Rudyk Tetiana
Candidate of Physical and Mathematical Sciences, Associate Professor,
Associate Professor at the Department of Mathematical Physics
National Technical University of Ukraine
«Igor Sikorsky Kyiv Polytechnic Institute»
Рудик Татьяна Александровна
кандидат физико-математических наук, доцент,
доцент кафедры математической физики
Национальный технический университет Украины
«Киевский политехнический институт имени Игоря Сикорского»

Sulima Olha
Candidate of Physical and Mathematical Sciences, Associate Professor,
Associate Professor at the Department of Mathematical Physics
National Technical University of Ukraine
«Igor Sikorsky Kyiv Polytechnic Institute»
Сулима Ольга Викторовна
кандидат физико-математических наук, доцент,
доцент кафедры математической физики
Национальный технический университет Украины
«Киевский политехнический институт имени Игоря Сикорского»

METHODS OF TRAFFIC REGULATION AND USER REPUTATION HANDLING IN THE BITTORRENT PEER-TO-PEER NETWORKS

МЕТОДЫ РЕГУЛИРОВКИ ТРАФФИКА И ОБРАБОТКИ ПОЛЬЗОВАТЕЛЬСКОЙ РЕПУТАЦИИ В ОДНОРАНГОВЫХ СЕТЯХ BITTORRENT

Summary. *Various methods for traffic regulation and reputation handling in the distributive and client context of BitTorrent network are analyzed. The overview of the methods for the calculation of the user reputation in the private trackers and corresponding reputation-based access systems are performed.*
Key words: *traffic regulation, BitTorrent networks, handling reputation.*

Аннотация. *Рассмотрены различные методы регулировки трафика и обработки пользовательской репутации в контексте сетей BitTorrent. Выполнен обзор методов вычисления пользовательской репутации и способов разграничения качества сервиса на их основе.*
Ключевые слова: *регулировка траффика, сети BitTorrent, обработка репутации.*

Introduction. Peer-to-peer networks were not nearly new at the beginning of modern century, and their concept was briefly outlined in the times of Internet very inception back in 1969. Although the contributors could not possibly have predicted the future scale of worldwide distribution of what was then a single link between just two mainframe computers, the idea of interconnected peer nodes was already there.

User interface terminals at the time were nowhere near to compare with host computers (mainframes), and were essentially lacking any computing and storage facilities whatsoever, hence the vision of peering networks remained dormant for long time since.

Only as the mainstream computers surged into the consumer market during 1970-s and 1980-s, the legacy of what we know today as "client-server architecture" was to be dominant for decades to come. It was assumed that should there be a network, it is naturally divided into servers (that provide access to resources) and clients (that make use of provided resources). The performance and capacity gap between server and client

hardware and, which is more important, a difference between network interconnections was still too obvious.

At that time, peering was common practice when dealing with server software and network architecture. TCP/IP routing schemes was essentially peering to the point that the very word "peering" made it into the specific technical term on internetworking routing, despite the fact that actual physical channels had (and still have) visible relevance to national backbones and traffic exchange points, making them more or less subordinate to each other. However, Usenet and e-mail servers were communicating with each other and there were no such thing as primary layer or central hub(s) through which all traffic should be passed — which is peering network.

Outside of Internet, attempts to build peering networks were also undertaken. One of the most successful of those attempts was FidoNet — amateur worldwide computer network, initially consisting of independent bulleting board systems (BBS), built on packet-switching principle over regular telephone lines using dialup modems. Unlike Internet, FidoNet is not online-network and all user interaction could be and mostly done in offline state. Host software, however, is required to maintain online availability during the certain policy-defined hours each day.

Right upon emerging, the FidoNet was truly peering, in the sense that each originating node accessed its addressee directly by calling its address (phone numbers in this case). Later in 1990-s, however, FidoNet had also "suffered" from infrastructure growth, when the network had exploded into thousands of nodes worldwide. These times of FidoNet development were marked with strict hierarchical structure, roughly based on geography and various regulating authorities within the network. It is worth noting, that unlike Internet (IPv4 address space making up 2^{32} addresses, including non-routable and reserved), hierarchical address structure of FidoNet theoretically allowed address space of 2^{48} network nodes alone and 2^{64} connection points in total.

Despite all aforementioned advances and peeks into the future concept, truly peer-to-peer online networks as we understand them today were far from reach before the advent of third millennium.

The commercial grounds for real peer-to-peer networks have appeared not until permanent Internet connections (also called then "leased lines") built on technologies such as ADSL or DOCSIS gained significant consumer market at homes and offices. In addition, not until average home and office computer hardware was closing to the average server hardware (often being built from the same parts indeed) was it plausible to build peer-to-peer networks with evenly distributed computing and storage resources [1, p.336].

It is widely believed, that commercial applications of the concept started to appear and gained much popularity in the beginning of XXI century.

An introduction to BitTorrent technology.

One of the modem peer-to-peer network protocols, BitTorrent, was conceived in 2001 and to date remains responsible for largest part of consumer-generated Internet traffic, sometimes prompting Internet Service Providers (ISPs) to implement special, often unpopular, filtering measures and devices.

Unlike other popular peer-to-peer networks such as eDonkey2000 or Gnutella networks, BitTorrent does not constitute a single addressing or naming space. It is not even a network itself, because BitTorrent operates as multitude of independent content-tracking servers, called "trackers". Each tracker maintains the list of published content entities, and for each entity, it maintains the list of peers associated with it. Most trackers do not communicate with each other, as eDonkey2000 servers do, unless they are sharing same content and are specially designed to exchange information among themselves.

Due to the absence of overhead related to maintaining global naming or addressing space, BitTorrent networks are quite faster in comparison with eDonkey2000 or Gnutella in terms of download and upload speed and length of download queues. BitTorrent clients are most likely to consume their bandwidth to exhaustion, despite the fact that BitTorrent does not imply sophisticated load-balancing algorithms for upload, reward scores and so on [2, p.1].

Typical content lifecycle in BitTorrent could be described as the following, preparation — content publisher prepares torrent file, which describes the number, names and size of files and the control checksums of each slice of binary stream made up from content files. Publication — publisher uploads torrent file in such a way that tracker became aware of its existence, not necessarily knowing all the details specified in the torrent file. Distribution — publisher distributes torrent file among clients who wish to download its content. It is usually done through web-based forums, either public or private or via other means. It is worth noting that publication and distribution is not the same process, although in most cases they are done simultaneously in the scope of one server. For example, uploading torrent file as file attach to the message on forum automatically registers torrent contents in the tracker. Initial seeding — publisher running BitTorrent-compliant client starts accepting incoming requests for content. Leeching — other clients proceed to download published torrent file, requesting tracker for the address of initial seeder and requesting initial seeder for content. Downloading — clients actively downloading content file will enable already downloaded slices to be shared among other clients, effectively speeding up the transfer for them. Secondary seeding — clients that completed the download, engage in seeding it by themselves. End of interest — all involved clients finishes and became seeders, and no downloading clients are left in the

swarm. Fadeout — seeders stop seeding one by one, and eventually there are neither seeders nor downloading clients associated with this torrent.

Once the content entity is fully downloaded (the transition between stage 6 and 7), the BitTorrent client must ascertain the data integrity of it. In this part BitTorrent specification seems to be slightly under-developed in comparison with its counterparts of eDonkey2000 and Gnutella networks. While the latter does use sophisticated tree-hashing algorithms designed to minimize traffic overhead, BitTorrent simply calculates hashing stream from binary stream with variable-sized chunks. If an error is detected, the whole chunk needs to be re-downloaded.

Analysis of Load-Balancing technique.

Most peer-to-peer network will eventually encounter the phenomenon called "leeching". The network client involving in leeching will only download content and not share it among others. Although such behavior is necessary for some time just after initial publication of the content (since some time is required to download at least one complete shareable piece of data), leeching beyond necessary period and for long time is considered bad, because it forces excess resource usage on other clients interested in the same content [3, p.150].

Peer-to-peer networks often employ various sophisticated algorithms to discourage leeching. One of prominent example is the credit reward system found on popular eDonkey2000 clients. Such clients maintain a "performance record" for each incoming client, who expressed interest in published content.

Typically, incoming clients are arranged into queue in order of time of their appearance. The foremost client in queue is served by the content piece and then rescheduled at the end of queue, therefore advancing other queue members.

However, incoming client can advance queue member by more than single step in the queue, taking into account its contribution (in case the sharing client is not completed seeder, of course). That is, the more content pieces were provided by the incoming client, the faster it progresses in the queue. This effectively places 'had' leechers to the end of queue and slows their advance.

Unfortunately, no such reward system is currently employed by the majority of the BitTorrent clients. There are number of reasons for it, including the aforementioned difference in distribution speed (BitTorrent content usually distributes faster than comparable eDonkey2000 counterpart due to small size of swarm). However, similar scheme are designed in so called "private trackers".

As BitTorrent is developing technology, new protocol extensions are constantly added to improve the overall efficiency of content sharing. These include, for example, so-called "Fast Peer Extensions" to allow new peers bootstrap into swarm more rapidly. Although it

is uncertain whether the performance itself is nearly topping its potential for the current BitTorrent development stage, it is beyond the scope of this paper.

Public vs. Private Trackers.

Roughly, trackers can be called "public" or "private". Public tracker, such as famous Sweden-based ThePirateBay usually does not require invitation or registration to be able to download its advertised content, therefore do not maintain download and upload rating records of its users.

In contrary, private trackers, such as Torrents.Ru or many others running Tor- rentPier software, do implement some restrictions against anonymous access. This is possible using so-called private keys — special passwords attached to the announce URL of tracker, designed so that the tracker could ascertain the user identity of every announce or update request coming from BitTorrent clients.

Private trackers often employ rating system, where rating is a value calculated using various formulas including overall download and overall upload amount of a particular user. Users with low rating are restricted from further downloading or they are potential candidates to be banned from tracker. Users with high rating have certain privileges such as ability to download more torrents simultaneously, priority to access and search across tracker, etc.

Hence in order to encourage content sharing and discourage leeching, tracker server must somehow be made aware of how much some particular BitTorrent client did download and upload to others. This is currently made by issuing special HTTP request ("tracker updates") to the tracker. Such requests usually contain user identity, content identity (hash), client activity state, amount of downloaded and uploaded data and other relevant information [4, p.33].

Proposed speed-up based on logical distance measurement.

As either public or private trackers became popular, commonly encountered overloading problems may arise. Although trackers itself do not store any shared content and the storage of torrent-files themselves require comparatively low resources, the "tracking" itself takes much up the processor speed and memory consumption. This was the reason for many popular public trackers to separate tracking services from forum and torrent file storages to a dedicated server or server clusters.

However efficient this solution might be, we believe that the expansive approach is not the only nor it is optimal. As peer-to-peer technology develops rapidly, the traffic its implementations generate became more and more noticeable in overall Internet traffic, as mentioned above. Modem end-user connection technologies such as ADSL, DOCSIS and end-user optical fiber etc, made high-speed Internet connections available to virtually every technically experienced customer.

Despite this fact, the network latency still plays important role in peer-to-peer applications. It is usually up to the vision of tracker software authors, how to report seeds and peers available to new clients. Every tracker implement its own balancing mechanism, some tend to shift balance to non-completed peers about to become seeds, others tend to report seeds more than ordinary peers.

Complex methods involving calculations regarding which parts are distributed across swarm more frequently than others, are currently not implementable, as BitTorrent protocol does not allow specific piece information to be sent in regular tracker update request.

However, what was left obvious is the load balance based on logical proximity of network nodes. Although it is commonly encountered phenomenon whereas a network packet designated to neighboring building may travel slower than the packet designated to another continent, the understanding of the relative logical position of network nodes may help packet to travel faster. It is widely used practice to build national Internet Traffic Exchange Points (IXes). Countries such as Ukraine, have single exchange point (UA-IX), whereas the geographically large country such as United States of America, have seven exchange points.

Implementation of exchange points generally allow involved members to peer internet traffic to each other on mutually free-of-charge agreements thus implicitly providing customers with higher traffic speeds with resources linked under the same exchange point.

Consider the single shared content over BitTorrent network swarm, to which the newly interested client connects and requests. The tracker, which is generally unaware of the logical proximity of new client to the existing peers in swarm, reports them either randomly or based on some internal optimization algorithm. Client then proceeds to request each received peer for shared content, and, naturally, might experience faster responses if some of the remote party happened to be located under the same internet exchange point, or even linked to the same ISP.

Conclusion and recommendation. Implementation of "logical topology" — based algorithms for peer selection in either public or private trackers could potentially speed up the content distribution in BitTorrent swarms as well as with any other similar peer-to-peer technology, where clients are obliged to inquire many peer clients periodically.

Social engineering means to encourage content downloaders may also help distribute shared content more efficiently, for example, in the systems where the number of peers and their actual network proximity depends on the user rating or otherwise calculated contribution value.

References

1. Stephanos Androutsellis-Theotokis, Diomidis Spinellis. A Survey of Peer-to-Peer Content Distribution Technologies / ACM Computing Surveys, 2004. — 36(4). — P. 335–371.

2. Stefan Saroiu, P. Krishna Gummadi, Steven D. Gribble. A Measurement Study of Peer-to-Peer File Sharing Systems. Technical Report UW-CSE-01-06-02, University of Washington, Department of Computer Science and Engineering, July 2001.

3. Poryev G. V. The Application of the Peer-to-Peer Network Technologies / Proceedings of Scientific Workshop of Donetsk National Technical University. Issue № 12(118) "Computing Technology and Automation". — DNTU, Donetsk (Ukraine), 2007. — P. 150.

4. Poryev G. V. Data Integrity Control in the Distributed Networks / Western-European Magazine on Advanced Technologies. Issue № 4/2(22). — KNURE, Kharkiv (Ukraine), 2006. — P. 32–35.