey the substance
15 of their work to others skilled in the art. These operations, while described functionally or logically, are understood to be implemented by computer programs. Furthermore, it has also proven convenient at times, to refer to these arrangements of operations as modules or code devices, without loss of generality.

[0044] It should be borne in mind, however, that all of these and similar terms are to be
20 associated with the appropriate physical quantities and are merely convenient labels applied to these quantities. Unless specifically stated otherwise as apparent from the present discussion, it is appreciated that throughout the description, discussions utilizing terms such as "processing" or "computing" or "calculating" or "determining" or "displaying" or the like, refer to the action and processes of a computer system, or similar
25 electronic computing device, that manipulates and transforms data represented as physical (electronic) quantities within the computer system memories or registers or other such information storage, transmission or display devices.

[0045] Certain aspects of the present invention include process steps and instructions described herein in the form of an algorithm. It should be noted that the process steps and
30 instructions of the present invention could be embodied in software, firmware or hardware,

and when embodied in software, could be downloaded to reside on and be operated from different platforms used by real time network operating systems.

[0046] The present invention also relates to an apparatus for performing the operations herein. This apparatus may be specially constructed for the required purposes, or it may
5 comprise a general-purpose computer selectively activated or reconfigured by a computer program stored in the computer. Such a computer program may be stored in a computer readable storage medium, such as, but is not limited to, any type of disk including floppy disks, optical disks, CD-ROMs, magnetic-optical disks, read-only memories (ROMs), random access memories (RAMs), EPROMs, EEPROMs, magnetic or optical cards,
10 application specific integrated circuits (ASICs), or any type of media suitable for storing electronic instructions, and each coupled to a computer system bus. Furthermore, the computers referred to in the specification may include a single processor or may be architectures employing multiple processor designs for increased computing capability.

[0047] The algorithms and displays presented herein are not inherently related to any
15 particular computer or other apparatus. Various general-purpose systems may also be used with programs in accordance with the teachings herein, or it may prove convenient to construct more specialized apparatus to perform the required method steps. The required structure for a variety of these systems will appear from the description above. In addition, the present invention is not described with reference to any particular programming
20 language. It is appreciated that a variety of programming languages may be used to implement the teachings of the present invention as described herein, and any references to specific languages are provided for disclosure of enablement and best mode of the present invention.

[0048] Finally, it should be noted that the language used in the specification has been
25 principally selected for readability and instructional purposes, and may not have been selected to delineate or circumscribe the inventive subject matter. Accordingly, the disclosure of the present invention is intended to be illustrative, but not limiting, of the scope of the invention.

[0049] We claim:

CLAIMS

1. A method for rate-limiting traffic in a peer-to-peer network, the method comprising:

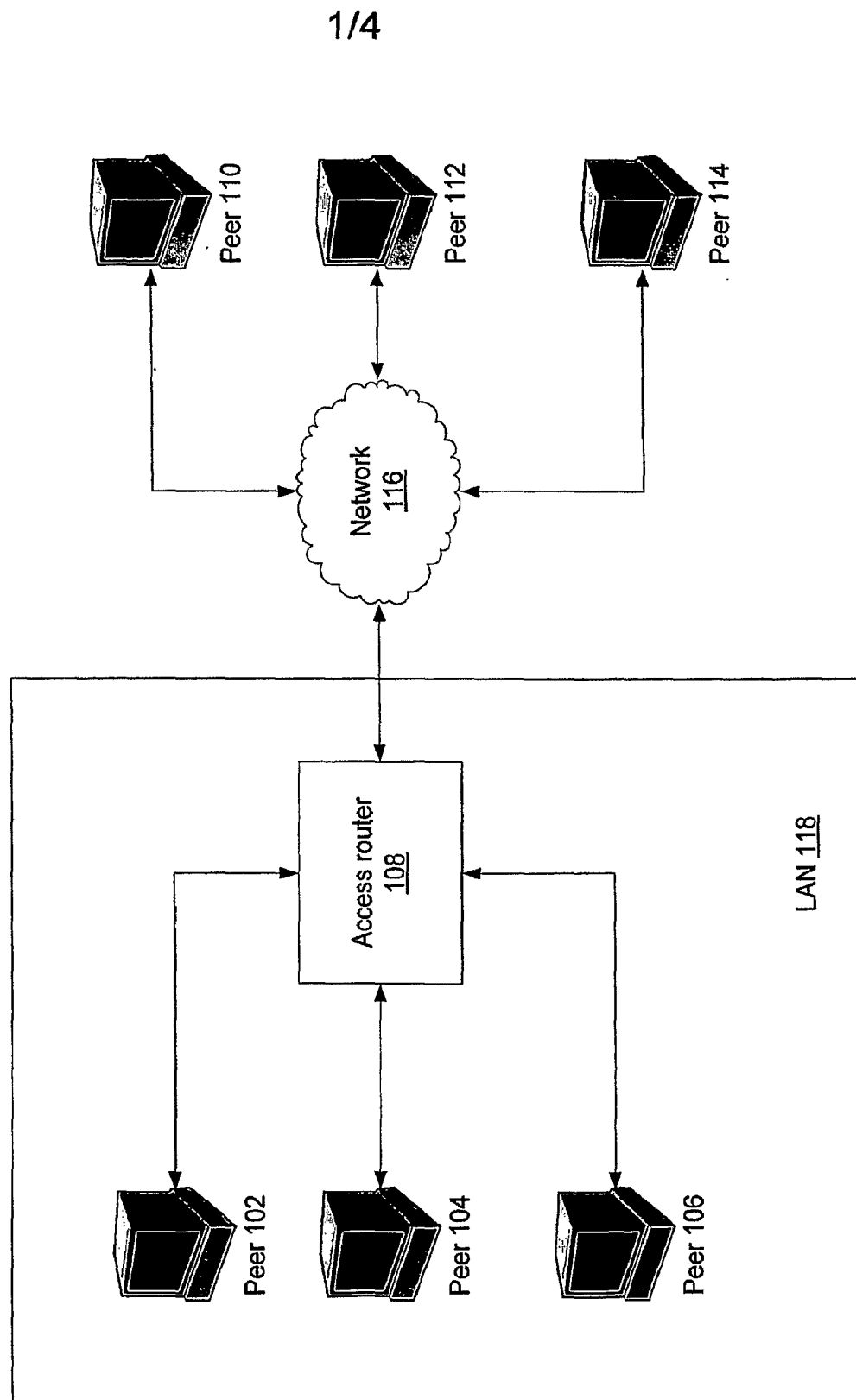
identifying a path in the network shared by a plurality of peers;

5 estimating congestion in the network;

determining a congestion control law associated with the estimated congestion; and

providing a bounding mechanism on the congestion control law.

Fig. 1



2/4

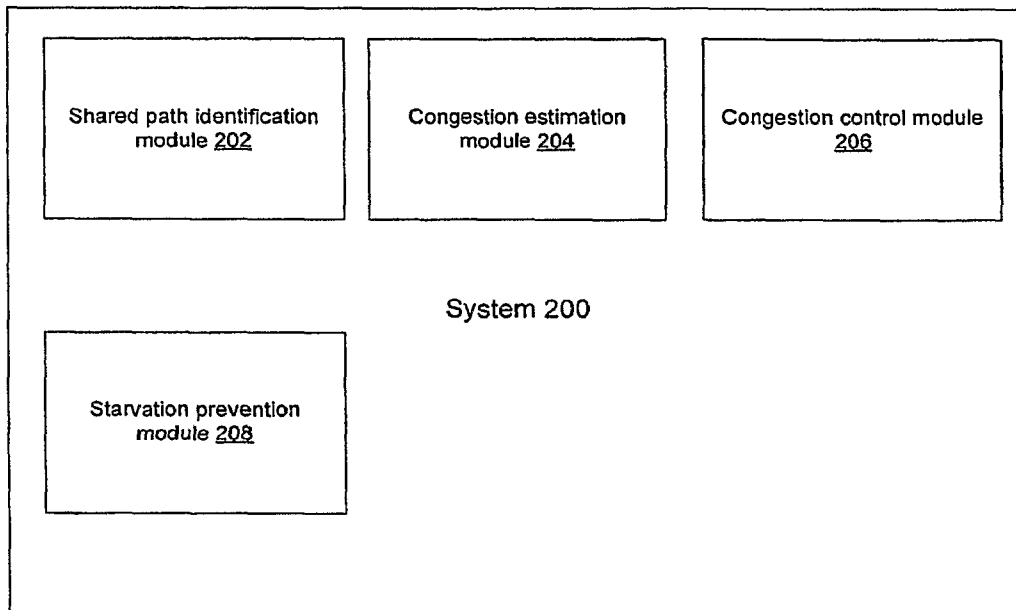


Fig. 2

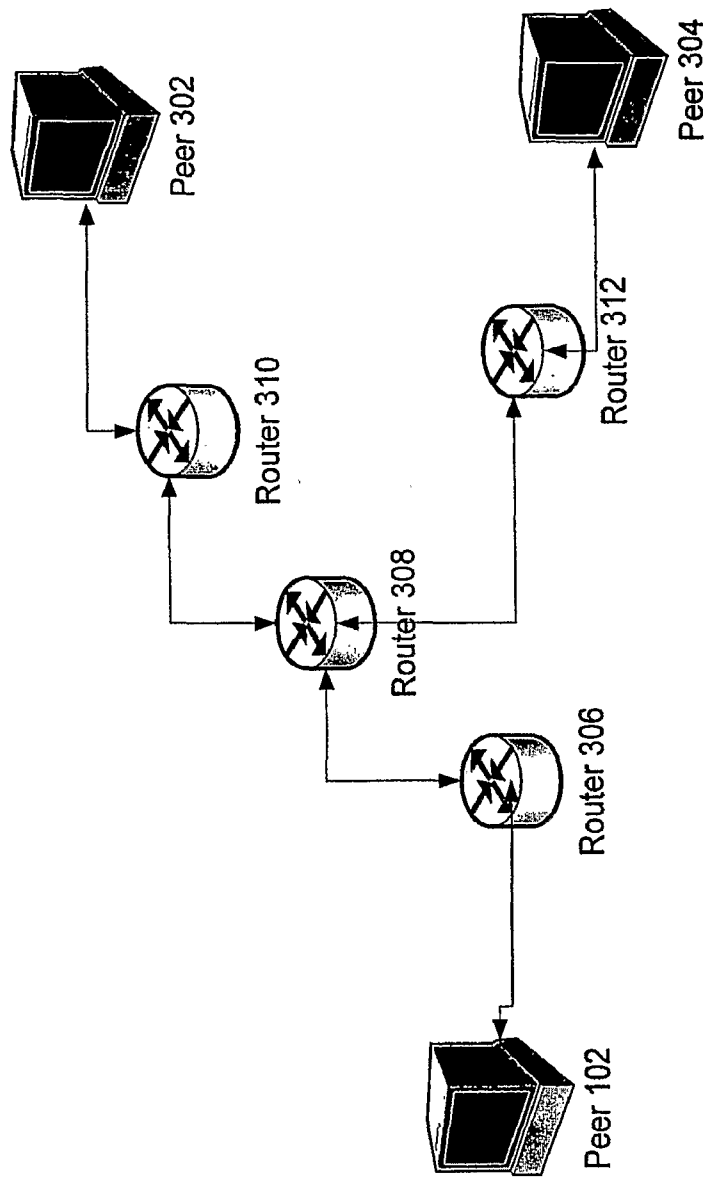
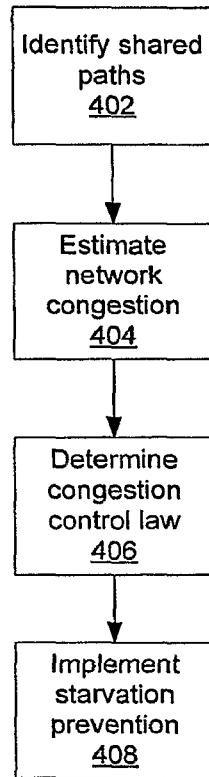


Fig. 3

4/4

**Fig. 4**