A Framework for a Content Delivery Network Based on Peer-to-Peer

Xinjun Mo, Chris Todd
University College London

Abstract: In this paper an introduction of a modified Peer-to-Peer structure based on hybrid network is presented. In order to optimize the peer-to-peer system, a virtual entity is introduced into the peer-to-peer overlay. The evaluation is based on BitTorrent protocol as the protocol efficiently avoid the free rider problem and automatically search to make most contribution to the total download rate. Measurement is done and proper metrics is sought to evaluate the virtual peer state and virtual link state, from which the network performance can be judged.

1, Introduction

In the recent years the peer-to-peer (p2p) application traffic on the Internet already generates a new level of network abstraction. Although there is still debate in the precise definition about the peer-to-peer concept, the exchanges of information in these systems typically depend on the voluntary participation of peers. The p2p application generates a virtual overlay above the application layer. Different P2P network architectures have been developed and can be classified into the centralized and decentralized P2P structure. Decentralized p2p network is the “pure” p2p system which delegates the functions like searching, resource index storage, and resources allocation equally into all the members of the virtual network. In centralized architecture there exists a central component handling the statistics from all the participant members of the p2p network while the recourses exchange is done directly between peers. One of the most popular centralized P2P applications is BitTorrent. Decentralized infrastructure enjoys more advantage in scalability and robustness while centralized p2p networks has more efficiency in file searching and identification of the current p2p overly.

Various P2P applications from simple file sharing towards advanced stream delivery, more complex resource sharing and even the wireless and wired converging network management, requires much effort to introduce control and management into the existing P2P architecture so that they can cooperate to provide a useful service to the community of users. However, the advantage of the P2P network is the decentralized and autonomy characteristics so there lays an issue of the balance between these advantages and the introduction of the central control.

Some challenges in the P2P networks are illustrated here:
Reorganize: Logical rearranging of the P2P network topology to match the service provider physical network topology.
Route: Re-routing of P2P downloads to lowest-cost network paths.
Access Control: Re-identify the value of content: Copyright Issue and confidential requirement.

2, Methodology

2.1 BitTorrent Protocol

Based on the different structures there is a number of p2p applications based on different protocols. This paper considers the BitTorrent protocol for the file distributing process. BitTorrent protocol adopts a centralized p2p structure. Participating nodes are divided into two roles: trackers and peers.

Trackers are servers that collecting the statistics of the torrents, which include the certain resource and all the peers that share it. The tracker responds to HTTP GET requests and returns a list of peers that having the complete or part of the resource. The real resource distribution is based on the peer wired protocol, the peers using the peer list from the tracker to request different pieces of the file parallel, adoptingTit-for-Tat strategy to guarantee every peer contribute for the file distribution and avoid the “free rider problem”. Also the protocol specifies a mechanism that every peers searching for the best upload rate to improve the whole rate of file distribution.

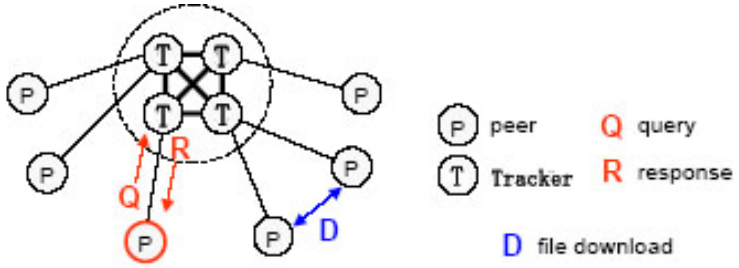
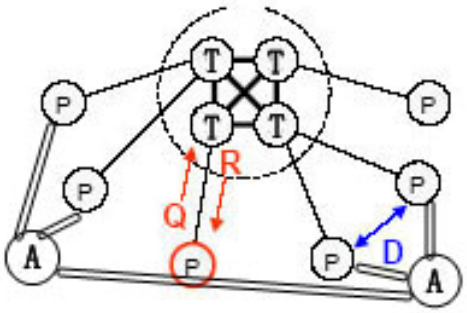


Figure 1 BitTorrent protocol network structure

2.2 Crawling the peer-to-peer network

In order to optimize a proposed peer-to-peer system, the participating peers’ characteristics must be understood and taken into account. For example, peers usually have different rate to connect to the internet, if some peers in a file-sharing system have low-bandwidth, high-latency bottleneck network connections to the Internet, delegating large or popular portions of the distributed pieces to those peers will obviously cause an overwhelming state, making that portion unavailable to other peers and lower the whole performance of the system. Similarly, considering the structure of P2P overlay is quite spontaneous as peers can join or leave the system at any time they want, the typical duration that peers choose to remain connected to the infrastructure has implications for the degree of redundancy necessary to keep data or index metadata highly available. In short, the appropriateness of a given peer for a specific task must be judged by the system before explicitly or implicitly allocating that task to the peer.



A is the virtual ad ministration Points whose function can be carried out by a servelet software running in any nodes

Figure 2 Revised BitTorrent network structure

Bandwidth (Virtual link state) & Peer capacity (Virtual peer state) are two key factors that determine the congestion in the p2p system. Both of virtual link and peer state are affected by hardware and software factors. Hardware factors include end terminal processor figures, routers’ capability, different access technologies like ADSL/ ISDN/ fiber… and Ethernet, X.25, etc in the data link layer. Link state influenced by software can be viewed as the bandwidth problem, like routing function not be optimized, resource allocation and peer topology changes quickly so link may not be stable (robust of the network). Peer state is usually affected by resource allocation and the physical location of the peers,

for example peers may be after a firewall and be limited in connection by NAT…. If congestion occurs from the user point of views there is a decline in QOS like influent media stream, low rate and long time consuming of downloading, etc. From the network operator point of view, traffic amount of the network exceed the capacity of the whole scope or part of the network.

Quantitated parameters should be derived from the combination statistics collected in all layers.

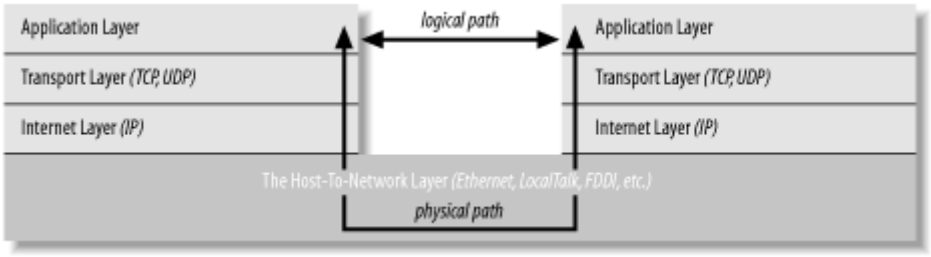


Figure 3 Internet connection model

In application layer, using the application logs from the server and the client (in the case of BitTorrent or any other hybrid P2P network) we can tell the peer state and kinds of message that the measurement are operating on. In transport layer, the characteristics in end-to-end latency, life-time, and bottleneck bandwidth can be chosen as the criteria to adjust the system topology and the connection between peers.

The tracker URL is announced in a .torrent file which is usually stored in a HTTP server. Because no direct access to mete-information maintained by the central BitTorrent servers is available, the only way that we could discover the set of peers participating in the system at any time is by parsing the announce tracker URL from the file, issuing queries to the tracker for files, and keeping a list of peers referenced in the queries' responses. The typical BitTorrent client uses the HTTP GET message which includes metrics from clients that help the tracker keep overall statistics about the torrent to start a downloading. The peer list in the response is length 50 by default selected randomly by the tracker. In order to discover the largest possible set of peers, we issued queries with the names of popular movies published in the web. We also use another form of request which is also supported by BitTorrent protocol called "Scrape". By convention most trackers support these command which queries the state of a given torrent (or all torrents) that the tracker is managing. Here is an example of the client request and tracker response:

Request:GET/scrape?info_hash=n%05hV%A9%BD%20%FC)%12%1Ap%D3%12%5D%E6U%0A%85%E1 HTTP/1.0

Response:d5:filesd20:n%05hV%A9%BD%20%FC)%12%1Ap%D3%12%5D%E6U%0A%85%E1d8:completei**5**e10:downloadedi**50**e10:incompletei**34**eeee

The response tells us there are 5 seeders, 34 incomplete, and 50complete downloads. By gathering this information we can tell the accuracy or coverage of the information gained in the former crawling step and determines the peer set to take into account. BitTorrent protocol supports announce tracker list which means the peer can query different trackers for peer list for the same content in parallel and this significantly enlarge the peer number that can be collected.

For each gathered peer population snapshot, direct measurements are done to collect additional properties of the peers. The goal is to capture data that would enable us to tell the virtual peer state and

virtual link state in the peer-to-peer file sharing system. However we must note the measurement is inevitably protocol based. Each protocol needs a specific crawler. The data collected includes the round trip latencies between peers and the measurement machine, the “lifetime” of the peers in the system, i.e., the frequency by which peers connect to the systems, and how long they remain connected.

The tracker response provides the list of peers’ IP-addresses, ports and unique peer-id. The round-trip latency between the peers and the measurement machines (our local PC) can then be calculated. The BitTorrent define the exchange of data between peers always starts from a handshake. For this, a simple transmission of a 80-byte TCP packet following the encode rule of BitTorrent is used to measures the Round-Trip time between a peer and our measurement host. TCP transfer has the famous congestion control feature which discriminates against flows with large round-trip times. This, together with the fact that the average size of files exchanged is in the range of 2-4 MB, makes latency a very important consideration when selecting amongst multiple peers sharing the same file.

Every peer in BitTorrent system connects with each other using a unique IP-address/port-number pair. There are three possible states for any peers that are listed in the tracker response. “Offline” means the peer is not connected to the internet or be blocked by a firewall or NAT problem occur. “Inactive” means the far-end terminal is connected to the internet but not listen to this p2p application port, while “active” means the peer is participating in the peer-to-peer system and involving in the file sharing. The former two state means the peer is disconnected from the p2p network. To determine which state a peer network sniffer “Ethereal” is used. First of all we send out a TCP SYN message to the destination peer. Checking the TCP header we can tell the peer is offline when no response is received. When the peer responses with a TCP RST message, it is in the inactive state. If the TCP 3-step handshake succeeds a TCP SYN/ACK packet will be received, from which we can tell the peer is in active state. However, the TCP header operation generates another traffic flow which occupies the network bandwidth since the measurement should be run for a continuous time so it might worth be considered to be combined with the latency measurement.

3. Conclusion

In this paper, a modified peer-to-peer system is presented seeking to adopt management and control into the spontaneous generated p2p topology. The network structure is based on the BitTorrent protocol and the measurement machine is “pretend” to perform like a common client side without modifying the server software and client software. Future works include the implication of the optimized protocol and find the proper size of the measurement packet so to reduce the bandwidth occupation.

Reference:

[1] BitTorrent *http://www.bittorrent.org*

[2] BitTorrent Specification *http://wiki.theory.org/BitTorrentSpecification?action=BackLinks*

[3] David Erman, Dragos Ilie, Adrian Popescu and Arne A. Nilsson “Measurement and Analysis of BitTorrent Signaling Traffic”

[4] Stefan Saroiu, P. Krishna Gummadi, Steven D. Gribble “A Measurement Study of Peer-to-Peer

File Sharing Systems” *Multimedia Computing and Networking (MMCN)*, San Jose, January, 2002