МЕТОДИ ТА ПРИЛАДИ КОНТРОЛЮ ТЕХНОЛОГІЧНИХ ПАРАМЕТРІВ

УДК 004.75; 004.724.2

PEER-TO-PEER NETWORKS AND THEIR IMPACT ON THE DEVELOPMENT OF THE INTERNET AND INFORMATION TECHOLOGIES

Poryev Gennadiy

National Technical University of Ukraine "KPI", 03056, Peremohy ave. 37, Kiev, Ukraine, e-mail: core@barvinok.net

The historic conditions for the emergence of peer-to-peer networks are reviewed. The BitTorrent network architecture is analyzed, including the lifecycle of publication, load balancing techniques and tracker operation specifics. New approach to increase the throughput of BitTorrent network based on the locality class metric estimation is proposed and experimentally validated.
Key words: peer-to-peer network, the lifecycle of publication, locality class, tracker, network, architecture.

Розглянуті історичні передумови виникнення однорангових мереж. Проаналізовано архітектуру мережі BitTorrent, зокрема, життєвий цикл публікації, засоби балансування навантаження та специфіка роботи трекерів. Запропоновано та експериментально перевірено новий підхід до підвищення ефективності обміну даними в мережах BitTorrent на основі оцінки метрики класу локальності.
Ключові слова: однорангова мережа, життєвий цикл публікації, клас локальності, трекер, мережа, архітектура.

Рассмотрены исторические предпосылки возникновения одноранговых сетей. Проанализирована архитектура сети BitTorrent, в частности, жизненный цикл публикации, средства балансировки нагрузки и специфика работы трекеров. Предложен и экспериментально проверен новый подход к повышению эффективности обмена данными в сетях BitTorrent на основе оценки класса локальности.
Ключевые слова: одноранговая сеть, жизненный цикл публикации, класс локальности, трекер, сеть, архітектура.

As of today, peer-to-peer (p2p) network architecture had gained considerable amount of usage worldwide. The p2p concept itself has been introduced much earlier than the beginning of XXI century, briefly outlined in the times of Internet very inception back in 1960s. 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 implemented.

Client terminals at the time were nowhere nearly comparable with host computers (mainframes), and were essentially lacking any computing power and storage facilities. Hence, the vision of networks with purely peering nodes remained dormant for decades.

As soon as the personal computers surged into the consumer market during 1970s and 1980s, the "client-server architecture" paradigm was destined to be dominant for decades to come. It was widely implied that in any network the nodes are naturally divided into servers (nodes that provide access to resources) and clients (nodes that make use of provided resources). The performance and capacity gap between server's and client's hardware and, more important, a difference in network interconnections techniques was still too obvious.

During that period, peering was common practice when dealing with server software and network architecture. TCP/IP routing schemes were 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 directly and there were no such thing as primary layer or central hub through which all traffic should be handled; therefore this is a peering network.

Internet was and still is not the only worldwide computer network. Outside of it, attempts to build peering networks were also underway. One of the most successful of those attempts was FidoNet — amateur worldwide computer network, initially consisting of independent bulletin board systems, 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 were mostly done in offline state. Host software of a FidoNet node, however, is required to maintain online availability during the certain policy-defined hours each day.

Since its inception, the FidoNet was truly peering, in the sense that each originating node accessed its addressee directly by calling its address (PSTN phone numbers). Later in 1990s, however, FidoNet had also "suffered" from infrastructure growth, when the network had exploded into thousands of nodes worldwide. The FidoNet had attracted new users by the free nature of itself while keeping its learning curve steep enough to admit only technologically skilled users.

In its early stages, FidoNet permitted only two-tier hierarchy — there were nodes addressed by number and networks, also numbered, under which the nodes were listed such that only the node-network numbers pair was unique in its address space.

Subsequent growth of the network required address space expansion due to the concerns for usability. The concept of a zone was introduced, whereby nodes from 1 to 6 corresponded to North America (1), Europe and Russia (2) and so forth. Also the point numbers were added such that each node can maintain up to 2^{16}-1 downlinks with no requirements for online availability.

This epoch of FidoNet history was marked with strict hierarchical structure, roughly based on geography and various regulating authorities within the network. It is worth noting, that unlike then IPv4-only Internet (whose address space is 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.

Nowadays due to the widespread presence of Internet connectivity and much cheaper access prices, FidoNet is experiencing its gradual decline, with most of the remaining nodes operating over Internet instead of PSTN as a transport layer, but keeping the legacy technology otherwise unchanged.

In spite of all advances and peeks into the future paradigm, truly peer-to-peer online networks as we understand them today were far from reach before the advent of XXI century.

The consumer market grounds for real user-driven 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 market share at homes and offices. This even spawned a special term — SOHO (Small Office and HOme).

Additionally, not until average home and office computer power neared to the average server power was it plausible to build peer-to-peer networks with evenly distributed computing and storage resources [1].

It is assumed that wide-scale applications of the concept started to gain much popularity in the beginning of XXI century.

BitTorrent technology

Recent estimates (fig.1) conducted by the Sandvine Intelligent Broadband Networks indicate that by first half of 2012 as much as 50% (upstream-wise) and 24% (downstream-wise) of Internet traffic on major backbones are relevant to p2p applications. Of all modern p2p applications that gained widespread usage, BitTorrent stands prominently.

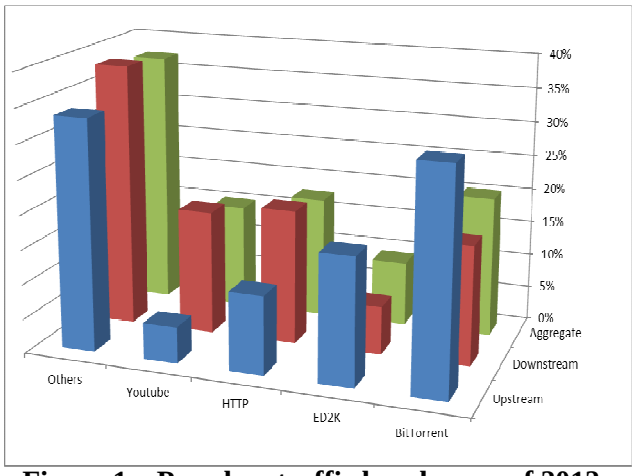

Figure 1 – Per-class traffic breakup as of 2012 according to Sandvine Intelligent Broadband Networks

The BitTorrent technology had first emerged in 2001. To date, it remains responsible for the 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 p2p 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 either operates as multitude of independent content-tracking servers, called “trackers” or functions purely as distributed hash table. 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 configured to exchange information among themselves.

Mostly because of absence of the overhead related to maintaining global naming-addressing space, BitTorrent network swarms are quite faster in comparison with eDonkey2000 or Gnutella in terms of file throughput and length of download queues [2].

Usually, the lifecycle of content consists of the following stages:

1) preparation — content publisher prepares torrent file or magnet link, which describes the number, names and size of files and the checksums of each slice of binary stream;

2) 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; with magnet links, this step is omitted;

3) distribution — publisher distributes torrent file or magnet link among clients who wish to download its content. It should be noted that publication and distribution is not the same process, especially with magnet links, 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 private tracker;

4) initial seeding — publisher running BitTorrent-compatible client starts initiating transfer connections content;

5) leeching — other clients proceed to download published torrent file or accepting magnet link, in the former case with requesting tracker for the address of initial seeder and requesting initial seeder itself for content;

6) downloading — clients actively downloading content file will enable already downloaded slices to be shared among other clients, effectively speeding up the transfer for them;

7) secondary seeding — clients that completed the download, engage in seeding it by themselves; additionally, once the content entity is fully downloaded, the BitTorrent client ascertains the data integrity of it;

8) end of interest — all involved clients finish and became seeders, and no downloading clients are left in the swarm;

9) fadeout — seeders stop seeding one by one, and eventually there are neither seeders nor downloading clients associated with this torrent or magnet link.

Overview of the balancing techniques

Most service-oriented p2p networks involve a lifecycle phase of leeching. It occurs when the network client will only download content and not share it among others. While on initial stages of the transfer such behavior is unavoidable, clients are expected to share content further upon completion of first slices. However, some of them may not follow the rule; 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]. Protocols designed for p2p networks often facilitate various sophisticated algorithms to discourage leeching.

A good example of balancing technique 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 in order of time of their appearance in the queue. The foremost client in queue is served by the content piece and then rescheduled in the queue again, advancing other queue nodes. However, incoming client can advance queue member by more than single step in the queue, taking into account its contribution into incoming transfer of the same file. Therefore, as more content slices are provided by the incoming client, it progresses faster. This effectively places “bad” leechers to the end of queue and slows their advance. No such reward system is currently employed or planned by the BitTorrent client software restricted by the protocol specification. However, “private” torrent trackers usually deploy similar schemes.

Overview of tracker types

All existing BitTorrent trackers may be called as either “public” or “private”. Public tracker usually do not require invitation or registration to be able to download its advertised content, and so do not maintain download and upload rating records of its users.

On the contrary, private trackers, mostly based on TorrentPier or TorrentTrader 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 facilitate the sophisticated rating system, where rating is a value calculated using various formulas including overall download and overall upload amount of a particular user, time spent seeding etc. Users with low rating are restricted from further downloading. Users with high rating have certain privileges such as ability to download more torrents simultaneously, priority to access and search across tracker, etc.

In order to encourage content sharing and discourage leeching, tracker server must be aware of how much some particular BitTorrent client did download and upload to others. This is currently done by 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].

New scheme of p2p speed-up based on locality estimation

When trackers are starting to be very popular, commonly encountered overloading problems may arise. Although trackers themselves do not store any shared content and the storage of torrent-files require comparatively low resources, the "tracking" itself takes much up the processor speed and memory consumption due to intensive database usage.

No matter how efficient this solution might be, we believe that the expansive approach is not the only or optimal. As p2p technology develops rapidly, the traffic generated by its implementations is becoming more and more noticeable in overall Internet traffic, as mentioned above. Modern end-user connection technologies made high-speed Internet connections available to virtually every technically experienced customer. Despite this fact, the network latency still plays important role in p2p applications.

The reporting scheme concerning seeds availability is up to the vision of tracker software designers. Every tracker implements 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. Advanced methods involving calculations on 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.

We propose the locality metric to leverage load balance between network nodes consisting p2p swarm. It is commonly encountered phenomenon whereas a network packet designated to neighboring building may travel slower than the packet designated to another country, thereby the understanding of the relative logical position of network nodes or even building common points of presence may help packet travel faster. Implementation of national or statewide traffic 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, keeping the maintenance cost low.

Consider the single published entity over BitTorrent network swarm, to which the newly interested client connects and requests. The tracker, which is generally unaware of the locality of new client relative to the existing peers in swarm, reports them 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.

It is therefore vital to provide each swarm node with the information relevant to its topological position with respect to the other nodes and traffic exchange points, so that intelligent software may decide to query "closer" nodes first.

We define such closeness as "locality class" which is a metric defined for two arbitrary nodes in the Internet and is supposed to be derived solely from their IP-addresses.

The viable solution for calculating the locality class would have to conform to certain requirements such as the following:

1) calculations have to be fast enough to return the results before the connection timeout occurs;

2) no service traffic is allowed at the time of calculation because for large peer-list that would jam the link even before actual data transfer begins;

3) vendor-independence in the sense that no information may be retrieved that required any paid subscription;

4) solution must be decentralized, to avoid a single point of failure on which the entire network is dependent.

The solution we propose [5] feature Regional Internet Registries as mandatory administrative entities that publish databases containing the definitions for IP ranges, associated autonomous systems and their sets (known as assets).

The reference implementation of the proposed solution is called CARMA which stands for Combined Affinity Reconnaissance Metric Architecture. The CARMA uses RIR databases for IP ranges, subranges, ASes and Assets to build an

unified model of the Internet. This model allows to differentiate a pair of nodes as belonging to one of eight locality classes, namely “subrange”, “range”, “as”, “asset”, “as-link”, “backbone”, “country”, “distant”.

Prominent features of CARMA, not found in any other solution, are: a) the ability to calculate locality class of an arbitrary nodes, not necessarily either one of them being local and b) no service traffic at the time of calculation which only takes a fraction of a second.

The demo interface to a CARMA is featured on fig.2.

By applying the metric used in CARMA the client software participating in any peer-to-peer network where the transfer of a large arrays of data is involved, may leverage the knowledge of relative topological affinity to its advantage, querying the nodes in the order of their locality class calculated against local IP number.

Using BTAPPS interface for the most popular BitTorrent client µTorrent we have implemented such scheme. Due to lack of any mechanism for direct manipulation of querying order, CARMA module is working by restricting the communication between local nodes and the remote nodes with certain locality class until all the connections to nodes with a lesser locality class are established.

The experiments involved repeated transfer of a large file over a constant speed link to a swarm with negligibly varied participant numbers. The resulting throughput gain is shown in fig.3.

Figure 2 – CARMA demo interface featuring two IP addresses with calculated locality class of “AS” and “backbone” respectively

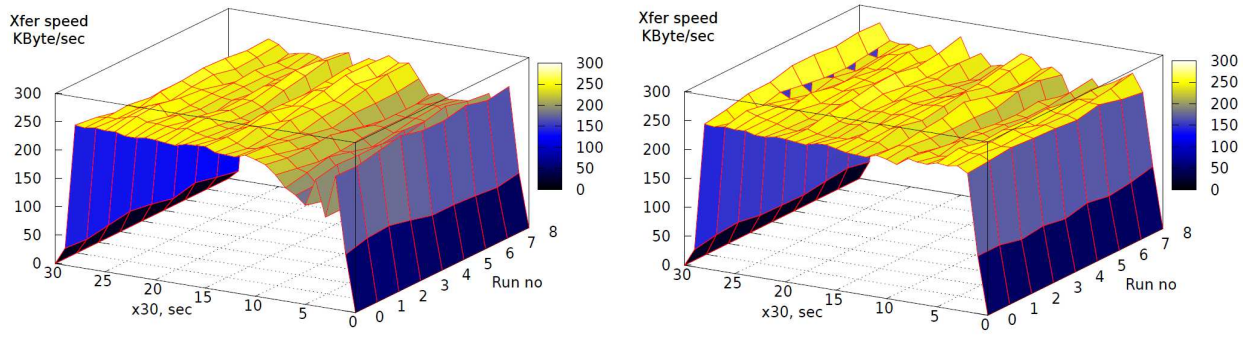

Figure 3 – Throughput gain when using locality estimation

It is apparent that the bandwidth saturation is reached in less than a minute when using CARMA module, contrary to 4 to 5 minutes with unaltered query order. Although the transfer speed did not increase as being limited by the physical link capacity, the average throughput was observed to be 2% more, which is a definitive and very promising result, giving the fact that the CARMA-based solution was deployed on a single node of the swarm.

In spite of a terabytes of p2p traffic circulating the average exchange points daily, 2% may well be worth the effort.

CONCLUSIONS

Facilitation of the locality-based algorithms for peer selection in BitTorrent trackers could potentially speed up the content distribution in swarms as well as with any other similar p2p technology, where clients are obliged to inquire many peer nodes 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.

The proposed solution based on locality class calculation does demonstrably increase the throughput in an average BitTorrent scenario by 2%.

1. Stephanos Androutsellis-Theotokis, Diomidis Spinellis. A Survey of Peer-to-Peer Content Distribution Technologies /Torrent/ 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. 5. Poryev G. V., Schloss H., Oechsle R. CARMA: A distance estimation method for internet nodes and its usage in P2P networks // International Journal on Advances in Telecommunications.— 2010.— Vol. 3, no. 3,4.— Pp. 114–128.

Поступила в редакцію 17.05.2012 р.

Рекомендував до друку докт. техн. наук, проф. Бурау Н. І.