Web Caching and Content Delivery
A. Bestavros and M. Rabinovich (Editors)

Comparative Measurements of Internet Traffic Using Cache-Triangle

Serge A. Krashakov [1], Lev N. Shchur [2]

Landau Institute for Theoretical Physics, 142432 Chernogolovka, Russia

Abstract

We discuss the possibility of comparative measurements of Web traffic. This paper is mainly focused on the traffic in the cache-mesh. We analyze the previous attempt based on the author's rewind-and-replay approach in detail. The new approach proposed uses a symmetric setup of cache-servers, a cache-triangle. We discuss the conditions under which such measurements could be done with a given accuracy.

Keywords: Cache hierarchy; Web network; Traffic measurement

1. Introduction

Optimization of the Web traffic is clearly one of the most practical problems in Internet. A number of approaches have been introduced in recent years, Web caching and content distribution [1]. The solution may depend on the purpose of the network. Commercial networks could be more efficient based on the content distribution network (CDN) concept, but the education networks could be preferably based on the cache-mesh architecture [2]. In this paper, we consider only the cache-mesh architecture for the simple reason that we work within the research and education network FREEnet.

In any case, for analyzing network efficiency, we need some traffic characteristics that can be measured reproducibly. In other words, we must define some quantities and the ideal experimental scheme in which these quantities could be measured. Clearly, these measurements should be reproducible, i.e., could be repeated by other groups and on other networks. This definition clearly reminds us of the measurement concept in physics.

Suppose that we must choose between two or more strategies for caching Web traffic. For example, we would like to choose the best path for getting information via a distributed cache-mesh. The following questions arise:

Q1: How can the strategies be compared to obtain the conditions under which some are preferable (are more efficient, minimize an average access time, or maximize channel efficiency, and so on)?

---

[1] E-mail: sakr@chg.ru
[2] E-mail: lev@landau.ac.ru

Q2: Are the measurements reproducible?

Q3: If so, what is the accuracy of the measurements?

Q4: Or, more simply, how long should the measurements be continued to attain an given accuracy, e.g., 5

In addition, we know some properties of Internet traffic that could lead to difficulties in measurement.

The first set of properties is connected with natural characteristics of human activity. In this connection, the daily working hours leads to a daily periodicity of the Web traffic, the work week leads to a weekly periodicity (weekends!), the annual calendar leads to an annual periodicity (winter and summer vacations!), and so on. Holidays and important events (political campaigns, etc.) can also affect the traffic.

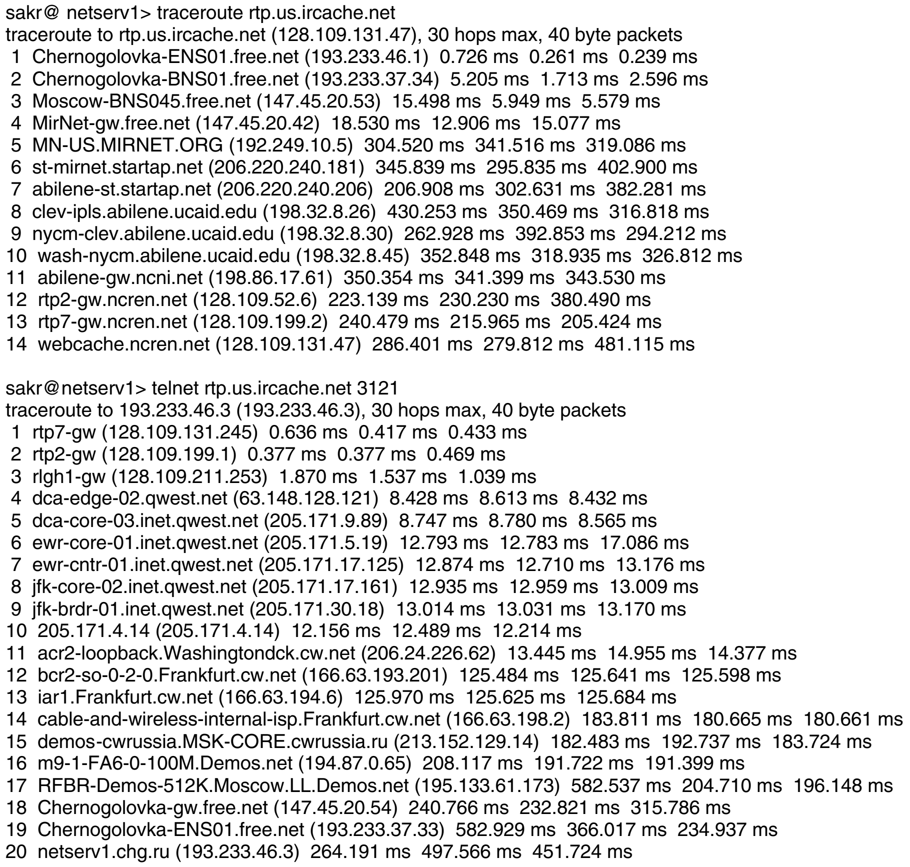
The second set of the Web traffic properties is connected with the fact that the path from one point in the Web (e.g., user) and another point (e.g., server) is not stable. The path is often quite complicated and consists of a number of routers, channels, caches, etc., which changes with time because the network constantly develops. We thus have the problem that the Internet traffic is not constant in time. This reminds us of the ancient philosopher Heraclites, who asserted “You cannot step twice into the same river.” We could say the same about the Internet river. It is perhaps early to discuss turbulence in the Internet, but we are really very close to formulating that concept.

The third set of properties further complicating the measurements discussed is connected with the dynamics of the Internet traffic for many autonomous systems (AS). This leads to random changes in the topology and therefore in the timing and loading characteristics. As a result, the Internet traffic route is sometimes very asymmetric (Fig. 1).

The last difficulty is connected with the resolution we use to examine the Internet. We could average the quantity of interest over the whole Internet; this could be the first approximation in which we answer any of the formulations of the question Q1 above. At first glance, the next approximation seems to be averaging on the scale of the top-level domains. But this is not the case at all, because many domains are inhomogeneous (i.e., have different outer-channel bandwidths and server response times for various reasons). Moreover, some are not compact. The best examples of noncompact top-level domains are *com* or *net*, which spread over the whole planet. Therefore, a special analysis is needed for understanding how this spreading or inhomogeneity could affect our measurement results. Probably, it is much better to choose ASs as the second level of coarse-graining we discuss. In practice, however, this needs more work and some modification of the existing software.

In fact, it is an independent and very interesting question to examine the data at the several scales of the global network. This interesting question will be the subject of independent research [3] for two reasons: first, we must define a clear and stable approach for the measurements and, second, this work requires considerable time.

One of the goals of the present work is to compare two different strategies for choosing the best way of getting information: directly from the original source, or indirectly from one of the parent cache servers in the cache hierarchy. The cache hierarchy allows forwarding requests to another cache server instead of the source server. This routing can be static by domains (e.g., one server is responsible for requests to the *.COM* domain, a second to the *.RU* domain, a third to the *.DE* domain, etc.) or by AS [4], or it can be dynamic as when the cache server can dynamically optimize the ‘distance’ from the



```
sakr@ netserv1> traceroute rtp.us.ircache.net
traceroute to rtp.us.ircache.net (128.109.131.47), 30 hops max, 40 byte packets
 1  Chernogolovka-ENS01.free.net (193.233.46.1)  0.726 ms  0.261 ms  0.239 ms
 2  Chernogolovka-BNS01.free.net (193.233.37.34)  5.205 ms  1.713 ms  2.596 ms
 3  Moscow-BNS045.free.net (147.45.20.53)  15.498 ms  5.949 ms  5.579 ms
 4  MirNet-gw.free.net (147.45.20.42)  18.530 ms  12.906 ms  15.077 ms
 5  MN-US.MIRNET.ORG (192.249.10.5)  304.520 ms  341.516 ms  319.086 ms
 6  st-mirnet.startap.net (206.220.240.181)  345.839 ms  295.835 ms  402.900 ms
 7  abilene-st.startap.net (206.220.240.206)  206.908 ms  302.631 ms  382.281 ms
 8  clev-ipls.abilene.ucaid.edu (198.32.8.26)  430.253 ms  350.469 ms  316.818 ms
 9  nycm-clev.abilene.ucaid.edu (198.32.8.30)  262.928 ms  392.853 ms  294.212 ms
10  wash-nycm.abilene.ucaid.edu (198.32.8.45)  352.848 ms  318.935 ms  326.812 ms
11  abilene-gw.ncni.net (198.86.17.61)  350.354 ms  341.399 ms  343.530 ms
12  rtp2-gw.ncren.net (128.109.52.6)  223.139 ms  230.230 ms  380.490 ms
13  rtp7-gw.ncren.net (128.109.199.2)  240.479 ms  215.965 ms  205.424 ms
14  webcache.ncren.net (128.109.131.47)  286.401 ms  279.812 ms  481.115 ms

sakr@netserv1> telnet rtp.us.ircache.net 3121
traceroute to 193.233.46.3 (193.233.46.3), 30 hops max, 40 byte packets
 1  rtp7-gw (128.109.131.245)  0.636 ms  0.417 ms  0.433 ms
 2  rtp2-gw (128.109.199.1)  0.377 ms  0.377 ms  0.469 ms
 3  rlgh1-gw (128.109.211.253)  1.870 ms  1.537 ms  1.039 ms
 4  dca-edge-02.qwest.net (63.148.128.121)  8.428 ms  8.613 ms  8.432 ms
 5  dca-core-03.inet.qwest.net (205.171.9.89)  8.747 ms  8.780 ms  8.565 ms
 6  ewr-core-01.inet.qwest.net (205.171.5.19)  12.793 ms  12.783 ms  17.086 ms
 7  ewr-cntr-01.inet.qwest.net (205.171.17.125)  12.874 ms  12.710 ms  13.176 ms
 8  jfk-core-02.inet.qwest.net (205.171.17.161)  12.935 ms  12.959 ms  13.009 ms
 9  jfk-brdr-01.inet.qwest.net (205.171.30.18)  13.014 ms  13.031 ms  13.170 ms
10  205.171.4.14 (205.171.4.14)  12.156 ms  12.489 ms  12.214 ms
11  acr2-loopback.Washingtondck.cw.net (206.24.226.62)  13.445 ms  14.955 ms  14.377 ms
12  bcr2-so-0-2-0.Frankfurt.cw.net (166.63.193.201)  125.484 ms  125.641 ms  125.598 ms
13  iar1.Frankfurt.cw.net (166.63.194.6)  125.970 ms  125.625 ms  125.684 ms
14  cable-and-wireless-internal-isp.Frankfurt.cw.net (166.63.198.2)  183.811 ms  180.665 ms  180.661 ms
15  demos-cwrussia.MSK-CORE.cwrussia.ru (213.152.129.14)  182.483 ms  192.737 ms  183.724 ms
16  m9-1-FA6-0-100M.Demos.net (194.87.0.65)  208.117 ms  191.722 ms  191.399 ms
17  RFBR-Demos-512K.Moscow.LL.Demos.net (195.133.61.173)  582.537 ms  204.710 ms  196.148 ms
18  Chernogolovka-gw.free.net (147.45.20.54)  240.766 ms  232.821 ms  315.786 ms
19  Chernogolovka-ENS01.free.net (193.233.37.33)  582.929 ms  366.017 ms  234.937 ms
20  netserv1.chg.ru (193.233.46.3)  264.191 ms  497.566 ms  451.724 ms
```

Fig. 1. One of the snapshots of the asymmetry in the traffic routing.

information source to its consumer. The distance can be measured by number of hops and round-trip time (RTT) for test packets. This possibility (ICMP-pinging and netprobe database) is realized, for example, in Squid starting Version 1.1.9 [5,6]. We are not aware of any publications on the NLANR cache-mesh dynamic routing performance.

2. Rewind-and-replay

In this section, we describe our first approach toward the understanding the problem of comparative measurements. This approach is based on the rewind-and-replay idea. Running squid we accumulated an archive of log files. Their records contain the queries with the timestamp, destination URL and so on (we emphasize only the information we need). These records could be used to repeat queries to the same destination at the same time of day and day of week. It seems that we could thus avoid the first problem mentioned in the previous section, i.e., the influence of the daily and weekly periodicity.

To be definite, suppose we are interested in the latency time t_l experienced by a user surfing Web-space [7]. We assume that this time is a function of the position x_u

of the user computer in Internet and of the position x_d of the destination server, i.e., $t_l(x_u, x_d)$. We can neglect the difference between the user-computer position x_u and the our experimental-setup position x_e because the transaction time t_i inside the campus network is typically much shorter then the latency time t_l. Finally, we use the notation $t_l(x_d)$ omitting the first variable: all measurements are performed from the same point, the experimental set placed in the NOC of AS 9113.

The position of the destination x_d could be regarded as the same for all servers within the given domain. This is a very rough assumption that is good for some extremes, like the domain *jp* or *tw*. For these domains, the main influence on the latency time is due to the overseas links. But such an assumption seems unsuitable for other extremes, like communication with the domains *com*, *org* and *net*, which are geographically noncompact. In this case, we can expect that the distribution function of the latency time for the destination servers in these domains might be multimodal.

We expect that in the limit of an infinite number of user requests R_D to the given domain D, the mean value $T_l(D)$ of latency times $t_l(D)$ exists:

$$\frac{1}{R_D}\sum_{r=1}^{R_D} t_l(D) \longrightarrow T_l(D).$$

We should not expect the distribution of the latency times to be Gaussian. Moreover, it could be multimodal in the case where the domain D is divided in some parts with essentially different throughputs of the external connections [8] (due to different link bandwidths and/or router performance). Therefore, the distribution could typically contain several characteristic times. Changing some external parameter, we could make one maximum larger than other and vice versa; therefore, a parameter value parameter should exist at which the two maximums are equal, demonstrating a phenomenon similar to the first-order phase transition in statistical mechanics. The simple and clear model of that phenomena will be analyzed elsewhere [8].

In the case where the distribution is unimodal, it seems to have a log-normal body with a power-law long tail [9,10]. Many authors usually use the median value $T_l^{(m)}(D)$ as the adequate measure for the latency time [10]. It is clear that the number of requests R_D over which we sum in the above equation should correspond to a time period T much larger than the inverse of the dominant frequency ω of traffic 'oscillations,' $T \gg 1/\omega$, in order to smooth their influence. This means that the average will have a reasonable value when taken over at least some number of days.

How could the latency times for two strategies of Internet traffic management be compared analytically?

We use the rewind-and replay procedure to compare measurements of the latency time T_l as follows. We take the log files for several periods and use them to imitate user requests. We send the imitation requests at the same time of the same weekday as the original ones, and we send them twice to the cache-servers with different strategies. We then compare the values of the averaged quantities. We can thus diminish the influence of the day-of-week and hour-of-day influences. The ratio $\mathcal{R} = T_l^I / T_l^{II}$ of the mean values of the latency times t_l^I and t_l^{II} for two different strategies I and II can be expected to be measured with reasonable accuracy.

As an example of the rewind-and-replay approach, we used two test machines imitating user activity. These machines used a number of log files grouped into set-1, set-2, set-3, and set-4, taken from the cache server www-cache.chg.ru – proxy server of the Scientific Center of the Russian Academy of Sciences at Chernogolovka. These sets are log files for the 24 hours accumulated on different weekdays (Monday, Tuesday, Wednesday and Thursday) in January and February 2000.

The same queries were simultaneously sent to two different cache servers. Both servers used identical hardware with 256MB RAM and 9GB-disk space under the FreeBSD operation system running the cache software squid-2.3.Stable2. The second server, setup #2, was configured for direct access, and the first one, setup #1, was configured to be able to use five main NLANR [2] cache servers (uc.cache.nlanr.net, bo.cache.nlanr.net, pb.cache.nlanr.net, sd.cache.nlanr.net, sv.cache.nlanr.net) as parents with the closest-only option. This option means that our setup #1 sent requests to the parent only when this way was expected to be the shortest one (in sense of RTT).

Setup #2 with direct access used default routing via several external channels: the 512 kbps sub-channel demos.net (Moscow) – cw.net (New York) rented from Demos Ltd, the channel RBnet (Moscow) – TeleGlobe (New York) for access to some Canadian European networks (for example, NorduNet), and 6 Mbps MIRnet channel to StarTap (Chicago) for access to some American universities, members of vBNS/Internet2. According to the agreement between FREEnet and NLANR, setup #1 was able to use the 6 Mbps MIRnet channel for access to all the NLANR cache servers.

The resulting log files were analyzed using the calamaris-2.29 package [12], which calculates the average speed of the data transfer $v = S/t_l$ as the ratio of the document size S to the latency time t_l and the ratio $R = T_l^I/T_l^{II}$. Here, $T_l^I/$ and T_l^{II} are the mean values of latency times measured by the respective servers setup #1 and setup #2.

Fig. 2 shows the mean efficiency parameter R averaged additionally over all four sets (actually, the most heavy weakly traffic), measured for different top-level domains. This figure clearly shows the advantages of using cache mesh. The value of the efficiency parameter R averaged for all requests (all domains) is about 2. Only some domains (*edu*, *ca*, *se*, and *de*) have values of R close to 1. This is probably because direct access to those domains use alternative channels (MIRnet or RBnet) which are the same (MIRnet in the

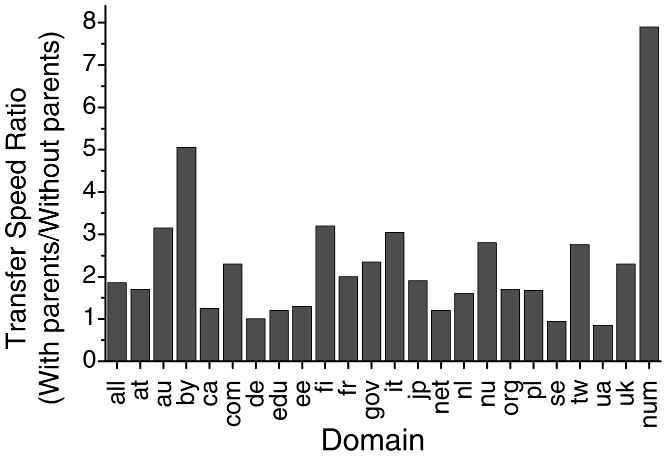


Fig. 2. The ratio of mean speed of the document retrieving from the Web for some top-level domains with and without cache mesh.

case of edu) or relatively underloaded (RBnet in the case of *ca*, *se*, and *de*) as with setup #1 using MIRnet.

Although this method seems to be a good candidate in the arsenal of measurements it has a drawback. This method practically doubles Internet traffic and could therefore change the conditions under which the measurement is performed (reminiscent of the observer problem in quantum mechanics). However, we could not estimate the value of these changes.

Analysis of the results shows that this procedure, unfortunately, does not give stable results for the latency times as expected [7] and the latency times seem to be random. At first, we believed that this was due to insufficient statistics. But, in fact, the greatest effect comes from the change of the network itself. We will discuss this in another publication.

3. Cache-triangle

We introduce the cache-triangle experimental setup of servers in order to minimize the influence of the network properties discussed above on the results of the comparative study of Internet traffic. It is constructed from three cache servers as show on Fig. 3. The first one, the Master, only sends queries to two slaves, the Left-slave and the Right-slave. Both slaves can communicate with the rest of the world. The Master sends queries to slaves alternatingly, for example, odd queries to the Left-slave and even queries to the Right-slave. It is obvious that for a sufficiently large number of queries, we could not distinguish between right and left. Therefore, our configuration is fully symmetric in the limit of an infinite number of queries. It is very important that this symmetry holds for any configuration of slaves. As a consequence, we could compare statistics of the Left- and

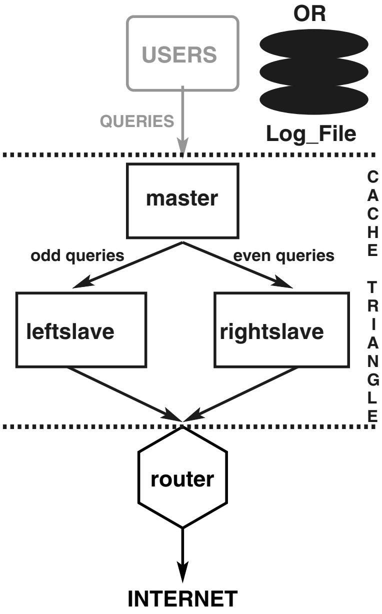


Fig. 3. The symmetric configuration of servers for the measurements — cache-triangle.

Right-slaves being sure they are not subject to daily or weekly periodicity, or the fact that sysadm Smith at the University of the Middle of Nowhere has changed the routing of his domain. He is not able to destroy the symmetry of our cache-triangle.

Moreover, we occasionally check the sensitivity of our experimental device. We found asymmetry in the first measurements, and this was due to a wrong configuration of the two cache-servers on the next level of the hierarchy. These servers were wrongly communicating with one of the slaves as with a parent.

The first configuration used was built with Master-cache (P-II/300, 192MB RAM, 8GB disk cache), which was configured to send all odd queries to the Left-slave cache (P-II/400, 192MB RAM, 8GB disk cache) and all even queries to the Right-slave cache (Celeron/433, 192MB RAM, 8GB disk cache). All cache servers were running the same version of the software Squid-2.2.Stable. The Master served real user requests (acted as a real proxy.chg.ru server) and was configured with the options ‘parent no-query round-robin htcp no-netdb’. Both slaves were configured identically with the options ‘parent closest-only’ to use NLANR cache servers [8], and with the options ‘parent closest-only htcp’ to communicate with ikia.ru.ircache.net and dau.ru.ircache.net servers, and with the option ‘sibling’ with webcache1.free.net. This setup produced log files on the Master running three weeks in the period from 13 December 2000 to 4 January 2001.

The resulting log files were processed with the seafood-1.13 software [13] (version dated 09.06.2000) and only real non-HIT queries from the master to both slave caches were taken into account. The traffic speed was calculated in Kbit/sec as the ratio of $\sum \text{Kbit}/\sum \text{sec}$.

3.1. Measurement results over three weeks of the symmetric configuration

First, we determined how symmetric are (1) the total number of queries, (2) the total traffic, and (3) the average speed of the traffic.

(1) We plot the number of queries for different top-level domains in double log scale in Fig. 4. Data is ordered in descending order for the Left-slave (solid stars), and

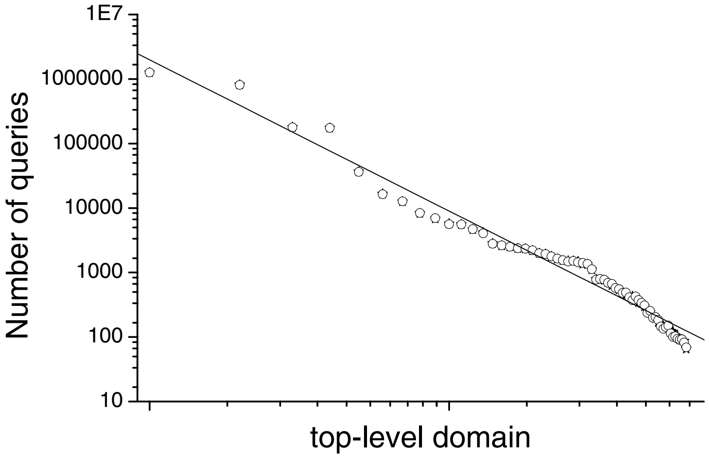


Fig. 4. Total number of queries during the three weeks of work of the cache-triangle in symmetric configuration. See text for the details.

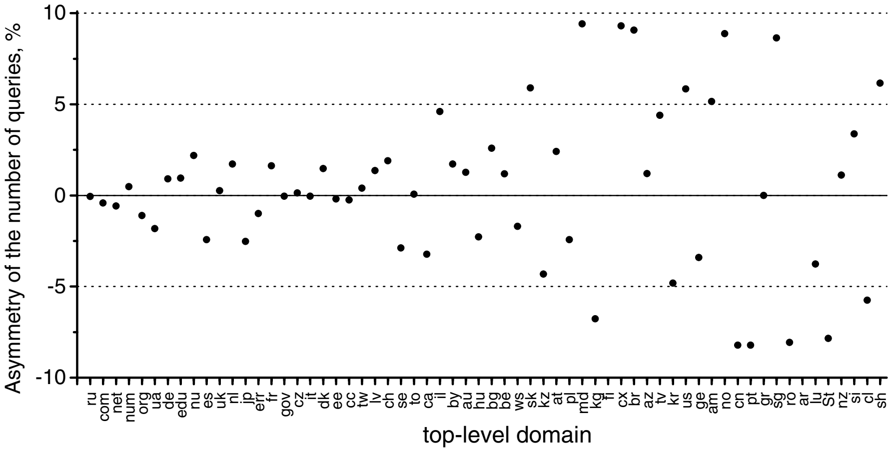


Fig. 5. Relative difference in the number of queries served by the Left-slave and by the Right-slave.

the corresponding number of queries for the Right-slave are marked by open circles. Clearly, the picture demonstrates that the number of queries coincides in the log–log scale. It is interesting that the number of queries decreases with the domain number by an approximate power law $n^{-2.34(5)}$ shown by the solid line, although this law is not perfect and there are some oscillations of the number of queries. Here and throughout paper, n denotes the rank of a given domain in the ordered sequence. The one-sigma uncertainty in the last digit is shown in parentheses. We call the effect of the power-law decay in the number of queries the *power-law decay of domain popularity*. This is very similar to Zipf's law distributions of Web objects [10,11].

To check asymmetry in the number of queries, we plot the relative difference in the number of queries, measured in percent, $\frac{N_r - N_l}{N_l} 100\%$, where N_l (N_r) is the number of queries served by Left-slave (Right-slave), in Fig. 5. The respective labels *num* and *err* denote unresolved domain names and requests with errors in URL. The two horizontal lines show the 5% acceptance interval chosen as the good accuracy for the measurements. All events are within this accuracy for the domains with more than 700 queries. The fluctuations are larger than 5%, but still less than 10% in the tail of the decay in domain popularity.

(2) In the Fig. 6, we plot the total traffic in Megabytes as a function of domain-name rank. Domains are ordered as described in 1. The open circles (closed stars) represent the total traffic for the Right-slave (Left-slave). Both curves almost coincide. The decay of the traffic is described by the power law $n^{-2.25(16)}$, which is the same as for the number of queries within a two sigma error.

Nevertheless, there is a visible asymmetry for some domains. We examine this in more detail.

In Fig. 7 in double log scale, we plot the relative difference of the total traffic coming through the Left- and Right-slaves $100(t_l - t_r)/t_l$, where t_l (t_r) is total traffic for Left-slave (Right-slave). We exclude all domains with less than 250 queries because of the very large fluctuations. The horizontal solid lines show the 10% acceptance interval we choose for the accuracy of the symmetry of the total traffic. There is a large asymmetry for some domains.

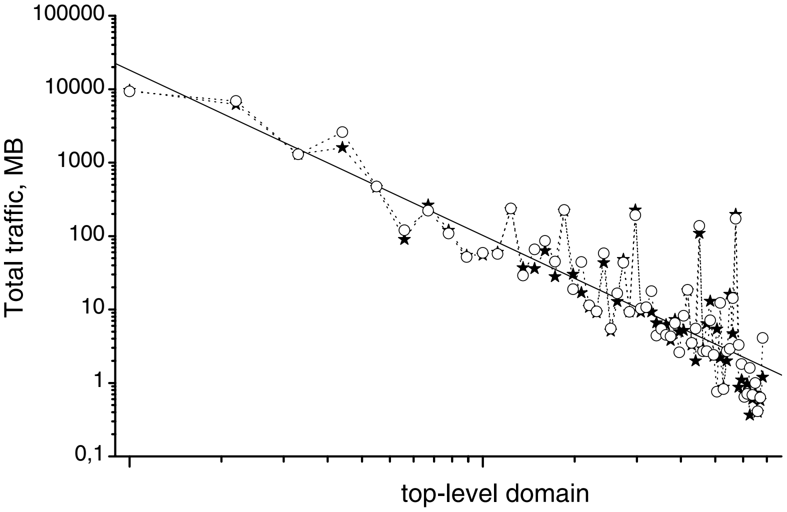


Fig. 6. The total traffic served by the Left-slave (closed stars) and by the Right-slave (open circles) in double-logarithmic scale. Solid line is the linear fit to the data.

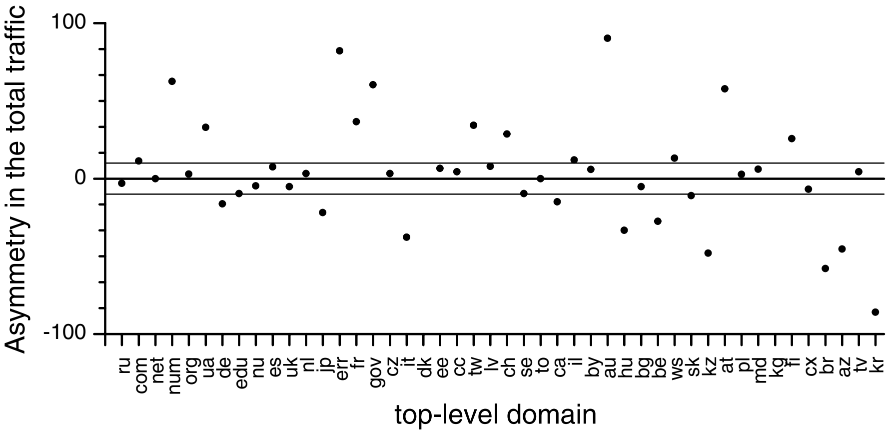


Fig. 7. Relative asymmetry of the total traffic shown on Fig. 6. Solid lines is the 10% interval.

The large deviation for the domain with more than 1000 queries is for the artificial domains *num* and *err*, which could be dropped from the analyses, and for the domains *ua* (33), *jp* (−22), *fr* (37), *gov* (60), *it* (−38), *dk* (163), *tw* (34), *ch* (28.7), and *au* (90). What is the origin of these deviations? We examine the average document size *m* for a given top-level domain. It could be calculated as the ratio of the total traffic to the total number of queries. The average document size for the Left-slave is 8.3 KB and for the Right-slave is 8.9 KB. The asymmetry in the average package size is about 7% and is rather large. Although asymmetry in the number of queries is 0.1%, it is 6.6% in the total traffic. This explains our choice for the acceptance interval for deviations from symmetry.

The largest document sizes are 1347KB (Left) and 1287KB (Right) for the domain *pt*, 256KB and 370KB for the domain *fi*, 152KB and 133KB for the domain *ca*, and

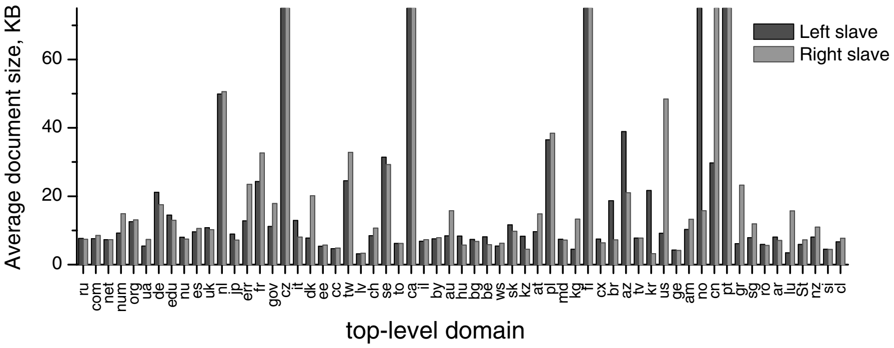


Fig. 8. Average document size for the symmetric configuration.

93KB and 95KB for the domain *cz*. Size of the packages for the domain *fi* have a very simple explanation. In Russia, the most popular Finnish network source is the ftp server ftp.funet.fi. It is clear that in the case of a large average file size for a given domain, we will have large fluctuations in the averages.

The typical document size distribution is known to be log-normal [9,10]. The rare requests for very large files could lead to enormous fluctuations in the averages. Analysis of the data supports this idea: the largest deviations occur for such domains where the average sizes of documents are far from the average value for all domains. This is especially well demonstrated with the domains *no* (average package sizes are 95KB and 16KB) and *cn* (30KB and 97KB).

In Fig. 8, we show average document sizes for the top-level domains. We exclude data for the above-mentioned domains to make the figure more readable. Generally, the average size is less than 10KB, and totally this picture clearly supports the influence of the rare and large packages on the distribution of the average package size.

In fact, the most interesting quantity is the average latency time, which is connected to the speed with which we receive packages.

(3) In Fig. 9, we show values of the average speed for the domains ordered in the same way as before.

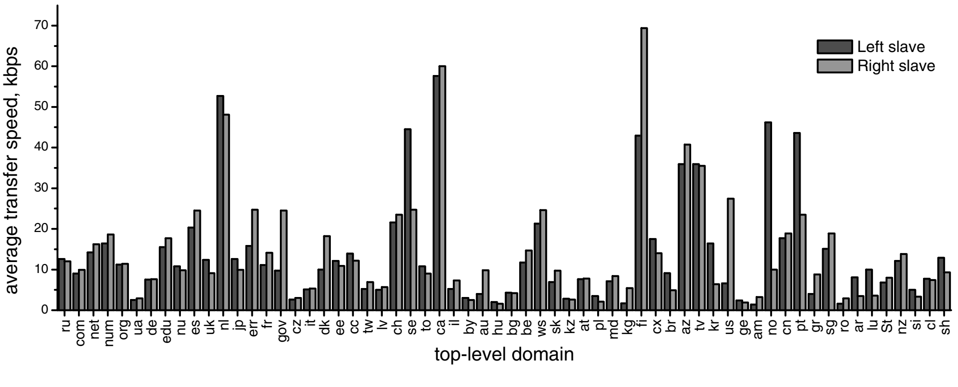


Fig. 9. Average documents transfer speed in Kbit/sec, which served by symmetric cache-triangle configuration.

Clearly, there is some correlation of the document transfer speed served by the Left- and Right-slaves, although fluctuations are quite large. The origin of these fluctuations is clearly connected to the above-mentioned long-tail distribution of the package sizes: the maximum speed and the difference in speed are for *fi*, *ca*, *pt*, *no*, etc. However, this is not the case, for example, for the domain *cz*. This quantity is probably also very sensitive to the nonuniformity of some domains and/or to the geographical spreading of some domains.

We have discussed the results of analyzing three weeks of measurement. We also analysed the data for each of the three weeks separately in the same manner as for the summary of three weeks. We did not find any essential difference. This means that averaging over three weeks gives some stable results, which give us confidence that the effects we discussed are not something cached from the random flow of Internet traffic rivers.

3.2. Measurement results over three weeks of an asymmetric configuration

For the next series of experiments, we increased the symmetry and power of the cache-triangle and introduced an asymmetry in the slave strategies. The Master (proxy.chg.ru) was based on the CPU AMD Athlon/850 with 512 Mb RAM and 30 Gb disk cache. Both slaves, the Left (ath2.chg.ru) and the Right (ath3.chg.ru), were absolutely identical and based on the CPU AMD Athlon/850 with 384 MB RAM and 20 GB disk cache. All three servers were running Squid-2.3.Stable4 under FreeBSD-4.2.

Master related to both slaves as ‘parent no-query round-robin no-netdb’. The Left-slave (ath2) communicated with the NLANR-servers (*uc, pb, sd, bo, sv* as ‘parent closest-only’ and with ikia.ru.ircache.net, dau.ru.ircache.net and webcache1.free.net as ‘parent closest-only htcp’. The Right slave was configured to obtain information only directly from the origin.

The Master served real user requests, and the log files accumulated in the period from 10 February 2001 to 3 March 2001 was processed with the seafood-1.13 software [13] (version dated 09.06.2000) and only real non-HIT queries from master to both slave caches were taken into account.

The total number of queries decayed with the same exponent as in the previous experiment. Moreover, the data coincide quite well. This is additional support that statistics over about three weeks are stable. Analysis shows that the same statement is also valid for the total traffic.

The average document size served by the Left-slave configured to use cache-mesh and through the Right-slave with direct access to the origin is shown in Fig. 10. Comparison of Fig. 10 with Fig. 8 immediately shows the difference. Namely, the average package sizes are larger in the case of cache-mesh usage.

We note that the following domains are excluded from the picture: *pt* (average document size is 2923.3KB(!) with 90 Left requests and 32.7KB with 102 Right requests), *kr* (211KB and 275KB), and *it* (231KB and 218KB).

In Fig. 11, we show the average document transfer speed. We note that there are no visible differences between the two strategies for many domains, but the preference of using cache-mesh is clear for many others (*com*, *net*, *org*, *ua*, *tw*, *fr*, *ca*, *pl*). Probably, we must find some another parameter in this case. This reminds us of one more statement of Heraclites. We do not hesitate to place it here as the ending message of our analyses: ‘The hidden harmony is better than the open’. We are going to hunt it!

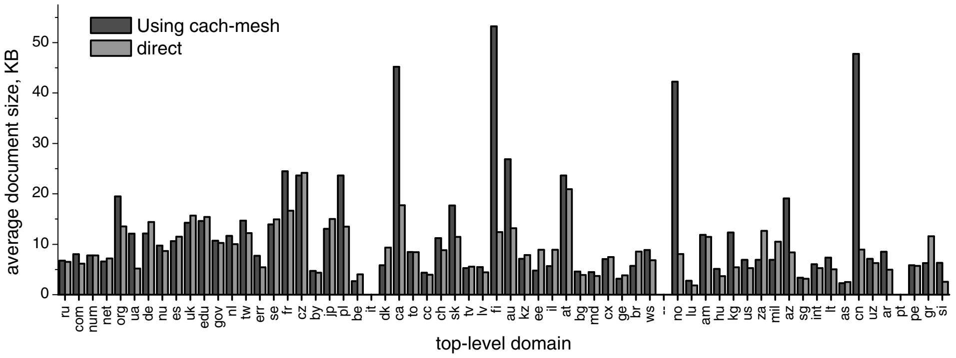


Fig. 10. Average size of the documents served by the Left-slave server using cache-mesh and by the Right-slave server with the direct connection to the origins.

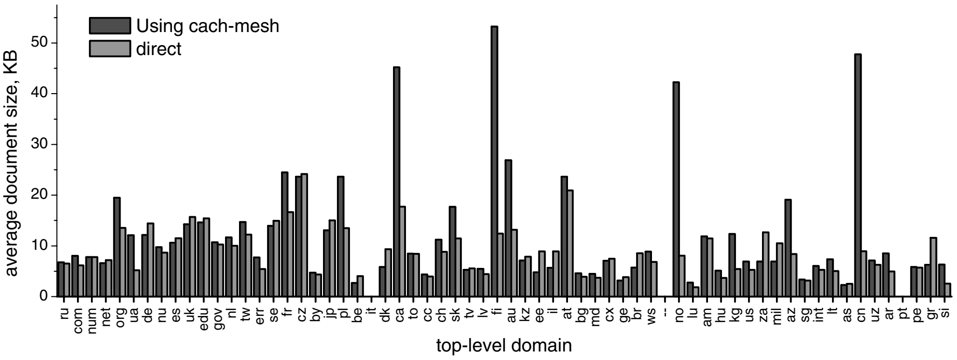


Fig. 11. The average documents speed in Kbit/sec. See caption to Fig. 10.

4. Conclusion and proposals

The analyses presented here show that the cache-triangle could be used for comparative measurements of Internet traffic. We have started a new set of measurements in which we combine both our ideas. Namely, we use the rewind-and-replay strategy with the cache-triangle. The preliminary results look very stable. We hope to extract more detailed information on the structure of Internet flow from this set of experiments.

Acknowledgments

This work was supported in part by the RFBR (Grant Nos. 99-07-90084 and 01-07-90119). One of the authors (LNS) acknowledges the CARIPLO fellowship and the Landau Network–Centro Volta for the financial support and University Milano-Bicocca for the kind hospitality during the final stage of this work.

References

[1] S. Gadde, J. Chase, M. Rabinovich, Web caching and content distribution: A view from an interior, *Computer Communs.*, 24, 222–231 (2001).

[2] Duane Wessels, The NLANR Cache System, http://www.ircache.net/Cache/

[3] L.N. Shchur and S.A. Krashakov, Coarse-graining of network and Internet traffic, in progress.

[4] C. Grimm, J.-S. Vockler, H. Pralle, Request routing in cache meshes, *Computer Networks and ISDN Systems*, 30, 2269–2278 (1998).

[5] D. Wessels, K. Claffy, ICP and Squid Web cache, *IEEE J. Selected Areas in Communs.*, 16, 345–357 (1998).

[6] D. Wessels, Squid Web proxy cache, http://www.squid-cache.org

[7] S.A. Krashakov, L.N. Shchur, Active measurements (experiments) of the Interne traffic using cache-mesh, *International Journal of Modern Physics C (Physics and Computers)*, 2001, to be published.

[8] L.N. Shchur, and S.A. Krashakov, First order phase transition in the Internet traffic, in progress.

[9] D. Helbing, B.A. Huberman, S.M. Maurer, Optimizing traffic in virtual and real space, In: *Traffic and Granular Flow'99*, Ed. D. Helbing, H.J. Hermann, M. Schreckenberg, D.E. Wolf, Springer: Berlin, 2000.

[10] P. Barford, A. Bestavros, A. Bradley, M. Crovella, Changes in Web client access patterns. Characteristics and caching implications, *World Wide Web J., Spec. Issue on Characterization and Performance Evaluation*, 2, 15–28 (1999).

[11] L. Breslau, P. Cao, L. Fan, G. Phillips, S. Shenker, Web caching and Zipf-like distributions: Evidence and implications, *Proc. IEEE INFOCOM'99*, 126–134 (1999).

[12] Cord Beerman, Calamaris, http://Cord.de/tools/squid/calamaris/

[13] J.-S. Vockler, Seafood: Extended cache statistics, http://www.cache.dfn.de/DFN-Cache/Development/seafood.html